Melodic Track Identification in MIDI Files Considering its Imbalanced Context*

Raúl Martín, Ramón A. Mollineda, and Vicente García

Department of Programming Languages and Information Systems
University Jaume I of Castellón, Spain
{martinr, mollined, vgarcia}@uji.es

Abstract. In this paper, the problem of identifying the melodic track of a MIDI file in imbalanced scenarios is discussed. A polyphonic MIDI file is a digital score that consists of a set of tracks where usually only one of them contains the melody and the remaining tracks hold the accompaniment. This fact leads to a two-class imbalanced problem which, unlike previous works, is managed by over-sampling the melody class (the minority one) and by under-sampling the accompaniment class (the majority one) until both classes have the same size. Experimental results over three different music genres prove that learning from balanced training sets clearly overcomes the standard classification process.

Keywords : Melody finding, music information retrieval, class imbalance problem, classification.

1 Introduction

This paper aims to solve the problem of automatic identification of the melodic line from a polyphonic MIDI file. A MIDI file is a kind of digital score as MusicXML and other XML music formats. It consists of a set of tracks (this is why it is referred to as polyphonic), where only one of them is usually the melodic track while the remaining tracks contain the accompaniment of that melody. An effective solution to this problem could have interest to a large number of applications. For example, it could speed up query operations in multimedia databases such as to recover a MIDI file whose melodic line matches with a hummed or whistled melody [1], to recommend songs similar to a given one by comparing their melodies, among others. Another application could be the plagiarism detection in the field of copyright management by identifying a percentage of the melody of one song in another one.

The automatic identification of the melodic line could be modelled as a two-class problem: the melody class and the accompaniment (or non-melody) class. The first one can be considered as the minority class because its number of samples, usually one track per MIDI file, is much lower than the size of the non-melody class which contains many accompaniment tracks per MIDI. Therefore, this relation generally leads to a class-imbalance problem.

* Our acknowledgements to the *Pattern Recognition and Artificial Intelligence Group* at the University of Alicante for providing us the datasets used in this paper.

Some previous papers have addressed the main goal of this work but ignoring its imbalanced nature. Most of them [2–4] represent a track by a vector of low-level statistical descriptors about its musical content, which are then used in a common learning/classification process. A different approach follows a structural paradigm by coding the sequence of notes as strings or trees [5].

This paper deals with the automatic identification of the melodic line in MIDI files but, unlike previous works, it takes into account the imbalanced nature of the problem and provides results for more than one classifier. The imbalance is managed by resampling classes in the training set as a preprocessing stage previous to classifier learning. This process balances the sizes of both classes by either over-sampling the minority class or under-sampling the majority class. Experiments are performed over corpora of MIDI files belonging to three different musical genres, crossing them for training and testing purposes. Most of classification results obtained from resampled training sets were significantly better than those derived from the corresponding original imbalanced training set.

2 Methodology

An overview of the solution is shown in Fig. 1 where the four main steps are remarked. Next subsections explain these steps.

2.1 Track Features Extraction

This step creates vector representations for all tracks of the MIDI files included in both the training and test corpora. As a result, two related sets of track vectors are obtained for training and testing purposes. Tracks are described by 38 features (many of them used in [2, 3]) summarizing its musical content and by a class label indicating whether the track contains the melody or not. All of these features (see Table 1) have continuous ranges of numeric values. This fact is necessary to apply the SMOTE technique explained in Sect. 2.2. The features were also pre-processed by a mean imputation to fill in missing values and finally, they were normalized in the range [0,1]. The feature *Song identifier*, that indicates the MIDI file to which a track belongs to, is only used in testing to assess the effectiveness of the MIDI classification process.

2.2 Resampling

As commented above, the original training set of track vectors is a two-class imbalanced problem because the number of melodic tracks is much lower than the number of non-melodic tracks. One possible solution to imbalance at data level is to resample the original training set either by over-sampling the minority class or by under-sampling the majority class until the class sizes are similar. In this work, one method of each strategy has been applied: *SMOTE* for over-sampling and *RUS* for under-sampling the training set.

