Support Vector Machines for Segmental Minimum Bayes Risk Decoding

Veera Venkataramani and Sourin Das
Center for Speech and Language Processing
Department of Electrical and Computer Engineering
Johns Hopkins University
Baltimore, MD 21218
{veera,sourin}@jhu.edu

Abstract

Segmental Minimum Bayes Risk (SMBR) Decoding is an approach whereby we use a decoding criterion that is closely matched to the evaluation criterion (Word Error Rate) for speech recognition. This involves the refinement of the search space into manageable confusion sets (ie, smaller sets of confusable words). We propose using Support Vector Machines (SVMs) as a discriminative model in the refined search space. The hope is we will be able to use SVMs effectively when the search problem is broken down into sequence of independent and simpler problems. Our first approach will be to use SVMs to make hard decisions, ie, the SVMs will output a word for each confusion set. We will then show that on using a simple voting scheme we improve upon the baseline significantly (10% relative at 9%WER) on a small vocabulary task.

1 Introduction

The common statistical model employed in most of the modern state-of-art speech recognition systems is the Hidden Markov Model (HMM). HMMs can be trained in various ways depending on the modeling assumption. The commonly used decoding scheme called Maximum A-Posteriori (MAP) gives optimal performance under Sentence Error Rate criterion. But for WER performance, other decoding techniques like SMBR proves to be better suited. In SMBR based decoding the search space is reduced and extremely refined discriminative training methods can be used for performance gain.

During decoding HMM use only simple acoustic scores for discriminating between two competing hypothesis paths. Extra data-dependent information obtained from the various sufficient statistics of the model parameters can be incorporated into the decoding mechanism. In this paper we have used *Fisher Scores* [1] for extracting extra information from the model. Generally these are multi-dimensional vectors consisting of first order derivative of the likelihood with respect to various model parameters [11, 12]. SVMs can then be trained as classifiers in this score space. This paper describes the application of SVMs trained in this score space for SMBR based decoding.

2 Speech Recognition System

2.1 Formulation

Given a sequence of acoustic observations $\mathbf{O} = o_1, o_2, \cdots, o_m \quad o_i \in \mathcal{O}$ and possible word sequence$\mathbf{W} = w_1, w_2, \cdots, w_n \quad w_i \in \mathcal{W}$, the problem of a Automatic Speech Recognition (ASR) system is to find a word string $\hat{\mathbf{W}}$ satisfying

$$\hat{\mathbf{W}} = \text{arg max}_{\mathbf{W}} P(\mathbf{W}|\mathbf{O})$$

Using Bayes rule the above problem is equivalent to

$$\hat{\mathbf{W}} = \text{arg max}_{\mathbf{W}} P(\mathbf{W})P(\mathbf{O}|\mathbf{W})$$

Hence a ASR system consists of three main components: a language model $P(\mathbf{W})$, an acoustic model $P(\mathbf{O}|\mathbf{W})$ and a decoder to find the string having maximum probability. In this paper we describe experiments based on isolated alpha-numeric character recognition, hence the language modeling component is not required. The component $P(\mathbf{O}|\mathbf{W})$ is modeled by a HMMs. The output distribution of the HMM is modelled by a mixture of Gaussian distributions.These models are trained using various criteria like Maximum Likelihood Estimation (MLE), Maximum Mutual Information Estimation (MMIE).

The commonly used decoding strategy is the MAP decoding criterion, which given an utterance $\mathbf{O}$ produces a sentence hypothesis according to $\hat{\mathbf{W}} = \text{arg max}_{\mathbf{W} \in \mathcal{W}} P(\mathbf{W}|\mathbf{O})$. It gives optimal performance under Sentence Error Rate. But the standard evaluation metric for ASR systems is WER. Hence other decoding techniques optimized for WER criterion are preferred. Such schemes include the Minimum Bayes-Risk (MBR) decoder and its variants the SMBR .

