TEL-AVIV UNIVERSITY
RAYMOND AND BEVERLY SACKLER
FACULTY OF EXACT SCIENCES
SCHOOL OF COMPUTER SCIENCE

Information Bottleneck for Speech and Speaker Recognition

Thesis submitted in partial fulfillment of the requirements for the M.Sc. degree in the School of Computer Science, Tel-Aviv University

by

Ron M Hecht

The research work for this thesis has been carried out at
Tel-Aviv University
under the supervision of Prof. Naftali Tishby
and Prof. Yishay Mansour

September 2007

Acknowledgments

This M.Sc. thesis took me quite a while. One of the rewards in ending it, is being able to thank all of you that helped me in this interesting voyage. I was fortunate to be supervised by a passionate scientist, Prof Naftali Tishby. I wish to thank him for the stimulating talks, beautiful ideas, encouragement and for his guidance. I wish to thank Prof. Yishay Mansour for his help.
I wish to thank my wife Hadas and my son Ido for the endless support and understanding. A very special thanks goes to my family, my mother, my father, my sister. Lalush and Izo. I wish also to thank, those that helped me in my struggle to finish this thesis, Shalev Itzkovitz, Ruth Aloni-Lavi and Ruth Fridberg.

.

Abstract

Identfying the relevant information in speech signals is an outstanding problem in speech and speaker recognition systems. The notion of relevant information is intimately tied to the goal of the system, e.g. phoneme, word, speaker, language or gender recognition, among others. Here, we explore a novel approach to the extraction of relevant information for speaker recognition and for speech recognition using a principled information theoretic framework - the Information Bottleneck method (IB)[TPB99]. The goal of this approach is to preserve only the relevant information in the speech signal about the speaker's identity or about the phoneme sequence. In this work, we focus on two specific cases of IB - the Gaussian Information Bottleneck (GIB) [CGTW03] and the Agglomerative Information Bottleneck (AIB) [ST00a]. We demonstrate that significantly smaller representations of the signal can be obtained that still capture most of the relevant information about phonemes or speakers. By applying GIB for the speaker recognition task, using the NIST SRE dataset, we were able to boost recognition performance significantly. The information curve enables us to quantify the tradeoff between relevant and irrelevant information and thus the difficulty of the task. Using the Information Bottleneck method seems to have significant implications for building more efficient speech and speaker recognition systems.

Contents

Chapter 1

Introduction

Speech recognition aims at extracting relevant features of a spoken conversation. These features may consist of the speaker identity, the language, the words spoken or others. The notion that is usually directing those systems is the generative notion. According to the generative notion an effort is made to build a model that explains the generation process of the signal.
In order to describe the speech recognition challenges more profoundly and to better understand the use of generative algorithm in speech recognition. I will first divide them into two subgroups. The first subgroup is the ”stationary challenges” and the second group is the ”transient challenges”. The term ”stationary challenges” refers to detection or identification of an aspect of a call that usually does not change throughout a conversion. Language identification, speaker recognition and gender recognition are good examples of members of that group. The language that is used in a conversion can be changed from one language to another; however that is rarely the case. In most of the conversions only a single language is used. The same holds for the speakers' identities and their gender. The second subgroup, the ”transient challenges”, involve the detection of aspects of the voice signal that change rapidly, aspects such as the words spoken. That subgroup is made out of a diverse set of challenges. Challenges that start from the simplest of tasks the VAD - voice activity detection and end at the crown jewel of speech recognition LVCSR Large Vocabulary Continuous Speech Recognition. (The VAD task is usually being referred as a simple one; however in the present of colored noise and in low SNR (Signal to Noise Ratio) that task become much more complicated)
We continue by deepening our understanding in the ”stationary challenges” subgroup. The aspects of a conversion that are part of ”stationary chal-

lenges" subgroup are usually of secondary importance (The speaker usually unintendedly added those aspects to a conversion). Therefore in order to tackle those aspects, one needs to overcome the main source of information in a conversation, the words that were pronounced in it. The commonly used method to achieve that word equalization effect is by apply a Bayesian generative model the GMM (Gaussian Mixture Model) [RQD00]. That approach tackles the equalization effect in an implicit manner. According to it, a conversion is sliced into small segments. Each audio segment is transformed into a point in a high dimensional space. Those points in their turn are treated as a set of i.i.d points. According to the GMM approach, each point is drawn from one of several possible Gaussian distribution. Each of those Gaussian distribution has a different prior weight. All of the mentioned above is formalized in the following equation.

$$\begin{aligned} P\left(O|\lambda\right) &= \prod\nolimits_{i=0}^{T} P\left(o_i|\lambda\right) \qquad (1.1) \\ &= \prod\nolimits_{i=0}^{T} \sum_{g=1}^{G} w_g \frac{1}{\sqrt{2\pi^n |\Sigma_g|^{-1}}} e^{-\frac{1}{2}(o_i-\mu_g)^T \Sigma_g^{-1} (o_i-\mu_g)} \end{aligned}$$

Intuitively, each Gaussian can be seen as representing a part of a phoneme in an implicit manner. The GMM algorithm which is the most common generative algorithm in the "stationary challenges" subgroup, is presented in the last equation.

Dealing with challenges from the second subgroup of the "transient challenges" is usually referred to as a much more complicated and diverse task. In order to tackle it, a variety of generative models and algorithms is used. We will address LVCSR framework as an example for the level of complication those tasks have and as an example to the generative model associated with it. The LVCSR challenge is considered as more complicated one, since in addition to the classification of an event, a detection of the location in the ausio segment is needed as well.

The speech signal has several sources of information. Each source of information represents a different level of the human understanding process. LVCSR system takes advantage of all of those sources in order to improve recognition accuracy. An even better results are achieved by using those sources simultaneously. Those sources can fell under main three categories:

- Acoustic Model That model is the lowest level model among the three models. That model matches between a vowel or a consonant and the sounds associated with them. Unfortunately, that model is not accurate enough to provide us with a reasonable level of recognition.

- Dictionary the second source of information handles the construction of worlds from their basic components (phoneme - vowels and consonants). Each word is represented by the list of phonemes that is used in its pronunciation. Each dialect has different dictionary associated with it and even some of the words have several pronunciations within the same dictionary.
- Language Model Some words have the tendency to follow other words. Understanding and modeling the relations between words fell under the language model responsibilities. One should remember that the language model changes among different dialect and even among different persons.

The most used generative method to model the acoustic level information is the Hidden Markov Model (HHM) [Rab89]. According to the HMM framework, a model is built for each phoneme. An HMM model assumes that each phoneme is made out of several stationary states. Each of those states has a distinct signature in the acoustic space, The observation associated with that state are located in a small area in the feature space. In addition, the probability of moving from one state to another is modeled by a Markov chain. The HMM framework gives us an elegant solution for three interesting questions:

- Efficient likelihood estimation question How do we estimate the likelihood of a set of observation $O = o_1, ..., o_T$ for a given model λ - $P(O|\lambda)$. Lets recall that in the transient challenges the i.i.d assumption does not hold and therefore it is possible that $P(O|\lambda) \neq \prod_{i=0}^{T} P(o_i|\lambda)$.
- Optimal word sequence How do we find the optimal word sequence $W = w_1, ..., w_N$ for a given set of observation $O = o_1, ..., o_T$ and a model λ - $argmax_{w_1,...,w_N} P(w_1, ..., w_N|O, \lambda)$
- Training question How do we estimate correctly the model parameters? $argmax_{\lambda} P(O|\lambda)$

Another limitation that LVCSR system faces is the real time factor. Slower then linear algorithms might prove themselves unpractical. Even maintaining linear time demands $O(t)$ is not enough. The linear factor plays a significant role as well. A great deal of research is being performed in order to reduce the real time demand. That kind of research involves the entire system. Starting from method to lower the feature space dimension, throw pruning unlikely paths in the Viterby Table and ending at a better language model constrains

Over the last paragraphs we have reviewed the work done using generative approach in speech recognition. We have seen it used in a wide variety of transient and stationary tasks. Although it is the most used method in speech recognition, that approach has two main drawbacks. The first drawback is the model itself. The generative process that is described is not the real one, it is only a model and therefore it is suitable only up to a certain amount. The second drawback is the model goal. The model goal is to explain the generation process of a signal, and not to discriminate between two different signals. The Information Bottleneck (IB) method is a novel approach to tackle those drawbacks [TPB99]. The Information Bottleneck method uses the mutual information with the classification targets as a guideline to the improve detection and gain discriminate capabilities. This allows flexibility by tailoring the model or features to the actual task at hand, as the parts of the speech which are indicative of the speaker are not necessarily the parts indicative of the language or content.