Table 1. Set of track features

- *Global properties*
  - Song identifier
  - Number of notes
  - Track duration
  - Polyphony rate
  - Occupation rate
- *Pitch*
  - Highest
  - Lowest
  - Range
  - Average
  - Standard deviation
  - Average relative
- *Syncopation*
  - Number of syncopated notes
- *Absolute intervals*
  - Highest
  - Lowest
  - Range
  - Average
  - Standard deviation
  - Average relative
  - Most repeated
  - Number of distinct
- *Note duration*
  - Shortest
  - Longest
  - Range
  - Average
  - Standard deviation
  - Average relative
- *Non-diatonic notes*
  - Count
  - Average
  - Standard deviation
  - Average relative
- *Rests*
  - Number of significant
  - Number of non-significant
  - Shortest duration
  - Longest duration
  - Duration range
  - Average duration
  - Standard deviation duration
  - Average relative duration

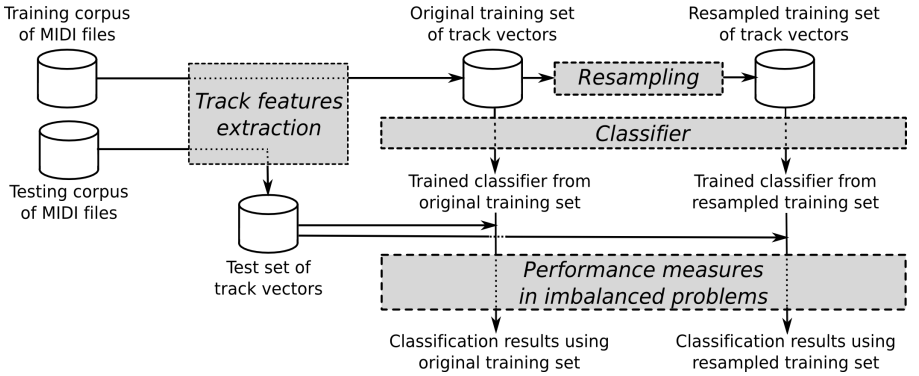

Fig. 1. Architecture of the solution

SMOTE (Synthetic Minority Oversampling TEchnique) [6] is a method that generates new synthetic samples in the minority class from a number of instances that lie together. For each sample in the minority class, this algorithm computes the k intra-class nearest neighbours, and several new instances are created by interpolating the focused sample and some of its neighbours randomly selected. Its major drawback is the increase of the computational cost of the learning algorithm. On the contrary, *RUS (Random Under-Sampling)* [7] is a non-heuristic method that aims to balance class distributions by randomly discarding samples of the majority class. Its major drawback is that it can ignore potentially useful data that could be important in the learning process.

2.3 Classifiers

The final purpose of the classification stage is to identify the melodic track in each MIDI file. This process is made up of two decision levels: i) *track level*, where individual tracks are classified into either melodic or accompaniment classes and

ii) *MIDI file level*, in which the identification of the melodic track of a MIDI file is carried out based on results at track level. The training process at track level is based on both the original and the resampled training sets of track vectors (see Fig. 1). As regards the set used to train (original or resampled), the effectiveness of the detection of the melodic track at MIDI file level is evaluated. A detailed description of this process can be stated as follows:

Track level

1. Given a track, a classifier assigns degrees of membership (probabilities) to both classes (melody and accompaniment)
2. Tracks are discarded when one of the following two conditions is satisfied:
   - the difference between both probabilities is lower than 0.1
   - the probability of being melody is higher than the non-melody probability, but lower than 0.6

MIDI file level

1. Given all non-discarded tracks from the same MIDI file, the one with the highest positive difference between the two probabilities of being melody and accompaniment respectively, is selected as the melodic track
2. The decision is considered as a *hit* if
   - *True Positive*: the selected track is originally labelled as melody, or
   - *True Negative*: in a file with no track labelled as melody, all its tracks have been discarded or they have negative differences between their probabilities
3. The decision is considered as a *mistake* if
   - *False Positive*: the selected track is originally unlabelled or labelled as accompaniment, or
   - *False Negative*: in a file with at least one track labelled as melody, all its tracks have been discarded or they have negative differences between their probabilities

The base classifiers used at track level are the k-Nearest Neighbour (k-NN), a Support Vector Machine (SVM), a Multi-Layer Perceptron (MLP) and a Random Forest (RF). They were chosen due to their diversity in terms of the geometry of their decision boundaries. In this paper, the experimental results are obtained by using the classifier implementations included in the WEKA toolkit [1] with their default parameters. In addition, k-NN was performed with $k = 1$ and $k = 5$, and RF (a weighed combination of decision trees) was configured with 10 trees, each of them using five features randomly selected.

