Primary Investigation of Sound Recognition for a domotic application using Support Vector

Mohamed A. Sehili*, Dan Istrate*
Jérôme Boudy**

**LRIT-ESIGETEL, 1 Rue du Port de Valvins,77210 Avon, France*
(Tel: 0331-60-72-70-51; e-mail:{ mohamed.sehili,dan.istrate}@esigetel.fr).
*** Telecom SudParis, 9 Rue Charles Fourier, 91000 Evry, France*
(e-mail: {jerome.boudy, mohamed.sehili}@it-sudparis.eu)

Abstract: The advent of modern communications and the low cost of some kinds of devices have resulted in a desire to equip elderly peoples' homes with sensors to monitor their activities and be forewarned of abnormal situations. In such an environment, sound may represent a rich source of information that can be exploited and this is considered as one of the most ergonomic and least intrusive solutions. However, this solution is often adversely affected by noise that is to say, mostly sounds of a type not taken into account in the creation of this system. Several methods were used to make it possible to classify sounds. In this work we tested Support Vector Machines to classify sounds in a domotic environment.

Keywords: Domotic system, sound classification, pattern recognition, support vector machines.

1. INTRODUCTION

Sound classification is a problem of pattern recognition where one aims to distinguish the class of a given sound from other classes. In a domotic environment, there are many kinds of daily sounds which require detection in order to obtain information about the status of elderly people and their activities. There are also some sounds considered as noise that the system should ignore. Speech is considered as one of the most informative sounds, it is by far the most important class for a telemonitoring system. In fact, a speech signal can carry useful information like emotions and may contain a distress expression. This is what has motivated researchers to attempt sound classification in a hierarchical fashion as in Istrate *et al*. (2009) where speech was first distinguished from other sounds before being transmitted to a second classification engine.

This research work take place in the framework of the Sweet-Home project which searches to provide a domotic HMI based on direct/indirect Speech/Sound recognition. The aim of this project is the safety of the persons and of goods using audio techniques. The interesting sound classes for this project are everyday life sounds (door clap, phone ring, dishes sounds, etc.) and abnormal sounds (screams, glass breaking, object falls, etc.).

The problem of sound classification can be compared to that of speaker identification as both are a multiclass pattern recognition task and rely on extracting and modeling the relevant features from the signal in order to differentiate them. In recent years, several statistical methods which have been successfully used for speaker identification, for example: Hidden Markov Models (HMMs), Gaussian Mixture Models (GMMs) Rose and Reynolds (1995) and Dynamic Time Warping (DWT), were used for sound classification. Previous work of the ANASON team applied GMMs to sound classification following the model described above. A combination of two or more classification methods was also used like in Bourouba *et al*. (2007) and Zhou *et al*. (2007).

Support Vector Machines (SVMs) is a hyperplane based method that has gained increasing attention in the pattern recognition community over the last few years and has been successfully applied to tasks like speaker identification and verification, and face recognition. From a theoretical point of view, this discrimination method is quite robust. For a linear classification problem, it attempts to choose a hyperplane that best separates data points from two classes. Moreover, it has been shown to perform a non-linear classification with accuracy via the use of appropriate Kernel functions. This makes the SVMs extremely valuable for the task of sound classification.

2. SUPPORT VECTOR MACHINES

SVMs belong to the family of binary classifiers. That means that an SVM attempts to assign one of two labels to data points from two distinct classes. The goal is to assign the exact label to each point given a set of labelled examples used to train the classifier.

The basic idea behind this method is to find a decision surface hyperplane which maximizes the margin between positive and negative examples. This implements the principle of structural risk minimization (SRM) (Burges,

1998) (Fig. 1). The hyperplane H_0 and the points which are mapped on it satisfy:

$$w \cdot x + b = 0 \quad (1)$$

The vector w is the normal to the hyperplane and b is the bias of the hyperplane from the origin. Given a set of N training examples (x_i, y_i) where are the points, and $y_i \in \{-1,+1\}$ are the associated labels, we need to find the maximum margin subject to the constraints:

$w \cdot x_i - b \geq 1$ for $y_i = 1$, and

$w \cdot x_i - b \leq -1$ for $y_i = -1$, which can be written as:

$$y_i(w \cdot x_i + b) - 1 \geq 0, \forall i \quad (2)$$

We find that the distance between the two margins H_1 and H_2 is $2/\|w\|$. Thus, the problem can be stated as minimize $\|w\|$ subject to (2).

