Universitat d'Alacant
Universidad de Alicante

Departamento de Lenguajes y Sistemas Informáticos
Escuela Politécnica Superior

Pattern Recognition for Music Notation

Jorge Calvo Zaragoza

Tesis presentada para aspirar al grado de

DOCTOR O DOCTORA POR LA UNIVERSIDAD DE ALICANTE
MENCIÓN DE DOCTOR O DOCTORA INTERNACIONAL
DOCTORADO EN APLICACIONES DE LA INFORMÁTICA

Dirigida por

Dr. Jose Oncina Carratala
Dr. Juan Ramón Rico Juan

Esta tesis ha sido financiada por el Ministerio de Educación, Cultura y Deporte a través del programa de ayudas para la Formación de Profesorado Universitario (Ref. AP2012-0939).

Para Bea, Juanita y K.T.;
por su cariño incondicional,
cada uno a su manera.

Agradecimientos

Me parece justo comenzar esta tesis mencionando a mis directores Jose Oncina y Juan Ramón Rico, del Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Alicante, cuya supervisión e ideas han hecho posible la investigación que se presenta en esta tesis.

Agradezco a mis compañeros del Grupo de Reconocimiento de Formas e Inteligencia Artificial que, de una u otra forma, han formado parte del buen desarrollo de esta tesis: Jose M. Iñesta, Luisa Micó, David Rizo, Antonio Pertusa y Plácido Román. También a aquellos compañeros de laboratorio que han contribuído a crear una situación agradable en el día a día, ya sea por su compañía durante los desayunos o por las reuniones de pasillo con las que amenizar el tiempo: José Bernabéu, Javi Gallego, Javi Sober, Pepe Verdú, Carlos Pérez y Miquel Esplà. No obstante, reservo una mención especial para José Javier Valero, el cual ha hecho que los largos días de trabajo fueran más que soportables.

Durante el tiempo que he estado realizando la tesis he pasado varios meses visitando a otros investigadores, a los cuales agradezco su hospitalidad y recibimiento: Isabel Barbancho, en el ATIC Research Group de la Universidad de Malaga; Colin de la Higuera en el Laboratoire Informatique de Nantes-Atlantique de la Universidad de Nantes; Andreas Rauber en el Institute of Software Technology and Interactive Systems de la Universidad Tecnológica de Viena; Enrique Vidal en el Instituto Tecnológico de Informática de la Universidad Politécnica de Valencia; e Ichiro Fujinaga en el Centre for Interdisciplinary Research in Music Media and Technology de la Universidad McGill.

Agradezco enormemente a mi padre Salvador todo lo que ha hecho por mi en esta vida y que, aun permaneciendo ajeno a todo lo que concierne a mi tesis, me ha apoyado como sólo un padre puede hacerlo, sea lo que sea que signifique eso.

Por último, un agradecimiento especial va para Beatriz, la persona con la que comparto mi vida, desde el cual me gustaría pedirle perdón por todas esas veces que ha hecho lo posible para que desconectara y no lo ha conseguido. Ella sabe que todo lo que he conseguido no hubiera sido posible sin su cariño, ánimo y apoyo constante.

Soporte

Quería aprovechar para mencionar a las entidades que me han proporcionado soporte económico para la realización de esta tesis: el Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Alicante, la Consejería de Educación de la Comunidad Valenciana mediante una plaza de Personal Investigador en Formación bajo el proyecto PROMETEO/2012/017; el Ministerio de Educación, Cultura y Deporte, tanto por su ayuda para la Formación de Profesorado Universitario (Ref. AP12-0939), como por las ayudas complementarias para estancias breves; y el Ministerio de Economía y Competitividad a través del proyecto TIMuL (financiado con fondos UE FEDER).

Jorge Calvo Zaragoza
Alicante, 6 de mayo de 2016

Síntesis en castellano

Introducción

La música constituye una de las principales herramientas para la transmisión cultural. Es por ello que, a lo largo de los siglos, numerosos documentos musicales se han preservado cuidadosamente en catedrales, bibliotecas o archivos históricos. No obstante, el acceso a estas fuentes no siempre es posible, pues su uso continuado podría comprometer su integridad. Esto implica que una importante parte de este patrimonio permanece alejado del estudio musicológico.

Desde hace años se ha invertido mucho esfuerzo en la transcripción de partituras a formato digital, ya que este proceso favorece la preservación de la música, así como su acceso, estudio y distribución. Para este propósito se han desarrollado muchas herramientas de distinta naturaleza. Por ejemplo, el uso de aplicaciones de edición de partituras está especialmente extendido. Éstas permiten crear partituras en formato digital a través de acciones con el ratón o el teclado. Otra posibilidad para transcribir partituras es utilizar instrumentos digitales (por ejemplo, un piano MIDI) que puedan ser conectados a un ordenador, de forma que la información musical se transfiera automáticamente a través de su interpretación. Desafortunadamente, este proceso no siempre puede captar todos los matices que se encuentran en una partitura.

Por otra parte, la digitalización masiva de documentos musicales abre diversas oportunidades para aplicar algoritmos de Extracción y Recuperación de Información Musical, que son de gran interés para el análisis musicológico. Independientemente del medio utilizado, la transcripción de partituras es un proceso que puede ser largo y tedioso —que a menudo requiere supervisión experta— por lo que el desarrollo de sistemas de transcripción automática ha adquirido importancia en los últimos años.

El Reconocimiento Óptico de Música (*Optical Music Recognition*, OMR) es la tecnología que proporciona a los ordenadores la capacidad de entender la información musical contenida en una partitura a partir del escaneo de su fuente. El proceso consiste, básicamente, en recibir una imagen de una partitura y exportar su contenido a algún tipo de formato estructurado como *MusicXML*, *MIDI* o *MEI*.

Hasta ahora, esta tarea ha sido enfocada desde un punto de vista de procesamiento de imagen. Sin embargo, representa un desafío similar al del Reconocimiento Óptico de Caracteres (*Optical Character Recognition*, OCR), que tradicionalmente ha sido tratado por la comunidad de Reconocimiento de Formas. La complejidad particular de la notación musical, no obstante, crea la necesidad de desarrollar algoritmos específicos.

Por otra parte, conviene tener en cuenta que las tecnologías actuales no permiten asegurar una transcripción libre de errores, y puede que nunca lo hagan. Es por ello que en los últimos años está surgiendo lo que se conoce como Reconocimiento de Formas Interactivo. Este paradigma está enfocado a la creación de sistemas de transcripción asistida por ordenador. En este caso, el usuario y la máquina colaboran para completar la tarea de reconocimiento con el mínimo gasto posible de recursos. El escenario más convencional asume que el ordenador propone soluciones a la tarea y el usuario tiene la responsabilidad de supervisar dicha salida. Si existe algún error, el usuario debe proporcionar retroalimentación a la máquina, que debe cambiar su respuesta teniendo en cuenta la nueva información recibida.

Este paradigma implica varios cambios con respecto al Reconocimiento de Formas tradicional:

- Comportamiento dinámico: las interacciones del usuario proveen información en línea relacionada con la tarea, lo que puede ayudar al sistema a variar su comportamiento. Por ejemplo, mediante el uso de nuevos datos etiquetados o propagando la corrección a otras partes de la hipótesis propuesta.

- Interacción con el sistema: es necesario invertir esfuerzo en que el usuario pueda utilizar una interfaz lo más ergonómica posible. Sin embargo, este tipo de interfaces pueden proceder de una señal no determinista, es decir, que a veces será necesario decodificar dicha interacción. Por lo tanto, el sistema tendrá que inferir, utilizando la nueva señal y la información inherente a la tarea, qué pretende comunicar el usuario. Esto abre la posibilidad a explotar la sinergia entre ambas modalidades de información.

- Medida de evaluación: como el esfuerzo del usuario, usualmente cuantificado como la cantidad de correcciones a realizar, se considera el recurso más importante, el objetivo del sistema no es tanto minimizar el número de errores sino el número de correcciones necesarias para completar la tarea. Esto puede provocar diferencias a la hora de elegir la hipótesis óptima.

Por todo lo expuesto anteriormente, esta tesis se centra en estudiar los aspectos del reconocimiento automático de notación musical que puedan ser enfocados desde una perspectiva de Reconocimiento de Formas, sin perder de vista el caso interactivo.

Objetivos

Desde una perspectiva global, el objetivo de esta tesis es explorar las posibilidades que puede ofrecer el Reconocimiento de Formas cuando se aplica al reconocimiento automático de notación musical.

En lo que atañe al propio campo de OMR, la idea es desarrollar nuevos algoritmos para algunas tareas específicas en las que queda margen de mejora para aplicar estrategias basadas en el Reconocimiento de Formas. Las tareas de OMR se suelen resolver siguiendo un cauce ampliamente establecido: binarización, borrado de líneas de pentagrama, detección de símbolos y clasificación; tradicionalmente, la implementación de estos pasos (salvo la clasificación) suele seguir enfoques basados en algoritmos de procesamiento de imagen.

A este respecto, en esta tesis queremos centrarnos en el paso encargado del borrado de líneas de pentagrama. A pesar de que estas líneas son necesarias para la interpretación humana, también complican la segmentación y clasificación automática de símbolos musicales. Aunque la detección y eliminación del pentagrama puede parecer una tarea sencilla, a menudo es difícil obtener resultados precisos. Esto se debe principalmente a irregularidades en la imagen tales como discontinuidades en las líneas o distorsión de la perspectiva, provocadas por la conservación del papel (especialmente en documentos antiguos) o el proceso de captación. Teniendo en cuenta que cuanto más preciso es este proceso, mejor es la detección de símbolos musicales, se ha llevado a cabo mucha investigación para mejorar este paso, que puede ser considerado hoy en día como un campo de estudio en sí mismo.

El hecho de que esta tesis haga especial hincapié en este proceso se debe a dos motivos principales. El primero, que las líneas de pentagrama tan sólo aparecen en documentos musicales, por lo que es un tema específico que no ha sido trabajado por otros campos relacionados con el análisis de documentos; en segundo lugar, que se estima que una gran cantidad de errores en posteriores etapas son causadas por fallos en este proceso (líneas de pentagrama no totalmente eliminadas o borrado en partes de símbolos musicales). Para esta tesis se plantearon dos objetivos relacionados con este proceso:

1. Investigar si es posible evitar el borrado de líneas de pentagrama. Aunque para notación moderna es complejo de asumir, esta estrategia parece factible en mucha notación musical antigua.

2. Investigar la resolución de este problema desde la perspectiva de Reconocimiento de Formas, afrontándolo como una tarea de clasificación supervisada. Nuestra hipótesis es que es posible, y quizá más provechoso, basar la bondad del proceso en algoritmos de aprendizaje automático.

Por otro lado, se pretende estudiar la interacción-humano máquina en tareas interactivas OMR. Como se ha comentado en la sección anterior, este paradigma está enfocado a la colaboración entre humano y máquina para resolver la tarea de la forma más eficiente posible. En nuestro caso, la idea es proveer al usuario encargado de supervisar la tarea de una interfaz ergonómica con la que trabajar. A pesar de los muchos esfuerzos invertidos en desarrollar editores de partitura cómodos para el usuario, la realidad es que la comunidad musicológica todavía prefiere trabajar de forma convencional con papel y lápiz. Utilizando un lápiz digital y una superficie electrónica, es posible desarrollar una interfaz que permita una interacción humano-máquina cómoda e intuitiva. El problema principal de este enfoque es que la interacción ya no es determinista, pues el sistema no puede estar seguro de qué es lo que el usuario está intentando comunicar; es decir, esta interacción tiene que ser decodificada, y esta decodificación puede contener errores.

Relacionado con este objetivo, esta tesis plantea estudiar el desarrollo de algoritmos de Reconocimiento de Formas que hagan que la máquina pueda entender interacciones recibidas a través de un lápiz digital. Típicamente, estas interacciones representarán símbolos musicales dibujados utilizando un lápiz digital. Nótese que esto puede ser utilizado tanto para interactuar con un sistema OMR como para proveer de un sistema de creación de partituras.

Otro de los procesos en los que más hincapié se quiere hacer en esta tesis es el de la propia clasificación de símbolos o trazos musicales, independientemente de si el origen es imagen o es un lápiz digital. En concreto, la regla del vecino más cercano (*Nearest Neighbour*, NN) representa una opción ideal desde un punto de vista interactivo por dos motivos principales: es naturalmente adaptativo, ya que la simple inclusión de nuevos prototipos en el conjunto de entrenamiento es suficiente (no es necesario volver a entrenar); si a través de este aprendizaje incremental, el conjunto de entrenamiento creciera demasiado, el tamaño podría ser controlado utilizando algoritmos de reducción basados en distancia.

Es por ello que esta tesis también se plantea ciertos objetivos relacionados con este tipo de clasificación. Por un lado, proponer mejoras en la clasificación de símbolos teniendo en cuenta algoritmos basados en un esquema NN. Adicionalmente, dado el carácter interactivo del campo de estudio de esta tesis, es importante que los clasificadores sean capaces de dar una respuesta rápida. Desafortunadamente, los clasificadores NN suelen ser computacionalmente ineficientes. Es por ello que también se plantea el objetivo de desarrollar esquemas que permitan utilizar este tipo de clasificadores de una forma más eficiente pero tratando, en la medida de lo posible, de no degradar su precisión.

Trabajos publicados

Los objetivos comentados anteriormente se han planteado desde una perspectiva general. Durante el transcurso de la tesis, no obstante, la investigación se ha ido matizando hacia aquellos aspectos que parecían más prometedores y más interesantes de tratar.

Dado que la tesis se defiende en la modalidad de compendio de publicaciones, los resultados troncales de la misma se encuentran reflejados en las distintas publicaciones en revistas o congresos de alto impacto que se han obtenido. A continuación se describe cada uno de ellas.

Publicación I

Referencia:

- Calvo-Zaragoza, J., Barbancho, I., Tardón, L. J., and Barbancho, A. M. (2015a). Avoiding staff removal stage in optical music recognition: application to scores written in white mensural notation. *Pattern Analysis and Applications*, 18(4):933–943

Esta investigación se llevó a cabo durante durante una estancia en la Universidad de Málaga. El trabajo presenta un sistema OMR para partituras escritas en notación mensural blanca del Archivo de la Catedral de Málaga. Estas partituras tienen un estilo de impresión específico que nos permite proponer un nuevo enfoque en el que se ha evitado la típica etapa de detección y borrado de líneas de pentagrama.

Dado que los archivos de la catedral deben ser cuidadosamente tratados, no se permite el escaneado de los mismos sino que las imágenes de entrada corresponden a fotografías tomadas desde una distancia fija. Por tanto, es necesario realizar una etapa de procesamiento previo con el fin de corregir tanto la rotación como la distorsión de la perspectiva de la entrada, de forma que el contenido quede alineado con el eje horizontal. En esta etapa también se aborda la binarización de la imagen de entrada por medio de diversos subprocesos (aumentar el contraste, compensar la iluminación y umbralizar), que será necesaria para los siguientes pasos.

La siguiente etapa comienza aislando cada sección de la partitura. Tras ello, seguimos una nueva estrategia para la detección de símbolos que no depende del borrado de líneas de pentagrama. Esta estrategia se basa en la combinación de un histograma vertical junto con un algoritmo de agrupamiento *k-means* para detectar los límites de cada región donde se encuentra un único símbolo. Con este procedimiento se logra una tasa de extracción superior al 96 %, demostrando ser suficientemente fiable para esta tarea.

Para clasificar cada uno de los símbolos, hacemos uso de un clasificador NN utilizando el operador de correlación cruzada normalizada como medida de disimilitud. Éste método obtiene unas tasas de clasificación superiores al 90 %.

Teniendo en cuenta los procesos de detección y clasificación, nuestro sistema transcribe los resultados con una precisión cercana al 90 %. En comparación con los resultados anteriores sobre este mismo archivo, nuestro trabajo mejora la detección de los símbolos, lo que demuestra que evitar la etapa de eliminación de pentagrama puede ser una opción muy interesante en estos términos. Además, la precisión de la clasificación también mejora, a pesar de mantener estas líneas.

Este trabajo abre nuevas vías para la construcción de sistemas OMR, demostrando que evitar el borrado de líneas de pentagrama merece consideración. Se ha comprobado que puede ser una manera de corregir algunos de los problemas de extracción y clasificación que se dan en los sistemas actuales.

Publicación II

Referencia:

- Calvo-Zaragoza, J., Micó, L., and Oncina, J. (2016a). Music staff removal with supervised pixel classification. *International Journal on Document Analysis and Recognition*, Online:1–9

En este trabajo presentamos un nuevo enfoque para la etapa de borrado de líneas de pentagrama. En la literatura, este proceso suele tratarse utilizando algoritmos de procesamiento de imagen basados en las principales características de los documentos musicales. Aunque la mayoría de los métodos propuestos son capaces de lograr un buen rendimiento en muchos casos, están lejos del óptimo cuando se cambia el estilo de la partitura. Nuestra intención es presentar un nuevo método que sea capaz de adaptarse a cualquier estilo de documento siempre y cuando se disponga de datos de aprendizaje adecuados.

En este contexto, nuestra estrategia consiste en modelar la tarea de borrar las líneas de pentagrama como si se tratara de un problema de clasificación de aprendizaje supervisado, en el que cada píxel de color se clasifica como *pentagrama* o *símbolo*, manteniendo tan sólo los últimos.

Dada una imagen binaria que representa una partitura musical, se propone recorre cada píxel con color y extraer un conjunto de características. Estas características se utilizan para entrenar un algoritmo de aprendizaje supervisado utilizando pares de partituras con y sin pentagrama.

En este trabajo las características de cada píxel de interés consisten en los valores de los píxeles vecinos, considerando una vecindad cuadrada de 3×3, 5×5, 7×7 o 9×9. Creemos que el entorno de cada píxel contiene suficiente información contextual para afrontar esta tarea con precisión. Además, esta información contextual puede ayudar a evitar errores de clasificación debidos al ruido o pequeñas deformaciones de la imagen.

Nuestros resultados experimentales muestran que el tamaño del conjunto de características es más relevante que el clasificador específico. En concreto, un clasificador de máquinas de vectores soporte (*Support Vector Machines*, SVM), teniendo en cuenta una vecindad de 9×9 (81 características), obtuvo los mejores resultados en promedio.

También se incluye una comparación con otros procesos de eliminación de líneas de pentagrama propuestos por otros investigadores. Nuestro método muestra un rendimiento muy competitivo, incluso logrando los mejores resultados en algunos casos a pesar de utilizar tan sólo una pequeña parte de la información de entrenamiento. También se lleva a cabo un experimento de prueba de concepto sobre documentos musicales antiguos, que demuestra la solidez de nuestra propuesta frente a otras opciones.

Por lo tanto, este nuevo enfoque reclama una mayor atención en este campo de investigación, ya que afrontar el proceso como una tarea de aprendizaje supervisado abre varias oportunidades para las cuales los métodos convencionales no son aplicables.

Publicación III

Referencia:

- Rico-Juan, J. R. and Calvo-Zaragoza, J. (2015). Improving classification using a confidence matrix based on weak classifiers applied to OCR. *Neurocomputing*, 151:1354–1361

Se propone una nueva representación para mejorar la clasificación de símbolos aislados de cualquier naturaleza. Esta representación se obtiene a partir de un conjunto de clasificadores débiles, de los cuales se obtiene la probabilidad *a posteriori* de que la entrada pertenezca a cada una de las categorías de la tarea. Este enfoque permite que las características iniciales sean transformadas a un nuevo espacio de meta-características, compactando su representación en una serie de valores más significativos.

La imagen de entrada se divide en sub-regiones, extrayendo de cada una características del símbolo, características del fondo y características del contorno. Para cada tipo de característica, se considera un clasificador débil distinto basado en NN, que mapea la entrada a un espacio de probabilidad. Los resultados de los clasificadores débiles se utilizan para crear una matriz de confianza, que es finalmente utilizada como conjunto de características para entrenar los clasificadores.

Nuestra experimentación demuestra que el uso de esta representación permite una mejora significativa en la precisión con respecto a utilizar el conjunto de características inicial. Estos resultados vienes avalados por una experimentación con cuatro bases de datos de símbolos ampliamente conocidas y el uso de pruebas de significancia estadística.

Publicación IV

Referencia:

- Calvo-Zaragoza, J. and Oncina, J. (2014). Recognition of pen-based music notation: The HOMUS dataset. In *22nd International Conference on Pattern Recognition, ICPR 2014, Stockholm, Sweden, August 24-28, 2014*, pages 3038–3043

Este artículo pretende ser un primer punto de referencia para el reconocimiento de la notación musical escrita a mano con un lápiz digital.

Este proceso se centra en el reconocimiento de símbolos musicales que se dibujan en una superficie digital utilizando un lápiz electrónico. De esta manera, se puede trabajar con notación musical digital sin recurrir a editores de partitura convencionales.

Se presentan algunos estudios previos que han trabajado en este tema. Sin embargo, todos ellos utilizan un conjunto de datos reducido y privado, por lo que todavía era necesario realizar experimentos comparativos que indicaran qué algoritmos son más adecuados para esta tarea.

Para resolver este problema, este trabajo presenta la base de datos *Handwritten Online Musical Symbols* (HOMUS). Este conjunto de datos contiene $15{,}200$ muestras de símbolos musicales a partir de 100 músicos expertos. Dentro de este conjunto se pueden encontrar 32 tipos diferentes de símbolos musicales. Se espera que el conjunto de datos proporcione muestras suficientes para que los resultados dependan de las técnicas utilizadas para la clasificación y no de la necesidad de más datos.

Cada muestra de la base de datos representa un símbolo musical aislado, que puede contener uno o varios trazos. Estos trazos — considerados como la forma dibujada entre los eventos *pen-up* y *pen-down* — producen un conjunto ordenado de puntos, que indican el camino seguido por el lápiz (modalidad *online*). No obstante, de cada símbolo se puede reconstruir una imagen de la forma dibujada, que también se puede utilizar para la clasificación como se haría en el reconocimiento a partir de imagen (modalidad *offline*). Esta modalidad da otra perspectiva del símbolo y podría ser más robusta frente a la velocidad del usuario, el orden seguido para dibujar un símbolo y el número de trazos usados.

Para establecer una primera línea base del reconocimiento de este tipo de datos, los experimentos se llevan a cabo con algoritmos de reconocimiento de formas ampliamente conocidos: para aprovechar la modalidad *online*, clasificadores NN y modelos ocultos de Markov (*Hidden Markov Models*, HMM); para clasificar las muestras de la modalidad *offline* se utilizan clasificadores NN, SVM, redes neuronales artificiales (*Artificial Neural Networks*) y HMM.

Se realizan dos experimentos para comprender mejor este conjunto de datos y extraer las primeras conclusiones sobre la clasificación de estos símbolos. El primer experimento consiste en medir la dificultad de reconocer un símbolo cuando proviene de un músico cuyo estilo no se ha visto durante el entrenamiento. En el segundo experimento, las muestras de cada músico se incluyen tanto en el conjunto de entrenamiento como en el de evaluación. Los resultados muestran que la dificultad principal se encuentra en el primer caso. Por otra parte, los algoritmos que aprovechan la naturaleza *online* de los datos han demostrado ser los más prometedores para la tarea de clasificación.

Publicación V

Referencia:

- Calvo-Zaragoza, J. and Oncina, J. (2015). Clustering of strokes from pen-based music notation: An experimental study. In *7th Iberian Conference Pattern Recognition and Image Analysis, IbPRIA 2015, Santiago de Compostela, Spain, June 17-19, 2015, Proceedings*, pages 633–640

Cuando se trata una tarea de reconocimiento de música basada en lápiz digital, la entrada consiste en una secuencia de trazos. A partir de un conjunto de datos de símbolos musicales, podemos obtener definiciones de cómo se construye cada símbolo a partir de trazos aislados. Si consideramos un etiquetado de trazos, se podría reducir el espacio de búsqueda de trazos mediante la asignación de la misma etiqueta a trazos similares.

En este punto, tenemos que lidiar con el problema abierto del conjunto de etiquetas o categorías a considerar para cada trazo aislado. Sería factible considerar un etiquetado *ad-hoc* pero no parece apropiado desde el punto de vista de la aplicación. Por tanto, proponemos un caso de estudio de etiquetado automático utilizando diferentes medidas de similitud entre trazos. El objetivo principal de este trabajo no es dar una única propuesta de etiquetado, sino medir la bondad y la generalización de cada medida de similitud considerada.

A este respecto, se consideran hasta 7 medidas de similitud. Algunas trabajan directamente con la secuencia ordenada de puntos en el plano 2D, mientras que otras son utilizadas tras pasar por un proceso de extracción de características.

Nuestro estudio experimental muestra que, aunque el proceso de agrupamiento es robusto cuando los símbolos provienen del mismo usuario, la tarea se vuelve compleja en el escenario en el que las muestras son diferentes estilos de escritura. En este caso, algunas medidas de similitud obtuvieron buenos resultados, mientras que otras, especialmente aquellas basadas en características extraídas de la imagen del trazo, se mostraron menos adecuadas para agrupar de forma compacta este tipo de información.

Publicación VI

Referencia:

- Calvo-Zaragoza, J., Valero-Mas, J. J., and Rico-Juan, J. R. (2015b). Improving kNN multi-label classification in Prototype Selection scenarios using class proposals. *Pattern Recognition*, 48(5):1608–1622

Con el fin de mejorar la eficiencia de un clasificador basado en la regla NN, han surgido una serie de técnicas que se centran en reducir el conjunto de entrenamiento. Uno de los enfoques más conocidos para este propósito es la selección de prototipos. La premisa principal de esta familia de algoritmos es que es posible mantener, o incluso mejorar, la precisión del clasificador teniendo en cuenta tan sólo un subconjunto de los datos de entrenamiento disponibles. Los criterios para seleccionar qué datos se mantienen dan lugar a diferentes algoritmos.

Desafortunadamente, en la mayoría de los casos, la reducción del conjunto de entrenamiento conlleva una pérdida de precisión en la clasificación. Para paliar esta situación, este trabajo propone una estrategia que tiene como objetivo buscar el equilibrio entre la precisión que se puede obtener con todo el conjunto de entrenamiento y la eficiencia que se puede alcanzar con algoritmos de selección de prototipos.

Nuestra estrategia reduce primero el conjunto de entrenamiento mediante el uso de un algoritmo de selección; la clasificación del nuevo elemento se realiza primero en ese conjunto reducido pero, en lugar de recuperar la clase más cercana, se propone un rango de clases de acuerdo a su similitud con la entrada; estas propuestas se utilizan para clasificar el elemento recibido en una versión filtrada de los datos de entrenamiento originales en los que solamente los elementos que pertenecen a las clases previamente seleccionadas se consideran para la tarea de clasificación.

Para comprobar la validez de nuestra estrategia, se lleva a cabo una experimentación exhaustiva, con múltiples bases de datos, diferentes escenarios y test de significancia estadísticos. Los resultados muestran que nuestra propuesta ofrece una nueva gama de soluciones equilibradas entre precisión y eficiencia. En los mejores casos, nuestra estrategia alcanza la precisión original utilizando tan sólo un 30 % del conjunto de entrenamiento. Además, en todos los casos considerados, las pruebas estadísticas revelaron que la precisión obtenida es significativamente mejor que la que se obtiene con los conjuntos reducidos.

Publicación VII

Referencia:

- Calvo-Zaragoza, J., Valero-Mas, J. J., and Rico-Juan, J. R. (2016b). Prototype Generation on Structural Data using Dissimilarity Space Representation. *Neural Computing and Applications*, Online:1–10

Dentro de las técnicas para reducir el tamaño del conjunto de entrenamiento podemos encontrar dos tipos de algoritmos: selección de prototipos, que selecciona los datos más representativos de los disponibles; y generación de prototipos, que se centran en la creación de nuevos datos que puedan representar la misma información que el conjunto original pero de forma más eficiente.

A pesar de que la generación de prototipos suele suponer una opción más eficiente, también es más restrictiva en su uso pues necesita que los datos estén representados por un conjunto de características numérico, siendo imposible de usar en datos estructurados como secuencias de símbolos de tamaño arbitrario, árboles o grafos.

En este trabajo proponemos el uso de lo que se conoce como *espacio de disimilitud*, que permite representar cualquier tipo de datos como vectores de características siempre y cuando se pueda establecer una medida de disimilitud. Utilizando este espacio es posible el uso de algoritmos de generación de prototipos en datos estructurados.

Dado que el uso de dicho espacio puede conllevar pérdida de representatividad, presentamos un estudio comparativo en el cual nuestra propuesta es enfrentada al uso de algoritmos de selección de prototipos sobre los datos originales.

Los resultados experimentales, avalados por el uso de varios conjuntos de datos y test de significancia estadística, muestran que la estrategia propuesta es capaz de obtener resultados significativamente similares que los obtenidos por la selección de prototipos. No obstante, utilizar un espacio de disimilitud presenta ventajas adicionales que refuerzan el uso de esta aproximación.

Trabajos no publicados

El trabajo sustancial de la presente tesis se encuentra en las publicaciones mencionadas anteriormente. Con carácter complementario, a continuación se describen trabajos realizados, pero pendientes de publicar, que completan algunos de los objetivos que se habían planteado, especialmente para el caso de la interacción basada en lápiz digital.

Trabajo I

- Título: *Recognition of Pen-based Music Notation with Finite-State Machines*

Este trabajo presenta un modelo estadístico para reconocer composiciones musicales basadas en lápiz digital utilizando algoritmos de reconocimiento de trazos y máquinas de estados finitos.

La secuencia de trazos recibida como entrada se transforma a una representación estocástica. Es decir, en lugar de asignarle una etiqueta concreta a cada trazo, se estima la probabilidad de que cada trazo sea cada una de las primitivas de trazo consideradas.

Esta representación se combina con un lenguaje formal que describe cada símbolo musical considerado en términos de secuencias de trazos (por ejemplo, el símbolo musical *negra* podría definirse por la secuencia de trazos *cabeza coloreada*, *plica*). Como resultado, se obtiene una máquina de estados probabilista que modela una distribución de probabilidad sobre todo el conjunto de secuencias musicales.

Con el objetivo de evitar secuencias incorrectas, este modelo se cruza con un lenguaje semántico que define todas aquellas que son gramaticalmente correctas.

Tras esto, obtenemos un modelo de estados que define una distribución de probabilidad sobre el conjunto de secuencias musicales gramaticalmente correctas. Con el fin de producir una hipótesis sobre la entrada recibida, se describen varias estrategias de decodificación de este tipo de máquinas.

Nuestra experimentación comprende varios algoritmos de reconocimiento de trazos, diversos estimadores de probabilidad, varias medidas de evaluación y diferentes escenarios. Los resultados muestran la bondad del modelo propuesto, obteniendo resultados competitivos en todos los casos considerados.

Trabajo II

- Título: *Pen-based Multimodal Interaction with Music Notation*

En este trabajo describimos una nueva forma de interacción humano-maquina para tareas de transcripción de notación musical. Este enfoque se basa en el uso de una interfaz de lápiz electrónico, donde se asume que el usuario va a calcar cada símbolo que el sistema haya obviado o clasificado incorrectamente. El sistema recibe, por tanto, una señal multi-modal: por un lado, la secuencia de coordenadas que indican la trayectoria seguida por el lápiz electrónico (modalidad *online*) y, por otro, la porción de partitura que subyace bajo el calco realizado (modalidad *offline*).

Hemos aplicado este enfoque a un pequeño repositorio de manuscritos de música española de entre los siglos XVI y XVIII en notación mensural blanca, visiblemente distinta de la notación moderna utilizada actualmente. De esta forma hemos obtenido $10{,}200$ muestras multi-modales, repartidas entre 30 tipos de símbolo.

El trabajo incluye experimentación con la base de datos recogida, considerando una clasificación que combina ambas modalidades. Se utiliza un clasificador NN que arroja una probabilidad por cada modalidad de que cada muestra pertenezca a cada uno de los posibles símbolos. Estas probabilidades son combinadas mediante una media ponderada en la cual se puede ajustar el peso otorgado a cada modalidad.

El análisis de estos experimentos revela que es provechoso utilizar ambas modalidades en el proceso de clasificación, ya que la precisión mejora notablemente con respecto a considerar cada modalidad por separado. En concreto, la mejor combinación encontrada obtiene alrededor de un 98 % de precisión mientras que se obtiene un 88 % y un 94 % para las modalidades individuales *offline* y *online*, respectivamente.

Conclusiones

Esta tesis doctoral estudia nuevos enfoques para el reconocimiento automático de notación musical basados en una perspectiva de Reconocimiento de Formas. El interés de la comunidad científica en el trabajo desarrollado se demuestra con las publicaciones en revistas y congresos de alto impacto, avalados por comités de revisión por pares. En concreto, indicar como indicios de calidad que 5 publicaciones están en revistas indexadas en el *Journal Citation Reports*, la mayoría de ellas situadas en los primeros cuartiles de impacto, y 2 han sido defendidas en congresos internacionales. Además, se han descrito otros trabajos que están en vías de publicación.

Como se demuestra en la diversidad de temáticas abordadas en los trabajos presentados, la tesis ha sido flexible en su línea principal, incorporando nuevas ideas surgidas en el transcurso de la propia investigación enmarcada dentro del reconocimiento automático enfocado en la notación musical. La investigación abarca diferentes partes del proceso como el borrado de líneas de pentagrama, nuevos enfoques para interactuar con el sistema y mejoras en la clasificación de símbolos, tanto en precisión como en eficacia.

El trabajo iniciado en esta tesis no puede considerarse como un camino finalizado, sino que precisamente ha sido la investigación llevada a cabo la que ha abierto nuevas vías que son interesantes para considerar en el futuro inmediato:

1. El enfoque de sistema OMR que evita borrar las líneas de pentagrama debe ser considerado para analizar otros tipos de partituras. Queda pendiente evaluar si esta estrategia puede establecerse definitivamente como una nueva alternativa para la construcción de estos sistemas o se reduce tan sólo a aquellas partituras que tengan un estilo de notación como el trabajado en esta tesis. Una cuestión a considerar es que la segmentación siga también un enfoque basado en aprendizaje automático, en lugar del uso de heurísticas.

2. Esta tesis ha demostrado que el borrado de líneas de pentagrama puede enfocarse como una tarea de clasificación supervisada. Siguiendo esta misma cuestión, sería interesante generalizar este proceso para que pueda utilizarse con imágenes en escala de grises, ahorrándose así los problemas inherentes al proceso de binarización. También debe dedicarse más investigación a superar el problema de conseguir datos suficientes para entrenar a los clasificadores cuando se recibe un nuevo estilo de partitura que no se había visto. Como idea preliminar, desarrollar sintéticamente un conjunto de entrenamiento más variado, que permita reconocer partituras de diferentes estilos.

3. Dado que el reconocimiento de notación musical basado en lápiz había sido poco explorado hasta el momento, el trabajo realizado durante esta tesis supone los primeros puntos de partida hacia esa dirección. No obstante, todavía queda mucho trabajo por realizar para explotar verdaderamente este tipo de información. Principalmente, incorporar esta modalidad en el flujo de trabajo de un sistema OMR funcional. Sería interesante comprobar cómo el propio sistema y el usuario colaboran para completar la tarea con el mínimo esfuerzo, haciendo que la interacción no sólo corrija errores producidos sino que ayude al sistema a modificar su comportamiento dinámicamente.

4. Hasta ahora la mayoría de sistemas OMR han seguido un cauce convencional basado en segmentación y clasificación. Como trabajo futuro, sería interesante analizar el rendimiento que obtienen algoritmos holísticos de reconocimiento de formas como HMM o redes neuronales recurrentes, que están dando buenos resultados en el reconocimiento automático de texto manuscrito.

Contents

III Unpublished works 99

IV Conclusion 133

Preface

Given that most of the research conducted as part of this thesis has been published in international peer-reviewed journals and conferences, this dissertation is configured as a *thesis by publication*. This means that the main part of the work is presented as reprints of such publications, keeping their original format.

The set of papers that form the PhD work are (in chronological order of publication):

1. Calvo-Zaragoza, J. and Oncina, J. (2014). Recognition of pen-based music notation: The HOMUS dataset. In *22nd International Conference on Pattern Recognition, ICPR 2014, Stockholm, Sweden, August 24-28, 2014*, pages 3038–3043

2. Rico-Juan, J. R. and Calvo-Zaragoza, J. (2015). Improving classification using a confidence matrix based on weak classifiers applied to OCR. *Neurocomputing*, 151:1354–1361

3. Calvo-Zaragoza, J., Barbancho, I., Tardón, L. J., and Barbancho, A. M. (2015a). Avoiding staff removal stage in optical music recognition: application to scores written in white mensural notation. *Pattern Analysis and Applications*, 18(4):933–943

4. Calvo-Zaragoza, J., Valero-Mas, J. J., and Rico-Juan, J. R. (2015b). Improving kNN multi-label classification in Prototype Selection scenarios using class proposals. *Pattern Recognition*, 48(5):1608–1622

5. Calvo-Zaragoza, J. and Oncina, J. (2015). Clustering of strokes from pen-based music notation: An experimental study. In *7th Iberian Conference Pattern Recognition and Image Analysis, IbPRIA 2015, Santiago de Compostela, Spain, June 17-19, 2015, Proceedings*, pages 633–640

6. Calvo-Zaragoza, J., Micó, L., and Oncina, J. (2016a). Music staff removal with supervised pixel classification. *International Journal on Document Analysis and Recognition*, Online:1–9

7. Calvo-Zaragoza, J., Valero-Mas, J. J., and Rico-Juan, J. R. (2016b). Prototype Generation on Structural Data using Dissimilarity Space Representation. *Neural Computing and Applications*, Online:1–10

Additionally, we also include some works that have not been published yet but contain substantial research that is strongly related to the objectives of the thesis.

Following the guidelines of the doctoral school of Universidad de Alicante for writing thesis as compilation of papers, the dissertation has to be organized as follows:

- **Part I: Introduction**. An initial section introducing the background of the thesis and a description of the set of contributions within the context of the thesis project.
- **Part II: Published work**. The compilation of papers that are already published or accepted for publication.
- **Part III: Unpublished work**. Finished works that are neither published nor accepted for publication yet.
- **Part IV: Conclusions**. Summary of the contributions, general conclusions and some lines about future research.

Taking into account that each presented paper is totally self-contained, Part I merely puts into context the research carried out without giving a deep insight into the background and related works. Similarly, the analysis of the results achieved can be found in each publication, so Part IV summarises the general discussion.

Part I

Preamble

Chapter 1

Introduction

A large number of music documents have been carefully preserved over the centuries. It is even a common practice nowadays, given the historical and cultural interest of these sources. The transcription of these documents allows a large-scale organization that would facilitate their access, search and study, while representing an indispensable element for their future maintenance. The problem, however, is that the transcription of scores is a long, tedious task —which often requires expert supervision— so the development of automatic transcription systems becomes an important need.

The automatic transcription of music documents has been approached so far from an image-processing point of view. However, it represents a challenge very similar to that of the Optical Character Recognition, which has been traditionally tackled by the Pattern Recognition community. Nevertheless, although part of the research carried out on characters could be of great utility in the context of music documents, the specific complexity of music notation forces the development of new algorithms and ideas.

On the other hand, it should be noted that ensuring error-free recognition systems (whichever the specific task) is not possible, and might never be. That is why a new paradigm, known as Interactive Pattern Recognition, is emerging in recent years. This paradigm is focused on the creation of computer-assisted transcription systems. It assumes a scenario in which users and machines collaborate to complete the recognition task efficiently. Therefore, many aspects ignored in the traditional scenario, such as the way humans interact with the machine or the ability of the system to adapt rapidly to the feedback, become of great interest in the interactive one.

Consequently, this dissertation focuses on the key aspects of automatic transcription of music documents that can be approached from a Pattern Recognition perspective, without losing sight of the interactive case.

Each contribution presented as a part of this thesis is a complete research work, thereby containing enough background information to be understood by themselves. Nevertheless, the following sections briefly introduce the different fields of research that are considered, given its interest for the present dissertation.

1.1 Pattern Recognition

Pattern Recognition is the field of computer science devoted to discovering patterns from data. It is assumed that there exists an unknown function that assigns a *category* or *class* to each sample. The goal is therefore to infer such a function from a set of representative examples.

Formally speaking, a Pattern Recognition task is defined by an input space $\mathcal{X}$, an output space $\mathcal{Y}$ and a function $\gamma : \mathcal{X} \rightarrow \mathcal{Y}$. The field can be broadly divided into two families of algorithms depending on how γ is estimated. If it is inferred by means of a set of labelled examples $T = \{(x_i, y_i) : x_i \in \mathcal{X}, y_i \in \mathcal{Y}\}_{i=0}^{|T|}$, the task is referred to as *supervised learning*. On the other hand, if the function has to be estimated from the data itself, it is called *unsupervised learning*.

Traditionally, a feature extraction process is performed, which consists of a function $f : \mathcal{X} \rightarrow \mathcal{F}$. This function maps the input space onto a feature space from which the task is expected to be solved more easily. Depending on the model used for representing the space $\mathcal{F}$, two fundamental approaches can be found: a first one, usually known as structural or syntactical, in which data is represented as symbolic data structures such as strings, trees or graphs; and a second one, known as statistical representation, in which the representation is based on numerical feature vectors that are expected to sufficiently describe the actual input. The election of one of these approaches has some noticeable implications and consequences: structural methods offer a wide range of powerful and flexible high-level representations, but only few algorithms and techniques are capable of processing them; statistical methods, in spite of being less flexible in terms of representation, depict a larger collection of Pattern Recognition techniques (Bunke and Riesen, 2012).

On the other hand, there are many Pattern Recognition tasks that are naturally sequential. For instance, Handwritten Text Recognition (Toselli et al., 2010) or Automatic Speech Recognition (O'Shaughnessy, 2000), for which the label to guess can be seen as a sequence rather than a single category. These tasks can be further modelled by an *alphabet* Σ, a finite non-empty set of symbols, and a *vocabulary* Ω. If some combinations of symbols are not acceptable, the vocabulary represents a subset of all possible sequences that can be formed with the alphabet ($\Omega \subset \Sigma^*$).

Furthermore, an input $x \in \mathcal{X}$ can be seen from different perspectives. On one hand, if considered as a whole, an output from Ω is directly inferred. These methods are also referred to as holistic or continuous models, for which systems based on Hidden Markov Models (Gales and Young, 2008) o Recurrent Neural Networks (Graves and Schmidhuber, 2009) are good representatives. On the other hand, the input can be approached as a sequence of smaller inputs, each of which has to be classified within the alphabet, as long as the final sequence belongs to the vocabulary. That is, the input is considered a sequence $x = (x_1, x_2, \ldots, x_n)$ and the estimated hypothesis $h = (h_1, h_2, \ldots, h_m)$ must accomplish that $h_i \in \Sigma$, $\forall 1 \leq i \leq m$ and $h \in \Omega$.

In the latter case, the Pattern Recognition task also entails a problem of segmentation, in order to decide how to divide each part of the raw input x into single units.

1.1.1 Interactive Pattern Recognition

It is widely known that current Pattern Recognition systems are far from being error-free. At least, that is the case in relevant fields like Automatic Speech Recognition (Graves et al., 2013), Handwritten Text Recognition (Yin et al., 2013) or Automatic Music Transcription (Benetos et al., 2013). If a high or full accuracy is a necessary issue, an expert supervisor is required to correct the mistakes. Traditionally, these corrections have been performed offline: the machine proposes a solution and the supervisor corrects the output off the system error by error. The Interactive Pattern Recognition (IPR) framework involves actively the user in the recognition process so as to reduce the effort needed in the previous scenario (Toselli et al., 2011).

A common IPR task is developed as follows (see Fig. 1.1):

1. An input is given to the system.

2. The system proposes a solution.

3. If some error is found, the user gives feedback to the system.

4. Taking into account the new information, the system proposes a new solution and returns to the previous step.

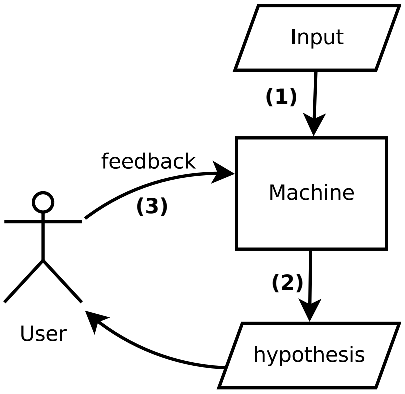


Figure 1.1: General scheme of an IPR task.

Note that the main goal of IPR is not to make the user learn how to perform such a task, which would be more related to fields like Interactive Learning (Lundvall, 2010), but to complete the work saving as much as possible the available resources.

Including a supervisor in the recognition process provides new ways to improve the efficacy of the system (Horvitz, 1999). For instance, corrections provide error-free parts of the solution, which can be helpful to be more accurate in the remaining ones. In addition, each interaction provides new context-related labelled data. Therefore, it might be interesting to consider instance-based classifiers which do not need to retrain the model when new data is available (Russell and Norvig, 2009).

However, the most important difference when dealing with an IPR task is the performance evaluation. Since the user is considered the most valuable resource, the performance of an IPR system must be related to the user effort needed to complete the task. This is commonly measured as the number of user interactions, regardless the nature of them (Vidal et al., 2007). In fact, if the number of user interactions is to be minimized, the optimum hypothesis changes with respect to conventional Pattern Recognition (Oncina, 2009)

Theoretically, this framework reduces the number of corrections that would be needed in a non-interactive scenario. Nevertheless, empirical studies with real users, such as those carried out under Transcriptorium (Romero and Sanchez, 2013) or CasMaCat (Sanchis-Trilles et al., 2014) projects, showed that the interactive approach may entail some drawbacks from users' point of view. For instance, if the human-computer interaction is not friendly enough or the user is not used to working in an interactive way, the time and/or effort needed to complete the task could even be worse than in the conventional post-editing scenario. As a consequence, some effort must be devoted to developing intuitive and ergonomic ways of performing the human-computer interaction.

1.2 Recognition of Music Notation

Digitizing music scores offers several advantages such as an easier distribution, organisation and retrieval of the music content. Since decades, much effort has been devoted to the development of tools for this purpose.

Nowadays, edition tools that allow actions based on *mouse and click* actions to place musical symbols in empty scores are available. Alas, its use is still tedious and very time-consuming. Moreover, digital instruments (such as MIDI keyboards) can also be found, from which the musical information can be directly transferred to the computer by playing the score. However, this mechanism cannot be completely accurate and capture all the nuances of the score. Furthermore, this method requires the user to be able to play the piece perfectly, which is not a trivial matter.

The emergence of Optical Music Recognition (OMR) (Bainbridge and Bell, 2001) systems represented a more comfortable alternative. Analogous to Optical Character Recognition with the case of text manuscripts, OMR is the field of computer science devoted to understanding a music score from the scanning of its source. The process basically consists in receiving an image of a music score and exporting its information to some kind of machine-readable format such as *MusicXML*, *MIDI* or *MEI*.

An OMR process usually entails a series of sub-procedures. A typical pipeline consists of the following steps:

1. Preprocessing. The preprocessing stage is focused on providing robustness to the system. If posterior stages always have as input an image with the staff lines aligned with respect to the horizontal axis, with equal relative sizes and where the only possible values for a pixel are background or foreground, systems tend to generalise more easily. Each of these steps can be addressed in different ways and each author chooses those techniques that are considered more appropriate in each case.

2. Staff lines removal. Although these lines are necessary for human readability, they complicate the detection and classification of musical symbols. Therefore, a common OMR system includes the detection and removal of staff lines. Next section goes deeper in this step, given its relationship with the research carried out in this dissertation.

3. Symbol detection and classification: symbol detection is performed by searching the remaining meaningful objects in the score after the removal of the staff lines. Once single pieces of the score have been isolated, an hypothesis about the type of each one is emitted in the classification stage. The main problem is that some of the musical symbols are broken by the earlier stages.

4. Post-processing: it entails a series of procedures that involve the reconstruction of music notation from symbol primitives and its transformation into some structured encoding.

For an extensive review of the state-of-the-art of all these steps, reader may check the comprehensive work published by Rebelo et al. (2012).

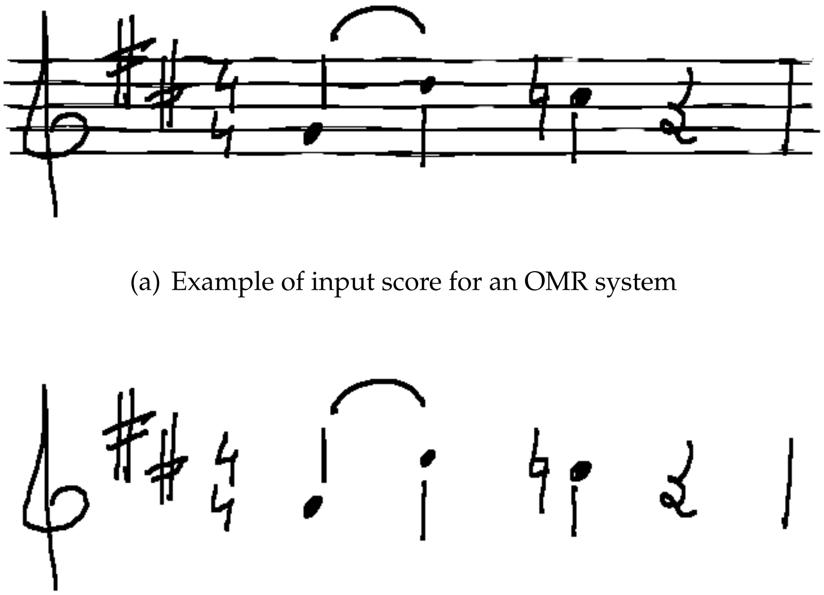
Yet, it should be stressed that the input of these systems can be quite varied. In addition to common notation, it is interesting to consider the automatic digitisation of any kind of old music manuscripts. This music is an important part of historical heritage, which is usually scattered across libraries, cathedrals and museums. Thereby making it difficult to access and study them appropriately. In order to analyse these documents without compromising their integrity, they should be digitised.

Nonetheless, conventional OMR systems are not effective transcribing these kind of music scores (Pinto et al., 2000). The quality of the sheet, the inkblots or the irregular levelling of the pages constitute some features to overcome. Moreover, it is extremely complex to build systems for any type of document because several notations can be found such as mensural, tablatures, neumes, and so on.

1.2.1 Staff detection and removal

OMR systems have to deal with many aspects of musical notation, one of which is the presence of the staff, the set of five parallel lines used to define the pitch of the notes. In fact, this stage is one of the most critical aspect of the OMR process since both the detection and the classification of musical symbol commonly relies on its accuracy.

It is important to note that this process should not only detect staff lines but also remove them in such a way that musical symbols remain intact (see Fig. 1.2).



(a) Example of input score for an OMR system

(b) Input score after staff removal

Figure 1.2: Example of a perfect staff removal process.

Problems mainly come from sheet deformations such as discontinuities, skewing or paper degradation —especially in ancient documents— or just a variation of the main features of the music sheet style (thickness, spacing or notation).

Given that, following conventional approaches, the more accurate this process the better the detection of musical symbols, much research has been devoted to this process, which can be considered nowadays as a task by itself (Dalitz et al., 2008). Although this stage has been approached in many ways, it finally becomes a trade-off between keeping information and reducing noise. Aggressive approaches greatly reduce the noise but can eliminate relevant information. On the contrary, less harmful processes end up producing a high amount of noisy areas.

1.2.2 Recognition of Pen-based Music Notation

Despite several efforts to develop light and friendly software for music score edition, many musicians still prefer pen and paper to deal with music notation.

On one hand, this is common during the composition of new music. Once the artistic process is over, however, they resort to this kind of tools to transcribe the musical content to some machine-readable format. Although this process is not always mandatory, it entails several benefits such as an easier storage, organization, distribution or reproduction of the music scores. A profitable way of solving the whole problem is by means of a pen-based music notation recognition system. Such systems makes use of an electronic pen, with which music symbols are drawn over a digital surface. The system collects user strokes and then processes them to recognize the music notation. As said before, this task can be considered very similar to the Optical Character Recognition task, for which pen-based (or online) research have been widely carried out (Plamondon and Srihari, 2000; Mondal et al., 2009; Liu et al., 2013).

On the other hand, such an interface could be used to amend errors made by OMR systems in a ergonomic way for the user, as has been proposed for automatic text recognition (Alabau et al., 2014). Handwriting is a natural way of communication for humans, and it is therefore interesting to use this kind of information as a mean of interaction with machines.

A straightforward approach to solve recognise pen-based music notation is to resort to OMR algorithms. That is, an image can be generated from pen strokes to make it pass through a conventional image-based system. Nevertheless, the performance of current such systems is far from optimal, especially in the case of handwritten notation (Rebelo et al., 2012).

Note that the main intention of a pen-based notation system is to provide musicians with an interface as friendly as possible. Therefore, they are expected to write without paying attention to achieving a perfect handwriting style so that notation would be even harder than usual to be recognised.

Fortunately, pen-based recognition brings new features that make the task be very different to the offline case. Therefore, it is interesting to move towards the development of specific pen-based algorithms.

1.3 The Nearest Neighbour classifier

The Nearest Neighbour rule (NN) is the most representative instance-based method for supervised classification. Most of its popularity in classification tasks comes from its conceptual simplicity and straightforward implementation. This method just require to work over a metric space, *i.e.*, that in which a distance between two instances can be defined. Thereby being independent of data representation used. More precisely, given an input x, the NN rule assigns to x the label of its nearest prototype of the training set. This rule can be easily extended to kNN, in which the decision is taken by querying its k-nearest prototypes of the training set.

An interesting advantage of this method is that it deals very well with interactive tasks. Due to its lack of model, the method does not need to perform any retune when new labelled data is considered. Therefore, the feedback received by the user within an interactive loop can be exploited rapidly. This increases the possibility of avoiding close errors related to the feedback received, thereby saving valuable user effort.

Additionally, this classifier is suitable for problems in which the set of possible labels contains more than two elements (*multi-class* classification). In this sense, the algorithm does not have to make any adjustment since it is naturally multiclass unlike other such as Support Vector Machines, which have to choose some kind of strategy to adapt to this scenario Hsu and Lin (2002).

On the other hand, this rule has some disadvantages related to its operation. For instance, it needs to examine all the training data each time a new element has to be classified due to the lack of model. As a consequence, it does not only depict considerable memory requirements in order to store all these data, which in some cases might be a very large number of elements, but also show a low computational efficiency as all training information must be checked at each classification task (Mitchell, 1997). Note that this is especially relevant for the interactive case, in which a stream of labelled data is expected to come through users' corrections.

These shortcomings have been widely analysed in the literature and several strategies have been proposed to tackle them. In general, they can be divided into three categories:

- **Fast Similarity Search:** family of methods which base its performance on the creation of search indexes for fast prototype query in the training set.

- **Approximated Similarity Search:** approaches which work on the premise of searching sufficiently similar prototypes to a given query in the training set instead of retrieving the exact nearest instance.

- **Prototype Reduction:** set of techniques devoted to lower the training set size while maintaining the classification accuracy.

While the two first approaches focus on improving time efficiency, they do not reduce memory consumption. Indeed, some of these techniques speed-up time response at the expense of increasing this factor. Therefore, when memory usage is an aspect to consider, the Prototype Reduction framework rises as a suitable option to consider.

1.3.1 Prototype Reduction

Prototype Reduction techniques are widely used in NN classification as a means of overcoming its previously commented drawbacks, being the two most common approaches Prototype Generation (PG) and Prototype Selection (PS) (Nanni and Lumini, 2011). Both methods focus on obtaining a smaller training set for lowering the computational requirements and removing ambiguous instances while keeping, if not increasing, the classification accuracy.

PS methods try to select the most profitable subset of the original training set. The idea is to reduce its size to lower the computational cost and remove noisy instances which might confuse the classifier. Given its importance, many different approaches have been proposed throughout the years to carry out this task. The reader may check the work of Garcia et al. (2012) for an extensive introduction to this topic and comprehensive experimental comparison of the different methods proposed.

On the other hand, PG methods are devoted to creating a new set of labelled prototypes that replace the initial training set. Under the reduction paradigm, this new set is expected to be smaller than the original one since the decision boundaries can be defined more efficiently. Reader is referred to the work of Triguero et al. (2012) to find a more comprehensive review about these methods, as well as a comparative experiment among different strategies.

Chapter 2

Contributions

This chapter broadly introduces the main contributions presented in this dissertation, and their relationships with the object of study.

It is important to emphasise that the main objective of this thesis is to explore the capabilities of Pattern Recognition strategies when dealing with the automatic recognition of music notation. Since this intention is rather general, we focus on specific tasks that can be approached from this perspective.

For the sake of clarity, the contributions are divided into three groups: general contributions to the OMR field, pen-based interaction for music notation, and improvements to the efficiency of the NN rule.

2.1 Optical Music Recognition

Among all the procedures involved in an OMR pipeline, this thesis pays special attention to the staff lines removal. This step is especially interesting for the purpose of this dissertation because staff lines represent a feature that only appears in music documents, so it is a specific stage that has not been addressed by other fields. Furthermore, it is estimated that a large number of errors in posterior stages are caused by inaccuracies during this process, *ie.* staff lines not entirely removed or removal of parts that belong to symbols.

In this respect, this work considers two questions related to this process:

1. Is it possible to avoid the staff removal stage in an OMR process?

   This question is addressed in Chapter 3 by proposing a new OMR system for printed Early notation that avoids the staff lines removal stage. This work was conducted during a research visit to Málaga (Spain), whose cathedral maintains an interesting music archive of printed Early music.

   The segmentation of symbols is done by means of an unsupervised learning analysis of the projection of the score over the x-axis. Once symbols are detected, classification is performed by using a common template matching method. Comparative results are provided, in which our strategy reports better figures in both detection and classification metrics than previous studies.

