

Speaker Clustering Using the Bayesian Information Criterion: CS281A Project

Kofi Boakye

December 2, 2003

Abstract

Clustering is a commonly used statistical method for analyzing various types of data. A number of algorithms for clustering exist, but a problem that arises for these algorithms is the determination of the number of clusters, either in the form of an initialization parameter or a stopping criterion. This paper presents an algorithm for clustering speakers based on the Bayesian Information Criterion (BIC) as a stopping criterion for agglomerative clustering. The data is that of segments containing speech feature vectors of audio obtained from the International Computer Science Institute (ICSI) Meeting Corpus. Results from select data sets suggest that the algorithm lacks robustness to outliers and overlap of speakers in the feature vector space, and as such falls short of an automatic general-purpose method of speaker clustering.

1 Introduction

In many types of real-world data, it is often the case that the data consists of underlying subpopulations, each of these subpopulations having some common characteristics (e.g., a particular generating process). It then becomes of interest whether, given some unlabelled data, one can infer these subpopulations. This is the problem of clustering, and a number of algorithms exist—statistical and otherwise—to provide solutions. The most common of these are K-means [1], Gaussian mixture models, as well as agglomerative (bottom-up) and divisive (top-down) hierarchical clustering.

The K-means algorithm produces K clusters by alternating between determining cluster centroids and assigning points to clusters based on the

nearest centroid. Here “nearest” can refer to any distance metric, but in practice a Euclidean distance is almost always used. In Gaussian mixture modelling the data is modelled as having been generated according to a distribution that is a weighted combination of some N Gaussians. That is, for a given data point x ,

$$p(x|\theta) = \sum_{i=1}^N \pi_i \mathcal{N}(x|\mu_i, \Sigma_i) \quad (1)$$

where $\mathcal{N}(x|\mu_i, \Sigma_i)$ is a normal distribution with mean μ_i and covariance matrix Σ_i , and π_i is a mixture weight constrained by $\sum_i \pi_i = 1$. The EM algorithm is run to produce model parameters (i.e., means, covariances, and mixture weights) which maximize the likelihood of the data and the assignment of a data point to a particular Gaussian is probabilistic. To produce concrete clusters one can make a hard assignment by associating a data point with the Gaussian that was most likely to have generated it. Hierarchical clustering proceeds by either merging (agglomerative) or splitting (divisive) clusters in a repeated fashion until some criterion is satisfied, enabling the process to stop. In agglomerative clustering, one starts with N clusters (usually one for each data point) and merges the two “nearest” clusters according to one of many distance metrics. In divisive clustering, all of the data initially belongs to one cluster and is split into two according to one of a number of non-hierarchical clustering algorithms, and the procedure continues recursively until the stopping criterion is satisfied.

From these descriptions, one sees that an issue that arises for all of these algorithms is the determination of the number of clusters. In the cases of K-means and mixture models, it is an initialization parameter, and in hierarchical clustering it is related to the stopping criterion. Often it is the case that this number is arrived at heuristically which, from the standpoint of statistical theory, is quite unappealing. This also prevents the clustering algorithm from proceeding in an automatic fashion.

This paper describes a project which utilizes a principled approach to clustering that obviates the need to determine the number of clusters in advance, namely the Bayesian Information Criterion. Here the data of interest is human speech from the International Computer Science Institute (ICSI) Meeting Corpus. Speech segments, which consist of a collection of speech feature vectors, are to be clustered with the expectation that each final cluster will contain all of the segments from a given speaker and the final number of clusters will equal the total number of speakers.

The paper is organized as follows: Section 2 describes the Bayesian In-

formation Criterion and its applications to clustering; section 3 describes the speech data used in the clustering and the processing used to obtain it; in section 4 design and implementation of the algorithm are covered; results are given in section 5 and section 6 covers conclusions as well as possible future avenues of exploration.

