STRING METHODS FOR FOLK TUNE GENRE CLASSIFICATION

Ruben Hillewaere and **Bernard Manderick**
Computational Modeling Lab
Department of Computing
Vrije Universiteit Brussel
Brussels, Belgium
{rhillewa,bmanderi}@vub.ac.be

Darrell Conklin
Department of Computer Science and AI
Universidad del País Vasco UPV/EHU
San Sebastián, Spain
IKERBASQUE, Basque Foundation for Science
Bilbao, Spain
conklin@ikerbasque.org

ABSTRACT

In folk song research, string methods have been widely used to retrieve highly similar tunes or to perform tune family classification. In this study, we investigate how various string methods perform on a fundamentally different classification task, which is to classify folk tunes into genres, the genres being the dance types of the tunes. A new data set *Dance-9* is therefore introduced. The different string method classification accuracies are compared with each other and also with n-gram models and global feature models which have been proven to be useful in previous folk song research. They are shown to yield similar results to the global feature models, but are outperformed by the n-gram models.

1. INTRODUCTION

In the history of Music Information Retrieval (MIR), folk song databases have often been used as test collections to evaluate computational models, especially the Essen Folksong Collection [18] has been the test set for various MIR methods [19, 4]. The availability of large databases of labelled folk tunes and the fact that many of these contain mainly monophonic tunes, make it an attractive test bed for machine learning algorithms applied to musical sequences.

However, there is also a deeper interest in folk music from an ethnomusicological point of view, which is growing with the progression of advanced music data mining methods and the computational possibility of dealing with large folk song corpora. Archives of folk music are being handed over to computational musicologists to be analysed, clustered and subdivided into comprehensible subgroups. A self-organizing map is used to identify and analyse motive collections of 22 folk music cultures in Eurasia [10]. Automatic pattern discovery has been applied to Cretan folk songs, in order to describe the characteristic features of each song type and region [6].

Besides these descriptive tasks, a common task in computational folk music analysis is music retrieval, which is related to the concept of melodic similarity. Symbolic melodic similarity tasks have been proposed at the MIREX contests in 2005, 2006 and 2007, and were reintroduced since 2010, with the Essen folksong database as test set. Most of the methods applied at these contests rely on sequence alignment algorithms, which are shown to be succesful [7, 20, 21].

In this paper, however, we are interested in the the predictive task of folk music classification, where the goal is to predict the class label of an unseen folk tune. Sequence alignment methods have been used for the classification of folk songs into tune families, which are ensembles of tunes that all derive from the same initial tune [22]. It is shown that they outperform global feature approaches. In our previous work, we have shown that n-gram models outperform global feature models for the classification of European folk tunes into their geographic region [8]. Given the conclusions of these previous papers on the topic of folk tune classification, the question arises how sequence alignment methods and more generally string methods compare with n-gram models. We will thus pursue our comparative study by adding this third category of models, in order to shed some light on the existing folk music classification methods and their performance.

This paper investigates the performance of three string methods for the task of genre classification on a newly developed folk tune database *Dance-9*, containing 2198 dances of 9 different types, which we call the genres. String methods rely on a sequential music representation which views a piece as a string of symbols. A pairwise similarity measure between the strings is computed and used to classify unlabeled pieces. Obvious examples of string methods are the sequence alignment methods mentioned above, but also less standard approaches which have been used in the field of text classification, such as compression based techniques [13] or the string subsequence kernel method [12]. Compression based techniques have been applied to music classification [11] and clustering [5], but these methods have not been thoroughly compared with other existing methods for folk tune classification. String subsequence kernels have never been applied to music classification, but relate to the method presented by Pérez-Sancho

Dance type	number of pieces	relative number
Bourrée	59	2.7%
Hornpipe	108	4.9%
Jig	793	36.1%
March	76	3.5%
Polska	339	15.4%
Reel	453	20.6%
Schottische	119	5.4%
Strathspey	123	5.6%
Waltz	128	5.8%
Total	2198	

Table 1. The *Dance-9* collection: the number of pieces of each dance type.

et al. [16], where n-words are used to represent musical pieces as Boolean feature vectors, in order to classify MIDI files into jazz or classical music.

