Journal of New Music Research, 2013
Vol. 42, No. 1, 1–18, http://dx.doi.org/10.1080/09298215.2012.718790

Routledge
Taylor & Francis Group

A Comparison between Global and Local Features for Computational Classification of Folk Song Melodies

Peter van Kranenburg[1], Anja Volk[2] and Frans Wiering[2]

[1]Meertens Institute, Netherlands; [2]Utrecht University, Netherlands

Abstract

In computational approaches to the study of variation among folk song melodies from oral culture, both global and local features of melodies have been used. From a computational point of view, the representation of a melody as a vector of global feature values, each summarizing an aspect of the entire melody, is attractive. However, from an annotation study on perceived melodic similarity and human categorization in music it followed that local features of melodies are most important to classify and recognize melodies. We compare both approaches in a computational classification task. In both cases, the discriminative power of features is assessed. We use a feature evaluation criterion that is based on the performance of a nearest-neighbour classifier. As distance measure for vectors of global features, we use the Euclidian distance. For the sequences of local features, we use the score of the Needleman–Wunsch alignment algorithm. In each of our comparisons, the local features correspond to the global features. In all cases, it appears that the local approach outperforms the global approach in a classification task for melodies, which indicates that local features carry more information about the identity of melodies. Therefore, locality is a crucial factor in modelling melodic similarity among folk song melodies.

1. Introduction

The question how to automatically determine the similarity between folk song melodies has a history of more than a century. In 1900, the Dutch musicologist Daniel François Scheurleer (1900) posed the question: 'Welche ist die beste Methode, um Volks- und volkmässige Lieder nach ihrer melodischen (nicht textlichen) Beschaffenheit lexikalisch zu ordnen?'.[1] To stimulate response, Scheurleer organized a competition, which marked the starting point of a long-lasting discussion about classification systems for folk song melodies. One of the main reasons to want such a system is the variability that exists between instances of folk songs as they are sung from memory. The melodies and the lyrics of folk songs are learned by imitation and participation rather than from written sources such as books. In the course of this oral transmission, changes occur to the melodies that range from minor alterations to deformation beyond recognition (Wiora, 1941). Therefore, any collection of folk song melodies that were gathered during field-work consists of groups of more or less related melodies. Later in the century, the concept of *tune family* (Bayard, 1950) was introduced to denote such a group of melodies that are supposed to have a common 'ancestor' in the line of oral transmission. The classification system Scheurleer envisioned has the aim to order a collection of folk song melodies such that melodies that are in the same tune family end up near each other in the resulting ordering. Important contributors to the discussion were, among others, Ilmari Krohn (1903), Béla Bartók and Zoltán Kodály (summarised by Suchoff, 1981), and Wolfgang Suppan and Wiegand Stief (1976). Features they used to order melodies are, for example, the number of phrases, the number of syllables in each phrase, the pitches of the cadence tones of the phrases,

[1]What is the best method for the lexical ordering of folk and folklike tunes? (Translation by Nettl 2005, p. 123).

Correspondence: Peter van Kranenburg, Meertens Institute, P.O. Box 94264, 1090 GG Amsterdam, Netherlands. E-mail: peter.van.kranenburg@meertens.knaw.nl

and so on. As early as in the 1940s, computers were used to process large collections of melodies. Bronson (1949) proposed a punch-card system to sort melodies according to various features. Bronson (1950) introduced a number of features for British–American folk song melodies. In decreasing order of importance, these are: (1) final cadence, (2) mid cadence, (3) first accented note, (4) first phrase cadence, (5) first accented note of second phrase, (6) penultimate stress of second phrase, etc. In an ordering according to these features, melodies from the same tune family are expected to be close to each other.

Most of the features that are mentioned thus far refer to specific locations in the melody. In contrast, most of the work by Steinbeck (1982) on melody groups in the Essen Folksong Collection (Schaffrath, 1995) is based on global features that summarize the entire melody in one value, such as the standard deviation of the pitch, the average interval size, and so on. For a set of 35 melodies, Steinbeck was able to cluster melodies into meaningful groups, such as hymns, children's songs and hunting songs, using 13 global features and a hierarchical clustering algorithm. An experiment with 500 melodies also led to clusters, but in this case the clusters were more difficult to characterize musically (Steinbeck, 1982, p. 346).

In a study based on the same corpus, Jesser (1991) employed global as well as local features. The set of global features consists of the frequencies of occurrence of a large number of intervals and duration-ratios (see Appendix A.2), while the local features include sequences of accent tones, sequences of cadence tones, form, and contour. Jesser did not achieve a fully automatic method to retrieve related melodies (p. 213), but she found related melodies by issuing a series of search commands among the feature values of the entire corpus. The tune family relationships between melodies that Jesser was able to find were mainly based on the representation of the melodies as sequences of local features, such as the sequence of cadence tones or the melodic contour. The global features did not provide much to the classification (Jesser, 1991, p. 258).

Zoltán Juhász's work is based on a vector representation of melodic contours (e.g. Juhász, 2009), which enables local comparisons between melodies. Eerola, Järvinen, Louhivuori and Toiviainen (2001), on the other hand, evaluate melodic similarity using global features, and Bohak and Marolt (2009) used global features for assessing similarity relations between Slovenian folk song melodies.

In a previous study, we showed that recurring motifs are important for the classification of tunes into tune families (Volk & Van Kranenburg, 2012). This conclusion was based on manual annotations by domain specialists at the Meertens Institute (Amsterdam), which hosts a large collection of Dutch folk song recordings, called *Onder de groene linde*. This prevalence of melodic motifs leads to the assumption that locality is important—perhaps indispensable—for the classification of melodies, and, consequently, for the (computational) modelling of similarity relations between melodies.

From a computational point of view, the employment of global features has the advantage that a melody is represented in a relatively simple way as a vector of feature values. Such a representation can be processed by numerous standard machine learning techniques. Therefore, if both approaches perform equally well, and if the aim is merely classification, it might be adequate to use such a standard approach, even when knowing that local features seem to play a major role in human assessment of melodic similarity. On the other hand, if the aim is to model the way humans relate melodies to each other, a local approach or a combined approach might be more suitable. These considerations are relevant for the research field of Music Information Retrieval, in which computational processing of music for retrieval purposes is studied (Downie, 2003). Therefore, the question how to model both musical content and (similarity) relations between musical objects is of central importance.

The question whether global or local features can be used equally well for recognition of melodies is interesting from a cognitive perspective as well. Is it because global features are more difficult to 'access' for humans that local features seem to be more important for the recognition of melodies, or do local features actually provide more information for recognition or classification? Though the topic of similarity in music has recently received substantial attention in music cognition research (see e.g. two special issues dedicated to this topic in *Musicae Scientiae*—Toiviainen, 2007, 2009), this question has not yet been studied systematically. For a broader overview on the relation of our study to the human assessment of melodic similarity we refer to Volk and van Kranenburg (2012).

In the current investigation, we compare the suitability of global and local features for discerning the members of a tune family in a corpus of melodies using a computational approach.

In a related study, Hillewaere, Manderick and Conklin (2009) did a similar comparison, taking the geographical origin as class labels (England, France, Ireland, Scotland, South East Europe, Scandinavia). In the current study, we confine to a single tradition (the Dutch) and we use the tune families as class labels instead, which directly relates the results of our study to previous and ongoing research in the area of Folk Song Research.

The general outline of the article is as follows. Section 2 provides further details on the data set we use and on the concept of tune family. In Section 3 we present the set of 88 global features that we use in our study, and we evaluate the discriminative power of both individual global features and sets of global features. In Section 4,

we compare the discriminative power of different sets of global features with the discriminative power of an alignment approach that uses corresponding local features, which was introduced in van Kranenburg (2010). We show that classification based on local features outperforms classification based on global features.

2. Data

The data set we work with is the collection *Onder de groene linde*, which consists of c. 7000 audio recordings of Dutch folk songs, made during the 1950s until the 1980s by ethnological field-workers Will Scheepers and Ate Doornbosch (Grijp, 2008). The collection is currently hosted by the Meertens Institute in Amsterdam, and accessible through the website of the Dutch Song Database.[2] Around 2500 of these recordings were encoded for computational processing. Next to the recordings, the collection also contains thousands of folk songs from written sources.

One of the tasks of the collection specialists of the institute is to classify the melodies into tune families. The concept of *tune family* is of central importance in the study of folk song melodies. This concept was introduced in the 1950s by Samuel Bayard to denote a group of melody instances that supposedly 'descend' from one single tune through the process of oral transmission. Later on, an extension of the concept was proposed by James Cowdery (1984). He considered other types of melodic relatedness than having a common 'ancestor' for establishing tune family membership. One of these is melodies being composed from the same 'pool' of motifs. Such a pool of motifs consists of concrete melodic material that is available to the folk musician for constructing a melody.

Our collection only contains 'end points' in the process of oral transmission. The full history of a melody, comprising the ancestral variants, is lost. Therefore, in practice, the assignment of a recorded tune to a tune family is done based on similarity relations between the melodies. As an example of the degree of variation that can be found among tune family members, Figure 1 shows incipits of four melodies from the tune family *Soldaat kwam uit de oorlog*.