2.2 Segmental Minimum Bayes-Risk Decoders

The MBR decoder tries to minimize the sentence hypothesis errors under a given loss function. Mathematically it finds the optimal hypothesis given by

$$\hat{W} = \text{arg min}_{W' \in \mathcal{W}} \sum_{W \in \mathcal{W}} l(W, W')P(W|O)$$

where $\mathcal{W}$ denotes the possible word string set from a N-Best list or lattices generated by the underlying HMM model and $l(W, W')$ is the loss function between word string W and W'. As a result the decoding becomes computationally expensive search problem. SMBR tries to tackle the problem by decomposing the original problem into a sequence of independent small search problems. Here we assume that the word sequence $W \in \mathcal{W}$ is segmented into N substrings consisting of zero or more words $W_1, W_2, \cdots, W_N$. This segments the associated lattice into N segment sets $\mathcal{W}_i, i = 1, 2, \cdots, N$. Then the MBR hypothesis $\hat{W}$ can be obtained by concatenating the individual hypothesis $\hat{W}_i, i = 1, 2, \cdots, N$ obtained for each lattice segment using

$$\hat{W}_i = \text{arg min}_{W' \in \mathcal{W}_i} \sum_{W \in \mathcal{W}_i} l(W, W')P_i(W|O)$$

There are various ways to segment the lattices. One particular approach starts with aligning the MAP hypothesis with the lattice. In region of low confidence of the MAP hypothesis other alternative paths from the lattice are retained for the new search space. But for high confidence region, the hypothesis is restricted to the MAP hypothesis. Hence the structure of the original lattice is retained only when we are not confident about the MAP hypothesis The new search space obtained by concatenating these small segments is called a *pinched lattice*. Typical examples of these search spaces are shown in fig 1.

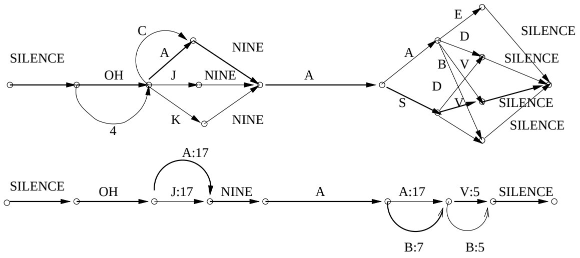

Figure 1: Lattice Segmentation: *Top:* First-pass lattice of likely sentence hypotheses with MAP path in bold. *Bottom:* Refined search space $\bar{\mathcal{W}}$ consisting of alternative paths to be discriminated

2.3 Refined discriminative training using SVMs

In a pinched lattice, those segments of the MAP hypothesis which are more error-prone are expanded using segments of the original lattice. Hence it opens up the possibility of training more refined models for those segments only. Previously separate acoustic models were trained specifically to discriminate only between the word sequences present in $\mathcal{W}_i$. Only training data corresponding to those specific lattice segments were used in building these acoustic models. The drawback of this approach lies in the fact that it uses only a scalar acoustic score to discriminate between the confusable alternatives. Instead we can use extra data dependent information from the generative model. In our work we have extracted extra information in form of fisher score. Then separate SVMs were trained for each confusable pair. Whenever similar confusion pairs were obtained in the test set, the corresponding SVM was used to select the most probable path.

3 Score-space formulation

This section gives a brief description of score-spaces and operators. Speech is a highly dynamic process. Hence the observation sequences are not of fixed-dimension. But to apply any static classifier like SVMs to speech problems, we need to map the observation sequence into fixed-dimension vectors. Score-space provides a mechanism of doing this.

Let $\mathbf{O} = (o_1, o_2, \cdots, o_t)$ be an observation sequence and let a set of possible generative probability models for that sequence be $\mathcal{P} = \{p_k(\mathbf{O}|\theta_{\mathbf{k}})\}$. This observation sequence can be transformed to a score-space using the following mapping:

$$\varphi^f_{\hat{F}}(\mathbf{O}) = \varphi_{\hat{F}} f(\{p_k(\mathbf{O}|\theta_{\mathbf{k}}\})$$

Here $f(.)$ is the score-argument and is function of the members of $\mathcal{P}$. $\varphi_{\hat{F}}$ maps from the scalar space of the score-argument into the score-space and hence called score-mapping. It is defined using a score-operator $\hat{F}$. The projection of the score-argument into the score space $\varphi^f_{\hat{F}}(\mathbf{O})$ is called the score. There are several candidates for score-argument and score-operator. Each of these combination give rise to a different score space.

The different dimensions of the score space are not orthogonal. Generally it should normalized or whitened to decorrelate the dimensions. The normalized score space is can be computed as below:

$$\varphi^N(\mathbf{O}) = \Sigma_{SC}^{-1/2}\varphi(\mathbf{O})$$

where

$$\Sigma_{SC} = \int (\varphi(\mathbf{O} - \mu_{SC})(\varphi(\mathbf{O} - \mu_{SC})^T p(\mathbf{O}|\theta) d\mathbf{O}$$

$$\mu_{SC} = \int \varphi(\mathbf{O})p(\mathbf{O}|\theta)d\mathbf{O}$$

are the covariance matrix and expectation in the score-space respectively.

