Speaker Diarization

M.Sc.
Nurit Spingarn
Supervised by Prof. Israel Cohen
January, 2015
Outline

- Introduction
- Related Work
- Background
- Short Utterances Speaker Diarization
- Results and Conclusion
Speaker Diarization is the process of partitioning an input audio stream into segments according to the speaker identity ("who spoke when?").
Motivation

Speaker diarization is required in many applications. For example:

- Audio indexing
  - Broadcast news
  - Telephone conversations
- Speaker verification and identification pre-processing tool
Problem Definition

- We assume that the recorded signal is comprised of clean speech, stationary and transient noises as follows:

\[ y(n) = x_{sp}(n) + x_{st}(n) + x_{tr}(n) \]

- The goal: To determine who spoke and when
  - Main difficulties:
    - Short speech utterances
    - Noisy environment
    - High number of involved speakers
Outline

- Introduction
- Related Work
- Background
- Short Utterances Speaker Diarization
- Results and Conclusion
Related Work

A typical speaker diarization system

Step 0  Step 1  Step 2
Related Work

• Benchmark Methods
  o The baseline system: Speaker change point detection followed by agglomerative clustering
  o Hao et al. 2012 - Dimensionality reduction approach for speaker diarization systems (PCA and LPP algorithm)
  o Ning et al. 2006 - Spectral clustering approach for speaker diarization system
Related Work

Baseline System

- Step 1: BIC-based speaker change point detector
  - Detects speaker change points within a window using a penalized likelihood ratio test (LRT)
  - Typical window length: 2-5 seconds
Baseline system

- Step 1: BIC-based speaker change point detector

\[ L_0 = \sum_{i=1}^{N_x} \log P(x_i|\theta_x) + \sum_{i=1}^{N_y} \log P(y_i|\theta_y) \]
\[ L_1 = \sum_{i=1}^{N_x} \log P(x_i|\theta_x) + \sum_{i=1}^{N_y} \log P(y_i|\theta_y) \]
\[ d = L_1 - L_0 - P\frac{\lambda}{2} \log N_z \]

\[ d = \Delta BIC = \frac{N_z}{2} \log |\Sigma| - \frac{N_x}{2} \log |\Sigma_x| - \frac{N_y}{2} \log |\Sigma_y| - P\frac{\lambda}{2} \log N_z \]

*Gopalakrishnn, 1998*
Baseline system

- Step 2: Agglomerative clustering algorithm
  - Initializing
  - Computing pair-wise distances between each cluster
  - Updating distances of remaining clusters
  - Iterating until a stopping criterion is met
Bottom-up algorithm
Speech Segment Example (two speakers)

Stop! You have two speakers.

Merge the two closest clusters

Merge the two closest clusters

Merge the two closest clusters
Related Work

Baseline System

• Change point detection
  o High miss rate of short utterances (2-5 sec)
  o Requires a detection error to be empirically tuned – tradeoff between pure segments and minimizing missing change points
  o Full search implementation is computationally expensive

• Directly affects on final diarization results.
Bottom-up algorithm
speech segment Ex.

Stop! You have two speakers.

Merge the two closest clusters

Merge the two closest clusters

Merge the two closest clusters

Related Work
Related Work

Baseline System

- **Main drawbacks**
  - High miss rate of short utterances
  - High dependency on penalized factor
  - Computationally expensive
  - Performance degradation in noisy environment
Outline

- Introduction
- Related Work
- Background
- Short Utterances Speaker Diarization
- Results and Conclusion
Graph Embedding

- Embedding algorithms are helpful for gaining insight into complex data sets described by high-dimensional features.
- Inherent in many data sets is a small set of natural parameters which capture the important sources of variation in the data.
- Hence, extracting these features reveals the underlying structure and can improve:
  - Data exploration
  - Visualization
  - Modeling and clustering
Graph Embedding

- These features describe a low-dimensional manifold which can be represented as a graph specifying which points on the manifold are neighbors.

- Motivation
  - Speech can lie on a low-dimensional manifold [1]
  - Suitable platform for GMM supervector
  - Successful methods in speaker recognition algorithms

Graph Notation

- Representing the data by similarity graph, $G = (V, E)$:
  - $V$ is a set of vertices – each vertex $v_i$ represents a data point
  - $E$ is a set of edges – each edge $e_{ij}$ between two vertices $v_i$ and $v_j$ carries a non-negative weight $W(i, j) \geq 0$ which is a measure of similarity between the corresponding points.

