A Framework for Classification and Segmentation of Massive Audio Data Streams

Charu C. Aggarwal
IBM T. J. Watson Research Center
19 Skyline Drive
Hawthorne, NY 10532
charu@us.ibm.com

ABSTRACT

In recent years, the proliferation of VOIP data has created a number of applications in which it is desirable to perform quick online classification and recognition of massive voice streams. Typically such applications are encountered in real time intelligence and surveillance. In many cases, the data streams can be in compressed format, and the rate of data processing can often run at the rate of Gigabits per second. All known techniques for speaker voice analysis require the use of an offline training phase in which the system is trained with known segments of speech. The state-of-the-art method for text-independent speaker recognition is known as Gaussian Mixture Modeling (GMM), and it requires an iterative Expectation Maximization Procedure for training, which cannot be implemented in real time. In this paper, we discuss the details of such an online voice recognition system. For this purpose, we use our micro-clustering algorithms to design concise signatures of the target speakers. One of the surprising and insightful observations from our experiences with such a system is that while it was originally designed only for efficiency, we later discovered that it was also more accurate than the widely used Gaussian Mixture Model (GMM). This was because of the *conciseness* of the micro-cluster model, which made it less prone to over training. This is evidence of the fact that it is often possible to get the best of both worlds and do better than complex models both from an efficiency and accuracy perspective.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications

General Terms

Algorithms

Keywords

Speaker Recognition

KDD'07, August 12–15, 2007, San Jose, California, USA

1. INTRODUCTION

The problem of speaker voice analysis and classification is useful in a number of applications such as real time monitoring, detection, and surveillance. In this paper, we are concentrating on the problem of *text-independent* speaker classification in which the actual textual content of the speech is not available for modeling purposes. A number of statistical and machine learning methods have been recently proposed for speaker classification. Some examples of such techniques may be found in [3].

A well known method for speaker classification and identification is that of Gaussian Mixture Modeling (GMM) [4]. The first step is to extract multi-dimensional feature vectors in order to represent portions of sampled speech. In this method, it is assumed that each data point extracted from the speech segments from a number of known speakers are used to estimate the parameters of a GMM model. Then the data points from an unknown speech segment are applied to each speaker model in order to estimate the maximum likelihood fit. The model with the highest fit is reported as the relevant class label. The GMM modeling method has been widely popular because of its intuitive appeal and effectiveness, and has therefore been used extensively.

In many applications, it is desirable to perform the speaker identification in real time. In such cases, we need to have a system which is adaptive enough to learn characteristics of new speakers in real time, and used these learned profiles in order to construct the final model for speaker classification. Unfortunately, popularly used models such as GMM are not very appropriate for real time speaker modeling. This is because the first step of determining the parameters of the mixture model requires an iterative computationally intensive approach known as the EM algorithm. The second stage of model fitting requires the evaluation of a log likelihood criterion on the data set for each model. These techniques can be computationally intensive in practice, and are often difficult to use effectively for real time modeling and classification. Furthermore, we are also interested in other mining variations of the classification problem in which it is desirable to model or match individual segments of speech from unknown speakers in real time. This is not possible with an iterative approach such as the EM algorithm. In addition, since we are making the stream assumption for all data processing, we assume that each point in the data can be scanned only once throughout the computation. This assumption is also violated by all other speaker identification systems [3] known to us. Therefore, we need to construct

a system which can work with the constraints of a one-pass system and still provide accurate results for speaker identification. In addition to speaker classification, we would like to design a system which is capable of detecting quick changes in the pattern of the underlying data stream. Such a system is useful in applications in which it is desirable to segment portions of speech into different speakers. We note that this segmentation may need to be done in an unsupervised way, since example segments of the different speakers may not be known in advance. In order to design a system which can work with such challenges, we construct a number of techniques which are designed towards *stream based online processing* of massive amounts of audio data. We adapt our earlier research results for stream micro-clustering [1] in order to create fast *cluster based signatures* of the data. These signatures are constructed and used in real time in order to perform the speaker identification.

This paper is organized as follows. In the next section, we will describe the speaker recognition system. In section 3, we will present the experimental results obtained from deploying these algorithms over a number of real data sets. Section 4 contains a number of conclusions and observations.

