A HYBRID GMM/SVM APPROACH TO SPEAKER IDENTIFICATION

Shai Fine *Jiří Navrátil* *Ramesh A. Gopinath*
IBM T.J. Watson Research Center, Yorktown Heights, NY 10598, USA
e-mail: {fshai,jiri,rameshg}@US.ibm.com

ABSTRACT

This paper proposes a classification scheme that incorporates statistical models and support vector machines. A hybrid system which appropriately combines the advantages of both the generative and discriminant model paradigms is described and experimentally evaluated on a text-independent speaker recognition task in matched and mismatched training and test conditions. Our results prove that the combination is beneficial in terms of performance and practical in terms of computation. We report relative improvements of up to 25% reduction in identification error rate compared to the baseline statistical model.

1. INTRODUCTION

Text-independent speaker identification, as a classical, purely acoustic recognition task, has been subject of intensive research efforts for the past several decades. Major challenges - namely the high variability of channel properties and the question of appropriate model structures to capture the characteristics of an individual voice - have been addressed by a wide range of feature extraction and modeling techniques. These cover pattern matching approaches, such as dynamic time warping, statistical modeling, e.g. hidden Markov models and Gaussian mixture models, and connectionist methods, e.g. multi-layer perceptrons.

Gaussian mixture models (GMM) represent the state-of-the-art technique in text-independent speaker recognition [3] and many other tasks including detection and segmentation [12, 15].

Particularly in the telephony environment, given relatively small amounts of training data (e.g. 30 sec), well-designed GMM systems show a good robustness to channel variations and seem to best achieve independency of text, topic and language.

In recent years a new classification methodology based on Support Vector Machines (SVM) [1, 14] has found an increased interest in the speech community. Favourable properties of the SVM such as their inherent class-discriminative model structure and the use of nonlinear-kernel methods represent an attractive way of enhancing the standard methods, mostly based on generative models (GMM, HMM), by complementary information and classification "power."

This paper describes a classification scheme that incorporates both the SVM and the statistical models in a way that the robustness advantage of the generative statistical models favourably combines with the discriminative power of the SVM. We apply this scheme to the task of text-independent speaker recognition and show that the above mentioned combination is both beneficial in terms of performance and practicable in terms of computation.

The paper is organized as follows: Section 2. describes the statistical (baseline) model, followed by Sec. 3. dedicated to the construction of a hybrid GMM/SVM classification system. The experimental task, data, and results are presented in Sec. 4. with subsequent discussion in Sec. 5.

2. THE BASELINE SYSTEM

In the first stage of our system, statistical models, namely Gaussian mixture models (GMM), serve as a parametric basis for the support vector machines and also as a baseline system for performance evaluation.

Due to the favourable properties mentioned above, GMMs are a good choice for our purpose: (1) they provide a suitable parametric structure for the SVM kernel (see Sec. 3.), and (2) they supply likelihood scores for reducing the size of the hypothesis set in the first stage to a small number of candidate classes, without great compromise in the accuracy.

For the class (speaker) modeling a hierarchical speaker modeling system [2] has been applied. Due to the exploratory nature of the experiments in this paper, we simplified this system by considering only the level without phonetic knowledge (i.e. global level). Thus, each speaker model θ with K mixture components is determined by the set $\theta_i = \{c_1^K, \mu_1^{\mathbf{K}}, \mathbf{\Sigma_1^K}, P_1^K\}_i$ with $\mu_1^{\mathbf{K}}$, $\mathbf{\Sigma_1^K}$, and P_1^K being the mean vectors, the covariance matrices and the priors for the K components respectively. Based on a sequence of training vectors belonging to a particular speaker, the Gaussian parameters are trained via Bayesian adaptation from a speaker independent GMM.

The likelihood of an observation $\mathbf{x}_t$ based on a speaker model θ is then calculated as follows:

$$p(\mathbf{x}_t|\theta) = \sum_{k=1}^{K} \frac{P(k|\mathbf{x}_t)}{(2\pi)^{n/2}|diag(\Sigma_k)|^{1/2}} \cdot$$
$$\cdot \exp\{-\frac{1}{2}(\mathbf{x}_t - \mathbf{m}_k)^t diag(\Sigma_k)^{-1}(\mathbf{x}_t - \mathbf{m}_k)\} \ . \quad (1)$$

3. THE CLASSIFICATION SYSTEM

The proposed classification system is formed by an ensemble of SVM binary classifiers which emerge from the GMMs of the baseline system using the Fisher kernel method. To better understand the system, we first review the main features of its principal ingredients and then describe the details of the classification process.

