Rapid Speaker Adaptation in Eigenvoice Space

by R. Kuhn, J-C. Junqua, P. Nguyen, and N. Niedzielski

Heeyoul (Henry) Choi
Dept. of Computer Science
Texas A&M University
hchoi@cs.tamu.edu
Outline

- Introduction
  - Speaker Adaptation
- Eigenvoice
- Comparison with others
  - MAP, MLLR, EMAP, RMP, CAT, RSW…
- Experiments
- Future work
- Summary
Introduction

- Speaker-dependent (SD) system
- Speaker-independent (SI) system

**Speaker Adaptation**

- Finding SD system for a new speaker with small data.

- This paper is about making the adaptation faster based on eigenvoice approach.
Speaker Adaptation

- Model-based algorithms
  - Adapt to a new speaker by modifying the parameters of the system’s speaker model.

- Maximum a posteriori (MAP), Maximum likelihood linear regression (MLLR).
  - Require significant amounts of adaptation data from the new speaker.
Speaker Adaptation

- **Speaker space algorithm**
  - Constrain the adapted model to be a linear combination of a small number of basis vectors from the reference speakers.
  - Faster and robust
    - Related to [speaker clustering](#) in fact that they reduce the parameter dimension to search.
    - Resemble [extended MAP (EMAP)](#) in fact that they use a priori information from reference speakers.
    - Actually, prior information is used to reduce the parameter space.

- Eigenvoice is one of these algorithm.
Speaker Adaptation

- Eigenvoice
  - Finds basis vectors that are orthogonal to each other
  - Efficient in the sense of variation.
  - Has all property of principal component analysis (PCA)

- PCA is applied to the parameter space.
Eigenface

- Eigenvoice is an analogy to eigenface in face images.
  - Face is a weighted sum of eigenfaces, which are eigenvectors of face images.

- PCA
  - Ordered by eigenvalues.
  - Guarantee the minimized mean-square error.
Eigenvoice

- Face recognition => speaker recognition? How?
- Speaker recognition vs. speech recognition

Efficient representation for speakers

A good SD model for the new speaker

Speech recognition of the new speaker
Eigenvoice

**OFFLINE STEPS**

HMMs

Train SD models for R speakers (+ 1 SI model)

From SD models, get R supervectors

Apply DRT to get R eigenvectors (eigenvoices): $e(0), e(1), ..., e(R-1)$

Keep first $K+1$ eigenvoices: $e(0), e(1), ..., e(K)$

**ONLINE STEPS**

PCA

Data from new speaker + eigenvoices + SI model

Estimate $K$ weights: $w(1), w(2), ..., w(K)$

Construct supervector for new speaker: $e(0)+w(1)e(1)+...+w(K)e(K)$

Adapted model for new speaker

Iterate

Texas A&M University
Eigenvoice (HMM)

- **Supervector**: Model parameters.
  - The means of HMM output Gaussians.
  - Not voice. (Actually, eigen_model_parameter)

- Instead of PCA,
  - independent component analysis (ICA) (Factorialvoice as in factorialface)
  - linear discriminant analysis (LDA) (generalized eigenvoice)

- They used correlation matrix instead of covariance matrix.
Eigenvoice (HMM)

- Hidden states of every SD model are from SI model.
  - Insufficient data for SD but enough for SI.
  - (SI + small data) is enough to make SD model.

- Hidden states means kind of Speaker invariant features.
  - Each speaker’s characters are from the mixture of Gaussian for each state after adaptation.
  - This makes sense to build new speakers model with the same states and transition probabilities as SI model.
Eigenvoice (PCA)

- Now, the adaptation procedure is redefined as finding K parameters.
  - Computationally cheap.
    - Only the weights of eigenvoices instead of all parameters.
  - Requires only a small amount of adaptation data
    - Think about the ‘curse of dimensionality’
  - And robust against noise
    - The discarded eigenvectors corresponding to small eigenvalues might be about ‘noise’.
Initialization of weights and the others (variances and transition probabilities) are from SI model.

