On the suitability of combining feature selection and resampling to manage data complexity *

Raúl Martín and Ramón A. Mollineda

Institute of New Imaging Technologies
Universitat Jaume I, Castellón, Spain
{martinr, mollined}@uji.es

Abstract. The effectiveness of a learning task depends on data complexity (class overlap, class imbalance, irrelevant features, etc.). When more than one complexity factor appears, two or more preprocessing techniques should be applied. Nevertheless, no much effort has been devoted to investigate the importance of the order in which they can be used. This paper focuses on the joint use of feature reduction and balancing techniques, and studies which could be the application order that leads to the best classification results. This analysis was made on a specific problem whose aim was to identify the melodic track given a MIDI file. Several experiments were performed from different imbalanced 38-dimensional training sets with many more accompaniment tracks than melodic tracks, and where features were aggregated without any correlation study. Results showed that the most effective combination was the ordered use of resampling and feature reduction techniques.

Keywords : Data complexity, feature reduction, class imbalance problem, melody finding, music information retrieval.

1 Introduction

Supervised classification methods are based on the inference of a decision boundary from a set of training samples. The quality of the classifier performance should be affected by the merits and shortcomings of the algorithm, and by the intrinsic difficulty of learning from those samples (*data complexity*) [1]. Some sources of data complexity are class overlap, irrelevant and redundant features, noisy samples, class imbalance, low ratios of the sample size to dimensionality, among others. These challenges are often managed before learning by means of preprocessing techniques in order to improve the generalization power of the training data. When two or more of these problems coincide, the original training set needs to be many times preprocessed, but in which order should the preprocessing techniques be applied?

* This research was partially supported by the Spanish Ministry of Innovation and Science under projects Consolider Ingenio 2010 CSD2007-00018 and DPI2006-15542, and by the FPI grant PREDOC/2008/04 from the Universitat Jaume I. We would also like to thank the *PRAI-UA Group* at the University of Alicante for providing us with the datasets used in this paper.

Little effort has been made to analize the relevance of the application order of several preprocessing techniques. A related paper is [2] where the combined effect of class imbalance and overlapping on classifier performance is analysed. Other studies focus on solutions to the co-occurrence of class imbalance and irrelevant features. A preliminary work [3], within the Web categorization domain, suggests that feature selection techniques are not very appropriate for imbalanced data sets. As a result, a feature selection framework which selects features for positive and negative classes separately is proposed, and the resulting features are explicitly combined. Another work [4] goes a step beyond and applies feature subset selection before balancing the original dataset to predict the protein function from amino acid sequence features. The modified training set feeds a Support Vector Machine (SVM), which gives more accurate results than those provided by the same classifier trained from the original data. Nevertheless, the contrary combination of techniques was not considered, so no conclusions can be drawn about their most suitable application order.

This paper focuses on the joint use of feature reduction and balancing techniques, and studies which is the application order that leads to the best classification results. Experiments are based on the problem of identifying the melodic track of a given MIDI file. This structure is a kind of digital score composed of a set of tracks where usually only one of them is the melody while the remaining tracks contain the accompaniment. This leads to a two-class imbalance problem which has many more accompaniment tracks (majority class) than melodic tracks (minority class). As in the previous work [5], several corpora of MIDI files of different music genres generate a collection of imbalanced training sets with 38 features that were aggregated without any previous study. This configures a suitable scenario to evaluate the goal of this paper.

2 Methodology

An overview of the solution scheme is shown in Figs. 1 and 2 where the five main steps are remarked. The following subsections explain these steps.

2.1 Track Feature Extraction

This step builds vector representations for all tracks of the MIDI files included in both the training and test corpora. As a result, as can be seen in Fig. 1, two related sets of track vectors are obtained for training (TRA) and testing purposes ($TEST$). Tracks are described using 38 features [5] and the class label (melody or accompaniment). These features summarize the musical content of each track by measuring some aspects such as the pitch, duration and syncopation of notes, intervals between notes, duration and importance of the rests, and so on.

