l_1-NORM SPARSE BAYESIAN LEARNING: THEORY AND APPLICATIONS

Yuanqing Lin

A DISSERTATION

in

Electrical and Systems Engineering

Presented to the Faculties of the University of Pennsylvania

in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

2008

Supervisor of Dissertation

Graduate Group Chair

Acknowledgements

I first want to thank my advisor, Dr. Daniel D. Lee, who offered me a fantastic opportunity to become a researcher in the field of machine learning. When I was starting my Ph.D. study in Spring 2003, I (literally) knew little about machine learning since my major had been optical engineering. It was very fortunate for me to have the chance to work with Dan. I am grateful for his knowledgable guidance and passionate support in the past five and half years. Without him, I would never be where I am.

I would also like to thank my committee members. I am thankful to Dr. Jingdong Chen (Bell Labs) who was also one major collaborator of mine on our acoustic signal processing projects. He provided tremendous help with recognizing the frontier of acoustic signal processing research as well as conducting experiments in an anechoic chamber. My discussions with him were always pleasant and fruitful. I thank Dr. Lawrence Saul for helpful discussions and suggestions, as well as our collaboration on a couple of papers. Moreover, his lectures have been very beneficial to me, and I am glad that I took all the courses that he had taught at Upenn before he moved to UCSD. I also thank Dr. Saleem A. Kassam for chairing the committee and providing much helpful comments and suggestions.

Besides Jingdong and Lawrence, I am thankful to my other collaborators during my Ph.D. study. I thank Dr. Youngmoo E. Kim (Drexel University) for allowing me to use the acoustic facility in his lab and beneficial discussions on advancing our acoustic signal processing projects. I also thank Dr. Bill Hanson and Dr. Erica Thaler (both from Hospital

of the University of Pennsylvania) for involving me in the electronic nose project where I learned and implemented many classification algorithms. I would also like to thank Dr. Ben Taskar for our collaboration on learning sparse Markov networks, although this work is not included in this thesis.

Meanwhile, I thank many of my wonderful colleagues and friends, Koby Crammer, Ricky Der, Jihun Hamm, Dan Huang, Yung-Kyun Noh, Fei Sha, Paul Vernaza, Qihui Zhu, for much joyful time together as well as many useful discussions.

I also want to thank Dr. Britton Chance for offering me the first opportunity to study in the US and later encouraging me to pursuit a Ph.D. degree. Without his kind help and encouragement, my transition to this Ph.D. program would have not been so smooth.

Last but not least, I want to thank my family. I am indebted to my parents, Yunchi Lin and Ailan Chen, and my grandmother Chunlian Yu. Although our family was not rich, they did everything they could to support my study (fortunately, I was able to finance myself after my third year in college). I also thank my bother Yuanwei Lin and my sister Lijuan Lin for their support. Most importantly, I want to thank my wife Fanglian He who is also my best friend and soulmate. She has been the critical factor that helps me to go through the ups and downs in my life and career. Finally, I thank my newborn daughter, Licia Lin. She is the most beautifully blessing to my life.

ABSTRACT

l_1-NORM SPARSE BAYESIAN LEARNING: THEORY AND APPLICATIONS

Yuanqing Lin

Supervisor: Daniel D. Lee

The elements in the real world are often *sparsely* connected. For example, in a social network, each individual is only sparsely connected to a small portion of people in the network; for certain disease (like breast cancer), even though human have tens of thousands of genes, only a small number of them are connected to the disease; for a filter modeling an acoustic room impulse response, only a small portion of filter coefficients are nonzero. Discovering the sparse representations of the real world is important since they provide not only the neatest insight for understanding the world but also the most efficient way for changing the world. Therefore, finding sparse representations has attracted a great amount of research effort in the past decade, and it has been a driving force for many exciting new fields, such as sparse coding and compressive sampling.

The research effort on finding sparse representations has covered both theories and applications. In its theoretic aspects, researchers have developed many approaches (such as nonnegative constraint, l_1-norm sparsity regularization and sparse Bayesian learning with independent Gaussian prior) for encouraging sparse solutions and established some conditions under which the true solutions (which are sparse) could be found by those approaches. Meanwhile, finding sparse representations has found its applications in a wide spectrum of fields such as acoustic/image signal processing, computer vision, natural language processing, bioinformatics, finance modeling, and so on.

However, despite of the intense studies in finding sparse solutions in the last decade, there is a fundamental issue still remained almost untouched, that is, how sparse is the *optimally* sparse in representing given data?

This thesis aims to answer the above fundamental question by establishing a theory of l_1-*norm sparse Bayesian learning*. In particular, using l_1-norm regularized least squares as an example, we show how the l_1-norm sparse Bayesian learning extends the conventional uniform l_1-norm sparsity regularization, where all variables desired to be sparse share a single scalar regularization parameter, to *independent* l_1-norm sparsity regularization, where each variable is associated with an independent regularization parameter. In the independent l_1-norm sparsity regularization, the optimal sparseness of solutions is then fully defined in a Bayesian sense via the optimal l_1-norm sparsity regularization parameters and inferred by learning directly from data. This is why we call our Bayesian approach *sparse learning*, which is very different from conventional methods where there is only single l_1-norm regularization parameter and it is determined by ad-hoc manners (like cross-validation).

The proposed l_1-norm sparse Bayesian learning shows superior performance in both simulations and real examples. Our simulation results demonstrate that the l_1-norm sparse Bayesian learning is able to accurately resolve the true sparseness in solutions even in very noisy data, and it provides better performance than the conventional uniform l_1-norm regularization and l_2-norm Bayesian sparse learning (also known as relevance vector machine). In real examples, we show the l_1-norm sparse Bayesian learning is effective for speech dereverberation and acoustic time different of arrival (TDOA) estimation in reverberant environments, both of which are hard problems and have remained open problems after a long history of research.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

It is well-known that we are in an era where data collected are growing exponentially along time. The exponential growing is powered by the growing in information hardware described by the Moore's law, which says that the number of transistors that can be inexpensively placed on an integrated circuit is increasing exponentially, doubling approximately every two years.

Consequently, there are many challenges arising in how to effectively handle the ever-growing amount of data. One challenge is about how to represent the data so that they can be stored and transferred *efficiently*. This challenge has attracted substantial research and engineering effort in the past four decades, and the field is well-known as data compression [1]. The second challenge is about how to represent the data in a *meaningful* way. The quote, *drowning in data, thirsting for knowledge*, vividly describes this challenge and reflects the lack of effective tools for discovering the meaningful information from the ever-growing amount of data. Both these two challenges can be addressed by finding sparse representation of data, as explained in the following sections. Simply speaking, finding a sparse representation of data is to represent the data in some way so that the representation has many zeros.

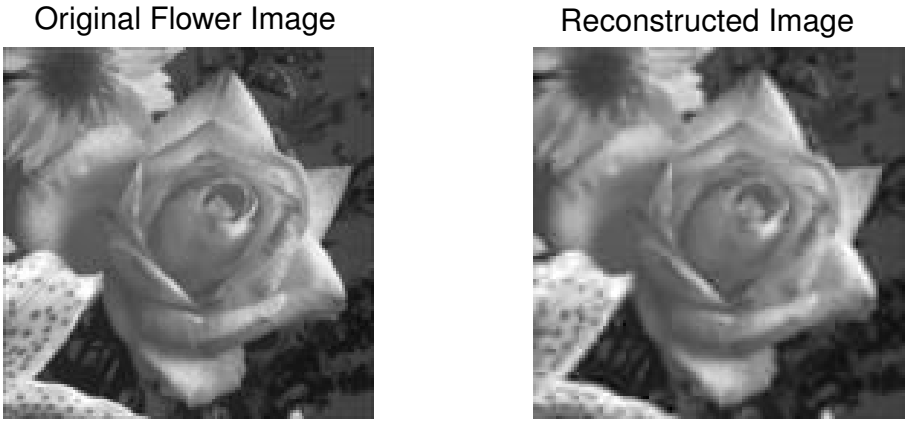


Figure 1.1: An original image and its reconstruction using only 25% of DCT coefficients.

1.1 Sparsity for efficiently representing data

As for the challenge on how to represent data efficiently, finding sparse representation of data in certain domain is an important rule for data compression. For example, in image compression, the current standard, JPEG [3], explores the fact that an image can be represented by a small number of components in the discrete cosine transform (DCT) domain [58] (as illustrated in Fig.1.1). In other words, the image compression is done by setting many DCT coefficients to be zero and representing the image using a sparse vector of DCT coefficients.

While DCT bases have been a popular choice for acoustic/image/video data compression, researchers have aggressively seeked for new bases (or code book) that allow data representations to be even more sparse [56] [64]. This is a very new field in data compression, and it is generally known as *sparse coding*. The sparse coding is now believed to be a ubiquitous sensing strategy employed in several different modalities across different biological organisms for maximizing data representation efficiency [57]. As such, sparse coding not only provides a promising way for data compression, but also is a valuable tool for studying how biology handle data and information.

A very exciting new paradigm for data compression is *compressive sampling* [12]. The

compressive sampling suggests that [15] [16], if a signal is known to be very sparse in frequency domain, it needs fewer samples for signal reconstruction than it would be required by the conventional Nyquist-Shannon sampling theorem. The compressive sampling is also known as *nonlinear sampling theorem*. In contrast to the conventional sampling theorem where *linear* interpolation is employed for signal reconstruction, the reconstruction in compressive sampling is done first by a nonlinear process which demands *the sparsest representation* in frequency domain for explaining the samples acquitted in time domain and then the time domain signal is reconstructed by the sparsest frequency domain representation. Since the Nyquist-Shannon sampling theorem is the foundation of the modern information theory, the nonlinear sampling theorem is now considered as a revolutionary step in information theory.

To summarize, finding sparse representation of data is a key magic for many data compression techniques that have been widely deployed as well as those that are active in research.

1.2 Sparsity for meaningfully representing data

The second challenge, how to represent data in meaningful ways, is closely related to knowledge discovery [27]. One powerful tool for knowledge discovery is dimensionality reduction, and learning sparse representations is a special approach for dimensionality reduction [26].

Figure 1.2 shows an example of gene selection, illustrating how finding sparse representations can be utilized for knowledge discovery. Today's microarray technology is able to measure the expression levels of tens of thousands genes [11]. Then, given gene expression measurements, one important task is to find which subset of genes is relevant to a certain biological phenotype, for example, breast cancer. In order to discover the genes

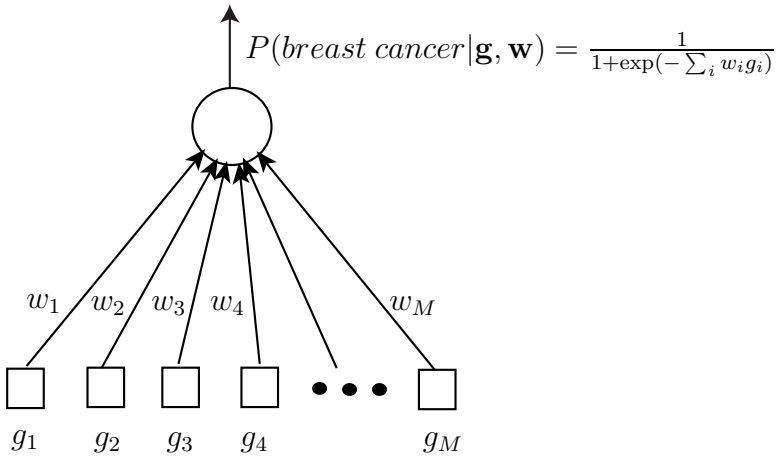


Figure 1.2: Gene selection.

that are relevant to breast cancer from microarray data acquired from both healthy people and breast cancer patients, we can build a simple binary logistic regression model (shown in Figure 1.2) [23]:

$$P(breast\ cancer|\mathbf{g},\mathbf{w}) = \frac{1}{1+\exp(-\sum_i w_i g_i)} \tag{1.1}$$

where $\mathbf{g} = [g_1; g_2; ...; g_M]$ are expression levels of all genes with M being the total number of genes, $\mathbf{w} = [w_1; w_2; ...; w_M]$ are weighting factors, and the probability of breast cancer is a function of the weighted sum of the expression levels of the genes. In this model, if a gene (g_i) is relevant to the breast cancer, it should be associated with a nonzero weight (w_i); in contrast, if a gene is irrelevant to the breast cancer, its weight must be zero. As a result, discovering the subset of relevant genes is about finding a sparse solution of the weight vector $\mathbf{w}$.

The gene selection example indeed describes a very typical scenario that we are facing today in knowledge discovery [2]. When building a system (for example, a disease diagnosis system, or a financial stock price predictor), we in principle can have as many sensors (or information sources) as possible to be the inputs of the system. However, the reality

is that only a small number of them are truly relevant to the system, and discovering the truly relevant inputs is a very challenging task. This becomes especially challenging when the inputs are in very high dimension. Fortunately, finding a sparse representation of data can be a powerful tool for sensor selection (or feature selection) [32] since it sets irrelevant sensors to be zero-weighted.

Besides variable selection, some knowledge discovery tasks are interested in finding the interaction between variables, for example, finding the regulatory pathway of a gene network [19]. The interactions between variables are described by a network with the variables being its nodes and interactions being described by its edges. As such, finding the interaction between variables is to find a proper network that represents them. Since the number of edges in a network is squared with respect to the number of nodes, learning a network is often very challenging because we have to deal with a problem with super-high dimension (say dimensionality is in the order of 10^6 or more). However, the good news is, in a real network, each variable only interacts with a small number of other variables in the network. Therefore, it is natural to assume that the network is sparse in the sense that the network is not fully connected and many edges do not exist. As a result, sparsity regularization plays an important role in discovery networks, as illustrated in some recent work on learning sparse Markov network [6] [68] [44].

To summarize, finding sparse solutions is an important task, and it becomes more and more essential to many applications (such as signal processing, pattern recognition, data mining, and so on) related to data storage and analysis.

This thesis is organized as the following. Chapter 2 reviews the previous work on how to find sparse solutions. We emphasize that, despite that much research effort has been devoted to studying how to find sparse solutions, a fundamental issue, how to find the *optimally* sparse representation of given data, has remained almost untouched. This motivates our proposal, l_1*-norm sparse Bayesian learning*, for finding the optimally sparse solutions

in a Bayesian sense. In Chapter 3, we describe our l_1-norm sparse Bayesian learning approach in details using examples such as l_1-norm regularized ordinary least squares and l_1-norm regularized nonnegative least squares. Simulations are employed to demonstrate that the proposed l_1-norm sparse Bayesian learning approach is able to accurately resolve the true sparseness in solutions. We also look into the possible applications of the l_1-norm sparse Bayesian learning to other problems like l_1-norm regularized logistic regression and finding sparse Markov networks. In Chapter 4 and Chapter 5, we show how the l_1-norm sparse Bayesian learning can be utilized for solving challenging real problems, speech dereverberation and acoustic time difference of arrival (TDOA) estimation in reverberant environments. At last, a brief discussion is conducted in Chapter 6.

Chapter 2

Related work in how to find sparse solutions

As we have described in Chapter 1, sparse representation is not only desirable but also crucial in many applications. As a result, how to find a properly sparse solution in a given problem is an important issue, and there has been much research effort under this topic in the past ten years. To make our presentation concrete, let's say the problem is to find a sparse solution that minimizes an objective function $f(\mathbf{w})$ and the optimization variable $\mathbf{w}$ ($M \times 1$ vector) is desired to be sparse, namely,

$$\begin{aligned} \mathbf{w}^* &= \arg\min_{\mathbf{w}} f(\mathbf{w}) \\ \text{subject to}: \ & \mathbf{w} \text{ is sparse.} \end{aligned} \tag{2.1}$$

The above sparsity constraint may be given quantitatively, for example, $\|\mathbf{w}\|_0 \leq c$ with c being some nonnegative constant, $\|\cdot\|_0$ denoting l_0-norm and $\|\mathbf{w}\|_0$ being the count of nonzero elements in $\mathbf{w}$. However, the sparsity constraint is often just a qualitative statement since the quantitative bound is unknown and the interest is to find some *appropriately*

sparse solution. The appropriateness of a solution is often justified by heuristic manners or by cross-validation.

There are many approaches for encouraging sparseness in the solutions of the optimization in Eq. 2.1, and they can be roughly classified into the following three categories: 1) enforcing nonnegative constraint on $\mathbf{w}$ when $\mathbf{w}$ is known to be nonnegative; 2) adding a sparsity regularization term (for example, l_1-norm of $\mathbf{w}$) to the objective function; 3) sparse Bayesian learning with independent Gaussian priors. We will review those three categories of approaches in the following.

2.1 Nonnegative constraint

If the variable $\mathbf{w}$ is known to be nonnegative in *priori*, then the optimization

$$\begin{aligned} \mathbf{w}^* &= \arg\min_{\mathbf{w}} f(\mathbf{w}) \\ \text{subject to}: \ & \mathbf{w} \geq 0 \end{aligned} \tag{2.2}$$

with nonnegative constraint $\mathbf{w} \geq 0$ will often yield a sparse optimal solution $\mathbf{w}^*$. This is because the optimal solution $\mathbf{w}^*$ is at the boundary of nonnegative orthant where many coordinates are zeros, as illustrated in Fig 2.1.

Nonnegative matrix factorization (NMF) [40] is a well-known example of using nonnegative constraint for enforcing spareness. NMF decomposes a matrix $\mathbf{V}$ into

$$\mathbf{V} \approx \mathbf{W}\mathbf{H} \tag{2.3}$$

where $\mathbf{V}$ is an $m \times n$ matrix, $\mathbf{W}$ is an $m \times r$ matrix, $\mathbf{H}$ is a $r \times n$ matrix, and the elements in both $\mathbf{W}$ and $\mathbf{H}$ are constrained to be nonnegative. In a typical NMF setting, r is much smaller than both m and n. It has been shown that, when the columns of $\mathbf{V}$ are pixel

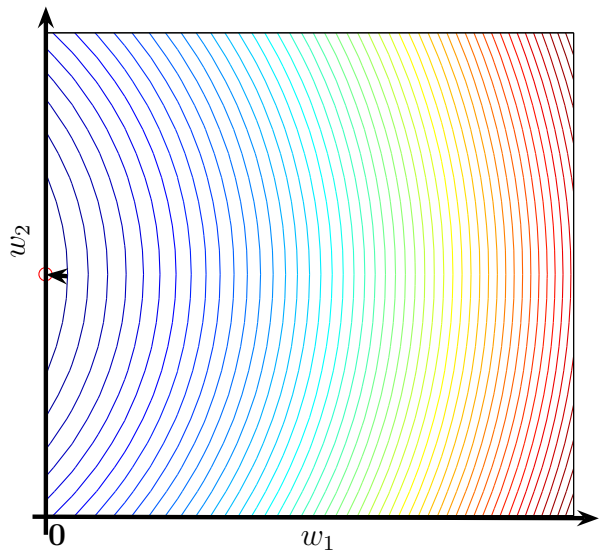


Figure 2.1: The contour plot of a function with nonnegative constraint. The optimal (indicated by the circle) is at the boundary where some coordinates are zeros.

values of human faces, images of the faces are decomposed into sparse bases (so-called parts) described by the columns of $\mathbf{W}$, and each image is a sparse linear combination of the sparse bases with the combination being described by the column of $\mathbf{H}$ associated with the image [40].