In a previous study, our aim was to understand how these assignments are established (Volk & Van Kranenburg, 2012). We asked the collection specialists to annotate similarity relations among melodies in a subset of 360 melodies in 26 tune families: the *Annotated Corpus*. Appendix A.1 shows its composition. This subset was carefully selected such that the kinds of melodic relations among the 360 melodies are representative for the kinds of melodic relations among the collection as a whole. One of our findings was that recurrence of characteristic motifs plays a major role in the classification of the melodies, even more so than similarity of melodic contour or rhythm.

Because the Annotated Corpus is relatively small, we also employ an additional set of 4470 melodies from other Dutch tune families. This additional set allows one to test for scalability: if we find a set of features that offers enough information to discern tune families within the Annotated Corpus, we will test whether this set of features is also appropriate to discern the same tune families in the large corpus. A positive result would confirm the validity of the set of features.

Since the assignments of the melodies to the tune families were done in a careful process by the domain experts from the Meertens Institute, we consider the resulting partitioning of the dataset of high quality, such that it is suitable to test the discriminative power of the various classification approaches.

3. Melodic classification using global melody feature sets

In this section we evaluate the usefulness of global features for the classification of melodies into tune families. First, we assemble a set of 88 features that have previously been used in various studies (Section 3.1). Next, we assess individual features to see whether we can find features that are discriminative for all tune families (Section 3.3). Finally, we investigate subsets of features (Section 3.4). For both single features and subsets of features, we test the discriminative power for each individual tune family as well as for all tune families in the Annotated Corpus. To evaluate the discriminative power of a feature subset, we need a feature evaluation criterion, which is presented in Section 3.2.

3.1 The set of global features

We use the following three sets of features that are well known from the literature:

- 12 features provided by Steinbeck (1982).
- 39 features provided by Jesser (1991).
- 37 rhythm, pitch and melody features implemented in jSymbolic by McKay (2004).

Steinbeck and Jesser specifically designed their feature sets to study relations between folk songs within the Essen Folk Song Collection that are connected through the process of oral transmission. Because our corpus consists of such folk song melodies, the evaluation of these two feature sets is particularly interesting. McKay's set was designed as a general purpose feature set for musicological classification tasks. It contains a number of features that are not in the sets by Jesser and Steinbeck.

[2]http://www.liederenbank.nl

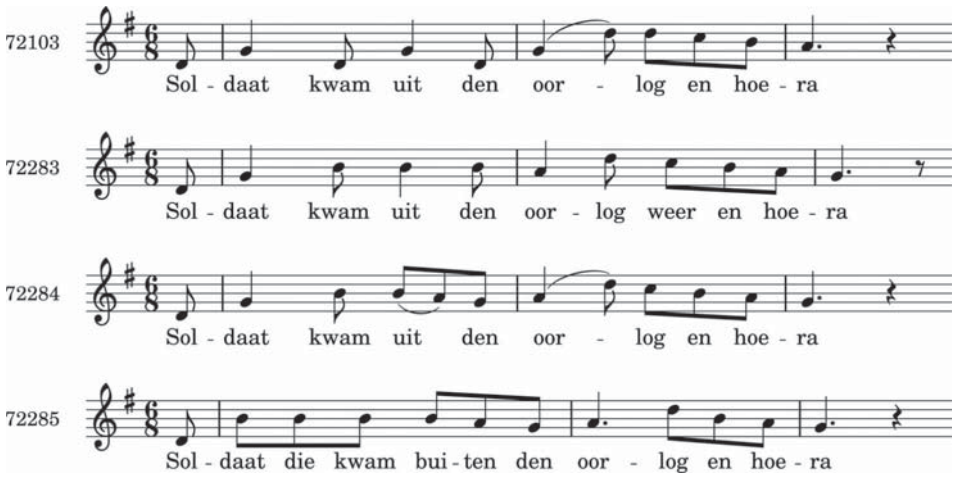

Fig. 1. Incipits of four members of the tune family *Soldaat*. The melodies are identified by their record number in the Dutch Song Database (http://www.liederenbank.nl). The full melodies and recordings can be accessed by entering the record number in the search box on the site of the Dutch Song Database.

All features for which absolute pitch is needed (e.g. Steinbeck's Mean Pitch) are discarded because the melodies are represented in various keys. The low-level, multidimensional features from the set of jSymbolic that are primarily needed to compute the values of other, higher-level features are discarded as well. Furthermore, categorical features and features that have the same value for all songs have not been included. Thus, we have a set of 88 features, which we characterize as 'global' because for each feature an entire song is represented by a single value. The complete list of features with descriptions is included in Appendix A.2.

Once all 88 feature values have been computed, a song is represented by a vector of 88 feature values, or, equivalently, by a point in the 88-dimensional feature space. The scaling of the values for the different features with respect to each other influences the distances between the song-representations in the feature space. Therefore, it is necessary to normalize the feature values such that they have comparable scales. For each feature we scale the values such that they have zero mean and a standard deviation of 1. This is achieved by subtracting the original mean and dividing by the original standard deviation. We do this both for the annotated set and for the full set separately.

3.2 A feature evaluation criterion

We need a measure to determine the discriminative power of a single feature, or of a set of features. In pattern recognition literature, such a measure is called a 'criterion'. It is commonly used by feature selection algorithms to find a subset of features that is particularly suited for a specific classification task. The subset with the highest criterion value is considered the 'best' subset (see e.g. Webb, 2002, p. 307ff).

The fraction of songs that is correctly classified into the right tune family using the feature set under consideration seems a good criterion for the discriminative power of that feature set. To take this approach, we need a classification algorithm. For that, we use the nearest neighbour classification rule: a song is classified into the tune family of the song that is closest in the feature space according to the Euclidean distance. This classification rule performs well if objects (songs) that belong to the same class (tune family) are close to each other in the feature space. Therefore, the criterion value indicates to what extent this is the case for the feature set under consideration. A feature selection algorithm using this criterion is expected to select a subset of features according to which the distances between songs from the same tune family are small compared to the distances to members of other tune families.

In most of our experiments, we are interested in classification of a relatively small set of songs among a larger collection. In any case, we are not directly interested in the classification performance for the additional 4470 songs. The tune families in the Annotated Corpus are selected by the domain specialists to be representative for the corpus as a whole. The additional songs just represent the 'rest of the world'. It is sufficient to know that they are from other tune families than the songs in the Annotated Corpus. As a consequence, we label all 4470 additional songs as 'Other', which results in very asymmetric class sizes. This leads to the following problem. In the case that most of the additional songs are classified correctly as 'Other', but none of the annotated songs are classified correctly, the overall classification accuracy is relatively high. In the extreme case that all songs in the data set would be classified as 'Other', the success rate would be $4470/4830 = 0.93$, while none of the songs we are interested in would have been classified

correctly. In our context, this situation constitutes a total failure, as the criterion should have value zero. We have to incorporate this selective interest in the criterion. Therefore, we compute the classification accuracy for the songs we are interested in and we correct it downwards for the songs in the rest of the corpus that are classified incorrectly into one of the tune families of the songs that we are interested in.

We define the set of songs **C** as the set of all songs involved in the experiment, and the subset $\mathbf{S} \subseteq \mathbf{C}$ as the songs in the tune families in which we are interested. We denote the set of features for which we want to compute the criterion with **F**.

Taking the above considerations into account, we define the following criterion:

$$J(\mathbf{C},\mathbf{S},\mathbf{F}) = \frac{tpr(\mathbf{S},\mathbf{F})}{1 + fpr(\mathbf{C},\mathbf{S},\mathbf{F})} = \frac{tp(\mathbf{S},\mathbf{F})}{|\mathbf{S}| + fp(\mathbf{C},\mathbf{S},\mathbf{F})},$$

where $tpr(\mathbf{S},\mathbf{F}) = tp(\mathbf{S},\mathbf{F})/|\mathbf{S}|$ the true positive rate, with tp(S,F) the number of true positives and $|\mathbf{S}|$ the number of songs in **S**, and $fpr(\mathbf{C},\mathbf{S},\mathbf{F}) = fp(\mathbf{C},\mathbf{S},\mathbf{F})/|\mathbf{S}|$ the false positive rate, where $fp(\mathbf{C},\mathbf{S},\mathbf{F})$ is the number of false positives. In this context, true positives are those songs in **S** that have another song from the same tune family as the nearest neighbour, and false positives are those songs *not* in **S** that have a song from a tune family present in **S** as nearest neighbour, but do not belong to that tune family. Since $tpr(\mathbf{S},\mathbf{F}) \in [0,1]$ and $fpr(\mathbf{C},\mathbf{S},\mathbf{F}) \geq 0$, $J(\mathbf{C},\mathbf{S},\mathbf{F}) \in [0,1]$. If **C** does not contain additional songs with respect to **S**, $fp(\mathbf{C},\mathbf{S},\mathbf{F})$ is zero, and thus $J(\mathbf{C},\mathbf{S},\mathbf{F}) = tp(\mathbf{S},\mathbf{F})/|\mathbf{S}|$, which is the nearest neighbour leave-one-out success rate. If, on the contrary, **C** does contain additional songs with respect to **S**, then J is the nearest neighbour leave-one-out success rate for the songs in **S** corrected by the false positives among the additional songs in **C**. The value of the criterion is a lower bound for the classification performance on the songs in **S**. A higher value for J indicates better class separability, since both the classes we are interested in can be separated and there is little or no interference from other classes.

