Speech Recording based Language Recognition

Leopold Cambier, Cindy Orozco Bohorquez, Matan Leibovich
lcambier@stanford.edu, orozcocc@stanford.edu, matanie@stanford.edu
Institute of Computational and Mathematical Engineering

**Overview**

- **Motivation**: Construct a real-time language classifier for communication purposes.
- **Method**: Construct ML estimator based on Gaussian Mixture Model (GMM) density estimations.
- **Results**: Classification error decreases with model complexity to a certain limit. Optimal number of gaussians varies significantly across languages. There is clustering pattern in the classification.

**Data**

Source: https://community.topcoder.com/longcontest/?module=ViewProblemStatement&rd=16555&pm=13978

- \( \mathcal{L} \): 176 different languages (some very exotic!)
- 376 10 sec. samples for each language
- Each sample divided into 210 ms long segments

**Feature Extraction: Shifted Delta Cepstral (SDC)**

- Hamming Window: \( x_{\text{Ham}}(t) = x(t) \text{ Ham}(t) \)
- Fourier Transform: \( \tilde{x}(f) = \mathcal{F}(x_{\text{Ham}}(t)) \)
- Transition to Mel Scale: \( \tilde{x}_{\text{Mel}}(f) = \tilde{x}(M(f)) \)

- 7-vectors cepstral coefficient for each 10 ms segment
- Use 3 adjacent cepstral coefficients to construct a 7-vector SDC coefficient
- For each 210ms sample ⇒ 49 SDC coefficients
- \( 46 \times 376 = 17206 \) data samples \( x_{\text{i}}^{(i)} \)

**Gaussian Mixture Model**

**Density Estimation**

- Use \( x_{\text{i}}^{(i)} \) to estimate mixture of \( N \) Gaussian densities
  \[
  p_{\ell}(x) = \sum_{i=1}^{N} w_i \mathcal{N}(x; \mu_{\ell,i}, \Sigma_{\ell,i}), \quad x \in \mathbb{R}^4
  \]
  \[
  P(x; \mu_{\ell,i}, \Sigma_{\ell,i}) = \frac{1}{\sqrt{(2\pi)^{4} | \Sigma_{\ell,i} |}} e^{-\frac{1}{2} (x-\mu_{\ell,i})^T \Sigma_{\ell,i}^{-1} (x-\mu_{\ell,i})}
  \]
  \( \mu_{\ell,i}, \Sigma_{\ell,i} \in \mathbb{R}^{4 \times 4} \)

- \( \Sigma_{\ell,i} \): either full or diag

- Training using Python’s sklearn toolkit.

**Classification**

- Given 10s sample, construct \( x_{\text{i}}^{(i)}, \quad i = 1, \ldots, 46 \)
- SDC vectors for each 210 ms segment.
- MLE estimator

\[
  \hat{\ell} = \arg \max_{\ell} \prod_{i=1}^{46} p_{\ell}(x_{\text{i}}^{(i)})
  \]

**Results**

- Model trained with full and diagonal covariances
- Optimal for full covariance at \( N = 10 \)

- Optimal \( N \) actually varies significantly from one language to another

**Future Work**

- Allow variability in \( N \) per language for model fitting
- Introduce a measure for the quality of the GMM fit
- Train the algorithm on a larger dataset
- Predict how well would the classifier fit the sample (confidence intervals)

**Bibliography**


---

**Discussion**

- Test error of 0.34 with \( N = 10 \) and full covariances, on a dataset with 176 (balanced) classes
- Increasing \( N \) (to some extent) decreases the validation error, which shows some consistency between the model and the data
- Relatively high variability in the accuracy depending on the language

---

**Construction of Cepstral Coefficients**

- DCT of logarithm: \( x_{\text{Cep}}(\ell) = \text{DCT}( \log \tilde{x}_{\text{Mel}}(f_{\ell}) ) \)
- SDC: \( \tilde{x}_{\text{SDC}}(\ell) = \tilde{x}_{\text{Cep}}(\ell) - \tilde{x}_{\text{M}}(\ell) \)

**Confusion Matrix Graph with Highlighted Communities**

**Distribution of Optimal \( N \) over Languages**

Error is non uniformly distributed but tends to cluster
Analysis of confusion matrix graph shows communities