DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING, IIT MADRAS

CS5691: Project - Speaker Diarization
Team 1

Sheth Dev Yashpal
CS17B106

Harshit Kedia
CS17B103

1 K-MEANS BASED SPEAKER DIARIZATION

K-Means with segment size 0.5 sec on dev data		
Metric	Time	Percentage of scored speech
SCORED SPEAKER TIME	3001.86	100.0
SPEAKER ERROR TIME	1710.11	57.0
OVERALL SPEAKER DIARIZATION ERROR = **58.04%**		

Table 1.1: Model Accuracy with K-Means Clustering and segment size of 0.5 seconds with overlapping data.

K-Means with segment size 1 sec on dev data		
Metric	Time	Percentage of scored speech
SCORED SPEAKER TIME	3001.86	100.0
SPEAKER ERROR TIME	1529.87	51.0
OVERALL SPEAKER DIARIZATION ERROR = **54.65%**		

Table 1.2: Model Accuracy with K-Means Clustering and segment size of 1 seconds with overlapping data.

K-Means with segment size 0.5 sec on dev data		
Metric	Time	Percentage of scored speech
SCORED SPEAKER TIME	3001.86	100.0
SPEAKER ERROR TIME	1723.28	57.4
OVERALL SPEAKER DIARIZATION ERROR = **59.72%**		

Table 1.3: Model Accuracy with K-Means Clustering and segment size of 0.5 seconds without overlapping data.

K-Means with segment size 1 sec on dev data		
Metric	Time	Percentage of scored speech
SCORED SPEAKER TIME	3001.86	100.0
SPEAKER ERROR TIME	1508.97	50.3
OVERALL SPEAKER DIARIZATION ERROR = **58.34%**		

Table 1.4: Model Accuracy with K-Means Clustering and segment size of 1 seconds without overlapping data.

K-Means with segment size 1 sec on eval data		
Metric	Time	Percentage of scored speech
SCORED SPEAKER TIME	2397.76	100.0
SPEAKER ERROR TIME	1389.35	57.9
OVERALL SPEAKER DIARIZATION ERROR = **63.40%**		

Table 1.5: Model Accuracy with K-Means Clustering and segment size of 1 seconds with overlapping data.

K-Means with segment size 1 sec on eval data		
Metric	Time	Percentage of scored speech
SCORED SPEAKER TIME	2397.76	100.0
SPEAKER ERROR TIME	1239.89	51.7
OVERALL SPEAKER DIARIZATION ERROR = **64.46%**		

Table 1.6: Model Accuracy with K-Means Clustering and segment size of 1 seconds without overlapping data.

- In K-Means, we first did the clustering for individual feature vectors however it gave completely random speaker transition even at switches of $10ms$. So we used windows of $0.5s$ and $1s$. However, the individual feature vector model, though not shown here, gave much better error rates - **41.20%** for both overlapping and non-overlapping data.
- Our overlapping data models always perform better than the ones where we ignored the overlap of data. Since, the overlapped vectors can support all the clusters of data which are clashing, using that we obtain better clustering and hence we get slightly better results.

2 Agglomerative Hierarchical Clustering based Speaker Diarization

2.1 Merger based on L2-norm of means

Agglomerative clustering with segment size 1 sec on dev data		
Metric	Time	Percentage of scored speech
SCORED SPEAKER TIME	3001.86	100.0
SPEAKER ERROR TIME	1127.06	37.5
OVERALL SPEAKER DIARIZATION ERROR = **41.23%**		

Table 2.1: Model Accuracy with Agglomerative (Bottom-up) Clustering and segment size of 1 seconds with overlapping data.

Agglomerative clustering with segment size 1 sec on dev data		
Metric	Time	Percentage of scored speech
SCORED SPEAKER TIME	3001.86	100.0
SPEAKER ERROR TIME	1165.40	38.8
OVERALL SPEAKER DIARIZATION ERROR = **46.90%**		

Table 2.2: Model Accuracy with Agglomerative (Bottom-up) Clustering and segment size of 1 seconds without overlapping data.

- In Agglomerative (Bottom-up) Hierarchical Clustering, we use time slice of $1s$ and assign a unique speaker to the entire segment. We calculate the mean of all the feature vectors of the segment and merge based on the Euclidean distance (L2-norm) between the means. We directly put the time slot as beginning + offset of $1s$. Due to this we get Missed and False Alarm Speaker Time more than normally what is expected. Hence our Speaker Error Time and Overall Error differ slightly.

Agglomerative clustering with segment size 1 sec on eval data		
Metric	Time	Percentage of scored speech
SCORED SPEAKER TIME	2397.76	100.0
SPEAKER ERROR TIME	1047.81	43.7
OVERALL SPEAKER DIARIZATION ERROR = **49.16%**		

Table 2.3: Model Accuracy with Agglomerative (Bottom-up) Clustering and segment size of 1 seconds with overlapping data.

Agglomerative clustering with segment size 1 sec on eval data		
Metric	Time	Percentage of scored speech
SCORED SPEAKER TIME	2397.76	100.0
SPEAKER ERROR TIME	943.48	39.3
OVERALL SPEAKER DIARIZATION ERROR = **52.10%**		

Table 2.4: Model Accuracy with Agglomerative (Bottom-up) Clustering and segment size of 1 seconds without overlapping data.

