Ensemble of state-of-the-art methods for polyphonic music comparison

David Rizo and José M. Iñesta
Departamento de Lenguajes y Sistemas Informáticos
University of Alicante
Alicante, 03080, Spain
e-mail: {drizo,inesta}@dlsi.ua.es

Kjell Lemström
Dept. of Computer Science
University of Helsinki
FIN-00014 Helsinki, Finland
e-mail: klemstro@cs.helsinki.fi

Abstract—**Content-based music comparison is a task where no musical similarity measure can perform well in all possible cases. In this paper we will show that a careful combination of different similarity measures in an ensemble measure, will behave more robust than any of the included individual measures when applied as stand-alone measures. For the experiments we have used five state-of-the-art polyphonic similarity measures and three different corpora of polyphonic music.**

I. Introduction

The effectivity of a content-based music comparison or a content-based music retrieval method is mainly down to the appropriateness and degree of success of the underlying similarity measure. To develop an effective musical similarity measure, however, is anything but straight-forward. There is no musical similarity measure that works well in all musical applications. This is partly because musical similarity is, for instance, a culture-dependent, genre-dependent, encoding-dependent, application-dependent — even a user-dependent — issue. Therefore, in literature one can find several musical similarity measures that are effective, in various degrees of success, for different music-related problems.

Typical music-related problems include, for instance, content-based music retrieval, CBMR (”musical pattern matching”); content-based music comparison, CBMC (”musical document similarity”) and motive discovery methods (”musical pattern discovery”). One way to make a distinction between methods developed for these problems is to observe their capability to deal either with music containing several *voices* and call such music *polyphonic*, or consider as *polyphonic* music that which simultaneous notes. The most straight-forward methods compare only the melodic lines of *monodies* [1]. One way to deal with polyphonic music is to reduce polyphonic structure in a monophonic form by using a heuristic, such as the skyline algorithm [12]. Naturally, after such a reduction, similarity measures developed for the monodies can be used for this case as well. There are also several methods capable of dealing with polyphony without any reduction, see e.g. [1], [13], [8].

In this paper we deal with symbolically encoded, polyphonic music and focus on the CBMC problem, where the similarity between given two full musical works is to be defined. In order to avoid problems resulting from a use of an unsuitable similarity measure, such as stuck on a local maxima, we suggest an approach that combines several similarity measures. As a careless selection of included similarity measures, such as the one including all possible measures, may result in an excessive running time and a severe bias caused by a ”clique of similarity measures”, we collect a minimal set of measures having a different aspect on the problem. In a successful compilation, or *ensemble*, of similarity measures, the included individual similarity measures fail and also succeed in different instances of the comparison problem, improving the robustness of the approach.

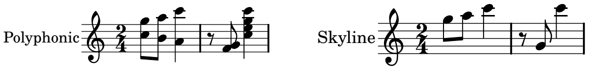

Fig. 1. Sample score (left) and its skyline reduction (right).

Based on our experiments on three polyphonic music databases and several polyphonic similarity measures reported earlier in the literature, the selection is carried out by using techniques of diversity measurement. To avoid overfitting the resulting ensemble to the data used in the experiments, we have used Kuncheva's overproduce and select method [2].

II. Background

As CBMC has been an active research area for over a decade, in literature one can find several similarity measures developed for the task. In this section we give an overview on similarity measures relevant to our case. The presented methods have been carefully chosen so that they both represent the state-of-the-art and that they give us the needed diversity so that the ensemble to be composed of them would work as well as possible.

We have generated monophonic versions of our three corpora by applying the skyline algorithm (see Fig.1) on the original corpora. In order to boost the diversity among the methods in the resulting ensemble, we have included some monophonic similarity measures that work on the skylined corpora. In addition, the included polyphonic measures are forced to work with both with the skylined and the original versions of the corpora.

Let us now briefly survey the methods and their associated parameters to be adjusted in our experiments. In each of the case, we will also show how the method encodes the short example depicted in Fig.1.

