CLASSIFICATION OF ACOUSTIC EVENTS USING SVM-BASED CLUSTERING SCHEMES

Andrey Temko[*], Climent Nadeu

TALP Research Center, Universitat Politècnica de Catalunya, Campus Nord, Edifici D5, Jordi Girona 1-3, 08034 Barcelona, Spain

ABSTRACT

Acoustic events produced in controlled environments may carry information useful for perceptually aware interfaces. In this paper we focus on the problem of classifying 16 types of meeting-room acoustic events. First of all, we have defined the events and gathered a sound database. Then, several classifiers based on support vector machines (SVM) are developed using confusion matrix based clustering schemes to deal with the multi-class problem. Also, several sets of acoustic features are defined and used in the classification tests. In the experiments, the developed SVM-based classifiers are compared with an already reported binary tree scheme and with their correlative Gaussian mixture model (GMM) classifiers. The best results are obtained with a tree SVM-based classifier that may use a different feature set at each node. With it, a 31.5% relative average error reduction is obtained with respect to the best result from a conventional binary tree scheme.

Keywords: acoustic event classification, support vector machines, clustering

1. INTRODUCTION

Activity detection and description is a key functionality of perceptually aware interfaces working in collaborative human communication environments like meeting-rooms or classrooms. In such types of environments the human activity is reflected in a rich variety of acoustic events, either produced by the human body or by objects handled by humans, so auditory scene analysis **[1]** by computer may help to detect and describe human activity as well as to increase the robustness of automatic speech recognition systems.

Acoustic event classification (AEC) is one of the problems considered by computational auditory scene analysis. Indeed, speech usually is the most informative acoustic event, but other kind of sounds may also carry useful information. For example: clapping or laughing inside a speech, a strong yawn in the middle of a lecture, a chair moving or door noise when the meeting has just started. When trying to deal with the problem of AEC in the framework of the CHIL project **[2]**, we soon noticed that reported works are scarce. Actually, classification of sounds has usually been carried out so far to segment digital audio streams using a limited number of categories, like music/speech/silence/environmental sound (see e.g.**[3]**). Usually those

[*] Corresponding author. Tel: +34-93-401-1627, Fax: +34-93-401-6447. Email address: temko@gps.tsc.upc.es (A.Temko)

works are intended to indexing and retrieval of multimedia documents. Audio retrieval is also the objective in **[4]**, using a relatively high number of sound classes (13) and without explicit segmentation, and also in **[5]**, where animal sounds are retrieved using natural language sentences. On the other hand, several works have been devoted to the problem of detection of single sounds, like laughter detection in **[6];** or **[7]**, where the authors built systems for detecting/classifying several sounds independently from each other. The AEC problem has also been considered in the framework of speech recognition in **[8]**. Aiming to improve the robustness of the ASR system, the authors in **[8]** dealt with the problem of classifying 92 types of isolated sounds that had been collected in an anechoic room, the RWCP sound scene database **[9]**. Some more information about the history and the state of the art in the problem of audio classification can be found in **[10]**.

In this paper we focus on acoustic events that may take place in meeting-rooms or classrooms and on the preliminary task of classifying isolated sounds. The number of sounds encountered in such environments may be large, but in this initial work we have chosen 16 different acoustic events, including speech and music, and a database has been defined for training and testing. While in **[8]** the authors looked at the problem from the point of view of speech recognition, applying the usual automatic speech recognition strategy (cepstral features, classifier based on Hidden Markov Models (HMM) and Gaussian Mixture Models (GMM)), in our work we consider, develop and compare several feature sets and classification techniques, aiming at finding the ones which are most appropriate for the problem we are dealing with. In this way, not only the parameters that are used in speech recognition to model the short-time spectral envelope of the signals and its time derivatives are considered, but also other perceptual features which may be more fitted to non-speech sounds. Moreover, HMMs require relatively large amount of data to accurately train the models, something that is not realistic in our task, since there are not many collections of meeting recordings and the number of samples of some type of sounds that can be found in them is small.

Recently, the Support Vector Machine (SVM) paradigm has proved highly successful in a number of classification tasks. As a classifier that discriminates the data by creating boundaries between classes rather than estimating class conditional densities, it may need considerably less data to perform accurate classification. In fact, SVMs have already been used for audio classification **[11]** and segmentation **[12]**. In this work we use SVM classifiers and compare them with GMM classifiers.

As SVMs are binary classifiers, some type of strategy must be employed to extend them to the multi-class problem. In **[11]**, the authors used the binary tree classification scheme to cope with several classes. That approach requires a relatively high number of classifiers and classification steps, and the number of classes has to be a power of 2 to get the most benefit from the technique. There are other ways of applying SVMs to the multi-class problem; see **[13]** for a comparison of different methods of multi-class SVM classification. In our work, we propose and develop several variants of a tree clustering technique. Relying on a given set of confusion matrices, that technique chooses the most discriminative partition and feature set at each step of classification, and, unlike the binary tree, works for any number of classes.

Comparative tests have been carried out using the two basic classifiers (GMM and SVM) and a number of classification schemes (binary tree and several clustering alternatives). The effects of using two different regularization parameters of the SVM classifiers to compensate data unbalance, and a confusion matrix based modification of those parameters are also investigated in this work.

The paper is organized as follows. In Section 2 we present the database of gathered sounds. Section 3 describes the features and explains the construction of feature sets. The basic theory of SVM and GMM classification techniques is reviewed in Section 4. The experiments and a discussion of the results are presented in Section 5. Finally, conclusions are given in Section 5.

2. DATABASE

The first problem we had to face when trying to develop a system for classifying acoustic events which take place in a meeting-room environment was the lack of data. As mentioned above, there exists a relatively large database of sounds, the RWCP sound scene database, but only a small part of the sounds included in that database can be considered as usual or at least possible in a meeting room.

The second column of **Table 1** shows the sixteen categories of sounds that were chosen. As can be seen in the third column, only four of them belong to the RWCP database. The other sounds have been found in a large number of websites, except the speech sounds, which were taken from the ShATR Multiple Simultaneous Speaker Corpus **[14]** and include short fragments from both close-talk and omnidirectional microphones. The number of samples is 100 or larger for the sounds taken from the RWCP database, but it is much smaller for a few classes. As shown in the fourth column of **Table 1**, chair moving and yawn events have only 12 samples in the database. The whole database amounts 53 min of audio (942 files).

	Event	Source	Number
1	Chair moving	I	12
2	Clapping	RWCP + I	100+7
3	Cough	I	47
4	Door slam	I	80
5	Keyboard	I	45
6	Laughter	I	26
7	Music	I	38
8	Paper crumple	RWCP	100
9	Paper tear	RWCP	100
10	Pen/pencil handwriting	I	30
11	Liquid pouring	I	40
12	Puncher/Stapler	RWCP	200
13	Sneeze	I	40
14	Sniffing	I	13
15	Speech	ShATR	52
16	Yawn	I	12

Table 1. The sixteen acoustical events considered in our database, including number of samples and their sources (I means Internet).

Indeed both the diversity in the number of samples per class and the small number of samples for some sounds are a challenge for the classifier. And, the fact that sounds were taken from different sources makes the task even more complicated due to the presence of several (at times even unknown) environments and recording conditions.

3. AUDIO FEATURES

The signals from all the sounds in the database presented above were downsampled to 8kHz, normalized to be in the range [-1 1], and partitioned in frames using: frame length=128, overlapping of 50%, and a Hamming window. The silence portions of the signals were removed using an energy threshold.