Support vector machine (SVM) is another interesting example of using nonnegative constraint for encouraging sparse solutions [61]. Given data $\{\mathbf{x}_i, y_i\}_{i=1}^N$ with $\mathbf{x}_i \in \mathbb{R}^M$ and $y_i \in \{-1, +1\}$, SVM is to find a hyperplane $y = \mathbf{w}^T\mathbf{x} + b$ that maximize the margin between the two classes of samples. The optimization problem for SVM is

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \frac{1}{2}\mathbf{w}^T\mathbf{w} \tag{2.4}$$
$$\text{subject to: } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \;\; i = 1, 2, ...N.$$

The above optimization is often solved by its dual:

$$\begin{aligned} \boldsymbol{\alpha}^* &= \arg\max_{\boldsymbol{\alpha}}(\sum_i \alpha_i - \frac{1}{2}\sum_{i,j} y_i y_j \alpha_i \alpha_j \mathbf{x}_i^T \mathbf{x}_j) \\ &\text{subject to: } \alpha_i \geq 0, \;\; i = 1, 2, ...N \\ &\sum_i y_i \alpha_i = 0, \end{aligned} \tag{2.5}$$

with the primal-dual relationship at the optimum:

$$\mathbf{w}^* = \sum_i \alpha_i^* y_i \mathbf{x}_i. \tag{2.6}$$

It is now well-known that the optimal solution $\boldsymbol{\alpha}^*$ is sparse, and those $\mathbf{x}_i$ associated with nonzero α_i^* are called *support vectors*. From the optimization point of view, the sparseness of $\boldsymbol{\alpha}^*$ is because of its nonnegative constraint.

The nonnegative constraint is a useful sparsity regulator, and this will be further illustrated in Section 3.2 in Chapter 3. The advantage of nonnegative constraint is its simplicity. It comes naturally without the plague of setting regularization parameters (although it can be incorporated with l_1-norm sparsity regularization where proper l_1-norm regularization parameters need to be set [34]).

2.1.1 Optimization methods

There are different algorithms for solving the optimization in Eq. 2.2, and the choice of algorithms depends on whether $f(\mathbf{w})$ is convex and whether the optimization has extra constraints other than the nonnegative constraint.

First, if the optimization in Eq. 2.2 is convex, it can be treated as a general constrained optimization problem and solved by an interior point method using log-barrier. It is known that, if the log-barrier interior point method employs Newton's update rule, the optimization

can be solved in a small number of steps ($\leq$ 30 steps) with high accuracy regardless of the size of $\mathbf{w}$. However, the log-barrier interior point method is feasible only when the optimization problem is *convex*. This is because, if the optimization problem is nonconvex, the centering procedure in the log-barrier interior point method may shift the solution to be far away from the initial solution, and the algorithm is possible to be stuck in an even worse solution than the initial due to the nonconvexity of the optimization problem. Therefore, the log-barrier interior point method is applicable only when the optimization is convex.

Second, if the optimization in Eq. 2.2 has only the nonnegative constraint, it can be solved by projected gradient descent [8]. In fact, when an optimization only has element-wise bound constrains, $l_i \leq w_i \leq u_i$ with l_i and u_i respectively being lower-bound and upper-bound of w_i, projected gradient descent is a usual choice for solving the optimization. This is because, in the case of such constraints, it is easy to project a point (say $\hat{\mathbf{w}}'$) onto the domain described by the constraints. It can be shown that the following simple operation indeed finds the closest point, denoted as $\hat{\mathbf{w}}$, in the domain to the point $\hat{\mathbf{w}}'$:

$$\hat{w}_i = \begin{cases} l_i & \text{when} \quad \hat{w}'_i < l_i \\ \hat{w}'_i & \text{when} \quad l_i \leq \hat{w}'_i \leq u_i \\ u_i & \text{when} \quad \hat{w}'_i > u_i. \end{cases}$$

The nonnegative constraints are a special case of bounded constraints with lower-bound being zeros and upper-bound being infinity. Therefore, the projected gradient descent is suitable for the optimization in Eq. 2.2. The work in [46] illustrated how the projected gradient descent was utilized for solving the optimization in NMF. The projected gradient descent algorithm is guaranteed to monotonically converge to a local minimum if its step sizes are determined by a Armijo like line search, as described in [8].

Third, there are also some problem specific algorithms for solving the optimization in Eq. 2.2. For example, one well-known algorithm for solving NMF is a multiplicative update

algorithm [41], a block descent algorithm that alternates optimizing $\mathbf{W}$ and optimizing $\mathbf{H}$. The multiplicative update was derived by constructing an auxiliary function around the current estimate of $\mathbf{W}$ and $\mathbf{H}$, and it is guaranteed to monotonically converge to a nearby local minimum. Similarly, Sha *et al.* [62] developed a multiplicative update algorithm for solving nonnegative quadratic programming problems.

2.2 l_1-norm sparsity regularization

Another category of approaches for enforcing sparse solutions in Eq. 2.1 is to add a sparsity regularization term to the objective function, namely,

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} f(\mathbf{w}) + \lambda' g(\mathbf{w}) \tag{2.7}$$

where $g(\mathbf{w})$ is for penalizing non-sparse solutions, and $\lambda' \geq 0$ is a regularization parameter that balances the trade-off between the original objective function and the sparseness solutions described by $g(\mathbf{w})$. The natural choice of $g(\mathbf{w})$ would be the l_0-norm of $\mathbf{w}$ (denoted as $\|\mathbf{w}\|_0$), which is the count of nonzero elements in $\mathbf{w}$. However, optimizing Eq. 2.7 with $g(\mathbf{w}) = \|\mathbf{w}\|_0$ involves combinatorial search, and it is very hard to solve. As a result, $g(\mathbf{w})$ is often chosen to be some relaxed form of the l_0-norm. Among many choices of the relaxation such as l_p-norm ($p \leq 1$ and $p\ = 0$) [31], l_1-norm (denoted by $\|(w)\|_1 = \sum_j |w_j|$ with $|w_j|$ being the absolute value of w_j) is the most popular choice [56] [65][18]. Figure 2.2 illustrates why l_1-norm regularization encourages sparse solutions but l_2-norm regularization does not. The main advantage of l_1-norm regularization comes from its convexity. The resulting optimization will be convex if $f(\mathbf{w})$ is convex with respect to $\mathbf{w}$. As we will see in Section 2.2.2, many l_1-norm regularized problems can be efficiently solved by a variety of optimization algorithms.

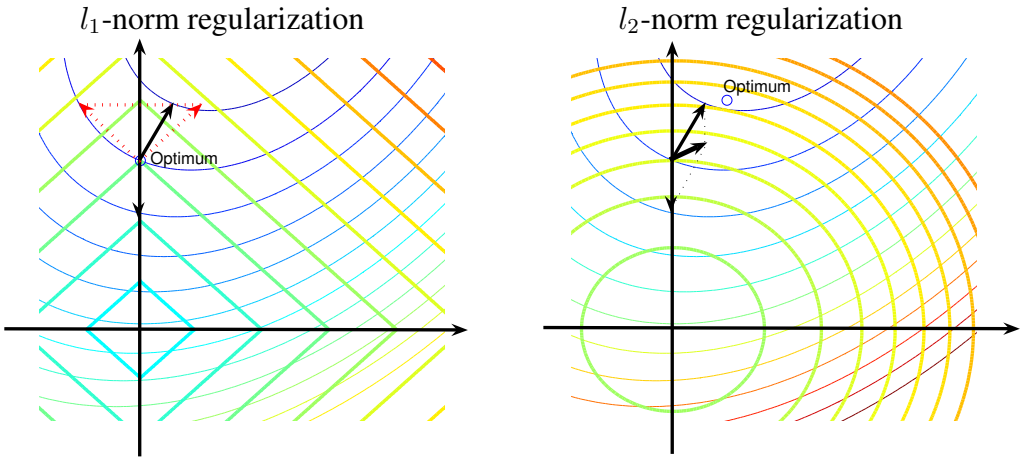


Figure 2.2: The contour plot of l_1-norm regularization (described by the series of tilted squares in the left figure), l_2-norm regularization (described by the series of circles in the right figure) and an original objective function $f(\mathbf{w})$ (described by the thin lines in both figures). For the l_1-norm regularization, when the optimal solution is on an axis, the derivative of the l_1-norm regularization term is a set (not a single vector). As such, the derivative of $f(\mathbf{w})$ is allowed to take different directions while the optimal solution keeps the same. The range of the allowance is indicated by the dashed triangle in the left figure. Therefore, with l_1-norm regularization, optimal solutions are often on axis and thus are sparse (since some coordinates are zeros). In contrast, for l_2-norm regularization, a slight deviation of the gradient of $f(\mathbf{w})$ from the axis direction will move a solution away from the axis.

When $g(\mathbf{w}) = \|\mathbf{w}\|_1$, the optimization in Eq. 2.7 becomes

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} f(\mathbf{w}) + \lambda' \|\mathbf{w}\|_1. \tag{2.8}$$

In the literature, the above optimization has two variant forms. One is known as LASSO [65] and has the form

$$\begin{aligned} \mathbf{w}^* &= \arg\min_{\mathbf{w}} f(\mathbf{w}) \\ \text{subject to}: \ & \|\mathbf{w}\|_1 \leq \beta', \end{aligned} \tag{2.9}$$

and the other is

$$\begin{aligned} \mathbf{w}^* &= \arg\min_{\mathbf{w}} \|\mathbf{w}\|_1 \\ \text{subject to}: \ & f(\mathbf{w}) \leq \gamma'. \end{aligned} \tag{2.10}$$

The form in Eq. 2.10 is especially popular when its constraints becomes equality constraints due to noiseless assumption in regression problems [12]. The Eqs. 2.8$\sim$2.10 are equivalent in the sense that, for example, given the optimization in Eq. 2.8 with λ', there exist a β' and a γ' such that all those three optimizations yield the same solution. Therefore, in the following presentation, we will ignore the specific form and assume that a statement for one form will easily lead to equivalent statements for the other two forms.

Enforcing sparseness using l_1-norm regularization has gained much research effort in the past decade, and the research is centered around the following three basic issues. First, *uniqueness* and *equivalence*: when a highly sparse solution is necessarily the sparsest possible solution in a given problem, and when the optimization with l_1-norm regularization will find the optimal solution of the one with l_0-norm regularization? Second, how to efficiently solve the optimizations in Eqs. 2.8$\sim$2.10? Third, how to choose the regularization

parameter λ' in Eq. 2.8 (or β' in Eq. 2.9, or γ' in Eq. 2.10)? In the following, we briefly review the research effort under these three issues.

2.2.1 Uniqueness and equivalence

The uniqueness and equivalence is a fundamental but very challenging issue in finding sparse solutions using l_1-norm regularization. Most existing research under this issue has been carried out under the simplest model possible, a linear noiseless model. Under such model, the optimization becomes

$$\begin{aligned} \mathbf{w}^* &= \arg\min_{\mathbf{w}} \|\mathbf{w}\|_1 \\ \text{subject to}: \ & \mathbf{\Phi w} = \mathbf{y} \end{aligned} \tag{2.11}$$

where $\mathbf{\Phi}$ is an $N \times M$ design matrix with full row rank and unitary columns ($\sum_i \Phi_{i,j}^2 = 1$ for $j = 1, 2, ..., M$), $\mathbf{y}$ is an $N \times 1$ data vector, $\mathbf{w}$ is an $M \times 1$ optimization variable, and $N < M$ (it is trivial if $N = M$, and it can be infeasible if $N > M$). Since $N < M$, the design matrix $\mathbf{\Phi}$ is often referred as *overcomplete* dictionary.

The *uniqueness* is to address the conditions under which a highly sparse solution (obtained by solving the optimization in Eq. 2.11 or other means) is necessarily the sparsest solution possible. In other words, when a given solution satisfies the conditions, it is surely the solution of the optimization with l_0-norm

$$\begin{aligned} \mathbf{w}^* &= \arg\min_{\mathbf{w}} \|\mathbf{w}\|_0 \\ \text{subject to}: \ & \mathbf{\Phi w} = \mathbf{b} \end{aligned} \tag{2.12}$$

which is known to be very hard to solve. Donoho *et al.* [21] showed that, if a solution $\mathbf{w}^*$

has its l_0-norm strictly less than $Spark(\mathbf{\Phi})/2$, namely,

$$\|\mathbf{w}^*\|_0 < Spark(\mathbf{\Phi})/2, \tag{2.13}$$

the solution $\mathbf{w}^*$ is necessarily the sparsest solution, where the $Spark$ of $\mathbf{\Phi}$, denoted as $\kappa = Spark(\mathbf{\Phi})$, is defined as the smallest possible number such that there exists a subgroup of κ columns from $\mathbf{\Phi}$ that are linearly dependent. Note that the $Spark$ of a matrix may sound similar to the rank of the matrix, they are two very different concepts. For example, for a matrix $\mathbf{\Phi}$ which has full row rank, $Spark(\mathbf{\Phi})$ can be as small as 2 (for example, when the $\mathbf{\Phi}$ matrix has two identical columns). The proof of the bound in Eq. 2.13 is fairly straightforward as shown in [21].

The *equivalence* is about the conditions under which a highly sparse solution to the l_0-norm minimization in Eq. 2.12 is also the solution to the l_1-norm minimization in Eq. 2.11. When the conditions are satisfied, one can solve the difficult optimization in Eq. 2.12 via solving a convex optimization problem in Eq. 2.11. D. L. Donoho *et al.* [21] showed that, if the solution to Eq. 2.12, denoted as $\mathbf{w}^*$, is sparse enough to satisfy either of the following two conditions,

$$\|\mathbf{w}^*\|_0 < (1 + 1/M(\mathbf{G}))/2 \tag{2.14}$$

$$\text{and} \quad \|\mathbf{w}^*\|_0 < \mu_{\frac{1}{2}}(\mathbf{G}), \tag{2.15}$$

the optimal solution $\mathbf{w}^*$ can be found by solving the optimization with l_1-norm regularization(in Eq. 2.11), where $M(\mathbf{G})$ is the largest off-diagonal entry in the Gram matrix $\mathbf{G} = \mathbf{\Phi}^T\mathbf{\Phi}$, and $\mu_{\frac{1}{2}}(\mathbf{G})$ denotes the smallest number m such that some m off-diagonal elements in a single row (or column) of $\mathbf{G}$ have absolute sum being at least $\frac{1}{2}$.

D. L. Donoho *et al.* [22] also showed the conditions for the uniqueness and equivalence

when the data vector ($\mathbf{y}$ in Eq. 2.11) was corrupt by noise.

Besides the above studies where the bound was established for absolute guarantees, recent studies have shown that, even with much looser bound, the equivalence between the l_1-norm regularization in Eq. 2.11 and the l_0-norm regularization in Eq. 2.12 still holds with overwhelming probability [15]. This indeed has led to a very promising new field, *compressive sampling* [12], which is also known as *nonlinear* sampling theorem in contrast to the conventional Nyquist−Shannon sampling theorem.

2.2.2 Optimization methods

There has been much research effort devoted to developing efficient algorithms for solving the optimizations in Eqs. 2.8∼2.10. In fact, if the function $f(\mathbf{w})$ is convex with respect to $\mathbf{w}$, all those three optimizations are convex since the l_1-norm $\|\mathbf{w}\|_1$ is convex. However, even when the optimizations are convex, how to efficiently solving them is not a trivial task. The main difficulty lies in the fact that $\|\mathbf{w}\|_1$ is not differentiable at $w_j = 0$ ($j = 1, 2, ..., M$). Someone may argue to use subgradient method [10] for solving the optimizations, but first-order gradient descent methods generally converge very slowly. In the following, we will review the existing methods for solving the optimizations in Eqs. 2.8∼2.10. They are divided into three categories of approaches: active set methods, entire regularization path methods, and interior point methods. We also developed some new algorithms for solving the optimization in Eq. 2.8, and they will be described in Section 3.1.2 in Chapter 3.

Active set methods

The active set methods aim to solve the optimization in Eq. 2.8 with a given regularization parameter λ'. The active set methods are able to not only solve l_1-norm regularized least square problems but also l_1-norm regularized other problems (such as logistic regression).

The main idea of active set methods is to manage an active set that are allowed to be nonzero and an inactive set that are fixed to be zeros. The existing active set methods for solving the optimization in Eq. 2.8 are the grafting algorithm [60] and the feature-sign search algorithm [42].

The grating algorithm [60] initialize the solution to be all zeros and the active set to be empty (inactive set to be full). Then, in each step, the algorithm moves a variable, say w_k, from the inactive set to the active set where the index k is chosen by $k = \arg\max_i |\frac{\partial f}{\partial w_i}|$. The sign of the new variable w_k is also determined to be $-\text{sign}(\frac{\partial f}{\partial w_k})$, and $|w_k|$ in the objective function in Eq. 2.8 becomes $-\text{sign}(\frac{\partial f}{\partial w_k})w_k$. Since the resulting objective function has no absolute function, the algorithm then uses general optimization techniques to solve the optimization with respect to the variables in the active set. The algorithm repeats to introduce a new variable to the active set until the partial derivative of $f(\mathbf{w})$ with respective to each variable in the inactive set has absolute magnitude less than λ'.

The feature-sign algorithm [42] is indeed one step further from the grafting algorithm. As you may have noticed, in grafting, before the optimization is performed with respect to all the variables in the active set, the sign of each variable was predetermined. However, the signs in the optimization result are not necessary to be consistent with the predetermined signs. The feature-sign algorithm provides a scheme for reconciliating the difference by performing a line search between the optimization result and the result before the optimization (or the result in the previous step). The line search procedure looks at all the points where any variable changes its sign and computes the objective function at all those points, then the algorithm picks the point with the smallest objective function as the new estimate. The algorithm repeats until all the variables meet the optimality conditions: for each nonzero element, its partial derivative must be equal to zero (or its magnitude is less than a threshold); for each zero element, the magnitude of its partial derivative of $f(\mathbf{w})$ must be no larger than λ'.

