2014-10

# Music genre classification using traditional and relational approaches

Brazilian Conference on Intelligent Systems, 3th, 2014, São Carlos.
http://www.producao.usp.br/handle/BDPI/48624

# Music genre classification using traditional and relational approaches

Jorge Valverde-Rebaza, Aurea Soriano, Lilian Berton, Maria Cristina Ferreira de Oliveira, Alneu de Andrade Lopes
Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo - Campus de São Carlos
13560-970 São Carlos, SP, Brazil
{jvalverr, asoriano, lberton, cristina, alneu}@icmc.usp.br

***Abstract*—Given the huge size of music collections available on the Web, automatic genre classification is crucial for the organization, search, retrieval and recommendation of music. Different kinds of features have been employed as input to classification models which have been shown to achieve high accuracy in classification scenarios under controlled environments. In this work, we investigate two components of the music genre classification process: a novel feature vector obtained directly from a description of the musical structure described in MIDI files (named as structural features), and the performance of relational classifiers compared to the traditional ones. Neither structural features nor relational classifiers have been previously applied to the music genre classification problem. Our hypotheses are: (i) the structural features provide a more effective description than those currently employed in automatic music genre classification tasks, and (ii) relational classifiers can outperform traditional algorithms, as they operate on graph models of the data that embed information on the similarity between music tracks. Results from experiments carried out on a music dataset with unbalanced distribution of genres indicate these hypotheses are promising and deserve further investigation.**

***Keywords*-Music genre classification, music features, relational classification, data graph models.**

## I. Introduction

Users who purchase or download music from online collections available on the Web often formulate their preferences in terms of genre. A non-expert person can identify the genre of a piece with 72% accuracy after hearing a three-second segmentation of the music [1]. Nonetheless, this could hardly be done manually with such a high accuracy on a large collection. Manual classification is a demanding process due to the time and expertise required. Moreover, results obtained would very likely be highly dependent on the experts's musical knowledge and previous experience. This scenario has motivated the development of several computational algorithms for automatic music classification [2], [3].

Any classification process requires a prior step of feature extraction. The choice of the type of feature is determined by the properties to be considered in the classification. In the specific case of music, features may be based on e.g., the instruments played, the performer, the time duration of the music, its genre, or other. Classification by genre is very usual and distinct approaches are reported in the literature to extract features capable of capturing the relevant properties to identify genre. One possibility is to focus on capturing elements of the underlying musical structure, which provides highly informative content for this purpose. The underlying structure of a music strongly relates to its high-level characteristics such as repetitions, interleaving of themes and choruses, presence of breaks, changes in time signatures, etc. Even considering these high-level features, it is still quite difficult to represent the music structure in a compact and meaningful form [4].

The studies reported in this paper take as input music descriptions given in the MIDI format (*Musical Instrument Digital Interface*) [5]. This representation allows reconstructing an approximate representation of the music as a symbolic score. Now established as a standard interchange format across multiple hardware devices and software platforms, MIDI descriptions of music are particularly attractive to users with musical knowledge and those interested in interactive applications and performances. MIDI is considered also a format more convenient than audio file formats to extract precise high-level musical information [6]. Due both to the symbolic nature of the information they carry and their processing speed, MIDI files also facilitate extracting features from entire recordings.

Once they are available, the descriptive feature vectors are input into the classification step. Classifiers such as *kNN* [7], *Support Vector Machines* [8]–[10], *Co-training* [2], [11] among others have been employed for music genre classification. These solutions achieve high accuracy in controlled environments, i.e., without considering class imbalance or feature vectors of different sizes. In particular, the work by Chai and Vercoe [12] addresses classification of MIDI recordings. The authors employ Hidden Markov Models to classify three types of Western folk music (Austrian, German and Irish), with 63% accuracy. Ponce de Leon and Inesta (2002) [13] describe a system to segment jazz and classical MIDI tracks in order to extract features, which are then classified employing Self-Organising Maps. They report that 77% of the instances have been classified correctly. Shank and Kuo (2003) [14] extract features based on melodies and chords, from pieces in four categories (Enya, Beatles, Chinese folk and Japanese folk). They had 38-55 recordings from each category, and achieved correctness rates ranging from 64% to 84%.

