Improving the Performance of Out-of-vocabulary Word Rejection by Using Support Vector Machines

HUANG Shilei, XIE Xiang, KUANG Jingming

Department of Electronic Engineering
Beijing Institute of Technology, Beijing, P.R.China
jason_huangshl@yahoo.com.cn, xiexiang@bit.edu.cn

Abstract

Support Vector Machines (SVM) represents a new approach to pattern classification developed from the theory of structural risk minimization [1]. In this paper, we propose an approach to improve the performance of confidence measurements for out-of-vocabulary word rejection by using SVM. Confidence measures are computed from the information of n-best candidates and anti-word by a Hidden Markov Model (HMM) based speech recognizer. The acceptance/rejection decision for a word is based on the confidence score which is provided by SVM classifier. And the decision is performed for each word in vocabulary separately. The performance of the proposed SVM classifier is compared with method based on posterior probability and anti-word probability. Experiments of Mandarin command recognition have showed that better performance can be obtained when using the proposed method.

Index Terms: speech recognition, confidence measure, support vector machines

1. Introduction

Recently, as speech recognition is deployed in an increasing number of applications, the system need to be flexible enough to deal with a wide range of user answers and behaviors, such as heavy accent, hesitations, and pause within a word. The system may also receive words that are out of the recognizer's vocabulary definition. So, it's important for a practical system to apply Out-Of-Vocabulary (OOV) words rejection.

A typical HMM-based keyword spotting system consists of two major phases: recognition (detection) and verification [2][3]. Usually in the verification phase, rejection of OOV words is decided by confidence measurement. The generalized confidence score is defined as a product of confidence scores obtained from confidence information sources such as likelihood, likelihood ratio, duration, duration ratio, language model probabilities, supra-segmental information etc [4]. All confidence information sources are converted into confidence scores by a confidence pre-processor.

Support vector machines have already been used to compute confidence measure values by integrating variance information and achieved better performance than traditional techniques[5][6][7].

In this work, information of n-best candidate's probability and anti-word probability was used to compute the score of confidence measurement. The score was obtained from SVM classifier and the SVM classifier is different for each word in vocabulary.

The remainder of this paper is organized as follows: section 2 describes the recognition system, and section 3 gives the introduction of Support Vector Machines. In section 4, the method of applying SVM to confidence measurement is introduced. Database and experimental results are presented in section 5. Finally, conclusions are given in section 6.

2. Recognition System

The whole recognition process in this paper is divided into two stages: in the first stage a HMM based recognizer to provide the original information including scores of n-best candidates and anti-word for second stage process. In the second stage process, confidence measurement is performed based on the score from first stage HMM-based recognizer.

Speaker independent HMM recognizer is used in this system. Acoustic feature used in our experiments were 12 Mel-frequency cepstral coefficients (MFCCs) and logarithmic energy, plus the corresponding delta coefficients which can consist of a 26-dimension vector. The acoustic unit is phoneme, and each phone is represented by 3-state, strictly left-to-right, Gaussian mixture continuous density HMM. 50 context independent phoneme models and silence model are trained from large vocabulary continuous speech corpus, and also 50 anti-phone model constructions from well-trained phoneme models[8]. Then during decoding process, first n-best candidates' probabilities and anti-word probability are obtained. If the word hypothesis of an utterance observation sequence $\boldsymbol{O}$ is $\boldsymbol{w}$, HMM output probability vector of $\boldsymbol{w}$ is defined as.

$$v_0(w) = [p_1(w), p_2(w), \cdots p_N(w), p_A(w)] \tag{1}$$

Where $\boldsymbol{p_i}$ is log probability of $\boldsymbol{i}$'th best candidate's probability, and $\boldsymbol{p_A}$ is the log probability of anti-word (according to the word hypothesis $\boldsymbol{w}$) probability. Sometimes the HMM who output the maximum probability perhaps is not the correct word. Although for normal recognition system, the result according to this HMM will be the answer, for more practical system, confidence measure based on the HMM decoding outputs will give better performance.

