A Mixture of Gaussians Front End for Speech Recognition

Matthew Stuttle

28th January 2003

Cambridge University Engineering Department

Web: http://svr-www.eng.cam.ac.uk/~mns25
Email: mns25@eng.cam.ac.uk

SVR Speech Seminar Series
Overview

- The GMM speech frontend
  - Motivation
  - Implementation
- Performance of GMM features
  - Baseline results
  - Concatenated with MFCCs
  - Streaming systems
- Confidence metrics
- Noise compensation
- Speaker Adaptation
- Conclusions
The case for formants in LVCSR

Motivation for using formants:

- Considered representative of underlying phonetic content
- Potentially useful in noisy or band-limited environments
- Formant positions important for human speech recognition

Existing formant schemes:

- Analysis by synthesis
- Linear prediction analysis
- Dynamic template matching of hand-labelled spectra
Problems with formants

Problems with existing formant extraction schemes:

- Not always well defined in spectra, (e.g., fricatives or nasalised sounds)
- Amplitude information required to distinguish certain phone types (e.g., nasalised phones and voiced vowels)

Statistical peak representations:

- Gravity Centroids: extract first and second moments from spectral subbands
- HMM-2: fit a second frequency HMM to the spectrum at each frame, each frequency state corresponds to a spectral peak or region
The Gaussian Mixture Model for feature extraction

Gaussian mixture model:

• Fits a set of Gaussian mixtures to the smoothed magnitude spectra of a speech signal
• Characterises the spectra in terms of spectral peaks, hence the features are ‘formant-like’.
• Can represent general spectral envelope
• Statistical representation
• Is not band-limited as Gravity Centriods
Gaussian Mixture Model front end

- Speech Signal
- Window
- Spectra
- Spectral Smoothing
- GMM Fit to spectra using EM algorithm
- MFCC features
  - means $\mu(t)$
  - Standard Deviation $\sigma(t)$
  - mix weights $e(t)$
- Stream 1
- Stream 2
- Concatenative Feature Vector
- Multiple Stream System
- Single Stream System
Single coded frame

Example single frame plot from test utterance, before and after smoothing.
A Mixture of Gaussians Front End for Speech Recognition

GMM front end trajectory plot

- Utterance “Where were you while we were away?”
- Four Gaussian components fitted per frame

- Extracts close approximation to formant positions
- No spectral smoothing or frame to frame constraints
Experimental details

All experiments were performed on the Resource Management (RM) task

- 3990 training sentences with roughly a 1000 word vocabulary, 109 training speakers and 1200 test sentences from 40 subjects
- Cross-word triphone context-dependent HMMs were made using a phonetic decision class tree as per HTK RM Recipe
- A word-pair grammar was used for recognition
- Results were tuned on the 300 sentence ‘feb89’ subset of data
- Word Error Rate averages over all 4 test sets quoted
Baseline Resource Management results

<table>
<thead>
<tr>
<th>Description</th>
<th>Total Features</th>
<th>% WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFCC</td>
<td>39</td>
<td>4.19</td>
</tr>
<tr>
<td>PLP</td>
<td>39</td>
<td>3.89</td>
</tr>
<tr>
<td>4 Component GMM</td>
<td>39</td>
<td>6.10</td>
</tr>
<tr>
<td>6 Component GMM</td>
<td>57</td>
<td>4.90</td>
</tr>
</tbody>
</table>

• Best GMM features result was 17% worse than the MFCC baseline
• Fitting six mixtures (GMM6) to spectra yields better result than four
• Errors were distributed evenly across phone classes
**Resource Management results for hybrid systems**

Gaussian means were appended directly onto the MFCC feature vector

<table>
<thead>
<tr>
<th>Parameterisation</th>
<th>Total Features</th>
<th>% Err</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFCC ( {c_1 \cdots c_{12}} )</td>
<td>39</td>
<td>4.19</td>
</tr>
<tr>
<td>MFCC + ( {c_1 \cdots c_{16}} )</td>
<td>51</td>
<td>4.29</td>
</tr>
<tr>
<td>MFCC + 4 Formant frequencies from ESPS</td>
<td>51</td>
<td>4.89</td>
</tr>
<tr>
<td>MFCC + 4 Gravity Centroids</td>
<td>51</td>
<td>4.08</td>
</tr>
<tr>
<td>MFCC + 6 Gravity Centroids</td>
<td>57</td>
<td>5.02</td>
</tr>
<tr>
<td>MFCC + 4 GMM Means</td>
<td>51</td>
<td>4.08</td>
</tr>
<tr>
<td>MFCC + 6 GMM Means</td>
<td>57</td>
<td>3.96</td>
</tr>
</tbody>
</table>