Suppose, as an example, that we are interested in the discriminative power of a certain feature set **F** for all of the 26 tune families (360 songs) of the Annotated Corpus among the entire corpus of 4830 songs. Then, **S** consists of the 360 songs of the Annotated Corpus and **C** consists of all 4830 songs. Suppose that we have a classifier that classifies 90 songs from the Annotated Corpus (**S**) correctly and that also classifies 300 songs that are not in **S** into one of the classes (tune families) that are in **S**. Then the total number of incorrectly classified songs is $270 + 300 = 570$, and thus, the total number of correctly classified songs is 4260. Therefore, the leave-one-out success rate for the whole data set is $4260/4830 = 0.88$. This seems a good result, but it is heavily biased by the asymmetric class sizes: the class 'Other' contains 4470 songs, while the typical size of the classes in **S** is in the order of 10 songs. In our example, only 90 songs in **S** have been correctly classified. Therefore a success rate of $90/360 = 0.25$ would better reflect the performance we are interested in. Still, for the discriminative power of the feature set that was used by the classifier, this is not the right value, since 300 other songs were classified into classes that are in **S**. Therefore we correct the success rate of 0.25 using the definition of the criterion:

$$J(\mathbf{C},\mathbf{S},\mathbf{F}) = \frac{\frac{90}{360}}{1 + \frac{300}{360}} = 0.14.$$

As another example, suppose that we are interested in the discriminative power of a certain feature set **F** for the melodies of tune family called *Ruiter 1* among the other melodies in the Annotated Corpus. Then, **S** consists of the 27 melodies from *Ruiter 1* and **C** consists of the 360 songs of the Annotated Corpus. Suppose, all 27 melodies from *Ruiter 1* are correctly classified as *Ruiter 1* using the nearest-neighbour rule, but also four melodies from other tune families are classified as *Ruiter 1*. Then the criterion value is:

$$J(\mathbf{C},\mathbf{S},\mathbf{F}) = \frac{\frac{27}{27}}{1 + \frac{4}{27}} = 0.87.$$

In our experiments, **S** will either contain the songs from a single tune family or the songs from all 26 tune families in the Annotated Corpus, and **C** will either contain the songs of the Annotated Corpus, or the full corpus of 4830 songs. In all cases, we label the songs in **S** with their respective tune family and the other songs in **C** with 'Other'.

As explained, the main idea of this criterion is to take false positives into account and 'penalize' for those. We defined a false positive as the case in which a song in **C**, but not in **S** has a song from **S** as nearest neighbour. However, in the case that the 'query' song belongs to a tune family of which only one member is present in the corpus, it is not possible to have a song from the same tune family as nearest neighbour, and consequently, in terms of similarity, it might be correct to find a song from **S** as nearest neighbour. Since there are 1460 'single' songs in the large set, this is an effect to be aware of. Still, we label the case that we find a song from **S** as nearest neighbour as a false positive, since it implies that the 'query' song is more similar to a song in **S** than to the 'rest of the world', which affects the discriminability of the tune families in **S**. Thus, with respect to this phenomenon, the criterion value gives a pessimistic estimation of the discriminative power of a feature set.

An implementation-specific advantage of this criterion is the efficiency of computation. We use the implementation of the nearest neighbour classifier as is provided in

the Matlab toolbox PRTools.[3] This toolbox offers a function (`testk`) that computes the leave-one-out success rate for an entire data set by only a few matrix operations instead of computing the error separately for each song and averaging afterwards. Rewriting this function to return the value of our criterion is straightforward. Fast computation of the criterion value is especially important for finding the optimal subset of features, which has a very large solution space.

3.3 Evaluation of individual global features

The main question of this subsection is: which of the single features are discriminative for which tune families? Furthermore, it is interesting to find out whether there are single features that are discriminative for all or many tune families, since such features could possibly be related to basic properties of melodic variability among melodies from oral tradition.

3.3.1 Method

For each of the individual 26 tune families, we compute for each of the 88 features the value of the criterion, both for the small dataset of 360 songs and for the large dataset of 4830 songs. Thus, **S** consists subsequently of the songs of the tune family under consideration, **C** consists either of the 360 songs from the Annotated Corpus or of all 4830 songs, and **F** consists subsequently of each single feature. In all of these cases we have a two-class classification problem. This results in a total of $26 * 88 * 2 = 4576$ different criterion values.

Besides this, we also compute the discriminative power of each single feature for all 26 classes from the Annotated Corpus as a whole, resulting for each feature in a 27-class classification problem. In this case, **S** contains all 360 songs from the Annotated Corpus. Again we do this both for the small and for the large data set. In the former case **C** consists of the 360 songs from the Annotated Corpus and in the latter case of all 4830 songs. Thus, for the small data set, we find the leave-one-out accuracy of the nearest neighbour classifier. The criterion value for the large data set indicates to what extent the annotated songs can still be recognized among thousands of other songs.

3.3.2 Results

The ten highest of the 4576 criterion values for the individual features and tune families are shown in Table 1, along with the tune families for which the features are discriminative. To compare scalability, the criterion value for the large data set is included as well. For all other combinations of features and tune families, the criterion value for the small set is less than 0.5.

In all cases, the discriminative power of the features decreases dramatically for the large data set. Overall, the highest criterion value for the large data set for individual tune families and features is 0.2857 for tune family *Nood* and feature FractionHalfDuration (which is not shown in Table 1). Four out of the eight songs of this tune family have been classified correctly using this single feature, while six songs from other tune families were erroneously classified as *Nood*. Inspection of the melodies shows that the rhythmic ratios of 1:2 and 2:1 are very common in this tune family indeed, as the incipit in Figure 2 shows. Apparently, there are only very few other melodies in the large corpus that show such a high occurrence rate of this rhythmical pattern.

For the separability of all 26 classes, we find that for the small data set the highest value of the criterion is 0.175, which occurs three times, namely for the features Melodic Thirds (14), DurationLineCorrespondence (49), and aminthird (53). For the large data set, the highest criterion value is 0.0217 for the features Amount of Arpeggiation (1) and Size of Melodic Arcs (32). These values are too low to justify further inquiry.

Both for single tune families and for the 26 tune families together, the majority of the cases yield criterion value zero. In these cases, none of the songs has a song from the same tune family as its nearest neighbour for the feature under consideration.

Table 1. The 10 features with the highest criterion value for the small dataset of 360 songs. The criterion value for the full dataset is also shown.

Tune family	Feature	J for small set	J for large set
Herderinnetje	FractionEqualDurations (47)	0.8182	0.0800
Herderinnetje	FractionHalfDuration (46)	0.7692	0.0500
Maagdje	Melodic Octaves (13)	0.7273	0.0256
Maagdje	aoctave (62)	0.7273	0.0299
Meisje	Most Common Melodic Interval Prevalence (17)	0.6875	0.0488
Halewijn 2	Polyrhythms (26)	0.6667	0.1404
Lindeboom	daugfourth (69)	0.6667	0.2667
Lindeboom	Melodic Tritones (15)	0.5000	0.0800
Meisje	aminseventh (60)	0.5000	0.2273
Halewijn 2	Number of Moderate Pulses (21)	0.5000	0.1765

[3]http://www.prtools.org (accessed 1 June 2011).

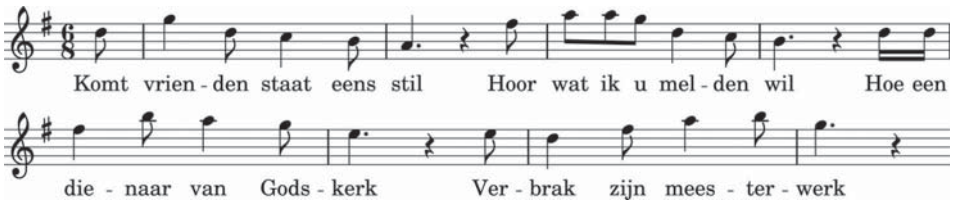

Fig. 2. Beginning of a representative song from the tune family *Nood*.

3.3.3 Conclusion

From these results we conclude that none of the individual features is 'strong' enough to discriminate all 26 tune families. The features that to some extent are discriminative for a single tune family are not discriminative for other tune families. Furthermore, we conclude that the results for the Annotated Corpus are not scalable: the features that are discriminative for tune families within the Annotated Corpus are not discriminative for the same tune families within the large data set. Apparently, there is quite some interference among the tune families concerning the values of individual features. Hence, to model similarity relations between tunes from oral tradition, employment of single features is not sufficient. Higher-dimensional approaches are needed. Therefore, in the next section, we evaluate sets of global features.