- Similarity matrix – a matrix whose $(i, j)$-th element equals to $W(i, j)$

- The degree of vertex $v_i$ is defined as

\[
d_i = \sum_j w_{ij}
\]
Spectral Clustering

- Constructing the normalized symmetric Laplacian matrix:
  \[ W = \exp \left( -\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \Rightarrow L = D^{-\frac{1}{2}}WD^{-\frac{1}{2}} \]
  where D is the degree matrix

- Compute the first k eigenvectors of \( L \) corresponding to the k largest eigenvalues:
  \[ U = (u_1, u_2, \ldots, u_k) \]

- Normalizing the matrix \( U \) as follows:
  \[ y_{ij} = u_{ij} / \left( \sum_k u_{ik}^2 \right)^{\frac{1}{2}} \]

- Applying K-means algorithm on matrix \( Y \)

Spectral Clustering

• Advantages:
  o Relies on analyzing the eigen-structure of an affinity matrix, rather than estimating some explicit model of data distribution
  o Underlying the structure of the data without assuming any type of structure in advance
  o Convenience algebra
  o Number of clusters can be estimated by the eigenvalues spectrum
Problem Definition

- We assume that the recorded signal is comprised of clean speech, stationary and transient noises as follows:

\[ y(n) = x_{sp}(n) + x_{st}(n) + x_{tr}(n) \]

- The goal: Determine who spoke and when.
- Focusing on:
  - Short utterances – rapid speaker change point
  - Speakers with similar voices
  - Noisy environment
Motivation

- Short utterances (2-5 seconds)
  - Hard to extract significant features for discrimination between speakers
  - High probability for missing speaker change point
- Speakers with similar voices (axon)
  - Difficult to discriminate
- Noisy environment degrades the system performance (segmentation, clustering, etc.)
Proposed Method

- Algorithm stages:
  - Segmentation (using VAD)
  - Feature vectors extraction
  - Utterance representation
  - Spectral clustering

- Assumptions:
  - Unknown number of speakers
  - Non overlapping speech segments
  - Short conversations (6-10 minutes)
Segmentation

- Using VAD for fine segmentation
  - Very short utterances are part of the current speaker
Feature Extraction

- Three categories:
  - High-level
  - Prosodic and Spectro-temporal features
  - Short-term spectral and voice source features

Learned

High-level
Phones, idiolect (personal lexicon), accent

Prosodic and Spectro-temporal
Pitch, energy, temporal features

Physiological

Short-term spectral and voice source
Spectrum, glottal, pulse features

Feature Extraction

- We chose the Mel Frequency Cepstral Coefficients (MFCCs) and its first and second derivatives and combine them as follows:

\[
Y(\cdot, t) = \begin{pmatrix}
Y_m(\cdot, t) \\
\Delta Y_m(\cdot, t) \\
\Delta \Delta Y_m(\cdot, t)
\end{pmatrix}
\]

where \( Y_m(\cdot, t) \) is the absolute value of MFCCs in frame \( t \).
Utterance Representation

- Reminder: GMM Mean Supervector
  - Adapting a target GMM
  - Concatenating the mean components
- We suggest to train the UBM on the tested conversation
  - Linguistic dependency
  - Relevant to the current conversation (noisy environment)
  - Less components
  - Easier MAP adaptation
Utterance Representation

The UBM
\((N\text{ components})\)

MAP Adaptation

Feature Extraction

Segment \(S_i\)

\[ SV_i = \begin{pmatrix} m_1 \\ m_2 \\ \vdots \\ m_N \end{pmatrix} \]

GMM Supervector of segment \(S_i\)

VAD

Feature Extraction

Utterances Representation

Spectral Clustering
Utterance Representation

- Why GMM mean Supervectors?
  - Mapping between an utterance and a high dimensional vector, fits well with the idea of spectral clustering (or any other dimensionality reduction approach)
  - Represents local first-order differences between the UBM and the adapted GMM
  - Was proven as successfully method for speaker verification and speaker clustering tasks
Utterance Representation

• The changes of GMM mean supervectors over time emphasize the difference between consecutive utterances.

