TEXT-INDEPENDENT SPEAKER IDENTIFICATION USING SPARSELY EXCITED SPEECH SIGNALS AND COMPRESSED SENSING

by

Karamichali Eleni

A thesis submitted to the faculty of

University Of Crete

in partial fulfillment of the requirements for the degree of

Master of Science

Computer Science Department

University of Crete

October 2010

ABSTRACT

TEXT-INDEPENDENT SPEAKER IDENTIFICATION USING SPARSELY EXCITED SPEECH SIGNALS AND COMPRESSED SENSING

Karamichali Eleni

Computer Science Department

Master of Science

Compressed Sensing (CS) is an emerging theory that claims that the Nyquist sampling theorem yields for more samples than necessary. According to the Nyquist sampling theorem, the sampling rate of a signal must be at least equal to the double of its maximum frequency. On the contrary, CS seeks to represent a signal using a small number of linear, non-adaptive measurements which are far less than the signal's bandwidth. Thus, CS accomplishes both compression and sampling in one low-complexity step. The only requirement for CS to be efficient is that the signal is sparse in some basis, which means it has only a few non zero elements in some basis.

Compressed sensing has been used for full signal reconstruction, but in our case it was used for feature recovery in order to perform text-independent speaker identification. Speaker identification is the act of recognizing a speaker under the condition that he is a part of a database which has been modeled be-

forehand using features extracted from each speaker's training set. Specifically, we trained a Gaussian Mixture Model for each speaker in the database, using Line Spectral Frequencies. Text-independent speaker identification means that the testing speech signals were not included in the training phase.

We chose to use CS theory for speaker identification for two reasons. The first one is that CS theory requires just a few samples to reconstruct a signal and this is very useful in environments like sensor networks where there are limitations in the data traffic that can be sent between the sensor nodes. Thus, although traffic is limited, we are still able to avoid information loss. The second reason is that CS algorithms are robust to noise. These algorithms force the signals to be sparse in some basis which results in neglecting noisy samples that have low energy.

After experimenting with some CS algorithms for signal reconstruction, we decided to use Orthogonal Matching Pursuit for our research because of its low complexity and the lowest feature distortion after the reconstruction.

The results may not be as good as the ones using features extracted from the original speech signals, but they are quite good regarding the number of samples that were used, and are very promising for future investigation and research.

ABSTRACT

Αναγνώριση Ομιλητή Ανεξάρτητη από το Κείμενο με χρήση Αραιών Σημάτων και της θεωρίας Συμπιεστικής Δειγματοληψίας

Ελένη Καραμιχάλη
Τμήμα Επιστήμης Υπολογιστών
Πανεπιστήμιο Κρήτης

Η θεωρία της Συμπιεστικής Δειγματοληψίας (Compressed Sensing) είναι μία αναπτυσσόμενη θεωρία που υποστηρίζει ότι το θεώρημα δειγματοληψίας των Νψχυιστ απαιτεί μεγαλύτερο ρυθμό δειγματοληψίας από ότι είναι απαραίτητο. Σύμφωνα με το θεώρημα δειγματοληψίας των Νψχυιστ, ο ρυθμός δειγματοληψίας πρέπει να είναι τουλάχιστον διπλάσιος από τη μεγαλύτερη συχνότητα του σήματος, ώστε να μπορούμε να το ανακατασκευάσουμε τέλεια. Αντίθετα, η θεωρία της Συμπιεστικής Δειγματοληψίας προσπαθεί να βρει μία αναπαράσταση του σήματος που απαρτίζεται από ένα μικρό αριθμό γραμμικών μετρήσεων, πολύ μικρότερο από το εύρος συχνοτήτων του σήματος. Έτσι η θεωρία της Συμπιεστικής Δειγματοληψίας επιτυγχάνει να συνδυάσει σε ένα βήμα χαμηλής πολυπλοκότητας, αφενός συμπίεση και αφετέρου δειγματοληψία του σήματος. Η μόνη προϋπόθεση για τα παραπάνω είναι το σήμα να είναι αραιό σε κάποια βάση, το οποίο σημαίνει να έχει πολύ λίγα μη μηδενικά στοιχεία σε κάποια βάση.

Η θεωρία της Συμπιεστικής Δειγματοληψίας έχει ως τώρα χρησιμοποιηθεί σε

εφαρμογές όπου χρειάζεται πλήρης ανακατασκευή του σήματος, αλλά σε αυτή την εργασία χρησιμοποιήθηκε για την ανακατασκευή χαρακτηριστικών με σκοπό την αναγνώριση ομιλητή ανεξάρτητα από το κείμενο. Η αναγνώριση ομιλητή είναι η διαδικασία εύρεσης του ατόμου που μιλάει με την προϋπόθεση ότι ανήκει σε μία βάση που απαρτίζεται από ομιλητές, και έχει προηγηθεί ¨εκπαίδευση’ ενός συστήματος με χαρακτηριστικά που έχουν εξαχθεί από τα σήματα ομιλίας κάθε ομιλητή στη βάση. Συγκεκριμένα, δημιουργήθηκε μία συλλογή κβαντισμένων χαρακτηριστικών για κάθε ομιλητή που ανήκει στη βάση μας, χρησιμοποιώντας ως χαρακτηριστικά γραμμικές φασματικές συχνότητες. Ο όρος ανεξάρτητη αναγνώριση από το κείμενο αναφέρεται στο γεγονός ότι τα σήματα ομιλίας που χρησιμοποιήθηκαν για τη δοκιμή του συστήματός μας, δεν είχαν συμπεριληφθεί στη φάση της ¨εκπαίδευσησ’ του συστήματος.

Επιλέξαμε τη θεωρία της Συμπιεστικής Δειγματοληψίας για την αναγνώριση ομιλητή για δύο λόγους. Ο πρώτος είναι ότι η θεωρία της Συμπιεστικής Δειγματοληψίας απαιτεί πολύ λίγα δείγματα για να πετύχει πλήρη ανακατασκευή ενός σήματος, και αυτό είναι πολύ χρήσιμο σε περιβάλλοντα όπως δίκτυα αισθητήρων όπου η κίνηση δεδομένων είναι περιορισμένη. Με αυτόν τον τρόπο, ενώ μπορούμε να στείλουμε πολύ λίγα δεδομένα, δεν έχουμε απώλεια πληροφορίας. Ο δεύτερος λόγος είναι ότι οι αλγόριθμοι ανακατασκευής σήματος της Συμπιεστικής Δειγματοληψίας είναι ανεκτικοί στο θόρυβο. Αυτοί οι αλγόριθμοι εξαναγκάζουν τα σήματα να είναι αραιά σε κάποια βάση, με αποτέλεσμα να μην λαμβάνονται υπόψη τα δείγματα του θορύβου εξαιτίας της χαμηλής τους ενέργειας.

Μετά τη διεξαγωγή πειραμάτων με κάποιους αλγορίθμους ανακατασκευής σημάτων, αποφασίσαμε να χρησιμοποιήσουμε τον αλγόριθμο Ορτηογοναλ Ματςηινγ Πυρσυιτ για την έρευνά μας λόγω της χαμηλής του πολυπλοκότητας και της μικρότερης διαστρέβλωσης που προκαλούσε η ανακατασκευή στο σήμα.

Τα αποτελέσματα μπορεί να μην είναι όσο καλά είναι όταν τα χαρακτηριστικά που έχουν εξαχθεί από τα πρωτότυπα σήματα ομιλίας, αλλά είναι πολύ καλά δεδομένου ότι τα χαρακτηριστικά ανακατασκευάζονται από ένα μικρό ποσοστό των αρχικών δειγμάτων και είναι πολύ ενθαρρυντικά για περαιτέρω ενασχόληση και έρευνα.

ACKNOWLEDGMENTS

At first, I would like to thank my supervisor professor mr. Athanasios Mouchtaris for his continuous support and help through the M.Sc. program. Professor Mouchtaris was constantly available for guidance and support and he would always listen to every question, significant or not. He was the first to involve me with research issues and showed me how to approach a problem from different points of view and how to think in order to find a solution to everything. I am very grateful for everything he taught me. On the one hand, for the knowledge about audio signals and algorithms and on the other hand, for the values, the encouragement, the guidance and motivation, the trust and the support during the last two and a half years of collaboration.

I would also like to thank professor mr. Ioannis Stylianoy for his encouragement and guidance through my undergraduate studies. He was the first to teach me about digital signal processing and because of the way he looked at things and his different approach to every problem, he inspired me to choose Digital Signal Processing as my field of expertise.

Except from the professors, the years of studying would be unbearable without some people that were always on my side. I would like to thank Elias Tragos, Mariana Karmazi, Maria Astrinaki, Elias Apostolopoulos, Panagiotis Gerovasilis, Frixos Terzakis, Maria Papageorgiou, Katia Lampropoulou, Sophie Karagiorgou, Georgina Tryfou, Thanasis Krontiris, Vicky Katirtzidaki, Kostas Tsitiridis, Savvas Kosmidis, and Despina Pavlidi, who were always

there for me to talk or discuss everything that troubled me, to support me and help me, to listen. I am grateful for their unconditional love although we all had our difficulties, and I would like to thank them for making me a better person and for everything I have learned from each one of them. I would also like to thank Christos Tzagkarakis and Anthony Griffin for helping me and giving me advice for code, algorithms and issues on my work.

Last but not least, I would like to thank the two people, that without their help nothing of these would be happening. Of course, I refer to my parents Elias and Theodosia Karamichali who supported me all these years without any complaints, who always believed in me even in times that I did not. I would like to thank them for giving birth to me in the first place, for their unlimited love, but also for my education, for their unconditional support to chase my dreams and interests, and the long conversations we have, always accompanied with a cup of coffee or alcohol and always with a cigarette. They gave me everything I needed in order to become who I am today, and would not change anything not even their different point of view in some matters. At last, I would like to thank my brother George Karamichalis for his support and the knowledge he gave on every aspect of life. Even if he did not know the answer to my questions, he was always there to help me overcome my problems. He also made me a better computer engineer because he would always find a problem in his computer that would need at least two days of work for me to solve.

Contents

List of Figures

Chapter 1

Introduction

According to the Shannon/Nyquist sampling theorem, if we want to store a signal without loosing any information, we have to sample it at least two times faster that the signal's maximum frequency, the so-called signal bandwidth. In fact, this principle underlies nearly all signal acquisition protocols used in consumer audio and visual appliances, radio receivers, and so on. In such applications, the Nyquist rate is so high resulting in too many samples, making compression a necessity prior to storage or transmission. In other applications, including imaging systems (medical scanners and radars) and high-speed analog-to-digital converters, increasing the sampling rate is very expensive. Also in gene expression studies one would like to infer the gene expression level of thousands of genes from a low number of observations. This work's focus is a recently developed theory called Compressed Sensing Theory. CS theory asserts that one can recover certain signals and images from far fewer samples or measurements than the Shannon/Nyquist sampling theorem requires. To make this possible, CS relies on two principles: sparsity, which pertains to the signals of interest, and incoherence, which pertains to the sensing modality. [2,3]

- Sparsity expresses the idea that the "information rate" of a continuous time signal may be much smaller than suggested by its bandwidth, or that a discrete-time signal depends on a number of degrees of freedom which is comparably much smaller than its (finite) length. More precisely, CS exploits the fact that many natural signals are sparse or compressible in the sense that they have concise representations when expressed in the proper basis Ψ.

