Ph.D. thesis

Stochastic Language Models for Music Information Retrieval

Carlos Pérez Sancho

Supervised by

José M. Iñesta
Jorge Calera Rubio

Universitat d'Alacant
Universidad de Alicante
Departament de Llenguatges i Sistemes Informàtics
Departamento de Lenguajes y Sistemas Informáticos

June 2009

A Vanessa y Javier

Agradecimientos

Quiero agradecer a todos los que han contribuido, directa o indirectamente, a que esta tesis haya sido posible. En primer lugar, quiero dar las gracias a mis dos directores. A José Manuel Iñesta, por haberme brindado la oportunidad de investigar en un área apasionante — la fusión de la informática y la música — y por orientarme en todo momento; y a Jorge Calera, por resolver todas mis dudas y sacarme de más de un atasco. A los dos, en general, por todo el apoyo, el esfuerzo y el tiempo que me han dedicado.

Quiero agradecer también a todo el Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Alicante por haberme proporcionado un ambiente de trabajo envidiable. En especial a mis compañeros del Grupo de Reconocimiento de Formas e Inteligencia Artificial, que han estado siempre dispuestos a ayudarme, y a Mikel L. Forcada por haber confiado en mí desde el principo. Gracias a Antonio, Pierre, David y Plácido, que han colaborado activamente en este trabajo y de los que he aprendido mucho en todo este tiempo, y a Rafael Ramírez y Stefan Kersten del *Music Technology Group* de la *Universitat Pompeu Fabra*, por su colaboración con el sistema de transcripción de acordes.

Gracias también a mis compañeros David y Felipe, que han compartido este camino conmigo, siempre con un gran sentido del humor. ¡Felicidades David por haber emprendido "ese otro camino"!

Quiero dar las gracias, por último, a las dos personas más importantes en mi vida. A Javier, que me alegra cada mañana con su sonrisa; y a Vanessa, por apoyarme desde el principio, por haberme animado hasta el final y por ayudarme en todo momento. Gracias por todo.

Finalmente, quiero expresar mi agradecimiento a las instituciones que han financiado parcialmente este trabajo a través de distintos proyectos: la red *PASCAL Network of Excellence* de la Unión Europea; los proyectos Consolider Ingenio 2010 (MIPRCV, CSD2007-00018), PROSEMUS (TIN2006-14932-C02) y TIRIG (TIC2003-08496-C04) del Ministerio de Educación y Ciencia, y los proyectos GV043-541 y GV06/166 de la *Generalitat Valenciana*.

Carlos Pérez Sancho
8 de junio de 2009

Resumen

Introducción

En los últimos años, diversos factores han contribuido a la popularización de las bases de datos con contenido multimedia. La gran expansión de las conexiones a internet de banda ancha, los bajos costes de los dispositivos de almacenamiento de gran capacidad o las mejoras en las tecnologías de compresión son sólo algunos de ellos. Todo esto ha provocado que tanto los consumidores como la industria hayan mostrado un interés creciente en este tipo de bibliotecas, prestando especial atención a las bibliotecas con contenidos musicales debido a su gran interés comercial.

Esta situación ha provocado un gran crecimiento de este tipo de bibliotecas, haciendo que la técnicas bibliográficas tradicionales, basadas principalmente en fichas de metadatos, sean cada vez menos apropiadas para manejar estos grandes volúmenes de información. Además, hay una demanda creciente de nuevas formas de acceso a la información digital que permitan hacer búsquedas más precisas sobre el contenido. Ante esta situación han surgido nuevos retos con respecto a la creación, gestión y acceso a las bibliotecas musicales, que requieren nuevos métodos para la generación automática de metadatos, la indexación de contenidos musicales y la creación de nuevos interfaces de acceso a la información.

Como respuesta a esta situación surgió una nueva área de investigación conocida como recuperación de información musical (*music information retrieval*, MIR). MIR es un área de investigación interdisciplinar, que agrupa áreas tan diversas como la informática, el procesamiento digital de la señal, la musicología y la ciencia cognitiva, entre otras. Esta diversidad ha permitido afrontar estos nuevos retos desde distintas perspectivas, integrando conocimiento de cada una de estas disciplinas para crear nuevos sistemas de tratamiento y acceso a la información musical. Una particularidad importante de la investigación en MIR es que está centrada en crear métodos de acceso a la información basados en contenidos, al contrario que los métodos tradicionales basados en metadatos. Esto ha permitido crear nuevas aplicaciones para la recuperación de información, organización e indexación automática, sistemas de recomendación e incluso análisis musical.

Buscar una canción en una base de datos, clasificar una nueva canción en una taxonomía existente o encontrar nueva música del estilo del usuario son tareas que requieren algoritmos capaces de calcular la similitud de un conjunto de canciones, de ahí que los conceptos de estilo y similitud musical son claves en MIR. En música, la similitud se pueden encontrar a distintos niveles, como la melodía, la armonía o el timbre, o puede referirse incluso a la forma en que la música se percibe (por ejemplo canciones románticas, tristes, etc.). Sin embargo, para construir un sistema capaz de realizar estas tareas

de la misma forma en que lo haría una persona, sería necesario conocer los mecanismos mentales que hay detrás de la percepción de la música. Aunque ya se conocen algunos de los mecanismos más básicos (Shmulevich et al., 2001; Temperley, 2006), éste es todavía un problema por resolver.

Ante esta situación, una buena alternativa para afrontar el desarrollo de estos sistemas son las técnicas de inteligencia artificial, un área de la informática que trata de imitar el comportamiento humano. El objetivo de estas técnicas es el de tomar decisiones inteligentes frente a un problema dado. De entre todas las disciplinas que abarca la inteligencia artificial, la más popular en MIR es el reconocimiento de patrones (*pattern recognition*, PR). Esta disciplina se define como "el acto de tomar datos sin tratar y tomar una decisión en función de la categoría a la que pertenecen estos datos" (Duda et al., 2000). Aplicaciones típicas de estas técnicas son la visión por computador, el reconocimiento de caracteres manuscritos, la diagnosis asistida, el reconocimiento del habla y la minería de datos.

Las técnicas de reconocimiento de patrones se dividen en dos grandes grupos: técnicas supervisadas y técnicas no supervisadas. En un esquema de clasificación supervisada, el clasificador es entrenado previamente usando un conjunto de datos etiquetado, asignando a los nuevos patrones la etiqueta de la clase a la que son más parecidos. Esta similitud se puede calcular de distintas maneras, dependiendo del método de clasificación utilizado. Algunas medidas de similitud típicas son la distancia de edición a un prototipo del conjunto de entrenamiento o la probabilidad de pertenecer a una función de densidad de probabilidad calculada a partir de las muestras de entrenamiento. En clasificación no supervisada los datos de entrenamiento no están etiquetados, y el sistema se encarga de agrupar estos datos en función de su similitud.

Las técnicas de PR se han aplicado satisfactoriamente en numerosas aplicaciones MIR, proporcionando algoritmos para la recuperación de información musical, sistemas de recomendación, clasificación de estilos musicales, composición automática, descripción y transcripción de señales de audio y análisis musical automático. Esta tesis se centra en el problema de clasificación de estilos musicales, utilizando para ello un esquema de aprendizaje supervisado.

Para poder aplicar satisfactoriamente los algoritmos de reconocimiento de patrones es necesario extraer las características adecuadas para representar los datos. En el caso de la música, las características que se pueden extraer dependen fuertemente del formato de representación de la información musical. Los formatos disponibles se pueden agrupar en dos grandes familias: datos de audio y datos simbólicos, aunque la elección del formato depende en gran medida del problema a resolver. Por ejemplo, la detección de instrumentos o de voz humana son dos problemas que requieren señales de audio, mientras que para realizar un análisis armónico de una pieza es necesario tener la información en formato simbólico. Esta

tesis utiliza una representación simbólica de la música, ya que este formato permite tener una representación precisa de la partitura de las canciones. Más concretamente, los métodos presentados en esta tesis exploran dos dimensiones de la música presentes en la partitura: la melodía (dimensión horizontal) y la armonía (dimensión vertical).

Se entiende por melodía una secuencia monofónica de notas (Pickens, 2001), es decir, una secuencia de notas donde no hay más de una nota sonando simultáneamente. La importancia de la melodía reside en que es la parte más sobresaliente de una canción, y por lo tanto la más fácil de recordar, y normalmente contiene los motivos principales. La codificación de melodías para tareas de recuperación de información musical se hace normalmente codificando las alturas y duraciones de las notas con símbolos alfanuméricos, de forma que que el resultado es una secuencia de símbolos similar a una cadena de texto, lo que permite utilizar técnicas de recuperación de información textual sobre un corpus musical (Doraisamy and Rüger, 2003; Downie, 1999).

La armonía en música se refiere al estudio de la simultaneidad de sonidos. A diferencia de la melodía, la armonía estudia la organización vertical de las notas musicales, el modo en que se perciben juntas en la música tonal. Por lo tanto, la armonía está estrechamente ligada al concepto de tonalidad. La tonalidad de una pieza musical es la escala de la que se toman la mayoría de sus notas y dicta el modo en que éstas se pueden combinar tanto vertical como horizontalmente. El principal objeto de estudio de la armonía son las progresiones de acordes, ya que en la música tonal cada acorde tiene su significado dependiendo de su composición (las notas que lo forman) y de su relación con los acordes que lo rodean. Las progresiones de acordes se codifican normalmente utilizando la notación estándar musical, obteniendo así una cadena de texto con los nombres de los acordes.

El problema de la clasificación de estilos musicales

El estilo musical es una cualidad de la música que la mayoría de la gente puede percibir de forma intuitiva. El estilo se usa para describir, clasificar e incluso comparar canciones o álbumes, a pesar de que no existe una definición precisa de qué es un estilo musical. Sin embargo, Fabbri (1999) proporcionó una definición interesante de este concepto:

> "(El estilo es) una organización recurrente de características en eventos musicales, típica de un individuo (compositor o intérprete), un grupo de músicos, un género, un lugar o un periodo temporal."

Esta definición coincide con el uso que se ha hecho normalmente del estilo en la literatura MIR, ya que un gran número de trabajos en esta

área se han centrado en distintos aspectos del estilo musical, como el género (Cruz-Alcázar and Vidal, 2008; Ponce de León and Iñesta, 2007), el compositor (Backer and van Kranenburg, 2005), la emoción (Hu et al., 2008), la estética (Manaris et al., 2005), la interpretación (Dannenberg et al., 1997), la producción (Tzanetakis et al., 2007) y el origen geográfico de canciones populares (Chai and Vercoe, 2001).

Métodos basados en la representación simbólica de la música

Los métodos de clasificación de estilos musicales basados en una representación simbólica de la música se pueden dividir en dos grupos: los que utilizan secuencias melódicas o los basados en secuencias armónicas (progresiones de acordes).

Entre los que se basan en la información contenida en secuencias melódicas se pueden distinguir, a su vez, dos grandes grupos, dependiendo de la forma en que la melodía es descrita. Por una parte se pueden encontrar trabajos que utilizan una descripción superficial de la melodía, mediante un conjunto de características estadísticas globales calculadas a partir de las notas. Por otro lado, otros trabajos se han centrado en describir las relaciones locales entre las notas de la melodía, usando la propia secuencia melódica como entrada al sistema de clasificación. Estos dos enfoques proporcionan un análisis musical a distintos niveles — global y local — y son, por lo tanto, complementarios.

Especialmente interesantes son los trabajos que han estudiado el paralelismo entre la música y el lenguaje natural, bajo la asunción de que el lenguaje musical posee una estructura similar al lenguaje natural humano. Estos trabajos utilizan técnicas de reconocimiento sintáctico de patrones, construyendo un modelo sintáctico para cada clase en el conjunto de entrenamiento, que más tarde se utiliza para analizar las nuevas melodías a clasificar (Chai and Vercoe, 2001; Cruz-Alcázar and Vidal, 2008; de la Higuera et al., 2005).

Con respecto a los métodos que utilizan información armónica, cabe señalar que han sido pocos los autores que han explotado esta fuente de información. Además, todos ellos han utilizado distintos vocabularios de acordes extraídos a su vez de distintas fuentes (ficheros de audio, MIDI o progresiones de acordes codificadas manualmente), por lo que los resultados obtenidos en estos trabajos son difíciles de comparar.

El enfoque en esta tesis

El principal objetivo de esta tesis es estudiar hasta qué punto se puede determinar el estilo de una obra musical basándose únicamente en la información contenida en su partitura. Para esto se han llevado a cabo dos tareas de clasificación usando un esquema supervisado: clasificación

de géneros musicales y el modelado del estilo de compositores. La hipótesis que hay detrás de estos experimentos es que la música, como método de comunicación humano, tiene una estructura similar al lenguaje natural, y por lo tanto puede ser estudiada usando las herramientas tradicionales del área de investigación del procesamiento del lenguaje natural. Estas herramientas se pueden usar para extraer los rasgos estilísticos característicos de un conjunto de piezas musicales pertenecientes a diversos géneros y compositores, usando varios corpus estiquetados de ficheros con información musical simbólica. Para esto se han explorado dos fuentes de información: la melodía y la armonía.

Al usar secuencias melódicas, se ha usado solamente la información proporcionada por la altura y la duración de las notas. La investigación presentada en esta tesis está especialmente influenciada por las conclusiones expuestas por Cruz-Alcázar y Vidal en (Cruz-Alcázar, 2004; Cruz-Alcázar and Vidal, 2008), donde los autores obtuvieron resultados excelentes usando modelos de n-gramas — una aproximación al modelado de lenguajes — usando dos corpus con un número reducido de géneros musicales. Se ha escogido también el modelado de lenguajes para los experimentos en esta tesis porque es una técnica que permite estudiar el paralelismo entre la música y el lenguaje natural, estudiando la habilidad de los modelos de lenguaje para predecir el estilo de nuevas canciones. Además de los modelos de n-gramas, se ha seleccionado para los experimentos otra técnica usada frecuentemente en tareas de clasificación de textos con muy buenos resultados, el clasificador *naïve Bayes*. Este método, bajo ciertas circunstancias, es equivalente a un modelo de lenguaje compuesto por unigramas.

Hasta el momento, pocos trabajos han estudiado cómo la información armónica puede ayudar a distinguir estilos musicales. En esta tesis se ha elegido la armonía tonal como característica para la clasificación de géneros musicales, ya que parece haber una fuerte conexión entre el género y el uso de distintas progresiones de acordes.

Finalmente, se ha estudiado también la posibilidad de usar estas técnicas para trabajar con ficheros de audio, usando algoritmos actuales de transcripción automática para obtener secuencias melódicas y armónicas a partir de ficheros de audio.

Clasificación de géneros musicales

En el problema de clasificación de géneros musicales se han realizado experimentos utilizando secuencias melódicas y armónicas por separado, y finalmente ambos enfoques se han combinado mediante una técnica de combinación de clasificadores. Con este propósito se ha recopilado un corpus de ficheros musicales en formato simbólico (llamado en adelante

Perez-9-genres), que contiene tanto secuencias melódicas como armónicas para todas las canciones. Este corpus contiene contiene nueve géneros musicales, organizados en una estructura jerárquica compuesta por tres estilos principales (música académica, jazz y popular), con tres subestilos para cada uno de ellos, lo que permite realizar experimentos a distintos niveles: bien utilizando una partición en tres clases para el nivel superior, o bien utilizando los nueve subgéneros, lo hace la tarea de clasificación más compleja. Además de este corpus, se han utilizado otros tres corpus (*Ponce-2-genres*, *Cruz-3-genres* y *Cruz-4-genres*) que fueron recopilados por otros autores (Cruz-Alcázar and Vidal, 2008; Ponce de León and Iñesta, 2007), lo que ha permitido comparar los resultados obtenidos en esta tesis con los obtenidos por dichos autores utilizando otras técnicas. Estos tres corpus contienen únicamente secuencias melódicas en formato MIDI.

Clasificación basada en secuencias melódicas

En estos experimentos se ha probado un sistema de clasificación de géneros musicales basado en secuencias melódicas. Este sistema está basado en una descripción local de las relaciones entre notas consecutivas en la melodía, codificando las secuencias melódicas como cadenas de texto, sobre las que se han aplicado dos métodos usados tradicionalmente en clasificación de textos: *naïve Bayes* y modelos de n-gramas. El formato de codificación escogido está basado en el propuesto por Doraisamy and Rüger (2003), y se basa en la codificación de secuencias de intervalos de alturas y proporciones de duraciones como secuencias de símbolos alfanuméricos. Se han estudiado dos versiones de este método de codificación, una de ellas que codifica los valores de altura y duración de forma acoplada (agrupando en un mismo símbolo ambos datos) y desacoplada (codificando por separado la altura y la duración). A diferencia del método de codificación utilizado en (Cruz-Alcázar and Vidal, 2008), este formato tiene la ventaja de que es capaz de codificar ficheros MIDI reales sin necesidad de realizar un preproceso ni una cuantización previa.

Para el clasificador *naïve Bayes* se han probado tres modelos estadísticos: modelo de Bernoulli multivariante, mezclas de Bernoulli y multinomial. Los resultados obtenidos con todos ellos han sido parecidos, y han mostrado un comportamiento similar al mostrado en trabajos que usan infomación textual. Sin embargo, el modelo de mezclas de Bernoulli no ha obtenido resultados tan buenos como se esperaba. Cuando se usa con textos, este modelo permite reflejar la posibilidad de que un mismo documento trate varios temas. Esto no ha sido así al usar secuencias musicales. Además de los tres modelos estadísticos, se ha probado un método de selección de características que permite seleccionar los símbolos que mejor ayudan a discriminar unas clases de otras, mejorando así los resultados de la clasificación. Sin embargo, no ha sido posible determinar el tamaño óptimo

del vocabulario, ya que los mejores resultados que se han obtenido con cada corpus han sido con distintos tamaños de vocabulario, por lo que no parece haber una relación aparente entre las características de los corpus y estos tamaños de vocabulario. Ésta es una de las principales debilidades de este método de clasificación, ya que si se establece un tamaño de vocabulario fijo basado en los resultados obtenidos con un corpus se pueden obtener resultados inesperados al evaluar un nuevo conjunto de datos.

Al usar n-gramas se han obtenido resultados excelentes con el corpus *Cruz-4-genres*, usando la codificación desacoplada para las melodías. Este método ha obtenido resultados superiores al clasificador *naïve Bayes*, a pesar de que la diferencia no se puede considerar estadísticamente significativa. En los experimentos con el corpus *Perez-9-genres*, sin embargo, sí que se ha podido observar una gran diferencia entre ambos métodos, obteniendo los mejores resultados con los modelos de n-gramas. Sin embargo, estos experimentos han puesto al descubierto las limitaciones de esta técnica, ya que se ha observado un gran descenso en la tasa de acierto en la clasificación, pasando de un 98% con el corpus *Cruz-4-genres* a un 64% con el corpus *Perez-9-genres*. Esto se debe principalmente al mayor número de géneros y la presencia de estilos más cercanos entre sí.

Con respecto a los formatos de codificación acoplada y desacoplada, los resultados obtenidos han sido mejores en general para la codificación desacoplada. La codificación acoplada permite codificar juntas varias notas de forma simultánea, incorporando así más información sobre el contexto de las notas que la desacoplada. Sin embargo, esto tampoco ha ayudado a mejorar los resultados, ya que al combinar varios símbolos musicales en la codificación, el número de posibilidades crece exponencialmente con el número de notas empleadas, lo que dificulta el proceso de aprendizaje de los clasificadores.

Para mejorar estos resultados la información melódica usada en estos experimentos debería combinarse con otras fuentes de información, de forma que se vean representadas otras dimensiones de la música. Una primera aproximación se presentó en (Ponce de León et al., 2006), donde algunos de los métodos propuestos en esta tesis se combinaron con clasificadores que usan una descripción estadística global de las melodías, combinando las decisiones de los clasificadores individuales usando técnicas de combinación de clasificadores. Los resultados obtenidos usando dichas técnicas superaron los mejores resultados individuales.

Se han realizado además experimentos con secuencias melódicas obtenidas a partir de ficheros de audio, usando para ello ficheros de audio sintetizados a partir de MIDI, y transcritos posteriormente con el sistema de transcripción polifónica descrito en (Pertusa and Iñesta, 2008). Los resultados obtenidos usando las melodías transcritas han sido bastante buenos, aunque se ha apreciado una pérdida significativa en la tasa de acierto en clasificación al usar modelos de n-gramas, mientras que al usar el clasificador *naïve*

Bayes los resultados han sido similares a los obtenidos con el conjunto de datos original. Esto se debe principalmente a que los errores de transcripción tienen un mayor impacto al codificar la melodía con n-gramas, ya que por cada nota incorrecta introducida por el transcriptor se generan n secuencias de n-gramas que no están presentes en la melodía original, distorsionando así las probabilidades estimadas.

Clasificación basada en secuencias armónicas

En estos experimentos se ha estudiado el modelado de estilos musicales usando secuencias armónicas. Para esto se han usado los métodos usados en los experimentos anteriores, el clasificador *naïve Bayes* y los modelos de n-gramas, para construir modelos de los géneros en el corpus construido para esta tesis, esta vez usando secuencias armónicas en lugar de melodías.

En primer lugar se han codificado las secuencias armónicas como progresiones de acordes, usando distintos conjuntos de características para codificar los acordes, donde cada uno de ellos proporciona distintos niveles de información. De esta forma se ha podido estudiar qué nivel de información sobre la estructura de los acordes es necesario para modelar con precisión los géneros musicales. Para las raíces de los acordes se han usado dos codificaciones: absoluta (la nota raíz) y relativa (grado relativo a la tonalidad). Sin embargo, en los experimentos de clasificación no se ha podido apreciar ninguna diferencia significativa entre los resultados obtenidos con ambas codificaciones. En lo que respecta a las extensiones de los acordes, se han usado cuatro formatos de codificación diferentes:

- *Acordes completos*: la raíz más 26 extensiones (major, 4, 7, 7+, 7♭5♯9, 7♭9♯11, 7♯11, 7♯9, 7sus♭9, 7alt, maj7, maj7♯5, 9♯11, maj9♯11, 11, 13♯11, 13alt, 13sus, m, m♯5, m6, m7, m7♭5, aug, dim, whole).
- *Tríadas*: todas las tríadas posibles (major, ♭5, aug, dim, m, m♯5, sus).
- *Acordes de 4 notas:* tríadas mayor y menor, y algunos acordes de séptima (major, 7, maj7, m, m7).
- *Mayor y menor:* solamente tríadas mayor y menor.

Los resultados obtenidos en estos experimentos son más interesantes, ya que se ha demostrado que no es necesario contar con la estructura completa de los acordes para construir modelos precisos de géneros musicales. Los resultados obtenidos con los conjuntos de acordes completos y de 4 notas alcanzaron una tasa de acierto en clasificación del 87%, superando los resultados obtenidos en los experimentos con secuencias melódicas. Al usar tríadas, sin embargo, los resultados fueron ligeramente peores, aunque aún así se pueden considerar satisfactorios teniendo en cuenta la simplicidad del vocabulario empleado.

En un intento de mejorar estos modelos se ha introducido información rítmica, extendiendo la codificación de las progresiones de acordes incorporando una representación del ritmo armónico. Sin embargo, estos experimentos han mostrado un comportamiento similar que al usar únicamente progresiones de acordes, por lo que se ha abandonado esta codificación al no proporcionar ningún beneficio a la hora de clasificar. Además, este método de codificación es más complejo, ya que se necesita conocer la estructura métrica de la canción, información que no siempre está disponible.

Los modelos armónicos se han probado también usando el corpus de ficheros de audio. Para esto se ha utilizado el algoritmo de extracción de acordes descrito en (Gómez, 2006). Los resultados obtenidos en estos experimentos han sido excelentes, alcanzando tasas de acierto iguales a las obtenidas con el conjunto de datos original, a pesar de los errores introducidos por el algoritmo de extracción de acordes. Este comportamiento se puede atribuir al hecho de que los ficheros de audio contienen más información estilística que el conjunto de datos simbólico. Los ficheros de audio han sido generados a partir de ficheros MIDI, en los que la melodía fue interpretada por un músico. Estas interpretaciones se realizan normalmente en función del género musical al que pertenece la canción, y proporcionan por lo tanto más información sobre el género de la que hay en la partitura original.

Combinación de clasificadores melódicos y armónicos

En todos los experimentos presentados anteriormente se ha probado la capacidad de dos algoritmos — *naïve Bayes* y modelos de n-gramas — para clasificar secuencias musicales, bien codificando secuencias melódicas o armónicas contenidas en ficheros simbólicos, o bien usando algoritmos de transcripción para obtener dichas secuencias a partir de ficheros de audio. Los resultados obtenidos han sido muy variables. En algunos casos los mejores resultados se han obtenido usando el clasificador *naïve Bayes*, mientras que en otros los modelos de n-gramas han funcionado mejor. Lo mismo ha ocurrido con respecto a los distintos parámetros de cada clasificador, e incluso con los formatos de codificación usados para las secuencias melódicas y armónicas, por lo que no es posible predecir con seguridad cuál sería el comportamiento de estos sistemas con un nuevo conjunto de datos.

Una forma de reducir esta incertidumbre es utilizar una combinación de clasificadores. Esta técnica permite combinar las decisiones tomadas por distintos clasificadores, y tiene la propiedad de que normalmente obtiene los mismos resultados — si no mejores — que el mejor clasificador individual usado en el conjunto (Moreno-Seco et al., 2006). De esta forma se puede construir un clasificador más robusto basado en las decisiones tomadas por las diferentes técnicas usadas en esta tesis, reduciendo de esta manera el riesgo de seleccionar un método inadecuado para nuevos conjuntos de datos.

Otra ventaja importante de esta técnica es que permite combinar diferentes decisiones tomadas usando varias fuentes de datos, de forma que se pueden combinar clasificadores basados en secuencias melódicas y armónicas para obtener resultados mejores que usando cada una de estas representaciones por separado. Para probar esta técnica se han realizado dos grupos de experimentos, combinando clasificadores entrenados con secuencias obtenidas a partir de ficheros simbólicos y de audio, respectivamente.

Los resultados obtenidos con esta técnica no han mostrado una importante mejora sobre los resultados de los mejores clasificadores individuales, aunque sí que han sido mejores en general. Sin embargo, esta técnica ha demostrado de nuevo ser más robusta, ya que en ningún caso han empeorado los resultados, y se evita así la necesidad de escoger un clasificador único con el riesgo que ello conlleva.

Modelado de compositores y atribución de autoría

Hasta el momento, todos los experimentos presentados abordan la tarea de clasificación de géneros musicales. Sin embargo, la metodología presentada en esta tesis no está limitada únicamente a este problema, sino que se puede generalizar para otras aplicaciones de modelado de estilos en general. Para demostrar esto se han propuesto varios experimentos para modelar el estilo de distintos compositores, usando modelos de n-gramas construidos a partir de secuencias melódicas para abordar los problemas expuestos en (van Kranenburg and Backer, 2004) y (van Kranenburg, 2006).

En un primer experimento se ha utilizado un corpus de cinco compositores, recopilado por van Kranenburg and Backer (2004), tres de ellos pertenecientes al estilo Barroco (Bach, Handel y Telemann) y otros dos al Clasicismo (Haydn y Mozart). Las tasas de acierto obtenidas han sido muy altas al clasificar pares de compositores pertenecientes a distintas épocas, alcanzando resultados entre el 94% y el 98% de acierto. Sin embargo, al clasificar entre Haydn y Mozart — compositores con estilos muy parecidos según los expertos — esta cifra ha descendido hasta el 75%.

En el segundo experimento se ha tratado de abordar un problema de atribución de autoría, presentado originalmente por van Kranenburg (2006). Esta tarea consiste en estudiar el origen de un conjunto de obras del catálogo de J. S. Bach, cuya autoría ha sido puesta en duda recientemente. En concreto, las obras estudiadas han sido nueve fugas (BWV 534, 536, 537, 555, 557, 558, 559, 560 y 565), y los compositores que se sospecha que pueden haber sido sus legítimos autores son W. F. Bach (hijo de J. S. Bach), J. L. Krebs (discípulo de J. S. Bach) o J. P. Kellner (copista de algunas composiciones para órgano de J. S. Bach).

Para este estudio se ha construido un modelo de cada compositor a partir de un corpus etiquetado de fugas, de las que no hay ninguna duda sobre su autoría. Después se han usado estos modelos para comprobar hasta qué punto son capaces de predecir las fugas polémicas, dando así una idea de cuál es el estilo al que más se parecen y, por lo tanto, el compositor que con mayor probabilidad fue su legítimo autor.

Los resultados obtenidos en este experimento, aunque no son concluyentes, respaldan las conclusiones obtenidas con otros métodos más sofisticados en trabajos anteriores, en los que es necesario un conocimiento exhaustivo del estilo de los compositores en disputa para seleccionar las características adecuadas para esta tarea. La principal ventaja del método empleado en esta tesis es que no requiere de un análisis complejo de las melodías, ya que solamente se emplean los intervalos de alturas y las proporciones de duración entre notas, y además se puede emplear para música monofónica o polifónica indistintamente. Por otro lado, en comparación con el conjunto de características propuesto por van Kranenburg, este método tiene el inconveniente de que no proporciona ninguna pista sobre cuáles son las diferencias entre los estilos de dos compositores. Solamente permite evaluar la similitud entre dos estilos, pero no se puede realizar ningún análisis más profundo sobre los resultados.

Conclusiones y trabajo futuro

En esta tesis se ha demostrado que es posible modelar diferentes aspectos del estilo musical usando un esquema de aprendizaje supervisado. Además, se ha estudiado la equivalencia entre el lenguaje natural y el musical, mediante la aplicación de técnicas usadas tradicionalmente en el área de procesamiento del lenguaje natural. Para esto se han empleado secuencias melódicas y armónicas obtenidas a partir de ficheros musicales con información simbólica, adaptadas mediante un formato de codificación para transformarlas en texto.

Las principales contribuciones de esta tesis han sido:

- La recopilación de un nuevo corpus de géneros musicales, con informacion melódica y armónica para cada canción. Este corpus es más completo que muchos de los usados en trabajos anteriores, ya que contiene más géneros con relaciones más estrechas entre ellos.

- Se ha demostrado las limitaciones de los métodos de clasificación de secuencia melódicas basados en modelos de n-gramas, usados en trabajos previos con resultados muy buenos. Estos resultados sugieren que es necesario explorar nuevas fuentes de información y métodos alternativos para mejorar en esta tarea.

- Se ha demostrado empíricamente que no es necesario utilizar información armónica completa para construir modelos armónicos precisos de géneros musicales. Además, se ha demostrado también que estos modelos se pueden usar con ficheros de audio, usando sistemas actuales de transcripción de acordes sin degradar la precisión del sistema de clasificación a pesar de los errores introducidos en el proceso de transcripción.

- Se ha demostrado además que la metodología propuesta no está limitada al problema de clasificación de géneros musicales, sino que se puede adaptar para resolver otros problemas en MIR. Para esto se ha propuesto un método para estudiar la autoría de piezas musicales, alcanzando las mismas conclusiones que otros métodos más sofisticados que usan representaciones más complejas del conocimiento musical.

Los métodos propuestos en esta tesis han demostrado ser útiles para el estudio de la similitud de estilos musicales. Sin embargo, estos métodos pueden ser útiles en otras tareas relacionadas con la recuperación de información musical. Estos métodos se podrían utilizar por ejemplo en un sistema de composición automática guiado por el estilo, de forma que permitan evaluar la calidad de las composiciones generadas por el sistema. Además, la combinación de modelos melódicos y armónicos permitiría generar piezas musicales más coherentes, ya que esto permitiría evaluar la calidad de estas composiciones a nivel local usando modelos melódicos, y global, usando modelos armónicos que reflejan relaciones a más largo plazo (Paiement, 2008). Otras posibles aplicaciones son la depuración de transcripciones polifónicas, ayudando a detectar secuencias improbables de notas; la segmentación automática, aprendiendo a detectar los puntos de corte en las melodías; y la extracción de motivos, mediante un análisis estadístico de las secuencias más representativas de una melodía.

Contents

List of Figures

List of Tables

1 Introduction

In the recent years, the widespread availability of broadband internet connections and the lowering costs of technology, specially high capacity storage devices, have favoured the development of large digital libraries with multimedia content. Well-known examples of this kind of libraries are YouTube[1], Magnatune[2], Last.fm[3], or Wikimedia Commons[4]. Among them, digital music libraries deserve special attention due to their high commercial interest. On the one hand, music consumers are shifting their preferences from the traditional physical media formats to the digital storage, a change mainly caused by the advances in compression technologies and the increasing offer of multimedia devices able to handle these new formats. On the other hand, artists and the music industry have seen this as an opportunity to open new markets, using internet as a distribution channel and thus increasing the offer of both music in digital format and all its surrounding technologies. This situation has created a virtuous circle, causing online music repositories and stores, and even personal collections, to grow with huge volumes of digital music contents.

Standard bibliographic techniques, based on descriptive files with metadata records, seem to be unable to handle these overwhelming volumes of data. First, the creation and maintenance of these records by hand requires a great investment of time and money that may be unaffordable for many individuals or institutions. Secondly, users are demanding new and more effective ways to browse these musical collections, based on their own musical preferences rather than on standard taxonomies. These and other factors have raised new challenges regarding the creation, management, and access to music libraries, that require new methods for automatic metadata computation, automatically indexing musical data, and also to provide more attractive browsing interfaces to users.

[1] http://www.youtube.com
[2] http://www.magnatune.com
[3] http://www.lastfm.com
[4] http://commons.wikimedia.org

1.1 Music Information Retrieval

Music Information Retrieval (MIR) is an interdisciplinary research area that aims at providing solutions to the problems exposed before. It encompasses research communities from many subject areas, such as Computer Science, Digital Signal Processing, Musicology, and Cognitive Science, to name a few. Researchers from these areas have contributed with their own experience from other similar tasks, as for example speech recognition or information retrieval from text databases, to face the problems that arise when working with musical information under their own points of view. This heterogeneity has proven to be very beneficial to the overall MIR community, because it allows to integrate knowledge from all the disciplines involved to develop new and exciting systems to access musical information. On the other hand, the coordination of MIR communities is often hindered by the differences in scientific backgrounds, methodology, and even terminology used by researchers from different disciplines. A good overview on these issues can be found in (Futrelle and Downie, 2003).

An interesting particularity of MIR research is that it is focused on providing *content-based* methods to access musical information, as opposed to traditional systems that are based on metadata or external descriptions of musical content. This new approach has allowed to create a wide range of interesting applications, that can be coarsely categorized as follows:

- **Information retrieval**. New music information retrieval systems allow to query a database by presenting an example — a musical excerpt — and the system returns a number of items from the database ranked by their similarity to the given example. Among them, special interest have received the so-called "query-by-humming" or "query-by-singing" systems, in which the user can enter the query by directly singing a fragment of the song to a microphone.

- **Automatic organization and indexation**. The purpose of these systems is to atomatically organize music pieces when their metadata are not available. This can be achieved using two different approaches. The first option is to analyze the musical content of the songs in order to obtain some semantically meaningful metadata, such as musical genre, in order to classify the pieces into a predefined taxonomy. The other possibility is to organize songs in groups based in their similarity, according to some similarity criterion.

- **Recommendation systems**. These systems help users to find new music which is similar to their likes, for example by abstracting common characteristics of the songs in their own music collections. These systems are in fact very similar to the ones used for automatically

organizing music libraries, since they can use the same techniques in order to find similar music in a database of new artists.

- **Musical analysis**. MIR research has also provided new tools for music students and scholars. These tools allow to perform automatic analyses of music, such as key estimation, pitch spelling, or harmonic analysis. Some of them are also suitable to obtain useful metadata for automatic indexing, as for example the tonality of a song.

Key concepts in MIR research are music similarity and style. Searching for a song in a database, categorizing a new song in a given taxonomy, or finding music in the style of a user require computer algorithms able to compute the similarity of a set of songs, although the precise meaning of similarity varies from one problem to another. Similarity in music can be found at different layers, such as melody, harmony, lyrics, or timbre, or it can be even perceived as a subjective experience depending on the feeling music arises in the listener (*mood*, as for example *sad* music). However, for building an ideal, human-like system for computing music similarity or perceive musical style it would be necessary to know the mechanisms of the human brain underlying the perception of music. Although some basic approximations can be done to model music perception (Shmulevich et al., 2001; Temperley, 2006), it is yet unknown how music is really perceived, and what makes two songs sound similar.

In this context, Artificial Intelligence techniques provide a good alternative to solve the problems that arise in the development of these systems. Artificial Intelligence (AI) is a field of computer science that tries to make computers imitate human behaviour, by taking *intelligent* decisions when facing a particular problem. Among the many subfields in AI, Pattern Recognition is one of the most popular within the MIR community.

1.2 Pattern recognition

Pattern recognition (PR) is a subfield of AI that tries to emulate the processes of the human brain underlying the identification of known patterns. Duda et al. (2000) define it as *the act of taking in raw data and taking an action based on the "category" of the pattern.* Thus, the main application of pattern recognition is the *classification* of patterns. However, this is a very general definition that masks the real complexity of the task. In practice, the categories to be identified — and the actions to be taken — must be defined depending on the nature of the tackled problem. Some typical applications of PR are machine vision, character recognition, computer-aided diagnosis, speech recognition, and data mining (Theodoridis and Koutroumbas, 2008).

All PR systems share the same basic structure. First, a sensor captures the patterns to be identified. Since this raw data is usually unstructured and difficult to handle, a set of suitable features are extracted. Then, these features are feeded to a classifier that outputs the corresponding class, and finally, the proper action is taken according to the class of the pattern. The core of a pattern recognition system are the feature extraction and classification steps. Feature extraction is maybe the most challenging part, since it usually requires knowledge from other disciplines. For example, digital signal processing techniques are needed to process digital audio signals captured by a microphone in a speech recognition task. Also, it is necessary to select the correct features and classification method, and sometimes there is not a priori knowledge of which of them are the most appropriate for a particular problem, so this decision must be taken by mere intuition. All these topics are thoroughly covered in (Duda et al., 2000; Theodoridis and Koutroumbas, 2008).

Classification methods are mainly divided in two groups: *supervised* and *unsupervised*. In supervised classification the classes are defined by a labeled set of training samples, which are used to train the classifier. In order to classify new patterns, they are assigned to the class to which they are more similar. This similarity can be computed in many different ways, depending on the classification method. Some typical similary measures are the edit distance to a class prototype or the probability of belonging to a probability density function estimated from the training classes. In an unsupervised classification scheme, the training samples are not labeled, and the classification task consists in grouping or *clustering* the samples based on their similarity.

1.3 Contributions of Pattern Recognition to MIR

Due to the nature of pattern recognition techniques, their most straightforward application in MIR is music classification. However, PR techniques have been also used in a number of MIR tasks, such as music retrieval or description of musical content.

The introduction of PR algorithms in information retrieval systems is not new to MIR research. Traditional (textual) retrieval systems use probabilistic models for estimating the probability of relevance of documents to user queries (Ponte and Croft, 1998). These techniques have been successfully applied to music, using melodic (Dannenberg et al., 2007) or harmonic (Pickens et al., 2002) sequences. PR techniques are also useful to improve the performance of such systems, as in (Little et al., 2007), where a query-by-humming system is trained to recognize the singing errors of the user using the feedback received on the retrieved melodies.

The ability of PR algorithms for modeling user input has also been profited by some recommendation systems. In (Moh and Buhmann, 2008), the system tracks user preferences on artists in order to offer new songs of artists in the user's profile. Other systems such as (Magno and Sable, 2008) and (Dopler et al., 2008) use unsupervised methods to build clusters of similar songs in order to generate playlists or recommend songs in the user's likes.

Musical style classifiers are useful tools for the automatic organization of music databases, because they can provide semantic descriptions of musical content. Stylistic labels such as genre (Scaringella et al., 2006), composer (van Kranenburg and Backer, 2004), or mood (Hu et al., 2008) are typical indexes used in such databases. Since PR techniques allow to capture the common characteristics shared by a set of songs, it is easy to build models of musical styles by presenting the system some examples in those styles, avoiding the need to formulate a computable definition of what a style is. Style modeling is a versatile tool that not only allows to perform classification, but also to use automatic composition algorithms to generate new songs in the learnt styles (Cope, 1996; Cruz-Alcázar and Vidal, 2008).

Using PR algorithms on digital audio signals it is possible to obtain interesting descriptions of musical content. For example, existing systems for detecting passages with human voice (Feng et al., 2008; Tsai et al., 2008), identifying musical instruments (Little and Pardo, 2008), or for discriminating pitched from unpitched sound (Camacho, 2008), can be used to build indexes for performing advanced queries on musical content. Nonetheless, the more interesting description that can be obtained from a music audio file is the original score. There are several reliable algorithms that are able to perform automatic transcription from monophonic signals, but automatic polyphonic transcription is a complex problem that is far from being completely solved. Although most works in this area use only digital signal processing techniques, some of them have used a pattern recognition approach (Marolt, 2004; Pertusa and Iñesta, 2004). A similar task is the automatic chord recognition in audio signals, where good results have been reported using PR algorithms and a limited set of chords (Bello and Pickens, 2005; Gómez, 2006; Lee and Slaney, 2006; Sheh and Ellis, 2003).

Finally, some contributions from the PR field to the automatic analysis of music can be also found. Illescas et al. (2008) proposed a melodic and harmonic analysis system, in which the probabilities of the transitions between chords are learnt from a corpus of previously analyzed scores. Another application is shown in (Pearce et al., 2008), where a melodic segmentation system is trained to recognize phrase boundaries.

This thesis is focused in the classification of musical style using a supervised approach. In the following sections this problem is discussed in detail, along with a review of how it has been approached in the literature, and a brief outline of the approach followed in this thesis.

1.4 Sources of musical information

In order to successfully apply pattern recognition algorithms it is necessary to extract the suitable features from the data. However, the features that can be extracted depend highly on the nature of the raw data. In the case of music, it can be found in many representation formats, that can be grouped in two big families: audio and symbolic.

1.4.1 Audio and symbolic music

Traditionally, the research on music information retrieval has been divided into the audio and symbolic domains. Digital audio files contain a digitized audio signal coming from a sound recording, and can be found in compressed (MP3) or uncompressed (WAV) formats. On the other hand, symbolic music files contain *digital scores*, i.e. the music notation and instructions necessary for a computer or synthesizer to play the song. These files can be sequenced (MIDI) or structurally represented (MusicXML).