2.2 Resampling

The original training set (TRA) leads to a two-class imbalance problem because it contains many less melodic tracks than accompaniment tracks. One way to

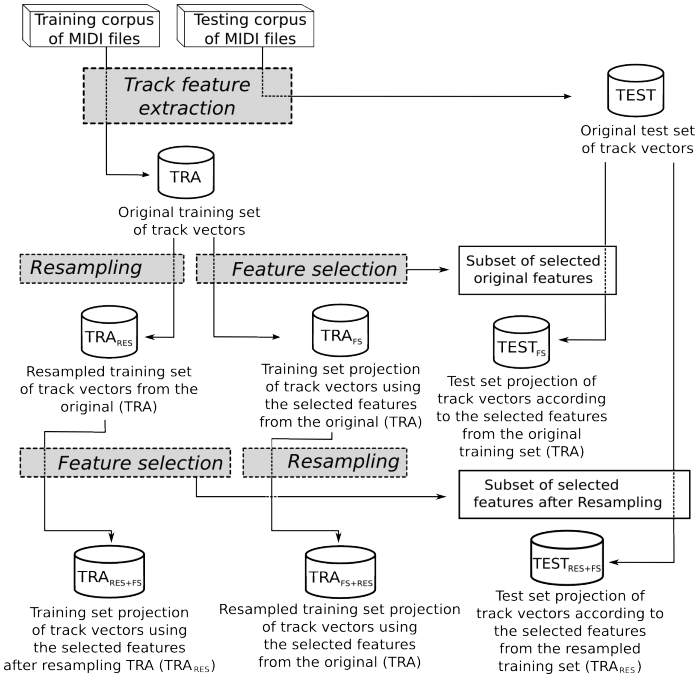

Fig. 1. Generation of datasets used in this work

deal with imbalance is to resample the original TRA either by over-sampling the minority class (melody track) or by under-sampling the majority class (accompaniment track) until the class sizes are similar. Considering that the complexity of the original TRA will be managed by joint preprocessing both imbalance and feature space, Fig. 1 shows the application of resampling methods before (TRA_{RES}) and after (TRA_{FS+RES}) using feature reduction techniques.

In this work two resampling methods have been used: *Synthetic Minority Over-sampling TEchnique (SMOTE)* [6] for over-sampling and *Random Under-Sampling (RUS)* [7] for under-sampling the training set. *SMOTE* is a method that generates new synthetic samples in the minority class. For each sample of this class, this algorithm computes the k intra-class nearest neighbours, in this paper $k = 5$, and several new instances are created by interpolating the focused sample and some of its neighbours randomly selected. Its major drawback is an increase in the computational cost. In contrast, *RUS* 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.

2.3 Feature reduction

Feature reduction [8] is an essential data preprocessing step prior to use a classifier. This process consists of reducing the dimensionality of data with the aim

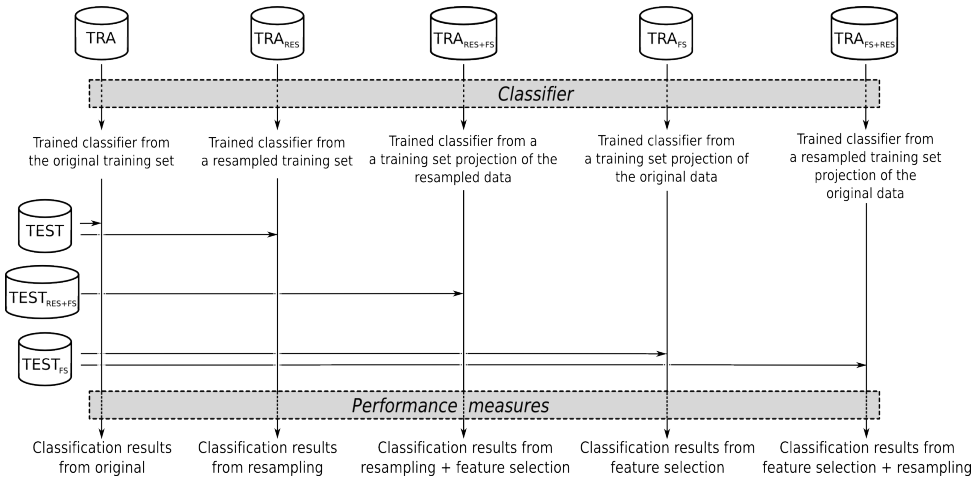

Fig. 2. Experimental design using the datasets introduced in Fig. 1

of allowing classifiers to operate faster and, in general, more effectively. Feature reduction methods can be separated according to many criteria, for example: i) selection versus extraction of discriminant features and ii) supervised versus unsupervised. In this work, a supervised selection technique and an unsupervised extraction method have been used. The former is *Correlation-based Feature Selection (CFS)* [8], and the latter is *Principal Components Analysis (PCA)* [9].

