CLUSTERING SPEAKERS BY THEIR VOICES

A. Solomonoff *A. Mielke* *M. Schmidt* *H. Gish*

GTE/BBN Technologies
70 Fawcett St, Cambridge, MA 02138 USA
asolomon@bbn.com,mschmidt@bbn.com

ABSTRACT

The problem of clustering speakers by their voices is addressed. With the mushrooming of available speech data from television broadcasts to voice mail, automatic systems for archive retrieval, organizing and labeling by speaker are necessary. Clustering conversations by speaker is a solution to all three of the above tasks. Another application for speaker clustering is to group utterances together for speaker adaptation in speech recognition. Metrics based on purity and completeness of clusters are introduced. Next our approach to speaker clustering is described and finally experimental results on a subset of the Switchboard corpus are presented.

1. INTRODUCTION

Given the rapid increase in the volume of computer-available recorded speech, for example, television and radio broadcasts, recordings of meetings and voice-mail, a growing need for automatic processing of such data exists. In this paper, the problem of organizing such audio data automatically by speaker is addressed. Applications include retrieving all conversations from a targeted speaker or labeling all conversations by speaker without having to listen to all of them. If, for example, 1000 conversations are partitioned into 20 clusters perfectly, then at most 20 utterances need to be searched to find all speech from a target of interest and only 20 utterances need to be labeled in order to have all 1000 conversations labeled. Another application of speaker clustering is speech recognition. Given more speaker specific training speech, performance of speaker adaptation methods improve.

The paper is organized into three main sections. Metrics for evaluating partitions are proposed along with a discussion of how to interpret clustering results. The second section describes our approach to clustering speakers. Finally, results are presented on a subset of the Switchboard corpus. Throughout, it is assumed that each speech message to be clustered contains speech from a single speaker.

2. CLUSTER EVALUATION

In order to do meaningful research on clustering algorithms, we need a method of scoring the quality of a partition. There are many ways of doing this, some more ad hoc than others. We want the scoring method to assign a good score to a perfect partition, and low scores to (for example) the partition that put each utterance in a separate cluster, or the partition that put all utterances in a single cluster. There should be a penalty for putting together utterances from different speakers, and also for not putting together utterances from the same speaker.

First, some definitions: Let n_{ij} be the number of utterances in cluster i that were spoken by speaker j. Let N_s be the total number of speakers, N_c the number of clusters, and let N_u be the total number of utterances. Furthermore, let $n_{\cdot j} = \sum_{i=1}^{N_c} n_{ij}$ be the total number of utterances from speaker j and let $n_{i\cdot} = \sum_{j=1}^{N_s} n_{ij}$ be the size of cluster i.

Rand Index

One method of scoring a partition is the Rand Index [1].

$$I_{Rand} = \frac{1}{2}\Big\{\sum_i n_{i\cdot}^2 + \sum_j n_{\cdot j}^2\Big\} - \sum_i \sum_j n_{ij}^2 \qquad (1)$$

In words, this is the number of utterance pairs that are from the same speaker and are not in the same cluster, or that are in the same cluster but are from different speakers. Note that smaller I_{Rand} is better.

BBN Metric

Another measure is one that we are calling the BBN Metric.

$$I_{BBN} = \sum_i \sum_j \frac{n_{ij}^2}{n_{i\cdot}} - Q \cdot N_c \qquad (2)$$

where Q is a user-specified parameter. Larger I_{BBN} is better. The parameter Q represents how badly the user prefers a few large clusters over the risk of merging things that don't really belong together.

Labeling Problem

The I_{BBN} metrics represents the value of a clustering in the following situation. Suppose that we have a large set of utterances, and we want to label them according to speaker identity or some quality that is connected to identity. We can have someone listen to an utterance and label it, but this is expensive. So a reasonable alternative is the following:

0-7803-4428-6/98 $10.00 © 1998 IEEE

1. Cluster the utterances on speaker ID somehow.
2. Select one utterance at random from each cluster
3. Have the labeling person label that utterance
4. Apply the same label to every other utterance in that cluster.

Then we could give a value V_c for each correctly labeled utterance, no points for incorrectly labeled utterances, and charge C_l for each utterance that the label person had to listen to.

Under these assumptions, one can show that the value of labeling the utterance set is (proportional to) I_{BBN}, with $Q = C_l/V_c$.