A. Point-pattern / line-segment similarity

Ukkonen et al. [13] considered the CBMR problem and suggested music to be modeled as sets of horizontal line segments in the Euclidean plane, formed by tuples of $\langle$ pitch, duration $\rangle$ (see Fig.2). A reduction of this representation, the point-pattern representation, considers only the starting points

[1] Monodic compositions either have only a single melodic line, or the composition is dominated by a single melodic line

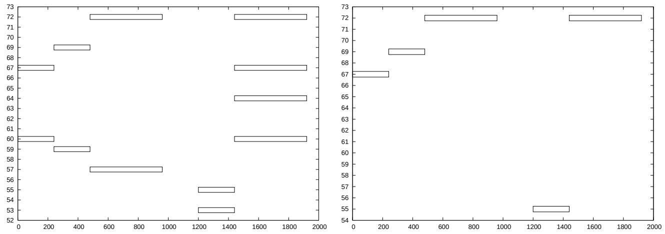

Fig. 2. Line-segment representation (left) and its skyline reduction (right) of Fig.1 on a pitch-against-time Euclidean plane.

Polyphony	r	Sets of [onset, pitch]
Skyline	4	{[0,69], [0,67], [1,72], [2,55], [3,72]}
Skyline	8	{[0,67], [1,69], [2,72], [5,55], [6,72]}
Polyphonic	4	{[0,69], [0,67], [0,60], [0,59], [1,72], [1,57], [2,55], [2,53], [3,72], [3,67], [3,64], [3,60]}
Polyphonic	8	{[0,67], [0,60], [1,69], [1,59], [2,72], [2,57], [5,55], [5,53], [6,72], [6,67], [6,64], [6,60]}

Fig. 3. PROMS representation of Fig.1. Onset is given relative to its position within the bar.

of the line segments, that is, it is formed of tuples ⟨ pitch, onset time ⟩. Out of their three problems, P2 and P3 are relevant to our case. Intuitively, the similarity of two pieces of music is computed by finding an alignment in the superimposition of two planes representing the considered pieces of music. The alignment should maximize the coincidence rate between the two planes either in point-pattern representation (problem P2) or in line-segment representation (problem P3). Recently, in [3], efficient indexing algorithms for P2 was given. We have included two of them, referred to as P2v5 and P2v6.

B. Bar-specific, quantized point-pattern similarity

Clausen et al. used inverted indices with a representation resembling the above point-pattern representation [1]. The onset times of the notes are quantized to a pre-selected resolution r. Thus, both the pitch and time dimensions become discrete. Moreover, onset times are represented relatively to their metrical position within the musical bar (see Fig.3). The information within a bar constitutes the unit for a query. The retrieval method finds occurrences (total and partial) that have similar metrical positions as the query. Local time and pitch fluctuations cannot be dealt with. Tempo invariance can be obtained by conducting a metrical structure analysis phase and transposition invariance by using a mathematical trick that outperforms the brute-force solution.

In the original algorithm, the approximate search is accomplished by allowing k dissimilarities, in maximum, between the query and the database document. To our needs, where whole pieces of music are to be compared, the original algorithm has been modified to return the normalized number of coincidences in the best alignment. In the sequel, we will refer to this technique as *PROMS similarity*, named after Clausen et al.'s original system.

C. Tree similarity

The tree representation [8] is based on a logarithmic subdivision along the time dimension of a musical bar. Each bar is encoded as a tree and the root-level correspond to the whole note covering the bar. Then, for instance in a binary division, in the next level we have two subtrees corresponding to the two half notes (think of them as fragmentations of the whole note of the root level). This division is continued until the preferred time resolution is met. All trees (i.e. bars) are finally rooted to a common parent representing the whole musical piece. Note that time is implicit in the representation when considering left-to-right ordering of the nodes. The building of a tree is started with pitch information stored in leaves at a level corresponding to its onset (with respect to its position within the bar) and duration. As we are dealing with polyphonic music, the labels of nodes are multisets that contain the cardinality of occurrences of a pitch class in that node. In a second phase, the node labels are bottom-up propagated: The contents of the children are merged in the parent node by using the multiset union operator. Moreover, in order to make the trees smaller, leaves at a level exceeding a depth threshold L are pruned.

