Kernel Metric Learning For Phonetic Classification

Jui-Ting Huang, Xi Zhou, Mark Hasegawa-Johnson, and Thomas Huang

Beckman Institute, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA
Dept. of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA
{jhuang29, xizhou2, jhasegaw}@illnois.edu
huang@ifp.uiuc.edu

***Abstract*—While a sound spoken is described by a handful of frame-level spectral vectors, not all frames have equal contribution for either human perception or machine classification. In this paper, we introduce a novel framework to automatically emphasize important speech frames relevant to phonetic information. We jointly learn the importance of speech frames by a distance metric across the phone classes, attempting to satisfy a large margin constraint: the distance from a segment to its correct label class should be less than the distance to any other phone class by the largest possible margin. Furthermore, an universal background model structure is proposed to give the correspondence between statistical models of phone types and tokens, allowing us to use statistical models of each phone token in a large margin speech recognition framework. Experiments on TIMIT database demonstrated the effectiveness of our framework.**

I. Introduction

While a sound spoken is described by a handful of frame-level spectral vectors, not all frames have equal contribution for either human perception or machine classification. For example, it has been showed that acoustic cues just after consonant release, and just before consonant closure, provide more phonetic information than acoustic cues during the closure interval for human and machine recognition [1]. Landmark-based speech recognition is one of the examples to consider salient acoustic cues (landmarks) in acoustic modeling. In [2], automatic speech recognition was performed by first detecting salient acoustic landmarks, then classifying the features of those landmarks. In [3], original spectral features were transformed into high-dimensional landmark-based representations by support vector machines. A Hidden Markov Model for each phone was then trained using the transformed features as input observations.

A key problem with the landmark-based method has always been its need for manually labeled data, in order to identify the critical phone boundary times that serve as anchor points with respect to which the timing of phonetic information is distributed [2], [3]. We seek, instead, to learn which frames are important directly from the data, because human annotations are expensive and somewhat sub-optimal. Particularly, a speech frame may have different importance in different phonemes, which implies the weights must be associated with phone classes. We propose to automatically weigh *important* acoustic observations relevant to phonetic information.

Recently, Frome *et. al* [4] proposed to adopt local distance functions to selectively weigh training patches for image classification. However, direct adaptation of their approach would be intractable to weigh feature frames of speech for two reasons. Firstly, directly estimating a frame-specific weight for every frame in a training database would be prone to overfitting as usually there are tens of millions speech frames. Secondly, the training process would need to iteratively compute the distance between all phone segment pairs; furthermore, without correspondence, the distance calculation exhaustively searches all the feature frame pairs, which exponentially increases the computation cost.

In this paper, we propose a new framework to automatically emphasize *important* acoustic observations relevant to phonetic information. In the framework, we first estimate an global Gaussian Mixture Model (GMM), called Universal background model (UBM), and then adapt it to obtain both phone-specific and token-specific (segment-specific) GMMs using a Maximum a posteriori (MAP) training criterion. Then we jointly learn the weights on a kernel distance metric across the phone classes based on the distances between segment-specific (token-specific) and phone-specific (type-specific) GMMs, attempting to satisfy a large margin constraint: the distance from a segment to its correct label class should be less than the distance to any other phone class by the largest possible margin. In this way, the weight of a Gaussian component of a phone-specific GMM is optimized, implicitly reflecting the importance of the acoustic frames associated with that component.

The new framework has five advantages: 1) Weighting on Gaussian components instead of feature frames controls the number of free parameters that need to be estimated and therefore makes the framework suitable on large scale problems. 2) UBM-MAP structure gives the correspondence across different GMMs, which greatly reduces the computation cost in the learning process. 3) UBM-MAP also provides a unified framework within which to compare phone types and segment tokens: each is a GMM. 4) Joint learning across the classes leads to a globally consistent distance metric that can be directly used in the testing phase. 5) Large margin constraints relate the kernel weights in a direct proportion to the number of misclassified phone segments, which matches the final evaluation criterion.

The paper is organized as follows: Section II-V discusses our approach in detail. In Section VI, we provide the phone classification experiments results on TIMIT dataset. Finally, Section VII draws the conclusion.

II. System Flow

The capability of UBM-MAP to represent small-sized samples, together with the correspondence of Gaussian components across different models adapted from UBM, allows us to propose a quite distinct framework from conventional speech recognition schemes: to learn a separate GMM statistical model for each segment token in the training database, and to let the segment models guide training of the phone token models using a large margin training criterion.