These string methods will be compared with both n-gram models and global feature models which we have studied in depth before [8], and the hypothesis of this study is that n-gram models will outperform both the global feature models and the string methods on the task of folk tune genre classification. It is unclear how the string methods will compare with the global feature models, since this classification task is essentially different than the tune family classification proposed by van Kranenburg [22]. Two folk dances, say for example two random waltzes, generally differ more than two tunes belonging to the same tune family.

The remainder of this paper is structured as follows. In the next section we discuss the data set and its representation that will be used for our experiments, then we describe the three string methods in detail, recapitulate the n-gram models and global feature models, before describing the experimental setup and reporting the results. We conclude with a discussion and future work.

2. DATA SET AND MUSIC REPRESENTATION

In this section we introduce a new folk tune database for our experiments, we illustrate two types of music representation and the features that will be used.

2.1 Data set : *Dance-9*

The corpus *Dance-9* is a large collection of European folk tunes which are subdivided into nine dance type categories, the largest ones being jigs, reels and polskas. An overview of the nine dance types and the class sizes is displayed in Table 1. The associated classification task is to predict the dance type of an unseen tune, which is what we call a *genre* classification task.

This corpus has been extracted from a much larger collection of approximately 14,000 folk songs transcribed in the ABC format, most of which are available on the web [1]. Many tunes contain metadata about their type of folk dance, and to construct *Dance-9* we only selected those with an unambiguous dance type annotation. Furthermore, we discarded all dance types that occurred insufficiently to have any statistical significance. To the remaining 2198 pieces, two preprocessing steps have been applied in order to end up with core melodies that fit for our research purpose: the first step ensures that all pieces are purely monophonic by retaining only the highest note of double stops which occured in some of the tunes, and in the second step we removed all performance information such as grace notes, trills, staccato, etc. Repeated sections and tempo indications were also ignored. Key and time signature information has been retained, even though they will not be explicitly used as musical features, as we explain in section 3.3. Finally, a conversion to clean quantized MIDI files is carried out with abc2midi. We removed all dynamic indications generated by the style interpretation mechanism of abc2midi.

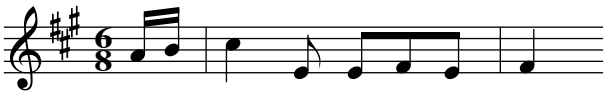
2.2 Music representation

In MIDI format, the folk tunes are reduced to a list of music events which are specified by their onset time, their pitch and duration. For the purpose of music data mining, one can represent a piece in various ways based on this information, and the chosen music representation is associated to the type of model one intends to use. We will discuss two main types of representation:

- *global feature vector*: a global feature describes an aspect of the whole piece with one single value, such as the average pitch or the fraction of ascending intervals. With a collection of global features, one can represent the piece as a multidimensional feature vector. There is a wide range of standard machine learning techniques available in toolboxes to classify such vectorized data.

- *string representation*: a piece can also be viewed as an ordered sequence of events, and every event is represented by an event feature of one's choice. In our case, the music events are note objects, with pitch and duration as basic event features, from which one can for example derive the melodic interval between the current and the previous note. Other examples are "duration ratio" or "melodic contour". This event feature sequence can be used directly for modelling, or it can first be mapped onto an ASCII symbol string.

Figure 1 illustrates these types of representation on the first measures of the Scottish jig "With a hundred pipers". The two upper lines show two global features "average pitch" and "rel. freq. M2", which is the relative frequency of major seconds. Some event features are illustrated on the next three lines, "pitch" being a basic one from which "melodic interval" is derived. The "interonset interval" tells the time span between the onset times of two successive notes (given in MIDI ticks here), which is similar to note duration, except when there are rests in the piece. The lower two lines show a possible mapping into strings given



average pitch	67.125							
rel. freq. M2	0.714							
pitch	69	71	73	64	64	66	64	66
melodic interval	⊥	+2	+2	-9	0	+2	-2	+2
interonset int.	⊥	12	12	48	24	24	24	24
int-string	'ffqjfdf'							
ioi-string	'lltiiii'							

Figure 1. Excerpt of the Scottish jig "With a hundred pipers", illustrating the difference between global features, event features and the string representation.

two event features, the melodic interval feature is mapped to "int-string" and the interonset interval to "ioi-string".