The choice for audio or symbolic formats is often guided by the purpose of the task. For example, instrument or voice detection does only make sense in the audio domain, while harmonic analysis can be only performed at the symbolic level. Other tasks, such as style classification, do not require any specific music source. However, there have been some claims that systems working with symbolic information do not really reflect user's needs, since real-world systems should work with audio databases. This is partially true, because many people store their music files in digital audio format, as well as music retailers do, but, on the other hand, music scholars always work with scores, never with audio files.

Audio files contain richer information than symbolic ones, because they bring together all the different elements that take part in a musical work: pitch, harmony, rhythm, timbre, lyrics, etc. However, this richness results in a bigger complexity, because all this information is mixed in the audio signal and it is very difficult to isolate each one of these elements from the others. For working with audio files it is necessary to use digital signal processing techniques in order to extract some features representing the musical content. For example, one of the most common features used for this purpose are Mel frequency cepstral coefficients, which provide information on the timbre of the piece (Aucouturier and Pachet, 2004). Other works use features regarding rhythm (Lidy and Rauber, 2005), texture (Tzanetakis and Cook, 2002), or pitch (Tzanetakis et al., 2003). However, these features only provide unaccurate descriptions of the musical content and are difficult to interpret.

Symbolic music files, on the other hand, provide very accurate information on the actual content of a musical work — the score — including pitch, harmony, rhythm, and lyrics, but lack some important features that

can be only found in the audio files, such as timbre and performance and production issues. Despite this limitation, musical scores contain stylistic traits that are inherent to their composer, and that are reflected in all their compositions (Cope, 1996). Also, recent advances in polyphonic transcription and chord recognition from audio files suggest that it is possible to use such algorithms to obtain a symbolic representation of music. Symbolic systems can be then used as a back-end, working with the symbolic sequences obtained from the audio files.

This thesis focuses in the symbolic approach, in order to explore the amount of stylistic information contained in the score of a musical piece. For this purpose, two different dimensions of music are explored: melody (horizontal dimension) and harmony (vertical dimension).

1.4.2 Melody

A melody is a monophonic sequence of pitches and durations, i.e. a sequence of musical notes where there is not more than one note sounding simultaneously (Pickens, 2001). Melodies usually contain the main motifs, and have some properties that make them able to be sung (Schönberg, 1967), so they are usually the most easy part to remember from a musical piece.

There are several authors that have studied different encoding schemes for melodies in MIR tasks. Downie (1999) studied the suitability of using melodic sequences encoded as small overlapping subsequences (n-grams) of pitch intervals for music retrieval. He called these subsequences *musical words*, trying to establish an equivalence between music and text. In these experiments, he concluded that there is an equivalence between these musical words and the words in a text, because they provide a similar amount of information to retrieve the class of the document (or musical piece) they belong to. In the same line, Doraisamy and Rüger (2003) extended these musical words, adding rhythm information by encoding together pitch intervals and duration ratios extracted from n-grams of notes. Using this encoding, they reported good results in the evaluation of their melody retrieval system, in both query-by-humming and query-by-example tasks.

The most complete study on melodic encoding so far has been carried out by Cruz-Alcázar and Vidal (2008). In their work, they studied several combinations of absolute and relative encodings of pitch and duration of notes, and evaluated their performance in two different tasks: classification of musical style and automatic composition. The conclusions of this work vary depending on the task. In the musical style classification task, the best results were obtained using a relative encoding of pitch (pitch intervals), but the results are not conclusive regarding the encoding of note durations, since excellent results were obtained with both absolute and relative encodings, depending on the training corpus. In the automatic composition task, however, only the absolute encoding of duration was tested, and the results

were more difficult to interpret since they were based on a subjective evaluation of the generated pieces.

To summarize, all these works have tried to find a suitable representation for melodies, mapping musical symbols into text sequences in order to be able to use techniques traditionally used with textual information. The encoding formats used in these systems are strongly influenced by psychoacousic studies that have shown that musical perception is based in the relationships between notes rather than in their absolute values (Parncutt and Drake, 2001), and this assumption has been supported by their experimental results.

1.4.3 Harmony

In music, harmony refers to the study of the simultaneity of sounds. In contrast to melody, harmony studies the vertical organization of musical notes, the way in which they are perceived together in tonal music. Thus, harmony is closely related to the concept of tonality. The tonality of a musical piece is the scale from which most of the notes are drawn, and dictates how they can be combined both vertically and horizontally. The name of the scale is called the key of a song.

A vertical arrangement of three or more notes is known as a chord, and a sequence of chords is called a chord progression. The organization of chord progressions is the main subject of study of harmony. Each chord has a different "meaning" depending on its composition and its relationship to the key and its surrounding chords. There are chords that produce a relaxation effect to the listener, while others introduce a tension that needs to be relaxed by future chords. The concatenation of those tensions and relaxations is the basis of composition in most Western music.

Chord progressions are usually encoded as a sequence of chord names, using standard musical notation. In this notation, each chord is represented by the root note and an extension that specifies the structure of the chord. Some examples are presented in Table 1.1. Chord inversions (i.e. when the lower note in the chord is not the root) are encoded by specifying the lower note after a slash. Using this notation, any chord progression can be encoded as a text sequence, which makes it a desiderable property for working with the same techniques usually applied with text or melodic sequences.

Other alternative encodings, as the ones used in (Harte et al., 2005) and (Paiement, 2008), include all the component notes of the chords, explicitly encoding their structure. Harte et al. (2005) proposed an unambiguous, context-independent syntax, in which each chord is denoted by the root note and the structure is specified by the list of halftones, measured from the root, of the rest of component notes of the chord. In (Paiement, 2008), a similar encoding is used, but explicit jazz voicings are given for each chord, providing precise information on what octave each note belongs to. This

Chord name	Component notes			
C	C	E	G	
Cm7	C	E♭	G	B♭
Cmaj7♯5	C	E	G♯	B
Cmaj7/B	B	C	E	G

Table 1.1: Examples of chord notations and their component notes.

way, the perceived loudness of each note can be computed as a function of their frequencies, and this information is used to compute a similarity measure of chords, based on psychoachoustic features of music.

1.5 State of the art in musical style classification

This section presents the problem of the classification of musical styles. Then, some of the most relevant works dealing with this problem are reviewed, focusing on those that use the symbolic approach.

1.5.1 Musical style

Musical style is a quality of music that most people can perceive intuitively. It is often used to describe, categorize, and even compare songs and albums, although there is not a formal definition of what a musical style is. However, some authors have given some interesting definitions of the term:

- *Style is a replication of patterning, whether in human behavior or in the artifacts produced by human behavior, that results from a series of choices made within some set of constraints* (Meyer, 1989).
- (Style is) *a recurring arrangement of features in musical events which is typical of an individual (composer, performer), a group of musicians, a genre, a place, a period of time* (Fabbri, 1999).

These two definitions coincide in that what makes a style is the repetition of some elements, and thus pattern recognition techniques seem to be the perfect tools to discover these patterns in order to model musical style. Moreover, the definition given by Fabbri fits perfectly the way this term has been used in the MIR literature, since many works in this area have focused in different aspects of musical style, including genre, composer, geographical origin, or historical periods, among others.

From all these different significations of style, the most widely studied has been the classification of music in genres. Musical genres arise mainly due to the influence of the music industry, in an effort to offer an organized

catalog of albums and artists and to orient consumers to choose the music they may like. However, there is a lack of consensus in the definition of these labels, and important inconsistencies can be found both in the number of labels used and in the way they are used (Aucouturier and Pachet, 2003; Lippens et al., 2004). Despite this, the organization of music in genres is still the most popular among music consumers, and it has caught the attention of many researchers from the two domains: audio (Aucouturier and Pachet, 2003; Scaringella et al., 2006) and symbolic (Gedik and Alpkocak, 2006; Karydis et al., 2006; Lin et al., 2004; Manaris et al., 2005; Ponce de León and Iñesta, 2007; Ruppin and Yeshurun, 2006; Tzanetakis et al., 2003).

Some authors have also studied the classification of composer styles (Backer and van Kranenburg, 2005; Buzzanca, 2002; Manaris et al., 2005; Margulis and Beatty, 2008; Wołkowicz et al., 2008), mostly focused on the recognition of classical composers. One advantage of this task over the classification of genres is that building a ground truth of pieces labeled with their composer is far easier than building a data set of musical genres, since everybody agrees in the authorship of pieces composed by well-known artists (except in some exceptional cases as it will be discussed in Chapter 6).

Other style classification tasks that have been addressed in the literature are the recognition of mood (Hu et al., 2008), aesthetics (Manaris et al., 2005), performance style (Dannenberg et al., 1997; Stamatatos and Widmer, 2005), production style (Tzanetakis et al., 2007), and identification of the geographic origin of folk songs (Chai and Vercoe, 2001).

In the following sections, some of the most relevant works on musical style classification using a symbolic representation of music are reviewed, either using melodic or harmonic sequences. Works dealing with audio data have been excluded from this review because most of them are focused mainly on the feature extraction techniques used to extract information from digital audio signals, a subject that falls out of the scope of this thesis. A good review on these works, although a bit outdated and focused on genre classification only, can be found in (Aucouturier and Pachet, 2003; Scaringella et al., 2006). The reader can also refer to the ISMIR conference proceedings[5], one of the most important conferences in the MIR field, where a fair amount of these works can be found.

It must be noted that it is rather difficult to compare these works because most of them use a different data set for their experiments. There are many reasons for this situation. First, the different conceptions of musical style make it impossible to compare works dealing with different tasks, such as genre or composer classification. However, even when the same task is carried out, most authors have gathered their own data sets in different file formats, because there is not any standardized corpus for testing style classification algorithms. Music files are usually copyrighted material, and

[5] http://www.ismir.net

this has prevented the MIR community from the creation of any publicly available data set, although some attempts have been done in this direction, such as the RWC database[6] or the Music Information Retrieval Evaluation eXchange (MIREX).

The RWC database is a copyright-cleared database which is available for researchers at distribution costs only. It contains several collections, as for example one collection of recorded instrument sounds and a music genre database. However, this database contains just 99 files divided in 10 categories and 33 subcategories, with 3 pieces in each, which provides insufficient data for training pattern recognition algorithms. The MIREX, on the other hand, is a MIR task evaluation contest that is held along with the ISMIR conferece series since 2005. In this contest, some MIR tasks are proposed yearly to allow researchers in this area to test and compare their algorithms with other researcher's using the same data set for all of them. In 2005 a symbolic genre classification task was carried out, where the participants had to send their algorithms to the organizers, instead of distributing the data set to avoid copyright issues. However, this task was discontinued and has no longer been carried out in succeeding editions of the contest.

1.5.2 Classification using melody

All the works that perform classification of musical styles using melodic information are based solely on the information contained in the musical score, and more precisely in the pitch and duration of the notes. Despite the obvious limitations of these methods, since they ignore important stylistic information such as timbre, in an experiment performed with human listeners by Iñesta et al. (2008), the authors found that people are able to distinguish two musical genres in absence of timbre, at least with just a 16% of error. Moreover, Zanette (2008) claims that music perception takes into account statistical properties of music, so statistical tools can be used to study musical qualities. Thus, it seems reasonable to think that pattern recognition techniques can find enough information in the melodic sequences to model the style of a set of musical pieces.

From all the works dealing with melodic sequences, two differentiated groups can be found according to the way music is described. In the first group we can find those works that use a shallow description of musical content, describing the melody using statistical features computed from the notes. In the other group are the works that focus on the local relationships of the notes in the melody, using the melodic sequence itself as the input to the classifier. Both approaches are useful to analyse music at different levels — global and local — and are thus complementary.

[6] http://staff.aist.go.jp/m.goto/RWC-MDB

Methods based on global descriptions

One of the first and most influential works in this area is the work by Dannenberg et al. (1997). In this work, the authors propose a method for automatically classifying the style of a live music performance, using a pitch-to-MIDI interface for capturing trumpet performances. The objective of this work was to identify the performing style using a predefined set of labels (lyrical, frantic, syncopated, pointillistic, blues, quote, high, and low), so these labels could be used by an automatic accompaniment system to play along with a human performer in the same style. The MIDI recordings were divided into fragments of 5 seconds, in order to test whether an on-line version of the system could be used in real time. From these fragments, 13 low-level statistical features were extracted, encoding average and standard deviations of pitch, duration, occupation ratio, and volume, as well as counts of notes, pitch bend and volume change messages. In the experiments three different classifiers were used: Bayesian, linear, and neural networks. The best results obtained were a 99.4% classification rate with 4 class labels and the linear classifier, and 90.0% with 8 classes and the bayesian classifier. However, it must be noted that the features used are somewhat related with the styles used in this task. For example, a "frantical" performance will have a higher number of notes than a "blues", or in a "pointillistic" style the occupation rate should be lower than in others. It is not clear, then, how would this system perform when classifying other aspects of musical style such as genre.

A simpler approach is used in (Tzanetakis et al., 2003), where each song is described using a four-dimensional vector, with four features regarding the two most frequent pitches in the song. A corpus of 500 MIDI files was used in the experiments, divided in five genres: electronica, classical, jazz, irish folk, and rock. Using a k-nearest neighbor classifier (k-NN), an average classification rate of 80% was achieved in pairwise classification, while it dropped to 50% when classifying the five genres. Although the representation format used in this work is quite limited, restricted only to the pitch of the notes, the authors show that it can be easily adapted to work with audio files using a multiple pitch detection algorithm. Despite the transcription errors made by the algorithm, the classifier obtained a 40% classification rate using a corpus of audio files in the same five genres, which is a good result considering the 20% baseline in this problem.

In order to find a suitable set of statistical features for musical genre classification, Ponce de León and Iñesta (2007) performed an extensive study on the discrimination power of a broad set of melodic, harmonic, and rhythmic descriptors. The authors defined a total of 28 different features, including counts, ranges, averages and standard deviations of several melodic elements: notes and rests, absolute and relative measures of the pitch and duration of notes, syncopation, and non-diatonic notes. Then, using a

feature selection procedure, three reduced sets of 6, 10, and 12 descriptors were selected. An interesting conclusion of the feature selection process is that the most discriminating features are those regarding the pitch and intervals of the notes, which were four out of the six topmost discriminant features in the analysis. Several experiments were performed on a corpus of 110 MIDI files from jazz and classical music, using the four feature sets (with 6, 10, 12, and the full set of 28 descriptors) and three different methods: a Bayesian classifier, k-NN, and self-organising maps (SOM). The best result was obtained with the k-NN classifier, reaching a 93% classification rate and the full set of descriptors, although similar results (91%) were obtained with the Bayesian classifier, using the 10 and 12 feature sets.

In the same work, the authors experimented also with the length of the sequences from which the statistical descriptors are computed, in order to test the feasibility of an online classification system as in (Dannenberg et al., 1997). For this purpose, a sliding window was used in the feature extraction step. Several window lengths and shift values were tested in order to find the optimal values for these parameters. These experiments showed that, in general, large melody segments (> 30 bars) are necessary to obtain good classification results. Using this method, better results were obtained than with the whole melody, reaching a 96.4% with the k-NN classifier and a window length of 95 bars.

In (Manaris et al., 2005), the authors establish a connection between natural language (and other natural phenomena) and music by means of Zipf's Law. They show that several musical features such as pitch, duration, and harmonic consonance follow a Zipf's distribution, and then use these distributions to train artificial neural networks in different classification tasks. In particular, three experiments were conducted building models of composer styles, genres, and an *aesthetic* quality of music defined as "pleasantness". In the composer modeling task, they reached a 95% accuracy using a corpus containing five classical composers: Scarlatti, Purcell, Bach, Chopin, and Debussy. In the genre modeling task, although no classification results are reported, they show how these features can help to distinguish musical genres using an analysis of variance. Finally, in the experiments performed to model the pleasantness of music, the authors built a data set of 12 pieces, six of them from classical composers, and the other six from twelve-tone composers. The pieces were first rated by 21 people with different musical background, that agreed in average to label the classical compositions as pleasant and the others as unpleasant. Then, two neural networks were trained using a leaving-one-out scheme to recognize both types of music, using musical fragments extracted from the pieces. This method correctly classified all the pieces. Only a fraction of a composition by Berg was misclassified, while for the others the entire pieces were correctly classified with a 100% accuracy.

The methods explained above compute a set of statistics from the melody, either using the whole melody or melodic segments selected at random. Karydis et al. (2006) proposed an improvement over these techniques, by using only significant parts of the melodies in the computation of the statistical features. These significant sections were selected using a repeating pattern discovery algorithm, that looks for the longest non-trivial patterns in a training corpus of melodies encoded using pitch and duration values. Then, a set of statistical features were computed separately for pitch and duration sequences. Pitch features were computed from a co-occurrence matrix of pitch values, allowing gaps between the musical symbols in the patterns, and for durations three features were computed regarding the two most frequent duration values. Finally, classification was performed using a weighted scheme, in which two different k-NN classifiers were trained using pitch and duration features separately, and two sets of k nearest neighbors were selected. For making the final decision, the votes of all the neighbors were weighted, giving a higher authority to those selected using pitch information. The voting weights were fixed to 70%–30% (pitch–duration) empirically. In the experiments using a corpus of 5 classical subgenres (ballad, choral, fugue, mazurka, and sonata), this method achieved a 92% accuracy, with a significant improvement over the method proposed in (Tzanetakis et al., 2003), that obtained an accuracy close to a 55% using the same data set.

All the works reviewed so far used different sets of statistical features. However, despite the differences, there is some homogeneity in the sense that all of them computed some statistics from the distributions of pitch and duration of the notes. Another interesting point is that most of these features were selected according to their interest from a musicological point of view, but not focusing in the particular data set used in the experiments. In can be expected, then, that these methods could be used in different style classification tasks with different data sets. Other authors, however, have developed specific features in order to solve a particular problem. Some of them are discussed below.

Gedik and Alpkocak (2006) used a classification scheme very similar to the one used in (Tzanetakis et al., 2003). However, in this work each song was described using a three-dimensional vector, with harmonic and rhythmic features specially designed to distinguish the three genres used: jazz, classical, and pop music. These features were selected in order to reflect the differences in the use of harmony and rhythm in these styles, such as the number of vertical and horizontal dissonant intervals, or the number of beats occurring at irregular positions. These features are computed from three-second windows extracted at random from the songs, and then classification is performed using a k-NN classifier. In the experimens using a corpus of 225 MIDI files, an average classification rate of 82% was achieved.

In (Backer and van Kranenburg, 2005) the authors used a set of features extracted from polyphonic compositions by five composers: Bach, Handel, Telemann, Haydn, and Mozart. The particularity of this work was that the features used were based on advanced musicological knowledge, and were only suitable to describe polyphonic music. In the experiments using a k-nearest neighbor classifier and different arrangements of classes, the results ranged between 79.4% and 95.2%.

Finally, Margulis and Beatty (2008) proposed the use of entropy as an analytical tool for studying musical style. Using three entropy measures (first-order, conditional, and normalized entropy), the authors study the differences in the styles of different composers, including Bach, Corelli, Handel, Telemann, Haydn, Mozart, and barbershop quartets (a capella music). Music pieces were represented using sequences encoding different attributes of pitch, texture and duration separately. This way, the authors measured the independent contribution of these features to distinguish the style of each composer from the others. Although the authors did not propose an automated method for distinguishing musical styles, they performed an analytical study of the styles of the composers in the corpus, showing that each style can be distinguished from the others using a different set of features. For example, vocal styles as barbershops or Bach's chorales exhibited a lower conditional entropy for pitch sequences than the others due to the range limitations of the human voice, while Haydn and Mozart were better differentiated from the others using rhythmic sequences. Another interesting finding of this work was that there was a correlation between the composer birth dates and the average entropy of their works, that reflects the stylistic evolution over time in the past centuries.

Methods based on local descriptions

Some authors have chosen to use the actual melodic sequence as the input to the classifier, instead of using statistical descriptors of its content. This way, the classifier learns the melodic constructions that are typical of each style, and uses this information to detect the same (or similar) structures in the new pieces to be classified. However, it is not possible to feed the classifiers using melodies in their original form, since they are usually contained in binary formats (MIDI) or other representation formats where the melody is mixed with formatting or structure information (**kern, MusicXML). It is necessary then to extract the relevant information on the pitch and duration of the notes, transforming the melody into a symbolic format suitable for this task. Most authors use an encoding format that maps musical symbols into text sequences, encoding pitch and duration values with alphanumeric characters. This way of encoding musical information has the additional advantage that it allows to use traditional text classification

methods, without the need of developing new techniques to deal with musical information.

The parallelism between music and text is not limited to the musical encoding. Some authors have also explored this connection using natural language processing tools that assume that music language has a similar structure and organization as natural language. This assumption is supported by generative theories as the one developed by Lerdahl and Jackendoff (1983), who proposed a grammatical theory to analyse tonal music. These works use a syntactic pattern recognition approach, builing a probabilistic model or grammar for each class in the training set of melodic sequences, which is later used to parse the melodies to be classified.

One of the first works in using this approach for classification of musical styles is (Chai and Vercoe, 2001). In this work the authors used Hidden Markov Models (HMM) to classify three different folk music styles: Irish, German, and Austrian, in order to find whether there are significant differences in the folk songs from different countries. For this, an individual HMM was trained for each style, and test melodies were evaluated to select the class of the model that gave the highest probability. Four different encoding formats were also tested: absolute pitch, absolute pitch and rhythm, pitch intervals, and pitch contour. In the experiments, the best results were obtained using the relative encoding of pitch as intervals, with 66%–77% recognition rates in two-way classification, and 63% with three classes. As expected most errors were committed when classifying between German and Austrian folk songs, due to the cultural and geografical proximity of both countries.

In (Cruz-Alcázar and Vidal, 2008), three different classification methods were used: two grammatical inference algorithms (ECGI and k-TSI), and language modeling using n-gram models. These methods were tested with two corpora of musical styles, containing three and four different styles respectively, and several encoding formats as discussed in section 1.4.2. These corpora are described in detail in the next chapter, since they are also used in this thesis. The best results in classification were obtained with the n-gram models, reaching a 98.3% classification rate with the three-classes corpus, and a 99.5% with the four-classes corpus, using the encoding formats discussed above.

A similar approach was used by de la Higuera et al. (2005). In this work, based on previous works by Cruz-Alcázar and Vidal, the same melody encoding formats and the corpus of four classes were used, but with a different classification method. This time a different grammatical inference algorithm (Minimum Divergence Inference, MDI) was used, reaching a classification rate of 96.25%.

Buzzanca (2002) used multi-layer feed-forward neural networks to distinguish the style of one composer from others. In this work a symbolic corpus of 13 composers was used, most of them Renaissance contrapunctists,

in order to build a model of the style of Giovanni P. da Palestrina, and to distinguish it from the others. Melodies were encoded using four symbols per note, encoding duration using absolute values, and pitch using both absolute and relative measures. A classification rate of 97% was achieved when distinguising Palestrina from the others.

Other authors have used a different approach to classify melodic sequences, using similarity measures to compare melodic strings, rather than using the syntactic approach. In (Lin et al., 2004), the edit distance was used as the measure to compute the similarity of a test song to each class in the training set. First, a pattern discovery algorithm was used to extract all the *significant repeating patterns* (SRP) in the training set. In the classification step, all possible SRP were also extracted from the test melody, and each class was scored by the edit distance of these SRP to those extracted from the class. This approach was tested with a data set of 500 MIDI files, from seven different genres: blues, country, dance, jazz, latin, pop, and rock music. The average classification rate was 49.2%, with a significant improvement over the method by Chai and Vercoe (2001), that achieved a 33.7% with the same data set.

Ruppin and Yeshurun (2006) used a compression distance to measure the similarity of pairs of melodies. This method was based on the idea that two melodies can be considered similar if both of them can be well compressed using the information provided by the other. Using the Lempel-Ziv algorithm for compressing melodies and a k-NN classifier, experiments were performed using a corpus of several artists from three genres: classical, pop, and traditional japanese music. This corpus allowed to perform experiments using two different definitions of style, either as a genre classification task using the genre labels, reaching an 85% classification rate, or as a composer identification task, with a 58% classification rate. It is worth noting that the notion of repeating patterns is also present in this work, since the repetition of patterns is the basis of compression algorithms.

Finally, in a recent work by Wołkowicz et al. (2008), n-gram profiles were used to model composer styles, using a corpus of five composers: J. S. Bach, L. van Beethoven, F. Chopin, W. A. Mozart, and F. Schubert. Melodies were encoded using relative measures for pitch and duration, and all the possible n-grams were extracted. Then a feature selection procedure was applied to select the most informative n-grams, and a profile vector was constructed for each composer, measuring the frequency of apparition of those n-grams in their compositions. Classification was performed by comparing the profile of a test song with all the composer's profiles. The best result obtained in the experiments was an 84% classification rate, using n-grams extracted from groups of seven notes.

1.5.3 Classification using harmony

Little attention has been paid in the MIR literature to how harmony can help in the recognition of musical styles. Unlike those works that use melodic sequences where the MIDI format seems to be a standard, there is not a standardized format to represent harmonic information, and thus the authors of these works have tried to obtain this information from different sources. Moreover, the chord vocabulary (i.e. the number of possible chords) varies from one work to another, making the possible comparisons even more difficult.

In Shan et al. (2002), a rule-based system is used to classify sequences of chords belonging to three categories: Enya, Beatles and Chinese folk songs. The main particularity of this work is that the data set used is made up of MIDI files, from which chord progressions are obtained using a set of heuristic rules. A vocabulary of 60 different chords was used, including triads and seventh chords. Classification accuracy ranged from 70% to 84% using two-way classification, and the best results were obtained when trying to distinguish Chinese folk music from the other two styles, which is a reasonable result as both western styles should be closer in terms of harmony, thus leading the system to a bigger confusion.

More recently, Lee (2007) has proposed genre-specific HMMs that learn chord progression characteristic for each genre. Although the ultimate goal of this work is using the genre models to improve the chord recognition rate, the author also presented some results on genre classification, with an 85.7% classification rate when distinguishing rock songs from another style labeled as "universal". For this task a reduced set of chords (major, minor, and diminished) were utilized.

Finally, Paiement (2008) used chord progressions to build probabilistic models of jazz harmony. In this work a set of 52 jazz standards was used, encoded as sequences of 4-note chords, each chord with a duration of 2 beats in a 4 beat meter. The author compared the generalization capabilities of a probabilistic graphical model against a Hidden Markov Model, both capturing stochastic properties of harmony in jazz, and the results suggested that chord structures are a suitable source of information to represent musical genres. However, as the author point out, these models should be tested in a more general framework with more genres in order to assess that they properly characterize different styles of music.

1.5.4 Classification using audio and symbolic data

Although the research in style classification has been traditionally divided in the audio and symbolic domains, some recent works have explored the combination of both domains. In this section some of these works are

presented, in order to illustrate how the combination of both methods can help to improve the results based on just one of these two domains.

Lidy et al. (2007) used a combination of features extracted from audio and symbolic data. For this, a polyphonic transcription algorithm was used in order to obtain a symbolic representation from a set of audio files. This symbolic representation was described using an extended set of features based on those presented in (Ponce de León and Iñesta, 2007), which were then combined with a set of rhythmic and timbral features extracted from the audio signal. In the experiments a support vector machine classifier and three different audio corpora of musical genres were used. These experiments showed that, although there was not a big difference in the results, the combination of audio and symbolic features led to better results than using audio features only.

In (Cataltepe et al., 2007) the approach was just the opposite. In this work the MIDI data set used in (McKay and Fujinaga, 2004) was synthesized into audio, and then a combination of k-nearest neighbor and linear discriminant classifiers was used to perform genre classification, using symbolic and audio features separately. A total of twelve different classifiers were used, with different combinations of classification techniques and feature sets, and their decisions were combined using a weighted majority scheme. As in the previous work, this method led to a small improvement over the individual classifiers using just symbolic or audio features on the root nodes of the genre hierarchy. However, in the leaf genres the improvement was only achieved over the symbolic classifier, obtaining the same result than the audio-only approach.

Finally, McKay and Fujinaga (2008) followed a more sophisticaded approach, by combining cultural features extracted from the web in addition to symbolic and audio features extracted from MIDI and audio files respectively. Several experiments were performed using all possible combinations of these feature sets, in order to test the accuracy gain obtained when aggregating different kinds of information. Significant improvements were obtained using various combinations of features, reaching a 13.7% gain when using all three feature sets over the classifiers using just one of them.

1.6 Objectives and the approach in this thesis

The main goal of this thesis is to study the amount of stylisic information contained in the score of musical pieces. For this purpose, two style classification tasks will be carried out using a supervised approach: genre classification and composer style modeling. The underlying hypothesis is that music, as a human way of communication, has a structure similar to natural language, and thus it can be studied using tools borrowed from the natural language processing field used traditionally in text classification

tasks. These tools will be used to extract the stylistic traits corresponding to a particular set of music pieces from several genres and composers, using labeled corpora of symbolic music files. Two different sources of information will be explored: melody and harmony.

When using melodic sequences, just the information provided by the pitch and duration of the notes will be used. As seen in previous works, there seems to be enough information in melodic sequences so as to determine the style of a piece. The research presented in this thesis is specially influenced by the conclusions reached by Cruz-Alcázar and Vidal in (Cruz-Alcázar, 2004; Cruz-Alcázar and Vidal, 2008), where the authors achieved excellent results using n-gram models — a language modeling technique — on two corpora with a few number of genres. Language modeling has been also selected for the experiments in this thesis because it allows to study the paralelism between music and natural language, studying the ability of language models built from a training corpus to predict the style of new songs. In addition to n-gram models, another technique frequently used with success in text classification tasks, the naïve Bayes classifier, has been chosen for these experiments. This method, under certain circumstances, is equivalent to a unigram language model.

Few works have studied how harmonic information can help to distinguish musical styles and, to the best of the author's knowledge, this is the first work where a ground truth of music pieces annotated with chord progressions is used in this task. Tonal harmony has been chosen as a feature to classify music in genres because there seems to be a strong connection between musical genre and the use of different chord progressions. In (Piston, 1987, chapter 29), the evolution of the harmony practice from the early Baroque period to the 20th century music is depicted. Some rules that were almost forbidden in a period have been accepted afterwards. Those rules include contrapunctual rules, musical form, and harmony issues like tonality modulations and valid chord progressions. Furthermore, it is well known that pop-rock tunes mainly follow the classical tonic–subdominant–dominant chord sequence, whereas jazz harmony books propose also different series of chord progressions as a standard (Herrera, 1998). Therefore, in each musical period or genre there is general agreement on the harmonic framework used to compose music, and works created by that time should move through those chord sequences, or at least, should not use progressions that were not valid.

Finally, as a secondary objective, the feasibility of using these techniques to work with audio files will be studied, using state of the art audio transcription algorithms to obtain melodic and harmonic sequences from audio music files.

These objectives are developed in the following chapters, which are organized as follows:

Chapter 2 describes the data used in the experiments and the encoding formats used for transforming symbolic melodic and harmonic sequences, in order to be used as the input for pattern recognition algorithms.

Chapter 3 explains the supervised pattern recognition algorithms used for the classification of musical styles, and a method for combining the decisions of different classifiers.

Chapter 4 presents the experiments on musical genre classification using melodic sequences.

Chapter 5 presents the experiments on musical genre classification using harmonic sequences and the combination of both melodic and harmonic approaches using a combination of classifiers.

Chapter 6 presents the experiments on composer style classification, and how these methods can be applied to an authorship attribution problem.

Chapter 7 summarizes the main contributions of this thesis and outlines some future research lines.

2 Experimental data

One of the main problems in the field of Music Information Retrieval (MIR) is the availability of data. In contrast to other disciplines such as Natural Language Processing, where many training resources are freely available for research purposes, there is not any standard data set for testing and comparing one's results with those reported by fellow researchers in this field. As it was discussed in Chapter 1, many authors have developed their own data sets, each one formed by files gathered from different sources and encoded in different formats. The design of these corpora is often guided not by the purpose of the task, but for the availability of data. This lack of a common benchmark has the drawback that the results reported in these works are hardly comparable or replicable.

Six different data sets have been used in this work: four in the genre classification task and two in the composer style and artist identification problem. In order to compare the results obtained in genre classification, the corpora used in (Cruz-Alcázar and Vidal, 2008) were requested to the authors, who kindly shared them with my research group at the University of Alicante. Another corpus used in previous works in my research group, built by Ponce de León and Iñesta (2007), was used, and the fourth corpus used in this task was built during the development of this work. Finally, the two corpora used in (Backer and van Kranenburg, 2005; van Kranenburg, 2006) were selected for the composer style modeling task. In the following section these data sets are described in detail, and the encoding methods used for melodic and harmonic sequences are explained in Sections 2.2 and 2.3.

For reading the following sections some knowledge of the MIDI standard is assumed. The reader can refer to (Selfridge-Field, 1997) and the MIDI Manufacturers Association web site at `http://www.midi.org` for further information on this standard.

2.1 Corpora

In the first stage of this work, one of the corpora used by Cruz-Alcázar and Vidal (2008) and that by Ponce de León and Iñesta (2007) were used. All of

them are sets of files encoded in MIDI format with the melody in a separate track, and were created for genre classification tasks. They were used for a preliminary evaluation of the methods presented in the next chapter and were very useful for guiding the posterior research, but soon their limitations came to light.

The main handicap in the corpus by Ponce de León and Iñesta (2007) is the sample size. It is very rich in the number of different composers, but it has a limited number of files, and pattern recognition techniques usually need larger amounts of data for building accurate models. The corpora by Cruz-Alcázar and Vidal (2008), on the other hand, are bigger in number of files, but lack the richness in their composition that can be otherwise found in the first one. Anyway, the advantages of working with other researcher's data is obvious: not only it allows to compare the results obtained with different methods, but it also grants one with the confidence that the data have been revised and validated by other peope. This is specially important when handling musical data.

One of the goals of this thesis is to go a step further in genre classification by tackling a more complex problem and also by exploring new sources of information. For this reason a new data set of music files was built from scratch. It is a compilation of melodic and harmonic sequences, with nine genres covering a wide music domain. It will be depicted in detail later in this section. The main advantage of this corpus is that, to the best of the author's knowledge, it is the first one that gathers together a ground truth of both melody and harmony for the same music pieces. It is also big enough to draw interesting conclusions from the experiments performed on it.

Building a data set of music files categorized in genres is far from being an easy task. The main conflictive aspect is the reliability of the data themselves and, as important as that, the accurateness of the metadata that can be extracted from the music files. Altough these metadata can be very useful in categorizing the pieces, they must be revised carefully and all the compiled songs must be listened to in order to assess their reliability. Fortunately, all this work was done with the help and advise of music experts.

For the other task tackled in this thesis, the composer style modeling problem, two more data sets have been gathered. They are the ones used in (Backer and van Kranenburg, 2005; van Kranenburg, 2006), and can be downloaded from the Internet.

The following sections describe the details of each individual data set used in this thesis.

2.1.1 Corpus *Cruz-3-genres*

This is one of the corpora used in (Cruz-Alcázar and Vidal, 2008), and is made up of 300 monophonic samples taken from MIDI files belonging to 3 musical genres: Gregorian (Middle Ages), sacred music by J. S.

Bach (Baroque), and Scott Joplin's Ragtimes for piano (beginning of 20^{th} century). The files in this corpus contain *musical phrases*, i.e. representative excerpts taken from the pieces, with an average length of 40 notes. All of them were step-by-step sequenced in order to keep the duration of the notes exactly as they appear in the score.

These characteristics make this corpus a good starting point for our task, because the genres selected are well defined and very differentiated from each other, and the scores are free of the inacuracies or *noise* that are usually introduced when sequencing MIDI files with an instrument.

2.1.2 Corpus *Cruz-4-genres*

This is the second corpus used in (Cruz-Alcázar and Vidal, 2008). It was designed to make the classification task more difficult, as it contains four musical genres, two of them being very close. The genres selected are Gregorian, Domenico Scarlatti's harpsicord sonatas, and two Celtic subgenres called Jigs and Reels. It contains 400 whole musical pieces, 100 per genre, with an average length of 550 notes.

The files in this corpus were downloaded from the Internet, which makes the task more realistic, as they can contain interpretation errors. The melodies in this corpus, as in the previous one, are all monophonic.

2.1.3 Corpus *Ponce-2-genres*

This is the corpus used in (Ponce de León and Iñesta, 2007). This corpus is a set of MIDI files from *Jazz* and *Classical* music collected from different web sources, without any preprocessing. The melodies were real-time sequenced by musicians, without quantization. The corpus is made up of 110 MIDI files, 45 of them being classical music and 65 being jazz music. All of them have several tracks — one per instrument — and the one containing the melody has been hand-labeled.

The length of the corpus is 9,966 bars (39,864 beats). Classical melody samples were taken from works by Mozart, Bach, Schubert, Chopin, Grieg, Vivaldi, Schumann, Brahms, Beethoven, Dvorak, Haendel, Paganini, and Mendelssohn. Jazz music samples are standard tunes from a variety of well known jazz authors including Charlie Parker, Duke Ellington, Bill Evans, and Miles Davis, among others.

The main difference of this corpus regarding the previous two is the variance of styles whithin each genre. Each class is a compilation of songs from different authors and subgenres, while in the others each genre is represented by just one author, except for gregorian and celtic music, but for those genres one can also expect less variance in style than in this corpus. This way it can be tested if the language modeling techniques presented in

this thesis are capable of modeling genres, with their own variety of styles, instead of modeling the style of one single composer.

2.1.4 Corpus *Perez-9-genres*

This corpus was specially designed for the experiments in this thesis, and contains both melodic and harmonic information (including tonality). Harmony has been obtained from files encoded in the format of the PG Music software named Band in a Box (aka BIAB)[1], and then converted into Musical MIDI Accompaniment format[2] (MMA) using the *biabconverter* program that can be found in the MMA webpage. The use of sequences made with this software permits us to have a reliable ground truth for the chords, because it is provided with a set of files from factory structured by genres. Anyway, they have been double-checked by experts. The melodies in the corpus are contained in MIDI files, and were all human-sequenced. These MIDI files contain also a number of accompaniment tracks, rendered from the chord sequences using the BIAB software, using style rules and patterns depending on the style of the songs. The classical corpus was extracted from the *classfake* folders in the BIAB installation. The rest of the files were obtained from links found at the Internet[3]. The complete set of songs is listed in Appendix A.

Music files from three "domains" have been utilized: popular, jazz, and academic music. Popular music data have been separated into three subgenres: *pop*, *blues*, and *celtic* (mainly Irish jigs and reels). For jazz, three styles have been established: a *pre-bop* class grouping swing, early, and Broadway tunes, *bop* standards, and *bossanovas* as a representative of latin jazz. Finally, academic music has been categorized according to historic periods: *baroque*, *classicism*, and *romanticism*. This hierarchical structure allows us to study our models at different levels, either at the first level with three broad genres, or at the second level with all nine subgenres, making the task more complex.

All these categories have been defined with the help and advice of music experts that have also collaborated in the task of assigning meta-data tags to the files and rejecting outliers in order to have a reliable ground truth for the experiments. For example, in academic music we can find authors, like Beethoven, that lived and composed in two different historic periods. Their works were carefully selected and assigned to the correct genre according to musicological criteria. In the case of popular music, blues and pop files were selected trying to keep both classes as separate as possible in terms of style, rejecting any song that could fall in between, as it was the case of many rock tunes.

[1] http://www.pgmusic.com
[2] http://www.mellowood.ca/mma/
[3] http://www.alisdair.com/gearsoftware/biablinks.html

The number of files eventually used for each genre is displayed in Table 2.1. The total amount of pieces was 856, providing 47 and a half hours of music data.

Academic	235	**Jazz**	338	**Popular**	283
Baroque	56	Pre-bop	178	Blues	84
Classical	50	Bop	94	Pop	100
Romanticism	129	Bossanova	66	Celtic	99

Table 2.1: Number of files per genre and subgenre in corpus *Perez-9-genres*.

One advantage of this corpus is that it allows to perform experiments at different levels of the genre hierarchy. Classification using this corpus can be seen as a 9-class problem, using all the subgenres in the corpus, or as a 3-class problem if classification is done between the three parent genres.

2.1.5 Corpus *Kranenburg-5-styles*

This corpus is made up of files from five different composers: Bach, Telemann, Handel, Haydn, and Mozart. It was built by van Kranenburg and Backer (2004) for a composer style classification task, and can be downloaded from the CCARH[4] web site in the Humdrum **kern format (Selfridge-Field, 1997). The list of utilized files can be found at the web site of the author[5]. All the files are polyphonic, containing several voices splitted in separate tracks.

As the melody encoding software used in this thesis only works with standard MIDI files (see Section 2.2), the corpus has been transformed using the *hum2mid* tool from the Humdrum toolkit[6]. Unfortunately, due to conversion errors, only 274 files out of the original set of 306 could be used. They can be grouped as follows:

- J. S. Bach: 28 cantata movements.
- J. S. Bach: 30 fugues from "The Well-Tempered Clavier".
- J. S. Bach: 11 movements from "The Art of Fugue".
- J. S. Bach: 6 movements from the violin concerts.
- G. F. Handel: 37 movements from the Concerti Grossi, op. 6.
- G. F. Handel: 14 movements from trio sonatas, op. 2 and op. 5.

[4] http://www.ccarh.org
[5] http://www.musical-style-recognition.net
[6] http://extras.humdrum.net/man/hum2mid/

- G. Ph. Telemann: 30 movements from the "Fortsetzung des Harmonischen Gottestdienstes".
- G. Ph. Telemann: 23 movements from the "Musique de table".
- F. J. Haydn: 49 movements from the string quartets.
- W. A. Mozart: 46 movements from the string quartets.

2.1.6 Corpus *Kranenburg-fugues*

This is the corpus used in (van Kranenburg, 2006) for studying the authorship of several disputed fugues from the catalog of J. S. Bach. Along with those pieces, it contains some fugues of undisputed authorship from J. S. Bach and the three other candidate composers proposed in the literature as the real authors of the pieces: W. F. Bach (J. S. Bach's son), J. L. Krebs (J. S. Bach's pupil), and J. P. Kellner (a copyist of Bach's organ compositions).

