New insights on speech signal modeling in a Bayesian framework approach

ADRIÀ CASAMITJANA DÍAZ

Master’s Degree Project
Stockholm, Sweden 2015

XR-EE-KT 2015:004
Abstract

Speech signal processing has always brought a lot of attention from the communication theory community. Speech communication, as the most natural way of communication between humans, is, indeed, a mature research topic with rich literature from even before the first digital hardware appeared to nowadays. The continuously increasing telephony market brought special attention to the discipline during the 80’s and 90’s, specially in speech coding and speech enhancement, where the most significant contributions were made. More recently, due to the appearance of novel signal processing techniques, the standard methods are being questioned. Sparse representation of signals and compressed sensing made significant contributions to the discipline, through a better representation of signals and more efficient processing techniques. In this thesis, standard speech modeling techniques are revisited. Firstly, a representation of the speech signal through the line spectral frequencies (LSF) is presented, with an extended stability analysis. Moreover, a new Bayesian framework to time-varying linear prediction (TVLP) is shown, with the analysis of different methods. Finally, a theoretical basis for speech denoising is presented and analyzed. At the end of the thesis, the reader will have a broader view of the speech signal processing discipline with new insights that can improve the standard methodology.
Acknowledgment

This thesis is the final step after 6 intensive years of study, starting at UPC (Barcelona) and finishing at KTH (Stockholm). During this time, I have received support from many people and I would like to thank everyone that has helped me to improve both academically and personally. Firstly, I would like to thank my supervisor and friend, Dr. Saikat Chatterjee, for giving me the opportunity to write this thesis with him. Thanks to his support and engagement I could put myself through a research-based project where what I price more is the learning process I have been involved. I would always be grateful for his endless help. In the same line, I thank Martin for his help and interesting discussions in the department as well as my examiner Dr. Mikael Skoglund. Special attention to my family, my parents Pilar and Jaume and my sister Laura, that have been very supportive during my university studies. Their unconditional support and advise have been really helpful. I would also like to thank my friends Martí, Àngel, Oriol, Arnau, Leyre, Piber, Angie, Pau, Jordi for our strong friendship that has last no matter the physical distance we had in between, and Mariona, for her continuous support during the early years of my degree. Another special mention to Raül, for every great moment we have shared in Barcelona. Finally, all my friends from high school that have been always very receptive when I was back home.
Contents

1 Introduction 3
  1.1 Motivation 4
  1.2 Outline and Contributions of the thesis 4

2 Background 6
  2.1 Speech Production and Hearing 8
    2.1.1 Physical Process 8
    2.1.2 The Source-filter Model of Speech-Production 9
  2.2 Linear Prediction of Speech 12
    2.2.1 Stability of the L2-norm 12
    2.2.2 Time Varying Linear Prediction 13
  2.3 Sparsity in Speech Signal Processing 15
    2.3.1 Greedy algorithms 15
    2.3.2 Convex relaxation 15
    2.3.3 Sparse Linear Prediction 16
  2.4 Bayesian Inference 16
    2.4.1 Sparse Bayesian Learning 18

3 Stability analysis 21
  3.1 Introduction 21
  3.2 Problem formulation 21
  3.3 Minimum-phase LP through LSF coefficients 23
  3.4 Performance analysis 27
    3.4.1 Spectral Envelope Modeling 27
  3.5 Peak-avoiding property 30
    3.5.1 Future Work 32

4 Time Varying Linear Prediction 34
  4.1 Introduction 34
  4.2 Time-varying linear prediction system 35
  4.3 Bayesian learning 36
    4.3.1 Underdetermined setup 36
    4.3.2 Determined setup 37
  4.4 Experiments 38
    4.4.1 Synthetic signals 38
    4.4.2 Real speech signals 39
  4.5 Conclusions and Future work 40
  4.6 Derivation of update equations 41
Chapter 1

Introduction

1.1 Motivation

Speech signal processing is one of the most evergreen disciplines within the signal processing and communication theory fields. It has numerous applications as it is the most natural way of communication between humans. The analysis of speech signals has been widely investigated in the past. However, the smart structure of languages that humanity has created is still a hard topic to be investigated in the future. In Fig. 1.1 we see that speech signals features can be split into several layers either language dependent or independent. Exploiting these characteristics and the relationship between them may improve the standard techniques and current systems as well as help to create new applications.

Figure 1.1: Speech signal
1.2 Outline and Contributions of the thesis

New signal processing techniques, specially regarding sparse representations of signals and compressed sensing, can be shown to improve the standard methodology in speech communication. Also, adapting the standard methods to a totally Bayesian framework is believed to improve the performance. A wide and critical analysis to the standard methods and its comparison to some novel techniques is performed at the beginning. Some new theoretical methods are presented and analyzed with proposed applications.

Chapter 2: Background

In this chapter we will introduce the old standard techniques of speech modeling. We will state the problems and benefits of the standard methodology and motivate why it should be improved. To complete the chapter, other general signal processing techniques will be introduced in order to help the further understanding of the thesis. This chapter can be summarized as:

- Introduction to the physics of speech production.
- Introduction to the standard speech modeling techniques.
- Introduction to sparse representation of signals.
- Introduction to Bayesian learning.

Chapter 3: Stability analysis

As a natural process, the generative process of speech is stable. This quality can’t be ensured for any kind of speech analysis method, and to the knowledge of the writer, there is no general method that is able to ensure it. Several applications of speech communications depend on the stability of the process (e.g. speech coding). The chapter can be summarized as:

- State the necessary conditions for stability.
- Analyze the standard methods of speech modeling in terms of stability.
- Propose a new general method of speech modeling with focus to the stability constraint.

Chapter 4: Time-varying linear prediction

Linear prediction (LP) is the most common technique in speech analysis and it has several applications, such as recognition or coding. However, this method assumes a short-time stationarity of the signal and a constant representation of the vocal tract. In the literature, one can find several solutions that try to overcome this problem. We propose a novel method using a Bayesian framework with a sparse bayesian learning approach. This approach has several benefits that will be discussed as well as several interesting applications to be analyzed in the future. The chapter can be summarized as:

- State the time-varying linear prediction approach.
• Analyze the standard methods in the literature.

• Develop a Bayesian framework to the time-varying linear prediction problem.

• Analysis and comparison of all the methods stated.

This chapter can be mainly found in the following paper (submitted):


Chapter 5: Speech Enhancement

A really interesting and challenging application in speech communication is speech enhancement. It includes several degradation sources: from noise to the room effect. In this chapter we will focus on speech denoising. We will introduce a novel framework to the problem using Bayesian learning, which will be a basis for further investigations. This chapter can be summarized as:

• State the speech denoising problem

• Describe our methodology

• Statistical modeling of the problem

• Analysis of the method in a particular case: Gaussian noise

• State the further modifications for a general case.
Chapter 2

Background

Speech signal processing is a field with a major research in the past and thus, a lot of standarized techniques. Recent research focuses in bringing new signal processing techniques that could overcome the limitations of the standarized ones. It has applications that range from telephony, which now allows to work with higher speech bandwidths, to information processing and analysis.

2.1 Speech Production and Hearing

Understanding the physics of any problem is important to design signal processing techniques and optimize algorithms. Speech signal processing is largely based on the knowledge of speech production and hearing.

2.1.1 Physical Process

Speech is produced with the help of several organs, and each one contributes in a different way Fig. [2.1]. The main components are:

- lungs: generate the energy
- trachea: transport the energy
- vocal cords: generate the signal
- vocal tract (pharynx, oral and nasal cavities): it is in charge of shaping the signal so that we can produce different sounds (phones), and can be seen as a time varying acoustic filter.

By contraction, the lungs produce an airflow that goes through the larynx, which mainly control the stream of air that enters the vocal tract via the vocal cords. The vocal cords are on charge of generating different kind of sounds. Voiced sounds are generated by the periodic interruption (vibration) of the airflow stream from the larynx. The area between the vocal cords, which is called the glottis, is opening and closing due to the increasing or decreasing air’s local pressure and causing the airflow interruption. Unvoiced sounds are generated by a non-periodic turbulent airflow. Finally plosive sounds are caused by building up the air pressure behind some constriction before the vocal tract followed by
a sudden opening. They can be described as a mixture of voiced and unvoiced sounds.

Vocal tract consists of pharynx, oral and nasal cavities. It can be seen as an acoustic resonator that changes the resonance frequencies by adapting the shape of the vocal tract through the different articulators (tongue, teeth, lips, velum, etc.)

Hearing anatomy is very important in the development of speech processing techniques. The whole hearing organ works as a sensor that receives the vibration of the acoustic waves, adapts it and transforms it into impulses that go to our nerve system. This last step is performed by the organ of Corti.

The main features of the hearing system are:

- Spectral resolution: hearing system is not able to distinguish with very close frequencies. One consequence of it is that one can model the hearing system as a filter bank and each filter applied in different frequency bands, separated in a non-uniform way by following the so-called Bark-scale.

- Masking: it occurs when a dominant sound renders a weaker sound inaudible. It is useful in several speech disciplines, such as perceptual coding.

- Frequency response: due to its construction, the ear doesn’t have a constant frequency response and thus, there will be frequencies that are accentuated and others that are attenuated, leading to the so-called hearing area as shown in Fig. 2.2. Note the peak around 2-4 kHz where the human ear is more sensible which is also where the human voice is centered.
2.1.2 The Source-filter Model of Speech-Production

Speech production modeling is not a trivial aspect. Achieving an accurate description of the real anatomy and physiology of the human speech system would be rather complicated. Through history, engineers have preferred to find a simple mathematical model that contains essential characteristics of speech signals. In analogy from the physical generative process of speech discussed above, it seems reasonable to design a two stage model as shown in Fig. 2.3.

The model, called the Source-Filter model, consists of two components:

- Excitation: it models the signal just before entering the vocal tract. Mainly, it takes into account the vocal cords and glottis effects.

- Vocal tract filter: it models the time-varying acoustic filter that comprise oral and nasal cavities by approximating it with a lossless-tube model.
Source Model

The source model features the influence of the lungs and vocal cords to the final speech signal. Depending on their production mechanisms, speech signals can be classified in three main categories: voiced sounds, unvoiced sounds and mixed sounds.

Voiced sounds are characterized by their fundamental frequency, i.e. the frequency at which the vocal cords vibrate. From the engineering point of view, this phenomena can be modeled as a train of pulses with periodicity equal to the fundamental frequency. In this thesis we will refer to this fundamental frequency as pitch (or pitch period). Unvoiced sounds are characterized by its spectral envelope. In analogy to its production mechanism, it can be modeled as a white gaussian sequence. Finally, mixed sounds\(^1\) are built up as a weighted sum of both models.

Vocal Tract

The vocal tract features the influence after the glottis delivers the airflow to the articulators that will contribute to the final speech signal. It will be modeled as a digital time-varying filter derived from the physics of sound propagation inside an acoustic tube.

All fine structures that shape the signal, can be approximated with a lossless cylindrical tube consisting of several cylindrical sections of equal length but different diameter. In [3] Atal show that by exploiting the relations of a lossless tube model with digital filters, it is enough with the formant frequencies and bandwidths to uniquely determine the tube model parameters. Besides, this tube model can be represented as a transfer function with \(P\) poles when the number of sections of the lossless tube is \(P\). In [3], Atal showed the consistency of the prediction filter with the speech production model, since the corresponding \(P\) poles carry all the information regarding the vocal tract model. One can identify each section of the tube model with a pole (see Fig.2.4)

\[
A_i(z) = 1 - p_i z^{-1},
\]

and the concatenation of \(P\) sections lead to a digital filter with \(P\) zeros (see Fig.2.5)

\[
A(z) = A_1(z) A_2(z) \ldots A_P(z) = \prod_{k=1}^{P} (1 - p_k z^{-1})
\]

\[\text{e}[n] \quad \frac{1}{A_1(z)} \quad \frac{1}{A_2(z)} \quad \ldots \quad \frac{1}{A_P(z)} \quad \text{x}[n]\]

Figure 2.4: Digital model of each section of the cylindrical tube

\(^1\)Examples of mixed sounds are plosive or fricative sounds
2.2 Linear Prediction of Speech

As we have seen above, the speech signal can be modeled as the output of a time-varying all-pole filter that is excited by a periodic pulse train for voiced speech and by dense noise in the case of unvoiced speech. In order to understand what this digital implementation of the source-filter model consists of, it is useful to tackle the problem from the spectral domain. The goal of the source-filter model is to decouple the spectral part that corresponds to the excitation (constant in the unvoiced speech and an harmonic structure for voiced speech) from the spectral part that corresponds to the vocal tract (spectral envelope). Thus, for unvoiced signals the power spectrum will be the same as the spectral envelope, whereas for voiced signals we will have an harmonic structure, with frequencies located at multiples of the pitch frequency, shaped by the vocal tract.