Then vector $\boldsymbol{v_0}$ is normalized by the frame number of each word. And distribution of duration of each word is not taken into account.

$$\begin{aligned} v_1(w) &= [p_1(w), p_2(w), \cdots p_N(w), p_A(w)] / L(w) \\ &= v_o(w) / L(w) \end{aligned} \tag{2}$$

where $\boldsymbol{L(w)}$ is length of $\boldsymbol{w}$.

In the verification stage, confidence measure is applied to decide whether the recognition result from first stage is an out-of-vocabulary word or not. There are many different ways to compute confidence measure of each word hypothesis by combining variant scores [2][3]. The confidence measurement in this paper is based on n-best candidates and anti-word probability. Generally, the confidence measure is defined as.

$$CM(w) = f(v_1(w)) \tag{3}$$

In Section 4, different methods of computing this value are discussed and compared.

3. Classification based on Support Vector Machines

Consider the problem of separating the set of m training vectors belonging to two different classes:

$$\{(x_1, y_1), ..., (x_m, y_m)\} \tag{4}$$

where $\boldsymbol{x_i} \in \boldsymbol{R^n}$ is a feature vector and $\boldsymbol{y_i} \in \{-1,+1\}$ is a class label, with a hyperplane of equation $\boldsymbol{w \cdot x + b = 0}$. Of all the boundaries determined by $\boldsymbol{w}$ and $\boldsymbol{b}$, the one that maximizes the margin would generalize better, as compared to other possible separating hyperplanes.

In the non-linear separable case, the set of training vectors of two classes are non-linearly separable. To solve the problem, Cortes and Vapnik [9] introduced non-negative variables, $\boldsymbol{\xi_i \geq 0}$, which measure the miss-classification errors. The optimization problem is now treated as minimization of the classification error [1]. The separating hyperplane must satisfy the following inequalities.

$$\begin{aligned} (w \cdot x_i) + b &\geq +1 - \xi_i, \text{ if } y_i = +1 \\ (w \cdot x_i) + b &\leq -1 + \xi_i, \text{ if } y_i = -1 \end{aligned} \tag{5}$$

The generalized optimal separating hyperplane is determined by the vectors, which minimize the function:

$$\Phi(\omega, \xi) = \frac{w^2}{2} + C\sum_{i=1}^{m} \xi_i \tag{6}$$

where, $\boldsymbol{\xi = (\xi_1, \dots, \xi_m)}$ and $\boldsymbol{C}$ are constants.

If a linear boundary is inappropriate, the SVM replaces the inner product $\boldsymbol{x_i \cdot x_j}$ of the classification function by a kernel function $\boldsymbol{K(x_i, x_j)}$, and then constructs an optimal hyperplane in the mapped space. Kernel function plays a very important role in avoiding explicit production of the mappings and the curse of dimensionality. There are several possible kernel functions such as linear function, polynomial function Radial Basis Function (RBF) and Sigmoid function. For a given kernel function, the classifier is given by:

$$D(x) = \sum_{SV} \alpha_i^0 y_i K(x_i, x) + b^0 \tag{7}$$

$$class(x) = \mathrm{sgn}[\sum_{SV} \alpha_i^0 y_i K(x_i, x) + b^0] = \mathrm{sgn}[D(x)] \tag{8}$$

where $\boldsymbol{SV}$ is support vectors, $\boldsymbol{\alpha_i^0}$, $\boldsymbol{x_i}$, $\boldsymbol{y_i}$ and $\boldsymbol{b^0}$ are parameters of SVM and determined during training.

4. Confidence Measure Using Support Vector Machines

The aim of the confidence measure technique in automatic recognition system is to estimate if the recognized words are correct or incorrect. To compute confidence scores, we simply define:

$$CM_0(w) = (p_1(w) - p_A(w)) / L(w) \tag{9}$$

where $\boldsymbol{w}$ is the word hypothesis after previous recognition.