The system is described below. First, a UBM is trained using all training data. Then for each phone model, the mean vector is adapted from the UBM by MAP adaptation; we call this a *phone-specific* GMM. In the same time, for each phone segment, we also apply MAP adaptation, using the frames belonging to the same segment, to the UBM to obtain a *segment-specific* GMM. The distance between a phone and a segment is then evaluated using a Gaussian kernel metric. In the testing (classification) phase, for an unknown segment, we label it with the phone class that gives the minimum distance to that segment. In the training phase, we optimize the Gaussian kernel metric by optimizing the weights associated with Gaussian components (of phone GMMs) to satisfy a large-margin constraint, and the optimization problem can be formulated as a convex optimization problem.

In the following sections, we will describe (1) the UBM-MAP System, (2) the definition of Gaussian kernel metric, and (3) the learning process for the weights in Gaussian kernel metrics.

III. UBM-MAP System

A. Universal Background Model

For ease of presentation, we denote z as an acoustic feature frame. Then, the distribution of the variable z is

$$p(z;\Theta)=\sum_{k=1}^{K}\lambda_k\mathcal{N}(z;\mu_k,\Sigma_k), \tag{1}$$

where λ_k, μ_k and Σ_k are the weight, mean and covariance matrix of the kth Gaussian component, respectively, and K is the total number of Gaussian components in a UBM. The density is a weighted linear combination of K unimodal Gaussian densities, namely,

$$\mathcal{N}(z;\mu_k,\Sigma_k)=\frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma_k|^{\frac{1}{2}}}\ e^{-\frac{1}{2}(z-\mu_k)^T\Sigma_k^{-1}(z-\mu_k)}. \tag{2}$$

Many approaches can be proposed to estimate the model parameters. Here we obtain a maximum likelihood parameter set using the Expectation-Maximization (EM) algorithm. For computational efficiency, the covariance matrices are restricted to be diagonal.

B. MAP Adaptation

We obtain the phone-specific distribution model by adapting the mean vectors of the UBM and retaining the mixture weights and covariance matrices. For each phone ϕ, the mean vectors $\{\mu_{\phi,k}:k=1,2,\ldots,K\}$ are adapted using MAP adaptation as an one iteration EM. In the E-step, we compute the posterior probability:

$$Pr(k|z_{\phi,t}) = \frac{\lambda_k\mathcal{N}(z_{\phi,t};\mu_k,\Sigma_k)}{\sum_{j=1}^{K}\lambda_j\mathcal{N}(z_{\phi,t};\mu_j,\Sigma_j)}, \tag{3}$$

$$n_{\phi,k} = \sum_{t=1}^{T(\phi)}Pr(k|z_{\phi,t}), \tag{4}$$

where $z_{\phi,t}$ is the t-th frame belonging to phone ϕ in the training set, and $T(\phi)$ denotes the total number of feature frames belonging to ϕ.

Then the M-step updates the mean vectors, namely

$$E_{\phi,k}(Z) = \frac{1}{n_{\phi,k}}\sum_{t=1}^{T(\phi)}Pr(k|z_{\phi,t})z_{\phi,t}, \tag{5}$$

$$\hat{\mu}_{\phi,k} = \alpha_{\phi,k}E_{\phi,k}(Z)+(1-\alpha_{\phi,k})\mu_{\phi,k}^{(0)}, \tag{6}$$

where $\alpha_{\phi,k}=n_{\phi,k}/(n_{\phi,k}+r)$; $\mu_{\phi,k}^{(0)}$ is a prior mean. The larger r, the larger the influence of the prior distribution on the adaptation.

Similarly, we estimate a segment-specific GMM for each phone segment using Equation (3)-(6), except that T in Equation (4) is the number of frames belonging to the specific segment.