2.4 Performance Measures in Class Imbalanced Problems

A typical metric for measuring the performance of learning systems is classification accuracy rate, which for a two-class problem can be easily derived from a 2×2 confusion matrix defined by i) TP (True positive) and ii) TN (True Negative), which are the numbers of positive and negative samples correctly classified,

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

Table 2. Corpora used in the experiments

CorpusID	*Music Genre*	*Midi Files*		*Tracks*		
		Total number	*without melody tracks*	*non-melody*	*melody*	*unlabeled*
CL200	Classical	200	1%	489	198	16
JZ200	Jazz	200	1.5%	561	197	11
KR200	Popular	200	20.5%	1171	159	338
CLA	Classical	131	35.88%	284	84	265
JAZ	Jazz	1016	0.98%	3131	1006	71
KAR	Popular	1358	8.18%	9416	1247	858

respectively, and iii) FP (False positive) and iv) FN (False Negative), which are the numbers of misclassified negative and positive samples, respectively. This measure can be computed as $Acc = (TP + TN)/(TP + FN + TN + FP)$.

However, empirical evidence shows that this measure is biased with respect to the data imbalance and proportions of correct and incorrect classifications [8]. Shortcomings of these evaluators have motivated search for new measures. Some straightforward examples of these alternative measures used in this work are: (i) *True positive rate* (also referred to as *recall*) is the percentage of positive examples which are correctly classified, $TPr = TP/(TP + FN)$; (ii) *Precision* (or *purity*) is defined as the percentage of samples which are correctly labelled as positive, $Precision = TP/(TP+FP)$; and (iii) *F-measure* combines TPr and Precision, $F\text{-}measure = (2 * TPr * Precision)/(TPr + Precision)$.

3 Experimental Results

3.1 Datasets

Experiments involve six datasets of track vectors obtained from a same number of corpora of MIDI files created in [2, 3]. These corpora contain MIDI files of three different music genres: classical music (CL200 and CLA), jazz music (JZ200 and JAZ) and popular music in karaoke format (KR200 and KAR). A more detailed description of these corpora is shown in Table 2. From each corpus, a corresponding dataset of 38-dimensional track vectors is available (see Sect. 2.1) where each vector has been manually labelled by a trained musicologist as melody, non-melody or unlabelled.

These corpora can be divided into two groups with regard to their data complexity and, also mostly, due to their sizes. A first cluster can include CL200, JZ200 and KR200, and they have in common the number of MIDI files (200). Moreover, most of them have well-defined melodic tracks which make them suitable for training purposes. On the contrary, CLA, JAZ and KAR are more heterogeneous corpora and, consequently, lead to more challenging tasks [2, 3].

3.2 Experimental Design

In the following experiments, different combinations of CL200, JZ200 and KR200 were employed for training, whereas CLA, JAZ and KAR are used as three

separated test sets. In a first series, the classifier is trained with two music genres (among CL200, JZ200 and KR200) and is tested with the remaining one (among CLA, JAZ and KAR). In the second experiment, only one training set, here named ALL200, is built from the union of CL200, JZ200 and KR200. The rationale of this experimental design is to find out whether the melodic track identification in a music genre depends on including samples of the same music genre in the training set. Unlike previous works [2, 3], conclusions are provided from the analysis of the results of more than one classifier (see Sect. 2.3).

Along with the previous objective, this paper aims at studying the convenience of managing the imbalanced nature of the training sets. Table 2 shows the imbalance between the distributions of both melody and non-melody classes in all corpora. In order to evaluate the relevance of imbalance in classification results, all previous experiments were performed over three versions of each training set: i) the imbalanced original case, ii) a balanced version by SMOTE, and iii) a balanced version by RUS (see Sec. 2.2).

Due to the random behaviour of SMOTE and RUS, each experiment over the balanced training sets was performed 10 times and the results were averaged. In the case of the RF classifier which selects random features, the experiments were also repeated 10 times in the imbalanced original training set.

For each experiment the Accuracy (Acc) was computed taking into account all MIDI files. However, TPr, Precision (Prec) and Fmeasure (Fm) ignored the MIDI files without melody tracks (see Table 2) as was stated in previous related works [2, 3]. Note that it affects the calculation of TN and FP (see Sect. 2.3).

3.3 Experiment I

This experiment evaluates the effectiveness of detecting the melodic track in MIDI files of a specific music genre, when no samples of this genre have been used in the training stage. In particular, the following three pairs of training and test sets were considered: i) (JZ200+KR200, CLA), ii) (CL200+KR200, JAZ) and iii) (CL200+JZ200, KAR). Its results are shown in Table 3.

The experiment can be analysed in two directions. The first one compares classification results with regard to the music genres used for training and testing. The second one is related to the evaluation of the influence of managing imbalance before classification.