In the pioneering work done by Atal [1] the all-pole coefficients are determined by minimizing the L2-norm of the difference between the observed signal and the predicted signal, i.e. the mean square error of the residual signal. That set-up leads to the well-known Yule-Walker equations for the autoregressive (AR) modelling. This solution corresponds to the maximum-likelihood (ML) estimator for a residual signal distributed as an i.i.d Gaussian signal, which is optimal in the mean square error (MSE) sense. Hence, for the case of unvoiced signals that can be modeled as white noise passing through an all-pole filter, it naturally leads to the L2-norm minimization principle. On the other hand, for the case of voiced signals, the quality of the L2 cost approach is questionable since the excitation signal is not i.i.d Gaussian and thus, it lacks of foundations in a statistical sense.

As announced in several works ([13], [15]) LP fails to decouple the vocal tract shape from the excitation signal in the case of unvoiced speech. As seen in Fig.2.6 there is a huge dependency of the pitch period. The LP tries to cancel the input voiced harmonics placing poles close to the unit circle and overestimates the spectral amplitude at the formant frequencies, providing a sharper contour than the original one. In order to solve this problem, several approaches have been considered ranging from rethinking the spectral modelling problem [15], changing statistical assumptions or more recently, using an L1-cost approach to minimize the residual signal is considered, seeking a sparse solution that will fit with the modeling assumptions (excitation signal is modeled as a train of impulses).

Even though the limitations of L2-norm minimization for speech analysis are well known, it is still the most popular criterion for speech analysis and coding for one main reason: simplicity.
Figure 2.6: LP envelope comparison between L1 cost (blue) and L2 cost (red). In (a) we have excited the all-pole filter with a periodic sequence of pitch \( f_p = 320 \text{Hz} \). In (b) the pitch is \( f_p = 160 \text{Hz} \). The sampling frequency for both is \( f_s = 8 \text{kHz} \) and the order of the filter is \( P = 10 \).

Characteristics

- The L2-norm minimization results in the Yule-Walker equations that can be efficiently solved by Levinson recursion and it will always find the global minimum as L2-norm cost function is strongly convex. As shown in [34] the zeros of the analysis filter \( A(z) \) are guaranteed to be inside the unit circle (minimum-phase) and thus, the all-pole synthesis filter \( \frac{1}{A(z)} \) is stable.

- Another interesting property is the time-frequency analogy through the Parseval theorem. L2-norm criterion intrinsically implies minimizing the error between the true and the estimated spectra. [33]

- Linear prediction estimates the spectral envelope of the signal. Minimizing the L2 cost or the Itakura Saito distortion measure (2.2) are equivalent. From this analogy, one can conclude that LP better estimates the peaks than the valleys. To get to this conclusion, note that Itakura Saito \( D_{IS} \) distortion measure computes the area of a function that depends on the distance \( Q(f) \) between the original spectrum \( S(f) \) and the estimated spectrum \( \hat{S}(f) \). As seen in Fig[2.7] this function is not symmetric and penalizes more the underestimate values than the overestimated.

\[
D_{IS} = \int_{-\frac{1}{2}}^{\frac{1}{2}} \left[ e^{Q(f)} - Q(f) - q \right] df
\]

\[
Q(f) = \log \frac{S_x(f)}{S_{\hat{x}}(f)}
\]
In the source-filter model of speech production, the all-pole filter needs to be stable in order to synthesize speech. According to the nature of speech, all signals are bounded and thus, correspond to a stable process. For an application point of view, it might not be mandatory for applications such as speech recognition, but it is a necessary condition in others such as speech coding. As mentioned above, L2-norm intrinsically results in stable all-pole filter. The optimum prediction polynomial $A(z)$ has all zeros $p_k$, $k = 1, ..., P$ inside the unit circle, that is $\|p_k\| < 1$ for all $k = 1, ..., P$, as long as the signal $x(n)$ is not a line spectral process, i.e. not fully predictable. However, this proof cannot be generalized for any Lp-norm.

**Proof** This proof appeared in [34]. The prediction filter in Fig. 2.8 can be redrawn as in Fig. 2.8 where $q$ can be any zero from the original polynomial $A(z)$, and $C(z)$ is a causal FIR with $P - 1$ zeros. Note that $1 - qz^{-1}$ is the optimal first-order prediction for the signal $y(n)$ in a L2-norm criterion since, otherwise, the residual signal could be made smaller and that contradicts the fact that $A(z)$ is the optimum LP(P) predictor for $x(n)$. This argument is key in order to develop the stability proof. To proof stability of the prediction filter, it is necessary to show that $|q| < 1$.

According to linear prediction theory,

$$q = \frac{R_{yy}(1)}{R_{yy}(0)},$$

(2.1)

where $R_{yy}(m)$ is the autocorrelation function of the process $y(n)$. From basic signal processing, we know that $R_{yy}(0) \geq R_{yy}(k)$ for any $k$ as long as the
signal $y(n)$ is not a line spectral process. Hence,

$$|q| \leq 1.$$  

To show that $|q|$ is strictly smaller than one, it is necessary to perform some more steps. Here, recall that

- The residual noise $e(n)$ is orthogonal to the previous data samples $x(n - 1), x(n - 2), \ldots x(n - P)$ (orthogonality principle).
- The signal $y(n - 1)$ is a linear combination of $x(n - 1), x(n - 2), \ldots x(n - P)$

$$(y(n) = x(n) - \sum_{k=1}^{P-1} c_k x(n - k)).$$

Then,

$$E[e(n)y^*(n - 1)] = 0; \quad (2.2)$$

By considering that $e(n) = x(n) - \sum_{k=1}^{P} a_k x(n - k) = y(n) - qy(n - 1)$, the error variance can be written as follows:

$$\sigma_e^2 = E[|e(n)|^2] = E[e(n)(y(n) - qy(n - 1))^*]$$

$$= E[e(n)y^*(n)] \text{ from (2.2)}$$

$$= E[(y(n) - qy(n - 1))y^*(n)|^2]$$

$$= R_{yy}(0) - qR_{yy}(1)$$

$$= R_{yy}(0)(1 - |q|^2) \text{ from (2.1)}$$

Finally, whenever $x(n)$ is not a spectral line process, we know that $\sigma_e^2 > 0$. Thus $(1 - |q|^2) > 0$ which actually proves that $|q| < 1$

### 2.2.2 Time Varying Linear Prediction

Vocal tract often varies slowly rather than as a sequence of abrupt jumps. By this motivation, Oppenheim introduced in [18] the so-called Time Varying Linear Prediction (TVLP). TVLP allows the coefficients of the all-pole model to change over time. Thus, each sample is defined by a different set of autoregressive coefficients. Besides, it allows to analyze larger windows of speech signals without any stationarity assumption. The first approach proposed by Oppenheim [18] allows the coefficients to vary in a parametric manner, as a linear combination of a set of known basis. Assuming that the coefficients are slowly varying, we can take any kind of basis to represent it. Different kinds of basis such as Legendre [23], Fourier [18], discrete prolate spheroidal functions [17] or wavelets [32], can be found in literature. Afterwards, TVLP using L1-norm minimization, such as in [10], is used in order to get rid of the Gaussian assumption of the voice excitation signal and better deal with voiced speech. Several applications arised from TVLP, such as formant tracking or detection of glottal openings and closings [27].

The main problem of TVLP is computation resources. Even though it allows smoother trajectories than standard LP analysis and it needs much less parameters to represent the speech signal (interesting for speech coding purposes), it deals with a great amount of data and it is computationally inefficient nowadays. To the author’s wish, let’s hope that in the proper years, we can explode TVLP much more.
### 2.3 Sparsity in Speech Signal Processing

Sparse approximations have been very successful in several signal processing applications during recent years. Specifically, the key application domains of sparse signal processing are sampling, coding, array processing, component analysis or spectral estimation among others. The basic idea behind sparse approximation is that many natural signals are sparse in some domain (time, frequency or space), dictionary or basis, that is just a few of its components are relevant, the others are insignificant or zero. Then, retaining only those elements with almost all the information, high precision approximations can be found. It is interesting to note that speech coding is one of the first problems that tackle sparse solutions. In [2] it is shown that one can produce speech of any desired quality by filtering a sufficient number of pulses through the synthesis filter. Finding the position and the amplitude of the pulses results in solving an inverse problem with sparsity constraints.

Mathematically speaking and under sparsity assumptions, we wish to recover a signal $x \in \mathbb{R}^M$ from a set of redundant measurements $y \in \mathbb{R}^N$

$$\begin{align*}
\text{minimize} & \quad \|x\|_0 \\
\text{s.t.} & \quad y = Ax
\end{align*}$$

(2.3)

where $M > N$ and $A \in \mathbb{R}^{N \times M}$ represents the redundant basis determined by the physics of the problem. The cost function $\| \cdot \|_0$ is the L0-norm and represents the cardinality of $x$. This approach basically seeks for the simplest explanation of the data given the measurement matrix. Another common approach involving measurement errors is the one in (2.4)

$$\begin{align*}
\text{minimize} & \quad \|x\|_0 \\
\text{s.t.} & \quad \|y - Ax\|_2 \leq \epsilon
\end{align*}$$

(2.4)

Unfortunately, this has little practical use since the optimization problems (2.3) and (2.4) are nonconvex and generally not possible to solve, since it implies intractable combinatorial search. To overcome this problem, several works announced algorithms that approximate the solution of (2.3) and (2.4) and even match perfectly the solution of the L0-norm under certain circumstances. Some examples are greedy algorithms [31], convex relaxations [6] and bayesian inference [29].
2.3.1 Greedy algorithms

The first approach to solve optimization problems in (2.3) and (2.4) are based on iterative greedy search (IGS), solving heuristically the sparse approximation by iteratively making locally optimal decisions with the hope that it will find the global optimal solution. Here, the matching pursuit (MP) algorithm introduced by S. Mallat and Z. Zhang in [25] is the basis for most of the algorithms in the literature. It decomposes any signal into a linear expansions of waveforms or basis (columns $A_i$) that belongs to an overcomplete dictionary ($A$). These basis functions are selected to best match the signal structures (normally, matched filtering) and in the literature are referred as atoms. Each atom corresponds to an index of the support set (i.e. non-zero elements of the sparse vector $x$). The main deficiency of MP type algorithms is: whenever an atom is selected and it was a wrong choice, there is no possibility to either correct its amplitude or even get rid of it. To overcome these problems several approaches are studied [8]. One of the most simple and prominent solutions is the orthogonal matching pursuit (OMP). It is based on MP but at each iteration it performs a L2-norm minimization (least-squares solution) in order to find the amplitude of the selected atoms (i.e. $x_i$). Other strategies are subspace pursuit AND look ahead techniques.

2.3.2 Convex relaxation

Another approach to solve (2.3) and (2.4) is to use convex optimization techniques. The first idea was introduced in [9] with the development of the basis pursuit (BP) method. Differing from MP and OMP, BP considers that the cardinality of a vector (i.e., $\ell_0$-norm) can be approximated by the absolute sum of its coefficients ($\ell_1$-norm), by replacing an optimization problem, that requires combinatorial search, with a problem solvable with convex tools. It also differs from general greedy search algorithms on the fact that it is based on global optimization problems and thus, it might find improved sparse solutions. The work done by E.Candès, et. al. in [6] analyze the stability of this method and its convergence, as well as the conditions to perfectly match the $\ell_0$-norm solution. The $\ell_1$-norm is chosen for this purpose as it is the closest convex norm to $\ell_0$-norm. Then, the solutions of (2.3) and (2.4) are found solving:

$$\minimize_x \|x\|_1 \quad \text{s.t.} \quad y = Ax$$  \hspace{1cm} (2.5)

$$\minimize_x \|x\|_1 \quad \text{s.t.} \quad \|y - Ax\|_2 \leq \epsilon$$  \hspace{1cm} (2.6)

Furthermore, recent algorithms have exploited the sparsity inducing property of the $\ell_1$-norm to improve the solution of the problems in (2.5) and (2.6) by iteratively reweighting the minimization process.

$$\minimize_x \sum_{i=1}^M w_i |x_i| \quad \text{s.t.} \quad y = Ax$$  \hspace{1cm} (2.7)

We assume that the weights are chosen as the inverse of the magnitude of every coefficient as in (2.3.2).
\[ w_i = \begin{cases} \frac{1}{|x_{0,i}|}, & \text{if } x_{0,i} \neq 0 \\ \infty, & \text{if } x_{0,i} = 0 \end{cases} \]

Then, if the true signal \( x_0 \) is \( K \)-sparse (i.e. \( \|x_0\|_0 = K \)), problem (2.7) is guaranteed to find the correct solution. In practice, since the precise weights cannot be found, we would use large values to discourage non-zero entries in the recovered signal and small weights could be used to encourage non-zero entries.