• Therefore, we suggest the following utterance representation:

\[
SV = \begin{pmatrix}
SV^m \\
\Delta SV^m \\
\Delta \Delta SV^m
\end{pmatrix}
\]

\[
\Delta SV^i_m = SV^i_m - SV^{i-1}_m
\]

\[
\Delta \Delta SV^i_m = \Delta SV^i_m - \Delta SV^{i-1}_m
\]

where \(SV^m\) is the GMM mean supervector.
Graph Embedding

• Embedding algorithms are helpful for gaining insight into complex data sets described by high-dimensional features.

• Assumption*: Speech can lie on a low-dimensional manifold.

• Therefore, there may be relatively fewer degrees of freedom in the underlying systems that generate this data.

• Extracting and reveling the underlying structure can improve: Data exploration, visualization and clustering.

* A. Jansen and P. Niyog
Spectral Clustering

- Weight matrix computation (Gaussian Kernel)
  \[ W = \exp\left(\frac{-d_{ij}^2}{2\sigma^2}\right) \]

- Option 1: Cosine metric (Proposed supervectors)
  \[ d_{ij}(SV_i, SV_j) = 1 - \frac{SV_i^T SV_j}{\sqrt{SV_i^T SV_i} \sqrt{SV_j^T SV_j}} \]
  - The Cosine metric takes into consideration the angle between vectors and neglects the magnitude.
  - Because speaker information is not part of the magnitude, the cosine metric is a suitable choice.
Spectral Clustering

• Option 2: KL divergence (GMM mean supervectors)

\[ d_{ij}(SV_m^i, SV_m^j) = \frac{1}{2} \sum_{l=1}^{N} \lambda_i (m_i^l - m_j^l)^T \Sigma_l^{-1} (m_i^l - m_j^l) \]

  o Natural and successful distance between two probability functions

• Final clustering obtained by using k-means algorithm
Spectral Clustering

- Number of speakers
  - There is a strong connection between number of clusters and the eigenvalues of the normalized Laplacian matrix using the GAP between consecutive eigenvalues
Block Diagram

Utterances

Feature Extraction

Mean MAP Adaptation

Supervectors Creation

Spectral Clustering

Speaker Clusters

UBM
Database

- Speech conversations with rapid speaker change points (short utterances of 2-5 seconds) based on:
  - TIMIT – similar axon
  - FESTVOX
- Real conversation records (cellphone or computer recording)
- Number of speakers: 2-6
- Additive noise signals:
  - Additive AWGN with different noise levels: 0dB, 5dB, 10dB, 20dB.
  - Additive transient interruption: Typing, door knocks and metronome.
Results and Conclusions

Scatter Plot - Four speakers
PCA → LPP → Spectral clustering

- 240 utterances
- 10dB SNR
- 2-5 seconds each utterance
Results and Conclusions

Estimating Number of Speakers

- Spectral clustering method
- Penalized BIC method

Error (%) vs. Number of Speakers
Performances Evaluation Measurements

- Diarization Error Rate (DER)

\[
DER = \frac{T_{FA} + T_{Miss} + T_{Confusion}}{T_{Ref}}
\]

- Average Cluster Purity (ACP)

\[
ACP = \frac{1}{N} \sum_{i=1}^{S} p_i n_i
\]

\[
p_i = \sum_{j=1}^{R} \left( \frac{n_{ij}}{n_i} \right)^2
\]

- R – number of speakers
- S – number of clusters
- \( n_{ij} \) – total number of utterances in cluster \( i \) spoken by speaker \( j \).
Discussion
System Comparison

- **PCA**
  - Set of mutually orthogonal basis functions that capture the directions of maximum variance in the data.
  - Sensitive to outliers

- **LPP**
  - Linear dimensionality reduction algorithm
  - Preserves the local neighborhood structure of the data set.
  - Seeks a linear approximation of nonlinear Laplacian Eigen-maps

- **Spectral clustering**
  - Optimally preserves the local neighborhood structure of the data set.
  - Does not limited to LINEAR constraints
Experiments
DER VS SNR
Experiments
ACP VS SNR

