LARGE MARGIN GAUSSIAN MIXTURE MODELING FOR PHONETIC CLASSIFICATION AND RECOGNITION

Fei Sha and Lawrence K. Saul

University of Pennsylvania
Department of Computer and Information Science
3330 Walnut Street, Levine Hall
Philadelphia, PA 19104, USA

ABSTRACT

We develop a framework for large margin classification by Gaussian mixture models (GMMs). Large margin GMMs have many parallels to support vector machines (SVMs) but use ellipsoids to model classes instead of half-spaces. Model parameters are trained discriminatively to maximize the margin of correct classification, as measured in terms of Mahalanobis distances. The required optimization is *convex* over the model's parameter space of positive semidefinite matrices and can be performed efficiently. Large margin GMMs are naturally suited to large problems in multiway classification; we apply them to phonetic classification and recognition on the TIMIT database. On both tasks, we obtain significant improvement over baseline systems trained by maximum likelihood estimation. For the problem of phonetic classification, our results are competitive with other state-of-the-art classifiers, such as hidden conditional random fields.

1. INTRODUCTION

Much of the acoustic-phonetic modeling in automatic speech recognition (ASR) is handled by Gaussian mixture models (GMMs) [1]. It is widely recognized that maximum likelihood (ML) estimation of GMMs does not directly optimize the performance of these models as classifiers. It is therefore of interest to develop alternative learning paradigms that optimize discriminative measures of performance [1, 2, 3].

Support vector machines (SVMs) currently provide state-of-the-art performance for many problems in pattern recognition [4]. The simplest setting for SVMs is binary classification. If the positively and negatively labeled examples are linearly separable, SVMs compute the linear decision boundary that maximizes the margin of correct classification—that is, the distance of the closest example(s) to the separating hyperplane. If the labeled examples are not linearly separable, the kernel trick can be used to map the examples into a nonlinear feature space and to compute the maximum margin hyperplane in this space. Alternately, or in conjunction with the kernel trick, the optimization for SVMs can be relaxed to permit margin violations (i.e., incorrectly labeled examples) in the training data.

For various reasons, it can be challenging to apply SVMs to large problems in multiway as opposed to binary classification. First, to apply the kernel trick (which is required for nonlinear decision boundaries), one must construct a large kernel matrix with as many rows and columns as training examples. Second, the training complexity increases with the number of classes, depending to some extent on the way that binary SVMs are generalized to multiway classification.

In this paper, we develop a framework for large margin classification by GMMs. As in SVMs, our approach is based on the idea of margin maximization. Intuitively, we show how to train "large margin" GMMs that maximize the Mahanalobis distance of labeled examples from the decision boundaries that define competing classes. As in SVMs, the parameters of large margin GMMs are trained by a convex optimization that focuses on examples near the decision boundaries. After developing the basic approach in section 2, we discuss extensions for segmental training and outlier handling in section 3 and report experimental results on phonetic classification and recognition in section 4.

Our approach has certain advantages over SVMs for large problems in multiway classification. For example, the classes in large margin GMMs are modeled by ellipsoids—which induce nonlinear decision boundaries in the input space—as opposed to the half-spaces and hyperplanes in SVMs. Because the kernel trick is not necessary to induce nonlinear decision boundaries, large margin GMMs are more readily trained on very large and difficult data sets, as arise in ASR.

2. LARGE MARGIN MIXTURE MODELS

We begin by describing large margin GMMs in the simple setting where each class is modeled by a single ellipsoid. We then extend this framework to the case where each class is modeled by one or more ellipsoids. Finally, we relate our framework to other discriminative paradigms that have been proposed for training GMMs.

2.1. Large margin classification

The simplest large margin GMM represents each class of labeled examples by a single ellipsoid. Each ellipsoid is parameterized by a vector "centroid" $\boldsymbol{\mu} \in \Re^d$ and a positive semidefinite "orientation" matrix $\boldsymbol{\Psi} \in \Re^{d \times d}$. These parameters are analogous to the means and inverse covariance matrices of multivariate Gaussians, but they are not estimated in the same way. In addition, a nonnegative scalar offset $\theta \geq 0$ for each class is used in the scoring procedure.