### 2.3.3 Sparse Linear Prediction

The standard methods used in LP of speech signals involve the minimization of the L2-norm of the residual. This intrinsically assumes that the residual is driven by a Gaussian source, what has actually been proven to be a false assumption since it is much closer to a Laplacian distribution. Hence, one can argue that a better scheme for speech analysis is not the one that minimizes the L2-norm, but the one with the least absolute error criterion. In the field of speech coding, for example, even though L2-norm is still employed to minimize the variance of the residual for efficient, sparse techniques are used to encode the signal. In regular-pulse excitation (RPE) coders, sparsity is motivated by psychoacoustics reasons, while in algebraic code-excited linear prediction (ACELP), sparse codes are used motivated by the dimensionality reduction of the excitation vector space. Therefore, usage of sparsity promoting techniques seems much reasonable than the L2-norm criterion. Sparse linear prediction, thus, tries to overcome those inconsistencies and better suit data characteristics.

The main goal of sparse linear prediction algorithms is to find a sparse signal that can represent the excitation signal by only a few non-zeros values representing the pitch and use it for more efficient encoding and representation of the data. Other approaches try to jointly estimate the short-term and the long-term linear prediction filters by exploiting sparsity in the overall filter to eliminate all the redundancies of the speech signal. Apart from those two applications, it is interesting to observe that in standard time varying linear prediction techniques introduced before, such as the one in ACELP, represents the autoregressive parameters by a few basis functions that one can see as a restricted subspace and thus, a sparse solution. This concept will be exploited in our work in TVLP introduced in Ch. [4], where we will introduce new sparse techniques to deal with the time varying linear prediction problem.

### 2.4 Bayesian Inference

Bayesian learning, or Bayesian inference, is a method for statistical inference based on the Baye’s rule to update the probability for a hypothesis as evidence is acquired.

\[ P(H|E) = \frac{P(E|H)P(H)}{P(E)} \quad (2.8) \]

where \( H \) stands for the hypothesis and evidence \( E \) corresponds to new data acquired. \( P(H) \) is the prior probability before the evidence is observed and \( P(H|E) \) is called the posterior distribution, which will be the prior probability when new evidence (data) is acquired.
Bayesian learning has some advantages over other statistical methods, such as ML or MAP:

- All variables are treated as random variables in a hierarchical manner (of course, up to a certain level defined by the user). It represents all the uncertainties in the choice of the model, that can adapt to the data observed.
- After the training task, we have both the model parameters and its probability, which can give a level of reliability.
- When it comes to iteratively train a system, it can be easily retrained and adapts to the new data is coming.

Even though all these advantages, there are some downsides regarding Bayesian Learning, which can be approximately be solved by variational Bayesian methods, but it goes beyond this thesis purposes.

- Information theoretically infeasible: it turns out that specifying a prior probability is mathematically extremely difficult.
- Computationally hard: even if an accurate prior probability can be found, there will be some problems where approximations should be made, either because of high-dimensionality of the data or too many parameters in the hierarchical model.
- Initial conditions: we might choose some initial distributions corresponding to the model parameters and there is no formal way to do it rather that subjectively.

Due to brevity and concerning only the purpose of this thesis, it will only be discussed the topic of linear regression, leaving other interesting discussions aside, with further information in [5]. The linear model for regression is the following

$$y = Ax + e$$  \hspace{1cm} (2.9)

where $y \in \mathbb{R}^N$ are the observations (evidence), $A$, are known basis functions or dictionary entries, $x \in \mathbb{R}^M$ are the model parameters or weights and finally $e \in \mathbb{R}^N$ is any kind of additive noise.

The distribution of the observed data given any parameter vector $f_{Y|X}(y|x)$ is called the sampling distribution.

The prior distribution is the distribution of the weights before any data is observed, i.e. $f_{X|\theta}(x|\theta)$, where $\theta$ are the so-called hyper-parameters that define the probability density function of the weights (e.g. if $x$ follows a Gaussian distribution, $\theta$ would be its mean and variance). Those hyper-parameters can also be modeled as random variables $f_{\theta|\lambda}(\theta|\lambda)$ and continue going more steps down in the modeling. However, this would increase the number of parameters to be learned and thus, we would need more and more data. In this thesis we will stop here and consider the hyper-parameters $\lambda$ deterministic (e.g. in the previous case of Gaussian distributed weights, the hyper-parameters for the mean can follow, for example, another Gaussian distribution with fixed mean and variance, and the variance can follow a Gamma distribution with also fixed
parameters). Depending on the application, it might or might not be so easy to have some previous knowledge about the parameters and determine the prior distribution for them. Then, we can make the prior information more or less informative by fixing its parameters. The less informative prior is the Jeffreys prior one \[21\]. It is convenient to choose a proper prior distribution that is a conjugate distribution to the sampling distribution, that is, the posterior density will be distributed as the prior just adapting its hyper-parameters to the data and its calculation may be expressed in a closed form.

The posterior distribution is the distribution of the weights after taking into account the observed data

\[
f_{X|\theta}(x|\theta^{(+)} = \frac{f_Y|X(x|\theta)f_X|\theta(x|\theta)}{f_Y|\theta(y|\theta)} = c(y) f_Y|X(x|\theta)f_X|\theta(x|\theta) \tag{2.10}
\]

We can use the posterior distribution to predict future observations by calculating the predictive density function

\[
f_{Y|Y|Y}(y|y, \theta^{(+)} = \int f_{Y|X}(y|x)f_X|\theta(x|\theta^{(+)} \tag{2.11}
\]

### 2.4.1 Sparse Bayesian Learning

In this section, we will introduce a general Bayesian framework to obtain sparse solutions to regression tasks using linear models in the parameters. As mentioned at some point, Bayesian inference might have too many parameters to infer (estimate) and thus, lead to overfitting. Sparse Bayesian Learning approach will help to shrink the number of parameters and just keep the ones that are relevant. This gives the name to this approach as relevance vector machine (RVM).

Consider the same set-up,

\[
y = Ax + e \tag{2.12}
\]

where \(y \in \mathbb{R}^N\) are the measurements, \(A\) are known system matrix, \(x \in \mathbb{R}^M\) are the weights and \(e \in \mathbb{R}^N\) is the additive noise term. RVM assumes Gaussian prior distributions

\[
e \sim \mathcal{N}(0, \beta^{-1}I) \\
x \sim \mathcal{N}(0, \Gamma^{-1})
\]

where \(\Gamma = \text{diag}(\gamma_1, \gamma_2, \ldots, \gamma_M)\) is a diagonal matrix with the precision of each independent dimension. Precisions (inverse variances) will also be considered as random variables that follow a Gamma prior:

\[
p(\Gamma) \sim \prod_{k=1}^{M} \text{Gamma}(\gamma_k | a + 1, b) \\
p(\beta) \sim \text{Gamma}(\beta| c + 1, d)
\]
where $\Gamma(\alpha|a + 1, b) \propto \alpha^a e^{-b\alpha}$. To make the hyper-priors non-informative we might fix their parameters to small values, ideally set to zero $a, b, c, d = 0$.

Furthermore, we can easily show that the likelihood function follows a Gaussian distribution:

$$p(y|x, \beta) = (2\pi)^{N/2} \beta^{N/2} \exp \left\{ -\frac{1}{2} \|y - Ax\|^2 \right\}.$$  \hspace{1cm} (2.13)

In order to infer from observed data, we are interested in computing the posterior distribution over all the unknowns given the measurements:

$$p(x, \Gamma, \beta | y) = \frac{p(x | y, \Gamma, \beta)p(x, \Gamma, \beta)}{p(y)} = p(x | y, \Gamma, \beta)p(\Gamma, \beta | y).$$  \hspace{1cm} (2.14)

From Bayes’ rule we derive the first equality but rapidly notice that we can’t compute it in a full analytical form, since we can’t calculate the normalizing term $p(y)$. Instead, the second equality calculations are performed. Note that the first term is the posterior distribution over the weights and that can be easily computed, since it only involves Gaussian distributed parameters. The posterior distribution over the weights is, thus, given by:

$$(x | y, \Gamma, \beta) \sim \mathcal{N}(\mu, \Sigma)$$

where $\mu = \beta \Sigma A^\top y$, $\Sigma = (\beta A^\top A + \Gamma)^{-1}$.

To compute the second term $p(\Gamma, \beta | y)$ we must make some approximations, and do so by representing the hyper-parameter posterior distributions by a delta function at its mode, that is, its most probable values $\Gamma_{MP}, \beta_{MP}$. Thus, relevance vector leaning becomes the search for the hyper-parameter posterior mode, i.e. maximizing $p(\Gamma, \beta | y) \propto p(y | \Gamma)p(\Gamma)p(\beta)$.

Here, there are two different approaches to solve the maximization. The first one is called evidence approximation and basically consists of computing MAP estimation using marginal likelihood function $p(y | \Gamma, \beta) \sim \mathcal{N}(0, \beta^{-1}I + A\Gamma^{-1}A^\top)$ \cite{22}. The second one exploits the expectation-maximization algorithm by treating the weights $x$ as hidden variables. Then, the function to be maximized is

$$E_x \left[ \log (p(y | x, \beta)p(x | \Gamma)p(\Gamma)p(\beta)) \right].$$

For this latter case, the updates fo the hyperparameters are:

$$\gamma_i = \frac{1 + 2a}{\mu_i^2 + \Sigma_{ii} + 2b}$$

$$\beta = \frac{N + 2c}{\|y - Ax\|^2 + \text{tr}(\Sigma A^\top A) + 2d}$$

where $\mu_i$ is the i-th element of the weight’s posterior mean $\mu$ and $\Sigma_{ii}$ is the i-th diagonal element of the weight’s posterior covariance $\Sigma$.

Differently from the standard approach of seeking a sparsity promoting function (e.g. L1-norm and reweighted L1-norm), here we seek for sparsity promoting distributions. One can analyze the ‘true’ weight distribution to see the level...
of sparsity promoting of the RVM scheme. By marginalizing $p(\mathbf{x} | \Gamma)$ with respect to the hyper-paramenters, it is shown that each weight follows a Student-t distribution.

$$p(x_i) = \frac{b^a \Gamma\left(a + \frac{1}{2}\right)}{(2\pi)^{0.5} \Gamma(a)} (b + \frac{x_i^2}{2})^{-(a+0.5)}$$  \hspace{1cm} (2.15)

From the analytical expression we see that, if we use a non-informative prior for the hyper-parameters, (i.e. $a = b = 0$) we obtain a prior that intuitively looks like sparsity promoting, thanks to its similarity to the popular Laplace prior $p(x_i) \propto \exp(-|x_i|)$. Even though we can compute the distribution of the weights, we can’t sucessfully proceed with the fully Bayesian framework, since the marginal distribution $p(\mathbf{x})$ is no longer Gaussian. Alternatively, we can look for the standard MAP estimation using the known distribution for the weights. This leads to a maximization of the likelihood function but with the regularization term $\sum_{k=1}^{M} \log|x_i|$ that can be seen as an L1 regularizer and, thus, it has been proven that RVM induces sparsity on the weights. However, this method is useless since we typically find that the likelihood function to maximize and hence, the posterior of the weights, is multimodal. This comes from the fact that the likelihood function (Gaussian distributed) overlaps the spines (see Student-t distribution in [29]) of the prior distribution.

In this thesis we further explore the case that the sparsity promoting distributions are on the noise part (i.e. sparse noise). It will be shown in Ch. [4].
Chapter 3

Stability analysis

3.1 Introduction

Linear Prediction is a widely used technique in speech signal processing (e.g. coding and recognition). Traditionally, minimization of the L2-norm of the residual has been the most common approach, as explained in Ch. [2]. This cost function implicitly assumes that the excitation signal follows a Gaussian distribution. This is the case of unvoiced speech, which corresponds to less than \( \frac{1}{3} \) of the speech utterances. For voiced speech, an alternative approach based on L1-norm minimization is used. Several works have shown its better modeling property by decoupling the vocal tract shape estimation from the pitch period, that is modeled by the residual noise [15].

L1-norm minimization approach has also shown beneficial properties in speech coding. Several speech coders, such as multi-pulse excitation (MPE) or algebraic code excited linear prediction (ACELP), are based on a multi-pulse signal that excites the synthesis filter. As a convex relaxation of the L0-norm, L1-norm minimization retrieve sparser solutions providing few high energy samples that correspond to the pitch period. Encoding of these few values lead to a more efficient coding of the residual signal. However, unlike the predictors found using the L2-norm cost function (using the Levinson recursion), L1-norm minimization doesn’t intrinsically imply stable predictors. Other works in the literature use higher order statistics to estimate the linear prediction filter and can’t neither be assumed to retrieve stable filters [28].

In the following, a novel and general method to estimate stable filters for any Lp-norm is presented. The motivation is the higher performance of the L1-norm minimization and its potential use in the future speech coding algorithms. The method is linear and use a convex optimization framework. It is compared to other approaches, such as the standard L2-norm minimization followed by pole reflection technique, that projects the unstable poles inside the unit circle.