Three basic types of acoustic feature were considered in this work. Two of them are spectrum envelope representations used in speech/speaker recognition, namely the typical mel-frequency cepstral coefficients (MFCC) plus the frame energy **[15]**, and the recently introduced frequency-filtered band energies (FFBE) **[16]**. Like in speech recognition, they will be considered either alone or together with their first and second time derivatives (the so-called delta and delta-delta features) **[15]**. We consider both types of features because we want to compare their discriminative capability in this application. The third type of features is a small set which includes perceptual features which are not considered in the above feature sets and may be more adequate for some kind of sounds (fundamental frequency and zero crossing rate), and also a reduced representation of the spectral envelope and its time evolution. We will call it perceptual feature set, since it has a more perceptually-oriented profile than the other two.

	Feature set	Content	Size
1	*Perc*	Perceptual features spectral	11
2	*Ceps+der*	E+MFCC+d+dd	39
3	*Ceps*	E+MFCC	13
4	*FF+der*	FFBE+d+dd	39
5	*FF*	FF	13
6	*Perc+ceps+der*	"Perc"+"Ceps+der"	50
7	*Perc+ceps*	"Perc" + "Ceps"	24
8	*Perc+FF+der*	"Perc" + "FF+der"	50
9	*Perc+FF*	"Perc" + "FF"	24

Table 2. . Feature sets that were used in this work, the way they were constructed from the basic acoustic features, and their size. **d** and **dd** denote first and second time derivatives, respectively, **E** means frame energy, and "+" means concatenation of features.

Thus, the acoustic features considered in this work are defined in the following way:

1. Perceptual features

- Short time signal energy, computed frame-by-frame.
- Sub-band energies: 4 subbands equally distributed along 20 mel-scaled logarithmic filter-bank energies (FBE) for each frame.
- Spectral flux: difference of spectrum values between two adjacent frames, for each of the above-defined 4 sub-bands. SF measures the changes of spectrum over time.
- Zero-crossing rate, computed as the number of zero crossings within a frame.
- Fundamental frequency: a simple cepstrum-based method was used to determine it for each frame in the range [70Hz, 500Hz]

2. Cepstral coefficients

12 mel-frequency cepstral coefficients (MFCC) were computed for each frame using 20 mel-scaled spectral bands. The zero-th cepstral coefficient was removed, but the frame energy was added to the set.

3. FF-based spectral parameters

Parameters based on filtering the frequency sequence of log FBEs (FFBE) **[16]**. We have used the usual second-order filter $H(z)=z-z^{-1}$, which implies subtraction of the log FBEs of the two adjacent bands. Before filtering, the sequence of log FBEs along frequency is extended with one zero at each side. In this way, the first and last parameters actually are the energies of the second and the second last sub-bands. That is the reason why the frame energy was not used with these features.

The three above defined types of acoustic features were combined to build the 9 different feature sets shown in **Table 2** which are considered in the experiments reported in Section 5. The mean and standard deviation of those features, estimated by averaging over the whole acoustic event signal, were taken for classification, thus forming one final statistical feature vector per audio event with a number of elements which doubles the length of the acoustic feature set.

4. CLASSIFICATION TECHNIQUES

Two basic classification techniques are considered in this work: Support Vector Machine (SVM) and Gaussian Mixture Model (GMM). The former is based on decision surfaces, and the latter models data with probability distributions. In this section, we will present both approaches, along with the SVM variants that are used in the experiments.

4.1. Support vector machines

The SVM is a discriminative model classification technique that mainly relies on two assumptions. First, transforming data into a high-dimensional space may convert complex classification problems (with complex decision surfaces) into simpler problems that can use linear discriminant functions. Second, SVMs are based on using only those training patterns that are near the decision surface assuming they provide the most useful information for classification.

4.1.1. Construction of a SVM

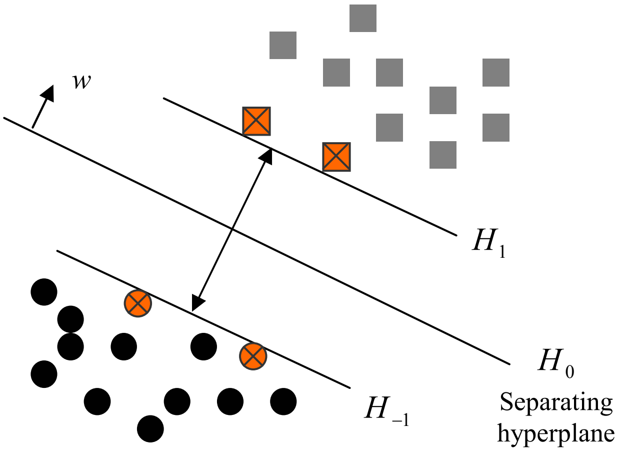


Figure 1. Two-class linear classification. The support vectors are indicated with crosses

Let us assume a typical two-class problem in which the training patterns (vectors) $x_i \in \Re^n$ are linearly separable, as in **[17]**, where the decision surface used to classify a pattern as belonging to one of the two classes is the hyperplane H$_0$. If x is an arbitrary vector ($x \in \Re^n$), we define

$$f(x) = w \cdot x + b \tag{1}$$

where $w \in \Re^n$ and $(\cdot)$ denotes the dot product. H$_0$ is the region of vectors *x* which verify the equation $f(x) = 0$ **[18]**, and H$_1$ and H$_{-1}$ are two hyperplanes parallel to H$_0$, and defined by $f(x) = 1$ and $f(x) = -1$, respectively. The distance separating the H$_1$ and H$_{-1}$ hyperplanes is

$$\frac{2}{\|w\|} \tag{2}$$

and it is called *margin*. The margin must be maximal in order to obtain a classifier that is not much adapted to the training data, i.e. with good generalization characteristics. As we will see, the decision hyperplane H$_0$ directly depends on vectors closest to the two parallel hyperplanes H$_1$ and H$_2$, which are called *support vectors*.

Consider a set of training data vectors $X = \{x_1,...x_L\}$, $x_i \in \Re^n$, and a set of corresponding labels $Y = \{y_1,...y_L\}$, $y_i \in \{1,-1\}$. We consider that the vectors are optimally separated by the hyperplane H$_0$ if they are classified without error and the margin is maximal. In order to be correctly classified, the vectors must verify

$$\begin{aligned} &f(x_i) \geq +1 \;\; for \;\; y_i = +1 \\ &f(x_i) \leq -1 \;\; for \;\; y_i = -1 \end{aligned} \tag{3}$$

Or, more concisely,

$$y_i f(x_i) \geq 1, \;\; \forall i. \tag{4}$$

Thus the problem of finding the SVM classifying function H$_0$ can be stated as follows:

$$\begin{aligned} &\text{minimize } \frac{1}{2}\|w\|^2 \\ &\text{subject to } y_i f(x_i) \geq 1, \;\; \forall i. \end{aligned} \tag{5}$$

This is called the *primal* optimization problem **[17][18][19]**. In order to solve it, we form the following Lagrange function

$$L(w,b) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{L} \alpha_i [y_i f(x_i) - 1] \tag{6}$$

where the Lagrange multipliers α_i verify

$$\alpha_i \geq 0, \quad \forall i. \tag{7}$$

The Lagrangian *L(w,b)* must be minimized with respect to *w* and *b*, so its gradient must vanish, i.e.

$$\frac{\partial}{\partial b} L(w,b) = 0, \; \frac{\partial}{\partial w} L(w,b) = 0 \tag{8}$$

From the two above equations, it follows, respectively, that

$$\sum_{i=1}^{L} \alpha_i y_i = 0, \tag{9}$$

and $$w = \sum_{i=1}^{L} \alpha_i y_i x_i \tag{10}$$

Substituting the conditions (9) and (10) into the Lagrangian (6), we arrive at the so-called *dual* optimization problem:

$$\begin{aligned} &\text{maximize } \sum_{i=1}^{L} \alpha_i - \frac{1}{2} \sum_{i=1}^{L} \sum_{j=1}^{L} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \\ &\text{subject to } \sum_{i=1}^{L} \alpha_i y_i = 0 \text{ and } \alpha_i \geq 0, \; \forall i \end{aligned} \tag{11}$$

The dual optimization problem is a (convex) quadratic programming problem that can be efficiently solved with a number of mathematical algorithms **[20].** In our work we use the decomposition method with conventional modifications **[19]**.