Each type of representation allows us to describe the music pieces on different levels of abstraction. When dealing with global features, one can create a large collection of features, each of them capturing information about a different musical aspect. The entire collection will be used to vectorize the pieces, and classification is achieved with standard machine learning algorithms. In the context of event features or the string representation, only one event feature is chosen for modelling, and this choice entirely implies the musical aspect to model and its granularity.

In order to do a fair comparison between the methods, we will only consider features that directly derive from *pitch* on the one hand and *duration* on the other hand, regardless of the used method. More precisely, for the string methods and n-gram models, separate models are built with the event features "melodic interval" and "interonset interval". For the global feature models, we manually created two collections of global features, one containing features derived from the pitches, and the other with features derived from the note durations. Any attributes that make use of other information such as the key or time signature are not included.

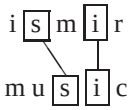
3. METHODS

3.1 String methods

In this section we give a detailed description of the string methods and the implementations that are used in our experiments.

3.1.1 Sequence alignment

The first category of string methods are the sequence alignment methods, which are very common in computational biology to compare protein sequences for example. Alignment algorithms define a similarity measure between two sequences of symbols, by estimating the minimal cost it takes to transform one sequence into the other by means of edit operations, such as substition, insertion and deletion. Therefore, this method is often referred to as "edit distance", which is in fact the Levenshtein distance. For example, the edit distance between the strings 'ismir' and 'music' is equal to 4, since the optimal alignment between them is given by



which means four edit operations are needed: two substitutions ('i' to 'm' and 'r' to 'c'), one insertion (the 'u') and one deletion (the 'm').

More advanced alignment algorithms and alignment scoring mechanisms have been developed depending on the application field. Mongeau and Sankoff [15] were among the first who designed a variant specifically for the alignment of musical sequences. For the purpose of our current research, we have used WEKA's implementation of the edit distance [2]. In a preliminary experiment, we tested this implementation on the melodic interval and interonset interval strings of the exact tune family database used by van Kranenburg [22]. We obtained one nearest neighbour classification accuracies of 94.2% and 80.6% respectively in comparison with his 92.0% and 74.0%, which shows that the general edit distance algorithm is sufficient for our comparative study at hand.

3.1.2 Compression based distance

The second type of string methods are compression based techniques, which also define a distance measure between two strings, by using the concept of information distance inherited from information theory. Ideally, this distance would be represented as

$$d(x,y) = \frac{\max(K(x|y), K(y|x))}{\max(K(x), K(y))},$$

where $K(x)$ is the Kolmogorov complexity of string x, and $K(x|y)$ is the conditional complexity of string x given string y. The underlying motivation behind this distance is to compute how much information is not shared between the two strings relatively to the information that they could maximally share. Since the Kolmogorov complexity $K(x)$ can not be exactly computed, it is approximated by the length of the compressed version of the string using a compressor C, denoted by $C(x)$. The information distance is thus approximated by the normalized compression distance [5]:

$$\mathrm{NCD}(x,y) = \frac{C(xy) - \min(C(x), C(y))}{\max(C(x), C(y))},$$

where xy represents the concatenation of the strings x and y. Various types of compressors can be used to estimate the Kolmogorov complexity, in our experiments we used `bzlib`, which is a block sorting text compression algorithm. For instance, the normalized compression distance

between 'ismir' and 'music' is computed as follows:

$$\begin{aligned} C(\text{'ismir'}) &= 344, \\ C(\text{'music'}) &= 336, \\ C(\text{'ismirmusic'}) &= 352, \end{aligned}$$

which are the compressed sizes in bits. So,

$$\text{NCD('ismir','music')} = \frac{352 - 336}{344} = 0.046512.$$

This number represents how different the two strings are, it is generally contained in $[0, 1]$.