IV. Gaussian Kernel Metric

Since we have converted phone segments into GMMs, the distance between a phone class ϕ and a phone segment i can be obtained through the distance between their corresponding GMMs. An approximation to the Kullback-Leibler divergence from a phone model GMM to a phone segment GMM [6] is used as our distance metric:

$$\begin{aligned}D(\phi,i)&=\sum_{k=1}^{K}-\left(\sqrt{\lambda_k}\Sigma_k^{-\frac{1}{2}}\mu_{\phi,k}\right)^T\left(\sqrt{\lambda_k}\Sigma_k^{-\frac{1}{2}}\mu_{i,k}\right)\\&=\sum_{k=1}^{K}d_{\phi i,k},\end{aligned} \tag{7}$$

where λ_k and Σ_k are the universal weight and covariance for the k_{th} Gaussian component, and $\mu_{\phi,k}$ and $\mu_{i,k}$ denote the adapted means for the k_{th} Gaussian Component, for ϕ and i respectively. Furthermore, taking into account unequal importance of different Gaussians in different phones, we modified Equation (7) such that different Gaussian components, indexed by k, in phone model ϕ are assigned possibly different weights $w_{\phi,k}$:

$$D(\phi,i)=\sum_{k=1}^{K}w_{\phi,k}d_{\phi i,k}, \tag{8}$$

where $w_{\phi,k}$ is a non-negative value indicating the importance of the k^{th} Gaussian kernel in phone model ϕ; the larger $w_{\phi,k}$ shows more importance of the k^{th} Gaussian kernel in phone model ϕ.

V. Kernel Metric Learning

A. Optimization Problem

Based on the model-to-segment distance we just defined, the classification rule is simply as follows. For a given phone segment i, we choose the phone class that minimizes the distance to the segment:

$$\hat{\phi} = \arg\min D(\phi, i). \tag{9}$$

Under this setting, we choose to learn $w_{\phi,k}$ in Equation (8) in a large margin fashion, both because of its discriminative and nice generalization properties. Specifically, for each training segment i, with its corresponding true label ϕ, we want to ensure that the following inequality holds,

$$D(\phi', i) \geq D(\phi; i) + 1 \quad \forall \phi' \neq \phi, \tag{10}$$

that is, the distance from the true phone model ϕ to the segment model i should be less than any other phone model ϕ' to i by a margin. Denote the number of training segments as N and the number of phonemes as Φ, the total number of constraints given by Equation (10) is $N(\Phi - 1)$. To make our formula clear, in the following we will first define some notations, depicting the constraints in a matrix manner.

We concatenate the weights imposed in Equation (8) into a weight vector $W = [w_{1,1} \dots w_{1,K} \dots w_{\phi,k} \dots w_{\Phi,K}]^T$, whose total length is ΦK, where K is the number of Gaussian kernels. Similarly, for each constraint with respect to (i, ϕ') in Equation (10), we introduce a distance vector $X_{i\phi'}$ to be a vector of the same length as W, with all of its entries being 0 except the subranges corresponding to the true model ϕ and the competitor ϕ' for i, which are set to $d_{\phi i}$ and $-d_{\phi' i}$ respectively ($d_{\phi i} = [d_{\phi i,1} \dots d_{\phi i,K}]^T$). In this way, the constraints formulated in Equation (10) can be reformatted as

$$W^T X_{i\phi'} \geq 1 \quad \forall i, \phi' \neq \phi. \tag{11}$$

However, in a real world situation, the constraints can not be possibly satisfied simultaneously for all (ϕ, i, ϕ'). Therefore, a relaxation is needed in the final objective function. We relax the constraints by introducing a penalty term that penalizes linearly for deviation from the constraint; the empirical loss of our model is defined as the sum of the hinge losses over all constraints,

$$\sum_{i,\phi' \neq \phi} [1 - W \cdot X_{i\phi'}]_+, \tag{12}$$

where $[z]_+$ denote the function $\max\{0, z\}$. On the other hand, the regularization on W is necessary to prevent over-fitting. To this end, we impose an L_2 regularization penalty on W. The relative importance of these two criteria is specified by a hyper-parameter C, thus

$$\begin{aligned} W = \arg\min_W \frac{1}{2} \|W\|^2 + C \sum_{i\phi'} \xi_{i\phi'} \\ \text{s.t.} \quad \forall i, \phi' : \xi_{i\phi'} \geq 0 \\ \forall i, \phi' : W \cdot X_{i\phi'} \geq 1 - \xi_{i\phi'} \\ \forall \phi, k : w_{\phi,k} \geq 0. \end{aligned} \tag{13}$$

Here we introduce a slack variable $\xi_{i\phi'}$, as in the standard SVM soft-margin form, to allow for some points to be on the wrong side of the margin.