Results and Conclusions
Experiments
DER VS Number of speakers

Results and Conclusions
## Experiments
DER at different transient noise environments + Babble noise

<table>
<thead>
<tr>
<th>Method</th>
<th>Door Knock</th>
<th>Keyboard stroke</th>
<th>Metronome</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>69.92%</td>
<td>71.23%</td>
<td>69.9%</td>
</tr>
<tr>
<td>PCA (Cosine)</td>
<td>57.6%</td>
<td>56.2%</td>
<td>58.1%</td>
</tr>
<tr>
<td>LPP</td>
<td>42.6%</td>
<td>43.9%</td>
<td>42.2%</td>
</tr>
<tr>
<td>Proposed (Cosine)</td>
<td>18.6%</td>
<td>19.3%</td>
<td>19.6%</td>
</tr>
</tbody>
</table>
Proposed VAD

- Learning a likelihood ratio function using spectral clustering concept.
- Based on unique kernel and features.
- Function extension using Laplacian pyramid algorithm.
- Outperform significantly conventional VADs under challenging noisy environments (especially transient noise).
Ning et al. Spectral Clustering Approach for speaker diarization

• Highlights:
  • Segmentation using BIC-based method
  • GMM construction using MFCC only
  • Spectral clustering:
    • Symmetric KL distance approximated by the unscented transformation. Based on:
      • 2d “sigma” points.
      • High dependency on chosen “sigma” points.
  • Different scaling factor
Results and Conclusions

Results

spectral clustering approaches

![Graph showing DER (%) vs SNR (dB)]
LLE

- Assuming well-sampled manifold.
- Expect each data point to lie on or close to locally linear patch.
- Characterize the local geometry in the neighborhood of each data point by linear coefficient that reconstruct the data point from its neighbors.
- Based on locally linear reconstruction errors.
- “Think globally fit locally”.
- Linear approximation for the LLE: The NPE algorithm.
Comparison to LLE and NPE methods

- LPP and spectral clustering represent in some sense, the following objective criterion:

\[ \sum_{i,j} (y_i - y_j)^2 w_{ij} \]

- LLE and NPE are both aims to minimize the locally linear reconstruction errors:

\[ \sum_i \left| x_i - \sum_j w_{ij} x_j \right|^2 \]

\[ \sum_i \left| y_i - \sum_j w_{ij} y_j \right|^2 \]
Comparison to other spectral methods

<table>
<thead>
<tr>
<th>Method</th>
<th>ACP</th>
</tr>
</thead>
<tbody>
<tr>
<td>LPP</td>
<td>72%</td>
</tr>
<tr>
<td>NPE</td>
<td>73.3%</td>
</tr>
<tr>
<td>LLE</td>
<td>79.2%</td>
</tr>
<tr>
<td>Spectral Clustering</td>
<td>86%</td>
</tr>
</tbody>
</table>

- Degradation in performance when using linear approximations method.
- LLE and NPE: Assuming highly sampled data and relevant low-dimension.
Results Summary

- The spectral clustering approach outperforms all compared methods.
- Local structure must be preserved and Non-linear manifold embedding is required
- The cosine metric performances outperform the KL divergence
  - Neglecting the magnitude
  - Using the first and second derivatives
- Linearity is not always a desired property
Summary

- We focused on short utterance diarization under noisy environment.
- We developed a unique VAD for fine segmentation and reducing of noise segments.
- We represented each utterance by a supervector:
  - UBM training
  - Derivatives
- We used spectral clustering in order to find the most informative and discriminative features in low dimensional space.
Main Contributions

Applicative:

- Short speech utterances diarization (rapid speaker changes)
- High robustness to noise
- Estimating number of speakers
- Handling with speakers characterized by similar voices
Main Contributions

Tools

- Graph embedding approaches
- GMM mean supervectors and its first and second derivatives
- Training the UBM on the tested conversation
- Combining cosine metric within the Gaussian kernel
- Fine segmentation (VAD)
Publications

Submitted

Published
Future Work

- Speaker overlap problem
- Audio-Visual speaker diarization
- Investigation of reverberation effects
- Combining at speaker verification systems
- Improving the suggested Algorithm:
  - Better features
  - Different weight matrix or distance metric
  - Iterative algorithm for handling with high number of involved speakers
Thank you!