Recognition and Rejection Performance in Wordspotting Systems Using Support Vector Machines

Yassine Ben Ayed, Dominique Fohr, Jean Paul Haton, Gérard Chollet
ENST, CNRS-LTCI, 46 rue Barrault, F75634 Paris cedex 13
LORIA/INRIA-Lorraine, BP239, F54506, Vandœuvre
France

Abstract: - Support Vector Machines (SVM) is one such machine learning technique that learns the decision surface through a process of discrimination and has a good generalization capacity [6]. SVMs have been proven to be successful classifiers on several classical pattern recogntion problems [9, 11]. In this paper, one of the first applications of Support Vector Machines (SVM) technique for the problem of keyword spotting is presented. It classifies the correct and the incorrect keywords by using linear and Radial Basis Function kernels. This is a first work proposed to use SVM in keyword spotting in order to improve recognition and rejection accuracy. The obtained results are very promising. The Equal Error Rate (EER) for the linear kernel is about 16,34% compared to 15,23% obtained by the radial basis function kernel.

Key-Words: - speech recognition, keyword spotting, hidden Markov model, support vector machines, radial basis function kernel, linear kernel

1 Introduction

Significant progress has been made in the development of Automatic Speech Recognition (ASR) technology for continuous speech. However, for widespread consumer applications, handling spontaneous speech, as opposed to strictly prescribed command words and phrases, remains a challenge in the deployment of ASR technology. In particular, the characteristics of spontaneous speech heavily contribute to the acoustic mismatch between speech data used to train a system and the speech input to a system during its operation. Spontaneous speech tends to contain out-of-vocabulary words and disfluencies such as filled pauses and false starts.

A recognizer must thus be able, in first time to spot a keyword embedded in speech, in second time to reject speech that does not include any valid keyword. However, word spotting and word rejection are interrelated such that a good word-spotting capability necessarily implies a good rejection performance.

Several rejection approaches have been suggested in previous research efforts, for rejection of putative hits in keyword spotting, for detection of out-of-vocabulary, and for utterance rejection [15, 16]. For example, we can find the filler (or garbage) models which are generally employed to act as a sink for out-of-vocabulary events and background noises. Confidence measure is also supposed to represent the reliability of the hypothesis. It can be used in a post-processing procedure that rejects the less reliable hypotheses by thresholding the computed confidence measure[8, 10].

In this paper, a new support vector machine based method is proposed for keyword spotting. The SVMs are gaining popularity due to many attractive features and promising empirical performance. The formulation embodies the Structural Risk Minimisation (SRM) principle, as opposed to the Empirical Risk Minimisation (ERM) approach commonly employed within statistical learning methods. SRM minimises an upper bound on the generalisation error, as opposed to ERM which minimises the error on the training data. It is this difference which equips SVMs with a greater potential to generalise, which is our goal in keyword spotting. The SVM can be applied to both classification and regression problems.[1, 4, 7, 13].

The organisation of this paper is as follows : Section 2 gives a brief description of the basic principles of the SVM. The details concerning database and recognition system are given in section 3. In section 4, we present the way we use SVM for keyword spotting. Experimental results are described in section 5, and conclusions are given in section 6.

2 Support Vector Machines (SVM)

Support vector machines have been recently introduced as a new technique for solving pattern recognition problems [14, 17, 18]. They perform pattern recognition between two point classes by finding a decision, determined by certain points of the training set, termed support vectors. This surface, which in some feature space of possibility infinite dimension can be seen as a hyperplane. It is obtained from the solution of the problem of quadratic programming that depends on regularization parameter.

2.1 The linear separable case

Consider the problem of separating the set of training vectors belonging to two separated classes,

$$D = \{(x_1, y_1), ..., (x_m, y_m)\},$$

where $x_i \in R^n$ is a feature vector and $y_i \in \{-1, 1\}$ a class label, with a hyperplane of equation :

$$w.x + b = 0$$

.

The goal is to find the hyperplane that separates the positive from the negative examples; the one that maximizes the margin would generalise better as compared to other possible separating hyperplanes.