2. THE SPEAKER IDENTIFICATION SYSTEM

In our online voice recognition system, we will use a non-parametric density estimation approach. The idea is to determine the probability distribution of the data for each speaker and map it to the behavior of unknown test segments in order to perform the matching. The reason for using a non-parametric approach is that most parametric methods such as the GMM model require an iterative approach to estimate the densities in the underlying data. On the other hand, a non-parametric approach adapts very well to an online application such as that of clustering data streams. A well known non-parametric method of finding the data distribution is that of kernel density estimation. However, density estimation is often an inefficient method since it requires us to estimate the data behavior over all regions of the data. In high dimensionality, most of the regions in the data are sparse, and the estimation will need to be performed over a very large number of data points (which is exponentially increasing with dimensionality) in order to provide a comprehensive overview of the data behavior. This can rapidly become untenable for non-uniform data distributions of even modestly high dimensionality. In order to substitute for using density distributions directly, we will use very fine grained *micro-clusters* in the data. Micro-clustering is a concept which is used to track very detailed level of statistics of clusters with high granularity. This is a desirable approach when the data sets are received in the form of massive data streams. The distribution of the data for different speakers over these fine grained clusters are used as a surrogate for the actual densities in the underlying data. We will see that this turns out to be a practical and scalable approach in most applications, and also simulates the underlying densities quite well. In order to design the voice recognition system, we will build upon some micro-clustering machinery developed in [1]. The micro-clustering framework discusses some summary statistics which are used to maintain the statistical information about the clusters. We assume that the dimensionality of the data stream is d.

The definition of a micro-cluster is as follows:

DEFINITION 1. *A micro-cluster for a set of d-dimensional points $\mathcal{C} = \{X_{i_1} \ldots X_{i_n}\}$ with time stamps $T_{i_1} \ldots T_{i_n}$ is defined as the $(2 \cdot d+2)$ tuple $(\overline{CF2^x}(\mathcal{C}), \overline{CF1^x}(\mathcal{C}), CF^t(\mathcal{C}), n(\mathcal{C}))$, wherein $\overline{CF2^x}(\mathcal{C})$ and $\overline{CF1^x}(\mathcal{C})$ each correspond to a vector of d entries. The definition of each of these entries is as follows:*

- *For each dimension, the sum of the squares of the data values is maintained in $\overline{CF2^x}(\mathcal{C})$. Thus, $\overline{CF2^x}(\mathcal{C})$ contains d values. The p-th entry of $\overline{CF2^x}(\mathcal{C})$ is equal to $\sum_{j=1}^{n} (x_{i_j}^p)^2$.*
- *For each dimension, the sum of the data values is maintained in $\overline{CF1^x}(\mathcal{C})$. The p-th entry of $\overline{CF1^x}(\mathcal{C})$ is equal to $\sum_{j=1}^{n} x_{i_j}^p$.*
- *The last update time (which is $max\{T_{i_1} \ldots T_{i_n}\}$) is maintained in $\overline{CF^t}(\mathcal{C})$.*
- *The number of data points is maintained in $n(\mathcal{C})$.*

The design of micro-cluster statistics is chosen so as to to make some important predictions about the data points in them. We make the following two observations about the statistics [1] stored in the micro-clusters:

OBSERVATION 2.1. *The mean and variance of the data points in a cluster can be determined from the micro-cluster statistics.*

In addition, the micro-clusters satisfy the *additivity property*. We omit the details of the proofs since they can be found in [5].

OBSERVATION 2.2. *The micro-cluster statistics satisfy the additivity property. The micro-cluster statistics for the union of two sets of data points is the sum of the micro-cluster statistics of the individual sets of points.*

As discussed later, the above observations are used during the clustering process. The clustering process is used to create and store summary frequency information. This summary information is used in order to perform effective classification. A set of a clusters can be used to create a *signature summary* of the data. This signature summary is defined in terms of the frequencies of the different data points drawn from the segment S, in terms of their distribution over the clusters $\mathcal{C}_1 \ldots \mathcal{C}_k$. The signature summary is used as a surrogate for the probability density of the data points in the stream. As we will observe later, this is not an unreasonable assumption when a fine granularity of clustering is maintained. Let us denote the data in the last segment of length S by $\mathcal{D}(S)$. We define the *signature summary* of the data set $\mathcal{D}(S)$ over the segment S with respect to the clusters $\mathcal{C}_1 \ldots \mathcal{C}_k$ as follows:

DEFINITION 2. *The signature summary $\beta(\mathcal{C}_1 \ldots \mathcal{C}_k, S)$ for a set of k clusters $\mathcal{C}_1 \ldots \mathcal{C}_k$ and segment of data points S is defined as the k-dimensional vector $(f_1 \ldots f_k)$, where f_i is defined as follows:*

$$f_i = |\mathcal{D}(S) \cap \mathcal{C}_i| / |\mathcal{D}(S)| \quad (1)$$

It is important to note that the signature summary essentially defines the relative distribution of the data points across the different clusters. When the individual clusters have very small variance, they can be used as a surrogate for the true density distribution. In order to explain the relationship between the probability density and the frequencies

of the data points across different clusters, we make the following observations.