3.4 Evaluation of sets of global features

3.4.1 Method

To find sets of features that separate the tune families, we perform forward floating feature selection (Pudil, Novovičivà, & Kittler 1994).[4] Starting with an empty feature subset, this algorithm successively adds or removes one feature in order to optimize the criterion. At the end, the subset yielding the highest criterion value is returned.

Again, we do this both for each individual tune family and for all 26 tune families from the Annotated Corpus together. In the former case **S** consists of the songs from the tune family under consideration and in the latter case **S** consists of all songs from the Annotated Corpus. In both cases, during feature selection, **C** consists of the 360 songs from the Annotated Corpus. Because of the infeasibly long computation time for the large set, we perform the feature selection only for the small data set. To test for scalability, we also compute the criterion value for the large set, using the feature subset that was selected for the small set.

[4]We use the Matlab-implementation of PRTools (http://www.prtools.org, accessed 1 June 2011).

3.4.2 Results

Table 2 shows for each tune family the indices of the selected features, and the value of the criterion for that set for both the small and the large corpus.

For almost all individual tune families the feature subset with the highest criterion value contains less than 10 features, while 62 out of the 88 features are represented in at least one of the selected feature sets. The most common feature is STBFractionStressed (44), which, however, occurs in only six of the 26 subsets. There are two features that occur five times, four features that occur four times, 12 features that occur three times, 17 features that occur two times, and 26 features that occur in only one of the selected subsets.

As in the previous experiments, the global feature approach is not scalable. For most tune families that have a high criterion value for the small data set, the value for the large set is very low. The extreme case is observed for the tune families *Nood* and *Stil*. Apparently, in these cases there is a lot of interference from tune families that are not in the Annotated Corpus. The only tune family for which a moderate performance has been obtained for the large data set is *Meisje*. Most of the features in the specific subset for *Meisje* are related to occurrence rates and sizes of intervals. This relates to the observation that most of the songs in tune family *Meisje* start with an upbeat of an ascending minor seventh or an ascending octave. An example is provided in Figure 3. Given the current results, this seems a relatively unique feature for this tune family among the rest of the corpus.

The selection procedure for separability of all 26 classes returns a feature subset of size 60 for the small data set, with a criterion value of 0.8194. The criterion value for the same feature subset using the large data set is 0.3402, which shows that there is quite some confusion between tune families from the Annotated Corpus and tune families in the rest of the large corpus, in the sense that a relatively high number of melodies from other tune families have a member from the Annotated Corpus as nearest neighbour.

Figure 4 shows the criterion values for selected optimal feature subsets of increasing size, for subsets of size 1 to 30. For feature subsets with more than around nine features, larger feature subsets only result in marginal improvement of the criterion value. The biggest

Table 2. The selected feature subset with the highest criterion value for the small dataset of 360 songs. The criterion value for the full dataset is also shown. Only for tune family *Ruiter 2*, more than 10 features are selected. For this tune family only the first 10 features are shown.

Tune family	Selected feature subset	J for small set	J for large set
Heer	58 59 60 84	0.3810	0.0625
Jonkheer	4 14 20 37 38 60	0.8333	0.0769
Ruiter 2	1 6 15 28 35 44 59 67 78 86	0.7778	0.1154
Maagdje	3 13	0.8000	0
Dochtertje	37 59 71 79	0.4118	0.0962
Lindeboom	8 23 69	0.8889	0.2414
Zoeteliefjes	3 6 44 54 78 87	1.0000	0.2692
Ruiter 1	24 38 53 70 82 84	0.6667	0.3158
Herderinnetje	47 87	1.0000	0.1000
Koopman	68 70	0.6842	0.0795
Meisje	2 3 4 5 13 16 17 41 60	1.0000	0.6471
Vrouwtje	9 44 48 49 84	0.9167	0.1739
Femmes	45 51 59 81	0.7143	0.0800
Halewijn 2	9 26	0.7273	0.2162
Halewijn 4	22 28 35 37 87	0.6667	0.1905
Stavoren	15 33 45 84	0.7778	0.0769
Zomerdag	27 39 66 67	0.7895	0.0909
Driekoningenavond	9 44 57 59 68 87	0.8462	0.2500
Stad	7 13 23 36 55	1.0000	0.4000
Stil	4 23 24 34 71 75	1.0000	0
Schipper	7 12 17 47 54 58 70	0.9333	0.4000
Nood	10 17 27 46 49	1.0000	0
Soldaat	6 16 49 71	0.6111	0.2041
Bruidje	29 44 49 84	1.0000	0.0556
Verre	8 25 35 42	0.7647	0.0400
Boom	4 10 24 27 39 42 44 66 70	0.7895	0.2963

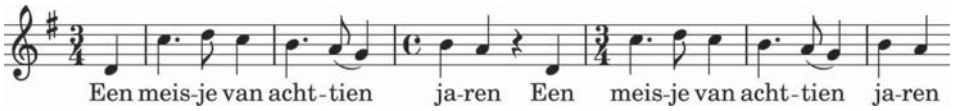

Fig. 3. Beginning of a representative song from the tune family *Meisje*.

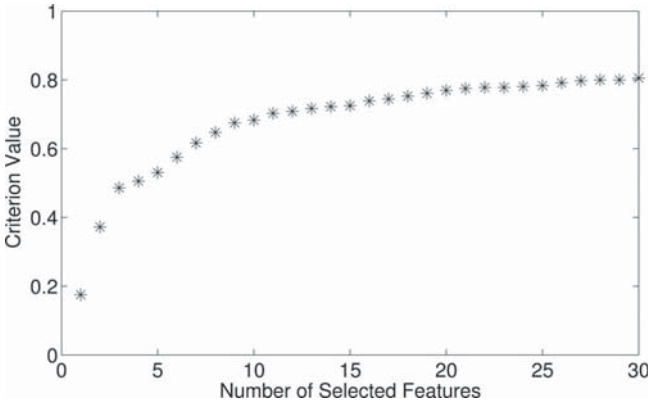

Fig. 4. Criterion values for feature subsets of various sizes for the Annotated Corpus.

improvements are reached for subsets of one, two and three features. The selected subset of three features contains FractionStressed (44), dminthird (66) and numlines (87). Interestingly, these three features are aspects of different dimensions of melodic similarity: meter, pitch and form.

3.4.3 Conclusion

Although discriminative subsets of features can be found for the Annotated Corpus, we conclude from these results that no feature subset can be found that is discriminative for all tune families in the large corpus. Nevertheless, some tune families can be better distinguished than others. Furthermore, we observe a large

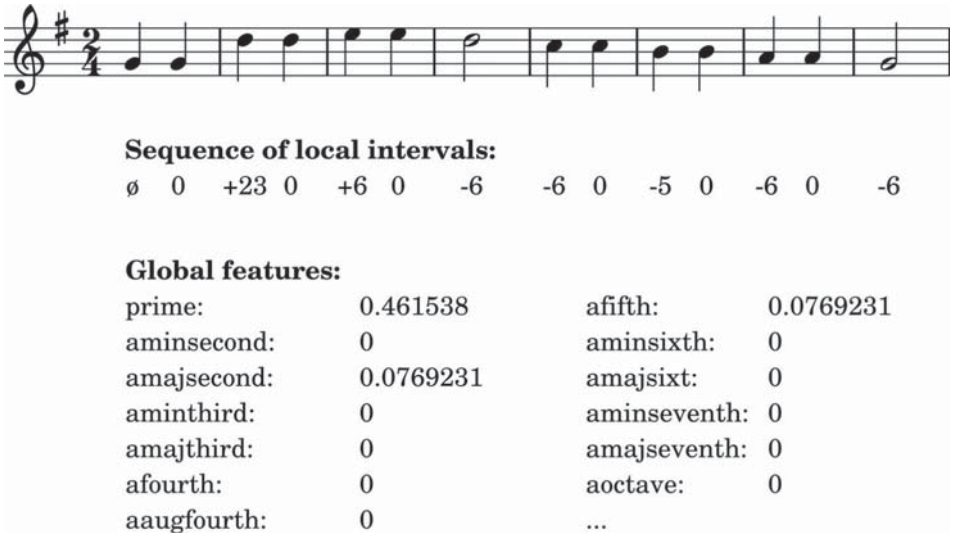

Fig. 5. Representation of a melody as sequence of local melodic intervals (directly below the notes, in base-40 encoding) and as set of several global features from the set of Jesser (inventory of ascending melodic intervals).

diversity among the specific feature subsets that have been selected for the individual tune families. There is not even a single feature that is present in a substantial number of selected subsets. It seems that, taking a global feature approach, one has to design separate models for each of the individual tune families.

In the next section, we incorporate local features in our investigation.

4. Comparison of global and local features

In this section we compare the global feature approach that was investigated in the previous section, with a local approach, in which a song is represented as a sequence of local feature values. In the local approach, the time-order of melodic events is preserved and local comparison of melodies is possible. In all our comparisons, we choose local and global features that are directly related.