The problem can be put as a quadratic programming problem as follows:

$$L_p = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} \alpha_i y_i (x_i \cdot w + b) + \sum_{i=1}^{N} \alpha_i \quad (3)$$

where the α_i's are the Lagrange multipliers.

In Fig. 1 it can be seen that few examples can be found on the margins H_1 and H_2. These are the support vectors and their associated α_i's are greater than 0.

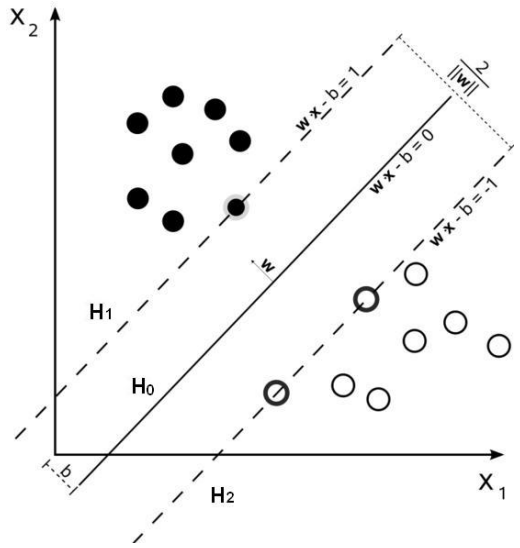

Fig. 1. Example of a linear classifier.

In most cases the data examples are not perfectly separable. In other words, there exists no hyperplane that can separate all points without making any erroneous classification. This has motivated to introduce *slack* variables, ξ_i, to allow some degree of misclassification for some examples while still maximizing the distance to the nearest cleanly separated examples. The problem becomes:

Minimize:

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \xi_i \quad (4)$$

subject to $y_i(w \cdot x_i + b) \geq 1 - \xi_i, \forall i$,

where C is the penalty parameter of the error term.

The above theory works well as long as the data is linearly separable. In many problems, including sound classification, the data is far from being linearly separable. To deal with such problems one solution is to map the data into an extremely high dimensional feature space so that a linear separation becomes possible. However, dealing with data from a high dimensional feature space can easily lead to high computation costs (Picone and Ganapathiraju, 2000). This can be avoided by using Kernel functions. Typically used Kernel functions are (Rose *et al*., 1995):

Linear: $K(x, y) = x \cdot y$ (5)

Polynomial: $K(x, y) = (\gamma x \cdot y + c)^p$ (6)

Gaussian RBF: $K(x, y) = \exp(-\Gamma|x - y|^2)$ (7)

The final decision function takes the form:

$$f(x) = \sum_{i=1}^{N_{sv}} \alpha_i y_i K(x, x_i) + b \quad (8)$$

and the sign of the function f gives the label of the input vector x.

3. APPLICATION TO SOUND CLASSIFICATION

3.1 Multiclass classification

In most cases a system has to deal with more than two classes of sounds. However SVM is a binary classification method. Although there exists a variant of SVM which can do multiclass classification, most researchers prefer splitting the problem into multiple binary problems and then using a binary classifier for each problem. There are two schemes most commonly used to do this; the one-to-all scheme and the one-to-one scheme. In the one-to-all scheme, C classifiers are created to represent C classes. Each classifier is trained by labeling examples from one class as $+1$ and examples from all the other classes as -1. An input example is thus evaluated using all the classifiers and is attributed to the class that yields the best distance. In the one-to-one scheme, a classifier is trained for each couple of classes and the final decision is achieved using a tree structure or a Directed Acyclic Graph (DAG) (Hyun-Chul *et al*, 2003; Seong-Whan and Byun, 2003).

In most cases, a sound consists of more than one vector (i.e. frame). In Zhaohui *et al*. (2006), where SVMs are applied to speaker identification, the score of an utterance

of N vectors is simply the arithmetic mean of the scores of the vectors it contains:

$$S = \frac{1}{N}\sum_{j=1}^{N}(\sum_{i}\alpha_i y_i K(x_j, x_i) + b) \quad (9)$$

Nevertheless we can also classify a sound using a majority voting on its vectors. This technique allows avoiding the influence of only some vectors misclassified.

Another way to use SVMs is to use an ensemble of classifiers. This may be very fruitful for sound classification especially when data is noisy. The idea is to obtain a set of classifiers for the same classification problem (Zhaohui *et al*., 2006). This can be achieved using bootstrapping or boosting (Hyun-Chul *et al*., 2003).