As the files in corpus *Kranenburg-5-styles*, they can be downloaded in **kern format from the web page of the author. The pieces used are the following:

- J. S. Bach (1685–1750): BWV 535a/2, 535/2, 538/2, 540/2, 541/2, 542/2, 543/2, 545/2, 547/2.
- J. L. Krebs (1713–1780): Fugue in C minor (I, 2), E major (I, 5), F minor (I, 6), G major (I, 8), F major (II, 13), F minor (II, 14), F minor (II, 15), B flat major (II, 19).
- W. F. Bach (1710–1784): Fk 33, 36, 37, Add. 211/1, Add. 211/2.
- J. P. Kellner (1705–1772): O08:01, O08:06, O08:07, O08[C], O08[F], O10:02.
- Disputed fugues: BWV 534/2, 536/2, 537/2, 555/2, 557/2, 558/2, 559/2, 560/2, 565/2.

When numbering */2 is used, it means that only the second movement of a prelude-fugue pair has been used. Fugues 555/2, 557/2-560/2 are in fact part of a collection of 8 pieces, but the author did not use the remaining three because they are small compositions (less than 30 bars). In order to compare the results in this thesis with those of van Kranenburg, the same set of files will be used.

2.2 Melody encoding

Since a text categorization approach will be used, there is a need to find an appropriate encoding, something like *music words*, that captures relevant information of the data and is suitable for that kind of algorithms to be applied.

As it was discussed in Chapter 1, the representation methods that obtained the best results in other works are those that combine pitch and note durations, using relative measures. Following those conclusions, the encoding used in this work has been inspired by the encoding method proposed in (Doraisamy and Rüger, 2003). This method makes use of pitch intervals and inter-onset time ratios (IOR) (see next section) to build series of symbols of a given length.

As in (Cruz-Alcázar, 2004), both coupled and decoupled encodings have been considered. Using the coupled encoding, intervals and IOR are encoded together, in a similar way it is done in music scores, where each symbol gathers together both pitch and duration information. On the other hand, this encoding establishes a strong relationship between pitch and duration. In order to alleviate this relationship the decoupled encoding is proposed, encoding intervals and IOR as separate symbols.

2.2.1 Coupled encoding

The encoding of MIDI files into a sequence of coupled intervals and IOR is performed through the following steps:

1. The melody track is extracted from the MIDI file and, if necessary, is converted into monophonic.

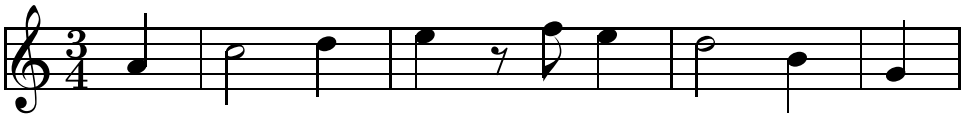
2. A sliding window is used to extract subsequences of notes, in a way similar to n-gram techniques.

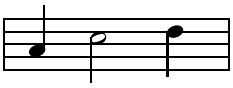
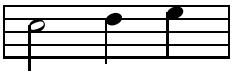
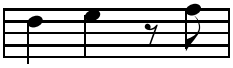
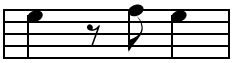
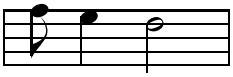
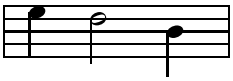
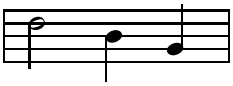
3. Each subsequence of notes is encoded using two different mapping functions for pitch intervals and IOR, respectively.

This way, each melody is converted into a sequence of *musical words*. In the following sections these steps are explained in detail, along with other considerations that affect the encoding. In order to illustrate the whole process, Figure 2.1 shows the encoding of a sample melody.

Melody extraction

The first step in the encoding process is to extract the melody from the MIDI file. For this reason, all the files in the corpora have been hand-labeled, selecting the track that contains the melody. Then, the list of MIDI events for each melody track is obtained and analyzed in search of polyphonic



window content	MIDI pitch & duration	interval & IOR	3-word	extended 3-word
	(69,120) (72,240) (74,120)	(+3,2) (+2,1/2)	CFB	CFBf
	(72,240) (74,120) (76,120)	(+2,1/2) (+2,1)	BfB	BfBZ
	(74,120) (76,180) (77,60)	(+2,3/2) (+1,1/3)	BDA	BDAh
	(76,180) (77,60) (76,120)	(+1,1/3) (-1,2)	Aha	AhaF
	(77,60) (76,120) (74,240)	(-1,2) (-2,2)	aFb	aFbF
	(76,120) (74,240) (71,120)	(-2,2) (-3,1/2)	bFc	bFcf
	(74,240) (71,120) (67,120)	(-3,1/2) (-4,1)	cfd	cfdZ

3-word encoding: CFB BfB BDA Aha aFb bFc cfd
extended 3-word encoding: CFBf BfBZ BDAh AhaF aFbF bFcf cfdZ

Figure 2.1: Example of the 3-word encoding of a MIDI file with 120 ticks per beat resolution. Sequences of pairs (*MIDI pitch, duration in ticks*) are extracted using a window of length 3. Then, intervals and IOR within the window are calculated and finally are encoded using the encoding scheme.

passages. Recall that in Chapter 1 melody was defined as a monophonic sequence of notes. Therefore when a polyphonic melody track is found, it is converted into monophonic by applying a polyphony reduction algorithm known as Skyline (Uitdenbogerd and Zobel, 1999). This algorithm is based on music perception research that suggests that, when there are multiple notes sounding at the same time, the notes with highest pitch are often perceived as the melody.

Once the melody track is monophonic, the pitch and duration for each note is obtained, and this sequence is feeded to the next step.

n-word extraction

Next, the melody is divided into n-note windows. For each window, a sequence of intervals and duration ratios is obtained, calculated using Equations (2.1) and (2.2) respectively.

$$I_i = Pitch_{i+1} - Pitch_i \qquad (i = 1, \ldots, n-1) \tag{2.1}$$

$$R_i = \frac{Onset_{i+2} - Onset_{i+1}}{Onset_{i+1} - Onset_i} \qquad (i = 1, \ldots, n-2) \tag{2.2}$$

where pitches are taken directly from the *note on* events, and the durations are computed as the time interval measured in ticks between the onset of one note and that of the next (inter-onset interval, IOI), ignoring intermediate rests. This measure is preferred over the actual duration of the notes (calculated as the time shift between *note off* and *note on* events), as it corresponds to the way rhythm is perceived by people (Parncutt and Drake, 2001). Then, each n-word is defined as a sequence of symbols:

$$[I_1 \; R_1 \; \ldots \; I_{n-2} \; R_{n-2} \; I_{n-1}] \tag{2.3}$$

Using this format, the encoding of a melody using a window of length 2 generates a sequence of pitch intervals, with no rhythm information. In order to test the influence of rhythm the previous encoding has been extended by adding the IOR of the last two notes in the window:

$$[I_1 \; R_1 \; \ldots \; I_{n-2} \; R_{n-2} \; I_{n-1} \; R_{n-1}] \tag{2.4}$$

This will be studied empirically in chapter 4. In this case, the duration of the last note is taken from the MIDI file, instead of looking at the onset of the next note ($Onset_{i+2} \equiv Offset_{i+1}$), because it falls outside the window.

n-words coding

The n-word values obtained in the previous step are mapped into sequences of alphanumeric characters (see Figure 2.1), that will be named n-words according to the equivalence we want to establish between melody and

text. Since the number of ASCII printable characters is lower than all possible intervals (26 characters against 255 intervals if we consider all the differences in pitch we can find using MIDI pitches), there is a need for a mapping function which assigns a character to each interval, using the same character for several different intervals. On the other hand, Downie's studies on melodic n-grams (Downie, 1999) show that the majority of intervals in a melody fall in the range of one octave ($[-12, +12]$ halftones). Thus, the best solution would be to assign a different character to each interval in this range (linear mapping) and to distribute the rest non-linearly amongst the set of higher intervals. This is done by establishing a non-linear mapping through a hyperbolic tangent with the following function:

$$C(I) = \text{int}(27 \cdot \text{tgh}(I/24)) \tag{2.5}$$

The values obtained with this function are mapped into ASCII characters. The direction of the intervals is encoded by assigning lowercase letters to negative intervals, uppercase letters to positive intervals and character 0 (zero) for the unison interval. Note that with this function a unique symbol is assigned to each interval in the range $[-12, +12]$, and the rest of interval values are binned to reduce the number of symbols in the vocabulary, as it can be seen in Figure 2.2. Note that all intervals larger than 4 octaves are mapped to the same character ('Z' if ascending, 'z' if descending).

For the IOR the mapping is performed as shown in Table 2.2. The IOR values have been selected by observing the most frequent proportions of note durations in a big corpus of melodies (Doraisamy and Rüger, 2003). Intermediate IOR values (as for example $1 < IOR < 6/5$) are encoded by rounding to the nearest value.

This way to encode durations has the additional advantage of performing an adaptative quantization of the original MIDI depending on the length of the pair of notes involved. Both mappings have also the property of imposing limits to the permitted ranges for both intervals and IOR.

IOR value	1	$6/5$	$5/4$	$4/3$	$3/2$	$5/3$	2	$5/2$	3	4	$> 4,5$
Symbol	Z	A	B	C	D	E	F	G	H	I	Y
IOR value		$5/6$	$4/5$	$3/4$	$2/3$	$3/5$	$1/2$	$2/5$	$1/3$	$1/4$	$< 1/4,5$
Symbol		a	b	c	d	e	f	g	h	i	y

Table 2.2: Symbol mapping for IOR values.

The encoding alphabet consists of 53 symbols for intervals and 21 for IOR. In order to illustrate the distribution of codes for both styles, histograms of intervals and IOR are displayed in Figures 2.3 and 2.4. Note the different frequencies for each style, that are in the basis of the recognition system. For the sake of clarity, histograms for corpus *Perez-9-genres* have

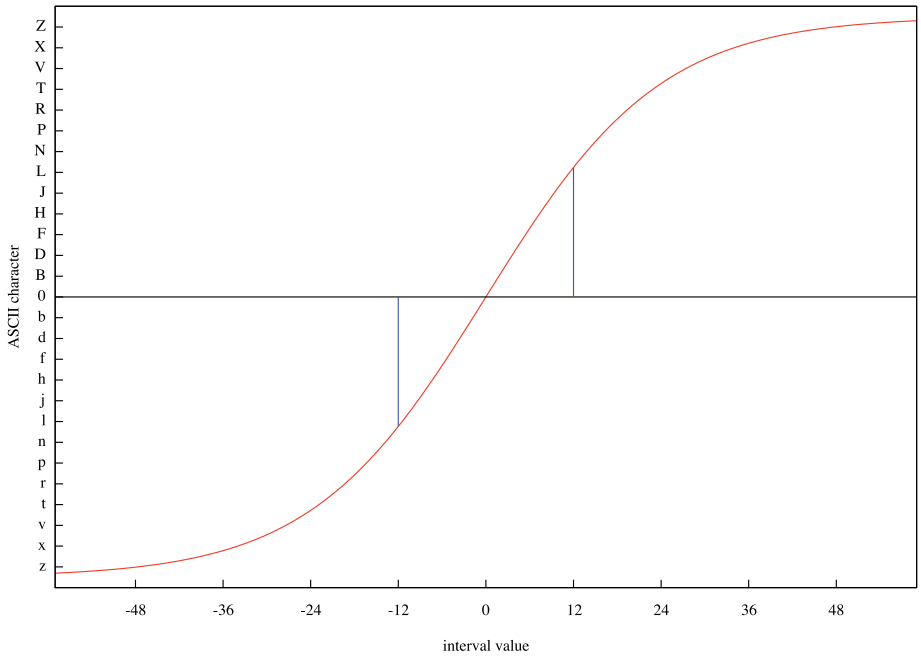


Figure 2.2: Mapping function for pitch interval values.

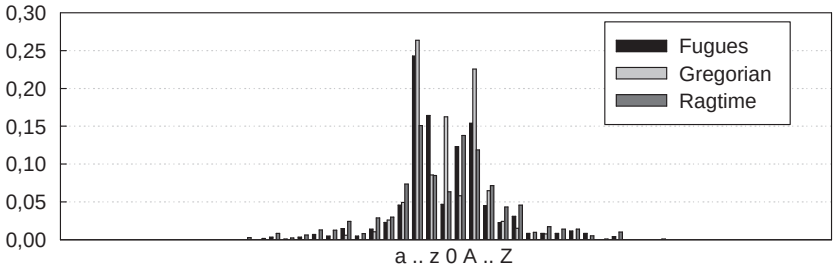
been only plotted using the three big domains: *academic*, *jazz*, and *popular* music.

In order to better illustrate the relationships between the 9 subclasses in corpus *Perez-9-genres*, the similarity between the vectors of symbol frequencies in each class has been computed. This similarity can be measured as the cosine of the angle formed by two vectors (Manning and Schütze, 1999, chap. 15), and is computed with Equation 2.6:

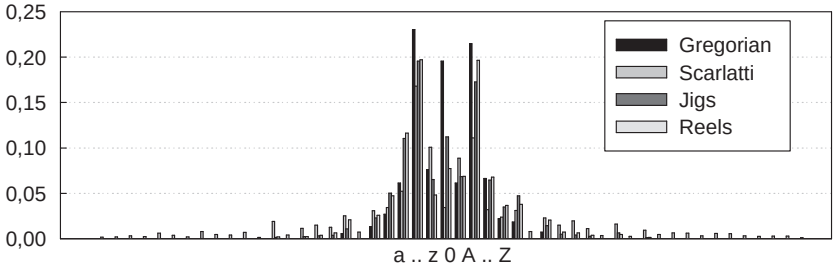
$$cos(\vec{p},\vec{q}) = \frac{\sum_{i=1}^{n} p_i q_i}{\sqrt{\sum_{i=1}^{n} p_i^2}\sqrt{\sum_{i=1}^{n} q_i^2}} \tag{2.6}$$

Figure 2.5 shows the similarities between all subgenres using a greyscale, where black color means that the distribution of symbols in both classes are identical ($cos(\vec{p},\vec{q}) = 1$). Looking at these similarities can help to make an approximate idea of the influence of both pitch and rhythm on the separability between classes.

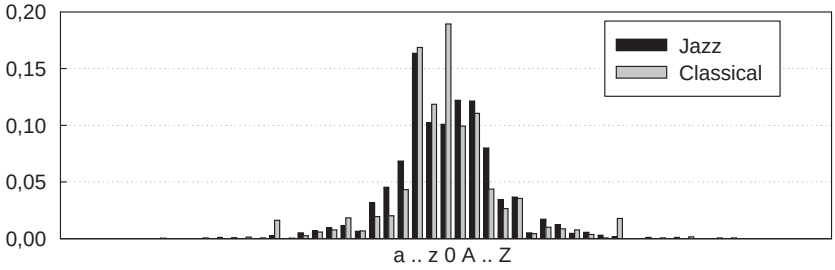
An analysis of the distribution of rhythm symbols in Figure 2.5b shows some interesting relationships between genres. As it can be seen, the strongest relations occur between genres belonging to the same domain, as in the case of all academic subgenres, *pre-bop* and *bop*, and also between *blues* and *pop* music. This is reasonable, as related styles of music usually share the



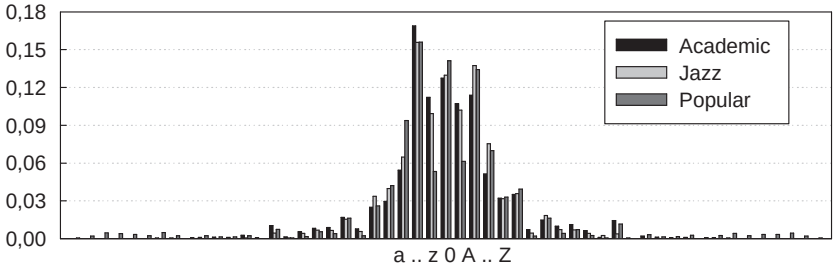
(a) Corpus *Cruz-3-genres*



(b) Corpus *Cruz-4-genres*

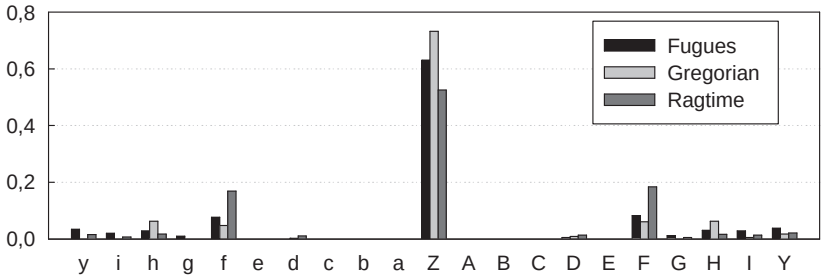


(c) Corpus *Ponce-2-genres*

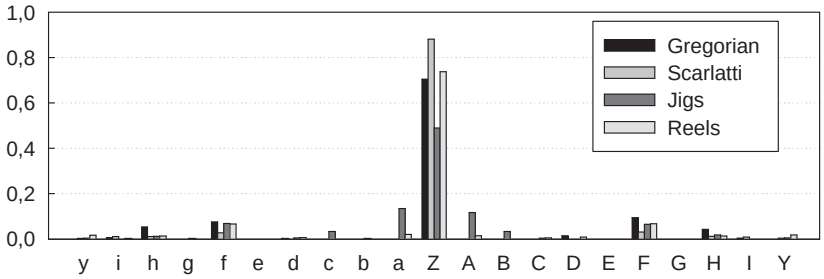


(d) Corpus *Perez-9-genres*

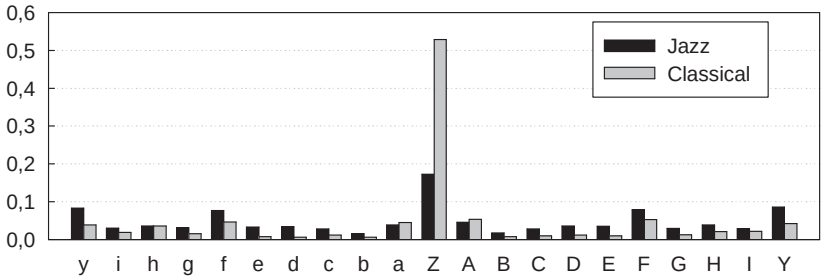
Figure 2.3: Histograms: normalized frequencies of intervals in the corpora. In the abcises, the coding letters are represented.



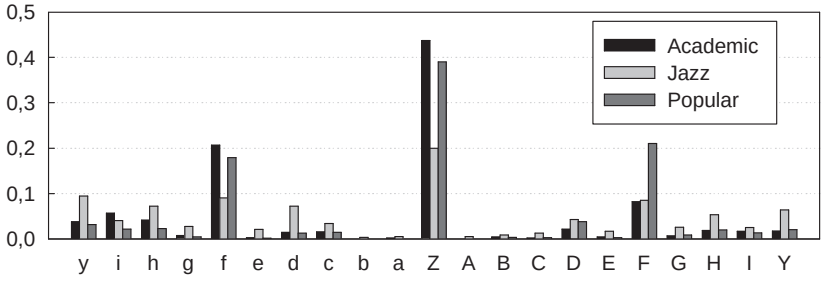
(a) Corpus *Cruz-3-genres*



(b) Corpus *Cruz-4-genres*

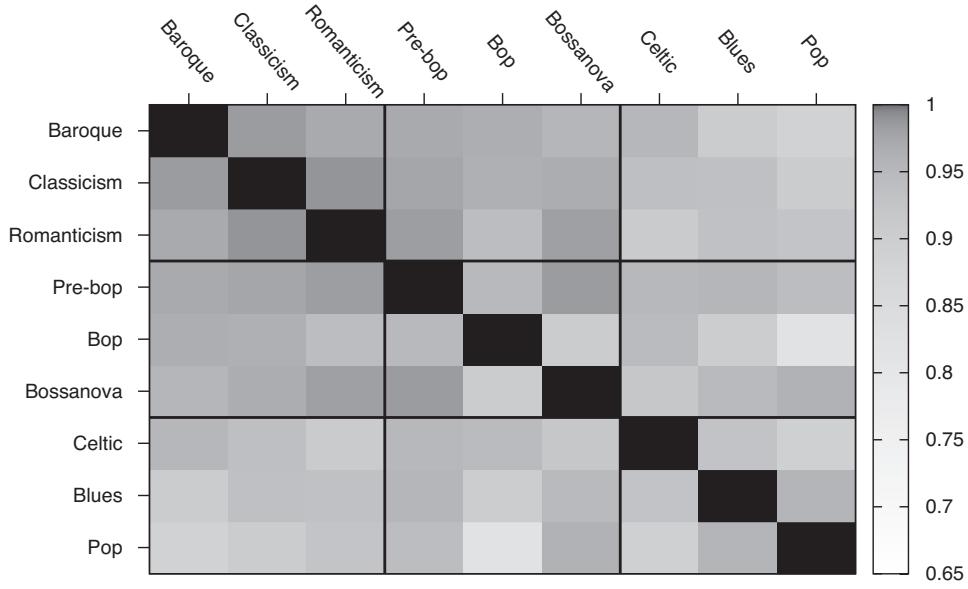


(c) Corpus *Ponce-2-genres*

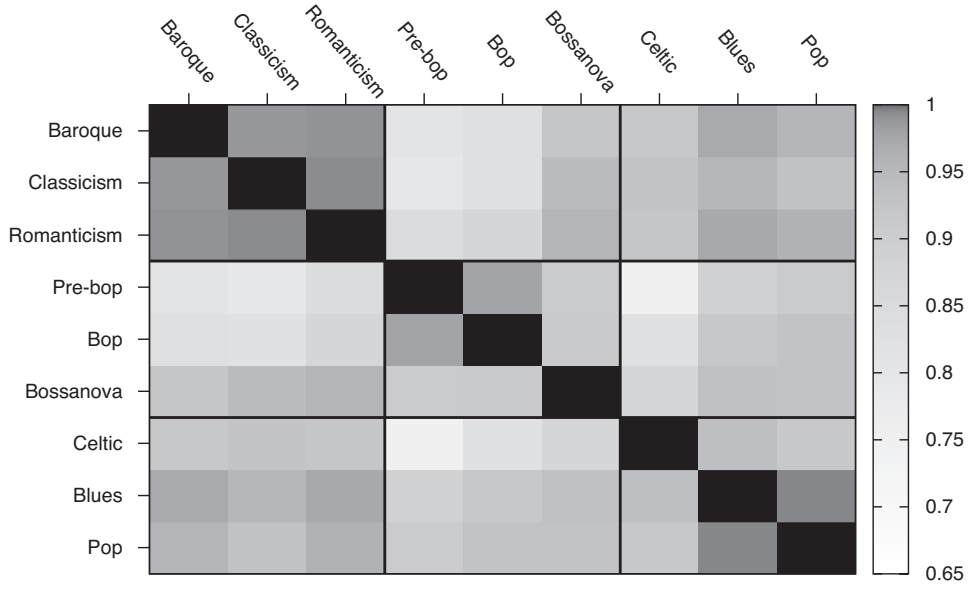


(d) Corpus *Perez-9-genres*

Figure 2.4: Histograms: normalized frequencies of inter-onset ratios in the corpora. In the abcises, the coding letters are represented.



(a) Pitch intervals



(b) IOR

Figure 2.5: Similarity measures between the subgenres in corpus *Perez-9-genres* using (top) pitch intervals, and (bottom) IOR. Darker colors mean that the distribution of symbols is more similar between genres.

same rhythmic patterns. There are also strong similarities between academic and popular music, while jazz subgenres seem to differentiate more from the others. This is also consistent with the distributions shown in Figure 2.4d. Pitch distributions, on the other hand, do not show any special relationship between genres.

Stop words filtering

The Implication-Realization (I-R) model (Narmour, 1990) establishes a continuity in how music is perceived in such a way that the expectations created when hearing an interval of two consequtive pitches in a melody are actually realized or denied when hearing a third pitch. The author developes a music perception theory around this fact that has provided very good results in music information retrieval tasks when it has been implemented (Grachten et al., 2005). The I-R model also establishes the concept of closure, that refers to the situation where a percieved interval does not imply a following interval, that is the inhibition of the listener's expectation. This kind of situations might commonly coincide with the finalization or completion of a musical whole, such as a phrase. Points of closure mark the boundaries of I-R structures and make neighbor notes not to be related by the listener perception in a sense of phrasal continuity. There are different situations for a closure point to be stablished, the most relevant for our analysis is the presence of long rests.

Long rests can be found in tracks belonging to accompaniment instruments that perform intermittently in a song, although relatively large rests can be also found whithin the main melody. Following the I-R model, consecutive notes separated by this kind of rests are not really related, so the next note can be considered as the beginning of a new melody. Then, a silence threshold is established, in a way that when a rest longer than this threshold is found, no words are generated across it. This restriction implies that, for each rest longer than this threshold, $n-1$ words less are encoded. In Chapter 4 different values for this threshold are studied.

Musical words length

In order to test the modeling performance for different n-word lengths, the range $n \in \{2,3,4,5,6,7\}$ has been established. Short n-words are less specific and provide more general information, while long n-words are more informative, but language models based upon them are much harder to train because the value of n has a great impact in the feature space size. As the value of n increases, the total number of possible symbols (i.e. combinations of pitch intervals and IOR) grows exponentially. From now on, the number of possible combinations will be referred to as *vocabulary* ($\mathcal{V}$). Using the encoding format in Equation 2.3 the vocabulary size is $|\mathcal{V}_n| = 53^{n-1} \times 21^{n-2}$,

and with the extended encoding described in Equation 2.4 the size of the vocabulary is $|\mathcal{V}_n| = (53 \times 21)^{n-1}$.

The maximum length for the n-words has been established at $n = 7$, as it has been reported that sequences of length around 5 to 7 notes can be used to uniquely identify a melody in MIR tasks (McNab et al., 1996). In our classification task it is desirable to build models able to generalize and recognize new melodies, not just to memorize the training data.

2.2.2 Decoupled encoding

The process for encoding melodies using the decoupled format for intervals and IOR is very similar to the one used for the coupled format. Once the melody is extracted from the MIDI file — applying polyphony reduction if necessary — it is encoded using Equation 2.5 for the intervals and the mapping shown in Table 2.2 for the IOR, with the difference that the symbols computed for each pair of notes are outputted separately. Also, the n-gramming step for building n-words is omitted, as it only made sense when symbols were grouped together. Thus, each pair of notes is encoded only once, and the resulting string is a sequence of alternate interval and IOR symbols. For example, the melody in Figure 2.1 would be encoded as follows:

decoupled encoding : C F B f B D A h a F b F c f d Z

2.3 Harmony encoding

In this thesis, harmony is represented as the sequence of chords found in a music piece. As it happens with melodies, it is necessary to transform these sequences into text using an appropriate encoding in order to be able for applying the classification methods described in Chapter 3.

Two different approaches have been considered. In the first one, sequences are encoded as chord progressions, where only chord changes are encoded. In the second, rhythm information is included, in order to represent some perceptual aspects of harmonic rhythm. Both approaches are described in the following sections.

2.3.1 Chord progressions

The encoding method chosen in this thesis for encoding chords is the standard musical notation, because it is the most common notation used by musicians in all styles. Despite its simplicity, it is a rich notation that provides all the information on the structure of the chords (see Section 1.4.3).

This notation allows to represent almost every combination of notes, and the number of all possible chords is rather high, so some simplifications were done in order to limit the vocabulary size. Thus, the chords found in

corpus *Perez-9-genres* (*raw chords*) were simplified to a smaller set using the most common 26 different extensions, and chord inversions were discarded because removing them does not affect the basic structure of the chord, just the ordering of its component notes.

In this corpus, all the files have annotated chords with a resolution of one beat. However, in real world tasks, harmonic information is hardly found. In those situations, chords can be obtained using different methods depending on the nature of the data available. If the score is given, a harmonic analysis can be performed in order to obtain the chord sequence. If dealing with audio data, a polyphonic transcription system can be used to extract the notes prior to the harmonic analysis. Another option is to use a chord extraction algorithm, able to detect the chords from an audio source. However, all of them are error-prone processes, and usually the chords that can be obtained are not as rich as the ones in this corpus.

In order to study how harmonic information could help in those situations, three reduced data sets were generated using different vocabularies. The first set was built by reducing all chords to its basic triad form, and the other two used the chord vocabulary that can be obtained using the audio to chord transcription algorithm described in (Gómez, 2006) — major and minor chord triads — and an extended version of it capable to recognize more chords (both versions of the algorithm will be used in Chapter 5). In summary, the following reductions were performed on corpus *Perez-9-genres*:

- *Full chords*: root note plus 26 extensions (major, 4, 7, 7+, 7♭5♯9, 7♭9♯11, 7♯11, 7♯9, 7sus♭9, 7alt, maj7, maj7♯5, 9♯11, maj9♯11, 11, 13♯11, 13alt, 13sus, m, m♯5, m6, m7, m7♭5, aug, dim, whole).
- *Triad chords*: all possible chord triads (major, ♭5, aug, dim, m, m♯5, sus).
- *4-note chords:* major and minor triads, plus some seventh chords (major, 7, maj7, m, m7).
- *Major-minor chords:* only major and minor chord triads.

The chords in a piece have different tonal functions depending on the underlying tonality, and therefore, a different meaning. While using just the name, Am is always the same chord, but using the degree it can be distinguished whether it is the first degree in the Am key or the minor sixth in the C major key. For example, the sequence Cm – B♭ – E♭ in Figure 2.6 should be read as i – VII – III for the C minor tonality but as vi – V – I for the E♭ major key. This second approach to represent chords (using their degrees instead of their names) has been also considered and the performance of both kinds of models will be studied comparatively. In Table 2.3 the vocabulary size obtained from the corpus when using each feature set is shown, and

Figure 2.7 shows an example of how a chord sequence is encoded using each feature set. Finally, in Appendix B the complete chord vocabulary is shown, along with the corresponding reductions for each chord.

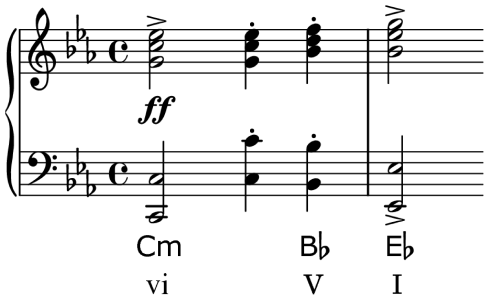


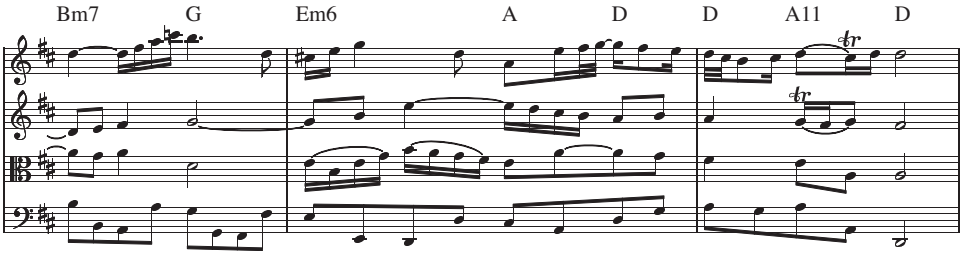
Figure 2.6: Fragment of "Der Freischtz" by Carl Maria von Weber, taken from Piston (1987).

2.3.2 Harmonic rhythm

Chord progressions are a common representation for harmonic sequences, but as they encode chord changes only, they lack information on the rhythm of the piece. However, rhythm is an important aspect of music, because musical pieces are often composed by repeating patterns that fit into the metrical structure (Paiement, 2008, chapter 5). Meter is perceived as a succession of strong and weak beats, and different effects can be achieved by placing chord changes at different positions in a bar. For example, placing steady changes in harmony at the first beat (strong) of every bar can induce a feeling of stability to the listener, while changing at weak positions (second or fourth beats in 4/4 meter) makes the opposite effect.

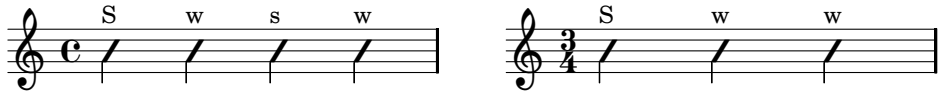
As it happens with chord progressions, the use of harmonic rhythm has changed over time, and different rhythmic patterns can be found in each musical period. Baroque and classical music usually followed regular rhythms, and in the Romantic period more complex patterns where introduced. In order to study how this information can help in the genre classification task, an encoding for the harmonic rhythm has been introduced to complement the chord progressions explained in the previous section. Thus, for each chord change in a chord progression, a new symbol is added indicating the type of beat where the change happens. Three beat categories have been considered: strong, semi-strong, and weak. Figure 2.8 shows where these beats are located in a bar.

One limitation of this encoding is that it does not account for chord durations, because only chord changes are included in the sequences. When the same chord is repeated in more than one bar, the total duration of



Full chord names:	Bm7	G	Em6	A	D	A11	D
Triad chord names:	Bm	G	Em	A	D	Asus	D
4-note chord names:	Bm7	G	Em	A	D	A7	D
3-note chord names:	Bm	G	Em	A	D	A	D
Full chord degrees:	vi7	IV	ii6	V	I	V11	I
Triad chord degrees:	vi	IV	ii	V	I	Vsus	I
4-note degrees:	vi7	IV	ii	V	I	V7	I
3-note degrees:	vi	IV	ii	V	I	V	I

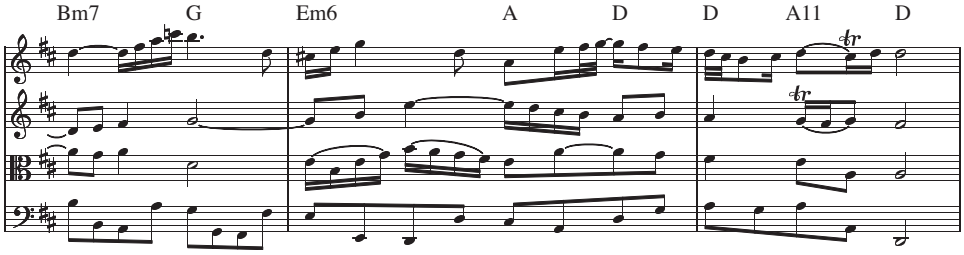
Figure 2.7: Extract from "Air on the G string" in D major by Johann Sebastian Bach, as it is encoded in each feature set. Score taken from the International Music Score Library Project (IMSLP).



(a) 4/4 meter (b) 3/4 meter

Figure 2.8: Position of the different types of beats within a bar: strong (S), semi-strong (s), and weak (w).

that chord is lost. In order to overcome this limitation, the chord placed at the first position of each bar is always outputted. This way, the length in bars of the chords is present in the resulting sequence. Figure 2.9 shows the encoding of a sample chord progression. Note that both coupled and decoupled versions of the encoding have been also considered.



decoupled : Bm7 S G s Em6 S A s D w D S A11 w D s
coupled : Bm7S Gs Em6S As Dw DS A11w Ds

Figure 2.9: Chord progression extracted from "Air on the G string" in D major by Johann Sebastian Bach, encoded using harmonic rhythm information.

	Full chords		Triads		4-note chords		major-minor chords	
	Degrees	Chords	Degrees	Chords	Degrees	Chords	Degrees	Chords
Baroque	79	72	42	41	48	44	25	24
Classical	58	56	39	37	38	36	24	22
Romanticism	120	106	68	61	55	49	26	24
Academic	127	114	69	62	58	52	26	24
Pre-bop	158	146	72	65	62	60	25	24
Bop	163	150	80	72	60	58	26	24
Bossanova	171	152	82	72	62	58	26	24
Jazz	238	218	95	83	64	60	26	24
Celtic	17	14	13	11	17	14	13	11
Blues	62	62	37	37	45	45	24	24
Pop	121	119	56	55	59	58	25	24
Popular	125	123	58	57	59	58	25	24
Total	**261**	**273**	**97**	**84**	**65**	**60**	**26**	**24**

Table 2.3: Vocabulary sizes for each feature set.

3 Methodology

In this chapter, the details of the pattern recognition techniques used in this thesis for the studied tasks are discussed. A supervised learning scheme has been adopted (Jain et al., 2000), i.e. all decisions are taken on the basis of a labeled set of training samples. For the classification task this means that the number of classes is determined by the training set, and classification consists in deciding, given a new test sample, which class it belongs to with highest probability. Similarly, in the author atribution task, a disputed music piece will be tested against a predefined set of candidate authors.

Music, as natural language, is a materialization of human thinking and a way of communication. Thus, one of the premises of this thesis is that standard Natural Language Processing techniques can be used for assigning categories to music data in a similar way texts are classified in a number of practical situations like spam filtering (Metsis et al., 2006), text retrieval (Lam et al., 1999) or machine translation (Juan and Vidal, 2002). For this reason, two techniques widely used in text categorization tasks have been selected: naïve Bayes and language modeling (n-grams). Both techniques allow the construction from examples of a statistical model for each genre. While n-grams make use of context information to compute the probabilities for each word progression, naïve Bayes computes the probability of each word on its own.

In this work, an analogy between music and text documents is established. This is achieved by using proper encoding methods that transform musical sequences — either melodic or harmonic — into text documents (see Chapter 2). These methods take a digital score as an input and map the musical symbols into ASCII characters. As a result, a sequence of *musical words* is obtained. Throughout this chapter, and for the sake of clarity, these musical words are referred to simply as words (w), and the sequence of words obtained from the encoding of a musical piece is considered a document.

3.1 Naïve Bayes

The naïve Bayes classifier (McCallum and Nigam, 1998) is a probabilistic classifier based on the *naïve Bayes assumption*. It assumes that all the

features (the words in a document, or the notes in a piece of music) are independent of each other, and also independent of the order they are generated. This assumption is obviously false in our task, because in music a single note or chord can create a very different effect to the listener depending on its context. In the case of melody classification, this assumption is partially overcome because the encoding method computes the relationships between consecutive notes (see Section 2.2). Moreover, when using the coupled encoding, a sliding window is used to extract words from groups of notes, in a way that each word shares some notes with its previous and following words, and thus they cannot be considered independent from each other. Anyway, despite its simplicity, it has been shown that naïve Bayes can obtain near optimal classification errors (Domingos and Pazzani, 1997).

In this framework, classification is performed as follows: given a set of classes $\mathcal{C} = \{c_1, c_2, \ldots, c_{|\mathcal{C}|}\}$ and a new document $\mathbf{x}$ whose class is unknown, it is assigned to the class $c_j \in \mathcal{C}$ with maximum a posteriori probability, in order to minimize the probability of error:

$$P(c_j|\mathbf{x}) = \frac{P(c_j)P(\mathbf{x}|c_j)}{P(\mathbf{x})} \quad , \tag{3.1}$$

where $P(c_j)$ is the a priori probability of class c_j, $P(\mathbf{x}|c_j)$ is the probability of $\mathbf{x}$ being generated by class c_j, and $P(\mathbf{x}) = \sum_{j=1}^{|\mathcal{C}|} P(c_j)P(\mathbf{x}|c_j)$.

The class-conditional probability of a document $P(\mathbf{x}|c_j)$ is given by the probability distribution of words in class c_j, that can be learned from a labeled training set of documents, $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{|\mathcal{X}|}\}$, using a supervised learning method. There are several methods for estimating those probabilities, each one based on different assumptions on how the words in a document are generated. Two of them have been considered: the multivariate Bernoulli and the multinomial models. Both of them make an estimation of the independent probabilities of words based on their relative frequencies in each class. Also, a finite mixture model of multivariate Bernoulli distributions (Carreira-Perpiñán and Renals, 2000; Juan and Vidal, 2002) has been considered in order to compensate the excessive simplicity of the Bernoulli model.

In these three models each document $\mathbf{x}$ is represented as a vector, where each component x_t codes statistical information about the presence of the word w_t in the piece. The set of possible words (vocabulary) is denoted as $\mathcal{V}$, and its size as $|\mathcal{V}|$. In the following subsections, the particularities of each model are discussed.

3.1.1 Multivariate Bernoulli model

In this model each document is represented as a binary vector $\mathbf{x} \in \{0,1\}^{|\mathcal{V}|}$, where each component x_t represents whether the word w_t appears at least

once in the coded document. Using this approach, each class follows a multivariate Bernoulli distribution:

$$P(\mathbf{x}|c_j) = \prod_{t=1}^{|\mathcal{V}|} x_t P(w_t|c_j) + (1 - x_t)(1 - P(w_t|c_j)) \tag{3.2}$$

where $P(w_t|c_j)$ are the class-conditional probabilities of each word in the vocabulary, and these are the parameters to be learned from the training set. Thus, a document can be seen as a set of independent Bernoulli experiments, one for each word in the vocabulary.

Bayes-optimal estimates for probabilities $P(w_t|c_j)$ can be easily calculated by counting the number of occurrences of each word in the corresponding class:

$$P(w_t|c_j) = \frac{1 + M_{tj}}{2 + M_j} \tag{3.3}$$

where M_{tj} is the number of songs in class c_j containing the word w_t, and M_j is the total number of songs in class c_j. Also, a Laplacean prior has been introduced in the equation above to smooth probabilities. Prior probabilities for classes $P(c_j)$ can be estimated from the training sample using a maximum likelihood estimate:

$$P(c_j) = \frac{M_j}{|\mathcal{X}|} \tag{3.4}$$

Classification of new songs is performed then using Equation (3.1), which is expanded using Equations (3.2) and (3.4).

3.1.2 Mixtures of multivariate Bernoulli distributions

Although the multivariate Bernoulli model is one of the most accurate classifiers for textual information, better results can be obtained by assuming that the probability distribution of words within a class follows a more complex distribution (Novovičová and Malík, 2002). This is usually done by using finite mixtures.

A finite mixture is a probability distribution formed by a number of components M which are combined together to model a complex probability function:

$$P(\mathbf{x}) = \sum_{m=1}^{M} \pi_m P(\mathbf{x}|\mathbf{p}_m) \tag{3.5}$$

where π_m are the mixing proportions, that must satisfy the restriction $\sum_{m=1}^{M} \pi_m = 1$; and $\mathbf{p}_m \in \{0,1\}^{|\mathcal{V}|}$ are the component prototypes. Since each class is modelled as a mixture of multivariate Bernoulli distributions, each component distribution $P(\mathbf{x}|\mathbf{p}_m)$ is calculated using Equation (3.2), substituting $P(w_t|c_j)$ with its corresponding value p_{mt}.

