A SUPPORT VECTOR MACHINES-BASED REJECTION TECHNIQUE FOR SPEECH RECOGNITION

Changxue Ma, Mark A. Randolph

Human Interface Laboratory Motorola Labs
1301 E. Algonquin Rd.
Schaumburg, IL 60196, USA
{Changxue.Ma, Mark.Randolph}@Motorola.com

Joe Drish

University of California, San Diego
Dept. of Computer science and Eng.
9500 Gilman Dr. LA JOLLA
CA 92037,USA

ABSTRACT

Support Vector Machines represent a new approach to pattern classification developed from the theory of Structural Risk Minimization[1]. In this paper, we present an investigation into the application of Support Vector Machines' to the confidence measurement problem in speech recognition. Specifically, based on the results from an initial decoding of an utterance during speech recognition, we derive a feature vector consisting of parameters such as word score density, N-best word score density differences, relative word score and relative word duration as input to the confidence measurement process in which hypothetically correct utterances are accepted and utterances determined to be incorrect are rejected. We propose a new approach to training Support Vector Machines. In this paper, we have trained and tested a Support Vector Machines classifier and compared the results with other statistical classification methods.

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, for example, is often ungrammatical; it tends to contain out-of-vocabulary words and dysfluencies such as filled pauses and false starts. Nonetheless, speech utterances having these characteristics are an element of human language behavior; therefore ASR technology that can gracefully handle spontaneous speech would contribute greatly towards user-friendly voice applications.

In a spoken language system, the characteristics of spontaneous speech can be modeled at several levels. For example, at the syntactic and semantic levels, *word spotting* grammars can be used to represent spontaneous speech and applied during a ”robust parsing” stage of processing after speech recognition is completed. At the acoustic level, *utterance verification* is an essential part of the speech recognition process for filtering out utterances that can not reasonably be accepted by the domain grammar[4, 5, 6]. During utterance verification the objective is to incorporate in the recognizer the ability accept *keywords* (i.e., words that are within the domain) and ignore or *reject* non-keywords.

Utterance rejection techniques are typically based on the by-products of the speech recognition decoding process. Specifically, a collection of features obtained from the decoder are combined and used to perform a 2-way classification of an utterance: ”correct” vs. ”incorrect”. The features that are selected and used in this classification assess the recognizer's *confidence* in its results and are often based on heuristics. For example, one of the confidence measurements that we define and use in this paper is based on having two sets of acoustic models: keyword models and non-keyword, *filler* or *garbage* models. The acoustic score of a keyword normalized by the acoustic score of the garbage models is used to detect the occurrence of non-keywords in an utterance, and therefore can be used to decide whether to reject the utterance.

In this paper we focus on two issues. First, in the next section, we propose a set of features for confidence measurement. In addition to the above-mentioned normalized acoustic score, we propose other score-related features, and in addition features based on word and speech segment duration. The second major area of focus is the problem of designing an effective classifier. One of the key characteristics of a classifier is its ability to *generalize* or form a robust classification rule based on a small number of training tokens. Support Vector Machines (SVM's), a relatively new approach to pattern classification developed from the theory of Structural Risk Minimization[1], have been shown to have a greater ability to generalize in comparison to other statistical classification methods. For this reason, SVM's have been successfully applied to problems in image and speech classification[2, 3]. In this paper, we'll compare and contrast SVM-based classifiers with other classification methods such as neural networks, logistic regression, polynomial methods and linear discriminant analysis.

2. WORD LATTICE BASED FEATURES

The features we use for confidence measurement in our investigation are based on having first derived a word lattice for an utterance. From this lattice we will extract features that are based on scores and duration information.