2. Is it possible to approach the staff removal stage from the supervised learning perspective?

   Our initial premise is that it might be possible to achieve accurate results as long as learning data is available. Chapter 4 presents a work in which the staff lines removal is solved as a classification problem at pixel level. Experiments show that the approach is very competitive, reaching state-of-art algorithms based on image processing procedures.

This thesis also deals with the recognition of isolated symbols. As aforementioned, we want to focus on classification based on the NN rule since it represents an ideal choice for the interactive case: it is naturally adaptive, since the mere inclusion of new prototypes in the training set is sufficient; and, if the training set grows too much due to the new labelled data received through user interactions, the size could be controlled by distance-based Prototype Reduction algorithms.

Chapter 5 describes a new ensemble method that takes into account different features from isolated symbols. These features are combined by means of weak classifiers based on the NN rule. Due to an editorial decision, the paper was presented as a method for classifying symbols of any nature. That is why that chapter includes additional results obtained for the case of music notation.

2.2 Pen-based Interaction for Music Notation

One of the specific objectives of the present dissertation is the study of human-computer interaction when dealing with music notation. The interactive framework is based on the collaboration between users and machines in order to complete the recognition task as efficiently as possible. In our case, we focus on providing a natural interface with which to work with music notation.

The premise is that it is possible to develop an ergonomic interface that allows an intuitive and comfortable interaction with the machine by means of an electronic pen (e-pen) and a digital surface. The main drawback of this interface is that the interaction is no longer deterministic, *ie.* the system cannot be sure what the user is trying to communicate. Therefore, this interaction has to be decoded and this decoding may contain errors.

Related to that problem, this thesis studies the development of Pattern Recognition algorithms that make the machine understand interactions received through an e-pen. These interactions might represent either isolated music symbols or complete music sequences.

Given that few research has been done over this issue, Chapter 6 presents a dataset of isolated symbols written using an e-pen. In addition, some baseline classification techniques are presented, considering both image and sequence features. Chapter 11 (unpublished work) extends this initial idea by considering a scenario in which the user traces the symbols over the music manuscript. This contribution shows that, taking into account the information provided by the user and the information contained in the score itself, it is possible to improve the classification results noticeably. Therefore, the interaction of the user is much better understood by the system, leading to a friendlier interaction with the machine.

On the other hand, Chapter 7 and 10 (unpublished work) develop the idea of using an e-pen and a digital surface to build a system in which a composer can write the music naturally and have it effortlessly digitised. Our proposal follows a learning-based approach and, therefore, it can be adapted to any kind of notation and handwriting style. The input of the system is the series of strokes written by the user. From that, Chapter 7 proposes an automatic labelling of these strokes, which is used as a seed to develop a complete system in Chapter 10. We show that the proposed approach is able to recognise accurately the music sequence written by the user.

2.3 Efficiency of the Nearest Neighbour rule

Given the interactive nature of the field of study in this dissertation, the Nearest Neighbour (NN) rule is a suitable classifier because of its intrinsic adaptiveness and dynamic behaviour. In turn, it is computationally inefficient, which can be harmful from a user-point of view: if the system takes too long to give any answer and the user has to wait too much, the interactive process loses all its sense.

Fortunately, one may resort to the use of Prototype Reduction techniques to develop schemes that allow using this kind of classifiers more efficiently. That is why this thesis devotes some effort to reducing the computational complexity of the NN when the task poses a multi-class problem, that is, when the set of possible labels is high (as it is in the case of music symbols classification). As introduced previously, among the techniques to reduce the size of the training set two types of algorithms can be found: Prototype Selection (PS), which selects the most representative data available; and Prototype Generation (PG), which focuses on creating new data that might represent the same information with fewer examples.

Alas, these techniques involve a loss of accuracy in most cases. To alleviate this situation, we propose in Chapter 8 a strategy that seeks for a trade-off between the accuracy obtained with the whole training set and the efficiency achieved with PS algorithms. In the best cases, our strategy achieves the original accuracy using only $30\ \%$ of the initial training set.

Moreover, although PG is often reported to be a more efficient option than PS, it is also more restrictive in its use because it needs data represented by a set of numerical features, being infeasible over structural data. Chapter 9 proposes the use of the so-called Dissimilarity Space (DS), which allows representing any type of data as feature vectors as long as a pairwise dissimilarity measure can be defined over the input space. Note that this is not a hard constraint under a NN scenario because such a dissimilarity function is necessary for the classification. Using this new space, it is possible to use PG algorithms over any kind of data. Experimental results show that the combined strategy DS/PG is able to obtain significantly similar results to those obtained by PS with the original data. However, the use of a DS representation provides additional advantages that enhance the use of this approach.

Part II

Published works

Chapter 3

Avoiding staff removal stage in Optical Music Recognition: application to scores written in white mensural notation

Calvo-Zaragoza, J., Barbancho, I., Tardón, L. J., and Barbancho, A. M. (2015a). Avoiding staff removal stage in optical music recognition: application to scores written in white mensural notation. *Pattern Analysis and Applications*, 18(4):933–943

SHORT PAPER

Avoiding staff removal stage in optical music recognition: application to scores written in white mensural notation

Jorge Calvo-Zaragoza · Isabel Barbancho ·
Lorenzo J. Tardón · Ana M. Barbancho

Received: 31 March 2014 / Accepted: 7 September 2014 / Published online: 21 September 2014

Abstract Staff detection and removal is one of the most important issues in optical music recognition (OMR) tasks since common approaches for symbol detection and classification are based on this process. Due to its complexity, staff detection and removal is often inaccurate, leading to a great number of errors in posterior stages. For this reason, a new approach that avoids this stage is proposed in this paper, which is expected to overcome these drawbacks. Our approach is put into practice in a case of study focused on scores written in white mensural notation. Symbol detection is performed by using the vertical projection of the staves. The cross-correlation operator for template matching is used at the classification stage. The goodness of our proposal is shown in an experiment in which our proposal attains an extraction rate of 96 % and a classification rate of 92 %, on average. The results found have reinforced the idea of pursuing a new research line in OMR systems without the need of the removal of staff lines.

Keywords Optical music recognition · Staff detection and removal · Ancient music · White mensural notation

1 Introduction

Since the emergence of computers, much effort has been devoted to digitizing music scores. This process facilitates music preservation as well as its storage, reproduction and distribution. Many tools have been developed for this purpose since the 1970s. One way of digitizing scores is to use electronic instruments (e.g., a MIDI piano) connected to the computer, so that the musical information is directly transfered. However, this process is not free of errors and inaccuracies could cause differences between the generated score and the original one. An additional bothersome feature of this method is that it requires the participation of experts who know how to perform the musical piece. On the other hand, software for creating and editing digital scores, in which musical symbols are placed in a staff based on 'drag and drop' actions, is also available. Nevertheless, the transcription of scores with this kind of tools is a very time-consuming task. This is why systems for automatic transcription of music scores became an important need.

Optical music recognition [1] (OMR) is the task of automatically extracting the musical information from an image of a score in order to export it to some digital format. A good review of OMR can be found in the work of Rebelo et al. [23], covering the state of the art and the remaining challenges.

In this work, we are interested in the process of recognition of musical symbols from ancient scores. Ancient music is a main source of historical heritage. This kind of music is scattered across libraries, cathedrals and museums, what makes it difficult to access and study them. In order to use these documents without compromising their integrity, they can be digitized. However, conventional OMR systems are not effective transcribing ancient music

J. Calvo-Zaragoza (✉)
Departamento de Lenguajes y Sistemas Informáticos,
Universidad de Alicante, Carretera San Vicente del Raspeig s/n,
03690, Alicante, Spain
e-mail: jcalvo@dlsi.ua.es

I. Barbancho · L. J. Tardón · A. M. Barbancho
Universidad de Málaga, ATIC Research Group, Andalucía Tech,
ETSI Telecomunicación, Campus de Teatinos s/n,
29071 Málaga, Spain

scores [18]. The quality of the sheet, the inkblots or the irregular leveling of the pages constitute some features to overcome. Moreover, it is extremely complex to build systems for any type of document because several notations can be found: mensural (white and black), tablature, neumes, etc. In the literature, some studies that have worked with some kinds of ancient scores can be found, such as those reported in [19] or [8].

The system described here focuses on analyzing ancient scores in white mensural notation. Specifically, our dataset consists of scores from the Archivo de la Catedral de Malaga (ACM). The ACM was created in the fifteenth century, and its library contains music scores from the tenth to the twentieth centuries. The scores of our dataset have a special feature: Unlike other ancient printed scores in which the printing house put the symbols over an empty staff, these symbols were printed jointly with a piece of staff over an empty sheet (see Fig. 1). It means that in each piece of the score, a single symbol is found. Furthermore, a noticeable distance between each musical symbol always exists. These features allow us to address the OMR process avoiding the common staff detection and removal stage.

Much research has been conducted in OMR concerning staff detection and removal [7, 25, 27]. This stage is one of the most critical aspects for both the detection and the classification of the musical symbols since they are based on symbol isolation. This stage is hardly sufficiently accurate, and it often produces noisy results. Although more aggressive methods that minimize noise can be used, they produce partial or total loss of some musical symbols. The trade-off between these two aspects, in addition to the accuracy of the techniques, has hitherto led to the inevitable production of extraction and classification errors [23]. Furthermore, this stage is usually very expensive in terms of time. For this reason, other authors decided to face OMR without the staff removal stage. In the work developed in [16], the whole score (including the staff) is thinned by a skeleton algorithm. The symbols are then detected seeking junctions and termination points. Pugin [22] also proposed a recognition scheme in which the score maintains the staff lines. His approach consisted in learning Hidden Markov Models based on low level.

Although these approaches are less common in the literature, we consider that this kind of procedure is an interesting option in different types of musical scores. Most of the current OMR systems are developed to handle contemporary notation, but same algorithms are performed later to early music, which is characterized by different types of scores. In this work, we propose an scheme that skips the staff removal stage. This approach is expected to help to reduce extraction and classification errors. Our aim is to show that this way of building OMR systems can be very effective for some music scores.

The type of scores selected from the ACM gives the possibility of detecting the musical symbols in a simple way. Since each symbol is on a different piece, there cannot be overlap. Therefore, in each piece of the score, there can be only one symbol. The extraction of the musical symbols only requires the detection of the portions of the staff in which each symbol begins and ends. Moreover, keeping the staff lines forces us to select appropriate techniques to classify the musical pieces of symbols. In this paper, a method based on template matching is proposed, since all the symbols to be detected come from a fixed font type due to the engraving mechanism. This approach has been successfully used for OMR tasks in some previous works [4, 30].

The remaining paper is structured in the same way as the recognition process (see Fig. 2): Sect. 2 details the preprocessing stage, Sect. 3 describes the score processing task, in which each staff of the score is isolated and each symbol is detected, and Sect. 4 presents the classification

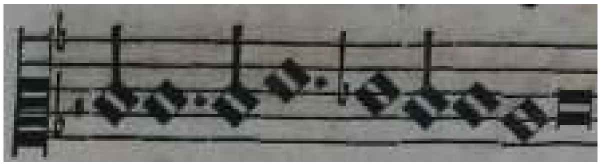


Fig. 1 Piece of staff in white mensural notation from the ACM. Each *musical symbol* is printed separately with its part of the staff

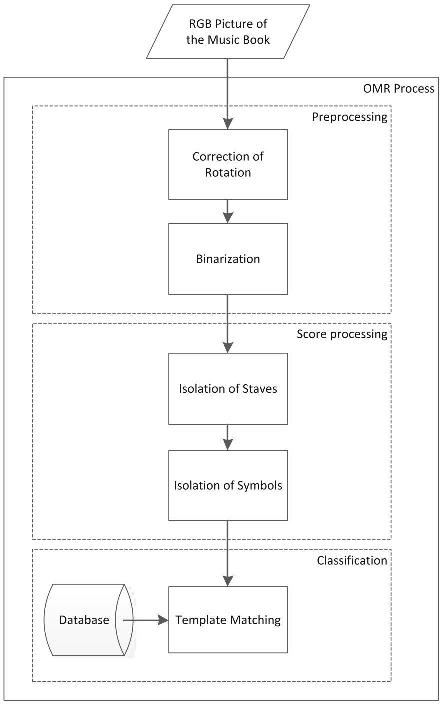


Fig. 2 General scheme of the recognition process

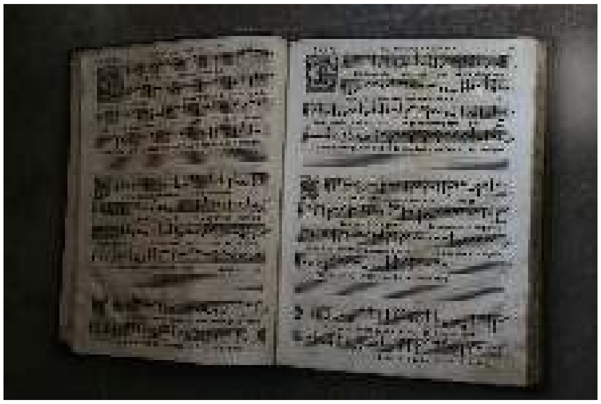


Fig. 3 Input image from the polyphony book 6 of the inventory of 1859 of the ACM (Francisco Guerrero, 1582)

step. Results are shown in Sect. 5, and some conclusions are drawn in Sect. 6. The steps to be performed after the recognition of symbols will not be addressed. An example of those processes for scores written in white mensural notation can be found in [29].

2 Preprocessing stage

In order to ensure the integrity of the documents, the images provided as input to the system correspond to pictures on polyphony books of the inventory of the ACM (Fig. 3), which consists of two pages each. A preprocessing of the image is a key step to perform the recognition task.

Often, the book appears rotated with respect to the image axes. Furthermore, the position of the book in the picture makes the perspective of the pages inconvenient. It is especially important to correct both the rotation and the perspective, so that the musical symbols can be detected and recognized correctly. Also, the background of the pages and ink is acquired with different color levels depending on their location due to the sheet conditions (irregular leveling, uneven lighting, paper degradation, etc.). Therefore, a binarization process that allows distinguishing accurately between the background and ink seems crucial for the performance of the system as well as for reducing the complexity of the recognition. These two steps are considered in the next subsections.

2.1 Correction of rotation

The process of transcription begins with the detection of the region of interest (ROI), which follows the same process as explained in [2]. The polygon that marks the boundaries of each page is found (Fig. 4). In addition to the separation of the pages, the vertexes of this polygon provide the key points to perform the correction of rotation.

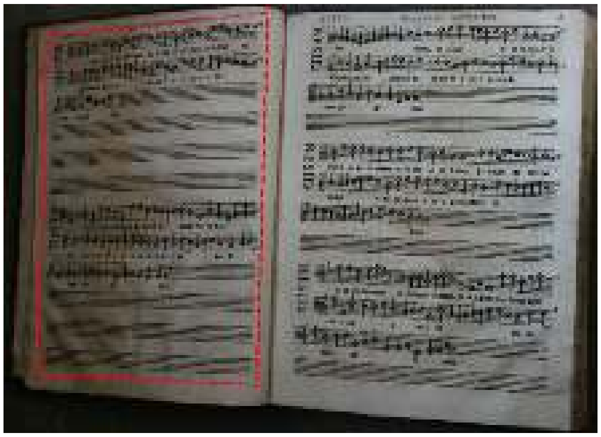


Fig. 4 Polygon over the ROI. The *polygon* identifies the boundaries of the page and provides the key points to correct the rotation

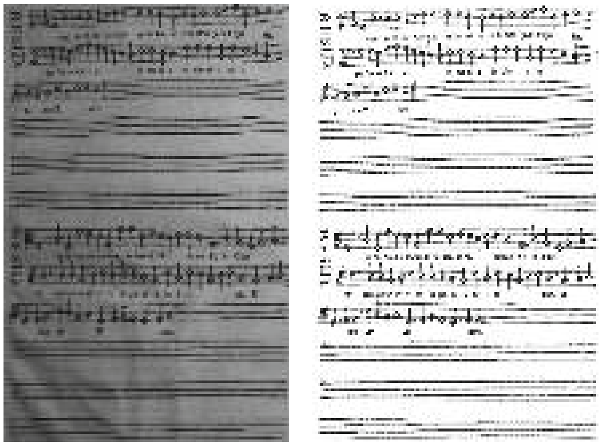
The objective of this step is to correct the rotation of the page. A perfect alignment with the image axes constitutes the starting point for the following stages since they are based on the horizontal and vertical histograms to detect the different parts of interest. In the case of these images, it is not sufficient to perform a simple rotation because the pages (their projection in the image) do not have the shape of a rectangle, but a trapezoid. Thus, the rotation is corrected by recovering the perspective distortion of the image with respect to the book pages.

In order to perform this rotation, we take the sides of the ROI polygon and split each pair into an equal number of segments to create a grid. Each pixel belonging to this grid is interpolated onto a rectangle. This process, when applied over a page of the input image, produces a result like the one shown in Fig. 5a. It can be observed that both the alignment with the image axes and the perspective are now adjusted successfully.

2.2 Binarization

The next step of the preprocessing stage is to binarize the image. We should be able to distinguish between meaningful pixels (music symbols and staves) and others (background, mold and noise). However, the binarization cannot be applied directly to the image with a typical adaptive method because of the presence of irregularities in the sheet. Hence, the binarization requires a more comprehensive process. The actions needed to better perform the binarization of these sheets are as follows:

- *RGB to grayscale conversion* The input images are in RGB color space. Since the relevant information of each pixel for our task relies only on its position and its intensity, the image is converted to grayscale by using a weighted average [10].



(a) Perspective-corrected image

(b) Perspective-corrected image binarized by using Otsu's method

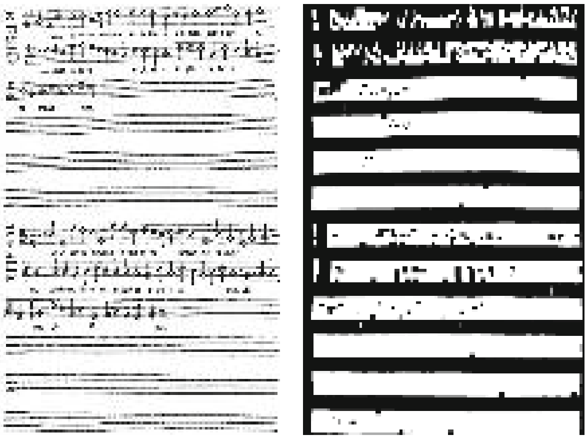
Fig. 5 Binarization of the perspective-corrected image

- *Contrast enhancement* In order to enhance the image, the contrast-limited adaptive histogram equalization (CLAHE) algorithm [20] is applied.
- *Illumination compensation* Since the illumination can vary largely among the set of images, the isolation of the reflectance—which keeps the meaningful information—is required. To this end, an aggressive symmetric Gaussian low-pass filter is used, so that an estimation of the illumination at each pixel can be obtained to correct the image. Preliminary experiments showed that a filter with size 80 and standard deviation 50 provided good results in the considered images. Nevertheless, results were not significantly different when using other similar parameters of the same order of magnitude.
- *Adaptive thresholding* An adaptive method is now needed to find the threshold that clusters the background pixels and the pixels with ink. At this stage, the Otsu's method [17]—which is reported as one of the fastest and most successful algorithms for this purpose [31]—is finally used to binarize the image.

An example of the result of the binarization process can be found in Fig. 5b.

3 Score processing

After the preprocessing stage, a binary image with perspective and rotation corrected is obtained. The next objective is to detect the musical symbols contained. As the scores are organized by staves, treating each staff separately is convenient. When the staves are isolated, the procedures for symbol detection can be performed more easily. In the next subsections, these two stages are described.



(a) Binary image of the page

(b) Mask over the staff regions

Fig. 6 Creation of a mask to detect staff regions

3.1 Isolation of staves

Staff detection consists in seeking the positions of five equally spaced parallel lines. The detection of the areas that contains these lines indicates the location of the staves. A common procedure is to compute the row histogram (or y-projection) of the image [28]. Staff features such as distance between staff lines, thickness of the staff lines and distance between staves are then computed from the histogram in order to isolate each staff. Alas, the presence in the scores of the ACM of other content such as lyrics or frontispieces among the staves complicates the process. Our approach handles this problem by creating a mask that keeps only the regions with horizontal lines. Unlike other works, we do not apply this mask to remove meaningless parts of the score but to directly isolate the staff parts on this mask.

First, an erosion over the binarized page is performed with a $1 - by - 20$ rectangular structuring element, which leads to the detection of parts with staves. A dilatation with a $20 - by - 1$ rectangular structuring element is then applied in order to span the entire space of the staff with the areas identified in the previous step. This way, a mask that indicates when a pixel is part of a staff region is estimated (Fig. 6).

It should be noted that by this mask, the extraction of staff features is not needed: staff splitting can now be performed with a row histogram analysis directly over the mask. Only a threshold is required in order to distinguish between rows with staff regions and rows with some remaining noise. Theoretically, each column of the

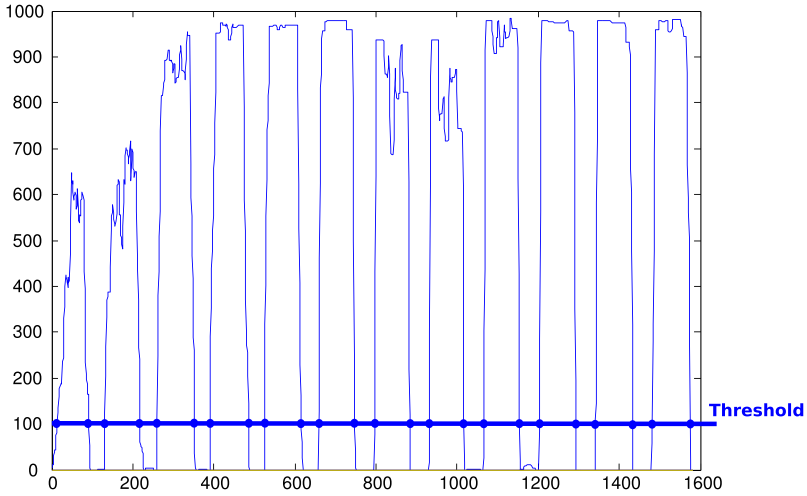


Fig. 7 Isolation of the staves. The intersection of the *threshold line* with the row histogram over the staff mask indicates the boundaries of each staff

histogram with a value higher than 1 should be considered part of a staff. Nevertheless, taking into account that previous steps are not error-free and staff parts get higher row-projection values, we decided to set a threshold which was a good margin with respect to the removal of noise and the detection of staff parts. Preliminary experiments established the threshold as 100 for the pages used in our experiments (1,600 × 1,000). This value achieved the best trade-off between noise removal and detection. Afterward, the intersection of the threshold line with the slopes of the histogram indicates where each staff is located in the original image (Fig. 7).

3.2 Isolation of symbols

After each staff has been isolated, the next goal is to detect the musical symbols contained. The common procedure at this point in typical OMR frameworks is the staff detection and removal. As aforementioned, we aim at exploring the possibilities of avoiding this step. The need of the removal of every part of the staff leads to delete some parts of the musical symbols, which produces unavoidable errors in posterior stages. Systems focused on contemporary scores need this process for the detection and classification of symbols. However, other scores—like the ones in our case—allow addressing the problem in a less aggressive manner and, eventually, less likely to delete important parts of the sheet. Thus, a novel approach for symbol detection and classification is presented.

Instead of staff removal and detection, we directly extract the column histogram of each staff obtained in the previous section. This histogram contains enough information to detect the musical symbols. Over this histogram, a k-means clustering [11], with $k = 3$, is applied to distinguish among the three column types considered: columns only with staff lines, columns with the head of a musical symbol and columns with the head of a musical symbol and its stem. Manhattan distance [5] is used in the clustering method instead of the Euclidean because it has proven to be more accurate for our system. After this process, the cluster with the lowest centroid—that corresponds to the areas without musical symbols—is removed. The histogram found is then used to partition the staff. This process is illustrated in Fig. 8.

3.2.1 *Special staff types*

The process explained so far performs well for common staves. However, there are two types of staff in the ACM scores that require some specific attention: staves with frontispiece (Fig. 9a) and half-filled staves (Fig. 10a). The special features of these staves distort the results of the clustering process and can lead to a poor segmentation. A slight preprocessing stage for these staves is required.

In the first case, in order to prevent parts of the frontispiece being treated as musical symbols, the beginning of the staff should be detected. The column histogram is used to detect the connected parts and keep only the widest one, which is expected to correspond to the staff (see Fig. 9).

In the case of half-filled staves, a correct clustering of the columns without symbols is difficult to perform because the number of such columns represents a very large percentage with respect to the total number of columns to analyze. The solution to this problem is to trim the image, so that the process is applied only to the parts that actually contain musical symbols. The detection of those parts is performed by means of a column histogram analysis. Starting from the left-hand side, it is checked whether

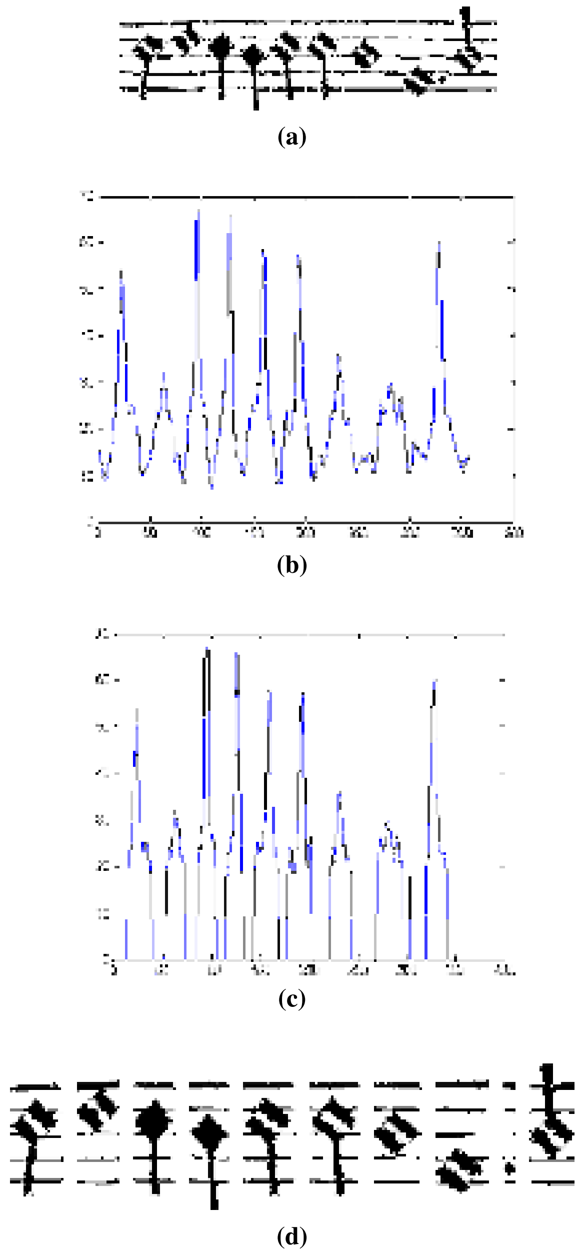


Fig. 8 Extraction of musical symbols from a piece of staff. **a** Piece of a musical staff. **b** Column histogram over the piece of the staff. **c** Column histogram without staff columns. **d** Example of the extraction of musical symbols by histogram analysis

the histogram stabilizes within a meaningful period. If this happens, it can be assumed that the rest of the staff is empty, so we trim the image at that point (see Fig. 10).

These two processes are applied to all the staves before the clustering process since they perform well regardless of the type of staff. It should be noted that only one vertical histogram is required to compute all the processes.

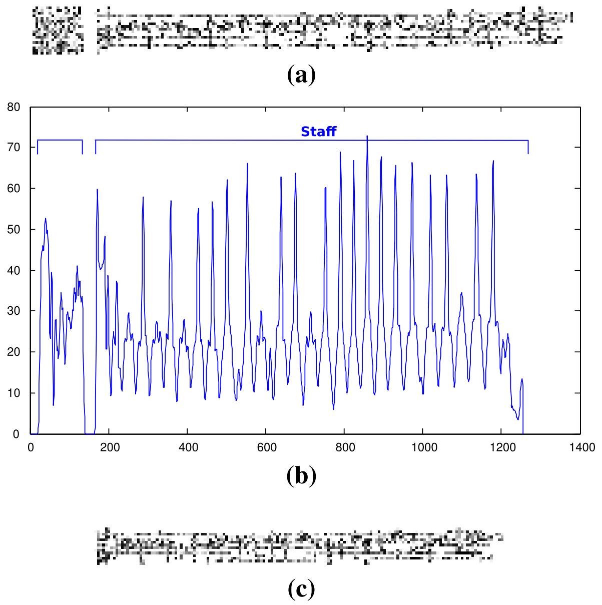


Fig. 9 Preprocessing of a staff with frontispiece. **a** Staff with frontispiece. **b** Column histogram over isolated staff: detection of the staff region. **c** Staff without frontispiece

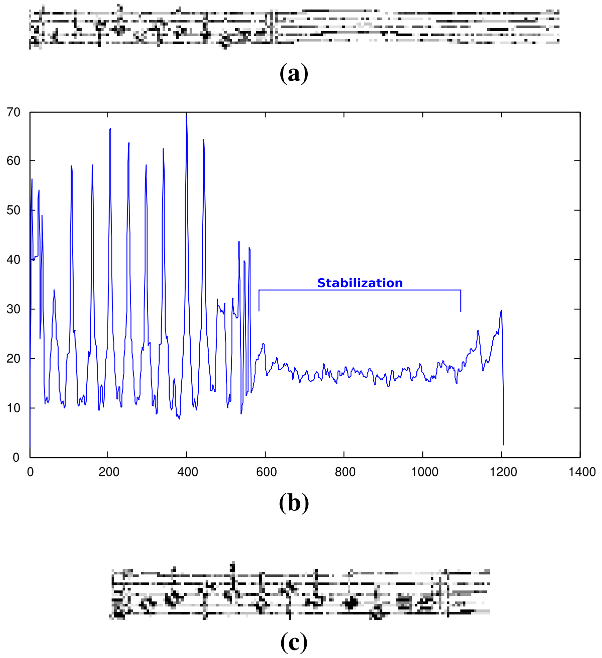


Fig. 10 Preprocessing of a half-filled staff. **a** Half-filled staff. **b** Column histogram over isolated staff: detection of the part without musical symbols. **c** Staff without the empty part

4 Classification

The output of the previous section is a set of ordered images containing a single musical symbol. The classification stage aims at labeling each of these images with the symbol contained in it. Typical OMR systems rely on feature extraction to classify the symbols. These features are then used to construct a set of samples to perform pattern recognition methods. Image feature extraction for recognition can be based on several techniques: Fourier descriptors [33], angular radial transform (ART) moments [12], chain codes such as Freeman's (FCCE) [9] or Vertex Chain Code (VCC) [3], etc. Unfortunately, these methods cannot be applied to these images as the presence of staff lines would represent an ineluctable obstacle. A classification method whose performance does not get severely damaged by the presence of the staff lines is required. This is the reason that led us to use the cross-correlation.

Cross-correlation [6] is a common method for template matching [24, 32]. Let $f(x, y)$ be an image and $w(x, y)$ be a template, the cross-correlation can be computed with the following equation:

$$\gamma(u, v) = \frac{\sum_{x,y} [f(x, y) - \overline{w}_{u,v}][t(x - u, y - v) - \overline{w}]}{\sqrt[2]{\sum_{x,y} f(x, y) - \overline{f}_{u,v}]^2 [w(x - u, y - v) - \overline{w}]^2}} \tag{1}$$

where $\overline{f}_{u,v}$ is the mean of $f(x, y)$ in the region under the template and $\overline{w}$ is the mean of the template. Equation (1) is commonly referred to as normalized cross-correlation [26]. The result of the normalized cross-correlation gives a value between -1 and $+1$ related to the presence of the template at each point of the image. In this work, a fast version of the normalized cross-correlation [15] is used.

It should be noted that the cross-correlation matrix can give high values despite being different symbols as long as some piece of the image looks like the template. Fortunately, it is known that if there is a very high value in the center of the matrix, the probability of being the same symbol is very high. This is because all symbol images in our dataset contain the symbols centered horizontally. Thus, we establish that the correlation values of interest are those that are well centered horizontally. We assume that if the cross-correlation attains its maximum value close to the vertical edges, it should be considered a misclassification. Hence, the classification process is governed by a range $R = (x_s, x_e)$, normalized with respect to the width of the image ($x_s, x_e \in [0, 1]$), that indicates which cells of the cross-correlation matrix must be taken into account for the classification.

Let s represent the $N \times M$ image of a symbol, W stands for the dataset of labeled symbols, $L(w)$ represents the label of a template w; let $M(m)$ denote the maximum value of a matrix m, let $[m]_{a:b,c:d}$ represent the sub-matrix of m formed by rows $a, \ldots, b$ and columns $c, \ldots, d$; and let $R = (r_1, r_2)$ denote a specific range, with $r_1, r_2 \in [0, 1]$; the label τ of s (τ_s) is determined by the following equation:

$$\tau_s = L\left(\arg\max_{w \in W} M\left([\gamma(s, w)]_{[Nr_1:Nr_2,1:M]}\right)\right) \tag{2}$$

In Eq. (2), the normalized cross-correlation between the extracted symbol and each labeled template in the database is applied. The template that achieves the best cross-correlation value within the width range R is used to label the symbol. It should be clear that, with this method, we can determine both the type and the pitch of the symbol as long as the labels in the database keep this information.

5 Experiments

In this section, some experiments are carried out to assess the accuracy of the proposed strategies. Our dataset is composed of 12 pictures, with two pages each one. The average number of staves in each page is 12. Over the entire dataset, 5,768 symbols are to be extracted and classified. The parameters involved in the process are as follows: the total number of musical symbols in the scores (T), the number of extracted symbols (E) and the number of correctly classified symbols (C). It should be noted that E can be divided into the number of musical symbols extracted (S_e) and the number of noise images extracted (N_e). All the symbols that either contain no musical information (e.g., parts of the frontispiece) or are partially (wrongly) extracted are considered as noise. Similarly, C can be divided into the number of correctly classified musical symbols (S_c) and the number of noisy symbols detected (N_c)—noise images classified as noise.

Since the extraction and the classification are two different processes that can be evaluated separately, an evaluation for each process is performed. A global evaluation of the system, involving both the extraction and the classification, is also included.

5.1 Evaluation of the extraction process

A good performance of the symbol extraction stage is the first requirement to perform a good transcription. The extraction process is related to the number of musical symbols correctly extracted as well as to the number of symbols lost or partially (wrongly) extracted. In order to assess this process, we use the extraction rate. This parameter can be calculated as the number of musical symbols that have been found during the segmentation process divided by the total number of musical symbols in the score:

Table 1 Performance results of the extraction process over the dataset

Fold	T	S_e	N_e	R_{ext} (%)	R_{noise} (%)
1	390	371	3	95.13	0.80
2	377	361	7	95.76	1.90
3	623	598	5	95.99	0.83
4	432	421	10	97.45	2.32
5	410	399	2	97.32	0.50
6	427	414	8	96.96	1.90
7	514	498	7	96.89	1.39
8	436	425	6	97.48	1.39
9	441	433	3	98.19	0.69
10	444	432	5	97.30	1.14
11	633	598	9	94.47	1.48
12	641	601	7	93.76	1.15
Whole	5,768	5,551	72	96.24	1.28

The table contains information about the number of musical symbols in each fold (T), the number of musical symbols extracted (S_e) and the number of noise images extracted (N_e), which are used to calculate the extraction rate (R_{ext}) and the noise rate (R_{noise})

Table 2 Classification rate over the dataset with a 12-fold cross-validation scheme

Classification rate ($R_{\text{classification}}$)						
Fold	E	Range $R = (r_1, r_2)$				
		(0, 1)	(0.1, 0.9)	(0.2, 0.8)	(0.3, 0.7)	(0.4, 0.6)
1	374	92.25	93.04	92.78	93.85	93.58
2	368	88.86	89.40	91.30	91.30	91.84
3	603	88.22	88.39	88.55	88.72	88.05
4	431	91.18	91.87	92.34	93.03	93.27
5	401	91.52	91.52	93.01	93.26	93.76
6	422	91.23	90.75	92.18	92.41	92.65
7	505	89.30	89.50	91.28	91.48	90.89
8	431	89.32	89.79	91.41	91.41	91.18
9	436	92.66	92.88	92.88	93.11	93.80
10	437	88.55	88.55	91.99	92.67	93.13
11	607	89.12	88.96	89.45	89.45	89.29
12	608	90.29	90.78	90.78	91.44	91.11
Whole	5,623	90.09	90.39	91.30	91.64	91.62

Different ranges R for the cross-correlation are presented

$$R_{\text{ext}} = \frac{S_e}{T} \tag{3}$$

Moreover, it is also important to quantify the noise introduced during the segmentation. The amount of noise can be evaluated by using the noise rate, based on the number of noise images extracted (N_e) and the total number of symbols extracted from the scores (E):

$$R_{\text{noise}} = \frac{N_e}{E} = \frac{N_e}{S_e + N_e} \tag{4}$$

Table 1 shows the extraction performance over our set of images. These results show that our extraction stage is able to achieve a rate over a 95 %, on average. All cases exceed a 93 %, even some of them are over 97 %. Moreover, the noise rate is low in almost all the cases, which means that our strategy accurately distinguishes between musical symbols and other objects of the scores. These values show the good performance of our symbol detection strategy.

Further analysis of these results revealed that the musical symbol dot is the most commonly missed symbol. The small width of the symbol makes it difficult to be detected. Changing the detection parameters, so that this symbol gets detected more accurately led to a larger noise rate. We consider that it is preferable to accept some dot misses rather than generate a more noisy output which may deteriorate the whole transcription process.

5.2 Evaluation of the classification process

The evaluation of the classification process aims at measuring the goodness of the method used to determine the type of the symbols found. As indicated in Sect. 4, the cross-correlation operator for template matching was chosen. In our system, we evaluate the accuracy of the classification strategy regardless of the type of symbols detected or the type of error made, so, in order to evaluate the performance, we use the common $0 - 1$ loss function. This function is able to measure the rate of misclassified symbols if a uniform weight for each symbol is established. Thus, the classification rate can be defined as the number of correctly classified symbols divided by the number of symbols extracted:

$$R_{\text{classification}} = \frac{C}{E} = \frac{S_c + N_c}{S_e + R_e} \tag{5}$$

The classification experiment is conducted by using a k-fold cross-validation scheme. Each fold is composed of one of the images of the dataset, while the labeled symbols of the rest of the folds are used as database for the cross-correlation operator. The results for each fold are shown in Table 2. A set of possible values for the range $R = (r_1, r_2)$ (Eq. 2) is confronted experimentally.

The results show that the classification rate obtained with the cross-correlation is larger than 90 % in all the cases considered. Also, it has been shown that the best range to use for the cross-correlation is between 30 and 70 % of the total width of the image, which yields a classification rate of 91.64 %, on average. However, it should be emphasized that the results among the different alternatives are not particularly remarkable, which is indicative of the robustness of the cross-correlation operator with respect to this parameter.

Table 3 Global results of the OMR systems over the dataset

Fold	T	N_e	C	W_{Acc} (%)
1	390	3	351	93.04
2	377	7	336	89.40
3	623	5	535	87.89
4	432	10	401	90.71
5	410	2	374	92.76
6	427	8	390	90.52
7	514	7	462	90.09
8	436	6	394	90.02
9	441	3	406	92.43
10	444	5	405	91.53
11	633	9	543	87.97
12	641	7	556	90.29
Whole	5,768	72	5,153	90.36

The table contains information about the number of musical symbols in each fold (T), the number of noisy images extracted (N_e) and the number of correct classifications (C). These parameters are used to calculate the word accuracy (W_{Acc})

5.3 Global evaluation

In the previous subsections, the extraction strategy and the classification strategy were evaluated. However, the OMR system has to be globally evaluated by involving both the extraction and the classification stages. In order to assess its performance, we use the well-known Word Error Rate (WER) [13].

The WER is based on the edit distance [14] and measures the difference between two sequences (in our case, two sequences of musical symbols). As the focus of OMR systems is to assist the human task, this metric can provide an estimation of the human effort needed to correct the output of the system. It involves the three common edit operations, which in this case are defined as follows:

- *Insertions* The difference between the number of musical symbols in the score and the number of extracted symbols ($T - S_e$).
- *Substitutions* The difference between the number of extracted symbols and the number of symbols correctly classified ($S_e - S_c$).
- *Deletions* The difference between the number of noise image extracted and the number of noise correctly classified ($N_e - N_c$).

Therefore, the WER can be calculated by summing up these three values and dividing it by the total number of musical symbols:

$$\begin{aligned} \mathrm{WER} &= \frac{(T - S_e) + (S_e - S_c) + (N_e - N_c)}{T} \\ &= \frac{T + N_e - C}{T} \end{aligned} \qquad (6)$$

Table 4 Comparison against previous work with scores from the ACM with average (%) results obtained in the recognition processes

	Extraction	Classification
Our results	**96.24**	**91.64**
Previous [29]	72.78	88.86

Bold values represent the best results, on average

The final results of applying our OMR process over the dataset with the best classification parameter selected [$R = (0.3, 0.7)$] are shown in Table 3. Note that since we are reporting the accuracy of the system, we show the results by using the word accuracy (W_{Acc}), which is defined as $1 - \mathrm{WER}$.

It can be observed that the results of the OMR system developed are all close to 90 % of W_{Acc}. This means that a person in charge of the transcription has to deal with just the remaining 10 % to get the perfect transcription of the score, which would result in a very important saving of time and effort.

In order to assess the relevance of our proposal, Table 4 provides a comparison against a previous work that makes use of musical scores from the ACM (see [29]). As mentioned above, the staff detection and removal stage is one of the main reasons for symbol detection losses. The results show that our approach, which circumvents the staff removal process, leads to a remarkably good extraction rate. On the other hand, our classification approach, based on cross-correlation operator, attains good performance.

6 Conclusions

This work presents a new approach to deal with the optical music recognition process for scores written in white mensural notation from the Archivo de la Catedral de Malaga. These scores have a special printing style that allows us to propose a new approach in which the very common staff detection, and removal stage has been avoided. This stage is critical in the detection and recognition of symbols, and it is often one of the main steps to improve the accuracy rates of current OMR systems.

A preprocessing stage is necessary in order to correct both the rotation and the perspective distortion of the input image. At this stage, a binarization process has also been performed to reduce the complexity of the subsequent task. The next stage isolates each staff of the score, and a new symbol detection strategy has been followed. This strategy is based on the combination of the use of the y-projection of the staff and k-means clustering to detect the boundaries of each symbol region.

These procedures have proven to be reliable as they have achieved extraction rate performance higher than

96 %. The cross-correlation operator has shown its effectiveness in this context for classifying symbols that maintain the staff lines. Classification rates higher than 90 % are attained in all cases. However, new techniques for symbol classification could be applied or developed in future works since there still is some room for improvement. An overall evaluation of the system has also been computed. Our system transcribed the scores with an accuracy close to 90 %.

In comparison with previous results on the ACM (see Table 4), our work attains very good extraction rate on average: 96.24 %, which proves that avoiding staff removal stage is a very valuable choice for the task in terms of symbol detection. In addition, the classification accuracy is also good: 91.64 %, on average, using a very simple classification strategy.

The work presented opens new avenues for building OMR systems. We believe that the avoidance of the staff detection and removal step deserves further research and can be a way to overcome some of the common misclassification problems that exist in current systems. This approach should be considered to analyze other types of scores to assess whether it can be definitely established as a new alternative for the construction of these systems.

Acknowledgments This work has been funded by the Ministerio de Educación, Cultura y Deporte of the Spanish Government under a FPU Fellowship No. AP20120939, by the Ministerio de Economía y Competitividad of the Spanish Government under Project No. TIN2013-48152-C2-1-R and Project No. TIN2013-47276-C6-2-R, by the Consejería de Educación de la Comunidad Valenciana under Project No. PROMETEO/2012/017 and by the Junta de Andalucía under Project No. P11-TIC-7154. This work has been done at Universidad de Málaga. Campus de Excelencia Internacional Andalucía Tech. The authors are grateful to the person in charge of the Archivo de la Catedral de Málaga, who allowed the utilization of the data sets used in this work.

References

1. Bainbridge D, Bell T (2001) The challenge of optical music recognition. Lang Resour Eval 35:95–121
2. Barbancho I, Segura C, Tardon LJ, Barbancho AM (2010) Automatic selection of the region of interest in ancient scores. In: MELECON 2010–2010 15th IEEE Mediterranean Electrotechnical Conference, pp 326–331
3. Bribiesca E (1999) A new chain code. Pattern Recognit 32(2):235–251
4. Chen YS, Chen FS, Teng CH (2013) An optical music recognition system for skew or inverted musical scores. Int J Pattern Recognit Artif Intell 27(07):1–23
5. Deza MM, Deza E (2009) Encyclopedia of Distances, first edn. Springer, New York
6. Duda RO, Hart PE (1973) Pattern classification and scene analysis, first edn. Wiley, Hoboken
7. Dutta A, Pal U, Fornes A, Llados J (2010) An efficient staff removal approach from printed musical documents. In: Pattern Recognition (ICPR), 2010 20th International Conference. pp 1965–1968
8. Fornés A, Lladós J, Sánchez G (2005) Staff and graphical primitive segmentation in old handwritten music scores. In: Proceedings of the 2005 conference on Artificial Intelligence Research and Development. IOS Press, Amsterdam, pp 83–90
9. Freeman H (1961) On the encoding of arbitrary geometric configurations. Electr Comput IRE Trans EC 10(2):260–268
10. Gonzalez RC, Woods RE (2007) Digital Image Processing. Prentice-Hall, Upper Saddle River
11. Hartigan JA (1975) Clustering algorithms. Wiley, Hoboken
12. Hwang SK, Kim WY (2006) Fast and efficient method for computing art. Image Process IEEE Trans 15(1):112–117
13. Jelinek F (1998) Statistical methods for speech recognition. The MIT Press, Cambridge
14. Levenshtein VI (1966) Binary codes capable of correcting deletions, insertions and reversals. Sov Phys Dokl 10:707
15. Lewis JP (1995) Fast template matching. In: Vision Interface. Canadian Image Processing and Pattern Recognition Society, Quebec City, pp 120–123
16. Ng KC, Cooper D, Stefani E, Boyle RD, Bailey N (1999) Embracing the composer: optical recognition of handwritten manuscripts. In: Proceedings of the International Computer Music Conference, Beijing
17. Otsu N (January 1979) A threshold selection method from gray-level histograms. IEEE Trans Syst Man Cybern 9(1):62–66
18. Caldas Pinto JR, Vieira P, Ramalho M, Mengucci M, Pina P, Muge F (2000) Ancient music recovery for digital libraries. In: Proceedings of the 4th European Conference on Research and Advanced Technology for Digital Libraries, ECDL '00. Springer, London, pp 24–34
19. João Rogério Caldas Pinto, Vieira P, João Miguel da Costa Sousa (2003) A new graph-like classification method applied to ancient handwritten musical symbols. IJDAR 6(1):10–22
20. Pizer SM, Johnston RE, Ericksen JP, Yankaskas BC, Muller KE (1990) Contrast-limited adaptive histogram equalization: speed and effectiveness. In: Visualization in Biomedical Computing, 1990, Proceedings of the First Conference, pp 337–345
21. Pruslin D (1966) Automatic recognition of sheet music. Sc.d. dissertation, Massachusetts Institute of Technology
22. Pugin L (2006) Optical music recognition of early typographic prints using hidden markov models. In: ISMIR, pp 53–56
23. Rebelo A, Fujinaga I, Paszkiewicz F, Marcal ARS, Guedes C, Cardoso JS (2012) Optical music recognition: state-of-the-art and open issues. Int J Multimed Inf Retr 1(3):173–190
24. Sarvaiya JN, Patnaik S, Bombaywala S (2009) Image registration by template matching using normalized cross-correlation. In: Advances in Computing, Control, Telecommunication Technologies, 2009. ACT '09. International Conference on, pp 819–822
25. Sotoodeh M, Tajeripour F (2012) Staff detection and removal using derivation and connected component analysis. In: Artificial Intelligence and Signal Processing (AISP), 2012 16th CSI International Symposium, pp 054–057
26. Stigler SM (1989) Francis Galton's account of the invention of correlation. Statistical Sci 4:73–79
27. Su B, Lu S, Pal U, Tan CL (2012) An effective staff detection and removal technique for musical documents. In: Document analysis systems (DAS), 2012 10th IAPR International Workshop, pp 160–164
28. Szwoch M (2005) A robust detector for distorted music staves. In: Gagalowicz A, Philips W (eds) computer analysis of images and patterns, vol 3691, Lecture notes in computer science. Springer, Berlin Heidelberg, pp 701–708
29. Tardón LJ, Sammartino S, Barbancho I, Gómez V, Oliver A (2009) Optical music recognition for scores written in white mensural notation. J Image Video Process 6
30. Toyama F, Shoji K, Miyamichi J (2006) Symbol recognition of printed piano scores with touching symbols. In: Pattern

Recognition, 2006. ICPR 2006. 18th International Conference, vol 2, pp 480–483

31. Trier OD, Taxt T (1995) Evaluation of binarization methods for document images. Pattern Analysis Mach Intell IEEE Trans 17(3):312–315
32. Wei SD, Lai SH (Nov 2008) Fast template matching based on normalized cross correlation with adaptive multilevel winner update. Image Process IEEE Trans 17(11):2227–2235
33. Zahn CT, Roskies RZ (March 1972) Fourier descriptors for plane closed curves. IEEE Trans Comput 21(3):269–281

Chapter 4

Music staff removal with supervised pixel classification

Calvo-Zaragoza, J., Micó, L., and Oncina, J. (2016a). Music staff removal with supervised pixel classification. *International Journal on Document Analysis and Recognition*, Online:1–9

ORIGINAL PAPER

Music staff removal with supervised pixel classification

Jorge Calvo-Zaragoza[1] · Luisa Micó[1] · Jose Oncina[1]

Received: 5 October 2015 / Revised: 22 December 2015 / Accepted: 24 February 2016

Abstract This work presents a novel approach to tackle the music staff removal. This task is devoted to removing the staff lines from an image of a music score while maintaining the symbol information. It represents a key step in the performance of most optical music recognition systems. In the literature, staff removal is usually solved by means of image processing procedures based on the intrinsics of music scores. However, we propose to model the problem as a supervised learning classification task. Surprisingly, although there is a strong background and a vast amount of research concerning machine learning, the classification approach has remained unexplored for this purpose. In this context, each foreground pixel is labelled as either *staff* or *symbol*. We use pairs of scores with and without staff lines to train classification algorithms. We test our proposal with several well-known classification techniques. Moreover, in our experiments no attempt of tuning the classification algorithms has been made, but the parameters were set to the default setting provided by the classification software libraries. The aim of this choice is to show that, even with this straightforward procedure, results are competitive with state-of-the-art algorithms. In addition, we also discuss several advantages of this approach for which conventional methods are not applicable such as its high adaptability to any type of music score.

Keywords Music staff removal · Optical music recognition · Pixel classification · Supervised learning

1 Introduction

Music constitutes one of the main tools for cultural transmission. That is why musical documents have been carefully preserved over the centuries. In an effort to prevent their deterioration, the access to these sources is not always possible. This implies that an important part of this historical heritage remains inaccessible for musicological study. Digitizing this content allows a greater dissemination and integrity of this culture. Furthermore, the massive digitization of music documents opens several opportunities to apply music information retrieval algorithms, which may be of great interest for music analysis. Since the manual transcription of music sources is a long, tedious task—which often requires expert supervision—the development of automatic transcription systems is gaining importance over the last decades [4,22,23].

Optical music recognition (OMR) can be defined as the ability of a computer to understand the musical information contained in the image of a music score. The process basically consists in receiving a scanned music score and exporting its musical content to some machine-readable format (Fig. 1). This task can be considered very similar to that of optical character recognition. Nevertheless, its higher complexity

This work has been supported by the Spanish Ministerio de Educación, Cultura y Deporte through a FPU Fellowship (Ref. AP2012–0939) and the Spanish Ministerio de Economía y Competitividad through Project TIMuL (No. TIN2013-48152-C2-1-R, supported by UE FEDER funds).

✉ Jorge Calvo-Zaragoza
jcalvo@dlsi.ua.es

Luisa Micó
mico@dlsi.ua.es

Jose Oncina
oncina@dlsi.ua.es

[1] Departamento de Lenguajes y Sistemas Informáticos, Universidad de Alicante, Carretera San Vicente del Raspeig s/n, 03690 Alicante, Spain

Published online: 12 March 2016

and particular notation, in comparison with text, lead to the need of specific developments [1].

OMR has to deal with many aspects of musical notation, one of which is the presence of the staff, the set of five parallel lines used to define the pitch of each musical symbol. Although these lines are necessary for human readability, they complicate the automatic segmentation and classification of musical symbols. Some works have approached the problem maintaining the staves [3,24,25]; however, a common OMR preprocessing includes the detection and removal of staff lines [28]. This task is aimed at removing the staff lines of the score, maintaining as much as possible the symbol information.

Although staff lines detection and removal may be seen as a simple task, it is often difficult to get accurate results. This is mainly due to problems such as discontinuities, skewing, slant or paper degradation (especially in ancient documents). Given that, the more accurate this process, the better the detection of musical symbols, much research has been devoted to this process, which can be considered nowadays as a research topic by itself.

Notwithstanding all these efforts, the staff removal stage is still inaccurate and it often produces noise, for example staff lines not completely removed. Although more aggressive methods that minimize noise can be used, they might produce partial or total loss of some musical symbols. The trade-off between these two aspects, in addition to the accuracy of the techniques, has hitherto led to the inevitable production of errors during this stage. Moreover, the differences between score style, sheet conditions and scanning processes lead researchers to develop some kind of *ad hoc* method for staff detection and removal, which usually presents little robustness when it is applied to different staves.

From another point of view, the process of removing staff lines can be defined as a classification problem in which, given some foreground pixel, it must be guessed whether that pixel is a part of a *staff* or a *symbol* (i.e. binary classification). Note that addressing the problem in this way, both staff detection and removal can be performed at the same time.

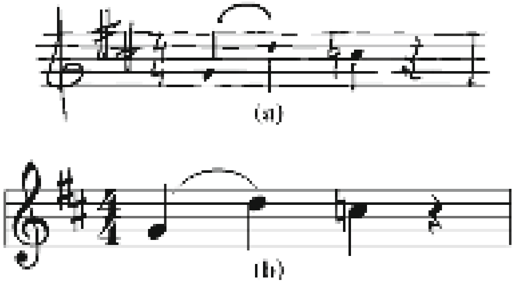


Fig. 1 The task of optical music recognition (OMR) is to analyse an image containing a music score to export its musical content to some machine-readable format. **a** Example of input score for an OMR system. **b** Symbolic representation of the input score

To the best of our knowledge, this approach still remains unexplored. Hence, this work aims at providing a first insight into the staff removal process modelled as a binary classification task. To this end, a set of features based on neighbourhood pixels is extracted at each foreground pixel. At the experimentation stage, several common pattern recognition algorithms will be applied using these features. Our main intention is to show that this simple and general approach deserves further consideration since its performance reaches the level of state-of-art methods while offering several advantages that the others cannot.

This paper is organized as follows: Sect. 2 presents background on staff detection and removal; Sect. 3 describes our approach to model the process as a classification task; Sect. 4 contains the experimentation performed and the results obtained; Sect. 5 discusses the pros and cons of our approach, and some additional considerations; and finally, Sect. 6 concludes the current work.

2 Background

Due to the complexity of music notation, OMR systems rely on music staff removal algorithms to perform the most relevant task of symbol isolation and segmentation [26]. Note that this process should not only detect staff lines but also remove them in such a way that musical symbols remain intact (see Fig. 2).

Unfortunately, this removal stage is hardly ever perfect. The need of eliminating every part of the staff often leads to delete some parts of the musical symbols, which produces unavoidable errors in posterior stages. The trade-off between keeping symbols and removing staff lines leads to inevitable production of extraction and classification errors later. That is why several methods have been proposed to tackle this process. A good comparative study, including a taxonomy of the different approaches, can be found in the work of Dalitz et al. [7].

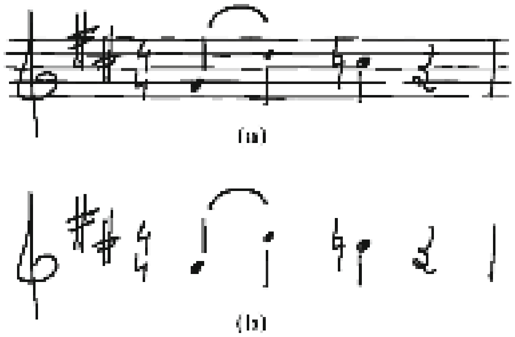


Fig. 2 Example of an accurate staff removal process. **a** Example of input score for an OMR system. **b** Input score after staff removal

In the last years, however, new strategies have been developed: Cardoso et al. [30] proposed a method that considers the staff lines as connecting paths between the two margins of the score. Then, the score is modelled as a graph so that staff detection is solved as a maximization problem. This strategy was improved and extended to be used on grey-scale scores [27]; Dutta et al. [10] developed a method that considers the staff line segment as a horizontal connection of vertical black runs with uniform height, which are validated using neighbouring properties; in the work of Piatkowska et al. [21], a swarm intelligence algorithm was applied to detect the staff line patterns; Su et al. [31] start estimating properties of the staves like height and space; then, they tried to predict the direction of the lines and fitted an approximate staff, which was posteriorly adjusted; Geraud [13] developed a method that entails a series of morphological operators directly applied to the image of the score to remove staff lines; and Montagner et al. [19] proposed to learn image operators, following the work of Hirata [17], whose combination was able to remove staff lines. Others works have addressed the whole OMR problem by developing their own, case-directed staff removal process [29,32].