3.1 Fisher score and kernel

The likelihood score-space given by a first-order derivative operator is called the fisher score and using the above notation can be represented as

$$\varphi^{lik}_{\nabla_\theta}(\mathbf{O}) = \nabla_\theta \ln p(\mathbf{O}|\theta)$$

Here θ represents the parameters of the HMM. Another choice of score-argument function is the log likelihood ratio.In that case the score is represented by

$$\varphi^{lik}_{\nabla_\theta}(\mathbf{O}) = \nabla_\theta \ln \frac{p_1(\mathbf{O}|\theta)}{p_2(\mathbf{O}|\theta)}$$

The normalized Fisher score space is given by

$$\phi = \Sigma^{-1/2}\varphi^{lik}_{\nabla_\theta}(\mathbf{O})$$

From Mercer's Theorem we know any kernel can be represented as a simple inner product between suitably chosen feature vectors

$$K(X_i, X_j) = \phi^T_{X_i}\phi_{X_j}$$

For any parametric class of models $P(X|\theta) \quad \theta \in \Theta$ the fisher information matrix I is given by

$$I = E_X\{U_X U_X^T\}$$

where

$$U_X = \nabla_\theta \log P(X|\theta)$$

is the Fisher score and the expectation is over $P(X|\theta)$. Hence a natural kernel in this mapping space can be obtained by taking the inner product of the feature vectors

$$K(X_i, X_j) \propto \phi^T_{X_i}\mathbf{I}\phi_{X_j} = U^T_{X_i}\mathbf{I}^{-1}U_{X_j}$$

This kernel is called the Fisher kernel.

4 SVMs in Automatic Speech Recognition: Challenges

Applying SVMs to Speech Recognition is not straight forward. One problem is that speech is a dynamic signal while SVMs are inherently static classifiers. We solve this problem by using Fisher kernels. Thus we map the variable length observation sequences into a fixed dimension space, the dimension being equal to the number of parameters in the generative model. Also, SVMs are inherently binary classifiers while speech is a multi-class problem. We get around this by using the SMBR decoding approach; we reduce the problem to a sequence of independent binary decision problems; then use SVMs for each of these simpler problems.

5 Proposed Method

The proposed method is as follows:

- Train HMMs and generate pinched lattices
- Go through the pinched lattices and choose 50 most confusable pairs

- Generate Fisher Score of those selected observation sequences and train separate SVMs for each pair
- Decode the test set
- For each confusion pair (W_i, W_j) in the test set,
  - Identify occurrences of the same in the training set
  - Using the trained SVM for this pair, choose hypothesis from (W_i, W_j).

6 Corpus and Baseline System Description

To test our approach, we present our results on the OGI-Alphadigit [3] corpus. This is a small vocabulary task of 37 words (26 letters and 11 numbers). There are around 46,000 utterances in the training set and around 3,000 in the test set. Each utterance is a sequence of 6 words. The error rate of discriminitively trained HMM-based systems on this corpus is around 10%. This ensures that are enough errors to allow for error analysis.

The HMM baseline system was built using the HTK toolkit [4]. The audio was encoded as 13 dimensional MFCC vectors and appended with first and second-order derivatives. Each word was modeled by left-to-right HMMs having around 20 states each. The models contained 12 mixtures per state, estimated by conventional HTK style training procedures. Starting from these models, three iterations of Maximum Mutual Information Estimation (MMIE) [5] [6] were performed to yield discrimanitively trained models. The Alphadigits task has no language model; thus a simple word loop was used for decoding and for generating lattices [7] on the training set for MMIE. The WER (9.07%) from these models were taken to be the baseline.

7 Experiments

We consider the 50 most frequently occurring confusion pairs in the training set. These were obtained by Period-1 risk-based lattice cutting [8]; we further simplify the problem by allowing only two competing paths in each confusion pair.