In chapter 2 we outline the Information Bottleneck method and demonstrate two commonly used algorithms: Agglomerative Information Bottleneck, and Gaussian Information Bottleneck. In the current work we have decided to focus on two derivatives of the information bottleneck method. These derivatives are quite different from one another. They can be seen as the two ends of a broad variety of spectrum of derivatives. By the time that the first derivative (AIB) handles a discrete feature space the second derivative (GIB) handles a continuous feature space. One of our goals in this work is to demonstrate the vast number of opportunities that the information bottleneck can provide us with. In this work we decided to focus on two different stages in two different applications, Feature extraction stage in the speech recognition application and the supervector stage in speaker recognition. Chapter 3 describes the feature extraction commonly used in speech recognition. Chapter 4 describes the steps in speaker recognition Gaussian Mixture Model, Score Normalization and Supervector/Test Utterance Parameterization. In Chapter 5 we describe a novel application of the Information Bottleneck method for the vector quantization step in the feature extraction. Theoretically speaking it looks like each application (Language Identification, Gender Identification, Speaker Recognition and Speech Recognition) should have a different set of features. However, in reality, the feature extraction stage is almost identical for all applications. Chapter 6 applies the Information Bottleneck for speaker recognition. In one of the last stages of the speaker identification process, each call is mapped into a single point in a high dimensional Euclidean space. During that stage a familiar problem arise, high dimensional data on one side and small number of observation on the other side. One of the methods to tackle the issue is to reduce the observation space. In this chapter we describe the dimension reduction method performed by the Gaussian Information Bottleneck. Chapter 7 summarizes the results and proposes future directions in the application of IB to speech recognition. The IB can be applied in all stages of the speech and speaker recognition tasks. In that chapter we have suggested ways to integrate the IB into all stages of the speech recognition application. Integration can be preformed from the first stage of the feature extraction, through the acoustic model to the language model.

Chapter 2

Information Bottleneck Method

Before we describe the Information bottleneck Method, we will start by introducing the notion of sufficient statistics and relevance. Many of the machine learning tasks suffer from a significant amount of data in each observation. These tasks are complicated to learn since it is hard to make the right deductions in a high dimensional space. One way to tackle this issue is by using a large training set. Another way is to reduce the dimension of each observation. Instead of using the observation as is, a representative of the observation is used. This representative is known as *statistic*. An example of a statistic is the number of heads in a coin toss sequence of N tosses. If one aims at estimating the 'head' probability of a coin, one does not need to use the entire sequence. An estimate can be made as the number of heads divided by the total number of tosses. In this example the estimation will not be improved by using the entire data-set. In such cases where the representative holds all the information for the task at hand the statistic is known as a **sufficient statistic**.

Till now we have described in general the notion of "sufficient statistic". In the following paragraph we are going to formalize it. We shall denote our signal $x \in X$, in the current work our signal is the speech signal or a transformation of that signal. Our goal signal will be denoted by $y \in Y$. In the speech recognition case y might be the speaker identity or the words that were spoken. Finally we will denote our statistic by $\hat{x}$. An ideal $\hat{x}$ is one that contains all the relevant information and only the relevant information. In the case of the speech signal for example, the commonly used vocoders reach a rate of 5-12Kb per second (In this case that is our X). This rate

is achieved by extracting features that are correlated to the position of the vocal tract and the vocal cords. By extracting that information we are able to reconstruct the original signal quite accurately (from the listener point of view) and thus preserve all the information that a listener needs. However if our goal is not to reconstruct the signal but to preserve the information about the word sequence (this is our Y) it might be possible that a better compression exists, an $\hat{x}$ that is sufficient statistics of x for the sequence y. Intuitively, in a second of audio there are about three words and decoding them require significantly less space.
We argued that an ideal $\hat{x}$ is one that contains all the relevant information and only the relevant information. Such $\hat{x}$ are hard to find, and a softer version of our demand must be applied - "Finding $\hat{x}$ that contains as much as possible relevant information and as least as possible **irrelevant information**". Intuitively speaking, all the information can be seen as the sum of the relevant information and the irrelevant information. Therefore the former demand is identical to - "Finding $\hat{x}$ that contains as much as possible relevant information and as least as possible **information**".
At first lets formalize the second part of the phrase " as least as possible information ". Since $\hat{x}$ is produced from x ($x \to \hat{x}$), the amount of information of $\hat{x}$ is the amount of information of x that exists in it. The amount of information that a random variable has on another one is known as the mutual information between the two and is labeled as $I\left(X;\hat{X}\right)$. This can be easily understood if we think about the mapping from x to $\hat{x}$. The volume of elements of x is $2^{H(x)}$ where $H(x)$ is the entropy of x and is defined as followed.

$$H(X) = -\sum\nolimits_{x \in X} P(x)\log P(x) \tag{2.1}$$

For a specific value of $\hat{x}$, the volume of elements of x that are mapped to it is $2^{H\left(X|\hat{x}=\hat{X}\right)}$ where $H\left(X|\hat{x}=\hat{X}\right)$ is defined as follows

$$H\left(X|\hat{x}=\hat{X}\right) = -\sum\nolimits_{x \in X} P(x|\hat{x})\log P(x|\hat{x}) \tag{2.2}$$

By taking the weighted average of the former value we get the average volume of x that are mapped to an element of $\hat{x}$

$$\begin{aligned} H\left(X|\hat{X}\right) &= \sum\nolimits_{\hat{x} \in \hat{X}} P(\hat{x})\, H\left(X|\hat{x}=\hat{X}\right) \\ &= -\sum\nolimits_{\hat{x} \in \hat{X}} P(\hat{x}) \sum\nolimits_{x \in X} P(x|\hat{x})\log P(x|\hat{x}) \\ &= -\sum\nolimits_{\hat{x} \in \hat{X}} \sum\nolimits_{x \in X} P(x,\hat{x})\log P(x|\hat{x}) \end{aligned} \tag{2.3}$$

The difference between the entropy of x with out knowing $\hat{x}$ and the entropy of x when $\hat{x}$ is known, is a lower bound on the amount of information in $\hat{x}$. This value is known as $I\left(X;\hat{X}\right)$ the mutual information between x and $\hat{x}$.

$$I\left(X;\hat{X}\right) = H\left(X\right) - H\left(X|\hat{X}\right) \tag{2.4}$$

We have so far addressed the second part of the phrase. Lets continue by addressing the first part of the phrase - "Finding $\hat{x}$ that contains as much as possible relevant information".
One approach to define relevance of variable $\hat{x}$ is by comparing it to the variable x, that $\hat{x}$ was generated from. The notion behind this approach is that since x contain relevant information and since $\hat{x}$ was generated from x, $\hat{x}$ can't posses more information then x posses. Therefore a good estimation to the amount of relevant information $\hat{x}$ posses can be achieved by using a specific kind of distance measure between $\hat{x}$ and x. Lets denote that distance measure as $d\left(\hat{x},x\right)$ and the total distance between all the pairs of x and $\hat{x}$ as $\sum_{\hat{x}\in\hat{X}}\sum_{x\in X}P\left(x,\hat{x}\right)d\left(x,\hat{x}\right)$. Thus our goal of preserving relevant information is equivalent to minimizing the distance according to d between $\hat{x}$ and x.
By joining the two parts of our phrase we formalize our goal as can be seen in the following equation.

$$R\left(D\right) = \min_{P(\hat{x}|x):\sum_{\hat{x}\in\hat{X}}\sum_{x\in X}P(x,\hat{x})d(x,\hat{x})<D} I\left(X;\hat{X}\right) \tag{2.5}$$

This approach is known as the "Rate Distortion Theory" [CT91] and it suffers from a significant drawback: defining the right distance measure is an extremely complicated task.
A second approach measures $\hat{x}$'s relevant information directly and empirically . Unlike the first approach that compares $\hat{x}$ to x, in that approach we will compare $\hat{x}$ to y, the relevant information itself. The amount of information that $\hat{x}$ has on y is $I\left(\hat{X};Y\right)$ the mutual information between the two. Equation 2.5 can be rephrased as follows:

$$R\left(I_0\right) = \min_{P(\hat{x}|x):I\left(\hat{X};Y\right)>I_0} I\left(X;\hat{X}\right) \tag{2.6}$$

This equation can be change by using Lagrange Multipliers where β is a tradeoff parameter.

$$L\left[P\left(\hat{x}|x\right)\right] = I\left(X;\hat{X}\right) - \beta I\left(\hat{X};Y\right) \tag{2.7}$$

This variational principle is known as the "Information Bottleneck Principle" [TPB99](That principle being referred as the "Information Bottleneck Method" as well). It has been succesfully applied to adress a wide verity of problems [ST00b] [SST$^+$02]. That approach has some significant advantages over the Rate Distortion approach. The first and most important one is that in the information bottleneck method approach, one doesn't need to define the distance measure, the empirical data does that for him. Another advantage lies in the variable we are conducting the comparison with. Instead of comparing $\hat{x}$ to x, hoping for the best, in the information bottleneck method approach we are comparing to y the relevant data itself.
The Information Bottleneck Method has many specific cases and approximations. In the rest of this section we will address two specific cases. Those two cases were picked in order to demonstrate the two sides of a broad range of information bottleneck derivatives. The first case that we will use is the Agglomerative Information Bottleneck (or AIB in short) [ST00a].The AIB deals with a discrete space observation set and is based on a greedy approximation. Our second case is the Gaussian Information Bottleneck (or GIB in short) [CGTW03]. Unlike the AIB, the GIB deals with a continuous observation space in which all the variables are Gaussian variables.