B. Dual Solver

To solve the optimization problem in Equation (13), we follow the work in [4], converting the problem into its dual form because the constraints on dual variables can be decoupled and thus easier to solve than the primal form. The dual form of the primal problem is[1]

$$\begin{aligned} \max_{\alpha, \Upsilon} & f(\alpha, \Upsilon) \\ \text{s.t.} \quad & \forall i, \phi' : 0 \leq \alpha_{i\phi'} \leq C \\ & \forall \phi, k : \upsilon_{\phi,k} \geq 0, \end{aligned} \tag{14}$$

where

$$f(\alpha, \Upsilon) = -\frac{1}{2} \left\| \sum_{i,\phi'} \alpha_{i\phi'} X_{i\phi'} + \Upsilon \right\|^2 + \sum_{i,\phi'} \alpha_{i\phi'}, \tag{15}$$

and $\Upsilon = [\upsilon_{1,1} \dots \upsilon_{1,K} \dots \upsilon_{\phi,k} \dots \upsilon_{\Phi,K}]^T$. In addition, the conversion to the dual gives the following relation between W and its dual vector Υ,

$$W = \sum_{i,\phi'} \alpha_{i\phi'} X_{i\phi'} + \Upsilon. \tag{16}$$

Since the constraints on the variable α and Υ in Equation (14) are all decoupled, and the objective function $f(\alpha, \Upsilon)$ is in a convex form, the dual problem can be easily solved by block coordinate methods [8], [4]. The basic idea is to update one variable at one iteration, minimizing the objective as other variables are fixed. In each iteration, the minimum point for $\alpha_{i\phi'}$ or Υ is obtained by setting the first partial derivatives of $f(\alpha, \Upsilon)$ to 0 and then clipping the values to the feasible regions (considering the boundary conditions in Equation (14)),

$$\hat{\alpha_{i\phi'}} \leftarrow \left[\frac{1 - \left\langle \left(\sum_{j,\psi \neq i,\phi'} \alpha_{j\psi} X_{j\psi} \right) \cdot X_{i\phi'} \right\rangle}{\|X_{i\phi'}\|^2} \right]_{[0,C]} \tag{17}$$

$$\Upsilon \leftarrow \max\left\{0, \textstyle\sum_{i,\phi'} \alpha_{i\phi'} X_{i\phi'}\right\} \tag{18}$$

Using Equation (16), updating Υ in Equation (18) is equivalent to updating W,

$$W \leftarrow \max\left(0, \sum_{i,\phi'} \alpha_{i\phi'} X_{i\phi'}\right). \tag{19}$$

To summarize, the updating process performs Equation (17) and Equation (19) iteratively, until the change of the dual function $f(\alpha, \Upsilon)$ is less than the threshold and most of

[1]The dual form is derived using a Lagrangian function associated with the primal problem. While the details are less relevant to the context of this paper, the interested reader is referred to Section 4.4.1 of [7] for the step-by-step derivation of a dual function.

the KKT conditions are satisfied. For our problem, KKT conditions are

$$\begin{aligned} \alpha_{i\phi'} = 0 &\Rightarrow \langle W \cdot X_{i\phi'} \rangle \geq 1 \\ 0 < \alpha_{i\phi'} < C &\Rightarrow \langle W \cdot X_{i\phi'} \rangle = 1 \\ \alpha_{i\phi'} > C &\Rightarrow \langle W \cdot X_{i\phi'} \rangle \leq 1. \end{aligned} \tag{20}$$

The practical optimizing procedure is detailed in Algorithm 1. Note that instead of sequentially updating $\alpha_{i\phi'}$ in the order of $\{(1,1)\cdots(N,\Phi)\}$, we randomly permute the order for each epoch to speed up the optimization process.

Algorithm 1 Dual solver for kernel selection

while $|\Delta f < \epsilon|$ **do**
  $A \leftarrow \{(1,1)\cdots(N,\Phi)\}$
  make a random permutation of A
  while $|\Delta f < \epsilon|$ **do**
    for $i \in A$ **do**
      if α satisfies KKT conditions **then**
        $A \leftarrow A \backslash i$.
        CONTINUE.
      else
        $g_i = W^T X_i - 1$
        $\bar{\alpha}_i \leftarrow \alpha_i$
        $\alpha_i \leftarrow \min(\max(\alpha_i - g_i / \|X_i\|^2), C)$
        $W \leftarrow \max(W + (\alpha_i - \bar{\alpha}_i) X_i, 0)$
      end if
    end for
  end while
end while

VI. Experiments

A. Experimental Setting

To evaluate the performance of our kernel metric learning, we conduct experiments on vowel classification using the TIMIT corpus [9]. A total of 16 vowels were used, including 13 monophthongal vowels /iy,ih,eh,ey,ae,aa,ah,ao,ow,uh,ux,er,uw/ and 3 diphthongs /ay,oy,aw/. The training set has 462 speakers, and a disjoint set of 50 speakers forms the evaluation set. The training and the evaluation set here are the same as the training and the development set defined in [10].