We first trained SVMs, using the giniSVM toolkit [9] [9], for the confusion pair `B<->V`. The alignments used were from the baseline system. There were around 5000 occurrences of `B<->V` in the training set and around 300 in the test set. To work with reasonable dimensions, we generated only the mean-space scores from the Baseline models. For the log-likelihood ratio score space that we used, this results in 20,000 dimension vectors. For the whitening of the scores, we used a diagonal approximation to the whitening matrix as calculated from the training data; this approximation was used to whiten both training and test scores. The training of the SVM, for the 20,000 dimension case, converged only for one value of the trade-off parameter (C=100). However, the performance of this SVM was dismal.

Previous research [11] indicates that the best generative models to get scores to train the best performing SVMs need not be from the best performing generative models themselves. The very high dimension of the scores probably has too many nuisance dimensions that the SVM cannot ignore. Thus, we then used 2 mixture Maximum Likelihood (ML) intermediate models to generate scores. Note there is now a mismatch between the models used to identify confusion pairs (12 mix MMIE) and those used to train SVMs (2 mix ML). The mean-scores now have a dimension of around 3000. Table 1 shows the error counts for the confusion pair as the trade-off parameter varies. We can see variation in performance with C; this indicates that some tuning of C will be required. We also found the `tanh` kernel produced a better Kernel feature map than the linear kernel; for C=1.0, the error count went down further to 86.

Table 1: *Error counts for* B<->V *confusion pair using a linear kernel*

SVM trade-off parameter (C)	Errors
Baseline	69
0.01	116
0.1	100
1.0	100
10.0	101
100.0	106

Table 2: *Word Error Rates on Full Systems.* r *indicates zeroth-derivative of the log-likelihood ratio scores,* a *indicates the transition probability scores and* m *the mean scores. The notation C=x->y indicates that for those SVMs which didn't converge with C=x, a SVM with C=y was used*

System	Error-Rates
Baseline	9.07
SVM m C=1.0->10.0	9.63
SVM r_a_m C=1.0 ->10.0	9.65
SVM r_a_m C=1.0 -> 3.0	9.44

We then added the zeroth-order derivative of the log-ratio scores and the transition probability derivatives of the HMMs to the score-space. For a dynamic task like speech recognition, one would expect the transition probabilities to help performance. Also, these additions resulted in a very small increase in the score dimension.

Finally, we trained SVMs on all 50 confusion pairs in these scores spaces. The outputs of these SVMs were then inserted in place of the confusion sets to yield full hypotheses for each utterance. This resulted in a full system whose performance we could measure against the baseline system.

Table 2 lists the Word Error Rates for various SVM systems. Thus the best SVM system that was trained is about 0.4% absolute WER worse. However, on error analysis [Figure 7], we find that SVMS do not perform uniformly worse. The SVMs outperform the baseline MMIE models on confusion pairs that had lesser amounts of training data (upto 5,000 training points). Beyond that, the SVM training most probably needs to tuned. To test this, we trained the C=1.0 -> 3.0 system (the final line of 2). The idea was that C has to be a multiple of the amount of training data upto which its performance was satisfactory. This indeed helped performance on confusion pairs with more training data; but was insufficient to beat the baseline.

On further analysis, we found that the 9.44% WER system and the 9.07% WER system were about 6.0% absolute apart. This situation is ideal for voting schemes. However, we noticed that the probabilities that the SVM was assigning to the two paths were not reliable confidence scores, *i.e.*, the errors that the SVM makes and the probabilities that it assigns were uncorrelated. However, it is well known that ASR errors can be reliably identified with HMM systems [13]. Thus we used a simple algorithm, based on posteriors, to filter out unreliable decisions of the HMMs and used SVM outputs in place of them. If the

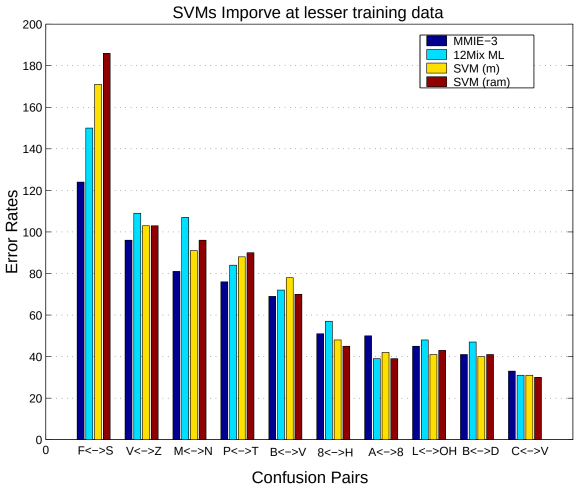

Figure 2: *Bar plot of error counts on confusion pairs of 4 different systems. The distance along the x-axis indicates decrease in the amount of training data; e.g.,* F<->S *had 15,000 training points, while* B<->V *had around 5,000. The SVMs outperform the baseline MMIE models on confusion pairs with lesser training data. For other confusion pairs, training needs tuning.*