2.1 Agglomerative Information Bottleneck (AIB)

A greedy approximation to the above optimization problem was introduced in [ST00a] . This is done using an agglomerative greedy hierarchical clustering algorithm which merges x points that result in the smallest loss of mutual information about Y. The algorithm starts with a trivial partition of singleton clusters where each element of the data X is in its own cluster (we denote the cluster set by $\hat{X}$). Each cluster induces a conditional probability distribution on the relevant variable $p\left(y|\hat{x}_i\right)$. At each step of the algorithm we merge two conditional distributions in the current partition into a single distribution in a way that locally minimizes the loss of mutual information on the relevant variable Y. Lets denote the two clusters that were merged by i and j. The new cluster that emerges is denoted by k.

$$\min \Delta I\left(Y;\hat{X}\right) = I\left(Y;\hat{X}^{before}\right) - I\left(Y;\hat{X}^{after}\right) = \tag{2.8}$$
$$= \sum_{y\in Y} p\left(y,\hat{x}_i^{before}\right) log\frac{p\left(y,\hat{x}_i^{before}\right)}{p\left(\hat{x}_i^{before}\right)p\left(y\right)}$$
$$+ \sum_{y\in Y} p\left(y,\hat{x}_j^{before}\right) log\frac{p\left(y,\hat{x}_j^{before}\right)}{p\left(\hat{x}_j^{before}\right)p\left(y\right)}$$
$$- \sum_{y\in Y} p\left(y,\hat{x}_k^{after}\right) log\frac{p\left(y,\hat{x}_k^{after}\right)}{p\left(\hat{x}_k^{after}\right)p\left(y\right)}$$
$$= p\left(\hat{x}_i^{before}\right)\sum_{y\in Y} p\left(y|\hat{x}_i^{before}\right) log\frac{p\left(y|\hat{x}_i^{before}\right)}{p\left(y|\hat{x}_k^{after}\right)}$$
$$+ p\left(\hat{x}_j^{before}\right)\sum_{y\in Y} p\left(y|\hat{x}_j^{before}\right) log\frac{p\left(y|\hat{x}_j^{before}\right)}{p\left(y|\hat{x}_k^{after}\right)}$$

The information loss in merging the distributions $p\left(y|\hat{x}_i\right)$ and $p\left(y|\hat{x}_j\right)$ is given by the Jensen-Shannon divergence [Lin91] between the distributions:

$$\begin{aligned}&JS_\pi\left[p\left(y|\hat{x}_i\right)||p\left(y|\hat{x}_j\right)\right] = \\ &\pi_i D_{KL}\left[p\left(y|\hat{x}_i\right)||\pi_i p\left(y|\hat{x}_i\right) + \pi_j p\left(y|\hat{x}_j\right)\right] + \\ &\pi_j D_{KL}\left[p\left(y|\hat{x}_j\right)||\pi_i p\left(y|\hat{x}_i\right) + \pi_j p\left(y|\hat{x}_j\right)\right]\end{aligned} \tag{2.9}$$

where $D_{KL}\lfloor p,q \rfloor$ is the Kullback Leibler divergence:

$$D_{KL}\left[p||q\right] = \sum p\left(x\right) log\frac{p\left(x\right)}{q\left(x\right)} \tag{2.10}$$

and $\pi_i = \frac{p(\hat{x}_i)}{p(\hat{x}_i)+p(\hat{x}_j)}$ for each i. This greedy merging yields hierarchical clusters that provide a simple approximation to the optimal solution of the IB problem in many cases. Throughout the merges we monitor the amount of information preserved by $\hat{X}$ clusters about Y , $I\left(\hat{X};Y\right)$ and compare it to the original value of $I\left(X;Y\right)$. This way we can stop the merging when we reach an acceptable level of relative information loss.

2.2 Gaussian Information Bottleneck (GIB)

In the special case that X and Y are jointly continuous multivariate Gaussian the IB tradeoff takes an especially convenient form, developed in [CGTW03]. In this case, we seek a reduced dimensional representation T of the original variable X. We start the description of GIB with a few basic definitions. For two continuous random variables, X and Y, we will denote the covariance matrices by Σ_x and Σ_y respectively. The cross covariance matrices of X and Y will be denoted by Σ_{xy} and Σ_{yx} , and the conditional covariance matrix will denoted by $\Sigma_{x|y}$, defined as follows:

$$\Sigma_{x|y} = \Sigma_x - \Sigma_{xy}\Sigma_y^{-1}\Sigma_{yx} \tag{2.11}$$

Solving the information bottleneck for general continuous distributions is complicated and involves an iterative solution. However, for the Gaussian case there is an elegant analytic solution to equation (2.7) [CGTW03]:

$$\min_{A,\Sigma_\xi} L = I(X;T) - \beta I(T;Y) \tag{2.12}$$

In this case, the optimal representation T, is the projection of X, as can be seen in the following eqution. Where ξ is an independent component.

$$\begin{aligned} T &= AX + \xi; \\ \xi &\ N(0, \Sigma_\xi) \end{aligned} \tag{2.13}$$

By replacing each of the mutual information components with its definition 2.4, the original equation can be transformed to the following form:

$$L = h(T) - h(T|X) - \beta h(T) - \beta h(T|Y) \tag{2.14}$$

By substituting the entropy with its explicit form for Gaussian variable $h(X) = \frac{1}{2}\log\left((2\pi e)^d |\Sigma_x|\right)$, the equation can be simplified yet further:

$$\begin{aligned} L &= (1-\beta)\log\left(|A\Sigma_x A^T + \Sigma_\xi|\right) \\ &- \log(|\Sigma_\xi|) + \beta\log\left(|A\Sigma_{x|y}A^T + \Sigma_\xi|\right) \end{aligned} \tag{2.15}$$

The optimal representation T, obtained as the solution of (2.15) is the projection of X on eigenvectors of the conditional covariance matrix $\Sigma_{y|x}\Sigma_x^{-1}$. The selected eigenvectors $\nu_1 \ldots \nu_N$ and their relative weights are determined by the tradeoff parameter β, as:

$$A = \left\{ \begin{array}{cc} \left[0^T; \ldots; 0^T\right] & 0 \le \beta \le \beta_1^C \\ \left[\alpha_1\nu_1^T; 0^T; \ldots; 0^T\right] & \beta_1^C \le \beta \le \beta_2^C \\ \left[\alpha_1\nu_1^T; \alpha_2\nu_2^T; 0^T; \ldots; 0^T\right] & \beta_2^C \le \beta \le \beta_3^C \\ \vdots & \vdots \end{array} \right\} \tag{2.16}$$

The left eigenvectors are sorted according to their eigenvalues in an ascending order. The scalars α's and the critical β's, where the dimensionality of the projection increases, are determined by the eigenvalues and eigenvectors as follows:

$$\beta_i^c = \frac{1}{1-\lambda_i} \tag{2.17}$$

$$\alpha_i = \sqrt{\frac{\beta^c(1-\lambda_i)-1}{\lambda_i r_i}} \tag{2.18}$$

$$r_i = \nu_i^T \Sigma_x \nu_i \tag{2.19}$$

Where λ are the eigenvalues of the matrix $\Sigma_{y|x}\Sigma_x^{-1}$.

Chapter 3

Speech Recognition

In the first part of this chapter we describe the human Speech Generation Process. In the second part we explore features used to represent speech and understand the motivation behind them.

3.1 Speech Generation Process

An intuitive way to understand the human speech production process is to look at it as if it were a machine transmitting a signal. Both of them have the same goal, transmitting a signal that contains some relevant information. In the current paragraph we will review the human production mechanism. People produce speech in three major stages:

- At the first stage the air is exhaled from the lungs toward the vocal cords. During this stage the signal gains its energy. However, currently the energy is scattered across a wide range of frequencies. Thus, if it were transmitted, its range would be quite small.
- The human production process overcomes this problem by using the vocal cords (The vocal cords are not always used in the process). By passing through the vocal cords the signal energy is centralized in a small set of energy bands. The vocal cords are made of two leafs that are located in the Larynx and are closing it almost entirely. The flow of air through the cords is causing it to vibrate in a specific frequency. This frequency is known as the pitch frequency or simply as F_0. Actually the cords don't vibrate at a single frequency, but rather in all of the pitch harmonies as well.

- Up to now we have produced what is known as a carrier signal. During the third stage the information that we want to transmit is added. For each phoneme, a different signal envelope is used. The signal envelope is determined by a set of all pole mechanical filters through which the signal passes. The resonance frequencies of those filters are knows as the Formant frequencies (usually there are up to four formants F_1, F_2, F_3, F_4). Those filters are the vocal tract, the nasal and oral cavities. Using the teeth, tongue and lips we change the structure of the cavities causing them to resonate in different frequency each time and thus achieve a different envelope.

By looking at the above transcribed mechanism it is easy to understand the goal of the front end stage of a speech recognition algorithm: slice the signal into small windows. For each window estimate correctly its envelope.
The channel between the speaker and listener changes the signal as well. The effect of that channel is also modeled as a filter. However whereas the filters of the vocal tract change rapidly as we speak, the channel filter remains unchanged throughout a conversation. All the stages that we described above can be described formally by the following equation:

$$P_{pitch}\left(e^{i\omega}\right) H_{vocal\,tract}\left(e^{i\omega}\right) H_{channel}\left(e^{i\omega}\right) = S\left(e^{i\omega}\right) \tag{3.1}$$

However, the rate of changing of the vocal tract filter is much higher than the channel changing rate. Thus we usually model the channel filter as a constant filter. Where S is the Fourier transform of the signal. It is important to mention that the described speech production process is relevant for the Indo-European languages. In other languages, like Mandarin, the pitch plays a much more significant part and in some African languages information is transmitted in the inhalation stage as well.

