# Unsupervised Discovery and Training of Maximally Dissimilar Cluster Models

*Françoise Beaufays, Vincent Vanhoucke, Brian Strope*

Google

fsb@google.com, vanhoucke@google.com, bps@google.com

## Abstract

One of the difficult problems of acoustic modeling for Automatic Speech Recognition (ASR) is how to adequately model the wide variety of acoustic conditions which may be present in the data. The problem is especially acute for tasks such as Google Search by Voice, where the amount of speech available per transaction is small, and adaptation techniques start showing their limitations. As training data from a very large user population is available however, it is possible to identify and jointly model subsets of the data with similar acoustic qualities.

We describe a technique which allows us to perform this modeling at scale on large amounts of data by learning a tree-structured partition of the acoustic space, and we demonstrate that we can significantly improve recognition accuracy in various conditions through unsupervised Maximum Mutual Information (MMI) training. Being fully unsupervised, this technique scales easily to increasing numbers of conditions.

**Index Terms**: acoustic modeling, unsupervised learning, clustering, too much data.

## 1. Introduction

Acoustic variability is a well-known problem in speech recognition: an application which works well for some class of users may not be functional for others, and a system that can be used by many in a quiet room may break down in harsh environments. This is especially true of complex systems such as Google Search by Voice [1], or Voice Search for short, where the language model imposes weaker constraints on the recognition search and where, at the same time, the generality of the application (web search) invites users with a wide variety of backgrounds to access the system anytime, from virtually anywhere.

Not surprisingly, a lot of research has been devoted to this issue. The table below illustrates the spectrum of commonly used acoustic modeling techniques, coarsely organized on a continuum between universal models and speaker-dependent models.

| Class of Acoustic Model | Ref. |
|---|---|
| Generic single model | |
| Task-adapted | [2] |
| Gender-dependent | [3] |
| Ensemble models | [4, 5] |
| Cluster-adapted | [6, 7] |
| Speaker-adapted | [8, 9] |
| Speaker-specific | |

Table 1: *List of common acoustic modeling techniques, in increasing order of expressiveness.*

Different tasks find themselves achieving optimal performance at different points on this continuum: most transcription tasks best operate on the lower end of this list, whereas tasks which involve short, independent, transactions such as Voice Search, have found that a better speed / complexity tradeoff can be achieved in the upper portion of this list.

Indeed Voice Search presents a special challenge: interactions are very short, making it difficult to estimate powerful transforms from a single interaction. Accumulating statistics across interactions has its own issues, including the complexity of the machinery needed to estimate, store, and serve in real-time millions of speaker-dependent transforms. Moreover, since users access the application from their mobile phone, it is not clear that speaker characteristics are indeed the most salient factor of variability; perhaps the noise conditions, or channel distorsion, or some combination of these, are more relevant.

The method we propose discovers factors of variability directly from the data, and does so in a hierarchical fashion, allowing an arbitrarily deep investigation into the long tail of acoustic conditions present in the data. Although adaptation techniques could be considered within this framework as well, this paper focuses on an approach more akin to ensemble models.

Interestingly, relatively little attention has been devoted to this portion of the modeling continuum sketched in Table 1, perhaps because ensemble models often partition the input space in ways that quickly result in data starvation and lost performance. We overcome this problem by using a totally unsupervised method which, by freeing us from the need for human transcriptions, gives us access to the virtually unlimited amount of speech that flows back from the Voice Search application into our servers.

This paper explores and expands a technique which is suprisingly effective at segregating data into clusters with maximal acoustic dissimilarity in a Kullback-Leibler sense [5]. The technique relies on representing every utterance as a relatively small sparse vector of Gaussian posteriors which summarizes in compact form the range of acoustic variability encountered in each acoustic state observed during recognition. These vectors are then partitioned iteratively using Vector Quantization (VQ) to yield a tree of increasingly small clusters of utterances which span the acoustic space.

The paper is divided into three main sections. In Section 2, we briefly review the clustering and tree generation algorithms, and describe an efficient distributed implementation using MapReduce [10]. In Section 3, we describe the recognition system, data sets, and baseline accuracy with unsupervised MMI training. In Section 4, we describe some recognition experiments with ensemble models trained for the nodes of a depth-3 clustering tree, and we analyze the nature of two early splits made by the clustering algorithm.