Data observed in real conditions are frequently affected by outliers. Sometimes they are caused by noisy measurements. If the outliers are taken into account, the margin of separation decreases so the solution does not generalize so well, and the data patterns may no longer be linearly separable. To account for the presence of outliers, we can *soften* the decision boundaries by introducing a *slack* positive variable ξ_i for each training vector **[18]**. Thus, we can modify the equations (3) in the following way:

$$\begin{aligned} &\underline{w}'\underline{x}_i + b \geq +1 - \xi_i \quad \textit{for } \; y_i = +1 \\ &\underline{w}'\underline{x}_i + b \leq -1 + \xi_i \quad \textit{for } \; y_i = -1 \end{aligned} \tag{12}$$

Obviously, if we take ξ_i large enough, the constraints (12) will be met for all *i*. To avoid the trivial solution of large ξ_i, we introduce a penalization cost in the objective function in (5), and thus the primal optimization formulation becomes:

$$\begin{aligned}&\text{mimimize } \left(\frac{1}{2}\|\underline{w}\|^2 + C\sum_{i=1}^{L}\xi_i\right)\\ &\text{subject to } y_i(\underline{w}'\underline{x}_i + b) \geq 1 - \xi_i, \quad \forall i,\end{aligned} \tag{13}$$

where C is a positive regularization constant which controls the degree of penalization of the slack variables $\xi_{i,}$ so that, when C increases, fewer training errors are permitted, though the generalization capacity may degrade. The resulting classifier is usually called *soft margin classifier.* If $C = \infty$, no value for ξ_i except 0 is allowed; it is the so-called *hard margin* SVM case.

The formulation (13) leads to the same dual problem as in (11) but changing the positivity constraints on α_i by the constraints $0 \leq \alpha_i \leq C$. Thus, it can be shown that the optimal solution has to fulfill the following conditions (known as Karush-Kuhn-Tucker optimality conditions) **[19]**:

$$\alpha_i = 0 \quad \Rightarrow \quad y_i f(x_i) \geq 1 \quad \textit{and} \quad \xi_i = 0 \tag{14}$$

$$0 < \alpha_i < C \quad \Rightarrow \quad y_i f(x_i) = 1 \quad \textit{and} \quad \xi_i = 0 \tag{15}$$

$$\alpha_i = C \quad \Rightarrow \quad y_i f(x_i) \leq 1 \quad \textit{and} \quad \xi_i > 0 \tag{16}$$

The above equations reveal one of the most important features of SVM: since most patterns lie outside the margin area, their optimal α_i's are zero (equation (14)). Only those training patterns x_i which lie on the margin surface (equation (15)) or inside the margin area (equation (16)) have non-zero α_i, and they are named support vectors. Consequently, the classification problem consists of assigning to any input vector x one of the two classes according to the sign of

$$f(x) = \sum_{j=1}^{M}\alpha_j y_j x_j \cdot x + b, \tag{17}$$

being M the number of support vectors. The fact that the support vectors are a small part of the training data set makes the SVM implementation practical for large data sets **[19]**.

In real situations, the distribution of the data among the classes is often not uniform, so some classes are statistically under-represented with respect to other classes. To cope with this problem in the two-class SVM formulation, we can introduce different cost functions for positively- and negatively-labeled points in order to have asymmetric soft margins, so that the class with smaller data size obtains a larger margin **[21]**. Consequently, the conventional soft margin approach can be generalized as

$$\text{minimize } \left(\frac{1}{2}\|\underline{w}\|^2 + C_- \sum_{i:y_i=-1} \xi_i + C_+ \sum_{i:y_i=1} \xi_i\right)$$

$$\text{subject to } y_i(\underline{w}'\underline{x}_i + b) \geq 1 - \xi_i, \quad \forall i. \tag{18}$$

As the formulation (18) suggests, when C_+ increases, the number of allowed training errors from positively-labeled data decreases, but at the expenses of increasing the allowed number of training errors from the negatively-labeled data. And the opposite occurs when C_- increases.

The resulting dual problem has the same Lagrangian as in (11), but the positivity constraints on α_i now become:

$$0 \leq \alpha_i \leq C_+ \text{ for } y_i = +1$$

$$0 \leq \alpha_i \leq C_- \text{ for } y_i = -1 \tag{19}$$

For a non-linearly separable classification problem we have first to map the data onto a higher dimensional (possibly infinite) feature space where the data are linearly separable. Accordingly, the Lagrangian of the dual optimization problem (11) must be changed to

$$\sum_{i=1}^{L} \alpha_i - \frac{1}{2}\sum_{i=1}^{L}\sum_{j=1}^{L} \alpha_i \alpha_j y_i y_j \phi(x_i) \cdot \phi(x_j) \tag{20}$$

Notice the input vectors are involved in the expression through a kernel function

$$K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j), \tag{21}$$

which can be thought as a non-linear similarity measure between two data points. According to the Mercer's theorem **[22]**, any (semi) positive definite symmetric function can be regarded as a kernel function, that is, as a dot product in some space, so we will look for (semi) positive definite symmetric functions that imply a data transformation to a new space where the classes can be linearly separated. Note that there is not need to know the mapping function ϕ explicitly, but only the kernel $K(x_i, x_j)$.

The most often used kernel functions in SVM applications are the following two:

$$\text{Radial Basis Function (RBF): } K(x_i, x_j) = e^{-|x_i - x_j|^2 / 2\sigma} \tag{22}$$

$$\text{Polynomial: } K(x_i, x_j) = (x_i \cdot x_j)^d \tag{23}$$

Thus, from equation (17) and the kernel concept, it follows that the two-class classification process with a SVM consists of assigning a positive/negative label to each input vector *x* through the following equation:

$$y(x) = \operatorname{sgn}(\sum_{j=1}^{M} \alpha_j y_j K(x, x_j) + b) \tag{24}$$

being *M* the number of support vectors.

As SVM is a binary classifier, we cannot employ it directly in our acoustic event classification problem, since we have a set of 16 classes. In the literature, several methods of extending from binary classifiers to multi-class classifiers can be found: *one against all*, *one against one*, DAGSVM, ECOC,… (see **[13][23]** for a comparison). In our experiments, we first use the scheme proposed in **[11]**, namely a binary tree with a SVM at each node. A disadvantage of the binary tree approach is that the number of classes has to be a power of two, otherwise the tree is unbalanced and some classes are more likely to be chosen than others. The alternative we propose in Section 5 is based on a decision tree that uses a specific feature set at each node, and it is trained with a clustering technique from a given set of confusion matrices. In this way, it uses the most discriminative feature set at each step of classification and works for any number of classes. The effect of a confusion matrix based modification of the generalization parameters C_+ and C_- of the SVM classifier is also presented in Section 5.

4.2. Gaussian Mixture Models

Gaussian mixture models are quite popular in speech and speaker recognition. In the design step, we have to find the probability density functions that most likely have generated the training patterns of each of the classes, assuming that they can be modeled by mixtures of Gaussians.

