HYBRID SVM/HMM ARCHITECTURES FOR SPEECH RECOGNITION

A. Ganapathiraju, J. Hamaker, and J. Picone

Institute for Signal and Information Processing
Department for Electrical and Computer Engineering
Mississippi State University, Mississippi State, MS 39762
{ganapath, hamaker, picone}@isip.msstate.edu

ABSTRACT

In this paper, we describe the use of a powerful machine learning scheme, Support Vector Machines (SVM), within the framework of hidden Markov model (HMM) based speech recognition. The hybrid SVM/HMM system has been developed based on our public domain toolkit. The hybrid system has been evaluated on the OGI Alphadigits corpus and performs at 11.6% WER, as compared to 12.7% with a triphone mixture-Gaussian HMM system, while using only a fifth of the training data used by triphone system. Several important issues that arise out of the nature of SVM classifiers have been addressed. We are in the process of migrating this technology to large vocabulary recognition tasks like SWITCHBOARD.

1. INTRODUCTION

Speech recognition can be viewed as a pattern recognition problem where we desire each unique sound to be distinguishable from all other sounds. Traditionally statistical models, such as Gaussian mixture models, have been used to "represent" the various modalities for a given speech sound. The parameters of the Gaussians are estimated using a Maximum Likelihood (ML) criterion [1]. The ML formulation for the representation of the acoustic space does not necessarily translate to better recognition performance since most of the optimization effort is spent in learning the intricacies of the training distributions.

Extensions of the HMM learning paradigm involving discriminative training techniques such as Maximum Mutual Information (MMI) and Minimum Classification Error (MCE) attempt to estimate parameters using both positive and negative examples [2]. Though they give consistent improvements in recognition performance, these techniques are computationally very expensive and are, thus, limited to small vocabulary tasks.

2. SUPPORT VECTOR MACHINES

Classifiers are typically optimized based on some form of risk minimization. Empirical risk minimization is one of the most commonly used technique where the goal is to find a parameter setting that minimizes the risk:

$$R_{emp}(\alpha) = \frac{1}{2l}\sum_{i=1}^{l}\left|y_i - f(x_i, \alpha)\right| , \quad (1)$$

where α is the set of adjustable parameters and y_i, x_i are the expected output and given input, respectively. However, minimizing R_{emp} does not necessarily imply the best classifier possible. For example, Figure 1 shows a two-class problem and the corresponding decision regions in the form of hyperplanes. All the hyperplanes $C0$, $C1$ and $C2$ achieve perfect classification and, hence, zero empirical risk. However, $C0$ is the optimal hyperplane because it maximizes the distance between the margins $H1$ and $H2$, thereby offering better generalization [4]. This form of learning is an example of Structural Risk Minimization (SRM) where the aim is to learn a classifier that minimizes a bound on the expected risk, rather than the empirical risk [4]. SVM learning is based on this SRM principle.

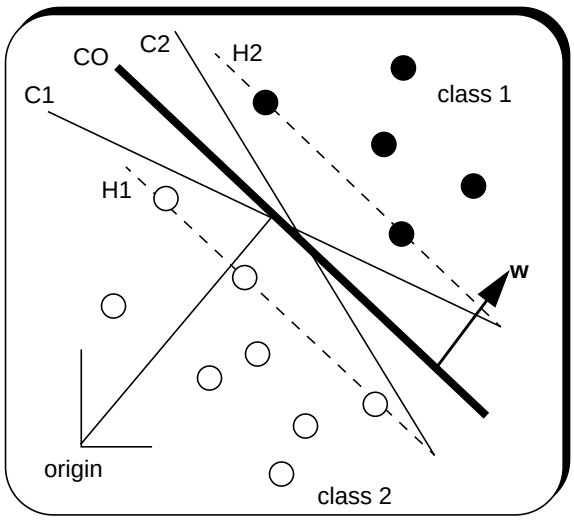

Figure 1: 2-class hyperplane classifier example

The power of SVMs lies in their ability to transform data to a high dimensional space where the data can be separated using a linear hyperplane. The optimization process for SVM learning therefore begins with the definition of a functional that needs to be optimized in terms of the parameters of a hyperplane. The functional is defined such that it guarantees good classification (if not perfect classification) on the training data and also maximizes the margin (e.g. the distance between H1 and H2 in Figure 1). The points that lie on the hyperplane satisfy,