- Incoherence extends the duality between time and frequency and expresses the idea that objects having a sparse representation in Ψ must be spread out in the domain in which they are acquired, just as a Dirac or a spike in the time domain is spread out in the frequency domain. Put differently, incoherence says that unlike the signal of interest, the sampling/sensing waveforms have an extremely dense representation in Ψ.

Compressed sensing has been used for full signal reconstruction. However, in this work we used CS theory for feature recovery in order to perform text-independent speaker identification. Speaker identification is the task of resolving who is talking, using features extracted from his or her voice. Moreover, text-independent speaker identification is the task of finding one's identity regardless the content of what was said. This task is accomplished by matching the features extracted from the unknown speaker's voice to a trained system that has been acquired from a database of speakers. In fact, it is a classification problem among a known existing database of speakers.

There are two reasons to use Compressed Sensing in speaker identification. On the one hand, CS theory uses only a portion of the samples that Nyquist's sampling theorem requires to reconstruct a signal. Thus, in environments where we are not able to send a large amount of data, we are able to avoid information loss regardless the limited packets we send. On the other hand, signal reconstruction is less affected by noise. When forcing the signal to be sparse in some basis, the part of the signal that is going to be neglected will be the one with the smallest energy, thus the noisy one. This is similar to signal de-noising by low-rank modeling. In the latter case, the amount of non-zero elements of the signal plays a key role because compressed sensing theory assumes that the signal is initially sparse in some basis. Thus, it is interesting to investigate whether speaker identification has better results when Compressed Sensing is used or when we use the samples of the original signal.

A key question is whether a speech signal can be considered to be sparse in some sense. For audio signals, it was recently showed that their sinusoidally modeled component can be considered to be sparse, and compressed sensing theory was applied to low-bitrate audio coding [4]. For speech signals, compressed sensing was recently applied to a sparse representation using the source/filter model in [5] for speech coding, and encouraging preliminary results were obtained. In this work, we extend the work of [5] by applying the proposed methodology to the problem of text-independent speaker identification. In that work, it was found that applying compressed sensing theory to speech signals modeled using the source/filter model, and assuming a sparse excitation, resulted in accurate estimation of the filter part (spectral envelope) of the speech signal.

This work is organized as follows: In the second chapter we will refer to the basics of the Compressed Sensing theory. In the third chapter, a baseline algorithm for speaker identification is described analyzing also the feature extraction process. In the fourth chapter, the algorithm that we used for the speaker identification process using Compressed Sensing Theory is described. In the fifth chapter, we discuss the problem that was spotted and the solution we propose, describing our work. In the sixth chapter we will present the results of our research. And at the last chapter, a small discussion takes place where we refer to our results and how it is possible to further investigate the subject worked on.

Chapter 2

Compressed Sensing

In this chapter we discuss the basics of Compressed Sensing, the general idea behind the theory and what lead to using CS theory.

2.1 The basics of Compressed Sensing

Compressed sensing seeks to represent a signal using a small number of linear, non-adaptive measurements. Usually the number of measurements is much lower than the number of samples needed if the signal is sampled at the Nyquist rate. Thus, compressed sensing combines compression and sampling of a signal into one low-complexity step. An important restriction is that compressed sensing requires that the signal is sparse in some basis, in the sense that it is a linear combination of a small number of basis functions-in order to correctly reconstruct the original signal. In this paragraph we will describe how all these are possible.

Let's consider a real valued signal x, with finite length and one dimension. x is a $N \times 1$ vector, with $x \in \Re$ and discrete time with elements $x[n]$ with $n = 1, 2, \text{ﬄ}, N$. This signal can be represented in terms of a $N \times N$ orthonormal basis Ψ, where every

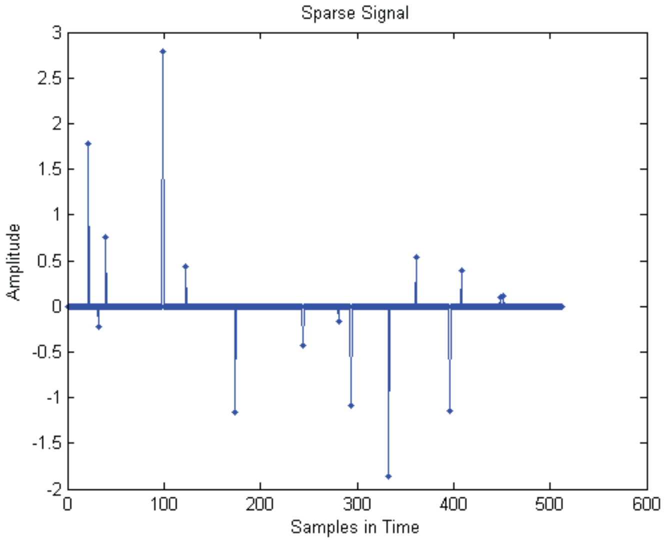

Figure 2.1 Example of a sparse signal

column is $\{\psi_i\}_{i=1}^{N}$. The signal x can now be expressed by the basis as:

$$\sum_{i=1}^{N} s_i\psi_i \quad or \quad x = \psi s \tag{2.1}$$

where s is the $N \times 1$ vector of weighting coefficients $s_i = \langle x, \psi_i \rangle = \psi_i^T x$. Actually, x and s represent the same signal, the first in the time domain, and the latter in the Ψ domain.

The signal x is K-sparse if it consists of only K non-zero elements, and $N - K$ zero elements. In a different point of view, a signal is K-sparse signal when it can be written as a linear combination of only K basis functions. Obviously, we are interested in the cases where K is much smaller than N (the length of the signal)(Figure 2.1).

The signal x is called compressible when it has only a few large values and all the other are very close to zero. In that case we can discard the small values because they are insignificant, and the signal is supposed to be sparse (Figure 2.2).

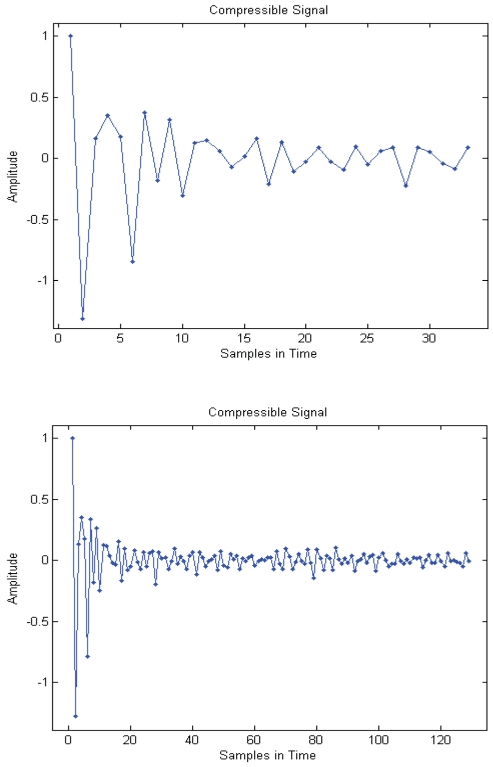

Figure 2.2 Examples of compressible signals

2.1.1 Transform Coding and its inefficiencies

Transform coding is based on the compressible signals, specifically on their property to transform in K-sparse signals. The process of transform coding consists of the following steps:

- The full signal must be acquired
- The transform coefficients $s = \Psi^T x$ must be computed
- The K largest values of s are located, and all the other that are near zero are discarded
- The K largest values and their locations are encoded.

This whole process though is very inefficient. First of all, one has to acquire all N samples of the original signal x, and then compute the measurements s. This means, that one would need a lot of memory space to save the signal no matter how large it is, and moreover CPU time to compute N coefficients, although only K will be needed and stored. Last but not least, the locations of the non zero elements must be encoded too, resulting in storing $2K$ elements instead of K.

This is why a need emerged for a new theory to surpass these problems. Compressed Sensing surpasses the above problems by acquiring from the start only the K non zero elements of the signal. It acquires $M < N$ inner products between x and a $M \times N$ matrix Φ consisting of N columns $\{\phi_j\}_{j=1}^{M}$. Let's consider the inner product, $M \times 1$ vector $y_j = \langle x, \phi_j \rangle$. Then using 6.1 we can rewrite ψ as:

$$y = \Phi x = \Phi \Psi s = \Theta s \tag{2.2}$$

where $\Theta = \Phi\Psi$ is a $M \times N$ matrix. The measurement process is not adaptive. This means that Φ is fixed and does not depend on the signal x.

The challenge though is twofold. In one hand, we should find a stable measurement matrix Φ such that the measurement process does not damage the important information of the signal, and on the other hand, we should find a reconstruction algorithm that recovers x from only M samples that were kept, and moreover recovers the right positions of the non-zero samples.

2.1.2 Designing matrix Φ

We have to reconstruct a $N \times 1$ vector signal, only by M measurements. However, $M < N$ so the problem is ill- conditioned. If our signal x is K-sparse and the positions of the non-zero elements are known, since $M \geq K$, the problem can be solved. For this problem to be well-conditioned, there is a necessary and sufficient condition. For any vector v sharing the same K nonzero entries as s and for some $\epsilon > 0$

$$1 - \epsilon \leq \frac{\|\Theta v\|_2}{\|v\|_2} \leq 1 + \epsilon \tag{2.3}$$

This inequality means that matrix Θ must preserve the length of the K-sparse vectors. However, we said earlier that this is in effect if the positions of the non-zero elements are known, which are not. A sufficient condition for a stable solution for both K-sparse and compressible signals is that Θ satisfies equation (2.3) for an arbitrary $3K$-Sparse vector v. This condition is referred to as restricted isometry property (RIP). A related condition referred to as incoherence, requires that the rows $\{\phi_j\}$ of Φ cannot sparsely represent the columns of Ψ and vise versa.

Direct construction of a matrix Φ such that $\Theta = \Phi\Psi$ has the RIP requires verifying equation (2.3) for all the possible combinations of K non-zero elements in the vector v of length N. However, both the RIP and incoherence can be achieved with high probability simply by selecting Φ as a random matrix.

If matrix Φ consists of independent and identically distributed random variables from a Gaussian probability density function, with zero mean and variance $1/N$, then because $y = \Phi x$, y will consist of M different weighted linear combinations of the elements of x. The Gaussian measurement matrix Φ has interesting properties:

- The matrix Φ is incoherent with a basis equal to an identity matrix and it can be shown that $\Theta = \Phi\Psi = \Phi$ has the RIP with high probability if $M \geq cK \log(N/K) \ll N$, with c a small constant. Therefore, K-sparse or compressible signals of length N can be recovered from only M random Gaussian measurements that obey the above inequality.

- No matter what the basis Ψ will be, matrix Θ will be i.i.d. Gaussian and thus have the RIP with high probability.

As long as we have chosen our measurement matrix, we have to chose a reconstruction algorithm. It has to recover all N samples of our original signal x or the sparse vector s, from only M measurements in the vector y, our measurement matrix Φ, and the basis Ψ.

We have stated that $y = \Theta s$. But for $M < N$ there are too many s' that satisfy that equation. This is because if $\Theta s = y$ then $\Theta(s + r) = y$ for any vector r in the null space $N(\Theta)$ of Θ. Therefore, the signal reconstruction algorithm tries to find the signal's sparse coefficient vector in the $(N - M)$-dimensional translated null space $H = N(\Theta) + s$.

There are three kinds of basic reconstruction algorithms. First of all, let's define the l_p norm of a vector s as $(\|s\|_p)^p = \sum_{i=1}^{N} |s_i|^p$.