The current performance of staff removal methods can be checked in the *GREC/ICDAR 2013 staff removal competition* [12,34]. This competition makes use of the CVC-MUSCIMA database [11], which contains handwritten music score images with a perfect ground truth on staff removal. Many of the most advanced methods showed a decreasing accuracy when different distortions were applied to the input scores. Indeed, the same behaviour may be expected by methods, especially suitable for some type of score that are subsequently applied to very different conditions. Taking into account the vast variety of music manuscripts—which is even wider considering old music—there is a need of developing staff removal methods that are able to deal with any kind of score.

In our work, we propose to model the staff removal stage as a pixel classification problem. That is, extract features from each foreground pixel and take a decision about keeping or removing it based on supervised learning classification techniques. Therefore, the accuracy of the method lies in data instead of in selecting the appropriate series of image processing steps. Although it may be worse in the cases in which specific staff removal algorithms have been developed, it allows us to present a robust approach since it can be effective in any type of score as long as labelled data are available. The strategy proposed is described in next section.

3 A classification approach for staff lines removal

As depicted above, several procedures for the staff detection and removal stage have been proposed. Although most of them are able to achieve a very good performance in many cases, they are far from optimal when the style of the score is changed. The intention of our strategy is to present a new method that is able to adapt to the actual score style as long as learning data are available.

To handle this issue, we propose to follow a supervised learning approach. That is, the task is based on building a classification model using a training sample with labelled data. After that, the model is able to receive new unseen samples and determine the class label [9].

In our context, given an image depicting a score, we extract a labelled set of features from each foreground pixel. These features are used to train a classification algorithm. At test phase, each of these pixels is classified between *symbol* or *staff*. Then, depending on what it is pursued—either staff detection or staff removal—it is removed from the image those pixels classified as symbol or those classified as staff. Without loss of generality, we shall assume from now on that our objective is the staff removal stage since it is the common preprocessing required in OMR systems.

In this work, the features of each pixel of interest consist of the values of its neighbouring region. We believe that the surroundings of each pixel contains contextual information that can be discriminative enough for this task. Furthermore, this contextual information can help to avoid misclassification due to noise or small deformations of the image.

We shall assume that the input score has been binarized previously, as it is usual in this field. Nevertheless, our feature extraction is not restricted to binary images, but it could be applied to any type of image.

Formally speaking, let $I : \mathbb{N} \times \mathbb{N} \rightarrow \{0, 1\}$ define the input score image. We use w_h and w_w to denote two integer values defining the opening of the neighbouring region in each dimension (horizontal and vertical, respectively). Given a position (i, j) of the input image, a set of $(2\,w_h + 1)(2\,w_w + 1)$ features $(\mathbf{f}_{i,j})$ is considered taking the values of the neighbourhood region centred at (i, j). That is, $\mathbf{f}_{i,j} = \{I(x, y) : |i - x| \leq w_h \wedge |j - y| \leq w_w\}$. Then, the values contained within this set are concatenated following some specific order (e.g. by columns) to obtain a proper feature vector. This process is illustrated in Fig. 3.

To obtain the training set, the feature extraction process is applied to the foreground pixels of a labelled data set of scores with and without staff lines. Given a pixel in the position (i, j), the feature extraction is applied in the score that contains staff lines (i.e. in the original one). After that, the value in the position (i, j) of the score without staff lines is used to obtain the actual label between *staff* or *non-staff*.

This training set is used to feed a supervised learning classifier. Then, when an input score is received, this classifier will be able to take a decision about each of its foreground pixels. If it is classified as *staff*, the pixel will be removed from the image.

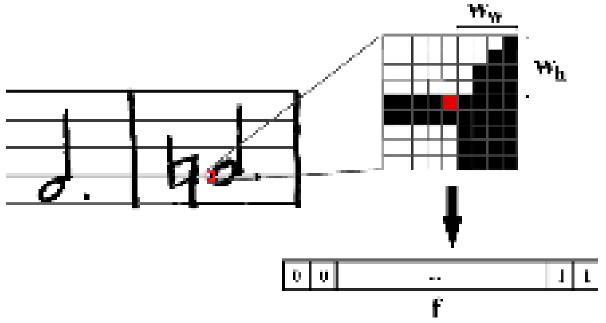


Fig. 3 Example of feature extraction considering $w_w = w_h = 4$. Cell in *red* represents the pixel from which features are being extracted (colour figure online)

It should be emphasized that the intention of this work is not to find the most suitable pair of features and classification algorithm, but to show that this approach allows dealing with the staff removal stage even with a very straightforward classification step. Pursuing the best configuration could cause results depend more on these aspects than on the approach itself. Thus, a more comprehensive feature extraction and classification research fall outside the actual intention of this work.

Next section will present experimentation with some common classification algorithms considering several values of w_h and w_w.

4 Experimentation

This section details the experimentation carried out to test our proposal. Taking advantage of the *GREC/ICDAR 2013 staff removal competition*, whose data set was publicly available,[1] we are going to follow the same experimental set-up to assure a fair comparison with state-of-art developments.

The data used in this contest are organized in train and test sets, with 4000 and 2000 samples, respectively. The test set is further divided into three subsets (TS1, TS2 and TS3) based on the deformations applied to the scores. Each sample consists of an image of a handwritten score in both binary and grey scale with its corresponding ground truth (the score without staves). We shall use here the binary ones. Figure 4 shows a piece from a score of that set. The number of foregrounds pixels per score is around 500,000 with 200,000 staff pixels, both on average.

The training set will be used to learn to distinguish between staff and symbol pixels by the classification algorithms. Due to the large amount of data available, it is infeasible to handle it completely. Thus, at each instance only one score of the training set, chosen randomly, will be used. If we also consider that one score contains around 500,000 foreground pixels, this is still too much information to use as training set.

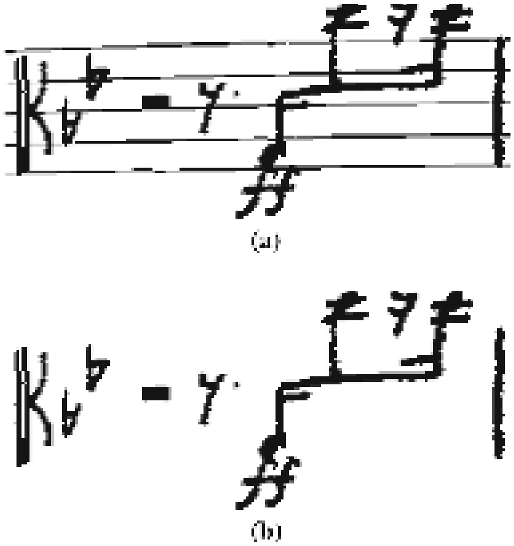


Fig. 4 Piece of sample from the *GREC/ICDAR 2013 staff removal competition* data set. **a** Score. **b** Ground truth for staff removal

We must bear in mind that the whole set of foreground pixels may be used to train the classifiers. Then, to further reduce the size of the training sample, the condensing algorithm [16] was applied. This algorithm removes the samples that are not considered relevant enough for the classification task. After that, the average size of the training sample was around 20,000. In other words, only 4 % of the foreground pixels of one score randomly selected have been used as training, which constitutes 0.001 % of the available training information.

On the other hand, test set will be used to assess the results achieved. As in the competition, the performance metric will be the F_1 score or *F-measure*:

$$F_1 = \frac{2\,\mathrm{TP}}{2\,\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}$$

where TP, FP and FN stand for true positives (staff pixels classified as staff), false positives (symbol pixels classified as staff) and false negatives (staff pixels classified as symbol), respectively.

4.1 Classification techniques

For the classification task, many supervised learning algorithm can be applied. In this work, we are going to consider the following methods:

- Nearest neighbour (NN) [6]: given a distance function between samples, this algorithm proposes a label of the

[1] http://dag.cvc.uab.es/muscima/.

input by querying its nearest neighbour of the training set. The Euclidean distance was used for our task.

- Support vector machine (SVM) [33]: it learns a hyperplane that maximizes the distance to the nearest samples (support vectors) of each class. It makes use of kernel functions to handle nonlinear decision boundaries. In our case, a *radial basis function* kernel was chosen.
- Random forest (RaF) [2]: it builds an ensemble classifier by generating several random decision trees at the training stage. The final output is taken by combining the individual decisions of each tree.

The methods described above have been applied using the Waikato Environment for Knowledge Analysis (WEKA) library [15], each one with their default parametrization unless where it has been told otherwise. Since our interest is not focused on finding the best classifier for this task, but to emphasize the supervised learning approach, we did not pursue the optimal tuning of these classifiers.

4.2 Results

For our experiments, we shall restrict ourselves to consider squared neighbouring regions. Concretely, regions with windows of length 1, 2, 3 and 4 in each direction centred at the pixel. Therefore, each pixel of interest is defined by 9, 25, 49 and 81 binary features, respectively. Results achieved are given in Table 1.

An initial remark is that the number of features has a stronger influence in the results than the algorithm used. For instance, classifiers showed a poor performance when 9 features are considered, but they increase noticeably the accuracy with 25 features. It is important to stress that each configuration outperforms results of any other configuration with less features, with independence of the algorithm used. This is also reported in the last row of the table, in which the average improvement obtained by increasing the number of features is depicted.

Table 1 Average F_1 score (%) achieved by the different classification techniques in combination with different values of neighbouring squared region over the three test subsets

Test set	Classifier	Features			
		9	25	49	81
TS1	NN	68.34	86.10	89.69	91.07
	SVM	40.72	87.14	93.95	**94.10**
	RaF	68.06	90.12	93.52	93.89
TS2	NN	77.32	90.05	95.24	96.06
	SVM	51.22	97.02	**98.11**	98.08
	RaF	76.46	93.86	96.95	97.78
TS3	NN	71.56	86.23	89.33	90.58
	SVM	48.07	87.81	93.92	**94.00**
	RaF	71.15	90.55	93.23	93.39
Average		63.43	89.87	93.77	94.32

Bold values indicate the best results, on average, at each subset. The average results obtained with each set of features are also showed

Results seem to be stabilized within the two highest number of features considered. Thereby including more than 81 is not expected to improve accuracy significantly. In addition, increasing the number of features may imply some drawbacks such as efficiency in both learning and testing phase.

Regarding classification techniques, SVM achieves best results for both each feature extraction considered and each test subset, although its difference with RaF is hardly significant. In turn, NN does present a lower accuracy than the others. Specifically, SVM with 81 is reported as the best configuration, on average, and it is also the best result in two of the three corpora used.

Figure 5 shows an example of the behaviour provided by our algorithm (SVM with 81 features), in which a great accuracy is achieved. Surprisingly, misclassification is not found in the edges between symbol and staff, but it mainly occurs along the staff lines. In fact, looking in more detail, very few symbol pixels are removed. It should be stressed that most of these remaining mistakes could be hopefully corrected by means of a post-processing step.

To analyse our performance against state-of-the-art methods, Table 2 shows a summary of the results achieved in the staff removal competition for each test set. These sets comprise different deformations applied over original scores: 3D distortions in TS1, local noise in TS2, and both 3D distortion and local noise in TS3. For a detailed description about each

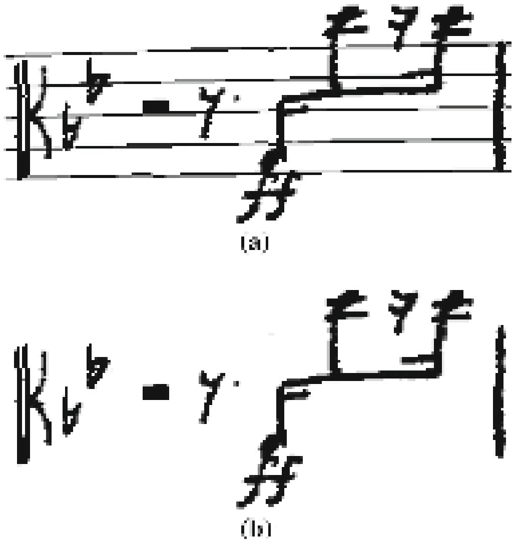


Fig. 5 Example of a staff removal process using SVM classifier, 81 features per pixel and only one *condensed* score as training set. **a** Input score. **b** Score after staff removal

Table 2 F_1 comparison between the best tuning of our method and the participants in the staff removal contest

Method	TS1	TS2	TS3
TAU	85.72	81.72	82.29
NUS	69.85	96.25	67.43
NUASI-lin	94.77	94.76	93.81
NUASI-skel	94.11	93.67	92.78
LRDE	**97.73**	96.86	**96.98**
INESC	89.29	97.72	88.52
Baseline	87.01	96.91	89.90
Our method	94.10	**98.08**	94.00

Best values, on average, achieved on each subset are highlighted

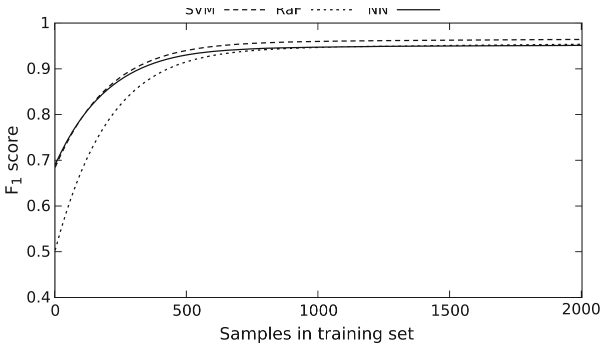


Fig. 6 Performance of the classifiers using 81 features with respect to the amount of training samples

participant and the deformation models applied, reader is referred to the report of the competition [34]. Our best average configuration (SVM with 81 features) is also included for comparison.

Most of the methods proposed in the contest follow a two-step approach: first, an estimation of the position of the staff lines and then staff lines removal while keeping symbol information. This second step is what usually produces the accuracy loss, since it is difficult to distinguish symbol pixels over a staff line. On the contrary, our method is directly focused on the final task without a first estimation of staff lines.

According to the results, our method shows the best accuracy against local noise (TS2). This is probably because local noise is less harmful for our feature extraction and classification. In turn, they are less generalizable to deal with 3D distortions (TS1 and TS3), for which our approach suffers some accuracy loss. Although we only achieve the highest score in one of the two subsets considered, our results are quite competitive as differences among best results and those obtained by our method are very small. In addition, our method surpasses many of the participants in the contest in all sets considered. It should be noted that not only this configuration is competitive but also most of the configurations given in Table 1, even with a little set of features. Moreover, we must also remember that for obtaining these results only 0.001 % of all available training information was used.

Finally, we focused on assessing whether the amount of data used to train the classifiers has a strong impact on the results. To this end, another experiment has been performed in which the number of training samples is iteratively increased, using a random subset of the training scores as validation set. As mentioned above, the specific size of the training set in our previous experiments is given by condensing algorithm, which keeps around 20,000 samples, on average. Figure 6 shows the curves of such experiment extracting 81 features per pixel (the highest value considered in our experiments). It can be seen that the performance is already stable when classifiers are trained with 2000 samples. This leads to the insight that results are not expected to improve significantly if more data were considered.

Given all of above, we consider that our proposal should merit high interest since its performance is competitive using a simple strategy that has not been studied so far. Next section extends the implications that our method has in the ongoing research on staff removal, supported by the results obtained.

5 Discussion

Since the work presented here is the first approach to the staff removal task as a pixel classification problem, it opens several lines of discussion that should be addressed.

The first thing to remark is that the performance of our method is very competitive, although it does not significantly outperform all the already proposed ones. While this fact may question the usefulness of the proposal, relevant additional advantages are shown. First of all, it is simple, easy to use and does not require additional knowledge of the field. In addition, a fine-tuning of the classifiers parameters, as well as using some kind of advanced feature extraction, clearly represents room for accuracy improvement.

Unfortunately, this method has also drawbacks that deserve consideration in future developments. For instance, approaching the task from a classification point of view is very expensive. Regardless the specific classifier speed, each foreground pixel of the score entails a classification process. Therefore, our method will be usually slower than conventional image processing methods.

From the learning-driven process point of view, the staff removal stage is as robust as its training set. That is, the process can be accurate if we have enough data of the target type of score. Foreground information such as handwritten notation or noise can also be addressed simultaneously as long as they appear in the training data. Furthermore, this paradigm allows the method to be adapted to any type of

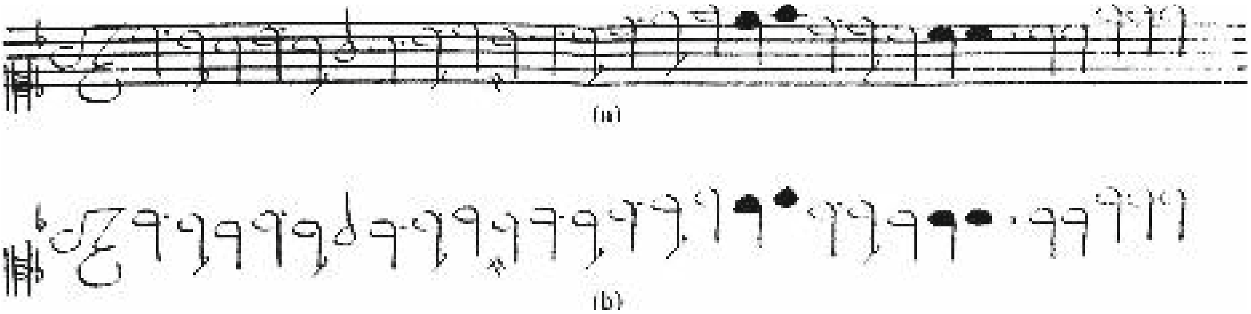


Fig. 7 Training set used for the *proof-of-concept* experiment over early music scores. **a** Score. **b** Ground truth for staff removal

music score, even those quite different such as Gregorian chant or guitar tablatures. In those cases, classical methods may fail because of the high variation with respect to classical notation or the variable number of staff lines.

To serve as an example, we have carried out a simple *proof-of-concept* experiment to compare the adaptiveness of our proposal against a classical one. The experiment is focused on early music manuscripts so as to analyse the behaviour of the methods when dealing with quite different musical scores.

In order to feed the supervised learning classifier, we have manually labelled a single line of staff of this type (see Fig. 7). Note that we are just using a very small piece as training set, which is expected to be available with small effort.

As a representative of classical image processing strategies, we have chosen the LRDE method, since it depicted the best performance in the contest. Its publicly available online demo[2] has been used for this test.

Figure 8 shows the results of a staff removal process applying both our proposal and this method to an early music piece of score. For the sake of further analysis, our method is trained with both specific data and data of CVC-MUSCIMA. It is important to remark that the LRDE method is not able to remove accurately staff lines in spite of being one of the best in the contest. On the other hand, our method achieves a very poor performance if data are not appropriate, as depicted in Fig. 8c. However, if specific data are used, results are fairly accurate (Fig. 8d). Although this comparison may not be totally fair, it clearly illustrates some drawback of developing image procedures to remove staff lines in contrast to a learning-based approach.

5.1 Further considerations

In addition to the advantages discussed previously, considering staff removal as a supervised classification problem allows us to explore many other paradigms that could be profitable for this task:

- Online learning: new data may be available through the use of the system [8]. For instance, when user corrects OMR mistakes, the information could be analysed to extract new labelled pixels for the staff removal process. This case may be useful when it is assumed that the data of the image in process are more relevant than the training set itself.
- Active learning: if it is assumed that a user must be supervising the OMR task, the system could query about the category (staff or symbol) of some piece of the score. The main goal is to reduce the user effort in the whole process, and therefore, some queries may be needed to avoid many of the potential mistakes in the classification stage [14].
- One-class classification: since the staff removal may entail an imbalanced binary classification with respect to the training data available, it could also be modelled as a one-class classification problem [18]. This case seems to be very interesting because it would need less data to train (just one of the two classes considered, the one whose data are more available) and some strategies could be applied to automatically extract labelled data of that class from score images.
- Deep learning: taking into account the huge amount of labelled data present in this task, this paradigm may learn the high-level representation inherent to each piece of the score to learn to distinguish between symbol and staff pixels more accurately. Convolutional neural networks have been reported to be especially suitable for performing such a task [5].

These points represent ideas that could be implemented to improve the process so that it becomes more adaptive, efficient and/or effective. Nevertheless, it should be noted that most of these paradigms can not be applied if conventional methods for staff removal are used.

On the other hand, one of the main obstacles in the preprocessing of degraded documents is the binarization step. However, the method proposed in this work could be trained to deal with grey-level images, although it would represent

[2] https://olena.lrde.epita.fr/demos/staff_removal.php.

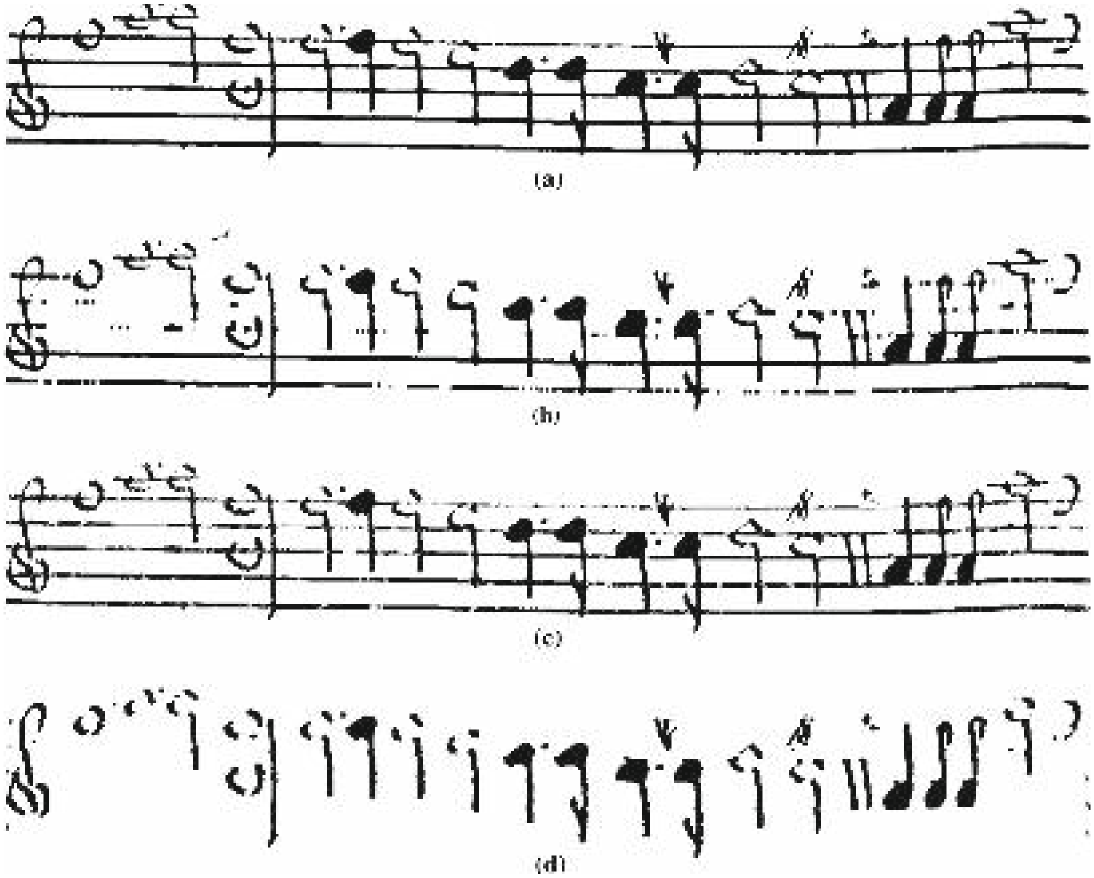


Fig. 8 Performance of LRDE method and our proposal (SVM classifier and 81 features per pixel) with general and specific data over an ancient score of early music. **a** Input score. **b** Input score after staff removal by LRDE method. **c** Input score after staff removal by our proposal with CVC-MUSCIMA data. **d** Input score after staff removal by our proposal with specific data

a different task. Since background pixels would have to be classified as well, the complexity of the process would be increased drastically.

For all the reasons above, we believe that this approach is worthwhile in its current form since the performance achieved is comparable to state of the art with a very straightforward procedure. Moreover, it is specially interesting when considering all the research avenues and opportunities opened, some of which could lead to a significantly higher performance than that obtained by the methods proposed so far.

6 Conclusions

In this work, we presented a novel approach for the staff removal stage, a key preprocessing step in the performance of most OMR systems. Our strategy models the task as a supervised learning classification problem, in which each foreground pixel is classified as *staff* or *symbol* using raw neighbouring pixels as features.

In our experiments, the feature set was demonstrated to be more relevant than the specific classifier in the accuracy results. SVM classifier, considering 81 features, reported the best results on average. In comparison with other state-of-the-art staff removal processes, our strategy showed a very competitive performance, even achieving the best results in some cases, using a very small piece of the training information. A proof-of-concept experiment over early music scores has also been carried out as an example of the robustness of our method. Therefore, this novel approach deserves further consideration in the field since it also opens several opportunities for which conventional methods are not applicable.

As future work, some effort should be devoted to overcoming the problem of getting enough data to train the classifiers. For instance, the conditions of the actual sheet—such as scale and deformation—could be learned online. Then, the same conditions could be applied to a reference data set so that specific labelled data are obtained for each type of score. The use of adaptive techniques for domain adaptation or transfer learning is another way to deal with this issue [20]. Similarly, considering an interactive OMR system, staff removal learning could be improved through user interaction.

Moreover, there is still plenty of room for improvement regarding the classification process such as seeking a better feature set or using other advanced techniques for supervised learning. Speeding up the process may be also of great interest. For instance, by classifying a relatively small block of the score at a once, instead of querying every single pixel of the image.

References

1. Bainbridge, D., Bell, T.: The challenge of optical music recognition. Comput. Humanit. **35**(2), 95–121 (2001). doi:10.1023/A:1002485918032
2. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001). doi:10.1023/A:1010933404324
3. Calvo-Zaragoza, J., Barbancho, I., Tardón, L.J., Barbancho, A.M.: Avoiding staff removal stage in optical music recognition: application to scores written in white mensural notation. Pattern Anal. Appl. **18**(4), 933–943 (2015)
4. Carter, N.P.: Segmentation and preliminary recognition of madrigals notated in white mensural notation. Mach. Vis. Appl. **5**(3), 223–229 (1992). doi:10.1007/BF02627000
5. Ciresan, D., Meier, U., Schmidhuber, J.: Multi-column deep neural networks for image classification. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3642–3649. IEEE (2012)
6. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Trans. Inf. Theory **13**(1), 21–27 (1967). doi:10.1109/TIT.1967.1053964
7. Dalitz, C., Droettboom, M., Pranzas, B., Fujinaga, I.: A comparative study of staff removal algorithms. IEEE Trans. Pattern Anal. Mach. Intell. **30**(5), 753–766 (2008). doi:10.1109/TPAMI.2007.70749
8. Diethe, T., Girolami, M.: Online learning with multiple kernels: a review. Neural Comput. **25**(3), 567–625 (2013)
9. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley, New York (2001)
10. Dutta, A., Pal, U., Fornes, A., Llados, J.: An efficient staff removal approach from printed musical documents. In: 2010 20th International Conference on Pattern Recognition (ICPR), pp. 1965–1968 (2010)
11. Fornés, A., Dutta, A., Gordo, A., Lladós, J.: CVC-MUSCIMA: a ground truth of handwritten music score images for writer identification and staff removal. Int. J. Doc. Anal. Recogn. **15**(3), 243–251 (2012)
12. Fornés, A., Kieu, V.C., Visani, M., Journet, N., Dutta, A.: The ICDAR/GREC 2013 music scores competition: Staff removal. In: 10th International Workshop on Graphics Recognition, Current Trends and Challenges GREC 2013, Bethlehem, PA, USA, August 20–21, 2013, Revised Selected Papers, pp. 207–220 (2013)
13. Géraud, T.: A morphological method for music score staff removal. In: Proceedings of the 21st International Conference on Image Processing (ICIP), pp. 2599–2603. Paris, France (2014)
14. Gosselin, P., Cord, M.: Active learning methods for interactive image retrieval. IEEE Trans. Image Process. **17**(7), 1200–1211 (2008)
15. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. SIGKDD Explor. Newsl. **11**(1), 10–18 (2009). doi:10.1145/1656274.1656278
16. Hart, P.: The condensed nearest neighbor rule (corresp.). IEEE Trans. Inf. Theory **14**(3), 515–516 (1968). doi:10.1109/TIT.1968.1054155
17. Hirata, N.S.T.: Multilevel training of binary morphological operators. IEEE Trans. Pattern Anal. Mach. Intell. **31**(4), 707–720 (2009)
18. Khan, S.S., Madden, M.G.: One-class classification: taxonomy of study and review of techniques. Knowl. Eng. Rev. **29**, 345–374 (2014)
19. Montagner, I.d.S., Hirata, R., Hirata, N.S.: A machine learning based method for staff removal. In: Pattern Recognition (ICPR), 2014 22nd International Conference on, pp. 3162–3167 (2014). doi:10.1109/ICPR.2014.545
20. Patel, V.M., Gopalan, R., Li, R., Chellappa, R.: Visual domain adaptation: a survey of recent advances. IEEE Signal Process. Mag. **32**(3), 53–69 (2015)
21. Piatkowska, W., Nowak, L., Pawlowski, M., Ogorzalek, M.: Stafflines pattern detection using the swarm intelligence algorithm. In: Bolc, L., Tadeusiewicz, R., Chmielewski, L.J., Wojciechowski, K. (eds.) Computer Vision and Graphics. Lecture Notes in Computer Science, vol. 7594, pp. 557–564. Springer, Berlin Heidelberg (2012)
22. Pinto, J.R.C., Vieira, P., Ramalho, M., Mengucci, M., Pina, P.,Muge, F.: Ancient music recovery for digital libraries. In:Proceedings of the 4th European Conference on Research and Advanced Technology for Digital Libraries, ECDL ’00, pp. 24–34.Springer, London, UK, UK (2000)
23. Pruslin, D.: Automatic recognition of sheet music. Sc.d. dissertation, Massachusetts Institute of Technology, UK (1966)
24. Pugin, L.: Optical music recognition of early typographic prints using hidden Markov models. In: ISMIR 2006, 7th International Conference on Music Information Retrieval, Victoria, Canada, 8–12 October 2006, Proceedings, pp. 53–56 (2006)
25. Ramirez, C., Ohya, J.: Automatic recognition of square notation symbols in western plainchant manuscripts. J. New Music Res. **43**(4), 390–399 (2014)
26. Rebelo, A., Capela, G., Cardoso, J.S.: Optical recognition of music symbols. Int. J. Doc. Anal. Recogn. **13**(1), 19–31 (2010)
27. Rebelo, A., Cardoso, J.: Staff line detection and removal in the grayscale domain. In: 2013 12th International Conference on Document Analysis and Recognition (ICDAR), pp. 57–61 (2013). doi:10.1109/ICDAR.2013.20
28. Rebelo, A., Fujinaga, I., Paszkiewicz, F., Marçal, A.R.S., Guedes, C., Cardoso, J.S.: Optical music recognition: state-of-the-art and open issues. IJMIR **1**(3), 173–190 (2012). doi:10.1007/s13735-012-0004-6
29. Rossant, F., Bloch, I.: Robust and adaptive OMR system including fuzzy modeling, fusion of musical rules, and possible error detection. EURASIP J. Appl. Signal Process. **2007**(1), 160–160 (2007)
30. dos Santos Cardoso, J., Capela, A., Rebelo, A., Guedes, C., Pinto da Costa, J.: Staff detection with stable paths. IEEE Trans. Pattern Anal. Mach. Intell. **31**(6), 1134–1139 (2009)
31. Su, B., Lu, S., Pal, U., Tan, C.: An effective staff detection and removal technique for musical documents. In: 2012 10th IAPR International Workshop on Document Analysis Systems (DAS), pp. 160–164 (2012). doi:10.1109/DAS.2012.16
32. Tardón, L.J., Sammartino, S., Barbancho, I., Gómez, V., Oliver, A.: Optical music recognition for scores written in white mensural notation. EURASIP J. Image Video Process. **2009** (2009). doi:10.1155/2009/843401
33. Vapnik, V.N.: Statistical Learning Theory, 1st edn. Wiley, Hoboken (1998)
34. Visani, M., Kieu, V., Fornes, A., Journet, N.: ICDAR 2013 Music Scores Competition: Staff Removal. In: 2013 12th International Conference on Document Analysis and Recognition (ICDAR), pp. 1407–1411 (2013). doi:10.1109/ICDAR.2013.284

Chapter 5

Improving classification using a Confidence Matrix based on weak classifiers applied to OCR

Rico-Juan, J. R. and Calvo-Zaragoza, J. (2015). Improving classification using a confidence matrix based on weak classifiers applied to OCR. *Neurocomputing*, 151:1354–1361

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Improving classification using a Confidence Matrix based on weak classifiers applied to OCR

Juan Ramon Rico-Juan, Jorge Calvo-Zaragoza*

Software and Computing Systems, University of Alicante, Carretera San Vicente del Raspeig s/n, E-03071 Alicante, Spain

ARTICLE INFO

Article history:
Received 27 May 2014
Received in revised form
3 September 2014
Accepted 24 October 2014
Communicated by Huiyu Zhou
Available online 5 November 2014

Keywords:
Confidence Matrix
Posterior probability
Weak classifiers
Feature spaces

ABSTRACT

This paper proposes a new feature representation method based on the construction of a Confidence Matrix (CM). This representation consists of posterior probability values provided by several weak classifiers, each one trained and used in different sets of features from the original sample. The CM allows the final classifier to abstract itself from discovering underlying groups of features. In this work the CM is applied to isolated character image recognition, for which several set of features can be extracted from each sample. Experimentation has shown that the use of CM permits a significant improvement in accuracy in most cases, while the others remain the same. The results were obtained after experimenting with four well-known corpora, using evolved meta-classifiers with the *k*-Nearest Neighbor rule as a weak classifier and by applying statistical significance tests.

1. Introduction

Classification systems have been widely studied in pattern recognition tasks. The classical classification scheme is based on a sequential model that consists in extracting features from a sample, using a classification technique and obtaining a final hypothesis [7]. This scheme has been exploited in order to attain fairly complex techniques with which to improve classification accuracy, such as Artificial Neural Networks [11] or Support Vector Machines [4]. The evolution in this field has led to the development of new schemes in supervised learning. For example, a new classification scheme has emerged based on the assumption that it is more robust to combine a set of simple hypotheses than to use just one complex hypothesis [13].

This scheme can be viewed from different perspectives according to the means used to combine decisions. On one hand, there are algorithms that combine the scores of individual classifiers (usually weak classifiers) to produce a final score, and which are commonly referred to as ensemble classifiers. A wide analysis of this kind of algorithms can be found in Kuncheva [14]. On the other hand, another approach has recently been proposed which uses several dissimilarity measures from the samples to obtain different scores that are subsequently combined. This approach has several approximations, which are described in Pekalska and Duin [19]. A more recent paper[12] proposes refinements of error correction for fusion strategies in the classification.

In this paper we propose a kind of combination of the two aforementioned schemes: first, each weak classifier –in our case classifiers based on the nearest neighbor rule (NN) [6]– provides the probability of belonging to each class. All these probabilities are in turn combined to form a Confidence Matrix (from here on referred to as CM) which is used as an input to a final classifier.

The construction of this matrix can be viewed as the same basic idea as that of the *Stacking* [32] family algorithms. These algorithms are based on the generation of meta-features. Each feature represents the *a posteriori* probability of the actual prototype belonging to each class depending on each weak classifier. In its initial version, Stacking is used to obtain the probability of each possible class using all the weak classifiers, and then it classifies the samples in the space of meta-features, principally through the use of a multi-linear regression approach. An evolved version known as Stacking-C [26] generates these meta-features class by class, adding an additional feature to indicate whether the sample belongs to the class being treated.

The construction of the CM just requires different groups of features to be extracted from the original signal. Each one of these groups has to be used with a different weak classifier, so that the final meta-classifier does not have to discover these underlying points of view by itself. Hence, our work establishes a case of study in which the CM representation is applied to the OCR task since this kind of data is known to allow several ways of extracting features [22].

In this paper, some meta-classifiers are tested by using original features and meta-features (CM). The experiments show how the power of this technique lies in the mapping of features onto meta-features. When the meta-feature space is used, any advanced classifier can be applied to recognize the samples without being

* Corresponding author.
E-mail addresses: JuanRamonRico@dlsi.ua.es (J.R. Rico-Juan), jcalvo@dlsi.ua.es (J. Calvo-Zaragoza).

http://dx.doi.org/10.1016/j.neucom.2014.10.058
0925-2312/© 2014 Elsevier B.V. All rights reserved.

limited to a set of algorithms based on linear regression. That is, the intention of this paper is to address the construction of a CM that can be used at the meta-feature level and combined with any meta-classifier. As discussed in the Experimental section, the accuracy of the results obtained using CM is, in most cases, significantly better or, at worst, the same as when using the original features. These empirical results are obtained by means of several experiments using different corpora, various evolved meta-classifiers and statistical analysis techniques.

The remainder of the paper is structured as follows: Section 2 describes the construction of the Confidence Matrix. Section 3 details the experimental setup. Section 4 shows the results obtained. The paper ends in Section 5 with our conclusions and future work.

2. A new classification scheme based on confidence measures

This section presents a new classification scheme based on the generation of a Confidence Matrix. This section will present a generic construction of this representation regardless the specific task or set of features. The application of this construction for its use in the OCR task will be addressed in the next section.

If Ω is a set of class labels and D is the set of weak classifiers, then a $|D| \times |\Omega|$ matrix is obtained. This matrix (CM) contains the confidence (represented as probabilities) that the weak classifiers give to each prototype belonging to each class. That is, CM_{ij} represents the probability that the sample belongs to the class Ω_i based on the weak classifier D_j. The CM can thus be viewed as a new feature representation (meta-features) that can be used to feed the final classifier rather than using the original features (see Fig. 1).

When this matrix is used, the final classifier does not need to distinguish the different points of views of the signal. In classical approaches, the final classifier has the responsibility of discovering them by itself. Furthermore, unlike that which occurs with the ensemble classifiers, this new scheme avoids the need to define distinct dissimilarity measures or types of weak classifiers. It is only necessary to group the input characteristics according to their nature, which is often relatively simple for a user with domain expertise.

In order to build the CM, it is necessary to train a set of weak classifiers, each of which is responsible for exploiting one group of features separately. One weak classifier is therefore trained for each set of features, thus producing confidence values that work on the different points of view of the input data. Each weak classifier must generate a vector of confidence values that are grouped to form the final CM (see Fig. 2). Each individual weak classifier can use different methods or measures to estimate the probability. In our case, the same methods are used based on different groups of input features, as will be shown in the Experimental section.

These confidence values should indicate the possibility of the sample belonging to a certain class. Although these confidence values do not have to exactly represent a probability, in this paper the values will be formalized as posterior probabilities. It is thus possible to state that the CM is composed of the likelihood of the sample belonging to each class according to each weak classifier considered.

From the algorithmic point of view, the CM representation entails some interesting advantages: (1) the implementation is very straightforward and it only requires weak classifiers; (2) the pipeline of the algorithm can be easily parallelized so that each weak classifier runs at the same time.

Additionally, there may be some scenarios in which the CM is not only helpful but necessary. For example, when several input signals from the same sample come from different, incompatible structures (e.g. trees, strings or feature vectors). In these cases, scores from weak classifiers trained separately with each kind of structure can be easily combined (early fusion) within the CM representation.

Note, however, that this new scheme does not produce a final decision. It merely maps the input features into another space (meta-features). This signifies that it is necessary to use an algorithm (meta-classifier) that employs the CM to make a decision.

From this point of view, CM representation is therefore related to both early and late fusions. Several inputs are combined with the CM structure which can be seen as an early fusion for the final meta-classifiers at the same time it is a late fusion from the weak classifiers point of view.

Our assumption is that it is usually simple to provide some good weak classifiers (at least, better than random decisions). If some good weak classifiers are provided, it is expected that the final meta-classifiers can detect which meta-features are most promising to use in the decision. Thus, if some weak classifier is giving bad meta-features, it is also expected that the final classifier can detect that it is better to avoid their scores.

Since we only provide a new representation of the input, it should be emphasized that the main goal is to prove that the use of the CM either improves on or maintains the accuracy obtained without it. To this end, a series of meta-classifiers for its use as a final algorithm will be considered in the experimentation.

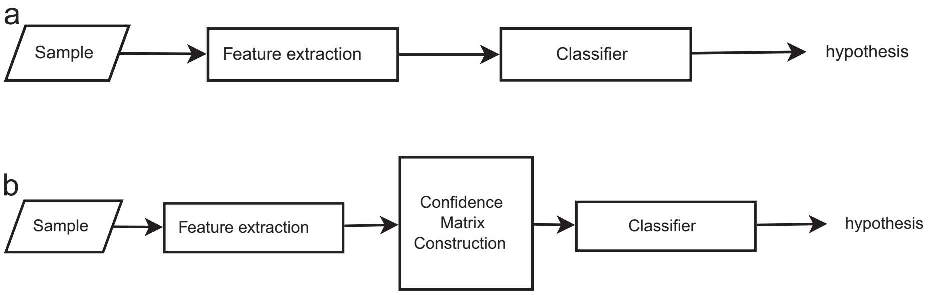


Fig. 1. Classification schemes with and without Confidence Matrix (CM). (a) Classification scheme without CM and (b) classification scheme with CM.

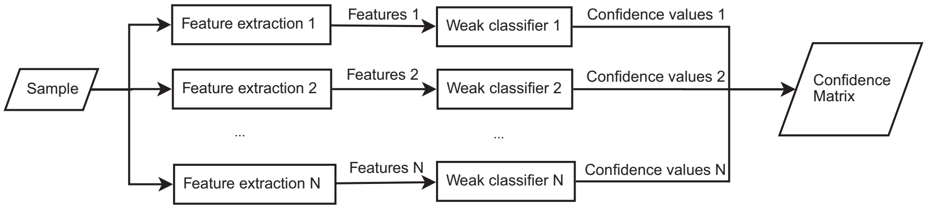


Fig. 2. Construction of the Confidence Matrix.

3. Experimental setup

In this paper, various experiments are carried out to compare the benefits attained by the use of the CM. The experiments are focused on classifying binary OCR images, using four different datasets:

- The NIST SPECIAL DATABASE 3 of the National Institute of Standards and Technology, from which a subset of the upper case characters (26 classes, [A–Z]) was randomly selected: 6500 images (250 examples per class) from 500 writers.
- The MNIST dataset of handwritten digits (10 classes, [0–9]) [16]: 60,000 training images and 10,000 test images. In our experiments, both sets will be mixed.
- The United States Postal Office handwritten digit (10 classes) dataset [9]: 9298 images.
- The MPEG-7 shape silhouette dataset (Core Experiment CE-Shape-1 part B, 70 classes) [15]: 1400 images. Although this dataset is not composed by characters, the images contained are very similar to that of previous datasets. Thus, it is included in the experiments to test the technique proposed.

3.1. Feature extraction from binary images

As it was mentioned previously, the inclusion of the CM is projected for tasks in which it is possible to extract different groups of features. The same feature extraction as that detailed in [22] was therefore used. Nevertheless, a brief explanation is provided in the following paragraphs.

As depicted in Section 2, the main idea is to obtain different kinds of features, each of which will be used on a specific weak classifier. A preprocessing stage is performed first, during which the image is binarized using a global histogram threshold algorithm [18]. A morphological closing filter [27] is then applied in order to correct any gaps and spurious points that may have appeared. In the next step, the character is located in the image and the region of interest (ROI) is selected. The ROI is divided into a sub-structure of smaller regions in order to extract local features. The number of sub-regions must be fixed according to each dataset (see Table 1).

Once the image has been preprocessed, the feature extraction takes place. Three types of features are extracted:

- *Foreground features*: A vector with the number of foregrounds pixels in each of the image sub-regions is produced.
- *Background features*: The background feature extraction is based on that of [30]. It computes four projections (up, down, left, and right) which are considered for each pixel in the image. When any of these projections touches the foreground object, the number associated with that pixel increases by one unit. It is thus possible to distinguish four different categories of background pixels, according to their projection values (1, 2, 3, 4). A fifth category is also added in order to provide more information: there are two situations that are similar in geometry but are totally different from a topological point of view. Our algorithm therefore assigns a value of 5 to the category if the pixel lies in an isolated background area, signifying that five vectors are extracted as features, one for each pixel projection category. Each vector represents the number of pixels with the particular category in each image sub-region.
- *Contour features*: The object contour is encoded by the links between each pair of 8-neighbor pixels using 4-chain codes in a manner proposed by [17]. These codes are used to extract four vectors (one for each direction), and the number of each code is counted in each image sub-region.

Table 1
Mean error rates (standard deviation) of 4-cross-validation preliminary experiment with different sub-region sizes. Results for each database using the MACP (Maximum Average Class Probability) algorithm are shown.

Subregion	NIST	MNIST	USPS	MPEG-7
02 × 02	39.7 (1.6)	35.5 (1.5)	38.5 (1.7)	41.2 (1.8)
03 × 03	19.1 (1.4)	21.2 (1.3)	23.1 (1.5)	26.0 (1.6)
04 × 04	10.1 (1.2)	16.4 (1.1)	16.4 (1.2)	12.4 (1.1)
05 × 05	8.5 (0.9)	13.1 (1.0)	10.2 (1.0)	9.3 (1.0)
06 × 06	**8.3** (**1.0**)	**11.2** (**0.9**)	6.1 (0.9)	**8.1** (**0.9**)
07 × 07	8.5 (1.0)	12.0 (1.0)	**4.2** (**0.8**)	9.2 (1.0)
08 × 08	8.7 (0.9)	13.3 (1.0)	5.3 (0.9)	10.0 (1.0)
09 × 09	10.0 (1.0)	15.4 (1.1)	6.8 (1.1)	12.5 (1.1)

3.2. Weak classifiers

In order to construct the CM, a set of weak classifiers with which to map each group of features onto confidence values is needed. In this case, a formula based in the Nearest Neighbor (NN) [6] rule was used since it is a common, robust and simple technique with which to produce a weak classifier. As discussed in Section 2, one weak classifier per group of features should be generated. Each weak classifier is trained using a leaving-one-out scheme: each single sample is isolated from the training set T and the rest are used in combination with the NN to produce the confidence values. The formula detailed below is used in Rico-Juan and Iñesta [23] and is inspired by Pérez-Cortés et al. [20]. If x is a training sample, then the confidence value for each class $w \in \Omega$ is based on the following equation:

$$p(w|x) = \frac{1}{\min_{x' \in T_w, x \neq x'} d(x, x') + \epsilon} \tag{1}$$

where T_w is the training set for w label and ϵ is a non-zero value provided to avoid infinity values. In our experiments, the dissimilarity measure $d(\cdot,\cdot)$ is the Euclidean distance. After calculating the probability for each class, the values are normalized such that $\sum_{w \in \Omega} p(w|x) = 1$. Once each training sample has been mapped onto the CM, the samples can be used in the test phase.

3.3. Evolved meta-classifiers

Once the CM has been calculated as explained in the previous section, it must be used in combination with a classifier to output a hypothesis. In this work we shall use well-known evolved meta-classifiers. The intention of this measure is to avoid the possibility that improvements in the results may be caused by the use of over simple classifiers. The underlying idea of meta-classification is to solve the problem of combining classifiers. A meta-classifier is in charge of gathering individual classification decisions in order to combine them into a unique final decision. We shall use the following three meta-classifiers: Maximum Average Class Probability, Stacking-C, and Rotation Forest.

Note that we are not trying to outperform the existing late fusion techniques. Since these three classifiers perform some kind of late fusion by their own, our intention is just to find out whether our CM representation can improve the performance achieved with classical feature vectors.

In addition to these three meta-classifiers, Support Vector Machines and Multi-Layer Perceptron algorithms are also included in the experimentation. All these techniques will be experimentally compared with and without the use of the CM.

3.3.1. Maximum average class probability

This meta-classifier is based on combining decisions by using the voting methods from the weak classifier hypothesis with the average rule [14]. In this case, each weak classifier classifies a new sample by

Table 2
Comparison of NIST classification average error rate per class with 4-cross-validation, comparing the methods with and without CM.

Dataset	RoF RaF		RoF J48		MACP		Stacking		SVM		MLP	
	With	Without	With	Without	With	Without	With	Without	With	Without	With	Without
A	3.5	4.0	5.0	6.5	7.0	6.0	9.5	8.5	9.0	18.5	1.0	10.5
B	6.5	6.0	6.5	10.0	6.0	5.5	7.5	14.5	11.5	14.5	2.5	58.0
C	4.0	3.0	4.5	6.0	5.5	4.5	8	8.5	11.5	12.5	4.0	8.5
D	10.5	14.0	11.0	18.5	14.5	11.5	16.5	22	33.5	43.5	8.0	61.5
E	2.5	6.0	4.0	11.0	5.0	6.0	6.5	9	1.5	20.5	6.0	35.5
F	2.5	3.0	4.5	4.0	2.0	3.5	3	5.5	6.0	14.0	3.5	30.0
G	7.5	9.0	8.0	10.0	9.5	8.5	10	17	8.5	25.0	7.0	32.5
H	3.5	7.5	6.5	6.5	5.0	7.5	5.5	13.5	2.0	22.0	4.5	33.5
I	6.5	10.5	4.5	8.0	6.0	7.5	9	11	42.5	45.5	5.5	72.5
J	5.0	6.0	8.0	5.0	7.5	5.5	8	9.5	13.5	45.0	5.0	58.0
K	4.0	7.0	5.0	9.0	9.5	9.0	7.5	18	5.0	26.0	4.0	46.5
L	3.5	4.0	5.5	6.0	4.5	6.0	4	3.5	25.5	28.5	4.5	7.5
M	4.0	2.5	5.5	7.0	5.0	3.5	4.5	6.5	7.0	18.0	5.0	34.5
N	7.0	8.5	10.5	10.0	10.5	8.5	12	20	8.5	38.5	6.0	55.5
O	12.0	13.0	12.0	14.5	18.5	11.0	19.5	15.5	16.5	18.0	10.0	38.0
P	4.0	5.0	5.0	5.0	3.0	3.5	4	5	16.0	13.0	5.0	34.5
Q	13.5	16.0	12.0	15.0	7.5	15.0	13	24	25.0	39.0	5.5	16.0
R	5.0	7.0	7.5	6.5	9.5	9.0	9	13	20.0	26.5	4.0	55.5
S	5.0	6.5	7.0	8.0	4.5	9.5	8	10.5	9.0	14.0	4.5	7.0
T	1.0	2.5	2.0	2.5	1.0	2.0	3.5	2.5	14.5	20.0	2.0	52.5
U	6.5	10.0	6.5	12.0	10.5	12.0	15	11	18.5	29.5	8.0	59.0
V	8.5	8.5	11.0	10.0	10.5	8.0	10.5	10	22.5	34.5	9.0	38.5
W	7.0	6.5	6.5	8.0	4.5	6.5	5	6.5	7.5	23.5	4.0	12.0
X	7.5	8.5	9.5	11.5	4.5	11.0	8.5	16.5	17.0	25.0	4.5	33.0
Y	8.0	12.5	9.0	10.5	6.5	8.5	7	14	12.0	27.5	7.0	57.0
Z	3.0	2.0	3.5	3.5	3.0	3.0	5	3.5	11.0	13.5	4.0	30.0
Avg. error	**5.8**	7.3	**6.9**	8.6	**7.0**	7.4	**8.4**	11.5	**14.4**	25.2	**5.2**	37.6

The values in bold type represent the best result obtained by each method (with or without CM). The underlined value emphasize the best dataset average error.

Table 3
Comparison of MNIST classification average error rate per class with 4-cross-validation, comparing the methods with and without CM.

Dataset	RoF RaF		RoF J48		MACP		Stacking		SVM		MLP	
	With	Without	With	Without	With	Without	With	Without	With	Without	With	Without
0	1.0	5.0	1.0	4.5	3.0	1.0	2.5	1.0	8.3	5.9	0.9	50.9
1	3.5	2.0	3.5	3.0	3.0	4.5	3.0	2.0	11.9	20.6	3.3	58.6
2	7.0	8.5	7.5	8.0	8.0	15.5	11.5	8.5	10.4	27.3	3.4	10.1
3	9.5	13.0	11.5	12.0	14.0	14.5	12.0	15.0	20.3	15.0	6.8	53.9
4	4.5	7.5	6.5	8.5	4.0	9.0	5.0	8.0	12.1	22.3	3.3	76.3
5	7.0	13.0	9.5	13.0	9.5	17.0	11.5	17.0	6.9	26.5	4.1	78.5
6	2.5	3.0	3.0	3.5	4.5	3.0	6.5	4.0	10.4	7.5	1.6	1.9
7	5.0	8.5	4.5	8.0	10.0	9.5	9.0	12.0	43.0	30.9	3.6	3.8
8	7.0	13.5	9.5	16.0	13.5	17.0	12.5	15.5	9.6	22.0	5.5	56.9
9	7.5	12.5	8.0	10.0	11.0	10.0	10.5	8.5	13.8	21.0	8.3	80.0
Avg. error	**5.5**	8.7	**6.5**	8.7	**8.1**	10.1	**8.4**	9.2	**14.7**	19.9	**4.1**	47.1

The values in bold type represent the best result obtained by each method (with or without CM). The underlined value emphasize the best dataset average error.

Table 4
Comparison of USPS classification average error rate per class with 4-cross-validation, comparing the methods with and without CM.

Dataset	RoF RaF		RoF J48		MACP		Stacking		SVM		MLP	
	With	Without	With	Without	With	Without	With	Without	With	Without	With	Without
0	3.0	4.5	3.0	3.9	3.0	2.0	5.0	2.0	0.6	0.0	1.6	37.2
1	4.5	7.0	5.5	5.5	3.9	3.9	3.9	5.5	1.5	65.8	0.5	27.7
2	2.5	3.9	3.0	3.5	3.5	3.5	3.9	3.5	0.2	98.7	1.3	40.8
3	4.0	6.0	6.9	6.5	4.5	4.5	4.0	5.0	82.7	90.7	1.8	34.9
4	4.0	6.5	4.5	7.0	4.5	4.5	5.0	5.5	34.2	99.7	1.8	99.7
5	3.0	5.0	3.5	6.0	5.0	4.0	4.5	4.0	79.7	79.7	2.1	79.7
6	1.5	3.0	2.5	2.0	1.5	1.0	1.5	0.0	5.7	84.7	1.0	51.1
7	3.5	4.0	4.5	4.5	3.5	3.5	2.5	4.5	25.1	73.2	1.1	31.1
8	2.0	4.5	2.0	4.0	4.0	3.5	2.5	3.5	82.7	82.7	1.2	19.9
9	1.5	3.0	2.0	3.0	4.0	1.5	2.5	2.0	42.1	88.2	1.5	18.9
Avg. error	**2.9**	3.7	**3.7**	4.6	3.7	**3.2**	3.5	3.5	**35.4**	76.3	**1.4**	44.1

The values in bold type represent the best result obtained by each method (with or without CM). The underlined value emphasize the best dataset average error.

computing the *a posteriori* probability for each class. The class that obtains the maximum average from among these values is selected. This method was chosen as baseline because of the good results obtained previously for this type of tasks [22]. In this previous work, the classification error rate was lower than 1-NN technique applied to each group of individual features (image, background, contour) and than a 1-NN technique gathering as input the three groups of features.

3.3.2. *Stacking-C*

Given that our classification scheme is based on the main idea of Stacking algorithms, we have included this algorithm to prove the improvement that can be obtained by means of the CM. We have selected one of the most successful algorithms from this family: Stacking-C [26], an extension to Stacking with which to accurately address multi-label classification problems.

Table 5
Comparison of MPEG-7 classification average error rate per class with 4-cross-validation, comparing the methods with and without CM.