2.1. Word-lattice Based Score Normalization

Continuous speech recognition is based on the fundamental probability relation

$$\tilde{w} = ARGMAX_w \frac{p(\mathbf{o}|w)p(w)}{p(\mathbf{o})} \tag{1}$$

$\tilde{w}$ is the maximum likelihood word sequence computed from $p(\mathbf{o}|w)$, the conditional probability (or likelihood) of the acoustic sequence given a hypothesized word string w, $p(w)$, w's *a priori* probability (typically obtained from a language model), and $p(\mathbf{o})$, the *a priori* probability of the acoustic sequence. The Viterbi decoding process finds the best word sequence according to the likelihood weighted by the language model score. That is, Equation (1) is evaluated with the denominator $p(\mathbf{o})$ being ignored. Since the *a priori* probability $p(\mathbf{o})$ is the same for all possible word sequences for a given utterance, the value of $p(\mathbf{o})$ will not change the rank order of N-best word sequence hypotheses output from a speech recognizer. However, the Viterbi score (or likelihood) for the word sequence w is not an absolute measure of probability and therefore cannot be reliably used to determine the goodness of match for a word to a sequence of acoustic observations. For measuring confidence or the probability of a word being correctly recognized, we need to use normalized Viterbi scores. We can define a confidence measure as

$$C = \frac{p(o|\tilde{w})p(\tilde{w})}{\sum_w p(o|w)p(w)} = \frac{\prod_i p(o_{t_i,t_{i+1}}, w_i)p(w_i, \sigma_i|\sigma_{i-1})}{\sum_w \prod_i p(o_{t_i,t_{i+1}}|w_i)p(w_i, \sigma_i|\sigma_{i-1})} \tag{2}$$

Alternatively, Equation (2) can be re-written as

$$C = \frac{\prod_i p(o_{t_i,t_{i+1}}, w_i)p(w_i, \sigma_i|\sigma_{i-1})}{\sum_h \prod_i p(o_{t_i,t_{i+1}}|h_i)p(h_i, \phi_i|\phi_{i-1})} \tag{3}$$

where σ_i, ϕ_i and s_i denote language model states, phone states and HMM states respectively. Equation (3) suggests that we can expand the denominator of the confidence formula above into phone-level or HMM state-level representations.

The confidence measure defined in Equation (3) can be applied to the best word path, where the best path is obtained from Viterbi decoding. However, although the best path is the optimal choice globally, some local word scores can be very low while others can be very high. As a consequence, rejection of an utterance based on the entire path score could be misleading. Alternatively, we should apply this confidence formula to individual words. Specifically, using the word lattice, we can calculate the normalized word scores using Equation (3) by allowing the denominator of this formula to be the sum of scores for all possible paths through the lattice; the numerator is simply the score corresponding to the best path.

2.2. Relative Duration and Relative Score

From the segmentation of word sequences in the lattice, we can find out how many alternative words are hypothesized for each segment of the best path. If a majority of alternative words whose segmentation overlaps with the that of a word in the best path have the same identity, it is reasonable to assume this word is probably correctly recognized. Based on this idea, we propose to use two other features from the word lattice, the relative duration and relative score for each word on the best path.

Basically, for each word on the best path, all words in the lattice will contribute to the total duration by the amount proportional to their overlapping part with the best word. And all words in the lattice with the same identity as the best word will contribute to the best word duration by the amount proportional to their overlapping part with the best word. The same process can be also applied to the word scores. We can also calculate the confidence measure based on the frame based score difference between the top two choices. The score per frame is computed as the global score divided by the frame number of the whole utterance.

2.3. Score differences of the top word choices

The score difference feature is obtained by re-scoring the best word segments with alternative words. These alternative words are extracted from the word lattice whose spectral features overlaps with the time span of the best word. If the scores from the best word and the maximum score of the alternative words are close, we can argue that these two words are confusable and the best word should be assigned a low confidence score. Otherwise, we can assign a high confidence score to it. Score differences can also be calculated between the best word score and average score in a phone lattice. As we have stated above, the phone lattice can be generated with looser constraints than the word lattice. This phone lattice could absorb the garbage words and out of vocabulary words. The phoneme scores can better indicate how well the acoustic features fit the phoneme models. When word scores are used, each sub-word unit contributes equally to the total score. In fact, different sub-words should contribute differently to the overall decision.