- The most classical approach to inverse problems of this type is to to find the vector in the translated null space with the smallest l_2 norm energy by solving

$$\widehat{s} = \arg\min \|s'\|_2 \quad such \quad that \quad \Theta s' = y \tag{2.4}$$

  This optimization has the convenient closed-form solution $\widehat{s} = \Theta^T(\Theta\Theta^T)^{-1}y$. Unfortunately, l_2 minimization will almost never find a K-sparse solution, returning instead a non sparse $\widehat{s}$ with many non-zero elements.

- Since the l_2 norm measures signal energy and not signal sparsity, consider the l_0 norm that counts the number of non-zero entries in s. (Hence a K-sparse vector has l_0 norm equal to K.) The modified optimization

$$\widehat{s} = \arg\min \|s'\|_0 \quad such \quad that \quad \Theta s' = y \tag{2.5}$$

  can recover a can recover a K-sparse signal exactly with high probability using only $M = K + 1$ i.i.d. Gaussian measurements (2.3). Unfortunately solving

(2.5) is both numerically unstable and NP-complete, requiring an exhaustive enumeration of all $\binom{N}{K}$ possible locations of the nonzero entries in s.

- Surprisingly, optimization based on the l_1 norm, which is basically the sum of the elements of the signal,

$$\widehat{s} = \arg\min \|s'\|_1 \quad such \quad that \quad \Theta s' = y \tag{2.6}$$

can exactly recover K-sparse signals and closely approximate compressible signals with high probability using only $M \geq cK \log(N\ K)$ i.i.d. Gaussian measurements (2.4). This is a convex optimization problem that conveniently reduces to a linear program known as Basis Pursuit, whose computational complexity is about $O(N^3)$. Equation 6.2 can be easily reformulated as:

$$\hat{s} = \arg\overset{\prime}{\underset{s}{\min}} \|y - \Theta s'\|_2 \quad such \quad that \quad \|s\|_0 = K \tag{2.7}$$

where the l_0 norm just counts the nonzero elements.

2.2 Sparsely Excited Signals

There are two parametric ways to represent speech or audio signals in the time domain. The one is by linear system models which can represent only speech signals, and the other one is sinusoidal modeling for representing both speech ad music. It is widely known that the most important features of a sound are their spectral ones and their harmonicity due to periodic excitation. That is why we represent signals by either linear prediction coefficients or their cepstrum analysis, in order to separate the periodic information from the spectral features.

Nevertheless, we can represent signals in other ways too, by projecting them onto bases that would make the representation sparse. For example, we can represent

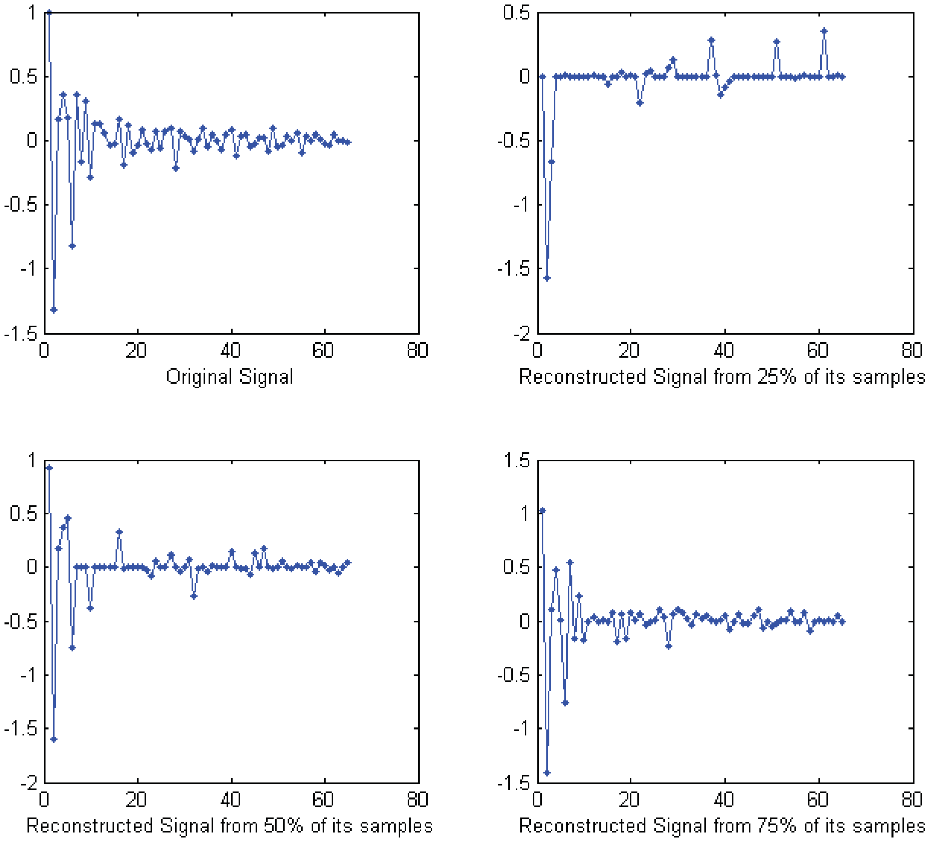

Figure 2.3 Signal Reconstruction using Matching Pursuit

the original signal as a linear combination of Discrete Cosine Transform (DCT) or Discrete Fourier Transform (DFT) coefficients. In the former case the measurement matrix would be a real valued transform matrix, but in the latter one it would be complex. However, the sparsity of the outcome will be unknown.

There is another linear sparse representation in the time domain, which is more suitable than the previously stated. In speech coding, the transform domain where the representation is required to be sparse is the prediction residual. [6, 7] In the

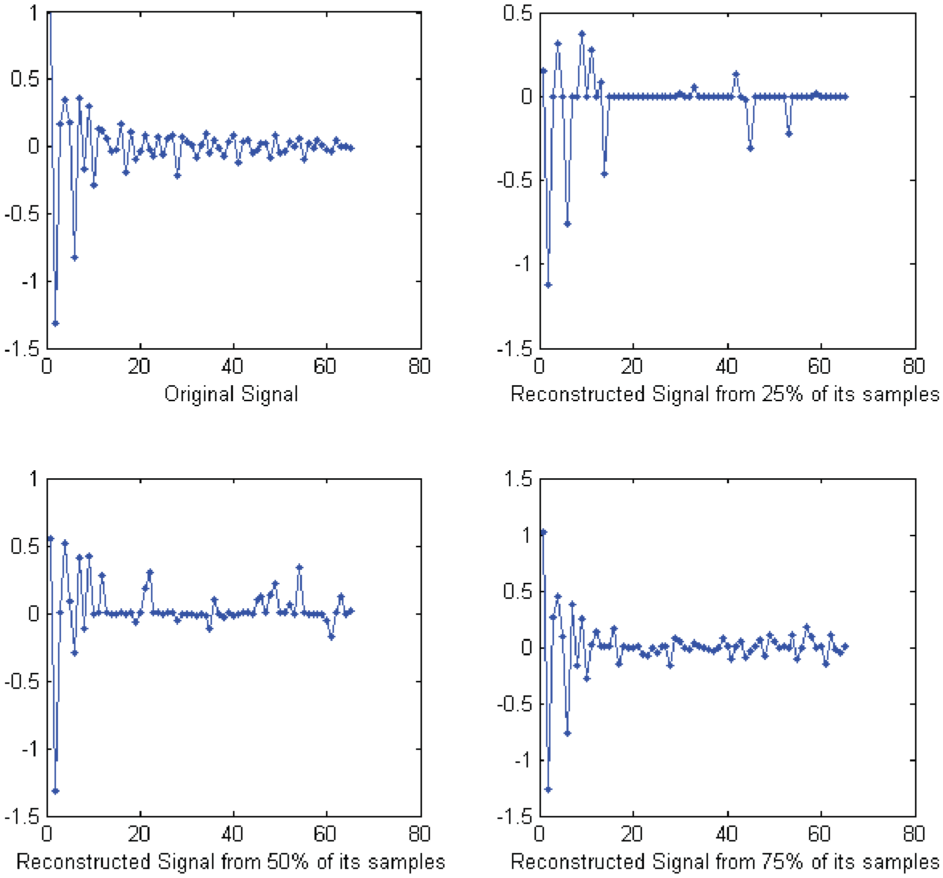

Figure 2.4 Signal Reconstruction using Basis Pursuit

simple case, a sparse linear predictor a of order P derives from

$$\hat{a} = \arg \min_{a \in \Re^P} \|x - Xa\|_1 \tag{2.8}$$

where

$$x = \begin{bmatrix} x(N_1) \\ \vdots \\ x(N_2) \end{bmatrix}, \quad X = \begin{bmatrix} x(N_1-1) & \cdots & x(N_1-P) \\ \vdots & & \vdots \\ x(N_2-1) & \cdots & x(N_2-P) \end{bmatrix}$$

and $\|\cdot\|_1$ is the $l1$ norm. The points N_1 and N_2 can be chosen with various ways. The most appropriate one is $N_1 = 1 \quad and \quad N_2 = N + P$.

The residual excitation component expressed in a $N \times 1$ vector can then be expressed as

$$r = Ax$$

and the signal can be reconstructed by

$$x = A^{-1}r = h \cdot r \tag{2.9}$$

where h is the signal domain impulse response of the smooth spectral envelope expressed in a $N \times N$ impulse response matrix.

$$h = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ a(1) & 1 & 0 & \cdots & 0 \\ \cdots & a(1) & 1 & \cdots & 0 \\ a(P) & a(P-1) & \cdots & 1 & 0 \\ 0 & a(P) & a(P-1) & \cdots & 1 \end{bmatrix}^{-1}$$

The matrix h would be $N \times K$ Toeplitz lower triangular for linear convolution and $N \times N$ circulant Toeplitz for circular convolution. Since h is signal dependent, in [5] it is proposed to use a codebook of size L of such matrices, produced by the training data. Then the reconstruction problem can be formulated similar to the basic formula of the orthogonal matching pursuit algorithm (which will be described in Chapter 4)

$$[\hat{r}, \hat{h}_l] = \arg\min_{h_l, r} \|y - \Phi \cdot h_l \cdot r\|_2, \quad \textit{such} \quad \textit{that} \quad \|r\|_0 = K, \tag{2.10}$$

$$\textit{and} \quad \hat{x} = \hat{h}_l \cdot \hat{r}$$

where K is the level of sparsity of the signal and $l = 1, 2, ..., L$. This representation is more suitable because, when the signal is represented in the time domain, it is processed in small time-windows because of the constant changes in the sound features. During these windows, the number of periods in the time domain are much fewer than the number of harmonics in the spectrum resulting in a sparser x.