Let $\overline{X}(\mathcal{C}_i)$ and $h(\mathcal{C}_i)^2$ be the centroid and variance of cluster $\mathcal{C}_i$. Then, the density estimate $\alpha(x)$ at any point x in the space can be constructed as follows:

$$\alpha(x) = \sum_{i=1}^{k} \frac{f_i}{\sqrt{2 \cdot \pi} \cdot (h(\mathcal{C}_i) + h')} \cdot e^{-\frac{|(x - \overline{X(\mathcal{C}_i))^2}|}{2(\cdot h(\mathcal{C}_i)^2 + h'^2)}} \quad (2)$$

We note that Equation 2 is a modification to the standard density function which is often used in kernel density estimation of individual data points. The primary difference is the cluster-specific value of the bandwidth $h(\mathcal{C}_i)$. Here h' is an additive bandwidth which is defined in the same way as standard kernel density estimation, except that we use the number of clusters rather than the number of points to define h'. According to the Silverman approximation rule, for a data distribution with n points (clusters) and variance σ of the data points (cluster centroids), the value of h' is chosen to be $1.06 \cdot \sigma \cdot n^{-1/5}$. We note that $\lim_{n \to \infty} h' \to 0$. In effect, the density estimate is equal to the sum of Gaussian kernels which are centered at different clusters, and have bandwidth which is defined by the variance of the data points in the cluster. For very fine grained clusters, such an approach provides a very good estimate to the process of kernel density estimation. This is because this limiting case defines a situation in which the value of $h(\mathcal{C}_i)$ tends to zero. On substituting the value of $h(\mathcal{C}_i) = 0$ in Equation 2, we obtain the same formula used for standard kernel density estimation. We summarize this observation as follows:

OBSERVATION 2.3. *In the limiting case, when each cluster contains only one data point, the value $\alpha(x)$ is equal to the standard kernel density estimate.*

The essential idea behind this exercise was to illustrate that the signature summary can encode very detailed information about the probability distribution of the data even for very fine grained clustering. We note that the difference in probability density between two distributions $\alpha_1(\cdot)$ and $\alpha_2(\cdot)$ may be defined as follows:

$$Diff(\alpha_1, \alpha_2) = \int_x |(\alpha_1(x) - \alpha_2(x)| \mathrm{d}x \quad (3)$$

This difference is essentially equal to the integral of the difference of the two distributions over the entire space. Intuitively, we would like to use this difference in density distribution as a measure for the fit between two density distributions. Note that even when probability densities can be estimated accurately, this difference can be difficult to compute in practice. Therefore, as the surrogate, we compute the difference in signature summaries between the two data streams.

DEFINITION 3. *The distance $Dist(H_1(\cdot), H_2(\cdot))$ between two signature summaries $H_1(\cdot) = (f_1 \ldots f_k)$ and $H_2(\cdot) = (f'_1 \ldots f'_k)$ is defined as follows:*

$$Dist(H_1(\cdot), H_2(\cdot)) = \sum_{i=1}^{k} |f_i - f'_i| \quad (4)$$

This distance is thus defined in terms of the L_1-norm. We note that this distance is essentially a discrete analogue of the difference in density distributions. In this case, we are summing the discrete differences in probabilities that a voice packet belongs to a particular micro-cluster.

