

# Dimensionality Reduction for Speech Recognition Using Neighborhood Components Analysis

Natasha Singh-Miller, Michael Collins, Timothy J. Hazen

Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology  
Cambridge, MA, U.S.A

natashas@mit.edu, mcollins@csail.mit.edu, hazen@mit.edu

## Abstract

Previous work has considered methods for learning projections of high-dimensional acoustic representations to lower dimensional spaces. In this paper we apply the *neighborhood components analysis* (NCA) [2] method to acoustic modeling in a speech recognizer. NCA learns a projection of acoustic vectors that optimizes a criterion that is closely related to the classification accuracy of a nearest-neighbor classifier. We introduce regularization into this method, giving further improvements in performance. We describe experiments on a lecture transcription task, comparing projections learned using NCA and HLDA [1]. Regularized NCA gives a 0.7% absolute reduction in WER over HLDA, which corresponds to a relative reduction of 1.9%.

**Index Terms:** speech recognition, acoustic modeling, dimensionality reduction

## 1. Introduction

Previous work [1] introduced heteroscedastic discriminant analysis (HLDA) as a method for learning projections of high-dimensional acoustic representations into lower-dimensional spaces. Acoustic vectors in the high-dimensional space can be created by concatenating the MFCC representations of multiple consecutive frames. The goal of projection methods is to find a lower-dimensional representation that captures the information required for discrimination of different phonemes, and can be used within a conventional speech recognizer that employs Gaussian mixture models for acoustic modeling. Finding a low-dimensional representation reduces the number of parameters that must be trained for the acoustic model and thereby has the potential to reduce the amount of overtraining in the model.

In this paper, we contrast HLDA with an approach for dimensionality reduction, *neighborhood components analysis* (NCA), introduced by Goldberger et al. [2]. NCA selects a projection that optimizes the performance of a nearest neighbor classifier in the projected space. NCA and HLDA both make use of training sets consisting of acoustic vectors and their associated class labels in order to learn projections that will be effective

at separating classes in the projected space. However, HLDA makes stronger assumptions about the distribution of samples in each class than NCA; specifically, HLDA assumes that each class of acoustic vectors have a normal distribution. Because NCA optimizes for a nearest neighbor classifier, the method makes weaker assumptions about the shape of the distribution in each class, making it a closer match to the use of mixtures of Gaussians which are eventually employed in modeling these distributions in the acoustic model.

We present the NCA method, along with discussion of specific implementation issues. We extend the method by introducing regularization which we determine can be useful in reducing WER further. Our end goal is to use these projections to lower WER in a large vocabulary speech recognition task. Academic lecture data [3, 4] is used to train and test our approach. In our experiments, we compare NCA, principal components analysis (PCA), and HLDA and show that NCA outperforms both other methods, showing a 2.7% absolute improvement in WER over a class-based PCA projection, and a 0.7% absolute (1.9% relative) improvement over HLDA.

## 2. Neighborhood Components Analysis

NCA was introduced by [2]; we describe the details of the method here for completeness. NCA learns a linear projection of vectors into a space that optimizes a criterion related to the leave-one-out accuracy of a nearest neighbor classifier on a training set. Specifically, NCA takes as input a training set consisting of vectors  $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$  where  $\mathbf{x}_i \in \mathcal{R}^m$  and an associated set of labels  $\{y_1, y_2, \dots, y_N\}$  where  $y_i \in \mathcal{L}$ . For example, in our experiments  $\mathbf{x}_i$  consist of concatenated vectors of MFCC measurements and  $y_i$  indicates the class of phonemic event described by the vector, such as /oy/. The method then learns a projection matrix  $\mathbf{A}$  of size  $p \times m$  that projects the training vectors  $\mathbf{x}_i$  into a  $p$  dimensional representation,  $\mathbf{z}'_i = \mathbf{A}\mathbf{x}_i$ , where a nearest neighbor classifier is effective at discriminating amongst the classes. This projection matrix  $\mathbf{A}$  defines a Mahalanobis distance metric that can be used by the nearest neighbor classifier in the projected space.

$$d(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{A}\mathbf{x}_i - \mathbf{A}\mathbf{x}_j)^T(\mathbf{A}\mathbf{x}_i - \mathbf{A}\mathbf{x}_j)$$

By selecting  $p < m$ , we can learn a lower dimensional representation of the original acoustic vectors.