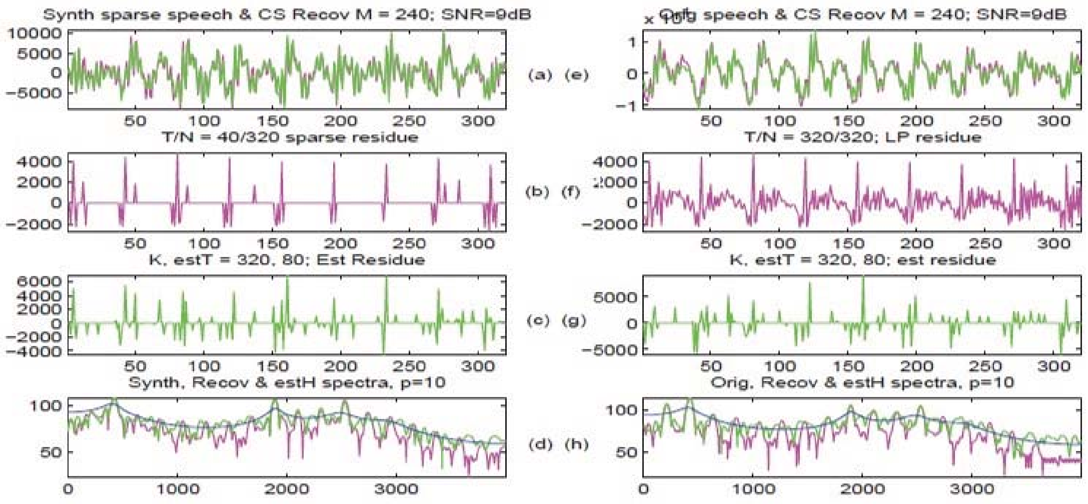

Figure 2.5 Example of CS recovery of speech: (a,e) 40ms speech frame, (b,f) residue signal (c,g) estimated residue (d,h) spectra of estimated LP, speech signal and recovered signal. Left column: signal with exact sparsity. Right column: original speech with approximate sparsity. [5]

Chapter 3

Background Information on Speaker Identification

In this chapter we will present a baseline speaker identification process, used to compare our results. We describe the data modeling, the way we extract the features chosen to represent our data and finally the computation of the identification certainty probabilities.

3.1 Text Independent Speaker Identification

There are two different tasks to which speaker recognition refers to, depending on the application. The one is speaker verification, which seeks the validity of the speaker's claim about his identity. The second one is the speaker identification, which tries to match the voice sample with one of the speakers in a given database. Furthermore, in either task, the voice sample can be constrained, for example a specific sequence of words, or unconstrained. The first case is called text-dependent recognition, and the second is called text-independent where the recognition depends only on the features of the speech signal and not on its content. [8]

Several speaker dependent spectral shapes tend to be represented by Gaussian densities. Moreover, Gaussian mixtures can model arbitrary densities. For these reasons, Gaussian Mixture Models (GMM) were used to represent the speaker database.

A Gaussian Mixture Model is a weighted sum of M component densities and given by the equation

$$p(\vec{x}|\Lambda) = \sum_{i=1}^{M} p_i b_i(\vec{x}) \tag{3.1}$$

where $\vec{x}$ is a D-dimensional vector, $b_i, i = 1, ..., M$, are the component densities and $p_i, i = 1, ..., M$, are the mixture weights. Each component density is a D-variate Gaussian function of the form

$$b_i(\vec{x}) = \frac{1}{(2\pi)^{\frac{D}{2}} |\Sigma_i|^{\frac{1}{2}}} \exp -\frac{1}{2}(\vec{x} - \vec{\mu}_i)' \Sigma_i^{-1} (\vec{x} - \vec{\mu}_i) \tag{3.2}$$

with mean vector $\vec{\mu}_i$ and covariance matrix Σ_i. The mixture weights satisfy the constraint: $\sum_{i=1}^{M} p_i = 1$. The complete Gaussian mixture density is parameterized by the mean vectors, covariance matrices and mixture weights from all component densities. These parameters are collectively represented by

$$\lambda = \{p_i, \vec{\mu}_i, \Sigma_i\} i = 1, ..., M \tag{3.3}$$

For speaker identification, each speaker is represented by a different GMM. The covariance matrix of every GMM can have several forms, but in this work we use diagonal ones.

3.2 Linear Prediction

The variables that represent each speaker cannot be the speech signal itself, but we have to extract some features by the signal that reflect the identity of the speaker.

The features selected are called Linear Prediction Coding coefficients. The LPC coefficients represent the spectral envelope of the signal and are widely used in audio signal processing. The basic idea is to formulate a linear system based on the input and previous outputs. Let's say that $x(n)$ is the input of the system and $y(n)$ its output (with $y(n-1), y(n-2), ...$ the previous outputs). The linear system will be:

$$\hat{y}(n) = \sum_{k=1}^{p} \alpha(k)y(n-k) + \sum_{k=0}^{N} b(k)x(n-k) \tag{3.4}$$

The $\hat{y}(n)$ denotes an estimation of the exit $y(n)$. The problem is to determine the $\alpha(k)$ and $b(k)$ constants such that the estimation of the future output is as accurate as possible. If the system is modeled as an all-pole one, then the prediction will be perfect if we know the input and the previous outputs. In practice, the prediction can never be perfect because the systems are not linear nor all-pole and there is always some noise. Moreover, the input $x(n)$ is unknown. Nevertheless, when we model the vocal tract with an all-pole model, the results are very good.

So if we form equation 3.4 as an all-pole system we will have:

$$\hat{y}(n) = -\sum_{k=1}^{p} \alpha(k)y(n-k). \tag{3.5}$$

Since $b(k) = 0$, we have to compute $\alpha(k)$ such that the prediction is as close as possible to the original output of the system. There are two ways to compute $\alpha(k)$, the autocorrelation method and the Levinson - Durbin Recursion.

Autocorrelation Method

In the autocorrelation method, the parameters $\alpha(1), ..., \alpha(p)$ are chosen in a way that

$$\sum_{n}(\hat{y}(n) - y(n))^2$$

is minimized. In the following the output $y(n)$ will be denoted as $s(n)$.

So we have a speech signal $s(n)$ with a finite number of nonzero elements. With the given prediction coefficients $\alpha(1), \alpha(2), ..., \alpha(p)$, the energy prediction error can be written as

$$\begin{aligned} E_p &= \sum_{n=-\infty}^{\infty} e(n)^2 \\ &= \sum_{n=-\infty}^{\infty} [s(n) - \hat{s}(n)]^2 \\ &= \sum_{n=-\infty}^{\infty} [s(n) - \sum_{k=1}^{p} -\alpha(k)s(n-k)]^2, \end{aligned}$$

where p is the length of the prediction filter and $\hat{s}(n)$ is the prediction of the $s(n)$.By having convention that $\alpha(0) = 1$, the energy E_p of the prediction error can be written as

$$E_p = \sum_{n=-\infty}^{\infty} [\sum_{k=0}^{p} \alpha(k)s(n-k)]^2.$$

Now we have to minimize E_p in terms of $\alpha(1), ..., \alpha(p)$. A necessary condition for optimality of the choice of $\alpha(i)$ is that the partial derivative of E_p with respect to variable $\alpha(i)$ equals zero. Notice that E_p depends on the variables $\alpha(1), \alpha(2), ..., \alpha(p)$ so it could be written as $E_p(\alpha(1), \alpha(2), ..., \alpha(p))$ but we omit this to keep the notation short. The partial derivative with respect to $\alpha(i), i = 1, 2, ..., p$ is:

$$\begin{aligned} \frac{\partial E_p}{\partial \alpha(i)} &= \frac{\partial \sum_n [\sum_{k=0}^{p} \alpha(k)s(n-k)]^2}{\partial \alpha(i)} \\ &= \sum_n 2[\sum_{k=0}^{p} \alpha(k)s(n-k)] \frac{\partial \sum_{k=0}^{p} \alpha(k)s(n-k)}{\partial \alpha(i)} \\ &= \sum_n 2[\sum_{k=0}^{p} \alpha(k)s(n-k)]s(n-i), \end{aligned}$$

where the differentiation rule $f(g(x))' = f'(g(x))g'(x)$ has been utilized. By regrouping this we get

$$\begin{aligned} &\sum_{n=-\infty}^{\infty} 2[\sum_{k=0}^{p} \alpha(k)s(n-k)]s(n-i) \\ &= 2\sum_{k=0}^{p} \alpha(k) \sum_{n=-\infty}^{\infty} s(n-k)s(n-i) \\ &= 2\sum_{k=0}^{p} \alpha(k)r(k,i) \end{aligned}$$

where

$$r(k,i) = \sum_{n=-\infty}^{\infty} s(n-k)s(n-i)$$

is in fact the autocorrelation of the signal $s(n)$ with delay $k-i$ which is

$$\sum_{n=-\infty}^{\infty} s(n)s(n-(k-i))$$

because,

$$\begin{aligned} r(k,i) &= \sum_{n=-\infty}^{\infty} s(n-k)s(n-i) \\ &= \sum_{n=-\infty}^{\infty} s((n+i)-k)s((n+i)-i) \\ &= \sum_{n=-\infty}^{\infty} s(n)s(n-(k-i)). \end{aligned}$$

Moreover the term $r(k,i)$ depends only on value $k-i$ so it can be denoted by one variable autocorrelation function

$$r(k-i) = r(k,i).$$

By setting the derivatives to zero, we obtain:

$$\begin{cases} 2\sum_{k=0}^{p} \alpha(k)r(k-1) = 0 \\ 2\sum_{k=0}^{p} \alpha(k)r(k-2) = 0 \\ \vdots \\ 2\sum_{k=0}^{p} \alpha(k)r(k-p) = 0 \end{cases},$$

which can also be written in the following form, with $\alpha(0) = 1$ and $r(k) = r(-k)$

$$\begin{cases} \sum_{k=1}^{p} \alpha(k)r(k-1) = -r(1) \\ \sum_{k=1}^{p} \alpha(k)r(k-2) = -r(2) \\ \vdots \\ \sum_{k=1}^{p} \alpha(k)r(k-p) = -r(p) \end{cases},$$

which in turn can be reformulated with matrices as:

$$\begin{bmatrix} r(0) & r(1) & r(2) & \cdots & r(p-1) \\ r(1) & r(0) & r(1) & \cdots & r(p-2) \\ r(2) & r(1) & r(0) & \cdots & r(p-3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r(p-1) & r(p-2) & r(p-3) & \cdots & r(0) \end{bmatrix} \begin{bmatrix} \alpha(1) \\ \alpha(2) \\ \alpha(3) \\ \cdots \\ \alpha(p) \end{bmatrix} = - \begin{bmatrix} r(1) \\ r(2) \\ r(3) \\ \cdots \\ r(p) \end{bmatrix}$$

The coefficient matrix is symmetric and Toeplitz due to $r(k) = r(-k)$ and $r(k, i) = r(k - i, i)$, which is crucial when deriving a fast computational method to find the coefficients $\alpha(1), \alpha(2), ..., \alpha(p)$.

At this point we have derived the equations (so called normal equations) for the prediction coefficients $\alpha(1), \alpha(2), ..., \alpha(p)$ based on the minimization of the prediction error. Now the coefficients could be solved by inverting the autocorrelation matrix, but this is computationally rather demanding.

Levinson - Durbin Recursive algorithm

In the Levinson - Durbin Recursive algorithm the basic idea is to solve the matrix equation

$$Rx = y$$

in steps, that is, by increasing the length of the vector x and by calculating a new solution based on the previous solution. The optimal coefficients satisfy

$$\sum_{i=0}^{p} \alpha(i) r(i) = E,$$

where E is the sum of squares of prediction error. By using this, the group of equations boils down to

$$\begin{bmatrix} r(0) & r(1) & r(2) & \cdots & r(p) \\ r(1) & r(0) & r(1) & \cdots & r(p-1) \\ r(2) & r(1) & r(0) & \cdots & r(p-2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r(p) & r(p-1) & r(p-2) & \cdots & r(0) \end{bmatrix} \begin{bmatrix} 1 \\ \alpha(1) \\ \alpha(2) \\ \cdots \\ \alpha(p) \end{bmatrix} = \begin{bmatrix} E \\ 0 \\ 0 \\ \cdots \\ 0 \end{bmatrix}$$

The matrix on the left is still symmetric and Toeplitz. Assume that we have already solved the equation when $p = 2$. Now, let us see how it helps us to solve

$\alpha_3(1), \alpha_3(2), \alpha_3(3)$ when $p = 3$, where the subscript refers to the degree of the equation. So this is what we have already solved:

$$\begin{bmatrix} r(0) & r(1) & r(2) \\ r(1) & r(0) & r(1) \\ r(2) & r(1) & r(0) \end{bmatrix} \begin{bmatrix} 1 \\ \alpha_2(1) \\ \alpha_2(2) \end{bmatrix} = \begin{bmatrix} E_2 \\ 0 \\ 0 \end{bmatrix}$$

.