In this paper, we investigate relational algorithms for music genre classification in an imbalanced environment. Relational approaches have been successfully applied in several data mining tasks [15], [16] but we are not aware of previous

978-1-4799-5618-0/14 $31.00 
DOI 10.1109/BRACIS.2014.54

efforts employing them to handle music genre classification. In our experiments, relational classifiers outperformed traditional classification techniques, such as *Decision Trees*, *Naïve Bayes*, *Neural Networks* and *Support Vector Machines*. Relational representations explore information about the instances that goes beyond the attribute values, as they operate on graph models built from the data. In our studies we have employed the *k*NN, mutual-*k*NN and regular-*k*NN graph models [17]–[19]. We also investigate a novel feature vector derived from the underlying musical structure, comparing its performance with that of state-of-the-art features typically used in music genre classification using both the relational and the traditional classifiers.

The remaining of the paper is organized as follows: Section II introduces the definitions and previous work on music feature extraction, graph models employed to represent tabular data and the relational classifiers employed in this investigation; Section III describes the experimental setup and discusses the experimental results obtained; finally, Section IV summarizes the conclusions.

## II. Background

This section describes several strategies to extract features from music, as well as the data graph models and the relational classifiers considered in this work.

### A. Music features

Albeit research related to feature extraction from music mostly consider acoustic data as input, some approaches exist that handle symbolic representations. In this work we consider three distinct strategies to extract features from symbolic MIDI representations: histograms of notes and statistical moments, assumed as the state-of-the-art in music classification, also structural features derived from concepts of music theory, which have not been previously employed in this context.

**Histograms of notes.** Histograms are useful representations of music signals described both in symbolic or acoustic form [20]. In our studies we have employed a histogram of the 12 possible musical notes. From the MIDI description node pitches have been extracted and their frequencies constitute the histogram of notes. The MIDI specification only allows for 128 discrete notes (values between 0-127), which differentiates two equal notes in different octaves. We adopt a scale of 12 discrete notes obtained applying the modulo operator $\%12$ on the total number of notes.

**Statistical moments.** Simple statistical measures are able to capture the global features of a music. One approach consists in quantifying both the speed and the musical notes of a piece, attempting to differentiate both rhythm and melody [21] and somehow capturing an approximation of the human perception of such properties. The statistical moments we take as features are the mean, standard deviation, entropy and uniformity, computed from both the histogram of notes and the histogram of speeds.

**Musical structure.** In order to identify music tonality and chords, Soriano et al. [22] employed very basic concepts in music theory to extract structural features from MIDI data files. The approach starts by segmenting a music into compasses. The next step is to identify the tone of the music considering its progression. The harmonic is then considered to obtain the chords in which rotates the music. The result of these steps is a sequence of chords.

In order to identify patterns, a string matching algorithm (e.g., the *Horspool* algorithm) is applied on the chords sequence to obtain subsets. Initially, we seek for chord subsequences of size $\frac{S}{2}$, where $S$ is the total number of chords. This process is iterative, with the subset size decreasing by 1 at each iteration. A list of patterns of different sizes is thus obtained, from which a feature vector is formed by numbers representing the size of each pattern identified.

### B. Data graph models

Formally, a graph $G = (V, E)$ consists of two sets $V$ and $E$. The elements of $V = \{v_1, v_2, \ldots, v_N\}$, with cardinality $|V| = N$, are the vertices of $G$, whereas the elements of $E = \{e_1, e_2, \ldots, e_M\}$ are the edges connecting vertex pairs, with $|E| = M$. A weighted graph also includes a set of values (weights) $W = \{w_1, w_2, \ldots, w_M\}$, $w_i \in \Re$ associated with the edges.

Graphs [1] are widely employed to model real-world problems. In many situations a 'natural' graph structure exists, such as in social, authorship or citation networks, the Internet, power grid networks, etc. In other cases the data is not inherently relational, but a graph may still be constructed from a data set given by an attribute-value matrix. Several authors have addressed the problem of building graph models from data, and many strategies are possible [17]–[19]. It is clear from previous work that the choice of the data graph model affects the outcome of graph-based data mining algorithms, i.e., the effectiveness of specific graph models will vary depending on the specific characteristics of the problem under analysis and the algorithms employed.