With respect to the analysis among the music styles, all classifiers except SVM seems to be sensitive to the genres used for training and testing both in imbalanced and balanced contexts. The most robust classifier is SVM, which obtained steady and high rates on the positive class (TPr, Prec and Fm). The worst results belong to RF as regards its low values in all measures, probably because of its random behaviour is more sensitive to the lack of samples of the test genre in the training set. In the case of Acc, its low values can be explained by the fact that the accuracy considers those test MIDI files without melodic track, which reach the amount indicated in Table 2.

In regard to the influence of managing imbalance, most of classifiers operating in the balanced contexts improve their results of the original imbalanced

Table 3. Averaged results of the Experiment I

Training estrategy	*Classifier*	*Train:* JZ200+KR200 *Test:* CLA				*Train:* CL200+KR200 *Test:* JAZ				*Train:* CL200+JZ200 *Test:* KAR			
		Acc	*TPr*	*Prec*	*Fm*	*Acc*	*TPr*	*Prec*	*Fm*	*Acc*	*TPr*	*Prec*	*Fm*
Original	*1-NN*	0.53	0.65	0.93	0.76	0.62	0.64	0.96	0.77	0.73	0.92	0.84	0.88
	5-NN	0.58	0.67	0.96	0.79	0.61	0.62	0.97	0.76	0.71	0.98	0.78	0.86
	SVM	0.69	0.95	0.99	0.97	0.87	0.89	0.99	0.94	0.85	0.97	0.94	0.96
	MLP	0.62	0.96	0.94	0.95	0.56	0.57	0.96	0.72	0.67	0.97	0.74	0.84
	RF	0.45	0.18	1	0.3	0.05	0.04	0.78	0.08	0.47	0.95	0.52	0.67
SMOTE	*1-NN*	0.57	0.73	0.97	0.83	0.74	0.76	0.97	0.85	0.74	0.94	0.84	0.89
	5-NN	0.64	0.84	0.97	0.9	0.84	0.87	0.97	0.92	0.73	0.99	0.8	0.88
	SVM	0.67	0.99	0.99	0.99	0.94	0.96	0.99	0.98	0.86	0.98	0.95	0.97
	MLP	0.63	0.87	0.94	0.91	0.68	0.71	0.94	0.81	0.41	0.77	0.49	0.6
	RF	0.43	0.23	1	0.38	0.06	0.05	0.8	0.09	0.59	0.95	0.66	0.78
RUS	*1-NN*	0.6	0.84	0.96	0.89	0.83	0.86	0.98	0.91	0.8	0.98	0.89	0.93
	5-NN	0.63	0.79	0.97	0.87	0.79	0.81	0.98	0.89	0.76	0.98	0.84	0.9
	SVM	**0.67**	**0.99**	**1**	**0.99**	**0.95**	**0.97**	**0.99**	**0.98**	**0.88**	**0.99**	**0.96**	**0.98**
	MLP	0.66	0.96	0.94	0.95	0.84	0.89	0.95	0.92	0.62	0.99	0.67	0.8
	RF	0.51	0.5	0.95	0.66	0.26	0.27	0.86	0.41	0.44	0.93	0.49	0.64

situation. This effect is accentuated in the case of JAZ, whose measures in the imbalanced scenario are, in general, the lowest among the three genres. The results of SVM obtained from both SMOTE and RUS are the highest and very similar, what suggests, for this problem, the use of RUS because it significantly reduces the complexity of the training set. Taking into account TPr, Prec and Fm, which only considers MIDI files with melodic track, RUS+SVM achieved results greater than 95%, most of them being greater or equal to 0.99%.

3.4 Experiment II

The second experiment copes with the same task of the first one but, in this case, the training set contains samples of all music genres. The training set is ALL200 (see Sect. 3.2) and the test sets are again CLA, JAZ and KAR. The results of this experiment are shown in Table 4.

Like in the first experiment, classification results based on balanced training sets prevail over those obtained from the original training set. Besides, most of classifiers clearly improve their behaviours in the previous series. This effect is highlighted in the case of RF due to its poor previous results. On the contrary, the improvement of SVM is negligible because its previous results were very high. However, RUS+SVM remains as the best choice and its results slightly improve those reported in previous works [2, 3], although these are obtained with less features in an imbalanced scenario.