Learning a multivariate Bernoulli mixture model consists in finding the optimal estimates for parameters $\Theta = (\pi_1, \ldots, \pi_M, \mathbf{p}_1, \ldots, \mathbf{p}_M)^T$. This can

be achieved using the EM algorithm (Dempster et al., 1977), but to be applicable, EM algorithm requires that the problem be formulated as an incomplete-data problem. To do this, we can think of each sample document $\mathbf{x}$ as an incomplete vector, where $\mathbf{z} \in \{0,1\}^{|M|}$ is the *missing data* and indicates which component of the mixture the document belongs to (with 1 in the position corresponding to the component and zeros elsewhere). Then, the EM proceeds iteratively to find the parameters that maximize the log-likelihood of the complete data:

$$\mathcal{L}_C(\Theta|X,Z) = \sum_{i=1}^{|\mathcal{X}|}\sum_{m=1}^{M} z_{im}\left(\log \pi_m + \log P(\mathbf{x}_i|\mathbf{p}_m)\right) . \tag{3.6}$$

In each iteration, the EM algorithm performs two steps. The E-step computes the expected value of the missing data given the incomplete data and the current parameters. Each z_{im} is replaced by the posterior probability of $\mathbf{x}_i$ being generated by component m:

$$z_{im} = \frac{\pi_m P(\mathbf{x}_i|\mathbf{p}_m)}{\sum_{k=1}^{M}\pi_k P(\mathbf{x}_i|\mathbf{p}_k)} \qquad (i = 1,\ldots,|\mathcal{X}|;\ m = 1,\ldots,M) \tag{3.7}$$

The M-step updates then the maximum likelihood estimates for the parameters:

$$\pi_m = \frac{\sum_{i=1}^{n} z_{im}}{n} \qquad (m = 1,\ldots,M) \tag{3.8}$$

$$\mathbf{p}_m = \frac{\sum_{i=1}^{n} z_{im}\mathbf{x}_i}{\sum_{i=1}^{n} z_{im}} \qquad (m = 1,\ldots,M) \tag{3.9}$$

Once the optimal parameters have been obtained, classification can be then performed expanding Equation (3.1) using Equations (3.5) and (3.4).

A common problem when using the EM algorithm is the possibility of falling in a "pathological" point of the parameter space, where the log-likelihood tends to $+\infty$. However, in the case of multivariate Bernoulli distributions, the log-likelihood is upper-bounded (Carreira-Perpiñán and Renals, 2000), and convergence is guaranteed if the parameters are initialized in the range $]0,1[$.

Another difficult problem is to automatically find the optimal number of mixture components. Some techniques have been proposed to solve this problem (Jain et al., 2000), although in this work the simplest approach has been adopted. It consists in studying the behaviour of the algorithm for different number of components, testing several values as in previous works on automatic text classification (Juan and Vidal, 2002).

3.1.3 Multinomial model

The multinomial model, as distinct from the previous two models, takes into account word frequencies in each piece, rather than just the occurrence or

non-occurence of words. In this model each document is encoded as a vector $\mathbf{x} \in \mathbb{N}^{|\mathcal{V}|}$, where each component x_t represents the number of occurrences of the word w_t in the piece.

The probability that the whole series of words has been generated by a class c_j is the multinomial distribution, assuming that the document length in words, $\mathcal{L}(\mathbf{x})$, is class-independent (McCallum and Nigam, 1998):

$$P(\mathbf{x}|c_j) = P(\mathcal{L}(\mathbf{x}))\mathcal{L}(\mathbf{x})! \prod_{t=1}^{|\mathcal{V}|} \frac{P(w_t|c_j)^{x_t}}{x_t!} \tag{3.10}$$

Now, Bayes-optimal estimates for class-conditional word probabilities are:

$$P(w_t|c_j) = \frac{1 + N_{tj}}{|\mathcal{V}| + \sum_{k=1}^{|\mathcal{V}|} N_{kj}} \tag{3.11}$$

where N_{tj} is the sum of occurrences of word w_t in the songs in class c_j. Class prior probabilities are also calculated as for the Bernoulli model using Equation (3.4).

3.1.4 Feature selection

The methods explained above use a representation of musical pieces as a vector of symbols. A common practice in text classification is to reduce the dimensionality of those vectors before training the system, through a feature selection procedure. This process is useful to avoid overfitting to the training data when there are limited data samples and a large number of features, and also to increase the speed of the system. This is done by selecting the words that contribute the most to discriminate the class of a document, using a ranked list of words extracted from the training set. A widely used measure to build this ranking is the *average mutual information* (AMI) (Cover and Thomas, 1991), which gives a measure of how much information about a class is provided by each single code. Informally speaking, we can consider that a word is informative when it is very frequent in one class and less in the others.

For the Bernoulli models, the AMI is calculated between the class of a document and the absence or presence of a word in the document. We define C as a random variable over all classes, and F_t as a random variable over the absence or presence of word w_t in a document, F_t taking on values in $f_t \in \{0, 1\}$, where $f_t = 0$ indicates the absence of word w_t and $f_t = 1$ indicates its presence. The AMI is calculated for each w_t as[1]:

$$I(C; F_t) = \sum_{j=1}^{|C|} \sum_{f_t \in \{0,1\}} P(c_j, f_t) \log \frac{P(c_j, f_t)}{P(c_j)P(f_t)} \tag{3.12}$$

[1] The convention $0 \log 0 = 0$ is used, since $x \log x \to 0$ as $x \to 0$.

where $P(c_j)$ is the number of documents for class c_j divided by the total number of documents; $P(f_t)$ is the number of documents containing the word w_t divided by the total number of documents; and $P(c_j, f_t)$ is the number of documents in class c_j having a value f_t for word w_t divided by the total number of documents.

In the multinomial model, the AMI is calculated between the class of the document from which a word occurrence is drawn and a random variable over all the word occurrences, instead of documents. In this case, Equation (3.12) is also used, but $P(c_j)$ is the number of word occurrences appearing in documents in class c_j divided by the total number of word occurrences, $P(f_t)$ is the number of occurrences of the word w_t divided by the total number of word occurrences, and $P(c_j, f_t)$ is the number of occurrences of word w_t in documents with class label c_j, divided by the total number of word occurrences.

3.2 n-gram models

The methods explained above model the statistical properties of texts — or any sequence of symbols — as if they were a collection of independent words ("bag of words") but, in many applications, this is not really true. Usually, these sequences are structured in patterns and the probabilities of single words are also dependent on their context.

Better models can be built that reflect this reality using statistical language modeling techniques. A language model is a probability distribution that assumes that the probability of a sequence of L words $W = w_1^L$ is the product of the probability of each word dependent on its previous context:

$$p(W) = p(w_1) \prod_{i=2}^{L} p(w_i | w_1^{i-1}), \tag{3.13}$$

where the sequence w_1^{i-1} is the history h_i of w_i. As the value of i increases, the number of possible histories of w_i grows exponentially, so estimating the probabilities of such a model can be an arduous task, and maybe computationally unaffordable when dealing with long sequences.

In order to overcome this problem, language models are often approximated using n-gram models. An n-gram is a sequence of n words in which the first $n-1$ words are considered as the context. Thus, the estimated probability of a word w_i given a context is computed as $P(w_i | w_{i-n+1}^{w_{i-1}})$. These probabilities are the parameters to learn, and can be estimated by maximizing the likelihood over a training set of sequences (Camastra and Vinciarelli, 2008). This probability can be easily calculated by dividing the number of occurrences of the n-gram by the number of occurrences of its

context in the given data set:

$$P(w_i|w_{i-n+1}^{w_{i-1}}) = \frac{\mathcal{N}(w_{i-n+1}^{i})}{\mathcal{N}(w_{i-n+1}^{i-1})} \quad . \tag{3.14}$$

3.2.1 Using n-gram models as classifiers

Language models built from a training set of documents can be also used as classifiers. For this, given a new, previously unseen, sequence of words, classification is done by selecting the class most likely to have generated that sequence.

The first step is to learn a model for each class in the data set. Then, the probability that a new sequence $w = w_1^k$ has been generated by model c is:

$$P_c(w) = \prod_{i=1}^{k} P_c(w_i|w_{i-n+1}^{i-1}), \tag{3.15}$$

where $P_c(w_i|w_{i-n+1}^{w_{i-1}})$ are computed using Equation (3.14), but using just the data for the class c in the training set. Thus, a test sample can be classified by following the maximum likelihood criterion, by assigning the test sample the class $\hat{c}$ of the model that holds $\hat{c} = \arg\max_c P_c(w)$.

Another method to evaluate a language model built from a class c, is by measuring its perplexity given a test sample w_1^k:

$$PP_c(w_1^k) = \left[\frac{1}{P_c(w_1^k)}\right]^k = \sqrt[k]{\frac{1}{\prod_{i=1}^{k} P_c(w_i|w_{i-n+1}^{i-1})}} \tag{3.16}$$

Perplexity is strongly correlated with the probability of generating a sample shown in Equation (3.15), and it can be intuitively interpreted as how surprised is this model when a new sample is presented to it: the lower the perplexity, the higher the probability that this model has generated that sample. Using this measure, classification can be performed by selecting the class of the model with lower perplexity $\hat{c} = \arg\min_c PP_c(w)$.

In this work, building and evaluation of the language models has been performed using the CMU SLM Toolkit (Clarkson and Rosenfeld, 1997), a free software package for creating statistical language models. Using this software it is not possible to retrieve the probabilities of the sequences, so perplexity has been used when evaluating the models.

3.2.2 Parameter smoothing

Even when the training set is big enough to build a good language model, there can be situations where words can be found in a test sample that have not been seen previously. When such situation occurs, the probability of the

n-grams containing those words is zero, thus causing the probability of the whole sequence being zero by the application of Equation (3.15).

To avoid this problem, several *smoothing* techniques can be used. A common procedure is to use a *discounting method*, which substracts a small probability from the set of known words. This quantity is then shared out among all unseen words. There are several techniques to calculate the optimal amount of probability that must be taken off, and what percentage of it must receive every unseen word. Cruz-Alcázar (2004) tested some of these techniques in a genre classification task using melodic information, and the best results he obtained were using the *Witten-Bell* discounting method (Witten and Bell, 1991).

Another common method is the *interpolation* of n-gram models of different order (Manning and Schütze, 1999). This method builds all the possible order models for values of $m = 1, \ldots, n$. Using a 3-gram model, the probability of a sequence is calculated as:

$$P(w_n|w_{n-2}w_{n-1}) = \lambda_1 p(w_n|w_{n-2}w_{n-1}) + \lambda_2 p(w_n|w_{n-1}) + \lambda_3 p(w_n) \quad (3.17)$$

where $0 \leq \lambda_i \leq 1$ and $\sum_i^m \lambda_i = 1$. These parameters can be adjusted by hand, although they can be automatically adjusted using the Expectation Maximization (EM) algorithm (Manning and Schütze, 1999).

In this work, a combination of both techniques — interpolation of models and the Witten-Bell discounting method — have been used.

3.3 Classifier ensembles

An ensemble of classifiers is a combination of different classification techniques, so that the individual decisions of the classifiers are weighted in some manner in order to reach a final decision (Dietterich, 2000). The rationale behind this is that the results for different classification techniques may differ substantially, although they can be also used to combine the decision of the same classifier using different data sources.

Different approximations exist on how the weights are calculated. One simple approach is to compute the weights based on a confidence measure over all the individual decisions, so that classifiers with a high confidence receive higher weights (Conklin and Witten, 1995). This has the advantage that the weights are computed once the decisions are taken, and no previous training is needed. Unfortunately, it is not advisable to use such a technique when using naïve Bayes classifiers. The probability estimates built with these techniques, despite their high success rates in classification, are usually very inaccurate due to the strong independence assumptions, and they tend to be overconfident on their decisions (Tóth et al., 2005).

Another way to obtain the weights is by taking into account empirical evidence on how the individual classifiers performed in a similar task. This

way, the ensemble can be thought of as a *committee of experts*, where higher authority is assigned to the classifiers that performed better with a training set. Performance for a classifier C_k is measured by the number of errors in training (e_k), and the authorities (a_k) are calculated based on these errors. This technique was thouroghly tested in (Moreno-Seco et al., 2006) and has shown the property that the ensembles usually perform at least as well as the best single classifier, but avoiding the risk of choosing the wrong classifier for a particular data set, so their decisions are more robust (or less risky). From the different weighting strategies presented in that work, the two that showed better performance have been selected.

3.3.1 Best-worst weighted vote (BWWV)

In this ensemble, the best and the worst classifiers C_k in the ensemble are identified using their estimated accuracy. A maximum authority, $a_k = 1$, is assigned to the former and a null one, $a_k = 0$, to the latter, being equivalent to removing this classifier from the ensemble. The rest of classifiers are rated linearly between these extremes (see figure 3.1a). The values for a_k are calculated using the number of errors e_k as follows:

$$a_k = 1 - \frac{e_k - e_B}{e_W - e_B} \quad ,$$

where

$$e_B = \min_k\{e_k\} \quad \text{and} \quad e_W = \max_k\{e_k\}$$

3.3.2 Quadratic best-worst weighted vote (QBWWV)

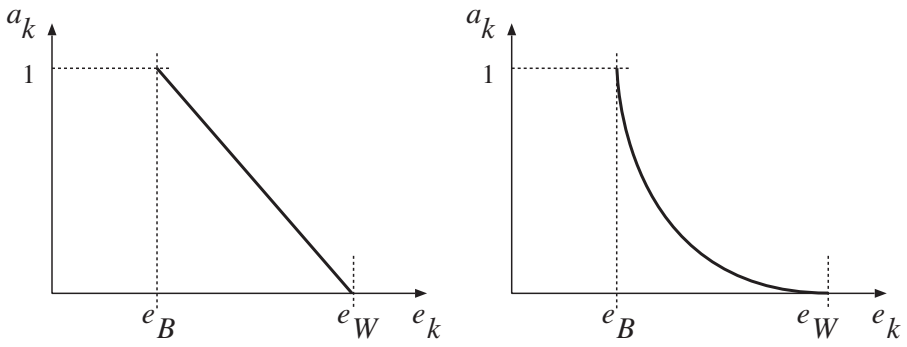
In order to give more authority to the opinions given by the most accurate classifiers, the values obtained by the former approach are squared (see figure 3.1b). This way,

$$a_k = \left(\frac{e_W - e_k}{e_W - e_B}\right)^2 \quad .$$

3.3.3 Classification

For these voting methods, once the weights for each classifier decision have been computed, the class receiving the highest score in the voting is the final class prediction. First, the weights of the classifiers (w_k) are obtained by normalizing their authorities:

$$w_k = \frac{a_k}{\sum_{i=1}^{K} a_i} \tag{3.18}$$



(a) Linear weighting (b) Quadratic weighting

Figure 3.1: Different models for giving the authority (a_k) to each classifier in the ensemble as a function of the number of errors (e_k) made on the training set.

Then, if $\hat{c}_k(\mathbf{x})$ is the prediction of C_k for the sample $\mathbf{x}$, then the prediction of the ensemble can be computed as

$$\hat{c}(\mathbf{x}) = \arg\max_j \sum_{k=1}^{K} w_k \delta(\hat{c}_k(\mathbf{x}), c_j) \quad , \tag{3.19}$$

being $\delta(a, b) = 1$ if $a = b$ and 0 otherwise.

4

Classification of music by melody

In this chapter, the experiments on genre classification of symbolic music by melody are presented. The idea behind this chapter is to study how and to what extent, melodic content can be utilized to perform music genre classification. For this reason, melodies are transformed into symbol sequences using the encoding methods explained in Section 2.2. Then, using the pattern recognition methods explained in Chapter 3, a different statistical model for each genre is built, capturing the different use of symbols — or patterns of symbols — in each genre.

In this task, the naïve Bayes classifier and n-gram models have been used, and the results obtained with both techniques will be analyzed comparatively. But, as both algorithms have different degrees of freedom, it is necessary to select the optimal combination of parameters for each of them first. Thus, a preliminary study using the naïve Bayes classifier was performed, in order to study the behaviour of the three statistical models considered, and also the influence of the feature selection procedure. The different variations of the coupled encoding have been also studied. These results are discussed in detail in Section 4.1. Then, in Section 4.2 the naïve Bayes classifier is compared with n-gram modeling, using both coupled and decoupled encodings. In Section 4.3 both methods are used to classify audio music, as a back-end to an automatic music transcription system. In Section 4.4 the results obtained in this work are compared with other works using the same corpora. Finally, Section 4.5 summarizes the conclusions obtained from these experiments.

4.1 Evaluation of the naïve Bayes classifier and coupled encoding

In this section, the performance of the naïve Bayes classifier will be studied. Three statistical models have been considered: multivariate Bernoulli, mixtures of multivariate Bernoulli, and multinomial. These three models have been thoroughly studied when applied in text classification tasks, achieving good classification rates (McCallum and Nigam, 1998; Novovičová

and Malík, 2002). However, there is not a prior knowledge of which statistical model will perform the best in this task. The behaviour of each model is highly dependent on the size of the vocabulary used, and while some data sets are better classified using a small vocabulary, others need a larger one.

In this thesis, musical sequences are encoded into text documents. However, the nature of the data used here is quite different from the data sets used in text classification tasks, so it is necessary to study how the classifier performs when applied to music. For this reason, several experiments were performed using the *Ponce-2-genres* and *Cruz-3-genres* corpora. In these experiments, different vocabulary sizes were also tested, ranging between 1 and $|\mathcal{V}|$ (the maximum vocabulary size extracted from the corpus). For each size tested (S), all the words in the vocabulary are ranked using their AMI value (see Section 3.1.4), and the first S words are selected.

These experiments were also useful to test the different variations of the coupled encoding. When encoding melodies using this method, consecutive notes in the melody are grouped in sets of a given size n, and their pitch intervals and IOR are encoded together as n-words. There are three parameters that directly affect the encoding process, namely:

- *Length of the n-words.* Long n-words provide more specific information, but the vocabulary coverage on the data sets is small. Shorter n-words, on the contrary, are more general but provide a better vocabulary coverage. The following values will be tested: $n \in \{2, 3, 4, 5, 6, 7\}$.

- *Last IOR.* The encoding format proposed in (Doraisamy and Rüger, 2003) does not include the IOR for the last two notes in the n-word. An extension to this format has been proposed by adding the last IOR using the duration of the last note. Both formats will be studied comparatively.

- *Rest threshold* for filtering the *stop words.* Although this parameter does not affect how words are encoded, it has an impact on the vocabulary coverage. Values $t \in \{1, 4\}$ beats will be tested. Since all the songs in corpora *Ponce-2-genres* and *Cruz-3-genres* are in 4/4 meter, $t = 4$ is equivalent to a rest of one bar.

The combination of all the possible values for these three parameters results in a total of $6 \times 2 \times 2 = 24$ different feature sets extracted from each corpus. Each one of these feature sets was used to train a different naïve Bayes classifier, using the three statistical models: Bernoulli (B), mixtures of Bernoulli (MB) and multinomial (M). Also, for the MB model, several component numbers in the mixtures were tested, $m \in \{2, 3, 4, 6, 8, 10\}$. Note that the particular case $m = 1$ would be equivalent to the standard Bernoulli model.

Due to the small size of the corpora, specially in the case of corpus *Ponce-2-genres*, the results have been evaluated using a *leaving-one-out* estimator: the training set is built using all the melodies but one, which is kept for testing the classifier. This process is repeated until all the melodies have been used for testing, and the performance of the classifier is measured as the percentage of correctly classified test samples. This allows to obtain a better estimation of the classification error despite the small size of the data sets.

Tables 4.1a and 4.1b show the best results obtained for corpora *Ponce-2-genres* and *Cruz-3-genres*. For each n-word size the best success rate is displayed, along with the combination of parameters for which these results were obtained. In the following subsections, the effects of each encoding parameter are discussed in detail.

n	accuracy (%)	last IOR	rest threshold	model	vocabulary size
2	92.1	yes	4	MB(2)	100
3	89.1	yes	4	MB(4)	7500
4	80.7	no	4	B	100
5	82.3	no	1	M	200
6	78.8	no	4	M	14000
7	73.9	no	4	M	10000

(a) Corpus *Ponce-2-genres*

n	accuracy (%)	last IOR	rest threshold	model	vocabulary size
2	90.6	yes	4	M	100
3	94.3	yes	4	MB(3)	2000
4	90.6	no	4	B	1000
5	84.6	no	1	M	7000
6	69.5	no	4	M	50
7	53.3	no	4	B	500

(b) Corpus *Cruz-3-genres*

Table 4.1: Best results obtained with corpora *Ponce-2-genres* (top) and *Cruz-3-genres* (bottom) with the naïve Bayes classifier and coupled encoding. *Model* column shows the statistical model used: multivariate Bernoulli (B), mixtures of multivariate Bernoulli (MB), and multinomial (M). For the mixture model, the number of components in the mixture is included in brackets.

4.1.1 Influence of the n-word length

The encoding parameter that most clearly influences the results is the length of the n-words, i.e. the number of notes represented in each word. The best results have been obtained using small words ($n = 2, 3$) for both corpora, and there is a decreasing trend in classification accuracy as the value of n increases. This is due to the small vocabulary coverage that is obtained when using long words, that affects the quality of the models built from the training data. As the features extracted from the melodies get more sparse, it is more difficult to train good probability estimates for each class, and also increases the probability to find unseen words in the test documents.

Success rates shown in Tables 4.1a and 4.1b correspond to the best overall results for the corpora. In order to better illustrate the decreasing trend in classification accuracy regarding the size of the words, Figure 4.1 shows the evolution of the classifier as a function of the vocabulary size, using different words sizes. In this figure, the behavior of the classifier regarding the value of n can be observed. As expected, best success rates are always obtained for small word sizes, and the results drop significantly for larger values.

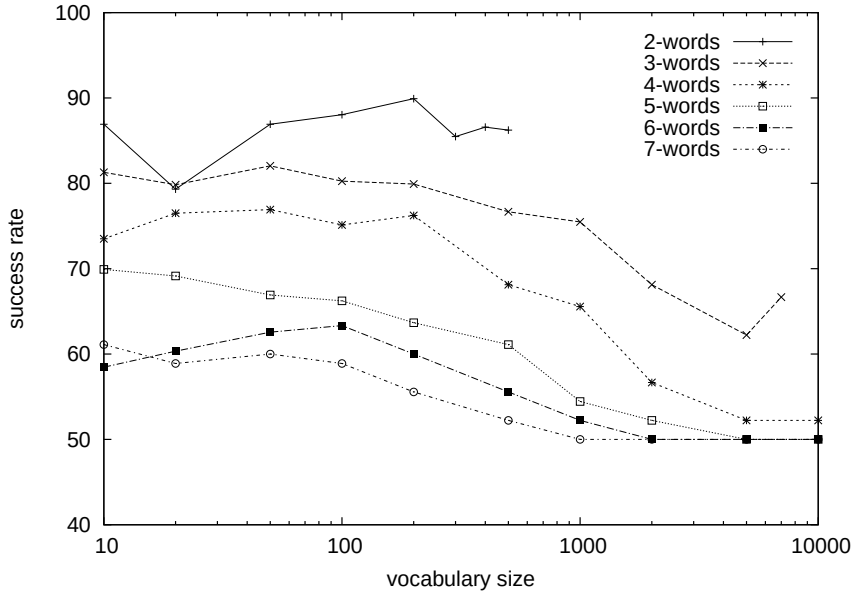
4.1.2 Last IOR

In Table 4.1 it can be seen that including the last IOR in the encoded words provides better results for n-word sizes $n \in \{2, 3\}$, while for longer words better results are obtained when it is not included in the encoding. The same behaviour can be observed in Figures 4.2 and 4.3.

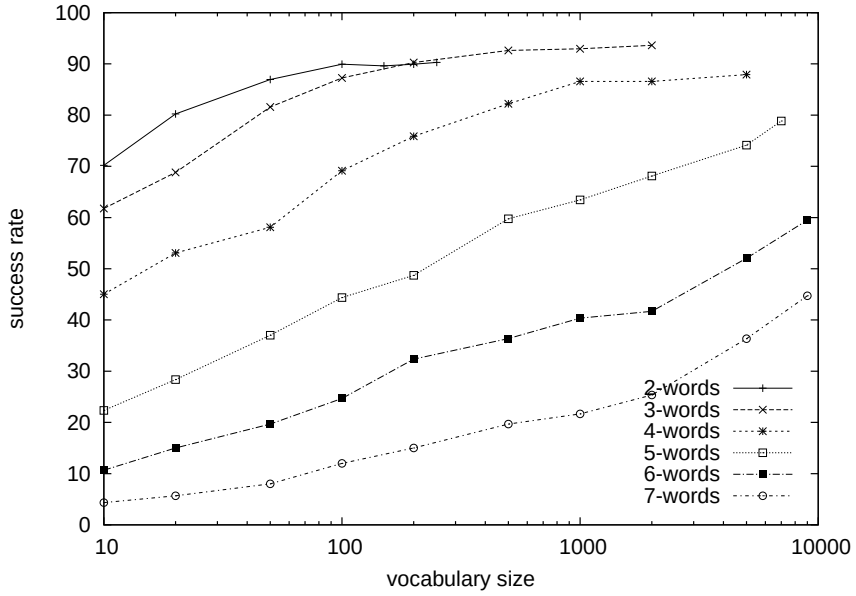
Again, this behaviour can be attributed to the vocabulary coverage. The n-words including the last IOR contain more information and are more specific, but when using small word sizes the vocabulary coverage is still relatively large. However, when using larger words, coverage decreases, so these words become too specific. In this case, better results are obtained using words without the last IOR.

4.1.3 Rest threshold

Values of 1 and 4 beats have been tested as a threshold to discard the n-words containing consecutive notes separated by large rests. Figure 4.4 shows some results using both values. In this figure it can be seen that this threshold does not have an important influence in the classification results, as only small variations can be appreciated in the experiments. But, although these differences are not significant, best results were obtained using a value of 4 beats in most cases, as shown in Tables 4.1a and 4.1b.

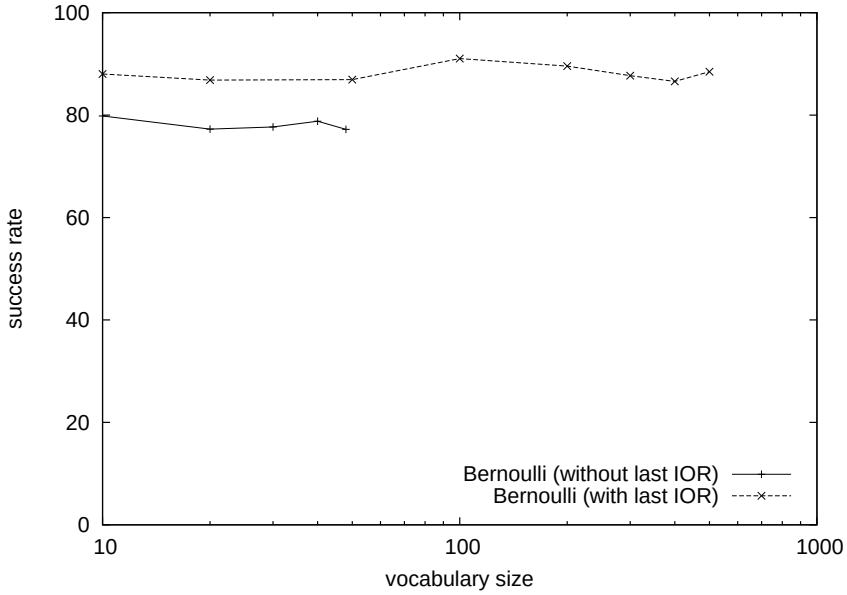


(a) Corpus *Ponce-2-genres*, multivariate Bernoulli

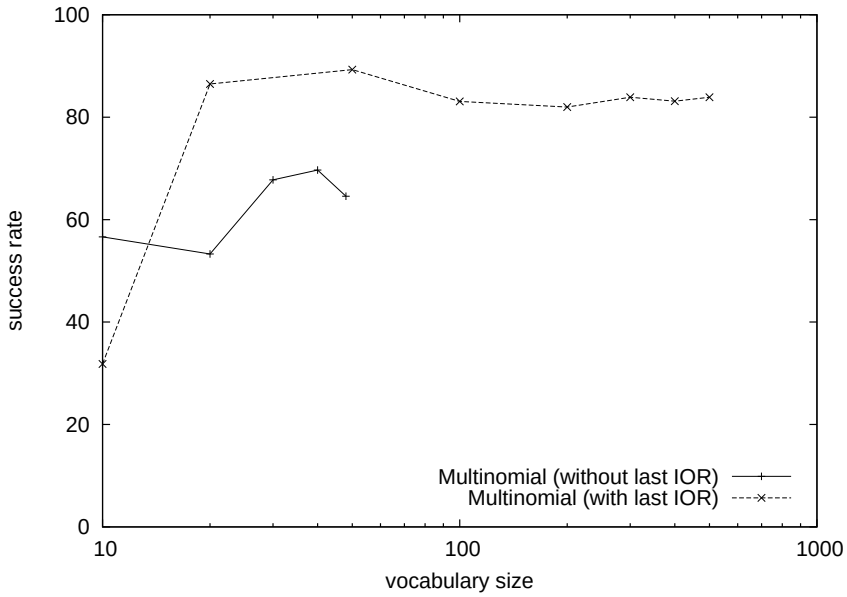


(b) Corpus *Cruz-3-genres*, multinomial

Figure 4.1: Success rate obtained for different word lengths with corpus *Ponce-2-genres* (top) and *Cruz-3-genres* (bottom).

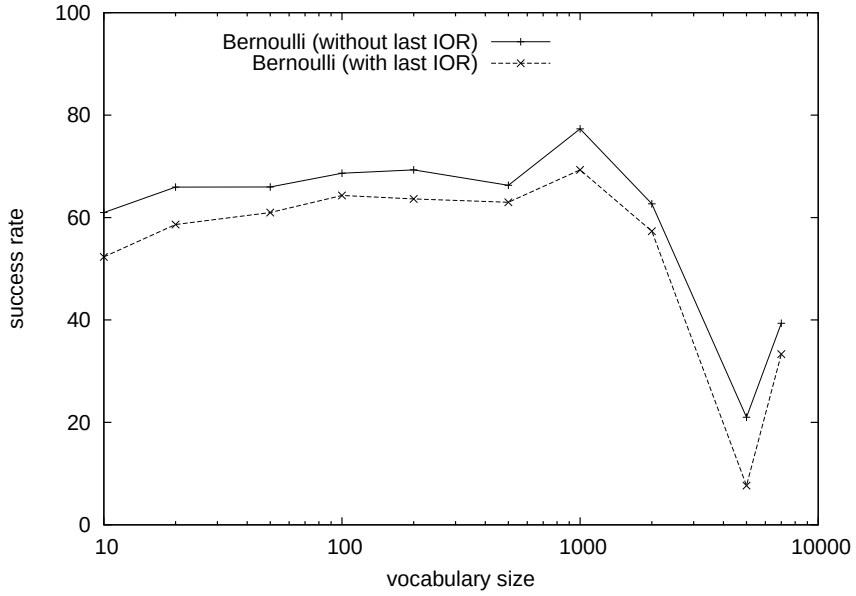


(a) multivariate Bernoulli

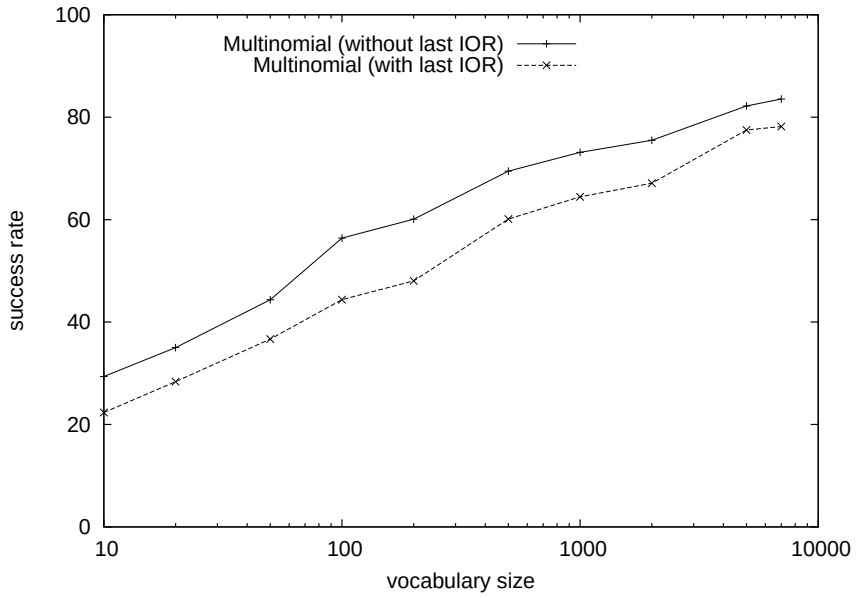


(b) multinomial

Figure 4.2: Comparison of the encoding formats including and not including the last IOR, for corpus *Ponce-2-genres*, using different statistical models and $n = 2$.

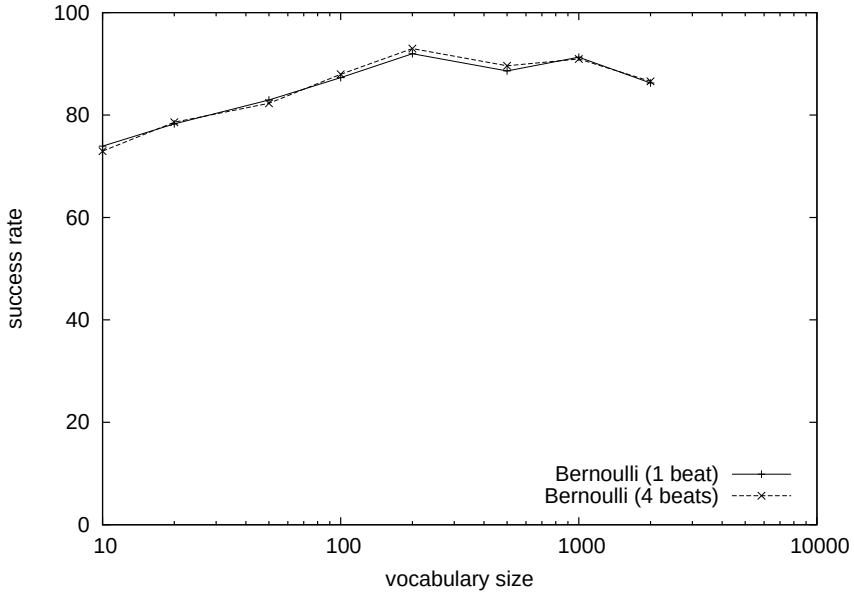


(a) multivariate Bernoulli

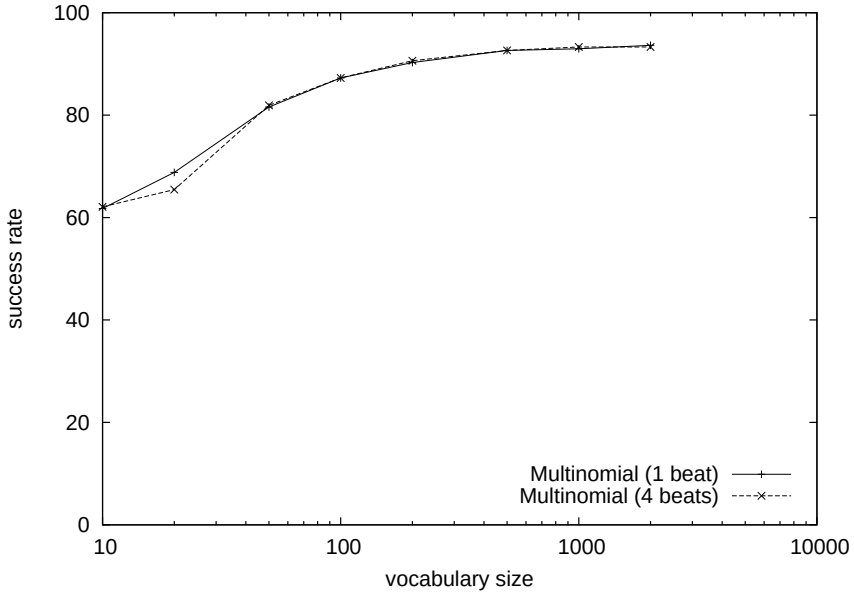


(b) multinomial

Figure 4.3: Comparison of the encoding formats including and not including the last IOR, for corpus *Cruz-3-genres*, using different statistical models and $n = 5$.



(a) multivariate Bernoulli



(b) multinomial

Figure 4.4: Comparison of the results obtained for the two *rest threshold* values, using corpus *Cruz-3-genres* and $n = 3$.

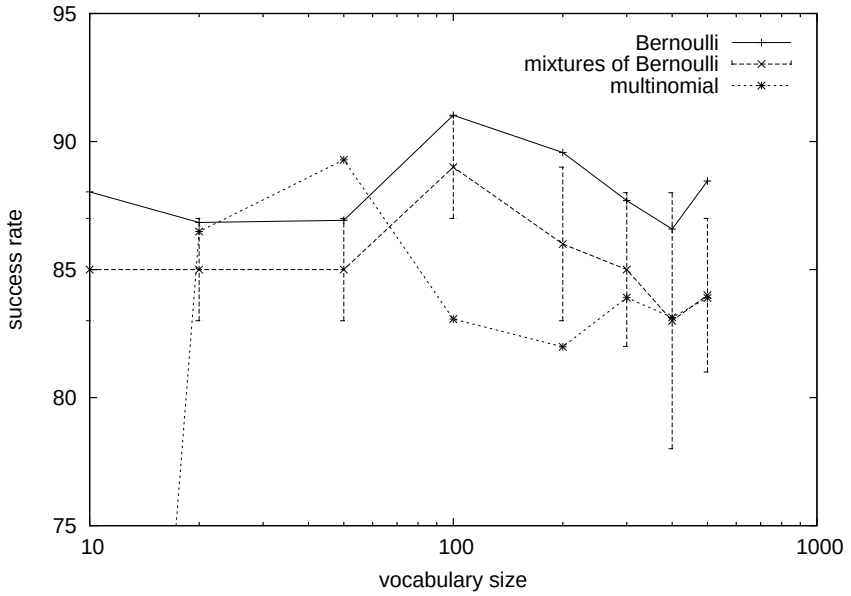
# components	accuracy (%)	time (sec.)
1	91.0	3.495
2	92.1	15.352
3	87.7	23.460
4	87.7	30.743
6	91.5	49.525
8	87.4	70.006
10	88.9	88.984

Table 4.2: Classification times for corpus *Ponce-2-genres* using different mixture models and a vocabulary of 100 words. A mixture model with just one component is equivalent to the multivariate Bernoulli model.

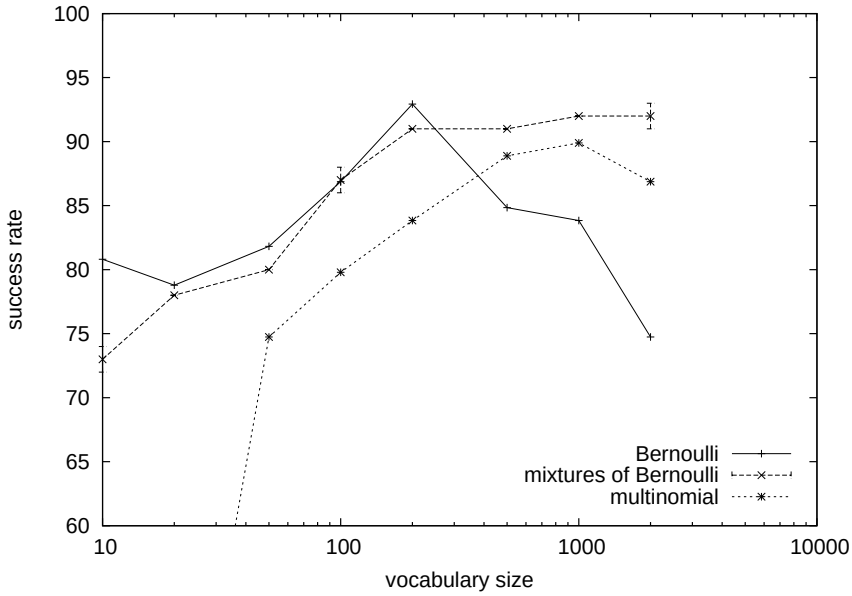
4.1.4 Statistical models

In the experiments, the three statistical models used have shown the same behaviour reported in other works using text data sets (McCallum and Nigam, 1998; Novovičová and Malík, 2002). Figure 4.5 shows the experiments where best results were obtained for each corpus. In general, when using small vocabularies, better results can be obtained with the Bernoulli models (both single and mixtures), while the multinomial model performs better as the size of the vocabulary increases. This happens because, when using large vocabularies, more infrequent words are used to classify. The multinomial model is well-suited to handle this situation, as it uses the frequency of each word to compute the probability of a document. In the multivariate Bernoulli model, on the contrary, each word in the vocabulary has the same weight in the decisions taken. The multinomial model also seems to be more robust to sparseness of data. In Tables 4.1a and 4.1b it can be seen that the best results for n-word sizes $n \in \{5, 6, 7\}$ were obtained using the multinomial model, reaching a success rate over 80%, while the best results obtained with the Bernoullis range from 50% to 60%.

Regarding the mixture models, the results are not conclusive in order to select an optimal number of components, as none of the values tested stood out from the others. Besides that, this model has not shown any significant improvement over the other two models. Although the best overall results for both corpora were obtained using the mixture model, in some cases the average performance of the mixtures is worse than that of the multivariate Bernoulli (see Figure 4.5a). Another factor to be taken into consideration is the running time of the algorithm. Estimating the probabilities for a mixture model has an increased cost of time over the single multivariate Bernoulli model. Table 4.2 shows the time necessary to classify all the files



(a) Corpus *Ponce-2-genres* ($n = 2$)



(b) Corpus *Cruz-3-genres* ($n = 3$)

Figure 4.5: Success rate obtained for *Ponce-2-genres* (top) and *Cruz-3-genres* (bottom) using the naïve Bayes classifier and the three statistical models: Bernoulli, mixtures of Bernoulli and multinomial. Results shown for the mixture model represent the average using different number of components $m \in \{2, 3, 4, 6, 8, 10\}$. Error bars show the standard deviation over this average.

in corpus *Ponce-2-genres*[1], using different number of components for the mixture. A mixture with 2 components is more than four times slower to train than the single multivariate Bernoulli, with only a small improvement in classification accuracy. This difference raises up to 25 times for a mixture with 10 components, and greater times are needed as the vocabulary size increases.