Table 3: *Error-Rates for different thresholds used for accepting the HMM output. A lower threshold would imply lesser confidence on the HMM system and more on the SVM system*

Threshold(t) (best path must have probability>t)	Error-Rates
0.0	9.07
0.9	8.87
.	.
.	.
$1-10^{-12}$	**8.25**
$1-10^{-13}$	8.30

posterior of a word's hypothesis was less than a chosen threshold, then we simply replace it by the SVM hypothesis. Using this combination approach, we can see from table 3, that we can reduce the error rate to 8.25%.

8 Conclusions and Future Work

We have presented a framework for training SVMs in the SMBR framework for Speech Recognition. The SVMs on combination with the baseline system shows significant improvement over the baseline performance on a small vocabulary task. It would be interesting to see if such improvements can still be obtained without recourse to the best HMMs available; *e.g.*, the confusion pairs on the test set were identified using 12 mixture MMIE models. It has been shown that scores generated from MMIE models yield SVMs with better performance then those from ML models [14]. The SVM trade-off parameter has to be tuned separately for each giniSVM; the amount of training data available seems to be a good criterion. The SVM output probabilities can also be used to perform re-scoring of the lattices.

Acknowledgments: The authors thank Shantanu Chakrabartty for helpful discussions, ideas and for the giniSVM toolkit. The MMIE models were trained by Vlasios Doumpiotis; the ML models were trained by Terri Kamm.

References

[1] T. Jaakkola and D. Haussler. “Exploiting generative models in discriminative classifiers.” In Advances in Neural Information Processing Systems 11, 1998.

[2] T. Jaakkola, M. Diekhans, and D. Haussler.”A discriminative framework for detecting remote protein homologies.” Journal of Computational Biology, 7(1,2):95–114, 2000.

[3] M. Noel, “Alphadigits,” CSLU, OGI, 1997, [Online]. Available: `http://www.cse.ogi.edu/CSLU/corpora/alphadigit.`

[4] S. Young et. al., *The HTK Book, Version 3.0*, July 2000.

[5] Y. Normandin, “Maximum Mutual Information Estimation of Hidden Markov Models,” *Automatic Speech and Speaker Recognition: Advanced Topics*, Chin-hui Lee, Frank K. Soong and Kuldip K. Paliwal, Eds. Kluwer, 1996.

[6] P. C. woodland and D. Povey, “Large Scale Discriminative Training for Speech Recognition” *Proc. ITW ASR*, ISCA, 2000.

[7] M. Mohri, F. Pereira and M. Riley, *AT&T General-purpose Finite-State Machine Software Tools*, 2001, [Online]. Available: http://www.reserarch.att.com/sw/tools/fsm/.

[8] S. Kumar and W. Byrne, “Risk Based Lattice Cutting for Segmental Minimum Bayes-Risk Decoding ” *ICSLP*, Denver, Colorado, USA, 2002.

[9] S. Chakrabartty and G. Cauwenberghs, “Forward-Decoding Kernel-Based Phone Sequence Recognition,” , *Adv. Neural Information Processing Systems (NIPS’2002)* , Cambridge: MIT Press, vol. 15, 2003.

[10] S. Chakrabartty, “The giniSVM toolkit”, Technical Report No. 48, CLSP, JHU, 2003.

[11] N. D. Smith and M. J. F. Gales. *Using SVMs to classify Variable Length Speech Patterns*, Technical Report CUED/F-INFENG/TR412, Cambridge University Eng. Dept., April 2002.

[12] N. D. Smith, M. J. F. Gales and M. Niranjan. *Data-dependent kernel in SVM Classification of speech patterns*, Technical Report CUED/F-INFENG/TR412, Cambridge University Eng. Dept., April 2002.

[13] J. Fiscus, “A Post-processing System to Yield Reduced Word Error Rates: Recognizer Output Voting Error Reduction (ROVER)”, *IEEE Wkshp. Spch. Recog & Und*, 1997.

[14] N. D. Smith and M. J. F. Gales.“Using SVMs and discrimanitive models for speech recognition,” ICASSP, 2002.