- Appending the GMM means gave a WER decrease of 5.5% relative to MFCC baseline
- Adding four Gravity Centroids reduced the WER by 2%
- All other features appended degraded performance
A Mixture of Gaussians Front End for Speech Recognition

**Synchronous stream system**

- Input vector $\mathbf{y}$ divided into 2 streams $\{\mathbf{y}_{MFCC}, \mathbf{y}_{GMM}\}$
- Output probability given by

$$b_j(\mathbf{y}) = \prod_{s=1}^{S} \left[ \sum_{m=1}^{M} c_{jsm} \mathcal{N}(\mathbf{y}_s; \boldsymbol{\mu}_{jsm}, \Sigma_{jsm}) \right]^{\gamma_s}$$

- Where $\gamma_s$ is the stream weight of stream $s$.
- Stream weights were constrained to sum to one.
- Only MFCCs were used to obtain alignments in Baum Welch training.
Optimal performance was for GMM6 system at stream weight of 0.8, giving 3.7% WER, a relative improvement of 10.9%.

Streaming MFCC and PLP features gave little improvement.
Confidence in GMM Fit Metrics

- Peaks are less reliably defined in unvoiced or quiet regions
- Define confidence metric $\xi(t)$ based on amplitude and curvature

$$\xi(t) = \beta \left[ \prod_{n=1}^{N} \frac{e_n(t) + 10.53}{\sigma_n(t)} \right]^{\frac{1}{N}}$$

- Use standard synchronous stream system

$$b_j(y(t)) = \prod_{r=1}^{R} \left[ \sum_{m=1}^{M} c_{jrm} N(y_r(t); \mu_{jrm}, \Sigma_{jrm}) \right]^{\gamma_r(t)}$$

- Stream weights $\gamma_r(t)$ set by confidence metric

$$\gamma_1(t) = 1 - \xi(t) \quad \gamma_2(t) \propto \xi(t)$$
A Mixture of Gaussians Front End for Speech Recognition

**Example Confidence Metric**

- Clean and noise-corrupted plots shown
- $\xi(t)$ is high in regions with peak-structures
- Is low in regions with low energy or no peaks
A Mixture of Gaussians Front End for Speech Recognition

Experimental setup

WSJ task

- 284 training speakers, 65,000 word vocabulary, Hub 1 dev and eval
- Cross-word triphone context-dependent HMMs
- Trigram language model
- Cepstral Mean Normalisation used on feature vectors
Results on WSJ using confidence metric

<table>
<thead>
<tr>
<th>Description</th>
<th>% WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFCC</td>
<td>9.75</td>
</tr>
<tr>
<td>MFCC+6 Means Concatenative</td>
<td>9.56</td>
</tr>
<tr>
<td>MFCC+6 Means Fixed Stream Weights</td>
<td>9.64</td>
</tr>
<tr>
<td>MFCC+6 Means Confidence Metric</td>
<td>9.52</td>
</tr>
<tr>
<td>GMM6</td>
<td>12.43</td>
</tr>
<tr>
<td>GMM6 feature mean normalisation</td>
<td>12.02</td>
</tr>
</tbody>
</table>

- Small improvements over fixed stream weights
- No significant improvement over concatenative feature vectors by using confidence metrics on clean speech
GMM Features in Noise

- Peak representations of speech are inherently robust to some noise sources.

- Noise sources with strong peak structures (ie background babble) can corrupt features significantly.

- Unlike most peak representations, can reconstruct spectrum from GMM features.

- Can compensate for noise at feature extraction stage by estimating clean speech parameters given noise model.