3.2 Acoustical parameters

The SVM are not applied directly on the time signal but on spectral extracted vectors named acoustical parameters. The acoustical parameters can be the MFCC (Mel Frequency Cepstral Coefficients), LFCC (Linear Frequency Cepstral Coefficients), LPC (Linear Prediction Coefficients), LPCC (Linear Prediction Cepstral Coefficients), etc. In this paper we have used LFCC because are more adapted for sound with high frequencies components. LFCCs are cepstral coefficients commonly used in speaker/speech recognition systems.

Their success is due to their ability to represent the speech amplitude spectrum in a compact form. They are commonly calculated as shown in Fig. 2.

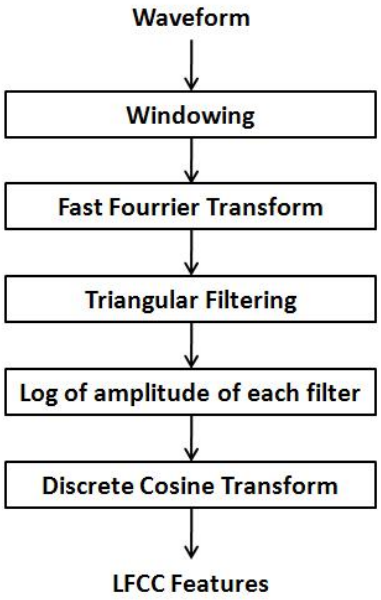

Fig. 2. Steps to derive LFCC.

In the first step the signal is divided into frames, usually by using a rectangular windowing function at fixed intervals and overlap. Thus, each frames can be considered as a cepstral feature vector. The discrete Fourier Transform is then applied to each frame and triangular filter of uniformly spaced frequency bins are applied (see Fig. 3). The logarithm is computed on each output energy of each triangular filter. The components are finally decorrelated using the Discrete Cosine Transform. This has the advantage to reduce the final number of features in each vector.

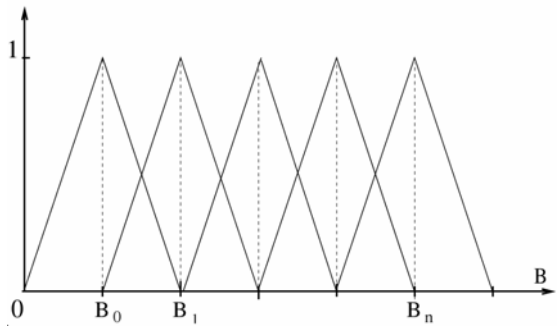

Fig. 3. Uniform Frequency scale.

4. FIRST EXPERIMENTS

In order to experiment with SVMs for sound classification we have used the SVM-light library (Joachims, 1998). We first made a test on a part of the dataset created by the ANASON team. The dataset consists of seven categories of sound related to daily human activities. Table 1 shows the classes used in these experiments.

Table 1. Classes of sound from the dataset.

Sound category	Number of files
Cough	42
Door bell	14
Laugh	10
Sliding door	19
Sneeze	26
Snore	20
Yawn	21

These sounds are 16kHz, 16 bits wav files. For this test, 24 order LFCCs (Linear Frequency Cepstral coefficients), energy and Zero Crossing Rate (ZCR) features were used. The frames were 16 ms of length with an overlap of 50%. Before explaining how to calculate LFCCs we first introduce briefly MFCCs (Mel-Frequency Cepstral coefficients.

The multiclass scheme used is the one-to-one, so a classifier is trained for each pair of classes. For each class, 50% of files are used for training and 50% for testing. To attribute a sound to a class, we first used the method consisting of calculating the sum the scores obtained by its vectors and then and then choosing the class according to the sign of the sum. This strategy yielded poor results. We then adopted a majority voting strategy which improved them moderately.

5. RESULTS

The proposed algorithm was evaluated on the data base through the good classified rate. The accuracy of the whole database is obtained by dividing the number of

correctly classified files by the total number of files. Table 2 shows the results obtained.

The method used is time consuming because of the nonlinear kernel (RBF in our case) where almost all training examples are retained as support vectors. This results in huge models.

In order to better use SVMs and improve the performances, many methods can be used to train a model like the use of hold-out set or cross-validation (Platt, 1999). In this work we used the techniques described in Chang *et al*. (2003) which consist in scaling, grid search and cross-validation.

The goal of scaling is to constraint each feature value to be in a specific range, for example $[-1,+1]$ or $[0,1]$.

This has the advantage to avoid features with greater values dominating those smaller values and to avoid numerical difficulties during calculation (Chang *et al*., 2003). A grid search is used to find the couple of C and Γ, which achieve the best accuracy on training data. Many combinations of these two parameters are thus used to train and test a classifier. One way to do this is to split the training data into two parts, train a classifier using one part and use the rest of data to determine which values of C and Γ allows for better performance.