The structure of **R** yields

$$\begin{bmatrix} r(0) & r(1) & r(2) \\ r(1) & r(0) & r(1) \\ r(2) & r(1) & r(0) \end{bmatrix} \begin{bmatrix} \alpha_2(2) \\ \alpha_2(1) \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ E_2 \end{bmatrix},$$

thus: symmetric Toeplitz matrices have the property that when the coefficient vector and the result vector are twisted upside down, the equation is still satisfied.

$$\begin{bmatrix} r(0) & r(1) & r(2) & r(3) \\ r(1) & r(0) & r(1) & r(2) \\ r(2) & r(1) & r(0) & r(1) \\ r(3) & r(2) & r(1) & r(0) \end{bmatrix} \left\{ \begin{bmatrix} 1 \\ \alpha_2(1) \\ \alpha_2(2) \\ 0 \end{bmatrix} + k_3 \begin{bmatrix} 0 \\ \alpha_2(2) \\ \alpha_2(1) \\ 1 \end{bmatrix} \right\}$$

$$= \begin{bmatrix} E_2 \\ 0 \\ 0 \\ q \end{bmatrix} + k_3 \begin{bmatrix} q \\ 0 \\ 0 \\ E_2 \end{bmatrix},$$

where $q = \sum_{i=0}^{2} \alpha_2(i) r(3-i)$.

For this to be a solution, we only require that all the elements, except the first one, in the vector on the right side are equal to zero. It will be so, if

$$q + k_3 E_2 = 0,$$

in other words

$$k_3 = -\frac{1}{E_2}\sum_{i=0}^{2}\alpha_2(i)r(3-i).$$

We notice that

$$\begin{aligned} E_3 &= E_2 + k_3 q \\ &= E_2 + k_3(-k_3 E_2) \\ &= E_2(1-k_3^2). \end{aligned}$$

Thus, we found that by trying a vector that is a sum of the lower degree solution and its twisted version multiplied by a constant, we get a solution to the problem of the higher degree. Same deduction works in general when increasing the size from $n-1$ to n. Thus, the results are

$$k_n = -\frac{1}{E_{n-1}}\sum_{i=0}^{n-1}\alpha_{n-1}(i)r(n-i),$$
$$E_n = E_{n-1}(1-k_n^2)$$

and

$$\alpha_n(i) = \alpha_{n-1}(i) + k_n\alpha_{n-1}(n-i).$$

Because $E_n \geq 0$ (E_n is the prediction error for the nth degree filter), it follows

$$|k_n| \leq 1.$$

The values k_n are called reflection coefficients. Levinson-Durbin recursion will be started with condition

$$r(0) = E_0,$$

which may be thought to be the error of the 0th degree predictor (no prediction at all).

There exist also other methods and variations to solve the coefficients but Levinson-Durbin recursion is the most commonly used one. Besides, calculating the coefficients in this way guarantees that the absolute values of the reflection coefficients are always ≤ 1, yielding a stable filter.

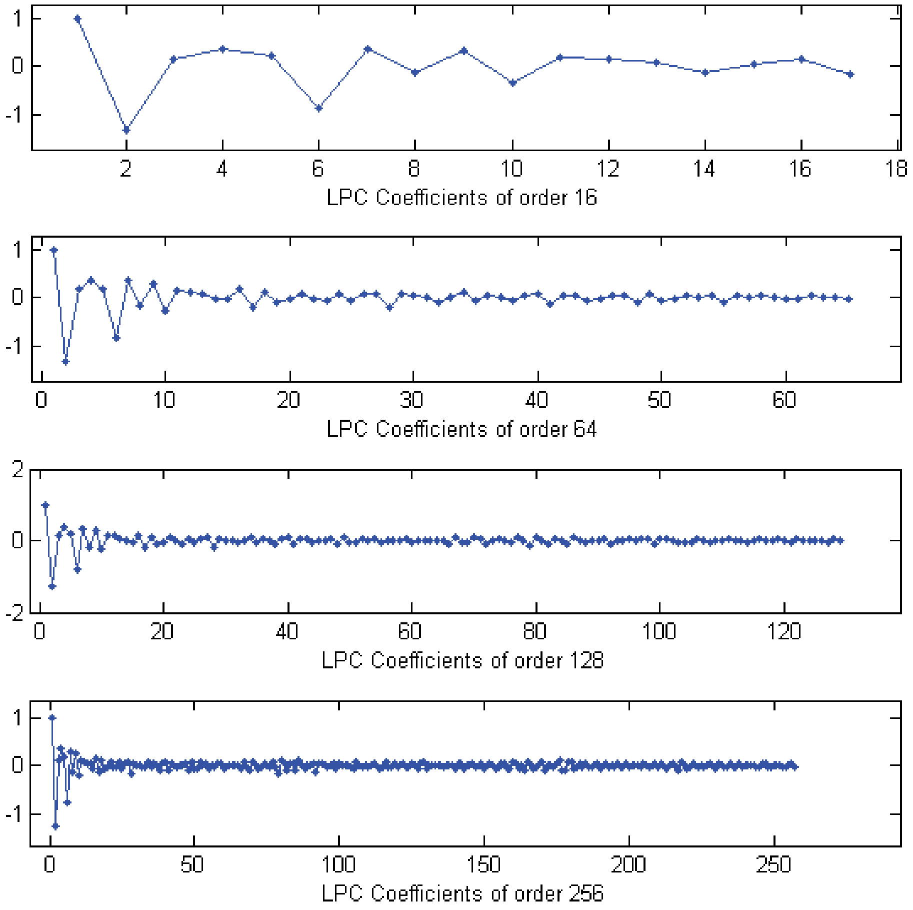

Figure 3.1 Examples of LPC coefficients of different orders

3.3 Line Spectral Frequencies vs LPC

Line Spectral Frequencies are widely used in speech coding, synthesis and recognition. They are alternatives to represent the all-pole spectrum of speech. They are a useful representation because the LPC coefficients are not a homogenous set and they do not quantize too well resulting on large spectral distortion. They also interpolate better than the LPC's because we cannot compute LPC's at two distinct times and expect to accurately predict the in between values. The zeros of the LPC polynomial are a better choice, since they all have the same physical interpretation. However, finding these zeros numerically entails a complex two dimensional search, while the the corresponding LSF zeros can be found by simple one-dimensional search techniques.

Let the m-th order inverse filter $A_m(z)$,

$$A_m(z) = 1 + \alpha_1 z^{-1} + ... + \alpha_m z^{-m}, \tag{3.6}$$

be obtained by the LP analysis of speech. The LSF polynomials of order $m + 1, \quad P_{m+1}(z) \quad and \quad Q_{m+1}(z)$,can be constructed by setting the $(m + 1)$-st reflection coefficient to 1 or -1. In other words, the polynomials $P_{m+1}(z)$ and $Q_{m+1}(z)$, are defined as

$$P_{m+1}(z) = A_m(z) + z^{-(m+1)} A_m(z^{-1}) \tag{3.7}$$

and

$$Q_{m+1}(z) = A_m(z) - z^{-(m+1)} A_m(z^{-1}) \tag{3.8}$$

The zeros of $P_{m+1}(z)$ and $Q_{m+1}(z)$ are called Line Spectral Frequencies and they uniquely characterize the LPC invert filter $A_m(z)$. $P_{m+1}(z)$ and $Q_{m+1}(z)$ are symmetric and anti-symmetric, respectively. They have the following properties:

- all of the zeros of the LSF polynomials are on the unit circle,

- the zeros of the symmetric and anti-symmetric LSF polynomials are interlaced,

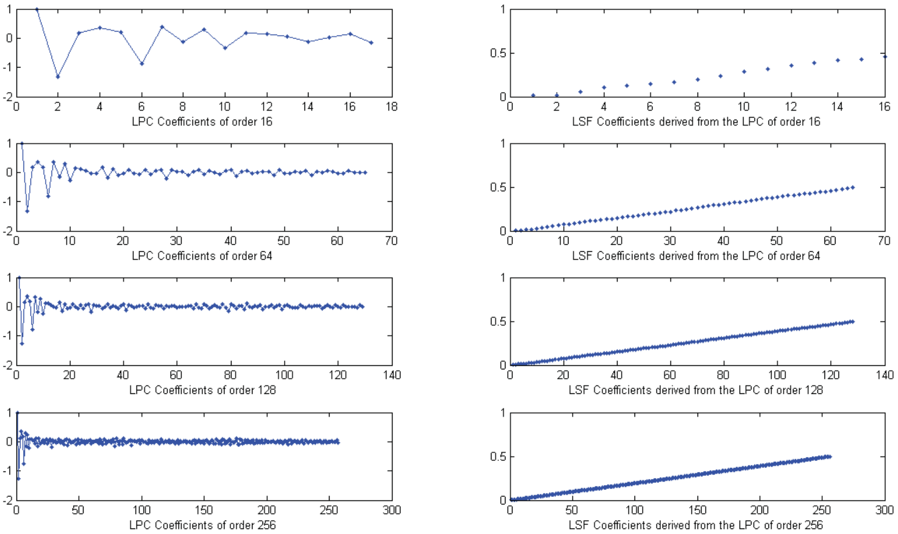

Figure 3.2 Examples of LSF coefficients derived from LPC of different orders

- the reconstructed LPC all-pole filter maintains its minimum phase property, and
- LSFs are related with the formant frequencies. [15]

3.4 Maximum Likelihood Parameter Estimation

As long as we have calculated the training features for each speaker, our goal is to find the set of λ that best matches the training features. There are several techniques for estimating the parameters of the GMM, but the most well-established if the Maximum Likelihood (ML) Estimation. For a sequence of T training vectors $X = \vec{x_1}, ..., \vec{x_T}$, the GMM likelihood can be written as

$$p(X|\lambda) = \prod_{t=1}^{T} p(\vec{x_t}|\lambda). \tag{3.9}$$

Unfortunately, this expression is not a linear function of the parameters of the GMM, thus direct maximization is not possible. However, we can use the Expectation Maximization(EM) algorithm instead to obtain the parameter estimates by iteration.