Algorithm *ConstructSpeechSignatures*(Speech Segment: S, Current Clusters: $\mathcal{C}_1 \ldots \mathcal{C}_k$, NumberOfClusters: k);
begin
for each speaker label i
  $H_i = (0, \ldots 0)$;
for the next data point $\overline{X}$ in S **do**
  begin
  Determine closest micro-cluster $\mathcal{C}_i$ to $\overline{X}$;
  Add $\overline{X}$ to $\mathcal{C}_i$ and update
    micro-cluster statistics for $\mathcal{C}_i$;
  Update signature counts for corresponding cluster in H_j
    where j is the label of the current data point;
  end
end

Figure 1: Constructing Summary Signatures

The micro-clustering algorithm is used in order to construct profiles from the speaker data. These profiles are essentially the signatures in the data stream. The micro-clustering algorithm uses a nearest neighbor clustering algorithm in which each data point is assigned to the mean of the closest cluster. As discussed in Observation 2.1, the mean of the cluster can be computed from the micro-cluster statistics. Once the closest micro-cluster has been computed, we add the statistics for that data point to the micro-cluster statistics. This is possible to achieve because of the micro-cluster statistics discussed in Observation 2.2. The overall process for signature construction is discussed in Figure 1.

One problem which may arise in a real application is that of cluster centroid drift. Cluster centroids can drift when the new data points which are added to the clusters have a different distribution from the original set of data points. Such a drift may affect the significance of the signatures over time, since different voice streams may be received one after another, and may have slightly different cluster centroids. This situation is also quite possible in real applications since the data distribution may vary over time. This is an undesirable situation, since it is difficult to fit the test data accurately. We note that the key function of cluster centroids is to ensure that the most relevant (dense) regions in the data are represented by these anchors. The exact location of these clusters is not very important, as long as they remain fixed over time, and represent most of the dense regions. Furthermore, we note that the only statistic which is used during the actual speaker identification process is the relative frequencies of the data points in the different clusters. Therefore, it is acceptable to fix the cluster centroids once each micro-cluster is sufficiently populated. Therefore, we pick a small threshold (say 100 voice packets) for each micro-cluster, and apply the micro-cluster update algorithm in a normal way till this limit. However, when the number of data points in the micro-cluster exceed this threshold, we do not add the assigned data point to the micro-cluster, but only update the count of the number of data points in it. The other statistics are updated by the corresponding multiplicative factor in order to reflect a larger number of data points. This ensures that the centroid of the micro-cluster remains fixed, but the frequency statistics are appropriately maintained.

During the training phase, we also maintain the signatures

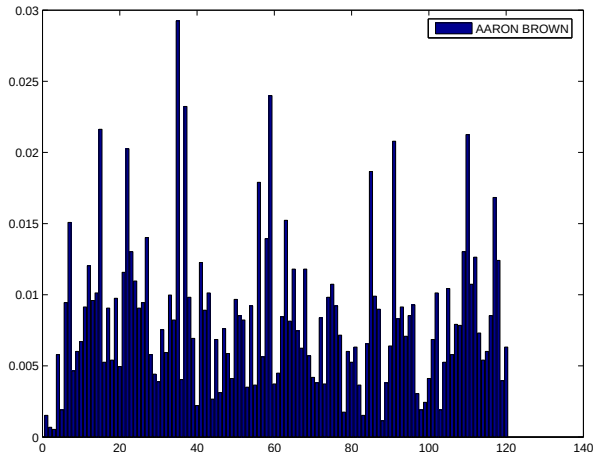

Figure 2: Training Signature for Aaron Brown

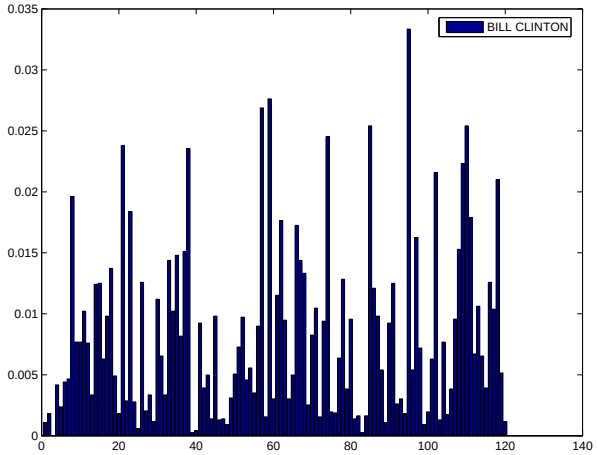

Figure 3: Training Signature for Bill Clinton

for the different speaker labels. These signatures are used for the speaker classification process. Specifically, we assume that the k-dimensional vector H_i is used in order to track the micro-cluster frequency behavior of the speaker i. Whenever a new data point is received, we add to the frequency for the corresponding micro-cluster in the signature for speaker i.