Dataset	RoF RaF		RoF J48		MACP		Stacking		SVM		MLP	
	With	Without	With	Without	With	Without	With	Without	With	Without	With	Without
Bat	6.3	6.3	18.8	18.8	6.3	6.3	18.8	18.8	25.0	31.3	6.3	43.8
Beetle	12.5	0.0	0.0	12.5	12.5	0.0	12.5	12.5	25.0	37.5	0.0	50.0
Bird	43.8	50.0	43.8	56.3	25.0	50.0	50.0	37.5	68.8	56.3	43.8	81.3
Butterfly	12.5	18.8	25.0	56.3	31.3	12.5	12.5	18.8	31.3	6.3	18.8	56.3
Camel	0.0	12.5	6.3	12.5	6.3	6.3	6.3	0.0	25.0	0.0	12.5	87.5
Cattle	0.0	0.0	0.0	6.3	0.0	0.0	0.0	0.0	25.0	12.5	6.3	68.8
Chicken	43.8	37.5	37.5	43.8	18.8	25.0	25.0	18.8	81.3	68.8	43.8	100.0
Classic	0.0	12.5	6.3	25.0	0.0	0.0	0.0	0.0	50.0	18.8	12.5	43.8
Comma	0.0	0.0	0.0	0.0	6.3	0.0	0.0	0.0	0.0	25.0	0.0	6.3
Crown	6.3	6.3	6.3	0.0	6.3	6.3	6.3	6.3	6.3	6.3	6.3	37.5
Cup	6.3	6.3	6.3	12.5	6.3	0.0	6.3	6.3	12.5	6.3	12.5	6.3
Deer	31.3	43.8	31.3	56.3	43.8	37.5	31.3	37.5	62.5	68.8	50.0	87.5
Device0	0.0	0.0	0.0	0.0	6.3	0.0	0.0	0.0	31.3	37.5	81.3	25.0
Device2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	6.3	43.8	31.3	0.0	100.0
Device3	0.0	0.0	0.0	0.0	0.0	0.0	6.3	0.0	43.8	62.5	0.0	31.3
Device4	0.0	6.3	0.0	0.0	0.0	0.0	0.0	0.0	18.8	37.5	0.0	43.8
Device6	0.0	0.0	0.0	0.0	0.0	0.0	12.5	0.0	0.0	18.8	0.0	0.0
Device9	6.3	0.0	6.3	6.3	0.0	0.0	6.3	12.5	31.3	6.3	0.0	68.8
Dog	6.3	0.0	12.5	0.0	6.3	6.3	12.5	18.8	18.8	50.0	18.8	43.8
Elephant	31.3	25.0	25.0	37.5	31.3	12.5	25.0	6.3	56.3	43.8	37.5	56.3
Fish	18.8	18.8	18.8	18.8	12.5	12.5	12.5	12.5	25.0	18.8	18.8	81.3
Flatfish	6.3	12.5	6.3	6.3	6.3	6.3	6.3	6.3	6.3	6.3	6.3	25.0
Fly	25.0	12.5	25.0	31.3	6.3	25.0	25.0	37.5	37.5	12.5	37.5	100.0
Fork	12.5	12.5	31.3	18.8	18.8	12.5	18.8	12.5	50.0	31.3	43.8	62.5
Frog	12.5	18.8	18.8	25.0	18.8	12.5	37.5	6.3	31.3	50.0	31.3	37.5
Guitar	12.5	12.5	18.8	18.8	0.0	6.3	18.8	6.3	93.8	43.8	43.8	62.5
Hammer	6.3	12.5	6.3	12.5	6.3	6.3	6.3	6.3	93.8	12.5	12.5	62.5
Horse	37.5	25.0	31.3	25.0	25.0	12.5	56.3	18.8	43.8	75.0	25.0	93.8
Horseshoe	0.0	0.0	0.0	0.0	0.0	0.0	12.5	0.0	0.0	18.8	0.0	68.8
Jar	0.0	12.5	0.0	25.0	0.0	12.5	12.5	25.0	50.0	75.0	6.3	75.0
Key	0.0	6.3	6.3	0.0	6.3	6.3	6.3	6.3	6.3	43.8	6.3	43.8
Lizzard	12.5	18.8	18.8	37.5	37.5	31.3	43.8	25.0	56.3	62.5	37.5	43.8
Lmfish	12.5	31.3	12.5	31.3	18.8	12.5	25.0	12.5	25.0	56.3	31.3	62.5
Misk	0.0	0.0	6.3	0.0	0.0	0.0	0.0	0.0	0.0	37.5	0.0	43.8
Octopus	18.8	25.0	12.5	18.8	12.5	18.8	12.5	12.5	37.5	50.0	18.8	62.5
Pencil	0.0	12.5	0.0	12.5	6.3	0.0	18.8	18.8	100.0	56.3	37.5	100.0
Personal_car	0.0	6.3	0.0	6.3	12.5	0.0	6.3	12.5	31.3	6.3	6.3	87.5
Pocket	6.3	6.3	6.3	6.3	6.3	6.3	6.3	6.3	12.5	12.5	0.0	6.3
Ray	18.8	25.0	0.0	25.0	6.3	0.0	31.3	12.5	25.0	37.5	0.0	37.5
Sea_snake	12.5	31.3	12.5	31.3	25.0	18.8	25.0	43.8	37.5	18.8	25.0	81.3
Shoe	0.0	0.0	0.0	6.3	0.0	0.0	0.0	0.0	12.5	12.5	0.0	62.5
Spoon	31.3	68.8	43.8	50.0	37.5	62.5	43.8	87.5	100.0	75.0	68.8	87.5
Spring	6.3	18.8	18.8	12.5	12.5	18.8	6.3	18.8	87.5	87.5	6.3	68.8
Stef	6.3	6.3	6.3	6.3	6.3	6.3	6.3	6.3	12.5	18.8	6.3	81.3
Tree	12.5	6.3	12.5	6.3	6.3	6.3	12.5	6.3	31.3	100.0	6.3	62.5
Turtle	31.3	31.3	31.3	37.5	6.3	18.8	31.3	12.5	37.5	12.5	18.8	93.8
Watch	6.3	12.5	6.3	18.8	12.5	6.3	6.3	25.0	12.5	6.3	6.3	68.8
Avg. error	**7.3**	9.6	**8.2**	11.9	7.4	**6.9**	10.3	**9.1**	**24.9**	29.8	**12.3**	52.2

The values in bold type represent the best result obtained by each method (with or without CM). The underlined value emphasize the best dataset average error.

Table 6
Summary of the average error rates obtained by the ensembles in the corpus considered.

Dataset	RoF RaF		RoF J48		MACP		Stacking		SVM		MLP	
	With	Without	With	Without	With	Without	With	Without	With	Without	With	Without
NIST	5.8	7.3	6.9	8.6	7.0	7.4	8.4	11.5	14.4	25.2	**5.2**	37.6
MNIST	5.5	8.7	6.5	8.7	8.1	10.1	8.4	9.2	14.7	19.9	**4.1**	47.1
USPS	2.9	3.7	3.7	4.6	3.7	3.2	3.5	3.5	35.4	76.3	**1.4**	44.1
MPEG-7	7.3	9.6	8.2	11.9	7.4	**6.9**	10.3	9.1	24.9	29.8	12.3	52.2

3.3.3. *Rotation Forest*

Rotation Forest (RoF) [24] is an ensemble method that is focused on building accurate and diverse classifiers. It trains a set of decision trees (forest), each of which uses an independent feature extraction. RoF makes use of a base classifier to generate the decision trees. In our work, two alternatives will be considered: C4.5 [21] (J48 implementation [8]) and Random Forest (RaF) [3]. The first alternative is proposed by the original RoF article and by WEKA Data Mining tool [8] as a default parameter, whilst the latter is considered in this paper due to its best performance in our OCR preliminary experiments despite not being a common ensemble in RoF experimentation.

3.3.4. *Support Vector Machines*

Support Vector Machines (SVM) is a common pattern recognition algorithm developed by Vapnik [29]. It seeks for a hyperplane which maximizes the separation (margin) between the hyperplane and the nearest samples of each class (support vectors). The libSVM implementation [8] with Polynomial kernel is used in our experimentation.

3.3.5. *Multi-Layer Perceptron*

Artificial Neural Networks is a family of structures developed in an attempt to mimic the operation of the nervous system to solve machine learning problems. The topology of a neural network can be quite varied. For this work, the common neural network called Multi-Layer Perceptron (MLP) [25] is used, as implemented in [8].

4. Results

The results of each dataset are detailed in the following subsections. Table 6 shows a summary of the average final results. A short discussion about the statistical significance of the results is also developed at the end of this section. The WEKA version 3.6.9 tool [8] has been used for testing RoF and Staking-C algorithms with their default parameters.

Note that our main goal is not to measure the goodness of each considered ensemble but to compare their results with and without using the CM representation proposed.

4.1. *NIST SPECIAL DATABASE 3*

For this dataset, the best number of image sub-regions is 6, signifying that $(6 \times 6)+(6 \times 6) \times 5+(6 \times 6) \times 4 = 360$ features are extracted from each of the samples in this set. The results of the experimentation is shown in Table 2. Upon viewing the results it will be noted that the inclusion of the CM has, on average, improved the results of all the algorithms.

Note that the improvement achieved by using the CM in both SVM and MLP is remarkably high. The latter case is specially interesting since this algorithm has the best error rate when using CM representation, which did not occur without it.

4.2. *MNIST*

The number of optimum image sub-regions in this dataset is also 6, signifying that 360 features are again used. Table 3 details the results of the experiment for this dataset. The results follow a similar trend to those of the NIST dataset. In this case, the improvement achieved by the inclusion of the CM would appear to be even more noticeable. MLP (with CM) was again reported as being the best classifier.

Table 7
Summary of the average (standard deviation) execution time in seconds obtained by the ensembles in the experiments.

Dataset	RoF RaF		RoF J48		MACP		Stacking		SVM		MLP	
	With	Without	With	Without	With	Without	With	Without	With	Without	With	Without
NIST	128.5 (0.5)	116.8 (0.4)	388 (4)	504 (6)	23.0 (0.1)	7.3 (0.4)	3517 (10)	2888 (12)	90 (3)	120 (4)	1650 (10)	3000 (90)
MNIST	23.5 (0.5)	38.0 (0.1)	35 (7)	150 (2)	8.3 (0.4)	4.0 (0.1)	75.3 (0.4)	265.3 (0.4)	85.0 (0.2)	297 (3)	313.7 (0.8)	7700 (90)
MPEG-7	49.3 (0.3)	26.2 (0.4)	224 (4)	97.0 (0.6)	4.3 (0.1)	2.7 (0.1)	923.7 (0.3)	436.2 (0.6)	7.2 (0.1)	15.2 (1.3)	750 (20)	2500 (60)
USPS	122 (2)	167.4 (0.6)	181 (3)	754 (4)	50 (2)	4.1 (0.1)	994 (4)	3845 (5)	73.0 (0.3)	248.1 (1.2)	291.2 (1.3)	794 (5)

Table 8
Wilcoxon statistical significances (p-values) reported in datasets considered with 4-cross-validation, assuming that the classifiers are better with CM than without it. The numbers in parentheses represent the opposite relationship (without CM is better than with it) when the main p-value is close to 1. The values in bold type represent a level of significance that is higher than α=0.95.

Dataset	RoF RaF	RoF J48	MACP	Stacking-C	SVM	MLP
NIST	**0.000004**	**0.000288**	0.404899	**0.000643**	**0.000007**	**0.000004**
MNIST	**0.007649**	**0.001312**	0.168282	0.132416	0.126279	**0.004317**
USPS	**0.0059**	**0.0043**	1 (0.796)	0.9582 (0.9582)	**0.016605**	**0.004317**
MPEG-7	**0.017337**	**0.002064**	1 (0.15168)	1 (0.399268)	**0.001516**	**0.000007**

4.3. USPS

The preliminary results as regards obtaining the best number of image sub-regions is 7 in the case of the USPS dataset, and $(7\times 7)+(7\times 7)\times 5+(7\times 7)\times 4=490$ features are therefore considered. The results of the final experiment are shown in Table 4. This is the first case in which the inclusion of the CM does not improve all the classifiers considered, since *MACP* increases its error rate from 3.2 to 3.7 when using the CM. The other classifiers decrease or maintain their error rates with the CM. Once again, the MLP (with CM) classification achieved the best classification result.

4.4. MPEG-7

As occurred for the NIST and MNIST, the best image sub-region size for the MPEG-7 database is 6, and 360 features are therefore used to classify the samples. Table 5 shows the results of the classification experiment with this database. The results of the datasets in which all the classifiers obtain a perfect classification have been removed owing to the size of the table. Note that some classifiers are enhanced with the inclusion of the CM while others are not. This also occurred in previous databases, but in this case the best classification was obtained for the MACP without CM.

4.5. Statistical significance

The intention of this experimental section is to assess whether the inclusion of the CM can achieve significantly better classification results. We shall therefore use the KEEL [1] software, which contains statistical tools. These tools will allow us to quantify the difference between the results with and without the CM. Specifically, a Wilcoxon 1×1 test was performed. The significance p-values considering all the experiments are shown in Table 8. These values represent the overlap between the two distributions, assuming that the classifiers are better with the CM. We can consider the p-values as a confidence measure for comparison. The significance of a low value is a high probability that the distributions compared are different.

As is shown in this table, most of the values are lower than 0.05, signifying that the use of CM significantly decreases the error rate at a confidence level of 95%. Special cases are reported for the USPS and MPEG-7 datasets using MACP and Stacking-C meta-classifiers for which the significance test yielded that the CM does not improve the accuracy although, in the opposite case (values in parentheses), CM does not worsen it either. This signifies that throughout the experiments the inclusion of the CM has significantly improved, or in the worst cases maintained, the results of the meta-classifiers.

Note the good performance of the MLP classifier when CM is included, given that it is significantly better and obtains some of the lowest error rates in the entire corpora.

Table 7 also shows a summary of the average execution time in order to assess how the inclusion of the CM affects the cost of the ensembles. In general, it is clear that the MACP obtains the lowest time because it computes few calculations, while Stacking-C obtains the highest times. With regard to the results obtained when using or not using CM, there is a considerable amount of variability depending on both the corpus and the algorithm, particularly in the case of decision tree based algorithms.

5. Conclusions

A new approach with which to transform original features into a Confidence Matrix (CM) is presented. This CM consists of a set of posterior probability values which were obtained by using several weak classifiers. This approach enables the features to be transformed into a new space (meta-features), thus allowing the dimensionality of the data (in our case) to be reduced and a more meaningful value to be provided in each dimension. This is expected to help to reduce the error rate of the final classifier.

In our experimentation, 1-NN was used as a weak classifier, and several algorithms (MACP, Stacking-C, RoF RaF, RoF J48) were considered as final meta-classifiers. A 4-fold cross-validation was conducted for each of the four different datasets, and a statistical significance test was also applied in order to measure the effect of including the CM in a comprehensive manner. These tests reported that the inclusion of the CM can significantly improve the results of evolved meta-classifiers. In most of the cases considered, the results were either improved or remained the same. With regard to the execution time, there is no clear trend in the results (see Table 7). The inclusion of the CM decreases the execution time of the most complex ensembles considered (RoF and Stacking-C) in some corpora and increases it in others.

Although our case of study is focused on OCR classification, our intention is that the CM representation can be used in any task as long as several different feature extraction techniques can be applied to the data. The main drawback is that it requires the user to be an active part of the process by extracting these different sets of features. Therefore, an interesting option for future work would be to find a way to extract different groups of features automatically. A bagging system could be explored in order to obtain different weak classifiers trained with a different subset of features. The way in which RoF builds its forests also represents an idea to explore.

One of the main lines of future research is to test our CM against other meta-classification schemes. On one hand, meta-feature extraction methods and early fusion methods could be applied in order to compare its performance with CM. Special interest arises in the comparison with embedding methods [19], since its requirements are quite similar. Nevertheless, embedding methods need to tune correctly some other parameters such as dimensionality or pivot selections so a comprehensive review and experimentation would be necessary. On the other hand, there are several late fusion algorithms (such as those reported in [28,5,10]). It would be of great interest to compare their performance both against our proposal and in combination with it.

The goodness of including the CM could also be tested against other benchmark corpora with lower features per prototype, such as UCI Machine Learning Repository [2]. In our experiments, the original prototypes have between 360 and 490 features, while the datasets of the UCI have between 17 and 35. What is more, the features of the datasets belonging to UCI have some unknown data, and a measure of similarity other than the Euclidean distance such as the HVDM (Heterogeneous Value Difference Metric) [31] would therefore have to be used to deal with these unknown data.

Acknowledgements

This work was partially supported by the Spanish CICyT through the project TIN2013-48152-C2-1-R, the Consejería de Educación de la Comunidad Valenciana through Project PROMETEO/2012/017 and a FPU fellowship (AP2012-0939) from the Spanish Ministerio de Educación Cultura y Deporte.

References

[1] J. Alcalá-Fdez, L. Sánchez, S. García, M.J.D. Jesus, S. Ventura, J.M. Garrell, J. Otero, J. Bacardit, V.M. Rivas, J.C. Fernández, F. Herrera, KEEL: a software tool to assess evolutionary algorithms to data mining problems, Soft Comput. 13 (3) (2009) 307–318.

[2] A. Asuncion, D. Newman, UCI Machine Learning Repository, 2007.

[3] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.

[4] C.J.C. Burges, A tutorial on support vector machines for pattern recognition, Data Mining Knowl. Discov. 2 (1998) 121–167.

[5] S.-F. Chang, Robust late fusion with rank minimization, in: Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) CVPR '12, IEEE Computer Society, Washington, DC, USA, 2012, pp. 3021–3028.

[6] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE Trans. Inf. Theory 13 (1) (1967) 21–27.

[7] R.O. Duda, P.E. Hart, Pattern Recognition and Scene Analysis, John-Wiley and Sons, New York, 1973.

[8] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The weka data mining software: an update, SIGKDD Explor. Newslett. 11 (1) (2009) 10–18.

[9] J. Hull, A database for handwritten text recognition research, IEEE Trans. Pattern Anal. Mach. Intell. 16 (5) (1994) 550–554.

[10] A. Jain, K. Nandakumar, A. Ross, Score normalization in multimodal biometric systems, Pattern Recogn. 38 (12) (2005) 2270–2285.

[11] A.K. Jain, J. Mao, K.M. Mohiuddin, Artificial neural networks: a tutorial, Computer 29 (3) (1996) 31–44.

[12] S. Kim, R. Duin, A combine-correct-combine scheme for optimizing dissimilarity-based classifiers, in: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, 2009, pp. 425–432.

[13] J. Kittler, M. Hatef, R.P.W. Duin, J. Matas, On combining classifiers, IEEE Trans. Pattern Anal. Mach. Intell. 20 (1998) 226–239.

[14] L.I. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms, John Wiley & Sons, 2004.

[15] L.J. Latecki, R. Lakämper, U. Eckhardt, Shape descriptors for non-rigid shapes with a single closed contour, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2000, pp. 424–429.

[16] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, in: Intelligent Signal Processing, IEEE Press, Montreal, Canada, pp. 306-351.

[17] H. Oda, B. Zhu, J. Tokuno, M. Onuma, A. Kitadai, M. Nakagawa, A compact on-line and off-line combined recognizer, in: Tenth International Workshop on Frontiers in Handwriting Recognition, vol. 1, 2006, pp. 133–138.

[18] N. Otsu, A threshold selection method from gray-level histograms, IEEE Trans. Syst. Man Cybern. 9 (1) (1979) 62–66.

[19] E. Pekalska, R. Duin, The Dissimilarity Representation for Pattern Recognition: Foundations and Applications, World Scientific Pub Co Inc, NJ, USA, 2005.

[20] J.C. Pérez-Cortés, R. Llobet, J. Arlandis, Fast and accurate handwritten character recognition using approximate nearest neighbours search on large databases, in: F.J. Ferri, J.M. Iñesta, A. Amin, P. Pudil (Eds.), Advances in Pattern Recognition, Lecture Notes in Computer Science, vol. 1876, Springer-Verlag, Berlin, 2000, pp. 767–776.

[21] Salzberg, S. (1994). C4.5: Programs for machine learning by J. Ross Quinlan. Morgan Kaufmann publishers, inc., 1993. Machine Learning, 16(3):235-240.

[22] J.R. Rico-Juan, J.M. Iñesta, Normalisation of confidence voting methods applied to a fast handwritten OCR classification, in: M. Kurzynski, E. Puchala, M. Wozniak, A. Zolnierek (Eds.), Computer Recognition Systems 2, Advances in Soft Computing, vol. 45, Springer, Wroclaw, Poland, 2007, pp. 405–412.

[23] J.R. Rico-Juan, J.M. Iñesta, Confidence voting method ensemble applied to off-line signature verification, Pattern Anal. Appl. 15 (2) (2012) 113–120.

[24] J. Rodriguez, L. Kuncheva, C. Alonso, Rotation forest: a new classifier ensemble method, IEEE Trans. Pattern Anal. Mach. Intell. 28 (10) (2006) 1619–1630.

[25] F. Rosenblatt, Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. Report Cornell Aeronautical Laboratory, Spartan Books, 1962.

[26] A.K. Seewald, How to make stacking better and faster while also taking care of an unknown weakness, in: Proceedings of the Nineteenth International Conference on Machine Learning ICML '02, Morgan Kaufmann Publishers Inc, San Francisco, CA, USA, 2002, pp. 554–561.

[27] J. Serra, Image Analysis and Mathematical Morphology, Academic Press, London, UK, 1982.

[28] O. Terrades, E. Valveny, S. Tabbone, Optimal classifier fusion in a non-Bayesian probabilistic framework, IEEE Trans. Pattern Anal. Mach. Intell. 31 (9) (2009) 1630–1644.

[29] V.N. Vapnik, Statistical Learning Theory, 1 edition, Wiley, NY, USA, 1998.

[30] E. Vellasques, L. Oliveira, Jr., A.B.A. Koerich, R. Sabourin, Modeling segmentation cuts using support vector machines. in: Tenth International Workshop on Frontiers in Handwriting Recognition, vol. 1, 2006, pp. 41–46.

[31] D.R. Wilson, T.R. Martinez, Improved heterogeneous distance functions. J. Artif. Intell. Res. (1997) 1–34.

[32] D.H. Wolpert, Stacked generalization, Neural Netw. 5 (1992) 241–259.

Juan R. Rico-Juan received the M.Sc. degree and Ph.D. degree in Computer Science from the Polytechnic University of Valencia in 1992 and from the University of Alicante in 2001, respectively. He is currently a permanent Lecturer in the Department of Software and Computing Systems at the University of Alicante. He is author of 27 book chapters, seven papers in international journals and 12 papers in international conferences. He has been involved in around 30 research projects (national and international) covering pattern recognition and artificial intelligence lines of work. His main research interests include rank prototypes for selection, quality prototypes for classification, incremental/adaptive algorithms, structured distances, mean string computation algorithms, ensemble classifiers and edition distances.

Jorge Calvo-Zaragoza received his M.Sc. degree in Computer Engineering from the University of Alicante in July 2012. He currently holds a FPU programme fellowship from the Spanish Ministerio de Educación. He is a Ph.D. candidate in the Software and Computing Systems of the University of Alicante, whose research is focused on statistical pattern recognition, soft computing and document analysis.

5.1 Results with handwritten music symbols

Given that this work could be generalized to deal with any kind of isolated symbols, we decided to include experimentation with several datasets so that the paper was useful to a wider audience. We also stated that it was focused to Optical Character Recognition (OCR), understood as any kind of isolated symbol but not only to alphanumerical characters. In fact, one of the dataset used contains silhouettes of general shapes (MPEG-7).

Considering the scope of this thesis, our main intention was to use this approach to classify images of isolated music symbols. Nonetheless, reviewers considered that it was better to restrict ourselves to the use of well-known datasets. Therefore, the paper finally lacked of results from isolated music symbols.

That is why this section includes the results obtained using the Handwritten Online Music Symbols (HOMUS) dataset. HOMUS is a labelled set of isolated symbols written using an electronic pen. As a part of this thesis, the dataset is described thoroughly in Chapter 6.

From the shape written by a user, an image of the symbol can be rendered so that a corpus of images showing isolated music symbols can be obtained. Table 5.1 shows the results of applying the strategy developed in this chapter to this set. The subregions considered per image are 8. For the sake of compactness, average results are directly reported.

Classifier	Original features	CM representation
RoF RaF	18.9 ± 0.9	15.4 ± 0.7
RoF J48	20.8 ± 0.8	16.6 ± 0.8
MACP	17.4 ± 1.4	19.0 ± 0.7
StackingC	19.2 ± 0.9	18.9 ± 1.1
MLP	75.4 ± 4.7	24.2 ± 4.3
SVM	14.6 ± 0.9	11.8 ± 0.7

Table 5.1: Classification average error rate (± standard devation) over HOMUS dataset with 4-cross-validation, comparing the classifiers with and without CM representation.

It can be seen that the premise demonstrated in the paper is still applicable for isolated music symbols: generally, it is more profitable to use the Confidence Matrix rather than the raw set of features.

Chapter 6

Recognition of Pen-Based Music Notation: The HOMUS Dataset

Calvo-Zaragoza, J. and Oncina, J. (2014). Recognition of pen-based music notation: The HOMUS dataset. In *22nd International Conference on Pattern Recognition, ICPR 2014, Stockholm, Sweden, August 24-28, 2014*, pages 3038–3043

Recognition of Pen-Based Music Notation: the HOMUS dataset

Jorge Calvo-Zaragoza and Jose Oncina
Department of Software and Computing Systems
University of Alicante
Spain
Email: {jcalvo,oncina}@dlsi.ua.es

***Abstract*—A profitable way of digitizing a new musical composition is by using a pen-based (online) system, in which the score is created with the sole effort of the composition itself. However, the development of such systems is still largely unexplored. Some studies have been carried out but the use of particular little datasets has led to avoid objective comparisons between different approaches. To solve this situation, this work presents the Handwritten Online Musical Symbols (HOMUS) dataset, which consists of 15200 samples of 32 types of musical symbols from 100 different musicians. Several alternatives of recognition for the two modalities –online, using the strokes drawn by the pen, and offline, using the image generated after drawing the symbol– are also presented. Some experiments are included aimed to draw main conclusions about the recognition of these data. It is expected that this work can establish a binding point in the field of recognition of online handwritten music notation and serve as a baseline for future developments.**

I. Introduction

Composing music with pen and paper is still a common procedure. However, there may be several reasons for exporting a music score to a digital format: storage, distribution and reproduction; using its information in the search of musical pieces; grouping of styles and detection of plagiarism; or for building digital libraries. Conventional digital score editors put musical symbols on a score by using *point and click* actions with the mouse. These tools represent a tedious effort for the user, leading to consume a lot of time. The use of digital instruments seems a more comfortable alternative. Digital instruments (such as a MIDI piano) can be connected directly to the computer and transfer the information while playing the musical piece. However, this type of transcription is not error-free and rarely catch all the nuances that may contain a score. Moreover, the music sheet can be scanned in order to use an automatic score transcription tool –commonly referred as Optical Music Recognition (OMR) systems [1]–. This option represent an effortless alternative for the user. Unfortunately, OMR systems are far from achieving accurate transcriptions, especially for handwritten scores [2]. Thus, the transcription has to be corrected afterwards.

Although one of the above methods can be used, it is more profitable digitizing the score at the same time the composer writes. In this way, the score is digitized with the sole effort of the composition itself. With an online transcription system, many of the problems above discussed can be avoided, plus additional advantages (e.g., the ability to quickly reproduce the current composition). Furthermore, recognition of this kind of musical symbols could have use in other contexts. For instance, it is feasible to think of a scenario in which an OMR system allows corrections using a digital pen, rather than having to use the conventional mechanism of a score editor. This approach has been already applied to Handwritten Text Recognition [3].

Some previous studies have been carried out but this field still remains largely unexplored. One of the major absences is a dataset that serve as reference for research. All the previous works have worked with its own dataset and its own set of musical symbols. Therefore, comparative studies to know which approaches perform better than others have not been conducted so it is still unclear what is the current status of the research. The present work aims to set a reference point for research on recognition of online handwritten musical symbols. To this end, a large dataset is provided for free access [1], covering the most used symbols in the composition of musical scores. To establish the first baseline, experimentation with well-known pattern recognition algorithms is presented so that more information about the dataset can be known such as the difficulty of the recognition task or which techniques seem more promising. It is also expected that the results can serve as baseline for future comparisons and developments.

The rest of the paper is structured as follows: Section II describes the nature of the recognition of online handwritten music notation. The description of the dataset is shown in Section III. Section IV presents some baseline techniques for this dataset. Experiments are presented in Section V. Finally, conclusions are drawn in Section VI.

II. Recognition of Pen-Based Handwritten Music Notation

Over decades, much research has been devoted to the development of friendly music score editors. Despite all these efforts, there is still no satisfactory solution. The emergence of tablet computer devices has open new avenues to approach this problem. With these devices, a musician can compose its music on a digital score using an electronic pen and have it effortlessly digitized.

The recognition of online (or pen-based) handwritten music notation task is defined as recognition of musical symbols at the time they are being written. The great variability in the manner of writing the musical symbols is the main difficulty to overcome. Figure 1 shows some examples of handwritten musical symbols from different musicians.

[1] The dataset is available at `http://grfia.dlsi.ua.es/homus/`

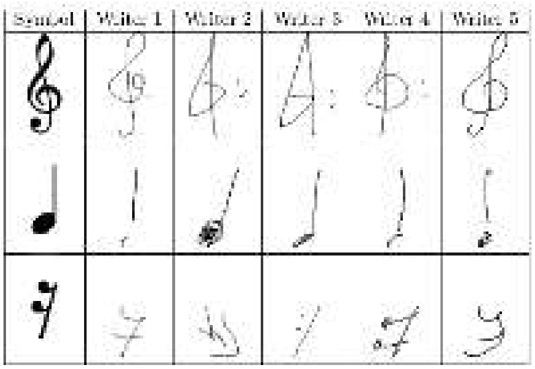


Fig. 1. Some examples of variability in handwritten musical symbols.

This variability is also a problem in OMR systems, but this scenario offers important advantages with respect to them: the staff lines (one of the main issues in offline OMR systems) do not interfere in the recognition since they are handle by the underlying system, the symbol detection could be intrinsically performed somehow, and the information about how the strokes are drawn is available.

These strokes –considered as the shape between pen-down and pen-up actions– produce an ordered set of points, which indicate the path followed by the pen. Similarly, each symbol can be drawn by one or more strokes. But not only this information can be extracted. An image of the shape itself can also be used for the classification (as it would be done in offline recognition). This modality gives another perspective of the symbol and it is more robust against the speed of the user, the order followed to draw a symbol and the number of strokes used.

A. Background

The first systems for pen-based recognition of musical scores were based on the use of simple gestures. This is the case of *Presto* system [4], which received as input short gestures that were generally mnemonic of the music symbols. These gestures were processed and translated to the actual musical symbols. With the same idea, Polácek et al. [5] created a new gesture alphabet especially designed for its use in low-resolution devices. The main drawback of these approaches is that they require an adaptation of the user to the gesture alphabet recognized by the system. Subsequently, there were other works that allowed writing symbols in its conventional manner. Miyao and Maruyama [6] based its system on the recognition of primitives (lines, circles, arcs, etc.), using information both the stroke path and the shape drawn. After the recognition, these primitives are combined to reconstruct the musical symbols. A similar approach was used in [7], in which document spatial structures were defined and combined with context-free grammars. However, depending on the musician writing, a musical symbol may consist of different primitives, so that the rules to rebuild the symbols lack the robustness needed to handle the different writing styles. Moreover, systems that have as their objective the recognition of complete musical symbol can also be found. George [8] used the images generated by the digital pen to learn an Artificial Neural Network (ANN) to recognize the symbols. Lee et al. [9] proposed the use of Hidden Markov Models (HMM) for recognition of some of the most common musical symbols using different features of the shape drawn by the pen. These studies have shown that the complete recognition of symbols written in the natural form of music is feasible.

TABLE I. Features of datasets used in previous works.

Work	Classes	Users	Data
George	20	25	4188 images
Miyao and Maruyama	12	11	13801 strokes
Lee et al.	8	1	400 symbols
Our dataset	32	100	15200 symbols

The recognition of online handwritten music notation is still a novel field so it is not yet established guidelines about which types of algorithms perform better. Aforementioned works have performed experiments that were only focused on finding the optimal parameters of the specific algorithm used. Each of them used its own dataset, its own set of musical symbols and its own nature of the input (see Table I), so it is unclear what dataset must be used to evaluate the performance of new approaches. This has hitherto led to a lack of comparative experiments to assess which of the proposed algorithms perform better in this context. To provide a solution to this problem, this work presents the HOMUS dataset, described in the next section.

III. The Handwritten Online Musical Symbols dataset

This section presents the Handwritten Online Musical Symbols (HOMUS) dataset. The objective is to provide a reference corpus for research on the recognition of online handwritten music notation. The dataset is available at `http://grfia.dlsi.ua.es/homus/`.

Analyzing previous works, it was observed that most of them only took into account a small set of the possible musical symbols. In addition, it is important to stress that each musician has its own writing style, as it occurs in handwritten text. Increasing both the set of musical symbols and the number of different writing styles is advisable if reliable results about the recognition of online handwritten music notation are pursued.

Following this way, the HOMUS was built by 100 musicians from the *Escuela de Educandos Asociación Musical l'Avanç* (El Campello, Spain) and *Conservatorio Superior de Música de Murcia "Manuel Massotti Littell"* (Murcia, Spain) music schools, among whom were both music teachers and advanced students. In order to cover more scenarios, some of them were experienced in handwritten music composition while other have few composition experience. Musicians were encouragingly asked to draw the symbols trying not to do it in a perfect manner, but in its own, particular style (which is reflected in the variability shown in Fig. 1). Each of them were asked to draw four times the 32 classes listed in Table II, which has resulted in 15200 samples spread over 38 templates [2]. Each sample of the dataset contains the label and the strokes composing the symbol. These strokes consists of a set of points relative to a coordinate center. Storing the data in this way allows covering all the possibilities considered: the image can be generated from the strokes, every single stroke can be

[2] The eighth, sixteenth, thirty-second, and sixty-fourth note symbols are written twice: right and inverted.

TABLE II. TYPES OF MUSICAL SYMBOLS IN THE HOMUS DATASET.

Note	whole, half, quarter, eighth, sixteenth, thirty-second, sixty-fourth
Rest	whole/half, quarter, eighth, sixteenth, thirty-second, sixty-fourth
Accidentals	flat, sharp, natural, double sharp
Time signatures	common time, cut time, 4-4, 2-2, 2-4, 3-4, 3-8, 6-8, 9-8, 12-8
Clef	G-clef, C-clef, F-clef
Others	dot, barline

extracted easily, and each individual symbol remains isolated. Since the pitch of the notes is based on its position over the staff, it is unnecessary to detect it in the classification, but it may be assigned in a post-processing stage.

It should be noted that not all musical symbols appear in the dataset. Less relevant symbols such as accidentals, ornaments or instrument-specific notation were left out although they could be added to the score with another mechanism (e.g., via a contextual menu). There are other symbols that can not be present because of their unfixed length (such as ties or slurs) for which an alternative mechanism of addition can also be found.

To create the dataset a *Samsung Galaxy Note 10.1* device was used and symbols were written using the stylus *S-Pen*. This device was chosen among the standalone friendly options because of its optimality to work with an e-pen. The device has a resolution of 1280×800 (149 ppi) and a sampling rate of 16 ms (60 fps). An application that request musical symbols to be drawn on an empty staff was developed. The staff was composed of five parallel lines with a line thickness of 3 and an equal staff line spacing of 14. These two values are provided as a reference for possible rescaling since they are the common features for this purpose in OMR systems [10].

In addition to the dataset, this paper is intended to provide a baseline of the classification rate that can be achieved. Some basic techniques to recognize HOMUS samples are described in the next section.

IV. BASELINE TECHNIQUES

In this section, some techniques for the recognition of the samples contained in the HOMUS dataset are presented. The goal is not to achieve high success rates, but provide some notions about the classification of the symbols. It is also expected that experiments identify the most promising techniques to recognize this kind of data and the results can be used as baseline to compare future developments.

The dual nature of the data –using the strokes and using the image– leads us to explore both ways in the classification of the symbols. Classification techniques for each of these modalities are presented in the following subsections.

A. Online Techniques

The online recognition modality uses the strokes made by the pen. These strokes provide information about how the shape has been generated segment by segment. This modality takes advantage of the local information, expecting that a particular musical symbol follows similar paths. Depending on the type of musical symbol and the pace of the user, a greater or lower number of points will be generated. Therefore, each sample has a different dimension. Due to this, most of the conventional techniques based on equal-sized feature vectors can not be applied. For this reason, we will restrict ourselves to the use of the Nearest Neighbor (NN) technique and Hidden Markov Models (HMM).

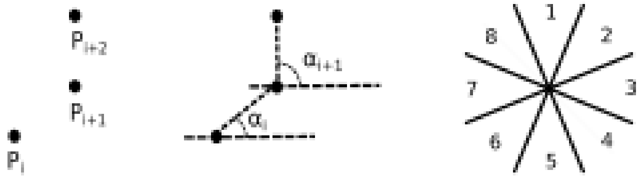


Fig. 2. FCC based on the angle between consecutive points.

1) Nearest Neighbor: Let $X = (x_1, \ldots, x_n)$ be a set of labeled samples and let $x' \in X$ be the sample that minimizes a dissimilarity measure $d(x, x')$ to a test point x. The NN rule [11] assigns to x the label associated with x'. The natural extension of this rule is to use the k-nearest samples (k-NN) and assign the most frequent label. The performance of this rule is strongly related to the dissimilarity measure $d(x, x')$ utilized. Two alternatives are presented in the following lines: Edit Distance with Freeman Chain Codes (FCC) and Dynamic Time Warping (DTW).

Given two strings, the edit distance (or Levenshtein distance) [12] is the minimum number of edit operations –usually insertion, deletion and substitution– to convert one string into another. To use this distance over the samples of the HOMUS, the set of points that represents a musical symbol has to be converted into a string. Codification based on Freeman Chain Code (FCC) [13] is applied. FCC is a typical method to build strings from image contours. It converts each pair of pixels into one code in function of the neighboring direction. In this case, instead of a contour we have a set of points that are not continuous (because of the device sampling rate). This situation can be approached in many ways. Between each pair of points a line that connects them can be interpolated. Thus we can establish a continuous path and applying the conventional FCC afterwards. Moreover, each pair of points can be replaced by a code based on the angle they form (see Fig. 2). These two approaches (FCC and FCC based on angle) will be evaluated experimentally. For symbols with multiple strokes, a specific code is concatenated at the end of each stroke.

On the other hand, DTW is a technique for measuring the dissimilarity between two time signals which may be of different durations. It was firstly used in speech recognition [14] although its use has widely extended to other fields [15], [16]. Let $x = (x_1, x_2, \ldots, x_n)$ and $y = (y_1, y_2, \ldots, y_m)$ be two time series, of length n and m respectively. DTW(i, j) is defined recursively as

$$\begin{cases} 0, & j = 0 \wedge i = 0 \\ \infty, & j = 0 \wedge i > 0 \\ \infty, & i = 0 \wedge j > 0 \\ d(x_i, y_i) + \min \begin{cases} DTW(i-1, j) \\ DTW(i, j-1) \\ DTW(i-1, j-1) \end{cases} , & otherwise \end{cases} \quad (1)$$

and therefore, DTW(x,y) = DTW(n,m). In our case, x_i and y_i are points in a 2-dimensional space. Hence, the distance $d(x_i, y_j)$ is the Euclidean distance between two points. The algorithm is implemented using a dynamic programming scheme, reducing the complexity to $O(nm)$. Details about the intrinsic operation of the algorithm can be found in [14].

2) Hidden Markov Models: Hidden Markov Models (HMM) [17] are statistical models that define an unobservable Markov process generating an observable output sequence. They have been successfully used in online handwritten recognition during the last years [18], [19]. In our work, a *continuous left-to-right* topology is used and the models are trained with the Baum-Welch algorithm [20]. Both the number of states and the number of Gaussian densities in the mixtures are adjusted in preliminary experiments.

Feature extraction is performed as described in the work of Kian et al. [9], which obtained good results for online music symbol recognition.

B. Offline Techniques

After drawing the symbol, an image can be obtained by creating lines between pair of consecutive points. After this, the lines are dilated to simulate a thickness of 3 as used to collect the samples. The information contained in these images provide a new perspective on the recognition and it does not overlap with the nature of the online recognition. The advantage of this representation is that it is robust against different speeds or different orders when writing the symbol.

The baseline showed here is inspired by the work of Rebelo et al. [21] on offline musical symbol recognition. The algorithms considered are k-Nearest Neighbor, Artificial Neural Network, Support Vector Machines and Hidden Markov Models. The images are resized to 20 × 20 and no feature extraction is performed (except for Hidden Markov Models).

1) k-Nearest Neighbor: The k-Nearest Neighbor (k-NN) rule, explained in the previous subsection, can also be used for recognition from images. In this case, a 400-dimensional vector with real values is received as input. To measure the dissimilarity between two samples, the Euclidean distance is used. Some different values for the parameter k will be evaluated experimentally (1, 3 and 5).

2) Artificial Neural Networks: Artificial Neural Networks (ANN) emerged as an attempt to mimic the operation of the nervous system to solve machine learning problems. An ANN comprises a set of interconnected neurons following a certain topology. Further details about ANN can be found in [22].

The topology of a neural network can be quite varied. For this work, the common neural network called Multi-Layer Perceptron (MLP) is used. This topology was also used for the same purpose in the work of George [8]. This kind of networks can be trained with the backpropagation algorithm [23]. The number of hidden states was fixed to 200.

3) Support Vector Machines: Support Vector Machines (SVM) is a supervised learning algorithm developed by Vapnik [24]. It seeks for a hyperplane

$$h(x) = w^T x + b = 0 \tag{2}$$

which maximizes the separation (margin) between the hyperplane and the nearest samples of each class (support vectors). Among the alternatives to extend the algorithm for multi-class problems, the *one-vs-one* scheme is used here.

SVM relies on the use of a Kernel function to deal with non-linearly separable problems. In this work, two kernel functions will be considered: radial basis function (RBF) kernel (Eq. 3) and polynomial (Poly) kernel (Eq. 4).

$$K(x, y) = e^{-\gamma \cdot \|x-y\|^2} \tag{3}$$

$$K(x, y) = \langle x, y \rangle^n \tag{4}$$

The training of the SVM is conducted by the Sequential Minimal Optimization (SMO) algorithm [25].

4) Hidden Markov Models: HMM are used here as explained for the online data. In this case, resizing and feature extraction are performed like in the work of Pugin [26].

V. Experimentation

The experimental part of this work focuses on providing the first classification results for the HOMUS dataset. In this way, we try to show what aspects of these data seem more appropriate or what are the main challenges to recognize the different musical symbols. To this end, two experiments are presented in this section. The first experiment is carried out to assess if the algorithms can detect the symbols regardless the particular style of each musician. The second experiment is aimed to analyze the accuracy of the algorithms when samples of the same user have been presented during the training stage. Next subsections describe these experiments.

A. User-independent experiment

In this experiment we aim to assess the difficulty of recognizing symbols from an unknown user. The samples of each musician are isolated from the whole dataset and used as test set (100 sets). Then, a 100-fold cross validation is conducted using a common $0-1$ loss function. The error rates obtained after applying the algorithms described in Section IV are shown in Fig. 3 (user-independent columns).

As seen in the results, algorithms are not very reliable in this scenario since all of them obtain error rates higher than 15 %. DTW obtains the lowest error rate (15.2 %). Among the offline techniques, SVM with RBF kernel provides the best error rate (26 %). To measure the significance of the results, a *Wilcoxon* statistical test was performed using KEEL software [27] (see Table III). It can be seen that DTW achieves significantly better results than other techniques.

B. User-dependent experiment

The latter experiment is focused on assessing how the classification results are affected when samples of the same musician are found in the training set. Each musician is divided into four sets and each one is used as a fold for a cross-validation experiment. To build the training set, two alternatives can be used: (1) using only the rest of the samples of the same musician (user set), and (2) using the rest of the dataset, including the remaining samples of the same musician

TABLE III. Summary of the Wilcoxon test for user-independent experiment. ●= the method in the row improves the method of the column. ○= the method in the column improves the method of the row. Upper diagonal of level significance $\alpha = 0.9$, Lower diagonal level of significance $\alpha = 0.95$

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
DTW (1)	-	●	●	●	●	●	●	●	●	●	●
String (2)	○	-	●	●	●	●	●	●	●	●	●
Angle (3)	○	○	-	●	●	●	●	●	●	●	●
HMM_{on} (4)	○	○	○	-	●	○	●	●	●	●	●
MLP (5)	○	○	○	○	-	○	○	○	○	○	○
RBF (6)	○	○	○	●	●	-	●	●	●	●	●
Poly (7)	○	○	○	○	●	○	-	●	●	●	●
1NN (8)	○	○	○	○	●	○	○	-	○	●	●
3NN (9)	○	○	○	○	●	○	○	●	-	●	
5NN (10)	○	○	○	○	●	○	○	○	○	-	
HMM_{off} (11)	○	○	○	○	●	○	○	○			-

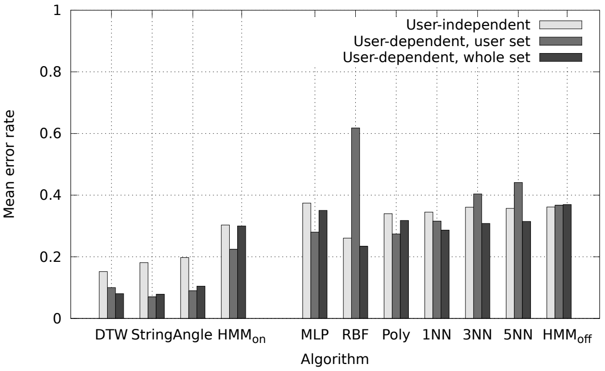


Fig. 3. Mean error rate of classification experiments. **String**: Freeman Chain Code, **Angle**: Freeman Chain Code based on angles, **DTW**: Dynamic Time Warping, $\mathbf{HMM_{on}}$: Hidden Markov Models with online features, **MLP**: Multi-Layer Perceptron, **Poly**: Support Vector Machine with Polynomial kernel, **RBF**: Support Vector Machine with Radial Basis Function kernel, **k-NN**: k-Nearest Neighbor using images, $\mathbf{HMM_{off}}$: Hidden Markov Models with offline features.

TABLE IV. Summary of the Wilcoxon test for user-dependent (user set) experiment. ●= the method in the row improves the method of the column. ○= the method in the column improves the method of the row. Upper diagonal of level significance $\alpha = 0.9$, Lower diagonal level of significance $\alpha = 0.95$

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
DTW (1)	-			●	●	●	●	●	●	●	●
String (2)		-		●	●	●	●	●	●	●	●
Angle (3)			-	●	●	●	●	●	●	●	●
HMM_{on} (4)	○	○	○	-	●	●	●	●	●	●	●
MLP (5)	○	○	○	○	-	○			○	○	●
RBF (6)	○	○	○	○	●	-	●	●	●	●	○
Poly (7)	○	○	○	○		○	-	○	○	○	●
1NN (8)	○	○	○	○		○	●	-	○	○	●
3NN (9)	○	○	○	○	●	○	●	●	-		○
5NN (10)	○	○	○	○	●	○	●	●		-	○
HMM_{off} (11)	○	○	○	○	○	●	○	○	●	●	-

TABLE V. Summary of the Wilcoxon test for user-dependent (whole set) experiment. ●= the method in the row improves the method of the column. ○= the method in the column improves the method of the row. Upper diagonal of level significance $\alpha = 0.9$, Lower diagonal level of significance $\alpha = 0.95$

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
DTW (1)	-			●	●	●	●	●	●	●	●
String (2)		-		●	●	●	●	●	●	●	●
Angle (3)			-	●	●	●	●	●	●	●	●
HMM_{on} (4)	○	○	○	-	●	○	●	○		●	●
MLP (5)	○	○	○	○	-	●	●	●	●	●	●
RBF (6)	○	○	○	●	○	-	○	○	○	○	●
Poly (7)	○	○	○	○	○	●	-				●
1NN (8)	○	○	○	●	○	●		-	○		●
3NN (9)	○	○	○		○	●		●	-		●
5NN (10)	○	○	○	○	○	●				-	●
HMM_{off} (11)	○	○	○	○	○	○	○	○	○	○	-

(whole set). This two options are confronted experimentally in a 400-fold (four per musician) cross validation. Figure 3 (user-dependent columns) show the results of this experiment, which is measure using the $0-1$ loss function as well.

Algorithms using the online nature of the data have the best performance while those exploiting offline modality still have higher error rates. Conventional FCC has reported the best error rate, on average (7 %). Regarding the two ways of building the training set, there are no clear trend in the results. Some algorithms have improved when using the whole dataset such as NN family and, especially, SVM with a RBF kernel (from 61 % to 23 %) because of its poor performance with few training data. However, in other algorithms, the error rate hardly varies or even rises, as in the case of SVM with a Polynomial kernel, the MLP or HMM with online features. The Wilcoxon statistical tests for these experiments are shown in Table IV and V. If each modality is seen as a whole, algorithms that work with the online data, except for HMM, achieve significantly better results than the others.

Comparing these results with those obtained in the previous experiment we can conclude that including samples of the same user during the training set can remarkably improve the performance of some algorithms. For instance, FCC has improved considerably its performance from 18 % to 7 % of error rate. Depending on the algorithm used, it is more convenient to do it with the rest of the dataset or only with the remaining samples of the same user. Algorithms that exploit the online modality of the data, except for the HMM, have shown a significantly better performance in both experiments. Specifically, DTW has proven to be the best technique since it improves significantly the results of other algorithms in the user-independent experiment and no one is significantly better in the user-dependent experiments. HMM deserves further consideration because its performance is closely linked to feature extraction. In any case, in this work we focused on features used in previous studies for the same task.

VI. Conclusions

The work presented here aims to become a first point of reference for recognition of online handwritten music notation. This process is focused on recognizing musical symbols that are drawn on a digital score using a friendly tablet device and an electronic pen. In this way, musicians can digitize their compositions without resorting to conventional music score editors.

Some previous studies that have worked on this issue have been presented. However, all of them used their own corpus, so there is still a lack of comparative experiments that indicate which algorithms are better for this task. To solve this problem, this paper has presented the HOMUS (Handwritten Online Musical Symbols) dataset. This dataset contains 15200 samples of musical symbols from 100 expert musicians. Within this set, 32 different types of musical symbols can be found. It is expected that the dataset provides sufficient samples so that the results depend on the techniques used for classification.

To establish the first baseline, experiments with well-known pattern recognition algorithms have been carried out. FCC, DTW and HMM have been used to take advantage of the online nature of these data while k-NN, SVM, ANN and HMM have been utilized to classify samples from the offline modality (image). Two experiments were conducted to better understand this dataset and draw the first conclusions on the classification of these symbols. The first experiment consists in measuring the difficulty of recognizing a symbol when it comes from an unknown musician (user-independent). In the second experiment, samples of the same musician are included in the training set (user-dependent). Results showed that recognizing symbols from unseen styles presents the main difficulty. Error rates of the user-independent experiment among 32 classes did not dropped below 15 % in any of the algorithms considered. Algorithms that exploit the online nature of the data has proven to be the most promising for the classification task, achieving results that improve the performance of those which use the offline modality. Considering all the experiments, DTW has shown the best performance. Nevertheless, results showed room for improvement.

These results has also led to the conclusion that a competitive system will need samples of the actual user. This scenario is feasible in real-world cases. The user can be asked to perform a training phase before using the system, in which he writes all the musical symbols with his own style. This extra effort can prevent a large number of classification errors that must be posteriorly corrected. Forcing the user to perform this phase can be actually seen as a way to minimize the human effort throughout the entire process. The user can also provide his writing style transparently by means of corrections where a misclassification is produced (user adaptation techniques).

As future work, the main challenge is to extend this work to recognize entire music scores.

Acknowledgment

This work was partially supported by a FPU fellowship (AP2012-0939) from the Spanish Ministerio de Educacin, Cultura y Deporte, the Spanish CICyT through the project TIN2009-14205-C04-C1 and Consejería de Educación de la Comunidad Valenciana through project PROMETEO/2012/017.

The authors are grateful to all the people who collaborate in the creation of the dataset.

References

[1] D. Bainbridge and T. Bell, “The Challenge of Optical Music Recognition,” *Language Resources and Evaluation*, vol. 35, pp. 95–121, 2001.

[2] A. Rebelo, I. Fujinaga, F. Paszkiewicz, A. Marcal, C. Guedes, and J. Cardoso, “Optical music recognition: state-of-the-art and open issues,” *International Journal of Multimedia Information Retrieval*, pp. 1–18, 2012.

[3] D. Martin-Albo, V. Romero, and E. Vidal, “Interactive off-line handwritten text transcription using on-line handwritten text as feedback,” in *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, 2013, pp. 1280–1284.

[4] J. Anstice, T. Bell, A. Cockburn, and M. Setchell, “The design of a pen-based musical input system,” in *Sixth Australian Conference on Computer-Human Interaction, 1996. Proceedings.*, 1996, pp. 260–267.

[5] O. Poláček, A. J. Sporka, and P. Slavík, “Music alphabet for low-resolution touch displays,” in *Proceedings of the International Conference on Advances in Computer Enterntainment Technology*, ser. ACE ’09. New York, NY, USA: ACM, 2009, pp. 298–301.

[6] H. Miyao and M. Maruyama, “An online handwritten music symbol recognition system,” *International Journal of Document Analysis and Recognition (IJDAR)*, vol. 9, no. 1, pp. 49–58, 2007. [Online]. Available: http://dx.doi.org/10.1007/s10032-006-0026-9

[7] S. Macé, Éric Anquetil, and B. Couasnon, “A generic method to design pen-based systems for structured document composition: Development of a musical score editor,” in *Proceedings of the First Workshop on Improving and Assesing Pen-Based Input Techniques*, Edinghburg, 2005, pp. 15–22.

[8] S. E. George, “Online pen-based recognition of music notation with artificial neural networks,” *Comput. Music J.*, vol. 27, no. 2, pp. 70–79, Jun. 2003.

[9] K. C. Lee, S. Phon-Amnuaisuk, and C.-Y. Ting, “Handwritten music notation recognition using hmm – a non-gestural approach,” in *International Conference on Information Retrieval Knowledge Management, (CAMP), 2010*, 2010, pp. 255–259.

[10] C. Dalitz, M. Droettboom, B. Pranzas, and I. Fujinaga, “A comparative study of staff removal algorithms,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 5, pp. 753–766, 2008.

[11] D. W. Aha, D. Kibler, and M. K. Albert, “Instance-based learning algorithms,” *Mach. Learn.*, vol. 6, no. 1, pp. 37–66, Jan. 1991.

[12] V. I. Levenshtein, “Binary codes capable of correcting deletions, insertions, and reversals,” Tech. Rep. 8, 1966.

[13] H. Freeman, “On the encoding of arbitrary geometric configurations,” *Electronic Computers, IRE Transactions on*, vol. EC-10, no. 2, pp. 260–268, 1961.

[14] H. Sakoe and S. Chiba, “Readings in speech recognition,” A. Waibel and K.-F. Lee, Eds. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1990, ch. Dynamic programming algorithm optimization for spoken word recognition, pp. 159–165.

[15] M. Faundez-Zanuy, “On-line signature recognition based on vq-dtw,” *Pattern Recognition*, vol. 40, no. 3, pp. 981 – 992, 2007.

[16] B. Hartmann and N. Link, “Gesture recognition with inertial sensors and optimized dtw prototypes,” in *IEEE International Conference on Systems Man and Cybernetics (SMC), 2010*, 2010, pp. 2102–2109.

[17] Z. Ghahramani, “Hidden markov models.” River Edge, NJ, USA: World Scientific Publishing Co., Inc., 2002, ch. An Introduction to Hidden Markov Models and Bayesian Networks, pp. 9–42. [Online]. Available: http://dl.acm.org/citation.cfm?id=505741.505743

[18] L. Hu and R. Zanibbi, “Hmm-based recognition of online handwritten mathematical symbols using segmental k-means initialization and a modified pen-up/down feature,” in *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, 2011, pp. 457–462.

[19] S. Azeem and H. Ahmed, “Combining online and offline systems for arabic handwriting recognition,” in *Pattern Recognition (ICPR), 2012 21st International Conference on*, 2012, pp. 3725–3728.

[20] F. Jelinek, *Statistical Methods for Speech Recognition*. Cambridge, MA, USA: MIT Press, 1997.

[21] A. Rebelo, G. Capela, and J. S. Cardoso, “Optical recognition of music symbols: A comparative study,” *Int. J. Doc. Anal. Recognit.*, vol. 13, no. 1, pp. 19–31, Mar. 2010.

[22] D. Graupe, *Principles of Artificial Neural Networks*, 2nd ed. River Edge, NJ, USA: World Scientific Publishing Co., Inc., 2007.

[23] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Neurocomputing: foundations of research,” J. A. Anderson and E. Rosenfeld, Eds. Cambridge, MA, USA: MIT Press, 1988, ch. Learning representations by back-propagating errors, pp. 696–699.

[24] V. N. Vapnik, *Statistical learning theory*, 1st ed. Wiley, Sep. 1998.

[25] J. C. Platt, “Advances in kernel methods,” B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds. Cambridge, MA, USA: MIT Press, 1999, ch. Fast training of support vector machines using sequential minimal optimization, pp. 185–208.

[26] L. Pugin, “Optical music recognitoin of early typographic prints using hidden markov models,” in *ISMIR*, 2006, pp. 53–56.

[27] J. Alcal-Fdez, L. Snchez, S. Garca, M. Jesus, S. Ventura, J. Garrell, J. Otero, C. Romero, J. Bacardit, V. Rivas, J. Fernndez, and F. Herrera, “Keel: a software tool to assess evolutionary algorithms for data mining problems,” *Soft Computing*, vol. 13, no. 3, pp. 307–318, 2009. [Online]. Available: http://dx.doi.org/10.1007/s00500-008-0323-y

Chapter 7

Clustering of Strokes from Pen-based Music Notation: An Experimental Study

Calvo-Zaragoza, J. and Oncina, J. (2015). Clustering of strokes from pen-based music notation: An experimental study. In *7th Iberian Conference Pattern Recognition and Image Analysis, IbPRIA 2015, Santiago de Compostela, Spain, June 17-19, 2015, Proceedings*, pages 633–640

Clustering of Strokes from Pen-Based Music Notation: An Experimental Study

Jorge Calvo-Zaragoza( ) and Jose Oncina

Departamento de Lenguajes y Sistemas Informáticos,
Universidad de Alicante, Alicante, Spain
{jcalvo,oncina}@dlsi.ua.es

Abstract. A comfortable way of digitizing a new music composition is by using a pen-based recognition system, in which the digital score is created with the sole effort of the composition itself. In this kind of systems, the input consist of a set of pen strokes. However, it is hitherto unclear the different types of strokes that must be considered for this task. This paper presents an experimental study on automatic labeling of these strokes using the well-known *k-medoids* algorithm. Since recognition of pen-based music scores is highly related to stroke recognition, it may be profitable to repeat the process when new data is received through user interaction. Therefore, our intention is not to propose some stroke labeling but to show which stroke dissimilarities perform better within the clustering process. Results show that there can be found good methods in the trade-off between cluster complexity and classification accuracy, whereas others offer a very poor performance.