2.2 Merger based on KL-divergence of Gaussians

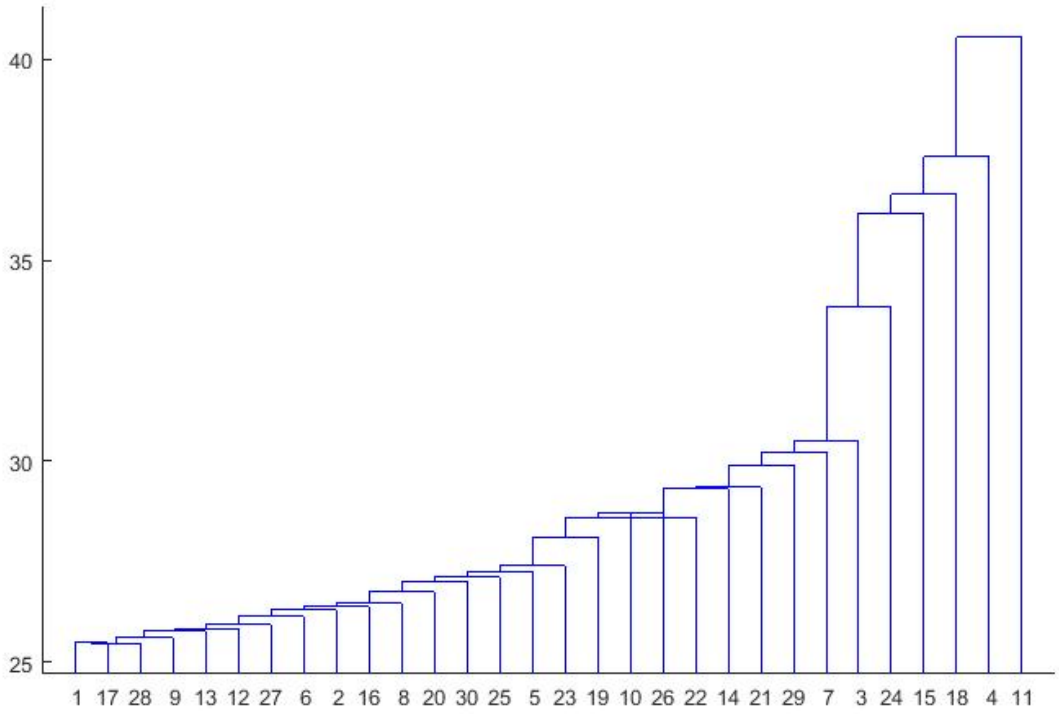

Figure 2.1: Dendrogram for hierarchical clustering based on KL-divergence of the Gaussians.

- The performance of this model is absolutely random. For every segment of 1s, we first calculate the mean and variance of the segment's feature vectors. Then, we merge the segments by using the KL-divergence of the two segments (average of both the values as it is asymmetric). When we cluster the data using this criterion, what we observe is that each point, individual feature vectors go and attach themselves to the clusters and hence finally we get $K-1$ cluster with one vector each and one cluster with the rest of the data. And hence there is no point in testing this model since there will be only one speaker predicted. You can refer to the attached dendrogram to better understand the problem.

Agglomerative clustering using GMM with segment size 4 sec on dev data		
Metric	Time	Percentage of scored speech
SCORED SPEAKER TIME	3001.86	100.0
SPEAKER ERROR TIME	1636.72	54.5
OVERALL SPEAKER DIARIZATION ERROR = **62.64%**		

Table 3.1: Model Accuracy wit GMM based Agglomerative Clustering and segment size of 4 secs with overlapping data.

Agglomerative clustering using GMM with segment size 4 sec on eval data		
Metric	Time	Percentage of scored speech
SCORED SPEAKER TIME	2397.76	100.0
SPEAKER ERROR TIME	1237.84	51.6
OVERALL SPEAKER DIARIZATION ERROR = **60.99%**		

Table 3.2: Model Accuracy wit GMM based Agglomerative Clustering and segment size of 4 secs with overlapping data.

Agglomerative clustering using GMM with segment size 0.1 sec on eval data		
Metric	Time	Percentage of scored speech
SCORED SPEAKER TIME	2397.76	100.0
SPEAKER ERROR TIME	1321.34	55.1
OVERALL SPEAKER DIARIZATION ERROR = **55.11%**		

Table 3.3: Model Accuracy wit GMM based Agglomerative Clustering and segment size of 4 secs with overlapping data, but allocation of of speakers done in intervals of 0.1 seconds within the 4 second segment.

- Here we fit a Gaussian Mixture Model into each of the segments of data. In the *EM* procedure, to avoid flooring covariance, we use only a *single, shared, diagonal covariance matrix*, and increase the segment size to 4 secs, giving us 400 feature vectors to start with.
- Now to merge these GMM's during bottom-up clustering, we compute the BIC (Bayesian Information Criterion) value for the GMM freshly generated for pooled data. We cannot just directly combine two different GMMs, therefore we train fresh GMMs pairwise for each segment with every other segment and then merge based on the minimum BIC obtained. The BIC for a model is given by the equation - $\ln(n)k - 2\ln(\hat{L})$; where $\hat{L}$ is the maximized likelihood value, k is the number of parameters to be estimated in the model and n is the number of input data points.
- Since the segment window is huge and there may be non-continuous data present in a single segment, we get some considerable amount of Missed Speaker Time and False Alarm Speaker Time if we directly output timestamp of starting feature vector, followed by offset of 4 secs on the rttm file. To avoid this, we break down the same 4 second interval into intervals of 0.1 seconds and then print the starting timestamp of these smaller segments and offset in the rttm file, using the same allocation of speaker for the whole segment, helping us to reduce the Overall Speaker Diarization Error.
- Here, we have neither used full-covariance matrix nor individual matrices for clusters within each segment which are both drawbacks. Also, none of our models are capturing the sequential pattern as in Hidden Markov Models. Possible fix to this is to build an ergodic HMM for the segments as they are generated with number of states as the number of sequences and then merge the segments according to BIC. After each merger, we can do a Viterbi Realignment of the data to get the appropriate segment to speaker mapping and adjust segment boundaries.