LEARNING IMPROVED LINEAR TRANSFORMS FOR SPEECH RECOGNITION

Andrew Senior[*], *Youngmin Cho*[†], *Jason Weston*[*]

[*] Google Inc.,
New York
{andrewsenior,jweston}@google.com

[†] Department of Computer Science and Engineering,
UC San Diego
yoc002@cs.ucsd.edu

ABSTRACT

This paper explores a novel large margin approach to learning a linear transform for dimensionality reduction in speech recognition. The method assumes a trained Gaussian mixture model for each class to be discriminated and trains a dimensionality-reducing linear transform with respect to the fixed model, optimizing a hinge loss on the difference between the distance to the nearest in- and out-of-class Gaussians using stochastic gradient descent. Results are presented showing that the learnt transform improves state classification for individual frames and reduces word error rate compared to Linear Discriminant Analysis (LDA) in a large vocabulary speech recognition problem even after discriminative training.

Index Terms— Linear discriminant analysis, LDA, speech feature transformation, margin Mahalanobis distance, stochastic gradient descent.

1. INTRODUCTION

Dimensionality reduction is a fundamental aspect of many machine learning and pattern recognition tasks. Many researchers have investigated methods of reducing the dimensionality of high-dimensional data to avoid the curse of dimensionality and improve modelling fidelity. Large vocabulary speech recognition is one problem for which dimensionality reduction is a standard technique. Although many variants have been proposed, linear discriminant analysis (LDA) is still widely used to project speech features from a high-dimensionality space to one with fewer dimensions where probability densities can be better modelled by mixtures of multivariate Gaussian distributions [1]. Since LDA makes the assumption that each class can be modelled by a single Gaussian, with common covariance, which is not valid empirically, we are interested in investigating better methods of dimensionality reduction.

In this paper we investigate a new way of constructing a dimensionality-reducing linear projection to improve the modelling of the mixtures of Gaussians in a speech recognition system.

2. PREVIOUS WORK

Linear discriminant analysis (LDA) [2] is a fundamental technique which has spawned many variants. LDA calculates a dimensionality-reducing linear projection that maximizes a class separation criterion by solving a generalised eigen system. Heteroscedastic linear discriminant analysis (HLDA) [3] relaxes the equal-covariance assumption of LDA by taking into account the covariance of each class, but still assumes that each class is Gaussian-distributed. A mixture of Gaussians has been used to train LDA [4], but this work simply treated each mixture component as a separate class and found performance to be worse than treating HMM states as separate classes. Demuynck *et al.* [5] show failure modes of HLDA and describe MMI and MCE criteria for linear discriminant analysis.

The technique of Semi-tied covariance (STC) matrices [6] computes an optimal rotation of the feature space for the model, aligning the space so that diagonal covariance matrices model the data well. STC is not a dimensionality reduction technique, since it finds a rotation within a given subspace, but in conjunction with STC, LDA is found to perform nearly as well as HLDA with STC.

There has also been work on modifying LDA to weight the sum of the between-class scatter matrices in the LDA eigen system, including weighting by distance and weighting according to pairwise classification error [7]. Hastie *et al.* [8] have described a regularized version of LDA called Penalized Discriminant Analysis. De la Torre [9] relates LDA to PCA and a variety of other techniques as examples of a least squares weighted kernel reduced rank regression (LS-WKRRR) formulation. Other authors have used neural network architectures, particularly bottle-neck autoencoders [10] for nonlinear dimensionality reduction of speech features.

In contrast to these methods, our model takes into account our assumption that each class can be modelled with a mixture of Gaussians, (as opposed to a single Gaussian) which is exactly the model that is trained *after* dimensionality reduction. While some work has used a mixture of Gaussians in training dimensionality reduction transformations, this has focused on classification problems without the scale and time-dependence of speech recognition, for instance Peltonen *et al.* [11] maximize the likelihood of the training data by alternately optimizing the subspace and (by Expectation Maximization) the mixture model parameters. Torkkola [12] maximizes a Mutual Information criterion and uses a GMM represenation to reduce the computation.