Popular data graph models are typically based on assessing neighborhoods, or pairwise similarities between data instances. Building a graph model from tabular data typically involves three steps: i) choosing a similarity function to obtain a pairwise similarity matrix; ii) defining the connections to obtain an appropriate graph model that represents the data manifold; and iii) applying a graph-based mining algorithm. The problem can be formulated as a supervised, unsupervised or semi-supervised mining paradigm [18].

Consider $N$ data points $\{X_1, X_2, \ldots, X_N\}$ assumed to be independently and identically distributed from some probability distribution $P$. Each data point is mapped to a graph vertex, with weighted edges representing the similarity between the corresponding data point pairs. A complete graph includes all pairwise relationships, but a strategy often found in the literature is to connect each vertex only to its $k$ nearest neighbors (i.e., the $k$ closest or most similar data points). The resulting nearest-neighbor graphs, known as *k*NN graphs, are

[1] We do not distinguish network and graph.

more adaptive to different domains. In this work we derive graph models of the music data considering three variations of the *k*NN strategy.

In the *symmetric kNN model*, or simply *k*NN, each vertex is connected with its $k$ closest neighbors as given by a similarity function, i.e. an edge is created between vertices $v_i$ and $v_j$ if $v_i \in k_N(v_j) \vee v_j \in k_N(v_i)$, where $k_N$ is the set of vertices formed by the $k$ nearest neighbors of $v$.

In the *mutual-kNN model* two vertices are connected only if the neighborhood pertinence condition is met by both, i.e. there is an edge between vertices $v_i$ and $v_j$ if $v_i \in k_N(v_j) \wedge v_j \in k_N(v_i)$. The mutual-*k*NN graphs tend to produce disconnected graphs, particularly for small values of $k$. To avoid this problem we combine this method with *minimum spanning tree* graphs.

The third graph construction strategy considered generates regular graphs, and is referred as *regular-kNN* [19]. All vertices have the same degree, unlike the *k*NN model, which may include vertices with degree higher than $k$.

### C. Relational classifiers

Relational classifiers require a fully described graph (vertices and edges) with known labels for some of the vertices to predict the labels of the remaining vertices. Traditional classifiers would not work properly on graphs because they ignore pairwise dependency relations between vertices [23]–[25].

*We considered four relational classifiers in this study: probabilistic relational neighbor (*prn*),* weighted vote relational neighbor (*wvrn*), *network-only Bayes* (*no-Bayes*) and *network-only link-based* (*no-lb*). The *prn* classifier estimates class membership probabilities by assuming that the label of a vertex depends only on its immediate neighbors and that linked vertices tend to belong to the same class [23]. Similarly, the *wvrn* classifier estimates class membership probabilities by assuming that linked nodes tend to belong to the same class and considering the weighted mean of the class-membership probabilities for the neighborhood of each vertex analyzed [25]. The *no-Bayes* classifier employs multinomial naïve Bayesian classification based on the classes of the neighborhood of each vertex [25]. These three relational classifiers use the relaxation label as a collective inference method.

The *no-lb* classifier creates a feature vector for a vertex by aggregating the labels of its neighborhood and then use logistic regression to build a discriminative model based on those feature vectors [24]. As proposed by Lu&Getoor [24], three aggregation methods have been considered: binary-link (*no-lb-binary*), mode-link (*no-lb-mode*) and count-link (*no-lb-count*). Another aggregation method considered is class-distribution-link (*no-lb-distrib*), proposed by Macskassy&Provost [25], which uses the iterative classification as a collective inference method.

## III. Experiments

We carried out experiments comparing the performance of traditional and relational classifiers on three distinct feature sets obtained from a music MIDI data set with imbalanced genre distribution.

### A. Data set

The music collection has been compiled by Soriano [22] and is publicly available for research purposes[2]. It includes 919 MIDI files describing audio tracks with different time durations and manually classified into four genres, namely *Classical*, *Brazilian Backcountry*, *Pop/Rock* and *Jazz*. The number of samples in each genre varies, as described in Table I.