## 2. Clustering And Tree Generation

The goal of this algorithm is to generate a very compact acoustic signature which faithfully represents the deviation between an acoustic model and a given training utterance. A desirable property of this signature is that it should be impervious to phonetic variability in order not to segregate utterances by content but rather solely by acoustic condition. In a typical ASR model, acoustic variability is well summarized by the Gaussian Mixture Model (GMM) distribution at each state. An effective way to measure acoustic similarity between an utterance and a model is to align its frames to the corresponding model states, and measure the divergence between the model and the sample distribution which is generated by the frame observations. This observation is at the core of many Maximum A Posteriori (MAP) speaker identification methods [11]. It was further argued in [5] that this divergence can be approximated using sufficient statistics which exclusively derive from the Gaussian posteriors accumulated over the frame observations. More specifically, if $p_{g,s,m,u}$ is the posterior probability of Gaussian $g$ for state $s$ in model $m$ and utterance $u$, and $w_{g,s,m}$ is the average posterior over the whole training data for model $m$ (denoted $w$ since it's the Gaussian's mixture weight under the Maximum Likelihood (ML) estimation criterion), the Kullback-Leibler divergence between the model and utterance $u$ for state $s$ can be approximated as:

$$\mathcal{D}(s,m,u) = \sum_{g \in s} p_{g,s,m,u} \log \frac{p_{g,s,m,u}}{w_{g,s,m}} \quad (1)$$

Since the total distance over all states should not depend on how many states are observed, the overall similarity measure between utterance $u$ and the model can be averaged over the set $O_u$ of states observed in $u$ to yield:

$$\mathcal{D}(m,u) = \frac{1}{|O_u|} \sum_{s \in O_u} \mathcal{D}(s,m,u) \quad (2)$$

This implies that each utterance can be represented by a sparse 'supervector' $\mathcal{S}(m,u) = [\dots p_{g,s,m,u} \dots]$ of Gaussian posteriors, whose intrinsic dimensionality is the total number of Gaussians in the system. An acoustic model can be represented by a non-sparse vector $\mathcal{W}(m) = [\dots w_{g,s,m} \dots]$ of the same dimensionality. Each entry in that vector is the mixture weight of each Gaussian in the system under the ML assumption. A major advantage of this coarse representation of acoustic variability is that since it is based on relatively sparse vectors of fixed length, it is very well suited to a wide range of data analysis algorithms which can operate very efficiently over datasets containing tens of millions of utterances. In particular, we will show how to apply distributed Principal Component Analysis (PCA) and Vector Quantization (VQ) algorithms to the dataset to cluster the training data.

### 2.1. Extracting the Dominant Source of Variability using Distributed PCA

While computing a full PCA over the entire dataset would be prohibitively expensive, one can use an iterative Expectation Maximization (EM) approach to extract the dominant principal component [12]. This approach parallelizes efficiently using the MapReduce framework, where a "Map phase" performs parallel computations whose intermediate results are then combined under a "Reduce phase" to produce the end results. Starting with a random principal component $\mathcal{P}$:

- For each utterance, compute (Map phase): $\mathcal{P}(m,u) = [\mathcal{P}.(\mathcal{S}(m,u) - \mathcal{W}(m))](\mathcal{S}(m,u) - \mathcal{W}(m))$,
- Accumulate the updated estimate (Reduce phase): $P(m) \leftarrow \frac{\sum_u \mathcal{P}(m,u)}{|\sum_u \mathcal{P}(m,u)|}$,
- Iterate until convergence.

In a 2-class scenario, a single PCA component is sufficient. Otherwise, the above process can be reapplied to yield principal components of lower orders. Note that PCA optimizes the L2 divergence, not the KL divergence we derived this model from. This will be addressed when refining the classifier using VQ clustering. In practice few iterations are required for a good separation.

### 2.2. Dataset Clustering using Distributed VQ

The PCA analysis can be used to boostrap a VQ clustering of the data using simple k-means directly optimizing the divergence in Eq. 2. Again using the MapReduce framework:

- Compute the class label associated with each utterance (Map phase). In the 2-class scenario, the class association can be boostrapped from the estimated Principal Component based on the sign of: $\mathcal{P}(m).(\mathcal{S}(m,u) - \mathcal{W}(m))$. In subsequent iterations, it is obtained by finding the class $m$ which minimizes $D(m,u)$,
- Accumulate the updated Gaussian posteriors for each cluster $m$: $w_{g,s,m} = \frac{1}{N_m} \sum_u p_{g,s,m,u}$, where $N_m$ is the number of utterances assigned to cluster $m$.
- Iterate until convergence.