Many algorithms have been proposed in the more general class of distance metric learning algorithms. Here, the goal is to transform, either linearly or nonlinearly, the feature representation of an object such that pairs of objects in the derived space have more semantically meaningful distances. If one constrains the transformation to be a mapping into a lower dimensional space than the original one, then this can be used as a dimensionality reduction technique, as in LDA. However, many methods [13, 14] learn a Mahalanobis distance so the dimensionality stays the same. Other nonlinear methods such as LLE [15] do learn a low-dimensional map, but it only applies to the data that it was trained on and does not readily generalize to out-of-sample points. In constrast to LDA the interesting thing about many of these metric learning approaches is that they do not make

[†]Work done while at Google.

a single Gaussian per-class assumption. Typically, metric learning trains on pairs of examples, and concentrates not on all pairs of distances, but only enforcing that points in the same class that are also near each other (by finding the k nearest neighbors in input space) should be mapped close to each other. As we will see, this approach is related to the method we use in this work.

3. ALGORITHM

Contrary to LDA, we start with the assumption that each class in our labeled training set will be modelled with a *mixture* of Gaussians. Our aim is to find an optimal linear mapping $A \in \mathbb{R}^{d \times D}$ from the original D-dimensional space to the lower d-dimensional space where Gaussian mixture modelling will be performed. To do this, we first assume that *we already have an initial matrix A^0 (from LDA+STC) and a trained Gaussian mixture model*. Our goal will then be to improve the parameters A (compared to A^0) with respect to the mixture model parameters. Subsequently, we can then try to improve the mixture model parameters with respect to A. We denote the Gaussian mixture model parameters as having means $C = \{c_i \in \mathbb{R}^d\}$ and inverse covariance matrices $\Psi_i \in \mathbb{R}^{d \times d}$, $i = 1, \ldots, |C|$ where there are $|C|$ Gaussians overall, and each Gaussian is assigned to a class label $Y_i \in \{1, \ldots, k\}$.

For a Gaussian mixture model with good discriminative ability, for a given example frame x with known label y we should expect that:

$$\min_{i:Y_i=y} (Ax - c_i)^\top \Psi_i (Ax - c_i) < \min_{i:Y_i \neq y} (Ax - c_i)^\top \Psi_i (Ax - c_i). \quad (1)$$

That is, the closest Gaussian to the example belongs to the same class as the example's label. If we can learn the parameters A such that as many of these constraints are fulfilled as possible, then we are in effect learning a linear transform that takes into account the assumption that our classes should be modelled by the mixture of Gaussians.

We thus write the objective function of our dimensionality reduction criterion as follows:

$$\min_A \sum_i \max\Big(0, \tau + \min_{j \in F(x_i, y_i)} (Ax_i - c_j)^\top \Psi_j (Ax_i - c_j) - \min_{j \in E(x_i, y_i)} (Ax_i - c_j)^\top \Psi_j (Ax_i - c_j)\Big) \quad (2)$$

where we have a training set $(x_i, y_i)_{i=1,\ldots,m} \in \mathbb{R}^D \times \{1, \ldots, k\}$ of labeled examples, and τ is the margin (a hyperparameter chosen in advance). Here, we have approximated counting the constraints (1) using the margin-based hinge loss, which is a well-known technique [13, 14].

We define $F(x_i, y_i)$ to be the set of candidate Gaussians of the same class ("friends") and $E(x_i, y_i)$ to be the set of candidate Gaussians of differing classes ("enemies") which the minimums operate over. Following Equation 1 we would define $F(x_i, y_i) = \{j : Y_j = y_i\}$ and $E(x_i, y_i) = \{j : Y_j \neq y_i\}$ however as we shall see this makes the optimization problem too expensive as large-scale speech systems can have hundreds of thousands of Gaussians. Hence, we make the following choice instead: $F(x_i, y_i)$ is defined as a "shortlist" of the closest n Gaussians to $A^0 x_i$ (precomputed using the initial linear transform) that belong to class y_i and $E(x_i, y_i)$ is similarly the closest n Gaussians to $A^0 x_i$ that do not belong to class y_i. We choose n to be small (e.g. 40), and because we use A^0 we can compute this fixed set in advance (before we optimize the parameters using our objective function). During optimization, we still search for the closest Gaussians with respect to the current parameters A, but searching over the shortlists rather than all Gaussians is much faster.