The parameters for the new speaker is

\[ P = \varepsilon(0) + w(1) \varepsilon 1 + \cdots + w(K) \varepsilon(K). \]

- The problem is to estimate the weight \( w(j) \) from data.

Maximum likelihood eigen-decomposition (MLED)

- Gaussian mean adaptation in a continuous density hidden Markov model (CDHMM).
Eigenvoice (EM-MLED)

- Likelihood ($\lambda = \{\text{means of Gaussian}\}$)
  \[
P(O \mid \lambda) = \sum_{m,s} P(O, m, s \mid \lambda) = \sum_{m,s} P(m, s \mid \lambda) P(O \mid m, s, \lambda)
  \]

- Auxiliary function
  \[
  Q(\lambda, \hat{\lambda}) = \sum_{m,s} P(O, m, s \mid \lambda) \log P(O, m, s \mid \hat{\lambda})
  
  P(O, m, s \mid \hat{\lambda}) = \prod_{t} P(o_t, m, s \mid \hat{\lambda}) = \prod_{t} P(o_t \mid m, s, \hat{\lambda}) P(m, s \mid \hat{\lambda})
  \]

- $\gamma_{m}^{(s)}(t) = P(m, s \mid \lambda, o_t)$ (s-m occupation prob.)
- $P(o_t \mid m, s, \hat{\lambda})$ is a Gaussian distribution.
Finally,

\[ Q(\lambda, \hat{\lambda}) = -\frac{1}{2} P(O|\lambda) \times \sum_s \sum_m \sum_t \gamma_m^{(s)}(t)f(o_t, s, m) \]

where

\[ f(o_t, s, m) = [-n \log(2\pi) - \log |C_m^{(s)}| + h(o_t, s, m)] \]

and

\[ h(o_t, s, m) = \left( \mu_m^{(s)} - o_t \right)^T C_m^{(s)-1} \left( \mu_m^{(s)} - o_t \right) \]
Eigenvoice (EM-MLED)

- In the Gaussians, the mean estimates are

\[
\hat{\mu} = \begin{bmatrix}
\hat{\mu}_1^{(1)} \\
\hat{\mu}_2^{(1)} \\
\vdots \\
\hat{\mu}_m^{(s)} \\
\vdots
\end{bmatrix} = \sum_{j=1}^{K} w(j)e(j)
\]

- Parameters are reduced from the D-dimensional means into the K-dimensional weights (K \ll D).
To maximize \( Q(\lambda, \hat{\lambda}) \), set \( (\partial Q / \partial w(j)) = 0, j = 1 \cdots K \); assuming the eigenvalues are independent, \( (\partial w(i) / \partial w(j)) = 0, i \neq j \). One obtains for \( j = 1 \cdots K \)

\[
\sum_s \sum_m \sum_t \gamma_m^{(s)}(t) \left( e_m^{(s)}(j) \right)^T C_m^{(s)-1} o_t
\]

\[
= \sum_s \sum_m \sum_t \gamma_m^{(s)}(t)
\]

\[
\times \left\{ \sum_{k=1}^K w(k) \left( e_m^{(s)}(k) \right)^T C_m^{(s)-1} e_m^{(s)}(j) \right\}
\]
Comparison (MAP and MLLR)

- **Model based** algorithm.
- **Maximum a Posteriori (MAP)**
  - Uses the prior information of parameters in Bayes rule.
  - Updates only the parameters of Gaussians that have observations $o_t$.
  - The number of parameters are large.
- **Maximum Likelihood Linear Regression (MLLR)**
  - Update all the parameters which is formulated by the linear regression.
  - Much less constrained by prior knowledge. A little by SI model.
  - The number of parameters are small. (the transformation matrix)

- **Eigenvoice**
  - puts a heavy emphasis on prior knowledge by eigenvectors
  - updates all the parameters by the weights. (Eigenvector is D-dimension)
  - The number of parameters are smaller than MLLR.
Comparison (RMP)

- Extended Maximum a Posteriori (EMAP)
  - Faster convergence by the correlations between observation.
    - Update all the correlated parameters with one observation as much as they are related.