TABLE I
Music genre distribution for the music collection used

| Genre | # Tracks |
|---|---|
| *Classical* | 31 |
| *Brazilian Backcountry* | 243 |
| *Pop/Rock* | 550 |
| *Jazz* | 95 |
| **Total** | 919 |

### B. Experimental setup

The experiments were carried out on distinct representations of the music collection (i.e., three sets of feature vectors) considering multiple alternatives of conventional and relational classifiers for the music genre classification task.

The music collection has been pre-processed to obtain: (i) the Histogram data set, in which each track is described by a feature vector formed by 12 attributes, obtained from the histogram of notes; (ii) the Moments data set, in which each track is described by a feature vector formed by 8 attributes generated from the statistical moments; and (iii) the Structure data set, formed by feature vectors that capture the structural patterns, as described in Section II. Each feature vector is described by a varying number of attributes (at most 250), depending on the number and size of the patterns identified.

Graph models have been constructed from each feature data set employing the *k*NN, mutual-*k*NN and regular-*k*NN strategies. For each strategy, undirected graphs have been built varying the choice of $k$ within the interval $[1, 15]$. For the Moments data set, graph edge weights indicate dissimilarity, taking the pairwise Euclidean distance between the data points as an approximation of data point dissimilarity, whereas for the Histogram and Structure data sets edge weights are determined computing the pairwise DTW (*Dynamic Time Warping* [26]) distance. DTW is widely employed to measure dissimilarity between time series, since it can account for time displacements and series of differents sizes [27]. It gave us an appropriate alternative to compute the pairwise dissimilarities for the Histogram and Structure feature sets.

The following traditional classification approaches have been considered: *decision tree* (J48), naïve Bayes (NB), *multilayer perceptron with backpropagatio*n (MLP) and *support vector machine* (SMO) classifiers. We used the Weka[3]

[2]http://www.icmc.usp.br/ asoriano/download.html
[3]http://www.cs.waikato.ac.nz/ml/weka

implementations with standard configuration. The following relational classification approaches have been considered: *weighted vote relational neighbor* (*wvrn*), *network-only Bayes* (*no-Bayes*), *probabilistic relational neighbor* (*prn*) and *network-only link-based* (*no-lb*) classifiers, in their Netkit-SRL[4] implementations with standard configuration. For the *network-only link-based* classifier we employed models mode-link (*no-lb-mode*), count-link (*no-lb-count*), binary-link (*no-lb-binary*) and class-distribution-link (*no-lb-distrib*). For both traditional and relational classification we adopted the 10-fold cross validation procedure.

### C. Results

Due to the class imbalance distribution in our music collection we analyze the results of the distinct classification alternatives using the AUC (area under the receiver operating characteristic curve) measure.

Table II shows AUC average values for the traditional classifiers considered. Tables III, IV and V show AUC average values for the relational classifiers operating on data graph models built with the *k*NN, mutual-*k*NN and regular-*k*NN strategies, respectively. For the relational classifiers each entry shows, in parentheses, the choice of $k$ which resulted in the model leading to the best performance. In all tables, the best results for the multiple configurations compared is emphasized in bold, whilst entries highlighted in gray indicate the classification configuration that achieved the best performance on a specific feature set (Histogram, Moments or Structure). The last row in all tables shows the average ranking of the performance of each classifier for the feature sets considered.

TABLE II
TRADITIONAL CLASSIFIERS PERFORMANCE MEASURED BY AUC

| | **J48** | **NB** | **MLP** | **SMO** |
|---|---|---|---|---|
| **Histogram** | 0.619 | 0.607 | 0.665 | 0.506 |
| **Moments** | 0.706 | 0.750 | 0.771 | 0.585 |
| **Structure** | **0.738** | **0.920** | **0.816** | **0.724** |
| **Average rank** | 2.667 | 2.000 | 1.333 | 4.000 |

From Table II, we observe that all traditional classifier perform better with the structural features. Observing the behavior of each classifier, MLP outperforms the others when histogram and moments features are used. NB outperforms the other classifiers when structural features are used.