We focus on vowels, rather than all phones, because most phone classification experiments have reported that vowels are more difficult than phones in general. In [10], for example, the set of all phones was classified with 78.5% accuracy, but the set of vowels was classified with only 71.5% accuracy. In [10], the classifier was a segmental classifier with five subsegments per token; our system, with only three subsegments per token, may achieve lower accuracy than that reported by [10]. Also, a different set of vowels was used in [10]. To our knowledge, the best vowel classification using only three subsegments per token, for the same 16 vowel categories as used in this paper, is about 63% phone classification accuracy [11].

Frame-based spectral features (12 PLP coefficients plus energy) with a 5 ms frame rate and a 25 ms Hamming window, along with their delta and delta-delta are calculated. For phonetic classification, we assume that the speech has been segmented into phone units correctly. Within each phone segment, we divide the frames into three regions with 3-4-3 proportion, and each of three regions has a corresponding GMM, formed by the method described in III. Consequently, each phone class has $K = 3k$ Gaussian kernels, where k is the total number of Gaussian components in a prototype UBM.

TABLE I
Error rates for phonetic classification on the TIMIT database.

Methods	Accuracy(%)
Leung and Zue [11]	63
UBM-MAP	65.61
UBM-MAP with KML	**68.91**

B. Vowel Classification Accuracy

As shown in Table I, our UBM-MAP system performs better than the best result in [11], for the same 16 vowel categories. Furthermore, with kernel metric learning (KML), the improvement is significant (absolute 3.3%).

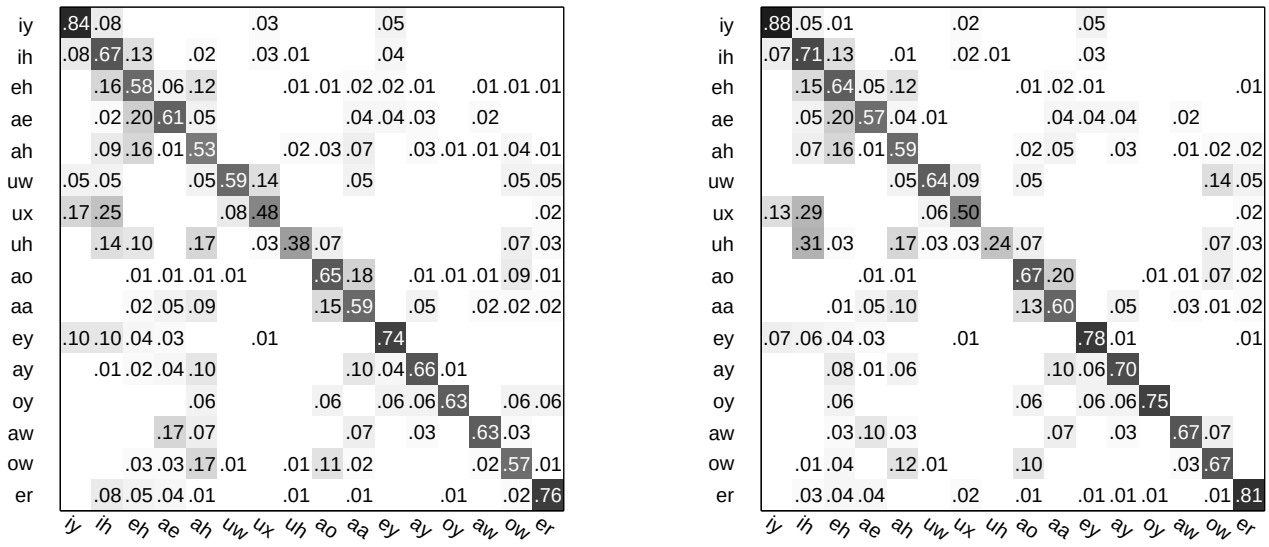
The classification errors also vary across different vowel/dithphong categories. To illustrate this, we show the confusion matrices of the classification results associated with UBM-MAP only and UBM-MAP with Kernel Metric Learning, respectively, in Figure 1. In our UBM-MAP only baseline, the long vowels/diphthongs generally attain higher classification accuracy than the short vowels. It can be explained by at least two causes. First, short vowels are subject to the reduction effect due to the phonetic context more severely. Second, long vowel segments comprise more frames, which can be better modeled under our framework as we apply MAP adaptation to each segment to obtain a segment-specific model, and more frames give a more reliable adapted model.