- Regression-based model prediction (RMP)
  - EMAP to CDHMM
  - Still faster than MAP.
  - Better performance than MAP.
  - Kind of a mixture of MLLR and MAP
Comparison (Clustering)

- Hard speaker clustering
  - Clusters of reference speakers – SI models (cf. codewords)
  - When the new speaker data is available, choose one model.
  - And then MLLR can be used.

- Soft speaker clustering
  - The new speaker’s model is a linear combination of reference speaker’s models.
  - Clustering + MLLR
    - Clustering works as prior to MLLR which has a little prior information.
Comparison (RSW)

- **Reference Speaker Weighting (RSW)**
  - The new speaker’s model is a linear combination of the reference models.
    \[ m = w(1)m(1) + \cdots + w(R)m(R) \]
  - The rest part is same as eigenvoice method.
  - Good with medium or large-vocabulary systems.
  - For class \( p \) and speaker \( r \)

\[
\mu_m^{(s)}(p, r) = c(p, r) + \nu_m^{(s)}
\]

\[
m(r) = [c(1, r), c(2, r), \cdots, c(P, r)] \text{; speaker dependent}
\]

- \( \nu \) is speaker independent
- For new speaker model, from \( m \), the vector \( m(S) \) is obtained by means of the ML. (equivalent to finding the weights)
- As the number \( R \) of reference speakers grows, it becomes more expensive in terms of memory and computation.
Comparison (Vowel Classification)

- Vowel Classification
  - PCA to the parameters of a vowel classifier (MoG)
Experiments

- **Database**
  - \( R = 120 \) (reference speakers) and 30 test speakers
  - \( D = 2808 \) (parameter dimension)
    - = 26 characters x 6 states (single Gaussian) x 18 features
  - PCA (0.,,, K eigenvoices ) and MLED (2 iterations)

- **In figure**
  - MLED.5 : MLED with K=5
  - MLED.10=>MAP : MAP with MLED.10 model as prior
  - Otherwise, SI is prior.
Experiments - Accuracy

% recognition

# of letters of adaptation data

Texas A&M University
# Experiments - Accuracy

**MLED.5 Recognition Rate (1 Letter of Adaptation Data)**

<table>
<thead>
<tr>
<th>RANK</th>
<th>LETTER</th>
<th>% CORRECT</th>
<th>RANK</th>
<th>LETTER</th>
<th>% CORRECT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>'V'</td>
<td>85.7</td>
<td>10</td>
<td>'L'</td>
<td>84.0</td>
</tr>
<tr>
<td>2</td>
<td>'D'</td>
<td>85.6</td>
<td>10</td>
<td>'X'</td>
<td>84.0</td>
</tr>
<tr>
<td>2</td>
<td>'T'</td>
<td>85.6</td>
<td>10</td>
<td>'Y'</td>
<td>84.0</td>
</tr>
<tr>
<td>3</td>
<td>'G'</td>
<td>85.5</td>
<td>11</td>
<td>'F'</td>
<td>83.9</td>
</tr>
<tr>
<td>3</td>
<td>'J'</td>
<td>85.5</td>
<td>11</td>
<td>'T'</td>
<td>83.9</td>
</tr>
<tr>
<td>4</td>
<td>'C'</td>
<td>85.3</td>
<td>11</td>
<td>'S'</td>
<td>83.9</td>
</tr>
<tr>
<td>4</td>
<td>'E'</td>
<td>85.3</td>
<td>12</td>
<td>'N'</td>
<td>83.8</td>
</tr>
<tr>
<td>5</td>
<td>'A'</td>
<td>85.2</td>
<td>13</td>
<td>'U'</td>
<td>83.7</td>
</tr>
<tr>
<td>5</td>
<td>'B'</td>
<td>85.2</td>
<td>14</td>
<td>'Q'</td>
<td>83.5</td>
</tr>
<tr>
<td>6</td>
<td>'P'</td>
<td>85.0</td>
<td>15</td>
<td>'R'</td>
<td>83.2</td>
</tr>
<tr>
<td>7</td>
<td>'H'</td>
<td>84.7</td>
<td>16</td>
<td>'M'</td>
<td>82.8</td>
</tr>
<tr>
<td>8</td>
<td>'K'</td>
<td>84.6</td>
<td>16</td>
<td>'O'</td>
<td>82.8</td>
</tr>
<tr>
<td>9</td>
<td>'Z'</td>
<td>84.4</td>
<td>17</td>
<td>'W'</td>
<td>82.2</td>
</tr>
</tbody>
</table>
robustness