The active set methods often work well when the active set at optimum are relatively small. This could be the case when the sparsity regularization parameter λ' is a big number and the optimal solution is extremely sparse. The extreme case is, when λ' is big enough so that the optimal solution is all zeros, the active set methods are able to find their optimal solution in one step. However, when the dimension of $\mathbf{w}$ is high and its optimal solution is moderate dense (say $20\% \sim 50\%$ of elements are nonzero), the active set methods can be computationally costly.

Entire regularization path methods

This category of optimization methods includes the homotopy method [59] and the modified least angle regression (LARS) method [25]. Both of them attempt to solve the optimization in Eq. 2.9 with all possible values of β' (and thus an entire regularization path) when the objective function $f(\mathbf{w})$ being the square errors of a linear model, namely

$$\begin{aligned} \mathbf{w}^* &= \arg\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{y} - \mathbf{\Phi}\mathbf{w}\|_2^2 \\ &\text{subject to : } \|\mathbf{w}\|_1 \leq \beta', \end{aligned} \tag{2.16}$$

where $\|\cdot\|_2$ denotes l_2-norm, $\mathbf{\Phi}$ is an $N \times M$ design matrix, $\mathbf{y}$ is an $N \times 1$ data vector, $\mathbf{w}$ is an $M \times 1$ optimization variable. For the modified LARS algorithm, the design matrix is normalized so that its columns are zero-mean and unit power.

Although the original homotopy method was formulated for the optimization in the form of Eq. 2.16 [59], it is easier to explain using its equivalent form in Eq. 2.8 [53], namely

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{y} - \mathbf{\Phi}\mathbf{w}\|_2^2 + \lambda'\|\mathbf{w}\|_1. \tag{2.17}$$

The homotopy algorithm manages an active set $J = \{j : w_j^* \neq 0\}$, an inactive set $I = \{i : w_i^* = 0\}$, and the sign vector $\mathbf{s}$ of $\mathbf{w}^*$ (in this sense, the homotopy algorithm is indeed also

an active set method). It starts with $\lambda' = \max_i |(\mathbf{\Phi}^T \mathbf{y})_i|$ where the optimal solution $\mathbf{w}^* = \mathbf{0}$, $I = [1; 2; ...; M]$ with M being the dimension of $\mathbf{w}$, J is empty, and $\mathbf{s} = [0; 0; ...; 0]$. Then the algorithm begins to decrease λ' while checking the optimality conditions

$$\mathbf{w}_J^* = [(\mathbf{\Phi}^T \mathbf{\Phi})_{JJ}]^{-1}[(\mathbf{\Phi}^T y)_J - \lambda' \mathbf{s}_J], \tag{2.18}$$

$$\text{and for } i \in I \qquad |(\mathbf{\Phi}^T \mathbf{\Phi} \mathbf{w}^* - \mathbf{\Phi}^T y)_i| \leq \lambda'. \tag{2.19}$$

where $(\mathbf{\Phi}^T \mathbf{\Phi})_{JJ}$ denotes the submatrix of $(\mathbf{\Phi}^T \mathbf{\Phi})$ with $[(\mathbf{\Phi}^T \mathbf{\Phi})_{JJ}]_{mn} = (\mathbf{\Phi}^T \mathbf{\Phi})_{J(m),J(n)}$, $\mathbf{w}_J^*$ is a vector such that $(\mathbf{w}_J^*)_m = \mathbf{w}_{J(m)}^*$, and the similar notation is also applied to $\mathbf{s}_J$. As the λ' decreases,

1. if the algorithm sees a sign change in $\mathbf{w}_J^*$ (say $w_{\hat{j}}^*$ has the sign change), then the corresponding index ($\hat{j}$) is moved from J to I;

2. if the algorithm sees a violation of the derivative bounds in Eq. 2.19 (say $|(\mathbf{\Phi}^T \mathbf{\Phi} \mathbf{w}^* - \mathbf{\Phi}^T y)_{\hat{i}}| > \lambda'$), then the corresponding index ($\hat{i}$) is moved from I to J.

The first index (say $\hat{i}_1$) to enter J is the one with largest magnitude in $\mathbf{\Phi}^T \mathbf{y}$ (namely $\hat{i}_1 = \arg\max_i |(\mathbf{\Phi}^T \mathbf{y})_i|$) because a slight decrease of λ' from $\max_i |(\mathbf{\Phi}^T \mathbf{y})_i|$ will cause violating the inequality in Eq. 2.19 for $i = \hat{i}_1$. As a result, after a slight decrease of λ', $J = \{j : j = \hat{i}_1\}$, $I = \{i : i = 1, 2, ..., M\} \setminus J$, $w_{\hat{i}_1}^*$ is computed using the equation in Eq. 2.18, and $s_{\hat{i}_1} = \text{sign}(w_{\hat{i}_1}^*)$. Now, we can imagine that, there must be some interval for λ' to further decrease before either the sign change of $w_{\hat{i}_1}^*$ or any inequalities in Eq. 2.19 to be violated. During this interval, the optimal solution can be computed analytically by Eq. 2.18 (with elements in I being zeros), and it is linear with respect to λ'. The λ' is gradually decreased to 0 while the operations in 1 and 2 are continuously performed. As a result, the optimal solution $\mathbf{w}^*$ is piece-wise linear with respect to the regularization parameter λ'.

The modified LARS algorithm is developed based on the optimization in Eq. 2.16. The

algorithm is parallel to the homotopy method but from a very distinct viewpoint. The *original* LARS algorithm proceeds in the following way. The algorithm initializes $\mathbf{w}^* = \mathbf{0}$ and picks the column in $\mathbf{\Phi}$ (say $\mathbf{\Phi}_{i_1}$) that has the largest correlation in magnitude with the residual. The residual is defined as $\boldsymbol{\mu} = \mathbf{y} - \mathbf{\Phi}\mathbf{w}^*$. As such, $i_1 = \arg\max_i |(\mathbf{\Phi}^T\mathbf{y})_i|$ since $\mathbf{w}^* = \mathbf{0}$. Then, $\mathbf{w}^*$ is updated with $\mathbf{w}^* = \gamma_1 s_{i_1}\mathbf{\Phi}_{i_1}$ where s_{i_1} being the sign of $(\mathbf{\Phi}^T\mathbf{y})_{i_1}$, and γ_1 grows positively from 0 until some other column of $\mathbf{\Phi}$ (say $\mathbf{\Phi}_{i_2}$) has as much correlation in magnitude with the current residual [namely, $|\mathbf{\Phi}_{i_2}^T(\mathbf{y}-\mathbf{\Phi}\mathbf{w}^*)| = |\mathbf{\Phi}_{i_1}^T(\mathbf{y}-\mathbf{\Phi}\mathbf{w}^*)|$]. Then, $\mathbf{w}^*$ is updated with

$$\mathbf{w}^* \leftarrow \mathbf{w}^* + \gamma_2\mathbf{u}_1 \tag{2.20}$$

where $\mathbf{u}_1$ is a direction equiangular between $s_{i_1}\mathbf{\Phi}_{i_1}$ and $s_{i_2}\mathbf{\Phi}_{i_2}$ with s_{i_2} being the sign of $\mathbf{\Phi}_{i_2}^T(\mathbf{y} - \mathbf{\Phi}\mathbf{w}^*)$. Similarly, γ_2 grows positively from 0 until some other column of $\mathbf{\Phi}$ (say $\mathbf{\Phi}_{i_3}$) has as much correlation in magnitude with the current residual as previously selected columns do [namely, $|\mathbf{\Phi}_{i_3}^T(\mathbf{y} - \mathbf{\Phi}\mathbf{w}^*)| = |\mathbf{\Phi}_{i_2}^T(\mathbf{y} - \mathbf{\Phi}\mathbf{w}^*)| = |\mathbf{\Phi}_{i_1}^T(\mathbf{y} - \mathbf{\Phi}\mathbf{w}^*)|$]. So on and so forth. Now the above LARS algorithm can be modified for solving the optimization in Eq. 2.16. The modification is, whenever there is a sign change (say on $w_{i_m}^*$, and it occurs when γ_k is growing positively from zero), the algorithm takes the step size (γ_k) at that point and drops i_m out of the current set $\{i_1, i_2, ...i_k\}$. It has been shown in [25] that the modified LARS algorithm finds the solutions of Eq. 2.16 in the entire regularization path.

One attractive advantage of the entire regularization path methods is that, although they provide the solutions to all possible (infinitely many) values of λ' (or β'), their computational cost is at the same order as solving the ordinary least square problem [25] without regularization or the optimization with single λ' (or β')[53]. Both entire regularization path methods are also able to accept a warm start and do not have to always start from scratch that $\mathbf{w} = 0$. Furthermore, both algorithms can proceed in a reverse direction which starts with the ordinary least squares solution and heads to the zero solution, though the reverse

direction is often more costly computationally.

Interior point methods

We have reviewed both the active set methods and entire regularization path methods for solving the optimizations in Eqs. 2.8$\sim$2.10. All those methods are specialized for solving l_1-norm regularized problems. However, when $f(\mathbf{w})$ in the optimizations is convex, they can also be solved by standard convex optimization algorithms. Among them, interior point methods are popular choices. The advantage of interior point methods is not only their nice theoretical framework (like polynomial time) but also their efficiency in practice. It has been a *happy* puzzle to optimization practitioners that, when interior point methods are combined with Newton's method, a convex optimization problem can often be solved accurately in a couple of tens of iterations regardless of the size of problems. In the following, we review some of the interior point methods for solving the optimization in Eqs. 2.8$\sim$2.10.

The work in [14] shows some interior point methods for solving l_1-norm regularized problems in the form of Eq.2.10 with $f(\mathbf{w})$ being the squared error of a linear model. In particular, when the data vector is noiseless, the optimization can be solved via linear programming using a primal-dual interior point method; when the data vector is corrupted by noise, the optimization can be solved via second-order cone programming (SCOP) using log-barrier interior point method. Since we are more interested in the case of noisy data, let's look at the latter optimization in more details. First let's recap the optimization problem in Eq. 2.10 with $f(\mathbf{w})$ being the squared error of a linear model

$$\begin{aligned} \mathbf{w}^* &= \arg\min_{\mathbf{w}} \|\mathbf{w}\|_1 \\ \text{subject to}: \ & \|\mathbf{y} - \mathbf{\Phi}\mathbf{w}\|_2 \leq c \end{aligned} \tag{2.21}$$

where $\|\cdot\|_2$ denotes l_2-norm, $\mathbf{\Phi}$ is an $N \times M$ design matrix, $\mathbf{y}$ is an $N \times 1$ data vector, $\mathbf{w}$ is an $M \times 1$ optimization variable, and c is some nonnegative constant. The above optimization can be transformed into

$$\begin{aligned} \mathbf{w}^*, \mathbf{u}^* &= \arg\min_{\mathbf{w},\mathbf{u}} \sum_i u_i \\ \text{subject to : } & \|\mathbf{y} - \mathbf{\Phi}\mathbf{w}\|_2 \leq c \\ & w_i - u_i \leq 0, -w_i - u_i \leq 0, \quad i = 1, 2, ..., M. \end{aligned} \tag{2.22}$$

Note that $u_i \geq 0, i = 1, 2, ..., M$ is automatically included in the last line of constraints (since $u_i \geq w_i$ and $u_i \geq -w_i$). Then, the log-barrier interior point method solves the above optimization by solving the following unconstrained optimization

$$\mathbf{w}^*(C), \mathbf{u}^*(C) = \arg\min_{\mathbf{w},\mathbf{u}} \sum_i u_i - \frac{1}{C}\{\log(c - \|\mathbf{y} - \mathbf{\Phi}\mathbf{w}\|_2) + \sum_i [\log(u_i - w_i) + \log(u_i + w_i)]\} \tag{2.23}$$

where $C > 0$ is a centering parameter. The algorithm of the log-barrier interior point method consists of outer loops and inner loops: the outer loops gradually increase the C parameter from a small number to a big number (for example $C \leftarrow 10C$); given a C parameter, inner loops solve the optimization in Eq. 2.23, an unconstrained convex optimization problem, using Newton's method. The algorithm may stop after C reaches some threshold. As such, while the constraints in Eq. 2.22 are always respected (otherwise the objective function would become $+\infty$), the log-barrier terms have little effect on the objective function when C becomes a large number.

S.-J. Kim *et al.* [36] also proposed a log-barrier interior point method, but it was devised to solve the optimization in the form of Eq. 2.17. The proposed method first transformed

the optimization in Eq. 2.17 into the following form

$$\begin{aligned} \mathbf{w}^*, \mathbf{u}^* &= \arg\min_{\mathbf{w},\mathbf{u}} \frac{1}{2}\|\mathbf{y} - \mathbf{\Phi}\mathbf{w}\|_2 + \sum_i u_i \\ \text{subject to : } & w_i - u_i \leq 0, -w_i - u_i \leq 0, \quad i = 1, 2, ..., M. \end{aligned} \tag{2.24}$$

Then, the log-barrier interior point method solves the following optimization problems

$$\mathbf{w}^*, \mathbf{u}^* = \arg\min_{\mathbf{w},\mathbf{u}} \frac{t}{2}\|\mathbf{y} - \mathbf{\Phi}\mathbf{w}\|_2 + t\sum_i u_i - \sum_i [\log(u_i - w_i) + \log(u_i + w_i)] \tag{2.25}$$

with gradually increasing the centering parameter t. Again, given a t parameter, the above optimization is an unconstrained convex optimization problem and it can be solved efficiently by Newton's method. In contrast to the standard procedure in a log-barrier interior point method, which has outer loops and inner loops and inner loops are Newton's method, the log-barrier interior point method proposed by S.-J. Kim *et al.* erased the boundary between the outer loops and inner loops by adaptively estimating t according to the remaining duality gap given the current solution of $\mathbf{w}$ and $\mathbf{u}$. By doing so, the optimization in Eq. 2.25 with a given t does not have to be solved very accurately (since t will be re-evaluated). This enables a so-called *truncated Newton's method* for solving Eq. 2.25 using a preconditioned conjugate gradient (PCG) descent. Because PCG only involves matrix-vector multiplication but not matrix inversion (as in Newton's method), the proposed algorithm is expected to be able to solve very-large scale problems.

2.2.3 Sparsity regularization parameters

In the optimizations in Eq. 2.8$\sim$2.10, there are sparsity regularization parameters, λ', β' and γ', which control the sparseness of solutions. We only concern one of those parameters since the three optimizations are equivalent. In some rare cases, one of the three regu-

larization parameters may be known *a priori* (for example, under an unrealistic noiseless assumption, the optimization in Eq. 2.11 is corresponding to $f(\mathbf{w}) = \|\mathbf{y} - \mathbf{\Phi}\mathbf{w}\|_2$ and $\gamma' = 0$). However, in most cases, none of them is known *a priori*. How to determine (one of) the regularization parameters is critical for deriving appropriate solutions since it controls the sparseness of solutions. Unfortunately, although the regularization parameters are so important for deriving sparse solutions, there is surprisingly little research effort that has been devoted to studying how to determine their settings.

There are two existing methods for determining the regularization parameters. One is heuristic approach introduced by Chen *et al.* [18] for determining the λ' in Eq. 2.17 using the following equation

$$\lambda' = \sigma\sqrt{2\log(M)} \tag{2.26}$$

where σ is the noise level of $\mathbf{y}$ (or $\mathbf{y} = \mathbf{\Phi}\mathbf{w} + \sigma\mathbf{n}$ with $\mathbf{n}$ being zero-mean Gaussian with unit variance), and M is the dimension of $\mathbf{w}$. The above setting has been shown to be near-optimal when the columns of $\mathbf{\Phi}$ are orthonormal. The other method for finding the regularization parameters is cross-validation. Using the form in Eq. 2.16 as an example, cross-validation approach divides the data $\mathbf{y}$ and $\mathbf{\Phi}$ into two sets, training set ($\mathbf{y}_1$ and $\mathbf{\Phi}_1$) and testing set ($\mathbf{y}_2$ and $\mathbf{\Phi}_2$). Then, it solves the following optimization with many possible (if not all) values of β':

$$\begin{aligned} \mathbf{w}^*(\beta') &= \arg\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{y}_1 - \mathbf{\Phi}_1\mathbf{w}\|_2^2 \\ \text{subject to : } & \|\mathbf{w}\|_1 \leq \beta', \end{aligned} \tag{2.27}$$

and pick the β' so that $\mathbf{w}^*(\beta')$ yields the smallest value on $\|\mathbf{y}_2 - \mathbf{\Phi}_2\mathbf{w}^*(\beta')\|_2^2$. To solve the above optimization (Eq. 2.27) with all values of β', the entire regularization path methods are attractive choices.

Unfortunately, both the heuristic approach and the cross-validation approach have their

obvious drawback. For the heuristic approach, the λ' in Eq. 2.26 is even not unit-consistent: if the unit of $\mathbf{y}$ is u_1 and the unit of $\mathbf{w}$ is u_2, then, the unit of $\|\mathbf{y} - \mathbf{\Phi}\mathbf{w}\|_2^2$ is u_1^2 while the unit of $\lambda'\|\mathbf{w}\|_1$ is $u_1 u_2$. Furthermore, the orthonormal assumption on the columns of $\mathbf{\Phi}$ is often not true. As for the cross-validation approach, it requires excessive data samples which are often not available. Moreover, as data become more and more heterogenous these days, there may be two or more distinct groups of weights, for example $\mathbf{w}_1$ and $\mathbf{w}_2$, and the optimization for finding sparse weights becomes

$$\mathbf{w}_1^*, \mathbf{w}_2^* = \arg\min_{\mathbf{w}_1, \mathbf{w}_2} f(\mathbf{w}_1, \mathbf{w}_2) + \lambda_1'\|\mathbf{w}_1\|_1 + \lambda_2'\|\mathbf{w}_2\|_1 \tag{2.28}$$

where it is natural to assume that $\mathbf{w}_1$ and $\mathbf{w}_2$ have different sparsity regularization parameters. However, it is very computationally expensive when there are two (or more) parameters to be cross-validated.

The problems of the heuristic approach and the cross-validation approach originates from the fact that neither of them defines any quantity regarding the optimal sparseness to optimize. In contrast, our proposal in this thesis, l_1*-norm sparse Bayesian learning*, is to put an l_1-norm regularized optimization in a Bayesian framework and explicitly define *the optimal sparseness* in a Bayesian sense. Using Eq. 2.17 as an example, we show how the optimal sparsity regularization parameter λ' can be inferred directly from data in a Bayesian framework. Furthermore, with the Bayesain inference, we can generalize the optimization in Eq. 2.17 into

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{y} - \mathbf{\Phi}\mathbf{w}\|_2^2 + \sum_j \lambda_j'|w_j| \tag{2.29}$$

where each weight w_j is associated with an *independent* sparsity regularization parameter λ_j', and the regularization parameters are to be inferred in the Bayesian framework. As such, the optimal sparseness of solutions is fully defined via the optimal sparsity regularization parameters, and it is derived by learning directly from data. This is why we call our

approach *sparse learning*, which is very different from previous approaches where there is a single sparsity regularization parameter and it is set in ad-hoc manners (like the heuristic approach and the cross-validation approach). We will describe the Bayesian framework for Eqs.2.17 and 2.29 in details in Chapter 3.