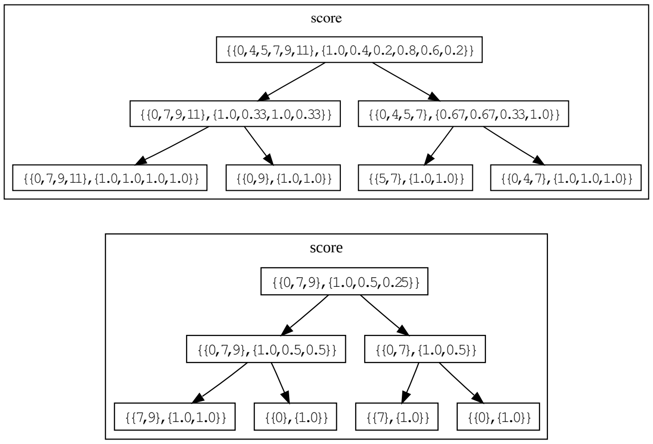

Fig. 4. Tree representations of the polyphonic version (top) and skyline version (bottom) of Fig.1 with $L = 2$. The node labels are of form "{set of pitch classes}, {corresponding cardinalities}".

The similarity of two pieces of music represented in this way is measured by combining Selkow's tree-edit-distance [10], that accounts for structural similarity, with a multiset distance measure accounting for musical surface level similarity. In [9] several multiset similarity measures to this end have been suggested.

D. n-gram similarity

Linear strings have often been used for encoding monophonic music. In doing so, one may also harness the general string matching techniques for the problem. In [11], Uitdenbogerd suggested the use of n-gramming on strings over a pitch interval alphabet, where the original interval values were reduced by performing $\mod 12$ over them.

We have implemented four methods based on the string matching framework performing over the aforementioned interval alphabet. The first uses the classical Levenshtein distance [4] (here called *NGEdit*). Then we have two modifications of the Levenshtein distance, one using a simple local alignment (called *NGLocal*) and one using so-called start-match alignment (called *NGMatch*). The last one is Uitdenbogerd's n-gramming technique.

Representation	Actual value
String	ophr
2-grams	{hr, op, ph}
3-grams	{oph, phr}

Fig. 5. A string and the corresponding 2- and 3-grams of skyline score in Fig.1

E. Graph similarity

In [7], Pinto modeled monodies by using directed graphs over the pitch profile. In his graph, each pitch class is associated with a node, and each melody transition from a pitch class to another with a label. The label represents the frequency of the interval, between the two associated nodes, within the represented monody (see Fig.6). The graph represents also the interval from the last note of the melody to the first one.

The similarity of the graphs is measured by using a their laplacian spectra as a feature vector. Laplacian spectra is invariant under shifts of the columns, that is, invariant under musical transpositions.

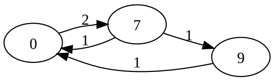

Fig. 6. Graph representation of skyline score in Fig.1

III. Classifier combination

As argued above, any single musical representation or similarity algorithm cannot give succesful results in all our cover version identification tasks. In [2], Kuncheva showed that a carefully chosen combination of classifiers, compiled into so-called *ensembles*, will result in more robust systems that perform at least at a comparable level when compared with any individual classifier. There are some constraints, however, that limit the performance of such ensembles, the most important being that the classifiers of the ensemble should differ in the kind of errors they make. This led us in chosing very different approaches, the ones described in Section II.

To measure the diversity amongst the chosen classifiers in the ensemble, we have used the *interrater-agreement* κ: the lower the κ, the higher the disagreement and hence higher diversity (see eq.(10.8) in [2]). The measure is computed for each pair of classifiers, and it is inversely proportional to the diversity.

To choose the ensemble from a set of available classifiers, Kuncheva proposes to use the *overproduce and select* method. In our case this works as follws: Given a training set and the classifications of all classifiers, choose the ones giving the highest diversity and the lowest average error rate. We obtain such a set of classifiers by computing a *pareto-optimal* set. The latter set is computed as follows: the κ and average error rate for each pair of classifiers is computed, then only those pairs with both best κ and average error rate are kept (they are said to be *non-dominated*). The *pareto-optimal* set is composed by the classifiers that form that kept pairs.