For each confidence measure, a specific threshold T is set up. If the confidence score is lower than this threshold, the recognition result is rejected:

$$w = \begin{cases} \text{Accept} & \text{if } \mathrm{CM}_0(w) > T \\ \text{Reject} & \text{otherwise} \end{cases} \tag{10}$$

If we look on vector of $\boldsymbol{(p_1(w)/L(w), p_A(w)/L(w))}$ as input vector x to a SVM classifier:

$$x(w) = (p_1(w) / L(w), p_A(w) / L(w)) \tag{11}$$

Class(x) in Eq(8) will be acceptance or rejection of word hypothesis. Moreover, we can use $\boldsymbol{D(x(w))}$ to compute the value of confidence measure and it can be modified as:

$$CM_1(w) = D(x(w)) = \sum_{SV} \alpha_i^0 y_i K(x_i, x(w)) + b^0 \tag{12}$$

$$class(x(w)) = \mathrm{sgn}[D(x(w)) - T] \tag{13}$$

$$w = \begin{cases} \text{Accept} & \text{if } class(x(w)) = +1 \\ \text{Reject} & \text{otherwise} \end{cases} \tag{14}$$

where $\boldsymbol{T}$ is the threshold for accepting/rejecting a word hypothesis.

To achieve better performance, more information should be included in input vector to a SVM classifier:

$$\begin{aligned} x(w) &= v_1(w) \\ &= (p_1(w) / L(w), ...p_N(w) / L(w), p_A(w) / L(w)) \end{aligned} \tag{15}$$

To train SVM classifier for confidence measure, extra database is needed. The utterances in this database were passed through the HMM recognizer and output probability vectors were obtained. And whether a result candidate should be accepted or rejected is marked as label of each output probability vector. Thus all output probability vector could be divided into two classes according to a uniform $\boldsymbol{T}$ value in Eq (13).

But this uniform SVM model for accepting/rejecting a word ignored the different output probability vector distributions of each word in vocabulary. To consider different output probability vector distributions of different words in vocabulary, we train SVM model for each word separately rather than using a single SVM model. And then a uniform threshold $\boldsymbol{T}$ is used to decide whether a result should be accepted or not. And Eq. (13) is modified to:

$$
\begin{aligned}
& class(x(w)) \\
&= \operatorname{sgn}[\sum_{SV(W_0)} \alpha_i^0(W_0) y_i(W_0) K(x_i, x(w)) + b^0(W_0) - T] \\
&= \operatorname{sgn}[D_{W_0}(x(w)) - T]
\end{aligned} \tag{16}
$$

where $SV(W_0)$ is support vectors, $\alpha_i^0(W_0)$, $x_i(W_0)$, $y_i(W_0)$ and $b^0(W_0)$ are parameters of SVM for word (W_0). For a given word sequence O in recognition, W_0 will be the most likely candidate word or word hypothesis.

Thus in the utterance verification process, multiple SVM classifiers will be used rather than single acceptance/rejection classifier based on SVM.

5. Experimental Results

Experiments of speaker-independent Mandarin isolated word recognition were carried out to evaluate the performance of proposed method. The corpus is about 60 speakers' (30 male and 30 female) 52080 utterances with 217 commands for controlling of hand holding device. The length of commands ranges from 2 syllables to 4 syllables. And 100 commands out of 217 commands were looked on as target words, and the rest 117 words were out-of-vocabulary words. The database for training (DB0) SVM model contains 30 speakers' (30 male and 30 female) 26040 utterances (Just half of the whole database), and each of the 100 commands has 120 samples. And the rest database for testing (DB1) also contains 30 speakers' 26040 utterances, each of the 100 commands has 120 samples.

To evaluate the performance of the proposed method, we use two evaluation rates:

The False Acceptance Rate, also called False Alarm Rate (FAR), define as:

$$FAR = \frac{Total\ False\ Acceptance}{Total\ False\ Attempts} \tag{17}$$

The False Rejection Rate (FRR), defined as

$$FAR = \frac{Total\ False\ Rejection}{Total\ True\ Attempts} \tag{18}$$

Plotting FRR versus FAR gives a Receiver Operating Characteristics (ROC) curve, and the Equal Error Rate (EER) is given by FAR=FRR.

In the baseline experiment (denoted as BL), confidence measure is given by Eq(9) The test was based on DB1, and the ROC curve is shown in figure 1, EER is about 32.2%.