After Kernel Metric Learning, diphthongs generally have significant marginal gains over our UBM-MAP baseline (/oy/: 63% to 75%, /ey/: 74% to 78%), whereas several short vowels generally improve with smaller gain (/ao/: 65% to 67%, /aa/: 59% to 60%) or even possibly have degradations (/uh/: 38% to 24%, /ae/: 61% to 57%). These changes are consistent with what we expect with our framework. Short vowels have static vowel quality along the speech frames, while diphthongs and some long vowels are more nonstationary. Thus the ideally learned weight by KML should be more uniformly distributed for short vowels, which implies that short vowels (closer to the baseline), might benefit less from our weight-learning framework.

VII. Conclusions

In this paper, we introduce a novel framework that can learn a phone-dependent kernel metric that weighs important speech frames in a discriminative way. We jointly learn the importance of speech frames by a distance metric across the phone classes, which leads to a globally consistent distance metric that can be directly used in the testing phase. Also, large margin training relates the kernel weights in a direct proportion to the number of misclassified phone segments,

Fig. 1. The confusion matrices for UBM-MAP (left) and UBM-MAP with Kernel Metric Learning (right). The entry in the i^{th} row and j^{th} column is the percentage of speech segments from phone i that were classified as phone j. (For better viewing quality, refer to the electronic PDF file.)



which matches the final evaluation criterion. A UBM-MAP structure structure is proposed to give correspondence across phone and segment models, which reduces the complexity of the learning process and makes our framework appropriate to a large scale problem. Experiments on TIMIT database demonstrated the effectiveness of our framework. We also found that our framework can improve the classification of diphthongs more than other vowel categories.

ACKNOWLEDGMENT

This work was funded in part by the Disruptive Technology Office VACE III Contract issued by DOI-NBC, Ft. Huachuca, AZ; and in part by the National Science Foundation Grant NSF 07-03624 and IIS-0534133.

REFERENCES

[1] S. Furui, "On the role of spectral transition for speech perception," *Journal of the Acoustical Society of America*, vol. 80, no. 4, pp. 1016–1025, 1986.

[2] C. Y. Espy-Wilson, T. Pruthi, A. Juneja, O. Deshmukh, "Landmark-Based Approach to Speech Recognition: An Alternative to HMMs," in *INTERSPEECH 2007*, 2007, pp. 886–889.

[3] S. Borys, "An SVM Front End Landmark Speech Recognition System," Master's thesis, University of Illinois at Urbana-Champaign., Illinois, USA, 2008.

[4] A. Frome, Y. Singer, F. Sha, and J. Malik, "Learning globally-consistent local distance functions for shape-based image retrieval and classification," in *Proceedings of IEEE 11th International Conference on Computer Vision*, 2007, pp. 1–8.

[5] D. Reynolds, T. Quatieri, and R. Dunn. "Speaker Verification using Adapted Gaussian Mixture Models," *Digital Signal Processing*, vol. 10, no. 1-3, pp. 19-41, 2000.

[6] W. Campbell, D. Sturim, D. Reynolds, and A. Solomonoff. "SVM Based Speaker Verification using a GMM Supervector Kernel and NAP Variability Compensation," *ICASSP*, vol. 1, pp. 97-100, 2006.

[7] A. Frome, "Learning Local Distance Functions for Exemplar-Based Object Recognition," PhD thesis, EECS Department, University of California, Berkeley, 2007

[8] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, September 1999.

[9] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren, "Darpa timit acoustic phonetic continuous speech corpus," 1993.

[10] A. K. Halberstadt, "Heterogeneous acoustic measurements and multiple classifiers for speech recognition," Ph.D. dissertation, Massachusetts Institute of Technology, 1998.

[11] H. Leung and V. Zue, "Phonetic classification using multi-layer perceptrons," *ICASSP*, vol. 1, pp. 525-528, 1990.