3.2 Feature Extraction

The goal of the feature extraction stage is to estimate the envelope of the signal. As mentioned in the previous paragraph, the signal should be sliced to small frames and the envelope of each frame should be estimated.
Choosing the optimal frame size entails a tradeoff. The longer the frame is, the more accurate is the signal envelope estimation. However, speech signals are generally not stationary, a characteristic which limits the size of the frame. A rule of thumb says that each third of a phone can be treated

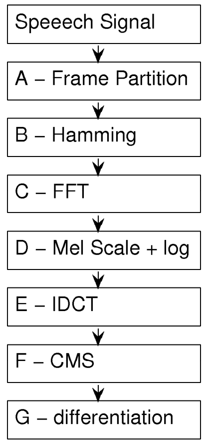

Figure 3.1: The feature extraction process.

as being stationary. By combining that with the average phone length, a 20msec frame is a reasonable choice. In order to overcome uncertainty in the phone actual starting point, a 50 percent overlap between two adjacent frames is used.
The feature estimation process consists of seven stages (figure 3.1). Some of the stages use information only from the current frame and some from adjacent frames as well. The first stage of the frame processing consists of multiplying the signal with a Hamming window. This is done to prevent leakage [OS99]. Only now can we achieve our first estimate of the signal spectrum and its envelope using the FFT algorithm. However estimating the envelope at the current stage suffers from a couple of disadvantages. First, it is not accurate and second, its dimension it extremely high.

A commonly used adjustment to the FFT estimation is known as melody scale [Sht00] or mel-scale in short. It is known that people are less sensitive to change in frequency at the high frequency range and much more sensitive to change in frequency at the low frequency range. This variation in sensitivity is mimicked by a filter bank. The bandwidth of a filter broadens as its middle frequency increases. Those filters have triangle shape as can be seen in figure 3.2. By using this technique, we are able to gain two advantages. The first is a scale that mimics the human ear and the second one is the signal dimension reduction. The dimension is reduced from the FFT size to a new dimension, the number of filters. Usually the signal dimension is reduced from about a hundred to about a twenty.

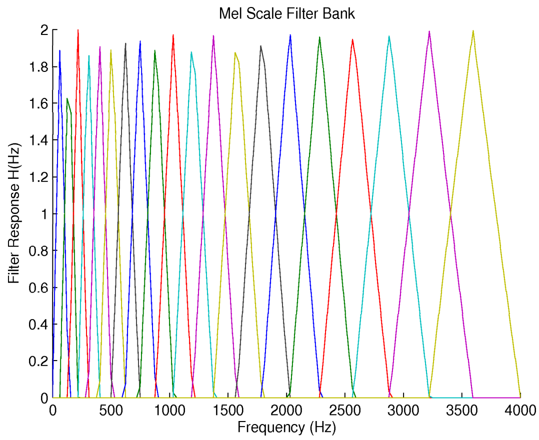

Figure 3.2: Mel Scale Filter Bank. Each Triangle represents a single filter.

The commonly used distance measure in speech recognition is the Cepstral Distance. It measures the distance between two frames and is defined as the Euclidian distance between the Cepstral coefficients of those frames. To better understand the Cepstral Coefficients we will first define a distance measure between the signals of two frames. Here, instead of using the power spectrum, we are using its logarithm. The distance measure between two frames is:

$$d^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} |\log|S\left(e^{i\omega}\right)| - \log|\acute{S}\left(e^{i\omega}\right)||^2 d\omega \tag{3.2}$$

Were the representation of each frames is $\log|S\left(e^{i\omega}\right)|$ According to Parseval theorem the signal can be represented as the following sum and that representation is unitary. The c_n are known as the Cepstral coefficients.

$$\log|S\left(e^{i\omega}\right)| = \sum_{n=-\infty}^{\infty} c_n e^{-i\omega n} \tag{3.3}$$

Lets recall one of the property of an inner product space.

$$\frac{1}{2\pi}\int_{-\pi}^{\pi}\log\left|S\left(e^{i\omega}\right)\right|e^{-i\omega m}d\omega =$$
$$= \frac{1}{2\pi}\int_{-\pi}^{\pi}\left(\sum_{n=-\infty}^{\infty}c_n e^{-i\omega n}\right)e^{-i\omega m}d\omega = c_m \tag{3.4}$$
$$\frac{1}{2\pi}\int_{-\pi}^{\pi}e^{-i\omega m}d\omega = \left\{\begin{array}{ll}1 & m=0\\ 0 & m\neq 0\end{array}\right.$$

By combining the last three equations it can be seen that our goal is equivalent to the Euclidian distance in the Cepstral space.

$$\begin{aligned} d^2 &= \frac{1}{2\pi}\int_{-\pi}^{\pi}\left|\log\left|S\left(e^{i\omega}\right)\right| - \log\left|\acute{S}\left(e^{i\omega}\right)\right|\right|^2 d\omega = \qquad (3.5)\\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi}\left|\sum_{n=-\infty}^{\infty}c_n e^{-i\omega n} - \sum_{n=-\infty}^{\infty}\acute{c}_n e^{-i\omega n}\right|^2 d\omega = \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi}\left|\sum_{n=-\infty}^{\infty}\left(c_n - \acute{c}_n\right)e^{-i\omega n}\right|^2 d\omega = \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi}\left(\sum_{n=-\infty}^{\infty}\left(c_n - \acute{c}_n\right)e^{-i\omega n}\right)\left(\sum_{m=-\infty}^{\infty}\left(c_m - \acute{c}_m\right)e^{i\omega m}\right)d\omega = \\ &= \frac{1}{2\pi}\sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty}\left(\left(c_n - \acute{c}_n\right)\left(c_m - \acute{c}_m\right)\int_{-\pi}^{\pi}e^{i\omega(n-m)}d\omega\right) = \\ &= \sum_{n=-\infty}^{\infty}\left(c_n - \acute{c}_n\right)^2 \end{aligned}$$

Let's look once again at our representation of a frame (3.1) using the cepstral coefficients.

$$\begin{aligned} & \log\left|S\left(e^{i\omega}\right)\right| = \qquad (3.6)\\ = \; & \log\left|P_{pitch}\left(e^{i\omega}\right)H_{vocal\,tract}\left(e^{i\omega}\right)H_{channel}\left(e^{i\omega}\right)\right| = \\ = \; & \log\left|P_{pitch}\left(e^{i\omega}\right)\right| + \log\left|H_{vocal\,tract}\left(e^{i\omega}\right)\right| + \log\left|H_{channel}\left(e^{i\omega}\right)\right| \end{aligned}$$

Each of these parts has a different behavior. The pitch, $\log\left|P_{pitch}\left(e^{i\omega}\right)\right|$, has most of its energy in the high Cepstral coefficients. These coefficients represent the part of the signal that has impressive harmonies. These harmonies are the pitch harmonies.

On the other hand, the vocal tract, $\log|H_{vocal\,tract}\left(e^{i\omega}\right)|$, has most of its energy in the low coefficients. These coefficients represent the part of the signal that doesn't have harmonies. In the vocal tract case, the lack of harmonies usually characterizes the formants and they in their turn are responsible for the envelope. Since our goal was to estimate the envelope, we achieve this by selecting only the lower coefficients. These coefficients contain the information about the formants without information about the pitch. Usually about ten to twenty coefficients are used.
Unlike the two former components of the signal, the channel component, $\log|H_{channel}\left(e^{i\omega}\right)|$, is spread across all the Cepstrum coefficients. Therefore we can't eliminate its effect by choosing a subset of the Cepstrum coefficients. However, since the channel filter is constant throughout the speech segment, by removing the mean of each coefficient its effect is eliminated. That technique is known as Cepstrum Mean Subtraction (CMS in short).
The analysis so far applied to vowels; however for consonants things are a bit different. While as vowels tend to be stationary, consonants have a transient nature. To overcome this, in addition to the Cepstrum coefficients, their derivatives are used as well. Speech recognition features fall under two main categories, discrete features and continuous ones. The features we have been considering so far have been continuous features. Transforming them into discrete features is done by using a Vector Quantization algorithm [ref].

Chapter 4

Speaker Recognition

Most state of the art speaker recognition systems are made of three main stages, a feature extraction stage and a Bayesian model stage followed by a decision stage. The first stage, the feature extraction stage, also known as the front end stage, was described in the previous chapter. During that stage the relevant information is extracted from the raw speech samples. Extraction of relevant features is known to have a significant impact on the system performance. The output of the first stage is a set of t points in a high n-dimensional space $O = \{o_1, \ldots, o_t\}$ $o_i \in \Re^n$. Most speaker recognition systems extract the first 10-20 coefficients of the Mel Frequency Cepstral transform and its first and second derivatives.
The second stage is the Bayesian model stage. During that stage, the likelihood of the input features is estimated $p(x|\lambda)$, where λ denotes a speaker model. The majority of the systems assume a Gaussian Mixture Model (GMM) when estimating the likelihood. The GMM is explained in section 4.1. The output of the second stage, the likelihood, needs to be normalized. There is a variety of normalization techniques for the different aspects of the calls, aspects varying from speaker and segment through channel and noise level to hand set and language. Only after the different normalizations have been performed can the decision take place. Normalization and decision are the last steps and they are reviewed in section 4.2.