The goal of the method is to learn a projection  $\mathbf{A}$  that maximizes the accuracy of a nearest neighbor classifier. In order to define a differentiable optimization criterion, the method makes use of “soft-neighbor” assignments instead of directly using the  $k$  nearest neighbors. Specifically, each point  $j$  in the training set has a probability  $p_{ij}$  of assigning its label to a point  $i$  that decays as the distance between points  $i$  and  $j$  increase.

$$p_{ij} = \frac{\exp(-\|\mathbf{A}\mathbf{x}_i - \mathbf{A}\mathbf{x}_j\|^2)}{\sum_{k \neq i} \exp(-\|\mathbf{A}\mathbf{x}_i - \mathbf{A}\mathbf{x}_k\|^2)}, \quad p_{ii} = 0$$

The method attempts to maximize the expected number of points correctly classified in a leave-one-out setting over the training set. This optimization criterion can be defined using the soft-neighbor assignments. First a quantity  $p_i$  is defined that denotes the probability of a point  $i$  being assigned the correct class label.

$$p_i = \sum_{j \in C_i} p_{ij}$$

$$C_i = \{j | y_j = y_i\}$$

The final optimization criterion  $f(\mathbf{A})$  can then be defined simply as the sum of the probabilities of classifying each point correctly.

$$f(\mathbf{A}) = \sum_i p_i$$

This criterion gives rise to a gradient rule that can be used to optimize the matrix  $\mathbf{A}$ . (Note that  $\mathbf{x}_{ij}$  is shorthand for  $\mathbf{x}_i - \mathbf{x}_j$ .)

$$\frac{\partial f}{\partial \mathbf{A}} = 2\mathbf{A} \sum_i \left( p_i \sum_k p_{ik} \mathbf{x}_{ik} \mathbf{x}_{ik}^T - \sum_{j \in C_i} p_{ij} \mathbf{x}_{ij} \mathbf{x}_{ij}^T \right)$$

This function can be optimized using a number of gradient methods, such as stochastic gradient ascent, or conjugate gradient ascent. Note that the function  $f(\mathbf{A})$  is not convex, so care needs to be taken when initializing the matrix  $\mathbf{A}$  in order to avoid sub-optimal solutions.

### 2.1. Computational Performance

The calculation of the above gradient can be computationally quite expensive. Calculating the soft-neighbor probabilities alone requires  $O(N^2p)$  calculations. However, many of these probabilities will be very close to zero, allowing us to truncate the gradient calculation.

Additionally, we can reduce the amount of computation by re-arranging terms of the gradient as follows.

$$\frac{\partial f}{\partial \mathbf{A}} = 2 \sum_i \left( p_i \sum_k p_{ik} (\mathbf{A}\mathbf{x}_{ik}) \mathbf{x}_{ik}^T - \sum_{j \in C_i} p_{ij} (\mathbf{A}\mathbf{x}_{ij}) \mathbf{x}_{ij}^T \right)$$

In our experiments we optimize  $f(\mathbf{A})$  using conjugate gradient ascent, which we parallelize across several machines.