CFS ranks feature subsets according to the degree of redundancy among the features. It searches subsets of features that are individually well correlated with the class but have low inter-correlation. *PCA* consists of a transformation of the original features into a smaller number of uncorrelated new features called principal components. PCA can be used for dimensionality reduction by retaining those principal components, from most to least importance, that accounts for a given proportion of the variance (in this work, the 95%).

As the complexity of the original TRA will be managed by joint preprocessing both feature space and imbalance, Fig. 1 shows the application of feature reduction techniques before (TRA_{FS}) and after (TRA_{RES+FS}) using resampling methods. Unlike the resampling methods which only involve training sets, the use of new feature spaces produces the projection of test sets on them giving rise to two new derived test sets: $TEST_{FS}$ and $TEST_{RES+FS}$.

2.4 Classifiers

The aim 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 identification of the melodic track of a MIDI file is carried out based on results at track level. As regards the sets used for training and testing

(see Fig. 2), the effectiveness of the detection of the melodic track at MIDI file level is evaluated. A detailed description of this process is as follows:

Track level

1. Given a track, a classifier assigns probabilities of membership of 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 a *hit* if
   - *True Positive*: the selected track is originally labelled as melody, or
   - *True Negative*: in a file with no melodic track, no track is classified as melody. However, all MIDI files used in the experiments have a well-defined melodic track, thus no True Negative cases occur
3. The decision is considered a *miss* if
   - *False Positive*: the selected melody track is originally labelled as accompaniment, or
   - *False Negative*: in a file with a melodic track, all its tracks have been discarded or classified as accompaniment

The base classifiers used at track level are 1-NN and SVM because of their diversity in terms of the geometry of their decision boundaries.

2.5 Performance Measures in Class Imbalance 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, 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 [10]. These shortcomings have motivated a search for new measures, for example: (i) *True positive rate* (or *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* which combines TPr and Precision giving a global vision focused on the positive class, $F\text{-}measure = (2 * TPr * Precision)/(TPr + Precision)$. Other well-known measures like AUC and *Gmean* can not be used due to their strong dependence on the *True negative rate* which is zero in our experiments (see Sect. 2.4).

Table 1. Corpora used in the experiments

CorpusID	*Music Genre*	*Midi Files*	*Tracks*	
			non-melody	*melody*
CL200	Classical	198	489	198
JZ200	Jazz	197	561	197
KR200	Popular	159	1171	159
CLA	Classical	84	284	84
JAZ	Jazz	1006	3131	1006
KAR	Popular	1247	9416	1247

3 Experimental Results

3.1 Datasets

Experiments involve six datasets of track vectors obtained from the same number of corpora of MIDI files created in [11] (see Table 1 for details). 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). From each corpus, a corresponding dataset of 38-dimensional track vectors is available (see Sect. 2.1) where each vector has been manually labelled as melody or non-melody by a trained musicologist.

These corpora can be divided into two groups with regard to their data complexity and also due to their sizes. A first group includes CL200, JZ200 and KR200 because they have in common a similar number of MIDI files (close to 200). Moreover, most of them have well-defined melodic tracks which make them suitable for training purposes. In contrast, CLA, JAZ and KAR are more heterogeneous corpora and, consequently, lead to more challenging tasks [11].

3.2 Experimental Design

In the following experiment, each classifier is trained with samples from two music genres taken from CL200, JZ200 and KR200, and is tested with samples of the remaining style taken from CLA, JAZ and KAR. 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). The rationale behind these data partitions is to maximize the independence between the training and the test sets, and to find out whether the effectiveness of music track identification depends on the music genres using for training and testing.

The main aim of the experiments is to study the importance of the order of applying two preprocessing techniques, resampling and feature reduction, to jointly reduce the complexity of training datasets. As it was seen in Sect. 2.2 and Sect. 2.3, two different algorithms of each preprocessing technique have been used: SMOTE and RUS for resampling, and CFS and PCA for feature reduction. The experimental design is based on the 2×2 crossing of these four methods considering the two possible sequences of each specific pair, producing