3.2 Problem formulation

Let \( x(n) \) be a signal modeled as a P-th order autoregressive (AR) process.
\[ x(n) = \sum_{k=1}^{P} a_k x(n-k) + e(n) \] (3.1)

where \( a_k \) are the predictor coefficients

\[ A(z) = 1 - \sum_{k=1}^{P} a_k z^{-k} \]

and \( e(n) \) is the excitation signal or driving noise process which, in general, might not be stationary and can follow any distribution. Now, let us consider a N-point sequence of the process \( x(n) \) as follows:

\[ x = Xa + e \]

where

\[
X = \begin{bmatrix}
  x_{N_1-1} & x_{N_1-2} & \cdots & x_{N_1-P} \\
  x_{N_1-2} & x_{N_1-3} & \cdots & x_{N_1-P+1} \\
  \vdots & \vdots & \ddots & \vdots \\
  x_{N_2-1} & x_{N_2-2} & \cdots & x_{N_2-P}
\end{bmatrix},
\]

\[ x_n = 0 \text{ for } n < 1 \text{ and } n > N \]

The general problem consists of finding the prediction coefficients \( a_k \in \mathbb{R}^P \) that better fits our model, generally written as follows:

\[ a^* = \arg \min_{a} \| x - Xa \|_p^p + \lambda \| a \|_k^k \] (3.2)

The choice of the starting \( N_1 \) and ending \( N_2 \) points, as well as the \( p, k \) brings up several solutions that fit different problems.

One of the most common choices is to use \( p = 2 \) and \( \lambda = 0 \). That is equivalent to the maximum likelihood estimator for a system excited by an i.i.d Gaussian signal, which is optimal in the mean square error (MSE) sense as \( N \to \infty \). The two most common choices for \( N_1 \) and \( N_2 \) are:

- \( N_1 = 1 \) and \( N_2 = N + P \). This choice is equivalent to the so-called autocorrelation method, which ensures to find a minimum-phase analysis filter. Note that, in addition to the \( x(n) \), we need consecutives samples, arbitrary set to 0.

- \( N_1 = P + 1 \) and \( N_2 = N \). We refer to this method as autocovariance method. It provides better estimates of the short term correlation function than the autocorrelation method, because we are not using arbitrary samples. However, it doesn’t ensure stable synthesis filters.
\[ \mathbf{a}^* = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{x} \quad (3.3) \]

However, it only gives an approximate solution for non-Gaussian excitation signals, such as the case of voiced speech sounds. In that case, the common choice is to use \( p = 1 \). This method is equivalent to the maximum-likelihood solution when the residual signal is assumed to be i.i.d Laplacian, which has recently been shown to model better the speech residual signal.

\[ \mathbf{a}^* = \arg\min_{\mathbf{a}} \| \mathbf{x} - \mathbf{Xa} \|_1 \]

This method doesn’t have a closed form expression and thus should be solved algorithmically. Furthermore, unlike the standard LPC method, L1-norm minimization doesn’t guarantee the stability of the corresponding all-pole filter. That will produce saturation whenever synthesizing speech is required. In most of the speech coders, where LSF coefficients are used to quantize the autoregressive parameters, stability of the LP filter is mandatory in order to be able to perform the transformation between AR parameters and LSF coefficients.

Stability is difficult to ensure as a general rule for any Lp-norm, since it follows a non-linear relationship of the AR coefficients. There are several techniques that can ensure stability, such as the ones proposed in [14]. We propose a general method that looks for best stable solution that solves (3.2) by using a convex optimization approach.

### 3.3 Minimum-phase LP through LSF coefficients

Line Spectral Frequencies (LSF) are used to represent the linear prediction coefficients. Firstly introduced by Itakura [20], LSF technique presents several properties that are very useful in applications such as speech coding:

- Robustness against quantization noise
- Stability easily ensured
- Better perceptual interpretation in the frequency domain.

To define LSF parameters, let us represent \( A(z) \) as the weighted sum of two different polynomials: \( A(z) = 0.5[P(z) + Q(z)] \), where

\[
P(z) = A(z) + z^{-(P+1)}A(z^{-1}) \]
\[
Q(z) = A(z) - z^{-(P+1)}A(z^{-1}).
\]

\( P(z) \) is called the palindromic polynomial and \( Q(z) \) the antipalindromic polynomial and have the following properties:

- The roots of \( P(z) \) and \( Q(z) \) lie on the unit circle
- The roots of \( P(z) \) are alternated with those of \( Q(z) \) around the unit circle.
- As for speech processing purposes, \( A(z) \) has real coefficients, the roots from \( P(z) \) and \( Q(z) \) occur in conjugate pairs.
Figure 3.1: Pole-zero plot of polynomials $A(z), P(z), Q(z)$. With red the zeros of $A(z)$. Zeros of $P(z)$ are represented with blue triangles. Zeros of $Q(z)$ are represented with blue circles.

Given the $P$ coefficients of the polynomial $A(z)$, the $P$ LSF parameters can be uniquely identified by finding the roots of $P(z)$ and $Q(z)$, i.e.

$$\{ w \mid P(z), Q(z) = 0 \mid z = e^{jw}, 0 < w < \pi \}$$

and they have some interesting features that are worth noting:

- Stability is easily preserved by making sure that the ordering of the LSF parameters is preserved after manipulation (e.g. quantization)

- They have better perceptual interpretation than standard LP coefficients. The spectral envelope of an AR process has several peaks (formants) that correspond to the phase of each pole of the polynomial $A(z)$. By construction, LSF coefficients are placed around these frequencies (see [33]) and thus, the crucial features from the power spectrum get coded into the LSF parameters. Basically, the frequency content of the LSF technique is the amplitude and bandwidth of each formant, depending on the phase of each LSF coefficient and its difference, respectively. This information allows us to perform bit allocation techniques that help to reduce the bit rate in speech coders (up to a 25% as reported in [33]) for a given perceptual quality.

- Close relationship with the acoustic tube model, as it can be seen as the open- and close-ended acoustic tube models as explained in [33]

As of interest to this work, we will exploit the ability of LSF technique to preserve stability of the all-pole model. Basically, it means that the zeros of the polynomial $A(z)$ lie strictly inside the unit circle and thus, it has minimum-phase properties.

Minimum phase condition is non-linear regarding LP coefficients $a_k$ and it requires algorithms to compute the roots of the filter $A(z)$. On the other hand, with the LSF parameters can be have easily ensured by preserving the increasing
Figure 3.2: Power spectrum of an AR process with the respective LSF coefficients represented. Dashed blue: Q(z) coefficients. Straight blue: P(z) coefficients

<table>
<thead>
<tr>
<th>Minimum-phase conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AR parameters</strong> a</td>
</tr>
<tr>
<td>$A(z) = 1 - \sum_{k=1}^{P} a_k z^{-k}$</td>
</tr>
<tr>
<td>$A(z) = \prod_{k=1}^{P} (1 - p_k z^{-k})$</td>
</tr>
<tr>
<td>$</td>
</tr>
</tbody>
</table>

Table 3.1: Stability conditions in different domains

order between the LSF parameters $f_k$. Furthermore, if we take into account the difference between the LSF parameters $\Delta f_k$, we see that the stability conditions form a convex set and thus, can be included in a convex optimization problem like the minimization of the Lp-norm as in speech analysis. Those relationships are specified in table (3.1).

The uniquely defined transformation between AR parameters $a_k$ and LSF coefficients $f_k$ is non linear and, additionally, doesn’t have a closed form expression. Thus, it is computed algorithmically using root finding algorithms. However, we can linearize it through a Taylor expansion around the operational point:

$$g(x + \Delta x) = g(x) + \nabla g(x) \Delta x + O(n^2)$$

(3.5)

where $g : \mathbb{R}^P \to \mathbb{R}^P$ is a bijective function uniquely relating two domains.

Let us define

$$f = g(a), \quad f = f_{op} + J_f(a_{op}) (a - a_{op})$$

(3.6)

where $a_{op}$ and $f_{op}$ are the operational points (OPs) and $J_f(a_{op})$ is the jacobian of the function evaluated at the operational point. The OP should be properly initialized since we linearize the function in a close boundary of this point.
In order to compute the jacobian of the function, we will use perturbation analysis. It basically approximates the derivative as the numerical variation in a close domain

$$\frac{\partial f_j}{\partial a_i} \simeq \frac{\Delta f_j}{\Delta a_i}$$

(3.7)

Then,

$$J_f(a_{\text{op}}) = \begin{bmatrix} \frac{\Delta f_1}{\Delta a_1} & \frac{\Delta f_1}{\Delta a_2} & \cdots & \frac{\Delta f_1}{\Delta a_P} \\ \frac{\Delta f_2}{\Delta a_1} & \frac{\Delta f_2}{\Delta a_2} & \cdots & \frac{\Delta f_2}{\Delta a_P} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\Delta f_P}{\Delta a_1} & \frac{\Delta f_P}{\Delta a_2} & \cdots & \frac{\Delta f_P}{\Delta a_P} \end{bmatrix},$$

(3.8)

Moreover, let us define the difference transform as follows:

$$\Delta f_i = f_{i+1} - f_i$$

(3.9)

$$\Delta f = Tf = \begin{bmatrix} \Delta f_1 \\ \Delta f_2 \\ \vdots \\ \Delta f_N \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 1 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{bmatrix}.$$  

(3.10)

Finally, our cost function in (3.2) will look as follows:

$$\|x - Xa\|^p_p = \|x - X[a_{\text{op}} + J_f(a)^{-1}(f - f_{\text{op}})]\|^p_p$$

$$= \|y - XJ_f(a)^{-1}f\|^p_p$$

$$= \|y - XJ_f(a)^{-1}T^{-1}\Delta f\|^p_p$$

(3.11)

and all possible solutions lie in the following convex set:

$$\Delta f_i > \epsilon$$

$$\sum_{i=1}^{P} \Delta f_i < \frac{1}{2} - \epsilon$$

(3.12)

where $\epsilon$ is a user tuned parameter that defines the minimum separation between LSF coefficients that can be used to avoid peaky spectrograms. We will refer to this parameter as the regularization term, and it is the minimum separation between LSF parameters. To get a better understanding, one can relate it with the frequency separation $\Delta f$ and the sampling frequency $f_s$:

$$\epsilon = \frac{2\pi \Delta f}{f_s}$$

(3.13)
Finally, the cost function in (3.2) together with the constraints in (3.12) can be solved via convex optimization techniques and will form the convex program in (11). This method allows us to find a stable all-pole model that describes the speech signal for a general Lp-norm cost function. Additionally, we can solve the problem of peaky spectrograms, which can cause a perceptual degradation of the reconstructed speech.

Algorithm 1: Lp-norm minimization through LSF coefficients

1. Set \( a_{\text{op}} = a_{\text{ini}} \), \( f_{\text{op}} = f_{\text{ini}} \), \( w = x - Xa_{\text{op}} \), \( C = \|x\|^p \epsilon \)
2. while \( C > \|w\|^p \epsilon \) do
3. \( \text{Compute } J_f(a_{\text{op}}), \Gamma \)
4. \( \text{Set } y = x - Xa_{\text{op}} + XJ_f^{-1}(a_{\text{op}})f_{\text{op}} \)
5. \( C = \|w\|^p \epsilon \)
6. minimize \( \|y - XJ_f(a_{\text{op}})^{-1}T^{-1}\Delta f\|^p \)
7. s.t. \( \Delta f_i > \epsilon \)
8. \( \sum_{i=1}^P \Delta f_i < \frac{1}{2} - \epsilon \)
9. \( w = x - Xa_{\text{hat}} \)
10. \( a_{\text{hat}} = a_{\text{op}} \)
11. end

However, we need an initial value to linearize this problem and that plays a key role: a good initialization will ensure that we find the global optimal point as well as being time-saving. The most time consuming part is the Jacobian computation. As the initial operational point, (see that it also requires to be stable) we will use the standard LP analysis.

3.4 Performance analysis

In this section we analyze the performance of the proposed method with the standard pole-inversion technique for different L1-norm and L2-norm measures. An overview of the methods compared and the acronyms used through the section are shown in table (3.2). Besides our method, stabilization is also performed using pole reflection. It needs of constant stability check and it consists of a root-finding algorithm that looks for the unstable poles and projects them in the unit circle by inverting and conjugating the pole. Pole inversion method keep the same amplitude response of the residual signal for both unstable and stabilized predictors.