Cluster Purity

We can define both I_{Rand} and I_{BBN} in terms of a quantity called cluster purity. In doing this, we will see how similar I_{Rand} and I_{BBN} are. The *purity* of cluster i is

$$p_i = \sum_j \frac{n_{ij}^2}{n_{i\cdot}^2} \qquad (3)$$

Purity is a quantity which describes to what extent all utterances in the cluster came from the same speaker. If all utterances from a cluster are from the same speaker, then $p_i = 1$. If most but not all are from one speaker, then the purity will be close to 1. If, many different speakers are in the cluster, then the purity will be tiny. For example, if the utterances are evenly divided between k speakers, the purity is $1/k$.

In terms of purity,

$$I_{BBN} = \sum_i n_{i\cdot} p_i - Q \cdot N_c \qquad (4)$$

and

$$I_{Rand} = \sum_i n_{i\cdot}^2 \left(\frac{1}{2} - p_i\right) + \frac{1}{2} \sum_j n_{\cdot j}^2 \qquad (5)$$

Note that the last term in (5) does not depend on the partition, and so can be ignored for purposes of analyzing a clustering.

Meaning of Purity

If two utterances are picked at random from cluster i, with replacement, then the purity p_i is the probability that they both came from the same speaker. To see this, note that with probability $n_{ij}/n_{i\cdot}$ the first utterance came from speaker i. The probability that they *both* came from speaker i is $(n_{ij}/n_{i\cdot})^2$. Now if they came from the same speaker, then they both came from speaker 1 or both from speaker 2, etc. Note that all these events are exclusive, and so we can sum them to get the probability of same speaker:

$$\text{prob(same speaker)} = \sum_j \frac{n_{ij}^2}{n_{i\cdot}^2} = p_i \qquad (6)$$

Large Pure Clusters are Better

We would like to demonstrate that both I_{Rand} and I_{BBN} give better scores to partitions having large pure clusters than ones with small impure clusters. First, if the partition is altered by merging two clusters, and the original clusters have the same purity as the new cluster, then I_{BBN} increases by Q. If two pure clusters containing utterances from different speakers are merged, the score will decrease by $2n_{i\cdot} - 2Q - (n_{i\cdot} - Q) = n_{i\cdot} - Q$ since before the merge the clusters have $p_i = 1$ and after the merge the purity is $p_i = 1/2$. The parameter Q represents how much large clusters are preferred at the expense of reduced purity.

With the Rand index, merging two pure clusters of size n into a cluster with purity $p = 1/2$ makes the index worse by n^2, while merging them into a pure cluster makes the index better by n^2. Recall that "better" means smaller for I_{Rand} and larger for I_{BBN}.

Both indices, therefore behave in a way that reflects our intuitive idea of what makes a good partition.

3. CLUSTERING

We do clustering in three stages. The first stage is to compute some distance-like measure between each pair of utterances. The second stage is to create a tree of clusters or "dendrogram" by starting with each utterance in its own cluster and recursively merging clusters according to some distance-related criterion. A variation on this is to start with all utterances in a single cluster and recursively split the clusters somehow. This gives a sequence of different partitions. The last stage is to somehow pick one of the partitions, a process we call *dendrogram cutting.*

Many other methods of clustering exist, of course.

Distance matrices

We start by modeling utterances/speakers with techniques very similar to those used in speech recognition. The speech signal is converted to frames of cepstra and difference cepstra, using overlapping windows, mel-warping and all of the other frills. Then the sequence of frames is modeled with a mixture of diagonal-covariance gaussians. The model is trained using EM.

Then the "distance" between two utterances is taken to be some symmetric expression involving the likelihood of utterances with respect to models. There are several possibilities. One is the generalized likelihood ratio test:

$$d_{GLR}(u_0, u_1) = \frac{l(u_0|M(u_0))l(u_1|M(u_1))}{l(u_{01}|M(u_{01}))} \qquad (7)$$

where $l(u|m)$ denotes the likelihood of the utterance u with respect to the model m, $M(u)$ denotes the model trained (using EM) from the utterance data u and u_{01} denotes the data set made by concatenating u_0 and u_1. The denominator term in (7) is the likelihood of combined data with respect to a combined model. If the two utterances were very different, then the combined model will be rather spread-out, and will give low probability to all utterances, and therefore d_{GLR} will be large.

Another distance mechanism is cross entropy, which is a sort of symmetrized version of the Kullback-Liebler information distance:

$$d_{CE}(u_0, u_1) = \log \frac{l(u_0|M(u_0))}{l(u_0|M(u_1))} + \log \frac{l(u_1|M(u_1))}{l(u_1|M(u_0))} \quad (8)$$

This uses the fact that we expect an utterance to have large likelihood with respect to its own model and small likelihood with respect to a different model.