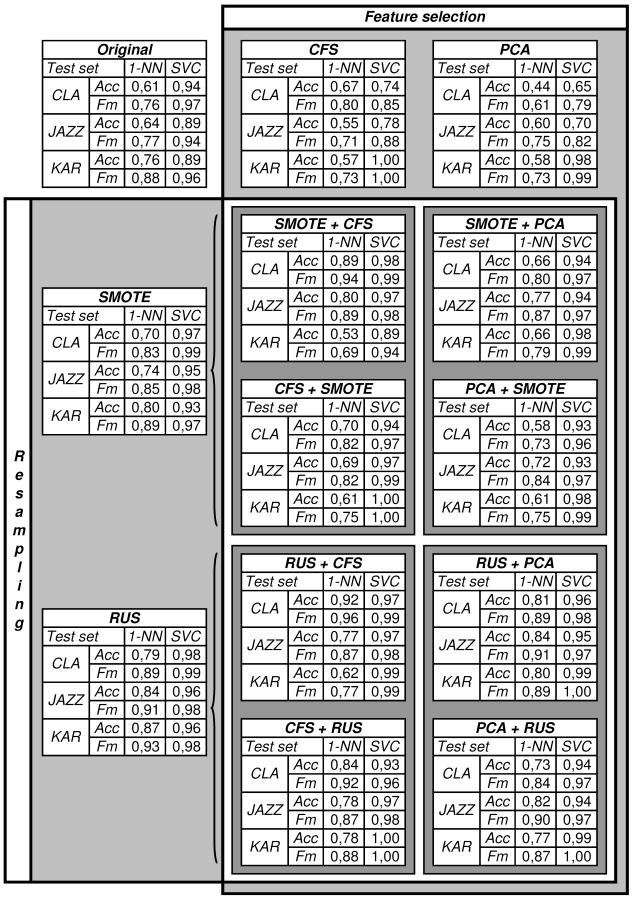

Fig. 3. Averaged results of experiments

eight different combinations. Fig. 3 shows all the results. Apart from comparisons among classification performances obtained from these ordered combinations of methods, these results are also compared with those provided by the only use of each particular technique and from the original imbalanced training set.

Due to the random behaviour of SMOTE and RUS, each experiment that involves these techniques was performed 10 times and their results were averaged. The classification results are given in terms of Accuracy (Acc) and Fmeasure (Fm), that were computed taking into account only MIDI files with at least one melody track in contrast with previous related works [5, 11] where Accuracy was computed including also MIDI files without any melody track.

The implementations of the classifiers and the feature reduction tecniques used are those included in the WEKA toolkit [1] with their default parameters.

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

3.3 Analysis of Results

The results presented in Fig. 3 are analysed in the following three ways organized from a low to a high level of detail. Each comparative analysis involves all the possible cases obtained from the combination of a classifier (see Sect.2.4), a data partition (see 3.2) and two performance measures (see 2.5). The percentage of favourable cases is computed and used as an index to explain the usefulness of a preprocessing technique or a combination of some of them.

- **Level A** (the highest level of analysis)
  - ***A.1 Resampling versus Original.*** The classification results obtained from the use of resampled training sets are compared with those provided by the corresponding original training sets (see the first column in Fig. 3). In the 100% of the cases, the former results were higher than those of the latter ones.
  - ***A.2 Feature reduction versus Original.*** The classification results obtained from the training and test sets whose dimensionality has been reduced by some feature reduction technique are compared with the results of the original 38-dimensional task (see the first row in Fig. 3). Only in the 25% of the cases, the former results were higher than those of the latter ones, so the fact of reducing dimensionality alone tends to deteriorate results.
- **Level B** (a middle level of analysis)
  - ***B.1 Resampling+Feature reduction versus Resampling.*** The classification results obtained from the joint application of resampling and feature reduction in this order, are compared with the results provided by the use of resampling alone. In this analysis, the boxes with titles in the forms *SMOTE+** and *RUS+** are compared with the boxes titled *SMOTE* and *RUS* respectively. In the 52% of cases, the joint use of resampling and features reduction produced performance measures equal to or greater than those obtained from the only application of resampling. Therefore, the ordered use of both techniques does not seem to guarantee better results with respect to resampling, which already produced a massive improvement on the original results (see level A.1).
  - ***B.2 Feature reduction+Resampling versus Feature reduction.*** The classification results obtained from the joint application of feature reduction and resampling in this order, are compared with the results provided by the use of feature reduction alone. In this study, the boxes with titles in the forms *CFS+** and *PCA+** are compared with the boxes titled *CFS* and *PCA* respectively. In the 97% of cases, the joint use of features reduction and resampling produced performance measures equal to or greater than those obtained from the only application of feature reduction. However, this result should be carefully considered because, as can be seen in the level A.2, the plain selection of features does not seem effective with respect to the original results.
- **Level C** (the lowest level of analysis)