The basic idea of the EM algorithm is, in every iteration, to estimate a new model $\bar{\lambda}$ such that if λ the model of the previous iteration, $p(X|\bar{\lambda}) \geq p(X|\lambda)$. The new model then becomes the initial one for the next iteration until some convergence threshold is reached. On each EM iteration, the following re-estimation formulas are used which guarantee a monotonic increase in the model's likelihood rate:

Mixture Weights:

$$\bar{p}_i = \frac{1}{T}\sum_{t=1}^{T} p(i|\vec{x_t}, \lambda) \tag{3.10}$$

Means:

$$\vec{\bar{m}}_i = \frac{\sum_{t=1}^{T} p(i|\vec{x}_t, \lambda)\vec{x}_t}{\sum_{t=1}^{T} p(i|\vec{x}_t, \lambda)} \tag{3.11}$$

Variances:

$$\bar{\sigma}_i^2 = \frac{\sum_{t=1}^{T} p(i|\vec{x}_t, \lambda)x_t^2}{\sum_{t=1}^{T} p(i|\vec{x}_t, \lambda)} - \bar{\mu}_i^2 \tag{3.12}$$

where $\sigma_i^2, x_t, and \mu_i$ refer to arbitrary elements of the vectors $\vec{\sigma}_i, \vec{x}_t, and \vec{\mu}_i$ respectively. The a posteriori probability for class/speaker i is given by

$$p(i|\vec{x}_t, \lambda) = \frac{p_i b_i(\vec{x}_t)}{\sum_{k=1}^{M} p_k b_k(\vec{x}_t)}. \tag{3.13}$$

3.5 Speaker Identification

For speaker identification, a group of S speakers $S = 1, 2, ..., S$ is represented by GMM's $\lambda_1, \lambda_2, ..., \lambda_S$. The objective is to find the speaker model which has the maximum a posteriori probability for a given observation sequence. Formally, under the

Maximum Likelihood framework

$$\hat{S} = \arg \max_{1 \leq k \leq S} p(X|\lambda_k). \tag{3.14}$$

Using logarithms and the independence between the observations, the speaker identification system computes

$$\hat{S} = \arg \max_{1 \leq k \leq S} \sum_{t=1}^{T} p(\vec{x}_t|\lambda_k). \tag{3.15}$$

in which $p(\vec{x}_t|\lambda_k)$ is given in 3.4.

Chapter 4

Speaker Identification Using Compressive Sensing

In this chapter we will describe two of the three signal reconstruction algorithms used for our experiments. The third one was Basis Pursuit and was described in the second chapter. Moreover, the identification process will be described which is used when applying CS theory on speaker identification application.

4.1 Orthogonal Matching Pursuit

Let's consider a real valued signal s that is m-sparse. That means that s consists of m nonzero elements. Moreover, let's consider a $N \times d$ measurement matrix Φ in $\Re^d$, independent to the signal s. In order to take measurements of the signal, we have to take the inner products of the signal with every row of the measurement matrix ϕ_n with $n = 1, ..., N$. We cannot take fewer than m measurements if we want to reconstruct the signal, and we can reconstruct it by solving the following statement:

$$\min_f \|f\|_1 \quad subject \quad to \quad \langle f, \phi_n \rangle = \langle s, \phi_n \rangle \quad for \quad n = 1, 2, ..., N. \tag{4.1}$$

providing that the measurement matrix is known. It has been shown that this recovery is possible by Candes-Tao [10] and of Rudelson-Vershynin [11] who have published Theorem 1.

Theorem 1 *Let $N \geq Km \ln(d/m)$, and draw N vectors $\phi_1, \phi_2, ..., \phi_N$ independently from the standard Gaussian distribution on $\Re^d$. The following statement is true with probability exceeding $1 - e^{kN}$. It is possible to reconstruct every m-sparse signal s in $\Re^d$ from the data $\{\langle s, \phi_n \rangle : n = 1, 2, ..., N\}$. The numbers K and k are universal constants.*

Particularly, Gaussian measurement vectors succeed for every m-sparse signal with high probability. The above statement is the main idea of the Basis Pursuit algorithm for signal reconstruction but although it is a linear programming problem, it may take a long time to solve and if optimization algorithms do not exist, it takes a lot of effort to construct or implement one.

For the above reasons, Orthogonal Matching Pursuit Algorithm (OMP) was used for signal reconstruction. The advantages of this algorithm is its ease of implementation and speed, although the performance of OMP was considered to degrade in cases that are not simple [13]. This choice was made because the negative results that were published for OMP were not misleading. Experiments have shown that OMP is capable to recover a m-sparse signal when the number of measurements is a multiple of m.

Theorem 2 *Fix $\delta \in (0, 0.36)$, and choose $N \geq Km \ln(d/\delta)$. Suppose that s is an arbitrary m-sparse signal in $\Re^d$. Draw N measurement vectors $\phi_1, \phi_2, ..., \phi_N$ independently from the standard Gaussian distribution on $\Re^d$. Given the data $\{\langle s, \phi_n \rangle : n = 1, 2, ..., N\}$, Orthogonal Matching Pursuit can reconstruct the signal with probability* *exceeding* $1-2\delta$. *For this theoretical result, it suffices that* $K = 20$. *When* m *is large, it suffices to take* $K \approx 4$.

As we mentioned, our signal is a m-sparse vector in $\Re^d$ and measurement matrix is Φ which is $N \times d$. The actual measurements will be an N-dimensional vector $v = \Phi s$. That means that vector v is a linear combination of m columns from Φ since s is a m-sparse signal. In order to recover the original signal, we have to determine which m columns take part in the measurement vector v and pick them in a greedy fashion. The basic idea of the OMP algorithm is to choose the column of Φ that is most strongly correlated with the remaining part of v in each iteration. Then subtract off its contribution to v and iterate on the residual. After m iterations, the algorithm will have identified the correct set of columns.

Analytically, the input of the algorithm is a $N \times d$ measurement matrix Φ, a N-dimensional data vector v and m (the sparsity level of the original signal). For the signal recovery the following steps are conducted in each iteration:

1. Initialize the residual as the measurements $r_0 = v$ and the the index set $\Lambda_0 = \emptyset$.

2. Find the solution in the optimization problem

$$\lambda_t = \arg \max_{j=1,\ldots,d} |\langle r_{t-1}, \phi_j \rangle|$$

   If the solutions are more than one, process the solutions deterministically.

3. Add the solution found in the index set, and ϕ_{λ_t} in the matrix of the chosen atoms.

4. Find a new signal estimate by solving

$$x_t = \arg \min_x \|\Phi x - v\|_2.$$

5. The new data approximation is $\alpha_t = \Phi_t x_t$ and the new residual is $r_t = v - \alpha_t$.

6. If iterations are not completed, return to Step 2.

7. The indices of the nonzero elements of the original signal are listed in the index set and the value of the signal in λ_j equals to the jth component of x_t.

The residual r_t in every iteration is orthogonal to the columns of the measurement matrix, thus the algorithm selects a new atom in each step and Φ has full column rank. OMP is a relatively-efficient iterative algorithm that produces one component of $\bullet x_t$ in each iteration, and thus allows for simple control of the sparsity of the signal. As the true sparsity is often unknown, the OMP algorithm is run for a pre-determined number of iterations, K, resulting in x being K-sparse.

4.2 SL0 complex

The main idea of the Smoothed $L0$ algorithm is to find a sparse solution for the optimization problem $As = x$ by directly minimizing the l_0 norm, that is the amount of the nonzero elements of the signal. It is called smooth because the l_0 norm is not a continuous function but if we try to minimize it we have to find a smooth approximation of it, in order to use gradient based methods and solve the sensitivity to noise.

This problem is said to be intractable as the dimensions increase because it requires a combinatorial search. This is why researchers tried other forms of solutions like minimizing the l_1 norm (Basis Pursuit). This solution is easy to find by Linear Programming and the algorithm is based on the idea that the Basic Pursuit's optimal solution is also the minimum l_0 norm minimum.

The l_0 norm of signal $s = [s_1, ..., s_n]^T$ is defined as the number of the nonzero

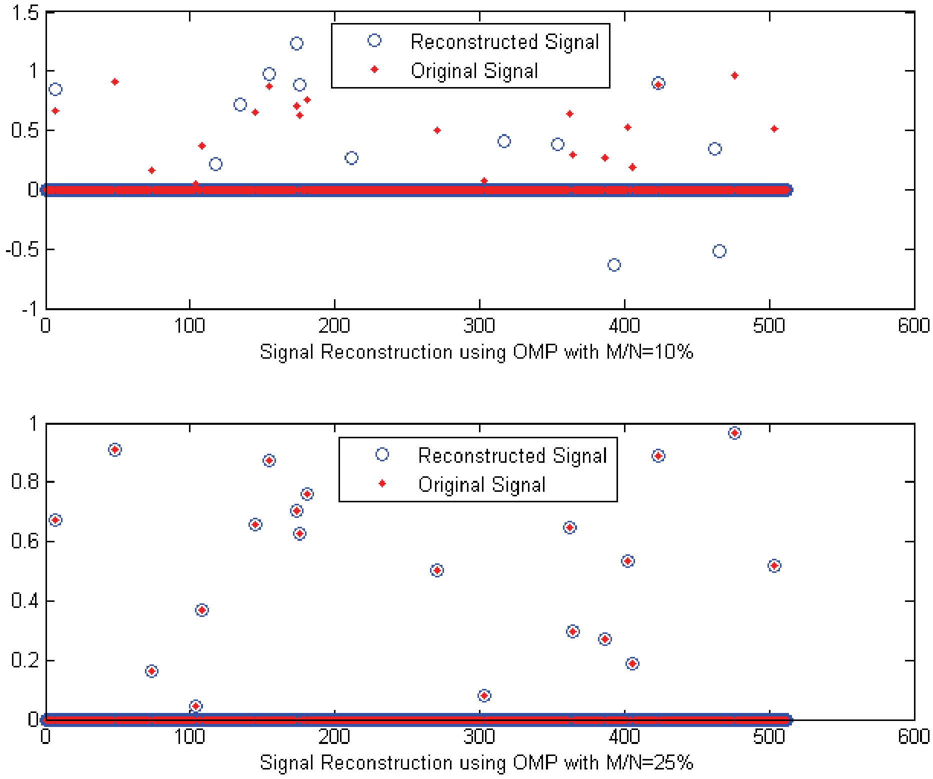

Figure 4.1 Signal Reconstruction using OMP.

elements of s. We can form this in the following equations:

$$v(s) = \begin{cases} 1 & s \neq 0 \\ 0 & s = 0 \end{cases} \tag{4.2}$$

then

$$\|s\|_0 = \sum_{i=1}^{n} v(s_i). \tag{4.3}$$

Equation 4.2 is the main problem of the minimization due to its discontinuities. This is why it is replaced by other functions such as zero mean Gaussian, in order to be differentiable. If we define:

$$f_\sigma = \exp(-s^2/2\sigma^2), \tag{4.4}$$

we have:

$$\lim \sigma \to 0 f_sigma(s) = \begin{cases} 1 & s = 0 \\ 0 & s \neq 0 \end{cases} = 1 - v(s). \tag{4.5}$$

If we define a function

$$F_\sigma(s) = \sum_{i=1}^{n} f_\sigma(s_i), \tag{4.6}$$

then the limit in 4.5 will be:

$$\lim \sigma \to 0 F_\sigma(s) = \sum_{i=1}^{n} (1 - v(s_i)) = n - \|s\|_0. \tag{4.7}$$

with $\|s\|_0 \approx n - F_\sigma(s)$. The value of σ specifies the trade-off between accuracy and smoothness of the approximation. The smaller the σ the better the approximation to the real minimum value of the l_0 norm, and the larger the σ the smoother the approximation. For small values of σ F_σ has a lot of local maxima which makes the maximization very difficult.

In order not to get trapped in a local maxima, the algorithm initializes the value of σ at ∞ and then gradually decreases it. The choice of σ to be initialized as ∞ was because of the the following theorem:

Theorem 3 *The solution of the problem: Maximize $F_\sigma(s)$ subject to $As = x$, where $\sigma \to \infty$, is the minimum l_2 norm solution of $As = x$, that is $s = A^T(AA^T)^{-1}x$.*

By maximizing $F_\sigma(s)$, s gets as sparse as possible.