The kernel functions used in our experiments are Linear Function and Radial Basis Function:

$$K(x_i, x_j) = x_i \cdot x_j \tag{19}$$

$$K(x_i, x_j) = \exp(-\gamma (x_i, x_j)^2) \tag{20}$$

where γ is a constant.

When applying SVM on vector given by Eq. (9) (denoted as OP1), only the 1-best probability and anti-word probability were passed to SVM. A 2-class SVM classifier (SVM_2) is trained from DB0. The two classes are acceptance-class and rejection-class. Figure 1 shows the ROC curve of SVM_2 with different kernel functions. We also carried out the experiments that 4-best probability and anti-word (OP2) were passed to SVM. The result ROC got from database DB1 is shown in figure 2.

When applying the proposed method, database DB0 for training was separated into many subsets according to a certain command (word). A 2-class SVM classifier is trained on each subset separately. Then for the 100-word vocabulary, 100 SVM classifiers were obtained (SVM_M). In the process of verification, one SVM classifier was chosen to decide whether the word should be accepted or rejected. This SVM classifier is chosen according to the 1-best hypothesis. Different kinds of input vector OP1 and OP2 were tested in experiments. Figure 3 shows the ROC when just using 1-best probability and anti-word probability in the multi-classifier case, and Figure 4 shows the ROC with 4-best probability and anti-word probability. The test database is DB1.

The EER values of all experiments are listed in table 1.

Table 1. *The EERs(%) of different methods.*

Method	OP1	OP2
Baseline	32.2	--
SVM_2 with Linear kernel	31.7	**27.1**
SVM_2 with RBF kernel	**30.6**	27.2
SVM_M with Linear kernel	**26.6**	**21.5**
SVM_M with RBF kernel	27.5	23.1

OP1 means that the input vector for SVM includes 1-best candidate's probability and anti-word probability.

OP2 means that the input vector for SVM includes 4-best candidates' probability and anti-word probability.

6. Discussion

As shown in Table 1, when applying multiple SVM classifiers to confidence measure in verification process, the obtained EER drops obviously. When using 1-best and anti-word probability as input vector passed to SVM, the EER concerning to linear kernel is about 26.6% in the case of multiple SVM classifiers compared to about 30.6% in the case of single SVM classifier (RBF kernel), drops about 13.1%. When using 4-best and anti-word probability as input vector passed to SVM, the EER concerning to linear kernel is about 21.5% in multiple SVM classifiers, compared to about 27.1% with linear kernel, single SVM classifier, drops about 20.7%.

It can also be seen from Table 1, that lower EER was obtained when more information was used in verification. In the case of single SVM classifier, EER drops from 31.7% to 27.1% (linear kernel) and from 30.6% to 27.2% (RBF kernel).In the case of multiple SVM classifiers, EER drops from 26.6% to 21.5%(linear kernel) and from 27.5% to 23.1% (RBF kernel).

In all experiments applying SVM, linear kernel and RBF kernel have close performance, and linear kernel is a little better in most cases.

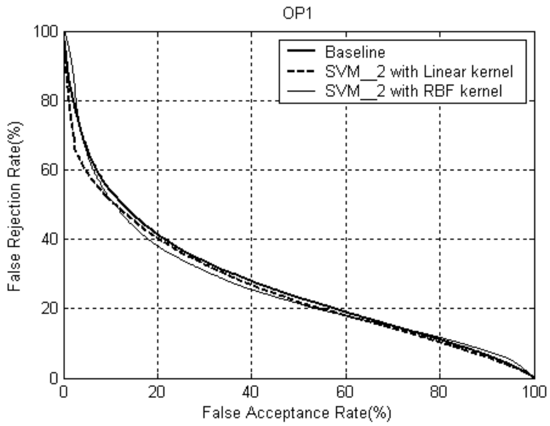

Figure 1 *ROC of one SVM classifier with 1-best and anti-word probability as input.*