1 Introduction

Still nowadays many musicians consider pen and paper as the natural tools for expressing a new music composition. The ease and ubiquity of this method, as well as the fact of avoiding tedious music score editors, favor this consideration. Nevertheless, after composition is finished, it may be appropriate to have the score digitized to take advantage of many benefits such as storage, reproduction or distribution. To provide a profitable way of performing the whole process, pen-based music notation recognition systems can be developed. This way, musicians are provided with a friendly interface to work with and save the effort of digitizing the score afterwards. Although offline music score recognition systems (also known as Optical Music Recognition) could be used, it is widely known that the additional data provided by the time collection sampling of a pen-based system can lead to a better performance since more information is captured. The process of recognizing handwritten music notation is very related to other pattern recognition fields, especially that of Optical Character Recognition (OCR). Despite similarities between text and music recognition processes, this latter presents several features that make it be considered a harder task [3]. Therefore, new recognition algorithms must be developed to deal with music scores. In the case of online recognition, the natural segmentation of the input

R. Paredes et al. (Eds.): IbPRIA 2015, LNCS 9117, pp. 633–640, 2015.
DOI: 10.1007/978-3-319-19390-8_71

is the set of strokes. Each stroke is defined as the data collected between pen-up and pen-down events over the digital surface. Nevertheless, given both the high variability in handwritten musical notation and differences among writer styles (see [4]), as well as the immaturity of the field itself, it is still unclear the classes of strokes that must be considered or which are the most accurate techniques to recognize them.

This paper presents an experimental study on automatic clustering of the strokes found in pen-based music notation. From the interactive system point of view, it is specially interesting to know which algorithms provide the best results since this clustering might be repeated in order to adapt the recognition to the style of the actual user. Therefore, this work does not intend to provide just a proposal of stroke labeling, but to find which techniques would be the most appropriate within this scenario. The paper is structured as follows: Sect. 2 addresses the intrinsics of the clustering problem described; techniques for measuring dissimilarity between strokes from pen-based music notation are presented in Sect. 3; Sect. 4 describes the experimental setup, results and analysis; finally, Sect. 5 concludes.

2 The Clustering Problem

When dealing with a pen-based music recognition task, raw input consists of a series of strokes. This is the natural segmentation of such systems since the beginning and ending of a stroke are easily detected by pen-down and pen-up events. From a labeled set of isolated handwritten musical symbol we can obtain definitions of these symbols in terms of strokes. If we considered a stroke labeling, we would reduce this set by assigning the same label to similar strokes. Then, the first step would be to classify each stroke within a set of labels. A label would represent a part of a musical symbol, *i.e.,* a white note head or a stem (Fig. 1(b)), or even a whole symbol (Fig. 1(a)).

(a) Two strokes (b) One stroke

Fig. 1. *Half Note* symbol written with different set of primitives.

At this point, we have to deal with the open problem of the set of primitives to be considered. Some *ad-hoc* labeling could be used but it might not be appropriate for this task due to several reasons: the data would be clustered considering human similarity perception instead of computer-based similarity, which is what it is applied in the final system; labels would be created from the data available at that moment, thus generalization could be poor; clustering may need to be repeated after user interaction, in which new data would be received and, therefore, the system must be adapted to actual user writing style. All these reasons lead us to perform an algorithmic-based clustering of the strokes found in pen-based music notation. As aforementioned, the main goal of this paper is not to give a good proposal of stroke labeling, but to measure the goodness and generalization of each possible clustering considered.

One of the key questions in any clustering problem is to choose the number of labels that must be considered (referred here as parameter k). Note that if music notation can be defined by a formal language, in which the alphabet is the primitives set, the lower the size of this set the less complex the language. Therefore, we are interested in lowering k as much as possible. On the other hand, low values of k can lead to ambiguous definitions, that is, more than one musical symbol defined by the same sequence of primitives. Considering that we should avoid these ambiguous definitions, our problem can be modeled as a constrained clustering problem.

Constrained clustering is the task of clustering data in which some conditions over the cluster assignments must be fulfilled. In the literature, several works on constrained clustering can be found [1,17]. The two considered cases are those of *must-link* and *cannot-link* conditions. The first defines pairs of data points that must be in the same cluster while the latter defines pairs of data points that must be in different clusters. The constraint in our case is to avoid more than one musical symbol defined by the same primitives. Let us consider just two musical symbols (*Whole Note* and *Half Note*). Let us assume that we have some definitions in which these symbols are described in terms of strokes. That is,

$$\begin{aligned}
\text{Whole Note} &\rightarrow s_1 \\
\text{Whole Note} &\rightarrow s_2\ s_3 \\
\text{Half Note} &\rightarrow s_4 \\
\text{Half Note} &\rightarrow s_5\ s_6
\end{aligned}$$

in which $s_1, s_2, \ldots, s_6$ denote strokes.

Let $\zeta(s)$ stands for the label assigned to stroke s. Then, we are looking for a labeling such that $\zeta(s_1) \neq \zeta(s_4)$ as well as $\zeta(s_2) \neq \zeta(s_5) \vee \zeta(s_3) \neq \zeta(s_6)$. This way, none but one symbol could be defined by the same sequence of primitives. Note that, although we are stating *cannot-link* conditions, we are not interested in just pairwise constraints but to *n-to-n* as shown above.

To our best knowledge, this kind of conditions is not approached in previous works on constrained clustering. Since developing such algorithm is out of the scope of the present work, we are going to follow a straightforward approach: unconstrained clustering will be performed and conditions will be checked afterwards. The lowest value of k that achieves a valid clustering will be considered. The problem with this approach is that it may lead to a very high number of k. Thus, some rate of ambiguous symbols will be allowed. We assume that some disambiguation can be solved by means of semantic music language models, as typically happens in offline Optical Music Recognition [11].

The unconstrained clustering process will be guided by a *k-medoids* algorithm [16], one of the most common and successful algorithms for data clustering [14]. This algorithm is very related to *k-means* but instead of taking the mean point at the expectation step, it searches the point of the cluster that minimizes the actual cost (set mean). In order to provide a more robust clustering, the initialization of the method is performed as described for *k-means++* algorithm [2]. This algorithm proposes a initialization (first centroids) that is expected to provide better results and faster convergence. It starts with a random centroid and the rest of the centroids are chosen randomly following a decreasing probability with respect to the distance to the nearest centroid already selected.

To perform the clustering we need to define some function that measures the distance or dissimilarity between two strokes. Next section will describe the techniques considered for such task.

3 Dissimilarity Functions for Pen-Based Music Notation Strokes

The data points of our clustering problem are handwritten strokes. Each stroke is composed of a sequence of consecutive two dimensional points defining the path that the pen follows. For the clustering algorithm we need to define some techniques to measure the dissimilarity between two given strokes. Below we present some functions that can be applied directly to the stroke data. Moreover, we also describe some ways of mapping the strokes onto feature vectors, for which other several dissimilarity measures can be applied.

Before computing these dissimilarities, a smoothing process will also be considered. Smoothing is a common preprocessing step in pen-based recognition to remove some noise and jitters [7]. It consists in replacing each point of the stroke by the mean of their neighbors points. Some values of neighborhood size will be considered at the experimentation stage.

3.1 Raw Stroke Distance

The digital surface collects the strokes at a fixed sampling rate so that each one may contain a variable number of points. However, some dissimilarity functions can be applied to this kind of data. Those considered in this work are the following:

- Dynamic Time Warping (DTW) [15]: a technique for measuring the dissimilarity between two time signals which may be of different duration.
- Edit Distance with Freeman Chain Code (FCC): the sequence of points representing a stroke is converted into a string using a codification based on Freeman Chain Code [5]. Then, the common Edit Distance [9] is applied.
- Edit Distance for Ordered Set of Points (OSP) [13]: an extension of the Edit Distance for its use over ordered sequences of points.

3.2 Feature Extraction

On the other hand, if a set of features is extracted from the stroke path, a fixed-sized vector is obtained. Then, other common distances can be applied. In this work we are going to consider the following feature extraction and distances:

- Normalized stroke (Norm): the whole set of points of the stroke is normalized to a sequence of n points by an equally resampling technique. Therefore, a stroke can be characterized by $2n$-dimensional real-valued feature vector. Given vectors x and y, two different distances are going to be considered:
  - Average Euclidean Distance (Norm+Euc) between the points of the sequences: $\frac{1}{n}\sum_{i=1}^{n} d(x_i, y_i)$
  - Average Turning Angle (Norm+Ang) between segments of the two sequences: $\frac{1}{n}\sum_{i=2}^{n} d_\Theta(x_{i-1}x_i, y_{i-1}y_i)$, where $x_{i-1}x_i$ represents the segment connecting points x_{i-1} and x_i, and d_Θ is the angular difference in radians. It has been chosen due to its good results in [8].
- Squared Image: an image of the stroke can be obtained by reconstructing the drawing made. Preliminary experimentation showed that the best results are obtained by simulating a pen thickness of 3. Images are then resized to 20×20 as done in the work of Rebelo et al. [10]. A 400-dimensional feature vector is obtained, for which the Euclidean distance is applied.
- Image Features: the image is partitioned into sub-regions, from which background, foreground and contour local features are extracted [12]. Then, similarity is measured using Euclidean distance.

4 Experimentation

This section contains the experimentation performed with the musical symbols of the Handwritten Online Musical Symbols (HOMUS) dataset [4]. HOMUS is a freely available dataset which contains 15200 samples from 100 musicians of pen-based isolated musical symbols. Within this set of symbols, 39219 strokes can be found. Taking advantage of the features of the HOMUS, two experiments will be carried out: user-dependent and user-independent scenarios. In the first, the clustering is performed separately for the samples of each writer since it is interesting to see how clustering behaves for small and similar data. In the latter, the whole dataset is used at the same time. However, since this can lead to an unfeasible computation in terms of time, only a subset of samples is selected at the beginning of the task. This subset selection is performed so that each symbol of any musician appears at least once. Clustering will be performed on this subset and the rest of the strokes will be assigned to their nearest cluster afterwards. In both experiments, some values of neighborhood parameter of the smoothing will be tested: 0 (no filtering), 1 and 2. Our experiments start with a low k that is increased iteratively until reaching a valid assignment (see Sect. 2), with a maximum established to 150. In both cases, we allow an ambiguity rate of 0.1 of the total number of symbols considered. When an acceptable clustering is obtained, we measure the classification accuracy using a *leaving-one-out* scheme.

For the classification step we are going to restrict ourselves to the use of the Nearest Neighbor (NN) rule with the same similarity used for the clustering. The obvious reason is to measure the goodness of the stroke dissimilarity utilized for both clustering and classification. Nevertheless, considering the interactive nature of the task (the system may be continuously receiving new labeled sample through user interaction), other reasons also justify this choice: distance-based classification methods such as NN (or k-NN) are easily adaptable to new data; Data Reduction techniques based on dissimilarity functions could be applied to not overflow the system [6]; in addition, fast similarity search techniques could also be used in order to provide fast response. It is clear, however, that once strokes are labeled conveniently, other advanced techniques can be applied to classify this data but that experimentation will be placed as future work.

4.1 Results

Results of the user-dependent experiment described above is shown in Table 1. Since dataset contains 100 different writers, average results are reported. For the user-independent experiment, average results from 10 different initial random subsets are shown in Table 2.

Table 1. Average results (**k**: number of clusters; **acc**: classification accuracy) of a 100-fold cross-validation with each writer subset. Several values of neighborhood for smoothing are considered (0, 1, 2).

Dissimilarity	Smoothing (0)		Smoothing (1)		Smoothing (2)	
	k	acc	k	acc	k	acc
DTW	18.1	88.9	18.4	89.4	19.1	88.8
FCC	14.9	87.6	15.7	88.0	15.4	87.8
OSP	15.4	87.7	15.4	88.3	15.1	89.1
Norm+Euclidean	17.5	89.1	17.6	89.0	17.8	88.9
Norm+Angular	18.7	78.7	19.1	79.0	21.0	79.2
Squared Image	30.6	79.0	30.2	78.7	30.5	80.5
Image Features	22.6	89.4	22.5	89.0	24.8	86.7

An initial remark to begin with is that smoothing demonstrates small relevance in the process since results hardly vary among the different values considered. Moreover, dissimilarities that make use of the image representation of the stroke obtain very poor results in both experiments. In fact, they obtain the worst results in the user-dependent experiment and none of them reach a low enough clustering value in the writer-independent experiment. Although variability is low when using small and similar data, differences in performance among methods are increased in the writer-independent experiment. Thorough the experimentation, OSP and FCC dissimilarities have reported the best results in terms of number of clusters, in spite of showing a lower accuracy rate than

Table 2. Average results (**k**: number of clusters; **acc**: classification accuracy) of a 10-fold cross-validation experiment with the whole dataset. Several values of neighborhood for smoothing are considered (0, 1, 2).

Dissimilarity	Smoothing (0)		Smoothing (1)		Smoothing (2)	
	k	acc	k	acc	k	acc
DTW	72.0	81.8	77.0	81.3	86.8	80.0
FCC	52.8	79.3	53.4	80.3	52.4	80.2
OSP	47.9	77.1	49.8	80.7	46.8	81.3
Norm+Euclidean	68.8	83.8	76.1	83.4	86.3	83.4
Norm+Angular	141.4	71.5	143.5	70.7	146.3	70.5
Squared Image	>150	-	>150	-	>150	-
Image Features	>150	-	>150	-	>150	-

DTW or Normalized strokes with Euclidean distance. Nevertheless, it is expected that both OSP and FCC methods may improve their accuracy performance by allowing them to use a high number of clusters. Results have reported that the dissimilarity applied has a big impact in the clustering process, especially when dealing with a high number of samples. Thus, if the process has to be performed when new data is available, it is profitable to use methods such as OSP or FCC that have shown a better ability to group the strokes.

5 Conclusions

This work presents an experimental study on clustering of strokes from pen-based music notation. The main goal is to show which dissimilarity measure between strokes performs better since we are interested in repeating the process when new data is received. Experimentation showed that, although the clustering process is robust in a user-dependent experiment, much attention should be devoted to the user-independent scenario. In this last, some techniques like OSP and FCC achieved good results whereas others, especially image-based techniques, were reported less suitable for grouping these strokes. As future work, there are several promising lines that should be explored with respect to clustering. These lines include approaching the unconstrained clustering problem when n-to-n constraints are required or developing an efficient clustering that repeats the process when new data is received taking advantage of the previous assignment. In addition, once a valid clustering is achieved, some advanced classification techniques could be considered instead of resorting to the NN rule.

Acknowledgements. This work was partially supported by the Spanish Ministerio de Educación, Cultura y Deporte through a FPU Fellowship (AP2012–0939), the Spanish Ministerio de Economía y Competitividad through Project TIMuL (No. TIN2013-48152-C2-1-R supported by EU FEDER funds) and Consejería de Educación de la Comunidad Valenciana through Project PROMETEO/2012/017.

References

1. de Amorim, R.: Constrained clustering with minkowski weighted k-means. In: 2012 IEEE 13th International Symposium on Computational Intelligence and Informatics (CINTI), pp. 13–17, November 2012
2. Arthur, D., Vassilvitskii, S.: K-means++: the advantages of careful seeding. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007, pp. 1027–1035. Society for Industrial and Applied Mathematics, Philadelphia (2007)
3. Bainbridge, D., Bell, T.: The challenge of optical music recognition. Lang. Resour. Eval. **35**, 95–121 (2001)
4. Calvo-Zaragoza, J., Oncina, J.: Recognition of pen-based music notation: the HOMUS dataset. In: Proceedings of the 22nd International Conference on Pattern Recognition, Stockholm, Sweden, pp. 3038–3043 (2014)
5. Freeman, H.: On the encoding of arbitrary geometric configurations. IRE Trans. Electron. Comput. **10**(2), 260–268 (1961)
6. García, S., Luengo, J., Herrera, F.: Data Preprocessing in Data Mining. Intelligent Systems Reference Library, vol. 72. Springer, Switzerland (2015)
7. Kim, J., Sin, B.K.: Online handwriting recognition. In: Doermann, D., Tombre, K. (eds.) Handbook of Document Image Processing and Recognition, pp. 887–915. Springer, London (2014)
8. Kristensson, P.O., Denby, L.C.: Continuous recognition and visualization of pen strokes and touch-screen gestures. In: Proceedings of the 8th Eurographics Symposium on Sketch-Based Interfaces and Modeling, SBIM 2011, pp. 95–102. ACM, New York (2011)
9. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. Technical report 8 (1966)
10. Rebelo, A., Capela, G., Cardoso, J.: Optical recognition of music symbols. Int. J. Doc. Anal. Recogn. **13**(1), 19–31 (2010)
11. Rebelo, A., Fujinaga, I., Paszkiewicz, F., Marçal, A.R.S., Guedes, C., Cardoso, J.S.: Optical music recognition: state-of-the-art and open issues. IJMIR **1**(3), 173–190 (2012)
12. Rico-Juan, J.R., Iñesta, J.M.: Confidence voting method ensemble applied to offline signature verification. Pattern Anal. Appl. **15**(2), 113–120 (2012)
13. Rico-Juan, J.R., Iñesta, J.M.: Edit distance for ordered vector sets: a case of study. In: Yeung, D.-Y., Kwok, J.T., Fred, A., Roli, F., de Ridder, D. (eds.) SSPR 2006 and SPR 2006. LNCS, vol. 4109, pp. 200–207. Springer, Heidelberg (2006)
14. Rokach, L.: A survey of clustering algorithms. In: Maimon, O., Rokach, L. (eds.) Data Mining and Knowledge Discovery Handbook, 2nd edn, pp. 269–298. Springer, New York (2010)
15. Sakoe, H., Chiba, S.: Readings in speech recognition. In: Waibel, A., Lee, K.-F. (eds.) Dynamic Programming Algorithm Optimization for Spoken Word Recognition, pp. 159–165. Morgan Kaufmann Publishers Inc., San Francisco (1990)
16. Theodoridis, S., Koutroumbas, K.: Pattern Recognition, 3rd edn. Academic Press Inc., Orlando (2006)
17. Wagstaff, K., Cardie, C., Rogers, S., Schrödl, S.: Constrained k-means clustering with background knowledge. In: Proceedings of the Eighteenth International Conference on Machine Learning, pp. 577–584. Morgan Kaufmann Publishers Inc., San Francisco (2001)

Chapter 8

Improving kNN multi-label classification in Prototype Selection scenarios using class proposals

Calvo-Zaragoza, J., Valero-Mas, J. J., and Rico-Juan, J. R. (2015b). Improving kNN multi-label classification in Prototype Selection scenarios using class proposals. *Pattern Recognition*, 48(5):1608–1622

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Improving kNN multi-label classification in Prototype Selection scenarios using class proposals

Jorge Calvo-Zaragoza*, Jose J. Valero-Mas, Juan R. Rico-Juan

Department of Software and Computing Systems, University of Alicante, Carretera San Vicente del Raspeig s/n, 03690 Alicante, Spain

ARTICLE INFO

Article history:
Received 18 July 2014
Received in revised form
18 November 2014
Accepted 22 November 2014
Available online 3 December 2014

Keywords:
K-Nearest Neighbor
Multi-label classification
Prototype Selection
Class proposals

ABSTRACT

Prototype Selection (PS) algorithms allow a faster Nearest Neighbor classification by keeping only the most profitable prototypes of the training set. In turn, these schemes typically lower the performance accuracy. In this work a new strategy for multi-label classifications tasks is proposed to solve this accuracy drop without the need of using all the training set. For that, given a new instance, the PS algorithm is used as a fast recommender system which retrieves the most likely classes. Then, the actual classification is performed only considering the prototypes from the initial training set belonging to the suggested classes. Results show that this strategy provides a large set of trade-off solutions which fills the gap between PS-based classification efficiency and conventional kNN accuracy. Furthermore, this scheme is not only able to, at best, reach the performance of conventional kNN with barely a third of distances computed, but it does also outperform the latter in noisy scenarios, proving to be a much more robust approach.

1. Introduction

Since its first proposal in 1951 [1], the *k*-Nearest Neighbor rule (kNN) constitutes one of the most well-known algorithms in Pattern Recognition (PR) for supervised non-parametric classification [2], case in which statistical knowledge of the conditional density functions of the classes involved is not available. Most of the kNN popularity in classification tasks comes from its conceptual simplicity and straightforward implementation, which can be described as a distance comparison between elements. More precisely, given an input x, the NN (kNN) rule assigns to x the label of the nearest (k-nearest) prototypes of the training set. An interesting theoretical property of this rule is that its probability of error is bounded above by twice the Bayes error rate [3]. kNN algorithm is usually described as a *lazy* learner which, in opposition to *eager* learners, does not build a classification model out of the training data until a new element has to be classified. Inside this lazy learning family, kNN is an example of instance-based method, meaning that no classification rules are obtained out of the training data, being part or the total amount of training information itself used for the classification task [4].

Despite the commented kNN popularity in PR, this method suffers from several drawbacks, out of which three clearly limit its application [5]: the first one is that, as an instance-based classifier, storage memory requirements tend to be high for keeping all the training data; the second limitation is its low computational efficiency since, each time new data has to be classified, many distance computations are repeated due to the lack of a model; the third disadvantage is that this method is sensitive to noisy instances, especially for low k values.

Prototype Selection (PS) is one of the most common techniques for overcoming the commented drawbacks [6]. This family of methods reduces the size of the initial training set so as to decrease the aforementioned computational cost and sensitiveness to noise by removing both redundant and noisy instances from the initial training set. However, although this process is expected to either maintain or even increase the classification results, in practical situations the accuracy obtained tends to be lower than with the initial set.

In this paper, in order to tackle the commented issue, we propose a strategy which aims to combine the classification accuracy of retaining all the training set with the time efficiency PS methods provide in kNN classification. Our proposal first reduces the training set by using a PS algorithm; on that reduced set, we perform the classification of the new element but, instead of retrieving the most convenient class, a rank of classes is proposed according to their suitability; these proposals are then used for classifying the new element on a filtered version of the initial training data in which only the elements belonging to the previously ranked classes are considered for the classification task. This scheme is expected to provide a profitable way of approaching a multi-label classification scenario as a large quantity of prototypes could be discarded.

* Corresponding author. Tel.: +349 65 903772; fax: +349 65 909326.
E-mail addresses: jcalvo@dlsi.ua.es (J. Calvo-Zaragoza), jjvalero@dlsi.ua.es (J.J. Valero-Mas), JuanRamonRico@ua.es (J.R. Rico-Juan).

http://dx.doi.org/10.1016/j.patcog.2014.11.015
0031-3203/© 2014 Elsevier Ltd. All rights reserved.

The rest of the paper is structured as follows: Section 2 introduces some related proposals to this topic; Section 3 thoroughly develops our proposed approach; Section 4 explains the evaluation methodology proposed; Section 5 shows the results obtained as well as a thorough discussion about them; finally, Section 6 explains the general conclusions obtained from the work and discusses about possible future work.

2. Related work

Among the different stages which comprise the so-called Knowledge Discovery in Databases (KDD), Data Preprocessing (DP) is the set of processes devoted to provide the information to the Data Mining (DM) system in the suitable amount, structure and format. Data Reduction (DR), which constitutes one of these DP possible tasks, aims at obtaining a reduced set of the original data which, if provided to the DM system, would produce the same output as the original data [7].

DR techniques are widely used in kNN classification as a means of overcoming its previously commented drawbacks, being the two most common approaches Prototype Generation (PG) and Prototype Selection (PS) [8]. Both methods focus on reducing the size of the initial training set for lowering the computational requirements and removing noisy instances while keeping, if not increasing, the classification accuracy. The former method creates new artificial data to replace the initial set while the latter one simply selects certain elements from that set. The work presented here focuses on PS techniques, which are less restrictive than PG as they do not require extra knowledge to merge elements from the initial set. However, reader is referred to [9] for a detailed introduction and thorough study of PG techniques. On the other hand, below we introduce the basics of PS methods due to its relevance in the present paper.

As aforementioned, PS methods aim to reduce the size of the initial training set to lower the computational cost and remove noisy instances which might confuse the classifier. Given its importance, many different approaches have been proposed throughout the years to carry out this task. Due to this large range of possible strategies, many different criteria have been posed in order to establish a taxonomy for these methods. However, in this paper we restrict ourselves to a criterion which basically divides them into three different families:

- *Condensing*: The idea followed by these methods is to reduce as much as possible the dataset size by keeping only the closest points to the different class decision boundaries. While accuracy on training set is usually maintained, generalization accuracy is lowered.
- *Editing*: This approach eliminates instances which produce some class overlapping, typical situation of elements located close to the decision boundaries or noisy data. Data reduction rate is lower than in the previous case but generalization accuracy is higher.
- *Hybrid*: These algorithms look for a compromise between the two previous approaches, which means seeking the smallest dataset while improving, or at least maintaining, the generalization accuracy of the former set.

For a thorough explanation regarding taxonomy criteria for PS algorithms, the reader may check [5] in which an extensive introduction to this topic as well as a comprehensive classification taxonomy for the different methods is discussed.

Even though PS methods are expected to keep the same accuracy as with the initial training set, in practice it becomes difficult to fulfill this requirement, reason why much research has been recently devoted to enhance these techniques through data reduction and learning techniques [7]. Some explored lines to improve accuracy results have been the use of ensembles together with PS [10] or hybridizing Feature Selection (FS) schemes with PS using Evolutionary Algorithms (EA) [11,12]. On the other hand, and in order to solve the scalability issue these algorithms show for very large datasets, some common methods have been the use of stratification [13] and distributed approaches [14].

In this paper it is proposed a scheme that tries to overcome the aforementioned drawbacks of PS algorithms in a very different way. Here, PS is used just as a preprocessing stage for selecting the most promising labels, which will be used for the actual classification in the original dataset. It should be noted that this approach does not constrain the development of PS algorithms as its performance, as a second stage process, is highly influenced by the initial PS step. In fact, the better the underlying PS algorithm is used, the better the performance is expected to be achieved with our scheme.

3. Improving prototype selection k-Nearest Neighbor classification

Let T be a training set which consists of pairs $\{(x_i, y_i) | x_i \in \mathcal{X}, y_i \in \mathcal{Y}\}_{i=1}^{|T|}$ drawn from an unknown function $f : \mathcal{X} \to \mathcal{Y}$. Typically, $\mathcal{X}$ is a feature space and $\mathcal{Y}$ is a discrete set of *labels* or *classes*. The main goal in supervised classification is to approximate this function *f*.

Given an input $x \in \mathcal{X}$, the k-Nearest Neighbor rule hypothesizes about $f(x)$ by choosing the most frequent label within its k nearest prototypes of T based on a dissimilarity function $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+ \cup \{0\}$.

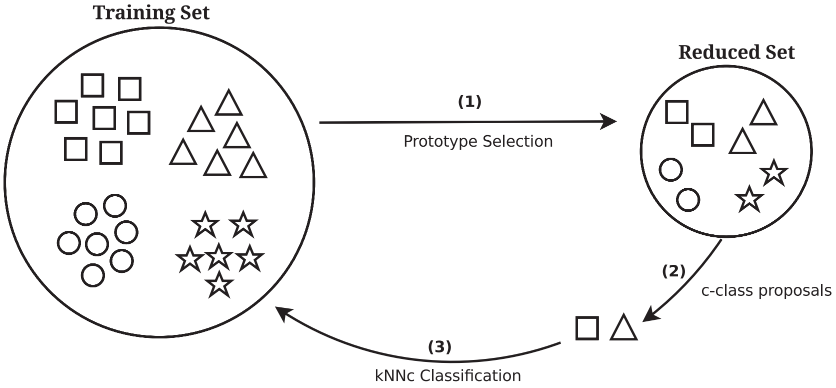


Fig. 1. General scheme of our classification strategy.

Similarly, a PS method takes T and gives a reduced set $R \subseteq T$ following some criteria (see Section 2). Due to the reduction of the original set, the approximation of the function may be different.

Considering the operation of kNN, a misclassification with R that is correctly classified with T has to be produced because of prototypes of the set $T \setminus R$. If we assume that PS is carried out due to time execution, then a profitable procedure is to recover the prototypes of $T \setminus R$ that play a key role in the approximation of $f(x)$. Obviously, finding out which ones of the whole set of prototypes must be reconsidered is not a trivial matter. In this work we propose a strategy that provides a heuristic solution to this situation.

Our classification strategy is based on a three-phase algorithm, which basically consists of the following steps (see Fig. 1 to find an illustration of the process):

1. A given PS algorithm is applied to the whole training set, producing a reduced set. This process is done just once in a preprocessing stage.
2. A new input x is given to the classification system. A reduced set of labels is selected as possible hypotheses for the input x taking into account only the reduced set. Specifically, we propose to select the c (parameter) nearest classes of input x.
3. The final hypothesis is decided using the kNN rule with the part of the initial training set restricted to the c labels proposed in the previous step (kNNc search).

The main idea is to use the reduced set as a fast recommending system, which only has to propose some of the possible labels. After that, the prototypes of those proposed labels are recovered and the final decision is then computed with them, thereby speeding-up the original NN classification.

Let us define $NN(x, k, T)$ as kNN rule for input x and training set T. Let $nearestLabels(c, x, R)$ denote the c-nearest labels of x, defined as a set $\mathcal{C}$ such that

$$\mathcal{C} \equiv \left\{ y \in \mathcal{Y} \middle| \min_{(x',y') \in R: y' = y} d(x, x') < \min_{(x',y') \in R: y' \in \mathcal{Y} \setminus \mathcal{C}} d(x, x') \right\} \text{ s.t} |\mathcal{C}| = c$$

That is, the first c labels that appear if we query the prototypes of the set R in ascendant order to the distance to x.

Let $T_w = \{(x, y) \in T | y = w\}$ be the prototypes of the training set with label w. Then, kNNc search can be performed following Algorithm 1. Note that the algorithm receives the reduced set R since PS can be performed offline, before the test stage.

Algorithm 1. kNNc search.

```
Require: $k, c \in \mathbb{N}; R$
  $\mathcal{C} \leftarrow$ nearestLabels$(c, x, R)$
  $T' \leftarrow \emptyset$
  for all $w \in \mathcal{C}$ do
    $T' \leftarrow T' \cup T_w$
  end for
  $h \leftarrow$ NN$(x, k, T')$
```

Our strategy requires an extra parameter: the scalar value c, which determines how many classes are recommended. This parameter allows tuning the classification since it is expected to affect inversely the accuracy and the computational time. In the experimentation section these two parameters will be analyzed in depth. Additionally, some dissimilarity $d(\cdot, \cdot)$ measure is required over the sample space since it is needed for both the kNN rule and *nearestLabels* function.

4. Experimental setup

This section presents the evaluation methodology for the assessment of the proposed approach, for which the most relevant issues are the classification strategies, the datasets utilized and the performance measurement. These three aspects are described in the following subsections.

4.1. Classification strategies

Our main goal is to compare the performance of our strategy against classical PS-based classification. To this end, we selected a representative set of PS algorithms published in the literature:

- *Condensing Nearest Neighbor* (*CNN*) [15]: Obtains a subset S out of the training set T such that every member of T is closer to a member of S of the same class than to a member of a different class. Prototypes of T are consulted randomly so different computations may give a different subset S.
- *Editing Nearest Neighbor* (*ED*) [16]: Selects a set S that starts equal to the original training set T. Each element of S which does not agree with its neighborhood is removed. As it happens with CNN, its result depends on the order the prototypes are consulted. A common extension to this technique is Multi-Editing (MED) [17], which computes repeatedly the ED algorithm until no more prototypes are removed.
- *Multi-Edit Condensing Nearest Neighbor* (*MCNN*) [18]: Applies ED algorithm and then applies CNN. The process is repeated until convergence is achieved.
- *Fast Condensing Nearest Neighbor* (*FCNN*) [19]: Computes a fast, order-independent condensing strategy based on seeking the centroids of each label. We also add a Multi-Edit Fast Condensing Nearest Neighbor (MFCNN) technique which combines the ideas of MCNN and FCNN.
- *Farther Neighbor* (*FN*) *and Nearest to Enemy* (*NE*) *rank methods* [20]: Give a probability mass value to each prototype following a voting heuristic. Then, prototypes are selected according to a parameter specified by the user that indicates the probability mass desired for each class in the reduced set.
- *Decremental Reduction Optimization Procedure 3* (*DROP3*) [21]: This algorithm applies an initial noise filtering step so as to eliminate the dependency on the order of presentation of the instances; after that, these instances are ordered according to the distance to their nearest neighbors and then, starting from the furthest ones, instances which do not affect the generalization accuracy are removed.
- *Iterative Case Filtering Algorithm* (*ICF*) [22]: Approach which bases its performance on the coverage and reachability premises to select the instances subset able to maximize the prototypes classification accuracy following the NN rule.
- *Cross-generational elitist selection, Heterogeneous recombination and Cataclysmic mutation algorithm (CHC)* [23]: Evolutionary algorithm commonly used as a representative of Genetic Algorithms in PS. The configuration of this algorithm has been the same as in [24], that is α=0.5, Population=50 and Evaluations=10,000.

All these algorithms will be confronted experimentally in order to measure its performance as PS base strategy of a kNNc search compared to the results obtained with the retrieval step proposed. Several values of k (1, 3, 5 and 7) and c (2 and 3) will be analyzed. Furthermore, kNN rule with the whole training set (no previous PS performed) will also be included.

4.2. Datasets

Our experiments are carried out with two different isolated character datasets: the NIST SPECIAL DATABASE 3 (NIST3) of the National Institute of Standards and Technology, from which a subset of the upper case characters was randomly selected (26 classes, 6500 images); and The United States Postal Office (USPS) handwritten digit dataset [25] (9298 images). In both cases, contour descriptions with Freeman Chain Codes [26] are extracted and the edit distance [27] is used as dissimilarity measure. Additionally, we include experiments with the Handwritten Online Musical Symbol (HOMUS) dataset [28]. This dataset is specially interesting for our work because it contains 15,200 prototypes of 32 different classes. Due to its good results in the baseline experimentation with this data, we will use Dynamic Time Warping [29] as dissimilarity measure.

We focused on datasets with many class labels since we consider that the main idea of kNNc is expected to provide interesting results in such data.

4.3. Performance measurement

In order to analyze the impact of our strategy in the PS-based classification, we take into account the following metrics of interest: accuracy of the strategy, the number of distances computed during the classification and the time in milliseconds. The two latter figures provide theoretical and empirical efficiency measures, respectively. Additionally, we provide an accuracy upper bound for kNNc classification measured as the percentage of times for which the correct label is within the *c*-classes' proposals. Another interesting property of PS-classification is the tolerance to noise. In order to analyze this metric, we will add synthetic noise to our data by swapping the labels of pairs of prototypes randomly chosen. The noise rates (percentage of prototypes that change their label) considered are 0.1, 0.2, 0.3 and 0.4 since these are the common values in this kind of experimentation [30].

Given some PS algorithms, the previous metrics are measured for both values considered for *c* as well as for PS-based classification without the *c*-classes retrieval step (except for the upper bound).

These measures allow us to analyze the performance of each considered strategy. Nevertheless, no comparison between the whole set of alternatives can be established so that we can determine which is the best one. The problem is that PS algorithms try to minimize the number of prototypes considered in the training set at the same time they try to increase classification accuracy. Most often, these two goals are contradictory so improving one of them implies a deterioration of the other. From this point of view, PS-based classification can be seen as a Multi-objective Optimization Problem (MOP) in which two functions want to be optimized at the same time: minimization of prototypes in the training set and maximization of the classification success rate. The usual way of evaluating this kind of problem is by means of *non-dominance* concept. One solution is said to dominate another if, and only if, it is better or equal in each goal function and, at least, strictly better in one of them. Therefore, the best solutions (there may be more than one) are those that are non-dominated.

Thus, the considered strategies will be compared by assuming a MOP scenario in which each of them is a 2-dimensional solution defined as (*acc*,*dist*) where *acc* is the accuracy obtained by the strategy and *dist* is the number of computed distances during its classification process. To analyze the results, the pair obtained by each scheme will be plotted in *2D* point graphs where the non-dominated set of pairs will be enhanced. In the MOP framework, the strategies within this set can be considered the best without defining any order among them.

5. Results

This section shows the results obtained using the approach presented in Section 3 with the experimentation described previously.

In sight of the large amount of experimentation carried out because of the number of possible combinations of schemes, noise scenarios and datasets considered, it is unpractical to present all the obtained results due to space limitations. Thus, figures presented actually constitute the average values of the three considered evaluation datasets.

For the sake of clarity, we are showing the obtained results in two different sections: a first one in which the considered datasets are evaluated in their current form and a second one in which the same evaluation is carried out with synthetic noise added to the data.

5.1. Non-added noise scenario

Results presented in this first subsection are obtained without adding artificial noise to the datasets. They can be checked in Table 1.

An initial remark to begin with is that, as no information is discarded, conventional kNN achieves the highest accuracy for all *k* values when only considering PS. However, the amount of distances computed is the maximum among all the algorithms.

ED and MED algorithms do not significantly reduce the size of the set, maintaining the accuracy in relation to the scores achieved by the kNN implementations. Due to this fact, the introduction of the kNNc approach does not produce a remarkable improvement over the simple PS implementation: accuracy is slightly increased as well as the amount of distances to be computed.

On the other hand, CNN and its extensions exhibit an interesting behavior: all of them achieve a great reduction rate, especially MCNN and MFCNN, as well as a great performance in terms of accuracy (for instance, the latter performs roughly 10% of the distances kNN does but obtaining only 4% less in terms of accuracy). On top of that, the introduction of kNNc does improve results in this case. Let us take the 3NN3 with CNN case: although the number of calculated distances is increased with respect to the PS classification, the accuracy is improved to the point of reaching performance of kNN with barely a third of distances to be computed.

EN and FN methods obtain some of the highest reduction rates (roughly ranging from 1% to 13% of the distances computed by kNN), also depending on its parameterization (the probability mass selected), though accuracy figures are noticeably affected: results get to achieve 15% points less in terms of accuracy with respect to the best result. As in the previous case, the inclusion of kNNc seems to come with some overall upturn: setting $c=3$, the accuracy is improved, in the best case scenario, to just 1% lower than the best score, despite being the number of distances to be computed around 29% of the maximum.

Hybrid algorithms DROP3 and ICF achieve great reduction rates as well (around 6–14% of the total of distances with respect to kNN), but they also experiment a significant decrease in their accuracies, with figures of about 10% and 20% lower than the maximum score. However, when using the proposed approach, there is a remarkable improvement: for instance, in the 3NN3 case, DROP3 increases its accuracy in a 10%, a result roughly 1% lower than the kNN one, computing just a fourth of the maximum number of distances.

The CHC evolutionary algorithm, just as the EN and FN methods when set to 0.1, performs one of the highest reduction rates as depicted in the 1NN case, in which the number of distances is reduced to just the 2% of the maximum, obtaining an accuracy close to 82% of the total. As in the other selection algorithms, when applying the kNNc method to CHC there is a general accuracy

Table 1
Average results obtained when no noise is added to the datasets. Bold elements correspond to the non-dominated points. Normalized results (%) of the different algorithms are obtained referring to the ALL method with the same *k* value.

k	Algorithm	Red. set size	Accuracy (%)			Upper Bound (%)		Distances (%)			Time (%)		
			PS	kNN2	kNN3	kNN2	kNN3	PS	kNN2	kNN3	PS	kNN2	kNN3
1	ALL	6898.7	90.6	90.6	90.6	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	ED	6231.2	89.4	90.2	90.4	95.5	97.4	90.6	91.6	92.0	90.7	103.4	107.7
	MED	6231.7	89.4	90.2	90.4	95.5	97.4	90.5	91.5	92.1	91.6	103.4	109.9
	MCNN	849.0	**86.9**	89.9	90.2	94.6	96.8	**11.8**	22.4	26.6	12.8	25.6	30.8
	CNN	1589.3	87.0	90.2	90.5	95.9	98.0	22.7	31.5	36.1	24.1	36.0	42.8
	MFCNN	830.9	86.8	90.0	90.4	94.9	97.0	11.7	21.9	26.9	12.1	24.8	29.4
	FCNN	1537.9	87.0	90.2	90.5	95.8	98.0	22.0	30.8	35.3	22.9	35.2	40.9
	1-FN$_{0.10}$	233.8	79.6	86.1	88.1	89.9	93.8	3.5	14.4	19.9	3.1	14.5	20.5
	1-FN$_{0.20}$	539.0	84.1	88.2	89.3	92.6	95.4	7.9	18.4	23.7	7.4	19.6	24.9
	1-FN$_{0.30}$	928.8	85.9	89.1	89.8	93.8	96.3	13.7	23.6	28.6	13.1	25.5	30.6
	1-NE$_{0.10}$	106.6	**75.3**	83.4	85.7	87.1	91.1	**1.6**	12.8	18.4	1.4	13.3	19.3
	1-NE$_{0.20}$	271.5	81.4	87.0	88.2	91.2	94.2	4.1	15.0	20.4	3.6	15.6	21.1
	1-NE$_{0.30}$	512.9	**84.9**	88.4	89.3	93.0	95.8	**7.7**	18.2	23.4	7.0	18.7	24.7
	2-FN$_{0.10}$	228.2	79.1	86.0	87.9	89.9	93.5	3.4	14.4	19.9	3.1	14.4	21.1
	2-FN$_{0.20}$	522.8	83.4	88.1	89.2	92.6	95.5	7.8	18.3	23.6	7.3	19.7	24.3
	2-FN$_{0.30}$	896.7	85.5	89.0	89.8	93.7	96.4	13.3	23.3	28.2	12.8	25.2	30.3
	2-NE$_{0.10}$	102.5	74.2	82.9	85.3	86.6	90.8	1.6	12.8	18.3	1.3	13.3	19.5
	2-NE$_{0.20}$	255.4	80.7	86.6	88.1	90.8	94.1	3.9	14.8	20.3	3.4	15.2	21.7
	2-NE$_{0.30}$	480.0	84.4	88.3	89.3	93.0	95.7	7.3	17.8	23.1	6.7	18.9	25.4
	DROP3	759.8	81.6	88.1	89.3	92.8	95.9	10.5	20.8	26.0	9.3	19.0	23.7
	ICF	987.3	71.8	81.8	85.0	85.9	91.0	14.2	24.2	29.2	14.1	23.7	27.9
	CHC	158.1	**81.7**	86.9	88.6	90.5	94.1	**2.3**	13.4	19.0	2.0	11.5	16.1
3	ALL	6898.7	90.9	90.9	90.9	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	ED	6232.0	89.4	90.4	90.6	95.4	97.4	90.6	91.6	92.1	93.8	106.7	111.7
	MED	6231.9	89.4	90.4	90.6	95.5	97.4	90.5	91.5	92.1	93.6	106.2	111.6
	MCNN	813.3	**85.8**	**89.8**	90.2	94.2	96.6	**10.9**	**21.2**	26.1	11.8	24.1	30.1
	CNN	1611.5	87.1	90.5	**90.9**	96.0	98.1	23.0	31.9	**36.1**	25.4	38.8	43.6
	MFCNN	831.3	86.8	**90.3**	**90.6**	94.9	97.1	11.7	**21.9**	**26.9**	12.7	25.5	30.9
	FCNN	1537.7	87.0	90.3	**90.8**	95.8	98.0	22.0	30.9	**35.3**	23.3	35.6	41.8
	1-FN$_{0.10}$	233.9	79.7	86.4	88.3	90.0	93.7	3.5	14.4	19.9	3.2	15.3	21.1
	1-FN$_{0.20}$	540.0	83.8	88.3	89.5	92.5	95.4	8.0	18.4	23.7	7.6	19.9	25.7
	1-FN$_{0.30}$	930.8	85.8	89.2	90.0	93.7	96.3	13.7	23.6	28.6	13.5	26.0	31.6
	1-NE$_{0.10}$	106.7	75.2	83.4	85.7	86.9	91.0	1.6	12.8	18.4	1.4	14.0	19.7
	1-NE$_{0.20}$	270.8	81.4	**87.1**	88.3	91.3	94.2	4.1	**15.0**	20.4	3.7	15.9	22.5
	1-NE$_{0.30}$	512.5	84.9	**88.7**	89.6	93.1	95.8	7.7	**18.2**	23.4	7.3	19.7	25.9
	2-FN$_{0.10}$	227.9	79.5	86.2	88.1	89.9	93.5	3.4	14.4	19.9	3.1	15.1	21.0
	2-FN$_{0.20}$	522.5	83.4	88.2	89.5	92.6	95.5	7.8	18.3	23.6	7.4	19.2	25.2
	2-FN$_{0.30}$	896.5	85.4	89.1	90.0	93.7	96.3	13.3	23.3	28.2	12.9	25.0	31.4
	2-NE$_{0.10}$	102.4	74.3	83.2	85.5	86.7	90.8	1.6	12.8	18.3	1.3	13.1	19.4
	2-NE$_{0.20}$	255.2	80.7	86.7	88.2	90.8	94.1	3.9	14.8	20.3	3.5	16.0	22.4
	2-NE$_{0.30}$	479.2	84.4	88.6	**89.6**	93.0	95.8	7.3	17.8	**23.1**	6.6	19.0	25.3
	DROP3	513.1	78.3	86.2	88.3	90.2	94.2	7.0	17.6	23.0	6.4	16.3	21.0
	ICF	917.4	73.0	82.3	85.2	86.0	90.7	13.4	23.5	28.5	13.4	22.6	27.9
	CHC	236.3	81.2	86.8	88.4	90.5	93.9	3.6	14.6	20.1	3.2	12.7	17.6
5	ALL	6898.7	90.7	90.7	90.7	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	ED	6231.2	89.4	90.1	90.5	95.5	97.4	90.6	91.5	92.1	90.6	103.2	107.9
	MED	6231.4	89.4	90.1	90.5	95.5	97.4	90.5	91.5	92.1	91.4	102.6	110.3
	MCNN	799.0	85.5	**89.2**	90.0	93.9	96.5	10.6	**20.9**	25.9	11.3	23.3	29.5
	CNN	1599.1	87.1	90.2	90.7	96.0	98.1	22.8	31.7	36.0	24.0	36.4	42.0
	MFCNN	832.9	86.8	89.8	90.4	94.9	97.0	11.7	21.9	26.9	12.3	24.2	30.6

	FCNN	1538.1	87.1	90.0	90.6	95.8	98.0	22.0	30.9	35.3	23.0	35.4	40.7
	1-FN$_{0.10}$	233.5	80.1	86.3	88.2	90.1	93.7	3.5	14.4	19.9	3.2	15.2	21.2
	1-FN$_{0.20}$	539.9	84.0	88.2	89.4	92.6	95.4	8.0	18.5	23.7	7.6	19.7	25.7
	1-FN$_{0.30}$	929.8	85.9	89.0	89.8	93.8	96.2	13.7	23.6	28.6	13.1	25.2	30.8
	1-NE$_{0.10}$	106.5	75.2	83.3	85.7	86.9	91.1	1.6	12.8	18.4	1.3	13.2	19.0
	1-NE$_{0.20}$	271.0	81.5	86.9	88.2	91.2	94.3	4.1	15.0	20.4	3.7	16.1	21.8
	1-NE$_{0.30}$	512.8	84.9	88.5	89.5	93.0	95.8	7.7	18.2	23.4	7.0	19.1	25.0
	2-FN$_{0.10}$	227.7	79.5	86.1	88.0	90.0	93.5	3.4	14.4	19.9	3.0	14.7	20.6
	2-FN$_{0.20}$	523.1	83.4	88.1	89.3	92.6	95.5	7.8	18.3	23.6	7.4	19.2	25.2
	2-FN$_{0.30}$	897.4	85.4	88.9	89.8	93.7	96.4	13.3	23.3	28.2	12.6	24.5	31.1
	2-NE$_{0.10}$	102.5	74.2	83.0	85.3	86.7	90.7	1.6	12.8	18.3	1.3	13.1	19.4
	2-NE$_{0.20}$	255.5	80.9	86.6	88.1	90.9	94.1	3.9	14.8	20.3	3.4	15.7	20.8
	2-NE$_{0.30}$	480.0	84.4	88.3	89.4	93.0	95.8	7.3	17.8	23.1	6.4	18.5	24.0
	DROP3	466.4	77.6	85.4	87.7	89.3	93.6	6.3	17.1	22.5	5.7	15.3	20.0
	ICF	898.7	73.2	82.3	85.1	86.1	90.8	13.2	23.3	28.3	13.1	22.4	26.9
	CHC	265.2	80.9	87.0	88.6	90.8	94.3	4.1	15.1	20.6	3.6	13.2	17.8
7	ALL	6898.7	90.3	90.3	90.3	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	ED	6227.0	89.4	89.7	90.1	95.5	97.4	90.5	91.6	92.0	92.0	103.9	110.4
	MED	6231.6	89.4	89.7	90.1	95.5	97.4	90.5	91.5	92.1	92.2	103.1	111.8
	MCNN	796.6	85.2	88.9	89.6	93.8	96.5	10.6	20.8	25.9	11.2	23.2	29.2
	CNN	1608.8	87.2	89.9	90.2	95.9	98.0	22.9	31.7	36.2	24.1	36.5	41.8
	MFCNN	831.0	86.8	89.6	90.0	94.9	97.1	11.7	21.9	26.9	12.3	24.9	30.5
	FCNN	1539.2	87.1	89.8	90.2	95.8	98.0	22.0	30.9	35.3	23.2	35.4	41.5
	1-FN$_{0.10}$	233.8	79.7	85.9	88.0	89.9	93.9	3.5	14.4	19.9	3.2	16.0	21.1
	1-FN$_{0.20}$	539.5	84.0	88.0	89.0	92.6	95.4	8.0	18.5	23.7	7.6	19.7	25.7
	1-FN$_{0.30}$	929.8	85.9	88.7	89.5	93.8	96.3	13.7	23.6	28.6	13.5	26.3	31.3
	1-NE$_{0.10}$	106.7	75.0	83.2	85.4	86.9	91.1	1.6	12.8	18.4	1.4	13.4	19.9
	1-NE$_{0.20}$	271.4	81.4	86.8	88.0	91.3	94.3	4.1	15.0	20.4	3.7	15.8	21.8
	1-NE$_{0.30}$	513.3	84.9	88.3	89.2	93.0	95.8	7.7	18.2	23.4	7.1	19.2	25.3
	2-FN$_{0.10}$	228.0	79.0	85.9	87.7	89.9	93.4	3.4	14.4	19.9	3.1	14.9	20.8
	2-FN$_{0.20}$	522.2	83.4	87.9	89.0	92.6	95.5	7.8	18.3	23.6	7.5	19.3	26.1
	2-FN$_{0.30}$	895.5	85.5	88.7	89.6	93.8	96.4	13.3	23.3	28.2	12.5	24.3	30.5
	2-NE$_{0.10}$	102.2	74.4	82.7	85.1	86.6	90.8	1.6	12.8	18.3	1.3	13.2	19.5
	2-NE$_{0.20}$	255.0	80.8	86.5	87.8	90.9	94.1	3.9	14.8	20.3	3.4	15.6	21.6
	2-NE$_{0.30}$	479.2	84.4	88.1	89.0	93.0	95.8	7.3	17.8	23.1	6.6	18.4	25.3
	DROP3	478.9	79.2	86.5	88.4	90.7	94.7	6.6	17.3	22.6	5.9	15.7	20.5
	ICF	887.9	73.6	82.2	84.8	86.0	90.6	13.0	23.1	28.2	13.1	22.4	26.8
	CHC	293.2	79.9	86.7	88.2	90.6	94.2	4.5	15.4	20.9	4.0	13.6	18.3

improvement of about 6% and 8% for $c=2$ and $c=3$ respectively, together with a 10% and 15% increase in the number of distances for the same cases.

On average, the accuracy improvement is more significant when passing from the basic PS scheme to the kNNc one than the gain obtained by increasing the number of proposals c, contrasting with the noticeable accuracy rise in the upper bounds in the same situation. This fact clearly points out that the major issue remains at the classification stage since, although the kNNc step is able to give highly accurate recommendations, the overall performance is not capable of reaching these upper limits.

The upper bound ratio does improve as the c value increases since a larger number c of classes are recommended. An increase in this c parameter causes a fixed rise in the number of distances to be computed since the classes in the datasets proposed are balanced. However, as it can be checked in the results, there is not such a linear relation between the upper bound figure and the number of distances computed: for instance, in MFCNN with $k=5$, the upper bounds are 94.9% for $c=2$ and 97% for $c=3$ with 21.9% and 26.9% of distances respectively, depicting that this 2% improvement is around a 5% increase in terms of computational cost but in order achieve a 100% upper bound (the remaining 3%), almost an additional figure of 73% of distances has to be computed. This non-linear behavior, which can be checked in all the other configurations as well, shows a clear dependency with the PS strategy used: a certain PS algorithm with an outstanding performance would require an elevated number of distances to show an improvement whereas an algorithm with a poor performance might exhibit a remarkable accuracy upturn without such distances increase. As a consequence, as the commented upper bounds are the ones which depict the maximum theoretical classification figures which can be expected, the obtained accuracy does also show this non-linearity with respect to the number of distances.

Finally, the increase in the k value does not have any noticeable effect on the accuracy obtained by each algorithm, possibly due to the fact that the datasets are hardly noisy.

As aforementioned, the PS-based classification can be seen as a MOP problem in which accuracy and distances computed have to be simultaneously optimized despite being typically opposed goals. Results of the strategies considered are shown in Fig. 2 facing these two metrics. Optimal solutions, defined using the non-dominance criterion described in Section 4, are remarked in this figure as well as being highlighted in Table 1. Since most of the algorithms gather in a small area, this particular region has been widened for a better visualization.

A first interesting outcome withdrawn from applying this criterion is that the kNN algorithm (with no PS) does not belong to the optimal set of solutions since kNN3 CNN scheme achieves the same accuracy with a lower number of distances computed.

Moreover, it can be also observed that, except for editing approaches, each main scheme – PS, kNN2 and kNN3, drawn in red, green and blue points respectively – entails a cloud of points in different regions of the space. Therefore, kNNc scheme is providing a great range of new options in the trade-off between distances and accuracy not explored in previous works. Furthermore, many kNN2 and kNN3 strategies are found within the optimal set of solutions. Therefore, the user is provided with a wide range of options from which to choose depending on the metric to emphasize (distances or accuracy). For example, let us assume a scenario like that depicted in Fig. 2 in which we are restricted to perform at maximum 25% of the distances. Thanks to 3NN2 scheme with MFCNN prototype selection, we could achieve an accuracy of 90.3% (just 0.6% below the best accuracy) with around 22% of distances computed.

5.2. *Noisy scenario*

In this subsection the figures obtained when synthetic noise is added are presented. Experimentation was carried out for each of the noise configurations considered in Section 4. As results show a qualitatively similar trend along these noise possibilities, remarks will not focus on a particular configuration but on the general behavior. In addition, and due to space limitations, results of only two of the noise scenarios tackled are shown: an intermediate situation (20% noise rate scenario), for which results can be verified in Table 2, and one for the most adverse situation considered (40% of synthetic noise rate), whose results can be checked in Table 3.

Synthetic noise addition to the samples changes the previous situation drastically. kNN, which scored the best results for each single k value, now exhibits a remarkable decrease in accuracy, becoming more noticeable as the noise rate is increased. As expected, the use of different k values does improve the accuracy, especially in the case of $k=7$, in which kNN scores the maximum classification rate, drawing in these terms with ED.

ED and MED algorithms are able to manage this noisy situation: the size reduction is sharper than in the previous case since the elements removed are the ones leading to confusion. As it happened in the non-added noise scenario, the use of kNNc with these algorithms does not carry a remarkable improvement in accuracy, though it does in the classification upper bounds, getting even to the point of decreasing the performance when low k values are used.

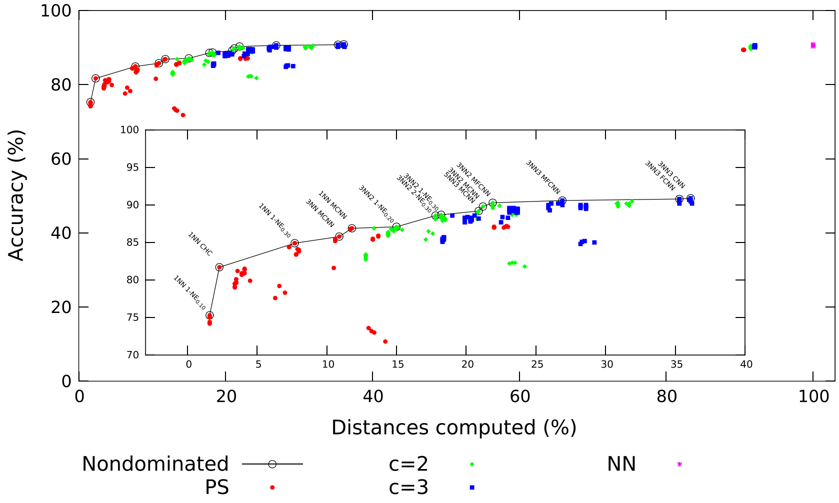


Fig. 2. Distances computed (%) against accuracy achieved by the algorithms. Average results when no noise is added to the samples. The non-dominated front is remarked.