Some things to notice are: d_{GLR} requires that a model be trained for each pair of utterances, making it rather expensive unless the models can be quickly trained. Cross-entropy distance only requires that one model be trained for each utterance. Both distances require the same amount of work to compute likelihoods.

Agglomerative Dendrogram Construction

The algorithm is to pick the closest pair of clusters according to the distance matrix, and merge them. This step is repeated until there is only one cluster.

Cluster Distances

The distance matrix only gives the distance between pairs of single utterances, so some method is required to construct a distance between clusters from distances between single utterances. There are several possibilities, all ad hoc:

Minimum pair/single linkage Pick the smallest distance between an utterance in one cluster and one in the other cluster. This has the un-distance-like property that if two clusters have a very close pair of points, the cluster distance is tiny no matter how different the clusters are otherwise. This has not worked well for us.

Maximum pair/complete linkage Pick the largest distance from a between-clusters pair of utterances. This has the un-distance-like property that the distance between a cluster and itself is nonzero unless all of its points are identical. Still, it works well for clustering.

Average pair/linkage This is the mean of all the utterance distances. It acts quite similar to the maximum pair distance.

It is not known whether the non-distance-like properties of these cluster distances reduces their performance for clustering. Very little is known about what qualities make a cluster distance good for clustering.

Reestimation

Another method for computing a distance between clusters is *Reestimation.* Instead of trying to construct a cluster distance from many utterance distance, this simply concatenates all the utterance data within each cluster into one long utterance and trains a model for the new larger data set. Then d_{GLR} or d_{CE} is computed just as if the cluster was a single utterance.

This works well, but is CPU-intensive unless model training is very fast. A great many models have to be trained, some on very large amounts of data.

Dendrogram Cutting

If we knew the purity and size of each cluster in a partition, we could compute the score of the partition. If we could do this for each cut/stage in the dendrogram, then we could select a single partition as the one having the best score.

We always know the size of each cluster in a partition, but to know its purity requires knowing the true speaker of each utterance. So we try to somehow estimate the purity.

Purity estimation can potentially be used for other clustering purposes as well. For example, refining a partition by greedy reassignment, i.e. examining each move of an utterance from one cluster to another, and making the transfers that increase I_{BBN} rather than clustering based on the GLR or CE distances, Cluster purity estimates may be interesting in and of themselves for giving a user some idea of whether a partition is any good.

Multi-level Dendrogram Cutting

Another way to use the purity estimator is in allowing more flexible cutting of the dendrograms. Sometimes an utterance set may consist of two or more groups whose statistical properties are different, for example, males might be more similar to other males than females to other females. In that case, when the dendrogram was being created, the males would cluster before the females, the best horizontal cut of the tree would be too late for the males and too early for the females.

A way to fix this is to first cut the tree at a point that gives only a few clusters, say 5 or 10. Each cluster corresponds to a subtree of the dendrogram. Then each subtree can have its own estimated score curve computed for it. A subtree cut can be found for each subtree that maximizes the subtree estimated score. Then the final partition is the union of the subtree partitions.

This is guaranteed to improve the estimated score over a single horizontal cut. It is not guaranteed to improve the true score, but in tests, some improvement was found.

Nearest Neighbor Purity Estimator

We do this in two stages. First we make an "utterance purity" for each utterance, and then average the utterance purities over all the utterances in a cluster to get the cluster purity. The ***utterance purity*** of a single utterance k with respect to the cluster it belongs to (cluster i) in the following way:

1. Sort all utterances in increasing order of their distance to utterance k.
2. Take the nearest $n_{i\cdot}$ utterances and count the number of them that belong to the cluster i. Call this n_{clus}.
3. Define ρ_k to be the fraction of the first $n_{i\cdot}$ near neighbors that are in cluster i: $\rho_k = n_{clus}/n_{i\cdot}$