2.3 Sparse Bayesian learning with independent Gaussian priors

Our l_1-norm Bayesian sparse learning was inspired by an l_2-norm Bayesian sparse learning framework [52] [28] [66], which is to learn the optimal independent l_2-norm regularization parameters $\boldsymbol{\alpha} = [\alpha_1; \alpha_2; ...; \alpha_M]$ for the following optimization

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} f(\mathbf{w}) + \frac{1}{2}\sum_{j}^{M} \alpha_j' w_j^2, \tag{2.30}$$

where $f(\mathbf{w})$ is a log-likelihood function parameterized by $\mathbf{w}$, and $\mathbf{w}$ is an $M \times 1$ vector desired to be sparse. When $f(\mathbf{w})$ is the squared error of a linear system, the above optimization becomes

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{y} - \boldsymbol{\Phi}\mathbf{w}\|_2^2 + \frac{1}{2}\sum_{j} \alpha_j' w_j^2 \tag{2.31}$$

where $\|\cdot\|_2$ denotes l_2-norm, $\boldsymbol{\Phi}$ is an $N \times M$ design matrix, $\mathbf{y}$ is an $N \times 1$ data vector. It has been shown that [66], when the l_2-norm regularization parameters $\boldsymbol{\alpha}$ are learned in a Bayesian framework, some of them would become infinity and thus their associated weights would become zeros. As a result, the Bayesian approach has a mechanism of *automatic relevance determination* for excluding the irrelevant columns in $\boldsymbol{\Phi}$ (by setting their associated weights to be zero), and it was named as *relevance vector machine* (RVM)

when the designed matrix $\mathbf{\Phi}$ was constructed by kernels. We briefly re-formulate the RVM approach for Eq. 2.31 in the following. Our formulation is from the l_2-norm regularization point of view, and it is slightly different from the original formulation in [66]. Nevertheless, the two formulations share the same objective in optimization.

2.3.1 Bayesian framework

The probabilistic model for Eq. 2.31 assumes that the data vector $\mathbf{y}$ is corrupted by zero-mean I.I.D. Gaussian noise, namely,

$$P(\mathbf{y}|\mathbf{w}, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{N/2}} \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{y} - \mathbf{\Phi}\mathbf{w}\|^2\right), \tag{2.32}$$

where σ^2 describes noise level, and the weights $\mathbf{w}$ are governed by an independent zero-mean Gaussian, namely

$$P(\mathbf{w}|\boldsymbol{\alpha}) = \prod_{j=1}^{M} (\frac{\alpha_j}{2\pi})^{1/2} \exp\{-\frac{\alpha_j}{2} w_j^2\} \tag{2.33}$$

where α_j is the hyperparameter for the Gaussian prior on w_j, $j = 1, 2, ..., M$. The parameters of the model are σ^2 and $\boldsymbol{\alpha}$. Their optimal setting can be found by maximizing the posterior $P(\sigma^2, \boldsymbol{\alpha}|\mathbf{y})$, which can be written as

$$\begin{aligned} \sigma^{2^*}, \boldsymbol{\alpha}^* &= \arg\max_{\sigma^2, \boldsymbol{\alpha}} P(\sigma^2, \boldsymbol{\alpha}|\mathbf{y}) \\ &= \arg\max_{\sigma^2, \boldsymbol{\alpha}} \frac{P(\mathbf{y}|\sigma^2, \boldsymbol{\alpha}) P(\sigma^2) P(\boldsymbol{\alpha})}{P(\mathbf{y})} \end{aligned} \tag{2.34}$$

according to the Bayes' rule. Now, if we assume the distributions, $P(\sigma^2)$ and $P(\boldsymbol{\alpha})$, are flat, then optimizing the posterior $P(\sigma^2, \boldsymbol{\alpha}|\mathbf{y})$ is equivalent to maximizing the *marginal*

likelihood $P(\mathbf{y}|\sigma^2, \boldsymbol{\alpha})$, which is marginalized with respect to $\mathbf{w}$, namely,

$$P(\mathbf{y}|\sigma^2, \boldsymbol{\alpha}) = \int_{\mathbf{w}} \mathbf{dw} P(\mathbf{y}|\mathbf{w}, \sigma^2) P(\mathbf{w}|\boldsymbol{\alpha}). \tag{2.35}$$

Since both $P(\mathbf{y}|\mathbf{w}, \sigma^2)$ and $P(\mathbf{w}|\boldsymbol{\alpha})$ are Gaussian as respectively shown in Eq. 2.32 and Eq. 2.33, the marginal likelihood can be computed in a close form, as shown in the following

$$P(\mathbf{y}|\sigma^2, \boldsymbol{\alpha}) = (2\pi)^{-N/2} \det(\sigma^2\mathbf{I} + \mathbf{\Phi}\Lambda^{-1}\mathbf{\Phi}^T)^{-1/2} \exp\{-\frac{1}{2}\mathbf{y}^T(\sigma^2\mathbf{I} + \mathbf{\Phi}\Lambda^{-1}\mathbf{\Phi}^T)^{-1}\mathbf{y}\} \tag{2.36}$$

where $\mathbf{\Lambda}$ is a diagonal matrix whose diagonal is $\boldsymbol{\alpha}$ (namely, $\Lambda_{jj} = \alpha_j$), and $\mathbf{I}$ is an $N \times N$ identity matrix.

2.3.2 Update rule

The optimal setting of σ^2 and $\boldsymbol{\alpha}$ can be found by maximizing the marginal likelihood in Eq. 2.36 (or more often, maximizing the logarithm of the marginal likelihood). Unfortunately, the optimization is not convex and it can be solved by standard algorithms (like interior point methods for dealing with the nonnegative constraint on both σ^2 and $\boldsymbol{\alpha}$). Therefore, M. Tipping [66] developed a specialized algorithm for maximizing the marginal likelihood. The algorithm was derived by solving fix-point equations that were obtained by setting the partial derives of the logarithm of the marginal likelihood with respect to σ^2 and $\boldsymbol{\alpha}$ to zero. The resulting update rule is

$$\gamma_i \equiv 1 - \alpha_i \Sigma_{ii} \tag{2.37}$$

$$\alpha_i \longleftarrow \frac{\gamma_i}{\mu_i^2} \tag{2.38}$$

$$\sigma^2 \longleftarrow \frac{\|\mathbf{t} - \mathbf{\Phi}\boldsymbol{\mu}\|_2^2}{N - \sum_i \gamma_i} \tag{2.39}$$

where

$$\boldsymbol{\Sigma} = (\sigma^{-2}\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \boldsymbol{\Lambda})^{-1} \tag{2.40}$$

$$\boldsymbol{\mu} = \sigma^{-2}\boldsymbol{\Sigma}\boldsymbol{\Phi}^T\mathbf{t} \tag{2.41}$$

and Σ_{ii} is the i^{th} diagonal element of $\boldsymbol{\Sigma}$. The above update rule has been demonstrated to perform well in practice, though it does not guarantee a monotonic increase of the marginal likelihood along iterations.

2.4 Why l_1-norm sparse Bayesian learning

As we have seen, finding sparse solution is an important tool for both *efficiently* and *meaningfully* representing today's exponentially growing data. Unfortunately, although there has been much research effort devoted to studying how to find sparse solutions in the last decade, how to find *optimally* sparse solutions is a relatively untouched issue. Our proposal in this thesis, *l_1-norm sparse Bayesian learning*, aims to address this fundamental issue.

Compared to the conventional l_1-norm sparsity regularization which has a single regularization parameter determined in ad-hoc manners (like heuristic approach or cross-validation), the l_1-norm sparse Bayesian learning approach associates each variable desired to be sparse with an independent regularization parameter and finds the optimally sparse solutions by learning the optimal regularization parameters in a Bayesian framework. This provides a natural way for defining the optimal sparseness in a Bayesian sense. Most importantly, as we will illustrate in Chapter 3, the l_1-norm sparse Bayesian learning is much more powerful than the conventional l_1-norm sparsity regularization in accurately resolving underlying sparse structures in noisy data.

The l_1-norm sparse Bayesian learning is also advantageous compared to the l_2-norm

sparse Bayesian learning (known as relevant vector machine). First, l_1-norm regularization is known to be much stronger than l_2-norm regularization for deriving sparse solutions. As illustrated by the simulations in Sections 3.1.5 in Chapter 3, the l_1-norm sparse Bayesian learning provides better performance in accurately resolving true sparse solutions than its l_2-norm counterpart. Second, l_1-norm sparse Bayesian learning bridges the two groups of research in finding sparse solutions, l_1-norm sparsity regularization and sparse Bayesian learning, which originally seemed to be two very different approaches.

Chapter 3

l_1-norm sparse Bayesian learning

This chapter describes how to learn the *optimal* l_1-norm regularization parameters $\boldsymbol{\lambda}' = [\lambda'_1, \lambda'_2, ..., \lambda'_M]^T$ of the following problem

$$\min_{\mathbf{w}} \sum_{i=1}^{N} L(\mathbf{x}_i, \mathbf{w}) + \sum_{j=1}^{M} \lambda'_j |w_j| \tag{3.1}$$

where $\{\mathbf{x}_i\}_{i=1}^N$ are N data samples, $\sum_{i=1}^N L(\mathbf{x}_i, \mathbf{w})$ is a data log-likelihood term parameterized by an $M \times 1$ weight vector $\mathbf{w}$, $|\cdot|$ denotes absolute value, and the second term $\sum_{j=1}^M \lambda'_j |w_j|$ is l_1-norm penalty for encouraging sparse solutions of $\mathbf{w}$. So, given the data $\{\mathbf{x}_i\}_{i=1}^N$ and knowing that the weight vector $\mathbf{w}$ is sparse, our goal here is to discover the *optimally* sparse solution of $\mathbf{w}$ without any prior knowledge about which subset of elements in $\mathbf{w}$ should be zeros. Our proposed approach, *l_1-norm sparse Bayesian learning*, is to define the *optimal sparseness* of solutions via the optimal sparsity regularization parameters $\boldsymbol{\lambda}'$ in a Bayesian framework and infer it by *learning* directly from data. This is why we call our method *sparse learning*, which is very different from convention methods which set the regularization parameters in ad-hoc manners (as reviewed in Section 2.2.3 in Chapter 2).

We describe the l_1-norm sparse Bayesian learning in details for an ordinary least squares

problem where the data log-likelihood term in Eq. 3.1 is the square errors of a linear model and the optimization takes the following form

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{y} - \mathbf{\Phi}\mathbf{w}\|_2^2 + \sum_{j=1}^{M} \lambda'_j |w_j| \tag{3.2}$$

with $\mathbf{y}$ being $N \times 1$ data vector, $\mathbf{\Phi}$ being $N \times M$ design matrix, and $\|\cdot\|_2$ denoting l_2-norm. We also show that the Bayesain framework can be easily adapted for finding the optimal scalar regularization parameter λ' in the conventional l_1-norm regularized least squares problem, namely

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{y} - \mathbf{\Phi}\mathbf{w}\|_2^2 + \lambda' \sum_{j=1}^{M} |w_j|. \tag{3.3}$$

The Bayesian framework indeed provides a universal approach for finding the optimal l_1-norm regularization parameters for a variety of problems. We show a similar Bayesian framework for a *nonnegative* least squares problem,

$$\begin{aligned} &\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{y} - \mathbf{\Phi}\mathbf{w}\|_2^2 + \sum_{j=1}^{M} \lambda'_j w_j \\ &\text{subject to : } \mathbf{w} \geq 0 \end{aligned} \tag{3.4}$$

and discuss the applicability of the Bayesian approaches for other problems like logistic regression [43] [38] [62], fitting sparse Gaussian Markov network [6] [29] and general Markov networks [44][68].

The Bayesian treatment in this Chapter was originally inspired by the work of relevance vector machine (RVM) [66], where Bayesian framework was employed for learning the optimal l_2-norm regularization parameters. Here we are interested in the Bayesian framework for l_1-norm regularization since it is well-known that the l_1-norm is more powerful for sparsity regularization than l_2-norm. However, the upgrading from l_2-norm to the l_1-norm in the Bayesian framework comes with extra computational cost. We mentioned in Eqs. 2.35

and 2.36 in Chapter 2 that, for ordinary least squares, learning the optimal l_2-norm regularization parameters is to maximize a marginal likelihood which turned out to have a close form. However, for the l_1-norm regularization, the marginal likelihood is not integrable any more. In this chapter, we show how to maximize the marginal likelihood, which does not have a close form, via an Expectation-Maximization (EM) approach and introducing a variational scheme for evaluating the expectations in EM steps. The variational approximation in each EM step involves solving an l_1-norm regularized problem with the current estimate of regularization parameters. As such, the resulting l_1-norm sparse Bayesian learning may be algorithmically viewed as a re-weighted l_1-norm regularization [13] with the clear goal of maximizing the marginal likelihood.

3.1 l_1-norm sparse Bayesian learning for ordinary least squares

This Section describes in details the Bayesian framework for ordinary least squares with *independent* l_1-norm regularization (shown in Eq. 3.2), and later it is adapted for *uniform* l_1-norm regularization (shown in eq. 3.3).

3.1.1 Bayesian framework for independent l_1-norm regularization

In the probabilistic model for Eq.3.2, the data $\mathbf{y}$ are assumed to be coupled with additive I.I.D. zero-mean Gaussian noise, namely,

$$P(\mathbf{y}|\mathbf{w},\sigma^2) = \frac{1}{(2\pi\sigma^2)^{N/2}} \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{y}-\mathbf{\Phi}\mathbf{w}\|^2\right), \tag{3.5}$$

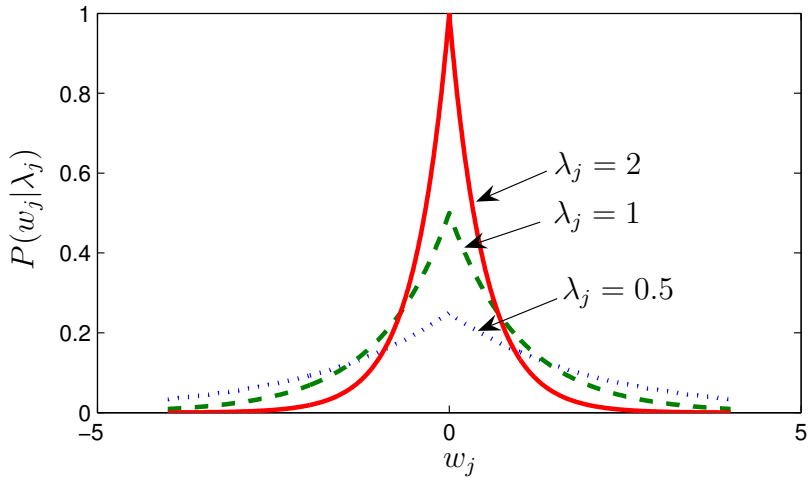


Figure 3.1: Laplacian distribution $P(w_j|\lambda_j) = \frac{\lambda_j}{2}\exp\{-\lambda_j|w_j|\}$ with $\lambda_j = 0.5, 1, 2$. The larger the λ_j, more concentrated the w_j is around zero.

and the prior on the weights is a factorized independent Laplacian distribution,

$$P(\mathbf{w}|\boldsymbol{\lambda}) = \prod_{j=1}^{M} \frac{\lambda_j}{2}\exp\{-\lambda_j|w_j|\}, \tag{3.6}$$

where $\boldsymbol{\lambda} = [\lambda_1; \lambda_2; ...; \lambda_M]$ and λ_j describes the slope of the Laplacian distribution on w_j, as shown in Fig. 3.1. It is easy to see that the optimization in Eq. 3.2 is basically a *maximum a posteriori* (MAP) estimation under the above probabilistic model with

$$\lambda'_j = \sigma^2\lambda_j \quad j = 1, 2, ..., M. \tag{3.7}$$

As such, finding the optimal regularization parameters $\boldsymbol{\lambda}'$ is to infer the optimal setting of σ^2 and $\boldsymbol{\lambda}$ given the observed data, $\mathbf{y}$ (and $\boldsymbol{\Phi}$). We choose to compute the optimal parameters, σ^2 and $\boldsymbol{\lambda}$, by maximizing the posterior distribution $P(\sigma^2, \boldsymbol{\lambda}|\mathbf{y})$. According to the Bayes' rule,

$$P(\sigma^2, \boldsymbol{\lambda}|\mathbf{y}) = \frac{P(\mathbf{y}|\sigma^2, \boldsymbol{\lambda})P(\sigma^2)P(\boldsymbol{\lambda})}{P(\mathbf{y})}. \tag{3.8}$$

Now, if the distributions, $P(\sigma^2)$ and $P(\boldsymbol{\lambda})$, are flat, maximizing the posterior $P(\sigma^2, \boldsymbol{\lambda}|\mathbf{y})$ is equivalent to maximizing the marginal likelihood:

$$\begin{aligned} P(\mathbf{y}|\boldsymbol{\lambda}, \sigma^2) &= \int_{-\infty}^{+\infty} d\mathbf{w}\, P(\mathbf{y}|\mathbf{w}, \sigma^2) P(\mathbf{w}|\boldsymbol{\lambda}) \\ &= \frac{\prod_{j=1}^{M} \lambda_j}{2^M (2\pi\sigma^2)^{N/2}} \int_{-\infty}^{+\infty} d\mathbf{w}\, \exp[-F(\mathbf{w})] \end{aligned} \tag{3.9}$$

where

$$F(\mathbf{w}) = \frac{1}{2\sigma^2}\|\mathbf{y} - \mathbf{\Phi}\mathbf{w}\|^2 + \sum_{j=1}^{M} \lambda_j |w_j|. \tag{3.10}$$

Unfortunately, this marginal likelihood can not be evaluated analytically. Our strategy is to maximize it via an Expectation-Maximization (EM) algorithm by treating $\mathbf{w}$ as hidden variables, σ^2 and λ as parameters. Then, the EM update rule is:

$$\lambda_j \longleftarrow \frac{1}{\int_{-\infty}^{+\infty} d\mathbf{w}\, |w_j| Q(\mathbf{w})} \quad j = 1, 2, ..., M, \tag{3.11}$$

$$\text{and} \quad \sigma^2 \longleftarrow \frac{1}{N} \int_{-\infty}^{+\infty} d\mathbf{w}\, \|\mathbf{y} - \mathbf{\Phi}\mathbf{w}\|^2 Q(\mathbf{w}), \tag{3.12}$$

where the expectations are taken over the distribution

$$Q(\mathbf{w}) = \frac{1}{\mathcal{Z}_w} \exp[-F(\mathbf{w})] \tag{3.13}$$

with normalization constant $\mathcal{Z}_w = \int_{-\infty}^{+\infty} d\mathbf{w}\, \exp[-F(\mathbf{w})]$. The EM procedure can be thought as iteratively re-estimating the optimal parameters (σ^2 and λ) from the current estimate of the weight statistics under $Q(\mathbf{w})$ distribution. Although the integrals in Eqs. 3.11 and 3.12 may be computed by Markov chain Monte Carlo (MCMC) based methods, they are expensive when the dimension of $\mathbf{w}$ is high. As a result, we seek to evaluate them via a variational approach, which approximates the distribution $Q(\mathbf{w})$ around its mode, $\mathbf{w}^{MP}$.