We perform the comparisons between the local and the global approach separately for pitch related features and rhythm related features. The respective local features are pitch intervals and duration ratios. In the first case, a note is represented by the melodic interval with the preceding note. In the second case, the note is represented by its duration divided by the duration of the preceding note. As an illustration, Figure 5 shows the representation of the same melody both as sequence of local pitch intervals in base-40 encoding (see Hewlett, 1992) and as vector of (several) global features from Jesser's interval inventory. This example shows a direct relation between the local and global features.

In the global approach, we compute the distance between two melodies as the Euclidian distance between the two corresponding vectors of feature values. Since the same set of global features is used for both melodies, these two vectors have the same size and correspond directly to each other. For the local approach we need another way to compute the distance between two melodies. In most cases, the sequences of local feature values of two melodies differ in length. These sequences are not directly comparable. Therefore, we use sequence alignment to compute the distance between the two sequences of local feature values. We take the extent to which an alignment can be computed as a measure of the similarity between the sequences. This will be further explained in Section 4.2.2. As soon as the distances between the songs are available, we can apply the nearest neighbour rule, and we can compute the value for the feature evaluation criterion that was defined in Section 3.2. This allows us to compare the classification performances of the global and local approaches.

4.1 Comparisons

We perform five comparisons, in which the melodies are represented by different kinds of features:

(1) Global: vector of interval features; Local: sequence of local intervals (Section 4.3.1).
(2) Global: vector of interval features and derived pitch features; Local: sequence of local intervals (Section 4.3.2).
(3) Global: vector of duration-ratio features; Local: sequence of local duration-ratios (Section 4.3.3).
(4) Global: vector of duration-ratio features; and derived global rhythmic features; Local: sequence of local duration-ratios (Section 4.3.4).
(5) Global: All global features; Local: sequence of local pitch, metric, and structure features (Section 4.3.5).

For comparisons 1 and 3, the local and global features correspond directly, while comparisons 2 and 4 include less closely related global features. The motivation for these five comparisons is to have both 'fair' comparisons (1 and 3) and comparisons in which the full available potential of the methods is used (2, 4, and 5). Table 3 shows the composition of each subset of global features, along with the global features that were selected by the feature selection algorithm.

4.2 Method

4.2.1 Method for global features

For each of the five sets of global features we perform forward floating feature selection (Pudil et al., 1994) in the same way as is described in Section 3.4. As a measure for the classification performance of the global feature subset under consideration, we compute the value of the criterion that was presented in Section 3.2. Just as before, we do this both for the Annotated Corpus and for the large corpus of 4830 melodies. In the former case, **C** consists of the melodies of the Annotated Corpus, and in the latter case, **C** consists of all 4830 melodies. In both cases, **S** consists of the melodies of the Annotated Corpus. The results of the feature selection are shown in Table 3.

4.2.2 Method for local features

For the comparison of two sequences of local melodic events, we use the Needleman–Wunsch–Gotoh alignment algorithm (Needleman & Wunsch, 1970; Gotoh, 1982). Needleman and Wunsch (1970) proposed an algorithm that finds an optimal alignment of two entire sequences of symbols. The quality of an alignment is measured by the alignment score, which is the sum of the alignment scores of the individual symbols. We interpret this alignment score as a similarity measure for melodies. Because these similarity values can easily be converted into distances, we can apply the nearest-neighbour classification rule, and, therefore, we can compute the value of our criterion, which allows for direct comparison between the local and global approach.

The Needleman–Wunsch algorithm takes two sequences of symbols, which we denote with $\mathbf{x} : x_1, \ldots, x_i, \ldots, x_n$, and $\mathbf{y} : y_1, \ldots, y_j, \ldots, y_m$. Symbol x_i can either be aligned with a symbol from sequence **y** or with a gap. Both operations have a score, respectively the similarity score and the gap score. The gap score is mostly expressed as penalty, i.e. a negative score. The optimal alignment and its score are found by filling a matrix D recursively according to:

$$D(i,j) = \max \begin{cases} D(i-1,j-1) + S(x_i, y_j), \\ D(i-1,j) - \gamma, \\ D(i,j-1) - \gamma, \end{cases} \quad (1)$$

where $S(x_i, y_j)$ is the similarity scoring function, γ is the gap penalty, $D(0, 0) = 0$, $D(i, 0) = -i\gamma$, and $D(0, j) = -j\gamma$. $D(i, j)$ contains the score of the optimal alignment up to x_i and y_j and therefore, $D(m, n)$ contains the score of the optimal alignment of the complete sequences. If needed, the alignment itself can be obtained by tracing back from $D(m, n)$ to $D(0, 0)$; the algorithm has both time and space complexity $O(nm)$.

The similarity scoring function, $S(x_i, y_j)$ reflects to what extent we want the symbols x_i and y_j to be aligned. It allows for incorporation of domain knowledge, which in our case is musical knowledge. We will define various functions in Section 4.3. All functions return values in the range $[-1, 1]$.

In our modelling, we use an extension of the algorithm proposed by Gotoh (1982), which employs an affine gap penalty function without loss of efficiency. In this approach, the extension of a gap gets a lower penalty than its opening. We take a gap opening penalty of 0.8 and a gap extension penalty of 0.2. Thus, the opening of a gap is preferred over a bad match, and the continuation of a gap is relatively 'cheap'.

Since songs are notated in different keys, the similarity measure should be transposition invariant. To achieve

Table 3. For each comparison, this table shows all involved global features and the subset of those features that has been selected by the floating feature selection algorithm. The indices refer to the list in Appendix A.2.

Comparison	Global features	Selected global features
1. Interval features	50–80	55 68 50 53 59 66 51 70 54 60 67 69 65 64 62 75 56 71
2. Pitch features	1–3 5–8 10–20 24 25 27–33 37–43 48 50–80 88	3 7 8 10 13 24 25 28 29 33 38 39 40 43 48 52 53 54 55 56 58 59 60 62 65 66 68 69 70 75 77 78 80
3. Duration-ratio features	45–47 81–86	45 47 81–86
4. Rhythm features	4 9 21–23 34 35 36 44–49 81–86	4 35 36 44 46 47 49 82 84
5. All features	1–88	2–15 20 24–31 35–46 48–51 53–55 54 55 57–59 64–68 70–72 78 81–84 87 88

this, a pitch histogram for both melodies is created that indicates for each pitch the total duration of all occurrences of that pitch in the entire song. Then the shift at which the normalized histograms have maximal intersection is computed. This can be interpreted as the interval with which the one melody should be transposed in order to compare it to the other. Applying this shift before computing the similarity score of two symbols ensures transposition invariance.

Since the score of an alignment depends on the length of the sequences, normalization is needed to compare different alignment scores. The alignment of two long songs results in a much higher score than the alignment of two short songs. Therefore, we divide the alignment score by the length of the shortest sequence. Thus, an exact match results in score 1, which is the maximal score. The normalized alignment score can be used as a measure for the similarity of two songs. The scores are converted into distances by taking one minus the normalized score.

Our approach is not unlike the approach of Hanna, Ferraro and Robine (2007), but more specifically aimed at folk song melodies. Our approach has already been shown to perform well for the collection under consideration (both for the small and the large corpus), as reported in van Kranenburg (2010, Chapter 6).

With the distances that are computed by the alignment algorithm, we compute the criterion value. We do this both for the Annotated Corpus only, and for the full set of 4830 melodies.

4.3 Features for the comparisons

In the following, we describe the five subsets of global features as introduced in Section 4.1, along with the local features and the specific similarity scoring functions for each of the five comparisons.

4.3.1 Comparison 1: Interval features

The subset of global features that is directly related to the intervals between the consecutive notes consists of the interval features as defined by Jesser: features 50–80 (see Appendix A.2). Each of these features measures the occurrence-rate of a certain interval.

The subset of interval features that is most discriminative for the Annotated Corpus as obtained by the floating selection algorithm is shown in Table 3.

For the local approach we take sequences of intervals between the successive pitches. Therefore, in this case, we have a perfect correspondence between the information used for the global features and for the alignment. We represent pitches in base-40 encoding (see Hewlett, 1992). As the similarity scoring function for the alignment algorithm we use:

$$S_{\text{interval}}(x_i, y_j) = \begin{cases} 1 & \text{if } melint\,(x_i) = melint\,(y_j), \\ -1 & \text{if } melint(x_i) \neq melint\,(y_j), \end{cases} \quad (2)$$

where $melint(x_i) = p(x_i) - p(x_{i-1})$ is the melodic interval between x_{i-1} and x_i for $i > 1$, in which $p(x)$ is the pitch of symbol x. $S_{\text{interval}}(x_0, y_j) = S_{\text{interval}}(x_i, y_0) = 1$, which allows for alignment of the first symbols of the respective sequences.

4.3.2 Comparison 2: Pitch features

The subset of global pitch features consists of all interval features used in comparison 1 along with a number of higher-level pitch-based features from the sets of Jesser, Steinbeck and jSymbolic as shown in Table 3.

For the local approach, again, we take sequences of intervals between the successive pitches, using the same similarity scoring function S_{interval}.

4.3.3 Comparison 3: Duration-ratio features

The subset of duration-ratio features consists of features that relate to the relative lengths of the notes (see Table 3). Jesser's features relate the duration of the note to the shortest duration in the melody, while Steinbeck's features relate the duration of a note to the duration of the preceding note.