2 The Bayesian Information Criterion

2.1 Motivations

The Bayesian Information Criterion (BIC) arises in statistics literature as a method for model selection [2]. Given a set of data $\mathcal{X} = \{x_i : i = 1, \dots, N\}$, we seek to choose from a set of parametric models $\mathcal{M} = \{M_i : i = 1, \dots, K\}$ a model that maximizes the likelihood of the data. Usually these models vary in the number of parameters and it is the model with the most parameters that will yield the highest likelihood. This is not always desirable, however, because selecting such a model may lead to overtraining. Take, for instance, the modelling of N points using a mixture of Gaussians. If we let the number of Gaussians (and thus, the number of parameters) be unconstrained, the maximum likelihood model will assign one Gaussian for each data point with that data point as the mean and a variance of zero. Clearly, this collection of deltas will not generalize well to new data.

The objective, then, is to strike a balance between the fit of the data to the model and the number of parameters used in the model. BIC does this by scoring a model according to a penalized log likelihood [3]. Let $L(\mathcal{X}, M)$ be the maximum likelihood of the data \mathcal{X} for a given model M , which is obtained by tuning the free parameters of M (e.g., mean, covariance) to maximize the likelihood. The BIC score is defined as:

$$BIC(M) = \log L(\mathcal{X}, M) - \lambda \frac{1}{2} \#(M) \times \log(N) \quad (2)$$

where the penalty weight $\lambda = 1$, $\#(M)$ is the number of free parameters in the model M , and N is the number of data points. Often in the literature BIC is written as the negative of the above expression, so as to relate it to the Minimum Description Length (MDL) principle, of which it has been shown [2] BIC is a derivative. The principle essentially states that one should select the model that gives the shortest description of the data. The negative of the above expression then becomes a measure of description length which one seeks to minimize, rather than maximize, as is the case here.

2.2 Applications to Clustering

In [3], Chen and Gopalakrishnan describe a method for applying BIC to hierarchical clustering. Let \mathcal{X} be as above and let $\mathcal{C}_k = \{c_i : i = 1, \dots, k\}$ be a clustering which has k clusters. If we model each cluster as a multivariate Gaussian, $\mathcal{N}(\mu_i, \Sigma_i)$, where μ_i is the estimated sample mean and Σ_i is the estimated sample covariance, the BIC score is:

$$BIC(\mathcal{C}_k) = \sum_{i=1}^k [\log f(\mathcal{X}_i | \mu_i, \Sigma_i)] - k \frac{\alpha}{2} \log(N) \quad (3)$$

where \mathcal{X}_i represents those data points assigned to cluster c_i and α is the number of free parameters for one Gaussian. For a multivariate Gaussian random variable of dimension d , $\alpha = d + \frac{1}{2}d(d+1)$. The first term represents the mean parameters and the second the number of free parameters in a $d \times d$ covariance matrix.

To find the best model according to BIC we would have to do a global search over all possible clusterings of the data, which would invariably prove too costly. For hierarchical clustering, it is possible to search in a greedy fashion to find a clustering. We can do bottom-up clustering for which at each stage we merge the two “nearest” clusters until we arrive at a single cluster. At each stage we also compute the BIC score for that clustering and the clustering which achieves the highest BIC score is selected. In the formulation in [3], the clustering stops if the next stage produces no increase in the BIC score, but this method does not necessarily result in a global maximum over all stages, so for the project it was altered to the above procedure.

3 The Data

3.1 The ICSI Meeting Corpus

The data utilized was obtained from the International Computer Science Institute (ICSI) Meeting Corpus [4]. The corpus consists of recordings of regular meetings which proceeded at the Institute and was collected to provide a data set on which to perform speech and speaker-related experiments in a multi-speaker environment. During the meetings, participants were outfitted with headset microphones which recorded their speech on individual channels, while tabletop microphones recorded all speakers present. The waveform for each channel (headset or tabletop mic) was then stored as a

separate file, sampled at 16 kHz and 16 bits. The meetings were labelled according to categories (e.g., Robustness, Network Services and Applications, etc.) and transcriptions with unique speaker labels were made. There are 75 meetings in the corpus, for a total of 72 hours, though this project utilizes a tiny subset of this, for purposes of simplifying the analysis.