3.4.1 Spectral Envelope Modeling

In this section we will analyze the modeling performance of the predictors in the case of voiced speech, unvoiced speech and a mixture of them. Different frame lengths \( N = 40, 80, 160, 320 \) are considered to perform the analysis. In order to know the reference generative process, we will generate synthetic speech in an analysis-by-synthesis (AbS) way. We will first analyze, using the LPC method, several speech utterances from different sentences in a random pattern from
Table 3.2: Different analysis methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LPC</td>
<td>Linear Predictive Coding (LPC) Traditional autocorrelation method, which minimizes (3.2). Used parameters are: $N_1 = 1$, $N_2 = N + P$ and $p = 2$.</td>
</tr>
<tr>
<td>LPI</td>
<td>Linear prediction Pole Inverse (LPI) Traditional covariance method, by minimizing (3.2) and stabilizing the synthesis filter through pole inversion. Used parameters are: $N_1 = P + 1$, $N_2 = N$ and $p = 2$.</td>
</tr>
<tr>
<td>LSF2</td>
<td>L2-norm minimization of (3.11) through finding the LSF coefficients. Used parameters are: $N_1 = P + 1$, $N_2 = N$ and $p = 2$.</td>
</tr>
<tr>
<td>L1</td>
<td>L1-norm minimization of (3.2) through finding directly the AR parameters and stabilizing the synthesis filter through pole inversion. Used parameters are: $N_1 = P + 1$, $N_2 = N$ and $p = 2$.</td>
</tr>
<tr>
<td>LSF1</td>
<td>L1-norm minimization of (3.11) through finding the LSF coefficients. Used parameters are: $N_1 = P + 1$, $N_2 = N$ and $p = 1$.</td>
</tr>
</tbody>
</table>

The criterion used to evaluate the quality of the spectral envelope modeling of the predictors, is the spectral distortion (SD) measure between the estimated all-pole model spectrum ($S(w; f) = S(w; a)$) and the ground truth found in the AbS stage $S(w)$. The SD is defined as

$$SD = \sqrt{\frac{1}{2\pi} \int_{-\pi}^{\pi} [10 \log_{10} S(w) - 10 \log_{10} \hat{S}(w; f)]^2 dw}$$  \hspace{1cm} (3.14)$$

Since both true and estimated signals are modeled as AR processes, $S(w)$ can be easily computed as follows:

$$S(w) = \frac{\sigma^2}{|A(e^{jw})|^2}$$  \hspace{1cm} (3.15)$$

The first conclusion we can infer from the results in Fig. 3.3 is that L1-norm methods outperform the L2-norm methods when the excitation is sparse or a mixture of sparse and gaussian sequences. As shown in several works, the L1-norm minimization is a better fit for sparse sequences and in the case of linear prediction that leads to a pitch independency analysis and better modeling of the excitation signal. In the case of gaussian noise, the L2-norm cost function is
optimal in terms of MMSE and has been shown to work better than any other norm.

In Fig. 3.3 we show the analysis case, where our method iterates at least once even though the initial estimate was good. This helps to ensure that it doesn’t overestimate the peaks no matter if the first estimation is better in terms of the mean squared error (MSE). Results show that standard methods followed by the pole reflection have indeed the best performance in terms of spectral distortion. This result might not be surprising to us. Firstly, the pole projection technique preserve the spectral envelope estimated with the possibly unstable poles which is, actually, rather good. It only effects the coefficients of the polynomial $A(z)$ and thus the residual signal. Secondly, we can see that for small frame lengths, $N = 40$ our method outperforms the pole reflection technique. This can have an explanation if we looks carefully in the unstability rate from the standard Lp-norm costs. Fig. 3.4 shows the unstability rate of methods LPI and L1 with respect to the frame length. High unstability rates are also linked with higher absolute values of the unstable poles and thus, it effects much more the final performance. For greater frame lengths $N \geq 160$ we see that the low unstability rate together with the fact that the unstable poles are so close to the unit circle have negligible effects to the final result. Finally, the generation procedure of synthetic signals, may effect the final result. The fact that standard LPC method allows to estimate spectra with high peaks which doesn’t correspond to the reality at all and they are mostly related with the possible unstable filters estimated. Thus, our method is not able to estimate the high power peaks that
can be generated using LPC estimated filters leading to a degraded performance (but certainly more accurate when using real speech signals).

To avoid this, we will do a further experiment using syntethic data from an analysis-by-synthesis procedure using the filters estimated in the previous analysis using the LSF method, and results are shown in Fig.3.5.

We finally conclude that standard methods perform well when they estimate stable filters. Otherwise, a better modeling is possible. The fact that the unstability rate is quite linked to the frame length and that our method is a bit more complex in terms of computations (due to the Jacobian calculation) makes our method attractive for low frame lengths where standard methods fail to ensure stability.

3.5 Peak-avoiding property

Conventional LP estimation is suited to estimate the spectral envelope since it emphasize the local spectral peaks. However, LP has its limitations. For medium pitch and high pitch voiced frames LP estimation does not provide good models. LP spectrum tends to overestimate and overemphasize spectral powers at formants, providing a sharper contour than the original vocal tract response. The use of L1-norm doesn’t tackle this problem although it succeeds to decouple the spectral envelope from the pitch period [15]. Several works on this topics have been done over the past years, being the most succesful approaches the lag window method [30] and the bandwidth expansion method [35], which are simple methods that prevent sharp spectral peaks.
The main reason why standard LP overestimates spectral peaks is that, in voiced frames, it tries to cancel out the high-power periodic pulses from the residual signal by placing poles close to the unit circle. In our method, one can tackle this problem using the regularization term $\epsilon$ from the convex set of possible predictors (3.12). The bandwidth separation between LSF parameters define the absolute value of the poles and thus, the bigger the separation between them is, the lower absolute value the poles have, which translates to a less estimated amplitude of the formant frequencies and hence, a smoother spectral envelope. Here, $\epsilon$ corresponds to the minimum bandwidth separation allowed between the LSF parameters.

However, the regularization parameter has an upper-bound depending on the prediction order and the sampling frequency, as shown in Fig. 3.6. Easily, one can see that the maximum separation that the LSF parameters can have, is the separation that will have it they were evenly distributed around the unit circle. It is interesting to see that this latter case corresponds to a white noise sequence.

How to choose the parameter $\epsilon$ such that it better fits some perceptual measure, is a question that still remains under consideration. One option might use different $\epsilon_i$ for each parameter in an adaptive way based in some perceptual measure since each formant frequency has different weight in the overall spectral envelope.

In order to see the difference, in Fig. 3.7 we show the power spectrum of a voiced frame. Clearly, it is seen that the first formant is overestimated using
3.5.1 Future Work

- Related to spectral envelope estimation, it seems reasonable to think about a method that adaptively selects the order depending on the data. Speech analysis through LSF coefficients might be a good approach to it. The method should automatically select those poles that are important, gather (quantize) the poles that lead to a close LSF coefficients into single parameter that characterizes the envelope rather than the spectral harmonics.

- Adapt the whole technique in a Bayesian Learning framework. Clearly, the Difference-LSF coefficients follow a Dirichlet distribution, and sparse bayesian learning (SBL) techniques can be used to model voiced speech. This might provide a better performance since the parameters are learned...
from data rather than being set by the user like in this approach. Variational Bayes techniques may be used since it is difficult to work with closed form solutions when system parameters are Diriclet distributed. This approach is under study although it goes behind the purpose of this thesis work.

• It can be useful to define higher order filters such as long-term prediction (LTP) or time-varying linear prediction (TVLP) where stability can be ensured, but it is not clear how to proceed.
Chapter 4

Time Varying Linear Prediction

Abstract We develop Bayesian learning algorithms for estimation of time-varying linear prediction (TVLP) coefficients of speech. Estimation of TVLP coefficients is a naturally underdetermined problem. We consider sparsity and subspace based approaches for dealing with the corresponding underdetermined system. Bayesian learning algorithms are developed to achieve better estimation performance. Expectation-maximization (EM) framework is employed to develop the Bayesian learning algorithms where we use a combined prior to model a driving noise (glottal signal) that has both sparse and dense statistical properties. The efficiency of the Bayesian learning algorithms is shown for synthetic signals using spectral distortion measure and formant tracking of real speech signals.

4.1 Introduction

Time-varying linear prediction (TVLP) model \cite{13, 27} of speech signals is a generalization over the much used linear prediction model \cite{24, 26}. In TVLP, each speech signal sample is predicted by a time-varying linear combination of past samples. Here, linear combination coefficients are time-varying unlike the case of standard linear prediction (LP) where the coefficients are fixed. Hence, for each speech signal frame there are more TVLP coefficients to estimate and the associated estimation problem becomes underdetermined. To keep the problem determined (or manageable), a smoothness constraint is typically imposed on time-varying coefficients. For example, time-varying coefficients have dynamics that can be modeled by a linear combination of low-frequency cosine functions, that means by using a fixed subspace in a known cosine basis. Several works considered different basis functions, such as Legendre \cite{23}, Fourier \cite{13}, discrete prolate spheroidal functions \cite{17}, wavelets \cite{32}. For a fixed subspace, a least-squares based method is predominant to estimate the TVLP coefficients. Independently of the basis functions used, the method can be viewed as a subspace estimation method.

In TVLP, a major consideration is modeling the driving noise. For speech signals, driving noise corresponds to a glottal signal. Typically a glottal signal is...
either assumed to be white noise for unvoiced sounds or periodic pulses at pitch frequency for voiced sounds. Unlike earlier works, we model the glottal signal as an additive combination of sparse (pulses) and dense (white noise) noise. Modeling of the glottal signal by purely sparse noise was recently considered in [11]. The authors in [11] considered an ideal data fit cost, that is, $\ell_0$-norm based cost minimization. The practical strategy was to use an iteratively re-weighted least-squares based algorithm (a deterministic solution). In this context, we mention that, for a standard linear prediction scheme, the use of sparse noise is considered in [15]. The work of [15] mainly used convex optimization to minimize a $\ell_1$-norm based data fit cost. Further, a Bayesian learning algorithm for a standard linear prediction was recently considered in [16] where a glottal signal was modeled as a combination of dense noise and block-structured sparse noise.

In this paper, we begin with the under-determined problem setup of TVLP and use Bayesian learning for estimation of the TVLP coefficients. We consider a data driven approach (no statistical stationarity is assumed) where the driving noise is modeled by a combination of sparse and dense noise. Bayesian learning methods are derived using expectation-maximization (EM) framework. Through simulations, we have found that the use of sparsity does not lead to a good performance for the underdetermined TVLP problem. Therefore, we convert the problem to a determined setup using a fixed subspace. For the determined setup, we show that the Bayesian learning method provides a better estimate of TVLP coefficients vis-a-vis competing methods.

4.2 Time-varying linear prediction system

In TVLP, the $n$’th speech sample $x_n$ is modeled as

$$x_n = \sum_{p=1}^{P} a_n(p) x_{n-p} + q_n \tag{4.1}$$

where $P$ is the order of the predictor and $\{a_n(p)\}_{p=1}^{P}$ are the TVLP coefficients at sample $n$. The term $q_n$ is the driving noise of the generative process model (4.1) and it is assumed to model the glottal signal. In our case, we assume that $q_n$ has two additive parts: the sparse noise $e_n$ to purely voiced sounds and the dense noise $w_n$ for purely unvoiced sounds. Let us consider that an $N$-point sequence of the signal $x_n$ is represented by a vector. Following (4.1) with $q_n = e_n + w_n$, we write

$$x = Xa + e + w \in \mathbb{R}^{N \times 1}, \tag{4.2}$$

where $x = [x_1, x_2, \ldots, x_N]$, $a = [a_1^T, a_2^T, \ldots, a_N^T]^T \in \mathbb{R}^{P \times 1}$ is the vector of TVLP coefficients, $e$ is the sparse noise vector and $w$ is the dense noise vector, and finally $X \in \mathbb{R}^{N \times PN}$ is the data matrix as below

$$X = \begin{bmatrix}
  x_1^T & 0 & 0 & \ldots & 0 \\
  0 & x_2^T & 0 & \ldots & 0 \\
  0 & 0 & x_3^T & \ldots & 0 \\
  \vdots & \vdots & \vdots & \ddots & \vdots \\
  0 & 0 & 0 & \ldots & x_N^T
\end{bmatrix}. \tag{4.3}$$
Here $\mathbf{x}_n = [x_{n-1}, x_{n-2}, \ldots, x_{n-P}]^\top$, and $\mathbf{a}_n = [a_n(1), a_n(2), \ldots, a_n(P)]^\top$. Let $\mathbf{A} \in \mathbb{R}^{P \times N}$ be formed by the column vectors $\mathbf{a}_n$ as $\mathbf{A} = [\mathbf{a}_1 \mathbf{a}_2 \ldots \mathbf{a}_N]$, and hence $\text{vec}(\mathbf{A}) = \hat{\mathbf{a}}$. Denoting the $k$'th row of $\mathbf{A}$ by $\mathbf{A}(k,:)$, we assume that the components of $\mathbf{A}(k,:)$ are slowly varying over $n$ and hence highly correlated. The correlation can be exploited by a decorrelating orthonormal transform $\mathbf{T}_1^{-1} \in \mathbb{R}^{N \times N}$ in the transform domain $\mathbf{T}_1^{-1}\mathbf{A}^\top$. Further the correlation between the components of rows of $\mathbf{T}_1^{-1}\mathbf{A}^\top$ can be exploited by another orthonormal transform $\mathbf{T}_2^{-1} \in \mathbb{R}^{P \times P}$, and hence the full transform can be realized as $\mathbf{T}_2^{-1}[\mathbf{T}_1^{-1}\mathbf{A}^\top]^\top$. Using the relation $\text{vec}(\mathbf{T}_2[\mathbf{T}_1^{-1}\mathbf{A}^\top]^\top) = \text{vec}(\mathbf{T}_2^{-1}\mathbf{A}[\mathbf{T}_1^{-1}]^\top) = [\mathbf{T}_1^{-1} \otimes \mathbf{T}_2] \text{vec}(\mathbf{A}) = [\mathbf{T}_1 \otimes \mathbf{T}_2]^{-1}\hat{\mathbf{a}}$, we can write (4.4) as