Once the most suitable classifiers are selected, any of the combination scheme in [5] can be used. In this paper, we have used the raw voting scheme.

IV. Experiments

The experiments are designed to check the suitability of the combination of different polyphonic music similarity paradigms in comparison to individual methods. To this end, first the performance of each individual method with all its possible setups has been tested, then the diversity of all possible pairs of classifiers has been studied using the κ statistic. Given this diversity measure the best classifiers have been chosen using several selection schemes, and combined using a voting ensemble.

A. Corpora

Three different corpora of cover versions have been collected in order to show the behaviour of the methods with different data. Each corpus is organized into songs or classes. For each song there is a main prototype and a set of variations or cover versions.

The first corpus, called *ICPS*, has 68 MIDI files corresponding to covers of the incipits of seven musical works: Schubert's "Ave Maria", Ravel's "Bolero", the children songs "Alouette", "Happy Birthday" and "Frère Jacques", the carol "Jingle Bells" and the jazz standard "When The Saints Go Marching In". All the works in this corpus have a similar kind of accompaniment tracks with a prominent melody track.

The second corpus, named *VAR*, consists of 78 classical works representing variations of 17 different themes as written by the original composer: Tchaikovsky "variations on a rococo theme" op.33, Bach "English suites" BWV 806-808 (suite 1 courante II, suite 2, 3, and 6 sarabande), and Bach "Goldberg variations". In this case, the variations are founded mostly on the harmonic structure of a main theme.

The third one, called *INET* is made of 101 whole MIDI files downloaded from the Internet corresponding to 31 different popular songs. It contains real-time sequences with mistakes and different arrangements of the original versions.

B. Quality measurement

The experiments have been performed using a query / candidates scheme, i.e., given a corpus, for each song its main prototype (acting as query) is compared with all the cover versions of all songs. The similarity values of all the comparisons are ordered and following a 1-NN rule, the best value is taken as the answer. This answer is correct if it corresponds to the same song of the query. Any other situation is considered as an error.

Thus, the success rate is measured as the rate of correct answers for all queries or main prototypes in a corpus.

C. Individual performances of methods

In the following points, the results for all possible setups of the tested methods are plotted. The reader can remind the meaning of each parameter in the figures in the introduction of the methods in Section II above.

The monophonic methods (graph and n-grams) have been fed with a skyline reduction of the corpora. The polyphonic methods have been tested using both the original corpora and also the same skyline reduction. In the plots, the corpora with the skyline applied are denoted with a "M-" prefix.

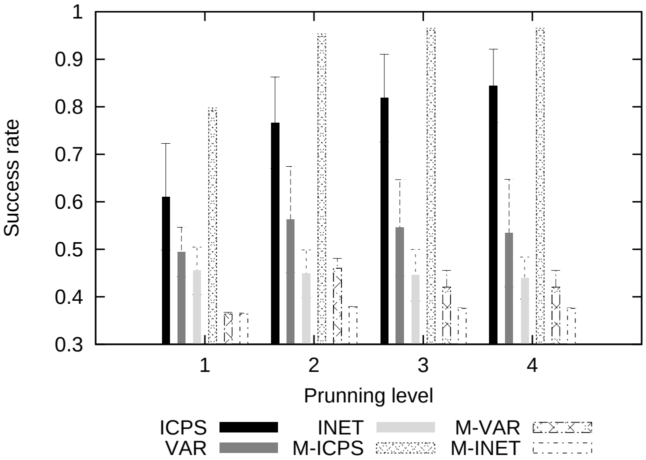

Fig. 7. Prunning levels

1) Trees results: The results in Fig. 7 show the averages and standard deviations for each pruning level with all multiset distances used, resulting in 44 different classifiers grouped into those 3 pruning levels. It can be noticed that for the corpus with a more evident melody line, the *ICPS*, the skyline reduction improves the classification rate. This is not the case for the other corpora, where a more important harmonic component is found, and the polyphony is required to identify versions. For the plots, the best pruning level after propagation is $L = 2$. For monophonic corpora, the system may require larger trees with the increase of classification times.