3.1.3 String subsequence kernel

The third kind of string method is the string subsequence kernel method (SSK), which has been developed for text classification [12]. This approach computes a similarity measure between strings based on the number and form of their common subsequences. Given any pair of two strings, SSK will find all common subsequences of a specified length k, also allowing non-contiguous matches, although these are penalized with a decay factor $\lambda \in (0, 1)$. For example,

$$\text{SSK}(k = 2, \text{'ismir','music'}) = \lambda^5 + \lambda^6,$$

because there are two common subsequences 'si' and 'mi', and the lengths of the matches are the exponents of λ :

	'ismir'		'music'		
	match	l_1	match	l_2	$l_1 + l_2$
'si'	'**s**m**i**'	3	'**si**'	2	5
'mi'	'**mi**'	2	'**m**us**i**'	4	6

To speed up the algorithm, one can reduce the search space of subsequences by specifying a maximal exponent m of λ, which is called λ-pruning; it has been shown there is little quality loss due to this pruning. In our experiments, we looked for short subsequences ($k = 2, 3, 5$) allowing few or no non-contiguous matches. The parameter λ was set to a default value of 0.5.

The general idea behind these three string methods is to determine a similarity measure between two "stringified" music pieces x and y. A similarity measure between strings is either derived from a distance metric $d(x, y)$, such as the edit distance or the normalized compression distance (NCD), or else it is computed directly from the strings, which is what the string subsequence kernel (SSK) does. Given a distance metric $d(x, y)$, one can simply use a nearest neighbour approach to classify unseen test pieces, replacing the usual Euclidean distance with $d(x, y)$. In the case of the string subsequence kernel, the computed similarity measure is considered as the kernel function of a support vector machine, a state of the art classifier that learns non-linear decision boundaries between classes.

3.2 n-gram models

In this section we briefly recall how an n-gram model can be employed for classification of music pieces, for more details we refer to our previous work [8]. In a first stage, every piece of the music data is transformed into an event feature sequence according to a feature of choice. In the training phase, for each class the n-grams are counted to estimate the probability distribution of the musical "words" in that particular class. Given a test piece represented by its event feature sequence, the piece probability is then computed as the joint probability of the individual events in the piece according to the learned distribution, with the assumption that the probability of an event only depends on the $n-1$ previous events. Finally, the test piece is assigned to the class with the highest piece probability, which is the most likely to have "generated" the piece.

Note that the music representation is basically the same as for the string methods, but the essential difference between these methods is that an n-gram model aims to model the transitions for a given class, whereas a string method computes a pairwise similarity measure between pieces.

3.3 Global feature models

In this section, we describe what global features were chosen for our experiments. Two separate global feature sets were made, with features derived from the pitch on the one hand and from the note durations on the other hand. The features were chosen among the following (see Table 2) :

- The *Alicante* set of 28 global features, proposed by P.J. Ponce de Léon and J.M. Iñesta in [17] to classify a collection of 110 MIDI tunes in the genres jazz and classical. Among these, 7 are derived from pitch, e.g. "average melodic interval" and 12 from duration, like "duration range".

- The *Jesser* set, containing 39 statistics designed by B. Jesser [9], 31 of which are pitch-based features. Most of these are basic relative interval counts, like "dminthird", measuring the fraction of descending minor thirds, for all ascending and descending intervals in the range of the octave. This set also includes 6 features derived from the note durations.

- The *McKay* set of 101 global features [14], which were used in the winning 2005 MIREX symbolic genre classification experiment and computed with McKay's software package jSymbolic [3]. This set is composed of a wide range of features, since it was intended to classify orchestrated MIDI files. We retained 34 features based on pitch, for example "Direction of motion", i.e. the fraction of melodic intervals that are rising rather than falling, and 4 based on duration.

All features derived from pitch were joined to obtain a set of 73 features, since there are not many overlapping features. With the same procedure applied to the duration features a set of 22 features was formed. We recall that any features derived from the meter or the key signature have not been retained, since we want to compare the methods and representations on the basis of the same information.

Global feature set	pitch	duration
Alicante	8	12
Jesser	31	6
McKay	34	4
Total	73	22

Table 2. Global features that were selected for our experiments, divided into those derived from pitch and those from duration.

