MIREX 2011: AUDIO TAG CLASSIFICATION USING FEATURE TRIMMING AND GRID SEARCH FOR SVM

Simon Bourguigne **Pablo Daniel Agüero**

Facultad de Ingenieria,
Universidad Nacional de Mar del Plata
Mar del Plata, Argentina
sbourguigne@fi.mdp.edu.ar

ABSTRACT

In our submission we use a straight forward method for the task of audio tag classification. This extended abstract, briefly describes the features used and the classification method.

1. INTRODUCTION

In this paper we present our system for Audio Tag Classification for the MIREX 2011 competition. The proposed system takes a very simple approach. It uses the standard procedure of frame-level audio feature extraction and posterior aggregation without considering it's time structure, that is to say, the bag-of-frames approach. Once these features are obtained they are fed to a number of SVMs binary-classifiers that equate with the number of tags. One of the difficulties lies at optimizing each classifier for better performance. In orther to do so, we iterate over different values for the SVM parameters doing a simple grid search and using a "feature trimming" algorithm until the stop criteria is met for each classifier. After each optimization iteration, the decision of which classifier works better is taken using an inner cross-validation for each given fold.

2. FEATURE EXTRACTION

The feature extraction process is done on a frame-based fashion using an in-house feature extractor named Ursula. We use 50ms hamming windows and a hop-size of 25ms. After the feature extraction each frame has a feature array associated to it. This is followed by grouping contiguous frames into a 1sec texture window and for each of the later we aggregate the features using mean and standard deviation values. Later on, we further aggregate these values computing the same statistics over them. This produces mean-mean, mean-std, std-mean, std-std values that we use to aggregate the whole set of features for the specific sound clip. Once the aggregation is done, each numeric feature will have four times it's length, ie mfcc will have 72 numeric values instead of 13. A detailed description of the features used is shown in Table 1.

Feature Description	Dim
lpcc	72
mfcc	52
lsp	72
spectral flatness	96
spectral crest factor	96
spectral flux	4
spectral decrease	4
loudness [5]	4
roll-off at 95%	4
zero crossing rate	4
formant band energy (250-2500Hz)	4
odd to even energy ratio	4
harmonic coefficient [2]	4
beat histogram (non-aggregated)	9

Table 1. The audio features used for classification.

3. THE CLASSIFICATION METHOD

In this section we present our classification method. The tags will be assumed to be independent, that means that only one binary classifier will be used for each of them. The chosen classifiers are SVMs with linear kernels, which have only two parameters to be optimized. A second step to boost the performance consists on dynamically trimming the feature set for each classifier. We then re-train each classifier with the remaining features.

3.1 Grid Search

In a linear SVM there are two parameters to be set, $C > 0$ is the penalty parameter of the error term of the classifier, and W is a penalty of the wrong classification for positive (+1) and negative (-1) examples. To find the optimal set of these parameters we perform a simple grid search. We define two arrays for different values for C and W. Later on the various pairs of (C; W) values are tried and the one with the best inner cross-validation f-measure is picked [4].

This document is licensed under the Creative Commons Attribution-Noncommercial-Share Alike 3.0 License.
http://creativecommons.org/licenses/by-nc-sa/3.0/
© 2011 The Authors.

3.2 Feature Trimming

In many supervised learning problems, feature selection is important for a variety of reasons: generalization performance, runtime requirements, and constraints and interpretational issues imposed by the problem itself. Support Vector Machines are not an exception. It is important to select a subset of features while preserving or improving the discriminative ability of a classifier. As a brute force search of all possible features is a combinatorial problem, it is necessary to take into account both the quality of solution and the computational cost of any given algorithm. Greedy methods are a simple heuristic solution to such problem. The number of features included in the feature vector grows step by step, each stage taking the results of the previous stage into account. The greedy algorithm begins with an empty initial feature vector, and in each stage appends an additional feature that contributes to a better global performance of the classifier. A different approach is taken by François [3]; instead of "eating" features, they train with all of them and in each iteration they "spit" the most useless one, after that they re-train with the new set of features and keep on spitting until they stop according to some predefined criteria. They called this the spitting method. As it has been proposed by [1] we use a Lazy Spitting Algorithm, which preserves the spitting behavior, but features are only marked to be spat. Later on the classifier is re-optimized when all marked features are finally deleted (or spat). This algorithm named lazy spitting method, begins with a full feature vector, and in just one stage deletes all the features that once removed do not impact in the global result of the classifier. For a detailed description of the whole optimization process see Figure 1.