3.2 Feature Extraction

In the speech and speaker recognition communities, analysis is seldom performed on speech waveforms, rather a more robust parameterization of the data is used. One of the most common parameterizations is that of feature vectors, or frames, composed of mel-frequency cepstral coefficients (MFCCs). The process of computing the features is as follows[5],[6]:

- 1) An FFT is taken of a windowed version of the waveform over an in-

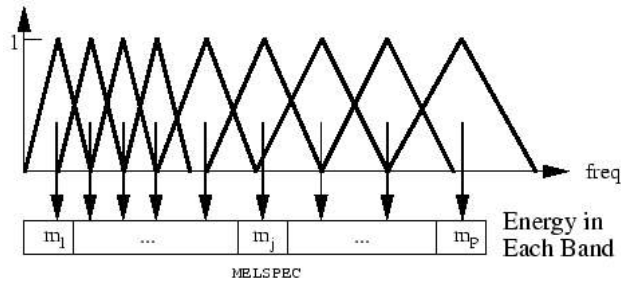


Figure 1: Mel-scale Filter Bank

terval ranging from 20-30 ms. For the project a Hamming window, as is typical, was used and the window interval was 25 ms.

- 2) The magnitudes of the FFT coefficients are then binned by correlating them with each triangular filter in a mel-scale filter bank as shown in figure 1. The mel-scale is an experimentally obtained frequency scaling based on human pitch perception and is obtained by:

$$\text{Mel}(f) = 2595 \log_{10}\left(1 + \frac{f}{700}\right). \quad (4)$$

Here binning means that each FFT magnitude coefficient is multiplied by the corresponding filter gain and the results accumulated. Thus, each bin

holds a weighted sum representing the spectral magnitude in that filter bank channel.

4) The log of these values is taken.

5) The cepstral features are computed according to:

$$c_i = \sqrt{\frac{2}{N}} \sum_{j=1}^N k_j \cos\left(\frac{\pi i}{N}(j - 0.5)\right) \quad (5)$$

where the k_j are the log filter bank coefficients and N is the feature vector dimension. c_0 serves as an energy parameter.

5) The first K cepstral features desired are kept and the rest discarded. This corresponds to a smoothing of the cepstral envelope. In the presented implementation, $K = 19$.

6) The window is shifted over by 10 ms and the process is repeated.

In addition to using the cepstral coefficients, it is often helpful to augment the feature vector with a difference of the cepstras (delta features) and a difference of these differences (delta-delta features). The delta features are obtained by:

$$d_t = \frac{\sum_{\theta=1}^{\Theta} \theta (c_{t+\theta} - c_{t-\theta})}{2 \sum_{\theta=1}^{\Theta} \theta^2} \quad (6)$$

and the delta-delta features can be computed in a similar fashion. For the project, Θ was set to be 2. Furthermore, delta-delta features were not utilized. Another feature utilized both in general practice and in the presented implementation is that of a log energy term. The energy is computed as the log of the signal energy. So for speech samples $\{s_n, n = 1, N\}$, the energy is given by:

$$E = \log \sum_{n=1}^N s_n^2 \quad (7)$$

The end result of this process is a sequence of feature vectors, $\vec{X}_1, \vec{X}_2, \dots, \vec{X}_M$ of 40 components: 19 mel cepstra, their delta coefficients, an energy coefficient, and the zeroth cepstrum, c_0 .

One key feature of the MFCC parameterization is that the coefficients tend to have very little correlation. As a result, when performing statistical modelling, a diagonal covariance matrix can often be assumed, which was done in this case.

