AUDIO ARTIST IDENTIFICATION MIREX 2005

Vitor Soares
FEUP - Faculty of Engineering
University of Porto
Rua Dr. Roberto Frias, s/n 4200-465
Porto, Portugal
vsoares@fe.up.pt

ABSTRACT

On the tasks of audio recognition with various simultaneous semantic contexts (vocals, music signals and environmental sounds), the already well known problem begins in the phase of selecting features, which can be illustrated by the “blanket concept”: "push it on one side, uncover on the other".

In light of this problem, motivation is sought aiming at the development and extension of various methods for feature selection. We propose a Multivariate Time Series datasets method, which we call taxoDynamic, which dynamically adjusts the subset of features at each node of the taxonomy.

1 ALGORITHM

A wide group of more than 400 signal processing features is considered including some transformations to the original features, such as derivatives and correlation between features. In the proposed algorithm of feature selection, the "wrapper" methodology was followed, which uses its own classifier to classify the quality of the features. In the first stage, we perform the feature selection class pair wise in the sense that a different subset of relevant features is fetched for every possible pair of classes. Part 1 of the algorithm executes a selection of features autonomously, not requiring a preset number of important features.

Algorithm part 1: Rank of group of features\( F_{\text{sorted}} \) using the inflection point adaptable to the classifier.

Output: \( \text{dfilter1} \) \{number of sub-optimal features\}

Requirements: Set \( F_{\text{sorted}} \) \{group of candidate features\};
1: From \( F_{\text{sorted}} \) the \( Cm \) feature is selected with less conditional probability density;
2: Multid. Gaussian, SVM \( \leftrightarrow \) Train\( _{iter(i)} \) with \( E \);
3: Fetch the accuracy of each model on iteration \( i \);
4: Comparison of the errorRate iteration \( i \) with iteration \( i-1 \);
5: If (errorRate\( _{i} \) < errorRate\( _{i-1} \)) do
   Add \( Cm \) to \( E \); remove \( Cm \) from \( F_{\text{sorted}} \) and return to step 1 Else
   Find dfilter1.

Algorithm part 2 Compute the correlation coefficient matrix and perform the vectorization using the upper triangle

Requirements: MTS dataset, \( B \) \{the number of chunks of observations considered in the data set\}, \( k \) \{number of variables found in Algorithm 1\};
1: for \( b = 1 \) to \( B \) do
2:   for \( i = 1 \) to \( k \) do
3:       \( C_{(i,j)} \leftrightarrow \) correlation coef. matrix of iTth MTS item
4:   endfor
5: endfor
6: \( \text{Cvectorized} \leftrightarrow [] \);
7: for \( i = 1 \) to \( k \) do
8:   \text{Cvectorized} \{\text{Cvectorized} \( C[i, (i + 1) : k] \)};
9: endfor

The part 3 algorithm aims to filter secondly the features, and assumes the knowledge of how strongly one characteristic implies the other in a certain artist. The added information to the selected original subset of features refines the model of each artist.

Algorithm part 3 Rank correlation features

Requirements: \text{Cvectorized} \{vector with correlation coefficients\}, \( \{\mu, \sigma^2\} \) of the gaussian model.
1: \( \mu, \sigma^2 \leftrightarrow \) Train Gaussian Model for each \text{Cvectorized} value;
2: Rank quality of features based on the best ratios \( (\mu_{\text{modelA}} - \mu_{\text{modelB}}) / \sigma^2 \);
3: \( D_{corr} \leftrightarrow \) Find rank dfilter2 using the Algorithm 1 and the subset of features found in 2.

Training Algorithm

Requirements: Dataset, \( N \) \{the number of audio classes involved in the classification scheme\}, Set \( F \leftrightarrow \{X_1, ..., X_D\} \) \{features extracted from the audio\};
1: for \( m = 1 \) to \( D \) do
   for \( c = 1 \) to \( N \) do
     Pererror\( _{c} \) \leftrightarrow \) Conditional PDE between \( c \) and \( c+1 \) endfor
   endfor
Set \( F_{\text{sorted}} \leftrightarrow \) ranks (Pererror);
Execute Algorithm part 1;
Execute Algorithm part 2;
Execute Algorithm part 3;
2: \( \text{dselected} \leftrightarrow \) Merge the group of features dfilter1 found in Algorithm 1 with the group \( D_{corr} \) of correlation features found in algorithm 3;
3: dtopfeatures \leftrightarrow \) calculate the mean of dselected, found in 2 throughout the \( N!(N-2)!2! \) combinations;
4: dfinalfeatset \leftrightarrow \) Calculate the non-normalized histogram of the top features
5: Train GMM with \( \text{dfinalfeatset} \) for each pair.