2.4. Word score per frame

The word score per frame is another attribute which indicates how well the acoustic segment of the word matches the HMM models. A high score indicates that the acoustic features fit the model well and a high confidence score should be given to the word. A low score means the acoustic features do not fit well into the model. This happens when the acoustic features are distorted or models were not trained to cover this situation. For large vocabulary speech recognition, the acoustic units are phoneme based. However, not all the models can be well trained, especially, the stop consonants. The word score per frame is just the average of phoneme scores. We can separate the models into categories each with different weights. The average score can be more indicative for the purpose of rejection.

3. SUPPORT VECTOR MACHINES BASED CLASSIFICATIONS

The features defined in Section 2 can be input into any statistical pattern classifer to decide whether to reject an utterance. In this investigation we focused our attention on the Support Vector Machine (SVM) as a classification method. An SVM learns the decision boundary between two classes by mapping the training sample vectors onto a higher dimensional space and then determining an optimal separating hyper-plane. SVM's create a classifier with minimized VC dimension which defines the capacity of a set of functions[1]. If the VC dimension is low, the expected probability of error is low as well. By not requiring fine-tuning of parameters, SVM's exhibit a greater ability to generalize in comparison to other statistical pattern classification methods.

Consider the training feature vectors of two classes,

$$(\mathbf{x}_i, y_i), \mathbf{x}_i \in R^n, y_i \in \{-1, +1\} \tag{4}$$

The Support Vector Machines (SVM) algorithms will find a pair of parallel optimal hyper-planes, defined as follows:

$$H_1: \; y = \mathbf{w} \cdot \mathbf{x} - b = +1; \{\mathbf{x} \in y_i = +1\}$$

$$H_2:\ y = \mathbf{w}\cdot\mathbf{x} - b = -1, \{\mathbf{x} \in y_i = -1\}$$

to separate the two classes, so that the margin, i.e. the distance between two hyper-planes, $2/\|\mathbf{W}\|$ is the largest. This is the sum of the shortest distance from the hyper-plane to the closest positive and negative examples. The training vectors on the hyper-lanes are called support vectors. The hyper-planes are located by solving the optimization problem:

$$\begin{array}{l} \min \|\mathbf{w}\|^2 + C\sum \xi_i \\ \text{subject to constraints:} \\ \mathbf{w}\cdot\mathbf{x} - b \geq +1 - \xi_i \\ \mathbf{w}\cdot\mathbf{x} - b \leq -1 + \xi_i \end{array} \tag{5}$$

If $\xi_i = 0$, the two classes are linearly separable and there are no data points between H_1 and H_2. If $\xi_i > 0$, the two classes are not linearly separable; for the data violating the maximum margin condition, a penalty controlled by $C > 0$ is given to balance margin maximization and classification errors. Using Wolfe duality theory, the problem can be transformed to the following dual problem:

$$\begin{array}{c} \max \sum_i^N \alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \\ \text{subject to constraints:} \\ \sum_i^N \alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq C \end{array} \tag{6}$$

where α_i are Lagrange multipliers. Therefore:

$$\mathbf{W} = \sum_i^N \alpha_i y_i \mathbf{x_i} \tag{7}$$

In the case where a linear boundary is inappropriate the Support Vector Machines can map the input vector into a high dimensional space through function $\Psi(\mathbf{x})$, where it can construct a linear hyper-plane in the high dimensional space. Since finding the SVM solutions involves the dot products of the sample vectors $\mathbf{x_i} \bullet \mathbf{x_j}$, kernel functions play a very important role in avoiding explicitly producing the mappings, and avoiding the curse of dimensionality, so that $\Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x}_j) = k(\mathbf{x}_i, \mathbf{x}_j)$. That is, the dot product in that high dimensional space is equivalent to a *kernel* function of the current space and the hyper-planes is expressed as $y = \sum_i^N \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) - b$. An example kernel function called Gaussian kernel is: $K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma^2}$

4. HIERARCHICAL SUPPORT VECTOR MACHINES TRAINING

Support Vector Machines exhibit a greater ability to generalize in comparison to other statistical pattern classification methods. However, it is observed that the SVM often consists of many support vectors that can be improperly biased by outliers along the boundaries. The resulting classifier is also suspect when the number of training vectors for one class are much larger than those in another class and when the two classes are not separable. The total penalties for each class could be heavily biased.