In the GMM, the likelihood function is defined as

$$p(x) = \sum_{i=1}^{P} w_i N(x; \mu_i, \Sigma_i) \tag{24}$$

where *P* is the number of Gaussians, the weights w_i verify

$$\sum_{i=1}^{P} w_i = 1 \text{ and } w_i \geq 0, \forall i \tag{25}$$

and $N(x; \mu, \Sigma)$ denotes the multivariate Gaussian distribution

$$N(x;\mu,\Sigma)=\frac{1}{(2\pi)^{\frac{|x|}{2}}\sqrt{|\Sigma|}}\exp\left(-\frac{1}{2}(x-\mu)^{\mathrm{T}}\Sigma^{-1}(x-\mu)\right) \tag{26}$$

being μ the mean vector and Σ the covariance matrix (often considered diagonal). As the goal is to maximize the likelihood (ML), the parameters of the GMM (w_i, μ_i, and Σ_i) are obtained via the Expectation-Maximization algorithm **[15]**. Unlike SVM, which is a two-class classifier, GMM-based classifiers can handle an arbitrary number of classes. The GMM-ML classifier belongs to the group of generative classifiers, unlike SVM, which is a discriminative classifier. Due to this different approach, GMM generally needs a larger training set than SVM and so it is usually considered more complex **[24]**.

In the next section, comparative tests are reported by using the two basic classifiers (GMM and SVM) and several classification schemes.

5. EXPERIMENTS

Several experiments were carried out to assess the classification performance of the selected feature sets and the classification systems, either based on SVM or GMM. To perform the evaluation, the acoustic event samples were randomly permuted within each class and indexed, so odd index numbers were assigned to training and even index numbers to testing. Also, 20 permutations were used in each experiment. Because of unevenness in the number of representatives of the various classes, the overall performance is computed as an average of the individual class performances.

As preliminary tests with the SVM classifier showed a superiority of the RBF kernel over the polynomial one, only the former was used in the evaluation. There are two main parameters (hyperparameters) that are to be specified using SVMs: σ from the RBF kernel and the regularization parameter *C* presented in Section 4.1.1. Regarding the setting of σ, 5-fold cross-validation **[17]** was applied. After that kernel parameter is found, the whole training set is used again to generate the final classifier.

5.1. Binary tree scheme

First of all, a binary tree with a SVM at each node was applied to our acoustic event classification problem. Figure 2 illustrates how the classifier works. In our implementation, the classes in the bottom level are ordered randomly. In **[11],** each SVM was trained using *C*=200; in our work, we chose *C*=1, since this value yielded better results in the experiments, a fact that may indicate that our data are more noisy (contains more outliers) than data used in **[11]**.

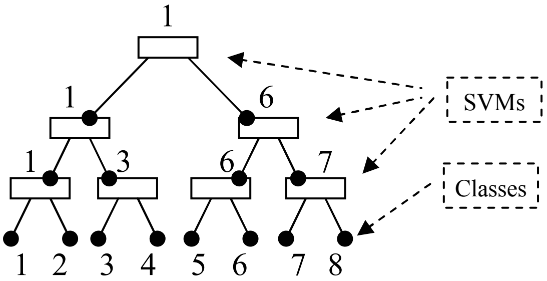


Figure 2. Binary tree structure for eight classes. Every test pattern enters each binary classifier, and the chosen class is tested in an upper level until the top of the tree is reached. The numbers 1–8 encode the classes. The figure shows a particular example, where class 1 is the class chosen by the classification scheme.

This SVM-based classification system was compared with a GMM classifier. The latter has one model per class and, for every test pattern, the model with maximal likelihood is chosen. Both a fixed and a variable number of Gaussians per class were tried; the best accuracy was achieved by using a variable number that depends on the amount of data per class.

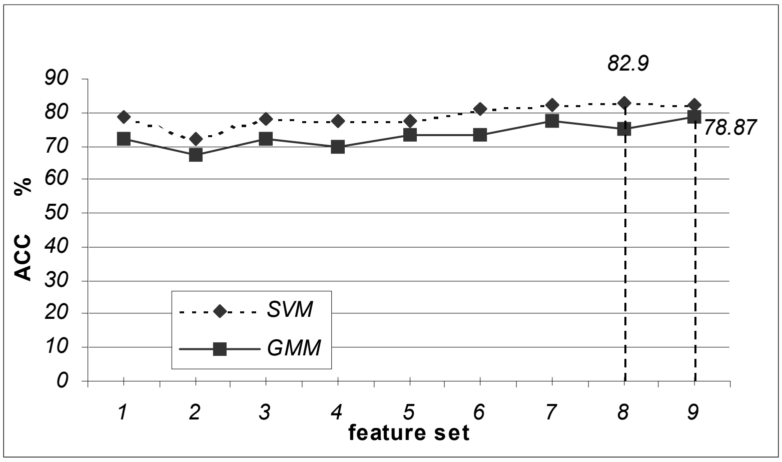


Figure 3. Percentage of classification rate for the SVM-based binary tree classifier and the GMM classifier on the defined feature sets

Figure 3 shows results for both classifiers. The best feature set in combination with the GMM classifier was the set number 9 (Perc + FF), with recognition rate 78,9%, whereas for the SVM classifier was the set number 8 (Perc + FF + der), with 82,9% recognition rate. Note that, in our experiments, the SVM approach shows a higher performance than the GMM one across all types of feature sets.

5.2. Confusion matrix based clustering scheme

We have developed a tree clustering algorithm which makes use of confusion matrices, one for each feature set. They are obtained from the experiments reported in the last section, by averaging over the 20 permutations, and normalizing their elements so that each row adds up 1. Those confusion matrices are used to find the best way of splitting the classes at a given node into two clusters with the least mutual confusion. As we have a relatively small number of classes, we can perform exhaustive search and get the global minimum. For the sake of homogeneity, we use confusion matrices obtained by SVM classifiers for SVM clustering, and GMM matrices for GMM clustering.

As our database contains a large variety of sounds, the feature set that gets the largest classification rate for a given class is not necessarily the best one for a different class. This fact is illustrated in **Figure 4**, where the three considered classes (liquid pouring, sneezing and sniffing) show their performance peaks at different feature sets and none of the sets is the 8th, the one that yields the best overall performance. Therefore, it is reasonable to assume that the performance can improve by using a specific feature set to discriminate within each pair of classes or groups of classes.

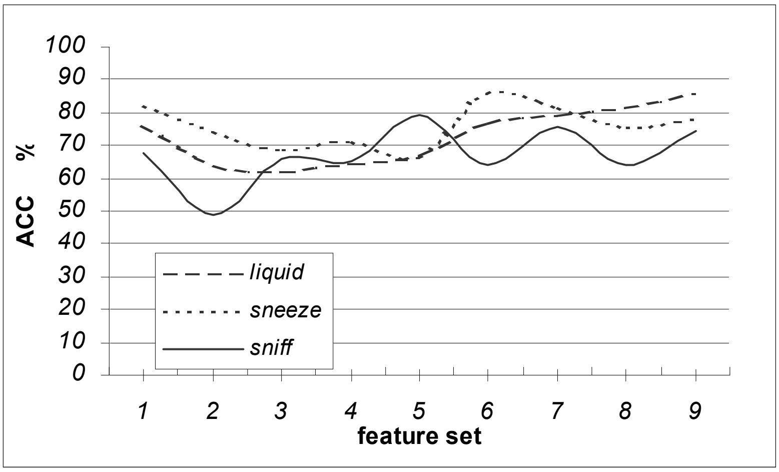


Figure 4. Dependence of performance of classifying "liquid_pouring", "sneeze" and "sniff" upon the feature sets using SVMs.

The clustering algorithm that selects a specific feature set for each tree node will be presented in the next section. The simpler case that uses the same feature set at every node is also considered in the experiments. We refer to them, respectively, as *variable-feature-set* and *fixed-feature-set clustering schemes*. In the following, we will present the former clustering algorithm since the latter is a particular case of it.