$$\boldsymbol{w} \cdot \boldsymbol{x} + b = 0 \tag{2}$$

where $\boldsymbol{w}$ is the normal to the hyperplane and b is the bias of the hyperplane from the origin. Let the N training examples be represented as tuples $\{\boldsymbol{x}_i, y_i\}, i = 1, \ldots, N$ where $y = \pm 1$ are the class labels. They satisfy the following constraints,

$$y_i(\boldsymbol{x}_i \cdot \boldsymbol{w} + b) - 1 \geq 0 \qquad \forall i \tag{3}$$

The distance between the margins can be shown to be $2/\|\boldsymbol{w}\|$ [4]. The goal of the optimization process should be to maximize the margin. Posing this as a quadratic optimization problem has several advantages and the functional can be compactly written as,

$$L_P = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{N} \alpha_i y_i(\boldsymbol{x}_i \cdot \boldsymbol{w} + b) + \sum_{i=1}^{N} \alpha_i \tag{4}$$

where the α_i 's are Lagrange multipliers.

As observed previously, only a few training examples have an impact on the functional and the optimal decision surface. This translates to the fact that, at the end of the optimization process, only a small percent of the training examples have non-zero multipliers. These examples are called Support Vectors. Note that we have assumed that the data are perfectly separable. This is not the case in most real data. This problem is handled by introducing *slack* variables into Equation 3:

$$y_i(\boldsymbol{x}_i \cdot \boldsymbol{w} + b) - 1 + \xi_i \geq 0 \qquad \forall i \ . \tag{5}$$

Note that the number of training errors can be characterized by $\sum_i \xi_i$.

We now have to address the need for learning classifiers that define non-linear decision regions. Notice that the linearity in the SVM design is manifested in the dot products. Suppose we transform the data into a higher dimension space where the data is linearly separable. The theory we have developed thus far holds in this case. So one could envision replacing all x_i 's with X_i 's in the above formulation where the X 's are in the high dimensional space. The theory of Kernel functions is used to avoid dealing directly with the high dimensional space and the excessive computations that result from such transformations [4,5].

Some of the commonly used kernels include,

$$K(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x} \cdot \boldsymbol{y} + 1)^d \qquad \text{(polynomial)} \tag{6}$$

$$K(\boldsymbol{x}, \boldsymbol{y}) = \exp\{-\Upsilon|\boldsymbol{x} - \boldsymbol{y}|^2\} \qquad \text{(RBF).} \tag{7}$$

The final classifier takes the form,

$$f(\boldsymbol{x}) = \sum_{i=1}^{L} \alpha_i y_i K(x, x_i) + b \tag{8}$$

where L is the number of support vectors. The class to which a sample belongs is decided by the sign of f.

3. HYBRID ASR SYSTEM

One significant drawback in SVMs is that, they are inherently static classifiers — they do not implicitly model temporal evolution of data. HMMs have the advantage of being able to handle dynamic data with certain assumptions about stationarity and independence [3]. Taking advantage of the relative strengths of these two classification paradigms we have developed a hybrid SVM/HMM system using our public domain speech recognition toolkit [9]. The toolkit includes a cepstral front-end, a Viterbi decoder capable of generating and rescoring word-graphs and a Baum-Welch training module. This system provided all components for the HMM portion of the hybrid system architecture. For estimating SVMs we used a publicly available toolkit, SVMLight [6].

An important issue that had to be addressed in this hybrid system is the fact that SVMs output a distance measure, while the Viterbi decoding algorithm typically uses likelihoods or posterior probabilities. We therefore estimate a warping function that maps SVM distances to posterior probabilities. There are several ways one could do this. One way would be to

estimate the class-conditional densities based on the histogram of the SVM distances for positive and negative examples. A posterior can then be estimated using the Bayes rule. A simpler approach to estimating the posterior is to assume that posterior takes the form of a sigmoid, and directly estimate the sigmoid [10].

$$p(y|f) = \frac{1}{1 + \exp(Af + B)} \quad (9)$$

In order to avoid severe bias in the distances for the training data, the free parameters, A and B are estimated on a cross-validation set. Once we have the posteriors, we replace the Gaussians in the HMM system with the SVM classifiers.