For the local approach we take the sequence of duration-ratios. The duration-ratio of a note is the duration of the note as a fraction of the duration of the preceding note. The similarity scoring function is defined as:

$$S_{\text{dratio}}(x_i, y_j) = \begin{cases} 1 & \text{if } dr(x_i) = dr(y_j) \\ -1 & \text{if } dr(x_i) \neq dr(y_j) \end{cases}, \quad (3)$$

where $dr(x_i) = d(x_i)/d(x_{i-1})$, the ratio between the durations $d(x_i)$ and $d(x_{i-1})$ of x_i and x_{i-1} for $i > 1$, and $S_{\text{dratio}}(x_0, y_j) = S_{\text{dratio}}(x_i, y_0) = 1$.

4.3.4 Comparison 4: Rhythmic features

The subset of rhythmic features contains the duration-ratio features along with other, higher-level, rhythmic features as shown in Table 3.

For the local approach, again, we take the sequence of duration-ratios using the same scoring function S_{dratio}.

4.3.5 Comparison 5: All features

For the comparison using all features, we take the full available potential of both approaches.

For the global approach, we take all global features as well as the optimal subset that was found by the floating selection algorithm among the full set of global features.

For the local approach, we use three types of features: pitch, metric weight and phrase-position. The pitch is the base40-representation of the pitch of the note. The metric weight is obtained using the Inner Metric Analysis (IMA) (see Volk, 2008), which computes a metric weight for each note solely based on the onset times of the notes instead of the notated meter. The phrase-position of a note is the scaled onset time of the note within a melodic phrase, such that the onset time of the first note of the phrase gets value 0 and the onset time of the last note of the phrase gets value 1. Thus, a melody is represented as a sequence of triplets. For each note, we have three feature values, which we can use in the similarity scoring function.

For each of the three local features, we define a separate similarity scoring function, which we, in the end, combine to get one similarity score to be used in the alignment algorithm. The definitions of these similarity scoring functions will now be presented.

The pitch-based similarity scoring function measures the difference in pitch height between a note in the first song and a note in the second song. The larger the difference, the lower the resulting similarity score.

$$S_{\text{pitchb}}(x_i, y_j) = \begin{cases} 1 - \frac{int(x_i, y_j)}{23} & \text{if } int(x_i, y_j) \leq 23, \\ -1 & \text{otherwise,} \end{cases} \quad (4)$$

in which $int(x, y) = |p(x) - p(y)|$ mod 40. A perfect fifth has value 23 in base-40 encoding. Thus, all intervals up to a perfect fifth get a positive similarity score and all larger intervals are considered a bad match.

We define the scoring function that uses the metric weights of the notes as computed by IMA as follows:

$$S_{\text{ima}}(x_i, y_j) = 1 - 2\big|w(x_i) - w(y_j)\big|. \quad (5)$$

Here, $w(x)$ denotes the metric weight of note x, scaled into the interval [0, 1]. For scaling, all weights are divided by the greatest weight in the song. For the free parameters in the IMA-algorithm (p and l) we take the values that are mostly used: $p = 2$, $l = 2$ (e.g. in Volk, 2008).

To use the information of phrase boundaries that is present in our data set, we use a scoring function based on the horizontal position of the notes within the phrase:

$$S_{\text{phrpos}}(x_i, y_j) = 1 - 2\big|phr(x_i) - phr(y_j)\big|, \quad (6)$$

in which $phr(x) \in [0,1]$ is a linear mapping of the horizontal position of symbol x between the onset of the first note and the onset of the last note of the phrase into the interval [0,1].

To get one similarity score, these three similarity scores are combined. We want alignments in which the aligned symbols are similar in all dimensions. Therefore, we multiply the individual scores:

$$S'_{\text{combination}}(x_i, y_j) = \prod_{k=1}^{n} S'_k(x_i, y_j), \quad (7)$$

in which $S'_k(x_i, y_j) = \frac{1}{2}(S_k(x_i, y_j) + 1)$, which is $S_k(x_i, y_j)$ scaled into the interval [0,1]. The final score $S_{\text{combination}}$ is $S'_{\text{combination}}$ scaled into $[-1, 1]$ back again. This scoring

Table 4. Criterion values for the various configurations, both for the small set consisting of the Annotated Corpus and for the large set consisting of the Annotated Corpus among 4470 other melodies.

Comparison		Features	J_{small}	J_{large}
1	global	interval features	0.52	0.20
		selected interval features	0.59	0.20
	local	interval sequence	**0.92**	**0.60**
2	global	pitch features	0.67	0.28
		selected pitch features	0.74	0.29
	local	interval sequence	**0.92**	**0.60**
3	global	duration-ratio features	0.38	0.09
		selected duration-ratio features	0.39	0.08
	local	duration-ratio sequence	**0.74**	**0.33**
4	global	rhythm features	0.49	0.12
		selected rhythm features	0.55	0.13
	local	duration-ratio sequence	**0.74**	**0.33**
5	global	all features	0.74	0.32
		selected features	0.82	0.34
	local	pitchband, IMA, phrasepos.	**0.99**	**0.73**

function was shown to be very successful in van Kranenburg (2010, Chapter 6).

4.4 Results and conclusions

Table 4 shows the criterion values for each of the five comparisons that were presented in Section 4.1, for both the local and the global approach.

In all comparisons, the performance for the large data set is considerably lower than for the small data set, both for the global and local approaches. Apparently, none of the tune families from the Annotated Corpus is completely isolated from the additional melodies in the large data set.

For the small data set, the selected subsets of global features yield better performance than the full feature sets. This is the case in all five comparisons. However, this improvement is as good as absent for the large data set.

The pitch-related features from comparisons 1 and 2 lead to better performances than the rhythm-related features from comparisons 3 and 4, both in the local and global approaches. Nevertheless, the criterion value for the alignment of duration-ratio sequences for the large corpus (0.33) can be considered quite high concerning the size of the total corpus: even among several thousands of other melodies, the rhythm of the melodies seems to provide enough information to classify a substantial part of the 360 annotated melodies correctly. But, purely rhythmic features are clearly not suitable to provide a basis for a classification method that is employable in a folk song research context.

The only case in which the global approach shows success is on the small corpus, when the optimal subset of all features is used. However, the performance drops considerably for the large corpus.

In all cases, the results show that the alignment approach is both more accurate and better scalable. The reported criterion values in comparison 5 (0.99 and 0.73 for the small and large sets respectively) indicate that this approach is useful for classification of melodies from oral tradition.

5. General conclusions

In this paper, we studied two approaches to classify melodies from Dutch oral tradition according to tune family membership. In the global approach, a melody is represented as a vector of values of global features, each of which summarizes an aspect of the entire melody. The classification is performed according to the nearest-neighbour rule using the Euclidian distance between vectors of global feature values. In the local approach, a melody is represented as sequence of local melodic events (such as pitches and duration ratios), and the classification is also performed according to the nearest-neighbour rule using the score of the alignment of two sequences as a similarity measure.

First, we evaluated the discriminative power of a large number of global melodic features for a set of melodies belonging to various tune families. We evaluated both individual global features and sets of global features. Next, we compared the classification performances of the global approach with the performances of a local approach, in which alignment of sequences of local features was used to determine the similarity between melodies.

We performed all tests with a small data set of 360 melodies, as well as with a large data set of 4830 melodies, in which the 360 melodies of the small data set are embedded. Thus, we can evaluate the classification performances with respect to scalability.

From the evaluation of individual features in Section 3.3, we conclude that there is no single global feature that is discriminative for all tune families. Only for the small data set, a few features have discriminative power for specific tune families.

The contents of the selected feature subsets in Section 3.4 show for each tune family which subset of global features is optimally discriminative. The selected sets differ to a large extent. The most common feature in all selected subsets has been selected for only 6 out of 26 tune families. This indicates that, concerning global features, each tune family is distinct from the rest of the corpus in a specific way. Therefore, it seems necessary to design separate models for the classification of the various tune families. In general, even the discovered optimal subsets do not lead to convincing classification performance for the large data set.

In all experiments with global features, success rates decrease considerably for the larger data set. This indicates that the global feature approach for recognition of melodies can only be taken if the data set contains a small set of tune families. This confirms the trend that was observed by Steinbeck (1982), who obtained meaningful results for a set of 35 melodies, but not for a set of 500 melodies.

In the comparisons we made in Section 4, the local, alignment-based, approach outperformed the global approach in all cases. The alignment approach is also better scalable: for the large data set, reasonable classification results could be obtained. This result is in accordance with the conclusions of both Jesser (1991) and Hillewaere et al. (2009). Both were able to get better retrieval or better classification results using local events and local features, rather than global features. However, the classification task in the current study is more difficult. Instead of classifying a heterogeneous collection of songs by country, as Hillewaere et al. did, we classify tune families within a single tradition. The Essen collection used by Jesser was sampled from a large variety of sources from different countries and regions, periods and types of sources and is therefore also far more heterogeneous than our data set.