5.2.1. The variable-feature-set clustering algorithm

The algorithm for clustering with a variable-feature-set approach is formally described in **Figure 5**. At the first step, all possible combinations of grouping 16 classes into two clusters (i.e. grouping 6 and 10, 8 and 8, etc) are searched over the available 9 confusions matrices that correspond to the 9 considered feature sets. For example, for the SVM clustering, we found that the 16 classes were best separated choosing the clusters C_1={9} and C_2={1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 15, 16}, and the 6th feature set. That process is carried out until we have single event clusters. Note in the expression of $S_k^{n,m}$ from **Figure 5** that the confusion measures e_{ij}^k are normalized by the corresponding accuracies e_{ii}^k to cope with the dispersion of performance rates among the classes. Regarding the GMM classifier, the algorithm also groups the classes into two clusters, but in this case two models are generated at each step, one for each cluster.

The above clustering technique is intended for a relatively small number of classes, as in our acoustic event classification task. When the number of classes is large either agglomerative hierarchical clustering or divisive hierarchical clustering **[25]** can be used if they are modified to handle several feature sets while searching; however, they do not guarantee to reach the global minimum.

```
1. Initialize $N$=16.
2. For $n$=1…$N$/2
    a. Determine $M$ combinations of grouping $N$ classes into two clusters $C_1$ and $C_2$
       containing $n$ and $N$-$n$ classes, respectively.
    b. For $m=1...M$
        • Having the m-th grouping combination, look up at each confusion matrix and
          measure how much are $C_1$ and $C_2$ confused for each feature set $k$, by computing
              $S_k^{n,m} = \frac{1}{|C_1||C_2|} \sum_{i=1}^{|C_1|} \sum_{j=1}^{|C_2|} \left( \frac{e_{ij}^k}{e_{ii}^k} + \frac{e_{ji}^k}{e_{jj}^k} \right)$
          where $e_{ij}^k$ denotes the i,j-th element of the k-th confusion matrix, and $|C_1|$ and
          $|C_2|$ are the number of classes (cardinalities) of the two clusters.
        • Find the minimum confusion measure over all feature sets
              $B_{n,m} = \min_k (S_{\mathrm{k}}^{\mathrm{n,m}})$
    c. Find the minimum confusion measure over all grouping combinations for the current
       number of classes at each cluster
              $T_n = \min_m (\mathrm{B}_{\mathrm{n,m}})$
3. Find the minimum confusion measure over all possible numbers of classes at each cluster
              $R = \min_n (T_n)$
4. Repeat steps 2-3 for each node of growing tree, initializing $N$ with $N$-$n$ for the right branch
   and $N$ with $n$ for the left one, until $N$=1 is reached.
```

Figure 5. Clustering algorithm based on an exhaustive search and using a set of estimated confusion matrices.

5.2.2. Dealing with the data unbalance problem

In our experiments, we have tried several ways of alleviating the problem of having a too much different amount of training data between the two clusters at a given tree node. A straightforward way of tackling that problem which has been considered in the experiments consists of restricting the exhaustive search in **Figure 5** to look for an equal number of classes at each cluster, i.e. having only the index value n=N/2 at step 2 of the algorithm. That solution is no longer optimal in terms of the tree structure, but the involved SVMs will work with more balanced data. Hereafter, we will refer to it as *restricted clustering*. **Figure 6** shows the trees obtained by the normal (unrestricted) and restricted clustering algorithms in the SVM case. Note that the two trees show a very different structure, but they have the same number of nodes (N-1), that is the same number of trained SVM classifiers. Indeed, the restricted tree shows a balanced structure, whereas, as it can be observed in **Figure 6**, in the normal clustering case we mostly have only one class separated on each clustering step. Actually, there is only one case where there are two classes grouped in the smaller cluster, which corresponds to classes 11 and 12. We have observed that the amount of confusions between both classes is a large portion of the total error for class 11. Regarding the GMM-based techniques, since each class model is trained without using information about the other classes it is not so much influenced by the problem of data unbalance. However, we will also consider both clustering schemes for the GMM case. The resulting schemes are similar to those in **Figure 6**.

The alternative way of coping with data unbalance used in our experiments (already mentioned in Section 4.1.1) is to introduce different regularization parameters for positively- and negatively-labeled training samples. Additionally, since a measure of confusions at each tree node can be obtained as a byproduct of the clustering algorithm, we have used these estimated measures to adapt the regularization parameters. The greater the confusion is, the larger the error should be allowed during training, and so the smaller the regularization parameters should be. Consequently, we force those parameters to be inversely proportional to the confusion measures. Indeed, we have a ∞ value at the beginning for normal clustering since the confusion at this step is 0. Note from **Figure 5** that if the performance of a class for a given feature set were 0 ($e_{ii}^{k} = 0$), the value of $S_k^{n,m}$ would be ∞. In order to decrease the contribution of that possible zeroth performance of a class to the computation of the confusion measures of the whole cluster, we substitute zero by a small value. In our algorithm, we use 0.001.

Three different methods of using and computing the regularization parameters in the SVM-based classifiers are considered in this work, along with the baseline method that uses only a constant parameter *C*=*K*. They are defined in the following, denoting by S_n the confusion measure at the *n-th* classification step:

1) Only one regularization parameter *C* computed as

$$C = K\frac{1}{S_n}. \tag{27}$$

2) Two different parameters C_+ and C_-, defined such that

$$C_+ = K\frac{A_-}{A_+},\ C_- = K\frac{A_+}{A_-} \tag{28}$$

where A_+ and A_- are the number of positive and negative training samples, respectively. In this way, the training errors of the two classes contribute equally to the cost of misclassification.

3) The effect of doing both adaptations simultaneously, namely,

$$C_+ = K\frac{A_-}{A_+}\frac{1}{S_n}, \quad C_- = K\frac{A_+}{A_-}\frac{1}{S_n} \tag{29}$$

In our tests, K was set to value 10 since it gave the best performance for the baseline method with constant C.

5.3. Results and discussion

Table 3 shows classification performance for GMM and SVM classifiers using either a variable- or a fixed-feature-set approach, and either normal (N) or restricted (R) clustering. The table also shows the standard deviation for each experiment, estimated over the 20 repetitions. The first column of results corresponds to *C*=*K*=10, and the other 3 columns correspond, respectively, to the three above-mentioned methods of computing the regularization parameters in the SVM cases. Note that SVM performs consistently better than GMM, and with SVM the highest accuracies are obtained using the third method.

The column *C*=*K* in **Table 3** shows that, without any adaptation, SVM-based restricted clustering performs equally well as normal clustering (and better than the binary tree scheme). In that table, we can notice that SVM-N takes advantage of using different *C* values for each class according to the simple equation of proportionality (27), since the training set sizes are largely spread across classes in our database. And SVM-R does not take any advantage due presumably to the balancing average implied by the half-to-half constraint. Additionally, as we can see from **Table 3**, introducing prior knowledge (about confusions) with the generalization parameter *C* (method 1) does not have a positive influence on the classification performance, while introducing it along with different *C* values for positive and negative classes (method 3) leads to an improvement for both types of clustering trees. The gain in performance, however, is not much significant, so there is a need to have a more sophisticated algorithm of introducing prior knowledge about confusions in the regularization parameters. In restricted clustering we can obtain only the global minimum of error within the constraint that is why the final performance of the SVM-R technique is worse than that of the normal one (**Table 3**, method 3). We can also observe that normal clustering seems to perform slightly better than restricted clustering for GMM.