We note that the great flexibility of the scheme is that the testing phase is essentially no different from the training phase, except that the micro-cluster statistics do not need to be updated. As in the training phase, we construct the signature for the test segment in order to create the summary signatures. This test segment is matched with the training segments using the relationship discussed in Equation 4. The closest matching signature is returned as the speaker identification.

The method discussed in this paper can also be used for speech segmentation. The essential idea in speech segmentation is to partition the conversations between one and more speakers in an unsupervised way. We note that this is often required as a pre-processing step to speaker identification in a real time application in which one does not have the time to perform the segmentation manually after using the content of the speech. In order to extend the methodology to these cases, we perform the micro-clustering continuously as in the previous case, except that we save the signatures in each window at intervals which are equivalent to the window size. For each window, we determine the distance between the signatures for the current window and the window just before it. This leads to a continuous alarm function over the data stream. This alarm function can be used in order to track sudden changes in the trends of speaker behavior. This can create a clear segmentation between the different speakers in a conversation. More details and experimental tests of the approach will be described in an extended version of this paper.

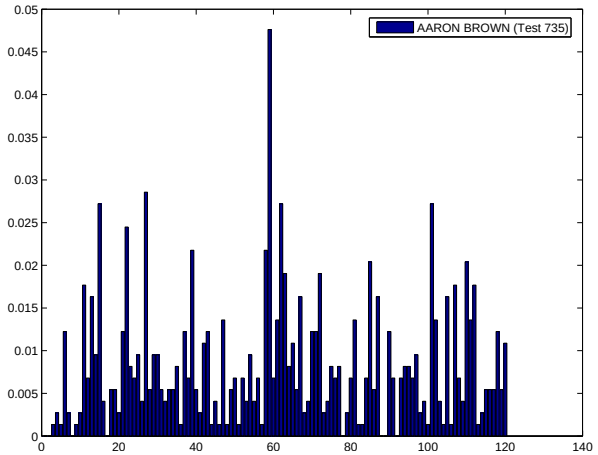

Figure 4: Testing Signature for Aaron Brown (735 records)

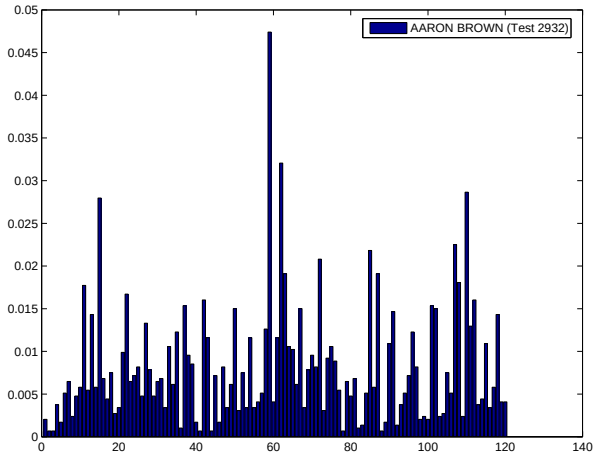

Figure 5: Testing Signature for Aaron Brown (2932 records)

3. EXPERIMENTAL RESULTS

In this section, we will discuss some empirical results illustrating the effectiveness and efficiency of the method on a variety of voice data sets. For the purpose of comparison, we will use a standard Gaussian Mixture model, which was derived from a software called netlab [2]. For each speaker, we first constructed a GMM on the training data set, and

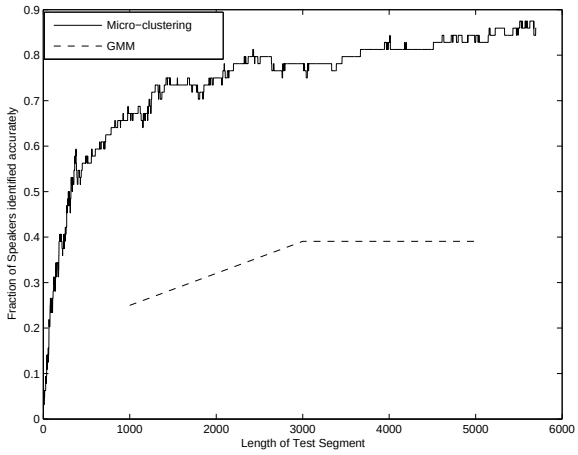

Figure 6: Accuracy of Classification for Test Segments of Different Sizes)

calculated the log-likelihood of each speaker in the training data on the test data segments. The maximum likelihood speaker was reported as the speaker for the test data set. We note that the use of GMMs is considered as state of the art for text independent speech recognition [4]. While such models often cannot be used easily for online training and detection, the accuracy provides a idea of the desired accuracy of an online system.