Table 2
Average results obtained when 20% of noise is added to the datasets. Bold elements correspond to the non-dominated points. Normalized results (%) of the different algorithms are obtained referring to the ALL method with the same *k* value.

k	Algorithm	Red. set size	Accuracy (%)			Upper Bound (%)		Distances (%)			Time (%)		
			PS	kNN2	kNN3	kNN2	kNN3	PS	kNN2	kNN3	PS	kNN2	kNN3
1	ALL	6898.7	73.1	73.1	73.1	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	ED	4329.4	88.1	88.3	87.8	94.6	96.8	62.8	66.9	69.0	63.0	76.1	80.1
	MED	4338.1	88.1	88.1	87.8	94.6	96.8	63.0	67.0	69.1	63.7	76.5	81.3
	MCNN	713.6	84.2	87.7	87.5	93.8	96.1	10.0	20.6	25.0	10.2	22.7	27.9
	CNN	4094.3	64.4	70.9	72.4	87.4	94.6	59.2	63.8	66.4	60.6	73.0	79.3
	MFCNN	673.5	**84.6**	87.6	87.6	93.6	96.2	**9.6**	19.8	25.0	9.8	22.3	27.4
	FCNN	3953.2	63.9	70.7	72.3	87.2	94.4	57.1	61.9	64.3	57.7	68.8	77.4
	1-FN$_{0.10}$	288.5	81.6	85.6	86.2	90.5	93.8	4.2	15.0	20.5	3.8	15.8	21.8
	1-FN$_{0.20}$	683.9	84.5	87.4	87.4	92.9	95.7	9.9	20.1	25.2	9.2	21.8	26.8
	1-FN$_{0.30}$	1157.0	85.2	87.4	87.4	93.6	96.3	16.8	26.2	30.9	15.5	27.3	33.9
	1-NE$_{0.10}$	233.3	80.9	84.9	85.7	89.8	93.2	3.4	14.3	19.7	2.9	14.8	20.3
	1-NE$_{0.20}$	574.6	**84.4**	87.2	87.1	92.8	95.4	**8.2**	18.6	23.8	7.4	19.3	25.3
	1-NE$_{0.30}$	1022.7	85.6	87.7	87.4	93.8	96.2	14.7	24.3	29.1	13.3	25.4	31.1
	2-FN$_{0.10}$	257.5	**81.6**	85.5	86.2	90.4	93.9	**3.8**	14.7	20.1	3.3	14.9	20.6
	2-FN$_{0.20}$	609.8	**84.5**	87.2	87.1	92.7	95.5	**8.9**	19.3	24.4	8.1	20.0	26.1
	2-FN$_{0.30}$	1061.7	85.0	87.6	87.3	93.8	96.2	15.5	25.1	29.9	14.2	26.0	32.6
	2-NE$_{0.10}$	186.9	80.2	84.6	85.6	89.5	93.0	2.8	13.7	19.2	2.4	14.5	20.5
	2-NE$_{0.20}$	472.5	84.0	87.0	86.9	92.4	95.2	6.9	17.3	22.6	6.2	18.1	24.8
	2-NE$_{0.30}$	864.5	85.2	87.7	87.3	93.6	96.2	12.5	22.4	27.3	11.5	24.0	29.9
	DROP3	759.8	68.7	80.6	83.4	86.2	92.2	10.5	20.8	26.0	9.3	18.9	23.7
	ICF	987.3	57.5	71.8	76.6	77.3	86.3	14.2	24.2	29.2	14.0	23.4	28.1
	CHC	160.5	**67.8**	79.2	81.7	84.3	89.8	**2.4**	13.4	19.0	1.9	11.4	16.1
3	ALL	6898.7	82.9	82.9	82.9	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	ED	4321.9	88.1	89.5	89.4	94.7	96.8	62.7	67.1	68.7	62.7	74.7	80.8
	MED	4334.9	87.7	**89.5**	89.6	94.6	96.8	62.9	**66.9**	69.1	62.9	74.9	81.1
	MCNN	679.1	82.8	88.6	89.3	93.2	96.0	9.2	19.6	24.7	9.6	21.5	28.0
	CNN	4081.8	64.3	82.4	82.7	87.3	94.6	59.1	63.8	66.2	59.4	71.5	77.8
	MFCNN	675.5	84.5	**88.8**	89.3	93.5	96.2	9.6	**19.8**	25.0	9.8	21.7	28.0
	FCNN	3951.3	63.9	82.1	82.6	87.2	94.4	57.0	61.9	64.3	57.4	69.7	75.7
	1-FN$_{0.10}$	288.4	81.6	86.7	87.9	90.7	93.8	4.2	15.0	20.5	3.7	15.8	21.7
	1-FN$_{0.20}$	682.6	84.5	88.3	89.1	92.9	95.8	9.9	20.1	25.2	9.0	20.4	26.9
	1-FN$_{0.30}$	1155.8	85.1	88.8	89.2	93.7	96.3	16.7	26.2	30.9	15.7	27.7	34.0
	1-NE$_{0.10}$	233.5	80.9	**85.9**	87.2	89.9	93.2	3.4	**14.3**	19.7	2.9	14.4	21.2
	1-NE$_{0.20}$	575.0	84.3	**88.2**	**88.9**	92.8	95.4	8.3	**18.6**	**23.8**	7.5	19.3	26.0
	1-NE$_{0.30}$	1023.3	85.5	88.8	89.3	93.7	96.2	14.7	24.3	29.1	13.6	25.8	31.8
	2-FN$_{0.10}$	257.9	81.5	**86.5**	87.7	90.5	93.9	3.8	**14.7**	20.1	3.4	15.4	21.1
	2-FN$_{0.20}$	610.3	84.5	88.2	88.9	92.8	95.5	8.9	19.3	24.4	8.2	20.1	26.1
	2-FN$_{0.30}$	1061.6	85.0	88.9	89.2	93.8	96.3	15.5	25.1	29.9	14.1	26.0	32.2
	2-NE$_{0.10}$	187.6	**80.3**	**85.6**	87.0	89.6	93.0	**2.8**	**13.7**	19.2	2.3	13.9	20.3
	2-NE$_{0.20}$	472.7	**84.1**	**88.0**	88.7	92.4	95.1	**6.9**	**17.3**	22.6	6.1	18.2	23.9
	2-NE$_{0.30}$	864.4	**85.4**	88.8	89.3	93.6	96.2	**12.5**	22.4	27.3	11.5	24.1	29.8
	DROP3	513.1	65.5	79.7	83.5	83.7	90.1	7.0	17.6	23.0	6.2	15.9	20.5
	ICF	917.4	58.8	74.1	79.3	77.8	86.1	13.4	23.5	28.5	13.1	22.4	27.1
	CHC	238.8	68.8	82.0	85.5	84.9	90.4	3.6	14.6	20.1	3.0	12.4	16.8
5	ALL	6898.7	87.6	87.6	87.6	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	ED	4331.0	87.7	89.3	88.8	94.6	96.8	62.9	67.0	68.9	63.6	75.8	82.1
	MED	4325.3	87.9	89.1	**89.7**	94.6	96.8	62.8	66.8	**69.0**	63.3	74.9	81.9
	MCNN	669.4	82.8	88.2	**89.3**	93.1	96.1	9.0	19.4	**24.4**	9.6	22.2	28.1
	CNN	4102.3	64.4	82.9	87.0	87.3	94.6	59.3	63.9	66.5	59.8	72.0	78.3
	MFCNN	676.3	84.5	88.6	**89.4**	93.5	96.3	9.6	19.9	**25.0**	9.8	21.9	27.9

Table 2 (*continued*)

k	Algorithm	Red. set size	Accuracy (%)			Upper Bound (%)		Distances (%)			Time (%)		
			PS	kNN2	kNN3	kNN2	kNN3	PS	kNN2	kNN3	PS	kNN2	kNN3
	FCNN	3953.7	63.9	82.6	86.9	87.2	94.4	57.1	61.9	64.3	58.3	71.7	76.2
	1-FN$_{0.10}$	288.6	**81.7**	86.5	87.9	90.7	93.8	**4.2**	15.0	20.5	3.8	15.8	21.5
	1-FN$_{0.20}$	683.2	84.5	88.0	89.1	92.9	95.8	9.9	20.1	25.2	9.3	21.5	27.6
	1-FN$_{0.30}$	1156.4	85.2	88.5	89.3	93.6	96.3	16.7	26.2	30.9	15.7	28.0	33.7
	1-NE$_{0.10}$	233.0	80.9	85.6	87.3	89.8	93.2	3.4	14.3	19.7	2.9	15.1	20.9
	1-NE$_{0.20}$	574.7	84.4	87.9	88.9	92.8	95.4	8.2	18.6	23.8	7.5	19.6	25.7
	1-NE$_{0.30}$	1022.5	85.7	88.6	89.4	93.7	96.2	14.7	24.3	29.1	13.6	26.1	31.5
	2-FN$_{0.10}$	257.7	81.5	86.3	87.8	90.5	93.9	3.8	14.7	20.1	3.4	15.3	21.4
	2-FN$_{0.20}$	609.9	84.5	88.0	89.0	92.7	95.5	8.9	19.3	24.4	8.2	20.6	26.0
	2-FN$_{0.30}$	1061.2	85.0	88.6	89.3	93.7	96.2	15.5	25.1	29.9	14.2	25.5	32.9
	2-NE$_{0.10}$	187.2	80.3	85.4	87.1	89.5	93.1	2.8	13.7	19.2	2.4	15.0	20.9
	2-NE$_{0.20}$	472.2	84.0	87.8	88.7	92.5	95.2	6.9	17.3	22.6	6.1	17.9	24.1
	2-NE$_{0.30}$	863.9	85.4	88.6	89.3	93.6	96.2	12.5	22.4	27.3	11.6	24.1	29.9
	DROP3	466.4	65.3	79.2	83.6	83.1	89.5	6.3	17.1	22.5	5.7	15.4	19.9
	ICF	898.7	58.9	74.2	79.7	77.8	86.1	13.2	23.3	28.3	13.2	22.6	27.4
	CHC	269.0	67.9	81.3	84.9	85.2	90.9	4.2	15.1	20.6	3.6	13.2	17.8
7	ALL	6898.7	88.1	88.1	88.1	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	ED	4335.3	88.1	89.0	89.5	94.6	96.8	62.9	66.9	69.0	63.2	76.2	80.3
	MED	4335.1	87.2	89.0	89.5	94.6	96.8	62.9	67.0	69.1	62.5	74.3	80.9
	MCNN	673.3	82.0	88.0	89.0	93.1	96.0	9.0	19.4	24.4	9.4	21.9	27.6
	CNN	4092.4	64.3	82.7	87.2	87.3	94.5	59.2	63.9	66.3	59.2	71.1	78.0
	MFCNN	678.7	84.3	88.4	89.2	93.5	96.2	9.6	19.9	25.0	9.9	21.6	28.3
	FCNN	3949.4	63.9	82.5	87.2	87.2	94.4	57.0	61.8	64.3	56.9	69.6	74.6
	1-FN$_{0.10}$	288.8	81.7	86.3	87.6	90.6	94.0	4.2	15.1	20.5	3.7	15.6	21.4
	1-FN$_{0.20}$	683.2	84.6	87.9	88.9	92.9	95.8	9.9	20.1	25.2	9.1	21.3	26.6
	1-FN$_{0.30}$	1156.7	85.2	88.3	89.1	93.6	96.3	16.7	26.2	30.9	15.6	27.7	33.6
	1-NE$_{0.10}$	233.1	**81.0**	85.5	87.1	89.8	93.2	**3.4**	14.3	19.7	2.9	14.6	20.8
	1-NE$_{0.20}$	574.0	84.4	87.8	88.6	92.8	95.4	8.2	18.6	23.8	7.4	19.2	25.6
	1-NE$_{0.30}$	1021.9	85.6	88.5	89.2	93.7	96.2	14.7	24.3	29.1	13.7	25.8	32.6
	2-FN$_{0.10}$	257.8	81.6	86.1	87.6	90.4	93.9	3.8	14.7	20.1	3.4	15.4	21.7
	2-FN$_{0.20}$	609.4	84.5	87.9	88.7	92.7	95.5	8.9	19.3	24.4	8.1	19.7	26.3
	2-FN$_{0.30}$	1060.6	85.0	88.5	89.0	93.7	96.2	15.5	25.1	29.9	14.1	25.9	32.7
	2-NE$_{0.10}$	187.3	80.3	85.2	86.8	89.6	93.1	2.8	13.7	19.2	2.3	14.4	19.9
	2-NE$_{0.20}$	472.8	84.1	87.5	88.5	92.3	95.1	6.9	17.4	22.6	6.2	18.2	24.4
	2-NE$_{0.30}$	864.4	85.4	88.4	89.1	93.6	96.2	12.5	22.4	27.3	11.4	23.9	29.5
	DROP3	478.9	66.4	80.6	85.0	83.6	89.9	6.6	17.3	22.6	5.7	15.3	19.9
	ICF	887.9	59.3	74.2	79.7	77.9	86.2	13.0	23.1	28.2	12.9	22.1	26.9
	CHC	296.2	67.8	81.7	85.7	84.7	90.7	4.5	15.5	20.9	3.9	13.5	18.0

Table 3
Average results obtained when 40% of noise is added to the datasets. Bold elements correspond to the non-dominated points. Normalized results (%) of the different algorithms are obtained referring to the ALL method with the same *k* value.

k	Algorithm	Red. set size	Accuracy			Upper bound		Distances			Time		
			PS	kNN2	kNN3	kNN2	kNN3	PS	kNN2	kNN3	PS	kNN2	kNN3
1	ALL	6898.7	60.4	60.4	60.4	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	ED	3037.8	85.1	85.4	83.9	93.4	96.1	44.1	50.2	53.5	44.0	55.1	63.2
	MED	3026.8	85.2	85.2	84.0	93.5	96.1	44.0	50.2	53.3	44.8	57.0	63.5
	MCNN	639.1	80.6	84.2	83.4	92.1	95.1	8.9	19.6	24.1	9.0	21.3	27.0
	CNN	5096.7	52.8	57.5	59.2	78.5	89.5	73.9	76.9	78.6	75.5	87.3	94.5
	MFCNN	596.3	80.4	83.7	83.7	92.2	95.6	8.4	18.7	24.0	8.5	20.8	26.9
	FCNN	4989.8	52.4	57.3	59.1	78.4	89.4	72.2	75.4	76.9	71.8	84.0	89.6
	1-FN$_{0.10}$	332.7	81.5	84.2	83.7	90.5	93.9	4.8	15.6	21.0	4.5	16.5	22.7
	1-FN$_{0.20}$	789.9	82.7	84.9	84.1	92.3	95.3	11.4	21.4	26.4	10.7	23.1	28.9
	1-FN$_{0.30}$	1342.9	78.1	81.1	81.9	91.4	95.1	19.4	28.5	33.1	18.3	30.4	36.7
	1-NE$_{0.10}$	304.6	81.6	84.4	83.7	90.6	93.9	4.4	15.2	20.6	3.9	16.0	21.7
	1-NE$_{0.20}$	740.8	83.2	85.0	84.1	92.4	95.5	10.7	20.7	25.8	9.8	22.4	27.5
	1-NE$_{0.30}$	1280.8	78.9	81.5	82.3	91.9	95.3	18.4	27.7	32.2	17.4	29.5	35.8
	2-FN$_{0.10}$	291.4	**81.8**	84.3	83.9	90.6	93.9	**4.3**	15.1	20.5	3.7	15.4	21.6
	2-FN$_{0.20}$	700.3	82.6	84.8	84.1	92.2	95.3	10.2	20.3	25.4	9.1	20.9	27.1
	2-FN$_{0.30}$	1187.9	81.1	83.7	83.1	92.4	95.5	17.2	26.6	31.3	15.3	27.1	33.4
	2-NE$_{0.10}$	246.9	81.5	84.2	83.5	90.3	93.4	3.6	14.5	19.9	3.0	14.4	20.2
	2-NE$_{0.20}$	613.4	**83.4**	85.2	84.1	92.6	95.6	**8.9**	19.1	24.3	7.7	18.7	25.7
	2-NE$_{0.30}$	1087.8	82.0	84.1	83.4	92.7	95.8	15.7	25.2	30.0	13.5	25.5	30.6
	DROP3	759.8	56.6	70.8	74.9	77.5	86.9	10.5	20.8	26.0	9.3	19.0	23.7
	ICF	987.3	47.3	62.7	69.6	68.3	80.7	14.2	24.2	29.2	14.0	23.6	28.1
	CHC	157.5	**54.5**	69.1	73.5	73.2	81.5	**2.3**	13.4	19.0	1.9	11.6	16.2
3	ALL	6898.7	72.5	72.5	72.5	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	ED	3043.8	85.5	87.8	87.7	93.5	96.1	44.2	50.4	53.5	45.3	57.9	64.2
	MED	3038.8	85.1	86.8	87.7	93.5	96.1	44.1	50.2	53.5	45.0	56.7	64.6
	MCNN	613.2	78.9	85.3	87.0	91.4	95.0	8.3	18.8	23.8	8.8	20.9	28.2
	CNN	5088.0	52.8	73.5	71.9	78.6	89.5	73.8	76.9	78.4	75.7	87.9	94.5
	MFCNN	596.9	81.2	86.8	87.4	92.2	95.6	8.4	18.8	23.9	8.8	21.5	27.7
	FCNN	4989.2	52.4	73.1	71.7	78.3	89.4	72.1	75.4	76.9	73.7	85.9	92.4
	1-FN$_{0.10}$	333.1	81.5	85.8	86.6	90.5	93.8	4.9	15.6	21.0	4.5	17.1	22.6
	1-FN$_{0.20}$	790.3	82.7	87.2	87.4	92.4	95.4	11.4	21.4	26.4	10.8	23.2	29.3
	1-FN$_{0.30}$	1343.0	78.4	86.2	86.5	91.5	95.1	19.4	28.5	33.1	18.6	31.2	37.1
	1-NE$_{0.10}$	304.7	81.6	86.0	86.7	90.5	94.0	4.4	15.2	20.6	3.8	15.8	21.6
	1-NE$_{0.20}$	741.3	83.2	87.3	87.5	92.4	95.6	10.7	20.7	25.8	9.9	21.8	29.6
	1-NE$_{0.30}$	1281.6	79.5	86.9	86.7	91.9	95.2	18.4	27.7	32.2	17.3	29.6	35.8
	2-FN$_{0.10}$	291.3	81.8	**86.0**	86.9	90.6	94.0	4.3	**15.1**	20.5	3.8	16.1	22.1
	2-FN$_{0.20}$	700.6	82.5	86.9	87.4	92.1	95.2	10.2	20.3	25.4	9.2	21.3	27.6
	2-FN$_{0.30}$	1188.2	81.0	87.1	87.0	92.5	95.5	17.2	26.6	31.3	16.1	28.8	34.7
	2-NE$_{0.10}$	247.3	81.5	**85.8**	86.4	90.4	93.5	3.6	**14.5**	19.9	3.0	14.8	20.8
	2-NE$_{0.20}$	614.0	83.3	87.2	87.5	92.5	95.5	8.9	19.1	24.3	8.0	20.2	26.0
	2-NE$_{0.30}$	1088.3	82.0	87.2	87.4	92.7	95.8	15.7	25.2	30.0	13.6	25.8	31.1
	DROP3	513.1	55.0	72.3	78.5	74.9	84.0	7.0	17.6	23.0	6.3	15.9	20.8
	ICF	917.4	48.2	66.3	74.3	69.1	81.0	13.4	23.5	28.5	13.3	22.7	27.8
	CHC	237.4	55.9	73.2	79.2	75.3	83.5	3.6	14.6	20.1	3.1	12.7	17.6
5	ALL	6898.7	82.8	82.8	82.8	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	ED	3038.8	85.3	87.8	88.4	93.5	96.1	44.1	50.3	53.4	44.1	56.3	62.7
	MED	3030.2	85.2	87.8	88.1	93.5	96.0	44.0	50.3	53.3	44.6	57.4	62.6
	MCNN	611.4	79.0	86.0	87.6	91.1	94.8	8.3	18.6	23.8	8.7	21.3	27.2
	CNN	5086.4	52.7	74.5	81.2	78.4	89.4	73.8	76.9	78.4	74.8	87.3	92.9
	MFCNN	594.1	81.3	**87.3**	88.2	92.3	95.6	8.4	**18.7**	24.0	8.6	20.8	27.4

Table 3 (*continued*)

k	Algorithm	Red. set size	Accuracy			Upper bound		Distances			Time		
			PS	kNN2	kNN3	kNN2	kNN3	PS	kNN2	kNN3	PS	kNN2	kNN3
	FCNN	4988.3	52.3	74.2	81.1	78.3	89.3	72.1	75.3	76.9	73.1	85.9	91.0
	1-FN$_{0.10}$	333.3	81.6	85.9	87.1	90.5	93.9	4.9	15.6	21.0	4.3	16.3	21.9
	1-FN$_{0.20}$	790.3	82.5	87.3	88.0	92.4	95.3	11.4	21.4	26.4	10.5	22.6	28.8
	1-FN$_{0.30}$	1342.9	78.7	86.7	88.0	91.5	95.1	19.4	28.5	33.1	18.5	30.7	37.0
	1-NE$_{0.10}$	304.9	81.7	86.0	87.3	90.5	94.0	4.4	15.2	20.6	3.9	16.0	21.6
	1-NE$_{0.20}$	741.0	83.1	87.4	88.0	92.4	95.6	10.7	20.7	25.8	9.6	21.7	28.0
	1-NE$_{0.30}$	1281.0	79.7	87.1	88.1	91.9	95.3	18.4	27.7	32.2	16.9	29.2	34.8
	2-FN$_{0.10}$	291.4	81.8	86.0	87.3	90.6	94.0	4.3	15.1	20.5	3.7	15.4	21.6
	2-FN$_{0.20}$	700.4	82.3	87.1	88.0	92.1	95.3	10.2	20.3	25.4	9.2	20.9	27.7
	2-FN$_{0.30}$	1187.6	81.2	87.5	88.3	92.5	95.5	17.2	26.6	31.3	15.5	27.4	33.8
	2-NE$_{0.10}$	247.4	**81.6**	85.8	86.8	90.5	93.5	**3.6**	14.5	19.9	3.0	14.5	20.7
	2-NE$_{0.20}$	613.5	83.4	**87.4**	88.1	92.6	95.5	8.9	**19.1**	24.3	7.7	19.6	25.6
	2-NE$_{0.30}$	1087.6	82.1	87.7	88.3	92.8	95.8	15.7	25.2	30.0	13.6	25.9	31.4
	DROP3	466.4	56.0	72.8	79.4	75.0	83.5	6.3	17.1	22.5	5.7	15.3	19.9
	ICF	898.7	48.3	66.8	75.5	69.1	81.2	13.2	23.3	28.3	13.1	22.6	27.2
	CHC	271.4	53.7	72.9	80.6	75.2	84.8	4.2	15.1	20.6	3.7	13.3	17.9
7	ALL	6898.7	85.5	85.5	85.5	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	ED	3045.2	85.5	87.9	88.4	93.5	96.1	44.2	50.5	53.4	43.9	56.9	61.3
	MED	3046.1	84.5	87.9	88.4	93.5	96.1	44.2	50.5	53.4	43.7	56.7	60.3
	MCNN	604.8	78.6	86.4	87.7	91.1	94.9	8.2	18.6	23.7	8.1	19.7	25.4
	CNN	5087.0	52.8	74.7	82.6	78.6	89.5	73.8	76.8	78.5	73.4	85.8	90.3
	MFCNN	593.5	81.2	87.2	**88.3**	92.2	95.5	8.4	18.7	**23.9**	8.3	20.7	25.8
	FCNN	4989.3	52.4	74.4	82.3	78.4	89.4	72.1	75.3	76.9	70.9	85.9	85.0
	1-FN$_{0.10}$	333.2	81.6	85.8	87.1	90.5	93.9	4.9	15.6	21.0	4.2	15.8	20.2
	1-FN$_{0.20}$	789.9	82.7	87.3	88.1	92.3	95.3	11.4	21.4	26.4	10.5	23.0	27.9
	1-FN$_{0.30}$	1343.1	78.4	86.5	87.9	91.5	95.1	19.4	28.5	33.1	17.4	29.3	33.7
	1-NE$_{0.10}$	304.8	81.7	85.9	87.2	90.6	94.0	4.4	15.2	20.6	3.9	16.8	20.8
	1-NE$_{0.20}$	741.0	83.2	87.4	88.2	92.5	95.6	10.7	20.7	25.8	9.0	20.5	25.1
	1-NE$_{0.30}$	1280.5	79.3	86.6	88.1	91.9	95.3	18.4	27.6	32.2	16.1	28.3	31.9
	2-FN$_{0.10}$	291.9	81.7	85.9	87.2	90.6	94.0	4.3	15.1	20.5	3.6	15.4	20.2
	2-FN$_{0.20}$	701.2	82.6	87.0	88.0	92.1	95.3	10.2	20.3	25.4	9.2	21.5	25.8
	2-FN$_{0.30}$	1188.6	81.1	87.1	88.3	92.5	95.6	17.2	26.6	31.3	15.3	28.5	31.2
	2-NE$_{0.10}$	246.8	81.6	85.7	86.7	90.3	93.4	3.6	14.5	19.9	2.9	15.1	18.2
	2-NE$_{0.20}$	613.3	83.4	87.3	88.1	92.5	95.5	8.9	19.1	24.3	7.6	18.8	24.5
	2-NE$_{0.30}$	1087.9	81.9	87.4	**88.4**	92.9	95.8	15.7	25.2	**30.0**	12.6	24.0	28.3
	DROP3	478.9	56.6	73.6	80.2	76.8	85.4	6.6	17.3	22.6	5.7	15.3	20.0
	ICF	887.9	48.8	67.1	75.6	69.4	81.1	13.0	23.1	28.2	12.9	22.4	26.9
	CHC	293.4	56.5	74.8	81.8	77.1	86.2	4.5	15.4	20.9	3.7	13.2	17.5

This particular effect is likely to happen since the samples added by our second step are, theoretically, the noisy ones discarded by the PS algorithm, thus confusing the classifier when low k values are used.

CNN and FCNN show some of the worst accuracies obtained in these experiments in terms of PS as they are very sensitive to noise: as stand-alone PS algorithms, they are not able to discard the noisy elements, thus leading to a situation in which there is neither an important size reduction nor a remarkable performance. Furthermore, the use of different k values does not upturn the accuracy results. On the other hand, the use of the second kNNc step does improve their accuracy, but still the results remain far from the classification bounds. However, as with ED and MED, introducing high k values enhances the obtained accuracy with respect to the low k values with the c class recommendation.

MFCNN and MCNN are not as affected as CNN and FCNN are at PS stage since they introduce an ED phase in the process: whereas the latter approaches obtained around 50% and 60% in terms of accuracy with around 60% and 70% of computed distances, the former algorithms do achieve precision rates around 80% with roughly 10% of the distances. The improvement obtained when using kNNc with these strategies is also noticeable with high k values. Moreover, as it happened in the non-added synthetic noise configuration, kNNc schemes are able to score results not significantly different from the ones achieved by the best performing algorithms (for instance, ED and MED with $k=7$) with just around 25% of the maximum distances computed.

EN and FN methods demonstrated to be interesting algorithms in the non-added noise scenario as they both obtained good accuracies while achieving some of the highest reduction rates. Attending now to the results obtained in the proposed noisy situations, these methods do also stand as an interesting alternative for PS as they behave amazingly well in both terms of accuracy and size, especially the 2-EN and 2-FN configurations: whereas, on average, hardly any of these algorithms score lower accuracies than 80%, the 2-EN and 2-FN ones are always able to score precision results above that mark, in some situations with distance ratios in the range from 3% to 10%. Including kNNc does improve the performance (as in the other cases, for the most part when using high-enough k values) and in spite of not outperforming other strategies (an exception to this assertion is the 2-NE$_{0.30}$ case with $k=7$ and $c=3$ in Table 3), accuracies obtained are not significantly different from the best performing algorithms. In addition, distances computed roughly range from 10% to 30% of the maximum, constituting a remarkable trade-off result. It is important to highlight that, in sight of the results obtained, these particular algorithms stand as an attractive alternative to some of the other studied methods for any noise situation as they do not perform any editing operation.

Hybrid algorithms DROP3 and ICF, just as CNN and FCNN, are not capable of coping with noisy situations either since accuracy results are similar or even lower (for instance, the ICF method in which with 40% of synthetic noise is not able to reach 50% of accuracy in any of the proposed PS schemes). However, it must be pointed out that, despite achieving similar accuracy rates, hybrid algorithms do it with a lower amount of distances: as an example, check the 7NN case with 40% of noise in which CNN achieves an accuracy of 52.8% with 73.8% of the total of distances while DROP3 gets 56.6% with just 6.6% of distances. On the other hand, adding the kNNc scheme remarkably enhances the accuracy achieved by these algorithms (in the 7NN3 configuration of ICF, the accuracy is improved in almost 30% with respect to the PS situation) but it also noticeably increases the number of distances to be computed up to, on average, 15% more.

Regarding the CHC evolutionary algorithm when tackling noisy situations, although it still shows one of the highest reduction figures amongst the studied methods (rates around 2–5%), its classification performance is significantly affected as no result is higher than 70%. In this case, the inclusion of kNNc has a similar effect to the hybrid algorithms as it shows a notorious accuracy increase (fixing $c=3$, classification figures improve around 20% and 25% for noise rates of 20% and 40% respectively) paired with a rise in the number of distances of about 10% and 15% points when setting $c=2$ and $c=3$ respectively.

In terms of the classification upper bounds defined with kNNc, as in the scenario without synthetic noise, bounds get higher as the number c of class proposals is increased. On average, and as already commented, there is not such a significant increase between $c=2$ and $c=3$ as the one observed when comparing PS and $c=2$. An exception to this remark can be observed, though, in both CNN and FCNN strategies as well as with the hybrid (DROP3 and ICF) and the evolutionary (CHC) algorithms in which, as they are not capable of coping with the noise effects, a high number c of class proposals are required.

As in the non-added noise scenario, it is possible to check the non-linear relation between the upper bound and the number of distances to be computed. For instance, for a figure noise of 40%, the 2-FN$_{0.10}$ algorithm with $k=1$ retrieves upper bounds of 90.6% for $c=2$ and 93.9% for $c=3$ with distance figures of 15.1% and 20.5% respectively, which shows that there is 3.3% of improvement with just a 5.4% increase in the total of distances to be computed. These

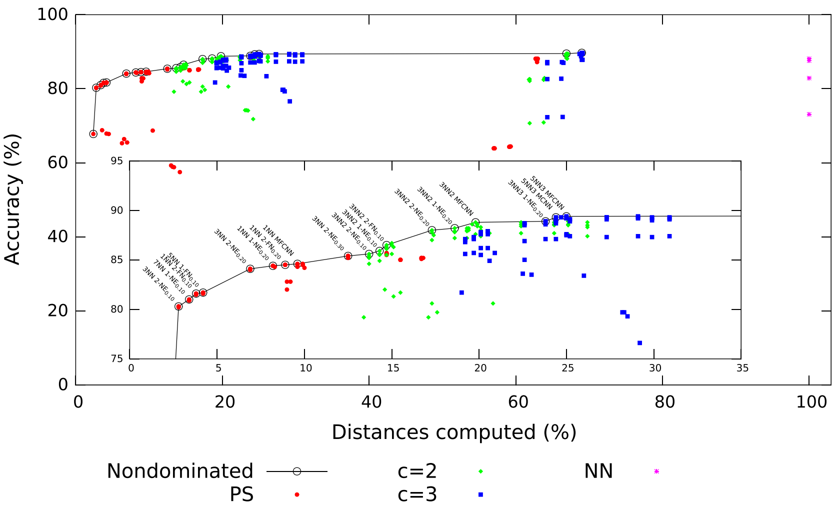


Fig. 3. Distances computed (%) against accuracy achieved by the algorithms. Average results when 20% of noise is added to the samples. The non-dominated front is remarked.

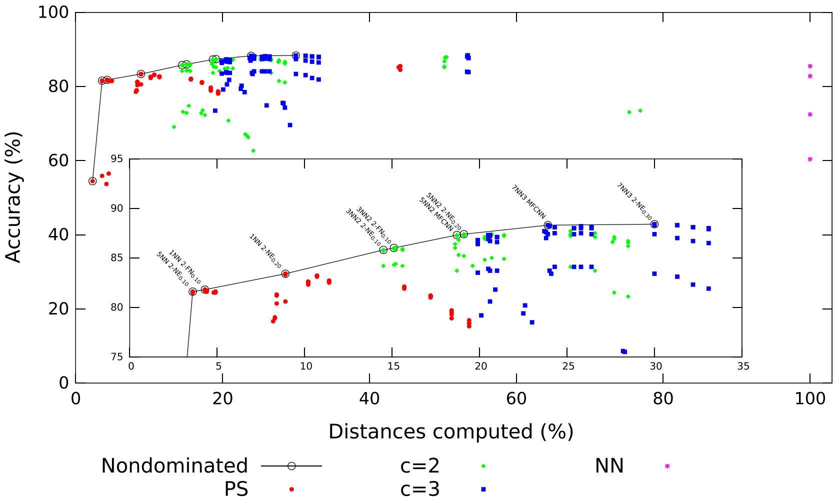


Fig. 4. Distances computed (%) against accuracy achieved by the algorithms. Average results when 40% of noise is added to the samples. The non-dominated front is remarked.

last figures clearly contrast with the 79.5% increase in the distances required so as to be able to improve the 93.9% bound figure to the maximum (barely, 6%). Also, as in the scenario without synthetic noise, the accuracy follows the same non-linear trend of the upper limits but with the remarkable influence of the noise, which might seriously affect the performance: for instance, in the same case of the 2-FN$_{0.10}$ algorithm with $k=1$ and a noise figure of 40%, there is an accuracy improvement from the initial 81.8–84.3% when going from the initial PS to the $c=2$ scheme by computing close to 11% more of distances; however, the use of $c=3$ takes an additional cost of 5% in the distances but, instead of improving results, there is an actual accuracy decrease of almost 0.5%. This clearly shows that the non-linear behavior is not only dependent on the PS algorithm but also on the noise the system is dealing with.

On the other hand, Figs. 3 and 4 show the results of the strategies considered facing accuracy and distances computed. Note that the optimal strategies (non-dominated solutions) are remarked. As occurred in the scenario without noise, PS and kNNc schemes are present within this optimal set. Following the MOP scenario no order can be established within the non-dominated solutions. Nevertheless, it can be checked that now the kNNc strategies are the most numerous.

As it happened in the previous situation in which no noise was added, the basic kNN algorithm is again out of the optimal set of solutions as now kNN3 is not only able to reach its performance with a lower number of distances computed, but it also does obtain a better classification accuracy. Specially interesting are the cases of 5NN3 MFCNN and 5NN3 MCNN, for 20% of noise, and 7NN3 MFCNN and 7NN3 2-NE$_{0.30}$, for 40% of noise, which achieve better performance with just around 25–30% of distances computed.

5.3. Statistical significance test

The aim of this section is to assess whether the inclusion of the second step of the kNNc scheme leads to significantly better classification accuracies. We shall therefore use the KEEL [31] software, which contains statistical tools that allow us to quantify the difference between the results with and without this step. Specifically, a Wilcoxon 1 × 1 test was performed between PS and each kNN2 configuration for the same algorithm as well as between kNN2 and kNN3. The first one checks whether there is a significant accuracy upturn between the kNN2 approach and the basic PS scheme, which is the main contribution of this paper. The second one is performed to assess whether the accuracy in kNN3 is significantly better than the one obtained in kNN2, which may justify providing more class proposals.

Table 4
Asymptotic p-value obtained in Wilcoxon 1 vs. 1 statistical significance test. First column assumes that accuracy of kNN2 is better than accuracy of PS. Second column assumes that accuracy of kNN3 is better than accuracy of kNN2. Bold values represent a level of significance higher than $\alpha=0.95$.

Noise (%)	k	kNN2 vs PS	kNN3 vs kNN2
0	1	**0.000032**	**0.000035**
	3	**0.000044**	**0.000038**
	5	**0.000051**	**0.000038**
	7	**0.000044**	**0.000044**
20	1	**0.00007**	0.117066
	3	**0.000035**	**0.00001**
	5	**0.000038**	**0.000047**
	7	**0.000048**	
40	1	**0.000065**	0.86278
	3	**0.000048**	**0.005203**
	5	**0.000044**	**0.000032**
	7	**0.000021**	**0.00001**

The significant (asymptotic) p-values considering all the experiments are shown in Table 4. These values represent the overlap between the two distributions, assuming that kNNc accuracy is better. We can consider the p-values as a confidence measure for the comparison. The significance of a low value is a high probability that the distributions compared are different.

As is shown in the first column, all the values are lower than 0.05, depicting that the inclusion of our second step leads to a significant accuracy improvement at a confidence level of 95%. Moreover, the second column shows that, except for the two particular configurations of $k=1$ with synthetic noise rates of 20% and 40%, proposing an additional label does lead to higher accuracy as the rest of the confidence values are also lower than 0.05.

6. Conclusions

k-Nearest Neighbor (kNN) classification is one of the most common, easy and simple algorithms for supervised learning which usually achieves an acceptable performance. Within this context, Prototype Selection (PS) algorithms have demonstrated their utility by improving some kNN issues such as computational time, noise removal or memory usage. Nevertheless, PS often leads to a decrease of the classification accuracy. To this end, we propose a two-step strategy in which the PS algorithm is exploited by using its reduced set to select the *c* nearest classes for a given input. Afterwards, only these *c* classes are taken into account in the classification stage with the original set. Therefore, some misclassification produced by using the reduced set can be corrected with neither increasing the computation too much nor requiring the whole training set to be stored in the memory at the same time.

Experimentation in which our strategy was faced against conventional PS-based classification was conducted. A representative set of PS algorithms was chosen and several metrics of interest were collected in classification experiments with some multi-label datasets.

Results showed that our proposal provides a new range of solutions in the trade-off between accuracy and efficiency. In the best cases, our strategy equals the accuracy of kNN classification with just 30% of distances computed. In addition, in the presence of noisy data, our search achieves a remarkably profitable performance since, in combination with the appropriate PS algorithm, it improves the kNN classification with a higher efficiency. Furthermore, in all cases considered, statistical tests revealed that kNNc accuracy is significantly better than the one obtained with just PS.

Some interesting conclusions were also drawn with respect to the tuning parameter *c*. The profitability of increasing *c* did show a non-linear tendency with respect to both the maximum achievable classification rate and the actual accuracy obtained. The improvement gain decreases as the number of recommendations *c* gets higher, depicting an asymptotic behavior. Therefore, an optimal *c* value may be found on the trade-off between accuracy and efficiency depending on the conditions of the considered scenario.

This work has opened some promising future work lines when computing a hypothesis in the second step. Results showed that there is a great gap between the upper bound of the classification (rate in which the correct label is within the *c* classes proposal) and the empirical classification rate. Therefore, other kind of search could be performed in this second step instead of resorting again to the kNN classification.

Conflict of interest

None declared.

Acknowledgements

This work was partially supported by the Spanish Ministerio de Educación, Cultura y Deporte through FPU Fellowship (AP2012–0939), the Spanish Ministerio de Economía y Competitividad through Project TIMuL (TIN2013-48152-C2-1-R), Consejería de Educación de la Comunidad Valenciana through Project PROMETEO/2012/017 and Vicerrectorado de Investigación, Desarrollo e Innovación de la Universidad de Alicante through FPU Program (UAFPU2014–5883).

References

[1] E. Fix, J.L. Hodges, Discriminatory Analysis, Nonparametric Discrimination: Consistency Properties, US Air Force School of Aviation Medicine Technical Report 4 No. 3, 1951, p. 477.

[2] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, 2nd ed., John Wiley & Sons, New York, NY, 2001.

[3] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE Trans. Inf. Theory 13 (1) (1967) 21–27. http://dx.doi.org/10.1109/TIT.1967.1053964.

[4] T.M. Mitchell, Machine Learning, 1st edition, McGraw-Hill, Inc., New York, NY, USA, 1997.

[5] S. Garcia, J. Derrac, J. Cano, F. Herrera, Prototype selection for nearest neighbor classification: taxonomy and empirical study, IEEE Trans. Pattern Anal. Mach. Intell. 34 (3) (2012) 417–435. http://dx.doi.org/10.1109/TPAMI.2011.142.

[6] J. Derrac, N. Verbiest, S. García, C. Cornelis, F. Herrera, On the use of evolutionary feature selection for improving fuzzy rough set based prototype selection, Soft Comput. 17 (2) (2013) 223–238.

[7] S. García, J. Luengo, F. Herrera, Data Preprocessing in Data Mining, of Intelligent Systems Reference Library, vol. 72, Springer, Cham, Switzerland (2015) http://dx.doi.org/10.1007/978-3-319-10247-4.

[8] L. Nanni, A. Lumini, Prototype reduction techniques: a comparison among different approaches, Expert Syst. Appl. 38 (9) (2011) 11820–11828. http://dx.doi.org/10.1016/j.eswa.2011.03.070.

[9] I. Triguero, J. Derrac, S. García, F. Herrera, A taxonomy and experimental study on prototype generation for nearest neighbor classification, IEEE Trans. Syst. Man Cybern. Part C: Appl. Rev. 42 (1) (2012) 86–100. http://dx.doi.org/10.1109/TSMCC.2010.2103939.

[10] N. García-Pedrajas, A. De Haro-García, Boosting instance selection algorithms, Knowl. Based Syst. 67 (2014) 342–360. http://dx.doi.org/10.1016/j.knosys.2014.04.021.

[11] C.-F. Tsai, W. Eberle, C.-Y. Chu, Genetic algorithms in feature and instance selection, Knowl. Based Syst. 39 (0) (2013) 240–247. http://dx.doi.org/10.1016/j.knosys.2012.11.005.

[12] J. Derrac, C. Cornelis, S. García, F. Herrera, Enhancing evolutionary instance selection algorithms by means of fuzzy rough set based feature selection, Inf. Sci. 186 (1) (2012) 73–92. http://dx.doi.org/10.1016/j.ins.2011.09.027.

[13] J.R. Cano, F. Herrera, M. Lozano, Stratification for scaling up evolutionary prototype selection, Pattern Recognit. Lett. 26 (7) (2005) 953–963. http://dx.doi.org/10.1016/j.patrec.2004.09.043.

[14] F. Angiulli, G. Folino, Distributed Nearest Neighbor-Based Condensation of Very Large Data Sets, IEEE Trans. Knowl. Data Eng. 19 (12) (2007) 1593–1606. http://dx.doi.org/10.1109/TKDE.2007.190665.

[15] P. Hart, The condensed nearest neighbor rule (corresp.), IEEE Trans. Inf. Theory 14 (3) (1968) 515–516. http://dx.doi.org/10.1109/TIT.1968.1054155.

[16] D.L. Wilson, Asymptotic properties of nearest neighbor rules using edited data, IEEE Trans. Syst. Man Cybern. 2 (3) (1972) 408–421. http://dx.doi.org/10.1109/TSMC.1972.4309137.

[17] P.A. Devijver, J. Kittler, Pattern Recognition: A Statistical Approach, Prentice Hall, London, United Kingdom, 1982.

[18] B.V. Dasarathy, J.S. Sánchez, S. Townsend, Nearest neighbour editing and condensing tools-synergy exploitation, Pattern Anal. Appl. (2000) 19–30.

[19] F. Angiulli, Fast nearest neighbor condensation for large data sets classification, IEEE Trans. Knowl. Data Eng. 19 (11) (2007) 1450–1464.

[20] J.R. Rico-Juan, J.M. Iñesta, New rank methods for reducing the size of the training set using the nearest neighbor rule, Pattern Recognit. Lett. 33 (5) (2012) 654–660.

[21] D.R. Wilson, T.R. Martinez, Instance pruning techniques, in: Proceedings of the Fourteenth International Conference on Machine Learning, ICML '97, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997, pp. 403–411.

[22] H. Brighton, C. Mellish, On the consistency of information filters for lazy learning algorithms, in: J. Zytkow, J. Rauch (Eds.), Principles of Data Mining and Knowledge Discovery, of Lecture Notes in Computer Science, vol. 1704, Springer, Berlin, Heidelberg, 1999, pp. 283–288. http://dx.doi.org/10.1007/978-3-540-48247-5_31.

[23] L.J. Eshelman, The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination, in: Proceedings of the First Workshop on Foundations of Genetic Algorithms. Bloomington Campus, IN, USA, July 15–18, 1990, pp. 265–283.

[24] J.R. Cano, F. Herrera, M. Lozano, On the combination of evolutionary algorithms and stratified strategies for training set selection in data mining, Appl. Soft Comput. 6 (3) (2006) 323–332. http://dx.doi.org/10.1016/j.asoc.2005.02.006.

[25] J. Hull, A database for handwritten text recognition research, IEEE Trans. Pattern Anal. Mach. Intell. 16 (5) (1994) 550–554. http://dx.doi.org/10.1109/34.291440.

[26] H. Freeman, On the encoding of arbitrary geometric configurations, IRE Trans. Electron. Comput. 10 (2) (1961) 260–268. http://dx.doi.org/10.1109/TEC.1961.5219197.

[27] R.A. Wagner, M.J. Fischer, The string-to-string correction problem, J. ACM 21 (1) (1974) 168–173. http://dx.doi.org/10.1145/321796.321811.

[28] J. Calvo-Zaragoza, J. Oncina, Recognition of pen-based music notation: the HOMUS dataset, in: Proceedings of the 22nd International Conference on Pattern Recognition, Stockholm, Sweden, 2014, pp. 3038–3043.

[29] H. Sakoe, S. Chiba, Dynamic programming algorithm optimization for spoken word recognition, in: A. Waibel, K.-F. Lee (Eds.), Readings in Speech Recognition, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990, pp. 159–165.

[30] N. Natarajan, I. Dhillon, P. Ravikumar, A. Tewari, Learning with noisy labels, in: Advances in Neural Information Processing Systems, 2013, pp. 1196–1204.

[31] J. Alcalá-Fdez, L. Sánchez, S. García, M.J.D. Jesus, S. Ventura, J.M. Garrell, J. Otero, J. Bacardit, V.M. Rivas, J.C. Fernández, F. Herrera, KEEL: a software tool to assess evolutionary algorithms to data mining problems, Soft Comput. 13 (3) (2009) 307–318.

Chapter 9

Prototype Generation on Structural Data using Dissimilarity Space Representation

Calvo-Zaragoza, J., Valero-Mas, J. J., and Rico-Juan, J. R. (2016b). Prototype Generation on Structural Data using Dissimilarity Space Representation. *Neural Computing and Applications*, Online:1–10

Selected paper of the 7th Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA 2015)

IBPRIA 2015

Prototype generation on structural data using dissimilarity space representation

Jorge Calvo-Zaragoza[1] · Jose J. Valero-Mas[1] · Juan R. Rico-Juan[1]

Received: 29 September 2015 / Accepted: 15 March 2016

Abstract Data reduction techniques play a key role in instance-based classification to lower the amount of data to be processed. Among the different existing approaches, prototype selection (PS) and prototype generation (PG) are the most representative ones. These two families differ in the way the reduced set is obtained from the initial one: While the former aims at selecting the most representative elements from the set, the latter creates new data out of it. Although PG is considered to delimit more efficiently decision boundaries, the operations required are not so well defined in scenarios involving structural data such as strings, trees, or graphs. This work studies the possibility of using dissimilarity space (DS) methods as an intermediate process for mapping the initial structural representation to a statistical one, thereby allowing the use of PG methods. A comparative experiment over string data is carried out in which our proposal is faced to PS methods on the original space. Results show that the proposed strategy is able to achieve significantly similar results to PS in the initial space, thus standing as a clear alternative to the classic approach, with some additional advantages derived from the DS representation.

Keywords kNN classification · Prototype generation · Structural pattern recognition · Dissimilarity space

1 Introduction

In the pattern recognition (PR) field, two fundamental approaches can be found depending on the model used for representing the data [12]: a first one usually known as structural or syntactical, in which data are represented as symbolic data structures such as strings, trees, or graphs, and a second one known as statistical or feature representation, in which the representation is based on numerical feature vectors that are expected to sufficiently describe the actual input.

The election of one of these approaches has some noticeable implications and consequences: Structural methods offer a wide range of powerful and flexible high-level representations, but only few PR algorithms and techniques are capable of processing them; statistical methods, in spite of being less flexible in terms of representation, depict a larger collection of PR techniques [5].

Independently of whether we use a structural or a feature representation, instance-based PR methods, for which the *k*-nearest neighbor rule (kNN) is the most representative, may be applied for classification tasks. Generally, these methods just require to work over a metric space, i.e., that in which a distance between two points can be defined. Instead of obtaining a set of classification rules out of the available information, they need to examine all the training data each time a new element has to be classified. As a consequence, they not only depict considerable memory requirements in order to store all these data, which in some cases might be a very large number of elements, but also show a low computational efficiency as all training information must be checked at each classification task [28].

Data reduction techniques, a particular subfamily of data preprocessing methods, try to solve these limitations by means of selecting a representative subset of the training

✉ Jorge Calvo-Zaragoza
jcalvo@dlsi.ua.es

[1] Department of Software and Computing Systems, University of Alicante, Carretera San Vicente del Raspeig s/n, 03690 Alicante, Spain

Published online: 26 March 2016

data [19]. Two common approaches for performing this task are prototype generation (PG) and prototype selection (PS). Both families of methods focus on reducing the size of the training set for lowering the computational requirements while maintaining, as far as possible, the classification accuracy. The former family creates new artificial data to replace the initial set while the latter one simply selects certain elements from that set.

It must be pointed out that the two aforementioned DR paradigms do not show the same dependency on the data representation used. PS algorithms have been widely used in both structural and feature representations as the elements are not transformed but simply selected. On the other hand, PG methods require to modify or create data in order to intelligently place new elements and, while this process can be easily performed in feature representations, it becomes remarkably difficult for structured data, at least in terms of developing a generic strategy for any type of data structure (e.g., strings, trees, or graphs).

In this paper, we study the possibility of applying PG methods to structured representations by means of using dissimilarity space (DS) methods so as to solve the aforementioned obstacle. By using DS techniques, the initial structural representation can be mapped onto a feature-based one, thereby allowing the use of statistical PG techniques not available in the original space. Our intention is to assess whether this approach deserves further consideration when faced against the classical choice of applying PS in the initial structural space.

This paper expands the initial idea proposed in the work of Calvo-Zaragoza et al. [8] by providing a more far-reaching experimentation, in which a broader number of DS methods is considered. Stronger statements about the performance of the proposal are drawn, supported by a comprehensive evaluation in terms of number of datasets and statistical significance tests.

The rest of the paper is structured as it follows: Sect. 2 introduces the task of data reduction; Sect. 3 explains the idea of dissimilarity space and its application to our case; Sect. 4 describes the evaluation methodology proposed; Sect. 5 shows and thoroughly analyzes the results obtained; finally, Sect. 6 explains the general conclusions obtained and discusses possible future work.

2 Background on data reduction

Among the different stages which comprise the so-called knowledge discovery in databases, data preprocessing is the set of tasks devoted to provide the information to the data mining system in the suitable amount, structure, and format [25]. Data reduction (DR), which constitutes one of these possible tasks, aims at obtaining a reduced set with respect to the original data which, if provided to the system, would produce the same output as the original data [19].

DR techniques are widely used in kNN classification as a means of overcoming its previously commented drawbacks, being the two most common approaches prototype generation (PG) and prototype selection (PS) [29]. Both methods focus on obtaining a smaller training set for lowering the computational requirements and removing ambiguous instances while keeping, if not increasing, the classification accuracy.

PS methods try to select the most profitable subset of the original training set. The idea is to reduce its size to lower the computational cost and remove noisy instances which might confuse the classifier. Typically, three main families can be considered based on the objective pursued during the process:

- *Condensing* The idea followed by these methods is to keep only the most representative prototypes of each class and reduce as much as possible the dataset. While accuracy on training set is usually maintained, generalization accuracy is lowered.
- *Editing* These approaches focus on eliminating instances which produce some class overlapping, typical situation of elements located close to the decision boundaries or noisy data. Data reduction rate is lower than in the previous case, but generalization accuracy tends to be higher.
- *Hybrid* These algorithms look for a compromise between the two previous approaches, which means seeking the smallest dataset while improving, or at least maintaining, the generalization accuracy of the former set.

Given its importance, many different approaches have been proposed throughout the years to carry out this task. The reader may check the work of Garcia et al. [18] for an extensive introduction to this topic as well as a comprehensive experimental comparison for the different methods proposed. Since trying to maintain the same accuracy as with the initial training set is difficult to fulfill in practical scenarios, much research has been recently devoted to enhance this process through the combination with other techniques. Some of these include feature selection [35], ensemble methods [20], or modifications to the kNN rule [7].

On the other hand, PG methods are devoted to creating a new set of labeled prototypes that replace the initial training set. Under the DR paradigm, this new set is expected to be smaller than the original one since the decision boundaries can be defined more efficiently. Depending on the focus where placing the new prototypes, three main families of strategies can be found:

- *Centroid-based* subsets of prototypes of the initial training set are grouped taking into account proximity, labeling, and representation criteria. Then, the centroid of this subset is generated as a new prototype for the final set.
- *Position adjustment* from an initial subset of the training set, selected following any strategy (for instance, a PS method), prototypes are moved around their neighborhoods following a particular heuristic. The objective was to find the location in which they can be more profitable for classification purposes.
- *Space partitioning* the idea is to divide the input space into regions of interest. Then, representatives of each space are generated. Variations in space division and generation within each one provide the different methods of this family.

Reader is referred to the work of Triguero et al. [34] to find a further extension to this introduction to PG methods.

Under the same conditions, PG is expected to perform better than PS since the former can be seen as a generalization of the latter. Nevertheless, while PS only needs information about similarity or proximity between different prototypes, for which one can use the same dissimilarity function considered for the kNN rule, PG needs information about the representation space. Indeed, the PG family represents a more restrictive option than the simple selection of prototypes because it is hard to be used under structural spaces. In these cases, it is difficult to develop generic operations such as "move a prototype toward a specific direction" or "find the centroid of a subset of prototypes." Thus, generating new prototypes in structural data is not a trivial matter.

Given the theoretical advantages of PG over PS methods, finding strategies to generate prototypes on structural data would be of great interest. In this work, it is proposed a method that fills this gap. It consists of a two-stage algorithm which first maps the structural data onto features vectors, after which common PG techniques can easily work. To perform this mapping, we resort here to the so-called dissimilarity space representation. Next section details our proposal.

3 Prototype generation over structural data using dissimilarity space representation

Current PG algorithms assume that data are defined over a vector space. Thus, it is feasible to perform geometric operations to find new points of interest in which new labeled prototypes can be generated. The intention is to maintain the accuracy of the kNN classifier with fewer prototypes than in the original training set.

Nevertheless, when working over a structural space, it is just known a distance function that allows knowing the proximity between two points of the space (this is also referred as metric space). In that case, PG algorithms are not able to generalize the geometric operations utilized in the vector space. Serve as an example the median operation: Its computation is easy for n-dimensional points, whereas it becomes NP-complete when points are strings [22]. Some examples of works addressing related issues include the work of Abreu and Rico-Juan [1], in which the median of a string set is approximated using edit operations, or Ferrer and Bunke [16], in which an iterative algorithm for the computation of the median operation on graphs is exposed. Nevertheless, all of them take advantage of the knowledge of the specific structural data to create these new prototypes. Therefore, generalization to other structural representations cannot be assumed.

We propose a new strategy as a possible solution to the problem stated above. The process itself follows a simple procedure which consists in mapping data onto a new vector, or feature, space. This process, known as *embedding*, has been extensively studied for decades [4, 23]. Once data are represented as feature vectors, conventional prototype generation strategies may be used.

In this work, we are going to restrict ourselves to a particular family of embedding algorithms known as dissimilarity space (DS) representation [13]. Broadly, DS representations are obtained by computing pairwise dissimilarities between the elements of a representation set, which actually constitutes a subset of the initial structural training data selected following a given criterion.

The choice of using DS instead of other techniques is justified by some reasons directly related to the actual object of study:

1. It only requires a distance or dissimilarity function between prototypes. Taking into account that this work focuses on DR techniques for improving kNN classification—which also needs this function—the requirement is assumed to be effortless.
2. The intention of the work is to measure the performance of PG on the new space. Therefore, it is preferable that results are more related to the PG technique instead of the quality of the embedding method. That is why it is considered a simple method (but with a strong background) rather than a more complex one.

During experimentation, the classification results obtained after applying a set of PG techniques to the DS representation will be compared to the results obtained when using PS techniques in the initial structural space so as to check whether our approach can be useful in these situations. On the other hand, below we introduce the DS transformation and the particular strategies considered.

3.1 Dissimilarity space transformation

Let $\mathcal{X}$ denote a structural space in which a dissimilarity function $d : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}^+$ is defined. Let Y represent the set of labels or classes of our classification task. Let T be a labeled set of prototypes such that $T = \{(x_i, y_i) : x_i \in \mathcal{X}, y_i \in Y\}_{i=1}^{|T|}$.

In order to map the prototypes of T onto a feature space $\mathcal{F}$, DS-based methods seek for a subset R out of the training set ($R \subseteq T$). The elements of R, usually known as *pivots*, are noted as r_i with $1 \leq i \leq |R|$. Then, a prototype $x \in \mathcal{X}$ can be represented in $\mathcal{F}$ as a set of features $(v_1, v_2, v_3, \ldots, v_{|R|})$ such that $v_i = d(x, r_i)$. This way, an $|R|$-dimensional real-valued vector can be obtained for each point in the space $\mathcal{X}$. Different heuristics were proposed in the work of Pekalska et al. [30] for the selection of pivots, some of which have been considered for our work and are briefly described below.

3.1.1 RandomC

The RandomC strategy selects a random subset of prototypes, in which the number of prototypes of each class is exactly c (tuning parameter), that is, $|R| = c|Y|$. In order to compare the influence of parameter c in the feature representation, some different values will be considered at experimentation stage.

3.1.2 kCenters

This strategy performs a k-medoids clustering process on every class considered. The initialization is performed as proposed in the work of Arthur and Vassilvitskii [3] (k-means++). The different centroids obtained after the process are included in R, i.e., $|R| = k|Y|$. As happened in the previous case, the value k may alter the representation of the new space so some tuning will be considered during the experimentation.

3.2 EditCon

The main idea behind EditCon is to select the most representative prototypes of the training set to be used as pivots. To this end, this technique applies two PS algorithms to the initial training set: As a first step, an editing process [37] is used to remove noisy information; then, a condensing process [21] is performed so as to keep only the informative elements. No parameters are considered in this case.

4 Experimentation

Figure 1 shows the implemented setup for performing the experimentation. As it can be checked, out of the initial structural elements, a feature representation is obtained using a particular DS method. DR techniques are then applied to both data representations but, while PS methods are applied to structural and feature representations, PG is only performed on the latter. Finally, the nearest neighbor (NN) algorithm, parameterized with $k = 1$, is used for the classification.