Algorithm 1 Online LTGMM Optimization

Input: labeled data (x_i, y_i), Gaussian mixture model (C, Ψ, Y), initial matrix A^0, margin τ and learning rate η.
for $i = 1$ to m **do**
  Compute shortlists $E(x_i, y_i)$ and $F(x_i, y_i)$ using C, Ψ, Y and A^0.
end for
Initialize $A \leftarrow A^0$.
repeat
  Pick a random labeled example (x_i, y_i)
  Compute closest "friend" and "enemy" Gaussians:
  $f^* = \text{argmin}_{f \in F(x_i, y_i)} (Ax_i - c_f)^\top \Psi_f (Ax_i - c_f)$
  $e^* = \text{argmin}_{e \in E(x_i, y_i)} (Ax_i - c_e)^\top \Psi_e (Ax_i - c_e)$
  if $\tau + (Ax_i - c_{f^*})^\top \Psi_{f^*} (Ax_i - c_{f^*}) - (Ax_i - c_{e^*})^\top \Psi_{e^*} (Ax_i - c_{e^*}) > 0$ **then**
    Make a gradient step:
    $A \leftarrow A - \eta \left(\Psi_{f^*} (Ax_i - c_{f^*}) - \Psi_{e^*} (Ax_i - c_{e^*}) \right) x_i^\top$
  end if
until validation error does not improve.

We note that one might ask how important it is to have a good approximation of these minima, given that they are expensive to compute? Indeed, other algorithms for example for metric learning [13, 14] methods are faced with a similar problem (see Section 2) but their approximations avoid computing minimums altogether. We will show in our experiments that such coarse approximations do not work well for our task, and in fact attempting to find the closest Gaussians is very important for good performance.

Unfortunately, our optimization problem has no closed form solution, but we can still optimize it using stochastic gradient descent [16]. Pseudocode of our method, called Linear Transform for a Gaussian Mixture Model (LTGMM), is given in Algorithm 1. Finally, after learning the transform, it is possible to fix A and then re-learn the parameters C, Ψ, Y of the mixture model, as described in Section 4.2.

4. EXPERIMENTS

Experiments were conducted with Google's Voice Search system [17]. This is a large vocabulary speech recognition system which recognizes user queries spoken into a mobile telephone and returns the Google search results for the recognized text. For this paper we experimented exclusively with the US English Voice Search data and recognizer. For training we use 1.9 million manually-transcribed anonymized utterances and 10 million utterances transcribed with our recognizer. A further 27,000 manually transcribed utterances were used for testing. The language model has of the order of 1 million tokens. All utterances were randomly selected among those received by our live production system.

This system has 41 phones and 7959 decision tree cross-word triphone states and uses a finite state transducer for search. Speech is represented as 13-dimensional PLP cepstral coefficients, with online cepstral mean normalisation. Training consists of maximum likelihood (ML) training using 39 dimensional PLP, delta and delta-delta features, followed by LDA to reduce dimensionality from 117 (a window of 9 frames) to 39 and maximum likelihood retraining,

Method	State Error rate (%)
LDA	60.05
LTGMM	58.13
LTGMM - no min over $E(x, y)$	73.92

Table 1. Context independent state classification error rates for original transform before and after training LTGMM with or without closest negative Gaussian selection. (7959 state model.)

with a global semi-tied covariance matrix transformation [6], followed by boosted maximum mutual information (BMMI) discriminative training [18].

Frame classification experiments with this model measured the error rate in assigning 10,000 speech frames, from an independent development set, to the 126 context-independent states, with ground truth given by forced alignment of the transcript. In testing, the projected frames were assigned to the most likely Gaussian and then labelled according to the context dependent state whose mixture model contained that Gaussian. Context dependent states were mapped to the corresponding context independent state.

In all experiments, LTGMM was initialized with the LDA projection.