4.1.5 Feature selection

In the experiments, the feature selection procedure has shown to have a key role in achieving the best classification rates. Only for large words ($n \geq 5$) and using the multinomial model, the whole vocabulary performed better than all the reduced sets using feature selection (see Figure 5.2b). There is no homogeneity in the results as to decide which is the optimal vocabulary size. However, even if the optimal size was found, it could change with the size of the training set, or if a new set of files were used (McCallum and Nigam, 1998).

4.1.6 Conclusions

From the results shown in the previous sections, it can be concluded which are the best combinations of encoding parameters and methods for further experimentation:

- *Length of the n-words.* The naïve Bayes classifier has proven to be highly sensitive to data sparseness, as the feature sets that provided a better vocabulary coverage performed the best. From now on, only word sizes $n \in \{2, 3, 4\}$ will be considered. Although for corpus *Ponce-2-genres* a better result was obtained with $n = 5$ than with $n = 4$, this is not the general behaviour.

- *Last IOR.* Best results were obtained using the n-words containing the last IOR and small word sizes, so the extended encoding format is preferred over the original.

- *Rest threshold.* Although there is not a significant difference in the results, the threshold of 4 beats will be kept because it provides better vocabulary coverage.

- *Statistical models.* It is not possible to know a priori which of the three statistical models is the best for classifying new data sets, as all of them performed comparably in this task. However, the mixture model has some disadvantages over the other two, despite its good performance: it is a time-consuming algorithm and the optimal

[1] The times were measured on a 2.2 GHz Intel Core 2 Duo machine with 2 GB RAM.

number of components needs to be found for each data set. Then, only the multivariate Bernoulli and multinomial models will be used in future experiments.

- *Feature selection.* There is not a fixed vocabulary size that provides optimal results, so different values need to be tested in order to find the optimal for each task.

4.2 Comparison of methods

In this section, n-gram models and the decoupled encoding method are introduced, in addition to the naïve Bayes classifier and the coupled encoding, which have been thoroughly studied in the previous section.

In (Cruz-Alcázar and Vidal, 2008), the authors obtained very good results with the *Cruz-4-genres* corpus, using n-grams and a decoupled encoding very similiar to the one used in this thesis. The purpose of the experiments in this section is to test how the naïve Bayes classifier performs compared to the n-gram models in the same task, and also to find the best combination of encoding and classification methods. The same experiments were also tested in a more difficult problem, using a bigger corpus with more genres, in order to see if the same results can be obtained when the differences between genres are more subtle.

4.2.1 Methodology

For the naïve Bayes classifier, the multivariate Bernoulli (B) and multinomial (M) models were used, following the conclusions in Section 4.1.6. Feature selection was also performed, testing different values from 1 to the maximum vocabulary size. For the n-grams, values of $n \in \{2, 3, 4\}$ were used.

Four feature sets were extracted from the melodies in each corpora using the coupled encoding (with groups of 2, 3, and 4 notes; and including the last IOR) and the decoupled encoding. For both methods, a rest threshold of 4 beats was used. Thus, the following combinations of classifiers and encoding methods have been tested:

- Naïve Bayes *(B, M)* – coupled encoding *(2-, 3-, 4-words)*
- Naïve Bayes *(B, M)* – decoupled encoding
- n-grams *(2-, 3-, 4-grams)* – coupled encoding *(2-words)*
- n-grams *(2-, 3-, 4-grams)* – decoupled encoding

Note that, in the combination of n-grams and coupled encoding, only words of size 2 were used because the n-gramming technique already makes groups of notes when builiding the statistical models.

Experiments with the *Ponce-2-genres* and *Cruz-3-genres* corpora were validated again using a leaving-one-out estimator. For corpora *Cruz-4-genres* and *Perez-9-genres* the experiments have been validated using a 10-fold cross-validation scheme for computational cost reasons, due to the bigger size of the data sets. Using this scheme the data set is split in two, 90% is used to train the classifier and the remaining 10% as test. This process is repeated 10 times, and the results obtained in each subexperiment are averaged, giving standard deviations as confidence intervals. For the naïve Bayes classifier, only the best results from all the vocabulary sizes tested are presented.

In order to compare the results obtained with the different combinations of encoding and classification methods, two statistical hypothesis tests (Howell, 1997) have been used: 1) a two-way ANOVA test that allows to study the interaction of both encoding and classification methods, and 2) a one-way ANOVA test with Tukey procedure for multiple comparison, in order to test the significance of the results and to find the best combination of methods. These tests are intended for comparing mean values from a set of measures by computing the ratio of the estimated variances of the samples, and require that the observations used to compute the means are drawn from a normal distribution. In order to compare the results obtained in the experiments, these tests are applied to the average accuracy rates calculated from the subexperiments performed with each corpus. When using a leaving-one-out estimator, the number of subexperiments is big enough to assume that the average classification rate follows a normal distribution due to the Central Limit Theorem (Feller, 1971). Then, sample variances are computed using the following formula:

$$\sigma^2 = \frac{p(1-p)}{N-1} \tag{4.1}$$

where p is the accuracy rate and N is the number of subexperiments performed. When using 10-fold cross-validation, it has been also assumed normality in order to perform the tests, although the number of observations made is not enough to ensure normality.

4.2.2 Corpus *Ponce-2-genres*

Table 4.3 shows the results obtained for corpus *Ponce-2-genres*. Results for the naïve Bayes classifier and coupled encoding are the same obtained in the previous section, with the exception of the mixture of multivariate Bernoulli model which has been excluded for the reasons exposed before. The best result was obtained using the combination *3-grams–decoupled*, reaching a 92%, comparable to the best result obtained with the mixture model. Similar results were also obtained using different combinations of encoding and classification methods, so it is not possible to identify the best

combination for this task. However, the n-gram approach has the advantage over naïve Bayes that it is not necessary to explore the vocabulary space in order to find the optimal vocabulary size using a feature selection procedure.

	naïve Bayes		n-grams		
	B	M	$n=2$	$n=3$	$n=4$
2-words	91 ± 3	89 ± 3	90 ± 3	85 ± 3	84 ± 3
3-words	83 ± 3	81 ± 3	–	–	–
4-words	77 ± 4	76 ± 4	–	–	–
decoupled	86 ± 3	91 ± 3	86 ± 3	$\mathbf{92 \pm 3}$	90 ± 3

Table 4.3: Classification results for corpus *Ponce-2-genres*.

4.2.3 Corpus *Cruz-3-genres*

Results with the *Cruz-3-genres* corpus are shown in Table 4.4. This time, best results were obtained using *naïve Bayes–coupled*, reaching also an accuracy rate close to the one obtained with the mixture model. For the n-gram models, best results were obtained again using the combination *3-grams–decoupled*, while the results for the coupled encoding were much lower. This is consistent with the results obtained in (Cruz-Alcázar and Vidal, 2008), although the encoding method proposed in this thesis has not reached the excellent results shown in that work.

	naïve Bayes		n-grams		
	B	M	$n=2$	$n=3$	$n=4$
2-words	89 ± 2	91 ± 2	67 ± 3	67 ± 3	68 ± 3
3-words	93.0 ± 1.5	$\mathbf{93.3 \pm 1.4}$	–	–	–
4-words	85 ± 2	87 ± 2	–	–	–
decoupled	76 ± 2	87 ± 2	77 ± 2	91 ± 2	88 ± 2

Table 4.4: Classification results for corpus *Cruz-3-genres*.

4.2.4 Corpus *Cruz-4-genres*

In this section, the experiments with the *Cruz-4-genres* corpus are presented (see Section 2.1.2). Table 4.5 shows the results obtained in the experiments. It can be seen that the best results were obtained with the combinations *(3,4)-grams–decoupled*, followed by *naïve Bayes–coupled*. The other combinations of encoding and classification methods obtained poorer results.

The two-way ANOVA test revealed that there is a significant interaction between both encoding and classification methods, with a 99.9% confidence level. From these results, it can be concluded that the naïve Bayes classifier

	naïve Bayes		n-grams		
	B	M	$n = 2$	$n = 3$	$n = 4$
2-words	95 ± 4	94 ± 3	80 ± 4	81 ± 4	80 ± 3
3-words	95 ± 2	92 ± 4	–	–	–
4-words	94 ± 4	95 ± 3	–	–	–
decoupled	70 ± 6	95 ± 3	93 ± 3	$\mathbf{98 \pm 2}$	$\mathbf{98 \pm 2}$

Table 4.5: Classification results for corpus *Cruz-4-genres*.

performs better when using coupled encoding, while the n-grams technique does with the decoupled encoding.

In order to find the best combination of encoding and classification methods, a one-way ANOVA test was performed. The results of this test revealed that there is not a statistically significant difference between the best results obtained with both classification methods, at a 95% confidence level. Then, from these results, it cannot be concluded that the n-grams are the best method to classify melodies in this corpus, although they reached the highest classification rate (98%).

4.2.5 Corpus *Perez-9-genres*

From all the corpora used in this thesis, this is probably the most interesting to study. It contains more genres than the others, and also a rich variety of authors and styles within each genre, so it makes the classification task closer to a real world problem. Moreover, the hierarchical taxonomy of genres allows to test the classification methods at the two levels of the genre hierarchy. The first level corresponds to the three broad domains: academic, jazz, and popular; and the second level to the nine subgenres shown in Table 2.1.

Three-genres classification

In this experiment, all the files in the corpus were grouped in three classes, corresponding to the three broad music domains considered: academic, jazz, and popular music. Table 4.6 shows the results using the different combinations of encoding and classification methods.

As it happened with corpus *Cruz-4-genres*, the best overall result was achieved using the combination *4-grams–decoupled*, but the comparison of the results using the hypothesis tests reavealed some differences in the behaviour of the classifiers. This time, n-gram models performed significantly better than the naïve Bayes classifier, regardless of the encoding method used. However, there is not a significant difference in the results between the coupled and decoupled encodings. It is worth noting that the results obtained using the combinations *(3,4)-grams–decoupled*

	naïve Bayes		n-grams		
	B	M	$n=2$	$n=3$	$n=4$
2-words	74 ± 4	73 ± 5	78 ± 5	80 ± 4	79 ± 5
3-words	75 ± 4	62 ± 5	–	–	–
4-words	69 ± 4	61 ± 5	–	–	–
decoupled	66 ± 5	72 ± 4	77 ± 4	83 ± 5	$\mathbf{84 \pm 3}$

Table 4.6: Classification results for corpus *Perez-9-genres* using three classes.

outperformed all the results obtained with naïve Bayes, while this difference was not significant when using *n-grams–coupled*. These results point to the fact that the best performance can be obtained when using n-grams ($n = 3, 4$) with decoupled encoding.

Nine-genres classification

Table 4.7 shows the results obtained in this experiment. As expected, classification accuracy has been lower than in the three-class experiment, due to several factors: the number of classes is higher, there are closer relationships between genres, and also because the number of files per class is lower than in the previous experiment. However, it is interesting to see that the best results were obtained using the combinations *(3,4)-grams–decoupled*. Again, these combinations outperformed all the results using naïve Bayes, with both coupled and decoupled encodings.

	naïve Bayes		n-grams		
	B	M	$n=2$	$n=3$	$n=4$
2-words	53 ± 3	52 ± 4	59 ± 4	53 ± 3	49 ± 4
3-words	54 ± 6	38 ± 5	–	–	–
4-words	43 ± 2	37 ± 4	–	–	–
decoupled	43 ± 3	52 ± 4	57 ± 3	63 ± 5	$\mathbf{64 \pm 2}$

Table 4.7: Classification results for corpus *Perez-9-genres* using nine classes.

Figure 4.6 shows the confussion matrix for the best result, obtained using *4-grams–decoupled*. In this figure it can be clearly seen that most errors were commited between close subgenres within each broad domain. Comparing this figure with the similarity measures shown in Figure 2.5, some interesting relationships can be observed. In some cases it seems that rhythm has a strong influence in the results, for example when comparing pre-bop and bop with all the academic subgenres. This influence is even stronger in the case of Celtic music, which had the highest classification rate (97%), and none of the other subgenres were misclassified as Celtic.

However, the behaviour of the classifier regarding pop music is more difficult to explain. Note that a high number of files in the other subgenres were assigned to this class. One possible explanation for this behaviour is that, even in the real world, this subgenre is ill-defined, and it contains an amalgam of different styles, at least regarding its melodic aspects. Then, it is easy that melodies from other subgenres fall within the limits of pop subgenre, thus leading the system to misclassifications.

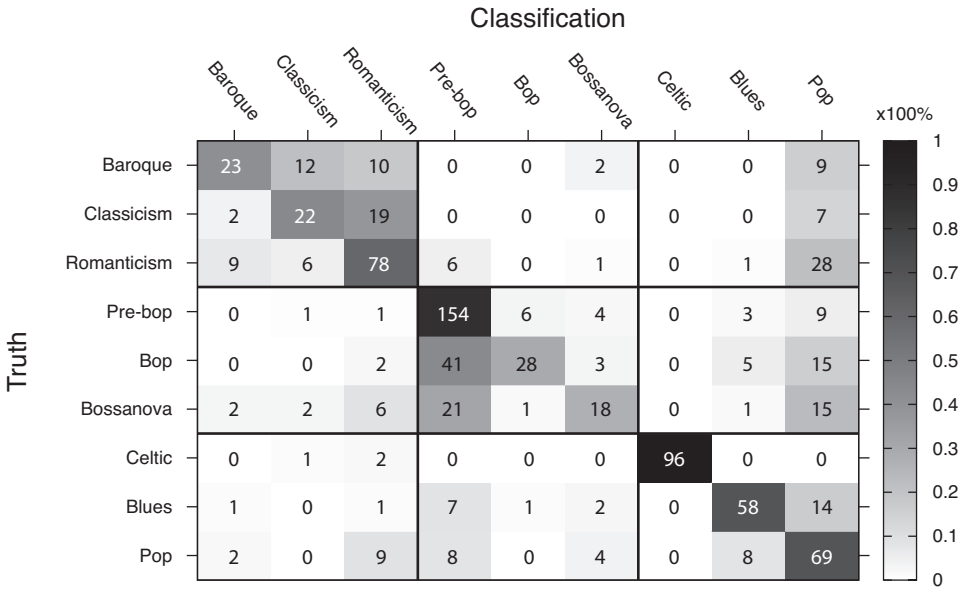


Figure 4.6: Confussion matrix for corpus *Perez-9-genres*, using 4-grams and the decoupled encoding. The greyscale represents the percentage over the total number of files in the class.

4.3 Classification of audio music

As it was discussed in Chapter 1, research in the MIR field has been traditionally separated in the symbolic and the audio domains. However, these two approaches are not completely isolated from each other. Methods used for audio music can be applied to symbolic sources by synthesizing them into audio (Cataltepe et al., 2007), and the opposite is also possible by using automatic transcription systems to transform audio music into symbolic (MIDI) files (Lidy et al., 2007). In this section, the latter approach will be followed, in order to study whether the models presented in this chapter can be applied to melodic sequences obtained from different sources.

For extracting the melody from an audio piece, it is first necessary to obtain the notes in the piece by using an automatic music transcription algorithm, but this problem is currently a subject of ongoing research and is far from being completely solved. Some state-of-the-art approaches include models of the human auditory system (Klapuri, 2005; Tolonen and Karjalainen, 2003), generative spectral models (Yeh et al., 2005), or machine learning techniques (Pertusa and Iñesta, 2005).

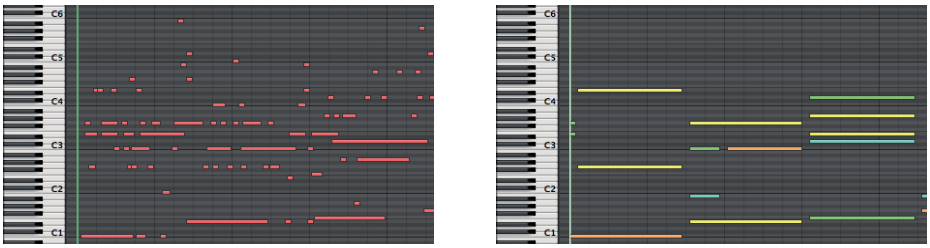
In order to test this approach, a synthetic experiment has been devised using the *Perez-9-genres* corpus and a polyphonic transcription algorithm developed by members of the Pattern Recognition and Artificial Intelligence Group at the University of Alicante (Pertusa and Iñesta, 2008). The details of the system are out of the scope of this thesis, so it will be just outlined in the next section. Then, in Section 4.3.2, some experimental results are presented.

4.3.1 Polyphonic transcription system

The goal of a polyphonic music transcription system is to obtain the score from an audio music piece. The core of these systems is usually a multiple fundamental frequency (f_0) estimation module, able to detect simultaneous notes from an analysis of the audio signal. In these experiments, the algorithm described in (Pertusa and Iñesta, 2008) has been used, tuned in order to obtain melodic sequences suitable for the encoding method used in this thesis.

The multiple pitch estimator converts a mono audio file into a sequence of MIDI notes. To detect all the pitches in the audio signal, a Short Time Fourier Transform is computed using a 93 ms Hanning windowed frame, with a 46 ms hop size. For each frame, the spectral peaks with an amplitude higher than a given threshold are extracted, and all the possible combinations of the extracted peaks are candidates to be the notes sounding in that frame. Then, the most salient set of notes is selected as the transcription of the frame. The salience of a set is computed as the sum of the saliences of each individual note, and for each note it is computed using the loudness of its harmonics and the distance of their spectral envelope to a Gaussian smoothed curve, using the assumption that the harmonics of a note produced by a real instrument usually have a smooth spectral envelope.

One drawback of this method is that, as most f_0 estimation systems, it tends to produce octave errors, i.e. it outputs notes in a different octave than the original ones, and usually these notes are only a few frames long. This is not an important problem when listening at the transcribed notes, but it is more important for the classification task because these high notes produce very large intervals and odd duration ratios in the skyline of the melody, which is used by the encoding method used in this chapter. To overcome this, the onset detection algorithm described in (Pertusa et al.,



(a) multiple f_0 estimation (b) transcription with onset detection

Figure 4.7: Effect of the onset detection system in the transcriptions. The figure on the left shows the output of the f_0 estimation system for a sample audio clip, and the figure on the right shows the transcription using onset detection.

2005) has been also used. Thus, the transcription is performed in two steps. First, all the onsets in the piece are detected, and second, the multiple pitch detection algorithm explained above is performed only for the immediate frame next to each onset, and the detected chord is assigned to the time span from its corresponding onset to the next. This way, most of the octave errors and spureus notes that usually appear at random positions different than the detected onsets are discarded. Figure 4.7 shows an example of the output of the transcription system with and without performing onset detection.

This system was evaluated in the MIREX 2007 *Multiple Fundamental Frequency Estimation & Tracking* task[2], obtaining an average F-measure of 0.408[3], which is a good result given the difficulty of the task (best result was 0.527).

4.3.2 Experiments

In order to test the transcription system, audio versions of the files in the *Perez-9-genres* corpus were synthesized using the TiMidity++ software[4], and then transcribed back into MIDI with the transcription system. This way, the effect of the transcription errors in the classification task can be studied, comparing the results in this section with those obtained using the original data set. In order to prune some spureous notes introduced by the transcription system, the resulting MIDI files were postprocessed, removing all the notes with a duration shorter than a sixteenth note.

The transcribed data set was encoded using both coupled and decoupled encodings, but for the coupled version only words of length 2 were used,

[2] http://www.music-ir.org/mirex/2007/index.php/Main_Page

[3] Measured for Task II, results based on onset only. The system is identified as PI2.

[4] http://timidity.sourceforge.net

because larger values did not show any improvement in the previous experiments. Experiments were performed again using 10-fold cross-validation, and the results are shown in Table 4.8. Note that the best results were obtained using the *naïve Bayes–coupled* combination, although they were not significantly better than those obtained with *n-grams–decoupled*.

		naïve Bayes		*n*-grams		
		B	M	$n=2$	$n=3$	$n=4$
3 genres	coupled	67 ± 5	$\mathbf{75 \pm 5}$	70 ± 4	69 ± 3	69 ± 2
	decoupled	42 ± 4	71 ± 4	70 ± 2	74 ± 4	70 ± 2
9 genres	coupled	50 ± 7	$\mathbf{54 \pm 6}$	50 ± 8	50 ± 7	51 ± 7
	decoupled	24 ± 8	51 ± 8	53 ± 6	53 ± 6	52 ± 7

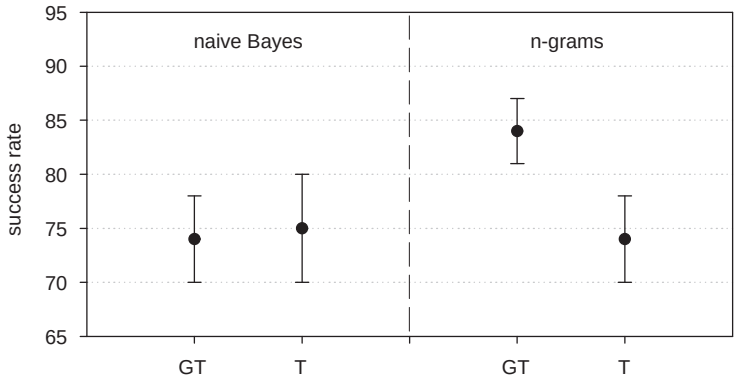
Table 4.8: Classification results for corpus *Perez-9-genres* using melodic sequences obtained from synthesized audio.

In Figure 4.8, the best results obtained with the transcribed melodies and the ground truth are shown comparatively. It is remarkable that, when using *n*-grams, the results obtained using the transcribed melodies are lower than with the original data set, around a 10% lower for both the three and nine-class problems. However, these results are still rather high considering the quality of the transcriptions. This time the naïve Bayes classifier and *n*-gram models performed comparatively, but naïve Bayes seems to be more robust to the errors introduced by the transcription system, presumably due to the feature selection procedure (vocabulary size crop that tends to exclude transcription errors due to their small appearance frequencies), and it has obtained results comparable to those obtained with the original data set. When using *n*-gram models, on the contrary, incorrect transcriptions have a higher impact because each incorrect note in the transcription introduces n different *n*-grams that are not present in the original melody, distorting the computed probabilities.

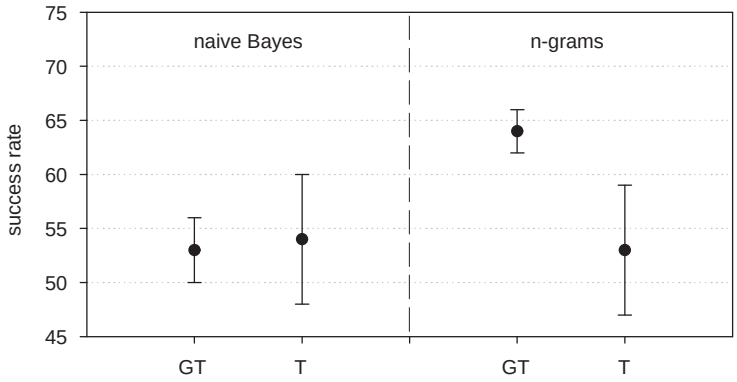
4.4 Comparison with other works

Since some of the corpora used in this chapter have been previously used by other authors, it is possible to compare the results obtained with different methods on the same data sets.

In (Cruz-Alcázar and Vidal, 2008) the authors tested several encoding methods for melodic sequences, using different representations for pitch and duration of the notes, including absolute, relative and contours. Both coupled and decoupled versions were also tested. Best classification rates were obtained with the decoupled encoding and an *n*-gram classifier, reaching a 98.3% for corpus *Cruz-3-genres* and relative representations for both pitch intervals and durations. Using corpus *Cruz-4-genres*, the



(a) 3 genres



(b) 9 genres

Figure 4.8: Comparison of the best results obtained with the ground truth of melodies (GT) and the transcribed melodies (T) using a polyphonic transcription algorithm.

accuracy rate was 99.5% using a relative encoding for pitch intervals and absolute note durations, and 96.7% when using both relative representations.

In this thesis, a similar methodology has been used. With the same classification method, the results obtained were 91% for corpus *Cruz-3-genres*, and 98% for corpus *Cruz-4-genres*. This latter result cannot be considered significantly different from the best obtained by Cruz-Alcázar and Vidal due to statistical variance. However, in corpus *Cruz-3-genres* there is a big difference in the results. Since the same classification method was used, this difference must be attributed to the encoding method. Although both works use a similar relative representation for pitch intervals and durations, the main difference is the number of symbols used. In this thesis the vocabulary size has been limited to 53 for pitch intervals, and 21 for duration ratios, while in the work by Cruz-Alcázar and Vidal all possible intervals and duration ratios are allowed (the encoding method is described in detail in (Cruz-Alcázar, 2004)). Maybe the higher rhythmic resolution had the biggest influence in these results, as this corpus was step-by-step coded using a vocabulary of 25 symbols for note durations, with 173 possible different duration ratios.

However, the encoding method used in that work has some disadvantages. When working with real MIDI files, they must be first quantized in order to smooth the note durations, which are then mapped into symbols representing absolute durations of the notes. From these symbols, duration ratios are finally computed. The advantage of the encoding method used in this thesis is that quantization is automatically performed when mapping IOR values into symbols, as shown in Table 2.2, and similar results were obtained with corpus *Cruz-4-genres*, in which melodies contain notes with "irregular" durations due to human interpretations.

Corpus *Cruz-4-genres* was also used in (de la Higuera et al., 2005), with the same encodings used by Cruz-Alcázar and Vidal (2008), but a different classification method. In that work, classification was performed using a grammatical inference algorithm called Minimum Divergenge Inference (MDI), achieving a classification rate of 96.25%. As no confidence interval is provided, it cannot be tested whether there is a significant difference with the other results on the same corpus.

Using corpus *Ponce-2-genres*, Ponce de León and Iñesta (2007) reached a 96.4% classification rate using a nearest neigbour classifier. In that work, melodies are divided into overlapping windows, and a set of shallow statistical descriptors are extracted from each window. After a feature selection procedure, the best descriptors were selected and encoded as vectors, which were then fed to the classifier. When using complete melodies instead of overlapping windows, classification rate dropped to a 93.0%, which is closer to the 92.1% obtained in this work with the naïve Bayes classifier. This method uses a different approach to represent melodic information, encoding global statistics of the melody instead of local relationships of the

notes as in this thesis. Both representation methods are complementary, and have proven to work well when used cooperatively. In (Ponce de León et al., 2006), a combination of both methods using an ensemble of classifiers was used, reaching a 98.2% classification rate for corpus *Ponce-2-genres*.

4.5 Conclusions

In this chapter, a genre classification framework of symbolic music by melody has been tested. It is based on a description of the local relationships of consecutive notes within a melody, encoding sequences of pitch intervals and duration ratios as a sequence of symbols. These sequences are then encoded into text strings, and two classification methods used traditionally for text classification tasks have been used: naïve Bayes and n-gram models.

For the naïve Bayes classifier, three statistical models have been tested: multivariate Bernoulli, mixtures of multivariate Bernoulli, and multinomial. All of them performed comparably, and have shown a similar behaviour than when used with textual information. This suggests that the underlying hypothesis that musical sequences share some characteristics with natural language is correct. However, the mixture model did not perform as well as expected. The main benefit of this model, when working with text documents, is that it allows to reflect the possibility that a single document deals with different topics. This has not been reflected when using musical sequences.

Using n-grams, excellent results have been obtained with corpus *Cruz-4-genres*, using a decoupled encoding for melodies. This method outperformed the naïve Bayes classifier, although the difference cannot be considered statistically significant. In the experiments with corpus *Perez-9-genres* there was a big difference between both methods, achieving best classification rates with n-grams. However, in these experiments the limitations of this classification method have arisen. From the 98% accuracy rate achieved with corpus *Cruz-4-genres*, results dropped down to a 64% with corpus *Perez-9-genres*, due to the higher number of genres and the presence of closer styles.

In order to improve these results, the melodic information used in this chapter should be combined with other sources of information, representing different dimensions of music. A first approach was presented in (Ponce de León et al., 2006). In that work, some of the methods presented in this chapter were combined with classifiers using shallow statistical descriptors, combining the individual decisions using an ensemble of classifiers. The results obtained using the combination of representation and classification methods outperformed those of the best single classifiers.

Finally, it has been also shown how the methods presented in this chapter can be integrated in a general MIR system working with audio data, acting as a back-end for state-of-the-art polyphonic transcription algorithms. The

results obtained in classification using the transcribed melodies were quite good, although there has been a significative loss in performance for the n-gram models, but the naïve Bayes classifier obtained classification rates comparable to those obtained with the original data set.

5

Classification of music by harmony

This chapter presents the experiments on genre classification of symbolic music using harmonic sequences. In the previous chapter it has been shown how the naïve Bayes classifier and n-gram models can help in the classification of genres using melodies. Here the same methods will be applied to the *Perez-9-genres* corpus, using a different source of symbolic information.

Several representations of the harmonic information have been tested. In Section 5.2, the experiments using chord progressions are presented, using different feature sets with different levels of chord structure information. In Section 5.3, harmonic rhythm is introduced in the encoding of chord sequences, and the results are compared with those using chord progressions only. In Section 5.4, the same methods are used for classifying chord sequences extracted from audio files, using a chord extraction algorithm. In Section 5.5, the methods used in this and the previous chapters are combined using an ensemble of classifiers, in order to test how the methods and music encodings used in this thesis perform when used cooperatively. Finally, Section 5.6 summarizes the conclusions drawn from the experiments in this chapter.

5.1 Methodology

All the experiments in this chapter have been performed using the *Perez-9-genres* corpus, using the harmonic sequences extracted from the BIAB files (see Section 2.1.4). Prior to encoding the chord sequences, all the songs have been transposed to C major / A minor in order to avoid the need of a corpus 12 times bigger and well balanced in terms of the different possible tonalities. This allows to compare the results in the same conditions according to the size of the respective vocabularies. Note that this is not exactly the same as using the relative encoding as degrees because, when encoding as degrees, a chord is encoded differently depending on whether the piece has a major or a minor tonality.

As for the classification methods, the same setup than in the previous chapter has been used. For the naïve Bayes classifier, the multivariate

Bernoulli (B) and multinomial (M) models have been used. Feature selection has been also performed, testing different values for the vocabulary size in order to find the optimal set of chords for this task. In the tables, the best result from all the vocabulary sizes tested is shown. For the n-gram models, values of $n \in \{2, 3, 4\}$ have been used.

All the experiments reported in this chapter have been validated using 10-fold cross-validation and, for each experiment, the average success rate and standard deviation are given. Also, the statistical significance of difference between the results has been tested using the one-way ANOVA test (see Section 4.2.1). When comparing results obtained with different encodings or classification methods, differences are reported according to the results of the ANOVA test.

5.2 Chord progressions

In this section, the results of the experiments using chord progressions are shown. In order to study how the amount of information available in codes affects the system performance, all the feature sets described in Section 2.3.1 have been used in the experiments. This way, it can be tested whether there is a loss in performance when complete harmonic information is not available. The results of these experiments provide a theoretical limit of the expected performance when chord sequences are obtained from a different source as, for example, a chord extraction algorithm, which usually uses a more limited chord vocabulary.

The encoding of chords using different levels of information is two-fold. Regarding the root of the chords, it can be encoded using the chord name or the degree relative to the tonality of the song. Regarding the quality of the chords, four different sets of extensions have been used: full, 4-note, triads, and major and minor chords. The combination of both root and extension encodings results in a total of eight feature sets extracted from the corpus of chord sequences. Each feature set will be referred to using the notation *root–extension*:

$$\begin{array}{cc} \textit{Root} & \textit{Extension} \\ \hline \left\{ \begin{array}{c} \textit{degree} \\ \textit{chord name} \end{array} \right\} & - \left\{ \begin{array}{c} \textit{full} \\ 4-\textit{note} \\ \textit{triads} \\ \textit{major}-\textit{minor} \end{array} \right\} \end{array}$$

As in the previous chapter, two different experiments have been performed. In the first experiment, the data set was divided in the three music domains considered: academic, jazz, and popular music. The aim of this experiment was to test whether the utilized models were able to distinguish among different music categories, as a first step to a more in deep framework.

The second experiment was performed using all the subgenres to evaluate to what extent this harmonic information can be used to distinguish musical genres when differences are more subtle.

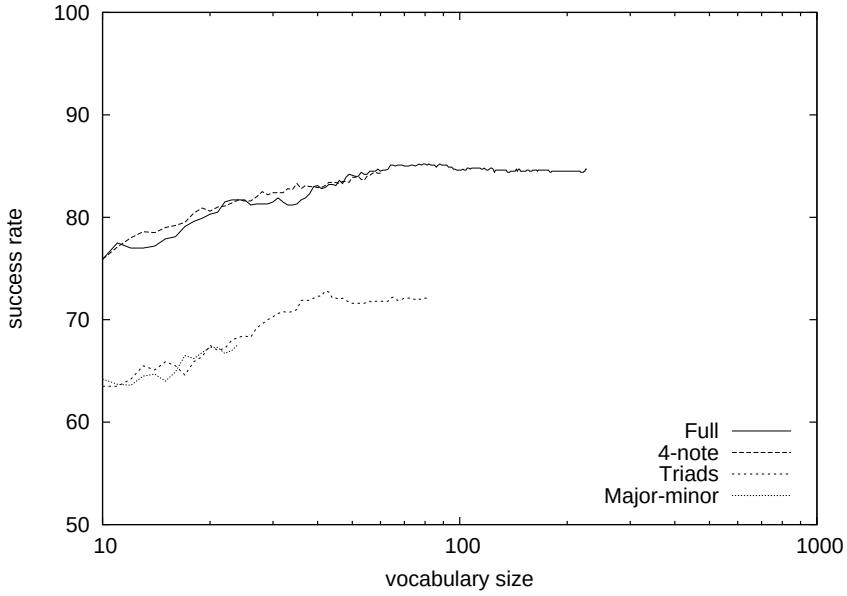
5.2.1 Three-genres classification

Prior to comparing the results of the two classification methods used, the results of the naïve Bayes classifier will be presented in order to study the behaviour of the two statistical models (multivariate Bernoulli and multinomial), and how the feature selection procedure affects the system performance. In Figures 5.1 and 5.2 the evolution of the success rate as a function of the utilized vocabulary size for all the feature sets can be observed. There were no significant differences in this experiment when using chord names or degrees. When comparing the behaviour of the two statistical models, it can be seen that slightly better results were obtained with the multivariate Bernoulli model for the full and 4-note feature sets (see also Table 5.1). For the triad and major-minor feature sets, these differences are smaller, obtaining even better results with the multinomial model in some cases. Regarding the feature selection procedure, the classification rates obtained with both models increased with the size of the vocabulary.

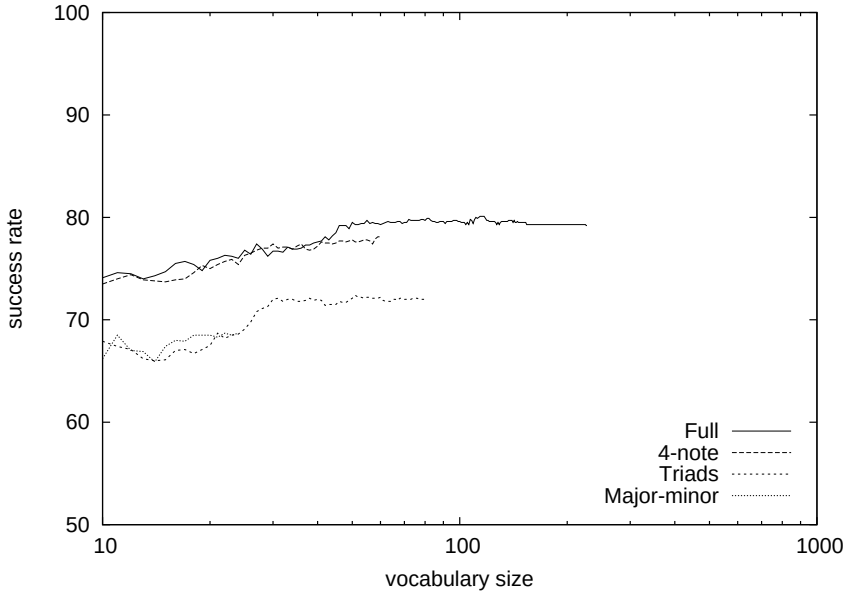
When comparing the results obtained with the four different extension sets more interesting conclusions can be drawn. Feature sets built with 4-note chords obtained classification rates similar to those obtained with the full set of extensions, and a similar behaviour can be observed with the triads and major-minor feature sets. Recall that the full and triad feature sets use a rich chord vocabulary that can be only obtained from a complex harmonic analysis of the songs, while the 4-note and major-minor sets can be considered their respective equivalents, but using a simpler vocabulary as the one extracted by state-of-the-art chord extraction algorithms. These results are encouraging in order to use these kind of algorithms as a source to obtain chord sequences for this task.

The best success rates obtained in the experiments using the naïve Bayes classifier and n-gram models are shown in Table 5.1. As it can be seen, the best results were obtained by the n-gram models, although naïve Bayes obtained similar results for the full and 4-note feature sets. Only the results obtained with naïve Bayes and the triads and major-minor feature sets were significantly lower than the others. On the contrary, n-gram models outperformed naïve Bayes when using these feature sets, because in these cases the use of context information is essential in order to make better decisions due to the small vocabulary size.

Finally note that, when using n-gram models, the classification accuracies obtained with the different values of n are very similar, and the use of larger n-grams does not improve the performance. This seems to be an effect of the interpolation process when evaluating the language models.

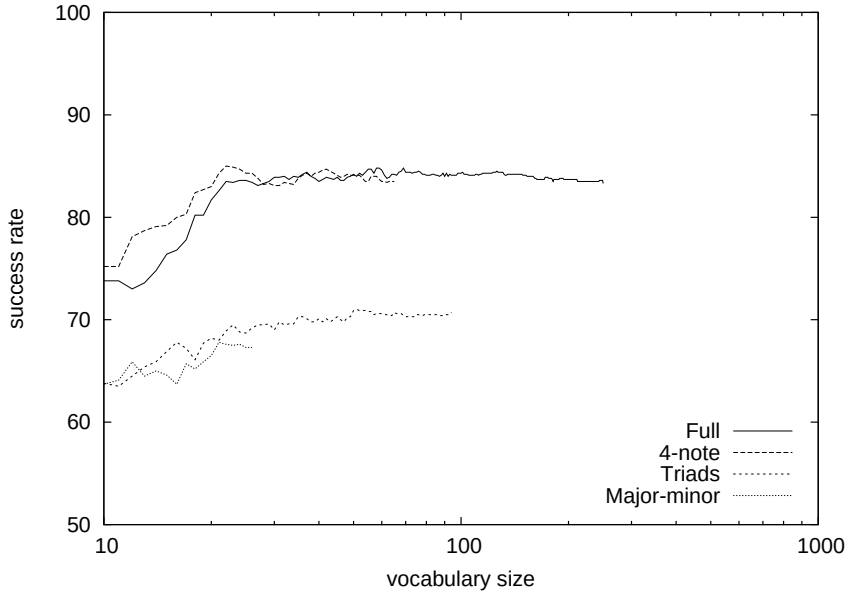


(a) multivariate Bernoulli

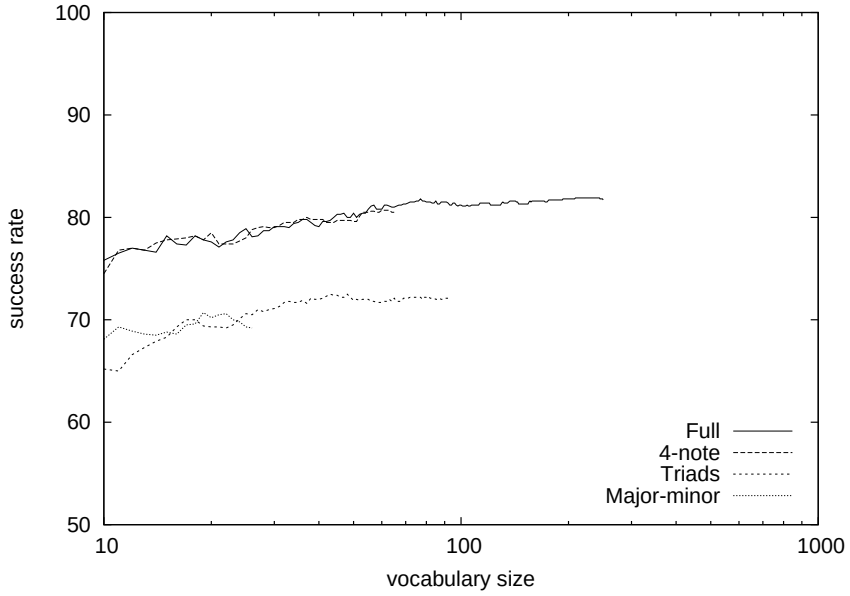


(b) multinomial

Figure 5.1: Results obtained with the naïve Bayes classifier in the 3-classes problem using the four extension sets and the root encoded as *chord names*. Results for the two statistical models are plotted: multivariate Bernoulli (top) and multinomial (bottom).



(a) multivariate Bernoulli



(b) multinomial

Figure 5.2: Results obtained with the naïve Bayes classifier in the 3-classes problem using the four extension sets and the root encoded as *chord degrees*. Results for the two statistical models are plotted: multivariate Bernoulli (top) and multinomial (bottom).