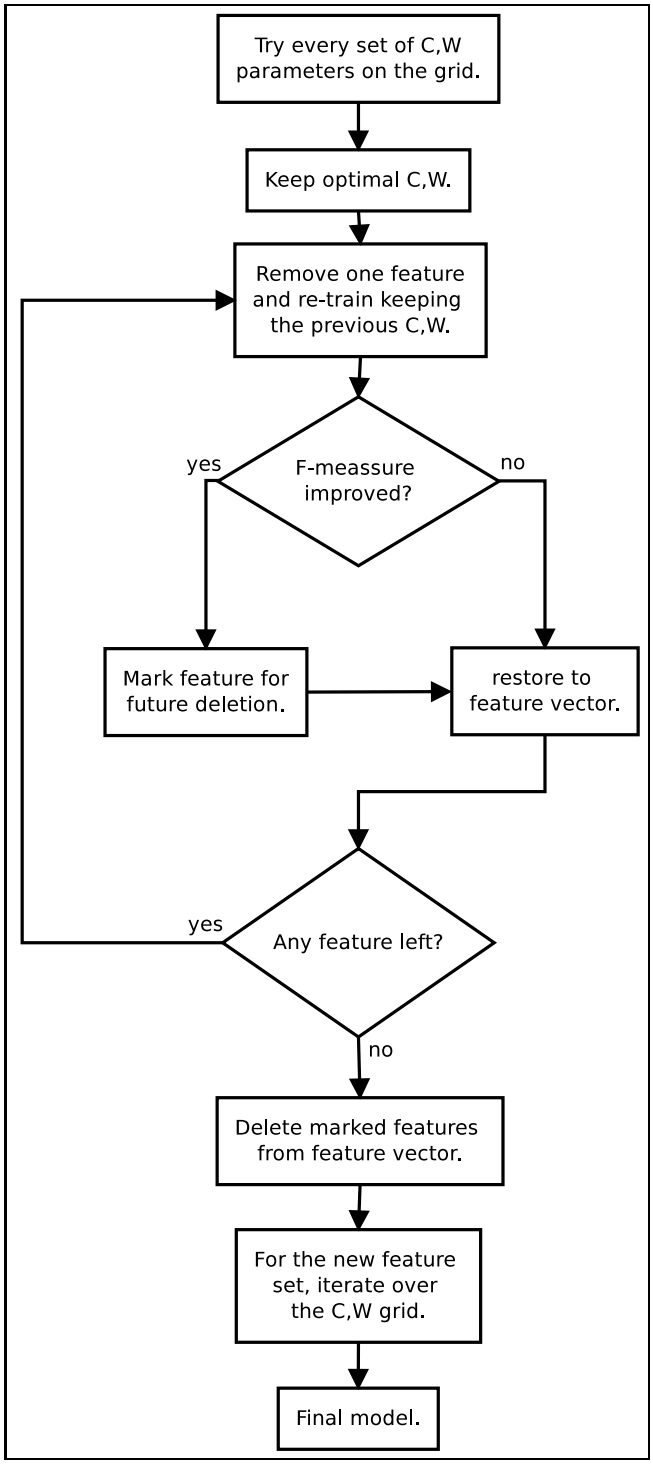

Figure 1. Flow chart description of the proposed method.

4. REFERENCES

[1] S. Bourguigne, P.D. Agüero, J.C. Tulli, E.L. Gonzales and A.J. Uriz : "Automatic Selection of Acoustic Features using a Lazy Spitting Method" *Proceedings of AST*, Cordoba, Argentina, 2011.

[2] W. Chou and L. Gi : "Robust singing detection in speech/music discriminator design," *Proceedings of ICASSP*, Salt Lake, 2001.

[3] H. Francois and O. Boeffard : "The greedy algorithm and its application to the construction of a continuous speech database," *Proceedings of LREC-2002*, Vol. 5, pp. 1420–1426, 2002.

[4] C.-W. Hsu, C.-C. Chang, C.-J. Lin : "A practical guide to support vector classification," *Technical report*, Department of Computer Science, National Taiwan University. July, 2003.

[5] S. Streich : "Music Complexity: a multi-faceted description of audio content," *Ph.D. Dissertation*, UPF, Barcelona, 2007.