4. EXPERIMENTAL SETUP

Figure 3 shows the hybrid architecture used for the recognition experiments. Given the SVM classifiers and an HMM system one would first attempt to train the classifiers on frame level data and use them as the classifiers in each state of the HMM. Since each classifier is trained as a one-vs-all classifier, the amount of training data is significant. To avoid burdening the quadratic optimizer, we chose to use segment-level data for our initial experiments. Using segment-level data also means that the HMMs we use are simple one state HMMs, though one could train classifiers for multi-state HMMs as well [7].

The HMM system is used to generate alignments at the phone level and each phone instance is treated as one segment. Since each segment could span a variable duration, we need to use some form of sampling to arrive at a fixed length vector for classification. Several methods have been attempted in this regard based on fixed and variable sampling techniques [11, 12]. One approach is to divide the segment into three regions in a set ratio and construct a composite vector from the mean vectors of the three regions. In our experiments we chose to follow empirical evidence and divide the frames in the segment into three regions in a 3-4-3 proportion. Figure 2 shows an example for constructing a composite vector for a phone segment. SVM classifiers in our hybrid system operate on such composite vectors.

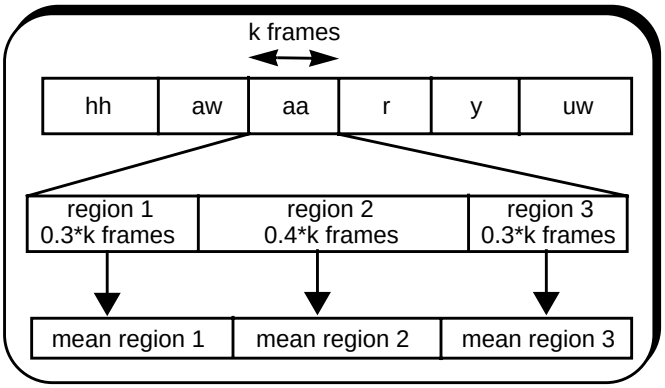

Figure 2: Example of a composite vector construction using a 3-4-3 proportion

At decode time, we get the segmentation information using a baseline HMM system — a cross-word triphone system with 8 Gaussian mixtures per state. Composite vectors are generated for each of the segments and posterior probabilities are hypothesized that are used to find the best word sequence using the Viterbi decoder. A better methodology to follow would be to generate segmentations for the hypothesis in an N-best list and reorder the list using the likelihoods generated by the SVMs [7].

5. RESULTS

The hybrid architecture has been benchmarked on the OGI alphadigit corpus that has a vocabulary of 36 words [8]. We used 29 phones to represent the pronunciations of the words, and therefore trained 29 SVM classifiers. The baseline HMM system was trained on 39-dimensional feature vectors comprised of 12 cepstral coefficients, energy, delta and acceleration coefficients. The training set had 50,000 sentences averaging 6 words a sentence. The SVM classifiers were trained using the composite feature vectors generated for only 9000 training sentences.

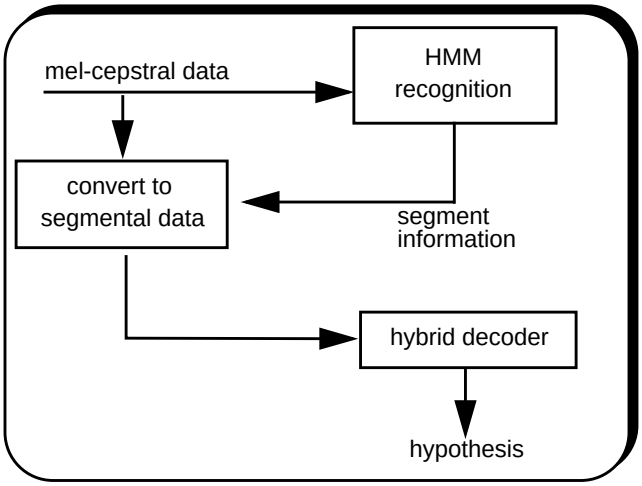

Figure 3: Hybrid system architecture

The test set was an open-loop speaker independent set with 1000 sentences. The composite vectors are also normalized to the range (-1,1) to avoid convergence problems with the quadratic optimizer.