$$x = \hat{\mathbf{X}}\mathbf{T}\mathbf{d} + \mathbf{e} + \mathbf{w},$$

where $\mathbf{T} \triangleq [\mathbf{T}_1 \otimes \mathbf{T}_2]$ and $\otimes$ denotes Kronecker product. In this paper, we use the standard discrete cosine transform (DCT) II for $\mathbf{T}_1^{-1}$ and identity matrix for $\mathbf{T}_2$. Hence $\mathbf{T}$ is known. Note that $\mathbf{x}$ is $N$-dimensional and $\mathbf{d}$ is $PN$ dimensional. Therefore, (4.4) is underdetermined by a factor of $P$. Denoting an estimate of $\mathbf{d}$ by $\hat{\mathbf{d}}$, we find the estimate of $\hat{\mathbf{a}}$ by $\hat{\mathbf{T}}\hat{\mathbf{d}}$. To estimate $\mathbf{d}$ we can use two approaches: (1) sparsity assumption on $\mathbf{d}$, and (2) restrict to the first $L$ coefficients of $\mathbf{d}$. The first approach is motivated by standard sparse representations and compressed sensing. The second approach deals with a determined system (subspace based estimation). Denoting the first $L$ columns of $\mathbf{T}$ by $\mathbf{T}_{(L)}$, and correspondingly first $L$ coefficients of $\mathbf{d}$ by $\mathbf{d}_{(L)}$, the determined setup is

$$x = \hat{\mathbf{X}}\mathbf{T}_{(L)}\mathbf{d}_{(L)} + \mathbf{e} + \mathbf{w} + \mathbf{n},$$

where $\mathbf{n}$ is the noise due to truncation ($\mathbf{d}$ to $\mathbf{d}_{(L)}$). We use $P \leq L \leq PN$. For TVLP, we must need $L > P$. Note that, as $\mathbf{T}_1$ and $\mathbf{T}_2$ are orthonormal matrices, the case $L = P$ corresponds to a standard linear prediction, that means $\forall n$, $a_n(p) = a(p)$.

### 4.3 Bayesian learning

Using EM framework, we consider Bayesian learning for the two approaches in the following subsections.

#### 4.3.1 Underdetermined setup

Here we deal with (4.4) where both $\mathbf{d}$ and $\mathbf{e}$ are assumed to be sparse, and $\mathbf{w}$ is dense. To reduce the number of parameters to be estimated as well as exploiting the structure of the transform matrix $\mathbf{T}$, we use block sparsity in $\mathbf{d}$. Let $\mathbf{d}$ comprises of $K$-dimensional sub-vectors $\mathbf{d}_i$ such that $\frac{PN}{K}$ is an integer. For Bayesian learning, we use the following Gaussian prior to promote block sparsity in $\mathbf{d}$

$$\mathbf{d} \sim \prod_{i=1}^{\frac{PN}{K}} \mathcal{N}(0, \gamma^{-1}_i \mathbf{I}) = \mathcal{N}(0, \mathbf{\Gamma}^{-1}),$$

$$\mathbf{\Gamma} = \text{diag}((\mathbf{\gamma}^\top \otimes \mathbf{1}_K)^\top),$$

(4.6)
where $\gamma = [\gamma_1, \gamma_2, \ldots, \gamma_{PN}]^T$ and $1_K$ denotes a constant vector of ones of size $K \times 1$. For $K = 1$, the prior induces sparsity in a usual sense (fully unstructured), and its use can be found in several earlier works [29, ?] including our work [?]. Then, motivated by our recent result in [?], we use a combined model prior for the joint noise as

$$e + w \sim \prod_{i=1}^{N} \mathcal{N}(0, \beta_i^{-1}) = \mathcal{N}(0, B^{-1}), \quad B = \text{diag}(\beta),$$

(4.7)

where $\beta = [\beta_1, \beta_2, \ldots, \beta_N]^T$. The precisions are $\{\gamma_i\}$ and $\{\beta_i\}$ that have Gamma distribution as hyper-priors

$$p(\gamma_i) = \text{Gamma}(\gamma_i | a+1, b), \quad p(\beta_i) = \text{Gamma}(\beta_i | c+1, d),$$

(4.8)

where Gamma$(\gamma_i | a+1, b) \propto \gamma_i^a \exp(-b \gamma_i)$. The hyper-parameters are $\{a, b, c, d\}$.

We find the maximum-a-posteriori (MAP) estimate of $\beta$ by maximization of $p(\beta|x, \gamma, \beta)$, as follows

$$\hat{\beta} = \Sigma (\bar{X}^T B \bar{X})^{-1}.$$

(4.9)

The precisions are updated by using the EM algorithm. Let $\theta \equiv \{\gamma, \beta\}$ denote the parameters that are updated in each iteration by maximizing the cost (EM help function in MAP estimation)

$$Q(\theta, \theta') + \ln p(\theta),$$

(4.10)

where $\theta'$ are the parameter values from the previous iteration. The function $Q(\theta, \theta')$ is defined as

$$Q(\theta, \theta') = E_{d|x, \theta'}[\ln p(x, d|\theta)],$$

(4.11)

where $E$ denotes the expectation operator. The maximization of (4.10) leads to following update equations over iterations

$$\gamma_{\text{new}} = \frac{K+2a}{\sum_{i=1}^{N} \gamma_i^{-1} + 2b' \bar{X}^T \bar{X}},$$

$$\beta_{\text{new}} = \frac{1+2c}{(x-X\hat{d})^T + (X^T \Sigma'(X^T))^2 + 2d},$$

where $\Sigma' = (\Gamma' + (X^T \Sigma B X)^{-1})^{-1}$. The derivation of the update equations is shown in section 4.6. Here we seek sparsity promoting solutions, and hence the requirement is that most of the precisions of the prior distribution will turn out to be high. In order to satisfy the requirement, we use non-informative hyper-priors (flat distribution) for the precisions $\gamma, \beta$ by fixing their parameters to small values: e.g. $a = b = c = d \sim 10^{-3}$.

### 4.3.2 Determined setup

Here, we deal with (4.5) by using the following isotropic Gaussian prior

$$d_{(L)} \sim \prod_{i=1}^{L} \mathcal{N}(0, \gamma^{-1}) = \mathcal{N}(0, \gamma^{-1} I).$$

(4.12)
The prior for $e + w + n$ is the same prior as for $e + w$ in section 4.3.1 as shown in (4.7). Using similar arguments as in section 4.3.1 the MAP estimate of $d(L)$ is

$$
\hat{d}(L) = \Sigma(L)(XT(L))^T B x,
$$

and the update equations for the precisions are

$$
\gamma_{\text{new}} = \frac{L + 2a}{\text{trace}(\Sigma(L) + \hat{d}(L)^T \hat{d}(L)) + 2b},
$$

$$
\beta_{i,\text{new}} = \frac{1 + 2c}{(x - XT(L) \hat{d}(L))^2_i + (XT(L) \Sigma(L)(XT(L))^T)_{ii} + 2d}.
$$

The derivation of the update equations is similar as before and hence not shown. In this case, we use a non-informative prior for $d(L)$ as we do not have a-priori knowledge about its properties. We achieve this behavior by fixing the hyper-priors for $\gamma$ as: $a \sim 10^{-3}$, $b \sim 10^3$. For the noise term $e + w + n$ we still use a sparsity promoting solution, and hence we set their hyper-priors to $c = d \sim 10^{-3}$.

### 4.4 Experiments

We evaluate the methods using synthetic signals as well as real speech where $P = 10$ is used and the sampling rate is 8 kHz. We considered window lengths of 20 ms, 40 ms and larger window length of 250 ms. The estimation methods we used are of two types: least-squares and Bayesian learning. In least-squares estimation, we compared following methods: [a] LP (stationary) - LP using statistical stationarity by solving Yule-Walker equations, [b] LP (least-squares) - solving (4.5) for $L = P$ by using a standard least squares $\hat{d}(L) = (XT(L))^T x$ (pseudo-inverse), [c] TVLP (least-squares) - solving (4.5) by pseudo-inverse, for $L = 30$ when window length is 20 ms, $L = 40$ when window length is 40 ms, $L = 60$ when window length is 250 ms. For Bayesian learning, we compared: [a] LP (Bayesian) - solving (4.5) for $L = P$ by using the relations in section 4.3.2, [b] TVLP (Bayesian) - solving (4.5) by using the relations in section 4.3.2, for $L = 30$ when window length is 20 ms, $L = 40$ when window length is 40 ms, $L = 60$ when window length is 250 ms, and [c] TVLP (U, Bayesian) - solving underdetermined setup (4.4) by using the relations in section 4.3.1 where the dimension of each block is $K = P = 10$. We also considered TVLP (U, Bayesian) for $K = 1$, that means using unstructured sparsity on $d$, which provided highly degraded performance and we do not report the degraded results.

#### 4.4.1 Synthetic signals

**Signal generation**

Synthetic signals are generated by the model (4.1) where we used different types of driving noise $q_n$:

i. Sparse noise: a pulse train with frequency 200 Hz to model a normal pitch frequency. The signal-to-driving-noise ratio (SDNR) is $\sigma_x^2 / \sigma_q^2 = 14.5$ dB.

ii. Dense noise: iid Gaussian noise signal such that SDNR = 13.1 dB.
iii. Joint noise: additive sparse noise and dense noise with equal variance such that SDNR = 12.5 dB.

To generate a stable signal \( x_n \), we used minimum phase TVLP coefficients, this means the Z-domain analysis filter \( 1 - \sum_{k=1}^{P} a_n(k) Z^{-k} \) has roots inside the unit circle for all \( n \). Further, to have a speech like signal, the TVLP coefficients are drawn from real speech signal analysis. Using a window length of 160 samples and window shift of one sample for a real 8 kHz speech signal, we performed autocorrelation based \( P = 10 \) order LP analysis that is guaranteed to provide stable filter coefficients. As the coefficients are drawn for every sample of the real speech signal, they can be used as known TVLP coefficients in the generative model (4.1).

Performance

For every \( n \)'th sample we compute the spectral distortion (SD) and report the average SD as the performance measure. The SD for the \( n \)'th sample is defined as

\[
SD = \left[ \frac{1}{2\pi} \int_{-\pi}^{\pi} \left[ 10 \log_{10} \frac{S_n(\omega)}{\hat{S}_n(\omega)} \right]^2 d\omega \right]^{\frac{1}{2}}, \tag{4.13}
\]

where \( S_n(\omega) = 1/|1 - \sum_{p=1}^{P} a_n(p)e^{-j\omega p}|^2 \) is the power spectrum and \( \hat{S}_n(\omega) = 1/|1 - \sum_{p=1}^{P} \hat{a}_n(p)e^{-j\omega p}|^2 \) is the reconstructed power spectrum. The average SD is computed for signal samples across 100 windowed signal frames.

Table 4.1 shows the performance of all competing methods. The ‘ground truth’ refers to the determined setup (4.5) where a baseline average SD is the minimum to arise due to the truncation of \( d \) to \( d_{(L)} \). For sparse driving noise, TVLP (Bayesian) provides the best performance. Further, for dense and joint driving noise types, the TVLP (Bayesian) provides good performance. The TVLP (U, Bayesian) turns out to be degraded. A possible reason for degraded performance in the underdetermined setup (4.4) can be the poor condition of system matrix \( \bar{X} \), as reflected in (4.3). The system matrix \( X \) is non-ideal as it is far away from the usual dense wide matrices used in standard sparse representations and compressed sensing. We have independently verified that the TVLP (U,Bayesian) learning algorithm provides good performance in experiments with ideal dense system matrices (with iid Gaussian entries) for the considered system sizes. For brevity, we do not report these experiments in this paper.