3.3 Segmentation

For the multi-speaker environment it is often the case that the segmentation of the speech presents a large problem in and of itself. Typically one has access to a single file containing multiple speakers who (hopefully) speak in turn. It then becomes necessary to identify the break points where one speaker ends and another begins and segment the data accordingly. Performing this procedure greatly aids in clustering as clustering can proceed at the segment level rather than the frame level, the latter being much more computationally intensive and also more prone to errors as it does not necessarily take into account continuity constraints of speech (we cannot have, for example, a speech segment consisting of one frame). As the main interest of the project was that of clustering rather than segmentation, the segmentation issue was completely bypassed by using expert segments; that is, the segments were obtained according to timing information contained in the true transcription. In addition, the speech segments for a given speaker were extracted from the waveform file recorded from his/her headset microphone to ensure the best audio—and consequently data—quality possible. This also minimizes corruption of segments from speakers talking simultaneously.

4 Design and Implementation

This section provides a more refined look at the overall process thus far described in implementing the proposed algorithm:

Segment

A given conversation is first segmented. The segmentation takes place for each individual speaker channel as mentioned in section 3.3. We now have the data represented as segments, which are collections of feature vectors. The segments have and retain the true speaker identity information, for purposes of evaluation at the conclusion of the process. In an attempt to ease the ability to cluster, a minimum duration of 2 seconds was imposed on the segment collection. This facilitates clustering by preventing very small segments—perhaps consisting of a single “um” uttered by the speaker, for example—from occurring. These small segments consist of a small number of frames and as such the estimates of their parameters are poor. The number of frames in a segment ranged from 200 to 1800, with an average of about 400 frames per segment.

Initialize

The clusters were initialized by letting each segment be its own cluster and each cluster be a multivariate Gaussian. The mean and covariance parameters were estimated according to the maximum likelihood estimates:

$$\vec{\mu}_{ML} = \frac{1}{N} \sum_{i=1}^N \vec{X}_i \quad (8)$$

and

$$\Sigma_{ML} = \frac{1}{N} \sum_{i=1}^N (\vec{X}_i - \vec{\mu}_{ML})(\vec{X}_i - \vec{\mu}_{ML})^T \quad (9)$$

Having assumed a diagonal covariance, we only take the diagonal elements of Σ_{ML} . The same results can be achieved by looking at each vector component individually and doing maximum likelihood estimation because of the diagonal assumption. At one point, the use of the KLT (a diagonalizing transform) was investigated to improve this assumption, but no improvement in system performance resulted.

Compute BIC Score

An initial BIC score is computed according to equation (3) and the cluster configuration is stored. The frames (i.e., vector data points) are taken to be independent so that $f(\vec{x}_1, \vec{x}_2, \dots, \vec{x}_N | \mu, \Sigma) = \prod_{i=1}^N f(\vec{x}_i | \mu, \Sigma)$. This independence of frames assumption, though incorrect, is frequently made in statistical modelling for speech and speaker recognition to facilitate analysis, and has often resulted in well-performing systems.

Merge Nearest Clusters

We seek to merge the two nearest clusters according to some distance metric. The metric selected was the symmetrized Kullback-Leibler (KL) divergence. The KL divergence, also known as relative entropy, is given by:

$$D(f||g) = \int f(x) \log \frac{f(x)}{g(x)} dx \quad (10)$$

and serves as a measure of the similarity of one distribution to another. The KL divergence is not a true distance, however, since it is neither symmetric nor satisfies the Triangle Inequality. It is possible to treat it as a metric, however, by symmetrizing. The symmetrized divergence D_{sym} is given by:

$$D_{sym}(f||g) = D(f||g) + D(g||f) \quad (11)$$

For multivariate Gaussians with diagonal covariance matrices this becomes:

$$D(f_1||f_2) = -d + \frac{1}{2} \sum_{i=1}^d \left\{ \frac{\sigma_{1i}^2}{\sigma_{2i}^2} + \frac{\sigma_{2i}^2}{\sigma_{1i}^2} + \frac{(\mu_{1i} - \mu_{2i})^2}{\sigma_{1i}^2} + \frac{(\mu_{1i} - \mu_{2i})^2}{\sigma_{2i}^2} \right\} \quad (12)$$

where d is the vector dimension. Once two clusters are merged, the parameters of the new cluster can be quickly updated according to:

$$\mu_3 = \frac{n\mu_1 + m\mu_2}{n + m} \quad (13)$$

and

$$\sigma_3^2 = \frac{m(\sigma_2^2 + \mu_2^2) + n(\sigma_1^2 + \mu_1^2)}{n + m} - \mu_3^2 \quad (14)$$

where n is the number of data points in cluster 1 and m is the number of data points in clusters 2. Rather than look at the divergence between two clusters, one could imagine examining the sum of the divergences from each cluster to the cluster which would result after the merger. This measure tries to take into account the future state of the system, and was compared to the previous measure. This latter metric showed poorer performance, however, and as a consequence was not used.

Iterate

The BIC score of the new configuration is computed, and if it exceeds that of all past BIC scores, the configuration is saved as the maximum BIC score clustering. We then continue to merge and compute BIC scores until we have only one cluster remaining. At completion the maximum BIC score clustering has been saved and will be selected.

Evaluation

Evaluating the quality of the resulting clustering is not straightforward. One obvious metric is a simple comparison of the resulting number of clusters to the true number of subpopulations. One problem with this is that it ignores the composition of the clusters. One could, for example, produce the correct number of clusters, but each cluster may contain a mixture of the members of the various subpopulations. Another aspect of composition is the number of points in each cluster. It may be the case, for example, that the correct number of clusters is obtained, but one cluster is a combination of two subpopulations and another consists of one or two data points of a subpopulation. An additional metric which helps to alleviate some of these

problems is that of cluster purity. The purity of a cluster is defined to be the ratio between the number of data points by the majority subpopulation in the cluster to that of the total number of data points in that cluster. Again, there are complications with this metric if one does not take into account the number of points in the clusters. As a result of these issues, for the project the number of clusters, the purity, and the composition were all used for analysis. In addition, the composition and purity were determined on the segment (versus frame) level, since it is segments which are merged and not frames.

5 Results

Meeting	#Speakers	#Clusters	Min. Purity	#Singletons	#Duos	#Trios
Bmr001	3	7	1.00	4	0	0
Bmr002	4	22	1.00	15	3	0
Bmr009	5	33	0.65	29	0	1

The above table shows the results of the clustering algorithm for three representative meetings. The terms 'singleton,' 'duo,' and 'trio' refer to clusters having just one, two, or three segments, respectively. An example of the BIC curve produced by a run of the algorithm is shown in figure 2.

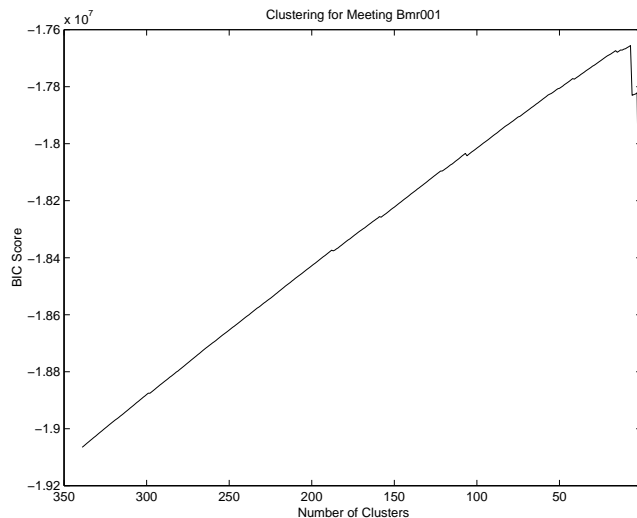


Figure 2: Example BIC curve for clustering algorithm

The first meeting, Bmr001, consists of three male speakers. The final clustering selected by the algorithm is that of seven clusters, though four of them are singletons. All of the clusters contain only segments from a single speaker, which means that the algorithm, with the exception of the singletons, was able to accurately cluster the segments of the three speakers. Indeed, if the segments found in the singletons are removed from consideration, it was shown that the clustering proceeds perfectly.