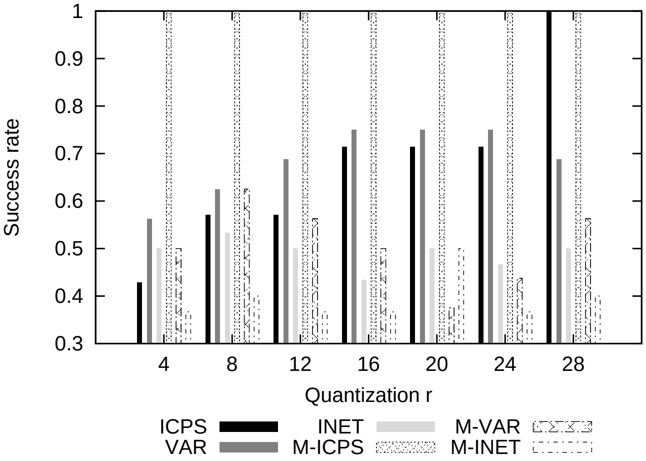

Fig. 8. PROMS, the x axis represents the different resolutions r in which each song bar is quantized

2) PROMS results: The different setups of the PROMS depend on the quantization per bar. We have tested from 4 (which means a resolution of quarter note for a 4/4 meter) to 28 where the system seems to stabilize.

The PROMS representation is very sensitive to quantization. This quantization produces chords with closer notes, being this fact noticeable in the behaviour of the monophonic corpus in front of the polyphonic one (see Fig. 8): a quantization of polyphonic content leads to too dense chords, and as the r raises, it gets less dense. On the other hand, more quantization of monophonic content helps in the classification.

Anyway, it seems that the best average setup is that of r=28, that is not tight to any meter as r=12 is to 3/4, or $r = 16$ to 4/4.

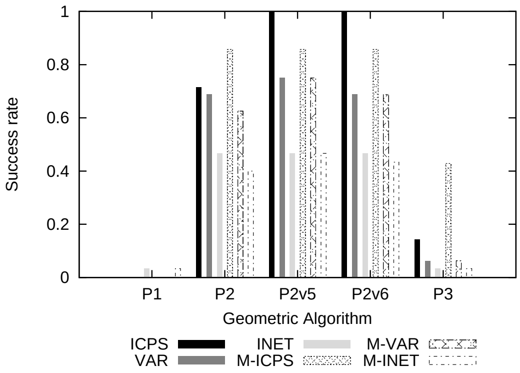

Fig. 9. Geometric algorithms

3) Geometric results: The five different geometric algorithms results are plotted in Fig. 9, showing that the best algorithms are P2v5 and P2v6. It is remarkable the robustness of these classifiers against the polyphony, behaving comparably with the original content and the “skylined” one.

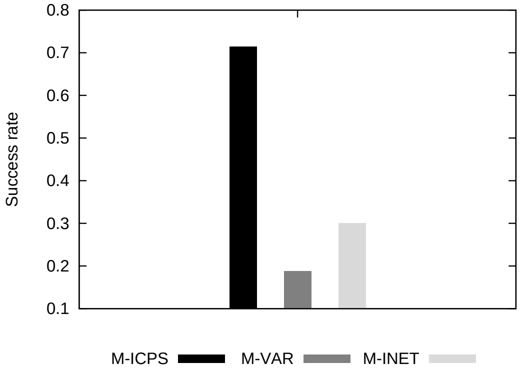

Fig. 10. Graphs

4) Graphs results: This method works only with the pitch component of monodies, so it uses limited information. Results shown in Fig. 10 evidence this fact. The only well managed corpus is that with a clear melody, *ICPS*. However, the best quality of this paradigm is its performance time, so it is worth to include it in the ensembles.

5) n-grams results: The results of the string matching approach (Fig. 11) and the n-grams classifying (Fig. 12) show that the later can deal better with skylined content than a simple string matching approach. The best results are located around the 5-grams. Anyway, as was previously introduced in [6], the best option is to combine various n-gram sizes, what will be done in the ensembles. Being the string methods very fast, we will include them in the combinations.