Notice in **Table 3** how the results for SVM fixed-feature-set clustering show just a slightly worse performance with respect to the variable-feature-set ones. This can be explained in the following way. On the one hand, for fixed-feature-set clustering, the chosen feature set is the one which yielded the best results in the previous experiments with binary tree, i.e. the 8th, which includes all kind of features: perceptual, envelope representation and time derivatives. On the other hand, the SVM classifier has somehow a built-in feature selection process. In fact, as it implicitly works with features in a transformed domain, if the kernel and the hyperparameters are appropriately chosen (so that good results are obtained), its transformation may imply emphasizing those features that are crucial for a good classification. That is why for the SVM classifier no feature selection technique leads to a huge classification improvement **[26]**. Moreover, using real-world data, it was shown in **[26]** that the best feature set was the one that included all types of features. Additional evidence from our experiments is given by the fact that the difference in performance between fixed- and variable-feature-set is more noticeable for the GMM classifiers than for the SVM ones. Nevertheless, in spite of that implicit feature selection process in SVM classifiers, and the fact that a fixed-feature-set scheme requires less computation, the variable-feature-set scheme may still be advantageous for the SVM case. In fact, apart from offering some information about the acoustical properties of the chosen classes, the variable-feature-set scheme obviously shows a smaller restriction bias than that of the fixed-feature-set clustering, thus resulting in a smaller inductive bias and a presumable higher overall accuracy **[27]**.

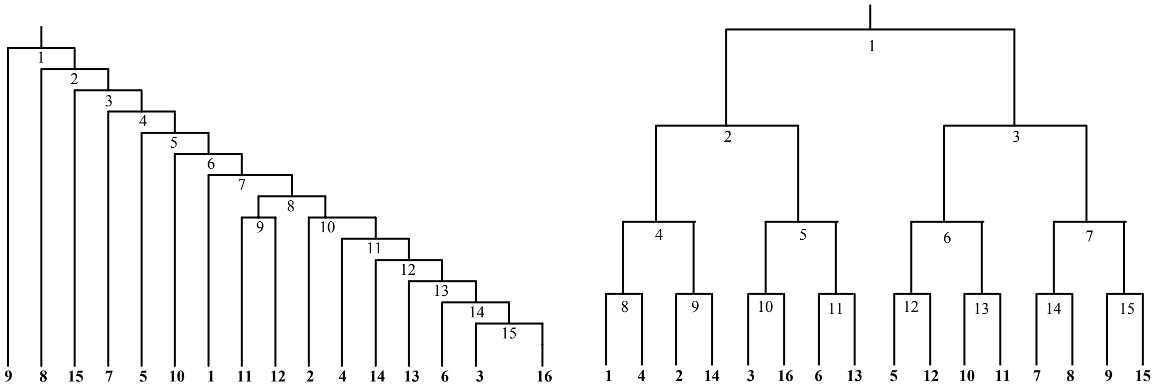


Figure 6. Normal and restricted clustering schemes for SVM classifiers

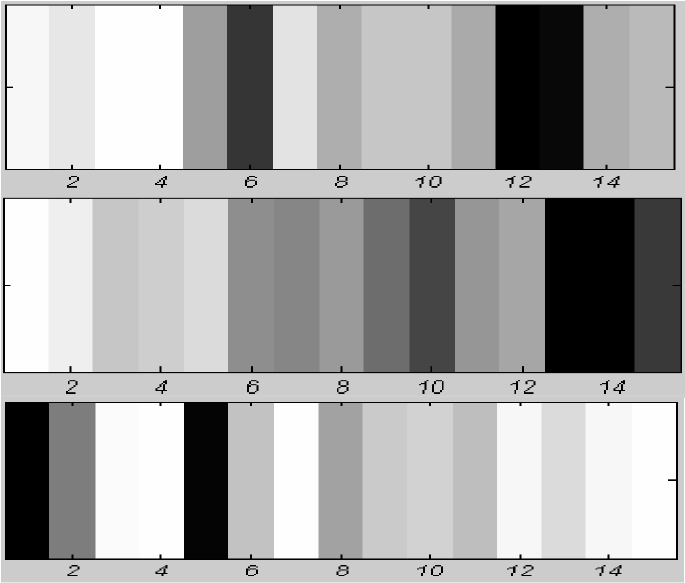
	C=K	**Method 1**	**Method 2**	**Method 3**
SVM-N variable	84.67 ± 2.5	84.05 ± 1.7	86.71 ± 1.4	88.29 ± 2.1
SVM-R variable	84.72 ± 2.6	84.88 ± 2.7	84.95 ± 2.2	87.20 ± 1.5
GMM-N variable		83.6 ± 2.2		
GMM-R variable		82.15 ± 2.3		
SVM-N fixed	84.6 ± 1.9	84.4 ± 1.6	86.6 ± 3.0	87.10 ± 1.8
SVM-R fixed	84.6 ± 2.7	83.8 ± 1.2	84.4 ± 2.3	87.06 ± 1.8
GMM-N fixed		81.2 ± 2.3		
GMM-R fixed		80.7 ± 2.4		

Table 3. Performances of variable-feature-set and fixed-feature-set classifiers using different adaptations of the regularization parameters for the SVM classifiers. -N and -R, denote normal and restricted clustering scheme, respectively. Standard deviations estimated over 20 repetitions are denoted with ± σ.

		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
	Confusion	*0*	*0.01*	*0.03*	*0.07*	*0.78*	*0.82*	*0.83*	*0.98*	3.90	1.27	1.57	2.57	8.44	**15.00**	**46.88**
SVM-N	***Features***	*6*	*3*	*6*	*7*	*7*	*8*	*3*	*7*	5	9	5	3	6	**4**	**9**
	Error	0.78	1.97	0	0	7.65	15.42	2.19	6.30	4.47	4.38	6.64	19.63	18.93	6.23	5.39
	Confusion	*0.41*	2.15	*0.04*	*0.15*	**15.74**	1.74	*0*	*0*	4.41	**46.88**	2.23	3.31	*0*	*0*	*0*
SVM-R	***Features***	*7*	9	*7*	*8*	**6**	9	*6*	*5*	5	**4**	8	8	*3*	*1*	*1*
	Error	23.59	12.35	1.14	0.69	23.1	6.01	0.46	9.12	5.32	4.76	6.51	1.29	3.95	1.20	0.53
	Confusion	*0.01*	*0.1*	*0.18*	*0.22*	*0.28*	*0.74*	1.10	1.14	2.41	3.77	3.00	7.55	**13.76**	**29.86**	**55.07**
GMM-N	***Features***	*6*	*9*	*9*	*3*	*7*	*7*	9	5	9	4	5	7	**1**	**6**	**1**
	Error	0.07	1.00	3.46	2.91	2.19	6.55	6.89	5.94	8.33	10.63	6.11	5.28	14.61	14.60	11.42
	Confusion	*0.53*	5.59	*0.10*	**15.60**	1.92	*0.58*	*0*	**12.03**	**55.07**	*0.81*	6.84	*0.77*	*0.92*	*0*	*0.1*
GMM-R	***Features***	*9*	6	*7*	**6**	8	*3*	*1*	**1**	**1**	*5*	5	*7*	*7*	*1*	*6*
	Error	25.35	24.27	3.27	15.43	3.23	3.18	0	3.18	9.50	2.63	7.18	0.51	1.83	0.38	0.07

Table 4. Confusion measure S_n (multiplied by 100), best separating feature set, and percentage distribution of the classification error (for the best results in **Table 3**) along the 15 nodes (depicted in **Figure 6** for SVM) for both normal and restricted clustering, and for the variable-features-set SVM classifier and the GMM classifier.

The proposed clustering schemes (both normal and restricted) show two computational advantages in front of the binary tree classifier. First, the required number of trained SVM is N-1, where N is the number of classes, while for the binary tree (N-1)N/2 trained SVM are needed. Second, the proposed schemes involve a smaller number of classification steps, 4 for restricted clustering, and between 1 and 14, depending on the input pattern, for normal clustering in our case (see **Figure 6**), whereas the binary tree requires 15. However, the proposed variable-feature-set scheme has an obvious disadvantage: with our choice of feature sets (see Table 2) up to 9 feature sets can be involved in testing, 7 in our case (numbers 3 4 5 6 7 8 9).