Table 1 shows the performance of the hybrid system in its various configurations. The system performs better than the baseline cross-word triphone HMM system with 8 Gaussian mixture components per state which gives 12.7% WER on this dataset. The best performance is achieved when the ratio of the segments in the composite feature vector is 3-4-3 which is in agreement with our notion that most of the information in a 3-state HMM is provided by the central state. From the results we also note that the RBF kernel is typically better at classification than the polynomial kernels owing to its ability to model decision regions where one class encloses the other. In terms of resource usage the SVM systems have about 13000 unique support vectors. This is an order of magnitude less than the number of free parameters in the cross-word triphone HMM system.

6. SUMMARY

In this work we have developed a paradigm for integrating SVMs into an HMM framework. The goal of this work was to augment HMMs with powerful classifiers, SVMs, that are trained discriminatively. Results on the OGI Alphadigits data show that the hybrid system gives a significant improvement (10% relative) over the baseline HMM system while using only a fifth of the training data. We expect that extending this approach to process N-best lists will give us further gains, especially in large vocabulary tasks like SWITCHBOARD. We are in the process of developing a method to convert variable length feature vectors into a fixed length vector based on the sufficient statistics generated using the Baum-Welch algorithm.

segment ratio	polynomial kernel		RBF kernel
	order-4	order-6	
1-1-1	13.2	13.6	12.8
3-4-3	12.1	13.4	**11.6**
2-4-2	13.1	13.5	12.5

Table 1: Performance of the hybrid system on OGI alphadigits (numbers show percent word error rate)

7. REFERENCES

[1] A.P. Dempster, et. al., "Maximum Likelihood Estimation from Incomplete Data," *J. of the Royal Statistical Society*, vol. 39, no. 1, pp. 1-38, 1977.

[2] V. Valtchev, *Discriminative Methods in HMM-based Speech Recognition*, Ph. D. Thesis, University of Cambridge, UK, 1995.

[3] J. Picone, "Continuous Speech Recognition Using Hidden Markov Models," *IEEE ASSP Magazine*, vol. 7, no. 3, pp. 26-41, July 1990.

[4] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag, New York, NY, USA,1995.

[5] B. Schölkopf, et. al., *Advances in Kernel Methods: Support Vector Machines*, MIT Press, Cambridge, MA, USA, December 1998.

[6] T. Joachims, SVMLight: Support Vector Machine, *http://www-ai.informatik.uni-dortmund.de/FORSCHUNG/VERFAHREN/SVM_LIGHT/svm_light.eng.html*, University of Dortmund, November 1999.

[7] A. Ganapathiraju, et. al., "Support Vector Machines for Speech Recognition," *Proc. of the ICSLP*, pp. 2923-2926, Sydney, Australia, November 1998.

[8] R. Cole et al, "Alphadigit Corpus," *http://www.cse.ogi.edu/CSLU/corpora/alphadigit*, Oregon Graduate Institute, 1997.

[9] N. Deshmukh, et. al., "Hierarchical Search for Large Vocabulary Conversational Speech Recognition," *IEEE Signal Processing Magazine*, vol. 1, no. 5, pp. 84-107, September 1999.

[10] J. Platt, Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods, In *Advances in Large Margin Classifiers*, MIT Press, Cambridge, MA, USA, 2000.

[11] S. Austin, et. al., "Speech Recognition Using Segmental Neural Nets," *Proc. of the ICASSP*, pp. 625-628, 1992.

[12] M. Ostendorf and S. Roukos, "A Stochastic Model for Phoneme-Based Continuous Speech Recognition," *IEEE Trans. on ASSP*, vol. 37, pp. 1857-1867, December 1989.