6) Best of each method and each corpus: In Table I the best of each method setup is detailed for each corpus. None

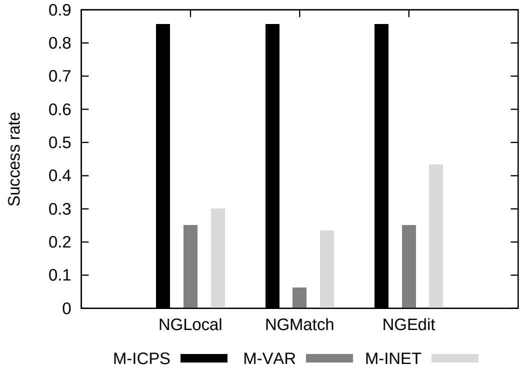

Fig. 11. Uitdenboderg Strings

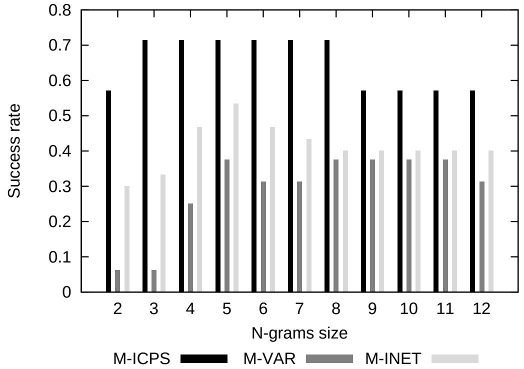

Fig. 12. Uitdenboderg ngrams

of the paradigms can be stated as the best for all situations, so a combination that takes advantage of the qualities of each seems to be the best option.

D. Ensemble methods

In this experiment, the ability of ensemble methods to improve overall classification rates has been tested. In order to choose only the best combinations in terms of number of classifiers included some selection methods have been compared.

These selections are based in choosing either the most diverse M classifiers, i.e., those giving the most different classifications, or choosing the classifiers with best trade-off between diversity and average error rate. Both selections are based on a plot of the κ statistic. Figures 13, 14, and 15 represent, for each corpus, that κ vs. average error rate for each pair of classifiers. The most diverse M classifier selection chooses the M rightmost points in these plots. The *pareto-optimal set* (aka. *Par.*) is shown by square points in the plots.

Fig.16 shows the behaviour of selection methods with the three corpora. From the plot, it can be stated that the best option is the use of the *pareto-optimal-set*, for the success rates and the number of classifiers involved.

Finally, the results of this ensemble compared to the best of individual methods (Table II) show that the most robust method is the ensemble, being significant the improvement in the most difficult method, the *INET*.

TABLE I
BEST SUCCESS RATES

Corpus	Method	Best method setting	Result (%)
VAR	Geometric	P2v5	75
M-VAR	Geometric	P2v5	75
M-VAR	Graph		19
VAR	PROMS	$r = 24$	75
M-VAR	PROMS	$r = 8$	63
VAR	Trees	Cosine Dist, $L = 2$	75
M-VAR	Trees	Cosine Dist, $L = 2$	75
M-VAR	String matching	Local, Edit	25
M-VAR	n-grams	$n \in \{5, 8, 9, 10, 11\}$	38
ICPS	Geometric	P2v5 and P2v6	100
M-ICPS	Geometric	P2,P2v5,P2v6	86
M-ICPS	Graph		71
ICPS	PROMS	$r = 28$	100
M-ICPS	PROMS	Any r	100
ICPS	Trees	Log dist, $L = 1$	100
M-ICPS	Trees	Var. dist, $L = 4$	100
M-ICPS	String matching	Any	86
M-ICPS	n-grams	$n \in [3..8]$	78
INET	Geometric	P2v6	47
M-INET	Geometric	P2,P2v5,P2v6	47
M-INET	Graph		30
INET	PROMS	$r = 8$	53
M-INET	PROMS	$r = 20$	50
INET	Trees	Log dist, $L = 1$	53
M-INET	Trees	Harmonic mean, $L = 4$	43
M-INET	String matching	Edit	43
M-INET	n-grams	n=5	53