The final SL0 algorithm is :

- Initialization
  1. Choose an arbitrary solution from the feasible set S, v_0.
  2. Choose a suitable decreasing sequence for $\sigma, [\sigma_1 ... \sigma_K]$
- for k = 1,...,K:
  1. Let $\sigma = \sigma_k$.
  2. Maximize (approximately) the function F_σ on the feasible set S using L iterations of the steepest ascent algorithm (followed by projection onto the feasible set):
     - Initialization: $s = v_{k-1}$.
     - for j = 1 ... L:
       (a) Let: $\Delta s = [s_1 \exp(-s_1^2/2\sigma_k^2), ..., s_1 \exp(-s_n^2/2\sigma_k^2)]^T$.

       (b) Let $s \leftarrow s - \mu\Delta s$.

       (c) Project s back onto the feasible set S:

       $$s \leftarrow s - A^T(AA^T)^{-1}(As - x)$$
  3. Set $v_k = s$.
- Final answer is $s = v_l$.

[14]

One advantage of the SL0 algorithm is that we do not need to spend much time to wait for it to converge. All the algorithm needs is to be near the absolute maximum of F_σ for escaping the local maxima, which can be achieved in just a few iterations (small L).

4.3 Speaker Identification Using CS

In order to perform speaker identification using compressed sensing, we have to construct a codebook of basis matrices from speech training data for each of the S speakers that we wish to identify, just like it was mentioned in section 2.2. This is essentially formed by performing a codebook of the LSF vectors of each speaker separately. This process is in fact similar to the GMM training for speaker identification, and is based on the assumption that LSF"s are suitable feature vectors for the classification task.

A simple way to do classification using compressed sensing is to find a basis for each of the C classes of interest, and then reconstruct a sparse vector from each of the class bases. The measured signal is then said to come from the class that produced the sparsest recovered vector. This can work well, but requires that the class bases be incoherent. In our case, the class bases would be the h_l"s for each speaker. Unfortunately these bases are far from incoherent. We thus need to find another method to perform speaker identification, and we proceed in the following manner.

We first find a residual excitation vector for each basis matrix from each speaker"s codebook using

$$\hat{r}_{s,l} = \arg\min_{r} \|y - \Phi h_{s,l} r\|_2 \quad such \quad that \quad \|r\|_0 = K. \tag{4.8}$$

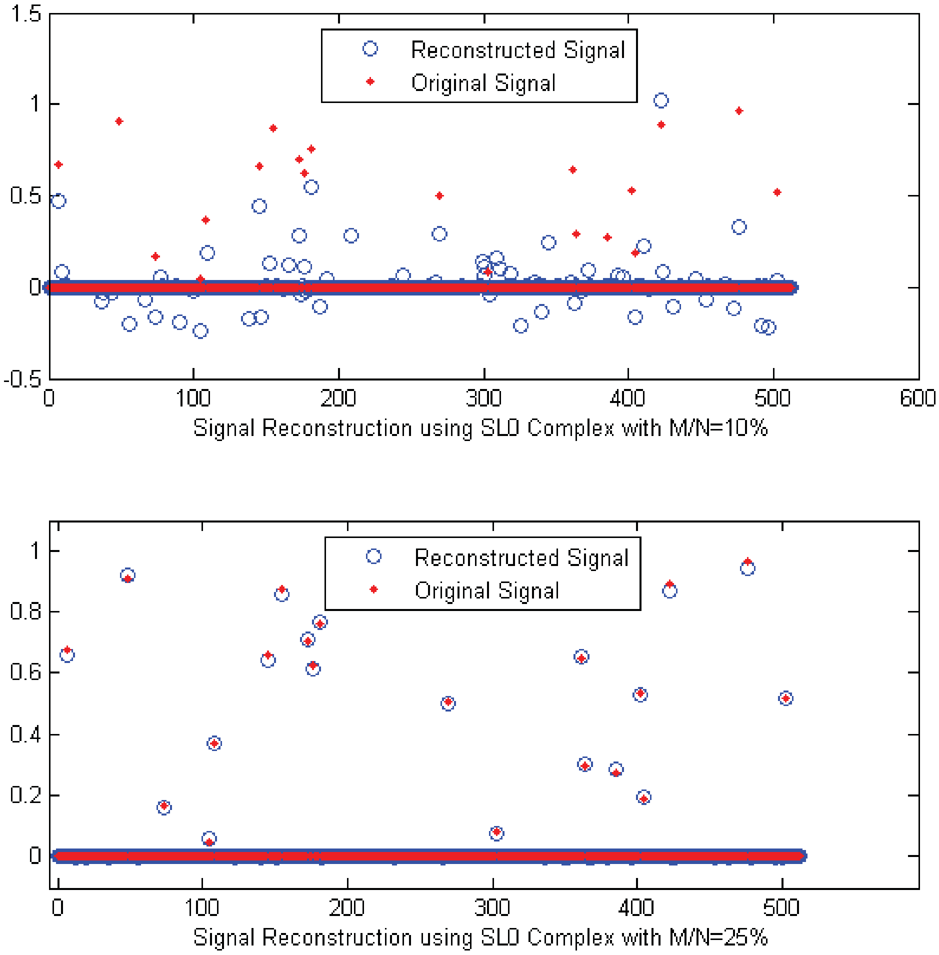

Figure 4.2 Signal Reconstruction using SL0 Complex.

Once these have been found, we then calculate

$$d_s = \min_l \|y - \Phi h_{s,l} \hat{r}_{s,l}\|_2, \tag{4.9}$$

which represents the minimum distance between the measurements y and measurements from the reconstructions from the s-th speaker”s codebook.

Now, let $d_{i,s}$ be the d_s calculated for the i-th frame. The actual speaker s* in the i-th frame should have the smallest distance, so that

$$d_{i,s*} < d_{i,s}, \quad \forall s \neq s* \,. \tag{4.10}$$

Thus if this is true we have chosen the correct speaker, and if not we have an error.

In practice, we can greatly improve the reliability of speaker identification by considering n frames at a time. This is based on the fact that the speaker will not change from frame to frame, and will rather be constant for a group of frames. Thus we use a sliding window to determine the most probable speaker as

$$\hat{s} = \arg\min_s \sum_{j=i-(n-1)}^{i} d_{i,s}, \tag{4.11}$$

to determine the speaker. Obviously if $\hat{s} \neq s*$ then the identification has failed for this particular segment. This approach is the same as the segment-based approach for identification.

Chapter 5

Results

In this chapter we discuss the results of our research. All the experiments were conducted using speech signals from the VOICES corpus, available by OGI's CSLU [16]. The speech signals, originally sampled at 22 kHz, were downsampled at 8 kHz, with N = 320 samples per frame and 50% overlapping between frames. The training data consisted of 30 sentences from 12 speakers, resulting in around 6000 frames per speaker. All the the simulations were performed using 10 sentences for each speaker different to those used in the training process. This provided more that 2000 frames of test data per speaker.

We performed the experiments using three types of signal reconstruction algorithms. SL0 complex, Basis Pursuit and Orthogonal Matching Pursuit. SL0 complex algorithm was implemented by Massoud Babaie-Zadeh and Hossein Mohimani and is available at http://ee.sharif.ir/ SLzero , Basis Pursuit was implemented by Justin Romberg, Caltech and is available in the toolbox l1-magic, and finally the OMP algorithm was implemented in Stanford and is available in the toolbox SparseLab.

For each of these cases experiments were conducted using a codebook of size L = 8, and the process described in chapter 4. For the case of OMP, experiments were conducted for larger codebooks too, and for different numbers of iterations.

5.1 SL0 complex and Basis Pursuit

As a first step, we tested the performance of the sparsity-based speaker identification for SL0 Complex. The measurement matrix Φ consisted of $M \times N$ Gaussian samples with zero mean and unit variance. The performance measure used was the probability of correct identification of the speaker using 4.11 with n equal to 140 frames (2.8 seconds), and averaged over all 12 speakers. The sparsity level range tested, was from 50% of the signal to 90% of the signal. The probability of correct speaker identification using these two reconstruction algorithm is shown below. The same conditions were tested using Orthogonal Matching Pursuit for reconstruction. The results were better so we continued the experiments only with OMP.

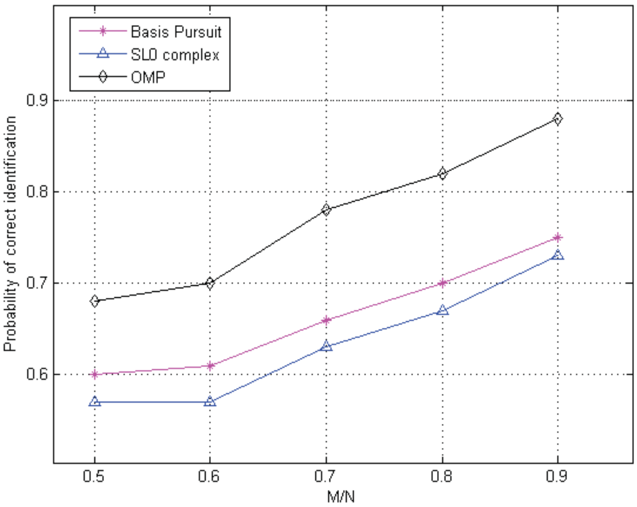

Figure 5.1 Probability of correct identification versus the number of iterations of the three reconstruction algorithms for a codebook size of 8.

5.2 Orthogonal Matching Pursuit

Initially, we tested the performance of the sparsity-based speaker identification. The measurement matrix Φ consisted of $M \times N$ Gaussian samples with zero mean and unit variance. The performance measure used was the probability of correct identification of the speaker using 4.11 with n equal to 140 frames (2.8 seconds), and averaged over all 12 speakers.

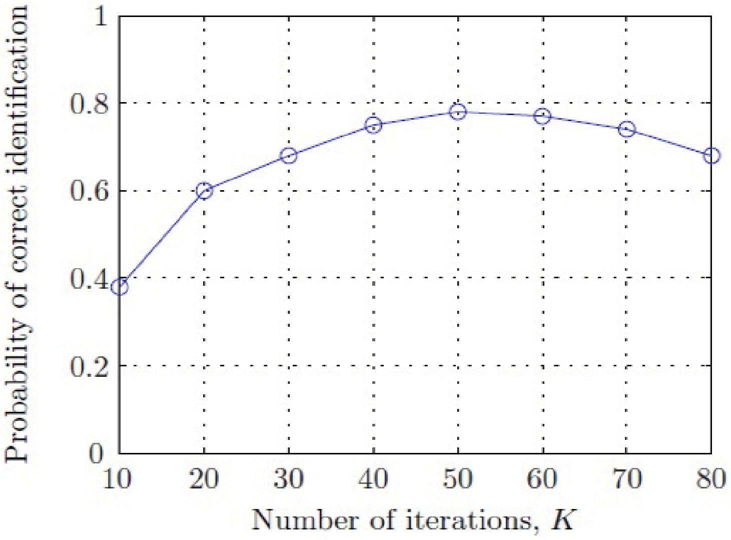

Figure 5.2 Probability of correct identification versus the number of iterations of the reconstruction algorithm for a codebook size of 16. The number of measurements is equal to half the Nyquist rate (M = N/2).

As an initial investigation, we looked at the effect of the number of iterations of the OMP algorithm, K, on our proposed method for M = N/2 measurements per frame. The results are shown in Figure 5.2 for a codebook size of $L = 16$. The identification process can be seen to not be very sensitive to K around $K \approx M/4$, and it is this value for K that we used in the rest of this work.