4 Conclusions and Future Work

This paper deals with the problem of identifying the melodic track in a MIDI file within its imbalanced context. This task is supported by a primary decision problem consisting of the classification of tracks either in the melody or in the accompaniment class. The higher amount of the latter with respect to the

Table 4. Averaged results of the Experiment II

Training estrategy	*Classifier*	*Train:* ALL200 *Test:* CLA				*Train:* ALL200 *Test:* JAZ				*Train:* ALL200 *Test:* KAR			
		Acc	*TPr*	*Prec*	*Fm*	*Acc*	*TPr*	*Prec*	*Fm*	*Acc*	*TPr*	*Prec*	*Fm*
Original	*1-NN*	0.64	0.88	0.97	0.92	0.89	0.91	0.99	0.95	0.73	0.92	0.84	0.88
	5-NN	0.66	0.87	0.97	0.92	0.91	0.92	1	0.96	0.63	0.88	0.74	0.81
	SVM	0.63	0.99	0.99	0.99	0.89	0.9	1	0.94	0.85	0.95	0.96	0.96
	MLP	0.61	1	0.94	0.97	0.87	0.88	0.99	0.93	0.53	0.75	0.69	0.72
	RF	0.63	0.82	0.9	0.86	0.71	0.73	0.97	0.83	0.56	0.76	0.73	0.75
SMOTE	*1-NN*	0.67	0.94	0.97	0.96	0.92	0.93	0.99	0.96	0.82	0.98	0.91	0.94
	5-NN	0.68	0.98	0.98	0.98	0.93	0.95	0.99	0.97	0.77	0.99	0.84	0.91
	SVM	0.64	1	0.99	0.99	0.93	0.94	1	0.97	0.91	1	0.99	0.99
	MLP	0.6	0.84	0.96	0.89	0.89	0.9	0.99	0.94	0.59	0.8	0.73	0.76
	RF	0.64	0.91	0.93	0.92	0.82	0.84	0.98	0.9	0.73	0.9	0.85	0.88
RUS	*1-NN*	0.7	0.98	1	0.99	0.93	0.94	0.99	0.97	0.84	0.99	0.92	0.95
	5-NN	0.67	0.99	0.9	0.8	0.94	0.95	0.99	0.97	0.71	1	0.77	0.87
	SVM	**0.64**	**1**	**0.99**	**0.99**	**0.94**	**0.95**	**0.99**	**0.97**	**0.91**	**1**	**0.99**	**1**
	MLP	0.63	0.99	0.95	0.97	0.91	0.93	0.98	0.96	0.67	0.96	0.74	0.84
	RF	0.62	0.92	0.94	0.93	0.86	0.89	0.98	0.93	0.75	0.97	0.82	0.89

melody tracks defines a two-class imbalanced problem. Unlike previous related works, imbalance is managed in the experiments by resampling before learning and several classifiers are used to draw the conclusions. Experiments study the melodic track identification within a music genre depending on the inclusion or not of samples of the same music style in the training set, both in balanced and imbalanced contexts. Most of results obtained from resampled training sets were significantly better than those derived from the corresponding original imbalanced training set. The best solution based on SVM provides high results which are independent of the music genres using for training and test.

Future lines of works could involve feature selection or extraction, and the segmentation of tracks in pieces that better match with the melody along with the corresponding labelling at piece level. This approach can be more suitable when the melody line moves across tracks.

References

1. Shen, H.C., Lee, C.: Whistle for music: using melody transcription and approximate string matching for content-based query over a midi database. Multimedia Tools Appl. **35**(3) (2007) 259–283
2. Rizo, D., Ponce de León, P., Pérez-Sancho, C., Pertusa, A., Iñesta, J.: A pattern recognition approach for melody track selection in midi files. In: Proc. of the 7th ISMIR, Victoria (Canada) (2006) 61–66
3. Rizo, D., Ponce de León, P., Pertusa, A., Iñesta, J.: Melodic track identification in midi files. In: Proc. of the 19th Int. FLAIRS Conf., AAAI Press (2006)
4. Madsen, S.T., Widmer, G.: Towards a computational model of melody identification in polyphonic music. In: IJCAI. (2007) 459–464
5. Habrard, A., Iñesta, J.M., Rizo, D., Sebban, M.: Melody recognition with learned edit distances. Lecture Notes in Computer Science **5342** (2008) 86–96
6. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: Synthetic minority over-sampling technique. J. Artif. Intell. Res. (JAIR) **16** (2002) 321–357

7. Kotsiantis, S.: Mixture of expert agents for handling imbalanced data sets. Annals of Mathematics, Computing & TeleInformatics **1** (2003) 46–55
8. Provost, F., Fawcett, T.: Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions. In: Proc. of the 3rd ACM SIGKDD. (1997) 43–48