4.1. Negative example selection

In a preliminary experiment, two methods of selecting the negative Gaussian were compared. Results are shown in Table 1. It was found that choosing the nearest negative example in Equation 2 was necessary to achieve reduction in the state classification rate. Choosing a random negative Gaussian from the shortlist failed to decrease the state classification error rate. Figure 1 shows the average state

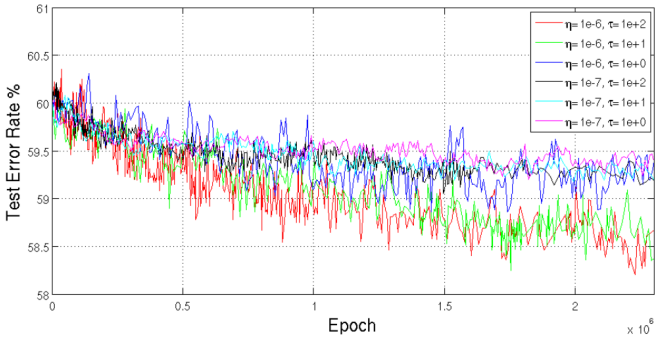

Fig. 1. State classification error rate against training epoch (in millions) for a variety of (η, τ) hyperparameter values.

classification rate evaluated periodically during training, for different values of η and τ. In practice $\eta = 10^{-6}$ and $\tau = 10$ provided good results on the frame classification task. For these parameters a single machine can complete around 450 epochs per second, or 10^6 epochs in 40 minutes. HLDA, with 200 iterations of gradient descent took over 4 hours on a similar machine.

Figure 2 shows the effect on the classification error for each state separately. The initial training is shown in the left-hand plot. All the states lie close to or below the line $y = x$, indicating that classification improves for all classes. However silence 98 is far below the line indicating that much of the net improvement is just from improving silence classification. After excluding silence from the

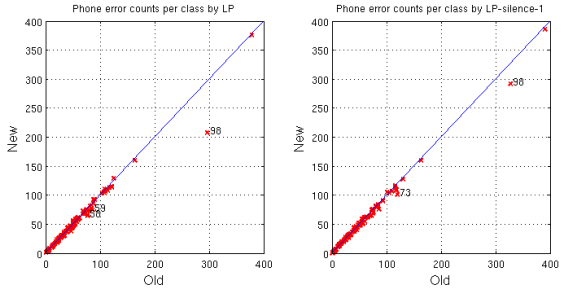

Fig. 2. Per-state state classification error counts before (x axis) and after (y axis) retraining with LTGMM. (left) training included silence frames. (right) training excluded silence frames.

Method	State Error rate (%)	WER
LDA	60.05	14.2
LTGMM	59.32	14.1

Table 2. Word error rate results of training LTGMM using the 7959 state baseline model. No silence frames were used in LTGMM training.

training, we find that class 98 is still better classified. Table 2 shows state classification and word error rates for the original LDA transform and the LTGMM-trained transform after training without silence frames. The transform used was the one with minimum development set phone classification error rate. The development error rate was reevaluated for each model every few minutes during training. A small reduction is seen in both frame classification and word error rates.

4.2. Retraining

Although the LTGMM optimization is constructed to improve the linear projection with respect to the current acoustic model, having constructed the projection we can retrain the acoustic model. Figure 3 shows the training schemes that have been applied, with circled numbers showing the systems evaluated. The baseline ML + BMMI training is ①.

These experiments were carried out on a 10,068 state model trained similarly on 1.7M automatically transcribed utterances (about 1700 hours of speech) which allowed for a reasonable turnaround for retraining experiments. Word error rates are show in Table 3 for the full large vocabulary voice search task with decoding parameters tuned for maximum accuracy. For these experiments a new manually-transcribed 27,000 utterance (over 150,000 words) test-set and new language model were used that both matched the more recent training data.

With the 10,068-state model, a linear projection was trained with LTGMM as in the previous section and tested with the original model ③. In addition, a new set of Gaussian mixture models was trained using the same training procedure as was previously used after LDA (Viterbi ML training followed by BMMI) but using the LTGMM transform ④. This is contrasted with similar training after HLDA ②. All CD retraining started with the alignments from the CD model marked $*$ in Figure 3.

As can be seen from Table 3, as in the previous experiment, the LTGMM transform performs slightly better than the original LDA linear transform. After ML retraining with STC and discriminative

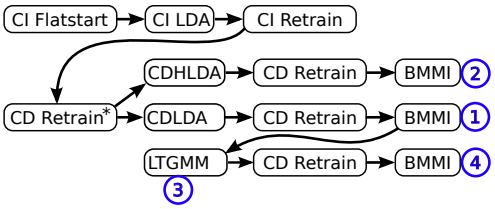

Fig. 3. Training alternatives explored with the 10068 state model, composed of context independent (CI) and context dependent (CD) maximum likelihood (ML) training and boosted MMI (BMMI). ①. . .④ are the evaluated systems of Table 3.