### 2.3. Generation of a Clustering Tree

Sections 2.1 and 2.2 showed how to split a set of utterances in two maximally distinct subsets. The process can be repeated hierarchically under the form of a binary tree as much as desired ... or as training data lasts. To this effect, an acoustic model must be estimated at each node. This can be made relatively lightweight by training a root-node context-independent model, and only doing a few additional iterations of training with a fixed amount of cluster-specific data at each node. The whole splitting and tree generation process takes only a few hours to generate a depth-5 tree with a few hundred CPUs.

## 3. Recognition System

### 3.1. Recognizer and Metrics

The speech recognition engine is a standard, large-vocabulary recognizer, with PLP features and LDA, decision trees, GMM-based triphone HMMs with variable numbers of Gaussians per state, STC [13] and an FST-based search [14]. ML training is followed by boosted MMI (BMMI) [15]. The language model is a 3-gram model trained from web and Voice Search queries.

Recognition performance is measured in terms of word error rate (WER), and 'normalized' sentence error rate (SER), where small variations such as multiword spacing, dashes, apostrophes, etc are normalized out prior to scoring.

### 3.2. Data Sets

The test set for the experiments reported below consists of 15K utterances collected from the field. Utterances containing side-speech (as marked by the transcribers) were discarded.

The training data consists of a set of 2M transcribed utterances collected roughly at the same time as the test set, and a set of 40M untranscribed utterances. These were selected out

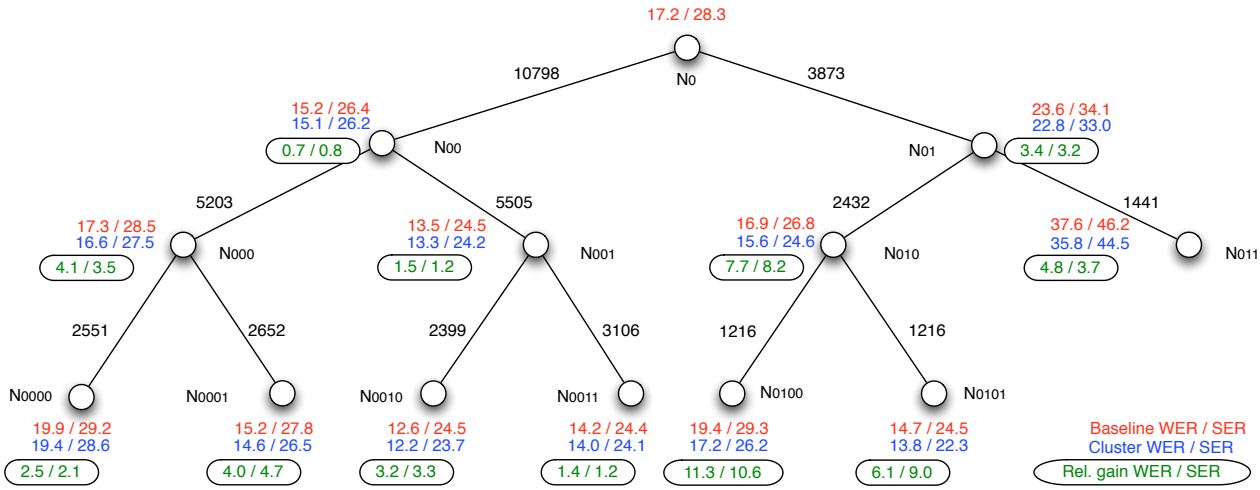


Figure 1: *Unsupervised BMMI performance of baseline and cluster models. The WER/SER pair for the baseline and cluster model is given at each node, with the relative improvement below in a round box. The number of test utterances in a node is listed along the branch leading to that node (total = 14581 utterances). Nodes are named with binary indices, with an additional bit at each tree level.*

of a more recent set of 80M utterances, of which the half with lowest confidence scores was discarded.

### 3.3. Baselines

A first acoustic model was trained from scratch with the 2M transcribed utterances, with ML and BMMI iterations. This model was carefully optimized for accuracy and latency.

Unsupervised training was then performed: we randomly selected 2M utterances out of the untranscribed 40M set, and did two more passes of BMMI training on top of the supervised models. The performance improvement is reported in Table 2. We use this new model as a baseline for our experiments which, as we will describe below, also involve unsupervised training with sets of 2M utterances. It is likely that performance could be further improved by using more untranscribed data, but we chose to keep a fairly lightweight process for experimental purposes.

| Model | WER | SER |
|---|---|---|
| 2M Sup. BMMI | 17.5% | 29.0% |
| 2M Sup. BMMI + 2M Unsup. BMMI | 17.2% | 28.3% |

Table 2: *Supervised and unsupervised BMMI root-node models.*

## 4. Experiments

### 4.1. Recognition Experiments

Using the 2M transcribed training utterances, we trained a clustering tree down to three splits, resulting in eight leaf nodes. (We later trained another tree from untranscribed data and found little difference between the two trees.)

