# Voice Processing

Marchioro Thomas

Project 4 – Speaker Identification

The goal of this project is designing an algorithm to identify a speaker from the Mel-Frequency Cepstrum Coefficients (MFCCs) among 40 possible speakers, whose recordings come from the TIMIT dataset. The coefficients extraction is done frame-by-frame employing a window of 20 ms and a shift of 5 ms, and for each frame $d = 25$ MFCCs are extracted. The prediction is based on the MAP criterion, assuming that the data are distributed as a Gaussian Mixture, i.e., the probability of producing a certain vector of coefficients $x$ by the speaker $k$ is given by

$$\Pr(x|q_k) = p_k \frac{1}{\sqrt{2\pi}^d \sqrt{|\Sigma_k|}} \exp\{-(x-\mu_k)^\top \Sigma_k^{-1}(x-\mu_k)\},$$

where $p_k$ is the a priori probability of the $k$-th speaker and $(\mu_k, \Sigma_k)$ are the mean and covariance of the $k$-th Gaussian of the mixture.

## 1 Unsupervised Speaker Identification

I first applied two clustering algorithm, namely, Viterbi EM and Kmeans, to build a Gaussian Mixture Model (GMM) without using any prior knowledge on the data (i.e., assuming that the identity of the speaker of the training data is unknown).

### 1.1 Viterbi EM algorithm

Viterbi EM algorithm initializes $K$ clusters at random and iteratively assigns the examples to the most likely cluster according to Bayes decision rule, which can be approximated by choosing the class $q_k$ maximizing

$$\log \Pr(q_k|x, \Theta^{(t)}) \simeq \log \Pr(x|q_k, \Theta^{(t)}) + \Pr(q_k|\Theta^{(t)})$$

where $\Theta^{(t)}$ is the GMM at iteration $t$. The parameters of the model $\Theta^{(t+1)}$ are updated at each iteration computing mean, covariance and a priori probability of each cluster, until convergence – or the maximum number od iterations – is reached.

I tested the algorithm on 2D synthetic data distributed according to a GMM with $K = 3$ Gaussians. The algorithm terminated after few iterations, separating the generated examples correctly, as can be seen from Figure 1. Nonetheless, worse results are obtained when the mean values of the Gaussians are close to each other and covariance is large (in terms of determinant), Viterbi EM algorithm often requires to be run multiple times before finding a reasonable partition. Moreover, being my implementation of the algorithm very slow, it is probably not the best solution for this application.

### 1.2 Kmeans

Since in this specific case the number of available examples is roughly the same for all the speakers, we can approximate the a priori probabilities with a uniform distribution, where $p_k = 1/K, \forall k = 1, \ldots, K$. Hence, the Maximum Likelihood criterion can be applied instead of Bayes decision rule

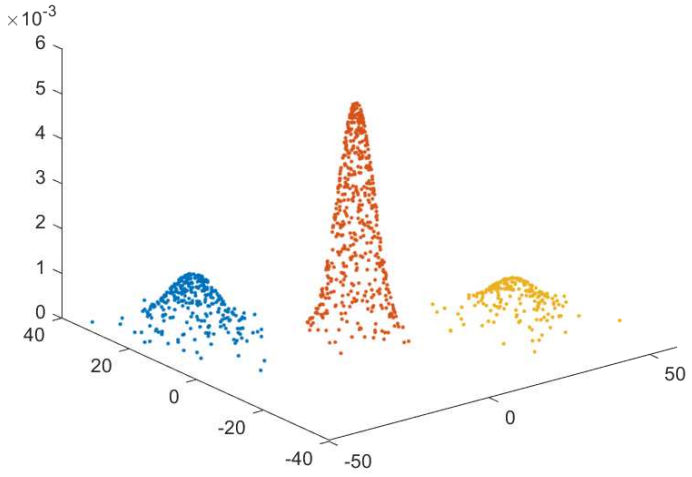


Figure 1: Execution of Viterbi EM algorithm on 2D synthetic data.

(which is based on MAP criterion). Moreover, since we assumed that data follow a GMM, if we also assume independent components, i.e.,

$$\Sigma_k = \mathrm{diag}(\sigma)^2 = \begin{bmatrix} \sigma_1^{(k)} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_d^{(k)} \end{bmatrix}^2 \tag{1}$$

and similar variance for all the Gaussians, it holds

$$\begin{aligned} \arg\max_{q_k} \Pr(q_k|x,\Theta) &= \arg\max_{q_k} \frac{1}{\sqrt{2\pi}^d \sqrt{|\Sigma_k|}} \exp\{-(x-\mu_k)^\top \Sigma_k^{-1}(x-\mu_k)\} && (2)\\ &\simeq \arg\max_{q_k} \exp\{-(x-\mu_k)^\top \Sigma_k^{-1}(x-\mu_k)\} && (3)\\ &= \arg\max_{q_k} -(x-\mu_k)^\top (x-\mu_k) && (4)\\ &= \arg\min_{q_k} \|x-\mu_k\|^2, && (5) \end{aligned}$$

and hence the Maximum Likelihood criterion can be approximated by a minimum distance criterion. This implies that Kmeans can be applied to build the GMM, since the principle is the same as Viterbi EM, but with Euclidean distance criterion instead of Bayes decision rule.

## 2 Supervised Speaker Identification

The aforementioned clustering algorithms construct a GMM in an unsupervised manner, without using any prior knowledge on the class of the training examples. Nevertheless, we actually have prior knowledge on the training data, since we know from which speaker the MFCCs are extracted. Therefore, the best GMM that can be built with such information is given by the unbiased estimators of the means obtained by averaging the training data

$$\hat{\mu}_k = \frac{1}{N_k} \sum_{x_i \in q_k} x_i$$

| Method | Training acc. | Test acc. |
|---|---|---|
| Viterbi EM | 0.3 | 0.375 |
| Kmeans | 0.45 | 0.5 |
| Supervised estimators | 1 | 0.975 |

Table 1: Accuracy of the GMM models obtained from the different algorithms.

and from the corresponding (biased) estimators of the covariance matrices

$$\hat{\Sigma}_k = \frac{1}{N_k} \sum_{x_i \in q_k} (x_i - \hat{\mu}_k)^\top (x_i - \hat{\mu}_k) .$$

# 3 Results

In the prediction phase, each frame is classified separately and the final predicted speaker for a whole track is the class with highest number of occurrences. The accuracy of the GMM models is computed as the ratio between the number of difference speakers predicted and the actual number of different speakers. The results on both training and test data are reported in Table 1, and clearly the supervised estimators provide the best performance. Also, one can infer that there is a problem of "underfitting" for the clustering algorithms, since the accuracy on the training data is even lower than the accuracy on test.