A separating hyperplane in canonical form must satisfy the following conditions :

$$w.x_i + b \geq 1 \quad if \quad y_i = 1$$

$$w.x_i + b \leq -1 \quad if \quad y_i = -1$$

These can be combined into one set of inequalities

$$y_i(w.x_i + b) \geq 1 \quad \forall i \in \{1, ..., m\}$$

The distance $d(w, b, x)$ of a point x from the hyperplane (w, b) is,

$$d(w, b, x) = \frac{|w.x + b|}{||w||}$$

The optimal separating hyperplane is given by maximizing the margin M given by the equation :

$$M = \min_{x_i|y_i=-1} d(w, b, x_i) + \max_{x_i|y_i=1} d(w, b, x_i) = \frac{2}{||w||}$$

To maximize the margin M, one need to minimize :

$$\Phi(w) = \frac{w^2}{2}$$

The solution to the optimisation problem is given by the saddle point of the Lagrange functional (Lagrangian)

$$L(w, b, \alpha) = \frac{1}{2} w.w - \sum_{i=1}^{m} \alpha_i [y_i(w.x + b) - 1]$$

Where α are the Lagrange multipliers. The Lagrangian has to be minimized with respect to w, b and with $\alpha \geq 0$. This problem can easily be transformed into the dual problem, and hence the solution is given by :

$$\alpha^0 = argmax \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (x_i.x_j)$$

with constraints,

$$\alpha_i \geq 0, \forall i \in 1, ..., m$$

$$\sum_{i=1}^{m} \alpha_i^0 y_i = 0$$

2.2 The non-linear separable case

In this case, the set of training vectors of two classes are non-linearly separable. To solve this problem, Cortex and Vapnik [3] introduce non-negative variables, $\xi_i \geq 0$, which measure the miss-classification errors. The optimisation problem is now treated as a minimization of the classification error [8]. The separating hyperplane must satisfy the following inequality :

$$(w.x_i) + b \geq +1 - \xi_i, \quad if \quad y_i = +1$$

$$(w.x_i) + b \leq -1 + \xi_i, \quad if \quad y_i = -1$$

The generalised optimal separating hyperplane is determined by the vector w, that minimizes the functional,

$$\phi(w, \xi) = \frac{w^2}{2} + C \sum_{i=1}^{m} \xi_i$$

Where $\xi = (\xi_1, ..., \xi_m)$ and C are constants.
The dual problem corresponding to this case is slightly different from the linear separable case, the goal now is to maximize :

$$\sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (x_i.x_j)$$

subject to

$$\sum_{i=1}^{m} \alpha_i^0 y_i = 0$$

$$0 \leq \alpha_i \leq C \quad \forall i \in \{1, ..., m\}$$

2.3 Kernel support vector machines

In the case where a linear boundary is inappropriate, the SVM can map the input vector into a high dimensional space through function $\phi(x)$, where the SVM constructs a linear hyperplane in the high dimensional space.

Since finding the SVM solutions involve the dot products of the sample vectors $x_i.x_j$, kernel functions play a very important role in avoiding explicit producing the mappings, and avoiding the curse of dimensionality, so that $\phi(x_i).\phi(x_j) = K(x_i.x_j)$, i.e., the dot product in that high dimensional space is equivalent to a kernel function of the current space [2, 12] .

In the linear and non-linear cases, the optimal separating hyperplane defined by w^0 and b^0 is determined as follows :

$$H : w^0.\phi(x) + b^0$$

where,

$$w^0 = \sum_{SV} \alpha_i \phi(x_i) y_i$$

and

$$b^0 = 1 - w^0.x_i \quad for \quad x_i \quad with \quad y_i = 1$$

The classification function is :

$$class(x) = Sign(w^0.\phi(x) + b^0)$$

$$class(x) = Sign[\sum_{SV} \alpha_i^0 y_i \phi(x_i).\phi(x) + b^0]$$

$$class(x) = Sign[\sum_{SV} \alpha_i^0 y_i K(x_i.x) + b^0]$$

Some widely used kernels are :

Linear : $K(x, y) = x.y$

Polynomial : $K(x, y) = (x.y + 1)^d$

Radial Basis Function (RBF) :

$$K(x, y) = exp[-\frac{|x - y|^2}{2\sigma^2}]$$

3 Database and recognition system

3.1 Database

For training, we use 5300 sentences of the French BREF80 database, pronounced by 80 speakers. These sentences are recorded at 16 KHz with 16 bits. This is a general purpose database, and the sentences have no relationship with our application.

The test database contains one hour of recording speech of radio braodcast news at 16 kHz. It is segmented into fragments of duration is 20s. This recorded speech have been pronounced by several speakers (different from the speakers of the training database). In our application we choose 10 different keywords.

3.2 Recognition system

The recognizer used in this work is a speaker independent Hidden Markov Model (HMM) system. The modeled unit is the phone, each phone is represented by 3-state, strictly left-to-right, continuous density HMM. A word is represented by the concatenation of phone models. The number of probability density function (pdf) per state is determined during the training phase [5].

The parameterization is based on MFCC (Mel-Frequency Cepstral Coefficients) parameters. The user can modify this parameterization : size of the analyzing window, shift, number of triangular filters, lower and upper frequency cut-off of the filter bank, and number of the cepstral coefficients. Finally the delta (the first derivation) and acceleration coefficients (the second derivation) are added. In the following experiments, the acoustic feature vectors are built as follows : 32ms frames with a frame shift of 10ms, each frame is passed through a set of 24 triangular band-pass filter resulting in a vector of 35 features, namely 11 static mel-cepstral coefficients (C_0 is removed), 12 delta and 12 delta delta coefficients.