4.1 Gaussian Mixture Model

The Gaussian Mixture Model or GMM in short is a Bayesian Model; as such it estimates the probability of a set of observation $O = \{o_1, \ldots, o_T\}$ given a model λ. Under this model we assume that the feature space distribution is stationary for all the frames.

$$P(O|\lambda) = \prod_{i=0}^{T} P(o_i|\lambda) \tag{4.1}$$

The GMM probability distribution function is the weighted sum of a set of G Gaussian distribution functions. Intuitively, this kind of distribution is relevant for speaker recognition. Each Gaussian can be seen as representing a specific speech event (a part of phoneme for example).

$$P(o_i|\lambda) = \sum_{g=1}^{G} w_g \frac{1}{\sqrt{2\pi^n |\Sigma_g|^{-1}}} e^{-\frac{1}{2}(o_i-\mu_g)^T \Sigma_g^{-1}(o_i-\mu_g)} \tag{4.2}$$

In order to reduce the number of model parameters a diagonal covariance matrix is assumed.

$$P(o_i|\lambda) = \sum_{g=1}^{G} w_g \frac{1}{\sqrt{2\pi^n |\Sigma_g|^{-1}}} e^{-\frac{1}{2}\sum_{j=1}^{n} \frac{\left(o_{i,j}-\mu_{g,j}\right)^2}{\sigma_{g,j}^2}} \tag{4.3}$$

Thus the model parameters are $w_g, \mu_{g,j}, \sigma_{g,j}$. The goal is $argmaxP(O|\lambda)$ Estimating the model tends to be a complicated task. First of all there is no analytic solution therefore the EM procedure is used. In addition since data is missing, MAP-estimation is used [RR95].

4.2 Score Normalization

One can model the speech signal as a multiplexing of many signals that originated from different sources like the channel or handset for example. Therefore, the likelihood of segments is affected by these sources, and not solely from the speaker identity. In order to eliminate the effect of these sources, a normalization procedure is applied. The simple normalization procedure is the background normalization. The score of the speaker sp on a segment seg is set to the likelihood ratio between the speaker model and a single model trained from a large number of speakers, the former model is known as background model (bkg in short).

$$\begin{aligned} Score(sp|seg) &= \log \frac{P(seg_i|\lambda_{sp})}{P(seg_i|\lambda_{bkg})} = \\ &= \log P(seg_i|\lambda_{sp}) - \log P(seg_i|\lambda_{bkg}) \end{aligned} \tag{4.4}$$

T Norm is an extension of the BKG normalization. According to this approach the likelihood of the target speaker is normalized by the mean and standard deviation of the likelihoods of reference models. Each of those models is trained from a single speaker.

$$Score_T\left(sp|seg\right) = \log P\left(seg_i|\lambda_{sp}\right) - \sum\nolimits_{j=0}^{SP} \log P\left(seg_i|\lambda_{sp_j}\right) \quad (4.5)$$

While as the last two techniques are for segment likelihood normalization, The Z Norm is for speaker likelihood normalization. It was observed that the models of some speakers are more likely than the models of other speakers. A method to over come this issue is by normalizing the likelihood of a speaker with its likelihood on other segments. A mean and standard deviation is applied.

$$Score_Z\left(sp|seg\right) = \log P\left(seg_i|\lambda_{sp}\right) - \sum\nolimits_{i=0}^{SEG} \log P\left(seg_i|\lambda_{sp}\right) \quad (4.6)$$

In addition to these three procedures there are quite a significant amount of normalization procedures. However, these three procedures are the most commonly used and they are the most significant ones from the performance point of view.

4.3 Supervector / Test Utterance Parameterization (TUP) model

The field of speaker recognition has developed significantly over the last few years. A most significant improvement was the GMM supervectors (also known as Test Utterance Parameterization TUP [ABA04]), which is one of the methods that enables us to represent a speaker as a point in a high dimensional Euclidean space [CCB06]. Unfortunately, the estimation of this representation tends to be very noisy, and it is desirable to eliminate the noise while preserving the relevant information about the speaker. This can be achieved by dimension reduction[AIB05][KBOD05].
Our baseline system is based on GMM supervectors, which are the extension of the commonly used GMM-UBM (Gaussian Mixture Model Universal Background Model) approach [RQD00]. Unlike the original approach, there is a full symmetry between the train and test segments. Instead of evaluating the GMM distributions just for the train segments, and calculating the likelihood of the test segments, we estimate GMMs for both. The likelihood is approximated by the KL distance between the two models, without using

the samples directly. To calculate the KL distance efficiently, each GMM is transformed into a point in a high-dimensional space (the dimension is the product of the number of Gaussians and the feature dimension) as in Eq. 4.7:

$$x_{g \cdot d+j} = \sqrt{w_g}\frac{\mu_{g,j}}{\sigma_{g,j}} \tag{4.7}$$

Where d is the feature space dimension, g indexes the Gaussians and j indexes the dimension. In this space, the Euclidean distance between two points provides a good approximation to the KL distance between the two original GMMs.
The front end of our system is based on MFCC feature extraction [ETS]. It extracts 13 cepstral coefficients and their first derivatives every 10ms. We are estimating the coefficient on a window of 25ms. The selection of the relevant frames is done by energy-based Voice Activity Detection (VAD). Mean and variance normalization is applied to the features. We used a single Universal Background Model (UBM); we didn't train different UBMs for different channels or different handsets. For normalization purposes we used two types of score normalizations: T-norm [ACLT00] and Z-norm.

Chapter 5

AIB oriented VQ

We apply the Information Bottleneck method to a commonly used representation of speech signals the Mel-cepstrum [Sht00]. We perform a vector quantization of the mel-cepstrum feature set, and use the resulting quantization as our initial data-space. We then extract compact representations of the data-space that preserve information about our target variables via clusters obtained through the agglomerative information bottleneck procedure (AIB) [ST00a]. This procedure takes into account the ultimate goal of recognition by iteratively merging cluster pairs which lead to the minimal reduction of mutual information with the target variable, phonemes in one case and speakers identity in the other. We show that this procedure efficiently preserves the relevant information for either phoneme recognition or speaker recognition, and makes it possible to use a much smaller representation without reducing recognition relevant information. We show that starting with a higher number of codebook vectors and reducing them using the AIB preserves significantly more relevant information than starting with a quantization of the same size. This has obvious implications for designing efficient speech recognition systems.

5.1 Relevant Speech Quantization

Speech is a complex signal with a high entropy rate. It contains ample information about the various components of its acoustic structure, such as the spoken language, specific utterances, identity of the speaker, his/her physical conditions, mood, etc. Yet most speech processing algorithms employ standard front-end processing that eliminates much of the entropy of the signal, but do it in a universal task independent way. Such a representation is bound to contain irrelevant information which thus reduces the efficiency and performance of the recognizer that follows. It is therefore a fundamental problem in speech technology to filter out only (or mostly) the task-relevant components of the signal. This is a difficult problem, as the relevant acoustic distortion measure is unknown and involves both complex perceptual and linguistic variables. Our approach to this problem is to utilize the available tagging of the signal (phonemes or speakers) to guide the selection of its representation.

We begin with the joint representation of the speech signal, denoted by X, and its relevant labeling signal whether phonemes, speaker identity, or other attributes denoted here by Y. We then estimate their joint distribution, $P(x, y)$. The amount of relevant information in X about Y is determined by Shannons mutual information between the variables $I(x; y)$.

Our goal is to find a compact representation of X, denoted by $\hat{X}$, that on the one hand compresses it by minimizing the mutual information between them $I(x; \hat{x})$ and on the other preserves as much as possible the information about Y $I(\hat{x}; y)$. To this end we minimize the Information-Bottleneck variational functional,

$$L[P(\hat{x}|x)] = I(x; \hat{x}) - \beta I(\hat{x}; y) \tag{5.1}$$

with respect to the (stochastic) mapping $P(\hat{x}|x)$, where β is a positive Lagrange multiplier .

5.2 Database and Feature Extraction

Our database was taken from the OGI multi-language phonetically transcribed corpus [MCO92]. We used files from the English story part. The database included over 100 PCM wave files of different speakers, each about one minute long. We used the standard Mel-cepstrum speech feature extraction (figure 5.1) [Sht00]. For each file the speech signal was divided into frames of 20 msec long. Each frame was multiplied by a Hamming window, and Fourier transformed. The power spectrum coefficients were mel-scaled. The log-mel-scaled coefficients passed through iDCT and cepstrum mean subtraction. The resulting features are a set of 16 coefficients of cepstrum values and their temporal derivative between adjacent frames. Each set of coefficients is tagged by the label phoneme (Y_1) and speaker (Y_2).

As the mutual information for the full continuous cepstral space is difficult to estimate, we performed vector quantization for discretizing the space. We used a training set taken from the database at different stages of the feature

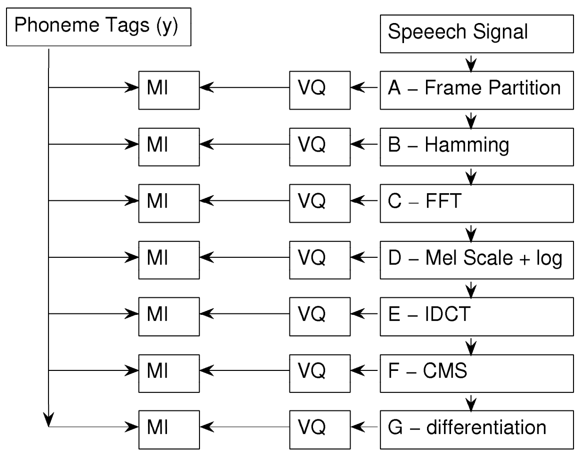

Figure 5.1: The feature extraction process.

extraction process to produce codebooks of different sizes, N, which henceforth serve as our compressed variable X. We then calculated the empirical mutual information between the tagged phoneme Y_1 and the discretized speech features X.