4.4.2 Real speech signals

We used clean speech signals from the Noizeus database. Considering the task of formant tracking, Fig. 4.1 shows spectrograms of different methods for a 250 ms speech signal instance. Fig. 4.1 (a) comprises of a series of periodograms where each periodogram is computed for a 10 ms window length to have a better time resolution and a 5 ms shift to have a better tracking of formants. It is clear that the standard LP method comes with a high level of granularity, as we observe in Fig. 4.1 (b). Finally, by visually comparing Fig. 4.1 (c) and (d), TVLP using Bayesian learning can be found to be the best method in the sense of formant tracking. The formant tracking is more prominent than the competing method of TVLP using least squares. We rely on visual inspection as there
Table 4.1: Comparison of methods for synthetic signals using average spectral distortion (average SD)

<table>
<thead>
<tr>
<th>Method</th>
<th>Noise types</th>
<th>0.85</th>
<th>0.85</th>
<th>0.85</th>
</tr>
</thead>
<tbody>
<tr>
<td>Window length = 20 ms</td>
<td>Ground truth (subspace)</td>
<td>0.85</td>
<td>0.85</td>
<td>0.85</td>
</tr>
<tr>
<td></td>
<td>LP (stationary)</td>
<td>2.26</td>
<td>2.88</td>
<td>2.78</td>
</tr>
<tr>
<td></td>
<td>LP (least-squares)</td>
<td>2.12</td>
<td>2.59</td>
<td>2.49</td>
</tr>
<tr>
<td></td>
<td>TVLP (least-squares)</td>
<td>1.53</td>
<td>2.86</td>
<td>2.64</td>
</tr>
<tr>
<td></td>
<td>LP (Bayesian)</td>
<td>1.93</td>
<td>2.59</td>
<td>2.33</td>
</tr>
<tr>
<td></td>
<td>TVLP (Bayesian)</td>
<td>1.07</td>
<td>2.95</td>
<td>2.38</td>
</tr>
<tr>
<td></td>
<td>TVLP (U, Bayesian)</td>
<td>2.65</td>
<td>4.57</td>
<td>3.93</td>
</tr>
<tr>
<td>Window length = 40 ms</td>
<td>Ground truth (subspace)</td>
<td>1.21</td>
<td>1.21</td>
<td>1.21</td>
</tr>
<tr>
<td></td>
<td>LP (stationary)</td>
<td>2.81</td>
<td>3.06</td>
<td>2.96</td>
</tr>
<tr>
<td></td>
<td>LP (least-squares)</td>
<td>2.80</td>
<td>2.87</td>
<td>2.82</td>
</tr>
<tr>
<td></td>
<td>TVLP (least-squares)</td>
<td>1.68</td>
<td>2.55</td>
<td>2.39</td>
</tr>
<tr>
<td></td>
<td>LP (Bayesian)</td>
<td>2.56</td>
<td>2.87</td>
<td>2.73</td>
</tr>
<tr>
<td></td>
<td>TVLP (Bayesian)</td>
<td>1.22</td>
<td>2.62</td>
<td>2.16</td>
</tr>
<tr>
<td></td>
<td>TVLP (U, Bayesian)</td>
<td>1.97</td>
<td>3.43</td>
<td>2.94</td>
</tr>
</tbody>
</table>

is no available performance measure to quantify the improvement of Bayesian learning over least-squares for real speech formant tracking. At this point, we mention that the standard LP used 250 coefficients, whereas the TVLP methods used 60 coefficients (i.e. \( L = 60 \)).

4.5 Conclusions and Future work

We deal with the problem of estimating TVLP coefficients from speech samples, corresponding to an underdetermined linear system. We have found that a direct assumption of unstructured sparsity does not lead to a good estimation performance. Instead a subspace approach provides reliable performance. Further, among several estimation methods, Bayesian learning is found to be the best, as it has higher generalization capability to model different statistics of model parameters and driving noise.

TVLP, in general, has several applications, such as in detection theory, formant tracking or coding. Here we propose some of the tasks that can be performed thanks to the contribution:

- The flexibility of the method, allowing to have several dimensions, makes it interesting for detection theory. As an output of the method we also have the posterior probability of the hidden variables, in that case, the TVLP filters. Thus, one can use them to detect changes, such as formant changes or glottal openings and closures.

40
Figure 4.1: Spectrograms as outputs of different methods for real speech signal. (a) DFT spectrum (periodogram) using window length of 10 ms and frame shift of 5 ms. (b) for LP using least-squares with window length of 20 ms and frame shift of 10 ms. (c) for TVLP using least-squares with window length of 250 ms. (d) TVLP using Bayesian learning with window length of 250 ms.

- Also, TVLP methods greatly decrease the number of parameters in the model what makes it interesting for speech coding. However the higher computational cost compared with the standard methods doesn’t allow to implement it in speech coders yet.

4.6 Derivation of update equations

To maximize $p(x, d|\theta)$ we use the EM algorithm with prior assumptions to its parameters $\theta = \{\gamma, \beta\}$.

- 1. Choose an initial setting for parameters $\theta'$
- 2. E-step. Evaluate $Q(\theta, \theta') = \mathcal{E}_{d|x, \theta'}[\ln p(x, d|\theta)]$

From Bayes rule,

$$\ln p(x, d|\theta) = \ln p(x|d, \theta) + \ln p(d|\theta),$$

where we use the following distribution functions,

$$p(x|d, \theta) \sim \mathcal{N}(XTd, B),$$
$$p(d|\theta) \sim \mathcal{N}(d, \Gamma).$$
Then
\[ Q(\theta, \theta') = \text{constant} + \frac{1}{2} \ln \det(B) \]
\[- \frac{1}{2} (x - XTd)^\top B (x - XTd) \]
\[ + \frac{1}{2} \text{tr} \left( (XT)^\top BXTY' \right) \]
\[ + \frac{1}{2} \ln \det(\Gamma) - \frac{1}{2} \text{tr} \left( \Gamma \left( \Sigma' + \hat{d}\hat{d}^\top \right) \right). \]

3. **M-step.** Evaluate \( \theta^* \) given by
\[ \theta^* = \arg \max_{\theta} Q(\theta, \theta') + \ln p(\theta), \]
where
\[ \ln p(\theta) = \ln p(\gamma) + \ln p(\beta), \]
\[ \ln p(\gamma_i) = a\ln(\gamma_i) + b\gamma_i + \text{constant}, \]
\[ \ln p(\beta_i) = c\ln(\beta_i) + d\beta_i + \text{constant}. \]

By setting derivatives to zero, we find the update equations as follows
\[ \frac{\partial}{\partial \beta_i} = \frac{1}{2\beta_i} \left( \frac{1}{2} (x - XTd)_i^2 \right) \]
\[ - \frac{1}{2} (XT\Sigma'(XT)^\top)_{ii} + \frac{c}{\beta_i} - d = 0, \]
\[ \beta_i = \frac{1 + 2c}{(x - XTd)_i^2 + (XT\Sigma'(XT)^\top)_{ii} + 2d}. \]
\[ \frac{\partial}{\partial \gamma_i} = \frac{K}{2\beta_i} - \frac{1}{2} \sum_{j=(i-1)K+1}^{iK} (\Sigma' + \hat{d}\hat{d}^\top)_{jj} \]
\[ + \frac{a}{\gamma_i} - b = 0, \]
\[ \gamma_i = \frac{K + 2a}{\sum_{j=(i-1)K+1}^{iK} (\Sigma' + \hat{d}\hat{d}^\top)_{jj} + 2b}. \]
Chapter 5

Speech Enhancement

5.1 Introduction

5.2 Problem Formulation

Let \( y(n) \) be a noisy speech signal and \( y \in \mathbb{R}^N \) and N-point sequence as \( y = [y_1, y_2, \ldots, y_n] \) that follows an additive noise model:

\[
y = x + w,
\]

where \( w \) is the general environmental noise.

Here, \( x \) is the clean speech signal that can be modeled as a P-th order AR process according to its generative model.

\[
x = \tilde{x} + e = \sum_{k=1}^{P} a_k x_{n-k} + e_n = Xa + e,
\]

where

\[
X = \begin{bmatrix}
x_0 & x_{-1} & \cdots & x_{-P+1} \\
x_1 & x_0 & \cdots & x_{-P+2} \\
\vdots & \vdots & \ddots & \vdots \\
x_P & x_{P-1} & \cdots & x_1 \\
\vdots & \vdots & \ddots & \vdots \\
x_{N-1} & x_{N-2} & \cdots & x_{N-P}
\end{bmatrix},
\]

and \( a = [a_1, a_2, \ldots, a_P]^\top \) is the AR coefficients vector and \( e \) is the excitation signal vector. Note that, in this case, we have the initial condition \( x_{\text{ini}} = [x_{-P+1}, x_{-P+2}, \ldots, x_0] \) that consists of the \( P \) previous samples of the signal to be estimated and can be either known or unknown. How to deal with this initial vector will be explained later in this chapter. Furthermore, \( e \) is assumed to be a combined sparse and dense noise willing to model the excitation signal. The sparse noise is meant for modeling voiced excitation and dense noise for unvoiced excitation. Hence, combining (5.1) and (5.2) we have:

\[
y = Xa + e + w.
\]
Equation (5.4) assumes that the signal is known, as it is represented in the system matrix $X$. This is totally a false assumption, since it is what we try to estimate for speech denoising purposes. Thus, let us rewrite the equation assuming that the LP filter $a$ is known and being the clean speech signal $x$ the unknown. Note that, now, the number of parameters to be estimated has increased as $N \gg P$. Thus, an overfitting of the problem may occur. However, we know that the excitation signal $e$ is much sparser and has less statistical variability than $x$. By exploiting those properties and estimating $e$, it might lead to a better estimate of the speech signal, since the inverse filtering process is known. From (5.2),

$$x = Ax + A_{ini}x_{ini} + e = Ax + e_{ini} + e.$$  \hspace{1cm} (5.5)$$

where $A$ is the system matrix representing the LP filtering,

$$A = \begin{bmatrix} 0 & 0 & \ldots & 0 & 0 & \ldots & 0 \\ a_1 & 0 & \ldots & 0 & 0 & \ldots & 0 \\ a_2 & a_1 & \ldots & 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_P & a_{P-1} & \ldots & a_1 & 0 & \ldots & 0 \\ 0 & a_P & \ldots & 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \ldots & a_P & a_{P-1} & \ldots & a_1 & 0 \end{bmatrix}.$$ \hspace{1cm} (5.6)$$

$e_{ini}$ is a known sequence related to $x_{ini}$ as follows:

$$e_{ini} = A_{ini}x_{ini} = \begin{bmatrix} a_P & a_{P-1} & \ldots & a_1 \\ 0 & a_P & \ldots & a_2 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & a_P \end{bmatrix} \begin{bmatrix} x_{-P+1} \\ \vdots \\ x_0 \end{bmatrix}.$$ \hspace{1cm} (5.7)$$

and $B(a) = (I - A)$ is a full-rank matrix that represents the LP filtering process. It has been empirically shown that the matrix $B(a)$ may have poor condition number for some estimates of $a$. Then, in order to avoid numerical problems from computing inverse the inverse matrix, (5.1) can be rewritten as follows:

$$B(a)y = B(a)x + B(a)w = e + e_{ini} + B(a)w.$$ \hspace{1cm} (5.8)$$

Finally, the basic idea of this speech denoising model is to iterate between equations (5.4) and (5.8), and update the system matrix $X$ by filtering noise estimates $\hat{e}$ through the inverse filter of $A(z) = 1 - \sum_{k=1}^{P} \hat{a}_k z^{-k}$. Note that a stability check of the filter coefficients $\hat{a}$ is required if the estimation procedure doesn’t provide it, as seen in Ch.4. Thanks to the results found there, we will use the pole inversion technique to deal with stability issues. This idea is summarized in Algorithm 2.
Algorithm 2: Description of the speech denoising procedure

1. Set $X = Y$, i.e. filled with observed values.
2. Set $x_{ini} = 0$
3. while estimation is improving do
   4. **Step 1**
      5. Estimate $\hat{a}$ from (5.4)
      6. Stability check and correction: $\hat{a}_{stable} = \text{stable}(\hat{a})$
      7. Form matrix $B(\hat{a})$ using the stable coefficients $\hat{a}_{stable}$
   8. **Step 2**
      9. Estimate $\hat{e}$ from (5.8)
     10. Estimate $\hat{x}$ from the filtered version of $\hat{e}$:
         \[
         x(n) = \sum_{k=1}^{P} a_k x(n-k) + e(n)
         \]
         using the initial values $x_{ini}$.
     11. Update matrix $X$ with $\hat{x}$ estimate values.
4. end

5.2.1 Update rule

The Algorithm 2 refers to a single speech frame. To reconstruct all the signal, we should split the speech signal in different frames and run the algorithm on each one. Then, we can use values of the last speech frame to initialize the variable $x_{ini} = [x(L - P + 1), \ldots, x(L)]^{k-1}$ where $L$ is the update length. This update length can be different from the analysis length, referred as $N$. Looking at the simple example of an AR(1) process one can argue that small values of $L < N$ should be used:

**AR(1) example**  Let us consider the simple example of an AR(1) process as

\[
    x(n) = ax(n-1) + e(n)
\]

where $e(n)$ is a i.i.d gaussian noise. Then, we observe some samples of this process and we use the Algorithm 2 to find out estimated values of $\hat{a}$ and $\hat{e}$. To recover the estimated signal $\hat{x}(n)$ we perform the inverse filtering operation:

\[
\begin{align*}
\hat{x}(0) &= 0 \\
\hat{x}(n) &= \hat{a} \hat{x}(n-1) + \hat{e}(n) \quad \forall \quad n = 1, \ldots, L
\end{align*}
\]