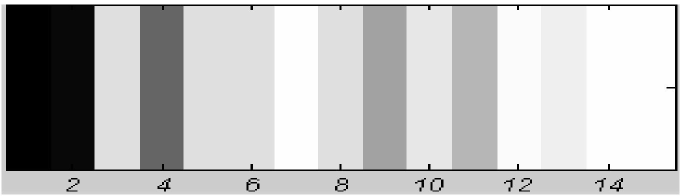


Figure 7. Distribution of the errors along the tree path for SVM-N, GMM-N, SVM-R and GMM-R. A darker cell means a larger error.

From **Table 4** we can extract some observations concerning the feature sets. Looking at bold numbers in the SVM case of **Table 4**, which correspond to a confusion measure larger than 10, it seems that the best separating feature sets for the most confused classes mostly are FFBE-based features (sets 4,5,8,9), while observing the italic numbers, which correspond to a confusion measure smaller than 1, it appears that the for the least confused classes the best separating feature sets are MFCC-based (sets 2,3,6,7). This fact may indicate that the FFBE-based features are more discriminative than the MFCC features for highly overlapped data distributions, while MFCC features appear to show the best performance when there is a clearer separation between classes. However, for the most confused classes in the GMM case (see bold numbers in the GMM part of **Table 4**) the average best feature set is the one we have called perceptual set. This may be due to the relatively low size of that feature set, which facilitates the estimation problem.

Note in **Table 4** that for normal clustering the largest errors are more located towards the end of the tree path while for restricted clustering they are towards the beginning. This effect, that is also illustrated in **Figure** 7, can be expected for the normal clustering technique, due to the way the clustering algorithm in **Figure 5** works. Apparently, the restrictions applied by restricted clustering make the largest errors are placed at the beginning. That information can be useful to improve classification by boosting, since the most erroneous

	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*
1	**96.67**	0	0	0	0	1.67	0	0	0	0	0	0	0	0	0	1.67
2	0	**96.79**	0	0.19	0.57	0.19	0	0	0.57	0	0	0.38	1.13	0.19	0	0
3	0	0.43	**88.70**	2.61	0	5.22	0	0	0	0	0	0.43	2.61	0	0	0
4	0	0.75	0.50	**96.50**	0	0.75	0	0	0.50	0	0.50	0	0.50	0	0	0
5	0	0	0	2.27	**87.73**	3.64	0	0	0	0	2.27	3.18	0.91	0	0	0
6	0.77	0	26.92	3.85	0	**48.46**	0	0	0	9.23	0	0	10.00	0.77	0	0
7	0	0	0	0	0	0	**100**	0	0	0	0	0	0	0	0	0
8	0	0.20	0	0.40	0	0	0	**98.8**	0.20	0	0	0.2	0.20	0	0	0
9	0	0	0	0	0	0	0	0.2	**99.80**	0	0	0	0	0	0	0
10	0	0	1.33	1.33	0	3.33	0	0	0	**92.67**	0	0	1.33	0	0	0
11	0	0	0	2.00	1.50	1.00	0	2.5	0	3.50	**77.00**	10.0	2.50	0	0	0
12	0	1.30	0	0	0	0.60	0	0.2	0	0	0.10	**97.2**	0.60	0	0	0
13	0	0.50	14.50	0	0	8.00	0	0	0	0.50	2.50	0	**74**	0	0	0
14	0	5.00	5.00	0	0	0	0	0	0	6.67	0	0	6.67	**76.67**	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**100**	0
16	0	0	11.67	0	0	3.33	0	0	0	1.67	0	0	1.67	0	0	**81.67**

Table 5. Confusion matrix corresponding to the best results (88.29 %)

steps generally contain rare class data and boosting the SVM that deal with rare categories has been shown to improve general performance in **[27]**.

Table 5 shows the confusion matrix corresponding to the best results. The resulting classification rates for the various types of sounds are diverse due to both the acoustic nature of sounds and the unevenness of the number of samples in the database. Notice that the sounds we could name *human vocal-tract non-speech* (HVTNS) sounds (numbers 3, 6, 13, 14, and 16) account for a large relative amount of confusions, since they only are 5/16 of the total number of classes and contribute with 69.7% of the total error. The only other sound with more than 10% error is number 11. In average, the HVTNS classes have a small number of samples in the database, but there are other sounds with similar number of samples (like *chair moving*), which do not show such a high error. Furthermore, the HVTNS sounds are mainly confused among themselves (the average for the 5 classes is 73.96%). Actually, although the proposed clustering schemes are based on acoustic features, some clusters can be interpreted from a semantic point of view, that is according to their source identity; e.g. the shaded cluster in **Figure 8** contains “cough”, “laughter”, “sneeze”, and “yawn”, sounds which belong to that HVTNS set.

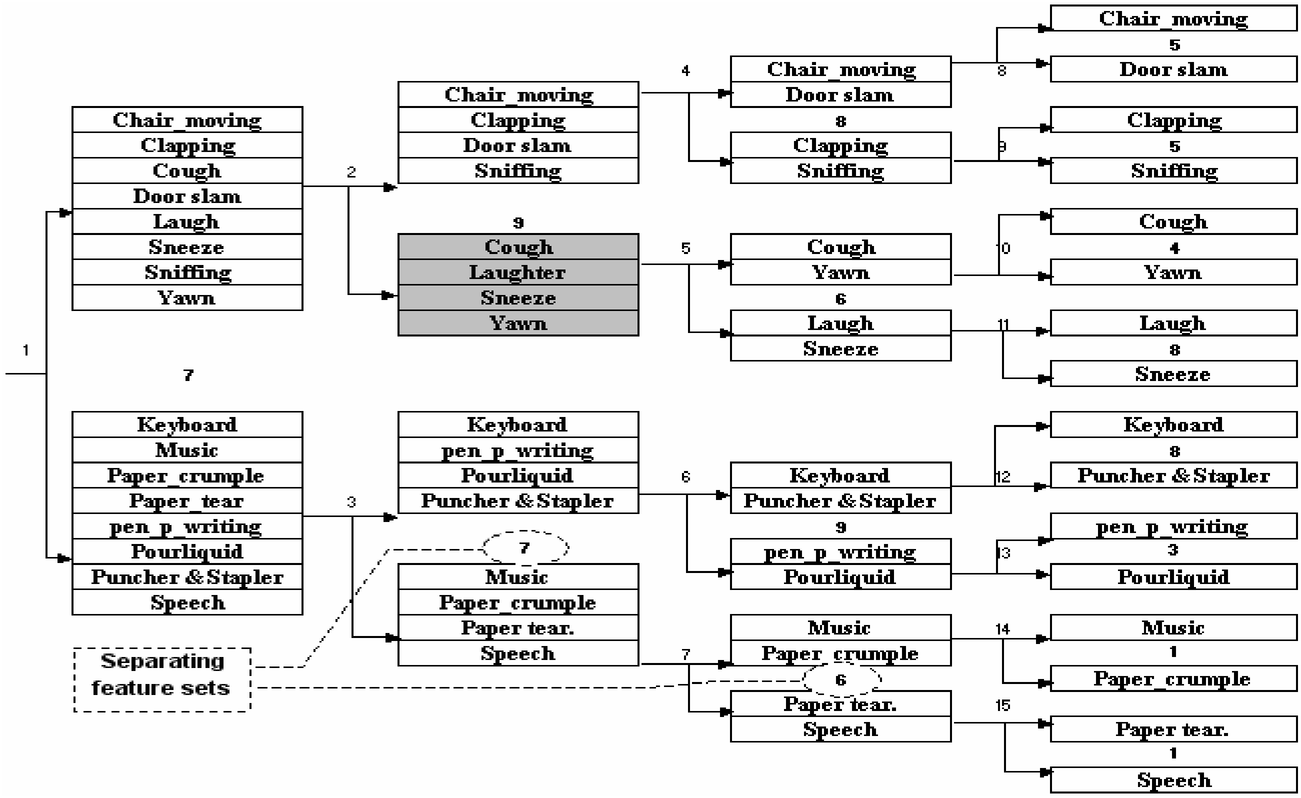


Figure 8. Restricted clustering tree based on SVM. The numbers in the nodes are the ordinal numbers of the 15 SVM classifiers, and the bold numbers between each pair of clusters denote the best separating feature sets.

