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

Jorge Calvo-Zaragoza and Jose Oncina

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

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

1 Introduction

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

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

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

2 The Clustering Problem

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

(a) Two strokes (b) One stroke

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

At this point, we have to deal with the open problem of the set of primitives to be considered. Some *ad-hoc* labeling could be used but it might not be appropriate for this task due to several reasons: the data would be clustered considering human similarity perception instead of computer-based similarity, which is what it is applied in the final system; labels would be created from the data available at that moment, thus generalization could be poor; clustering may need to be repeated after user interaction, in which new data would be received

and, therefore, the system must be adapted to actual user writing style. All these reasons lead us to perform an algorithmic-based clustering of the strokes found in pen-based music notation. As aforementioned, the main goal of this paper is not to give a good proposal of stroke labeling, but to measure the goodness and generalization of each possible clustering considered.

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

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

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

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

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

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

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

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

3 Dissimilarity Functions for Pen-based Music Notation Strokes

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

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

3.1 Raw stroke distance

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

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

3.2 Feature extraction

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

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

4 Experimentation

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

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

4.1 Results

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

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

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

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

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

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

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

5 Conclusions

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

Acknowledgements

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

References

1. de Amorim, R.: Constrained clustering with minkowski weighted k-means. In: Computational Intelligence and Informatics (CINTI), 2012 IEEE 13th International Symposium on. pp. 13–17 (Nov 2012)
2. Arthur, D., Vassilvitskii, S.: K-means++: The advantages of careful seeding. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms. pp. 1027–1035. SODA '07, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (2007)
3. Bainbridge, D., Bell, T.: The Challenge of Optical Music Recognition. Language Resources and Evaluation 35, 95–121 (2001)
4. Calvo-Zaragoza, J., Oncina, J.: Recognition of Pen-Based Music Notation: the HOMUS dataset. In: Proceedings of the 22nd International Conference on Pattern Recognition. pp. 3038–3043. Stockholm, Sweden (2014)
5. Freeman, H.: On the encoding of arbitrary geometric configurations. Electronic Computers, IRE Transactions on EC-10(2), 260–268 (1961)
6. García, S., Luengo, J., Herrera, F.: Data Preprocessing in Data Mining, Intelligent Systems Reference Library, vol. 72. Springer (2015)
7. Kim, J., Sin, B.K.: Online handwriting recognition. In: Doermann, D., Tombre, K. (eds.) Handbook of Document Image Processing and Recognition, pp. 887–915. Springer London (2014)
8. Kristensson, P.O., Denby, L.C.: Continuous recognition and visualization of pen strokes and touch-screen gestures. In: Proceedings of the 8th Eurographics Symposium on Sketch-Based Interfaces and Modeling. pp. 95–102. SBIM '11, ACM, New York, NY, USA (2011)
9. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. Tech. Rep. 8 (1966)
10. Rebelo, A., Capela, G., Cardoso, J.: Optical recognition of music symbols. International Journal on Document Analysis and Recognition 13(1), 19–31 (2010)
11. Rebelo, A., Fujinaga, I., Paszkiewicz, F., Marçal, A.R.S., Guedes, C., Cardoso, J.S.: Optical music recognition: state-of-the-art and open issues. IJMIR 1(3), 173–190 (2012)
12. Rico-Juan, J.R., Iñesta, J.M.: Confidence voting method ensemble applied to off-line signature verification. Pattern Anal. Appl. 15(2), 113–120 (Apr 2012)
13. Rico-Juan, J.R., Iñesta, J.M.: Edit distance for ordered vector sets: A case of study. In: Yeung, D.Y., Kwok, J., Fred, A., Roli, F., de Ridder, D. (eds.) Structural, Syntactic, and Statistical Pattern Recognition, Lecture Notes in Computer Science, vol. 4109, pp. 200–207. Springer Berlin Heidelberg (2006)
14. Rokach, L.: A survey of clustering algorithms. In: Data Mining and Knowledge Discovery Handbook, 2nd ed., pp. 269–298 (2010)
15. Sakoe, H., Chiba, S.: Readings in speech recognition. chap. Dynamic programming algorithm optimization for spoken word recognition, pp. 159–165. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1990)
16. Theodoridis, S., Koutroumbas, K.: Pattern Recognition, Third Edition. Academic Press, Inc., Orlando, FL, USA (2006)
17. Wagstaff, K., Cardie, C., Rogers, S., Schrödl, S.: Constrained k-means clustering with background knowledge. In: Proceedings of the Eighteenth International Conference on Machine Learning. pp. 577–584. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2001)