In a previous study (Volk & Van Kranenburg, 2012) we found for the same corpus that recurring, characteristic motifs are important for recognizing melodies. There are many kinds of motifs: a rhythmic figure, an uncommon interval, a leap, a syncopation, and so on. The current results suggest that it is not possible to grasp the discriminative power of motifs in only a few global features. This is an important shortcoming of the approach based on global features. Therefore, for the next steps in the research on automatic classification of melodies from oral tradition, a local approach is indispensable. Unlike early adaptors of computational methods, such as Bronson (1949), Steinbeck (1982), and Jesser (1991), we are currently able to employ computationally much more demanding methods due to the advances in computer hardware and computer science during the last few decades, which allows for more detailed representations of music and for comparisons between melodies that involve more complex algorithms. Another next step would be to explore a hybrid approach in which both local and global elements of melodies are used.

As a general conclusion we state that the global features that are known from recent computational studies are of limited use for the retrieval of related folk song melodies from a large database. Good results are only obtained for a few tune families within a small corpus. Using the local approach, we obtained good results for a large corpus, as well. Therefore, to design models of relations between melodies from oral culture, local melodic phenomena are indispensable. Given the fact that most of the Dutch melodies are in a Western tonal idiom, we expect this conclusion to apply to Western folk songs in general. Nonetheless, it would be a relevant next step to involve melodies from other traditions as well.

This conclusion confirms choices that were made in early Folk Song Research. Among others, Krohn (1903), Bartók and Kodály (see Suchoff, 1981), and Suppan and Stief (1976) all used sequences of local melodic events to order their respective collections of melodies. Although from a computational perspective the use of global features has advantages, the local approach must be preferred from a musicological point of view.

The local approach has not exhaustively been explored in the current study. We confined ourselves to a representation of melodies as a series of notes. A representation of a melody as a sequence of motifs seems a promising next step. Therefore, in future work, we will focus on similarity relations between melodies that are based on shared melodic motifs.

Acknowledgements

This work was funded by the Netherlands Organisation for Scientific Research. It was carried out within the WITCHCRAFT project (NWO 640-003-501), which is part of the Continuous Access to Cultural Heritage (CATCH) program, and in the Tunes & Tales project, which is part of the Computational Humanities program of the Royal Netherlands Academy of Arts and Sciences. Anja Volk is supported by the Netherlands Organisation for Scientific Research, NWO-VIDI grant 276-35-001. We thank Marcelo E. Rodriguez López (Utrecht University) for careful proof reading and suggestions.

The Annotated Corpus can be obtained from the authors. The implementation of the alignment algorithm is available as C++-library from: http://libmusical.sourceforge.net.

References

Bayard, S. (1950). Prolegomena to a study of the principal melodic families of British–American folk song. *Journal of American Folklore*, *63*(247), 1–44.

Bohak, C., & Marolt, M. (2009). Calculating similarity of folk song variants with melody-based features. In *Proceedings of the 10th International Conference on Music Information Retrieval (ISMIR 2009)*, Kobe, Japan, pp. 597–601.

Bronson, B.H. (1949). Mechanical help in the study of folk song. *The Journal of American Folklore*, *62*(244), 81–86.

Bronson, B.H. (1950). Some observations about melodic variation in British-American folk tunes. *Journal of the American Musicological Society*, *3*, 120–134.

Cowdery, J. (1984). A fresh look at the concept of Tune Family. *Ethnomusicology*, *28*(3), 495–504.

Downie, J.S. (2003). Music information retrieval. *Annual Review of Information Science and Technology*, *37*, 295–340.

Eerola, T., Järvinen, T., Louhivuori, J., & Toiviainen, P. (2001). Statistical features and perceived similarity of folk melodies. *Music Perception*, *18*(3), 275–296.

Gotoh, O. (1982). An improved algorithm for matching biological sequences. *Journal of Molecular Biology*, *162*, 705–708.

Grijp, L.P. (2008). Introduction. In L.P. Grijp & I. van Beersum (Eds.), *Under the Green Linden — 163 Dutch Ballads from the Oral Tradition* (pp. 18–27). Amsterdam: Meertens Institute + Music & Words.

Hanna, P., Ferraro, P., & Robine, M. (2007). On optimizing the editing algorithms for evaluating similarity between monophonic musical sequences. *Journal of New Music Research*, *36*(4), 267–279.

Hewlett, W.B. (1992). A base-40 number-line representation of musical pitch. *Musikometrika*, *4*, 1–14.

Hillewaere, R., Manderick, B., & Conklin, D. (2009). Global feature versus event models for folk song classification. In *Proceedings of the 10th International Conference on Music Information Retrieval (ISMIR 2009)*, Kobe, Japan, pp. 729–733.

Jesser, B. (1991). *Interaktive Melodieanalyse* (Vol. 12). Bern: Peter Lang.

Juhász, Z. (2009). Automatic segmentation and comparative study of motives in eleven folk song collections using self-organizing maps and multidimensional mapping. *Journal of New Music Research*, *38*(1), 71–85.

Krohn, I. (1903). Welche ist die beste Methode, um Volks- und volksmässige Lieder nach ihrer melodischen (nicht textlichen) Beschaffenheit lexikalisch zu ordnen? *Sammelbände der internationalen Musikgesellschaft*, *4*(4), 643–660.

McKay, C. (2004). *Automatic genre classification of MIDI recordings* (Master's thesis). McGill University, Canada.

Needleman, S.B., & Wunsch, C.D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, *48*(3), 443–453.

Nettl, B. (2005). *The Study of Ethnomusicology: Thirty-one Issues and Concepts* (2nd ed.). Urbana and Chicago: University of Illinois Press.

Pudil, P., Novovičivà, J., & Kittler, J. (1994). Floating search methods in feature selection. *Pattern Recognition Letters*, *15*(11), 1119–1125.

Schaffrath, H. (Ed.). (1995). *The Essen Folksong Collection*. Stanford, CA: Center for Computer Assisted Research in the Humanities.

Scheurleer, D. (1900). Preisfrage. *Zeitschrift der Internationalen Musikgesellschaft*, *1*(7), 219–220.

Steinbeck, W. (1982). *Struktur und Ähnlichkeit—Methoden automatischer Melodieanalyse* (Vol. XXV). Kassel: Bärenreiter.

Suchoff, B. (1981). Preface. In *The Hungarian Folk Song* (pp. ix–lv). Albany: State University of New York Press.

Suppan, W., & Stief, W. (Eds.). (1976). *Melodietypen des Deutschen Volksgesanges*. Tutzing: Hans Schneider.

Toiviainen, P. (Ed.). (2007). *Discussion Forum 4A. Similarity Perception in Listening to Music*. Special issue of *Musicae Scientiae*. Jyväskylä: European Society for the Cognitive Sciences of Music.

Toiviainen, P. (Ed.). (2009). *Discussion Forum 4B. Musical Similarity*. Special issue of *Musicae Scientiae*. Jyväskylä: European Society for the Cognitive Sciences of Music.

van Kranenburg, P. (2010). *A computational approach to content-based retrieval of folk song melodies* (PhD thesis). Utrecht University, Utrecht.

Volk, A. (2008). Persistence and change: Local and global components of metre induction using inner metric analysis. *Journal of Mathematics and Music*, *2*(2), 99–115.

Volk, A., & van Kranenburg, P. (2012). Melodic similarity among folk songs: An annotation study on similarity-based categorization in music. *Musicae Scientiae*. doi:10.1177/1029864912448329

Webb, A. (2002). *Statistical Pattern Recognition* (2nd ed.). New York: Wiley.

Wiora, W. (1941). Systematik der musikalischen Erscheinungen des Umsingens. *Jahrbuch für Volksliedforschung*, *7*, 128–195.

Appendix A

A.1 Tune families in the annotated corpus

Tune family (short)	Tune family (long)	Size
Heer	Daar ging een heer 1	16
Jonkheer	Daar reed een jonkheer 1	12
Ruiter 2	Daar was laatstmaal een ruiter 2	17
Maagdje	Daar zou er een maagdje vroeg opstaan 2	10
Dochtertje	Een Soudaan had een dochtertje 1	13
Lindeboom	Een lindeboom stond in het dal 1	9
Zoeteliefjes	En er waren eens twee zoeteliefjes	16
Ruiter 1	Er reed er eens een ruiter 1	27
Herderinnetje	Er was een herderinnetje 1	11
Koopman	Er was een koopman rijk en machtig	17
Meisje	Er was een meisje van zestien jaren 1	15
Vrouwtje	Er woonde een vrouwtje al over het bos	12
Femmes	Femmes voulez vous éprouver	13
Halewijn 2	Heer Halewijn 2	11
Halewijn 4	Heer Halewijn 4	11
Stavoren	Het vrouwtje van Stavoren 1	8
Zomerdag	Het was laatst op een zomerdag	17
Driekoningenavond	Het was op een driekoningenavond 1	12
Stad	Ik kwam laatst eens in de stad	18
Stil	Kom laat ons nu zo stil niet zijn 1	11
Schipper	Lieve schipper vaar me over 1	15
Nood	O God ik leef in nood	8
Soldaat	Soldaat kwam uit de oorlog	17
Bruidje	Vaarwel bruidje schoon	11
Verre	Wat zag ik daar van verre 1	15
Boom	Zolang de boom zal bloeien 1	18

A.2 The set of global features

The following features from the feature set of McKay (2004) are included in the set of global features that is used in this thesis.