Let $(\boldsymbol{\mu}_c, \boldsymbol{\Psi}_c, \theta_c)$ denote the centroid, orientation matrix, and scalar offset representing examples in class c. We label an example $\mathbf{x} \in \Re^d$ by whichever ellipsoid has the smallest Mahanalobis distance (plus offset) to its centroid:

$$y \;=\; \operatorname*{argmin}_c \left\{ (\mathbf{x} - \boldsymbol{\mu}_c)^{\mathrm{T}} \boldsymbol{\Psi}_c (\mathbf{x} - \boldsymbol{\mu}_c) + \theta_c \right\}. \qquad (1)$$

The goal of learning is to estimate the parameters $(\boldsymbol{\mu}_c, \boldsymbol{\Psi}_c, \theta_c)$ for each class of labeled examples that optimize the performance of this decision rule.

It is useful to collect the ellipsoid parameters of each class in a single enlarged $(d{+}1){\times}(d{+}1)$ positive semidefinite matrix:

$$\boldsymbol{\Phi}_c = \begin{bmatrix} \boldsymbol{\Psi}_c & -\boldsymbol{\Psi}_c \boldsymbol{\mu}_c \\ -\boldsymbol{\mu}_c^{\mathrm{T}} \boldsymbol{\Psi}_c & \boldsymbol{\mu}_c^{\mathrm{T}} \boldsymbol{\Psi}_c \boldsymbol{\mu}_c + \theta_c \end{bmatrix}. \qquad (2)$$

We can then rewrite the decision rule in eq. (1) as simply:

$$y = \operatorname*{argmin}_c \left\{ \mathbf{z}^{\mathrm{T}} \boldsymbol{\Phi}_c \, \mathbf{z} \right\} \quad \text{where} \quad \mathbf{z} = \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}. \qquad (3)$$

Here, $\mathbf{z} \in \Re^{d+1}$ is the vector created by appending a unit element to $\mathbf{x} \in \Re^d$. In this transformed representation, the goal of learning is simply to estimate the single matrix $\boldsymbol{\Phi}_c \in \Re^{(d+1)\times(d+1)}$ for each class of labeled examples.

We now consider the problem of learning in more detail. Let $\{(\mathbf{x}_n, y_n)\}_{n=1}^N$ denote a set of N labeled examples drawn from C classes, where $\mathbf{x}_n \in \Re^d$ and $y_n \in \{1, 2, \ldots, C\}$. In large margin GMMs, we seek matrices $\boldsymbol{\Phi}_c$ such that all the examples in the training set are correctly classified by a large margin—i.e., situated far from the decision boundaries that define competing classes. For the n^{th} example with class label y_n, this condition can be written as:

$$\forall c \neq y_n, \quad \mathbf{z}_n^{\mathrm{T}} \boldsymbol{\Phi}_c \mathbf{z}_n \;\geq\; 1 + \mathbf{z}_n^{\mathrm{T}} \boldsymbol{\Phi}_{y_n} \mathbf{z}_n. \qquad (4)$$

Eq. (4) states that for each competing class $c \neq y_n$, the Mahalanobis distance (plus offset) to the c^{th} centroid exceeds the Mahalanobis distance (plus offset) to the target centroid by a margin of at least one unit.

We adopt a convex loss function for training large margin GMMs. Analogous to SVMs, the loss function has two terms, one that penalizes margin violations of eq. (4) and one that regularizes the matrices $\boldsymbol{\Phi}_c$. Letting $[f]_+ = \max(0, f)$ denote the so-called "hinge" function, we can write the loss function for large margin GMMs as:

$$\mathcal{L} = \gamma \sum_n \sum_{c \neq y_n} \left[1 + \mathbf{z}_n^{\mathrm{T}} (\boldsymbol{\Phi}_{y_n} - \boldsymbol{\Phi}_c) \mathbf{z}_n \right]_+ + \sum_c \operatorname{trace}(\boldsymbol{\Psi}_c). \qquad (5)$$

The second term regularizes the orientation matrices $\boldsymbol{\Psi}_c$ which appear in the $d \times d$ upper left blocks of $\boldsymbol{\Phi}_c$. In *realizable* settings, where all the examples can be correctly classified, the second term favors the "minimum trace" Mahalanobis metrics consistent with the unit margin constraints in eq. (4). The relative weight of the two terms is controlled by a hyperparameter $\gamma > 0$ set by cross-validation.