3.1.2 Optimization of l_1-norm regularized least squares

The mode of the distribution $Q(\mathbf{w})$, $\mathbf{w}^{MP}$, is defined as the $\mathbf{w}$ that maximizes the $Q(\mathbf{w})$, namely

$$\begin{aligned}
\mathbf{w}^{MP} &= \arg\min_{\mathbf{w}} F(\mathbf{w}) \\
&= \arg\min_{\mathbf{w}} \frac{1}{2\sigma^2}\|\mathbf{y}-\mathbf{\Phi}\mathbf{w}\|^2 + \sum_{j=1}^{M}\lambda_j|w_j| \qquad (3.14)\\
&= \arg\min_{\mathbf{w}} \frac{1}{2}\mathbf{w}^T\mathbf{A}\mathbf{w} + \mathbf{b}^T\mathbf{w} + \sum_{j=1}^{M}\lambda_j|w_j|, \qquad (3.15)
\end{aligned}$$

where $\mathbf{A} = \sigma^{-2}\mathbf{\Phi}^T\mathbf{\Phi}$, and $\mathbf{b} = -\sigma^{-2}\mathbf{\Phi}^T\mathbf{y}$. Note that this minimization is equivalent to the one in Eq. 3.2 with $\lambda'_j = \sigma^2\lambda_j, j = 1, 2, ..., M$.

The optimization in Eq. 3.14 (or Eq. 3.15) can be solved by different optimization approaches such as active set methods, entire regularization path methods, and interior point methods. We have reviewed those methods in Section 2.2.2 in Chapter 2. However, the minimization problem in Eq. 3.14 (or Eq. 3.15) is slightly different from the conventional form of l_1-norm regularized least squares problem (as shown in Eq. 3.3) in that the different weights are associated with different regularization parameters. This can be fixed by a simple variable transform, $z_i = \lambda_i w_i$, and then the optimization in Eq. 3.14 and Eq. 3.15 become

$$\begin{aligned}
\mathbf{z}^{MP} &= \arg\min_{\mathbf{z}} \frac{1}{2\sigma^2}\|\mathbf{y}-\tilde{\mathbf{\Phi}}\mathbf{z}\|^2 + \sum_{j=1}^{M}|z_j| \qquad (3.16)\\
&= \arg\min_{\mathbf{z}} \frac{1}{2}\mathbf{z}^T\tilde{\mathbf{A}}\mathbf{z} + \tilde{\mathbf{b}}^T\mathbf{z} + \sum_{j=1}^{M}|z_j|, \qquad (3.17)
\end{aligned}$$

where the j^{th} column of $\tilde{\mathbf{\Phi}}$ is $\tilde{\mathbf{\Phi}}_j = \frac{1}{\lambda_j}\mathbf{\Phi}_j$, $\tilde{A}_{ij} = \frac{A_{ij}}{\lambda_i\lambda_j}$, and $\tilde{b}_j = \frac{b_j}{\lambda_j}$ for $i, j = 1, 2, ..., M$. Now, the optimizations in Eq. 3.16 (or Eq. 3.17) is ready to be solved by the existing

optimization methods reviewed in Section 2.2.2 in Chapter 2.

Although the existing methods for solving an l_1-norm regularized least squares problem often provide reasonable performance, they can be improved. For the active set methods (the grafting algorithm and the feature-sign algorithm) often perform well when the optimal solution is very sparse, but their convergence often slows down significantly when the optimal solution become denser. And that's similar for entire regularization path methods, which would not be very efficient if it would not be allowed to stop at a very early point of the entire regularization path where solutions are very sparse. In contrast, the interior point methods are not sensitive to how sparse the optimal solutions are. However, a log-barrier interior point method is often slow because it has an inner loop and an outer loop. S.-J. Kim *et at.* [36] proposed to erase the boundary between inner loops and outer loops by adaptively estimating centering parameters. However, the estimating of centering parameters involves heuristics, which are not easy to interpret. Here, we propose four different approaches for solving the optimization in Eq. 3.15 with the hope that they provide better performance than those existing methods.

Our first approach for solving the optimization in Eq. 3.15 is to construct a series of auxiliary functions using a property of an absolute function and solve the optimization by minimizing those auxiliary functions. We will show that, this approach has no need of determining step size, it is very easy to implement, and it is guaranteed to monotonically converge to the global optimizer.

Our other three approaches are for solving nonnegative quadratic programming (NNQP) problems, and the optimization in Eq. 3.15 can be transformed into an NNQP problem as

$$\begin{aligned} \mathbf{w}^{+*}, \mathbf{w}^{-*} &= \arg\min_{\mathbf{w}^+, \mathbf{w}^-} \frac{1}{2}(\mathbf{w}^+ - \mathbf{w}^-)^T \mathbf{A}(\mathbf{w}^+ - \mathbf{w}^-) + \mathbf{b}^T(\mathbf{w}^+ - \mathbf{w}^-) + \sum_{j=1}^{M} \lambda_j (w_j^+ + w_j^-) \\ \text{subject to} \quad & w_j^+ \geq 0,\ w_j^- \geq 0 \quad j = 1, 2, ..., M, \end{aligned} \tag{3.18}$$

where $\mathbf{w}^{MP}$ can be computed using $\mathbf{w}^{MP} = \mathbf{w}^{+*} - \mathbf{w}^{-*}$ after the above optimization is solved. This formulation is based on the observation that, in the optimal solution, for each pair w_j^{+*} and w_j^{-*}, at least one of them is zero (so that $w_j^+ + w_j^- = |w_j|$). Otherwise, there exists a better solution which is derived by subtracting both of them with the smaller number of them. This is because the better solution does not change the terms in objective function that involve $w_j^+ - w_j^-$, but it yields a decrease in the term $\lambda_j(w_j^+ + w_j^-)$. The above transform is different from the previously proposed transform in [14] and [36], and its advantage is that it has only simple nonnegative constraints that are much easier to handle. In fact, the optimization in Eq. 3.18 can be further simplified as a standard nonnegative quadratic programming problem, namely

$$\begin{aligned} \mathbf{v}^* &= \arg\min_{\mathbf{v}} \frac{1}{2}\mathbf{v}^T\mathbf{H}\mathbf{v} + \mathbf{q}^T\mathbf{v} \\ \text{subject to:} \quad & v_j \geq 0 \quad j = 1, 2, ..., 2M, \end{aligned} \tag{3.19}$$

where $\mathbf{v} = [\mathbf{w}^+; \mathbf{w}^-]$ ($2M \times 1$ vector), $\mathbf{q} = [\mathbf{b} + \boldsymbol{\lambda}; -\mathbf{b} + \boldsymbol{\lambda}]$ ($2M \times 1$ vector), and $\mathbf{H} = \begin{bmatrix} \mathbf{A} & -\mathbf{A} \\ -\mathbf{A} & \mathbf{A} \end{bmatrix}$ ($2M \times 2M$ matrix).

Our three approaches for solving the NNQP problem in Eq. 3.19 are from very different perspective. Our first approach is based on constructing an auxiliary function of $\mathbf{v}^T\mathbf{H}\mathbf{v}$. We will show that this auxiliary function based approach leads to a multiplicative update algorithm, which is easy to implement without the need of determining step size and guaranteed to monotonically converge to the global optimizer. Our second approach is Merhotra predictor-corrector primal-dual interior method [69]. This method does not have inner-outer loops, and it is known to provide even better line search directions than Newton's method because it has an extra corrector step (while the corrector step costs very little extra computation). Our experiments show that this approach often converges to an

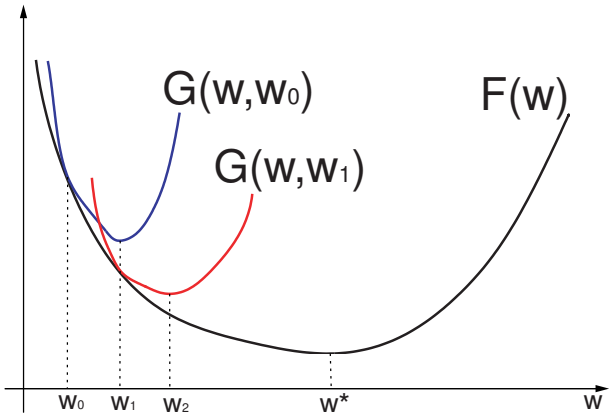


Figure 3.2: The iterative procedure of minimizing $F(\mathbf{w})$ via auxiliary functions $G(\mathbf{w}, \tilde{\mathbf{w}})$, with $\tilde{\mathbf{w}} = \mathbf{w}_0, \mathbf{w}_1, \mathbf{w}_2, ...$

extremely accurate solution (say relative dual gap being less than 10^{-14}) in less than 20 steps. Our third approach is projected gradient descent [8], an algorithm that is efficient when an optimization only involves coordinate-wise bound constraints. The projected gradient approach is also easy to implement with guaranteed monotonic convergence (through line search). Most importantly, it only involves matrix-vector multiplication but not matrix inversion, and it serves as a good candidate for developing on-line algorithms.

Auxiliary function (of an absolute function) based algorithm

We introduce here a method that solves the optimization problem in Eq. 3.15 by constructing auxiliary functions. Because of the concavity of a square-root function, $|w_j| = (w_j^2)^{1/2}$ is upper bounded as $|w_j| \leq |\tilde{w}_j| + \frac{1}{2|\tilde{w}_j|}(w_i^2 - \tilde{w}_j^2)$ for any $\tilde{w}_j$, and equality holds only when $w_j = \tilde{w}_j$. As a result, we construct the auxiliary function:

$$G(\mathbf{w}, \tilde{\mathbf{w}}) = \frac{1}{2}\mathbf{w}^T\mathbf{A}\mathbf{w} + \mathbf{b}^T\mathbf{w} + \sum_{j=1}^{M} \frac{\lambda_j}{2|\tilde{w}_j|} w_j^2 + \sum_{j=1}^{M} \frac{\lambda_j}{2}|\tilde{w}_j|, \tag{3.20}$$

which satisfies the two conditions: 1) $G(\tilde{\mathbf{w}}, \tilde{\mathbf{w}}) = F(\tilde{\mathbf{w}})$, and 2) $G(\mathbf{w}, \tilde{\mathbf{w}}) \geq F(\mathbf{w})$ for any $\mathbf{w}$. Then, the iterative update rule,

$$\tilde{\mathbf{w}} \longleftarrow \arg\min_{\mathbf{w}} G(\mathbf{w}, \tilde{\mathbf{w}}), \tag{3.21}$$

will converge to a local minimum of $F(\mathbf{w})$ [70], which is also the global minimum since $F(\mathbf{w})$ in Eq. 3.15 is convex. An example for illustrating the iterative scheme is shown in Fig. 3.2. At each iterative step, since the auxiliary function is a quadratic function, its optimal solution can be computed analytically:

$$\tilde{\mathbf{w}} \longleftarrow (\mathbf{A} + \mathbf{\Lambda})^{-1}\mathbf{b} \tag{3.22}$$

where $\mathbf{\Lambda} = \text{diag}([\lambda_1/|\tilde{w}_1|, \lambda_2/|\tilde{w}_2|, ..., \lambda_M/|\tilde{w}_M|])$. Because the columns in $\mathbf{\Phi}$ associated with zero solutions during the iterations can be pruned, the matrix inversion in Eq. 3.22 is performed on a gradually reduced matrix. Generally, the resulting algorithm for solving the optimization problem in Eq. 3.22 is easy to implement without the need of determining step size, has excellent convergence property, and is computationally efficient when the optimal solution is sparse.

Multiplicative update algorithm for NNQP

We developed a multiplicative update algorithm [62] for solving the NNQP problem in Eq. 3.19 by constructing a series of auxiliary functions using a scheme that we have illustrated in Fig. 3.2. Here, the auxiliary function is based on an upper bound function for $\mathbf{v}^T\mathbf{H}\mathbf{v}$. After the NNQP problem is solved, we can easily recover the optimal solution of the optimization in Eq. 3.15 by $w_i^* = v_i^* - v_{i+M}^*$ with $i = 1, 2, ..., M$.

The multiplicative algorithm first decompose the matrix $\mathbf{H}$ into its positive and negative

components such that $\mathbf{H} = \mathbf{H}^+ - \mathbf{H}^-$ where

$$H_{ij}^+ = \begin{cases} H_{ij} & \text{if} \quad H_{ij} > 0 \\ 0 & \text{if} \quad H_{ij} \leq 0 \end{cases} \quad H_{ij}^- = \begin{cases} 0 & \text{if} \quad H_{ij} \geq 0 \\ -H_{ij} & \text{if} \quad H_{ij} < 0 \end{cases} \tag{3.23}$$

In terms of the nonnegative matrices $\mathbf{H}^+$ and $\mathbf{H}^-$, the following is an auxiliary function that upper bounds Eq. 3.19 for all $\tilde{\mathbf{v}} > 0$ [62]:

$$\begin{aligned} g(\mathbf{v}, \tilde{\mathbf{v}}) &= \frac{1}{2} \sum_i \frac{(\mathbf{H}^+ \tilde{\mathbf{v}})_i}{\tilde{v}_i} v_i^2 - \frac{1}{2} \sum_{i,j} H_{ij}^- \tilde{v}_i \tilde{v}_j (1 + \ln \frac{v_i v_j}{\tilde{v}_i \tilde{v}_j}) + \mathbf{q}^T \mathbf{v} \\ &= \frac{1}{2} \sum_i \frac{(\mathbf{H}^+ \tilde{\mathbf{v}})_i}{\tilde{v}_i} v_i^2 - \sum_i (\mathbf{H}^- \tilde{\mathbf{v}})_i \tilde{v}_i \ln \frac{v_i}{\tilde{v}_i} - \frac{1}{2} \sum_{i,j} H_{ij}^- \tilde{v}_i \tilde{v}_j + \mathbf{q}^T \mathbf{v}, \end{aligned} \tag{3.24}$$

where the auxiliary function $g(\mathbf{v}, \tilde{\mathbf{v}})$ satisfies two conditions: 1) $g(\tilde{\mathbf{v}}, \tilde{\mathbf{v}}) = f(\tilde{\mathbf{v}})$ with $f(\mathbf{v}) = \frac{1}{2} \mathbf{v}^T \mathbf{H} \mathbf{v} + \mathbf{q}^T \mathbf{v}$ being the original objective function in Eq. 3.19; 2) $g(\mathbf{v}, \tilde{\mathbf{v}}) \geq f(\mathbf{v})$ for any $\mathbf{v}$. The first condition is easy to check, and proof for the second condition can be found in [63]. Now, the auxiliary function $g(\tilde{\mathbf{v}}, \tilde{\mathbf{v}})$ is separable coordinate-wise, and it can be optimized by solving scalar optimization problems. To optimize $g(\mathbf{v}, \tilde{\mathbf{v}})$, we take the derivative with respect to v_i and set it to be zero, namely

$$\frac{(\mathbf{H}^+ \tilde{\mathbf{v}})_i}{\tilde{v}_i} v_i - \frac{(\mathbf{H}^- \tilde{\mathbf{v}})_i \tilde{v}_i}{v_i} + q_i = 0, \quad i = 1, 2, ..., M. \tag{3.25}$$

This is equivalent to the following scalar quadratic equation

$$\frac{(\mathbf{H}^+ \tilde{\mathbf{v}})_i}{\tilde{v}_i} v_i^2 + q_i v_i - (\mathbf{H}^- \tilde{\mathbf{v}})_i \tilde{v}_i = 0, \quad i = 1, 2, ..., M. \tag{3.26}$$

The above scalar quadratic equation has analytic solutions, and the update rule for $\mathbf{v}$ becomes

$$v_i \longleftarrow v_i \frac{-q_i + \sqrt{q_i^2 + 4(\mathbf{H}^+ \mathbf{v})_i (\mathbf{H}^- \mathbf{v})_i}}{2(\mathbf{H}^+ \mathbf{v})_i}, \tag{3.27}$$

where the other solution of the scalar quadratic equation was discarded because it would lead to negative solutions of $\mathbf{v}$.

The update rule in Eq. 3.27 is easy to implement, does not need to determine step size, and is guaranteed to converge to the global optimizer [63]. Furthermore, the update rule only involves matrix-vector multiplication but not matrix inversion, and thus it is expected to be able to solve large-scale nonnegative quadratic programming problems.

Merhotra predictor-corrector primal-dual interior method for NNQP

A primal-dual interior point method for NNQP is based on solving the following equations that describe the optimality of the NNQP problem shown in Eq. 3.19 [69]:

$$\mathbf{Hv} + \mathbf{q} = \boldsymbol{\rho} \tag{3.28}$$

$$v_i \rho_i = 0, \quad i = 1, 2, ..., 2M \tag{3.29}$$

$$v_i \geq 0, \quad \rho_i \geq 0, \quad i = 1, 2, ..., 2M \tag{3.30}$$

where $\boldsymbol{\rho}$ is the dual variable, and Eq. 3.29 is so-called complementary conditions. Then, the primal-dual interior point method is to solve the Eqs. 3.28 and 3.29 using Newton's method while the nonnegative constraints in Eq. 3.30 are enforced by choosing proper step sizes using line search. One practical issue here is, if line search directions are computed directly based on Eqs. 3.28$\sim$3.30, Newton iterations would get to the nonnegative boundary very fast and they would not be able to take big step sizes. Therefore, most existing primal dual interior point methods solve the following equations instead

$$\mathbf{Hv} + \mathbf{q} = \boldsymbol{\rho} \tag{3.31}$$

$$v_i \rho_i = \eta\xi, \quad i = 1, 2, ..., 2M \tag{3.32}$$

$$v_i \geq 0, \quad \rho_i \geq 0, \quad i = 1, 2, ..., 2M \tag{3.33}$$

with $\eta \in [0, 1]$ being a centering parameter and $\xi = \mathbf{v}^T\boldsymbol{\rho}/(2M)$ describing the duality gap of the current solution $\mathbf{v}$ and $\boldsymbol{\rho}$. As such, the Newton's search direction $[\boldsymbol{\Delta}\mathbf{v}; \boldsymbol{\Delta}\boldsymbol{\rho}]$ at a point $[\mathbf{v}; \boldsymbol{\rho}]$ is computed by solving the following linear equation

$$\begin{bmatrix} \mathbf{H} & -\mathbf{I} \\ \mathbf{P} & \mathbf{V} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Delta}\mathbf{v} \\ \boldsymbol{\Delta}\boldsymbol{\rho} \end{bmatrix} = \begin{bmatrix} -\mathbf{r}_1 \\ -\mathbf{VPe} + \eta\xi\mathbf{e} \end{bmatrix} \tag{3.34}$$

where $\mathbf{I}$ is a $2M \times 2M$ identity matrix, $\mathbf{V}$ is a diagonal matrix with $V_{ii} = v_i$, $\mathbf{P}$ is a diagonal matrix with $P_{ii} = \rho_i$, $\mathbf{r}_1 = \mathbf{Hv} - \boldsymbol{\rho} + \mathbf{q}$, and $\mathbf{e} = [1; 1; ...; 1]$ with length $2M$.