Figure 1 shows results from a Nemenyi post-hoc test [28] where the critical difference (CD) calculated at 95 percentile is 2.71. The CD value is plotted just above the diagram, whereas the average ranks from Table II are plotted along the horizontal axis. The lowest (best) ranks are in the left side. We observe that the differences amongst the classifiers are not statistically significant, so they are connected by a black line in the diagram. Although there is no statistically significant difference among them, we observe that MLP and NB are those with the first and the second best overall performances, respectively.

[4]http://netkit-srl.sourceforge.net/index.html

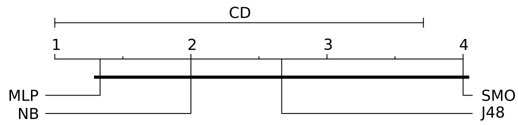



Fig. 1. Post-hoc test results for traditional classifiers performance

From Tables III, IV and V, we observe that all relational classifiers perform better with the structural features. In general, this scenario occurs with the networks built taking $k \geq 11$ for *no-lb-count*, *no-lb-distrib* and *wvrn* classifiers and taking $k \leq 9$ for the remaining relational classifiers. Inspecting the behavior of each classifier, *no-lb-count*, *no-lb-distrib* and *wvrn* outperform the others.

Figures 2, 3 and 4 show results from a Nemenyi post-hoc test, considering relational classifiers operating on data graph models obtained with the *k*NN, mutual-*k*NN and regular-*k*NN techniques, respectively. The critical difference (CD) for all post-hoc tests calculated at the 95 percentile is $5.20$. We observe that, although the performance differences are not statistically significant, the classifiers *no-lb-distrib*, *no-lb-count* and *wvrn* deliver the three best overall performances.

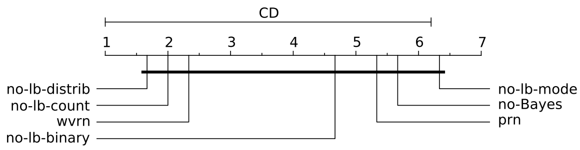



Fig. 2. Post-hoc test results for relational classifiers built on *k*NN networks

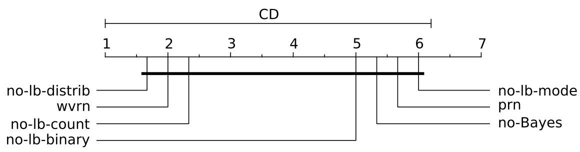



Fig. 3. Post-hoc test results for relational classifiers built on mutual-*k*NN networks

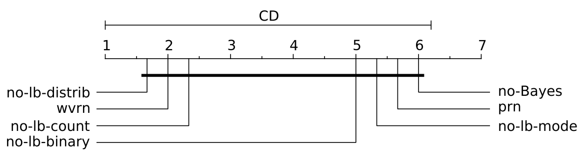



Fig. 4. Post-hoc test results for relational classifiers built on regular-*k*NN networks

Aiming to identify which networks resulted in the best classifier performances we pick just the best results from each relational classifier on each network built, i.e. entries

TABLE III
RELATIONAL CLASSIFIERS PERFORMANCE EVALUATED BY AUC IN $k$NN NETWORKS

| | **no-lb-mode** | **no-lb-count** | **no-lb-binary** | **no-lb-distrib** | **wvrn** | **no-Bayes** | **prn** |
|---|---|---|---|---|---|---|---|
| **Histogram** | 0.575 (k=11) | 0.723 (k=9) | 0.622 (k=2) | 0.71 (k=8) | 0.712 (k=9) | 0.515 (k=1) | 0.537 (k=1) |
| **Moments** | 0.547 (k=5) | 0.635 (k=13) | 0.575 (k=8) | 0.644 (k=7) | 0.644 (k=9) | 0.563 (k=2) | 0.571 (k=3) |
| **Structure** | **0.834 (k=7)** | **0.939 (k=14)** | **0.851 (k=4)** | **0.945 (k=14)** | **0.931 (k=15)** | **0.922 (k=15)** | **0.903 (k=9)** |
| **Average rank** | 6.333 | 2.000 | 4.667 | 1.667 | 2.333 | 5.667 | 5.333 |