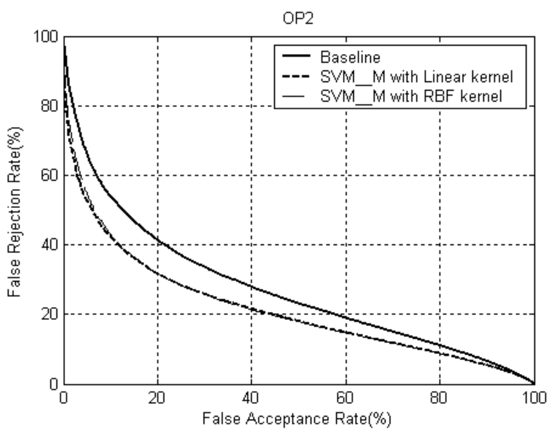

Figure 2 *ROC of one SVM classifier with 4-best and anti-word probability as input.*

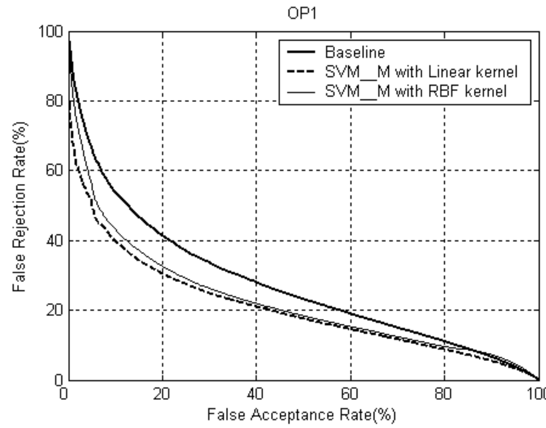

Figure 3 *ROC of multiple SVM classifiers with 1-best and anti-word probability as input.*

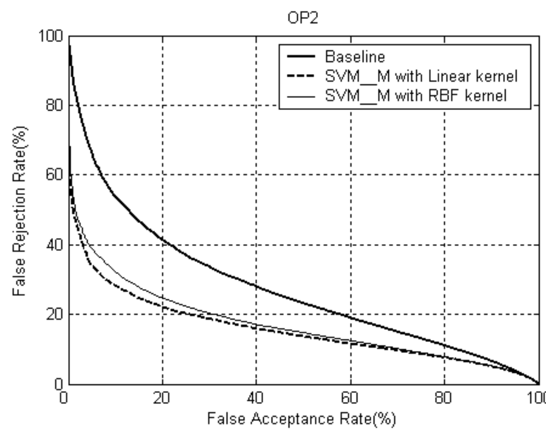

Figure 4 *ROC of multiple SVM classifiers with 4-best and anti-word probability as input.*

7. Conclusions

In this paper, we have proposed a new method to compute scores of confidence measure based on SVM and applied it to a command recognition system. Experiments results have shown that the proposed method achieved lower EER compared to conventional method. And the more information is used in computing confidence measure, the better performance will be achieved.

8. Acknowledgements

The research was supported in part by National Nature Science Foundation of P.R.China under Grant NSFC60372089.

9. References

[1] V. Vapnik, Statistical Learning Theory, John Wiley and Sons, 1998.
[2] Yoma N.B. et al, "Bayes-based confidence measure in speech recognition", Signal Processing Letters, IEEE, Vol 12, Issue 11, p745-748, 2005.
[3] Myoung-Wan Koo et al, "A new decoder based on a generalized confidence score", Proc of ICASSP '98, Vol 1, p213–216 1998.
[4] M. W. Koo, "An utterance verification system based on subword modeling for a vocabulary independent speech recognition system," in 6th European Conference on Speech Communication and Tech, pp. 287-290 1999.
[5] Benayed, Y et al, "Improving the performance of a keyword spotting system by using support vector machines", ASRU 2003 IEEE, p145–149, 2003.
[6] Benayed et al, "Confidence measures for keyword spotting using support vector machines" Proc of ICASSP 2003, Vol 1, p588-591 2003.
[7] Takehito Utsuro et al, "Confidence of agreement among multiple LVCSR models and model combination by SVM". Pro of. ICASSP 2003. Vol 1, pI-16-19, 2003.
[8] B Yan et al, "An approach of keyword spotting based on HMM", Proc of the 3rd World Congress on Intelligent Control and Automation,Vol 4, p2757–2759 2000.
[9] C. Cones and V. Vapnik, "Support-vector networks:' Machine Learning, vol. 20, no. 3, p273-297 1995.