For these experiments, different configurations of the c and k parameters of the RandomC and kCenters, respectively, have been tested. The values considered have been 5, 10, and 15 prototypes per class.

We shall now describe the different datasets, data reduction strategies studied, and the performance metrics considered for this study.

4.1 Datasets

Five different datasets of isolated symbols have been considered: the National Institute of Standards and Technology DATABASE 3 (NIST3), from which a subset of the upper case characters was randomly selected, the Mixed National Institute of Standards and Technology dataset (MNIST) [27] of handwritten digits, the United States Postal Office handwritten digits dataset (USPS) [24], the MPEG-7 shape silhouette dataset [26], and the Handwritten

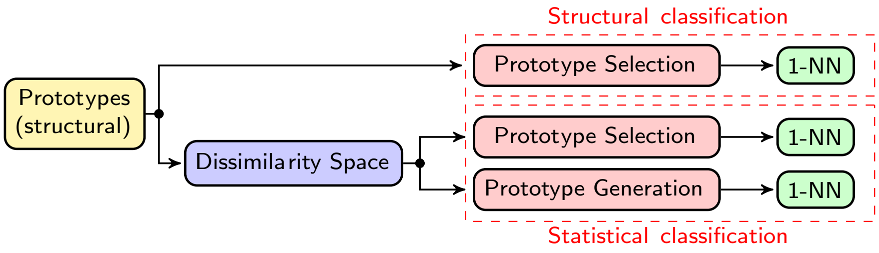


Fig. 1 Experimental setup tested. DS is used for mapping structural data into a feature-based space. PS is applied to both structural and feature data while PG is only performed on the latter. 1-NN is used for the classification in all cases

Table 1 Description of the datasets used in the experimentation

Name	Instances	Classes
NIST3	6500	26
MNIST	70,000	10
USPS	9298	10
MPEG-7	1400	70
HOMUS	15,200	32

Online Musical Symbol (HOMUS) dataset [6]. In terms of class representation, these datasets can be considered as being totally balanced. Freeman Chain Codes [17] have been considered as contour descriptors. Since these structural data are represented with strings, the well-known edit distance [36] is considered as dissimilarity. Once data are mapped onto feature vectors, the Euclidean distance is used.

A fivefold cross-validation process has been applied for each dataset to examine the variance to the training data.

Reader is referred to Table 1 to find more details about the composition of the datasets.

4.2 Data reduction strategies

A representative set of DR algorithms covering a wide range of selection variants was used for the experimentation. However, in order to perform a fair comparison between the two DR strategies, we are only showing the results for the PS algorithms retrieving similar size reductions to the PG algorithms. These techniques are briefly introduced in the following lines.

4.2.1 Prototype selection (PS) algorithms

- Fast condensing nearest neighbor (FCNN) [2] computes a fast, order-independent condensing strategy based on seeking the centroids of each label.
- Farther neighbor (FN) [31] gives a probability mass value to each prototype following a voting heuristic based on neighborhood. Prototypes are selected according to a parameter (fixed to 0.3 in our case) that indicates the probability mass desired for each class in the reduced set.
- Cross-generational elitist selection, heterogeneous recombination, and cataclysmic mutation algorithm (CHC) [14]: evolutionary algorithm commonly used as a representative of genetic algorithms in PS. The configuration of this algorithm has been the same as in [9].

This subset of techniques is expected to cover three typical searching methodologies of PS: FCNN as condensing, FN as heuristic approach, and CHC as evolutionary search.

4.2.2 Prototype generation (PG) algorithms

- Reduction by space partitioning 3 (RSP3) [32] divides the space until a number of class-homogeneous subsets are obtained; a prototype is then generated from the centroid of each subset.
- Evolutionary nearest prototype classifier (ENPC) [15] performs an evolutionary search using a set of prototypes that can improve their local quality by means of genetic operators.
- Mean squared error (MSE) [10] generates new prototypes using gradient descent and simulated annealing. Mean squared error is used as cost function.

The parameters of these algorithms have been established following the work of Triguero et al. [34]. As in the previous case, we try to consider a representative set of generation techniques: MSE as a classical method, ENPC as evolutionary search, and RSP3 as heuristic approach.

4.3 Performance measurement

In order to assess the results, we have considered as metrics of interest the classification accuracy of the reduced set as well as its size. While the former indicates the ability of the DR method to choose the most relevant prototypes, the latter one depicts its reduction skills.

For these figures of merit, we show the results obtained when averaging the scores for each dataset, which allows to understand the general performance of each scenario at a glance. Nevertheless, in order to perform a rigorous comparison among the strategies, a significance test has been performed facing accuracy and set size figures.

It must be considered that, although these measures are suitable to evaluate the performance of each single strategy, it is not possible to establish a clear comparison among the whole set of alternatives to determine the best one. DR algorithms aim at minimizing the number of prototypes considered in the training set while, at the same time, increasing the classification accuracy. Most often, these two goals are contradictory so improving one of them implies a deterioration of the other. From this point of view, classification in DR scenarios can be seen as a multi-objective optimization problem (MOP) in which two functions have to be simultaneously optimized: reduction in the training set and maximization of the classification success rate. Usually, the evaluation of this kind of problems is carried out in terms of the *non-dominance* concept. One solution is said to dominate another if, and only if, it is better or equal in each goal function and, at least, strictly better in one of them. The set of non-dominated elements represents the different optimal solutions to the MOP. Each of them is usually referred to as Pareto optimal solution, being the whole set usually known as Pareto frontier.

Finally, classification time is also considered in this study to assess the influence of the type of data representation in these terms.

5 Results

Average results in terms of classification accuracy and set size obtained on the different datasets are presented in Table 2. Additionally, Table 3 shows the corresponding average classification times. Normalization (in %) is done with respect to the whole dataset. ALL refers to results obtained when using the whole training set (no DR algorithm is applied). Furthermore, Table 4 shows the average number of attributes obtained in each dataset when applying the different DS processes to the initial structural space.

For a better understanding, Fig. 2 shows graphically the results in a 2D representation where accuracy and size are confronted. Non-dominant elements representing the Pareto frontier are highlighted.

A first initial remark is that, on average, the DS process implies a reduction in classification accuracy. For a given algorithm, when comparing the accuracy results obtained in the initial space with any of the corresponding DS cases, there is a decrease in these figures. For instance, when considering the ALL case, average classification accuracy goes from 90.8 % in the initial space to figures around 88 % in the different DS spaces considered, which is around a 3 % decrease in accuracy simply because of the mapping stage.

For both structural and feature-based representations, PS techniques depict a decrease in the classification accuracy when compared to the ALL case. This effect is a consequence of the reduction in the set size. In the DS space, however, PG achieves slightly better classification results

Table 2 Results obtained with the different DS algorithms configurations considered

	ALL		PS						PG					
			FCNN		1–FN$_{0.3}$		CHC		RSP3		ENPC		MSE	
	Acc	Size	Acc	Size	Acc	Size	Acc	Size	Acc	Size	Acc	Size	Acc	Size
No DS	90.8	100	87.9	22.0	84.6	13.9	81.0	3.6	–	–	–	–	–	–
RandomC(5)	87.4	100	84.1	26.8	80.7	16.5	75.5	3.6	85.8	31.5	85.6	15.0	83.3	14.3
RandomC(10)	88.0	100	84.5	26.2	81.3	16.5	75.8	3.3	86.2	31.0	86.0	14.4	85.5	14.4
RandomC(15)	88.2	100	84.9	26.0	81.6	16.5	76.6	3.3	86.7	31.3	86.1	14.3	84.1	14.4
kCenters(5)	87.9	100	84.1	26.4	81.2	16.5	76.6	3.4	86.2	30.2	85.9	14.8	83.8	14.3
kCenters(10)	88.0	100	84.5	26.1	81.1	16.5	76.7	3.6	86.6	30.9	86.2	14.4	84.2	14.4
kCenters(15)	88.3	100	85.1	25.7	81.4	16.5	77.0	3.5	86.7	30.7	86.3	14.1	84.2	14.4
EditCon	88.0	100	84.9	25.9	81.5	16.6	75.8	3.7	86.5	32.6	86.2	14.5	83.7	14.4

Figures shown represent the average of the results obtained for each single dataset. No DS depicts results obtained in the initial structural space. Selection and generation techniques are regarded as PS and PG, respectively. ALL stands for the case in which no selection or generation is performed. Normalization (%) is performed with respect to ALL case of each dataset separately

Table 3 Average classification time (in seconds) for the different DS algorithms configurations considered

	ALL	PS			PG		
		FCNN	1–FN$_{0.3}$	CHC	RSP3	ENPC	MSE
No DS	877.3	221.3	136.7	50.7	–	–	–
RandomC(5)	3.15	0.91	0.56	0.13	1.09	0.46	0.13
RandomC(10)	5.07	1.5	0.96	0.28	1.71	0.72	0.14
RandomC(15)	6.72	2.03	1.36	0.47	2.21	0.92	0.18
kCenters(5)	3.17	0.90	0.56	0.13	1.06	0.47	0.08
kCenters(10)	5.05	1.48	0.96	0.29	1.68	0.71	0.14
kCenters(15)	6.73	2.01	1.36	0.48	2.18	0.9	0.18
EditCon	20.75	7.2	5.39	2.92	6.53	2.48	0.39

Figures shown represent the obtained when processing each single dataset. No DS depicts results obtained in the initial structural space. Selection and generation techniques are regarded as PS and PG, respectively. ALL stands for the case in which no selection or generation is performed

Table 4 Number of features in the dissimilarity space for each DS algorithm and dataset

	RandomC(5)	RandomC(10)	RandomC(15)	kCenters(5)	kCenters(10)	kCenters(15)	EditCon
NIST3	130	260	390	130	260	390	520
MNIST	50	100	150	50	100	150	650
USPS	50	100	150	50	100	150	680
MPEG-7	350	700	1050	350	700	1050	210
HOMUS	160	320	480	160	320	480	1760
Average	148	296	444	148	296	444	764

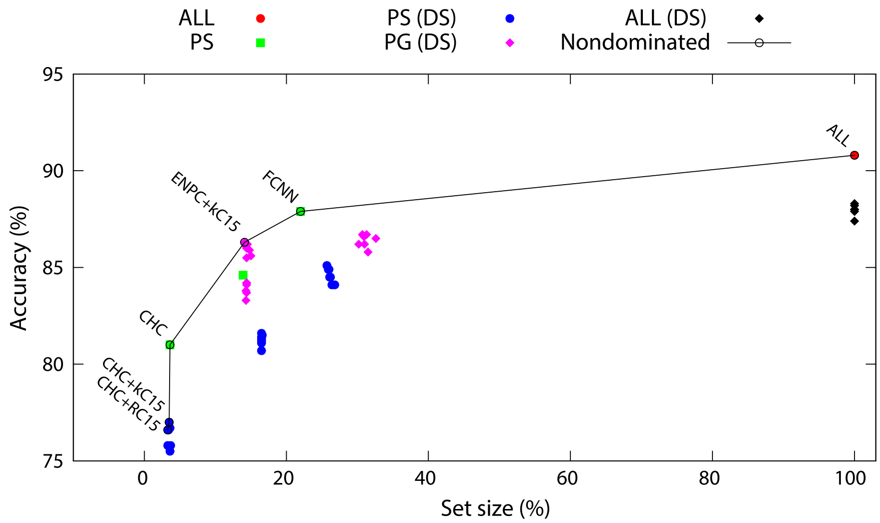


Fig. 2 Average results of the different configurations considered, facing accuracy and size of the reduced set. Non-dominated elements defining the Pareto frontier are *highlighted*

with similar reduction rates than the PS algorithms, somehow showing the superior robustness of these methods. As an example, for RandomC(5), ENPC achieves an accuracy of 85.6 % with a set size of 15 %, whereas 1-FN$_{0.3}$ roughly gets to a classification rate of 80.7 % with a 16.5 % of the initial set.

Nevertheless, the main outcome out of the results obtained by the PG algorithms is that the scores obtained in the DS space are quite similar to the ones obtained by PS schemes in the initial structural space. Although this point shall be later thoroughly assessed through statistical tests, these figures may allow us to qualitatively see the proposed strategy as a clear competitor of PS in structural data.

In terms of classification times, results show DS strategies as much faster than structural ones (several orders of magnitude) due to the complexity reduction achieved by using Euclidean distance instead of Edit distance.

Regarding the considered DS strategies, it can be checked that the results are not remarkably affected by the DS algorithm considered as neither accuracy values nor sizes show dramatic changes among them. In the same sense, parameters of RandomC and kCenters do not seem to have a remarkable influence either as figures obtained by the different configurations are very similar.

When considering the non-dominance criterion, we can see that most elements defining the Pareto frontier are PS configurations in the structural space, more precisely CHC, FCNN, and the ALL configuration (see Fig. 2). When mapping to the statistical space, CHC extends the frontier as, despite its accuracy loss, it achieves remarkable reduction rates. Concerning our proposal of PG in the DS space, we can see that the different configurations fill some areas of the space where the rest of the considered approaches do not have a relevant presence. It is also interesting to point out the presence of ENPC as part of the non-dominant elements set, thus remarking the interest of the strategy proposed in the paper.

Finally, some remarks can be done attending to the information in Table 4 regarding the number of attributes in the feature space for the datasets considered together with the general performance information in Table 2. As it can be seen, the election of a particular DS method implies a great difference in the number of attributes. For instance, RandomC(5) supposes one-third of the number of attributes in RandomC(15) and around one-seventh of the ones retrieved by the EditCon algorithm. Nevertheless, accuracy results (cf. Table 2) do not report a clear difference in the results. As an example, in the ALL situation, kCenters(5) and EditCon report a very similar average accuracy (around 80 %) but with a great difference in terms of number of attributes.

5.1 Statistical significance

As aforementioned, in order to statistically estimate the competitiveness of the proposed strategy, a Wilcoxon rank-sum test [11] has been performed. As we aim at assessing the competitiveness of using PG in DS spaces against PS in the initial space, accuracy and set size figures shall be compared. Table 5 shows the results of this test when considering a significance $p < 0.05$.

We note that PG strategies are not competitive in accuracy against the ALL case in the structural space as they achieve significantly lower classification rates. In terms of reduction, as expected, all PG strategies significantly outperform the ALL case, as the latter does not perform any kind of reduction.

When compared to the PS algorithms in the structural space, it can be checked that RSP3 does not achieve a remarkable reduction rate as set sizes are significantly higher than the ones in the initial space. However, regarding classification rate, RSP3 stands as a clear competitive algorithm as results are never significantly worse than the ones by the PS strategies.

The evolutionary algorithm ENPC achieves noticeable reduction rates as, except when compared to CHC, figures are significantly similar to, or even better than, the considered PS strategies. Classification rates are, in general, similar to the ones in PS except for the CHC algorithm, in which ENPC always shows a significant improvement, and some particular cases of FCNN, in which ENPC shows a significant decrease.

MSE shows the poorest performance of the considered algorithms with respect to accuracy. This can be clearly seen when compared to the FCNN or the CHC cases in which the results of the tests are significantly worse than the ones of the other PG strategies. Although this poor performance could be due to a sharp reduction rate, this is not the case. For instance, if we check the ENPC and MSE

Table 5 Results obtained for the statistical significance tests comparing PG in the DS space with PS in the structural one

PG	DS method	ALL		PS					
				FCNN		1-FN$_{0.3}$		CHC	
		Acc	Size	Acc	Size	Acc	Size	Acc	Size
RSP3	RandomC(5)	✗	✔	=	✗	=	✗	✔	✗
	RandomC(10)	✗	✔	=	✗	✔	✗	✔	✗
	RandomC(15)	✗	✔	=	✗	✔	✗	✔	✗
	kCenters(5)	✗	✔	=	✗	✔	✗	✔	✗
	kCenters(10)	✗	✔	=	✗	✔	✗	✔	✗
	kCenters(15)	✗	✔	=	✗	✔	✗	✔	✗
	EditCon	✗	✔	=	✗	✔	✗	✔	✗
ENPC	RandomC(5)	✗	✔	✗	=	=	=	✔	✗
	RandomC(10)	✗	✔	=	✔	=	=	✔	✗
	RandomC(15)	✗	✔	=	✔	=	=	✔	✗
	kCenters(5)	✗	✔	✗	=	=	=	✔	✗
	kCenters(10)	✗	✔	=	✔	=	=	✔	✗
	kCenters(15)	✗	✔	=	✔	=	=	✔	✗
	EditCon	✗	✔	✗	=	=	=	✔	✗
MSE	RandomC(5)	✗	✔	✗	=	=	=	=	✗
	RandomC(10)	✗	✔	✗	=	=	=	=	✗
	RandomC(15)	✗	✔	✗	=	=	=	✔	✗
	kCenters(5)	✗	✔	✗	=	=	=	✔	✗
	kCenters(10)	✗	✔	✗	=	=	=	✔	✗
	kCenters(15)	✗	✔	✗	=	=	=	✔	✗
	EditCon	✗	✔	✗	=	=	=	=	✗

For each comparison, accuracy and set size are assessed. Symbols ✔, ✗, and = state that results achieved by elements in the rows significantly improve, decrease, or do not differ, respectively, to the results by the elements in the columns. Significance has been set to $p < 0.05$

cases with RandomC(10) against FCNN, we can see that, while the former achieves accuracy results similar to the PS algorithm with a significantly lower set size, MSE shows worse classification results than the PS strategy with a similar set size.

5.2 Discussion

Experiments show that the performance of PG in the feature-based space seems to be somehow bounded by the DS mapping process: The PG configurations considered are capable of retrieving classification rates similar to the ones achieved when not performing data reduction in this new space; however, these figures are still far from the ones achieved in the original space without any reduction either. While this could be a particularity of a precise DS method, our experiments show that this effect is inherent to the mapping process itself. A possibility to consider to palliate this effect would be the use of more robust embedding algorithms.

Taking this limitation into account, we can see the proposed strategy of PG in the DS space as very competitive when compared to PS in the initial space: Considering the performance limitation due to the space mapping, and except for the case in which we compare MSE with FCNN, accuracy results achieved by PG are similar or even better than the ones by PS. This proves that PG algorithms can cope with the aforementioned drop.

Regarding the reduction capabilities, the proposed scheme achieves similar figures to the ones obtained by PS in the initial space: Except when considering RSP3, which does not achieve great reduction figures, or when comparing to CHC, which performs the sharpest reduction, sizes do not significantly differ in the comparison.

In general, we can see that the proposed strategy of mapping the initial structural representation to a statistical one for then performing PG is able to achieve classification and reduction rates significantly similar to the ones obtained by PS in the initial space. This fact clearly questions the usefulness of the method as it does not improve over the results obtained in the classical scenario. However, if we consider computational cost for the classification, we can see that the proposed strategy stands as a very interesting alternative as it achieves statistically similar results in significantly shorter (several orders of magnitude) time lapses (see Table 3) than the structural representations. Additionally, if speed is the major concern, the proposed DS mapping with PG still stands as an interesting approach since a remarkable amount of fast search algorithms have been proposed for feature-based space, in contrast to fast searching in metric spaces [33].

6 Conclusions

Prototype generation techniques for data reduction in instance-based classification aim at creating new data out of the elements of a given set so as to lower memory requirements while precisely defining the decision boundaries. Although these methods are commonly used in statistical pattern recognition, they turn out to be quite challenging for structural data as the merging operations required cannot be as clearly defined as in the former approach. It has been proposed the use of dissimilarity space representations, which allow us to map structural data representations onto feature ones, to benefit from the advantages prototype generation methods depict.

The experimentation performed shows some important outcomes. In our results, PG approaches applied to structural data using DS representation are capable of competing with PS methods in the original space even though the mapping process implies information losses. Nevertheless, when compared to the figures obtained in the non-reduced structural space, PG methods depict lower accuracy results. Finally, classification using DS representations has been proved as a faster option than the one performed in the structural space as costly distance functions like edit distance are replaced by low-dimensional Euclidean distance. This evinces the proposed approach as an interesting tradeoff option between precision and time consumption.

Given the accuracy drop observed in the dissimilarity space mapping process, more sophisticated methods should be considered to check whether that loss could be somehow avoided. Additionally, experimentation could be extended including other prototype generation algorithms not considered in the present study.

Acknowledgments This work was partially supported by the Spanish Ministerio de Educación, Cultura y Deporte through a FPU fellowship (AP2012–0939), Vicerrectorado de Investigación, Desarrollo e Innovación de la Universidad de Alicante through FPU program (UAFPU2014–5883), and the Spanish Ministerio de Economía y Competitividad through Project TIMuL (No. TIN2013-48152-C2-1-R supported by EU FEDER funds).

References

1. Abreu J, Rico-Juan JR (2014) A new iterative algorithm for computing a quality approximated median of strings based on edit operations. Pattern Recognit Lett 36:74–80
2. Angiulli F (2007) Fast nearest neighbor condensation for large data sets classification. IEEE Trans Knowl Data Eng 19(11): 1450–1464
3. Arthur D, Vassilvitskii S (2007) K-means++: the advantages of careful seeding. In: Proceedings of the eighteenth annual ACM-SIAM symposium on discrete algorithms, SODA '07Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, pp 1027–1035

4. Borzeshi EZ, Piccardi M, Riesen K, Bunke H (2013) Discriminative prototype selection methods for graph embedding. Pattern Recognit 46(6):1648–1657
5. Bunke H, Riesen K (2012) Towards the unification of structural and statistical pattern recognition. Pattern Recognit Lett 33(7):811–825
6. Calvo-Zaragoza J, Oncina J (2014) Recognition of pen-based music notation: the HOMUS dataset. In: Proceedings of the 22nd international conference on pattern recognition, ICPR, pp 3038–3043
7. Calvo-Zaragoza J, Valero-Mas JJ, Rico-Juan JR (2015) Improving kNN multi-label classification in prototype selection scenarios using class proposals. Pattern Recognit 48(5):1608–1622
8. Calvo-Zaragoza J, Valero-Mas JJ, Rico-Juan JR (2015) Prototype generation on structural data using dissimilarity space representation: a case of study. In: Paredes R, Cardoso JS, Pardo XM (eds) 7th Iberian conference on pattern recognition and image analysis (IbPRIA). Springer, Santiago de Compostela, pp 72–82
9. Cano JR, Herrera F, Lozano M (2006) On the combination of evolutionary algorithms and stratified strategies for training set selection in data mining. Appl Soft Comput 6(3):323–332
10. Decaestecker C (1997) Finding prototypes for nearest neighbour classification by means of gradient descent and deterministic annealing. Pattern Recognit 30(2):281–288
11. Demsar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30
12. Duda RO, Hart PE, Stork DG (2001) Pattern classification. Wiley, New York
13. Duin RPW, Pekalska E (2012) The dissimilarity space: bridging structural and statistical pattern recognition. Pattern Recognit Lett 33(7):826–832
14. Eshelman LJ (1990) The CHC adaptive search algorithm: how to have safe search when engaging in nontraditional genetic recombination. In: Proceedings of the first workshop on foundations of genetic algorithms, Indiana, USA, pp 265–283
15. Fernández F, Isasi P (2004) Evolutionary design of nearest prototype classifiers. J Heuristics 10(4):431–454
16. Ferrer M, Bunke H (2010) An iterative algorithm for approximate median graph computation. In: Pattern recognition (ICPR), 20th international conference on, pp 1562–1565
17. Freeman H (1961) On the encoding of arbitrary geometric configurations. Electron Comput IRE Trans EC-10(2):260–268
18. Garcia S, Derrac J, Cano J, Herrera F (2012) Prototype selection for nearest neighbor classification: taxonomy and empirical study. IEEE Trans Pattern Anal Mach Intell 34(3):417–435
19. García S, Luengo J, Herrera F (2015) Data preprocessing in data mining. Springer, Berlin
20. García-Pedrajas N, De Haro-García A (2014) Boosting instance selection algorithms. Knowl Based Syst 67:342–360
21. Hart P (1968) The condensed nearest neighbor rule (corresp.). IEEE Trans Inform Theory 14(3):515–516
22. de la Higuera C, Casacuberta F (2000) Topology of strings: median string is NP-complete. Theor Comput Sci 230(1–2):39–48
23. Hjaltason G, Samet H (2003) Properties of embedding methods for similarity searching in metric spaces. Pattern Anal Mach Intell IEEE Trans 25(5):530–549
24. Hull J (1994) A database for handwritten text recognition research. IEEE Trans Pattern Anal 16(5):550–554
25. Kotsiantis SB, Kanellopoulos D, Pintelas PE (2007) Data preprocessing for supervised learning. Int J Comput Electr Autom Control Inf Eng 1(12):4091–4096
26. Latecki LJ, Lakmper R, Eckhardt U (2000) Shape descriptors for non-rigid shapes with a single closed contour. In: Proceedings of IEEE conference computer vision and pattern recognition, pp 424–429
27. LeCun Y, Bottou L, Bengio Y, Haffner P (2001) Gradient-based learning applied to document recognition. In: Haykin S, Kosko B (eds) Intelligent signal processing. IEEE Press, Piscataway, NJ, USA, pp 306–351
28. Mitchell TM (1997) Machine learning. McGraw-Hill Inc, NY
29. Nanni L, Lumini A (2011) Prototype reduction techniques: a comparison among different approaches. Expert Syst Appl 38(9):11820–11828. doi:10.1016/j.eswa.2011.03.070
30. Pekalska E, Duin RPW (2005) The dissimilarity representation for pattern recognition: foundations and applications (machine perception and artificial intelligence). World Scientific Publishing Co., Inc, Singapore
31. Rico-Juan JR, Iñesta JM (2012) New rank methods for reducing the size of the training set using the nearest neighbor rule. Pattern Recognit Lett 33(5):654–660
32. Sánchez J (2004) High training set size reduction by space partitioning and prototype abstraction. Pattern Recognit 37(7):1561–1564
33. Serrano A, Micó L, Oncina J (2013) Which fast nearest neighbour search algorithm to use? In: Sanches JM, Micó L, Cardoso JS (eds) 6th Iberian conference on pattern recognition and image analysis (IbPRIA). Funchal, Madeira, Portugal
34. Triguero I, Derrac J, García S, Herrera F (2012) A taxonomy and experimental study on prototype generation for nearest neighbor classification. IEEE Trans Syst Man Cybern C 42(1):86–100
35. Tsai CF, Eberle W, Chu CY (2013) Genetic algorithms in feature and instance selection. Knowl Based Syst 39:240–247
36. Wagner RA, Fischer MJ (1974) The string-to-string correction problem. J ACM 21(1):168–173
37. Wilson DL (1972) Asymptotic properties of nearest neighbor rules using edited data. IEEE Trans Syst Man Cybern 2(3): 408–421

Part III

Unpublished works

Chapter 10

Recognition of Pen-based Music Notation

Abstract

This work presents a statistical model to recognize pen-based music compositions using stroke recognition algorithms and finite-state machines. The series of strokes received as input is mapped onto a stochastic representation, which is combined with a formal language that describes musical symbols in terms of stroke primitives. Then, a Probabilistic Finite-State Automaton is obtained, which defines probabilities over the set of musical sequences. This model is eventually crossed with a semantic language to avoid sequences that does not make musical sense. Finally, a decoding strategy is applied in order to output a hypothesis about the musical sequence actually written. Comprehensive experimentation with several decoding algorithms, stroke similarity measures and probability density estimators are tested and evaluated following different metrics of interest. Results found have shown the goodness of the proposed model, obtaining competitive performances in all metrics and scenarios considered.

10.1 Introduction

Despite several efforts to develop light and friendly software for music score edition, many composers still prefer pen and paper to express their new music compositions. Once the artistic process is over, however, they resort to this kind of tools to transcribe the musical content to some digital format. Although this process is not always mandatory, it entails several benefits such as an easier storage, organization, distribution or reproduction of the music scores.

A profitable way of performing the whole process is by means of a pen-based music notation recognition system. Such systems make use of an electronic pen, with which music symbols are drawn over a digital surface. The system collects user strokes and then processes them to recognize the composition. The goal is to present the score actually written to the user in the desired format. It should be noted that this task can be considered very similar to the Optical Character Recognition (OCR) task, for which pen-based research has been widely carried out (Plamondon and Srihari, 2000; Mondal et al., 2009; Liu et al., 2013). Nevertheless, the complexity of musical notation in comparison to text leads to the need of specific developments as pointed out by Bainbridge and Bell (2001).

A straightforward approach to solve the task stated above is to resort to Optical Music Recognition (OMR) systems, which are devoted to understanding music scores from their image. That is, an image can be generated from pen strokes to make it pass through an OMR system (offline recognition). Nevertheless, the performance of current OMR systems is far from optimal, especially in the case of handwritten notation (Rebelo et al., 2012). Note that the main intention of a pen-based score composition system is to provide musicians with an interface as friendly as possible. Therefore, they are expected to compose without paying attention to achieving a perfect handwriting style so that notation would be even harder than usual to be recognized.

Fortunately, pen-based (or online) recognition brings new features that make the task be very different to the offline case, some of which include:

- Staff lines: a staff is composed of five parallel lines, in which musical symbols are placed in different heights depending of their pitch. Staff detection and removal is one of the most difficult issues to overcome in offline OMR systems (Dalitz et al., 2008), since symbol detection and recognition are based on the accuracy of this step. Nevertheless, in a pen-based system the problem is harmless because the staff lines are handled by the system itself and their removal can be done effortlessly.

- Segmentation: the input of a pen-based system is naturally segmented by pen strokes. Each stroke is easily detected by pen-down and pen-up actions over the digital surface. This allows avoiding a lot of potential mistakes that may be caused by a bad segmentation in OMR systems.

- Online data: drawing symbols in the pen-based scenario produces a time signal of coordinates indicating the path followed by the pen. Although the image of the score can be rebuilt from the strokes, online data is available to be used in the recognition. This dynamic information is valuable for shape recognition (Kim and Sin, 2014).

All these features lead towards the development of specific pen-based algorithms that are able to improve the performance of current offline OMR systems.

In this work it is proposed an approach for solving this task using finite-state machines and dissimilarity measures between strokes. For a given input, the combination of these artefacts is able to produce a probabilistic model that defines the probability of each possible musical sequence. The use of decoding algorithms (for instance, searching the most probable sequence) provides a hypothesis about the sequence actually written.

The rest of the paper is structured as follows: Section 10.2 presents some related work about pen-based music recognition; Section 10.3 delves in the details of the construction of the probabilistic model; Section 10.4 describes the experimentation carried out and the results obtained; finally, Section 10.5 draws the main conclusions and discusses some future work.

10.2 Related works

Notwithstanding the benefits offered by pen-based music recognition systems on music composition, few attention has been paid to their development. Decades ago, some works were proposed based on the use of a gesture alphabets so that each musical symbol was associated with a simple gesture (Anstice et al., 1996; Ng et al., 1998). These gestures were generally mnemonic of the actual symbols they represented. The main concern of these approaches is that they did not provide a natural interface to musicians, who had to learn a new way of writing music. Poláček et al. (2009) developed a similar idea for its use on low-resolution displays.

More recently, many works have dealt with the problem of recognizing pen-based isolated musical symbols. For instance, George (2003) used the images generated by the digital pen to learn an Artificial Neural Network to recognize the symbols. Lee et al. (2010) proposed the use of Hidden Markov Models for the recognition of some of the most common musical symbols using different features of the shape drawn by the pen. Calvo-Zaragoza and Oncina (2014)[1] presented a free dataset of pen-based music symbols written by different musicians, as well as an experimental baseline study taking into account several recognition algorithms.

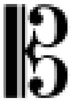
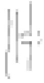
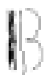
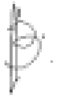
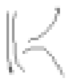
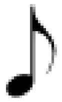
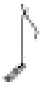
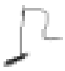
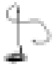
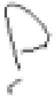
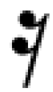
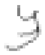
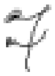
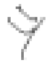
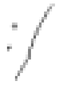
While the recognition of isolated symbol might have its interest, the actual challenge is the recognition of pen-based music sequences. On this issue, there have been less attempts. Miyao and Maruyama (2004, 2007) proposed a system based on the recognition of predefined strokes primitives (such as note-heads, lines, dots, etc.). Once the strokes were classified by using both time-series data and image features, musical symbols were detected following a heuristic approach that made use of a set of predefined rules. Macé et al. (2005) proposed a generic approach for pen-based document recognition applied to music scores. This approach was based on the use of a stroke recognizer and *ad-hoc* Context-Free Grammars, which defined the spatial structure of the document.

Unfortunately, these solutions were not satisfactory from a user point of view. The only way of having symbols recognized is by following the rules of the system. Therefore, these solutions forced users to adapt to system style when what should be pursued is just the opposite.

[1] This cite corresponds to the work presented in Chapter 6.

For all comments above, there is still a need of developing a user-centred music composition system. The main goals are to provide an ergonomic interface, which is indeed fulfilled with the use of the e-pen, and to provide an adaptive behaviour. To serve as an example, Table 10.1 shows some musical symbols written by different musicians. This implies that recognition must be guided by a learning process, which allows the system to know how musicians are going to write their music.

Table 10.1: Different handwriting styles for some musical symbols.

Label	Symbol	Style 1	Style 2	Style 3	Style 4
C-Clef					
Eighth Note					
Sixteenth Rest					

Our proposal follows a stroke-based approach, in which the writing style is learned from labelled data. By using pairwise stroke similarity functions it is possible to build a finite-state machine that represents the stochastic set of solutions. Then, following some decoding strategy, a hypothesis about the sequence actually written is provided. Next section will describe in depth how to build such model.

10.3 Recognition with Finite-State Machines

This section describes our approach to recognize handwritten musical sequences written with an e-pen.

In this work it is assumed that a training set with samples of how the musical symbols are written by users is available. This corpus might be obtained by either asking the user to go through a training phase before using the tool or by using some existing dataset (like the one mentioned in the previous section). This will allow defining a set of construction rules from stroke primitives to musical symbols.

For a given input, the probability of each stroke to belong to each stroke primitive is computed. This estimation, as well as the construction rules defined in the training phase, will be used to obtain a probabilistic machine that is able to give a probability to each of the possible musical sequences. Nevertheless, given the formal system in which music is framed, it is known that there exists several sequences of musical symbols that do not make sense. Therefore, a semantic model will be used to tune this probabilistic machine in order to avoid those sequences that are not well-formed.

Finally, several decoding strategies will be applied to provide an hypothesis that will be considered as solution to the task.

10.3.1 Symbol generation from stroke primitives

The input to the system consists of the set of time-ordered strokes drawn by the user. Each symbol may be written with a single stroke or with several ones. For instance, a *Quarter Note* (♩) can be a *black note head* followed by a *stem* (Fig. 10.1(a)), or just the *quarter note* primitive if the symbol was written with a single stroke (Fig. 10.1(b)). If symbols are to be recognized from this kind of input, it is needed to know how each musical symbol can be written from strokes.

(a) Two strokes (b) One stroke

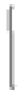
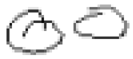
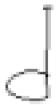
Figure 10.1: *Quarter Note* written with different sets of strokes.

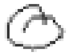
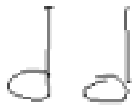
As mentioned above, it is assumed that a dataset of labeled pen-based musical symbols is available. Therefore, we have a set of series of strokes labeled by the musical symbol they represent. Due to the stroke feature space, it is very unlikely to have equal strokes written more than once so using this dataset to learn may be useless. Nevertheless, if *similar* strokes are grouped under the same label, the complexity of this process is diminished, allowing a higher generalization as well.

Once strokes have been labelled (we will revisit this issue later), musical symbols can be defined from sequences of stroke primitives. Let us denote $\Sigma = \{\sigma_1, \ldots, \sigma_{|\Sigma|}\}$ the set of musical symbols and $\Pi = \{\pi_1, \ldots, \pi_{|\Pi|}\}$ the set of stroke primitives (possible labels assigned to a stroke). Table 10.2 illustrates an example of dataset of musical symbols, in which strokes are also labelled. In this case, $\Sigma = \{$𝅗𝅥, 𝅝$\}$ and $\Pi = \{st, wh, hn\}$.

From the dataset of labelled musical symbols followed by their labelled sequence of strokes, a set of construction rules $R = \{(\sigma, (\pi_{i_1}, \ldots, \pi_{i_n})) : \sigma \in \Sigma, \pi_{i_j} \in \Pi\}_{i=1}^{|R|}$ can be obtained. That is, sequences of stroke primitives describing how specific musical symbols can be written. For instance, the dataset showed previously would be represented by the following set of rules:

Table 10.2: Example of a dataset of pen-based musical symbols with labeled strokes.

Stroke Primitives (Π)		
stem (st)	white notehead (wn)	half note (hn)
		

Musical Symbols (Σ)	
Whole Note (𝅝)	Half Note (𝅗𝅥)
	

$$\begin{array}{l} (\text{𝅝}, \text{wn}) \\ (\text{𝅗𝅥}, \text{hn}) \\ (\text{𝅗𝅥}, \text{st wn}) \end{array} \tag{10.1}$$

It should be stressed that it is possible to find the same musical symbol defined many times by the same sequence of stroke primitives. In other words, there are some stroke sequences that are more likely to describe a musical symbol than others. We use this fact to define a prior probability for each rule in R.

Let $R_\sigma = \{(\sigma, \bar{\pi}) \in R : \sigma \in \Sigma, \bar{\pi} \in \Pi^+\}$ represent the set of rules whose musical symbol is σ. Then, the prior probability of a rule $(\sigma, \bar{\pi}) \in R_\sigma$, denoted by $p(\bar{\pi}|\sigma)$, is defined as:

$$p(\bar{\pi}|\sigma) = \frac{\#((\sigma, \bar{\pi}))}{\sum_{(\sigma,\bar{\pi}') \in R_\sigma} \#((\sigma, \bar{\pi}'))} \tag{10.2}$$

where $\#((\sigma, \bar{\pi}))$ denotes the number of times that rule appears in the dataset. Note that $\sum_{(\sigma,\bar{\pi}) \in R_\sigma} p(\bar{\pi}|\sigma) = 1$, for any $\sigma \in \Sigma$.

At this point, there still exists the open question of how to label each stroke. It is clear that the alphabet of musical symbols is defined by the task itself, but it is not for the stroke primitives. This problem has been covered in the work of Calvo-Zaragoza et al. (2015a)[2], in which an automatic labelling of strokes is proposed given an *ambiguity* rate allowed. A stroke labelling is *ambiguous* if, and only if, two different musical symbols can be defined by the same set of primitives. We use

[2] This cite corresponds to the work presented in Chapter 7.

here this approach to label the strokes of our dataset. Several ambiguity rates will be considered during the experimentation. The more ambiguity allowed, the more accurate the stroke recognition. In turn, the actual recognition of music symbols becomes harder.

10.3.2 Input processing

Next lines describe how to build a finite-state machine that defines probabilities over the allowed musical sequences. This machine will be conditioned by both the input received and the construction rules presented in the previous section, as well as a formal language that defines which sequences make musical sense.

Probability estimation

The input to the system is given by a sequence of strokes $\bar{s} = (s_1, \ldots, s_{|\bar{s}|})$, in which each stroke is defined by an ordered sequence of 2D coordinates.

The first step to recognize this input is to know which types of strokes have been actually written. Since this process is not error-free, a way of approaching it is by computing the probability of each received stroke to be each stroke primitive considered.

Although there exists several ways of computing probabilities from labeled data (for instance, Hidden Markov Models), our estimation is going to be governed by dissimilarity functions between strokes. We denote this dissimilarity as $d(\cdot, \cdot)$. Our choice is justified by the fact that the stroke labeling is directed by a dissimilarity function. Furthermore, this paradigm is specially suitable for interactive scenarios like the one found in our task, as the simple addition of new prototypes to the training set is sufficient for incremental learning, whereas the size of the dataset can be controlled by dissimilarity-based data reduction algorithms (García et al., 2015).

To estimate a probability from a given dissimilarity, two different strategies are considered: Parzen Window and Nearest Neighbour.

- Parzen Window (Parzen, 1962) is a non-parametric technique to estimate probability density functions from training samples. Given a series of samples $x_1, x_2, \ldots, x_n$ from an unknown distribution p, an estimated density $\hat{p}$ in a point x following Parzen Window method is

  $$\hat{p}(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{h} \varphi \left(\frac{d(x, x_i)}{h} \right) \tag{10.3}$$

  The term φ refers to the *window function*, a symmetric function that integrates to one. The parameter h, called the bandwidth of the window, should be defined according to the volume of the considered region.

One of the main issues of the Parzen Window estimation is the choice of the window function φ. In practice, it is commonly assumed a standard Gaussian kernel:

$$\varphi(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}u^2} \tag{10.4}$$

- Nearest Neighbour: the problem of choosing the adequate window function φ can be avoided by only considering the nearest sample of the training data. This is called Nearest Neighbour estimation (Duda et al., 2001).

  Given a series of samples $x_1, x_2, \ldots, x_n$ from an unknown density function p, a common estimation for a point x becomes

$$\hat{p}(x) = \frac{1}{\min_{i=1}^{n} d(x, x_i)} \tag{10.5}$$

Note that a dissimilarity function between strokes is needed to make use of the previous probability density function estimators. The digital surface collects the strokes at a fixed sampling rate so that each one may contain a variable number of 2D points. Several functions for measuring dissimilarity can be applied to this kind of data. Those considered in this work include:

- Dynamic Time Warping (DTW): a technique for measuring the dissimilarity between two time signals which may be of different duration. It was firstly used in speech recognition Sakoe and Chiba (1990), although its use has been extended to other fields Hartmann and Link (2010); Kim and Sin (2014).

- Edit distance with Freeman Chain Code (FCC): the sequence of points representing a stroke is converted into a string using a codification based on Freeman Chain Codes Freeman (1961). Then, the common Edit distance Levenshtein (1966) between strings is applied.

- Normalized Stroke (NS): the whole set of points of the stroke is normalized to a sequence of n points by an equally resampling technique. Therefore, a stroke is characterized by an n-dimensional feature vector of 2D coordinates. Given two vectors of this kind, an accumulated Euclidean distance between the points of the sequences can be computed.

- Edit distance for Ordered Set of Points (OSP) Rico-Juan and Iñesta (2006): an extension of the edit distance for its use over sequences of consecutive points.

Building a Probabilistic Automaton

At this point it is known both the probability of each stroke to be each stroke primitive and the musical symbol construction rules from stroke primitives. This knowledge can be merged to obtain a machine that defines the probability of sequences of musical symbols.

The machine to be built is a Probabilistic Finite-State Automaton (PFA). A PFA is a generative device for which there are a number of possible definitions (Paz, 1971; Vidal et al., 2005).

Definition 1. *A* Probabilistic Finite-State Transducer (PFA) *is a tuple* $\mathcal{A} = \langle \Sigma, Q, \mathbb{I}, \mathbb{F}, \delta \rangle$, *where:*

- Σ *is the alphabet;*
- $Q = \{q_1, \ldots, q_{|Q|}\}$ *is a finite set of* states*;*
- $\mathbb{I} : Q \to \mathbb{R} \cap [0, 1]$ *(initial probabilities);*
- $\mathbb{F} : Q \to \mathbb{R} \cap [0, 1]$ *(final probabilities);*
- $\delta : Q \times \Sigma \times Q \to \mathbb{R} \cap [0, 1]$ *is the complete transition function;* $\delta(q, a, q') = 0$ *can be interpreted as "no transition from* q *to* q' *labelled with* a*".*

  $\mathbb{I}$, δ *and* $\mathbb{F}$ *are functions such that:*

$$\sum_{q \in Q} \mathbb{I}(q) = 1, \tag{10.6}$$

and $\forall q \in Q$,

$$\mathbb{F}(q) + \sum_{a \in \Sigma,\ q' \in Q} \delta(q, a, q') = 1. \tag{10.7}$$

Given $x \in \Sigma^*$, an *accepting* x-path is a sequence $\gamma = q_{i_0} a_1 q_{i_1} a_2 \ldots a_n q_{i_n}$ where $x = a_1 \cdots a_n$, $a_i \in \Sigma$ and $\delta(q_{i_{j-1}}, a_j, q_{i_j}) \neq 0$, $\forall j$ such that $1 \leq j \leq n$. Let $\Gamma_{\mathcal{A}}(x)$ be the set of all paths accepting x. The probability of the path γ is defined as $Pr_{\mathcal{A}}(\gamma) = \mathbb{I}(q_{i_0}) \cdot \prod_{j=1}^{n} \delta(q_{i_{j-1}}, a_j, q_{i_j}) \cdot \mathbb{F}(q_{i_n})$ and the probability of the sequence x is obtained by summing over the probabilities of all the paths in $\Gamma_{\mathcal{A}}(x)$.

The construction of our PFA is done as described in Algorithm 1. The machine generates as many states as strokes are in the input plus 1. The ith state represents that every stroke from the first until the $(i-1)$th has been processed (state 0 means that no stroke has been processed so far). This is why the only initial state is the first and the only final state is the last one. For each state, the set of construction rules is queried and a new edge is created for every single rule. These edges go from the current state to a state as far as strokes primitives contain the rule. The label of the edge is given by the musical symbol of the rule. Note that those edges that would end beyond the last state will be discarded. Finally, the probability of

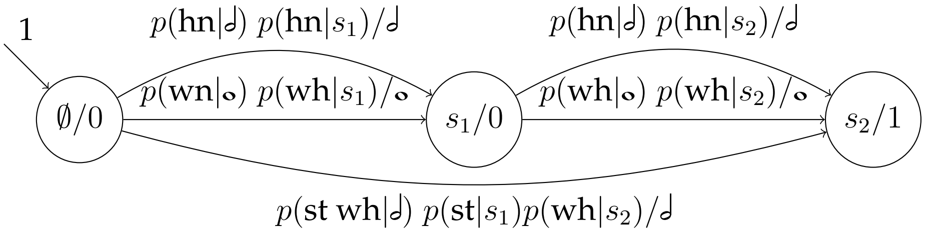


Figure 10.2: Example of PFA generated from an input of two strokes and the set of construction rules defined in Eq. 10.1. An arrow toward a state represents its initial probability (omitted when 0). Text inside states represents the last stroke processed and the probability of stop. Text over edges represents the probability of the transition and its label.

these edges is given by the product of the probability of the strokes to be the primitives of the rule as well as by the prior probability of the rule itself. Calculation of these probabilities was showed above so it is assumed that they are available when running Algorithm 1.

```
**Data**: $\bar{s} = (s_1, \ldots, s_{|\bar{s}|})$, $R = \{(\sigma, (\pi_{i_1}, \ldots, \pi_{i_n})) : \sigma \in \Sigma, \pi_{i_j} \in \Pi\}^{|R|}$
**Result**: $(Q, \Sigma, \mathbb{I}, \mathbb{F}, \delta)$ : PFA
$Q \leftarrow \{q_0, \ldots, q_{|\bar{s}|}\}$
$\mathbb{I}(q_0) \leftarrow 1$
$\mathbb{F}(q_{|\bar{s}|}) \leftarrow 1$
**forall the** $q_i \in Q$ **do**
    **forall the** $(\sigma, (\pi_{i_1} \ldots \pi_{i_n})) \in R$ **do**
        **if** $i + n \leq |\bar{s}|$ **then**
            $\delta(q_i, \sigma, q_{i+n}) \leftarrow p(\pi_1 \ldots \pi_n|\sigma) \prod_{k=0}^{n} p(\pi_{k+1}|s_{i+k})$
        **end**
    **end**
**end**
```

Algorithm 1: Building a PFA from an input sequence and the set of symbol construction rules.

Figure 10.2 shows an example of PFA given a input sequence $\bar{s} = s_1 s_2$ and the set of construction rules of Eq. 10.1. Although this is not the case, different paths may have the same probability depending on the set of rules.

Avoiding not well-formed sequences

The machine obtained in the previous section is able to define a probability for every sequence in Σ^*. However, it is clear that not all these sequences are *grammatically* correct. Hence, the next step is to ensure that only well-formed sequences have a non-null probability of being produced.

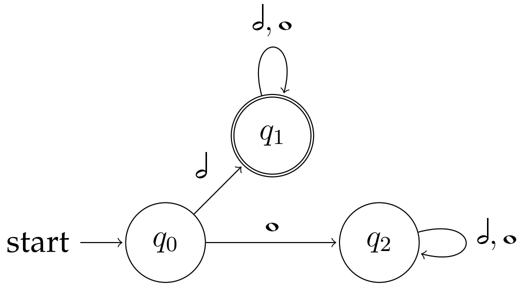


Figure 10.3: An example of DFA that accepts only sequences whose first symbol is a *Half note*. Double circle indicates a final state.

To this end, it is assumed that well-formed musical sequences can be defined by means of a regular language. That way it is possible to build a Deterministic Finite-State Automaton that only accepts those sequences that fulfill the music constraints.

Definition 2. *A* Deterministic Finite-State Automaton (DFA) *is a tuple* $\mathcal{D} = (Q, \Sigma, \delta, q_0, F)$ *where:*

- Q *is the set of states;*
- Σ *is the alphabet;*
- $\delta : Q \times \Sigma \to Q$ *is the transition function;*
- q_0 *is the initial state;*
- $F \subseteq Q$ *is the set of accepting states.*

A sequence $\bar{\sigma} \in \Sigma^*$ *is accepted by the* DFA *if, and only if,* $\delta^*(q_0, \hat{\sigma}) \in F$.

Let us once again consider a little alphabet of musical symbols $\Sigma = \{$𝅗𝅥, 𝅝$\}$. Figure 10.3 shows a toy DFA which only accepts sequences that begin with a *Half note* (𝅗𝅥). Note that this language does not make any musical sense but it is used here as an easy example to guide the explanation.

The semantic information provided by the DFA can be merged with the previous PFA. Our goal is to change the probabilities so that sequences that do not belong to the language are nullified (zero probability). To generate this machine, the intersection of the DFA and the PFA is computed. The output of this intersection is a new probabilistic machine for which sequences that do not belong to the language finish in a non-final state, *ie.* a state with a stop probability equal to 0.

The intersection between such machines can be obtained computationally by following Algorithm 2. Given a PFA $\mathcal{A}$ and a DFA $\mathcal{D}$, we get a new PFA $\mathcal{B}$. This new machine has $Q_{\mathcal{A}} \times Q_{\mathcal{D}}$ states. We denote each of these states using a pair (q_a, q_d). The first element indicates the state of $\mathcal{A}$ and the second one the state of $\mathcal{D}$ that the new state is representing. The initial probabilities of $\mathcal{B}$ are equal to those in $\mathcal{A}$

for every state that also represents an initial state in $\mathcal{D}$. Otherwise, probabilities are equal to 0. Similarly, the final probabilities of $\mathcal{B}$ are equal to those in $\mathcal{A}$ as long as the state is also in the set of final states of $\mathcal{D}$. Finally, the probability from a state (q_a, q_d) to $(q_{a'}, q_{d'})$ with a symbol σ is equal to the transition from q_a to $q_{a'}$ in $\mathcal{A}$ with this symbol as long as a transition from q_d to $q_{d'}$ with σ is allowed in $\mathcal{D}$. A post-processing step may normalize the new PFA so that it fulfils the conditions stated in Def. 1.

```
Data: $\mathcal{A} = (Q_{\mathcal{A}}, \Sigma, \mathbb{I}_{\mathcal{A}}, \mathbb{F}_{\mathcal{A}}, \delta_{\mathcal{A}})$ : PFA, $\mathcal{D} = (Q_{\mathcal{D}}, \Sigma, \delta_{\mathcal{D}}, q_{\mathcal{D}_0}, F_{\mathcal{D}})$ : DFA
Result: $\mathcal{B} = (Q_{\mathcal{B}}, \Sigma, \mathbb{I}_{\mathcal{B}}, \mathbb{F}_{\mathcal{B}}, \delta_{\mathcal{B}})$ : PFA
$Q_{\mathcal{B}} \leftarrow Q_{\mathcal{A}} \times Q_{\mathcal{D}}$
forall the $q_b = (q_a, q_d) \in Q_{\mathcal{B}} : q_d = q_{\mathcal{D}_0}$ do
    $\mathbb{I}_{\mathcal{B}}(q_b) \leftarrow \mathbb{I}_{\mathcal{A}}(q_a)$
end
forall the $q_b = (q_a, q_d) \in Q_{\mathcal{B}} : q_d \in F_{\mathcal{D}}$ do
    $\mathbb{F}_{\mathcal{B}}(q_b) \leftarrow \mathbb{F}_{\mathcal{A}}(q_a)$
end
forall the $q_b = (q_a, q_d) \in Q_{\mathcal{B}}$ do
    forall the $q_{b'} = (q_{a'}, q_{d'}) \in Q_{\mathcal{B}}$ do
        forall the $\sigma \in \Sigma$ do
            if $\delta_{\mathcal{D}}(q_d, \sigma) = q_{d'}$ then
                $\delta_{\mathcal{B}}(q_b, \sigma, q_{b'}) \leftarrow \delta_{\mathcal{A}}(q_a, \sigma, q_{a'})$
            end
        end
    end
end
```

Algorithm 2: Building a new PFA from the information provided by a PFA and a DFA.

An example of this intersection is showed in Fig. 10.4. For the sake of clarity, unreachable states are omitted. It should be stressed that now the path (𝅝, 𝅗𝅥) has the same probability as before but the final probability of the sequence is 0 due to the null stop probability of state (s_2, q_2). Useless paths could be removed to reduce the complexity of the generated machine.

10.3.3 Decoding strategies

At this point, our model is able to assign a probability to each valid sequence of musical symbols. The last step is to apply some kind of decoding strategy to output a hypothesis about the input sequence of strokes. The possibility of approaching this stage following different strategies is another interesting advantage of our approach. The decoding strategies considered here are listed below:

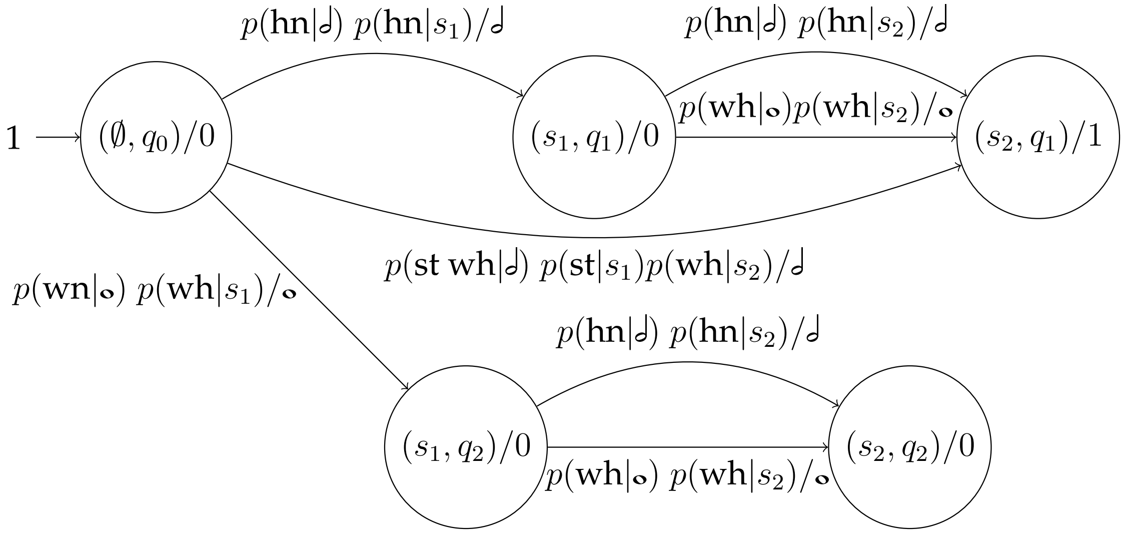

$p(\text{hn}|$𝅗𝅥$)$ $p(\text{hn}|s_1)/$𝅗𝅥
$p(\text{hn}|$𝅗𝅥$)$ $p(\text{hn}|s_2)/$𝅗𝅥
1
$(\emptyset, q_0)/0$
$(s_1, q_1)/0$
$p(\text{wh}|$𝅝$)p(\text{wh}|s_2)/$𝅝
$(s_2, q_1)/1$
$p(\text{st wh}|$𝅗𝅥$)$ $p(\text{st}|s_1)p(\text{wh}|s_2)/$𝅗𝅥
$p(\text{wn}|$𝅝$)$ $p(\text{wh}|s_1)/$𝅝
$p(\text{hn}|$𝅗𝅥$)$ $p(\text{hn}|s_2)/$𝅗𝅥
$(s_1, q_2)/0$
$(s_2, q_2)/0$
$p(\text{wh}|$𝅝$)$ $p(\text{wh}|s_2)/$𝅝

Figure 10.4: PFA obtained by the intersection of the PFA showed in Fig. 10.2 and the DFA showed in Fig. 10.3. An arrow toward a state represents its initial probability (omitted when 0). Text inside states represents the intersection of the original states where it comes from, as well as the probability of stop. Text over edges represents the probability of the transition and its label.

- Most probable path (MPP): it seeks for the most probable path of the PFA, and then it outputs the sequence associated to this path. This is typically computed with an efficient Viterbi alike algorithm.

- Most probable sequence (MPS): searching the most probable sequence of a distribution defined by a PFA is known to be NP-Hard (Casacuberta and de la Higuera, 2000). Nevertheless, a recent development allows its computation with a complexity of the inverse of the probability of the most probable sequence (de la Higuera and Oncina, 2014). We use here this approach.

- Optimum decoding to minimize number of corrections (MNC): if it is assumed that the output is going to be corrected by a human supervisor and after each correction the machine is allowed to output a new hypothesis (interactive approach), the optimum way of minimizing the expected number of sequential corrections is by computing the algorithm developed by Oncina (2009).

Each of these strategies will be compared experimentally.

10.4 Experimentation

This section describes the experimentation performed to assess the goodness of our proposal.