Compared to the general primal-dual interior point methods that compute their (Newton's) search direction using Eq. 3.34, Merhotra predictor-corrector primal-dual interior point method computes its search direction $[\boldsymbol{\Delta}\mathbf{v}; \boldsymbol{\Delta}\boldsymbol{\rho}]$ at a point $[\mathbf{v}; \boldsymbol{\rho}]$ using

$$\begin{bmatrix} \boldsymbol{\Delta}\mathbf{v} \\ \boldsymbol{\Delta}\boldsymbol{\rho} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Delta}\mathbf{v}^p \\ \boldsymbol{\Delta}\boldsymbol{\rho}^p \end{bmatrix} + \begin{bmatrix} \boldsymbol{\Delta}\mathbf{v}^c \\ \boldsymbol{\Delta}\boldsymbol{\rho}^c \end{bmatrix} \tag{3.35}$$

where $[\mathbf{v}^p; \boldsymbol{\rho}^p]$ and $[\mathbf{v}^c; \boldsymbol{\rho}^c]$ are respectively computed by solving the following linear equations

$$\begin{bmatrix} \mathbf{H} & -\mathbf{I} \\ \mathbf{P} & \mathbf{V} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Delta}\mathbf{v}^p \\ \boldsymbol{\Delta}\boldsymbol{\rho}^p \end{bmatrix} = \begin{bmatrix} -\mathbf{r}_1 \\ -\mathbf{VPe} \end{bmatrix} \tag{3.36}$$

and

$$\begin{bmatrix} \mathbf{H} & -\mathbf{I} \\ \mathbf{P} & \mathbf{V} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Delta}\mathbf{v}^c \\ \boldsymbol{\Delta}\boldsymbol{\rho}^c \end{bmatrix} = \begin{bmatrix} 0 \\ \eta\xi\mathbf{e} - \boldsymbol{\Delta}\mathbf{V}^p\boldsymbol{\Delta}\mathbf{P}^p\mathbf{e} \end{bmatrix} \tag{3.37}$$

where $\boldsymbol{\Delta}\mathbf{V}^p$ is a diagonal matrix with $\Delta V_{ii}^p = \Delta v_i^p$, $\boldsymbol{\Delta}\mathbf{P}^p$ is a diagonal matrix with $\Delta P_{ii}^p = \Delta\rho_i^p$, and the extra term $-\boldsymbol{\Delta}\mathbf{V}^p\boldsymbol{\Delta}\mathbf{P}^p\mathbf{e}$ in Eq. 3.37 is for correcting the linearization effect introduced in computing a Newton's search direction. Because of the correction step, Merhotra predictor-corrector primal-dual interior method provides better search directions

than the Newton's method. Meanwhile, since the linear equation in Eqs.3.37 has the same coefficient matrix as the one in Eqs.3.36, computing $[\mathbf{v}^c; \boldsymbol{\rho}^c]$ costs little extra computational effort.

Projected gradient descent algorithm for NNQP

The projected gradient descent algorithm [8] is able to efficiently solve the optimization in Eq. 3.19 because the constraints are coordinate-wise bound constraints, which makes projection almost a trivial task as we will explain shortly. For the optimization in Eq. 3.19, the projected gradient descent algorithm consists of the following two steps:

$$1) \quad \text{gradient update: } \tilde{\mathbf{v}}^{new} = \tilde{\mathbf{v}} - \alpha \mathbf{dv}; \tag{3.38}$$

$$2) \quad \text{projection } \mathcal{P} : \tilde{\mathbf{v}}^{new} \rightarrow \tilde{\mathbf{v}}^p \text{ with } \tilde{v}_j^p = \begin{cases} \tilde{v}_j^{new} & \text{if} \quad \tilde{v}_j^{new} > 0 \\ 0 & \text{if} \quad \tilde{v}_j^{new} \leq 0 \end{cases} \tag{3.39}$$

where $\tilde{\mathbf{v}}$ is the current estimate of $\mathbf{v}$, $\mathbf{dv}$ is the gradient of the objective function in Eq. 3.19 at $\tilde{\mathbf{v}}$, and $\alpha > 0$ is step size. Then, $\tilde{\mathbf{v}}$ will be updated by $\tilde{\mathbf{v}}^p$ if $\tilde{\mathbf{v}}^p$ satisfies some conditions (for example, it sufficiently decreases the objective function). It can be show that the simple projection operation in 3.39 indeed projects $\tilde{\mathbf{v}}^{new}$ onto its closest point in the nonnegative orthant, $\tilde{\mathbf{v}}^p$.

One implementation issue in the projected gradient descent algorithm is about how to determine the step size α. We adopt the Armijo type line search for determining α [8]. Let $f(\mathbf{v}) = \frac{1}{2}\mathbf{v}^T\mathbf{H}\mathbf{v} + \mathbf{q}^T\mathbf{v}$ be the objective function in Eq. 3.19, then α is determined by $\alpha = t^k$ with k being the smallest integer that satisfies

$$f(\mathcal{P}(\tilde{\mathbf{v}} - t^k\mathbf{dv})) - f(\tilde{\mathbf{v}}) \leq \tau \mathbf{dv}^T(\mathcal{P}(\tilde{\mathbf{v}} - t^k\mathbf{dv}) - \tilde{\mathbf{v}}) \tag{3.40}$$

where t and τ are constants that are $0 < t < 1$ and $0 < \tau \leq 0.5$ (for example, a typical

setting is $t = 0.5$ and $\tau = 0.01$). The intuition of Eq. 3.40 is that, the right hand side is a negative number and the algorithm wants the descent in each step to be *significant* enough to be no worse than some linear bound. It can be shown that, with the step size being set by the above line search, the projected gradient is guaranteed to monotonically converge to a local optimizer [8], which is also a global optimizer for the optimization in Eq. 3.19 because of its convexity.

In our implementation, in each iteration of projected gradient descent, we also set one element in each pair, v_j (or w_j^+) and v_{j+M} (or w_j^-), to be zero by subtracting both of them with the smaller of them. This explores the fact that one element of each pair has to be zero at optimum and the operation guarantees to decrease the objective function.

It is worth noting that, the project gradient descent algorithm is also able to achieve superlinear convergence by properly scaling the gradient like Newton's method, and the resultant algorithm was named *two-metric* projected gradient descent since the scaling and the projection are done in two different metric spaces. For more details about the two-metric method, please refer to [8].

3.1.3 Variational approximation

After $\mathbf{w}^{MP}$ is computed, one may approximate $Q(\mathbf{w})$ in Eq. 3.13 as a δ-function at $\mathbf{w}^{MP}$. Unfortunately, this simple treatment may cause divergence of the updates when σ^2 and $\boldsymbol{\lambda}$ are not initialized properly. To overcome this difficulty, we require a better approximation to properly account for variability in the distribution $Q(\mathbf{w})$.

The solution of $\mathbf{w}^{MP}$ naturally partitions itself into two distinct groups: non-zero elements (indexed by J) and zero elements (indexed by I). As a result, we choose to approximate the joint distribution $Q(\mathbf{w})$ as a factorized distribution, namely,

$$Q(\mathbf{w}) \approx Q_J(\mathbf{w}_J)Q_I(\mathbf{w}_I). \tag{3.41}$$

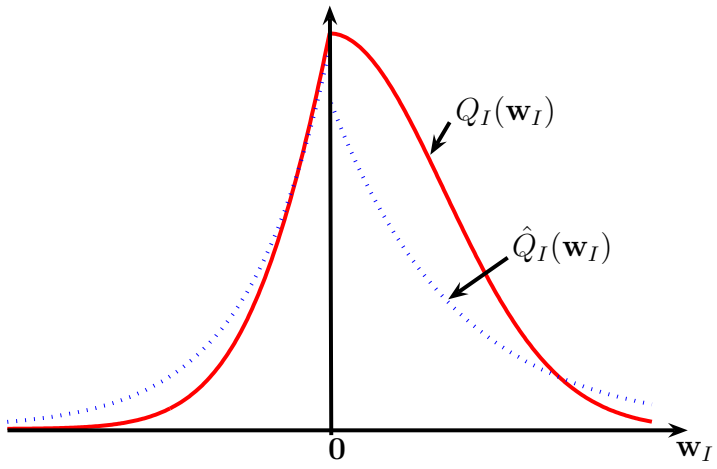


Figure 3.3: The schematic of $\hat{Q}_I(\mathbf{w}_I)$ distribution for approximating $Q_I(\mathbf{w}_I)$ distribution, which has its mode at zero.

Since $\mathbf{w}_J \neq 0$, the first order derivative vanishes $[(\nabla F(\mathbf{w})|_{\mathbf{w}^{MP}})_J = 0]$. Therefore, $Q_J(\mathbf{w}_J)$ is approximated as a Gaussian distribution with mean $\mathbf{w}_J^{MP}$ and covariance $\mathbf{A}_{JJ}^{-1}$ with $\mathbf{A}_{JJ}$ being the sub-matrix of $\mathbf{A}$ [namely, $(\mathbf{A}_{JJ})_{mn} = A_{J(m),J(n)}$].

For $Q_I(\mathbf{w}_I)$, $\mathbf{w}_J$ are treated as point mass (or δ-functions), namely,

$$\begin{aligned} Q_I(\mathbf{w}_I) &= Q(\mathbf{w})|_{\mathbf{w}_J=\mathbf{w}_J^{MP}} \\ &= \frac{1}{\mathcal{Z}_{\mathbf{w}_I}} \exp\{-\frac{1}{2}\mathbf{w}_I^T\mathbf{A}_{II}\mathbf{w}_I - [(\mathbf{A}\mathbf{w}^{MP})_I + \mathbf{b}_I]^T\mathbf{w}_I - \sum_{i\in I}\lambda_i|w_i|\} \end{aligned} \quad (3.42)$$

where $\mathcal{Z}_{\mathbf{w}_I}$ is the normalization factor so that $\int_{\mathbf{w}_I} \mathbf{dw}_I Q_I(\mathbf{w}_I) = 1$, $\mathbf{A}_{II}$ is the sub-matrix of $\mathbf{A}$ [namely, $(\mathbf{A}_{II})_{mn} = A_{I(m),I(n)}$].

Unfortunately, it is still hard to evaluate the integrals in Eqs. 3.11 and 3.12 because it is not easy to compute expectations over the $Q_I(\mathbf{w}_I)$ distribution which has absolute function in its exponent. Therefore, we choose to approximate $Q_I(\mathbf{w}_I)$ using a mean-field approximation. Because $\mathbf{w}_I^{MP} = 0$ and the first order derivative $(\nabla F(\mathbf{w})|_{\mathbf{w}^{MP}})_I \neq 0$, we approximate $Q_I(\mathbf{w}_I)$ with a factorized asymmetric Laplacian distribution (as illustrated in Fig. 3.3), namely

$$Q_I(\mathbf{w}_I) \approx \hat{Q}_I(\mathbf{w}_I) = \prod_{i \in I} \hat{Q}_i(w_i), \tag{3.43}$$

with

$$\hat{Q}_i(w_i) = \begin{cases} \frac{1}{2\mu_i^-} e^{w_i/\mu_i^-} & \text{when} \quad w_i < 0 \\ \frac{1}{2\mu_i^+} e^{-w_i/\mu_i^+} & \text{when} \quad w_i \geq 0, \end{cases} \tag{3.44}$$

where $\boldsymbol{\mu}^+ \geq 0$ and $\boldsymbol{\mu}^- \geq 0$ are variational parameters, and they are defined by minimizing the Kullback$-$Leibler (KL) divergence between Q_I and $\hat{Q}_I$, namely,

$$\begin{aligned} \boldsymbol{\mu}^{+*}, \boldsymbol{\mu}^{-*} &= \arg \min_{\boldsymbol{\mu}^+, \boldsymbol{\mu}^-} D_{KL}[\hat{Q}_I(\mathbf{w}_I) \| Q_I(\mathbf{w}_I)] \\ &= \arg \min_{\boldsymbol{\mu}^+, \boldsymbol{\mu}^-} \int_{\mathbf{w}_I} \mathbf{dw}_I \{\ln[\hat{Q}_I(\mathbf{w}_I)] - \ln[Q_I(\mathbf{w}_I)]\} \hat{Q}_I(\mathbf{w}_I) \end{aligned} \tag{3.45}$$

where

$$\begin{aligned} & \int_{\mathbf{w}_I} \mathbf{dw}_I \ln[\hat{Q}_I(\mathbf{w}_I)] \hat{Q}_I(\mathbf{w}_I) \\ = & \sum_{i \in I} \int_{w_i} dw_i \ln[\hat{Q}_i(w_i)] \hat{Q}_i(w_i) \\ = & \sum_{i \in I} \int_{-\infty}^{0} dw_i (-\ln 2 - \ln \mu_i^- + w_i/\mu_i^-) \frac{1}{2\mu_i^-} e^{w_i/\mu_i^-} \\ & + \sum_{i \in I} \int_{0}^{\infty} dw_i (-\ln 2 - \ln \mu_i^- + w_i/\mu_i^+) \frac{1}{2\mu_i^+} e^{-w_i/\mu_i^+} \\ = & -\frac{1}{2} \sum_{i \in I} (\ln u_i^- + \ln \mu_i^+) - \ln 2 - 1 \end{aligned} \tag{3.46}$$

and

$$\begin{aligned} & \int_{\mathbf{w}_I} \mathbf{dw}_I \ln[Q_I(\mathbf{w}_I)] \hat{Q}_I(\mathbf{w}_I) \\ = & -\ln \mathcal{Z}_{\mathbf{w}_I} - \int_{\mathbf{w}_I} \mathbf{dw}_I \{\frac{1}{2} \mathbf{w}_I^T \mathbf{A}_{II} \mathbf{w}_I + [(\mathbf{A}\mathbf{w}^{MP})_I + \mathbf{b}_I]^T \mathbf{w}_I + \sum_{i \in I} \lambda_i |w_i|\} \hat{Q}_I(\mathbf{w}_I) \\ = & -\ln \mathcal{Z}_{\mathbf{w}_I} - \sum_{i \in I, j \in I} \frac{1}{2} A_{i,j} \langle w_i w_j \rangle - \sum_{i \in I} (b_i + (\mathbf{A}\mathbf{w}^{MP})_i) \langle w_i \rangle - \sum_{i \in I} \lambda_i \langle |w_i| \rangle \end{aligned} \tag{3.47}$$

with $\langle\cdot\rangle$ denoting the expectation under $\hat{Q}_I(\mathbf{w}_I)$ distribution. It is easy to see that

$$\begin{aligned}
\langle w_i\rangle &= \frac{1}{2}(\mu_i^+ - \mu_i^-); && (3.48)\\
\langle |w_i|\rangle &= \frac{1}{2}(\mu_i^+ + \mu_i^-); && (3.49)\\
\langle w_i w_j\rangle &= \frac{1}{4}(\mu_i^+ - \mu_i^-)(\mu_j^+ - \mu_j^-) \quad i \neq j; && (3.50)\\
\langle w_i^2\rangle &= (\mu_i^+)^2 + (\mu_i^-)^2. && (3.51)
\end{aligned}$$

Combining the Eqs. 3.45$\sim$3.51, the optimization for finding the mean field parameters is:

$$[\boldsymbol{\mu}^{+*}; \boldsymbol{\mu}^{-*}] = \arg\min_{\boldsymbol{\mu}\geq 0} \frac{1}{2}\boldsymbol{\mu}^T \hat{\mathbf{A}}\boldsymbol{\mu} + \hat{\mathbf{b}}^T\boldsymbol{\mu} - \sum_i \ln \mu_i, \tag{3.52}$$

where $\boldsymbol{\mu} = [\boldsymbol{\mu}^+; \boldsymbol{\mu}^-]$, $\hat{\mathbf{b}} = [(\mathbf{A}\mathbf{w}^{MP} + \mathbf{b} + \boldsymbol{\lambda})_I; (-\mathbf{A}\mathbf{w}^{MP} - \mathbf{b} + \boldsymbol{\lambda})_I]$, and $\hat{\mathbf{A}} = \begin{bmatrix} \hat{\mathbf{A}}_{11} & \hat{\mathbf{A}}_{12} \\ \hat{\mathbf{A}}_{21} & \hat{\mathbf{A}}_{22} \end{bmatrix}$, where $\hat{\mathbf{A}}_{11} = \hat{\mathbf{A}}_{22} = \frac{1}{2}\mathbf{A}_{II} + \frac{3}{2}\text{diag}(\mathbf{A}_{II})$, and $\hat{\mathbf{A}}_{21} = \hat{\mathbf{A}}_{12} = \frac{1}{2}\mathbf{A}_{II} - \frac{1}{2}\text{diag}(\mathbf{A}_{II})$ with $\mathbf{A}_{II}$ being the sub-matrix of $\mathbf{A}$, and $\text{diag}(\mathbf{A}_{II})$ denoting the diagonal matrix whose diagonal is the diagonal of $\mathbf{A}_{II}$.