We propose to train the Support Vector Machines hierarchically. Starting at the root node of the tree from the training vectors of two classes, the Support Vector Machine is trained with the data centroids of the data from the two classes. After each iteration, according to some rules we split the data associated with some nodes of the tree into two parts and find the centroid for each node. The centroids from the bottom nodes will be used to train the Support Vector Machines. The maximum margin of the Support Vector Machines will increase and saturate in the end. This way we can also reduce the memory usage for training which is a significant problem with training Support Vector Machines. Since we are only interested in pattern classification, we can choose a hyper-plane separating two classes with a certain tolerance region. That means the boundary in the input space can be smoothed by clustering some data points along the boundaries and therefore reducing the number of supporting vectors. In this way, we have trained classifiers for classifying digits. Figure 1 shows the number of support vectors as a function of variance in the Gaussian kernel for two training methods for the classification of *oh vs. zero*.

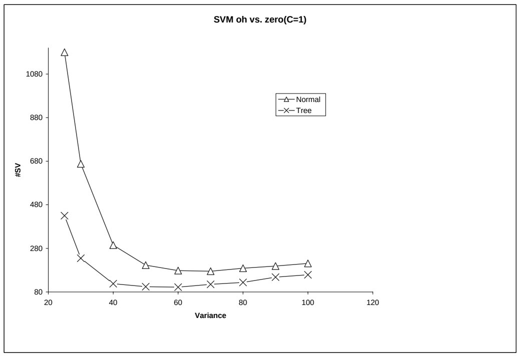

Figure 1: Comparison of the number of support vectors from two training methods.

5. EXPERIMENT I

As we proposed in the above, we have several features, such as the normalized score, score per frame, word duration, speech rate, etc, which can heuristically contribute to determine the acceptability of a word. We can formulate the confidence measure as a statistical modeling problem. Given an acoustic segment o, which has been transformed into a feature set X, and its correctness, we have to find the posterior probability, $P(w = c|X)$. Given the features, there are many ways to generate the conditional posterior probability for the two categories.

We can view this as a parametric statistical modeling of the training experiment results or a pattern classification problem. In the experiment, we collect features from a digit recognition task. For each word, four features such as density score, score difference, relative scores and relative duration are used for the analysis. We have analyzed the results from statistical classification methods such as Neural Networks (NN), Linear Discriminant Analysis (LDA), Logistic regression (LR) and Polynomial Regression(PL). There is a heavy imbalance between the correct and the incorrect examples from the training data, because recognition just has an error rate of a few percent. We expect that the effectiveness of the Support Vector Machines given only a small amount of data would ease the problem. The Gaussian kernel is used in the experiment and the variance is one. Figure 2 shows the histogram of the distance of the feature vectors to the hyper-planes of the Support Vector Machines. We show the comparative performance of the different classifiers we used in the experiment in Figure 3.

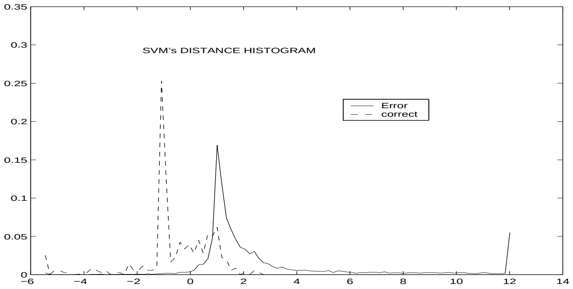

Figure 2: SVMs distance histogram for digit rejection.

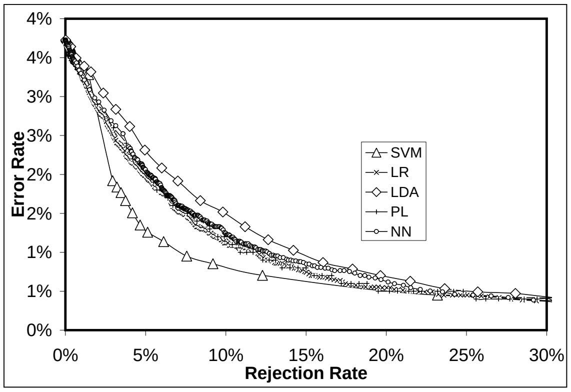

Figure 3: Digit rejection results from SVM, LDA, NN, PL and LR classifiers.