### 2.2. Regularization

We can introduce regularization into the NCA optimization criterion in order to alleviate a few possible problems with the method. Regularization can help counteract over-fitting effects we might see with the training data. The other problem we seek to address with regularization is specifically related to the definition of the soft-neighbor assignments used by the method. Because soft-neighbor assignments  $p_{ij}$  decay very rapidly with distance, as the magnitude of  $\mathbf{A}$  increases the effective number of nearest neighbors influencing the labeling of a point decreases. If the magnitude of  $\mathbf{A}$  grows sufficiently large, the method might simply consider just the one closest neighbor, which could lead to a quite suboptimal projection of the data. We therefore introduce the following regularized version of the optimization function where  $C$  is a constant chosen by optimizing the leave-one-out performance over a development set.

$$f_{reg}(\mathbf{A}) = \frac{1}{N} \sum_i p_i - C \sum_{j,k} A_{j,k}^2$$

where  $A_{j,k}$  indicates the element at the  $j$ th row and  $k$ th column of matrix  $\mathbf{A}$ .

## 3. Experiments

We compare NCA and HLDA using speech recognition experiments on academic lectures [3, 4]. We train the HLDA projection using the code provided by [1] for HLDA using full covariance matrices. For training we have 121 hours of speech collected from a wide variety of lectures predominantly obtained from the MIT World collection [5, 6]. For testing we have 6 hours of similar data. We use the SUMMIT recognizer [7] in our experiments.

Each acoustic sample is represented using a 112-dimensional feature vector, consisting of the concatenation of eight 14-dimensional feature vectors. Each of these vectors contain 14 MFCC measurements taken at eight telescoped time intervals around the point of the acoustic sample.

The training set was manually transcribed, with time alignments of words and phonetic events obtained via forced transcription using our baseline recognizer. From the forced transcriptions, labels for each acoustic feature vector are extracted. These labels correspond to 1837 phonetic classes including context independent phone internal labels (e.g. /f/ or /ae/) and diphone transition labels (e.g. /f/ -> /ae/ or /s/ -> /t/). In our experiments, we make use of 53 context-independent internal phone classes only. We use a small portion of this data, 500 samples from each of 53 phonemic classes, to

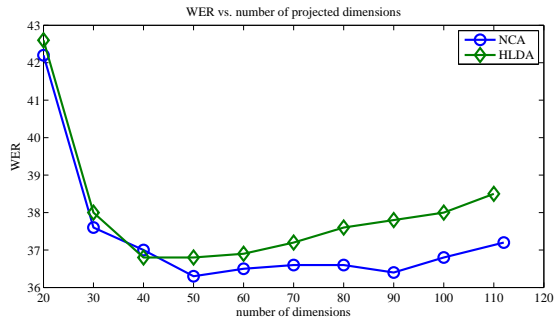


Figure 1: WER of HLDA and NCA on a test set for several different dimensionalities.

train a projection using the NCA method. These samples are randomly selected and come from a number of different speakers. To train the HLDA projection we use all of the training data across the 53 phonemic classes, around 4 million samples.

Because the optimization function for NCA is non-convex, care needs to be taken when initializing  $\mathbf{A}$ . In our experiments, we initialize the matrix randomly for both the NCA and regularized NCA projections.

### 3.1. Speech Recognition

Our end goal is to use NCA in order to reduce speech recognition word error rates. Because our recognizer employs mixtures of Gaussians with diagonal covariance matrixes, and because we model 1837 instead of 53 classes used to train the projections, we apply a class-based PCA after learning NCA and HLDA. This class-based whitening transform is performed by applying PCA to a covariance matrix obtained by pooling the covariances of the individual phonetic classes. We compare the WER achieved by both the unregularized version of NCA and HLDA for a number of dimensions in the projected space in Figure 1. As the number of dimensions is increased, initially large improvements in WER are achieved for both methods, but these level-off quickly at around 40 dimensions. The minimal WER achieved by NCA occurs at 50 dimensions. As the number of dimensions increases beyond 90, the performance of the recognizer begins to deteriorate, indicating an over-training effect. A similar trend is seen with the HLDA method, with optimal performance achieved at 40 and 50 dimensional projections.