A better way to determine the best parameters is to use cross-validation. In n-fold cross-validation the straining dataset is split into n subsets of equal size. Each subset is then used to test the classifier trained on the other subsets. In our experiments we used 5-fold cross validation. Table 3 shows the results obtained after using the procedures above. It can be seen that these results outperform the previous one.

Furthermore, in table 3, and contrary to table 2, the performances of the two strategies are almost comparable. This fact is due to scaling the data before training and test. We have also noticed that the models obtained after scaling the data are fairly of smaller size than those obtained with non scaled data. This may be very interesting for real time systems as the time required to classify one vector is closely related to the size of the model.

Table 2. The scores obtained with the first tests using two strategies of classification

	Classification strategy	
	Score sum	Majority voting
Cough	0.33	0.57
Door bell	0.57	1.00
Laugh	1.00	1.00
Sliding door	0.00	0.00
Sneeze	0.15	0.23
Snore	0.40	0.80
Yawn	0.18	0.18
Whole dataset	0.31	0.48

Table 3. The scores obtained after scaling the data and using cross-validation

	Classification strategy	
	Score sum	Majority voting
Cough	0.90	0.95
Door bell	1.00	1.00
Laugh	1.00	1.00
Sliding door	0.20	0.00
Sneeze	0.38	0.38
Snore	0.70	0.70
Yawn	0.18	0.18
Whole dataset	0.61	0.60

6. CONCLUSIONS

This paper presents an application of SVMs to classify sound in a domotic environment. The sound classification is a multiclass problem but SVM are binary classifiers; two techniques was used one-against-one and one-against-all. The use of techniques like scaling and detecting the best parameters by using cross-validation allows improving the performances. Although the first obtained results are encouraging, there are still several methods that can be used to better exploit SVMs and deal with the noise like the use of ensemble of classifiers with bootstrapping or boosting.
Future tests will aim to evaluate the noise influence on the SVM recognition performances and also the possibility to combine GMM with SVM in order to obtain a better system through score fusion.

ACKNOWLEDGMENTS

We would like to thank the ANR (French National Research Agency) and, especially VERSO program, for funding the Sweet-Home project, the framework of this research activity.

Note that conferences impose strict page limits, so it will be better for you to prepare your initial submission in the camera ready layout so that you will have a good estimate for the paper length. Additionally, the effort required for final submission will be minimal.

REFERENCES

Burges Christopher, J. C. (1998). A tutorial on support vector machines for pattern recognition, *Data Min. Knowl. Discov*, pp. 121–167.

Bourouba Hocine, Djemili Rafik, and Mouldi Bedda (2007). A hybrid gmm/svm system for text independent speaker identification, *Int. J. of Computer Science and Engineering*, pp. 22–28.

Chang Chih-Chung, Chih-Wei Hsu, and Chih-Jen Lin (2003). *A practical guide to support vector classification.*

Hyun-Chul Kim, Shaoning Pang, Hong-Mo Je, Daijin Kim, and Sung Yang Bang (2003). Constructing support vector machine ensemble, *Pattern Recognition*, pp. 2757 – 2767.

Istrate, D., Rougui, J.E., and Souidene, W. (2009). Audio sound event detection for distress situations and context awareness, *31st Annual Int. Conference of the IEEE EMBS*, pp. 3501–3504.

Joachims Thorsten (1998). *Making large-scale support vector machine learning practical*.

Picone Joseph and Ganapathiraju Aravind (2000). *svm/hmm architectures for speech recognition*.

Platt John, C. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods, *in Advances in Large Margin Classifiers*, MIT Press, pp. 61–74.

Rose Richard, Reynolds C., and Douglas, A. (1995). Robust text-independent speaker identification using gaussian mixture speaker models, *IEEE Transactions on Speech and Audio Processing*, pp. 72–80.

Seong-Whan Lee and Hyeran Byun (2003). A survey on pattern recognition applications of support vector machines, *Int. J. of Pattern Recognition and Artificial Intelligence*, pp. 459–486.

Zhaohui Wu Zhenchun Lei, Yingchun Yang (2006). Ensemble of support vector machine for text-independent speaker recognition, *Int. J. of Computer Science and Network Security*, pp. 163–167.

Zhou Xianzhong Luo, Wen He Xin, and Guo Ling (2007). Hybrid support vector machine and general model approach for audio classification, *ISNN'07 - Proc. of the 4th Int. Symp. on Neural Networks*, pp. 434–440.