The data set used is the HUB-64 data set, which contains the voice signal of 64 different public personalities such as Wolf Blitzer, Bill Clinton, Al Gore, Candy Crowley etc. In each case, a training data set and test data set was constructed from the data. In most real time applications, the size of the test data can be quite small and cannot be controlled. Furthermore, it is desirable to use as small a test segment as possible to perform the classification, since this affects the latency of speaker identification. On the other hand, very large sets of training data can often be made available in most applications for well known speakers by compiling data from multiple sources. Therefore, we chose to divide the training data such that the training data size was twice the test data in each case.

First, we will discuss some examples of signatures which were obtained by using micro-clustering on the different data sets. In Figures 2 and 3, we have illustrated some examples of the signatures obtained from the Aaron Brown, and Bill Clinton respectively. In each case, we have used 120 micro-clusters in order to construct the signatures. In each case, the cluster-id is illustrated on the X-axis, whereas the relative frequency of the corresponding cluster is illustrated on the Y-axis. It is clear that in each case, the corresponding signatures are quite distinct. The distinct natures of these signatures are very useful in performing an exact classification process on different test segments. In Figures 4 and 5, we have illustrated the signatures obtained on test segments of different sizes for the signatures corresponding to Aaron Brown. It is clear that the signature in Figure 5 from the test segment of larger size is closer to the signature on the training data set. As a result, the accuracy of classification increases with increasing size of the test segment. In Figure 6, we have illustrated the accuracy on test segments of different sizes. In each case, we trained for all 64 speakers simultaneously, and classified a test segment into one of these 64 speakers. This is significantly more difficult than a problem in which we try to distinguish a particular speaker from a control or mixed signal. This is because many speakers may have inherent similarity in their voice patterns, and it often becomes difficult to distinguish between a very large number of speakers using a single model. In Figure 6, we have illustrated the number of records on the X-axis, and the classification accuracy over the 64 speakers on the Y-axis. In the same Figures, we have illustrated the accuracy using Gaussian Mixture Models on test segments of different sizes. The accuracy of classification increases considerably with increasing test segment size because it is easier to build more accurate models on larger test segments. It is interesting to see that the accuracy of the micro-clustering process was significantly higher than the more computationally intensive (and offline) GMM approach. Our initial goals in designing this system were only to construct a system which could work in a one-pass approach for an online data stream, and we did not expect the micro-clustering approach to outperform the GMM model in terms of accuracy. Therefore, the (substantial) qualitative advantage of the micro-clustering approach over the GMM model was particularly surprising. We believe that the higher accuracy is because of the more compact representation of the micro-clustering approach.

4. CONCLUSIONS AND SUMMARY

In this paper, we discussed an online system for voice recognition in data streams. We discussed a micro-clustering approach which is not only efficient, but is significantly more accurate than the state-of-the-art GMM model. The greater accuracy of the micro-clustering approach is particularly surprising considering the fact that the GMM model is designed using the iterative EM-approach which does not have the constraints of the one-pass stream based micro-clustering model. While the simplicity and compactness of the micro-clustering signature representation was originally designed only for speed, we found that it achieves the dual purpose of avoiding the overtraining of the parameter-heavy GMM model − a classic example of the "less is more" paradigm in machine learning.

5. REFERENCES

[1] C. C. Aggarwal, J. Han, J. Wang, P. Yu. A Framework for Clustering Evolving Data Streams. *VLDB Conference*, 2003.

[2] I. Nabney. Netlab: Algorithms for Pattern Recognition. *Advances in Pattern Recognition*, *Springer-Verlag*, Germany, 2001.
URL: http://www.ncrg.aston.ac.uk/netlab/down.php

[3] M. Prybocki, A. Martin. NIST's Assessment of Text Independent Speaker Recognition Performance. *http://www.nist.gov/speech/publications/index.html*

[4] D. Reynolds, T. Quateiri, R. Dunn. Speaker Verification using Adapted Gaussian Mixture Models. *Digital Signal Processing*, Vol. 10, pp. 42–54, 2000.

[5] T. Zhang, R. Ramakrishnan, M. Livny. BIRCH: An Efficient Data Clustering Method for Very Large Databases. *ACM SIGMOD Conference,* 1996.