Since global features represent every instance as a multidimensional feature vector, any standard machine learning classifier can be applied to get a performance accuracy.

4. RESULTS

In this section we describe the experimental setup and the classification results on the folk tune data set *Dance-9*. Since we are interested in the relative performance of the string methods, the n-gram models and the global feature models, we have computed 10-fold cross validation classification accuracies for each of the methods. Care has been taken to use the exact same cross validation folds in all experiments, and the classifier parameters (if applicable) have always been set to standard values to do an unbiased comparison between the methods.

The string methods edit distance and NCD have been evaluated using a one nearest neighbour approach (1NN), whereas SSK implies one works with a support vector machine. Different lengths of short subsequences have been examined ($k = 2, 3, 5$), and it was found that the best performances were obtained with contiguous matches; only those will be displayed. For the n-gram models, we constructed trigram and pentagram models, in direct comparison with the SSK method. The global feature vectors have been classified with nearest neighbour approaches as well as with a regular SVM kernel with a Radial Basis kernel Function (RBF). For all experiments with SVM, the parameter determining the softness of the decision boundary has been set to its default value, after verifying this does not penalize any of the methods.

The results are reported in Table 3, which have to be compared to a baseline classification accuracy of 36.1% one obtains by always choosing the largest class "Jig". The first column contains the results using only features related to pitch or melodic interval sequences, whereas the second column gives the results with the duration features and interonset interval sequences. It appears immediately that the latter leads to superior classification accuracies regardless of the method, with a difference of approximately 20% on average. This shows that the recognition of folk dance types on this corpus is easier to achieve with the duration representations than with melodic ones, which is not surprising since folk dances are commonly distinguished by their rhythmic patterns.

SSK appears to be the most powerful string method, especially with contiguous subsequences of length $k = 3$. However, when we increase the length to $k = 5$ the performance drops when using the melodic interval strings. NCD does not lead to any promising result, whereas the edit distance does reasonably well, especially if one keeps in mind its computation time is a lot shorter than for SSK.

String methods	melodic int.	interonset int.
EditDist (1NN)	50.0	70.0
NCD (1NN)	48.0	68.0
SSK ($k = 2$, $m = 4$)	54.0	71.2
SSK ($k = 3$, $m = 6$)	60.8	72.9
SSK ($k = 5$, $m = 10$)	38.4	68.9
n-gram models	melodic int.	interonset int.
$n = 3$	60.7	71.9
$n = 5$	**66.1**	**76.1**
Global feature models	pitch	duration
1NN	40.3	66.9
5NN	44.8	69.3
SVM, RBF-kernel	53.5	67.7

Table 3. The 10-fold cross validation classification accuracies with all methods using the interval and duration representation.

The comparison across all the methods reveals that the pentagram model clearly outperforms the other approaches, with both the melodic interval and the interonset interval features. The trigam model also outperforms most other methods with both representations, except for SSK with $k = 3$ that achieves very similar results. On this corpus, the string methods and global feature models yield similar results with the melodic features, but on the rhythmic features there is a slight advantage for all the string methods except NCD. For the global feature models, the SVM with RBF-kernel performs better than both nearest neighbour models with the melodic features, but with the rhythmic features there is no benefit in using the more sophisticated SVM classifier over the simple nearest neighbours approach.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we thorougly examined the classification performances of three string methods and compared them with well-known other classification methods on a large folk dance dataset with nine classes. We have described the difference between the underlying types of music representation and features, and have shown that features based on duration lead to better classification models than features based on pitch, no matter if they are used to represent the music with an event feature sequence or string, or with a collection of global features.

The comparison between the methods has revealed that the n-gram models outperform both the string methods and the global feature models, which is in agreement with our earlier survey [8]. This result proves the effectiveness of modelling the transitions within a musical sequence and supports our hypothesis that the n-gram model should be the default model for folk tune classification.

However, the string methods generally perform slightly

better than the global feature models, particularly with the string subsequence kernel which obtains the highest accuracies among the string methods. This first result on music classification with the string subsequence kernel is encouraging for future work. The alignment methods which have been shown to be efficient in tune family classification [22] cannot measure up to the pentagram model on this genre classification task. We are currently doing more research on other folk song databases to get a broader view of the alignment method performance. In particular we are interested in discovering which models are most effective with respect to the precise classification task at hand.