5.3 Results

Our first interesting observation is that the mutual information of X and Y_1 increases at each stage of the standard feature extraction procedure (as shown in figure 5.2), despite the fact that each stage reduces entropy and thus discards information. This can be explained by the facts that first, the information discarded is less relevant for phoneme identification as it should be and second, after quantization we are left with a more efficient representation. This suggests the code-book obtained from the final processing stage, G, as the best choice of data-space for our relevant compression procedure.

The application of the agglomerative information bottleneck procedure leads to the desired result. For a given number of clusters, the amount of mutual information between the clusters and the target variable Y is much higher when starting from a larger codebook and then reducing its size using the AIB algorithm (figure 5.3). As an example, the mutual information between a code book of size $N = 256$ extracted directly from the vector quan-

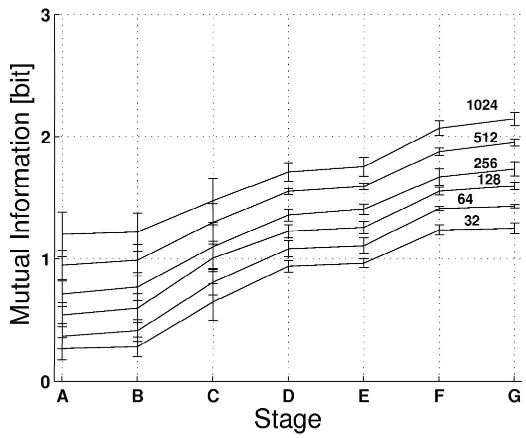

Figure 5.2: The mutual information between the VQ codebook and the phonemes. Note the monotonic increase of the relevant information along the stages as well as with the VQ size. Error bars are for 7 cross-validation batches.

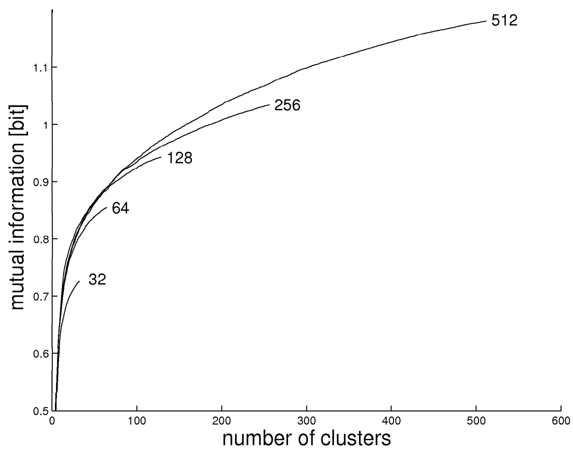

Figure 5.3: Mutual information between the clusters ($\hat{X}$) and the phonemes (Y_1) at different stages of the AIB algorithm. Shown are results for N=32, 64, 128, 256, and 512

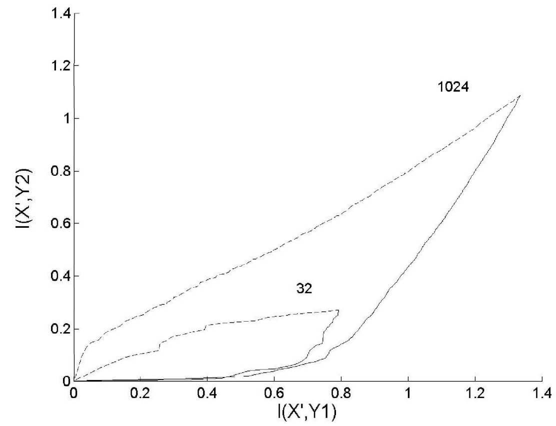

Figure 5.4: Mutual information between clusters and phonemes, $I(\hat{x}; y_1)$ vs. the mutual information between clusters and speakers $I(\hat{x}; y_2)$. Results shown are for initial data-spaces of sizes $N = 1024$ and $N = 32$. The results shown are for the AIB applied to the phonemes (dashed) and the AIB applied to the speakers (solid).

tization stage described in section 2.1 is lower than the mutual information between the target Y and a codebook of size 256 obtained by AIB applied to an initial data-space of 512 vectors. An important advantage of the IB algorithm is flexibility in choosing the relevant target. When performing the IB algorithm for purposes of maximizing the mutual information with the tagged phoneme Y_1 we observe a large range in which the number of clusters decreases without a considerable decrease in the mutual information about Y_1 (figure 5.4). On the other hand, the mutual information with the speaker target (Y_2) drops rapidly (see figure 5.4). When applying the IB algorithm with the goal of maximizing the mutual information about the speaker's identity Y_2, we observe a larger range where $I(\hat{x}; y_2)$ barely drops, while $I(\hat{x}; y_1)$ decreases rapidly (figure 5.4). Interestingly, this is more noticeable when starting from a smaller data-space size N. The IB indeed captures the relevant information about the target, whether phonemes or speaker, as it is designed to do. The difference between the phoneme-cluster information and the speaker-cluster information is about 50 percent of the mutual information, which (roughly) predicts a similar improvement in recognition performance for each task (respectively). Moreover, the analysis of the resulting clusters can identify the speaker vs. phoneme dependent components of the Mel-cepstra, identifying them as clearly different components of the signal.

Table 5.1: *Entropies and Mutual Information (bits) for the phonemes and speakers for the initial size $N = 1024$ VQ and after applying the AIB algorithms for phonemes and speakers with reduced $N = 32$ clusters.*

	Entrophy	VQ 1024 MI	MI after AIB to speakers	MI after AIB to Phonemes
Phonemes	3.58	1.33	0.35	0.82
Speaker	4.0	1.08	0.35	0.15

Chapter 6

GIB oriented Speaker Recognition

In the previous chapter we demonstrated the advantages of the Information Bottleneck method for extracting a discrete feature set for speech signals. We showed that the IB method selects features that encapsulate the relevant information for the speech property at hand, be it phoneme or speaker recognition. In this chapter we will apply the Information Bottleneck method for another important step in the speech processing scheme - dimensionality reduction of the supervectors in the TUP method. This will be demonstrated for the problem of speaker recognition.
In chapter 4 we described the TUP procedure, which maps the GMM parameters to a point in a high dimensional space. In speaker recognition systems, each speaker is mapped to one point in this space. The ultimate speaker recognition consists of selecting the GMM point with the smallest Euclidean distance from the GMM point of the sample. In this chapter we show that the IB method is very efficient in reducing the dimensionality of the TUP space. We compare and contrast the GIB solution to another method for dimensionality reduction - Fisher's Linear Discriminant Analysis (LDA) [DHS01]. We show that the IB dimensionality reduction increases the success rate of speaker recognition.

6.1 GIB for speaker recognition

The optimal projection according to the GIB algorithm consists of the eigenvectors of the matrix $\Sigma_{x|y}\Sigma_x^{-1}$ (equation (2.16)). In our case the X vectors are the GMM supervectors of the voice segments (a 256*26 dimensional vector). The Y vectors are the GMM supervectors of the speakers. The GMM supervector of a speaker was set to be the average vector of all the segments that were generated by the speaker:

$$y_s = \frac{1}{n_s} \sum_{i=1}^{n_s} x_{si} \tag{6.1}$$

y_s is the GMM supervector of speaker s
x_{si} is the GMM supervector of segment i of speaker s
n_s is the number of segments made by speaker s
It can be seen in the following equation that under these conditions the cross covariance matrices Σ_{xy}, Σ_{yx} are equal to the speaker covariance matrix Σ_y.

$$\Sigma_{xy} = \Sigma_{xy} = \frac{1}{SP} \sum_{s=1}^{SP} \left(\frac{1}{n_s} \sum_{i=1}^{n_s} y_s x_{si}^T \right) = \tag{6.2}$$
$$= \frac{1}{SP} \sum_{s=1}^{SP} \left(y_s \left(\frac{1}{n_s} \sum_{i=1}^{n_s} x_{si}^T \right) \right) = \frac{1}{SP} \sum_{s=1}^{SP} y_s y_s^T = \Sigma_y$$

By using the equation (6.2) we can simplify the optimal projection equation.

$$\begin{aligned} \Sigma_{x|y}\Sigma_x^{-1} &= \left(\Sigma_x - \Sigma_{xy}\Sigma_y^{-1}\Sigma_{yx}\right)\Sigma_x^{-1} = \\ &= \left(\Sigma_x - \Sigma_y\Sigma_y^{-1}\Sigma_y\right)\Sigma_x^{-1} = \left(\Sigma_x - \Sigma_y\right)\Sigma_x^{-1} \end{aligned} \tag{6.3}$$

Due to the scarcity of training data, a full covariance matrix cannot be estimated properly. In order to overcome this problem and to simplify the estimation, we assumed that the two matrices are block-diagonal [NA06]. This is equivalent to assuming that there is no correlation between values of x that originate from different cepstral coefficients and no correlation between an MFCC and its derivative. Therefore, the size of each block was the number of Gaussians. In addition, we used a fewer Gaussians than the commonly used 1024 or 2048 Gaussians, specifically 256.