Method	ML WER %	BMMI WER %
LDA ①	22.2	16.9
HLDA ②	22.0	16.6
LTGMM + original GMM model ③	–	16.8
LTGMM + retrained GMM ④	22.1	16.6

Table 3. Word error rates after training the linear transform A using LTGMM and subsequently retraining the mixture model parameters with full Viterbi training followed by BMMI. These are compared to baseline systems with LDA and HLDA transforms. ①. . .④ are the systems of Figure 3.

training with BMMI the model using the LTGMM transform outperforms the seed LDA transform but the word error rate is the same as for HLDA.

5. CONCLUSIONS

We have shown that we can exploit the non-Gaussian distribution of speech states to learn an improved linear transformation for speech recognition. Compared to LDA, the learnt transform can improve both state classification accuracy and overall speech word error rate on a large vocabulary task, and delivers similar performance to HLDA. Improvements were seen both while keeping the mixture model fixed and after retraining the mixture model. Further iterations of retraining A and c_i, Ψ_i parameters may bring about further improvements but have not yet been attempted.

In future work, we have begun to investigate learning nonlinear transformations instead of the linear transformation used here. This technique might also be applied to dimensionality reduction on higher-dimensional features than the conventional PLP features used here.

6. REFERENCES

[1] M.J.F. Gales and S.J. Young, "The application of hidden Markov models in speech recognition," *Foundations and Trends in Signal Processing*, vol. 1, no. 3, pp. 195, 2007.

[2] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification*, Wiley Interscience, 2001.

[3] N. Kumar and A.G. Andreou, "Heteroscedastic discriminant analysis and reduced rank HMMs for improved speech recognition," *Speech Communication*, vol. 26, no. 4, pp. 283–297, 1998.

[4] R. Haeb-Umbach and H. Ney, "Linear discriminant analysis for improved large vocabulary continuous speech recognition," in *ICASSP*, 1992, vol. I, pp. 13–16.

[5] K. Demuynck, J. Duchateau, and D. Van Compernolle, "Optimal feature sub-space selection based on discriminant analysis," in *Eurospeech*, 1999.

[6] M.J.F. Gales, "Semi-tied covariance matrices for hidden Markov models," *IEEE Transactions on Speech and Audio Processing*, vol. 7, pp. 272–281, 1999.

[7] H.S. Lee and B. Chen, "Linear discriminant analysis feature extraction using weighted classification confusion information," in *Interspeech*, 2008.

[8] T. Hastie, A. Buja, and R. Tibshirani, "Penalized discriminant analysis," *Annals of Statistics*, vol. 23, no. 1, pp. 73–102, 1995.

[9] F. de la Torre, "A least-squares framework for component analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence*, September 2011.

[10] F. Grezl, M. Karafiat, S. Kontar, and J. Cernocky, "Probabilistic and bottle-neck features for LVCSR of meetings," in *ICASSP*, 2007.

[11] J. Peltonen, J. Goldberger, and S. Kaski, "Fast semi-supervised discriminative component analysis," in *MLSP*, 2007.

[12] K. Torkkola, "Learning discriminative feature transforms to low dimensions in low dimensions," in *NIPS*, 2001.

[13] K.Q. Weinberger, J. Blitzer, and L.K. Saul, "Distance metric learning for large margin nearest neighbor classification," in *NIPS*.

[14] G. Chechik, V. Sharma, U. Shalit, and S. Bengio, "Large scale online learning of image similarity through ranking," *J. Mach. Learn. Res.*, vol. 11, pp. 1109–1135, 2010.

[15] S.T. Roweis and L.K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[16] L. Bottou, "Stochastic gradient learning in neural networks," in *Proceedings of Neuro-Nîmes 91*, 1991.

[17] J. Schalkwyk, D. Beeferman, F. Beaufays, B. Byrne, C. Chelba, M. Cohen, M. Kamvar, and B. Strope, "Google Search by Voice: A case study," in *Visions of Speech: Exploring New Voice Apps*. Springer, 2010.

[18] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah, "Boosted MMI for model and feature-space discriminative training," in *ICASSP*, 2008.