TABLE IV
RELATIONAL CLASSIFIERS PERFORMANCE EVALUATED BY AUC IN MUTUAL-$k$NN NETWORKS

| | **no-lb-mode** | **no-lb-count** | **no-lb-binary** | **no-lb-distrib** | **wvrn** | **no-Bayes** | **prn** |
|---|---|---|---|---|---|---|---|
| **Histogram** | 0.621 (k=1) | 0.712 (k=10) | 0.626 (k=2) | 0.735 (k=12) | 0.727 (k=13) | 0.555 (k=1) | 0.571 (k=1) |
| **Moments** | 0.570 (k=1) | 0.657 (k=14) | 0.588 (k=2) | 0.633 (k=15) | 0.630 (k=14) | 0.578 (k=1) | 0.574 (k=1) |
| **Structure** | **0.864 (k=1)** | **0.955 (k=14)** | **0.818 (k=2)** | **0.963 (k=15)** | **0.964 (k=14)** | **0.913 (k=6)** | **0.902 (k=2)** |
| **Average rank** | 6.000 | 2.333 | 5.000 | 1.667 | 2.000 | 5.333 | 5.667 |

TABLE V
RELATIONAL CLASSIFIERS PERFORMANCE EVALUATED BY AUC IN REGULAR-$k$NN NETWORKS

| | **no-lb-mode** | **no-lb-count** | **no-lb-binary** | **no-lb-distrib** | **wvrn** | **no-Bayes** | **prn** |
|---|---|---|---|---|---|---|---|
| **Histogram** | 0.608 (k=1) | 0.724 (k=8) | 0.611 (k=2) | 0.737 (k=12) | 0.730 (k=12) | 0.544 (k=1) | 0.553 (k=1) |
| **Moments** | 0.569 (k=1) | 0.652 (k=13) | 0.571 (k=1) | 0.620 (k=8) | 0.625 (k=6) | 0.560 (k=1) | 0.565 (k=1) |
| **Structure** | **0.904 (k=1)** | **0.948 (k=11)** | **0.82 (k=1)** | **0.967 (k=15)** | **0.966 (k=15)** | **0.923 (k=3)** | **0.904 (k=2)** |
| **Average rank** | 5.333 | 2.333 | 5.000 | 1.667 | 2.000 | 6.000 | 5.667 |

emphasized in black in Tables III, IV and V, and analyze them statistically. Figure 5 shows the critical difference diagram obtained from this analysis where the CD value calculated at 95 percentile is $1.25$. We observe that differences in classifier performance considering distinct networks are not statistically significant, but regular-$k$NN network strategy provided the best network to leverage the relational classifiers.

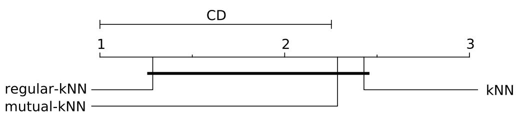



Fig. 5. Post-hoc test results for identify the influence of network construction techniques in relational classifiers

Figure 6 shows the performance of all relational classifiers in regular-$k$NN networks obtained with different values of neighborhood size $k$. From Figure 6 we observe that all classifiers maintain a stable behavior as $k$ varies, with the exception of the *no-lb-binary* classifier, for which performance decays as $k$ increases when employing regular-$k$NN networks built from the structural features. When regular-$k$NN networks are built from the histogram and moments features the classifiers performance remains under $0.75$ of AUC. When regular-$k$NN networks are built based on structural features most relational classifiers perform above $0.8$ of AUC for any choice of $k$. We also observe that *no-lb-distrib*, *no-lb-count* and *wvrn* outperform the other classifiers in all regular-$k$NN networks built, for all choices of $k$.