We then percolated the 40M untranscribed training utterances down the tree, and selected at each node a random subset of 2M utterances, which we used to do BMMI unsupervised iterations on top of the root-node supervised model. This process could be interpreted as a form of unsupervised discriminative adaptation, in that the structure and dimension of the model are not affected, but its parameters are re-estimated based on cluster-specific data. At run-time, every test utterance is percolated down the tree (a fairly low-cost operation since it only requires a VQ cluster assignment at each node), and then recognized with the BMMI model associated with the selected leaf node.

We also evaluated a slightly different model training strategy, where each node is adapted from its immediate parent node, rather than from the root node. This gave small but consistent improvements over the previous strategy, possibly because it exposes the model to more data and thus to more errors the algorithm can learn from. These results are shown in Fig. 1.

As indicated in Fig. 1, some clusters benefit highly from retraining (e.g. $N_{0100}$, with a relative SER reduction of almost 11%), others less so. One cluster, $N_{0000}$, is actually better served by its parent node model than by its own (28.3% SER instead of 28.6%). It would thus make sense in a production environment to retain only some of these models. The choice of which ones depends on the goal: improving accuracy where we best can, or minimizing the overall test set error rate. A sensible choice, for example, might be to keep, in addition to the root node model, the models for $N_{0001}$, $N_{0100}$ $N_{0101}$ and $N_{011}$, which would improve the recognition of about 62% of the data, for an average relative SER reduction of a little over 5%.

A related question that we are currently exploring is how to best scale both the baseline and the individual cluster models with more data. The choice we made of working with sets of 2M utterances was dictated in part by our desire to have a fast experimental turn-around, but also because this appears to be the "knee" in the growth curve of our current models. In other words, these models are fairly saturated with data, and doubling or quadrupling the amount of data at the root node gives little incremental improvement, on the order of 0.2 or 0.3%. In the limit, with an infinite set of training data, the question of how much data can be used to train a model is irrelevant. The interesting question instead becomes how to scale the system so it can absorb more data, be it by growing larger models, training more models, or both. So far, our experiments showed that there is significant promise in training specific models for various acoustic conditions, and that "staged" MMI adaptation of cluster models from their parent nodes seem to make good use of the training data.

### 4.2. Interpreting the Clustering Decisions

As described in Section 2, the data clustering is unsupervised: the algorithm discovers at each node the best dimension along which to split the data. It is therefore tempting to try to interpret some of the decisions it made.

In particular, one might expect the first split, $N_0 \rightarrow (N_{00}, N_{01})$, to be related to gender, although the difference of accuracy between the 2 child nodes, about 30% relative, indicates that there must be some other factor as well [16].

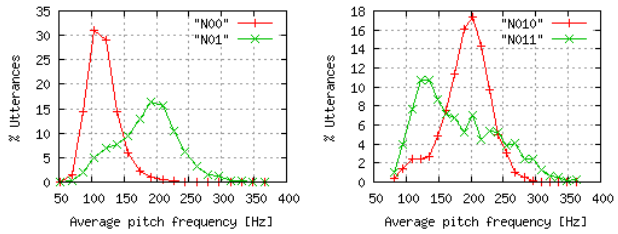


Figure 2: *Pitch histograms: left:* $N_{00}, N_{01}$*, right:* $N_{010}$*,* $N_{011}$.

To explore this hypothesis, we compared the pitch profiles of $N_{00}$ and $N_{01}$ in Fig. 2 (left). In first approximation, the histograms do show a gender bias, with peaks at 100 and 200 Hz. The higher-pitch node, $N_{01}$, however, has a rather broad distribution. The pitch histograms for its child nodes, in the right-hand-side figure, show a mostly-female cluster, $N_{010}$, and a broader cluster seemingly dominated by male voices, $N_{011}$.

To further analyze this broader cluster, we looked at various signals. Utterance loudness, computed as the 95th percentile of the average filterbank energy distribution, was the most revealing (see Fig. 3). Indeed, whereas the first split (left figure) shows little difference in loudness between the two top-level nodes, the second split (right figure) reveals a bimodal distribution for $N_{011}$, indicating a large fraction of loud data. This likely correlates with the rather poor performance of that node (37.6% WER). It is difficult to measure noise and SNR with high-end cellular phones as noise is typically suppressed by the device, but we guess that this loud speech was indeed produced by users speaking over noise.