3.1. Support Vector Machines

In a nutshell, Support Vector Machine (SVM) is a classification learning methodology that incorporates two key ideas:

- Out of all possible separating hyperplanes (linear classifiers) - the *Optimal Hyperplane* [14] is the one with the maximal margin, with respect to a labeled training set, i.e.

$$f^* = \arg\max_f \min_i y_i f(x_i) \quad (2)$$

where $f(x) = (x \cdot w) + b$ for $x, w \in \mathbb{R}^N$ and $b \in \mathbb{R}$, $y_i \in \{-1, 1\}$ are the labels corresponding to the training set $\{x_i\}$, and $sign(f(x))$ is the classification rule. Intuitively, the choice of a classifier positioned in the "middle" between the two classes seems reasonable, and one can also expect to achieve robustness with respect to both the sample space as well as the hypothesis (function) space: a slight perturbation in either one of them should not affect the resulting classification. Without imposing any constraints, the optimization problem (2) is ill-posed. To get around it, a regularization term, the norm of w is introduced. This term serves as a complexity/capacity control mechanism (much like in the MDL methodology) that realizes Vapink's *Structual Risk Minimization* principle [13, 14].

- If an algorithm can be described merely by dot-product operations - then one can in principle avoid the need to explicitly represent the acting vectors. This calls for a separation between the input space, in which the input vectors reside, and the feature space, in which the vectors act. There may be such a separation if one applies some sort of transformation (linear or nonlinear) placing the input vectors in another space, hopefully more suitable for classification. By the assumption that the learning algorithm acts purely upon dot-product values it follows that, given an efficient method of computing those values - the algorithm's computational complexity will depend on the dimension of the input space and the training set size, rather then the dimension of the feature space. Hence, the "curse of dimensionality" may be alleviated.

Example 1 *The polynomial kernel, $(x \cdot y)^d$ corresponds to a map into the space spanned by all products of exactly d dimensions $\mathbb{R}^N$. Adding a shift parameter $c > 0$, i.e. $(x \cdot y + c)^d$, accounts for all products up to d. Using these kernel operations amounts to performing dot-product operations in an $O(N^d)$ space, while the cost of the actual computation scales with N.*

Another important ingredient of the SVM methodology is the *Soft Margin* which is introduced in order to enable learning even if the sample set is in fact linearly inseparable: the existence of *outliers,* namely mislabeled samples, is tolerated by incorporating positive slack variables ξ_i in the SVM optimization problem and assigning an extra cost for the errors, which scales with $\sum_i \xi_i^m$. With the choice $m = 1$ the SVM (primal) optimization problem can be stated as follows:

$$\begin{array}{ll} \min_{w,b,\xi} & \|w\|^2 + C\sum_i \xi_i \\ \text{s.t.} & y_i\left(\langle w \cdot x_i\rangle + b\right) \geq 1 - \xi_i \\ & \xi_i \geq 0 \end{array} \qquad (3)$$

(3) has a global, tractable solution, and is described solely using the Lagrange multipliers and dot-product values in the feature space, i.e. by kernel operations.

3.1.1. The Fisher kernel

Finding an appropriate kernel function for a particular application can be difficult and remains largely an unresolved issue. One of the recent innovations in the field of kernel engineering has been made by Jaakkula and Haussler [7] who formed a link between generative and discriminative models: Generally speaking, generative models (such as GMM, HMM or graphical models) will focus on providing an efficient description of the data while discriminative models will strive for a better description of a decision boundary between the various classes. The fact that usually discriminative methods outperform generative models at classification tasks should also be attributed to the fact that the training set (for discriminative methods) is labeled, which obviously provides significant amount of extra information. Jaakkula and Haussler suggested to encode the descriptive power of generative models in the design of a mapping function that will map the input data to a (fixed dimension) feature space in which Denote $p(x|\theta)$ a generative model, where θ are its parameters, the mapping function is an analogous quantity to the model's sufficient statistics, known as the *Fisher score*:

$$U_\theta(x) = \nabla_\theta \log\left(p(x|\theta)\right) \qquad (4)$$

Each component of U_x is a derivative of the log-likelihood score for the input vector x with respect to a particular parameter. In the GMM case, the feature vector components are derivatives with respect to the priors, the mean vectors and the covariance matrices (sec. 2.). The magnitude of the components specify the extent to which each parameter contributes to generating the input vector. The natural kernel for this mapping is the inner product between these feature vectors, scaled by a positive definite matrix, M

$$k_M^{nat}\left(x, x'\right) = U_\theta(x)\, M^{-1} U_\theta\left(x'\right) \qquad (5)$$

and the *Fisher kernel* is obtained by choosing $M = I = E_p\left(U_\theta(x)\, U_\theta(x)^T\right)$ i.e. *Fisher information* matrix. Another reasonable choice is to set $M = \mathbf{1}$, if I is too difficult to compute. In our experiment we have used a normalization procedure to scale each component of the feature vector to the $[-1, 1]$ interval.

The Fisher kernel defined above provides a "natural" means for comparing examples induced by the generative model. It was also shown [7] that subject to some mild assumptions, a kernel classifier employing the Fisher kernel would be at least as powerful as the original generative model, and in most cases will actually improve the discriminative power of the generative model. Recently, Oliver et al. [8] suggested another justification for the utility of Fisher kernels by providing a regularization-theoretic analysis of this approach which extend the set of kernels to a class of natural kernels. Our applied scaling procedure may thus be considered as a construction of a kernel in that extended class.

3.1.2. Handling massive data sets

In large scale problems (which are so common in real world applications such as speech, document classification, OCR, etc.) the optimization problem becomes impracticable. Several approaches to handle this problem have been suggested in the past few years which basically trade storage requirements for an increase in the overall computational complexity. In the current study we took an additional step and traded training convergence with an increased number of classifiers we are actually using. Constructing an *Ensemble of Classifiers* is applied in many popular learning techniques, such as *Boosting, Bagging* and *Error-Correcting Output Coding (ECOC)* (see sec. 3.2.). The strength of an ensemble stems from the observation that non-correlated errors made by individual classifiers can be removed simply by (un/weighted) voting[1]. We use an all-pair ECOC technique to shift from binary to multiclass problems allowing for a very crude convergence at individual SVM problems,

[1]For an ensemble of L classifiers, if the error rate of every individual classifier is less than or equal to $P < 0.5$, then the probability that the outcome of a majority vote will not be correct is upper bounded by P

knowing that the voting scheme implicit in the ECOC will compensate for that. By using the SMO algorithm [9] to solve the optimization problem, and by imposing very loose convergence criteria we obtain a solution of about $75-80\%$ of the optimum (measured by the value of the objective function). This, however, brings a dramatic decrease in computational complexity - from an order of several hours to a few minutes in convergence time.

3.1.3. Fitting Sigmoid

To enable post processing, it is necessary to calibrate the output of the SVM classifier with values (scores) from other parts of the system. While there may be numerous ways to do so, it seems reasonable to assign probabilistic reasoning to the calibrated classification scores. In [10] Platt suggests to train an additional sigmoid function to map the SVM outputs to posterior probabilities. The resulted SVM+Sigmoid classifier can be regarded as a "soft" linear classifier acting in the feature space and smoothing the original "hard" SVM decisions according to the following probability law:

$$P(y=1|f^*) = \frac{1}{1+\exp(Af^*+B)} \tag{6}$$

where f^* is the maximal margin classifier and A, B are the sigmoid parameters (found by maximizing the likelihood of f^* scores on the training data). Apart from providing a useful means for post-processing, the engagement of the sigmoid also seems to add certain amount of robustness to noise (as evidenced in our experiments).

3.2. Handling Multiclass Classification via Error Correcting Output Codes

While the resulting SVM classifiers (with or without the additional sigmoid fitting) are binary classifiers, the speaker recognition task is essentially multi-class. Diettrich and Baikiri [4] presented a general framework, based on error-correcting codes, to obtain a multi-class decision using binary classifiers: The original set of classes are partitioned into complementary (two) subsets. Each such partition defines a binary problem which is used to train a binary classifier. By repeating this process L times (each time using a different partition), we obtain an ensemble of L classifiers. The partitions are defined by (usually pre-determined) error-correcting code matrix. This matrix combinies the (binary) classifications to yield a multi-class decision: Each class is encoded as an L-bit codeword, such that the l-th bit is predicted by the l-th classifier. When the L classifiers are applied to a testing point, their predictions from an L-bit output word and based on the code matrix, we choose the class whose codeword is the closest (in Hamming distance) to the L-bit output word.