The loss function in eq. (5) is a piecewise linear, *convex* function of the matrices $\boldsymbol{\Phi}_c$, which are further constrained to be *positive semidefinite*. Its optimization can thus be formulated as a problem in semidefinite programming [5]. Such problems can be generically solved by interior point algorithms with polynomial time guarantees (though we implemented a special-purpose solver using gradient-based methods for the results in this paper). Most importantly, eq. (5) has the desirable property that its optimization is not plagued by spurious local minima.

2.2. Mixture models

We now extend the previous model to represent each class by multiple ellipsoids. This is analogous to modeling each class by its own GMM, as opposed to a single Gaussian. Let $\boldsymbol{\Phi}_{cm}$ denote the matrix for the m^{th} ellipsoid in class c. The most straightforward extension is to imagine that each example $\mathbf{x}_n$ has not only a class label y_n, but also a mixture component label m_n. The latter labels are not provided a priori, but we can generate "proxy" labels by fitting a GMM to the examples in each class by ML estimation, then for each example, computing the mixture component with the highest posterior probability under this GMM. Given joint labels (y_n, m_n), we replace the large margin criterion in eq. (4) by:

$$\forall c \neq y_n, \quad -\log \sum_m e^{-\mathbf{z}_n^{\mathrm{T}} \boldsymbol{\Phi}_{cm} \mathbf{z}_n} \;\geq\; 1 + \mathbf{z}_n^{\mathrm{T}} \boldsymbol{\Phi}_{y_n m_n} \mathbf{z}_n. \qquad (6)$$

Eq. (6) implies that the match to *any* centroid m in any competing class $c \neq y_n$ is worse than the match to the target centroid m_n in class y_n by a margin of at least one unit. To see this, note that $\min_m a_m \geq -\log \sum_m e^{-a_m}$.

The loss function for mixture models is a simple extension of eq. (5). We replace the hinge loss in the first term by $[1 + \mathbf{z}_n^{\mathrm{T}} \boldsymbol{\Phi}_{y_n m_n} \mathbf{z}_n + \log \sum_m e^{-\mathbf{z}_n^{\mathrm{T}} \boldsymbol{\Phi}_{cm} \mathbf{z}_n}]_+$, which penalizes violations of the margin inequalities in eq. (6). The regularizer in the second term changes only to sum over different classes and mixture components: $\sum_{cm} \operatorname{trace}(\boldsymbol{\Psi}_{cm})$. Due to the "softmin" operation over mixture components, the resulting loss function is no longer piecewise linear in the matrices $\boldsymbol{\Phi}_{cm}$; however, it is easy to verify that it remains convex. Thus, even the optimization of this more general loss function for large margin GMMs is quite tractable.

2.3. Relation to previous work

Our framework differs in important aspects from previous frameworks for discriminative training of GMMs. Suppose

the class-conditional densities $p(\mathbf{x}|y)$ are modeled by GMMs. One common approach to discriminative training [2] estimates the means, covariance matrices, and mixture weights of these models that maximize the *conditional* log-likelihood $\sum_n \log p(y_n|\mathbf{x}_n)$. Such models generally outperform GMMs that are estimated by maximizing the *joint* log-likelihood $\sum_n \log p(\mathbf{x}_n, y_n)$. In contrast to our framework, however, the optimization of GMM parameters in this way is not convex. Moreover, as a loss function, the conditional log-likelihood does not focus on examples near the decision boundaries nor incorporate the idea of a large margin.

Recently, Liu et al [6] proposed a margin-based framework for discriminative training of GMMs in continuous-density hidden Markov models (HMMs). Unlike our work, however, the optimization in their framework is not convex, and they focus on GMMs with diagonal covariance matrices.

Finally, we revisit the comparison between large margin GMMs and SVMs, as discussed in section 1. In large margin GMMs, classes are modeled by one or more ellipsoids (as opposed to half-spaces); hence, the kernel trick is not required to induce nonlinear decision boundaries. Though one can generate quadratic decision boundaries in SVMs using polynomial kernels, large margin GMMs differ from such SVMs by restricting their quadratic forms to be positive semidefinite, thus imagining the support of each class as some *bounded* region in input space. In addition, the large margin GMMs in section 2.2, with *multiple* ellipsoids per class, cannot be represented by SVMs with polynomial kernels.

3. EXTENSIONS

Two further extensions of large margin GMMs are important for problems in ASR: handling of outliers, and segmental training. We describe each extension in isolation, assuming for simplicity that each class is modeled by a single ellipsoid, as in section 2.1. The generalization to the large margin GMMs described in section 2.2 is straightforward, as is the handling of outliers in combination with segmental training.