Root	Extensions	naïve Bayes		n-grams		
		B	M	$n=2$	$n=3$	$n=4$
Degrees	Full	86 ± 4	83 ± 4	$\mathbf{87 \pm 4}$	85 ± 4	86 ± 3
	4-note	86 ± 5	82 ± 4	$\mathbf{87 \pm 4}$	85 ± 4	86 ± 4
	Triads	72 ± 5	73 ± 7	$\mathbf{83 \pm 4}$	$\mathbf{83 \pm 4}$	$\mathbf{83 \pm 4}$
	Major-minor	68 ± 6	71 ± 5	$\mathbf{84 \pm 3}$	$\mathbf{84 \pm 4}$	$\mathbf{84 \pm 3}$
Chord names	Full	83 ± 3	81 ± 3	$\mathbf{86 \pm 4}$	85 ± 5	85 ± 3
	4-note	85 ± 3	79 ± 5	$\mathbf{87 \pm 4}$	85 ± 4	85 ± 3
	Triads	74 ± 6	73 ± 6	82 ± 5	$\mathbf{83 \pm 3}$	$\mathbf{83 \pm 3}$
	Major-minor	69 ± 5	70 ± 6	83 ± 4	$\mathbf{84 \pm 4}$	$\mathbf{84 \pm 4}$

Table 5.1: Average classification rates obtained for the 3-classes problem using naïve Bayes and n-gram models. Baseline accuracy by selecting the a priori most probable class is 39.5%.

When using a value of $n > 2$, the number of observations of any given n-gram can be so small that the system falls back to shorter n-gram models to compute its probability, instead of using bad probability estimates.

5.2.2 Nine-genres classification

Once the capabilities of the system to distinguish among the broad music domains have been tested, the second experiment tries to investigate how these models performed in a more complex task. In this experiment the data set consists of nine different classes, corresponding to the nine sub-genres described in section 2.1.

Table 5.2 shows the average success rate for each method and feature set. The performance is poorer now, a 64% in the best case, and the naïve Bayes classifier shows the same trend than in the previous experiment: the richer the chord vocabulary, the better the results obtained. However, the results obtained with the n-gram models show the opposite trend. This fact suggest that the system is, actually, overfitting the training set. In this task the size of each genre is smaller, and it seems that using more specific encodings results in a loss of generalization power.

When these results are studied in more detail (see Figure 5.3) it can be seen that the errors mainly occur within the three broad domains. For example, it is particularly difficult for the system to properly distinguish between baroque and classical music, or between pop-rock and country, or among jazz sub-genres. On the other hand, misclassifications among different domains are much less frequent. For example, only one of the academic pieces was classified as a jazz sub-genre, and vice versa. These results are in keeping with those presented for the three-genre case: the model is able to capture well the harmonic differences among music domains but it performs poorer for closer music genres.

Root	Extensions	naïve Bayes		n-grams		
		B	M	$n=2$	$n=3$	$n=4$
Degrees	Full	$\mathbf{62 \pm 3}$	60 ± 7	41 ± 8	42 ± 10	42 ± 10
	4-note	$\mathbf{61 \pm 7}$	51 ± 7	51 ± 3	49 ± 7	49 ± 8
	Triads	48 ± 7	50 ± 7	50 ± 11	$\mathbf{52 \pm 11}$	50 ± 10
	Major-minor	47 ± 4	46 ± 6	53 ± 7	$\mathbf{54 \pm 7}$	53 ± 9
Chord names	Full	$\mathbf{64 \pm 4}$	62 ± 5	40 ± 10	40 ± 7	40 ± 7
	4-note	$\mathbf{61 \pm 6}$	60 ± 4	49 ± 5	48 ± 6	49 ± 5
	Triads	50 ± 9	$\mathbf{52 \pm 8}$	50 ± 9	49 ± 7	49 ± 9
	Major-minor	44 ± 7	47 ± 5	$\mathbf{55 \pm 9}$	54 ± 11	53 ± 11

Table 5.2: Average classification rates obtained for the 9-classes problem using naïve Bayes and n-gram models. Baseline accuracy by selecting the a priori most probable class is 20.8%.

When looking closely at the errors committed by the system, it can be found that many of them are confusions between pop and others, or academic and celtic music. For example, there were a large number of baroque, classical, and romantic files misclassified as celtic. The reason is that celtic music makes use of a very small vocabulary of chords, as seen in Table 2.3, being all of them triads or seventh chords (Table 5.3 shows the most frequent 3-chord progressions found in the celtic corpus). Thus, a language model built from celtic music assigns a high probability to all progressions that use these simple chords. Therefore, all test files in which progressions of simple chords prevail are always classified as celtic. Such is the case, for example, of "Ave Maria" by Gounod. Note that the first bars in this piece (C Dm G7 C Dm G7 C Am) can be found among the most frequent celtic progressions shown in Table 5.3.

Rank	Progression			Rank	Progression		
1	C	G7	C	11	G7	C	Dm
2	G7	C	G7	12	C	Dm	G7
3	C	Dm	C	13	F	G7	C
4	Am	G	Am	14	C	F	G7
5	G	Am	G	15	C	G	C
6	C	F	C	16	G	F	G
7	G7	C	F	17	F	G	F
8	Dm	C	Dm	18	C	Am	C
9	Dm	G7	C	19	F	C	F
10	F	C	G7	20	G	C	G

Table 5.3: Most frequent chord progressions found in celtic music using $n = 3$.

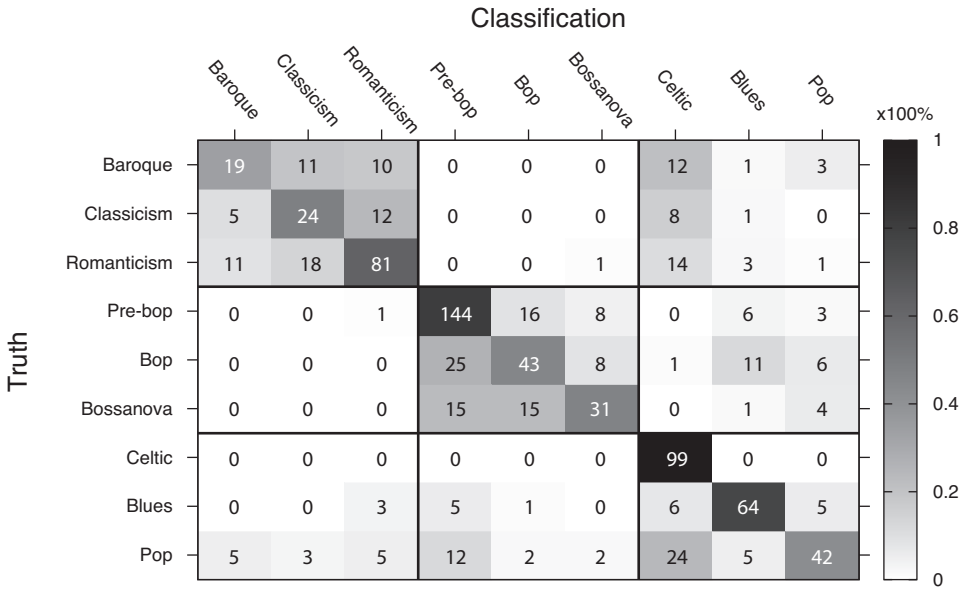


Figure 5.3: Confussion matrix for corpus *Perez-9-genres*, using naïve Bayes and feature set *chord names–full*. The greyscale represents the percentage over the total number of files in the class.

5.3 Harmonic rhythm

In the previous section it has been shown that chord progressions can achieve quite good results in the genre classification task. However, the encoding used in those experiments does not take into account any rhythmic information, as only chord changes are encoded in the sequences. As it was discussed in Section 2.3.2, harmonic rhythm — i.e. where chord changes happen — is an important aspect in music perception. In this section the experiments using chord sequences including harmonic rhythm are presented. For this, all the chord sequences in the *Perez-9-genres* corpus were encoded again, using the encoding described in Section 2.3.2. In the encoding, the same feature sets as in the previous section were used, this time including harmonic rhythm using the coupled and decoupled variants. Thus, a total of 16 feature sets were generated for all the possible combinations of *root–extension–coupling*. Using these feature sets, experiments were performed again at the two levels of the genre hierarchy.

Despite the perceptual importance of harmonic rhythm, the results obtained in these experiments were very disappointing because they did not show any significant improvement over the results obtained using just chord progressions. The complete set of results obtained in these experiments can be found in Appendix C, Tables C.1 and C.2. In order to better illustrate

what happened, Figure 5.4 shows a comparison of the best results obtained using chord progressions with and without harmonic rhythm. As it can be seen, the improvements were just around a 1% in most cases, and they were never statistically significant. Only in the 9-class problem and using n-gram models this difference was bigger. In this case it seems that including rhythm information helps to overcome the overfitting effect seen in the previous section.

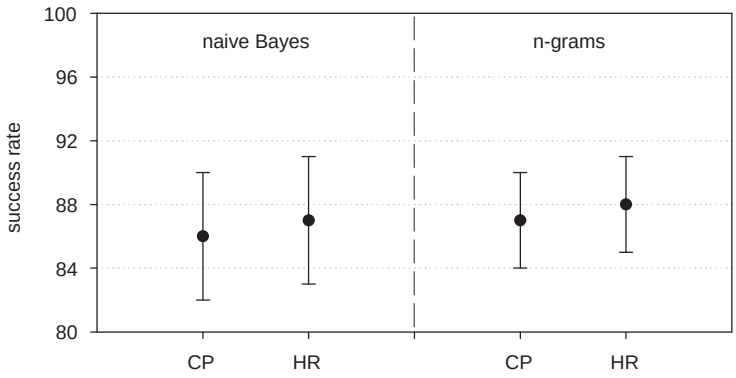
In order to find an explanation for these unexpected results, it was decided to explore the behaviour of the system using the academic subgenres. The use of the harmonic rhythm during these periods has been well studied, and it is known that, while there were not significant differences between the Baroque and Classical periods, a different use of rhythm was made in the Romantic compositions. Thus, the use of the harmonic rhythm encoding should be of help at least for distinguishing these genres.

In order to test this hypothesis, the set of academic files was divided in two classes: *Baroque–Classical*, and *Romantic*; and the files in these classes were encoded using the *degree-full* feature set for chord progressions, and *degree-full-coupled* for the harmonic rhythm encoding. The results obtained in these experiments are shown in Table 5.4. Again, similar results were obtained using both encodings. Including rhythm information did not help to obtain better results, even when trying to distinguish between the academic subgenres.

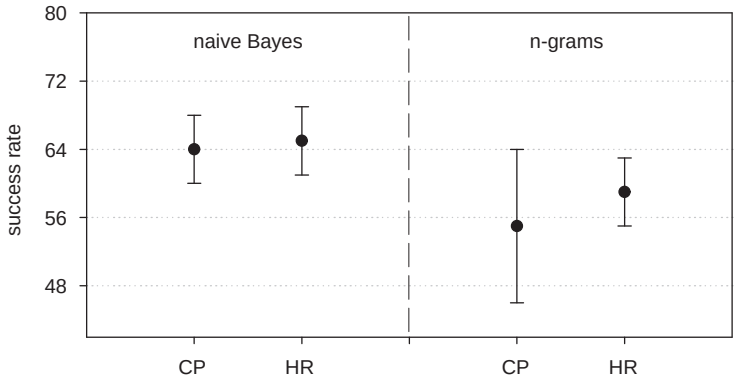
	naïve Bayes		n-grams		
	B	M	$n=2$	$n=3$	$n=4$
Chord progressions	68 ± 10	66 ± 10	$\mathbf{73 \pm 10}$	67 ± 10	65 ± 10
Chords and rhythm	$\mathbf{72 \pm 6}$	$\mathbf{72 \pm 7}$	68 ± 9	67 ± 6	67 ± 7

Table 5.4: Results obtained in the classification of Baroque and Classical versus Romantic music using chord progressions and the same sequences including harmonic rhythm.

Considering these results, it was decided not to continue using this encoding, because it did not provide any improvement over the chord progressions encoding and, moreover, it is necessary to know the metrical structure of the piece in order to be able to use it. However, it cannot be concluded that harmonic rhythm is not useful to build better musical models, because just one simple representation has been tested. In the future, different representations of rhythm should be explored, in order to find one more suitable for this task.



(a) 3 genres



(b) 9 genres

Figure 5.4: Comparison of the best results obtained with chord sequences encoded as chord progressions (CP) and the same sequences including harmonic rhythm (HR).

5.4 Classification of audio music

Harmonic information, in spite of being a useful description of music, is a kind of information difficult to find. Although it is easy to retrieve large amounts of chord sequences from the Internet for many different genres, they are not usually reliable given that labelling chords in songs by ear is a difficult task even for trained people.

In order to test how the models presented in the previous sections perform when harmonic sequences are obtained from a different source, a similar framework than that in Section 4.3 has been established. The same audio files generated in those experiments have been used, but this time a different transcription algorithm has been used to obtain harmonic sequences from the audio files.

Several chord extraction systems exist able to extract chord sequences, either at frame or beat level, from an audio signal. These works have reported good results using Hidden Markov Models with chromagrams extracted from real recordings of the Beatles (Bello and Pickens, 2005; Sheh and Ellis, 2003), with chord recognition rates around 75%. Using similar techniques and characteristics, Lee and Slaney (2006) obtained a 92% chord recognition rate with classical music, but using the Melisma Music Analyzer[1] to build the ground truth, instead of using hand labeled samples. Another common feature in these works is the fact that they only use small subsets of chord names, with a vocabulary size ranging from 24 (major and minor triads) to 36 (adding diminished triads). Although in (Sheh and Ellis, 2003) the original vocabulary comprised 147 possibilities, including triads and seventh chords, only 32 from the original 147 were found in the data set.

In this chapter a chord extraction algorithm, developed by members of the Music Technology Group from the University Pompeu Fabra (Barcelona) will be used. This algorithm is based on the computation of harmonic pitch class profiles, and will be outlined in the next section. One advantage of this system is that it is able to recognize a rich set of chords, including major and minor sevenths, equivalent to the *4-note* feature set described in Section 2.3.1. Recall that, in the experiments performed in the previous sections, this feature set obtained similar results than the feature set containing full harmonic information.

5.4.1 Chord extraction system

The perception of musical pitch has two main attributes: pitch height and chroma. Pitch height moves vertically and tells which octave a note belongs to, while chroma (or pitch class) indicates its position in relation to others within an octave. A chromagram, or pitch class profile, is a 12-dimensional vector representation of a chroma, which represents the relative intensity in

[1] http://www.link.cs.cmu.edu/music-analysis/

each of twelve semitones in a chromatic scale. Since a chord is composed of a set of tones, and its label is only determined by the position of those tones in a chroma, regardless of their heights, chromagram seems to be an ideal feature to represent a musical chord.

The algorithm used in this chapter is based on the computation of harmonic pitch class profiles (HPCP). The HPCP is an enhanced pitch class distribution (or chroma) feature, computed in a frame-by-frame basis using just the local maxima of the spectrum within a certain frequency band. For this, the algorithm described in (Gómez, 2006) is applied, which distributes spectral peak contributions to several adjacent HPCP bins and takes peak harmonics into account. In addition to use the local maxima of the spectrum, HPCPs are tuning independent (i.e. the reference frequency can be different from the standard tuning) and consider the presence of harmonic frequencies. The resulting HPCP is a 36-bin octave-independent histogram representing the relative intensity of each 1/3 of the 12 semitones of the equal tempered scale. The detailed description of the algorithm can be found in (Gómez, 2006).

The feature vectors are then processed to obtain a symbolic representation consisting of a sequence of chords in the form of triads (e.g. Am) or extended triads (e.g Am7). Thus, each song is represented as a string of chord progressions, within an alphabet of symbols. Two different alphabets have been considered: major and minor triads (24 symbols) and major and minor seventh chords in addition to the major and minor triads (72 symbols). These two alphabets are equivalent to the ones used in the *major-minor* (24 symbols) and *4-note* (72 symbols) feature sets described in Section 2.3.1.

5.4.2 Experiments

In this section the results obtained using the output of the chord extraction system with the *Perez-9-genres* corpus are presented. In these experiments, a similar methodology than in Section 4.3 was used. Using the set of audio files synthesized from the MIDI sequences, the chord extraction algorithm was applied to obtain the chord sequences, with a resolution of one chord per frame. Then, the resolution of these sequences was converted into beat level by using a fixed window of 500 ms (equivalent to a tempo of 120 bpm). This was done by selecting the chord with highest probability from all the chords detected in the frames within the window. Using a fixed time window for all the songs in the corpus is not an optimal solution, because many pieces may have a different tempo than the one used here, but it is a compromise solution that has proven to perform quite well. In previous experiments, the *BeatRoot* beat tracking algorithm (Dixon, 2007) was used at this point, but it did not show any improvement over the results presented here.

This process was performed twice, using the two chord vocabularies equivalent to the *4-note* and *major-minor* feature sets. Finally, these

sequences were encoded also as chord names and degrees, resulting in a total of four feature sets extracted from the audio files. Note that for encoding chords as degrees it is necessary to know the tonality of the song, but this information is not available in the audio files, so the key estimation algorithm described in (Temperley, 1999) was used.

Root	Extensions	naïve Bayes		n-grams		
		B	M	$n=2$	$n=3$	$n=4$
Degrees	4-note	65 ± 6	70 ± 4	79 ± 6	$\mathbf{81 \pm 4}$	80 ± 5
	Major-minor	61 ± 6	57 ± 8	67 ± 4	$\mathbf{74 \pm 6}$	71 ± 2
Chord names	4-note	75 ± 6	74 ± 6	$\mathbf{88 \pm 3}$	87 ± 3	86 ± 3
	Major-minor	62 ± 7	63 ± 5	74 ± 4	$\mathbf{76 \pm 4}$	71 ± 3

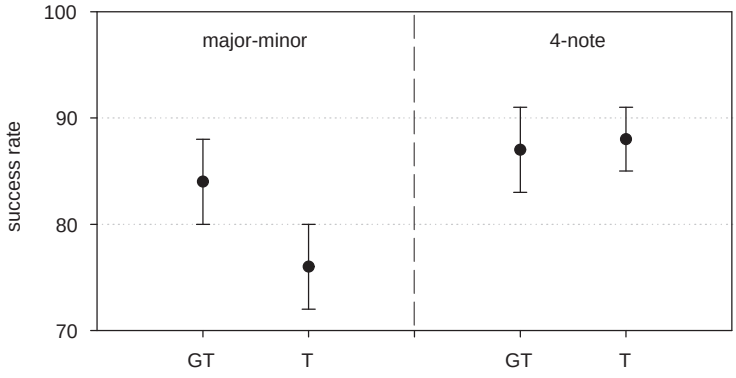
(a) 3-genres

Root	Extensions	naïve Bayes		n-grams		
		B	M	$n=2$	$n=3$	$n=4$
Degrees	4-note	44 ± 5	46 ± 9	$\mathbf{58 \pm 7}$	57 ± 10	56 ± 10
	Major-minor	38 ± 10	32 ± 9	45 ± 8	$\mathbf{50 \pm 8}$	48 ± 5
Chord names	4-note	59 ± 4	63 ± 6	$\mathbf{67 \pm 8}$	64 ± 5	64 ± 5
	Major-minor	40 ± 7	36 ± 8	50 ± 7	$\mathbf{52 \pm 9}$	49 ± 9

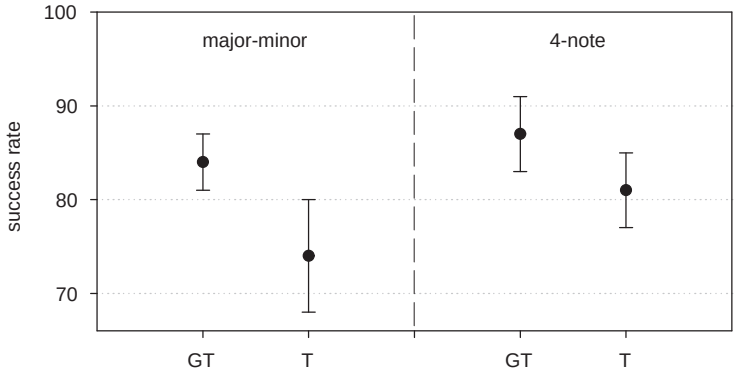
(b) 9-genres

Table 5.5: Classification results for corpus *Perez-9-genres* using harmonic sequences obtained from synthesized audio.

Table 5.5 shows the results obtained for the three and nine-genres tasks, and a comparison of the best results obtained for each feature set using both the ground truth and the transcribed sequences are shown in Figures 5.5 and 5.6. As it can be seen, very good results have been obtained using the 4-note chord names, reaching classification rates similar to those of the ground truth in the three and nine-genre tasks. Although it was expected that these results would be lower due to the errors introduced in the chord sequences by the transcription system, the results show the opposite trend. This surprising behaviour has, however, a simple explanation. It must be noted that the audio files have been obtained from MIDI files containing *interpretations* of the original pieces, and these interpretations are highly influenced by the style of the pieces. For example, two musicians from different styles, as jazz and pop, would not play the same chord progression in the same way. In jazz music, it is usual to introduce ornaments, playing notes or chords that are not in the original chord progression. Thus, there is more information on the genre of the pieces implicit in the audio files than in the chord sequences of the ground truth.

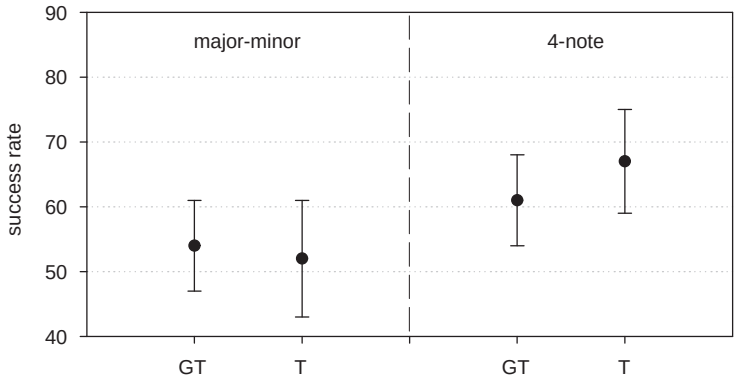


(a) chord names

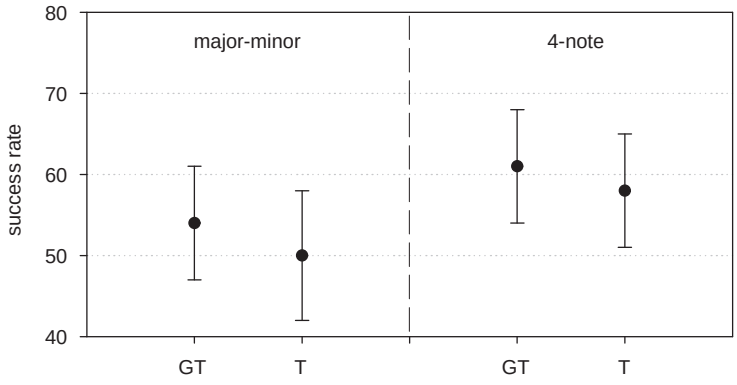


(b) degrees

Figure 5.5: Comparison of the best results obtained in the *three-genres* task with the ground truth of chord progressions (GT) and the transcribed chord sequences (T) using a chord extraction algorithm.



(a) chord names



(b) degrees

Figure 5.6: Comparison of the best results obtained in the *nine-genres* task with the ground truth of chord progressions (GT) and the transcribed chord sequences (T) using a chord extraction algorithm.

On the other hand, the results obtained with the relative encoding are poorer than those with the chord names. Recall that the system needs the tonality of the song for transforming each chord to its corresponding degree, and the tonality is also obtained from the audio using a key estimation algorithm. In this task, the algorithm used had an error rate of 32.2%, from which only a 6.5% were relative major/minor mistakes (e.g. estimating the tonality Am instead of C or vice versa). This error propagates through the encoding to the chord sequences, and finally to the models built upon them, resulting in poorer classification results.

5.5 Melodic and harmonic ensembles

In all the experiments presented before, the performance of two classification methods — naïve Bayes and n-gram models — has been tested, using symbolic sequences taken from different sources, either directly encoding symbolic music files containing melodic or harmonic sequences, or by using a transcription algorithm to obtain these sequences from a set of audio files. In those experiments, variable results have been obtained regarding the configuration of each classifier. In some cases best results were achieved using the naïve Bayes classifier, while in others the n-gram models performed the best. More variability can be also found in the performance of each classifier. For naïve Bayes, it is not clear which of the statistical models studied is the best, and it cannot be neither determined an optimal vocabulary size when doing feature selection. For the n-gram models, different values of n achieved the best results in different experiments.

The same variability has been observed regarding the different encoding formats used for melodic and harmonic sequences. In Table 5.6 the best results obtained for the *Perez-9-genres* corpus using both representations are summarized, along with the combination of classification technique and encoding format for which that results were obtained. Although in this table only the best results are shown, recall that similar results were also obtained using different configurations of the parameters, and it cannot be known with a high certainty whether this behaviour would be the same with a different data set.

Under the light of these results, it seems clear that a reasonable way to tackle this uncertainty is to use an ensemble of classifiers (see Section 3.3). This technique allows to combine the decisions taken by different classifiers, and it has shown the property that it usually obtains similar results — if not better — than the best single classifier in the ensemble (Moreno-Seco et al., 2006). This way, a more robust classifier can be built based on the decisions taken by the different techniques used in this thesis, reducing the risk of choosing the wrong method for new data sets.

Representation	Source	Success (%)	Classifier	Encoding
Melody	symbolic	84 ± 3	4-grams	decoupled
	audio	75 ± 5	naïve Bayes (M)	coupled
Harmony	symbolic	87 ± 4	2-grams	chord–4-notes
	audio	88 ± 3	2-grams	chord-4-notes

(a) 3-genres

Representation	Source	Success (%)	Classifier	Encoding
Melody	symbolic	64 ± 2	4-grams	decoupled
	audio	54 ± 6	naïve Bayes (M)	coupled
Harmony	symbolic	64 ± 4	naïve Bayes (B)	chord–full
	audio	67 ± 8	2-grams	chord-4-notes

(b) 9-genres

Table 5.6: Best results in the genre classification task for the *Perez-9-genres* corpus using different sources of information: melodic or harmonic sequences, obtained directly from symbolic files or by applying a transcription algorithm to audio files.

Another important advantage of this technique is that it allows to combine different decisions based on various data sources. Until now, all the experiments performed have been done using a single source of information: either melodic or harmonic sequences. By using an ensemble, both sources can be combined, and this allows to study whether this combination can help to achieve better results than when using each representation separately.

In order to test this technique, a new set of experiments were performed using the *Perez-9-genres* corpus. These experiments were divided in two groups. In the first set of experiments, the data sequences were obtained from symbolic music files, and in the second set the transcribed sequences obtained from the audio files were used. This way, it can be studied the expected performance of the system depending on what kind of data is available.

5.5.1 Symbolic sources

In these experiments, the decisions taken by some of the classifiers shown in Sections 4.2.5 and 5.2 were combined, using the combination method explained in Section 3.3. This technique requires some diversity in the decisions taken by the different classifiers, so different combinations of classification and data encoding methods have been selected, in order to ensure this diversity. Ten different classifiers were selected, five of them using melodic sequences, and the other five using chord progressions:

- Melodic sequences:

  - naïve Bayes (B)–coupled
  - naïve Bayes (M)–coupled
  - 2-grams–coupled
  - 3-grams–decoupled
  - 4-grams–decoupled

- Harmonic sequences:
  - naïve Bayes (B)–4-note chords
  - naïve Bayes (M)–4-note chords
  - 2-grams–4-note chords
  - 3-grams–4-note chords
  - 4-grams–4-note chords

These single classifiers were selected because they obtained some of the best results in the previous experiments. However, when building the ensembles, each classifier must be assigned a weight based on how it performed with a different data set, and it would be incorrect to compute those weights based on the results obtained in the previous experiments to classify again the same data set. For this reason, the experiments in this section were performed as follows:

1. First, the data sets were divided in two: a 90% of the files was used to train the ensembles (i.e. to compute the classifier weights), and the remaining 10% was kept for testing purposes.

2. Using the training set (90% of the files), all the classifiers listed above were evaluated using a 10-fold cross-validation. Once all the classifiers were evaluated, each one of them was assigned a weight based on the number of errors committed with this training set.

3. In order to obtain the decisions for each classifier, they were trained once again, this time using the whole training set, and then they were evaluated using the remaining 10% of the files.

4. Finally, the decisions obtained in the previous step were combined using the weights computed in step 2, reaching a single decision for each file in the test set.

5. Steps 1-4 were repeated 10 times, using different splits of the data sets, until all the files were used for testing the ensembles.

Using the results of these experiments, three different ensembles were built: one combining the decisions of the melodic classifiers only, another one using just harmonic classifiers, and an ensemble of melodic and harmonic

classifiers. This way, it can be studied whether the combination of melodic and harmonic information results in an improvement of the classification results. For each ensemble, the two voting methods explained in Section 3.3 were used: Best-Worst Weighted Vote (BWWV) and Quadratic Best-Worst Weighted Vote (QBWWV).

The results obtained in these experiments are shown in Table 5.7. As it can be seen, the combination of melodic and harmonic information achieved the best classification rates, with a significant difference over the ensembles using melodic classifiers only in the three-genres task. However, the difference with the harmonic ensembles cannot be considered significant. Comparing these results with the best of the singles classifiers shown in Table 5.6, it can be seen that the ensembles built with either melodic or harmonic classifiers achieved classification rates similar to that of the best single classifiers, and only when comparing the combination of melody and harmony with the best melodic classifier, a significant improvement was obtained.

When comparing the results obtained in the nine-genres task, any significant differences can be found, even when comparing them to the results of the best single classifiers. However, the best result was achieved again using the combination of melodic and harmonic classifiers.

	Melodic ensemble		Harmonic ensemble		Melody & harmony	
	BWWV	QBWWV	BWWV	QBWWV	BWWV	QBWWV
3-genres	83 ± 4	84 ± 4	88 ± 4	88 ± 4	$\mathbf{90 \pm 3}$	89 ± 4
9-genres	65 ± 3	64 ± 4	62 ± 5	60 ± 8	$\mathbf{68 \pm 6}$	66 ± 4

Table 5.7: Results obtained for the sequences extracted from the symbolic files in the *Perez-9-genres* corpus, using ensembles of classifiers with two different weighting methods.

5.5.2 Audio sources

In order to find the best classification scheme when dealing when audio music files, the same experiments than in the previous section were performed, this time using the melodic and harmonic sequences obtained from the synthesized audio files, using the transcription algorithms described in Sections 4.3 and 5.4. Again, the ten classifiers which performed the best with the transcribed melodic and harmonic sequences were selected:

- Melodic sequences:
  - naïve Bayes (B)–coupled
  - naïve Bayes (M)–coupled

  - 2-grams–decoupled
  - 3-grams–decoupled
  - 4-grams–decoupled

- Harmonic sequences:
  - naïve Bayes (B)–4-note chords
  - naïve Bayes (M)–4-note chords
  - 2-grams–4-note chords
  - 3-grams–4-note chords
  - 4-grams–4-note chords

The results obtained in these experiments are shown in Table 5.8. It can be seen that the ensembles built using either melodic or harmonic sequences obtained similar results than their respective best single classifiers. Again, the advantage of this technique is shown: by using an ensemble the risk of choosing a wrong classifier is avoided and the results obtained are always similar to those of the best classifier. However, this advantage is not enough to overcome the poor results obtained when using the melodic sequences transcribed from the audio files, as these results were significantly lower than those obtained using the extracted chords. Finally, the combination of melodic and harmonic classifiers did not shown any significant improvement. In this case the results obtained were comparable to those of the harmonic ensembles.

	Melodic ensemble		Harmonic ensemble		Melody & harmony	
	BWWV	QBWWV	BWWV	QBWWV	BWWV	QBWWV
3-genres	74 ± 4	74 ± 3	87 ± 3	87 ± 3	$\mathbf{89 \pm 3}$	87 ± 3
9-genres	54 ± 6	54 ± 7	67 ± 6	$\mathbf{68 \pm 8}$	67 ± 6	67 ± 6

Table 5.8: Results obtained for the transcribed sequences from the synthesized files of the *Perez-9-genres* corpus, using ensembles of classifiers with two different weighting methods.

5.6 Conclusions

In this chapter, modeling of musical genre using harmonic sequences has been studied. These models have been evaluated in a genre classification task using a naïve Bayes classifier and n-gram models, and their results have been compared in order to find the best combination of classification method and encoding format for this task.

As a first step, harmonic sequences were encoded as chord progressions, using different feature sets for encoding the individual chords, each one of them providing different levels of information. Regarding the root of the chords, they were encoded using absolute and relative encodings (i.e. using the chord name or the degree relative to the key of the song), but any significant differences were found in these experiments. Regarding the structure of the chords, four different feature sets were used. In these experiments it was shown that the full structure of the chords is not necessary to build accurate models of musical genre, as similar results were obtained using a simpler set of chords encoding just major and minor triads, and major and minor seventh chords. Using these two feature sets, an 87% accuracy rate was achieved in the three-genres problem, a result even higher than the best obtained in the previous chapter using melodic sequences. When using only triad chords, however, the results obtained were slightly lower, but very satisfactory anyway taking into account the simplicity of the vocabulary used.

An attempt to improve these models using rhythm information was made, extending the chord progressions with a representation of harmonic rhythm. However, these experiments showed a similar performance than those using just chord progressions, so this encoding is not recommended because it does not provide any benefit in the classification task. Moreover, this encoding method is more complex because it needs to know the metrical structure of the piece, and this information is not always available.

The models built from chord progressions were also tested using a corpus of audio files, using a state-of-the-art chord extraction algorithm to obtain the chord sequences from the audio files. Excellent results were obtained in these experiments, reaching classification rates equal to those obtained with the ground truth of chord progressions, despite the transcription errors introduced by the chord extraction algorithm.

Finally, a classification framework using classifier ensembles was tested, combining the decisions of different classifiers and different representations for musical sequences. In these experiments, the models built in this chapter were combined with those built in the previous chapter using melodic sequences. These ensembles did not obtain a significant improvement over the best single classifiers. However, they have shown to be a robust classification method, because they always obtained near-optimal results without the risk of choosing the wrong classifier.

6

Composer style modeling

In the previous chapters it has been shown how pattern recognition techniques can help in the task of classifying music by genre, using different sources of information. This chapter presents a different application of the same techniques. As discussed in Chapter 1, modeling of musical style can be viewed from different viewpoints, depending on how the concept of *style* is interpreted. In this chapter, a different meaning for style will be assumed, modeling the musical style of different composers in order to solve an authorship attribution problem for some disputed musical pieces.

The task of studying the authorship of a work is known as *stylometry*, and it has been traditionally done by experts in the field "by hand", looking for characteristic traits inside an anonymous or disputed work that could reveal the identity of its author.

Since the introduction of machine learning techniques to this task, good results have been obtained in the study of the authorship of texts, and recently also of musical pieces, although the case of text attribution has been much more thoroughly studied. An extensive review of stylometry applied to textual data can be found in (Koppel et al., 2008). In all those works, the key point for performing a good analysis is choosing proper features that are able to capture the style of the data analyzed. Many features have been proposed which make use of linguistic knowledge, such as syntax and parts-of-speech, but also very good results have been obtained using character n-grams. This will be the approach in this chapter. Language models built from a training corpus will be used in order to evaluate the disputed works.

In the musical domain, less works on stylometry can be found that make use of pattern recognition techniques. They are discussed in detail in Section 6.1. In order to test the method proposed here, it has been applied to solve the problem posed in one of those works: the attribution of the authorship of some disputed J. S. Bach's fugues. In Section 6.2, the proposed method is tested using a corpus of five composers with relatively different styles. Then, in Section 6.3 the disputed fugues are compared with models built from a set of candidate composers, in order to determine their authorship.

6.1 Previous work

Little work has been done in the modeling of composer styles, and it has been mainly done in the audio domain. In the several editions of the Music Information Retrieval Evaluation eXchange (MIREX), an *audio artist identification task* has been proposed, including in 2007 and 2008 a *classical composer identification subtask*. In this task, experiments were performed using a data set of 30-second audio clips from 11 composers: Bach, Beethoven, Brahms, Chopin, Dvorak, Handel, Haydn, Mendelssohnn, Mozart, Schubert, and Vivaldi. The best results reported are around a 53% accuracy (Mandel and Ellis, 2008), using Support Vector Machines on spectral features extracted from the audio. It is remarkable that, for all the algorithms presented, Bach is one of the composers for which better results were obtained, just a step behind Chopin.

In the symbolic domain, we can find the works by van Kranenburg (2006) and Ogihara and Li (2008), using melodic and harmonic information respectively.

In (Ogihara and Li, 2008), the authors explore the capabilities of n-grams of chord progressions to characterize the style of several jazz musicians and The Beatles. Songs are encoded using n-gram profiles, where each n-gram is weighted using its relative duration measured in beats over the whole sequence. Then, the cosine of the product of two profiles is used as a similarity measure to study the separability between the different composers and their links, using a hierarchical clustering.

The authors also study different levels of chord information encoding, using chord triads, 6th and 7th chords, and extensions (9th, 11th and 13th). They conclude by selecting 20 *style markers* (4-grams of 7th chords) as the best to characterize the eight styles studied. However, no classification is performed to empirically support their conclusions.

Special attention will be paid to the work of van Kranenburg. In (Backer and van Kranenburg, 2005; van Kranenburg, 2006), the authors perform a stylometric study on some organ fugues in the J. S. Bach (BWV) Catalogue, for which the authorship is disputed. For this, a set of features was developed based on musicological criteria. Fugues are a particular style of composition in which a main theme is developed in several voices, which imitate each other in a way similar to a canon. This kind of composition is highly polyphonic and there are strict rules that state, for example, the intervals that are allowed and forbidden between voices. For this reason, the 20 features selected by the authors refer mainly to the polyphonic relationships between voices. They include:

- Vertical intervals weighted by duration
- Parallel motion

- Dissonance treatment
- Voice density
- Entropy measures
- Stability of the time intervals between changes

These features are extracted using a 30 bars sliding window. Then, a feature selection is performed in order to select the set of features that contribute the most to discriminate between the training data sets, made up of compositions from the catalogue of the candidate composers. Finally, all the windows extracted from the piece under study are classified using a nearest neighbor classifier, and the individual decisions are combined to reach a final decision.

This framework was first tested with a data set of five composers: Bach, Handel, Telemann, Haydn, and Mozart (van Kranenburg and Backer, 2004), reaching classification rates between 79.4% and 95.2% using several configurations of classes. High error rates were due to the presence of Haydn and Mozart, which are composers with very similar styles.

Once this method proved to be useful for distinguishing composer styles, it was used to study the authorship of nine fugues originally attributed to J. S. Bach, but which were later attibuted by musicologists to other composers such as W. F. Bach, J. L. Krebs, and J. P. Kellner. The results obtained, though not conclusive, support most of those studies and prove that pattern recognition techniques can be used as a complement to "traditional" methods.

However, this method has two main drawbacks. First, the feature selection procedure must be repeated for each configuration of data sets, since not all the features perform the same to distinguish between different composers. And most important, those features are only useful if working with polyphonic compositions: bars that are not strictly polyphonic must be discarded during the encoding process.

In this thesis, a different approach is proposed to overcome those drawbacks, using the encoding and methods used for melodic classification in Chapter 4. In order to test this new approach, the same experiments reported in (van Kranenburg and Backer, 2004) and (van Kranenburg, 2006) will be carried out using the proposed method, and the results will be compared with the ones in those works.

6.2 Composer style classification

The first experiment consists in modeling the styles of the five composers in corpus *Kranenburg-5-styles*: Bach, Handel, Telemann, Haydn, and Mozart. This data set is made up of polyphonic MIDI files with one

track per instrument, containing only melodic information. Thus, the same methodology as in Chapter 4 will be used. However, in the study of the disputed fugues, a closer look will be taken at the output of the classifiers and, as said in Section 3.3, the naïve Bayes classifier tends to produce bad probability estimates, so it was decided not to use it in this task, only n-gram modeling will be used.

The encoding process of the MIDI files was slightly different than that described in Section 2.2. The files in this corpus include fugues, concerts, trios, and quartets and, in general, all the tracks contain rich melodic passages. Also, the melody can be played by different instruments, passing from one track to another, so selecting one track as the one containing the melody would be a difficult and inaccurate task. For this reason, all the tracks in the MIDI files were encoded separately — applying the polyphony reduction to each one — and the resulting strings were concatenated as one single file.

	Decoupled encoding			Coupled encoding		
	$n=2$	$n=3$	$n=4$	$n=2$	$n=3$	$n=4$
Bach vs. Handel	83.3	88.1	**88.9**	86.5	86.5	87.3
Bach vs. Haydn	85.5	93.6	**96.8**	91.9	93.6	93.6
Bach vs. Mozart	90.1	95.0	**97.5**	95.9	95.9	95.9
Bach vs. Telemann	88.3	94.5	**95.3**	94.5	93.8	93.8
Handel vs. Haydn	89.0	92.0	**94.0**	**94.0**	92.0	92.0
Handel vs. Mozart	85.6	**94.9**	92.8	93.8	93.8	92.8
Handel vs. Telemann	87.5	90.4	**93.3**	83.7	83.7	89.4
Haydn vs. Mozart	67.4	70.5	66.3	65.3	**74.7**	68.4
Haydn vs. Telemann	90.2	**98.0**	96.0	94.1	95.1	94.1
Mozart vs. Telemann	86.9	93.9	**98.0**	96.0	96.0	97.0

Table 6.1: Success rates in pairwise classification of corpus *Kranenburg-5-styles* using n-gram modeling.

In order to find the best combination of encoding and n-gram length for this task, pairwise classification was performed between all the classes in the corpus. Leaving-one-out success rates are shown in Table 6.1. As it happened in the genre classification task, best results were obtained using the decoupled encoding, with very high success rates for most of the pairs. The results for the *Bach vs. other* pairs were very satisfactory, and suggest that this method is suitable for differentiating the style of J. S. Bach from other composers, and thus for the authorship attribution task which will be studied in the next section.