- ***C.1 Resampling+Feature reduction versus Feature reduction + Resampling.*** The results of the two ways of combining resampling and feature reduction analysed in level B are compared to find out which order is more effective. It involves the 8 central boxes with titles made of two acronyms. Each box is contrasted with the one that has its reverse title, for example, *SMOTE+CFS* versus *CFS+SMOTE*. Considering the four comparable pairs of boxes, in the 79% of cases, Resampling+Feature reduction gave better results than the contrary combination.
- ***C.2 SMOTE versus RUS.*** The two resampling techniques used in the experiments are compared (SMOTE and RUS). Each box that involves a particular resampling technique is compared with the corresponding one, i.e., that with the same title pattern as a function of the other resampling method. Considering the five comparable pairs of boxes, in the 90% of cases, results of RUS are equal to or outperform the results of SMOTE. From this analysis, RUS seems to be more appropriate than SMOTE to manage the task complexity. In addition, RUS reduce the size of the training set and hence the time to build a classifier.
- ***C.3 CFS versus PCA.*** The two feature reduction methods used in the experiments are compared (CFS and PCA). Each box that involves a particular feature reduction technique is compared with the corresponding one, i.e., that with the same title pattern as a function of the other feature reduction method. Considering the five comparable pairs of boxes, in the 70% of cases, results of CFS are equal to or outperform results of PCA. From this analysis, CFS seems to be more appropriate than PCA to deal with the problem. Besides, CFS reduces significantly the data dimensionality by choosing specific features, while PCA requires the 38 original features before to transform them into principal components.
- ***C.4 1-NN versus SVC.*** The two classifiers used in the experiments are compared (1-NN and SVC). This analysis involves all the boxes where each 1-NN result is compared with the corresponding internal SVC result. In the 100% of cases SVC outperforms 1-NN results. SVC appears to be more robust than 1-NN regarding to the genres used for training and testing, and also, considering both imbalanced and balanced contexts.

Taking into account all the previous analysis, the best solution is that where training data are preprocessed by RUS and CFS in this order, and test data are filtered by CFS and classified with SVM. In general, the ordered use of resampling and feature reduction leads to better results than the reverse combination. The posterior use of CFS produced a drastic reduction of the number of features, from 38 to an average of 11, while keeping or improving the performance results provided by the plain application of RUS. Besides, this combination of techniques obtained high and similar results for the three music genres tested, so it seems to be independent of the music style of the training samples.

4 Conclusions

This paper studies the effectiveness of the joint application of two preprocessing techniques, resampling and feature reduction, considering their order of use. They were validated over the problem of identifying the melodic track of MIDI files belonging to three music genres: classical, jazz and popular. The higher number of accompaniment tracks compared to the number of melody tracks defines a two-class imbalance problem what, along with the wide set of primary features, explains the combined use of resampling and feature reduction methods.

As in [4], our experiments show that benefits associated to resample training data are greater than those related to the use of feature reduction. However, unlike [4], we consider all the possible ways in which they can be applied, either individually or jointly in both directions. It supports the thesis suggested by [3], that the application of feature reduction methods on imbalanced data has a low efectiveness. The most effective solution was the joint use of resampling and feature selection methods in this order because, apart from sharing the best classification results with the application of resampling alone, this approach significantly reduced the data dimensionality. Besides, these good results are similar for the three music genres tested which could indicate the independence of the solution from the music style used for training.

References

1. Basu, M., Ho, T.: Data Complexity in Pattern Recognition. Springer-Verlag New York, Inc., Secaucus, NJ, USA (2006)
2. García, V., Alejo, R., Sánchez, J.S., Sotoca, J.M., Mollineda, R.A.: Combined effects of class imbalance and class overlap on instance-based classification. In: IDEAL. (2006) 371–378
3. Zheng, Z., Wu, X., Srihari, R.: Feature selection for text categorization on imbalanced data. SIGKDD Explor. Newsl. **6**(1) (2004) 80–89
4. Al-shahib, A., Breitling, R., Gilbert, D.: Feature selection and the class imbalance problem in predicting protein function from sequence. Applied Bioinf. **4** (2005)
5. Martín, R., Mollineda, R., García, V.: Melodic track identification in midi files considering the imbalanced context. In: 4th IbPRIA, Póvoa de Varzim (2009)
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. Hall, M.: Correlation-based feature subset selection for machine learning (1999)
9. Jolliffe, I.: Principal Component Analysis. Second edn. Springer (2002)
10. 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
11. 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