3.1. Handling of outliers

Many discriminative learning algorithms are sensitive to outliers. The loss function in eq. (5), in particular, does not closely track the classification error rate when the training data has many outliers. We adopt a simple strategy to detect outliers and reduce their malicious effect on learning.

Outliers are detected using ML estimates of the mean and covariance matrix of each class. These estimates are used to initialize matrices $\mathbf{\Phi}_c^{\mathrm{ML}}$ of the form in eq. (2). Then, for each example $\mathbf{x}_n$, we compute the accumulated *hinge loss* incurred by violations of the large margin constraints in eq. (4):

$$h_n^{\mathrm{ML}} = \sum_{c \neq y_n} \left[1 + \mathbf{z}_n^{\mathrm{T}}(\mathbf{\Phi}_{y_n}^{\mathrm{ML}} - \mathbf{\Phi}_c^{\mathrm{ML}})\mathbf{z}_n\right]_+ \quad (7)$$

Note that $h_n^{\mathrm{ML}} \geq 0$ measures the decrease in the loss function when an initially misclassified example $\mathbf{x}_n$ is corrected during the course of learning. We associate outliers with large values of h_n^{ML}.

Outliers distort the learning process by diverting its focus away from misclassified examples that could otherwise be easily corrected. In particular, correcting *one* badly misclassified outlier decreases the cost function proposed in eq. (5) more than correcting *multiple* examples that lie just barely on the wrong side of a decision boundary. To fix this situation, we reweight the hinge loss terms in eq. (5) involving example $\mathbf{x}_n$ by a multiplicative factor of $\min(1, 1/h_n^{\mathrm{ML}})$. This reweighting equalizes the losses incurred by all initially misclassified examples, thus reducing the malicious effect of outliers. We compute the weighting factors once from the ML estimates and hold them fixed during discriminative training. In practice, this scheme appears to work very satisfactorily.

3.2. Segmental training

The margin constraints in eq. (4) apply to individually labeled examples. We can also relax them to apply, collectively, to multiple examples known to share the same class label. This is useful for ASR, where we can train on variable-length "segments", consisting of multiple consecutive analysis frames, all of which belong to the same phoneme. Specifically, let p index the ℓ frames in the n^{th} phonetic segment $\{\mathbf{x}_{np}\}_{p=1}^{\ell}$. For segmental training, we rewrite the constraints in eq. (4) as:

$$\forall c \neq y_n, \quad \frac{1}{\ell}\sum_p \mathbf{z}_{np}^{\mathrm{T}}\mathbf{\Phi}_c\mathbf{z}_{np} \;\geq\; 1 + \frac{1}{\ell}\sum_p \mathbf{z}_{np}^{\mathrm{T}}\mathbf{\Phi}_{y_n}\mathbf{z}_{np}, \quad (8)$$

where the scores on both sides have been normalized by the segment length. The segment-based constraint in eq. (8) is especially well motivated if a segment-based decision rule is used for classification (e.g., $y = \mathrm{argmin}_c \sum_p \mathbf{z}_p^{\mathrm{T}}\mathbf{\Phi}_c\mathbf{z}_p$) as opposed to the frame-based rule in eq. (1).

4. EXPERIMENTAL RESULTS

We applied large margin GMMs to well-benchmarked problems in phonetic classification and recognition on the TIMIT database [3, 7, 8, 9]. We used the standard partition of training and test data and the same development set as in earlier work [3, 9]. All *sa* sentences were excluded. We mapped the 61 phonetic labels in TIMIT to 48 classes and trained ML and large margin GMMs for each class. Results were evaluated by mapping these 48 classes to 39 phones to remove further confusions, as in previous benchmarks. Our front end computed mel-frequency cepstral coefficients (MFCCs) with 25 ms windows at a 10 ms frame rate. We retained the first 13 MFCC coefficients of each frame, along with their first and second time derivatives. GMMs modeled these 39-dimensional feature vectors after they were whitened by PCA.

# of mixture components	classification		recognition	
	baseline	margin	baseline	margin
1	32.1%	24.3%	40.1%	34.7%
2	30.1%	23.4%	36.5%	33.5%
4	27.8%	22.3%	34.7%	32.7%
8	25.9%	**21.1%**	32.7%	31.1%
16	26.0%	21.4%	31.7%	**30.1%**

Table 1. Error rates for phonetic classification and recognition on the TIMIT database. Large margin GMMs are compared to baseline GMMs trained by ML estimation. See text for details.