The error rate in the figure is percentage of remaining errors after rejection against the total number of reference words. The rejection rate is the percentage of words being rejected. We see a better result from the Support Vector Machines classification. The Linear Discriminant Analysis is not as good as the other approaches. However, the disadvantage of the Support Vector Machines approach is that it takes much more parameters than other methods we presented in the paper.

6. EXPERIMENT II

Experiments are also performed by using Support Vector Machines classifier to validate the recognition results or to compute the confidence. To classify speech data with SVMs, the speech features have to be encode into a fixed length vector. The alignment information from the Viterbi decoding of speech is therefore used to divide the feature segment for each model into three equal regions and the averages of Mel Cepstral feature vectors of 39 dimension in each region are formed into feature vectors of dimension of 118. The SVM classifers are trained in a *one vs. all* fashion. The target model belongs to one class and other models in the model set belong to another class. In digits recognition twelve classfiers are trained and the posterior probabilities are estimated from outputs of the decision function. For comparison, the Guassian mixture models with two and five mixtures of full covariance matrice are also trained for each model. We have investigated this post validation method on a digits recognition task. We have estimated the distribution of the output of the SVM classifier with Guassian kernel function and those for the classifiers using Gaussina mixture models of a mixture of two and five. The rejection rate vs. error rate classification curve are constructed for all these classfiers. We also show this rate when only the numerical scores generated by the HMM alignment are used for rejection. Figure 4 demonstrates the results of this comparison.

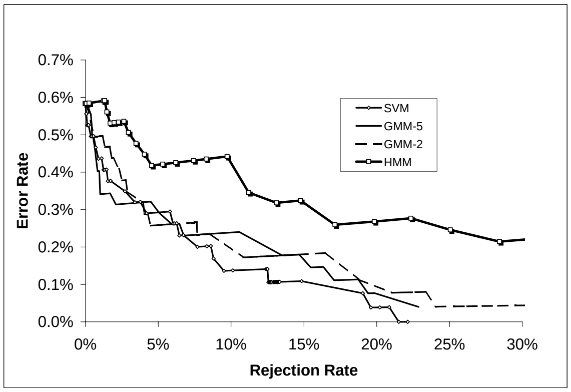

Figure 4: Digits rejection results by post-validation.

7. CONCLUSION

In this paper, we addressed the word rejection problem and proposed the use of features extracted from word lattice for rejection. We applied Support Vector Machines to the classification problem and presented in the paper a new approach to train Support Vector Machines for pattern classification. By Comparison with other classification methods, we showed that Support Vector Machines approach performs better. However, this approach requires more storage for parameters. Further research is needed to apply this approach to more complex speech recognition tasks.

8. REFERENCES

[1] Vapnick, V. *The Nature of Statistical Learning Theory.* Springer-Verlag, New York, 1995.

[2] Clarkson, P., and Moreno, P.J. "On the use of support vector machines for phonetic classification," ICASSP-99, pp: 585 - 588 vol.2

[3] Niyogi, P., Burges, C., and Ramesh, P. "Distinctive feature detection using support vector machines," ICASSP-99, pp: 425 - 428 vol.1

[4] Rose, R.C., Yao, H., Riccardi, G. and Wright, J. "Integrating multiple knowledge sources for utterance verification in a large vocabulary speech understanding system," IEEE Proceeding ASRU, 1997. pp.215-222.

[5] Rahim, M.G., Chin-Hui Lee, Biing-Hwang Juang, and Wu Chou, "Discriminative utterance verification using minimum string verification error (MSVE) training," ICASSP-1996, pp.3585 - 3588, vol. 6.

[6] Rahim, M.G., Chin-Hui Lee, and Biing-Hwang Juang, "Robust utterance verification for connected digits recognition," ICASSP-95, pp: 285 - 288 vol.1