In the recognition phase, we adjust parameters in order to have no deletion keywords (as consequence we obtained a great number of insertion keywords).

4 SVM for keyword Spotting

In this section, we describe the way we have utilised the SVM for the keyword detection. After the recognition phase, the goal is to classify a sequence of detected keywords into correct and incorrect keywords.

For each keyword, we compute the frames assigned to each phone state and extract the acoustic features. In order to have good information about the correctness of a word hypothesis, we must use the feature parameters of each word. In this case we choose the logarithm of the acoustic observation likelihood $Log[P(O_t|ph_i)]$ to provide a good

classification.

Where ph_i is the i^{th} phone of the spoken utterance phone sequence. $PH = \{ph_1, ...ph_N\}$, and O_t belongs to $O = \{O_1, ...O_T\}$, which is the acoustic observation sequence for the utterance. Each O_t element is equivalent to $O_t = \{O_{b[t]}, ..., O_{e[t]}\}$, where $b[t]$ and $e[t]$ represent respectively, the beginning and the ending frames of the i^{th} phone.

For each phone in the spoken utterance, we compute the acoustic observation likelihood, $P(O_t|ph_i)$ using Viterbi algorithm, then we obtain for each word a feature vector of dimension equals to the number of phones in it.

Thus, for each keyword (insertion keyword, recognized correct keyword) we compute a feature vector which is used as input for the SVM. However, SVM system need the same number of input for all keywords. For this reason, we fix the size of the input vector SVM as the largest word size. In case of shorter words, we complete the feature vector with zeros.

In our work the insertion keyword belongs to the class labelled -1 and the correct keyword is assigned to the class labelled +1. Thus, we classify the correct and the incorrect keywords.

The SvmFu package was used for these experiments. It is available as freeware on : http://www.kernel-machines.org/

5 Experimental Results

The database used in the second phase of our experiments (after the recognition phase) is composed of 600 keywords for the training data, and 560 keywords for the test data. In this work, we use linear and Radial Basis Function (RBF) kernels.

To evaluate the performances of our recognizers, we use two evaluation rates :

- The False Acceptance Rate also called False Alarm Rate (FAR). It is defined by the equation :

$$FAR = \frac{Total\ \ False\ \ Acceptance}{Total\ \ False\ \ Attempts}$$

- The False Rejection Rate (FRR). FRR is defined by the equation :

$$FRR = \frac{Total\ \ False\ \ Rejection}{Total\ \ True\ \ Attempts}$$

Plotting a graph of FRR versus FAR gives a Receiver Operating Characteristics (ROC) graph.

Figure 1: ROC curves on test data using a linear kernel by varying the value of C

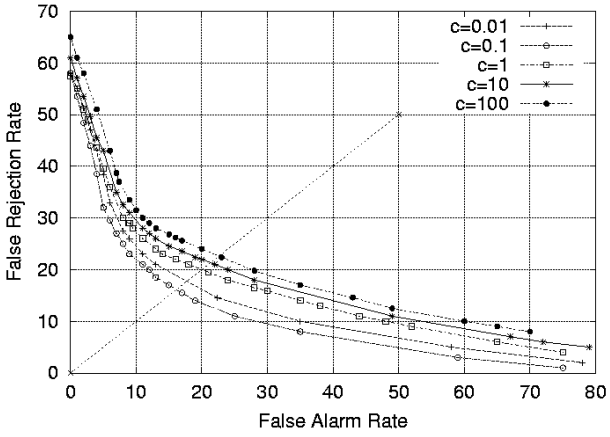
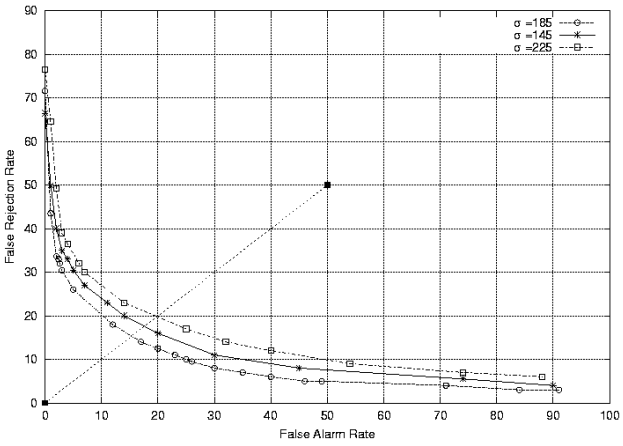


Figure 2: ROC curves on test data using a RBF kernel by varying the value of σ



The resulting ROC curves, using linear SVM by varying the value of the parameter C, $C \in \{0.01, 0.1, 1, 10, 100\}$ are presented in figure 1.