4.1. Phonetic classification

Phonetic classification is an artificial but instructive problem in ASR. One assumes in this case that the speech has been correctly segmented into phonetic units, but that the phonetic class label of each segment is unknown. The input to the classifier is the "segment" of consecutive analysis frames that spans precisely one phoneme. We trained large margin GMMs using the segment-based margin criteria in section 3.2 and compared them to baseline (full covariance) GMMs trained by ML estimation. The baseline GMMs were also used to determine the proxy labels for mixture components in eq. (6) and to detect and reweight outliers, using eq. (7). We used the development data set to choose the hyperparameter $\gamma > 0$ in eq. (5), to tune a unigram language model, and to perform early stopping of the optimization procedure. The training time on 1.1M frames (roughly, 140K segments) ranged from 2-9 hours depending on the model size.

Table 1 shows the percentage of incorrectly classified phonetic segments on the TIMIT test set. Large margin GMMs consistently and significantly outperform baseline GMMs with equal numbers of mixture components. The best large margin GMM also yields a slightly lower classification error rate than state-of-the-art results (21.7%) obtained by hidden conditional random fields [3].

4.2. Phonetic Recognition

The same baseline and large margin GMMs were used to build phonetic recognizers. The recognizers were first-order HMMs with one context-independent state per phonetic class. Baseline or large margin GMMs were used in these HMMs to compute the log probabilities (or scores) of observed frames. The weighting of log transition probabilities in all HMMs was optimized on the development set. Table 1 compares the phone error rates of these HMMs, obtained by aligning the results of Viterbi decoding with the ground-truth phonetic transcriptions [7]. Again, the large margin GMMs lead to consistently lower error rates, here computed as the sum of substitution, deletion, and insertion error rates.

5. CONCLUSION

We have shown how to learn GMMs for multiway classification based on similar principles as large margin classification in SVMs. Classes are represented by ellipsoids whose location, shape, and size are discriminatively trained to maximize the margin of correct classification, as measured in terms of Mahalanobis distances. The required optimization is convex over the model's parameter space of positive semidefinite matrices. On problems in phonetic classification and recognition, large margin GMMs led to significant improvement over baseline GMMs. In ongoing work, we are investigating the use of context-dependent phone models, which are known to reduce phone error rates [7, 8]. We are also studying schemes for integrating the large margin training of GMMs with sequence models such as HMMs and/or conditional random fields [3].

Acknowledgments

This work was supported by NSF award 0238323. We thank A. Gunawardana (Microsoft Research) and K. Crammer (University of Pennsylvania) for many useful discussions and helpful correspondence.

6. REFERENCES

[1] S. Young, "Acoustic modelling for large vocabulary continuous speech recognition," in *Computational Models of Speech Pattern Processing*, Keith Ponting, Ed., pp. 18–39. Springer, 1999.

[2] A. Nadas, "A decision-theoretic formulation of a training problem in speech recognition and a comparison of training by unconditional versus conditional maximum likelihood," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 31, no. 4, pp. 814–817, 1983.

[3] A. Gunawardana, M. Mahajan, A. Acero, and J. C. Platt, "Hidden conditional random fields for phone classification," in *Proceedings of Eurospeech 2005*, Lisbon, 2005.

[4] B. Schölkopf and A. J. Smola, *Learning with Kernels*, MIT Press, Cambridge, MA, 2002.

[5] L. Vandenberghe and S. P. Boyd, "Semidefinite programming," *SIAM Review*, vol. 38(1), pp. 49–95, March 1996.

[6] Chaojun Liu, Hui Jiang, and Xinwei Li, "Discriminative training of CD-HMMs for maximum relative separation margin," in *Proceedings of ICASSP 2005*, Philadelphia, 2005, pp. 101–104.

[7] K. F. Lee and H. W. Hon, "Speaker-independent phone recognition using hidden Markov models," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 11, pp. 1641–1648, 1988.

[8] T. Robinson, "An application of recurrent nets to phone probability estimation," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 298–305, 1994.

[9] A. K. Halberstadt and J. R. Glass, "Heterogeneous acoustic measurements for phonetic classification," in *Proceedings of Eurospeech 97*, Greece, 1997, pp. 401–404.