6.2 Comparing the GIB and the LDA algorithms

In the current section, we will compare the LDA method to our current version of GIB, in which Y (the speaker vector) is defined as the average of all the segments of the speaker. The goal of LDA is to find the eigenvectors of the matrix $S_W^{-1} S_B$ that have the highest eigenvalues. S_B is the between-class scatter matrix, and is defined as Σ_y . S_W is the within-class scatter matrix and we will now prove that it is equal to $\Sigma_{x|y}$.

$$\begin{aligned} S_w &= \frac{1}{SP}\sum_{s=1}^{SP}\left(\frac{1}{n_s}\sum_{i=1}^{n_s}\left(y_s - x_{si}\right)\left(y_s - x_{si}\right)^T\right) = \\ &= \frac{1}{SP}\sum_{s=1}^{SP}\left(\left(\frac{1}{n_s}\sum_{i=1}^{n_s}\left(x_{si}x_{si}^T\right)\right) - y_s y_s^T\right) = \\ &= \left(\Sigma_x - \Sigma_y\right) = \Sigma_{x|y} \end{aligned} \tag{6.4}$$

Thus, the optimal projection according to the LDA algorithm consists of the eigenvectors of the matrix $\Sigma_{x|y}\Sigma_x^{-1}$ with the *highest* eigenvalues.
We have shown in equation 6.3 that in our version of GIB, the goal is to find the eigenvectors of the matrix $\Sigma_{x|y}\Sigma_x^{-1} = I - \Sigma_y\Sigma_x^{-1}$ that have the *smallest* eigenvalues.

Table 6.1: *LDA and GIB matrices.*

Algorithm	Matrix
LDA	$\Sigma_{x\|y}^{-1}\Sigma_y\vec{\nu} = \lambda\vec{\nu}$
GIB	$\vec{\omega}^T\Sigma_{x\|y}\Sigma_x^{-1} = \gamma\vec{\omega}^T$

An eigenvector and its corresponding eigenvalue of the LDA matrix are denoted by $\vec{\nu}$,λ and those of the GIB matrix are denoted by $\vec{\omega}$,γ .

$$GIB: \quad \vec{\omega}^T\Sigma_{x|y}\Sigma_x^{-1} = \gamma\vec{\omega}^T \tag{6.5}$$

$$\Sigma_x^{-1}\Sigma_{x|y}\vec{\omega} = \gamma\vec{\omega}$$

$$\begin{aligned} LDA: \quad & \Sigma_{x|y}^{-1}\Sigma_y\vec{\nu} = \lambda\vec{\nu} \\ & \Sigma_{x|y}^{-1}\left(\Sigma_x - \Sigma_{x|y}\right)\vec{\nu} = \lambda\vec{\nu} \\ & \left(\Sigma_{x|y}^{-1}\Sigma_x - I\right)\vec{\nu} = \lambda\vec{\nu} \\ & \left(\Sigma_{x|y}^{-1}\Sigma_x\right)\vec{\nu} = (1+\lambda)\,\vec{\nu} \\ & \frac{1}{1+\lambda}\left(\Sigma_{x|y}^{-1}\Sigma_x\right)\vec{\nu} = \vec{\nu} \end{aligned} \tag{6.6}$$

By placing the eigenvector $\vec{\nu}$ of the LDA algorithm in the GIB equation, we get:

$$\Sigma_x^{-1}\Sigma_{x|y}\vec{\nu} = \Sigma_x^{-1}\Sigma_{x|y}\vec{\nu}\left(\frac{1}{1+\lambda}\left(\Sigma_{x|y}^{-1}\Sigma_x\right)\vec{\nu}\right) = \frac{1}{1+\lambda}\vec{\nu} \tag{6.7}$$

Therefore any eigenvector of the LDA matrix, is also an eigenvector of the GIB matrix, with some other eigenvalues. However, since there is a monotonously decreasing mapping $\lambda \longrightarrow (1+\lambda)^{-1}$ between the eigenvalues, choosing the largest for LDA, is exactly the same as choosing the smallest for GIB. Therefore, both methods will select exactly the same eigenvectors for a given reduced dimension.

6.3 Results

Our experiments were conducted on the male part of the common task of the NIST SRE 2005. Evaluation of the background model was done from the male training segments of common task of the NIST SRE 2004 [NIS04][NIS05]. A total of about 250 segments from about 100 different speakers were used for the background training. The models of the target speakers were estimated by using MAP adaptation from the single background model. As was mentioned in chapter 4, in the GMM supervector approach each of either the train or test segments is represented by a point in a high dimensional space. This creates an appealing symmetry between the train set and the test set. Similarly there is now an equivalence between T -norm and Z -norm, as both normalizations are done in the same space. All the segments (train and test) of the male part of the common task of the NIST SRE 2004 were used to compute the T -norm and Z -norm of each sample. There are about 700 segments that were generated by about 100 different speakers. These segments were used in the evaluation of the three matrices $\left(\Sigma_x, \Sigma_y, \Sigma_{x|y}\right)$ as well.

We first conducted three experiments. The first one was a baseline experiment, in which the original representation of the model space was left untouched, i.e. no dimensionality reduction transformation was applied to it. In the second and third experiments, the LDA and GIB transformations were applied respectively to the model space. In both transformations, all the eigenvectors were used. The results of those experiments can be seen in figure 6.1 [MDK+97]. We performed two additional sets of experiments, one using the GIB method and the other using LDA. In these experiments, we measured the degradation in recognition performance due to the reduction of the number of eigenvectors that were used. The results of these sets of experiments can be seen in figures 6.2 and 6.3. In figure 6.2 the performance measure is the equal error rate (EER) and in figure 6.3 the performance is measured using the detection cost function (DCF). In both figures, the baseline system always uses the original GMM supervector, without any kind of

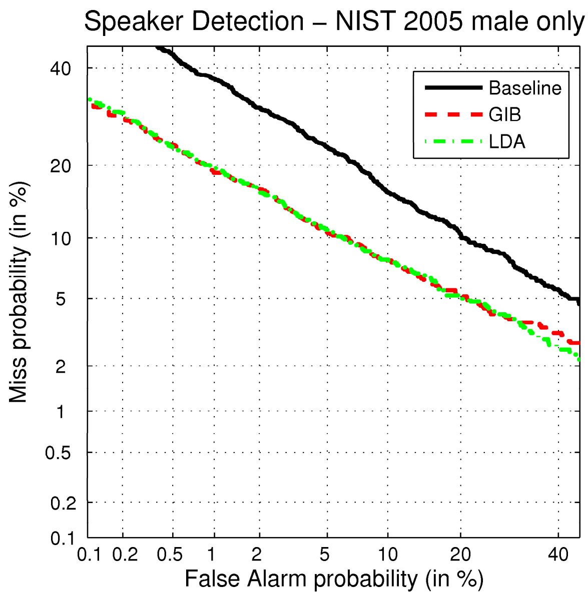

Figure 6.1: DET Curve of three systems: the best GIB system, the best LDA system and the baseline system

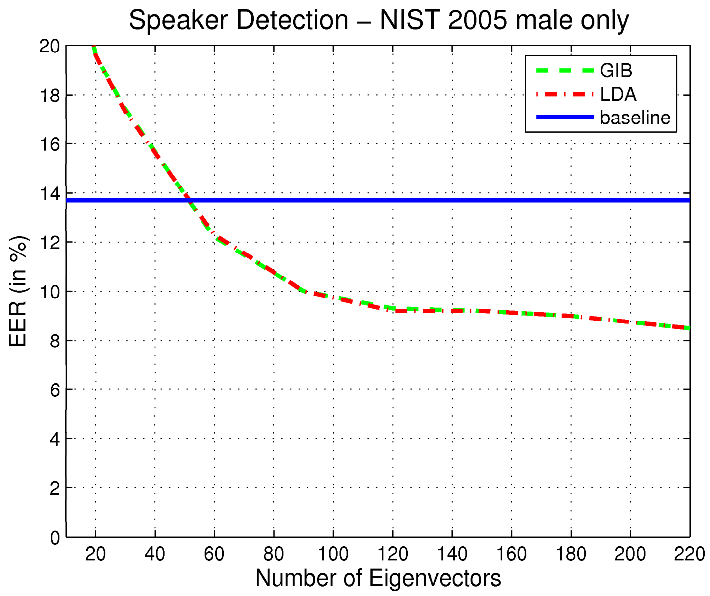

Figure 6.2: EER vs. the new dimension of each block

dimensionality reduction. Therefore, the x-axis is irrelevant for the baseline, and it is plotted as a horizontal line for purely aesthetic reasons.

6.4 Information Curve

A better understanding of the recognition process can be gained by examining the information curve the dependence of the relevant information, $I(T;Y)$, on the representation complexity $I(T;X)$. The easier the task - the steeper this curve. It can be seen that even when using a small dimension (80), the majority of the information about the speaker can be preserved. Furthermore, there is a correlation between the mutual information about the speaker and the detection performance. Still, the fact that there is no clear knee in the curve means that information on the speaker exists on all scales and all dimensions.