Finally, we compare the performance of traditional and relational classifiers. For this purpose we consider the two best performing traditional classifiers from Table II, i.e. MLP and NB, as well as the two best performing relational classifiers from Table V, i.e. *no-lb-distrib* and *wvrn* from regular-$k$NN network. Figure 7 shows results from a Nemenyi post-hoc test comparing the best traditional and relational classifiers. The CD value calculated at 95 percentile is $2.71$. We observe that although differences are not statistically significant, the relational classifiers are the best positioned.

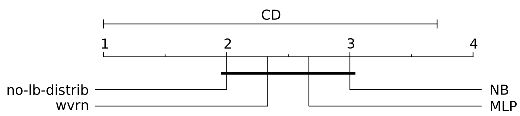
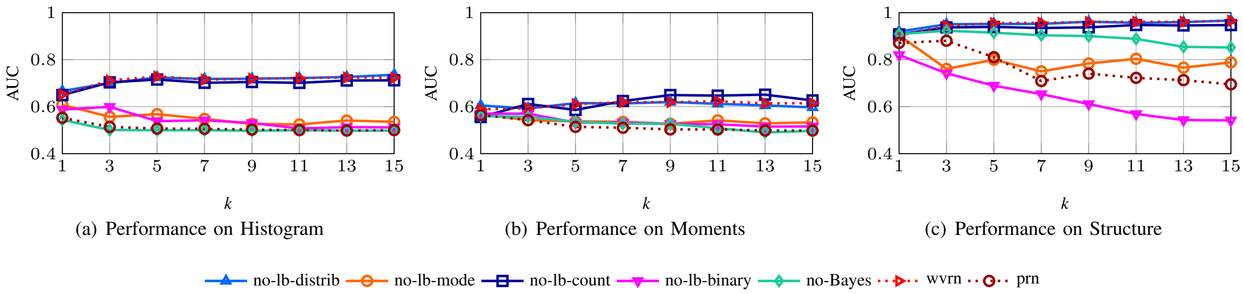



Fig. 7. Post-hoc test results for the comparison between the best traditional and relational classifiers

## IV. CONCLUSION

We report an investigation on the performance of relational algorithms for music genre classification. These approaches require deriving a relational representation of the input features. We considered several alternative graph construction strategies to obtain a relational representation of a particular music collection described in the MIDI format.

In addition to traditional features employed in the literature to represent music instances, such as histograms of notes and statistical moments, we also consider a novel feature vector, obtained from a MIDI description, that summarizes the musical structure and compare its effectiveness with that




(a) Performance on Histogram (b) Performance on Moments (c) Performance on Structure

Fig. 6. Relational classifiers performance evaluated by AUC in regular-$k$NN networks using different $k$ values

of the conventional features. We evaluated the classifiers on a particular music collection characterized by an imbalanced distribution of four music genres. The performance of the classifiers has been evaluated using the AUC.

Regarding the music features considered, results suggest that musical structure yields better classification performance as compared to histograms of notes or statistical moments, i.e., the features computed by inspecting the musical structure resulted in improved performance from both traditional and relational classifiers.

Regarding the relational data graph models considered, results suggest that best results are obtained with the regular-$k$NN graphs as compared to the graph models obtained with the $k$NN and the mutual-$k$NN graph construction strategies, i.e., the regular-$k$NN networks provided the relational model most suitable to improve the performance of the relational classifiers.

Regarding the performance of the relational versus the traditional classifiers, results suggest that relational classifiers perform better on this problem, i.e., they are more effective in capturing the musical features and their described dissimilarities resulting in improved accuracy in the music genre classification task.

## Acknowledgments

Supported by grants 2011/21880-3, 2011/14165-6, 2011/22749-8, 2012/24537-0 and 2013/12191-5 from São Paulo Research Foundation (FAPESP) and 151836/2013-2 from National Council for Scientific and Technological Development (CNPq).