Note that the inverse filtering operation can also be written as:

\[
\hat{x}(n) = \sum_{k=0}^{n} \hat{a}^k \hat{e}[n - k]
\]

and from here one can see that the error increases as $n \uparrow \uparrow$ increases.
Proof. We will model the estimation error as \( \delta a, \delta e, \delta x \)

\[
\hat{x}(n) = \sum_{k=0}^{n} (a + \delta a)^k (e + \delta e)[n-k] = \sum_{k=0}^{n} \left( \sum_{i=0}^{k} \frac{k!}{i!(k-i)!} a^i \delta a^k \right) (e + \delta e)[n-k]
\]

\[
= \sum_{k=0}^{n} \left( \sum_{i=0}^{k} \frac{k!}{i!(k-i)!} \left( \frac{\delta a}{a} \right)^i \right) (e + \delta e)[n-k]
\]

\[
= \sum_{k=0}^{n} a^k e[n-k] + \sum_{k=0}^{n} a^k e[n-k] \left( \sum_{i=1}^{k} \frac{k!}{i!(k-i)!} \left( \frac{\delta a}{a} \right)^i \right) + \sum_{k=0}^{n} a^k \delta e[n-k] \left( \sum_{i=0}^{k} \frac{k!}{i!(k-i)!} \left( \frac{\delta a}{a} \right)^i \right)
\]

\[
= x(n) + \delta x(n)
\]

where both

\[
\sum_{i=1}^{k} \frac{k!}{i!(k-i)!} \left( \frac{\delta a}{a} \right)^i, \quad \sum_{i=0}^{k} \frac{k!}{i!(k-i)!} \left( \frac{\delta a}{a} \right)^i
\]

are monotonically increasing functions with \( k \). As \( k = 0, \ldots, n \), they are also monotonically increasing functions with \( n \).

We consider the error \( \delta e \) to be Gaussian distributed. Thus, the estimation error of the signal \( \delta x \) has two terms, each of them is a AR process multiplied each time \( n \) with a monotonically increasing function. Then the total variance of the process \( \delta x \) will increase as \( n \) ↑↑ increases.

5.3 Statistical model definition

The estimation process stated in Algorithm 2 can be tackled from different perspectives. In this thesis we propose a fully Bayesian framework estimation. Hence, we need to specify the probabilities of all the variables involved.

Let us consider first the equation (5.4), where \( y = Xa + e + w \).

We will consider that the system parameters are distributed as follows:

\[
a \sim \mathcal{N}(0, \gamma I)
\]

\[
e \sim \prod_{k=1}^{N} \mathcal{N}(0, \Gamma_k)
\]

(5.10)

\[
w \sim \mathcal{N}(0, \alpha^{-1})
\]

where \( a \in \mathbb{R}^P \) and come from an i.i.d Gaussian source, \( e \in \mathbb{R}^N \) is an independent Gaussian variable with different precisions for each dimension, i.e. with a diagonal matrix \( \Gamma = \text{diag}(\Gamma_1, \Gamma_2, \ldots, \Gamma_N) \), and \( w \in \mathbb{R}^N \) is the general additive noise term, defined with a general covariance matrix capable to deal with different kind of noises.

The precision of each variable are treated as hyperpriors distributed as follows:

\[
\alpha \sim \mathcal{N}(0, \alpha^{-1})
\]
\[ p(\gamma) \sim \text{Gamma}(\gamma|a+1,b) \]
\[ p(\Gamma_i) \sim \text{Gamma}(\Gamma_i|c+1,d) \quad \forall i = 1,2,\ldots,N \]  
(5.11)
\[ p(\alpha) \sim W_N(\nu_\alpha, S_\alpha) \]

where
\[
\text{Gamma}(\gamma|a+1,b) \propto \gamma^a e^{-b\gamma}
\]
\[
\log (W_N(\alpha|v_\alpha, S_\alpha)) \propto \frac{v_\alpha - N - 1}{2} \log(\alpha) - \frac{1}{2} \text{tr}(S_\alpha^{-1} \alpha)
\]  
(5.12)

We seek a sparsity promoting distribution for \( e \) in order to better fit the voiced excitation. For this purposes, as stated and proved in Ch.2, we will use weakly informative hyperprior, that is, small values for \( c = d \sim 10^{-3} \). On the other hand, for the filter parameters \( a \) we seek a weakly informative prior, that means a flat prior which corresponds with high variance \( \gamma \). To this purposes, we set the hyperpriors parameters to \( a = 10^{-3} \) and \( b = 10^2 \). As a prior for the precision matrix \( \alpha \) we will use a Wishart distribution. Wishart distribution is the conjugate prior for precisions matrices. We will use its degree of freedom \( v_\alpha \) equals to its dimension (i.e. \( N \)) and the scale matrix \( S_\alpha \) as an initial guess of the covariance matrix. However, they require quite a lot of parameters to be estimated and, as reported in [4], they are known to have some modeling problems. For denoising purposes and looking at empirical results, Wishart prior works for the noises treated in this thesis.

Typically, in practice, in order to infer the precisions, one seeks the ML estimate \( p(y|\gamma, \Gamma, \alpha) \). Instead, we take the alternative (fully Bayesian) approach of maximizing directly the joint probability \( p(y,\gamma, \Gamma, \alpha) \) using EM algorithm and \( a \) as a hidden variable. Both become equivalent if non-informative hyperpriors are used for the precisions, but due to the undetermined set up of the problem, one should provide some extra information to relax the problem demands and avoid overfitting.

As stated in the Ch.2, the estimation procedure consists of finding the best a posteriori estimate of \( a \) fixing the precision values and iteratively update the precisions. For brevity, we translate (5.4) to the equivalent setup:
\[
y = \begin{bmatrix} X & I_N \end{bmatrix} \begin{bmatrix} \hat{a}^\top \\
\hat{e}^\top \end{bmatrix} + w = Gz + w
\]  
(5.13)
\[
C = \begin{bmatrix} \gamma I_p & 0 \\
0 & \Gamma \end{bmatrix}.
\]  
(5.14)

where \( C \) is the precision matrix of \( z \). Then, we first estimate the joint vector \( [a^\top e^\top]^\top \)
\[
\begin{bmatrix} \hat{a}^\top \\
\hat{e}^\top \end{bmatrix} = \Sigma^{-1} G^\top \alpha y
\]  
(5.15)
\[
\Sigma = (C + G^\top \alpha G)^{-1},
\]  
(5.16)

and update the precisions using an EM framework with \( z \) as a hidden variable and maximize the posterior probability found in the previous iteration:

47
maximize $Q(\theta, \theta^{new}) = \mathbb{E}_z[p(z|y, \gamma, \Gamma, \alpha)]$ \hspace{1cm} (5.17)

$\theta = \{\gamma, \Gamma, \alpha\}$ \hspace{1cm} (5.18)

Then, the updates of the hyperparameters result as follows:

$$\gamma^{new} = \frac{P + 2a}{||\hat{a}||^2 + \text{tr}(\Sigma_{11} + 2b)}$$ \hspace{1cm} (5.19)

$$\Gamma_i^{new} = \frac{1 + 2c}{\hat{e}_i^2 + ||\Sigma_{22}||_{ii} + 2d}$$ \hspace{1cm} (5.20)

$$\alpha^{new} = (v_\alpha - N - 1) \left( (y - G\hat{z})(y - G\hat{z})^\top + G\Sigma G^\top + S_\alpha \right)^{-1}$$ \hspace{1cm} (5.21)

where

$$(C + G^\top \alpha G)^{-1} = \begin{pmatrix} \gamma I_P + X^\top \alpha X & X^\top \alpha \\ \alpha X & \Gamma + \alpha \end{pmatrix} = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}$$ \hspace{1cm} (5.22)

Note that although $\alpha$ is symmetric, we use a full matrix derivative rather than the formula for symmetric matrices. This means that we look for a minima over all matrices and not just the symmetric matrices. However, since the optimum is found to be a symmetric matrix, this still gives the correct minima.

Now, let’s consider the second equation (5.8), where

$$t = By - e_{ini} = e + B(a)w = e + n,$$

assuming that we know the filter parameters $a$, that is, approximating it with the previous estimate $\hat{a}$. We will assume that the parameters are distributed as follows:

$$e \sim \prod_{k=1}^N \mathcal{N}(0, \Gamma_i)$$

$$n \sim \mathcal{N}(0, \Upsilon^{-1})$$

with $\Upsilon = \text{diag}(\Gamma_1, \Gamma_2, \ldots, \Gamma_N)$. Here, we also seek an sparsity promoting distribution for the excitation signal $e$ and a general covariance matrix for the additive noise term $n$, that is able to deal with any kind of noise. For this purpose, we define the following hyperpriors:

$$p(\Gamma_i) \sim \text{Gamma}(\Gamma_i|a + 1, b) \quad \forall i = 1, 2, \ldots, N$$ \hspace{1cm} (5.23)

$$p(\Upsilon) \sim \mathcal{W}_N(S_{\Upsilon}, v_\Upsilon)$$ \hspace{1cm} (5.24)

where the parameters of the hyperprior distribution will be set to small values $a = b = 10^{-3}$. As before, we will use an iterative framework, where first we fix the hyperparameters to estimate the desired signal $e$ and then we use the EM algorithm to update the hyperparameters update, assuming the previous estimate $e$ as the hidden variable. Thus, the estimated excitation signal will be
\[
\hat{e} = \Sigma^{-1} \Upsilon z
\]
\[
\Sigma = (\Upsilon + \Gamma)^{-1},
\]
with the updates of the precisions as follows:
\[
\Gamma_{i}^{\text{new}} = \frac{1 + 2a}{\hat{e}_{i}^{2} + \Sigma_{ii} + 2b}
\]
\[
\Upsilon^{\text{new}} = (v_{r} - N - 1) \left( (t - \hat{e})(t - \hat{e})^{T} + \Sigma + S_{r} \right)^{-1}.
\]
where we also find out that a symmetric matrix is the optimal among all matrices.

### 5.4 Experimental results

In this section we will analyze the case where the additive noise follows a Gaussian distribution. Other kind of noises are under investigation since it turns out that modeling the covariance with a general wishart prior doesn’t work properly. The reasons can be:

- When the covariance is full, too many parameters might be estimated.
- The use of nearly non-informative prior will make difficult to reduce the overall complexity of the problem and lead us to overfitting.

Some improvements that can be done are:

- If the noise follows some structure, it will reduce drastically the number of parameters to be estimated.
- Prior knowledge of the noise can make the prior more informative and thus improve the overall system.

Thus, while in future works we will experiment the general noise case, with special attention when the noise follow an AR process (source separation problem), the current results make us optimistic to achieve also good performance in the general noise case.

In order to carry out some experiments, we will use clean speech signals sampled at \( f_s = 16 \) kHz from the TIMIT database. As said in the model definition, we will use different window lengths to analyze the signal and to update the denoised version

and in our case we will use the following values:

\[ L = 40 \text{ samples}, \quad N = 40, 160, 320 \]
\[
x_{i\text{ini}} \text{ updated from the last estimation, i.e.}
\]
\[
x_{i\text{ini}}^{\text{update}} = [\hat{x}^{i-1}(L-2), \hat{x}^{i-1}(L-1), \ldots, \hat{x}^{i-1}(L)]
\]

---

The results obtained are shown in Fig. 5.2. The first conclusion is that for small analysis frame lengths \((N \sim 40)\) the performance is not good enough, it completely fails for \(SNR > 5dB\). On the other hand, we have reasonable performance when \(N > 160\) samples because the AR parameters are better found at the first equation. Besides, one easily sees that for low input SNR values \((SNR < 0dB)\) the improvement is much better. The reason for that is that when a frame has an instantaneous SNR low, the method doesn’t try to recover the signal, that is, the output \(SNR \to 0dB\). For this reason, we have great improvement for low input SNR and low values of analysis frame length.

5.4.1 Conclusions and future work

Speech denoising is an old topic. The first approaches were using Wiener filter in an iterative way in the time domain. More recently, the most used techniques are in frequency domain (spectral components) or working in the time-frequency...
domain (spectrograms). Several methods involving both domains have been reported to be successful. Nowadays, array processing techniques when several microphones are available are also investigated, which means that another domain is taken into account: space.

In this chapter, we have proposed a fully Bayesian framework to recover clean speech signals in frequency domain. This approach might be difficult because it involves finding out the AR parameters of the clean speech signal in presence of noise. Dealing with noises generically distributed and with full covariances is also a complicated issue. However, we have achieved quite good results that encourage us to continue investigating and improving this method. The final goal of the writer is to develop a method that is able to deal with noises with an AR structure and thus, tackle the source separation problem.
Bibliography


52