The HOMUS dataset (Calvo-Zaragoza and Oncina, 2014) of pen-based musical symbols will be used in this experimentation. This set contains 15200 samples from 100 different musicians. Taking advantage of its configuration, the series of experiments consists in recognizing semi-synthetic pen-based music scores from two different scenarios: user-dependent, in which both learning and test data come from the same musician, and user-independent, when all data available is mixed. The former scenario is aimed at measuring the performance when the user is known by the system, whereas the latter one simulates a more general situation.

The input of both experiments is a series of 1000 sequences of musical symbols, generated randomly respecting the language model defined. Each sequence is used to generate a pen-based score using the available data from the HOMUS. Figure 10.5 illustrates an example of this generation. Sequences generated contain 17.1 musical symbols, on average (± 3 of standard deviation), with a variable number of strokes depending on the synthetic generation.

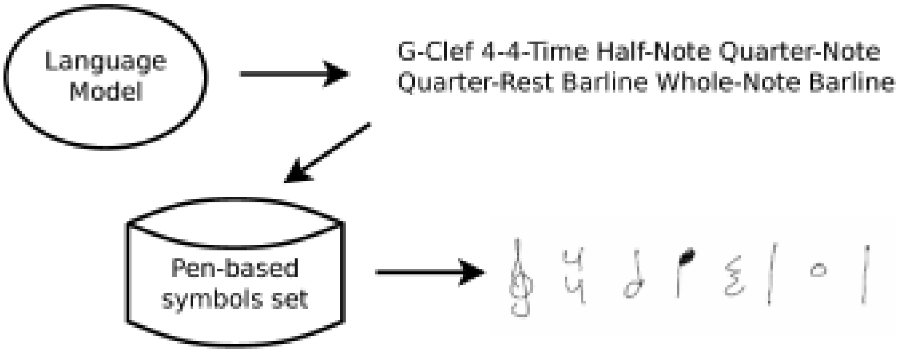

Language Model
G-Clef 4-4-Time Half-Note Quarter-Note
Quarter-Rest Barline Whole-Note Barline
Pen-based symbols set

Figure 10.5: Example of the procedure used to generate semi-synthetic pen-based scores for experimentation.

The stroke labeling process is performed automatically as mentioned in Section 10.3.1. We will allow different ambiguity rates, denoted as α, ranging from 0 to 0.3 depending on each scenario. Higher values of ambiguity are expected to give very poor results.

Recalling from Section 10.3.2, we need to develop a DFA that indicates which musical sequences are allowed. For these experiments we shall restrict ourselves to scores with a *common time* metric ($\frac{4}{4}$ or $\mathbf{c}$), one of the most common in modern Western music.

With respect to the performance evaluation, there are different ways of measuring the goodness of a hypothesis depending on the main goal pursued. To this end we take the following metrics of interest:

- Error rate (E_r): it provides the number of times (over the total) that the first hypothesis provided by the system is the actual input. It is used when a $0/1$ loss function is assumed.

- Average edit distance (E_d): it provides the expected number of corrections per sequence by computing the average edit distance between the solutions of the system and the actual inputs. It is the unnormalized version of the Word Error Rate.

- Average number of corrections (C): a good way of measuring the performance of this kind of systems is by counting the number of corrections the user would have to make until getting the sequence actually written (Vidal et al., 2007). After each correction, the system is allowed to recompute a new hypothesis, which is expected to be more accurate since some parts of the solution are known.

Next subsections present the results achieved in each scenario. It is important to stress that a comparative experiment with other approaches is not included. As stated in Section 2, no fully learning-based music notation recognition has been proposed so far and, therefore, a fair comparison is not possible.

10.4.1 User-dependent experiment

The user-dependent experiment consists in the following steps: for every sequence, a user is randomly selected and an automatic stroke labeling is performed with different ambiguity rates ranging from 0 to 0.2. For each of them, data is split into test and train set. The test set is used to build the pen-based score fulfilling the definition of the sequence generated, whereas the train set is used for both extracting the set of construction rules of musical symbols and training the stroke probability estimators. The machine obtained is then decoded to produce a hypothesis about the input sequence. Finally, aforementioned performance evaluation is applied.

Table 10.3 shows the results achieved in such experiment. Although further analysis is presented below, an initial remark to begin with is that performance is quite promising. Results yield that $90\ \%$ of these sequences are perfectly recognized ($E_r = 0.10$). From another point of view, results report 0.3 mistakes per sequence (E_d) or 0.28 corrections needed per sequence (C). This implies that the post-processing user correction phase could be assumed effortlessly, regardless of the use of an interactive approach.

Looking in more detail, the allowable ambiguity rate in stroke labeling has a special impact on the results. In most cases, results of the same probability estimator with the same decoding strategy are noticeably worse as ambiguity rate becomes higher. For instance, NN estimation with FCC and decoded by MPS achieves an error rate of 0.10 with $\alpha = 0$, but error rates of 0.17 and 0.38 with $\alpha = 0.1$ and $\alpha = 0.2$, respectively, are obtained. This tendency is depicted in

Table 10.3: Mean results of the user-dependent experiment with 1000 random sequences of length 17.1 (on average). Several allowable ambiguity rates (α) for the automatic stroke labeling are considered.

Prob.	Dissim.	Decod.	$\alpha = 0$			$\alpha = 0.1$			$\alpha = 0.2$		
			E_r	E_d	C	E_r	E_d	C	E_r	E_d	C
NN	FCC	MPP	0.11	0.31	0.36	0.19	0.62	0.31	0.42	1.54	0.73
		MPS	**0.10**	**0.30**	0.36	**0.17**	**0.55**	0.28	**0.38**	**1.39**	0.65
		MNC	**0.10**	0.31	0.35	0.18	0.59	**0.26**	0.40	1.46	**0.62**
	DTW	MPP	0.16	0.56	0.45	0.32	1.12	0.55	0.54	2.08	1
		MPS	0.16	0.57	0.44	0.33	1.13	0.53	0.48	1.97	0.9
		MNC	0.17	0.58	0.44	0.32	1.13	0.52	0.51	2.04	0.91
	NS	MPP	0.19	0.70	0.60	0.36	1.38	0.63	0.54	2.37	1.09
		MPS	0.19	0.70	0.59	0.34	1.33	0.59	0.52	2.25	1.02
		MNC	0.19	0.69	0.58	0.34	1.32	0.58	0.54	2.37	1.05
	OSP	MPP	**0.10**	**0.30**	0.51	0.23	0.8	0.37	0.45	1.87	0.81
		MPS	**0.10**	**0.30**	0.51	0.19	0.73	0.34	0.40	1.7	0.75
		MNC	**0.10**	**0.30**	0.51	0.2	0.75	0.35	0.43	1.8	0.77
Parzen	FCC	MPP	0.14	0.50	0.34	0.23	0.83	0.44	0.43	1.71	0.76
		MPS	0.13	0.50	0.33	0.2	0.74	0.38	0.39	1.57	0.68
		MNC	0.13	0.50	**0.31**	0.21	0.78	0.39	0.42	1.65	0.71
	DTW	MPP	0.27	1.01	0.79	0.49	2.11	1.14	0.65	3.06	1.60
		MPS	0.26	0.94	0.72	0.45	1.96	1.03	0.61	2.89	1.50
		MNC	0.26	0.94	0.70	0.45	1.92	0.99	0.65	2.94	1.46
	NS	MPP	0.19	0.72	0.52	0.37	1.56	0.79	0.57	2.69	1.29
		MPS	0.19	0.74	0.48	0.34	1.48	0.74	0.56	2.6	1.24
		MNC	0.19	0.74	0.48	0.34	1.51	0.75	0.57	2.66	1.22
	OSP	MPP	0.17	0.57	0.48	0.29	1.05	0.54	0.46	1.9	0.88
		MPS	0.16	0.56	0.45	0.23	0.88	0.46	0.43	1.73	0.79
		MNC	0.16	0.59	0.46	0.25	0.96	0.45	0.45	1.8	0.8

Fig. 10.6, which shows the average results among the different decoding strategies and probability density estimator for each ambiguity rate considered. Note that results degenerate as α becomes higher. A remarkable exception is presented for NN estimation with FCC when number of corrections (C) is considered as evaluation metric. In this case, the best result (which is also the best with respect to the whole experiment) are that with $\alpha = 0.1$.

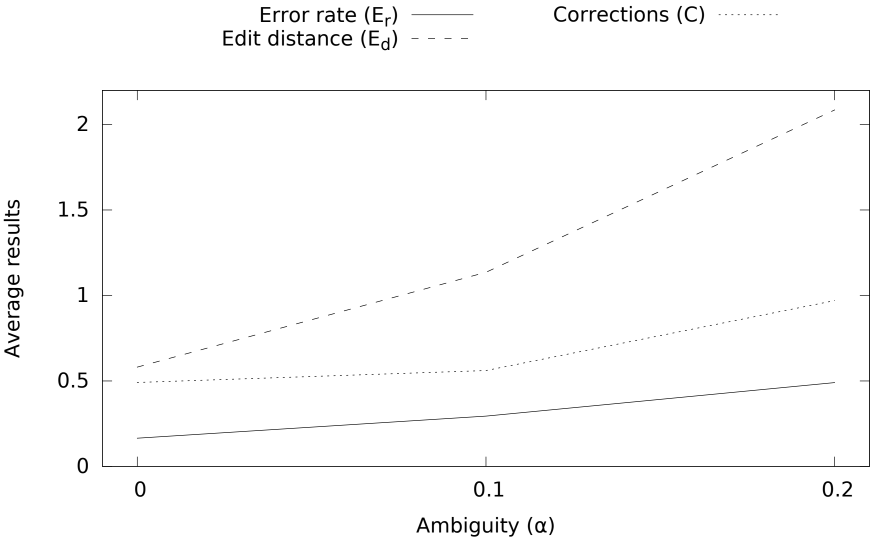


Figure 10.6: Impact of the stroke labelling ambiguity over the results achieved. Average results among all decoding strategies and probability density estimators are showed with respect to the ambiguity rate allowed.

Not surprisingly, stroke probability estimation is also a relevant factor. However, accuracy seems to be more in the dissimilarity function rather than the probability density estimator. As a whole conclusion to this respect, NN with FCC seems to be the most accurate stroke probability estimator for this task. On the other hand, results hardly vary among the decoding strategies considered, especially in the case of $\alpha = 0$. Despite this fact, best average results for E_r and E_d are achieved by the MPS strategy, whereas MNC has the lower number of corrections needed. MPP does have the worst average results in most of the cases.

10.4.2 User-independent experiment

The user-independent experiment exposes a more general scenario, that in which the system does not know what kind of handwriting style will receive and, therefore, must learn from samples of many different musicians. Given the large number of different strokes found in this scenario, the automatic labeling fails to obtain a non-ambiguous clustering using less than 500 labels (maximum considered). This is why the experiments are shown with ambiguity rates of 0.1, 0.2 and 0.3.

Table 10.4 shows the results obtained in this experiment. As a general analysis, the results are worse than in the previous case, since the complexity of the task is higher, but the best values obtained are still quite competitive: $64\ \%$ of the sequences were perfectly recognized with the first hypothesis; on average, only 1.10 of editing operations are necessary, with only 0.54 corrections considering an interactive case. These figures demonstrate that the use of an interactive approach is very profitable in this scenario, unlike what happened in the previous one.

Once again, the ambiguity rate seems to be the most important factor for the recognition (Fig.10.7 shows the average tendency with respect to this factor). Since a null ambiguity rate is not possible, it is still unknown the best case within the user-independent scenario. However, results with $\alpha = 0.1$ are at the same level than those of $\alpha = 0.2$ in the user-dependent case. This lead us to believe that the user-independent scenario would not be very far from the user-dependent scenario if an optimal stroke labelling could be performed.

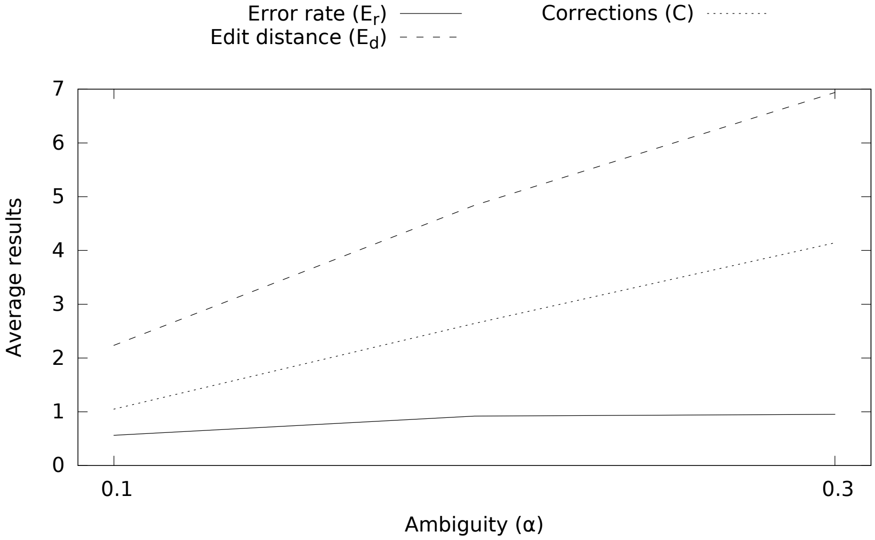


Figure 10.7: Impact of the stroke labelling ambiguity over the results achieved. Average results among all decoding strategies and probability density estimators are showed with respect to the ambiguity rate allowed.

On the other hand, probability estimation has a higher impact in this scenario. For instance, Parzen estimation using DTW achieves pretty poor results in all cases considered. Depending on each particular case, NN using FCC or Parzen using NS are reported as the best configurations.

The decoding algorithm depicts less relevance on the results, as happened in the previous scenario. Nevertheless, MPS generally achieves slightly better results for E_r and E_d, whereas MNC does for C.

Table 10.4: Mean results of the user-independent experiment with 1000 random sequences of length 17.1 (on average). Several allowable ambiguity rates (α) for the automatic stroke labelling are considered.

Prob.	Dissim.	Decod.	$\alpha = 0.1$			$\alpha = 0.2$			$\alpha = 0.3$		
			E_r	E_d	C	E_r	E_d	C	E_r	E_d	C
NN	FCC	MPP	0.39	1.52	0.59	0.94	4.70	2.37	0.94	6.81	4.11
		MPS	**0.36**	**1.10**	0.56	0.85	4.10	2.23	0.92	6.50	3.55
		MNC	**0.36**	1.13	**0.54**	0.89	4.32	2.20	0.93	6.60	3.55
	DTW	MPP	0.59	2.04	1.07	0.94	4.78	2.76	0.98	7.43	4.64
		MPS	0.54	1.90	0.90	0.90	4.58	2.46	0.95	7.04	4.12
		MNC	0.56	2.00	0.90	0.91	4.58	2.46	0.97	7.11	4.07
	NS	MPP	0.57	2.18	0.91	0.90	4.73	2.58	0.96	7.06	4.32
		MPS	0.55	2.13	0.88	0.94	4.91	2.50	0.94	6.60	4.07
		MNC	0.56	2.13	0.88	0.94	4.91	2.51	0.95	6.68	4.04
	OSP	MPP	0.54	2.18	0.93	0.92	4.29	2.40	0.95	6.57	3.86
		MPS	0.54	2.26	0.91	0.90	4.11	2.18	0.93	6.59	3.73
		MNC	0.54	2.28	0.90	0.91	4.16	2.17	0.93	6.59	3.73
Parzen	FCC	MPP	0.68	2.90	1.31	0.92	5.13	2.83	0.97	6.94	4.45
		MPS	0.48	1.75	0.85	0.91	4.31	2.32	0.95	**6.37**	**3.62**
		MNC	0.50	1.76	0.84	0.91	4.36	2.31	0.95	**6.37**	**3.62**
	DTW	MPP	0.90	4.75	2.62	1.00	6.73	4.13	0.98	7.86	5.68
		MPS	0.79	3.53	1.71	0.97	6.05	3.39	0.98	7.73	4.78
		MNC	0.79	3.54	1.69	0.97	6.05	3.38	0.98	7.73	4.74
	NS	MPP	0.45	1.57	0.73	0.91	4.93	2.74	0.94	6.95	4.20
		MPS	0.37	1.16	0.62	**0.83**	**3.90**	2.33	0.94	6.60	3.64
		MNC	0.37	1.19	0.60	0.85	4.39	**2.32**	0.94	6.60	3.64
	OSP	MPP	0.76	3.58	1.71	0.96	5.89	3.33	0.98	7.67	5.02
		MPS	0.66	2.53	1.30	0.96	5.10	2.76	0.96	7.07	4.11
		MNC	0.67	2.53	1.23	0.96	5.16	2.75	0.96	7.07	4.11

10.5 Conclusions

This work presented a novel approach to the recognition of pen-based music compositions using stroke similarity algorithms and finite-state machines. Our approach is able to learn writing styles from data and, therefore, notation is not restricted to predefined rules or gestures.

The series of strokes received as input is mapped onto stroke primitive probabilities with similarity-based probability density estimators. These probabilities are combined with a set of learned rules describing musical symbols in terms of these primitives. Then, a probabilistic model is obtained, which is eventually crossed with a formal language to avoid those sequences that do not make musical sense. A decoding algorithm is finally applied to produce a hypothesis as solution to the task.

A comprehensive experimentation has been carried out in which several metrics of interest have been evaluated considering a number of probability density estimators, stroke similarity functions and decoding strategies. Two main scenarios were considered: user-dependent and user-independent.

As expected, the user-dependent experiment showed better recognition results. Its best results yielded that only 10 % of the hypotheses were wrong, whereas the other only needed few corrections (below 0.3, on average). However, the user-independent scenario also showed competitive results, obtaining only 36 % of erroneous hypotheses, needing around 1 correction (0.5 in the interactive case), on average, otherwise.

It was found in both scenarios that the accuracy of the recognition was closely related to the degree of allowed ambiguity in the automatic stroke labelling process. An option to be considered is to improve this process. In fact, the method used was unable to get a non-ambiguous stroke labelling in the user-independent scenario and, therefore, there is still room for improvement. The dissimilarity measure utilized, to a lesser extent, also proved to be an important parameter to consider. In this regard, NN estimation using FCC dissimilarity was reported as the best choice in a broad sense.

As future work the interest is on the development of an interactive system, in which user corrections are used to continuously improve the performance of the system. The main concern is to provide a transparent and user-friendly way to receive feedback while the user is using the system, and how this feedback can be efficiently exploited. A final user-end application that takes advantage of the research explained here is also to be considered.

Chapter 11

Pen-based Multimodal Interaction with Music Notation

Abstract

The need of digitizing early music sources is demanding the development of new ways of dealing with music notation. This paper describes a work carried out under the development of a project focused on the automatic transcription of early music manuscripts. Assuming that current technologies cannot guarantee a perfect transcription, our intention is to develop an interactive system in which user and software collaborate to complete the task. Since conventional score post-editing might be tedious, the user is allowed to interact using an electronic pen. Although this provides a more ergonomic interface, this interaction must be decoded as well. In our framework, the user traces the symbols using the electronic pen over a digital surface, which provides both the underlying image (offline data) and the drawing made by the e-pen (online data) of each symbol to improve classification. Applying this methodology over 70 scores of the target musical archive, a dataset of 10230 samples of 30 different symbols was obtained and made available for research purposes. Classification over this data is presented, in which symbols are recognized by using the two modalities extracted from the sources. The combination of modes has demonstrated its good performance, decreasing the error rate of using each modality separately and achieving an almost error-free performance.

11.1 Introduction

Music constitutes one of the main tools for cultural transmission. That is why musical documents have been preserved over the centuries, scattered through cathedrals, museums, or historical archives. In an effort to prevent their deterioration, the access to these sources is not always possible. This implies that an important part of this historical heritage remains inaccessible for musicological study. Occasionally, these documents are transcribed to a digital format for an easier access, distribution and study, without compromising their integrity.

On the other hand, it is important to point out that the massive digitization of music documents also opens several opportunities to apply Music Information Retrieval algorithms, which may be of great interest. Since the handmade transcription of these sources is a long, tedious task, the development of automatic transcription systems for early music documents is gaining importance over the last few years.

Optical Music Recognition (OMR) is a field devoted to providing computers the ability to extract the musical content of a score from the optical scanning of its source. The output of an OMR system is the music score encoded in some structured digital format such as MusicXML, MIDI or MEI. Typically, the transcription of early music documents is treated differently with respect to conventional OMR methods due to specific features (for instance, the different notation or the quality of the sheet). Although there exist several works focused on early music documents transcription (Pinto et al., 2003; Pugin, 2006), the specificity of each type of notation or writing makes it difficult to generalize these developments. This is especially harmful to the evolution of the field because it is necessary to implement new processing techniques for each type of archive. Even worse, new labelled data are also needed to develop techniques for automatic recognition, which might imply a significant cost.

Notwithstanding the efforts devoted to improving these systems, their performance is far from being optimal (Rebelo et al., 2012). In fact, assuming that a totally accurate automatic transcription is not possible, and might never be, user-centred recognition is becoming an emergent framework. Instead of a fully-automatized process, *computer-aided* systems are being considered, with which the user collaborates actively to complete the recognition task (Toselli et al., 2010).

The goal of this kind of systems is to facilitate the task for the user, since it is considered the most valuable resource (Andrés-Ferrer et al., 2011). In the case of the transcription of early music documents, the potential user is the expert musicologist who understands the meaning of any nuance that appears in the sheet. However, very often these users find the use of a pen more natural and comfortable than keyboard entry or drag-and-drop actions with the mouse. Using a tablet device and e-pen, it is possible to develop an ergonomic interface to receive feedback from users' drawings. This is specially true for score post-edition where the user, instead of sequentially inputting symbols has to correct some of them, and for that, direct manipulation is the preferred interaction style.

Such an interface could be used to amend errors made by the system in a simpler way for the user, as has been proposed for automatic text recognition (Alabau et al., 2014). However, there are studies showing that, when the task is too complex, users prefer to complete the task by itself because the human-machine interaction is not friendly enough (Romero and Sanchez, 2013). Therefore, this interface could also be used to develop a manual transcription system that would be more convenient and intuitive than conventional score editors. Moreover, this transcription system might be useful in early stages of an OMR development, as it could be used to acquire training data more efficiently and ergonomically, which is specially interesting for old music notations.

Unfortunately, although the user is provided with a more friendly interface to interact with the system, the feedback input is not deterministic this way. Unlike the keyboard or mouse entry, for which it is clear what the user is inputting, the pen-based interaction has to be decoded and this process might have errors.

For all reasons above, this article presents our research on the capabilities of musical notation recognition with a system whose input is a pen-based interface. To this end, we shall assume a framework in which the user traces symbols on the score, regardless of the reason of this interaction (OMR error correction, digitizing the content, acquire labelled data, etc.). As a result, the system receives a multimodal signal: on one hand, the sequence of points that indicates the path followed by the e-pen on the digital surface, usually referred to as *online* modality; on the other hand, the portion of image below the drawn containing the original symbol, which represents the *offline* modality. One of the main hypothesis of this study is that the combination of both modalities leads to better results than using just either the pen data or the symbol image.

The rest of the chapter is structured as follows: Section 11.2 introduces the corpora collected and utilized, which comprises data of Spanish early music written in White Mensural notation; Section 11.3 describes a multimodal classifier that exploits both offline and online data; Section 11.4 presents the results obtained with such classification; and Section 11.5 concludes the present work.

11.2 Multimodal data collection

This work is a first seed of a case study to digitize a historical musical archive of early Spanish music. The final objective of the whole project is to encode the musical content of a huge archive of manuscripts dated between centuries 16th to 18th, handwritten in mensural notation, in the variant of the Spanish notation at that time (Ezquerro, 2001). A short sample of a piece from this kind of document is illustrated in Fig. 11.1.

This section describes the process developed to collect multimodal data of isolated musical symbol from images of scores. A massive collection of data will allow us to develop a more effective classification system and to go deeper into the analysis of this kind of interaction. Let us note that the important point in our interactive system is to better understand user actions. While a machine is assumed to make some mistakes, it is unacceptable to force the user to draw the same symbol of score many times. To this end, our intention is to exploit both offline data (image) and online data (e-pen user tracing) received.

Our idea is to simulate the same scenario of a real application. Therefore, we loaded the images of the scores on a digital surface to make users trace the symbols using the electronic pen. The natural symbol isolation of this kind of input

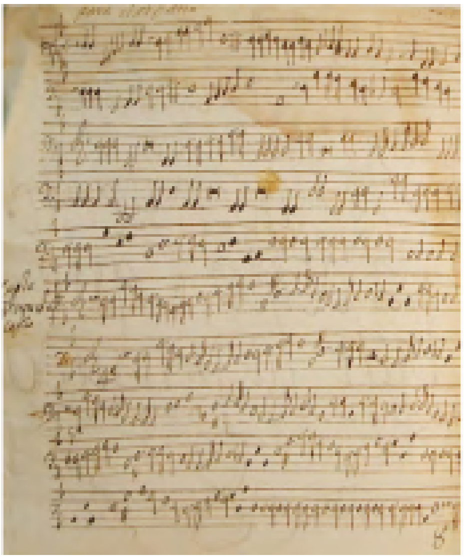


Figure 11.1: Example of page of a music book written in handwritten white mensural notation from Spanish manuscripts of centuries 16th to 18th.

is the set of strokes —data collected between pen-down and pen-up actions. To allow tracing symbols with several strokes, a fixed elapsed time is used to detect when a symbol has been completed. If a new stroke starts before this time lapse, it is considered to belong to the same symbol than the previous one.

Once online data is collected and grouped into symbol classes, the offline data is also extracted from this information. A bounding box is obtained from each group of strokes belonging to the same symbol, storing the maximum and minimum values of each coordinate (plus a small margin) among all the trace points collected. This bounding box indicates where the traced symbol can be found in the image. Therefore, with the sole effort of the tracing process, both online and offline data are collected. Note that the extraction of the offline data is driven by the tracing process, instead of deciding at every moment the bounds of each symbol.

Figure fig:tracing illustrates the process explained above for a single symbol. Although the online data is drawn in this example, the actual information stored is the sequence of 2D points in the same order they were collected, indicating the path followed by the e-pen.

Following this approach, several advantages are found: the final effort of collecting multimodal data is halved, since the online data collection simultaneously provides the offline data collection; the collected data mimics the scenario that might be found in the final application, when the user interacts with the machine; and the process becomes more user-friendly, which usually leads to a lower number of errors.

The collection was extracted by five different users from 70 different musical scores of different styles from the Spanish white mensural notation of 16th-18th

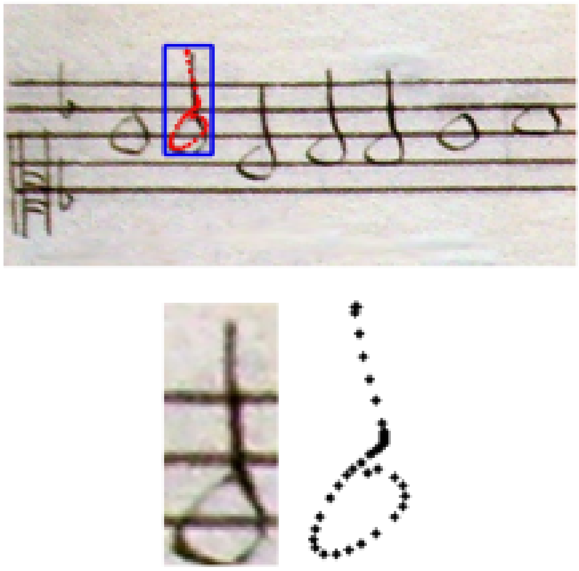


Figure 11.2: Example of extraction of a *minima*. Above, the sequence of points collected by the e-pen. The box represents the bounding box of the sequence. Below, the multimodal data extracted from the same sample.

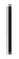
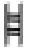
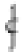
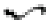
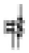
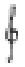
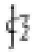
centuries. The *Samsung Galaxy Note Pro 12.2* device (247 ppi resolution) was used and symbols were written by means of the stylus *S-Pen*. All the score images used are in the same scale, in which staff lines spacing is about 24 DP.[1] Due to the irregular conditions of the documents, this value is approximate but it can be used for normalizing with respect to other scores.

The obtained dataset consists of 10230 samples, each of which contains both a piece of image and the strokes followed during its tracing. These samples are spread over 30 classes. Table 11.1 lists the set of labels, including a typographic example and the number of samples per each. The number of symbols of each class is not balanced but it depicts the same distribution found in the documents.

Every symbol that must be differentiated for preservation purposes was considered as a different class. For instance, there are two *f-clef* types because the graphical symbol is quite different despite having an equal musical meaning. However, the orientation of the symbols does not make a different class since the same graphical representation with a vertical inversion can be found. In the case it was needed, the orientation could be obtained through an easy post-processing step.

[1] DP stands for *device independent pixels* in (Android) mobile application development

Table 11.1: Details of the dataset obtained through the tracing process over 70 scores (images from *Capitán* font).

Label	Symbol	#	Label	Symbol	#
barline		46	brevis		210
coloured brevis		28	brevis rest		171
c-clef		169	common time		29
cut time		56	dot		817
double barline		73	custos		285
f-clef 1		52	f-clef 2		43
fermata		75	flat		274
g-clef		174	beam		85
longa		30	longa rest		211
minima		2695	coloured minima		1578
minima rest		427	semibrevis		1109
coloured semibrevis		262	semibrevis rest		246
semiminima		328	coloured semiminima		403
semiminima rest		131	sharp		170
proportio maior		25	proportio minor		28

11.3 Multimodal classification

This section provides a classification experiment over the data described previously. Two independent classifiers are proposed that exploit each of the modalities presented by the data. Eventually, a late-fusion classifier that combines the two previous ones will be considered.

Taking into account the features of our case of study, an instance-based classifier was considered. Specifically, the Nearest Neighbour (NN) rule was used, as it is one of the most common and effective algorithms of this kind (Cover and Hart, 1967). The choice is justified by the fact that it is specially suitable for interactive scenarios like the one found in our task: it is naturally adaptive, as the simple addition of new prototypes to the training set is sufficient (no retraining is needed) for incremental learning from user feedback. In addition, the size of the dataset can be controlled by distance-based prototype reduction algorithms (García et al., 2015) .

Decisions given by NN classifiers can be mapped onto probabilities, which are needed for the late fusion classifiers. Let $\mathcal{X}$ be the input space, in which a pairwise distance $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is defined. Let $\mathcal{Y}$ be the set of labels considered in the classification task. Finally, let T denote the training set of labelled samples $\{(x_i, y_i) : x_i \in \mathcal{X}, y_i \in \mathcal{Y}\}_{i=1}^{|T|}$.

Let us now assume that we want to know the posterior probability of each class $y \in \mathcal{Y}$ for the input point $x \in \mathcal{X}$ ($P(y|x)$) following the NN rule. A common estimation makes use of the following equations (Duda et al., 2001):

$$p(y|x) = \frac{1}{\min_{(x',y') \in T : y'=y} d(x,x') + \epsilon} \tag{11.1}$$

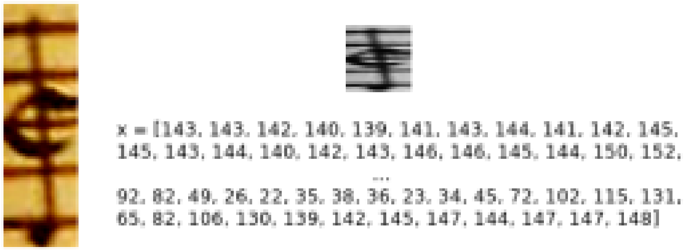
$$P(y|x) = \frac{p(y|x)}{\sum_{y' \in \mathcal{Y}} p(y'|x)}, \tag{11.2}$$

where ϵ is a contemptible value used to avoid infinity calculations. That is, the probability of each class is defined as the inverse of the distance to the nearest sample of that class in the training set. Note that the second term is used to ensure that the sum over the probability of each class is 1. Finally, the decision $\hat{y}$ of the classifier for an input x is given by a *maximum a posteriori* criterion:

$$\hat{y} = \arg\max_{y} P(y|x) \tag{11.3}$$

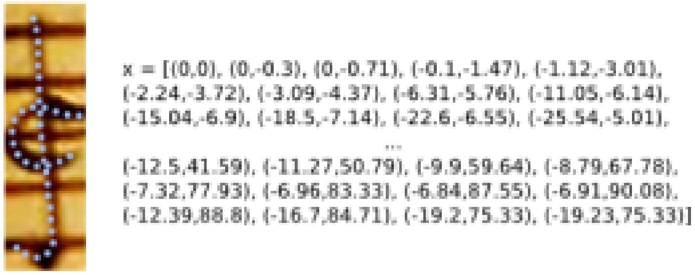
11.3.1 Offline classifier

The offline classifier takes the image of a symbol as input. To simplify the data, the images are converted to greyscale. Then, since they can be of different sizes, a fixed resizing process is performed, in the same way that can be found in other works, like that of Rebelo et al. (Rebelo et al., 2010). At the end, each image is represented by a integer-valued feature vector of equal length that stores the



```
x = [143, 143, 142, 140, 139, 141, 143, 144, 141, 142, 145,
145, 143, 144, 140, 142, 143, 146, 146, 145, 144, 150, 152,
...
92, 82, 49, 26, 22, 35, 38, 36, 23, 34, 45, 72, 102, 115, 131,
65, 82, 106, 130, 139, 142, 145, 147, 144, 147, 147, 148]
```

Figure 11.3: Offline modality of a *cut time* symbol for classification: feature vector containing the greyscale value of each position of the rescaled image.



```
x = [(0,0), (0,-0.3), (0,-0.71), (-0.1,-1.47), (-1.12,-3.01),
(-2.24,-3.72), (-3.09,-4.37), (-6.31,-5.76), (-11.05,-6.14),
(-15.04,-6.9), (-18.5,-7.14), (-22.6,-6.55), (-25.54,-5.01),
...
(-12.5,41.59), (-11.27,50.79), (-9.9,59.64), (-8.79,67.78),
(-7.32,77.93), (-6.96,83.33), (-6.84,87.55), (-6.91,90.08),
(-12.39,88.8), (-16.7,84.71), (-19.2,75.33), (-19.23,75.33)]
```

Figure 11.4: Online modality of a *cut time* symbol for classification: sequence of coordinates indicating the path followed by the e-pen during the tracing process.

greyscale value of each pixel (see Fig. 11.3). Over this data, Euclidean distance can be used for the NN classifier. A preliminary experimentation fixed the size of the images to 30×30 (900 features), although the values within the configurations considered did not vary considerably.

11.3.2 Online classifier

In the online modality, the input is a series of 2D points that indicates the path followed by the pen (see Fig. 11.4). It takes advantage of the local information, expecting that a particular symbol follows similar paths. The information contained in this modality provides a new perspective on the recognition and it does not overlap with the nature of the offline recognition.

The digital surface collects the strokes at a fixed sampling rate so that each one may contain a variable number of points. However, several distance functions can be applied to this kind of data. Those considered in this work are the following:

- Dynamic Time Warping (DTW) (Sakoe and Chiba, 1990): a technique for measuring the dissimilarity between two time signals which may be of different duration.

- Edit Distance with Freeman Chain Code (FCC): the sequence of points representing a stroke is converted into a string using a codification based on Freeman Chain Code (Freeman, 1961). Then, a Edit Distance (Levenshtein, 1966) can be applied to measure distance.

- Edit Distance for Ordered Set of Points (OSP) (Rico-Juan and Iñesta, 2006): an extension of the Edit Distance for its use over ordered sequences of points, such those collected by the e-pen.

11.3.3 Late-fusion classifier

A straightforward late fusion has been used here. The idea is to combine linearly the decisions taken by the two base classifiers. That is, probabilities of individual classifiers are combined by a weighted average:

$$P_{\text{fusion}}(y|x) = \alpha \cdot P_{\text{on}}(y|x) + (1 - \alpha) \cdot P_{\text{off}}(y|x) \tag{11.4}$$

where P_{off} and P_{on} denote the probabilities obtained by offline and online classifiers, respectively. A parameter $\alpha \in [0, 1]$ is established to tune the relevance given to each modality. We will consider several values of α ranging from 0 to 1 during experimentation.

11.4 Experimentation

Experimentation followed a 10-fold cross-validation scheme. The independent folds were randomly created with the sole constraint of having the same number of samples per class (where possible) in each of them. All the dissimilarities described in the previous section for the online classifier will be tested.

Table 11.2 illustrates the error rate (%) achieved with respect to α for this experiment. Note that $\alpha = 0$ column yields the results of the offline classifier as well as $\alpha = 1$ is equal to the online classifier. A summary of the average results is also illustrated in Fig 11.5.

An initial remark to begin with is that the worst results of the late-fusion classifiers are achieved when each is modality is used separately, with an average error of 11.77 for the offline modality and of 11.35, 9.38 and 5.26 for DTW, FCC and OSP, respectively. Not surprisingly, best results are those that combine both natures of the data, satisfying the hypothesis that two signals are better than one.

Results also report that the tuning of α is indeed relevant since it makes the error vary noticeably. An interesting point to mention is that, although the online modality is more accurate than the offline one by itself, the best tuning in each configuration always gives more importance to the latter. This might be caused by the lower variability in the writing style of the original scribes.

The best results, on average, are reported by the late-fusion classifier considering OSP distance for the online modality, with an $\alpha = 0.2$. In such case, just 2 % of error rate is obtained, which means that the interaction is well understood by the system in most of the cases. Note that a more comprehensive search of the best α may lead to a better performance (for instance, in the range $(0.2, 0.4)$), but the improvement is not expected to be significant.

Table 11.2: Error rate (average ± std. deviation) obtained for a 10-fold cross validation experiment with respect to the value used for tuning the weight given to each modality (α) and the distances for the online modality (DTW, FCC and OSP). Bold values represent the best average result for each configuration considered.

α	DTW	FCC	OSP
0.0	11.8 ± 1.5	11.8 ± 1.5	11.8 ± 1.5
0.1	5.3 ± 0.4	5.5 ± 1.1	4.9 ± 0.7
0.2	**4.7** ± **0.4**	4.1 ± 1.0	3.0 ± 0.2
0.3	5.4 ± 0.4	**3.9** ± **0.8**	2.2 ± 0.4
0.4	6.4 ± 0.3	4.1 ± 0.7	**2.0** ± **0.4**
0.5	7.4 ± 0.5	4.6 ± 0.7	2.2 ± 0.5
0.6	8.2 ± 0.6	5.2 ± 0.9	2.5 ± 0.5
0.7	9.1 ± 0.5	5.9 ± 1.0	3.0 ± 0.6
0.8	9.8 ± 0.5	6.6 ± 0.9	3.4 ± 0.6
0.9	10.5 ± 0.8	7.3 ± 0.9	4.2 ± 0.5
1.0	11.3 ± 0.8	9.3 ± 0.7	5.2 ± 0.5

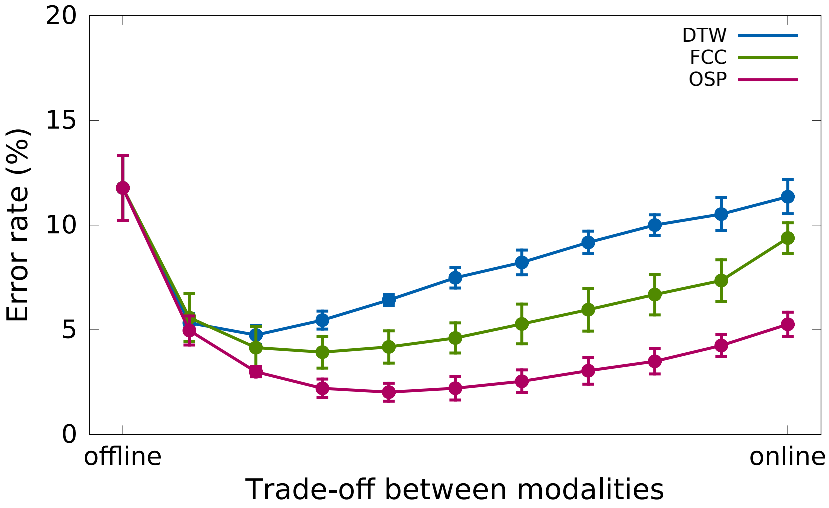


Figure 11.5: Average results with respect to the weight (α) given to each modality for the configurations considered, from *offline* ($\alpha = 0$) to *online* ($\alpha = 1$).

Although the results report a fair accuracy, the use of semantic music models is expected to avoid some of these mistakes by using contextual information. Therefore, a nearly optimal performance could be obtained during the interaction with the user.

11.5 Conclusions

This paper presents a new approach to interact with musical notation, based on the use of an electronic pen. Our framework assumes that the user traces each musical symbol of the score, and the system receives a *multimodal* input accordingly: the sequence of coordinates indicating the trajectory of the e-pen (online mode) and the underlying image of the score itself (offline mode).

This framework has been applied to a music archive of Spanish music from the 16th to 18th centuries, handwritten in white mensural, with the objective of obtaining data for our experiments. The result of processing this collection has been described and made available for research purposes.

Experimentation with this dataset is presented, considering several classifiers. The overall analysis of this experiments is that it is worth to consider both modalities in the classification process, as accuracy is noticeably improved with a combination of them than that achieved by each separately.

As a future line of work, the reported analysis will be used to build a whole *computer-aided* system, in which the user interacts with the system by means of an electronic pen to digitize music content. Since the late-fusion classifier is close to its optimal performance, it seems to be more interesting to consider the development of semantic models that can amend misclassifications by using contextual information (*e.g.*, a score starts with a clef). In addition, further effort is to be devoted to visualization and the user interface.

Part IV

Conclusion

Chapter 12

Concluding remarks

This chapter concludes the dissertation. It includes a summary of the research carried out, a brief discussion of the general contributions and some of the main opened lines of future work.

12.1 Summary

This thesis aims at proposing contributions to the automatic recognition of music notation based on a Pattern Recognition perspective. The research focuses its efforts on specific aspects that dovetail with this overall objective.

Chapters 3, 4 and 5 deal with general aspects of the OMR field. In the first one, an new approach for old printed music is proposed. The distinctive feature of this system is that it avoids the staff lines removal. Results show that, at least for some kind of notation, it is advisable to address the problem in this way since many errors in both extraction and classification are prevented. Chapter 4 continues with aspects related to staff lines. It is proposed to solve that step as a classification problem at pixel level. Experimentation reports that this approach is competitive with respect to state-of-the-art strategies, including some additional advantages related to supervised classification. Moreover, Chapter 5 proposes a new algorithm to classify symbols based on the NN rule. It combines ideas of ensemble classifiers and dissimilarity spaces to improve the accuracy.

Chapters 6, 7, 10 and 11 develop the human-machine interaction for music notation by means of e-pen technologies. Those chapters deal with both the automatic recognition of isolated handwritten symbols as well as the use of such an interface to interact with an OMR system or to develop a system for pen-based score creation.

Finally, Chapters 8 and 9 address improvements to the efficiency of classification based on the NN rule. This classifier is a good choice from the point of view of an interactive scenario. In turn, it leads to a high computational complexity. Therefore, the use of strategies that make use of Prototype Reduction algorithms is proposed to alleviate this situation. Specifically, Chapter 8 proposes a new heuristic to combine the efficiency of using Prototype Selection algorithms with the accuracy of using the full available data. Chapter 9, however, proposes a way to use Prototype Generation algorithms in tasks for which the input is not represented as a features vector. Both studies have demonstrated significant improvements treating the database presented in Chapter 6. Nevertheless, in order to reach a wider audience, they also include experimentation with several well-known datasets that allows generalising these contributions to other classification tasks.

12.2 Discussion

Given that the substantial part of the dissertation consists of a compilation of papers, the main discussions of each contribution can be found in their corresponding chapter. From a global point of view, however, this thesis has shown that it is profitable to develop those aspects of OMR in which Pattern Recognition can make significant contributions.

As depicted in the diversity of issues addressed in the publications, it can be seen that the thesis has been flexible in its main line, incorporating new ideas emerged in the course of the research on the automatic recognition focused on musical notation. The research covers different parts of the process such as the removal of the staff lines, new approaches to interact with the system and improvements in the classification of symbols, both in accuracy and efficiency.

It can be concluded that the research conducted involves an interesting research for the scientific community, as demonstrated by publications in high-impact journals and conferences, backed by peer-review committees. As evidence of quality, it can be pointed out that 5 of these works are published in journals indexed in the *Journal Citation Reports*, located within the first quartiles, whereas the rest are published in relevant international conferences. Additionally, two works have been described that are still to be considered for publication on the day of the submission of this dissertation.

On the other hand, this series of publications not only demonstrate that the research carried out is able to depict a relevant contribution in the field of study but also proves that the knowledge needed for research dissemination has been acquired.

12.3 Future work

The research lines started in this thesis can not be considered as completely finished. On the contrary, the conducted research has opened new avenues that are interesting to consider in the near future:

1. The approach to build OMR systems that avoid the removal of staff lines must be considered to analyse other types of scores. It remains to evaluate

whether this strategy can definitely be established as a new alternative for the construction of these systems or reduced only to those who have a sheet style like the one considered in this thesis. Moreover, one issue to consider is that segmentation of symbols also follows a supervised learning approach, instead of using heuristics based on unsupervised learning. For example, a classifier could learn to discriminate which parts of the staff represent the bound of a music symbol.

2. It has been shown that removing staff lines can be approached as a supervised classification task. In this regard, it would be interesting to generalize this process so that it can be used with greyscale images, thereby avoiding the problems produced by a wrong binarization. In addition, more research should be devoted to overcoming the problem of getting enough data to train these classifiers when a new type of sheet is received.

3. The recognition of pen-based music notation had been little explored so far. The work done in this thesis represents an important seed but there is still room for further research in that direction. The addition of pen-based interaction in the workflow of a fully functional OMR system is still to be studied. It would be interesting to see how the system itself and the user collaborate to complete the task with minimal effort. In addition, the interaction should not only correct errors produced but help the system to dynamically modify its behaviour.

4. So far, most of OMR systems have followed a conventional pipeline based on segmentation and classification steps. As future work, the performance of holistic Pattern Recognition should be considered for this task. For instance, considering the use of models such as Hidden Markov Models or Recurrent Neural Networks, which are giving good results in the automatic recognition of handwritten text.

5. In recent years, deep neural networks have been a remarkable leap in the ability to learn intrinsic representations of input data. Specifically, Convolutional Neural Networks have shown a great ability in classification tasks involving images (Ciresan et al., 2012). Therefore, it would be interesting to consider this new paradigm to improve the results obtained in the OMR field.

It should be noted that depending on the degree of depth pursued, each of these lines can be seen as either a sequel of the work presented in this dissertation or the start of a specific research project.

Bibliography

Alabau, V., Martínez-Hinarejos, C. D., Romero, V., and Lagarda, A. L. (2014). An iterative multimodal framework for the transcription of handwritten historical documents. *Pattern Recognition Letters*, 35:195–203.

Andrés-Ferrer, J., Romero, V., and Sanchis, A. (2011). *Multimodal Interactive Pattern Recognition and Applications*, chapter General Framework. Springer, 1st edition.

Anstice, J., Bell, T., Cockburn, A., and Setchell, M. (1996). The design of a pen-based musical input system. In *Sixth Australian Conference on Computer-Human Interaction*, pages 260–267.

Bainbridge, D. and Bell, T. (2001). The challenge of optical music recognition. *Computers and the Humanities*, 35(2):95–121.

Benetos, E., Dixon, S., Giannoulis, D., Kirchhoff, H., and Klapuri, A. (2013). Automatic music transcription: challenges and future directions. *Journal of Intelligent Information Systems*, 41(3):407–434.

Bunke, H. and Riesen, K. (2012). Towards the unification of structural and statistical pattern recognition. *Pattern Recognition Letters*, 33(7):811–825.

Calvo-Zaragoza, J., Barbancho, I., Tardón, L. J., and Barbancho, A. M. (2015a). Avoiding staff removal stage in optical music recognition: application to scores written in white mensural notation. *Pattern Analysis and Applications*, 18(4):933–943.

Calvo-Zaragoza, J., Micó, L., and Oncina, J. (2016a). Music staff removal with supervised pixel classification. *International Journal on Document Analysis and Recognition*, Online:1–9.

Calvo-Zaragoza, J. and Oncina, J. (2014). Recognition of pen-based music notation: The HOMUS dataset. In *22nd International Conference on Pattern Recognition, ICPR 2014, Stockholm, Sweden, August 24-28, 2014*, pages 3038–3043.

Calvo-Zaragoza, J. and Oncina, J. (2015). Clustering of strokes from pen-based music notation: An experimental study. In *7th Iberian Conference Pattern Recognition and Image Analysis, IbPRIA 2015, Santiago de Compostela, Spain, June 17-19, 2015, Proceedings*, pages 633–640.

Calvo-Zaragoza, J., Valero-Mas, J. J., and Rico-Juan, J. R. (2015b). Improving kNN multi-label classification in Prototype Selection scenarios using class proposals. *Pattern Recognition*, 48(5):1608–1622.

Calvo-Zaragoza, J., Valero-Mas, J. J., and Rico-Juan, J. R. (2016b). Prototype Generation on Structural Data using Dissimilarity Space Representation. *Neural Computing and Applications*, Online:1–10.

Casacuberta, F. and de la Higuera, C. (2000). Computational Complexity of Problems on Probabilistic Grammars and Transducers. In *Proceedings of the 5th International Colloquium on Grammatical Inference: Algorithms and Applications*, ICGI 2000, pages 15–24, London, UK, UK. Springer-Verlag.

Ciresan, D., Meier, U., and Schmidhuber, J. (2012). Multi-column deep neural networks for image classification. In *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3642–3649. IEEE.

Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27.

Dalitz, C., Droettboom, M., Pranzas, B., and Fujinaga, I. (2008). A comparative study of staff removal algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5):753–766.

de la Higuera, C. and Oncina, J. (2014). Computing the Most Probable String with a Probabilistic Finite State Machine. In *11th International Conference on Finite-State Methods and Natural Language Processing*, pages 1–8.

Duda, R. O., Hart, P. E., and Stork, D. G. (2001). *Pattern Classification*. John Wiley & Sons, New York, NY, 2 edition.

Ezquerro, A., editor (2001). *Música de la Catedral de Barcelona a la Biblioteca de Catalunya*. Biblioteca de Catalunya, Barcelona.

Freeman, H. (1961). On the encoding of arbitrary geometric configurations. *IRE Transactions on Electronic Computers*, EC-10(2):260–268.

Gales, M. and Young, S. (2008). The application of hidden markov models in speech recognition. *Foundations and trends in signal processing*, 1(3):195–304.

Garcia, S., Derrac, J., Cano, J., and Herrera, F. (2012). Prototype Selection for Nearest Neighbor Classification: Taxonomy and Empirical Study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):417–435.

García, S., Luengo, J., and Herrera, F. (2015). *Data Preprocessing in Data Mining*, volume 72 of *Intelligent Systems Reference Library*. Springer.

George, S. E. (2003). Online pen-based recognition of music notation with artificial neural networks. *Computer Music Journal*, 27(2):70–79.

Graves, A., Mohamed, A.-R., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6645–6649.

Graves, A. and Schmidhuber, J. (2009). Offline handwriting recognition with multidimensional recurrent neural networks. In *Advances in neural information processing systems*, pages 545–552.

Hartmann, B. and Link, N. (2010). Gesture recognition with inertial sensors and optimized DTW prototypes. In *IEEE International Conference on Systems Man and Cybernetics (SMC)*, pages 2102–2109.

Horvitz, E. (1999). Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 159–166. ACM.

Hsu, C.-W. and Lin, C.-J. (2002). A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425.

Kim, J. and Sin, B.-K. (2014). Online handwriting recognition. In Doermann, D. and Tombre, K., editors, *Handbook of Document Image Processing and Recognition*, pages 887–915. Springer London.

Lee, K. C., Phon-Amnuaisuk, S., and Ting, C.-Y. (2010). Handwritten music notation recognition using HMM – a non-gestural approach. In *International Conference on Information Retrieval Knowledge Management, (CAMP), 2010*, pages 255–259.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. Technical Report 8.

Liu, C., Yin, F., Wang, D., and Wang, Q. (2013). Online and offline handwritten chinese character recognition: Benchmarking on new databases. *Pattern Recognition*, 46(1):155–162.

Lundvall, B. (2010). *National systems of innovation: Toward a theory of innovation and interactive learning*, volume 2. Anthem Press.

Macé, S., Éric Anquetil, and Couasnon, B. (2005). A generic method to design pen-based systems for structured document composition: Development of a musical score editor. In *First Workshop on Improving and Assesing Pen-Based Input Techniques*, pages 15–22, Edinghburg.

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill, Inc.

Miyao, H. and Maruyama, M. (2004). An online handwritten music score recognition system. In *Proceedings of the 17th International Conference on Pattern Recognition*, volume 1, pages 461–464.

Miyao, H. and Maruyama, M. (2007). An online handwritten music symbol recognition system. *International Journal of Document Analysis and Recognition (IJDAR)*, 9(1):49–58.

Mondal, T., Bhattacharya, U., Parui, S. K., Das, K., and Roy, V. (2009). Database generation and recognition of online handwritten bangla characters. In *Proceedings of the International Workshop on Multilingual OCR*, MOCR '09, pages 9:1–9:6, New York, NY, USA. ACM.

Nanni, L. and Lumini, A. (2011). Prototype reduction techniques: A comparison among different approaches. *Expert Systems With Applications*, 38(9):11820–11828.

Ng, E., Bell, T., and Cockburn, A. (1998). Improvements to a pen-based musical input system. In *Australasian Computer Human Interaction Conference*, pages 178–185.

Oncina, J. (2009). Optimum algorithm to minimize human interactions in sequential computer assisted pattern recognition. *Pattern Recognition Letters*, 30(5):558–563.

O'Shaughnessy, D. (2000). *Automatic Speech Recognition*, pages 367–435. Wiley-IEEE Press.

Parzen, E. (1962). On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076.

Paz, A. (1971). *Introduction to probabilistic automata*. Academic Press, New York.

Pinto, J. C., Vieira, P., Ramalho, M., Mengucci, M., Pina, P., and Muge, F. (2000). Ancient music recovery for digital libraries. In *Research and Advanced Technology for Digital Libraries*, pages 24–34. Springer.

Pinto, J. R. C., Vieira, P., and da Costa Sousa, J. M. (2003). A new graph-like classification method applied to ancient handwritten musical symbols. *International Journal on Document Analysis and Recognition (IJDAR)*, 6(1):10–22.

Plamondon, R. and Srihari, S. N. (2000). On-line and off-line handwriting recognition: A comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):63–84.

Poláček, O., Sporka, A. J., and Slavík, P. (2009). Music alphabet for low-resolution touch displays. In *International Conference on Advances in Computer Enterntainment Technology*, ACE '09, pages 298–301, New York, NY, USA. ACM.

Pugin, L. (2006). Optical music recognition of early typographic prints using hidden markov models. In *ISMIR 2006, 7th International Conference on Music Information Retrieval, Victoria, Canada, 8-12 October 2006, Proceedings*, pages 53–56.

Rebelo, A., Capela, G., and Cardoso, J. S. (2010). Optical recognition of music symbols. *International Journal on Document Analysis and Recognition (IJDAR)*, 13(1):19–31.

Rebelo, A., Fujinaga, I., Paszkiewicz, F., Marcal, A., Guedes, C., and Cardoso, J. (2012). Optical music recognition: state-of-the-art and open issues. *International Journal of Multimedia Information Retrieval*, 1(3):173–190.

Rico-Juan, J. R. and Calvo-Zaragoza, J. (2015). Improving classification using a confidence matrix based on weak classifiers applied to OCR. *Neurocomputing*, 151:1354–1361.

Rico-Juan, J. R. and Iñesta, J. M. (2006). Edit distance for ordered vector sets: A case of study. In *Structural, Syntactic, and Statistical Pattern Recognition*, volume 4109 of *Lecture Notes in Computer Science*, pages 200–207. Springer Berlin Heidelberg.

Romero, V. and Sanchez, J. (2013). Human Evaluation of the Transcription Process of a Marriage License Book. In *12th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1255–1259.

Russell, S. and Norvig, P. (2009). *Artificial Intelligence: A Modern Approach*. Prentice Hall Press, Upper Saddle River, NJ, USA.

Sakoe, H. and Chiba, S. (1990). Readings in Speech Recognition. In Waibel, A. and Lee, K.-F., editors, *Readings in Speech Recognition*, chapter Dynamic Programming Algorithm Optimization for Spoken Word Recognition, pages 159–165. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Sanchis-Trilles, G., Alabau, V., Buck, C., Carl, M., Casacuberta, F., García-Martínez, M., Germann, U., González-Rubio, J., Hill, R. L., Koehn, P., et al. (2014). Interactive translation prediction versus conventional post-editing in practice: a study with the CasMaCat workbench. *Machine Translation*, 28(3-4):217–235.

Toselli, A., Vidal, E., and Casacuberta, F. (2011). *Multimodal Interactive Pattern Recognition and Applications*. Springer.

Toselli, A. H., Romero, V., Pastor, M., and Vidal, E. (2010). Multimodal interactive transcription of text images. *Pattern Recognition*, 43(5):1814–1825.

Triguero, I., Derrac, J., Garcia, S., and Herrera, F. (2012). A taxonomy and experimental study on prototype generation for nearest neighbor classification. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 42(1):86–100.

Vidal, E., Rodríguez, L., Casacuberta, F., and García-Varea, I. (2007). Interactive Pattern Recognition. In *Machine Learning for Multimodal Interaction (MLMI)*, pages 60–71.

Vidal, E., Thollard, F., de la Higuera, C., Casacuberta, F., and Carrasco, R. C. (2005). Probabilistic finite-state machines-part I. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(7):1013–1025.

Yin, F., Wang, Q.-F., Zhang, X.-Y., and Liu, C.-L. (2013). ICDAR 2013 Chinese Handwriting Recognition Competition. In *12th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1464–1470.