## References

[1] D. Perrot and R. Gjerdigen, "Scanning the dial: An exploration of factors in the identification of musical style," in *SMPC*, 1999, p. 88.
[2] Y. Yaslan and Z. Cataltepe, "Audio genre classification with semi-supervised feature ensemble learning," in *IWMLM*, 2009.
[3] S. Poria, A. Gelbukh, A. Hussain, S. Bandyopadhyay, and N. Howard, "Music genre classification: A semi-supervised approach," in *Pattern Recognition*, 2013, pp. 254–263.
[4] N. Orio, "Music retrieval: A tutorial and review," *Found. Trends Inf. Retr.*, vol. 1, no. 1, pp. 1–96, 2006.
[5] H. Casabona and D. Frederick, *What Is MIDI?* Alfred Music Publishing, 1988.
[6] C. Mckay, "Automatic Genre Classification of MIDI Recordings," Master's thesis, McGill University, 2004.
[7] E. Pampalk, A. Flexer, G. Widmer *et al.*, "Improvements of audio-based music similarity and genre classificaton." in *ISMIR*, vol. 5, 2005, pp. 634–637.
[8] M. I. Mandel and D. P. Ellis, "Song-level features and support vector machines for music classification," in *ISMIR*, 2005, pp. 594–599.
[9] T. Lidy and A. Rauber, "Evaluation of feature extractors and psycho-acoustic transformations for music genre classification." in *ISMIR*, 2005, pp. 34–41.
[10] N. Scaringella and D. Mlynek, "A mixture of support vector machines for audio classification," *IEEE MIREX, London*, 2005.
[11] Y. Xu, C. Zhang, and J. Yang, "Semi-supervised classification of musical genre using multi-view features," in *ICMC*, 2005, pp. 5–9.
[12] W. Chai and B. Vercoe, "Folk music classification using hidden markov models," in *ICAI*, 2001.
[13] P. de Leon and J. Inesta, "Musical style identification using self-organising maps," in *WEDELMUSIC*. IEEE Computer Press, 2002, pp. 82–89.
[14] M.-K. Shan, F.-F. Kuo, and M.-F. Chen, "Music style mining and classification by melody." in *ICME (1)*. IEEE, 2002, pp. 97–100.
[15] J. Valverde-Rebaza and A. A. Lopes, "Exploiting behaviors of communities of twitter users for link prediction," *Social Network Analysis and Mining*, vol. 3, no. 4, pp. 1063–1074, 2013.
[16] N. M. Llerena, L. Berton, and A. A. Lopes, "Graph-based cross-validated committees ensembles," in *CASoN*, 2012, pp. 75–80.
[17] M. Maier, M. Hein, and U. Luxburg, "Cluster identication in nearest-neighbor graphs." in *ICALT*, 2007, pp. 196–210.
[18] T. Jebara, J. Wang, and S.-F. Chang, "Graph construction and b-matching for semi-supervised learning," in *ICML*, 2009, pp. 441–448.
[19] D. A. Vega-Oliveros, L. Berton, A. M. Eberle, A. A. Lopes, and L. Zhao, "Regular graph construction for semi-supervised learning." in *J.Phys.:Conf. Ser.*, 2014.
[20] G. Tzanetakis, A. Ermolinskyi, and P. Cook, "Pitch histograms in audio and symbolic music information retrieval," in *ISMIR*, 2002, pp. 31–38.
[21] O. Lartillot, P. Toiviainen, and T. Eerola, "A matlab toolbox for music information retrieval," in *GfKl*, 2007, pp. 261–268.
[22] A. Soriano, F. Paulovich, L. Nonato, and M. Oliveira, "Visualization of music collections based on structural similarity," in *SIBGRAPI 2014*, ser. XXVII Conference on Graphics, Patterns and Images, 2014.
[23] S. A. Macskassy and F. J. Provost, "A simple relational classifier," in *2nd Workshop on Multi-Relational Data Mining*, 2003.
[24] Q. Lu and L. Getoor, "Link-based classication," in *ICML*, 2003, pp. 496–503.
[25] S. A. Macskassy and F. J. Provost, "Classification in networked data: A toolkit and a univariate case study," *JMLR*, vol. 8, pp. 935–983, 2007.
[26] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 26, no. 1, pp. 43–49, 1978.
[27] R. Niels, "Dynamic time warping," Master's thesis, University of Nijmegen, 2004.
[28] J. Demsar, "Statistical comparisons of classifiers over multiple data sets," *JMLR*, vol. 7, pp. 1–30, 2006.