Methods for selecting the code matrix ranges from the simplest "one-against-all" approach (in which each bit corresponds to a prediction whether the given point belongs to a particular class) to picking codes which are specifically designed to have strong error-correcting properties. In our experiments we followed the *All-Pairs* technique of Hastie and Tibshirani [6] who suggested to compare all pairs of classes (i.e. ignore the data from other classes at training) and then combine the pairwise decisions to form the multi-class decision. In this case, the code matrix alphabet is extended to include an *NC* symbol which states that this bit should be ignored when calculating the distance between the output and the code word.

3.3. Putting It All Together: Building a Hybrid Classification System

The basic building block of the classification system is a binary classifier constructed by a Fisher kernel SVM classifier with a fitted sigmoid: Based on the GMMs of the baseline system, it is possible to associate a unique Fisher mapping (4) to each speaker[2] and then construct a kernel matrix (5) that will be plugged into the SVM optimization problem. For each resulting linear classifier we trained a sigmoid to map its score values to posterior probabilities. The final binary classification score is given by Eq. (6). Such binary classifiers are trained for all possible pairs of speakers and their scores combined using All-Pairs ECOC scheme. The final classification rule is the maximum decision over the speakers scores.

Clearly this classification process is very much involved and thus should be used to resolve cases which are too "hard" for the baseline system. We therefore constructed the hybrid system as followes:

1. The baseline system produces an N-best list, based on the GMMs scores.
2. The classification system considers *only* the N-best speakers and selects the one with the maximal score.

This two step procedure not only significantly reduces the work load, but also increases the accuracy of the overall system, since the baseline system serves as a filter passing only the relevant information for classification. However, if the baseline system fails to do so, a classification error occurs.

4. EXPERIMENTS

4.1. Database

The publicly available Lincoln Lab Handset Database LLHDB [11] was used to train and test both system parts in text-independent mode. The database contains telephone-bandwidth speech from 52 (male and female) speakers recorded over various types of microphones and thus allows for targeted evaluations in matched and mismatched acoustic conditions. In particular, we used data from four types of carbon-button microphones denoted CB1 through CB4. Each speaker recorded two long (30 sec) and ten short sentences (scripts from the TIMIT database) through each of the transducers. In all our experimental configurations, one long sentence of each speaker, namely the "rainbow" text, served for system training and the short utterances were used for testing, giving a grand total of about 2000 tests across the four microphone conditions.

4.2. The Baseline System

As front-end features for the baseline GMM system, 19-dimensional MFCC and their first derivatives were used, forming a 38 dimensional feature vector calculated every 10 ms with subsequent cepstral mean subtraction.

Using the rainbow passages, each GMM consisting of approximately 30 Gaussians was adapted from a speaker-independent (SI) model using the MAP criterion [5]. The SI model was created on a population of few hundreds speakers taken from an internal telephone-quality speech database.

Two separate systems were built: the first using training data from microphone CB1, the second on the CB3. The latter was chosen for comparison, based on the fact that the first system (CB1) performed worst on tests from this particular type CB3. Tests were carried out using all four types CB1 through CB4 on these two systems, thus having results for one matched and three mismatched condition rounds, for each of the two systems.

[2]Note that for a binary problem, this transformation implies asymmetric role for the positive and negative classes (since it depends on the generative model of only one of them).

System	Test Condition			
	CB1	C2	CB3	CB4
Baseline GMM CB1	6.9	11.0	54.9	35.4
Hybrid GMM/SVM CB1	5.1	9.8	54.9	35.4
Rel.red.% CB1	25.7	10.7	0.0	0.0
Baseline GMM CB3	53.0	54.0	14.2	29.9
Hybrid GMM/SVM CB3	51.8	54.3	11.4	27.0
Rel.red.% CB3	2.2	-0.7	20.3	9.7

Table 1. Identification error rates on the CB1- and CB3-trained systems for the four types of carbon button microphones (3-5 sec).