Now the estimated purity of the cluster is

$$p'_i = \frac{1}{n_{i\cdot}} \sum_{k \in \text{cluster } i} \rho_k. \quad (9)$$

Theoretical Justification

This estimator gives the correct purity if the (true) clusters are well-enough separated. Specifically, every utterance must be closer to any utterance from the same speaker than to any utterance from a different speaker. In that case, the nearest $n_{\cdot j}$ utterances are from speaker j, for all utterances from speaker j. Recall that the cluster i contains n_{ij} utterances from speaker j. This is the number of near utterances that are in the cluster. Note that $n_{ij} < n_{\cdot j}$. So the point purity for utterance k is

$$\rho_k = \frac{n_{ij}}{n_{i\cdot}} \tag{10}$$

Averaging this over the cluster gives

$$\begin{aligned} p_i' &= \frac{1}{n_{i\cdot}} \sum_{k \in \text{cluster } i} \rho_k \\ &= \frac{1}{n_{i\cdot}} \sum_j n_{ij} \rho_k \\ &= \frac{1}{n_{i\cdot}} \sum_j n_{ij} \frac{n_{ij}}{n_{i\cdot}} \\ &= p_{i\cdot} \end{aligned} \tag{11}$$

Systematic errors

If the true clusters are not well-enough separated, then this estimator can give a purity less than the true purity.

4. SWITCHBOARD EXPERIMENTAL RESULTS

In this section we describe one clustering experiment and display some aspects of the results. The utterance set contained 60 1-minute utterances, 3 each from 20 speakers. Half of the speakers were male, the other half female. They were taken from the Switchboard corpus. For each speaker, 2 utterances were on one channel, and the third was on a different channel.

Data vectors were created from 20 msec frames with 10 msec overlap. Data vectors had 38 elements, 19 cepstra and 19 difference cepstra. Models were GMMs with 128 diagonal covariance terms, trained with 3 iterations of EM.

The distance matrix was the GLR distance, reestimating models after each merge.

Figure 1 shows the value of I_{BBN} at each stage in the dendrogram, with $Q = 1/2$. The two curves denote the true score, calculated using true purity, and the estimated score, calculated using the nearest neighbor purity estimator. It can be seen that at the many-clusters end of the dendrogram, the estimated score is quite close to the true score, but at the few-clusters end, it is completely in error. This is typical.

Table 1 displays the actual clusters obtained by cutting the dendrogram at the stage having the best estimated score. This partition has 29 clusters. The 20-cluster dendrogram cut was better, but a user would have no way to know this.

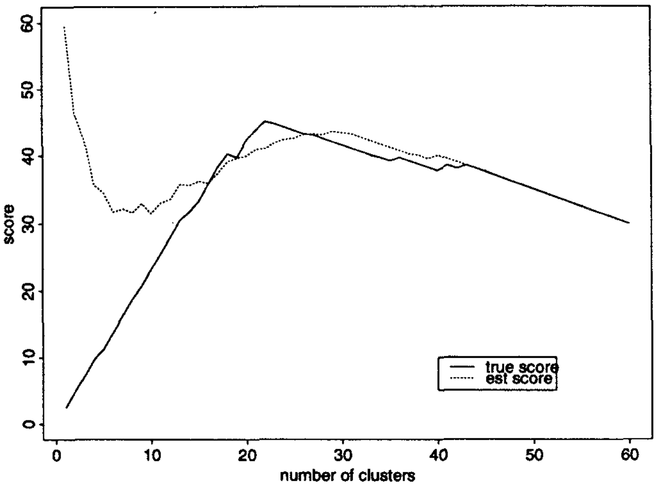

Figure 1. True and estimated clustering performance as measured by the BBN metric (Q=1/2). The estimated purity has a local peak at 29 clusters.

Aa	AE	Bi	B	b	CCc
DDd	Ee	FF	fGGg	HHh	II
JJj	KKk	LL	l	MMm	NNn
OO	o	PPp	Q	Qq	R
R	r	SSs	TT	t	

Table 1. Utterance set partition generated by generalized likelihood ratio and reestimation. The dendrogram was cut at the max-estimate score stage.

Each speaker is denoted by a letter of the alphabet, from A to T. Utterances from the first channel are denoted by capitol letters, and the second channel by lowercase letters. The clusters are not presented in any particular order.

5. CONCLUSION

Summarizing and indexing recorded speech by speaker, topic, etc., is becoming increasingly important given explosion of such data. An approach to clustering speakers by their voices was outlined. We described the metrics used for evaluating our performance and showed results of an experiment on Switchboard data. Cluster purity estimates are a key component of determining good clusters with the metrics and systems presented.

REFERENCES

[1] L. Hubert, P. Arabie "Comparing Partitions", Journal of Classification, 2:193-218(1985)

[2] L. Yanguas, George Doddington, Marc Zissman, Internal DoD Document. (1996)