We demonstrated that the IB method in general [For05] [ST00a] [HT05] and the GIB method in particular are efficient principled methods for improving the performance of speaker recognition. The IB, through its discriminative nature, provides a subspace that contains as much information about the

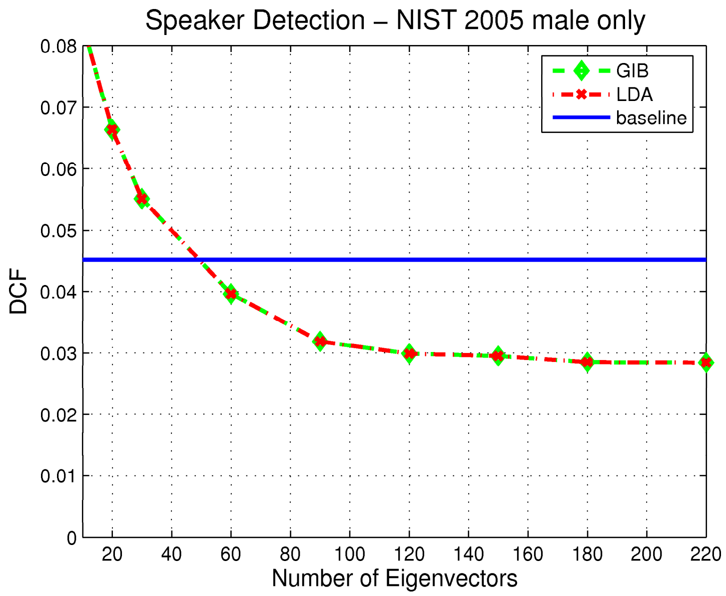

Figure 6.3: DCF vs. the new dimension of each block

speaker as possible, while minimizing irrelevant acoustic information. We also showed that the classical LDA is a special case of GIB, where the relevance variable Y is the average of the X vectors for each speaker. Even for this simple case, where the two variables (X and Y) are not from truly different sources, we obtain a significant improvement in performance.

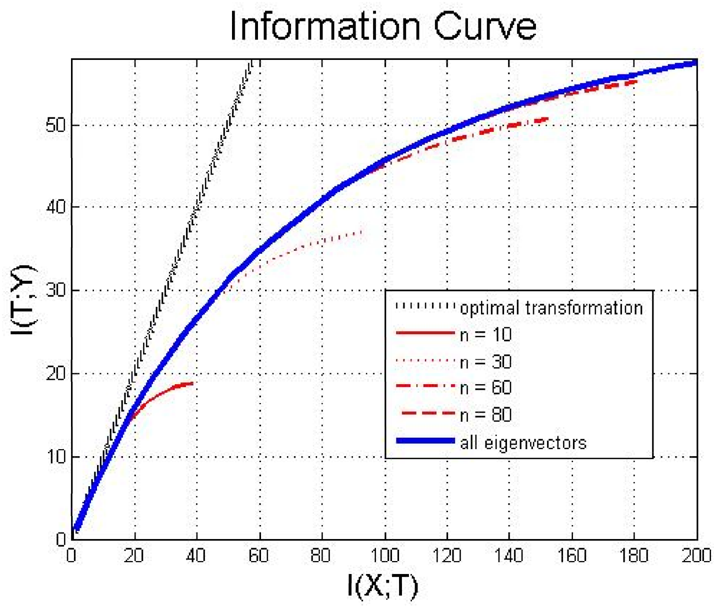

Figure 6.4: Speaker Recognition Information curve. n the number of eigenvectors per block

Chapter 7

Conclusion

In this thesis we demonstrated that using the information bottleneck method for speech recognition offers an improvement in recognition performance. We chose two applications at the two ends of the recognition scheme. In the first application we used AIB to cluster the discrete space of the front end stage of the speech recognition task. We have shown improvement in preserving the mutual information between the signal representation and its label, be it phoneme spoken or speaker identity. In the second application we used GIB to reduce the dimension of the TUP representation of the continuous GMM space. We have shown improvement in recognition performance as measured by EER and contrasted and compared to another dimensionality reduction method - LDA.

Chapter 8

Future Work

In this chapter I will discuss different directions in which this work can be extended. All the directions that will be addressed fall under the LVCSR (Large Vocabulary Continuous Speech Recognition) framework [WOVY94]. Currently this is considered the most complicated task in speech recognition. The goal is to recognize the word spoken in spontaneous speech using a dictionary of tens of thousands of words.
An LVCSR system consists of several components. Each of these components is equivalent to a different level of the speech understanding process. The information bottleneck method can be used to improve the following components:

- The first component is the feature extraction process. In this thesis we have used the AIB algorithm for a discrete feature space. However, state of the art systems use a high dimensional continuous feature space instead (The cepstrum feature space is commonly used). This feature space is reduced today by using HLDA (Heteroscedastic Linear Discriminant Analysis [Gal00]). The IB can become an alternative to the HLDA.

- Another component in which the IB can contribute significantly is the phoneme model level. The two commonly used phoneme models are the left to right HMM model and the left to right with jumps HMM model [RPKM$^+$05]. In these models the feature space in each state is modeled by a Gaussian mixture. The parameters of these models are usually estimated using the maximum likelihood criterion. The IB can become an alternative discriminative training criterion. Some models are estimated using discriminative criteria (MCE, MPE, MMI

[RPKM$^+$05]), however the IB may hold advantages over these methods by facilitating reduction in the number of Gaussians.

- An important part in the training of a speech recognition model is known as the tying stage. During this stage, models that are similar are merged to a single model. Here, as in the phoneme model level, the tying mechanism is based on the maximum likelihood criterion. The IB can be assessed as an alternative criterion.

Bibliography

[ABA04] H. Aronowitz, D. Burshtein, and A. Amir. "speaker indexing in audio archives using test utterance gaussian mixture modeling". In *Proceedings of International Conferences on Spoken Language Processing (ICSLP) - Interspeech*, 2004.

[ACLT00] R. Auckenthaler, M. Carey, and H. Lloyd-Thomas. score normalization for text-independent speaker verification systems. *Digital Signal Processing*, 10, 2000.

[AIB05] H. Aronowitz, D. Irony, and D. Burshtein. "modeling intra-speaker variability for speaker recognition". In *Proceedings of Interspeech*, 2005.

[CCB06] M. Collet, D. Charlet, and F. Bimbot. "a weighted measure of similarity for speaker tracking". In *Proceedings of Odyssey*, 2006.

[CGTW03] G. Chechik, A. Globerson, N. Tishby, and Y. Weiss. "information bottleneck for Gaussian variables". In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2003.

[CT91] T. M. Cover and J. A. Thomas. *"Elements of Information Theory"*. "John Wiley & Sons, Inc", 1991.

[DHS01] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. John Wiley & Sons, Inc, 2001.

[ETS] ETSI. speech processing, transmission and quality aspects (stq) distributed speech recognition; front-end feature extraction algorithm; compression algorithms.

[For05] O. Forkosh. "multi level hierarchical mixture model". Master's thesis, The Weizmann Institute of Science, 2005.

[Gal00] Mark J. F. Gales. Factored semi-tied covariance matrices. In *NIPS*, 2000.

[HT05] R. M. Hecht and N. Tishby. "extraction of relevant speech features using the information bottleneck method". In *Proceedings of Interspeech*, 2005.

[KBOD05] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel. "factor analysis simplified". In *Proceedings of IEEE International Conference Acoustics, speech, signal processing (ICASSP)*, 2005.

[Lin91] J. Lin. "divergence measures based on the shannon entropy". *IEEE Trans. on IT*, 37:145–150, 1991.

[MCO92] Y. K. Muthusamy, R.A. Cole, and B.T. Oshika. "the ogi multi-language telephone speech corpus". In *Proceedings of International Conferences on Spoken Language Processing (ICSLP)*, 1992.

[MDK+97] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki. "the DET curve in assessment of detection task performance". In *Proceedings of Eurospeech*, 1997.

[NA06] E. Noor and H. Aronowitz. "efficient language identification using anchor models and support vector machines". In *Proceedings of the Speaker and Language Recognition Workshop. IEEE*, 2006.

[NIS04] NIST. The nist year 2004 speaker recognition evaluation plan, 2004.

[NIS05] NIST. The nist year 2005 speaker recognition evaluation plan, 2005.

[OS99] A.V. Oppenheim and R.W. Schafer. *"Discrete-Time Signal Processing"*. Prentice-Hall, 1999.

[Rab89] L. Rabiner. "a tutorial on hidden markov models and selected applications in speech recognition". In *Proceedings of IEEE*, 1989.

[RPKM+05] S. Matsoukas R. Prasad, C.-L. Kao, J. Ma, D.-X. Xu, T. Colthurst, O. Kimball, and R. Schwartz. "the 2004 bbn/limsi 20xrt english conversational telephone speech recognition system". In *Inerspeech*, 2005.

[RQD00] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn. "speaker verification using adapted gaussian mixture models". *Digital Signal Processing*, 10, 2000.

[RR95] D. A. Reynolds and R. C. Rose. robust text-independent speaker identification using gaussian mixture speaker models. *IEEE Transactions on Speech and Audio Processing*, 3, 1995.

[Sht00] Y. J. Shtein. *"Digital Signal Processing A Computer Science Perspective"*. "John Wiley & Sons, Inc", 2000.

[SST+02] E. Schneidman, N. Slonim, N. Tishby, R. de Ruyter van Steveninck, and W. Bialek. "analyzing neural codes using the information bottleneck method". In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2002.

[ST00a] N. Slonim and N. Tishby. "agglomerative information bottleneck". In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2000.

[ST00b] N. Slonim and N. Tishby. "document clustering using word clusters via the information bottleneck method". In *Proceedings of SIGIR*, 2000.

[TPB99] N. Tishby, F. Pereira, and W. Bialek. "the information bottleneck method". In *Proceedings of the 37-th Annual Allerton Conference on Communication, Control and Computing*, 1999.

[WOVY94] P.C. Woodland, J.J. Odell, V. Valtchev, and S.J. Young. "large vocabulary continuous speech recognition using htk ". In *Proceedings of IEEE International Conference Acoustics, speech, signal processing (ICASSP)*, 1994.