The second meeting, Bmr002, consists of three male speakers and one female speaker. In this case the selected number of clusters differs greatly from the true number of speakers, but we see that the majority of the excess clusters is singletons, and the remainder is duos. Again, all clusters are 100% pure, so the situation is that of the four speakers, minus the outlier segments, accurately clustered. This is confirmed by removing these segments, which yields perfect clustering.

In the final meeting shown, Bmr009, the speakers are three males and two females. Here the algorithm performs even more poorly in the choice of clusters. We see, though, that the singleton issue plays a key role in this. There is another issue in this case, however, as the clusters are not all 100% pure. Closer analysis revealed that the two female speakers were completely merged as a single cluster, as were two of the male speakers. Even after removing the singletons and the trio this occurs, suggesting the phenomena are independent.

Given the results it seems that the proposed algorithm suffers from issues common to clustering algorithms: namely those of robustness to outliers and overlap of subpopulations in the feature space. The singletons, duos, and trios, which remain once the algorithm has arrived at a final clustering are segments that are not merged at any previous step and as such are in some sense “distant” from the large clusters and the segments contained therein. The overlapping of subpopulations is supported by the fact that the two mixed clusters each contained speakers of one gender; two females, for example, are more likely to occupy the same acoustic feature space than a male and a female. In addition, the possibility of overlap increases with the number of speakers. In our above cases, the phenomenon did not appear until five speakers were involved, for example.

6 Conclusions and Future Work

In this paper, an approach to speaker clustering using the Bayesian Information Criterion in an agglomerative clustering setting was presented. From

an analysis of the results taken from a few data sets, it appears that the clustering algorithm, though in some situations well-performing, in general lacks robustness to outliers and subpopulation overlap. The latter issue becomes more serious as the true number of subpopulations (i.e., speakers) grows in size. These problems are common to clustering algorithms and the solutions are not clear.

One possible change that could be examined is the distance metric, which affects the merging procedure. Though the symmetrized KL divergence proved to perform well, various other measures exist. Wilcox et al. in [7], for example, describe a metric related to segment likelihoods that could be utilized. Another area of exploration is the features used. There exist many choices regarding features and some may produce better speaker separation, and consequently clustering, than those utilized in the present implementation.

References

- [1] Jordan, M., An Introduction to Probabilistic Graphical Models, chap.10, pp.4-9, unpublished (2003).
- [2] Hansen, M. and B. Yu, "Model Selection and the Principle of Minimum Description Length," J. Amer. Statist. Assoc., vol. 96, 746-774, 2001.
- [3] Chen, S.S. and P.S. Gopalkrishnan, "Clustering Via the Bayesian Information Criterion with Applications in Speech Recognition." ICASSP, pp. 645-648, 1998.
- [4] Janin, A., D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters, "The ICSI Meeting Corpus," ICASSP, pp. 364-367, 2003.
- [5] Hidden Markov Model Toolkit (HTK) Book is available at <http://htk.eng.cam.ac.uk/docs/docs.html>
- [6] Gold, B. and N. Morgan, Speech and Audio Signal Processing: Processing and Perception of Speech and Music, pp.271-277, 2000.
- [7] Wilcox, L., F. Chen, D. Kimber, and V. Balasubramanian, "Segmentation of Speech Using Speaker Identification," ICASSP, pp. 161-164, 1994.