4.3. The Hybrid Classification System

The Fisher mapping transformed the original 38 dimensional input vector to a 2464 dimensional feature space. We trained 2652 binary classifiers (SVM classifiers and corresponding sigmoids). The time spent to the binary classifier training never exceeded 2-3 minutes. The same training and testing sets that were used for the baseline system were also used to train and test the hybrid system. For classification, the size of the N-best list was set to 10.

Table 1 shows the identification rates for the two systems (CB1 and CB3) and tests across all conditions (CB1 through CB4) as described above. Obviously, the microphone mismatch in training and testing is the most influent factor in performance degradation, however also the particular type of microphone appears to play a role, which can be seen in the difference between matched tests for CB1 and CB3.

The utility of the hybrid system is presented by the relative improvement in identification rate over the baseline system. In the matched (CB1/CB1 and CB3/CB3) and low mismatched (CB1/CB2 and CB3/CB4) tests the improvement ranges from 25.7% to 9.7% while in the strong mismatch the performances are in the same ballpark as the baseline system.

5. CONCLUDING REMARKS

Our main motivation to construct the hybrid classification system is to win both worlds by combining the descriptive strength of the baseline system with the high performance classification capabilities of SVM classifiers. Another reason to believe that one can gain form this type of combination is attributed to the fact that the training of discriminative models is done in a supervised manner, which obviously injects significant amount of additional information to the system.

Our experiments confirm these assumptions by showing that we can outperform the results of the baseline GMM system without increasing the training set. On the down side we observed that classifiers are much more affected by mismatch (since they focus on the decision boundary rather than the center of mass). Still, the combined performances never fall short of the baseline system.

We observed a significant amount of decorrelation between the error regions of the baseline and the hybrid classification system, even when their performances were comparable. By taking advantage of this phenomenon, one may hope to gain another improvement over the current results. This is the subject of our further research.

REFERENCES

[1] B. Boser, I. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In D. Haussler, editor, *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152. ACM Press, 1992.

[2] U.V. Chaudhari, J. Navrátil, and S.H. Maes. Multigrained data modeling for speaker recognition with sparse training and test data. In *Proc. of the International Conference on Spoken Language Processing (ICSLP)*, Beijing, October 2000. submitted.

[3] U.V. Chaudhari, J. Navrátil, S.M. Maes, and G. Ramaswamy. Very large population text-independent speaker identification. In *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 2001. submitted.

[4] T. G. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286, 1995.

[5] J.-L. Gauvain and C.-H. Lee. Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 2(2):291–8, April 1994.

[6] T. Hastie and R. Tibshirani. Classification by pairwise coupling. In M. S. Kearns, S. A. Solla, and D. A. Cohn, editors, *Advances in Neural Information Processing Systems*, volume 10. MIT Press, 1998.

[7] T. S. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In M. S. Kearns, S. A. Solla, and D. A. Cohn, editors, *Advances in Neural Information Processing Systems*, volume 11. MIT Press, 1999.

[8] N. Oliver, B. Schölkopf, and A. J. Smola. Natural regularization from generative models. In A. J. Smola, B. Schölkopf, P. L. Bartlett, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, chapter 4, pages 51–60. MIT Press, 2000.

[9] J. C. Platt. Fast trining support vector machines using sequential mininal optimization. In B. Schölkopf, C. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods*, chapter 12, pages 185–208. MIT Press, 1999.

[10] J. C. Platt. Probabilities for sv machines. In A. J. Smola, B. Schölkopf, P. L. Bartlett, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, chapter 5, pages 61–74. MIT Press, 2000.

[11] D.A. Reynolds. Htimit and llhdb: Speech corpora for the study of handset transducer effects. In *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1535–1538, Munich, Germany, May 1997.

[12] D.A. Reynolds, R.B. Dunn, and J.J. McLaughlin. The lincoln speaker recognition system: Nist eval2000. In *Proc. of the International Conference on Spoken Language Processing (ICSLP)*, Beijing, China, October 2000.

[13] V. N. Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer-Verlag, 1982.

[14] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.

[15] F. Weber, B. Peskin, M. Newman, and L. Gillick. Speaker recognition in two-speaker data: Recent results from dragon systems. In *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Istanbul, Turkey, 2000.