Next, another set of experiments was performed in order to compare this method with the one used in (van Kranenburg and Backer, 2004), using the same configuration of classes. The data sets used in each experiment are shown in Table 6.2, and the results for both methods are

shown comparatively in Table 6.3. Although the results obtained with the n-grams were poorer for most of the data sets, they are quite good considering that a general purpose encoding has been used, compared to the other specialized feature set, and the good results obtained in pairwise classification encouraged to continue to the next step.

data set	classes
I	{Bach}, {Telemann}, {Handel}, {Haydn}, {Mozart}
II	{Bach}, {Telemann}, {Handel}
III	{Bach}, {Telemann, Handel}
IV	{Bach}, {Telemann, Handel, Haydn, Mozart}
V	{Telemann}, {Handel}
VI	{Haydn}, {Mozart}
VII	{Telemann, Handel}, {Haydn, Mozart}

Table 6.2: Combination of classes in each data set.

data set	4-grams	van Kranenburg
I	78.8	**80.1**
II	87.2	**93.0**
III	88.3	**95.2**
IV	89.4	**94.0**
V	**93.3**	91.6
VI	66.3	**79.4**
VII	**95.0**	93.5

Table 6.3: Success rates using 4-grams and the decoupled encoding (left) compared with those obtained by van Kranenburg (right).

6.3 Authorship attribution

In this section, the experiments on authorship attribution with corpus *Kranenburg-fugues* are presented (see Section 2.1.6). This task consists in trying to clarify, using pattern recognition methods, the disputed authorship for some of the fugues in the J. S. Bach catalogue. During the 20th century, some authors have questioned the original source of those works using traditional methods, based on finding characteristic patterns in common with compositions of undisputed authorship. The goal of this work is to find empirical evidence to support those conclusions by using language modeling tools. However, as stated by van Kranenburg (2006), the results obtained with pattern recognition methods on this data set should be handled cautiously, because it is possible that the real author of the pieces was someone different to the four composers considered.

Before studying the authorship of the disputed pieces, an experiment was performed with the undisputed pieces of the four composers, in order to see to what extent the method proposed can help in this task. All the MIDI files were encoded using all the available tracks, and classification was done using 4-gram models and leaving-one-out cross-validation. The confussion matrix for this experiment is shown in Table 6.4. As it can be seen, all the pieces from W. F. Bach and J. P. Kellner were misclassified, and assigned mainly to J. S. Bach. Although this is a discouraging result, it must be taken into account that only a small set of files has been used, and it is a sensible result as J. S. Bach was a big influence in the styles of the other composers.

	J. S. Bach	W. F. Bach	J. L. Krebs	J. P. Kellner
J. S. Bach	7	0	2	0
W. F. Bach	4	0	1	0
J. L. Krebs	2	0	6	0
J. P. Kellner	4	0	2	0

Table 6.4: Confussion matrix for the classification between J. S. Bach, W. F. Bach, J. L. Krebs, and J. P. Kellner. In the columns, the number of files assigned to its corresponding class are shown.

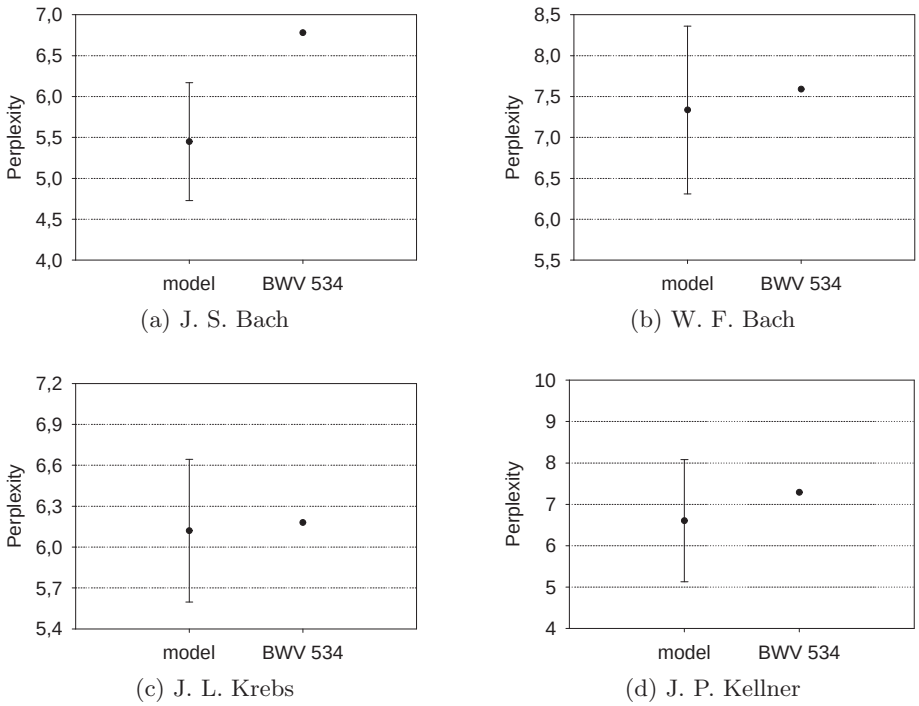
It is interesting to see that the classifier performed quite well when distinguishing between J. S. Bach from J. L. Krebs. As Krebs is one of the candidates for some of the disputed pieces, we can expect plausible conclusions from their analysis. In the next sections, each one of the disputed pieces is analyzed, evaluating them against the models built from the composers which have been proposed as their legitimate authors in the musicological literature (van Kranenburg, 2006). The results in this thesis will be also contrasted with the conclusions in that work.

6.3.1 BWV 534/2

The Fugue in F minor (BWV 534/2) was first rejected as a composition by J. S. Bach in 1985, and later attributed to his son, W. F. Bach. However, van Kranenburg rejected W. F. Bach as the actual composer, being J. L. Krebs or J. P. Kellner the composers with highest probability, although the results were not conclusive.

Figure 6.1 shows the evaluation of the piece using the models built from the four composers. For each model, the perplexity for the undisputed pieces has been computed using a leaving-one-out estimator, and the average value is plotted along with the standard deviation. Then, the disputed fugue has been evaluated using the whole model for the composer, and its perplexity is

plotted in order to show whether it falls within the variance of the composer. If so, that composer can be accepted as a possible author for the piece.



(a) J. S. Bach (b) W. F. Bach

(c) J. L. Krebs (d) J. P. Kellner

Figure 6.1: Evaluation of Fugue BWV 534/2 against the models of J. S. Bach (a), W. F. Bach (b), J. L. Krebs (c), and J. P. Kellner (d).

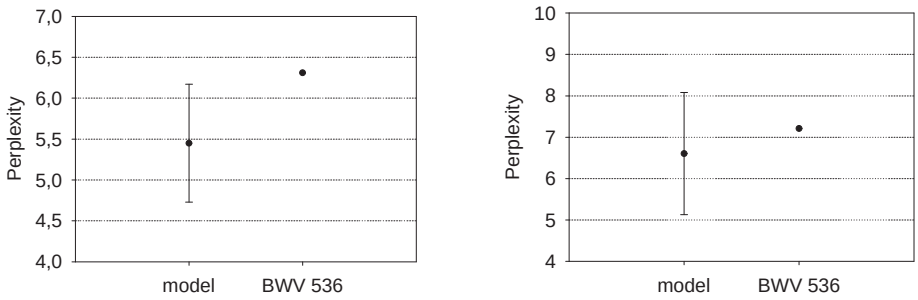
From these results, it seems safe to reject Fugue BWV 534/2 as a composition of J. S. Bach (see Figure 6.1a). It is not clear which of the other composers is the most probable author, although it seems closer to the style of J. L. Krebs, as suggested by van Kranenburg. However, his hypothesis that W. F. Bach is not the author should be rejected according to figure 6.1b.

6.3.2 BWV 536/2

The Fugue in A major (BWV 536/2), rejected as a J. S. Bach composition in 1989, has been claimed to be a work by J. P. Kellner. Van Kranenburg, on the contrary, did not support this hypothesis, but he also found that this is not a typical J. S. Bach fugue.

In Figure 6.2a it can be seen that, effectively, this piece deviates from the typical Bach's fugues, though the difference is not very high. When comparing it with J. P. Kellner's pieces (Figure 6.2b), it can be seen that

it is in fact probable that he was the actual composer, as it falls within the perplexity range of Kellner's genuine fugues.



(a) J. S. Bach (b) J. P. Kellner

Figure 6.2: Evaluation of Fugue BWV 536/2 against the models of J. S. Bach (a) and J. P. Kellner (b).

6.3.3 BWV 537/2

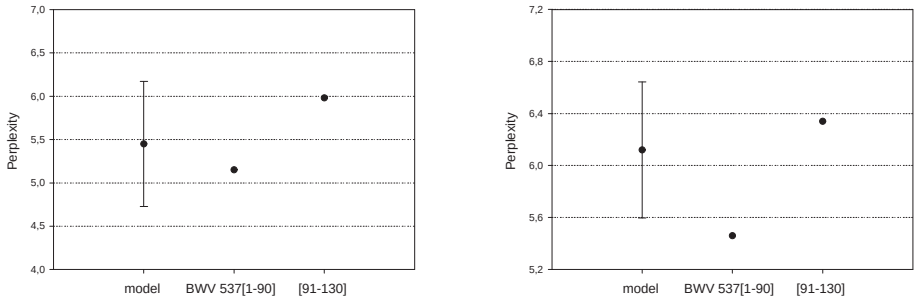
The Fugue in C minor (BWV 537/2) is a special case within the disputed fugues under study. There are studies that claim that it is an original work by J. S. Bach, but he left it unfinished. Some time later, J. L. Krebs would have finished this piece, adding 40 more bars to the 90 composed by Bach. This theory was supported by van Kranenburg, who showed how this piece evolves from the style of Bach to that of Krebs in the last 40 bars.

In order to test this hypothesis, the piece has been splitted in two, evaluating the first 90 bars and the last 40 separately with the styles of J. S. Bach and J. L. Krebs. The results are shown in Figure 6.3. In this figure, the same trend described by van Kranenburg can be observed. The first part of the piece can be seen as a typical Bach fugue, and it falls outside the style of Krebs. On the other hand, for the last 40 bars it is probable that Krebs was the author, although Bach cannot be discarded either.

6.3.4 BWV 555/2, 557/2, 558/2, 559/2, and 560/2

These fugues have been grouped together because they are part of a collection named *Acht kleine Präeludien und Fuguen*. This collection is very controversial due to its low quality, and some authors have rejected J. S. Bach as its composer, proposing others candidates, with J. L. Krebs among them. In his study, van Kranenburg rejected J. S. Bach and W. F. Bach, and proposed J. L. Krebs as the possible author, although he did not reject the possibility that this collection belongs to other anonymous composer.

For evaluating this collection, all the pieces have been evaluated together against the models of the four composers considered. The results are shown



(a) J. S. Bach (b) J. L. Krebs

Figure 6.3: Evaluation of the two parts of Fugue BWV 537/2 (bars 1–90 and 91–130) against the models of J. S. Bach (a), and J. L. Krebs (b).

in Figure 6.4. From these results it can be concluded that neither J. S. Bach nor J. L. Krebs were the composers of the collection. Although these pieces are closer to the style of W. F. Bach and J. P. Kellner, it is probable that they belong to other composer.

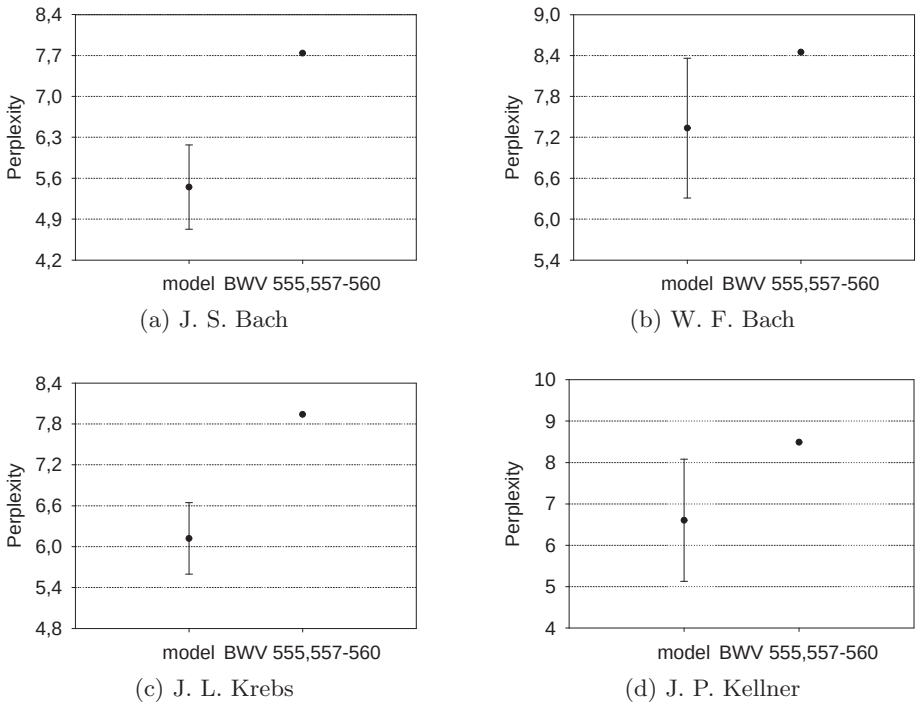
6.3.5 BWV 565/2

The last fugue in this study, Fugue in D minor (BWV 565/2), is an organ piece which has been rejected as a J. S. Bach's work because it differs significantly from all his other organ compositions. One of the candidates for this fugue is J. P. Kellner. The results obtained by van Kranenburg for this piece were not conclusive, because it shares characteristics with both composers: it is harmonically closer to the style of Kellner, but the rhythm resembles more the style of Bach.

Figure 6.5 shows the results of the evaluation of the piece with both composers. The style of this piece is close to that of both J. S. Bach and J. P. Kellner, and none of them can be stated to be more probable than the other.

6.4 Conclusions

In this chapter, it has been shown how n-gram models can be used to model the style of different composers. When applied to the classification of composers with differentiated styles, this method has proved to be very effective. Classification rates for pairs of composers in different genres (Baroque and Classical) stayed very high, ranging from 94% to 98%. However, composers in the same genre are more difficult to distinguish, and accuracy in classification dropped close to 75% for the pair Haydn and Mozart, authors that the experts agree to consider very close in their



(a) J. S. Bach (b) W. F. Bach

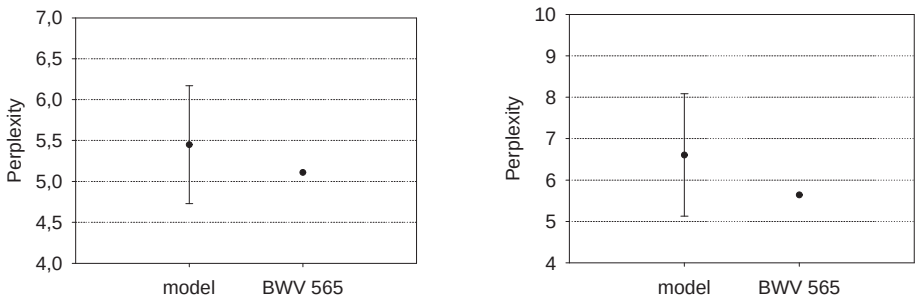
(c) J. L. Krebs (d) J. P. Kellner

Figure 6.4: Evaluation of Fugues BWV 555/2, 557/2, 558/2, 559/2, and 560/2 against the models of J. S. Bach (a), W. F. Bach (b), J. L. Krebs (c), and J. P. Kellner (d).

composing style. This difficulty increases as the styles of the composers get closer, as it was made evident in the authorship attribution task. Using a data set of pieces in the same style and with the same structure, classification of pieces belonging to different composers was not possible.

However, this study has shed some light on the attribution of authorship of the disputed pieces. Although the results obtained are not conclusive, it has been proved that the rejection of J. S. Bach as the composer of some of the pieces is well-founded. Also, some of the hypothesis suggested in the literature using traditional studies have been corroborated. The experiments in this chapter also support the conclusions reached by van Kranenburg, but the features used here are much simpler than theirs.

The encoding method used in this thesis has some advantages over other more sophisticated methods. There is no need to perform a complex analysis of the melody for extracting the features, and it can be used for any kind of music, not necessarily polyphonic. On the other hand, compared to the set of features proposed by van Kranenburg, this method has the drawback that it does not provide any insight on how the style of one composer differentiates



(a) J. S. Bach (b) J. P. Kellner

Figure 6.5: Evaluation of Fugue BWV 565/2 against the models of J. S. Bach (a), and J. P. Kellner (b).

from other. It can serve to evaluate the similarities between styles, but no deeper analysis can be extracted from the results.

7

Conclusions and future work

In this thesis, it has been shown that it is possible to model different aspects of musical style using a supervised learning approach. For this purpose, two pattern recognition techniques traditionally used in natural language processing tasks have been used, establishing an equivalence between music and text by using an appropriate encoding method to transform musical sequences into text documents.

The main contributions of this work can be summarized in the following points:

- The construction of a new corpus of musical genres, containing melodic and harmonic information for all the pieces. This corpus is available under request in order to ensure the reproducibility of the results presented in this thesis, and also to provide a benchmark for future research in musical modeling tasks. It is tagged by genre and melody track. This corpus poses a more complex problem to study than many corpora used in previous works, as it contains more genres with closer relationships than the corpora used in those works.

- In previous works, modeling of musical genres using n-grams and melodic sequences has been already studied, achieving high classification rates. However, in this work the limitations of this technique have been shown using the new corpus. These results suggest that different sources of information and modeling methods need to be explored.

- It has been shown empirically that it is not necessary to use complete harmonic information in order to build accurate harmonic models of musical genre. Also, it has been shown that these models can be used as a back-end for state-of-the-art chord extraction algorithms from audio files without degrading the performance in classification despite the transcription errors.

- In order to show that the proposed methodology is not tuned to solve a single problem but is of a wider applicability in music information retrieval, a simple method for studying the authorship of disputed musical pieces has been proposed, reaching the same conclusions than

much more sophisticated methods using complex representations of musical knowledge.

These contributions are explained in greater detail in the following summary.

7.1 Summary

In this thesis, music modeling using local descriptions of musical content has been studied. For this purpose, two different representations of musical content have been considered, each one of them describing different aspects of music: melody and harmony. Two pattern recognition techniques have been used to build statistical models from some annotated symbolic corpora using those representations, and finally these models have been evaluated in two different tasks: music genre classification and composer style modeling.

In the genre classification task, experiments were performed using melodic and harmonic representations separately, and finally both approaches were combined by using an ensemble of classifiers. To this end, a new corpus of symbolic music files was built, containing both melodic and harmonic sequences for all the pieces. This corpus has a bigger number of files and genres than other corpora used in previous works, and it has a hierarchical structure of genres that allows to perform experiments using a three-class split of the files, or to make the task more complex using all the subgenres as a nine-class problem. Other corpora used in previous works were also used in this thesis to evaluate the methods proposed, reaching similar results than in those works.

In the experiments with melodic sequences, a similar methodology than that in (Cruz-Alcázar and Vidal, 2008) was followed, using the same data sets and a similar encoding method for melodies. However, the encoding proposed in this thesis has the advantage over the one used in that work that it is capable to deal with "real world" MIDI files (not step-by-step sequenced) without the need of performing a previous quantization or manual cleaning. Like in that work, coupled and decoupled variations for the pitch and duration symbols were considered. In general, better classification rates were obtained using the decoupled encoding. Extending the context for the coupled version by joining together symbols extracted from more than two notes did not help to improve the results, due to the low vocabulary coverage obtained in those cases. Note that combining musical symbols results in a high number of possibilities, and also that the vocabulary size grows exponentially with the length of the subsequences used in the encoding. In order to overcome this problem, further experimentation should be carried out with a much bigger data set. However, it must be taken into account that building a reliable corpus of music files is an arduous and time consuming task, and it is left for future work.

Regarding the classification methods, the naïve Bayes classifier and n-gram models were used. In the melody classification task, best results were obtained using the n-gram models in conjunction with the decoupled encoding. Using this combination of methods, excellent results were obtained with one of the corpus used in (Cruz-Alcázar and Vidal, 2008), reaching a 98% classification rate. However, when evaluating the models with the new corpus, the limitations of these methods came into light. In these experiments classification rates dropped to an 84% in the three-class problem, and to 64% with nine classes, due to the higher number of genres and the presence of closer styles.

Special attention was paid to the naïve Bayes classifier, because this is the first time that it is used for the tasks presented in this thesis. Three different statistical models have been tested: multivariate Bernoulli, mixtures of multivariate Bernoulli, and multinomial. However, any of them showed a better performance than the others. This is a disappointing result because it was expected that the mixture model would perform better, because it is able to model more complex distributions than the other two. This fact, together with the high computational cost of this method, makes it inadvisable for the modeling of musical styles. The effect of a feature selection procedure for this classifier has been also studied. When performing feature selection, better results than with the whole vocabulary were obtained. However, it was not possible to determine the optimal vocabulary size, because the best results for each corpora were obtained with different vocabulary sizes and there does not seem to be an apparent relationship between the characteristics of the corpora and these vocabulary sizes. This is the main weakness of this method because, if a vocabulary size is fixed based on the results obtained with one corpus, unexpected results can occur if a new data set is evaluated.

The use of harmonic sequences for modeling musical genres has been also studied. In these experiments, harmonic sequences were encoded as chord progressions, using different chord vocabularies in order to study how the amount of information available on chord structure affects the classification task. Four different extension sets were used, ranging from a set containing all possible chord extensions, to the simplest one encoding just major and minor triads. All these feature sets obtained similar results in the experiments, and it is remarkable that, when using a vocabulary of just five extensions (major, minor, major seventh, minor seventh, and dominant seventh), the same results than with the full set of extensions were obtained. Absolute and relative encodings for chord roots were also tested, but in this case there were no significant differences in the classification rates.

In order to incorporate rhythm information to the chord progressions, an extension to this encoding was proposed, appending a symbol to each chord change indicating where the change has happened within a bar. This encoding was selected because it represents perceptual aspects of harmonic

rhythm. However, it did not provide any significant improvement over the standard chord progression encoding.

The models presented in this thesis were also tested as a back-end for audio processing systems, in order to explore different sources of musical information. Two different state-of-the-art transcription systems were used: a polyphonic transcription algorithm for obtaining melodic sequences from audio files and a chord extraction algorithm for obtaining chord progressions. Surprising results were obtained with the extracted chord progressions, unexpectedly achieving similar results than with the original symbolic corpus. This behaviour can be attributed to the additional information that is present in the audio files that is not in the symbolic harmonic sequences. The audio files were generated from MIDI files containing interpretations of the songs. These interpretations are usually dependent on the genre and thus provide more information on the genre than it is in the original score. On the other hand, when using the transcribed melodies, the results obtained were significantly lower than with the ground truth due to the errors committed by the transcription algorithm.

All the models used in the experiments described above use information of a single aspect of music, encoding information of melodies or harmonies separately. In order to integrate these models, an ensemble of classifiers was used, to combine the decisions of single classifiers using different methods and descriptions of musical data. The results obtained with the ensembles did not show a significant improvement over the best results obtained with the single classifiers, although they were better in average. However, this technique has proven to provide a robust classification framework, as it always provides classification rates at least equal to those of the best single classifier and reduces the risk of choosing the wrong methods for classifying new data sets.

Finally, in order to show that the methodology presented in this thesis is suitable to model different aspects of musical style, not only for genre classification, a composer style modeling task has been also proposed. A first attempt was done using n-gram models and the decoupled encoding of melodies with a corpus of melodic sequences from different composers. In this task, good results were obtained, but again poorer classification rates were obtained when trying to distinguish composers with close styles. Then, these models were used in a more difficult task, the study of an authorship attribution problem for some disputed pieces of J. S. Bach. The results obtained in these experiments, although not conclusive, support the conclusions obtained with much more sophisticated methods reported in previous works, that need a deep knowledge of the styles of the candidate authors for selecting the appropriate features or *style markers*. The main advantage of the proposed method is that it does not need a complex analysis of the melodies (just pitch intervals and duration ratios are used), and it can be used for monophonic or polyphonic music indistinctly.

7.2 Future lines of work

There are a number of possible lines of work that can be followed to study more in depth the approaches proposed in this thesis. Some of them are related to the tasks presented here and others can be considered a continuation of this work by applying the same methods to other music information retrieval tasks.

In some of the experiments performed in this thesis, the relatively small size of the new corpus presented had a negative impact in the results. This is the case of the experiments perfomed with n-words constructed with more than two notes, or in the experiments using the nine subgenres with chord progressions. In order to overcome this limitation, this corpus should be improved, increasing the number of files per genre. Also, more genres should be added to make the classification task more interesting, with closer relationships between genres.

Regarding the encoding of harmonic sequences, different representations of harmonic rhythm can be explored. For example, a simple encoding similar to the one used for melodies can be used, encoding chord durations in either absolute or relative format.

It would be also interesting to study how the interaction between melodic and harmonic information could help in the classification task. In this thesis, the combination of both melodic and harmonic sequences has been done by combining the decisions of different classifiers trained with different music representations separately. It has been shown, however, that there exists a statistical relationship between chord progressions and the notes in their respective melodies (Paiement, 2008). A more complex music modeling system could profit from this relationship by computing the joint probability of melodic and harmonic sequences together. Prior to this, the sequences in the corpus must be aligned, establishing the necessary links between chords and melodies, and proper encoding formats should be also studied.

Finally, some possible applications of the methods studied in this thesis to other music information retrieval tasks are outlined in the following list:

- *Evaluation of automatic composition systems*: in (Espí et al., 2007), the melodic models presented in Chapter 4 were used in an automatic composition system, in conjunction with the shallow statistical description framework described in (Ponce de León and Iñesta, 2007). These models were used together as a fitness function to evaluate melodies generated by a genetic algorithm. The main problem of this approach is that the generated melodies usually have a lack of a musically coherent structure. The introduction of harmonic models into the system could help to improve the quality of these compositions. This could be done either by analyzing the overall harmonic goodness of the melodies generated or by

dividing the composition process in two steps: first generating chord progressions as a seed, and then using the joint melodic and harmonic models suggested above to generate suitable melodies for those chord progressions.

- *Debugging of polyphonic transcription methods*: as it was discussed in Section 4.3, automatic transcription systems as the one used in this thesis usually commit octave errors, placing notes in an octave different than the actual one. These and other transcription errors could be debugged using a statistical analysis of the output of the system. This way, a genre/style oriented transcription system could be built, automatically discarding sequences with small probability with respect to statistical models built from musical corpora.

- *Automatic segmentation of melodies*: a key point in many music information retrieval systems is the segmentation of the melodic line in musically coherent parts. This process could be carried out by using a statistically trained model of musical segments. Such models could be trained to identify *cut points* within a melody.

- *Motif extraction*: another research line in music information retrieval is the automatic extraction of motifs — short sequences of notes that are representative of the melody and allow to identify it — that can be used as indexes or *thumbnails* in music databases. Statistical analysis of musical sequences combined with automatic segmentation of melodies could be used to build a system for automatic motif extraction.

7.3 Publications

Some parts of this thesis have been published in journals and conference proceedings. Here is a list of papers in chronological order (in brackets, the chapter, or chapters, to which each paper is related):

Journal articles:

- Carlos Pérez-Sancho, José M. Iñesta, Jorge Calera-Rubio (2005). Style recognition through statistical event models. *Journal of New Music Research*, 34:331–340. **[Chapter 4]**

- Carlos Pérez-Sancho, David Rizo, José M. Iñesta (2009). Genre classification using chords and stochastic language models. *Connection Science*, 21:145–159. **[Chapter 5]**

- Carlos Pérez-Sancho, David Rizo, José M. Iñesta, Pedro J. Ponce de León, Stefan Kersten, Rafael Ramírez (accepted). Genre classification of music by tonal harmony. *Intelligent Data Analysis*. **[Chapter 5]**

Book chapters:

- Pedro J. Ponce de León, Carlos Pérez-Sancho, José M. Iñesta (2006). Classifier ensembles for genre recognition. *Pattern Recognition: Progress, Directions and Applications*, pages 41–53. **[Chapters 4 & 5]**

Conference proceedings:

- Carlos Pérez Sancho, José M. Iñesta, Jorge Calera-Rubio (2004). Style recognition through statistical event models. In *Proceedings of Sound and Music Computing, SMC'04*, pages 135–139. Paris, France. **[Chapter 4]**

- Carlos Pérez-Sancho, José M. Iñesta, Jorge Calera-Rubio (2005). A text categorization approach for music style recognition. *Lecture Notes in Computer Science 3523 (Pattern Recognition and Image Analysis, Second Iberian Conference, IbPRIA 2005)*, pages 649–657. Estoril, Portugal. **[Chapter 4]**

- Carlos Pérez-Sancho, Pedro J. Ponce de León, José M. Iñesta (2006). A comparison of statistical approaches to symbolic genre recognition. In *Proceedings of the International Computer Music Conference, ICMC 2006*, pages 545–550. New Orleans, USA. **[Chapter 4]**

- David Espí, Pedro J. Ponce de León, Carlos Pérez-Sancho, David Rizo, José M. Iñesta, Antonio Pertusa (2007). A cooperative approach to style-oriented music composition. In *Proceedings of the International Workshop on Artificial Intelligence and Music, MUSIC-AI*, pages 25–36. Hyderabad, India. **[Chapter 4]**

- Carlos Pérez-Sancho, David Rizo, José M. Iñesta (2008). Stochastic text models for music categorization. *Lecture Notes in Computer Science 5342 (Structural, Syntactic, and Statistical Pattern Recognition, Joint IAPR International Workshop, SSPR & SPR 2008)*, pages 55–64. Orlando, USA. **[Chapters 4 & 5]**

- Carlos Pérez-Sancho, David Rizo, Stefan Kersten, Rafael Ramírez (2008). Genre classification of music by tonal harmony. In *Proceedings of the International Workshop on Machine Learning and Music, MML 2008*, pages 21–22. Helsinki, Finland. **[Chapter 5]**

A

Corpus *Perez-9-genres*

Here is a listing of the musical pieces contained in corpus *Perez-9-genres*, which is presented in detail in Chapter 2. For the academic music, composer and song name are listed for all the pieces. For jazz and popular music, just song names are listed.

Baroque

Bach – Air on the G String
Bach – Anna Magdalena Aria
Bach – Ave Maria (Gounod)
Bach – Benedictus Aria
Bach – Brandenburg Concerto #1 in F, 1st mvt.
Bach – Brandenburg Concerto #1 in F, 2nd mvt.
Bach – Brandenburg Concerto #1 in F, 3rd mvt.
Bach – Brandenburg Concerto #1 in F, Minuetto
Bach – Brandenburg Concerto #2 in F, 1st mvt. Allegro
Bach – Brandenburg Concerto #2 in F, 2nd mvt. Andante
Bach – Domine Deus
Bach – Gloria Intro
Bach – Goldberg Variation #1, Allegro
Bach – Goldberg Variation #2, Andante
Bach – Goldberg Variation #3, Andante
Bach – Goldberg Variations, "Aria"
Bach – Jesu, Joy of Man's Desiring, Cantata 147
Bach – Kyrie Duet Intro
Bach – Kyrie
Bach – Laudamas Te Intro
Bach – Loure
Bach – Lute Prelude
Bach – Menuet
Bach – Minuetto
Bach – Praeludium #1 in C, BWV 846
Bach – Praeludium #17 in Ab, BWV 862

Bach – Qui Sedes Aria Intro
Bach – Two Part Invention #1, Allegro
Bach – Two Part Invention #2 in Cm
Bach – Two Part Invention #3 in D
Bach – Two Part Invention #4 in Dm
Bach – Two Part Invention #5 in Eb
Bach – Two Part Invention #8 in F
Handel – Air in F, Watermusic
Handel – Bourree in Dm "Fireworks Music"
Handel – La Paix , "Fireworks Music"
Handel – La Rejouissance, "Fireworks Music"
Handel – Largo
Handel – Largo in G from the Opera "Xerxes"
Handel – Menuet 1, "Fireworks Music"
Handel – Menuet 2, "Fireworks Music"
Handel – Menuet in F , "'Watermusic"
Handel – Menuetto in F, "Watermusic"
Handel – Overture, "Fireworks Music"
Handel – Overture in F, "Watermusic"
Vivaldi – The Four Seasons "Autumn" 1st mvt.
Vivaldi – The Four Seasons, "Autumn" 2nd mvt.
Vivaldi – The Four Seasons, "Autumn" 3rd mvt
Vivaldi – The Four Seasons, "Spring" 1st mvt
Vivaldi – The Four Seasons, "Spring" 2nd mvt.
Vivaldi – The Four Seasons, "Spring" 3rd mvt.
Vivaldi – The Four Seasons, "Summer" 1st mvt
Vivaldi – The Four Seasons, "Summer" 3rd mvt
Vivaldi – The Four Seasons, "Winter" 1st mvt.
Vivaldi – The Four Seasons, "Winter" 2nd mvt.
Vivaldi – The Four Seasons, "Winter" 3rd mvt

Classical

Beethoven – An Einen Saugling
Beethoven – Aus Goethe's Faust
Beethoven – Piano Sonata No. 6 in F, Op. 10, No. 2, 1st mvt.
Beethoven – Sonata in G, Op. 49, No. 2
Beethoven – Sonata in Gm, Op. 49, No. 1
Beethoven – Symphony 1 in C
Beethoven – Symphony 1, 2nd mvt. in F
Beethoven – Symphony 2 in D, Op.36, 1st mvt
Beethoven – Symphony in D, Op. 36, 4th mvt.
Carulli – Grand Etude

Carulli – Sonata in A
Gluck – Andante
Haydn – Allemande
Haydn – Sonata in F, 1st mvt.
Haydn – Sonata in F, Minuet
L. Mozart – Bourree
Mozart – Adagio in B♭, Piano sonata 1, 2nd mvt.
Mozart – Adagio in Fm, Piano Sonata 2, 2nd mvt.
Mozart – Adagio in G, Sonata 3 (K545), 2nd mvt.
Mozart – Allegro Finale in E♭, Symphony 39 (K543), 4th mvt.
Mozart – Andante in B♭, Sonata in F (K533), 2nd mvt.
Mozart – Andante in C, Sonata in G (K189h), 3rd mvt.
Mozart – Eine Kleine Nachtmusic (K525), 1st mvt.
Mozart – Eine Kleine Nachtmusic (K525), 3rd mvt.
Mozart – Eine Kleine Nachtmusic (K525), 4th mvt.
Mozart – Menuetto in E♭, Symphony 39 (K543), 3rd mvt.
Mozart – Among Those Who Love
Mozart – Bird Charmer
Mozart – Eine Kleine Nachtmusic (K525), 2nd mvt.
Mozart – Prague Symphony 38 (K504), 1st mvt.
Mozart – Rondo in C, Piano Sonata 3, 3rd mvt.
Mozart – Rondo in F, Piano Sonata 4, 3rd mvt.
Mozart – Sonata 1, 1st mvt.
Mozart – Sonata 15, 3rd mvt.
Mozart – Sonata 15, Adagio
Mozart – Sonata 16, 1st mvt.
Mozart – Sonata 16, Menuetto
Mozart – Sonata 2, 1st mvt.
Mozart – Sonata 3, 1st mvt.
Mozart – Sonata 4, 1st mvt.
Mozart – Sonata 5, 1st mvt.
Mozart – Sonata 8, Andante
Mozart – Sonata 9, 1st mvt.
Mozart – Sonata 9, Adagio
Mozart – Sonata 9, Rondeau
Mozart – Sonata 9, Tema
Mozart – Symphony 38 (K504), “Prague” 2nd mvt.
Paganini – Adagio in Bm, Op. 6, 2nd mvt.
Paganini – Concerto in D, Op. 6, 1st mvt.
Paganini – Rondo in D, Op.6, 3rd mvt.

Romanticism

Beethoven – Abshiedsgesang
Beethoven – An Die Hoffnung
Beethoven – Für Elise
Beethoven – Opferlied
Beethoven – Sonata, Op. 31, No. 3
Beethoven – Symphony 3 “Eroica - Marcia Funebre”
Beethoven – Symphony 3 in Eb “Eroica” Op. 55, 1st mvt.
Beethoven – Symphony 4, Op. 60
Beethoven – Symphony 5
Beethoven – Symphony 5, Op. 67, 1st mvt.
Beethoven – Symphony 6 in F, Op. 68, 1st mvt. Allegro
Beethoven – Symphony 6, 2nd mvt.
Beethoven – Symphony 7 in A
Beethoven – Symphony 7, 2nd mvt.
Beethoven – Symphony 8, Op. 93
Beethoven – Symphony 9, Op. 125, 3rd mvt. Adagio
Beethoven – Turkish March
Bellini – Norma March
Bohm – La Zingara
Bohm – The Rain
Brahms – Hungarian Dance 5
Brahms – Symphony 1 in Cm, Op. 68 1st mvt.
Brahms – Symphony 1, Op. 68, 2nd mvt. Andante
Brahms – Symphony 1, Op. 68, 3rd mvt.Allegretto
Brahms – Symphony 1,Op. 68, 4th mvt.
Brahms – Symphony 2 in D, Op. 73, 1st mvt.
Brahms – Symphony 2, Op. 73, 2nd mvt.
Brahms – Symphony 2, Op. 73, 3rd mvt.
Brahms – Symphony 3 in F, Op. 90 1st mvt. Allegro
Brahms – Symphony 3, Op. 90 2nd mvt. Andante
Brahms – Symphony 3, Op. 90, 3rd mvt.
Brahms – Waltz 1 in B, Op. 39
Brahms – Waltz 2 in E, Op. 39
Brahms – Waltz 3 in G#m, Op. 39
Brahms – Waltz 4 in Em, Op. 39
Brahms – Waltz 6 in C#, Op. 36
Chopin – Etude in Cm
Chopin – Etude, Op. 25, No. 9
Chopin – Fantaisie Impromptu in Db
Chopin – Fantaisie Impromtu 4
Chopin – Funeral March
Chopin – Mazurka In Am, Op. 67, No. 4

Chopin – Mazurka in Ab, Op. 24, No. 3
Chopin – Mazurka in Am, Op. 67, No. 2
Chopin – Mazurka in Am, Op. 7, No. 2
Chopin – Mazurka in B, Op. 63, No. 1
Chopin – Mazurka in Bb, Op. 7, No. 1
Chopin – Mazurka in Bm, Op. 33, No. 4
Chopin – Mazurka in C, Op. 33, No. 3
Chopin – Mazurka in C, Op. 67, No. 3
Chopin – Mazurka in D, Op. 33, No. 2
Chopin – Mazurka in G#m, Op. 33, No. 1
Chopin – Mazurka in Gm, Op. 67, No. 2
Chopin – Nocturne in B, Op. 32, No. 1
Chopin – Nocturne in Eb, Op. 9, No. 2
Chopin – Nocturne in F# , Op. 15, No. 2
Chopin – Nocturne in Fm Op. 55, No. 1
Chopin – Nocturne in Gm, Op. 15, No. 3
Chopin – Nocturne, Op. 27, No. 2
Chopin – Nocturne, Op. 37, No. 1
Chopin – Op. 10, No. 2
Chopin – Op. 25, No. 8
Chopin – Polonaise In Ab, Op. 53
Chopin – Polonaise in A, Op. 40, No. 1
Chopin – Polonaise, Op. 26, No. 1
Chopin – Prelude in A, Op. 28, No. 7
Chopin – Prelude in Bm, Op. 28, No. 6
Chopin – Prelude in Cm, Op. 28, No. 20
Chopin – Prelude in Db, Op. 28, No. 15
Chopin – Prelude in Em, Op. 28, No. 4
Chopin – Valse Brilliante, Op. 34, No. 2
Chopin – Valse Posthumous
Chopin – Valse in Ab, Op. 42
Chopin – Valse in Cm, Op. 64, No. 2
Chopin – Waltz in Ab, Op. 69, No. 1
Chopin – Waltz in Am, Op. 34, No. 2
Chopin – Waltz in Bb, Op. 39, No. 8
Chopin – Waltz in Bm, Op. 69, No. 2
Delibes – Coppelia Valse
Delibes – Pas Des Fleurs
Dvorak – Humoreske, Op.101, No.7
Dvorak – Largo from New World Symphony
Dvorak – Slavonic Dance 2 in Em
Dvorak – Slavonic Dance 3 in Ab
Dvorak – Slavonic Dance 4 in F
Dvorak – Slavonic Dance 6 in D

Gounod – Faust Ballet
Grieg – Album Leaf
Grieg – Dance Of Anitra, Op.46, No.3
Grieg – Norwegian Dance
Liszt – Consolation
Mendelssohn – Spring Song
Mendelssohn – Symphony 1 in Cm, Op.11 2nd mvt.
Mendelssohn – Symphony 1 in Cm, Op.11 1st mvt.
Mendelssohn – Symphony 4, 3rd mvt.
Mendelssohn – Symphony 4, Andante
Mendelssohn – Venetian Boat Song 1, Op.19, 6
Moszkowski – Melodie, Op. 18, No 1
Moszkowski – Serenata, Op. 15
Offenbach – Barcarolle
Ponchielli – Dance of the Hours
Schubert – L'abeille
Schubert – Serenade
Schubert – Symphony 3 in D (D200) 2nd mvt.
Schubert – Symphony 3 in D, 3rd mvt.
Schubert – Symphony 4 "Tragic" in Cm, D417
Schubert – Symphony 6 in C (D589) 2nd mvt.
Schubert – Symphony 6 in C (D589) 4th mvt.
Schubert – Valse Sentimentale 1 in D, D.779:12
Schubert – Valse Sentimentale 2 in A, D.779:13
Schumann – Traumerei, Op.15, No.7
Strauss – A Thousand and One Nights Waltz, Op.346
Strauss – Andante, Op. 8
Strauss – Cagliostro Waltz 1
Strauss – Emperor Waltz, Op.437
Strauss – Kiss Waltz
Strauss – Light Character Polka
Strauss – The Blue Danube
Strauss – "Artist's Life" Waltz
Strauss – "Danube Mermaid" Waltz, Op. 427
Strauss – "Enjoy Life" Waltz
Strauss – "Viennese Blood" Waltz
Strauss – "Voices of Spring" Waltz
Strauss – "You And You" Waltz
Suppe – Light Calvary Overture
Tschaikowsky – Chant Sans Paroles
Wagner – Flying Dutchman
Wagner – Lohengrin Swan Song
Wagner – Tannhauser March