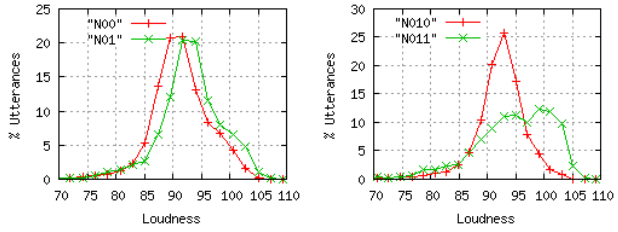


Figure 3: *Loudness histograms: left:* $N_{00}, N_{01}$*, right:* $N_{010}$*,* $N_{011}$.

Interestingly, our analysis also revealed a perplexity difference: 172 for $N_{011}$ vs. 135 for $N_{010}$, itself close to $N_{00}$ which has a perplexity of 145.

Thus the first split seems to separate clean male data from female plus both-gender "formerly noisy" data, which then separates at the next split into female clean data and both-gender harder data. This probably explains why we were more successful training 2 separate models for the child nodes, $N_{010}$ and $N_{011}$, rather than a single one for their parent node, $N_{01}$, whose composition is more heterogenous (2% abs. SER reduction instead of 1%). We expect similar behaviors at deeper tree levels, with recognition improvements showing each time we segregate a fairly homogeneous acoustic condition.

## 5. Conclusions

We described a clustering technique which discovers factors of variability in the acoustic data, and allows a hierarchical partitioning of the input space into maximally dissimilar clusters. We showed that discriminative techniques such as BMMI can be used to model these clusters using only untranscribed data. Because the proposed technique rests completely on unsupervised training, it may be used to explore and hopefully improve arbitrarily deep pockets of difficult conditions in large bodies of data, such as Voice Search's. A small step was taken in this direction by training unsupervised BMMI models for the nodes of a depth-3 clustering tree, and showing that we could significantly improve the recognition performance of about half our test data using such models.

## 6. Acknowledgements

We would like to thank Mitch Weintraub, Yun-Hsuan Sung, and Eugene Weinstein for their insightful discussions and inputs.

## 7. References

[1] http://www.google.com/mobile

[2] Sankar A. and Kannan A., A comprehensive study of task-specific adaptation of speech recognition models, Speech Communication, Vol. 42, #1, 2004.

[3] Murveit, H., Weintraub M., Cohen M., Training Set Issues in SRI's DECIPHER Speech Recognition System, Proc. Workshop on Speech and Natural Language, Hidden Valley, PA, 1990.

[4] Cook G. and Robinson, T., Boosting the Performance of Connectionist Large Vocabulary Speech Recognition, Proc. ICSLP, 1996.

[5] Mao M., Vanhoucke V., Strope B., Automatic Training Set Segmentation for Multi-Pass Speech Recognition, Proc. ICASSP, 2005.

[6] Gales M.J.F., Cluster adaptive training of hidden Markov models, IEEE Trans. on Speech and Audio Processing, Vol. 8, #4, 2000

[7] Kuhn R., Nguyen P., Junqua J.C., Goldwasser L., Niedzielski N., Fincke S., Field K., Contolini M., Eigenvoices for speaker adaptation, Proc. ICSLP, 1998

[8] Anastasakos Y., McDonough J., Schwartz R., Makhoul J., A compact model for speaker-adaptive training, Proc. ICSLP, 1996

[9] Digalakis V., Rtischev D., Neumeyer L., Fast speaker adaptation using constrained estimation of Gaussian mixtures, IEEE Trans. on Speech and Audio Processing, 1995

[10] Dean J. and Ghemawat S., Map Reduce: Simplified data processing on large clusters, Communications of the ACM, Vol. 51, #1, 2008

[11] Campbell, W., Generalized linear discriminant sequence kernels for speaker recognition, Proc. ICASSP, 2002.

[12] Roweis S., EM Algorithms for PCA and SPCA, Advances in Neural Information Processing Systems, MIT Press, 1998

[13] Gales M., Semi-Tied Covariance Matrices for Hidden Markov Models, Proc. IEEE Trans. SAP, May 2000

[14] OpenFst Library, http://www.openfst.org

[15] Povey, D., Kanevsky, D., Kingsbury, B., Ramabhadran, B., Saon, G., Visweswariah, K., Boosted MMI for model and feature-space discriminative training, Proc. ICASSP, 2008

[16] Adda-Decker, M. and Lamel, L., Do speech recognizers prefer female speakers?, Proc. EUROSPEECH, 2005.