The minimization problem in Eq. 3.52 can not be solved analytically, but it can be solved by many optimization methods. First, the optimization in Eq. 3.52 can be treated as an unconstrained optimization since the log-terms act as a natural log-barrier. Consequently, the optimization in Eq. 3.52 can be solved by Newton's method or gradient descent algorithms with a proper line search scheme for determining step sizes. Second, the optimization in Eq. 3.52 can also be solved by constructing a series of auxiliary functions. The auxiliary functions are similar to the one we employed for solving the NNQP problem in Eq. 3.19. So, using a similar upper bound as the one in Eq. 3.24, the auxiliary function for

Eq. 3.52 is:

$$
\begin{aligned}
g(\boldsymbol{\mu}, \tilde{\boldsymbol{\mu}}) &= \frac{1}{2}\sum_{i\in I}\frac{(\hat{\mathbf{A}}^{+}\tilde{\boldsymbol{\mu}})_i}{\tilde{\mu}_i}\mu_i^2 \\
&\quad -\sum_{i\in I}(\hat{\mathbf{A}}^{-}\tilde{\boldsymbol{\mu}})_i\tilde{\mu}_i\ln\frac{\mu_i}{\tilde{\mu}_i} - \frac{1}{2}\sum_{i,j\in I}\hat{A}^{-}_{ij}\tilde{\mu}_i\tilde{\mu}_j + \hat{\mathbf{b}}_I^T\boldsymbol{\mu} - \sum_{i\in I}\ln\mu_i,
\end{aligned}
\tag{3.53}
$$

where $\hat{\mathbf{A}} = \hat{\mathbf{A}}^{+} - \hat{\mathbf{A}}^{-}$ is the decomposition of $\hat{\mathbf{A}}$ into its positive and negative components in the same way as the decomposition of $\mathbf{H}$ matrix in Eq. 3.23. Now, similar to Eq. 3.24, $g(\boldsymbol{\mu}, \tilde{\boldsymbol{\mu}})$ is coordinate-wise separable, and it can be minimized analytically. Take the derivative of $g(\boldsymbol{\mu}, \tilde{\boldsymbol{\mu}})$ with respect to $\boldsymbol{\mu}$ and set it to be zero, then

$$
\frac{(\hat{\mathbf{A}}^{+}\tilde{\boldsymbol{\mu}})_i}{\tilde{\mu}_i}\mu_i - (\hat{\mathbf{A}}^{-}\tilde{\boldsymbol{\mu}})_i\tilde{\mu}_i\frac{1}{\mu_i} + \hat{b}_i - \frac{1}{\mu_i} = 0, \quad i \in I. \tag{3.54}
$$

These equations are equivalent to the following quadratic equations:

$$
\frac{(\hat{\mathbf{A}}^{+}\tilde{\boldsymbol{\mu}})_i}{\tilde{\mu}_i}\mu_i^2 + \hat{b}_i\mu_i - [(\hat{\mathbf{A}}^{-}\tilde{\boldsymbol{\mu}})_i\tilde{\mu}_i + 1] = 0, \quad i \in I, \tag{3.55}
$$

and they can be solved analytically. As a result, the update rule for solving the optimization in Eq. 3.52 is

$$
\mu_i \longleftarrow \mu_i\frac{-\hat{b}_i + \sqrt{\hat{b}_i^2 + 4(\hat{\mathbf{A}}^{+}\boldsymbol{\mu})_i[(\hat{\mathbf{A}}^{-}\boldsymbol{\mu})_i + \frac{1}{\mu_i}]}}{2(\hat{\mathbf{A}}^{+}\boldsymbol{\mu})_i}. \tag{3.56}
$$

which is a multiplicative update and guaranteed to monotonically converge to the global optimizer.

After the variational parameters $\boldsymbol{\mu}^{+*}$ and $\boldsymbol{\mu}^{-*}$ are derived, the mean $\bar{\mathbf{w}}$, the absolute mean $\overline{|w_i|}, i = 1, 2, ..., M$, and the covariance $\mathbf{C}$ of $\mathbf{w}$ under the approximated distribution

$Q_J(\mathbf{w}_J)\hat{Q}_I(\mathbf{w}_I)$can be computed:

$$\bar{w}_i = \begin{cases} w_i^{ML} & \text{if} \quad i \in J \\ (\mu_i^{+*} - \mu_i^{-*})/2 & \text{if} \quad i \in I \end{cases}, \tag{3.57}$$

$$\overline{|w_i|} = \begin{cases} |w_i^{ML}| & \text{if} \quad i \in J \\ (\mu_i^{+*} + \mu_i^{-*})/2 & \text{if} \quad i \in I \end{cases}, \tag{3.58}$$

$$C_{ij} = \begin{cases} (\mathbf{A_{JJ}}^{-1})_{ij} & \text{if } i,j \in J \\ \delta_{ij}[\frac{(\mu_i^{+*}+\mu_i^{-*})^2}{4} + \frac{(\mu_i^{+*})^2+(\mu_i^{-*})^2}{2}] & \text{otherwise.} \end{cases} \tag{3.59}$$

From these statistics, the integrals in Eqs. 3.11 and 3.12 can be evaluated analytically, and the update rules for estimating $\boldsymbol{\lambda}$ and σ^2 becomes:

$$\lambda_j \longleftarrow \frac{1}{\overline{|w_j|}}, \quad j = 1, 2, ..., M \tag{3.60}$$

$$\sigma^2 \longleftarrow \frac{1}{N}[(\mathbf{y} - \boldsymbol{\Phi}\bar{\mathbf{w}})^T(\mathbf{y} - \boldsymbol{\Phi}\bar{\mathbf{w}}) + \text{Tr}(\boldsymbol{\Phi}^T\boldsymbol{\Phi}\mathbf{C})]. \tag{3.61}$$

3.1.4 Bayesian framework for uniform l_1-norm regularization

The above Bayesian framework can be adapted for learning the optimal scalar regularization parameter λ' in the uniform l_1-norm regularized least squares problem, as shown in Eq. 3.3. Compared to the $\boldsymbol{\lambda}$ vector in independent l_1-norm regularization in Eq. 3.2, the scalar λ' is a uniform l_1-norm regularization parameter shared by all weights ($w_j, j = 1, 2, ..., M$). Accordingly, to formulate the uniform l_1-norm regularized least squares problem in a Bayesian framework, the weights are assumed to have a uniform sparsity prior,

ALGORITHM OF l_1-NORM SPARSE BAYESIAN LEARNING FOR ORDINARY LEAST SQUARES

1. Initialize σ^2 and λ (for example, $\sigma^2 = 0.01$ and $\lambda = 10$).
2. Solve the optimization in Eq. 3.3 for computing $\mathbf{w}^{MP}$.
3. Do the variational approximation by solving the optimization in Eq. 3.52.
4. Compute the statistic of $\mathbf{w}$ in Eqs. 3.57$\sim$3.59 under the approximated $Q(\mathbf{w})$ distribution.
5. Update σ^2 and λ using the update rule in Eq. 3.61 and Eq. 3.65, respectively.
6. Repeat Step 2$\sim$Step 5 until convergence.
7. Initialize $\boldsymbol{\lambda} = [\lambda; \lambda; ...; \lambda]$ ($M \times 1$ vector with M being the dimension of $\mathbf{w}$).
8. Solve the optimization in Eq. 3.2 for computing $\mathbf{w}^{MP}$.
9. Do the variational approximation by solving the optimization in Eq. 3.52.
10. Compute the statistic of $\mathbf{w}$ in Eqs. 3.57$\sim$3.59 under the approximated $Q(\mathbf{w})$ distribution.
11. Update σ^2 and $\boldsymbol{\lambda}$ using the update rule in Eq. 3.61 and Eq. 3.60, respectively.
12. Repeat Step 8$\sim$Step 11 until convergence.

Figure 3.4: The algorithm of l_1-norm sparse Bayesian learning for ordinary least squares problems in Eqs. 3.2 and 3.3.

namely,

$$P(\mathbf{w}|\lambda) = (\frac{\lambda}{2})^M \exp\{-\lambda \sum_i |w_i|\} \tag{3.62}$$

while the data vector $\mathbf{y}$ is still assumed to be corrupted by Gaussian noise, as shown in Eq. 3.5. In the Bayesian formulation for this uniform case, the iterative estimates are similar to before except that Eq. 3.10, Eq. 3.11 and Eq. 3.60 respectively become [49]

$$F(\mathbf{w}) = \frac{1}{2\sigma^2}\|\mathbf{y} - \mathbf{\Phi w}\|^2 + \lambda \sum_{j=1}^{M} |w_j|. \tag{3.63}$$

$$\lambda \longleftarrow \frac{1}{\sum_{j=1}^{M} \int_{-\infty}^{+\infty} d\mathbf{w}\, |w_j| Q(\mathbf{w})} \tag{3.64}$$

$$\lambda \longleftarrow \frac{1}{\sum_{j=1}^{M} \overline{|w_j|}}. \tag{3.65}$$

Since there are fewer parameters to estimate, a uniform regularization prior is convenient to use at the beginning of the algorithm. Using an independent prior results in stronger sparsity regularization, but with more parameters to estimate. Thus, in our implementation of l_1-norm sparse Bayesian learning, a uniform regularization is initially used in the beginning iterations, and then the solution is optimally refined using independent regularization.

To summarize, Fig. 3.4 shows the algorithm of l_1-norm sparse Bayesian learning for ordinary least squares problems in Eqs. 3.2 and 3.3.

3.1.5 Simulations

We employ simulations to demonstrate the performance the algorithm of l_1-norm sparse Bayesian learning for ordinary least squares problems. The algorithm was summarized in Fig. 3.4. We show the convergence of the algorithm, demonstrate its capability to accurately discover true sparse structures in data, and compare its performance with the its l_2-norm counterpart, relevance vector machine (RVM) regression.

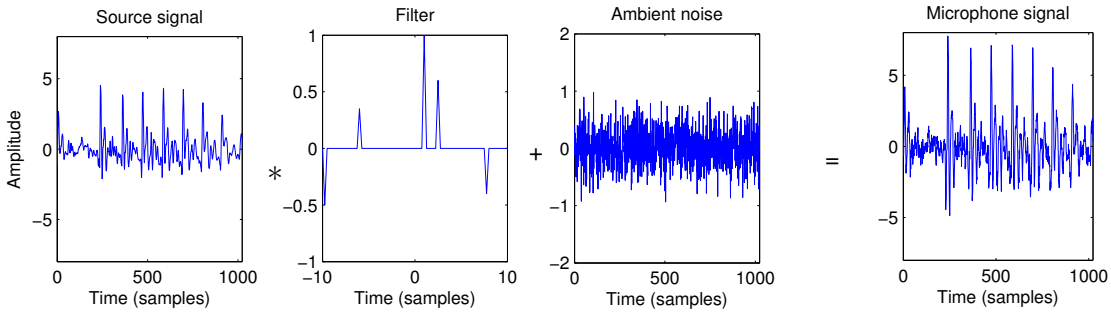


Figure 3.5: Simulated signals for the FIR filter identification example with sub-sample resolution. The microphone signal is the convolution (denoted by $*$) of the source signal and the filter, corrupted by zero-mean Gaussian noise. The time resolution of the filter is 4 times higher of the one in the source and microphone signals.

Comparison between l_1-norm sparse Bayesian learning with conventional l_1-norm regularization approaches

Here we employ a simulated FIR filter identification example to demonstrate that the derived update rule for l_1-norm sparse Bayesian learning converges to the correct noise level (σ^2) and the Bayesian approach is able to accurately discover the true sparse solution. In particular, with inferred optimal independent regularization parameters, the optimization problem in Eq. 3.2 is able to accurately resolve the correct sparseness of the solution even in very noisy data. In comparison, the conventional l_1-norm regularization approaches, which is the optimization in Eq. 3.3 with the scalar regularization parameter λ' set by either heuristic approaches or cross-validation, yield sub-optimal results.

The simulated signals are shown in Fig.3.5. A speech segment (1024 samples, sampling frequency was 16,000Hz) was employed as the source signal $\mathbf{s}$. The simulated sparse FIR filter $\mathbf{w}$ had nonzero amplitudes of -0.5, 0.35, 1, 0.6, and -0.4 at -9.75T_s, -6T_s, 1T_s, 2.5T_s, and 7.75T_s (T_s denotes the sample interval), respectively, and had zero amplitude elsewhere. We intended to make the problem more challenging by setting the goal to identify such a super-resolution FIR filter. Then the observation $\mathbf{y}$ was the convolution of the source

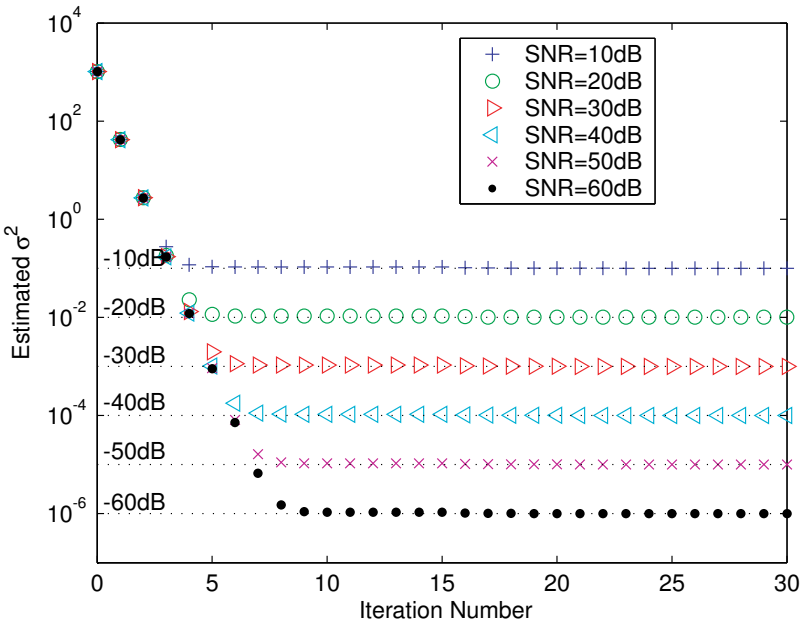


Figure 3.6: Convergence of σ^2 estimation in Bayesian L_1-norm sparse learning. The source signal was normalized so that it had unit power.

with the simulated filter, corrupted by I.I.D. sampled zero-mean Gaussian noise. The task of filter identification was to discover the filter $\mathbf{w}$ given the source $\mathbf{s}$ and the observation $\mathbf{y}$.

We utilized the proposed l_1-norm sparse Bayesian learning algorithm for the least squares problem in Eq. 3.2 to identify the super-resolution filter. The columns of the designed matrix $\mathbf{\Phi}$ are the delayed patterns of the source with delays from $-10T_s$ to $+10T_s$ incremented by $0.25T_s$. Due to the fact that the adjacent columns in $\mathbf{\Phi}$ are very similar to each other and the matrix $\mathbf{\Phi}^T\mathbf{\Phi}$ is ill-conditioned, sparsity regularization is crucial for deriving a correct solution.

Figure 3.6 illustrates the convergence of the σ^2 estimation under different noise levels (from -60dB to -10dB) using the update rules of l_1-norm sparse Bayesian learning derived in Eqs. 3.60, 3.61 and 3.65. In the simulation, uniform regularization was employed in the first 15 iterations, and then independent regularization was utilized in the next 15 iterations to further refine the solution. From Figure 3.6, we observe that the σ^2 estimate converges

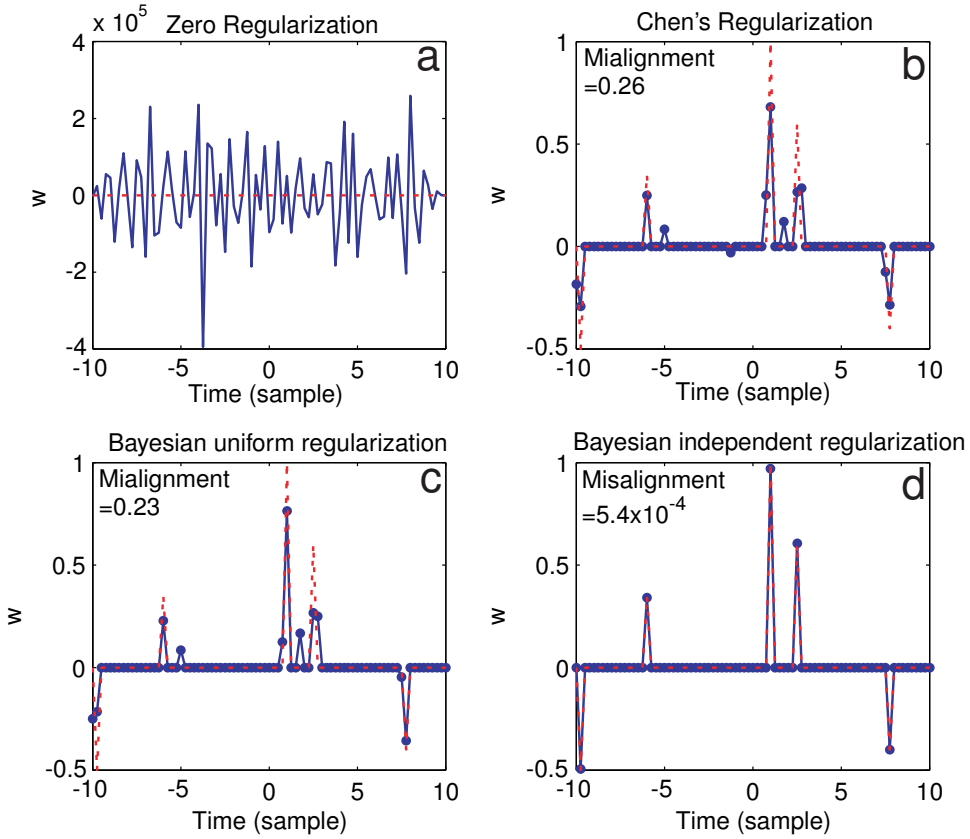


Figure 3.7: Filter identification result by different l_1-norm regularization schemes. a) no regularization; b) the regularization proposed by S. S. Chen *et. al* [18]($\lambda' = 0.94$); c)Bayesian uniform regularization, ($\lambda = 28$ and $\sigma^2 = 0.1$, thus $\hat{\lambda} = 2.8$); d) Bayesian independent regularization. The dot lines in the figures indicate the ground truth of the filter, while the solid lines with dots are the estimates.

to the true value even with bad initialization.

The resulting filter estimate when SNR=10dB is shown in Figure 3.7 (d). Compared to the estimate of the first 15 iterations with uniform regularization (shown in Figure 3.7 (c)), the result of an additional 15 iterations with independent regularization exhibits the same sparseness as the true solution and has very small misalignment (defined as $\|\hat{\mathbf{w}} - \mathbf{w}_0\|^2/\|\mathbf{w}_0\|^2$ with $\hat{\mathbf{w}}$ being the estimate and $\mathbf{w}_0$ being the ground truth). By contrast, other approaches that empirically determine the regularization parameter often yield sub-optimal solutions, as shown in Figure 3.7 (b). Because the simulated deconvolution

is ill-conditioned without sparsity regularization, the estimate in Figure 3.7 (a) with no regularization fluctuates widely, containing little information about the true filter.