- Alternatively can generate noise compensated model set given clean model set and noise model.
Front End Noise Compensation

- Compensate at feature extraction stage

- Assumes noise model $\hat{\theta}^{(n)} = \{\hat{\epsilon}^{(n)}, \hat{\mu}^{(n)}, \hat{\sigma}^{(n)}\}$

- Estimate clean speech feature parameters given noise model

$$l(x(t)|\theta(t), \hat{\theta}^{(n)}) =$$

$$\sum_{k=1}^{K} \ln \left( \sum_{q=1}^{Q} \hat{\epsilon}_q^{(n)} \mathcal{N} \left( x_k(t); \hat{\mu}_q^{(n)}, \hat{\sigma}_q^{(n)2} \right) + \sum_{n=1}^{N} e_n(t) \mathcal{N} \left( x_k(t); \mu_n(t), \sigma_n^2(t) \right) \right)$$
Model Compensation

- Adapts the static mean parameters of clean HMM model trained on GMM parameters

- Reconstructs spectra $x_{jm}$ from GMM parameters of each state $j$ and component $m$ in model

- Noise corrupted spectra is formed by adding spectra from noise spectrum $q$

- Parameters for noisy data $\hat{\theta}_{jm}$ are re-estimated

$$l(x_{jm} + q|\hat{\theta}_{jm}) = \sum_{k=1}^{K} \left( \ln \sum_{n=1}^{N} \hat{e}_{jmn} \mathcal{N} \left( x_{jmk} + q_k; \mu_{jmn}, \sigma^2_{jmn} \right) \right)$$
Additive Noise

- Noise source is Operations Room noise from the Noisex database
- Data corrupted by adding noise at waveform level
- Coloured noise disrupts peak structure severely
- Noise spectrum and corrupted spectrum shown
RM Results in additive noise - I

Results using

- **UC** Uncompensated clean speech models
- **MC** Mean compensated models
- **NM** Noise matched models

<table>
<thead>
<tr>
<th>18 dB SNR</th>
<th>UC</th>
<th>MC</th>
<th>NM</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFCC</td>
<td>32.3</td>
<td>14.0</td>
<td>8.1</td>
</tr>
<tr>
<td>MFCC+GMM Concat.</td>
<td>30.6</td>
<td>13.1</td>
<td>7.1</td>
</tr>
<tr>
<td>+ Confidence</td>
<td>29.6</td>
<td>12.6</td>
<td>7.1</td>
</tr>
</tbody>
</table>

- Adding GMM parameters to MFCCs gives improvements in noisy conditions
- Confidence metric yields small improvements for model compensated data
- Frontend compensation to the GMM parameters gave 28.3% WER
Adding GMM features to MFCCs gives small improvements over a range of SNRs.
Speaker adaptation

- GMM features are directly represented in spectrum - position of component means are frequency bin values

- Can implement a VTLN approach by scaling the component means

- CMN approach approximates VTLN for GMM system

- Diagonal feature transforms will scale features for VTLN and spectral tilt effects.
Speaker adaptation

- Obtained a constrained diagonal MLLR transform for WSJ speakers
- Regression fit to GMM means warpings yields VTLN factors correlated to MFCC Brent estimated ML search parameters.
**Unconstrained MLLR**

- Adapting the data using a speech/silence full MLLR transform

<table>
<thead>
<tr>
<th>Type of Transform</th>
<th>MFCC</th>
<th>MFCC + 6 Means</th>
<th>GMM6</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>9.75</td>
<td>9.56</td>
<td>12.0</td>
</tr>
<tr>
<td>UC MLLR</td>
<td>8.69</td>
<td>8.36</td>
<td>10.37</td>
</tr>
<tr>
<td>C MLLR</td>
<td>8.77</td>
<td>8.84</td>
<td>11.26</td>
</tr>
<tr>
<td>C MLLR + SAT</td>
<td>7.98</td>
<td>8.45</td>
<td>11.32</td>
</tr>
</tbody>
</table>

- 4% improvement incorporating GMM features with MFCCs and using UC MLLR
- Performance degrades when feature space transforms are used
- Systems using diagonal feature transforms did improve in CMLLR systems
Conclusions

- Fitting a GMM to speech provides features with information complementary to MFCC parameterisation.

- Incorporating GMM features with MFCCs by concatenating feature vectors reduces error rates on RM task.

- Combining MFCCs with GMM features using synchronous streams measure of confidence yields no significant improvement over concatenating into a single feature vector.


**Conclusions**

- Including GMM features with MFCCs gives improved performance in an additive noise environment.

- The static mean parameters of GMM features can be rapidly adapted to additive noise environments.

- Relative improvements incorporating GMM features with an MFCC parameterisation are maintained with a MLLR adaptation.

- GMM features are not suited to feature-space transforms and constrained MLLR approaches.