In Table 1 we compare the performance of the recognizer using class-based PCA, HLDA, NCA, and the regularized version of NCA for 50-dimensional projections. Regularized NCA achieves a large improvement over classed-PCA alone of 2.7%, and a significant improvement over HLDA as well of 0.7% (1.9% relative improvement). Regularization also improves the performance of NCA, with the regularized method achieving an

	WER
PCA	38.8
HLDA, all training data	36.8
HLDA, 500 samples per class	37.1
NCA, 500 samples per class	36.3
NCA (regularized), 500 samples per class	36.1

Table 1: Word error rate of recognizer using PCA, HLDA, NCA, and regularized NCA to learn 50 dimensional projections.

improvement of 0.2% over the baseline NCA method.

We additionally experimented with increasing the number of data points per class used to train NCA to 1000 and 5000. These experiments led to negligible differences in speech recognition WER.

### 3.2. Discussion

By looking at the classification accuracy achieved in a kNN setting using both HLDA and regularized NCA, we can identify some of the differences between the two methods. We can calculate the accuracy of a kNN classifier on a test set of acoustic samples (i.e. a set of samples not used to train the NCA or HLDA projections). The classification performance of the internal phonemic classes ordered by reduction in error rate achieved using regularized NCA are shown in Table 2. A 50-dimensional projection is learned for both regularized NCA and HLDA and both are trained with 500 samples from each of the 53 internal phonemic classes. NCA achieves large improvements in classification accuracy across almost all the phonemic classes.

Another fact to note is that while the kNN performance of NCA and HLDA are very different, the difference in recognition performance of the two methods in terms of WER is not as large. This suggests that large increases in kNN performance does not necessarily mean large improvements in WER. A direction for future work would be to try to more directly employ the kNN framework to acoustic modeling, as there may be a mismatch between the NCA framework and the mixture of Gaussians employed by the acoustic model.

## 4. Related Work

There are several alternatives to HLDA that also learn discriminative projections. In [8], an alternative to HLDA is presented that optimizes a minimum phoneme error (MPE) criterion. This type of discriminative projection can also be used for speaker adaptation [9]. One problem with NCA is that because the optimization function is non-convex, it is easy to converge to local optima. An alternative method to NCA [10] presents a convex optimization function related to NCA that may help solve this problem.

Phoneme	NCA Acc.	HLDA Acc.	% Red. in Err.
/epi/	88.60	62.40	69.68
/em/	79.00	56.60	51.61
/ah_ɪp/	72.80	48.80	46.88
/g/	77.80	58.80	46.12
/b/	80.40	66.40	41.67
/pɪ/	63.40	39.80	39.20
/zh/	78.00	65.20	36.78
/p/	67.80	50.00	35.60
/jh/	55.00	34.20	31.61
/uw/	56.60	37.20	30.89
/oy/	51.80	30.60	30.55
/sh/	67.00	52.60	30.38
/w/	72.20	60.40	29.80
/ey/	62.40	46.60	29.59
/tɪ/	49.60	28.80	29.21
/y/	67.20	54.40	28.07
/ay/	48.00	28.40	27.37
/uh/	58.20	42.60	27.18
/axr/	56.00	40.20	26.42
/v/	53.60	37.00	26.35
/en/	55.60	40.60	25.25
/d/	46.40	29.00	24.51
/z/	56.40	43.00	23.51
/f/	63.80	52.80	23.31
/ch/	51.00	37.00	22.22
/ng/	59.40	48.00	21.92
/k/	62.20	52.40	20.59
/dx/	49.60	36.60	20.50
/th/	52.40	40.60	19.87
/dh/	55.80	45.80	18.45
/t/	53.40	43.40	17.67
/ae/	37.40	24.40	17.20
/aw/	39.40	27.60	16.30
/hh/	54.20	45.40	16.12
/dɪ/	39.20	27.80	15.79
/bɪ/	50.00	41.80	14.09
/ao/	45.20	36.40	13.84
/iy/	55.40	48.40	13.57
/gɪ/	42.00	33.00	13.43
/er/	43.40	35.20	12.65
/el/	66.80	62.00	12.63
/l/	25.40	14.80	12.44
/ow/	31.80	22.80	11.66
/r/	40.40	33.00	11.04
/ih/	32.60	25.40	9.65
/eh/	31.80	24.60	9.55
/aa/	30.00	23.40	8.62
/ah/	27.20	21.20	7.61
/n/	39.00	34.20	7.29
/ax/	25.60	20.60	6.30
/m/	31.60	27.60	5.52
/kɪ/	45.20	43.20	3.57
/s/	57.20	57.40	-0.47