Pre-bop

A sunday kind of love
Accentuate the positive
Accustomed to her face
After you've gone
Ain't misbehavin'
Alone together
All of me
All of you
All the things you are
Angel eyes
Any time
Anything goes
April in paris
Autumn in new york
Autumn leaves
Acid change
Back home again in Indiana
Beautiful love
Beginning to see the light
Bewitched
Billy boy
Blue moon
Blue room
Blue skies
Body and soul
But beautiful
But not for me
Can't we be friends
Candyman
Come sunday
Confessin' (that i love you)
Cottontail
Cry me a river
Cute
Charlston
Chattenooga choo
Cherokee
Christmas song
Dancing on the ceiling
Darn that dream
Dear old stockholm
Dearly beloved

Deed i do
Deep purple
Django
Do nothin' 'til you hear from me
Don't blame me
Don't get around much anymore
Dream
Easy living
Embraceable you
Everything i have is yours
Exactly like you
Fascinatin' rhythm
Fine romance
Fly me to the moon
For heaven's sake
Frosty the snowman
Gettin' sentimental over you
Ghost of a chance
Girl talk
God bless the child
Green dolphin street
Hello dolly
Honeysuckle rose
How about you?
How high the moon
How long has this been goin' on
I can't give you anything but love
I cover the waterfront
I cried for you
I don't know why
I hear a rhapsody
I left my heart in San Francisco
I let a song (go out of my heart)
I remember you
I should care
I won't dance
I'll be around
I'll be seeing you
I'll get by
I'll never smile again
I'm a fool to want you
I'm all smiles
I'm in the mood for love
I've got a crush on you

In a mellowtone
In a sentimental mood
Isn't it romantic
It could happen to you
It don't mean a thing
It had to be you
It's only a paper moon
It's you or no one
Just friends
Just in time
Just the way you look tonight
Lady is a tramp
Laura
Like someone in love
Lover man
Lulu's back in town
Make someone happy
Mame
Manhattan
Margie
Mean to me
Memories of you
Misty
Mood indigo
Moonglow
Moonlight in vermont
More than you know
My blue heaven
My foolish heart
My heart belongs to daddy
My old flame
My romance
My shining hour
My ship
Nature boy
New York New York
No moon at all
Oh you beutiful doll
On a clear day
On a clear day
On a slow boat to China
On the street where you live
On the sunny side of the street
Once in a while

Opus one
Pennies from heaven
Perdido
Pick yourself up
Polka dots and moonbeams
Portrait of jennie
Prelude to a kiss
Put on a happy face
Quiet now
Red top
Rosetta
Route 66
Satin doll
Sentimental journey
September song
Shiny stockings
Solitude
Some other time
Someday you'll be sorry
Sophisticated lady
Speak low
Spring is here
Star crossed lovers
Star eyes
Stardust
Stompin' at the savoy
Summertime
Swingin' shepherd blues
Tangerine
Tenderly
The nearness of you
The song is you
There is no greater love
There will never be another you
There's a small hotel
These foolish things
Till there was you
Time after time
Touch of your lips
We'll be together again
What am i here for
When sunny gets blue
When you're smiling
Yesterdays

You are too beautiful
You don't know what love is
You took advantage of me
You've changed

Bop

'round midnight
A night in Tunisia
Afro blue
Afternoon in paris
All blues
Anthropology
Ask me now
Au privave
Ba-lue Bolivar ba-lues are
Bags' groove
Beautiful friendship
Bemsha swing
Bessie's blues
Big nick
Black narcissus
Blue comedy
Blue monk
Blue trane
Blues for alice
Bluesette
Boplicity
Bright Mississippi
Brilliant corners
Bye
Confirmation
Daahoud
Dexterity
Dolphin dance
Donna lee
E.s.p.
Epistrophy
Equinox
Eronel
Evidence
Excercise 3 (Missouri uncompromized)
Fall

Fee fi fo fum
Fifty second street theme
Five-0-two blues
Footprints
Four
Four on six
Gloria's step
Goodbye pork pie hat
Grand central
Groovin' high
Hackensack
Half nelson
House of jade
I mean you
I remember clifford
If you could see me now
Impressions
In walked bud
In your own sweet way
Inner urge
Invitation
Isotope
Israel
Jackie
Jinriksha
Jordu
Joy spring
Lady bird
Let's cool one
Little rottie tootie
Missouri uncompromised
Miyako
Moment's notice
Monk's mood
Moose the mooche
Mysterioso
Naima
Nica's dream
Off minor
Oleo
One by one
Ornithology
Pannonica
Peace

Peri's scope
Played twice
Reflections
Rhythm
Ruby my dear
Scrapple from the apple
Solar
Speak no evil
Stolen moments
Straight no chaser
Think of one
Unit seven
Well you needn't

Bossanova

A Felicidade
A Ilha
A Ra
African Flower
Agua de beber
Amazonas
Ana Maria
Andanca
Aquele Abraco
Batida diferente
Blue Bossa
Bridges (Travessia)
Ceora
Contrato de Separacao
Corcovado
Cheganca
Chiclete com banana
De pois do amor o vazio
Demais
Desafinado
Doce Vampiro
Dois pra la dois pra ca
Dores de Amores
Ela e carioca
Emocoes
Eu sei que vou te amar
Falando de amor

Fato Consumado
Feitico da Vila
Flor de liz
Fotografia
Futuros Amantes
Garota de Ipanema
Gentle Rain
Here's that rainy day
How insensitive
If you never come to me
Ipanema (the girl from)
Look to the sky
Lucky southern
Lugar comum
Manha de Carnaval
Marina
Mas Que Nada
Meditation
Meu Bem Querer
Misterios
Moda de Sangue
Nuvem Negra
O morro nao tem vez
Once I loved
One note samba
Pecado Original
Pensativa
Pretty World (Samarina)
Recado
Saudade fez um samba
So louco
Tanto que aprendi de amor
Tarde em Itapoa
The Dolphin
Tintim por tintim
Tipo Zero
Trem das Cores
Valsinha
Wave

Celtic

Baby Brat
Banish Misfortune
Behind the Haystack
Billy In The Lowground
Black Reel 5
Capers
Coleraine's jig
Colonel Rodney
Congress reel
Cottage in the Grove
Cherish the ladies
Dawn
Dingle Regatta
Draught of Ale
Drowsy Maggie
Eleanor Kane
Father O'Flynn
Flowers of Red Hill
Frost is All Over
Gallagher's Frolics
Gander in the Pratie Hole
Ger the Rigger
Gravel Walk
Green Fields of Rossbeigh
Hag with the Money
Hare in the Corn
Harper's Frolic
Hayden Fancy
Jackie Coleman's reel
Julia Delaney
Kathleen Hehir's
Kevin Burke Polka 1
Kevin Burke Polka 2
Kevin Burke Polka 3
Kevin Burke Polka Set
Kilavel Jig
Killarney Boys of Pleasure
Kitty come down to Limerick
Loch Giel
Martin Wynne's 2
Mason's Apron
Merrily Kiss the Quaker

Mike's Fancy
Miss McLeod's Reel
Mountain Road
Mulvihill's
Music In the Glen
My Love is in America
Nine Points of Roguery
Old Hag you have killed me.
Old John's Jig
Paddy Carthy
Pretty Pegg
Queen of the Fair
Rakes of Kildare
Rolling on the Ryegrass
Salmon Tails Up the Water
Sean Ryan's Reel
Sean Sa Ceo
Sligo Maid
Spey in Spate
Sporting Paddy
Strayaway Child
Tar Road to Sligo
The Battering Ram
The Boys of Ballysodare
The Connaughtman's Rambles
The Cup of Tea
The Drunken Landlady
The Earl's Chair
The Girl that Broke my Heart
The Hag's Purse
The Jig of Slurs
The Lark in the Morning
The Lark on the Strand
The Longford Tinker
The Maid of Mount Cisko
The Musical Priest
The Oak Tree
The Orphan
The Otter's Den
The Pigeon on the Gate
The Pinch of Snuff
The Rambling Pitchfork
The Rose in the Heather
The Scholar

The Shores of Loch Gowna
The Swallow's Tail
Tom Ward's Downfall
Tongs by the Fire
Toss the Feathers
Trip To Windsor
Tripping Up the Stairs
Whelan's Jig
Whisky Before Breakfast
Willie Coleman's Jig
Wind that Shakes the Barley
Winter Ducks
Wise Maid

Blues

A New Approach
After Midnight
All The Time
Another Day Older
At Hop
Bad Boy
Baptist Blues
Bet Your Soul
Big Jive
Blowin' the Blues
Blue Gene
Blue Rhumba
Boys
City Trouble
Cosmic Smile
Cryin' Today
Cheap Thrills
Dizzy Miss Lizzie
Don't Say It
Empty Street
Esoteric Jubilee
Eyeballin'
Flip Flop Fly
Fake ID
Funk Favors
Gimme Five
Goin' Downtown

Going To See My Lord
Gospel Truth
Got That Feelin'
Half Baked Blues
Hard
He Is Faithful
He Talks To Me
He Washed My Sins Away
High & Dry
Holy Is His Name
Hope Eternal
I'm Sure You Know
Jack's Groove
Just Asking for the Blues
La Valse Bleue
Leavin' Town
Lightning Rod
Little Freda
Long Tall Sally
Low Profile
Meaning of It All
Misty Blues
Mommas Prayer
Money
Mucho Jammin'
My Heart Has Joy
Night Train
On My Mind
Outa' Town Blues
Pale Blues
Pass the Slaw
Please Blues go on away
Positive Repartee
Praise His Name
Rattlesnake Blues
Right By My Side
Rock and Roll Music
Skanky Times
Sly Luck
So Far From Home
Supercilious Blues
Sweet Jam
Talkin Bout Jesus
That's What It Is

The Circumventor
The Maryland farmer
Third Caper
To Be True
Troublesome Blues
Turn It Around
Twenty
Twice Thrice Blues
Twisted River
Walkin' Dude
Way Up On High
We Come to Praise Him
Your mama don't dance

Pop

A Little Bit More
Adoro
Alfie
Alone Again Naturally
All that she wants
Anna (Go To Him)
Arthur's Theme
Arrow Through Me
August Outcome
Autumn Morning
B-Holly
Baby I'm A Want You
Blue Bayou
Bohemian Rhapsody
Bop Illusion
Both Sides Now
Breathe
Bridge Over Troubled Water
California Dreamin'
Candle On The Water
Carefree Highway
Circus
Come In From The Rain
Do I Love You Because You're Beautiful
Do You Feel Like We Do
Don't Stop Thinking About Tomorrow
Dream Dream Dream

Dust In The Wind
Evergreen
Ferryman
Get Ready
Here comes the Sun
Homburg
Honey
House Of The Rising Sun
How do you sleep?
I Am Your Captain
I Just Want To Stop
I'll Be Watching You
If You Leave Me Now
If
Insane Load
Isn't She Lovely
Just you and I
Knock Three Times Medley
Lonely People
Make the World Go Away
Midnight Blue
Monday Mirror
More Today Than Yesterday
Mother's Little Helper
Never Mind
New World
November Groom
Out of Time
Paloma Blanca
Peace
Povo
Puppet on a string
Raw Thrash
Red Sails In The Sunset
Ride On Baby
Run For The Roses
Sad Breeze
Sittin on a Fence
Smokey Mountain Rain
Somewhere Out There
Sound of Silence
Streets of London
Summertime Blues
Sweet Transvestite

Take It or Leave It
Taste Of Honey
Tell Me (You're Coming Back)
The Best Of My Love
The Fool on the Hill
The Letter
The One That You Love
The Party's Over
The Rose
The Water is Wide
This Guy's In Love With You
Time Is On My Side
Traces
True Love Ways
Tuesday Afternoon
Twist and Shout
Vem Vet (Who Know)
Walk On By
What Have They Done to the Rain
Who's Sorry Now?
Why Did You Not Come?
Words of Love
Yes I'm Ready
Yesterday
You Make Me Feel Like Dancin'
You Really Got A Hold On Me
You are Beautiful
You've Got A Friend
Your song

B

Chord vocabulary

Table B.1 shows the chord vocabulary used in the encoding of harmony (see Section 2.3). Left column shows chords as found in the music files, and the other columns show the encodings used in the four reduced sets: *full chords*, *triad chords*, *4-note chords*, and *major-minor chords*. The feature sets that encode degrees instead of chord names use the same symbols, but replacing chord roots by degrees according to the tonality.

Raw chord	Full chord	Triads	4-note chord	major-minor chord
C	C	C	C	C
C2	C	C	C	C
C5	C	C	C	C
C6	C	C	C	C
C69	C	C	C	C
Cmaj	C	C	C	C
C4	C4	Csus	C	C
Csus	C4	Csus	C	C
C+	Caug	Caug	C	C
Caug	Caug	Caug	C	C
C7	C7	C	C7	C
C7+	C7+	Caug	C7	C
C7alt	C7alt	C	C7	C
C7♯5	C7alt	Caug	C7	C
C7♯5♯9	C7alt	Caug	C7	C
C7♯5♭9	C7alt	Caug	C7	C
C7♯9	C7♯9	C	C7	C
C7♯9♭13	C7alt	Caug	C7	C
C7♯11	C7♯11	C	C7	C
C7aug	C7alt	Caug	C7	C
C7♭9♭13	C7alt	Caug	C7	C
C7♭5	C7alt	C(b5)	C7	C
C7♭5♯9	C7♭5♯9	C(b5)	C7	C
C7♭9	C7alt	C	C7	C
C7♭9♯11	C7♭9♯11	C	C7	C
C7♭13	C7alt	C	C7	C

C7♭5♭9	C7alt	C(b5)	C7	C
C7sus	C11	Csus	C7	C
C7sus9	C11	Csus	C7	C
C7sus♭9	C7sus♭9	Csus	C7	C
Cmaj7	Cmaj7	C	Cmaj7	C
Cmaj7♯5	Cmaj7♯5	Caug	Cmaj7	C
C9	C7	C	C7	C
C9♯5	Cwhole	Caug	C7	C
C9♭5	Cwhole	C(b5)	C7	C
C9♯11	C9♯11	C	C7	C
Cmaj9♯11	Cmaj9♯11	C	Cmaj7	C
Cmaj9	Cmaj7	C	Cmaj7	C
C11	C11	Csus	C7	C
C13	C7	C	C7	C
Cmaj13	Cmaj7	C	Cmaj7	C
C13alt	C13alt	C	C7	C
C13♯5	C13alt	Caug	C7	C
C13♭5	C13alt	C(b5)	C7	C
C13♯9	C13alt	C	C7	C
C13♭9	C13alt	C	C7	C
C13♯11	C13♯11	C	C7	C
C13sus	C13sus	Csus	C7	C
Cwhole	Cwhole	Caug	C7	C
Cm	Cm	Cm	Cm	Cm
CmMaj7	Cm	Cm	Cm	Cm
Cm♯5	Cm♯5	Cm(+5)	Cm	Cm
Cm7	Cm7	Cm	Cm7	Cm
Cm9	Cm7	Cm	Cm7	Cm
Cm11	Cm7	Cm	Cm7	Cm
Cm6	Cm6	Cm	Cm	Cm
Cm69	Cm6	Cm	Cm	Cm
Cm7♭5	Cm7♭5	Cdim	Cm7	Cm
Cdim	Cdim	Cdim	Cm	Cm

Table B.1: Chord vocabulary found in the training sets encoded using different reductions of the vocabulary.

C Tables

The following tables correspond to the results of the experiments on harmonic rhythm, presented in Chapter 5, Section 5.3.

Root	Extensions	naïve Bayes		n-grams		
		B	M	$n=2$	$n=3$	$n=4$
Degrees	Full	86 ± 4	84 ± 3	87 ± 3	87 ± 3	87 ± 3
	4-note	86 ± 4	83 ± 3	87 ± 2	87 ± 3	88 ± 3
	Triads	78 ± 4	79 ± 3	82 ± 3	84 ± 4	84 ± 2
	Major-minor	77 ± 4	75 ± 5	84 ± 3	84 ± 4	85 ± 3
Chord names	Full	86 ± 5	83 ± 3	86 ± 4	86 ± 4	86 ± 4
	4-note	85 ± 5	81 ± 3	87 ± 3	87 ± 3	87 ± 3
	Triads	78 ± 6	77 ± 4	84 ± 3	84 ± 3	84 ± 2
	Major-minor	77 ± 4	75 ± 5	85 ± 3	85 ± 3	86 ± 2

(a) coupled

Root	Extensions	naïve Bayes		n-grams		
		B	M	$n=2$	$n=3$	$n=4$
Degrees	Full	87 ± 5	84 ± 3	83 ± 3	87 ± 3	86 ± 4
	4-note	86 ± 3	82 ± 3	84 ± 3	86 ± 2	87 ± 2
	Triads	75 ± 5	76 ± 4	74 ± 3	82 ± 4	84 ± 2
	Major-minor	73 ± 5	74 ± 5	76 ± 5	83 ± 3	84 ± 4
Chord names	Full	87 ± 4	82 ± 4	81 ± 3	87 ± 3	88 ± 3
	4-note	85 ± 4	80 ± 4	81 ± 4	86 ± 3	87 ± 3
	Triads	76 ± 5	75 ± 5	75 ± 4	82 ± 3	84 ± 4
	Major-minor	73 ± 4	74 ± 4	76 ± 5	83 ± 3	85 ± 3

(b) decoupled

Table C.1: Average classification rates obtained for the 3-classes problem using chord progressions and harmonic rhythm.

Root	Extensions	naïve Bayes		n-grams		
		B	M	$n=2$	$n=3$	$n=4$
Degrees	Full	63 ± 5	61 ± 3	42 ± 7	42 ± 7	43 ± 6
	4-note	61 ± 6	59 ± 5	50 ± 6	51 ± 8	50 ± 8
	Triads	55 ± 7	54 ± 6	51 ± 5	54 ± 6	53 ± 4
	Major-minor	54 ± 7	51 ± 5	54 ± 8	59 ± 4	57 ± 5
Chord names	Full	65 ± 4	62 ± 2	41 ± 6	42 ± 6	41 ± 6
	4-note	63 ± 6	61 ± 2	49 ± 5	51 ± 7	49 ± 6
	Triads	56 ± 6	55 ± 3	50 ± 9	52 ± 6	51 ± 6
	Major-minor	54 ± 9	52 ± 4	54 ± 8	57 ± 5	56 ± 6

(a) coupled

Root	Extensions	naïve Bayes		n-grams		
		B	M	$n=2$	$n=3$	$n=4$
Degrees	Full	63 ± 6	62 ± 5	29 ± 8	45 ± 5	45 ± 7
	4-note	61 ± 6	58 ± 6	35 ± 10	53 ± 5	52 ± 5
	Triads	53 ± 8	54 ± 6	37 ± 10	52 ± 6	53 ± 4
	Major-minor	50 ± 7	50 ± 7	44 ± 5	57 ± 5	57 ± 5
Chord names	Full	63 ± 5	63 ± 4	30 ± 9	44 ± 6	44 ± 7
	4-note	62 ± 6	62 ± 3	37 ± 12	54 ± 5	54 ± 4
	Triads	53 ± 5	55 ± 5	40 ± 7	54 ± 7	54 ± 8
	Major-minor	49 ± 8	52 ± 4	45 ± 6	57 ± 9	58 ± 7

(b) decoupled

Table C.2: Average classification rates obtained for the 9-classes problem using chord progressions and harmonic rhythm.

Bibliography

Aucouturier, J.-J. and Pachet, F. (2003). Representing musical genre: A state of the art. *Journal of New Music Research*, 32(1):83–93. 1.5.1

Aucouturier, J.-J. and Pachet, F. (2004). Improving timbre similarity: How high is the sky? *Journal of Negative Results in Speech and Audio Sciences*, 1(1). 1.4.1

Backer, E. and van Kranenburg, P. (2005). On musical stylometry: a pattern recognition approach. *Pattern Recognition Letters*, 26(3):299–309. (document), 1.5.1, 1.5.2, 2, 2.1, 6.1

Bello, J. P. and Pickens, J. (2005). A robust mid-level representation for harmonic content in music signals. In *Proceedings of the 6th International Conference on Music Information Retrieval, ISMIR 2005*, pages 304–311. 1.3, 5.4

Buzzanca, G. (2002). A supervised learning approach to musical style recognition. In *Proceedings of the Second International Conference on Music and Artificial Intelligence, ICMAI*. 1.5.1, 1.5.2

Camacho, A. (2008). Detection of pitched/unpitched sound using pitch strength clustering. In *Proceedings of the 9th International Conference on Music Information Retrieval, ISMIR 2008*, pages 533–537, Philadelphia, Pennsylvania, USA. 1.3

Camastra, F. and Vinciarelli, A. (2008). *Machine Learning for Audio, Image and Video Analysis*. Springer, 1st. edition. 3.2

Carreira-Perpiñán, M. A. and Renals, S. (2000). Practical identifiability of finite mixtures of multivariate Bernoulli distributions. *Neural Computation*, 12(1):141–152. 3.1, 3.1.2

Cataltepe, Z., Yaslan, Y., and Sonmez, A. (2007). Music genre classification using midi and audio features. *EURASIP Journal on Advances in Signal Processing*, 2007. 1.5.4, 4.3

Chai, W. and Vercoe, B. (2001). Folk music classification using hidden markov models. In *Proceedings of the International Conference on Artificial Intelligence*, Las Vegas, USA. (document), 1.5.1, 1.5.2

Clarkson, P. R. and Rosenfeld, R. (1997). Statistical language modeling using the CMU-Cambridge Toolkit. In *Proceedings of ESCA Eurospeech*. 3.2.1

Conklin, D. and Witten, I. (1995). Multiple viewpoint systems for music prediction. *Journal of New Music Research*, 24(1):51–73. 3.3

Cope, D. (1996). *Experiments in musical intelligence*. A-R Editions, Inc. 1.3, 1.4.1

Cover, T. M. and Thomas, J. A. (1991). *Elements of Information Theory*. John Wiley. 3.1.4

Cruz-Alcázar, P. P. (2004). *Técnicas de Reconocimiento de Formas para el Modelado de Estilos Musicales*. PhD thesis, Universidad Politécnica de Valencia. (document), 1.6, 2.2, 3.2.2, 4.4

Cruz-Alcázar, P. P. and Vidal, E. (2008). Two grammatical inference applications in music processing. *Applied Artificial Intelligence*, 22(1 & 2):53–76. (document), 1.3, 1.4.2, 1.5.2, 1.6, 2, 2.1, 2.1.1, 2.1.2, 4.2, 4.2.3, 4.4, 7.1

Dannenberg, R. B., Birmingham, W. P., Pardo, B., Hu, N., Meek, C., and Tzanetakis, G. (2007). A comparative evaluation of search techniques for query-by-humming using the musart testbed. *Journal of the American Society for Information Science and Technology*, 58(5):687–701. 1.3

Dannenberg, R. B., Thom, B., and Watson, D. (1997). A machine learning approach to musical style recognition. In *Proceedings of the International Computer Music Conference*, pages 344–347. International Computer Music Association. (document), 1.5.1, 1.5.2

de la Higuera, C., Piat, F., and Tantini, F. (2005). Learning stochastic finite automata for musical style recognition. In *Proceedings of the 10th International Conference on Implementation and Application of Automata, CIAA*, volume 3845 of *Lecture Notes in Computer Science*, pages 345–346. Springer. (document), 1.5.2, 4.4

Dempster, A. P., Laird, N. M., and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39:1–38. 3.1.2

Dietterich, T. G. (2000). Ensemble methods in machine learning. *Lecture Notes in Computer Science*, 1857:1–15. 3.3

Dixon, S. (2007). Evaluation of the audio beat tracking system beatroot. *Journal of New Music Research*, 36(1):39–50. 5.4.2

Domingos, P. and Pazzani, M. (1997). Beyond independence: conditions for the optimality of simple bayesian classifier. *Machine Learning*, 29:103–130. 3.1

Dopler, M., Schedl, M., Pohle, T., and Knees, P. (2008). Accessing music collections via representative cluster prototypes in a hierarchical organization scheme. In *Proceedings of the 9th International Conference on*

Music Information Retrieval, ISMIR 2008, pages 179–184, Philadelphia, Pennsylvania, USA. 1.3

Doraisamy, S. and Rüger, S. (2003). Robust polyphonic music retrieval with n-grams. *Journal of Intelligent Information Systems*, 21(1):53–70. (document), 1.4.2, 2.2, 2.2.1, 4.1

Downie, J. S. (1999). *Evaluating a Simple Approach to Music Information Retrieval: Conceiving Melodic n-grams as Text*. PhD thesis, University of Western Ontario. (document), 1.4.2, 2.2.1

Duda, R. O., Hart, P. E., and Stork, D. G. (2000). *Pattern Classification, Second Edition*. Wiley-Interscience. (document), 1.2

Espí, D., Ponce de León, P. J., Pérez-Sancho, C., Rizo, D., Iñesta, J. M., Moreno-Seco, F., and Pertusa, A. (2007). A cooperative approach to style-oriented music composition. In *Proceedings of the International Workshop on Artificial Intelligence and Music, MUSIC-AI*, pages 25–36, Hyderabad, India. 7.2

Fabbri, F. (1999). Browsing music spaces: categories and the musical mind. In *Proceedings of the IASPM Conference*. (document), 1.5.1

Feller, W. (1971). *An introduction to probability theory and its applications*. John Wiley and Sons, 2nd. edition. 4.2.1

Feng, L., Nielsen, A. B., and Hansen, L. K. (2008). Vocal segment classification in popular music. In *Proceedings of the 9th International Conference on Music Information Retrieval, ISMIR 2008*, pages 121–126, Philadelphia, Pennsylvania, USA. 1.3

Futrelle, J. and Downie, J. S. (2003). Interdisciplinary communities and research issues in music information retrieval: ISMIR 2000–2002. *Journal of New Music Research*, 32(2):121–131. 1.1

Gedik, A. C. and Alpkocak, A. (2006). Instrument independent musical genre classification using random 3000 ms segment. In *Proceedings of the 14th Turkish Symposium on Artificial Intelligence and Neural Networks, TAINN*. 1.5.1, 1.5.2

Gómez, E. (2006). *Tonal Description of Music Audio Signals*. PhD thesis, MTG, Universitat Pompeu Fabra, Barcelona, Spain. (document), 1.3, 2.3.1, 5.4.1

Grachten, M., Arcos, J. L., and de Mántaras, R. L. (2005). Melody retrieval using the Implication/Realization model. In *Proceedings of the 6th International Conference on Music Information Retrieval, ISMIR 2005*. 2.2.1

Harte, C., Sandler, M., Abdallah, S., and Gómez, E. (2005). Symbolic representation of musical chords: a proposed syntax for text annotations. In *Proceedings of the 6th International Conference on Music Information Retrieval, ISMIR 2005*, pages 66–71. 1.4.3

Herrera, E. (1998). *Teoría musical y armonía moderna*, volume II. Antoni Bosch Editor (in spanish). 1.6

Howell, D. C., editor (1997). *Statistical Methods for Psychology*. Duxbury Press, London, 4th edition. 4.2.1

Hu, X., Downie, S. J., Laurier, C., Bay, M., and Ehmann, A. F. (2008). The 2007 mirex audio mood classification task: Lessons learned. In *Proceedings of the 9th International Conference on Music Information Retrieval, ISMIR 2008*, pages 462–467, Philadelphia, Pennsylvania, USA. (document), 1.3, 1.5.1

Illescas, P. R., Rizo, D., and Iñesta, J. M. (2008). Learning to analyse tonal music. In *Proceedings of the International Workshop on Machine Learning and Music, MML 2008*, pages 25–26, Helsinki, Finland. 1.3

Iñesta, J. M., Ponce de Len, P. J., and Heredia-Agoiz, J. L. (2008). A ground-truth experiment on melody genre recognition in absence of timbre. In *Proceedings of the 10th International Conference on Music Perception and Cognition (ICMPC10)*, pages 758–761, Sapporo, Japan. 1.5.2

Jain, A. K., Duin, R. P. W., and Mao, J. (2000). Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1). 3, 3.1.2

Juan, A. and Vidal, E. (2002). On the use of Bernoulli mixture models for text classification. *Pattern Recognition*, 35(12):2705–2710. 3, 3.1, 3.1.2

Karydis, I., Nanopoulos, A., and Manolopoulos, Y. (2006). Symbolic musical genre classification based on repeating patterns. In *Proceedings of the 1st ACM workshop on Audio and music computing multimedia*, pages 53–58, Santa Barbara, California, USA. 1.5.1, 1.5.2

Klapuri, A. (2005). A perceptually motivated multiple-f0 estimation method. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 291–294. 4.3

Koppel, M., Schler, J., and Argamon, S. (2008). Computational methods in authorship attribution. *Journal of the American Society for Information Science and Technology*, 60(1):9–26. 6

Lam, W., Ruiz, M., and Srinivasan, P. (1999). Automatic text categorization and its application to text retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 11:865–879. 3

Lee, K. (2007). A system for automatic chord transcription using genre-specific hidden markov models. In *Proceedings of the International Workshop on Adaptive Multimedia Retrieval*, Paris, France. 1.5.3

Lee, K. and Slaney, M. (2006). Automatic chord recognition from audio using a supervised HMM trained with audio-from-symbolic data. In *AMCMM '06: Proceedings of the 1st ACM workshop on Audio and music computing multimedia*, pages 11–20, New York, NY, USA. ACM. 1.3, 5.4

Lerdahl, F. and Jackendoff, R. (1983). *A Generative Theory of Tonal Music*. The MIT Press. 1.5.2

Lidy, T. and Rauber, A. (2005). Evaluation of feature extractors and psycho-acousic transformations for music genre classification. In *Proceedings of the 6th International Conference on Music Information Retrieval, ISMIR 2005*, pages 34–41. 1.4.1

Lidy, T., Rauber, A., Pertusa, A., and Iñesta, J. (2007). Improving genre classification by combination of audio and symbolic descriptors using a transcription system. In *Proceedings of the 8th International Conference on Music Information Retrieval, ISMIR 2007*, pages 61–66, Vienna, Austria. 1.5.4, 4.3

Lin, C.-R., Liu, N.-H., Wu, Y.-H., and Chen, A. L. P. (2004). Music classification using significant repeating patterns. In *Proceedings of Database Systems for Advanced Applications*, pages 506–518. 1.5.1, 1.5.2

Lippens, S., Martens, J. P., Leman, M., Baets, B., Meyer, H., and Tzanetakis, G. (2004). A comparison of human and automatic musical genre classification. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2004*, pages 233–236. 1.5.1

Little, D. and Pardo, B. (2008). Learning musical instruments from mixtures of audio with weak labels. In *Proceedings of the 9th International Conference on Music Information Retrieval, ISMIR 2008*, pages 127–132, Philadelphia, Pennsylvania, USA. 1.3

Little, D., Raffensperger, D., and Pardo, B. (2007). A query by humming system that learns from experience. In *Proceedings of the 8th International Conference on Music Information Retrieval, ISMIR 2007*, pages 335–338, Vienna, Austria. 1.3

Magno, T. and Sable, C. (2008). A comparison of signal-based music recommendation to genre labels, collaborative filtering, musicological analysis, human recommendation, and random baseline. In *Proceedings of the 9th International Conference on Music Information Retrieval, ISMIR 2008*, pages 161–166, Philadelphia, Pennsylvania, USA. 1.3

Manaris, B., Romero, J., Machado, P., Krehbiel, D., Hirzel, T., Pharr, W., and Davis, R. B. (2005). Zipf's Law, music classification and aesthetics. *Computer Music Journal*, 29(1):55–69. (document), 1.5.1, 1.5.2

Mandel, M. I. and Ellis, D. P. W. (2008). LabROSA's audio classification submissions. http://www.music-ir.org/mirex/2008/index.php. 6.1

Manning, C. D. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. The MIT Press. 2.2.1, 3.2.2, 3.2.2

Margulis, E. H. and Beatty, A. P. (2008). Musical style, psychoaesthetics, and prospects for entropy as an analytic tool. *Computer Music Journal*, 32(4):64–78. 1.5.1, 1.5.2

Marolt, M. (2004). A connectionist approach to automatic transcription of polyphonic piano music. *IEEE Transactions on Multimedia*, 6(3):439–449. 1.3

McCallum, A. and Nigam, K. (1998). A comparison of event models for naive bayes text classification. In *AAAI-98 Workshop on Learning for Text Categorization*, pages 41–48. 3.1, 3.1.3, 4.1, 4.1.4, 4.1.5

McKay, C. and Fujinaga, I. (2004). Automatic genre classification using large high-level musical feature sets. In *Proceedings of the 5th International Conference on Music Information Retrieval, ISMIR 2004*. 1.5.4

McKay, C. and Fujinaga, I. (2008). Combining features extracted from audio, symbolic and cultural sources. In *Proceedings of the 9th International Conference on Music Information Retrieval, ISMIR 2008*, pages 597–602, Philadelphia, Pennsylvania, USA. 1.5.4

McNab, R. J., Smith, L. A., Witten, I. H., Henderson, C., and Cunningham, S. J. (1996). Towards the digital music library: Tune retrieval from acoustic input. In *Digital Libraries'96, Proceedings of the ACM Digital Libraries conference, Bethesda, Maryland*, pages 11–18, New York. Association for Computing Machinery. 2.2.1

Metsis, V., Androutsopoulos, I., and Paliouras, G. (2006). Spam filtering with naive bayes – which naive bayes. In *Third Conference on Email and Anti-Spam (CEAS)*. 3

Meyer, L. B. (1989). *Style and Music: Theory, History, and Ideology*. University of Chicago Press, Chicago. 1.5.1

Moh, Y. and Buhmann, J. M. (2008). Kernel expansion for online preference tracking. In *Proceedings of the 9th International Conference on Music Information Retrieval, ISMIR 2008*, pages 167–172, Philadelphia, Pennsylvania, USA. 1.3

Moreno-Seco, F., Iñesta, J. M., de León, P. P., and Micó, L. (2006). Comparison of classifier fusion methods for classification in pattern recognition tasks. *Lecture Notes in Computer Science*, 4109:705–713. (document), 3.3, 5.5

Narmour, E. (1990). *The Analysis and Cognition of Basic Melodic Structures: The Implication-Realization Model*. University of Chicago Press, Chicago. 2.2.1

Novovičová, J. and Malík, A. (2002). Text document classification using finite mixtures. Technical Report 2063, Academy of Sciences of the Czech Republic, Institute of Information Theory and Automation. 3.1.2, 4.1, 4.1.4

Ogihara, M. and Li, T. (2008). N-gram chord profiles for composer style representation. In *Proceedings of the 9th International Conference on Music Information Retrieval, ISMIR 2008*, pages 671–676, Philadelphia, Pennsylvania, USA. 6.1

Paiement, J.-F. (2008). *Probabilistic Models for Music*. PhD thesis, École Polytechnique Fédérale de Lausanne. (document), 1.4.3, 1.5.3, 2.3.2, 7.2

Parncutt, R. and Drake, C. (2001). Psycology: Rythm. In Sadie, S., editor, *New Grove Dictionary of Music and Musicians, 20*, pages 535–538, 542–553. Macmillan Publishers Ltd. 1.4.2, 2.2.1

Pearce, M. T., Müllensiefen, D., and Wiggins, G. A. (2008). A comparison of statistical and rule-based models of melodic segmentation. In *Proceedings of the 9th International Conference on Music Information Retrieval, ISMIR 2008*, pages 89–94, Philadelphia, Pennsylvania, USA. 1.3

Pertusa, A. and Iñesta, J. (2004). Pattern recognition algorithms for polyphonic music transcription. In Fred, A., editor, *PRIS 2004: Pattern Recognition in Information Systems*, pages 80–89, Porto, Portugal. 1.3

Pertusa, A. and Iñesta, J. M. (2005). Polyphonic monotimbral music transcription using dynamic networks. *Pattern Recognition Letters. Special Issue on Artificial Neural Networks in Pattern Recognition*, 26(12):1809–1818. 4.3

Pertusa, A. and Iñesta, J. M. (2008). Multiple fundamental frequency estimation using gaussian smoothness. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2008*, pages 105–108, Las Vegas, USA. (document), 4.3, 4.3.1

Pertusa, A., Klapuri, A., and Iñesta, J. M. (2005). Recognition of note onsets in digital music using semitone bands. *Lecture Notes in Computer Science*, 3773:869–879. 4.3.1

Pickens, J. (2001). A survey of feature selection techniques for music information retrieval. Technical report, Center for Intelligent Information Retrieval. (document), 1.4.2

Pickens, J., Bello, J. P., Monti, G., Crawford, T., Dovey, M., Sandler, M., and Byrd, D. (2002). Polyphonic score retrieval using polyphonic audio queries: A harmonic modeling approach. In *Proceedings of the 3rd International Conference on Music Information Retrieval, ISMIR 2002*, pages 140–149. 1.3

Piston, W. (1987). *Harmony*. Norton, W. W. & Company, Inc., 5th. edition. 1.6, 2.6

Ponce de León, P. J. and Iñesta, J. M. (2007). A pattern recognition approach for music style identification using shallow statistical descriptors. *IEEE Transactions on Systems Man and Cybernetics C*, 37(2):248–257. (document), 1.5.1, 1.5.2, 1.5.4, 2, 2.1, 2.1.3, 4.4, 7.2

Ponce de León, P. J., Iñesta, J. M., and Pérez-Sancho, C. (2006). Classifier ensembles for genre recognition. In Pla, F., Radeva, P., and Vitrià, J., editors, *Pattern Recognition: Progress, Directions and Applications*, chapter 3, pages 41–53. Centre de Visió per Computador. Universitat Autònoma de Barcelona. (document), 4.4, 4.5

Ponte, J. M. and Croft, W. B. (1998). A language modeling approach to information retrieval. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–281, New York, NY, USA. ACM Press. 1.3

Ruppin, A. and Yeshurun, H. (2006). MIDI music genre classification by invariant features. In *Proceedings of the 7th International Conference on Music Information Retrieval, ISMIR 2006*, pages 397–399, Victoria, Canada. 1.5.1, 1.5.2

Scaringella, N., Zoia, G., and Mlynek, D. (2006). Automatic genre classification of music content: a survey. *Signal Processing Magazine, IEEE*, 23(2):133–141. 1.3, 1.5.1

Schönberg, A. (1967). *Fundamentals of Music Composition*. Faber & Faber. 1.4.2

Selfridge-Field, E., editor (1997). *Beyond MIDI: the handbook of musical codes*. MIT Press, Cambridge, MA, USA. 2, 2.1.5

Shan, M.-K., Kuo, F.-F., and Chen, M.-F. (2002). Music style mining and classification by melody. *Proceedings of the IEEE International Conference on Multimedia and Expo*, 1:97–100. 1.5.3

Sheh, A. and Ellis, D. P. W. (2003). Chord segmentation and recognition using em-trained hidden markov models. In *Proceedings of the 4th International Conference on Music Information Retrieval, ISMIR 2003*. 1.3, 5.4

Shmulevich, I., Yli-Jarja, O., Coyle, E., Povel, D.-J., and Lemström, K. (2001). Perceptual issues in music pattern recognition: Complexity or rhythm and key finding. *Computers and the Humanities*, 35. (document), 1.1

Stamatatos, E. and Widmer, G. (2005). Automatic identification of music performers with learning ensembles. *Artificial Intelligence*, 165(1):37–56. 1.5.1

Temperley, D. (1999). What's key for key? the krumhansl-schmuckler key-finding algorithm reconsidered. *Music Perception*, 17(1):65–100. 5.4.2

Temperley, D. (2006). A probabilistic model of melody perception. In *Proceedings of the 7th International Conference on Music Information Retrieval, ISMIR 2006*, Victoria, Canada. (document), 1.1

Theodoridis, S. and Koutroumbas, K. (2008). *Pattern Recognition, Fourth Edition*. Academic Press. 1.2

Tolonen, T. and Karjalainen, M. (2003). A computationally efficient multipitch analysis model. *IEEE Transactions on Speech and Audio Processing*, 8(6):708–716. 4.3

Tóth, L., Kocsor, A., and Csirik, J. (2005). On naive bayes in speech recognition. *International Journal of Applied Mathematics and Computer Science*, 15(2):287–294. 3.3

Tsai, W.-H., Liao, S.-J., and Lai, C. (2008). Automatic identification of simultaneous singers in duet recordings. In *Proceedings of the 9th International Conference on Music Information Retrieval, ISMIR 2008*, pages 115–120, Philadelphia, Pennsylvania, USA. 1.3

Tzanetakis, G. and Cook, P. (2002). Musical genre classification of audio signals. *IEEE Transactions on speech and audio processing*, 10(5):293–302. 1.4.1

Tzanetakis, G., Ermolinskyi, A., and Cook, P. (2003). Pitch histograms in symbolic and audio music information retrieval. *Journal of New Music Research*, 32(2):143–152. 1.4.1, 1.5.1, 1.5.2

Tzanetakis, G., Jones, R., and McNally, K. (2007). Stereo panning features for classifying recording production style. In *Proceedings of the 8th International Conference on Music Information Retrieval, ISMIR 2007*, pages 441–444, Vienna, Austria. (document), 1.5.1

Uitdenbogerd, A. and Zobel, J. (1999). Melodic matching techniques for large music databases. In *MULTIMEDIA '99: Proceedings of the seventh ACM international conference on Multimedia (Part 1)*, pages 57–66, New York, NY, USA. ACM. 2.2.1

van Kranenburg, P. (2006). Composer attribution by quantifying compositional strategies. In *Proceedings of the 7th International Conference on Music Information Retrieval, ISMIR 2006*, pages 375–376, Victoria, Canada. (document), 2, 2.1, 2.1.6, 6.1, 6.3, 6.3

van Kranenburg, P. and Backer, E. (2004). Musical style recognition - a quantitative approach. In Parncutt, R., Kessler, A., and Zimmer, F., editors, *Proceedings of the Conference on Interdisciplinary Musicology (CIM04)*, Graz, Austria. (document), 1.3, 2.1.5, 6.1, 6.2

Witten, I. and Bell, T. (1991). The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on Information Theory*, 37(4):1085–1094. 3.2.2

Wołkowicz, J., Kulka, Z., and Keselj, V. (2008). N-gram based approach to composer recognition. *Archives of Acoustics*, 33(1):43–55. 1.5.1, 1.5.2

Yeh, C., Robel, A., and Rodet, X. (2005). Multiple fundamental frequency estimation of polyphonic music signals. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages iii/225–iii/228. 4.3

Zanette, D. (2008). Playing by numbers. *Nature*, 453(7198):988–989. 1.5.2