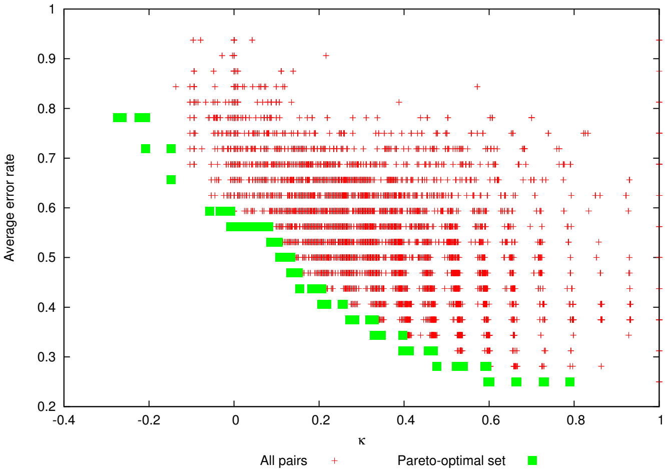

Fig. 13. κ vs. average error rate in corpus *VAR*

V. Conclusions and Future Works

In this paper, we have considered five different approaches to measure musical similarity. In order to study how they perform, both as a stand-alone measure and as part of an ensemble measure, we have experimented on them using three distinct music corpora. An ensemble measure, or classifier, produced by *overproduce and select* method, has been shown to be superior to any of the individual stand-alone classifiers.

Now that we know the ensemble classifier to give good results, our next step is to build a large corpus with a good variety in genres. Having such a corpus, we would be able to apply standard cross-validation rules in training and testing with new data. We will also apply some more sophisticated combination schemes that are expected to improve the rates of correct classification.

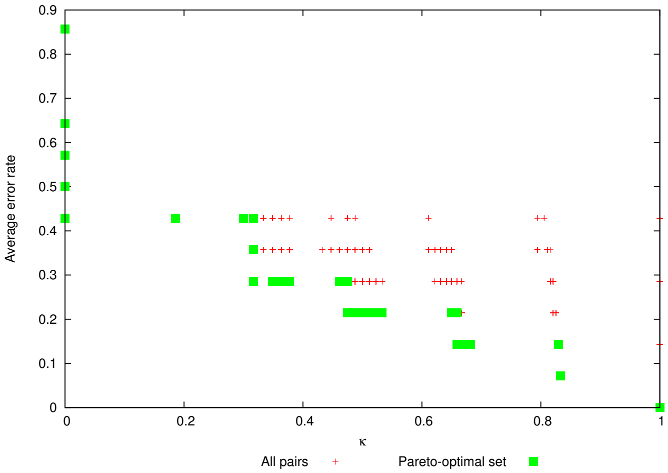

Fig. 14. κ vs. average error rate in corpus *ICPS*

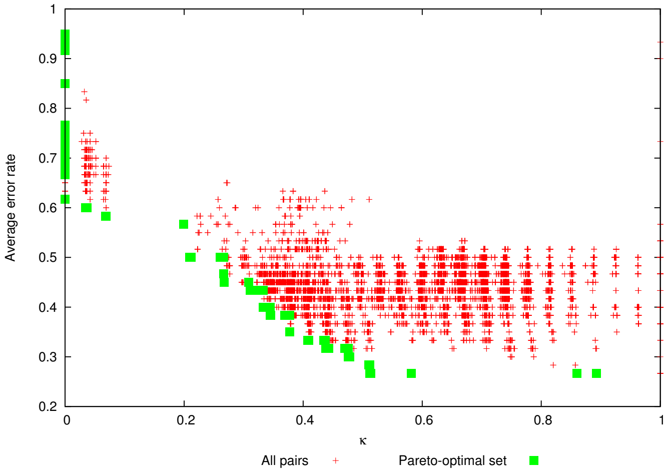

Fig. 15. κ vs. average error rate in corpus *INET*

TABLE II
BEST SUCCESS RATES OF ALL INDIVIDUAL AND COMBINED METHODS

Corpus	Method	Best Result (%)
VAR	Individual	75
VAR	Combined	84
ICPS	Individual	100
ICPS	Combined	100
INET	Individual	53
INET	Combined	82

VI. ACKNOWLEDGMENTS

Kjell Lemström was supported by the Academy of Finland (Grant 108547).