Table 2: Accuracy of a kNN classifier on a test set of acoustic vectors with their associated phonemic labels. Vectors are first projected into a 50-dimensional space using HLDA or regularized NCA trained on a training set of 500 points per class.

HLDA has also been effectively applied to speaker adaptation [11, 12]. For a single speaker, HLDA will most likely perform better than in the speaker-independent projections we learn here because multiple speakers can introduce a high amount of variability in the data. Specifically, the Gaussian assumption made by HLDA may be more likely to hold for a single speaker. When training a projection for a single speaker, it is possible that some of the gains seen when using NCA instead of HLDA would diminish.

## 5. Conclusion and Future Work

We have shown that NCA can deliver significant improvements in speech recognition WER over PCA and HLDA. However NCA has drawbacks, including computational cost and a non-convex optimization function. Efficient online methods of optimizing the NCA criterion should be investigated. Additionally, future work can compare NCA against similar methods with a convex optimization function. Finally, there is a mismatch between the nearest neighbor framework used by NCA and the mixture of Gaussians used by the recognizer. A more direct application of the NCA framework within the recognizer is a promising area for future exploration.

## 6. References

- [1] N. Kumar, *Investigation of silicon auditory models and generalization of linear discriminant analysis for improved speech recognition*, Ph.D. thesis, Johns Hopkins University, Baltimore, Maryland, 1997.
- [2] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov, "Neighbourhood components analysis," in *Advances in Neural Information Processing Systems 17*, L. K. Saul, Y. Weiss, and L. Bottou, Eds. 2005, pp. 513–520, MIT Press.
- [3] J. Glass, T. J. Hazen, L. Hetherington, and C. Wang, "Analysis and processing of lecture audio data: Preliminary investigations," in *HLT-NAACL 2004 Workshop on Interdisciplinary Approaches to Speech Indexing and Retrieval*, 2004, pp. 9–12.
- [4] A. Park, T. J. Hazen, and J. R. Glass, "Automatic processing of audio lectures for information retrieval: Vocabulary selection and language modeling," in *International Conference on Acoustics, Speech, and Signal Processing*, 2004, pp. 497–500.
- [5] "Mit open course ware," . <http://ocw.mit.edu>.
- [6] "Mit world," . <http://mitworld.mit.edu>.
- [7] J. Glass, "A probabilistic framework for segment-based speech recognition," *Computer, Speech, and Language*, vol. 17, no. 2-3, pp. 137–152, 2003.
- [8] B. Zhang and S. Matsoukas, "Minimum phoneme error based heteroscedastic linear discriminant analysis for speech recognition," in *International Conference on Acoustics, Speech, and Signal Processing*, 2005, pp. 925–928.
- [9] L. Wang and P.C. Woodland, "MPE-based discriminative linear transform for speaker adaptation," in *International Conference on Acoustics, Speech, and Signal Processing*, 2004, pp. 321–324.
- [10] A. Globerson and S. Roweis, "Metric learning by collapsing classes," in *Advances in Neural Information Processing Systems 18*, Y. Weiss, B. Scholkopf, and J. Platt, Eds. 2006, pp. 513–520, MIT Press.
- [11] G. Saon, G. Zweig, and M. Padmanabhan, "Linear feature space projections for speaker adaptation," in *International Conference on Acoustics, Speech, and Signal Processing*, 2001, pp. 325–328.
- [12] S. Matsoukas and R. Schwartz, "Improved speaker adaptation using speaker dependent feature projections," in *Workshop on Automatic Speech Recognition and Understanding*, 2003, pp. 273–278.