Figure 5.3 presents the performance of our proposed method as the number of

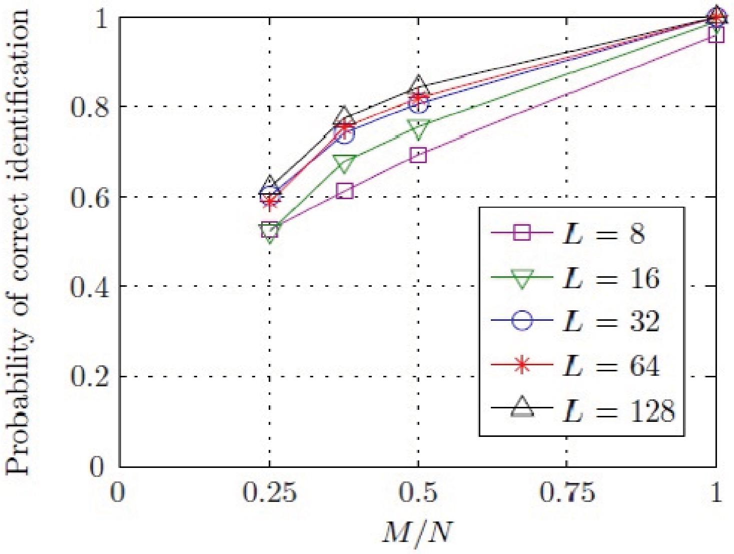

Figure 5.3 Probability of correct identification versus the number of measurements for various speaker codebook sizes. The number of iterations of the reconstruction algorithm is equal to one quarter of the number of measurements (K = M/4).

measurements M and the size of each speaker"s codebook L are varied. These results are intuitively satisfying; as M decreases, the reconstruction quality will degrade, and thus the probability of correct identification decreases. The results for $M/N = 1$ do not use compressed sensing, and this can be thought of as the best possible performance. The performance also improves as L increases, although there seems to be diminishing returns after L = 32, and each increase in L increases the complexity of the identification process. Thus for L = 32 with 50% measurements the probability of correct identification is about 0.8, and if the measurements are lowered to 25% this probability drops to about 0.6.

All the previous results are for noise-free speech. We also explored the effect of additive white Gaussian noise on the probability of correct identification for the $L = 32$, $M = N/2$ case, and this is presented in Figure 5.4, along with the corresponding

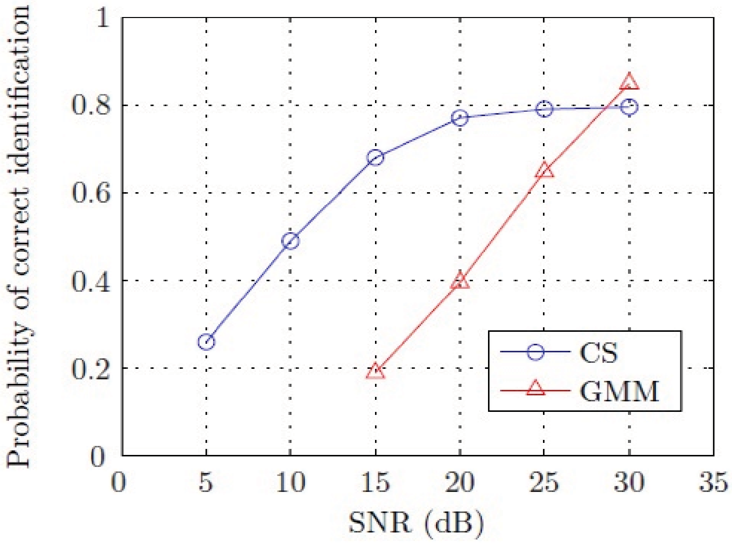

Figure 5.4 Probability of correct identification versus the signal to noise ratio of the speech signal for the Gaussian mixture model (GMM) method and the proposed compressed sensing (CS) method.

results for the GMM method discussed in Chapter 3. The GMM method used 32 diagonal mixtures and the same training and testing data as the compressed sensing (CS) method. It is clear that the CS method outperforms the GMM method once the signal to noise ratio (SNR) is below 30dB. In fact, there is very little loss in performance for the CS method down to an SNR of 20dB, and even an SNR of 15dB only affects the performance mildly.

Chapter 6

Discussion

In this work, we examined the effect of Compressive Sensing Theory on a speaker identification application using three different algorithms for signal reconstruction. The experiments showed that speaker identification is possible using Orthogonal Matching Pursuit although the probability of a right recognition could be improved.

Assuming the two methods (CS and baseline GMM method) were used in a sensor with limited power resources, the CS method would require slightly more processing than the GMM method in the sensor, as it needs to calculate the measurements, although efficient measurement methods do exist. However the CS method would require half the bandwidth of that of the GMM method to transmit the measurements back to a central processor. This transmission power gain and the robustness to noise for the CS method come at the cost of increased complexity in the speaker identification algorithm, but for many applications this is acceptable.

We have presented a novel method for speaker identification based on a sparse signal model and the use of compressed sensing. The use of compressed sensing permits the use of less transmission power for the sensor recording the voice. Additionally, our method has been shown to be robust to noise in the recorded speech signal. This is an encouraging result, and warrants further investigation.

In order to have a more robust recognition we could utilize a part of the idea of [17]. The idea of this publication is on the one hand to enhance the training database by adding samples with various amounts of noise. On the other hand, instead of using the whole testing vector used to represent the speech signals, they use only the parts that have a strong connection with the corresponding trained system.

The motive to use this specific publication was that the authors try to solve the problem of speaker identification in noisy environments, and we noticed from the experiments conducted that CS algorithms are already robust in noise. It describes a method that combines a multicondition model training and missing feature theory to model noise with unknown spectral characteristics. Multicondition training is performed by generating multiple copies of the original training set which contains only clean speech. Each copy gets corrupted by additive noise with specific characteristics. Particularly, white noise at various signal-to-noise ratios is added to simulate the corruption. The augmentation of the training set results in a new likelihood function. If we assume that Φ_0 is the original training set and $\Phi_i, i = 1, ..., L$ the corrupted ones, then the likelihood will be:

$$p(X|S) = \sum_{l=0}^{L} p(X|S, \Phi_l)p(\Phi_l|S) \tag{6.1}$$

where $p(X|S, \Phi_l)$ is the likelihood function of frame vector X trained on set Φ_l, and $P(\Phi_l|S)$ is the prior probability of the noise condition Φ_l for speaker S. This equation is called a multicondition model.

Another step to make the speaker identification process further robust, is to choose which part of the testing vector is going to participate in the process. One way to do that is to ignore the heavily mismatched subbands and focus the score only on the matching subbands. If $X = (x_1, x_2, ..., x_N)$ a test frame vector and $X_l \subset X$ be a subset of X containing all the subband features corrupted at noise condition Φ_l. Then (6.1) will be refined as:

$$p(X|S) = \sum_{l=0}^{L} p(X_l|S, \Phi_l)P(\Phi_l|S) \tag{6.2}$$

where $p(X_l|S, \Phi_l)$ is the marginal likelihood of the matching feature subset X_l, derived from $p(X|S, \Phi_l)$ with the mismatched subbands features ignored to improve mismatch robustness between test frame X and the training noise condition Φ_l. The approach expressed in 6.2 extends the traditional approaches because traditional approaches determine the importance of a feature against the clean data, while the new approach assesses this against the data containing variable degrees of corruption. This allows the model to use not only clean data, but also noisy features that match the noisy training conditions for recognition.

The posterior union model introduced by the authors of [17] is formulated finally as:

$$P(S, \Phi_l|X_{sub}(M)) = \frac{p(X_{sub}(M)|S, \Phi_l)P(S, \Phi_l)}{\sum_{S',l'} p(X_{sub}(M)|S', \Phi_{l'})P(S, \Phi_{l'})} \tag{6.3}$$

where $X_{sub}(M)$ is the subset of test frame X with length M and is considered as a function of the length of the subset of the subbands we keep and not of the content.

Based on the basic idea of 6.2 we could expand the algorithms used for speaker identification of this work. Instead of using a dictionary made of clean speech, we could apply the same principle as the multicondition model and use a larger dictionary with simulated corruption. Thus, even if CS algorithms are robust to noise because they ignore very small elements considered as noise, if they cannot be ignored they can be recognized correctly.

Bibliography

[1] A. Griffin, E. Karamichali and A. Mouchtaris, ”Speaker Identification Using Sparsely Excited Speech Signals and Compressed Sensing”, Proceedings of the 2010 European Signal Processing Conference (EUSIPCO-2010), Aalborg, Denmark, 23-27 August, 2010.

[2] R.G. Baraniuk, “Compressive sensing,” IEEE Sig.Proc. Mag., pp. 118•120, July 2007.

[3] Emmanuel J. Candes and Michael B. Wakin, ”An Introduction To Compressive Sensing Theroy”, IEEE SIGNAL PROCESSING MAGAZINE, pp. 21-30, Mar. 2008.

[4] A. Griffin, T. Hirvonen, A. Mouchtaris, and P. Tsakalides, “Encoding the sinusoidal model of an audio signal using compressed sensing,” in Proc. IEEE Int. Conf. on Multimedia Engineering (ICME”09), New York, NY, USA, June 2009.

[5] T. V. Sreenivas and W. B. Kleijn, “Compressive sensing for sparsely excited speech signals,” in Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP), Taipei, Taiwan, 2009, pp. 4125•4128.

[6] D. Giacobello, M. G. Christensen, J. Dahl, S. H. Jensen, and M. Mooren, ”Sparse Linear predictors for speech coding”, in Proc. Interspeech, 2008, pp. 1353-1356.

[7] D. Giacobello, M. G. Christensen, M. N. Murthi, S. H. Jensen, and M. Mooren, ”Speech Coding based on Sparse Linear predictors”, in Proc. EUSIPCO, 2009, pp. 2524-2528.

[8] Douglas A. Reynolds, and Richard C. Rose, ”Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models”, IEEE Transactions on Speech and Audio Processing, Vol. 3, No. 1, pp 75-83, Jan. 1995.

[9] E. Erzin, and A. E. Cetin, ”Line Spectral Frequency Representation of subbands for speech recognition”, Signal Processing, Vol. 44, pp. 117-119, Mar. 1995.

[10] E. J. Candes and T. Tao, ”Decoding by linear programming.”, IEEE Trans. Info. Theory, Vol. 51, pp. 4203•4215, Dec. 2005.

[11] M. Rudelson and R. Veshynin, ”Geometric approach to error correcting codes and reconstruction of signals.”, Int. Math. Res. Not., Vol. 64, pp. 4019•4041, 2005.

[12] J. A. Tropp and A. C. Gilbert, ”Signal recovery from random measurements via orthogonal matching pursuit”, IEEE Trans. Inform. Theory, vol. 53, no. 12, pp. 4655-4666, December 2007.

[13] R. DeVore and V. N. Temlyakov, ”Some remarks on greedy algorithms.”, Adv. Comput. Math., Vol. 5, pp. 173•187, 1996.

[14] G. Hosein Mohimani1, Massoud Babaie-Zadeh1 and Christian Jutten, ”A fast approach for overcomplete sparse decomposition based on smoothed l_0 norm”, IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 57, NO. 1, pp.289-301, Jan. 2009.

[15] K. K. Paliwal, ”On the use of Line Spectral Frequency parameters for speech recognition”, Digital Signal Processing, A Review J., Vol. 2, pp.80-87, April 1992.

[16] A. Kain, ”High Resolution Voice Transformation”, Ph.D. thesis, OGI School of Science and Engineering at Oregon Health and Science University, October 2001.

[17] J. Ming, T. J. Hazen, J. R. Glass, and D. A. Reynolds, “Robust speaker recognition in noisy conditions,” IEEE Trans. on Audio, Speech and Language Processing, vol. 15(5), pp. 1711•1723, July 2007.