David Rizo and José M. Iñesta are supported the Spanish Ministry projects: TIN2006–14932–C02 and Consolider Ingenio 2010 (MIPRCV, CSD2007-00018), both partially supported by EU ERDF. Special thanks to José F. Bernabeu for his technical support, Iman Suyoto, and Alberto Pinto for their collaboration.

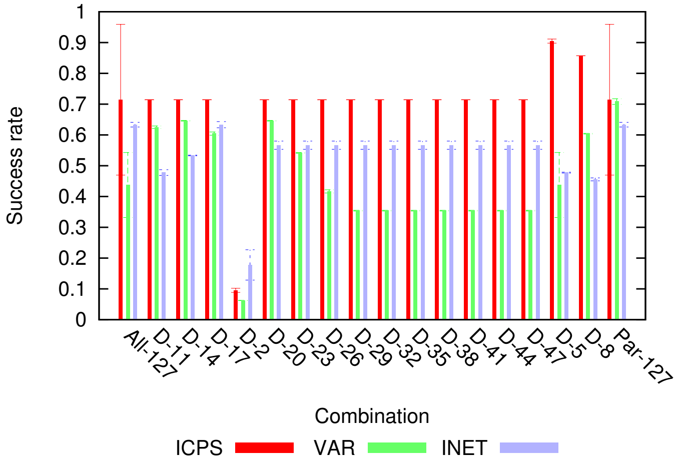

Fig. 16. Classifier selection methods. *All* means no selection, that is, use all classifiers. DM is the ensemble built from the classifiers included in the M most diverse pairs. *Par.* is the *pareto-optimal-set*.

REFERENCES

[1] Michael Clausen, Ronald Engelbrecht, D. Meyer, and J. Schmitz. Proms: A web-based tool for searching in polyphonic music. In *ISMIR*, 2000.

[2] Ludmila I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, July 2004.

[3] Kjell Lemström, Niko Mikkilä, and Veli Mäkinen. Fast index based filters for music retrieval. In *Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR 2008)*, September 2008.

[4] Vladimir I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710, 1966.

[5] F. Moreno-Seco, José M. Iñesta, P. Ponce de León, and L. Micó. Comparison of classifier fusion methods for classification in pattern recognition tasks. *Lecture Notes in Computer Science*, 4109:705–713, 2006.

[6] Nicola Orio. Combining multilevel and multifeature representation to compute melodic similarity. MIREX, 2005.

[7] Alberto Pinto, Reinier H. van Leuken, M. Fatih Demirci, Frans Wiering, and Remco C. Veltkamp. Indexing music collections through graph spectra. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR'07)*, Vienna, Austria, September 2007.

[8] David Rizo, Kjell Lemström, and José M. Iñesta. Tree structured and combined methods for comparing metered polyphonic music. In *Proc. Computer Music Modeling and Retrieval 2008 (CMMR'08)*, pages 263–278, Copenhagen, Denmark, Copenhagen, Denmark, May 19-23 2008.

[9] David Rizo, Kjell Lemström, and José M. Iñesta. Tree representation in combined polyphonic music comparison. *Lecture Notes in Computer Science, selected papers from the CMMR 2008 (to appear)*, 2009.

[10] Stanley M. Selkow. The tree-to-tree editing problem. *Inf. Process. Lett.*, 6(6):184–186, 1977.

[11] Alexandra L. Uitdenbogerd. N-gram pattern matching and dynamic programming for symbolic melody search. MIREX, 2007.

[12] Alexandra L. Uitdenbogerd and Justin Zobel. Manipulation of music for melody matching. In *ACM Multimedia*, pages 235–240, 1998.

[13] Esko Ukkonen, Kjell Lemström, and Veli Mäkinen. Sweepline the music! In *Computer Science in Perspective*, pages 330–342, 2003.