- **Sensitiveness** to changes in the reference speaker models.

### MLED—Changing Training Data Quantity

<table>
<thead>
<tr>
<th>Type</th>
<th>2 prod, 120 spkrs</th>
<th>2 prod, 60 spkrs</th>
<th>2 prod, 30 spkrs</th>
<th>1 prod, 120 spkrs</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLED.1</td>
<td>85.0</td>
<td>82.0</td>
<td>81.5</td>
<td>84.7</td>
</tr>
<tr>
<td>MLED.5</td>
<td>87.1</td>
<td>86.1</td>
<td>85.4</td>
<td>86.2</td>
</tr>
<tr>
<td>MLED.10</td>
<td>88.1</td>
<td>86.3</td>
<td>85.5</td>
<td>87.5</td>
</tr>
</tbody>
</table>

### Adaptive Training Experiments

<table>
<thead>
<tr>
<th>Type</th>
<th>Training</th>
<th>Full</th>
<th>balanced 17 letters</th>
<th>random (17-let. average)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLED.1</td>
<td>ML</td>
<td>85.0</td>
<td>81.8</td>
<td>84.3</td>
</tr>
<tr>
<td>MLED.1</td>
<td>adaptive</td>
<td>84.9</td>
<td>84.1</td>
<td>84.2</td>
</tr>
<tr>
<td>MLED.5</td>
<td>ML</td>
<td>87.1</td>
<td>81.0</td>
<td>85.6</td>
</tr>
<tr>
<td>MLED.5</td>
<td>adaptive</td>
<td>87.4</td>
<td>86.1</td>
<td>85.9</td>
</tr>
<tr>
<td>MLED.10</td>
<td>ML</td>
<td>88.1</td>
<td>81.0</td>
<td>85.9</td>
</tr>
<tr>
<td>MLED.10</td>
<td>adaptive</td>
<td>88.0</td>
<td>86.1</td>
<td>86.6</td>
</tr>
</tbody>
</table>
“it is by no means universally true.” – I. T. Jolliffe

- 1\textsuperscript{st} eigenvector: sex (strong)
- 2\textsuperscript{nd} eigenvector: amplitude (strong)
- 3\textsuperscript{rd} eigenvector: second-formant (maybe)
- 4\textsuperscript{th} eigenvector: changes in pitch (maybe)
Future work

- Extensions of the Eigenvoice Approach
  - Hybrid
    - such as MLED+MAP in Fig. 2. (Done)
    - How about allowing K to rise as the data increases?
    - MLED + MLLR
  - Discriminative training of the reference SD models;
  - Environment adaptation
  - LDA rather than PCA
  - Learning basis vectors by ML rather than PCA
  - Eigenvoice adaptation of state transition probabilities and Gaussian standard deviations
    - I don’t think so. There are definitely some correlations between states and Gaussian, though.
Future work

- Training models for Large-Vocabulary Systems
  - There will be insufficient data per reference speaker.
  - The computational and storage requirements of this naïve extension of small-vocabulary methodology would be onerous.

- Principals
  - Inter-speaker variability (in K-space)
  - Intra-speaker variability (by the Gaussians)
  - Efficient pooling of training data from different speakers
    - Remove speaker-dependent characteristics from training data at the beginning of training, rather than based on SI model.
Summary

- Fast way to speaker adaptation based on Eigenvoices.
- Maximum Likelihood Eigen-Decomposition
  - Better than MLLR and MAP.
- For large vocabulary data, MLED+MAP is best.