The Equal Error Rate (EER) which is given by FAR=FRR, is about 16,34% obtained by linear kernel in case $C = 0.1$

For the kernel RBF, we use several kernel parameters σ and different parameters C : $\sigma \in \{1, 5, 10, ..., 250\}$ and $C \in \{0.01, 0.1, 1, 10, 100\}$. We try all combinations. We present in Table 1 best results obtained by different values of the pair (C, σ).

Table 1: Recognition accuracy for RBF kernel SVM by varying the value of the pair (C, σ)

(C, σ)	**ACC (in %)**
(0.01, 100)	88,96
(0.1, 200)	89,67
(1, 50)	90,56
(1, 90)	*93,23*
(1, 130)	92,52
(**10**, 145)	93.95
(**10**, 185)	**94,66**
(**10**, 225)	93.23
(100, 200)	92,23
(100, 240)	*93,59*
(100, 280)	92,59

As shown in Table 1, a recognition accuracy of 94,66 % was obtained with $\sigma = 185$ and $C = 10$.

According to these obtained results, we decided to choose the value $C = 10$ to achieve next experiments for the RBF kernel SVM.

Figure 2 presents ROC curves corresponding to the performance obtained using RBF kernel by varying the value of the parameter σ, with $C = 10$.

As mentioned, in our experiments, we test several values of σ, but in order to alleviate the figure, we choose to present only the interesting curves for $\sigma \in \{145, 185, 225\}$.

These results demonstrate that the performance of the RBF kernel are better than results obtained for the linear SVM.

The EER concerning the RBF kernel is about 15,23% compared to 16,34% obtained by the linear kernel.

6 Conclusion

This paper presents the results achieved by SVM techniques for the keyword spotting problem using linear and RBF kernels. Taking into account that is a first approach using support vector machine in keyword spotting, the results obtained seem to be very promising. In the near future, feature vector will be adjusted for each keyword in order to have more information about it. Other different kernel types and parameters will be experimented.

References:

[1] C. Burges. A tutorial on support vector machines for pattern recognition. In *Data Mining and Knowledge Discovery*, volume 2(2), 1998.

[2] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. In *Machine Learning*, volume 46, pages 131–159, 2002.

[3] C. Cortes and V. Vapnick. Support vector networks. In *Machine Learning*, volume 20, pages 1–25, 1995.

[4] T. Evgeniou and M. Pontil. Support vector machines: Theory and applications. volume 2049, pages 249–259, 2001.

[5] D. Fohr, O. Mella, and C. Antoine. The automatic speech recognition engine ESPERE : experiments on telephone speech. In *Proc. IEEE Int. Conf. Spoken Language Processing*, 2000.

[6] A. Ganapathiraju. *Support Vector Machines for Speech Recognition.* PhD thesis, Mississipi State University, 2002.

[7] S. Gunn. Support vector machines for classification and regression. In *Technical Report ISIS-1*, 1998.

[8] S. O. Kamppari and T. J. Hazen. Word and phone level acoustic confidence scoring. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, 2000.

[9] J. Kharroubi, D. Petrovska, and G. Chollet. Combining gmm's with suport vector machines for text-independent speaker verification. In *EuroSpeech*, 2001.

[10] B. Maison and R. Gopinath. Robust confidence annotation and rejection for continuous speech recognition. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, 2001.

[11] E. Osuna, R. Freund, and F. Girosi. Support vector machines : Training and applications. In *A.I .Memo 1602, MIT*, 1997.

[12] E. Osuna, R. Freund, and F. Girosi. Training support vector machines : an applications to

face detection. In *Conference on Computer Vision and Pattern Recognition*, 1997.

[13] M. Pontil, R. Rifkin, and T. Evgeniou. From regression to classification in support vector machines. Technical Memo AIM-1649, 1998.

[14] R. Rifkin, M. Pontil, and A. Verri. A note on support vector machine degeneracy. In *Proceedings of the 10th International Conference on Algorithmic Learning Theory*, volume 1720, pages 252–263, 1999.

[15] R. C. Rose and D. B. Paul. A hidden Markov model based keyword recognition system. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, 1990.

[16] R. A. Sukkar and C. H. Lee. Vocabulary independent discriminative utterance verification for nonkeyword rejection in subword based speech recognition. In *IEEE Trans. on Speech and Audio Processing*, volume 4, pages 420 – 429, 1996.

[17] V. Vapnick. The nature of satistical learning theory. In *Springer-Verlag*, 1995.

[18] V. Vapnick. Satistical learning theory. In *Johon Wiley*, 1998.