6. REFERENCES

[1] `http://trillian.mit.edu/~jc/cgi/abc/tunefind`.

[2] `http://www.cs.waikato.ac.nz/ml/weka/`.

[3] `http://jmir.sourceforge.net/jSymbolic.html`.

[4] W. Chai and B. Vercoe. Folk music classification using hidden Markov models. In *Proceedings of International Conference on Artificial Intelligence*, Seattle, Washington, USA, 2001.

[5] R. Cilibrasi and P. Vitányi. Clustering by compression. *Information Theory, IEEE Transactions on*, 51(4):1523–1545, 2005.

[6] D. Conklin and C. Anagnostopoulou. Comparative pattern analysis of Cretan folk songs. *Journal of New Music Research*, 40(2):119–125, 2011.

[7] C. Gómez, S. Abad-Mota, and E. Ruckhaus. An analysis of the Mongeau-Sankoff algorithm for music information retrieval. In *Proceedings of the 8th International Conference on Music Information Retrieval*, pages 109–110, Vienna, Austria, 2007.

[8] R. Hillewaere, B. Manderick, and D. Conklin. Global feature versus event models for folk song classification. In *Proceedings of the 10th International Society for Music Information Retrieval Conference*, pages 729–733, Kobe, Japan, 2009.

[9] B. Jesser. *Interaktive Melodieanalyse*. Peter Lang, Bern, 1991.

[10] Z. Juhász. Motive identification in 22 folksong corpora using dynamic time warping and self organizing maps. In *Proceedings of the 10th International Society for Music Information Retrieval Conference*, pages 171–176, Kobe, Japan, 2009.

[11] M. Li and R. Sleep. Melody classification using a similarity metric based on Kolmogorov complexity. In *Sound and Music Computing Conference*, Paris, France, 2004.

[12] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins. Text classification using string kernels. *The Journal of Machine Learning Research*, 2:419–444, 2002.

[13] Y. Marton, N. Wu, and L. Hellerstein. On compression-based text classification. In *Advances in Information Retrieval*, volume 3408 of *Lecture Notes in Computer Science*, pages 300–314. Springer Berlin / Heidelberg, 2005.

[14] C. McKay and I. Fujinaga. Automatic genre classification using large high-level musical feature sets. In *Proceedings of the 5th International Conference on Music Information Retrieval*, pages 525–530, Barcelona, Spain, 2004.

[15] M. Mongeau and D. Sankoff. Comparison of musical sequences. *Computers and the Humanities*, 24(3):161–175, 1990.

[16] C. Pérez-Sancho, J. M. Iñesta, and J. Calera-Rubio. Style recognition through statistical event models. *Journal of New Music Research*, 34(4):331–339, 2005.

[17] P. J. Ponce de Léon and J. M. Iñesta. Statistical description models for melody analysis and characterization. In *Proceedings of the 2004 International Computer Music Conference*, pages 149–156, Miami, USA, 2004.

[18] H. Schaffrath. The Essen folksong collection in the Humdrum Kern format. *Stanford, California: Center for Computer Assisted Research in the Humanities*, 1995.

[19] P. Toiviainen and T. Eerola. A method for comparative analysis of folk music based on musical feature extraction and neural networks. In *3rd International Conference on Cognitive Musicology*, pages 41–45, Jyväskylä, Finland, 2001.

[20] A. L. Uitdenbogerd. N-gram pattern matching and dynamic programming for symbolic melody search. *Proceedings of the Third Annual Music Information Retrieval Evaluation eXchange*, 2007.

[21] J. Urbano, J. Lloréns, J. Morato, and S. Sánchez-Cuadrado. Mirex 2010 symbolic melodic similarity: Local alignment with geometric representations. *Music Information Retrieval Evaluation eXchange*, 2010.

[22] P. van Kranenburg. A computational approach to content-based retrieval of folk song melodies. *SIKS dissertatiereeks*, 2010(43), 2010.