Index	Feature	Description as given by McKay (2004)
1	Amount of Arpeggiation	Fraction of horizontal intervals that are repeated notes, minor thirds, major thirds, perfect fifths, minor sevenths, major sevenths, octaves, minor tenths or major tenths.
2	Average Melodic Interval	Average melodic interval (in semi-tones).
3	Chromatic Motion	Fraction of melodic intervals corresponding to a semi-tone.
4	Combined Strength of Two Strongest Rhythmic Pulses	The sum of the frequencies of the two beat bins of the peaks with the highest frequencies.
5	Direction of Motion	Fraction of melodic intervals that are rising rather than falling.
6	Distance Between Most Common Melodic Intervals	Absolute value of the difference between the most common melodic interval and the second most common melodic interval.
7	Dominant Spread	Largest number of consecutive pitch classes separated by perfect 5ths that accounted for at least 9% each of the notes.
8	Duration of Melodic Arcs	Average number of notes that separate melodic peaks and troughs in any channel.
9	Harmonicity of Two Strongest Rhythmic Pulses	The bin label of the higher (in terms of bin label) of the two beat bins of the peaks with the highest frequency divided by the bin label of the lower.
10	Interval Between Strongest Pitch Classes	Absolute value of the difference between the pitch classes of the two most common MIDI pitch classes.
11	Interval Between Strongest Pitches	Absolute value of the difference between the pitches of the two most common MIDI pitches.
12	Melodic Fifths	Fraction of melodic intervals that are perfect fifths.
13	Melodic Octaves	Fraction of melodic intervals that are octaves.
14	Melodic Thirds	Fraction of melodic intervals that are major or minor thirds.
15	Melodic Tritones	Fraction of melodic intervals that are tritones.
16	Most Common Melodic Interval	Melodic interval with the highest frequency.
17	Most Common Melodic Interval Prevalence	Fraction of melodic intervals that belong to the most common interval.
18	Most Common Pitch Class Prevalence	Fraction of Note Ons corresponding to the most common pitch class.
19	Number of Common Melodic Intervals	Number of melodic intervals that represent at least 9% of all melodic intervals.
20	Number of Common Pitches	Number of pitches that account individually for at least 9% of all notes.
21	Number of Moderate Pulses	Number of beat peaks with normalized frequencies over 0.01.
22	Number of Relatively Strong Pulses	Number of beat peaks with frequencies at least 30% as high as the frequency of the bin with the highest frequency.
23	Number of Strong Pulses	Number of beat peaks with normalized frequencies over 0.1.
24	Pitch Class Variety	Number of pitch classes used at least once.
25	Pitch Variety	Number of pitches used at least once.
26	Polyrhythms	Number of beat peaks with frequencies at least 30% of the highest frequency whose bin labels are not integer multiples or factors (using only multipliers of 1, 2, 3, 4, 6 and 8) (with an accepted error of $+/-3$ bins) of the bin label of the peak with the highest frequency. This number is then divided by the total number of beat bins with frequencies over 30% of the highest frequency.
27	Range	Difference between highest and lowest pitches.

(*continued*)

Appendix A.2 (*Continued*).

Index	Feature	Description as given by McKay (2004)
28	Relative Strength of Most Common Intervals	Fraction of melodic intervals that belong to the second most common interval divided by the fraction of melodic intervals belonging to the most common interval.
29	Relative Strength of Top Pitch Classes	The frequency of the 2nd most common pitch class divided by the frequency of the most common pitch class.
30	Relative Strength of Top Pitches	The frequency of the 2nd most common pitch divided by the frequency of the most common pitch.
31	Repeated Notes	Fraction of notes that are repeated melodically.
32	Size of Melodic Arcs	Average melodic interval separating the top note of melodic peaks and the bottom note of melodic troughs.
33	Stepwise Motion	Fraction of melodic intervals that corresponded to a minor or major second.
34	Strength of Second Strongest Rhythmic Pulse	Frequency of the beat bin of the peak with the second highest frequency.
35	Strength of Strongest Rhythmic Pulse	Frequency of the beat bin with the highest frequency.
36	Strength Ratio of Two Strongest Rhythmic Pulses	The frequency of the higher (in terms of frequency) of the two beat bins corresponding to the peaks with the highest frequency divided by the frequency of the lower.
37	Strong Tonal Centers	Number of peaks in the fifths pitch histogram that each account for at least 9% of all Note Ons.

The following features from the feature set of Steinbeck are included in the set of global features that is used in this thesis.

Index	Feature	Description (page numbers refer to Steinbeck, 1982)
38	StdPitch	Standard deviation of the pitch (p. 156ff).
39	Ambitus	Difference between the highest and lowest pitch in the melody (p. 155).
40	MeanInterval	Mean of the size of the intervals. The intervals between the phrases are not taken into account (p. 165ff).
41	StdInterval	Standard Deviation of the size of the intervals (p. 165ff).
42	ChangingDirection	The fraction of the intervals that cause a change of direction (p. 149f).
43	MeanSteepness	The steepness is the deviation in pitch between two turning points divided by the duration. This feature is the mean of these steepnesses (p. 173ff).
44	FractionStressed	The sum of durations that start on a stressed beat as fraction of the total duration (p. 178ff).
45	FractionDottedDuration	The fraction of transitions between pitches that has duration quotient 3:1 (p. 152ff).
46	FractionHalfDuration	The fraction of transitions between pitches that has duration quotient 2:1 or 1:2 (p. 152ff).
47	FractionEqualDurations	The fraction of transitions between pitches that has duration quotient 1:1 (p. 152ff).
48	PitchLineCorrelation	The correlation of the pitch contours of the individual lines. For each line the maximum of the correlations with the other lines is taken. Of these values the mean is computed (p. 299ff, p. 93).
49	DurationLineCorrespondence	Similarity of the sequence of durations. This is computed in the same way as the previous feature, but instead of correlation the fraction of durations that corresponds is taken (p. 299ff).

The following features from the feature set of Jesser (1991) are included in the set of global features that is used in this thesis.

Index	Feature	Description
50	prime	fraction of the melodic intervals that is a prime.
51	aminsecond	fraction of the melodic intervals that is an ascending minor second.
52	amajsecond	fraction of the melodic intervals that is an ascending major second.
53	aminthird	fraction of the melodic intervals that is an ascending minor third.
54	amajthird	fraction of the melodic intervals that is an ascending major third.
55	afourth	fraction of the melodic intervals that is an ascending perfect fourth.
56	aaugfourth	fraction of the melodic intervals that is an ascending augmented fourth.
57	afifth	fraction of the melodic intervals that is an ascending perfect fifth.
58	aminsixth	fraction of the melodic intervals that is an ascending minor sixth.
59	amajsixth	fraction of the melodic intervals that is an ascending major sixth.
60	aminseventh	fraction of the melodic intervals that is an ascending minor seventh.
61	amajseventh	fraction of the melodic intervals that is an ascending major seventh.
62	aoctave	fraction of the melodic intervals that is an ascending perfect octave.
63	ahuge	fraction of the melodic intervals that is larger than an ascending octave.
64	dminsecond	fraction of the melodic intervals that is a descending minor second.
65	dmajsecond	fraction of the melodic intervals that is a descending major second.
66	dminthird	fraction of the melodic intervals that is a descending minor third.
67	dmajthird	fraction of the melodic intervals that is a descending major third.
68	dfourth	fraction of the melodic intervals that is a descending fourth.
69	daugfourth	fraction of the melodic intervals that is a descending augmented fourth.
70	dfifth	fraction of the melodic intervals that is a descending perfect fifth.
71	dminsixth	fraction of the melodic intervals that is a descending minor sixth.
72	dmajsixth	fraction of the melodic intervals that is a descending major sixth.
73	dminseventh	fraction of the melodic intervals that is a descending minor seventh.
74	dmajseventh	fraction of the melodic intervals that is a descending major seventh.
75	doctave	fraction of the melodic intervals that is a descending perfect octave.
76	dhuge	fraction of the melodic intervals that is larger than a descending octave.
77	astep	fraction of the melodic intervals that is an ascending step.
78	aleap	fraction of the melodic intervals that is an ascending leap.
79	dstep	fraction of the melodic intervals that is a descending step.
80	dleap	fraction of the melodic intervals that is a descending leap.
81	shortestlength	shortest duration such that all durations are a multiple of this shortest duration, except for triplets.
82	doublelength	fraction of the notes with duration of twice the shortest duration.
83	triplelength	fraction of the notes with duration of three times the shortest duration.
84	quadruplelength	fraction of the notes with duration of four times the shortest duration.
85	dotted	fraction of the notes that is dotted.
86	triplets	fraction of the notes that belongs to a triplet.
87	numlines	number of lines.
88	numpitchclasses	number of distinct pitch classes.