6. CONCLUSION

This paper is a preliminary attempt to deal with the problem of classifying acoustic events that occur in a meeting-room environment. A database has been defined, and several feature sets and classification techniques have been tested with it. In our tests, the SVM-based techniques show a higher classification capability than the GMM-based techniques, and the best results were consistently obtained with a confusion matrix based variable-feature-set clustering scheme, arriving with SVM to a 88,29 % classification rate, which implies a 31.5% relative average error reduction with respect to the best result from the conventional binary tree scheme. That good performance is mostly attributable to the presented clustering technique, and to the fact that SVM provides the user with the ability to introduce knowledge about data unbalance and class confusions.

7. ACKNOWLEDGEMENTS

The authors wish to thank Enric Monte for his valuable insights into pattern recognition and machine learning, and also Jaume Padrell for his valuable suggestions during the work. This work has been partially sponsored by the EU-funded project CHIL, IP506909, and the Spanish Government-funded project ALIADO.

8. REFERENCES

[1]. A. Bregman, *Auditory Scene Analysis*. MIT Press, Cambridge,1990.
[2]. CHIL - Computers In the Human Interaction Loop, http://chil.server.de/
[3]. L. Lu, H-J. Zhang, H. Jiang, "Content Analysis for Audio Classification and Segmentation"*, IEEE Transactions on Speech and Audio Processing*, V.10, N. 7, pp. 504-516, October. 2002.
[4]. D. Hoiem, Y. Ke, and R. Sukthankar, "SOLAR: Sound Object Localization and Retrieval in Complex Audio Environments", *International Conference on Acoustics, Speech, and Signal Processing, Philadelphia, March* 2005.
[5]. M. Slaney, "Mixtures of Probability Experts for Audio Retrieval and Indexing", *IEEE International Conference on Multimedia and Expo*, Lausanne, August 2002.
[6]. L. Kennedy, D. Ellis, "Laughter Detection in Meetings", NIST Meeting Recognition Workshop, International Conference on Acoustics, Speech, and Signal Processing, Montreal, May 2004.
[7]. J. Pinquier, J. Arias, and R. André-Obrecht, *"*Audio Classification by Search of Primary Components", *International Workshop on Image, Video and Audio Retrieval and Mining*, Sherbrooke, October, 2004.
[8]. T.Nishiura, S. Nakamura, K. Miki, K. Shikano,"Environmental Sound Source Identification Based on Hidden Markov Model for Robust Speech Recognition", *Eurospeech 2003,* Geneva, pp.2157-2160, September, 2003.
[9]. S.Nakamura, K.Hiyane, F.Asano, T.Nishiura, and T.Yamada, "Acoustical Sound Database in Real Environments for Sound Scene Understanding and Hands-Free Speech Recognition", *2nd International Conference on Language Resources & Evaluation*, Athens, 2000.
[10]. D. Gerhard, "Audio Signal Classification: History and Current Techniques", Technical Report TR-CS 2003-07 November, 2003.
[11]. G. Guo, Z. Li, "Content-based Audio Classification and Retrieval using Support Vector Machines", *IEEE Transactions on Neural Networks*, vol. 14, pp 209-215, January, 2003.
[12]. L.Lu, S. Z. Li, and H. Zhang, "Content-based Audio Classification and Segmentation by Using Support Vector Machines", *ACM Multimedia Systems Journal*, 8 (6), pp. 482-492, March 2003.
[13]. C.W.Hsu, C.J. Lin, "A Comparison of Methods for Multi-class Support Vector Machines", *IEEE Transactions on Neural Networks*, Vol. 13, pp. 415-425, 2002.
[14]. ShATR Multiple Simultaneous Speaker Corpus, http://www.dcs.shef.ac.uk/research/groups/spandh/projects/shatrweb/index.html
[15]. L. Rabiner, B.H. Juang, *Fundamentals of Speech Recognition*, Prentice Hall, 1993.
[16]. C. Nadeu, J. Hernando, and M. Gorricho "On the decorrelation of filter-bank energies in speech recognition", *European Speech Processing Conference (Eurospeech'95)*, Madrid, pp. 1381-1384, September, 1995.
[17]. C. Burges, "A tutorial on support vector machines for pattern recognition", *Data Mining Knowledge Discovery,* 2, pp.955–975, 1998.
[18]. B. Schölkopf, A. Smola, *Learning with Kernels*, MIT Press, Cambridge, MA, 2002.
[19]. K.Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, "An Introduction to Kernel-Based Learning Algorithms", *IEEE Transactions on Neural Networks*, vol. 12, pp.181-202, March 2001.
[20]. D. Bersekas, *Nonlinear programming*, Athena Scientific, 1995.
[21]. K. Veropoulos, C. Campbell, N. Cristianini, "Controlling the Sensitivity of Support Vector Machines", *Proceedings of International Joint Conference on Artificial Intelligence,* pp. 55-60, 1999.
[22]. I. Gradshteyn, I. Ryzhik,. *Tables of Integrals, Series, and Products,* 5th ed., Academic Press, p.1101, 1979.
[23]. R. Rifkin, A. Klautau; "In Defense of One-Vs-All Classification", *Journal of Machine Learning Research*, vol. 5, pp.101-141, 2004.
[24]. R. Duda, P. Hart, D. Stork, *Pattern Classification,* (2nd Edition), Wiley-Interscience, 2000.
[25]. E. M. Voorhees, "Implementing Agglomerative Hierarchical Clustering Algorithms for use in Document Retrieval," *Information Processing and Management*, vol. 22, pp. 465--476, 1986.
[26]. J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik, "Feature Selection for SVMs", *Proceedings of NIPS*, 2000.
[27]. Y. Liu Y. Yang, J. Carbonell, **"**Boosting to correct inductive bias in text classification", *International conference on Information and Knowledge Management (CIKM)*, McLean, pp.348 – 355, November 2002.

About the author – Andrey Temko received the B.Sc. and the engineer degree in computer science from Dniepropetrovsk National University, Dniepropetrovsk, Ukraine, in 2001 and 2002, respectively. He is currently a PhD student at TALP Research Center in Universitat Politècnica de Catalunya (UPC), Barcelona, Spain. His main research interests include acoustic event recognition and kernel methods.

About the author – Climent Nadeu received the Telecommunication Engineering degree in 1977 and the Doctoral degree in 1982, both from the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain. Since 1977, he has been with the UPC where he is currently a Professor of signal processing. He has approximately 130 publications in books, scientific journals, and conference proceedings, mainly in the area of speech technologies. Also, he has been Director of the Research Center for Technologies and Applications of Language and Speech (TALP) from 1998 to 2004, and he has been a Visiting Researcher at AT&T Bell Laboratories, Murray Hill (NJ), at the International Computer Science Institute, Berkeley (CA), and at Griffith University, Brisbane (Australia). His current research interests lie in the areas of signal processing, pattern recognition, and multimodal interfaces.