Comparison between l_1-norm and l_2-norm sparse Bayesian learning

The l_2-norm sparse Bayesian learning is also known as relevance vector machine (RVM) regression [66], as reviewed in Section 2.3 in Chapter 2. To compare the performance between l_1-norm and l_2-norm sparse Bayesian learning, we employed two simulated examples. One was the commonly used sinc function regression example, and the other was the filter identification example that we described in Fig. 3.5.

In the sinc function regression example, we intended to using the same setup as the one in [66]. We were given data vectors $\mathbf{x}$ and $\mathbf{y}$, where $\mathbf{x}$ is a 100×1 vector sampled from [-10, 10] with equally spaced sampling intervals, and $\mathbf{y} = \sin(\mathbf{x})/\mathbf{x} + 0.1\mathbf{n}$ with the $\sin$ and division operation being element-wise and $\mathbf{n}$ (100×1 vector) being I.I.D. zero-mean unit variance Gaussian noise. Then, the task of regression here is to find a function $y = f(x)$ that mimic the sinc function ($y = sinc(x)$) given data $\mathbf{x}$ and $\mathbf{y}$. The function is parameterized by a weight vector $\mathbf{w} = [w_0; w_1; ...w_{100}]$ and has the form

$$y = \sum_{j=1}^{100} w_j K(x, x_j) + w_0; \tag{3.66}$$

where $K(x, x_j) = e^{-\eta(x-x_j)^2}$ is a Gaussian kernel with $\eta = 1/9$. Therefore, the weights may be find by least squares fitting. However, since the number of variables (which is 101) is larger than the number of data points (which is 100), the least squares problem will be ill-posed. To overcome the illness of the problem, M. Tipping [66] proposed l_2-norm sparse Bayesian learning, namely

$$\mathbf{w}^*_{l2} = \arg\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{y} - \mathbf{\Phi}\mathbf{w}\|_2^2 + \frac{1}{2}\sum_{j=0}^{100} \alpha'_j w_j^2 \tag{3.67}$$

	Mean root squared error	Average number of relevance vectors
l_2-*norm SBL*	0.032	6.2
l_1-*norm SBL*	0.059	4.9

Table 3.1: Result of Sinc function regression using l_1-norm and l_2-norm sparse Bayesian learning (SBL) on 100 experiments.

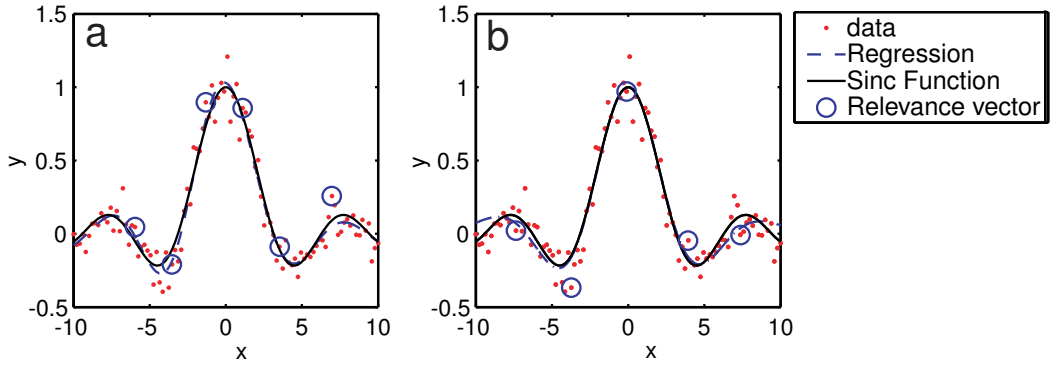


Figure 3.8: Sinc function regression result. a) By L_2-norm sparse Bayesian learning, b) by By L_1-norm sparse Bayesian learning.

where $\boldsymbol{\alpha}' = [\alpha'_0; \alpha'_1; ...; \alpha'_{100}]$ were l_2-norm regularization parameters to be learned in the Bayesian framework described in Section 2.3 in Chapter 2, and $\boldsymbol{\Phi}$ was the design matrix whose first column is $[1; 1; ...; 1]$ and the j^{th} column is $[K(x_1, x_j); K(x_2, x_j); ...K(x_{100}, x_j)]$. In the result, those data samples corresponding to nonzero weights are called relevance vector. For l_1-norm sparse Bayesian learning, we solved the optimization in Eq. 3.2 with the l_1-norm regularization parameters $\boldsymbol{\lambda}'$ to be learning in a Bayesian framework via σ^2 and $\boldsymbol{\lambda}$ using the algorithm described in Fig. 3.4.

The results of the l_1-norm and l_2-norm sparse Bayesian learning are shown in Table 3.1. The results were averaged over 100 independent trails and different trails had different Gaussian noise samples (vector $\mathbf{n}$ were different among trials). We see that the l_1-norm sparse Bayesian learning yielded sparser solutions than its l_2-norm counterpart. Figure 3.8 shows one example of the regression results by the l_1-norm and l_2-norm sparse Bayesian learning. The l_1-norm sparse Bayesian learning was not able to achieve dual improvement

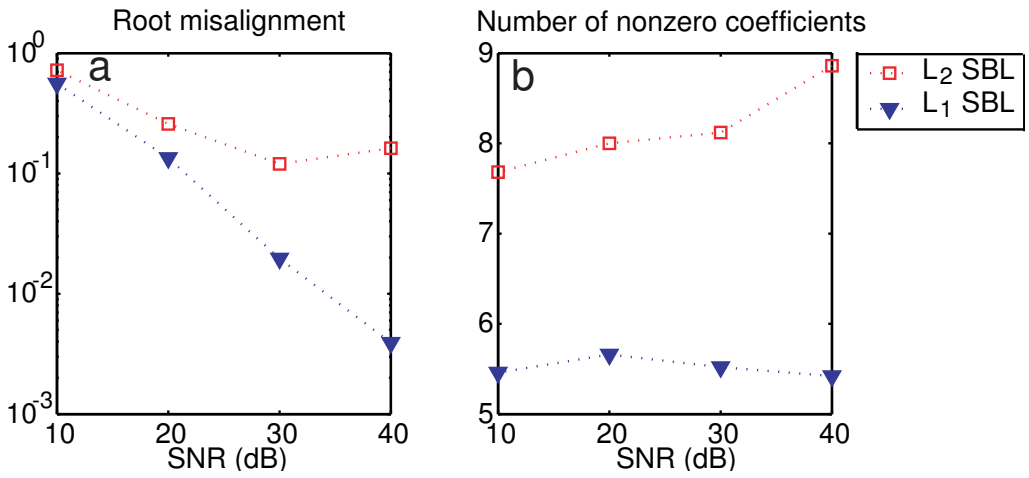


Figure 3.9: The FIR filter identification results by l_1-norm and l_2-norm sparse Bayesian learning. The results at each noise level were averaged over 50 independent trials. a) Mean root misalignment; b) average number of non-zero elements (The true number of non-zero elements is 5).

(both more sparseness and less fitting error) in this simulation. This probably is because the setting $\eta = 1/9$ is somehow an optimized choice for l_2-norm sparse Bayesian learning.

We also employed the simulated super-resolution FIR filter identification example in Fig. 3.5 for comparing the performance between the l_1-norm and l_2-norm sparse Baysian learning. Compared to the sinc function regression example, this simulation knew the ground truth of solutions. To see the performance of the two sparse Bayesian learning approaches at different noise levels, the microphone signals were corrupted with Gaussian noise at four signal to noise ratios (SNR), 10dB, 20dB, 30dB and 40dB. At each noise level, we did 50 trials where each trail sampled an independent noise vector. Then, we performed l_1-norm and l_2-norm sparse Bayesian learning to identify the filter given the source and the noisy microphone observations.

Figure 3.9 shows the results of the l_1-norm and l_2-norm sparse Bayesian learning at the different noise levels. At each noise level, the results were averaged over 50 trials. The results demonstrate that, compared to the L_2-norm Bayesian sparse learning, our proposed

L_1-norm Bayesian sparse learning achieved significant improvement in both reducing the root misalignment ($\sqrt{\|\mathbf{h}-\mathbf{h}_0\|^2/\|\mathbf{h}_0\|^2}$, where $\mathbf{h}$ is the estimated filter, $\mathbf{h}_0$ is the true filter) and the sparseness of solutions. In particular, the l_2-norm sparse Bayesian learning seems to have trouble when the noise level was low. This probably was due to the illness of the problem, which attempted to identify a filter with much higher resolution than the available signals. In contrast, the l_1-norm Bayesian sparse learning was able to overcome the illness. This is not a surprise since l_1-norm regularization is known to be more effective than l_2-norm for finding sparse solutions and has been demonstrated to be good at discriminating solution degeneracies, for example, in learning overcomplete dictionary [45].

3.2 l_1-norm sparse Bayesian learning for *nonnegative* least squares

The Bayesian formulation in Section 3.1 can be adapted for other l_1-norm regularized problems, and simplest adaptation probably is for the l_1-norm regularized *nonnegative* least squares problem, shown in Eq. 3.93. As we described in Section 2.1 in Chapter 2, nonnegative constraint itself is a very useful form of sparsity regularization, and nonnegative least squares problem is essential for some popular algorithms like nonnegative deconvolution [51] and nonnegative matrix factorization (NMF) [40]. To be further motivated by the nonnegative constraint, let's look at an example of nonnegative deconvolution for acoustic time delay estimation.

The task of acoustic time delay estimation is illustrated In Fig. 3.10. In a linear time-invariant system, the microphone signal $y(t)$ is described as

$$y(t) = \int_\tau h(\tau)s(t-\tau)d\tau + n(t) \tag{3.68}$$

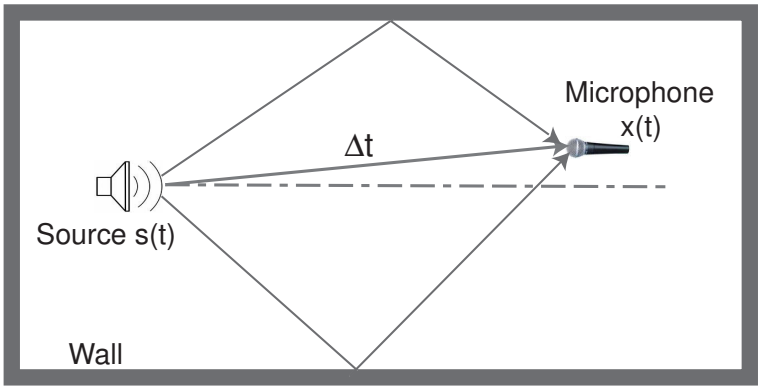


Figure 3.10: Illustration of an acoustic system for time delay estimation. A microphone observation consists of a direct path signal, multipath reflections, and ambient noise. The task of time delay estimation is to estimate how long it takes the source signal to travel from the speaker to the microphone in the direct path. The time delay is denoted as Δt.

where $y(t)$ is the convolution of the acoustic source signal $s(t)$ and the room impulse response $h(t)$, corrupted by additive noise $n(t)$. An room impulse response models multi-path reflections in a room, and time delay estimation relies on identifying the room impulse response by solving the deconvolution problem:

$$\min_{h(t)} \int dt \, \frac{1}{2} \left\| y(t) - \int dt' \, h(t') s(t - t') \right\|^2 . \tag{3.69}$$

Here we assume the source signal is known, which could be true for a robot sound source since we usually know what sound a robot is playing [50].

Conventional cross-correlation can be viewed as an optimization of Eq. 3.69 under the assumption that the room impulse response is a delta function, namely, $h(t) = \alpha_\tau \delta(t - \tau)$. With this assumption, the optimal estimates of τ and α_τ given the source signal $s(t)$ and the measured signal $y(t)$ are [37]:

$$\Delta t = \arg\max_{\tau} \int y(t) s(t - \tau) dt \tag{3.70}$$

$$\hat{\alpha}_\tau = \frac{\int y(t) s(t - \Delta t) dt}{\int s(t)^2 dt}. \tag{3.71}$$

Eqs. 3.70 and 3.71 show that the optimal estimates are related to the maximal value of the cross-correlation between $s(t)$ and $y(t)$. Because of its computational simplicity, cross-correlation has been widely adopted for applications such as time delay estimation. However, the underlying assumption of a delta-function impulse response causes cross-correlation estimates to degrade in reverberant environments where multipath reflections are not negligible. Generalized cross-correlation techniques such as the phase alignment transform pre-whiten the signals before performing cross-correlation to help alleviate some of these difficulties [37], but their effectiveness is still limited by the underlying simple delta-function assumption on the room impulse response.

On the other hand, the least squares optimization of Eq. 3.69 without any constraints on the impulse response is equivalent to conventional linear deconvolution [39, 30]. With discrete-time signals, Eq. 3.69 can be written in matrix form as:

$$\min_{\mathbf{w}} \frac{1}{2} \|\mathbf{y} - \mathbf{\Phi}\mathbf{w}\|^2 \tag{3.72}$$

where $\mathbf{y}$ ($N \times 1$ vector) is the discrete version of $y(t)$, $\mathbf{\Phi}$ is an $N \times M$ Toeplitz convolution matrix with first column being $\mathbf{y}$ and first row being $[y(1); 0; ...; 0]^T$, and $\mathbf{w}$ ($M \times 1$ vector) is the discrete version of the impulse response $h(t)$. When the number of measurements N is larger than the number of time lags M, the resulting matrix optimization can be solved by taking the pseudo-inverse:

$$\mathbf{w}^*_{LS} = (\mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T\mathbf{y}, \tag{3.73}$$

and time delay estimation can be acquired by examining the directly path coefficient in $\mathbf{w}^*_{LS}$.

According to the image model of room acoustics [4], the filter coefficients are approximately nonnegative. Therefore, we can enforce nonnegative constraint on filter coefficients

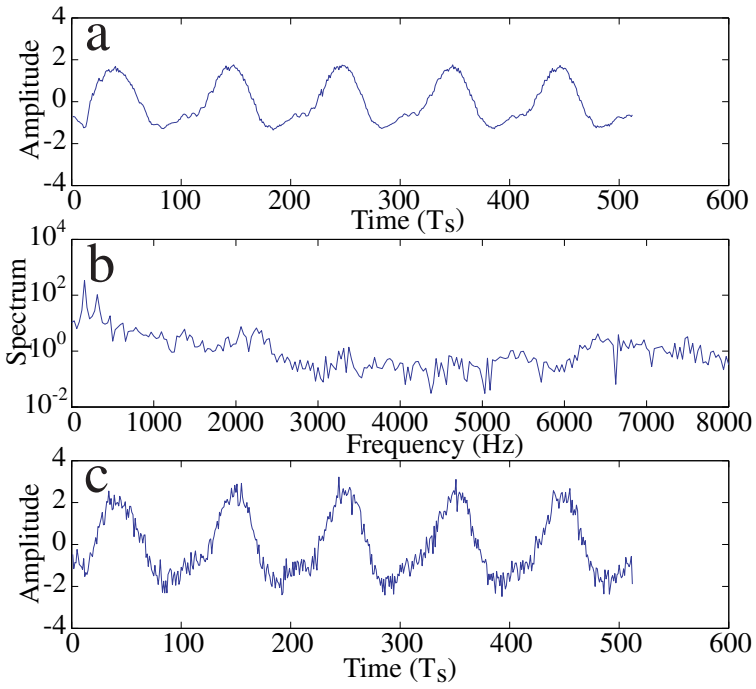


Figure 3.11: The signals used for the simulation: a) source signal $s(t)$, b) source spectrum, $|S(f)|$, c) the simulated measurement $y(t) = s(t - T_s) + 0.5s(t - 8.75T_s) + n(t)$, where $T_s = 62.5\mu s$ is the sample interval and $n(t)$ is varying levels of ambient noise.

in Eq. 3.72, resulting a nonnegative deconvolution formulation for time delay estimation

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{y} - \mathbf{\Phi w}\|^2 \tag{3.74}$$
$$\text{subject to: } \mathbf{w} \geq \mathbf{0}$$

In the following simulated example of time delay estimation, we show that the nonnegative constraint can be very useful.

Figure 3.11 shows the simulated signal for time delay estimation. The source signal is short segment of human speech (512 samples at 16 bit resolution and sampling time $T_s = 62.5\mu s$). This speech signal was convolved with a simulated room impulse response to model the measured microphone signal. For now, let's look at noiseless case (ambient noise $n(t) = 0$).

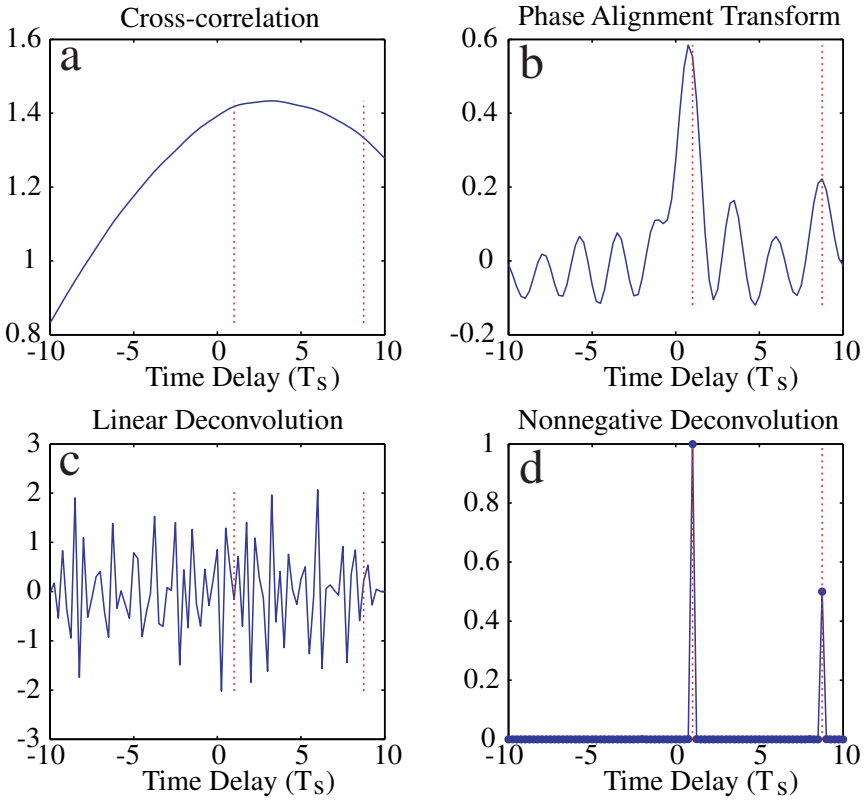


Figure 3.12: Time delay estimation of the room impulse response $h(t) = \delta(t - T_s) + 0.5\delta(t - 8.75T_s)$ by a) cross-correlation, b) phase alignment transform, c) linear deconvolution, d) nonnegative deconvolution. The vertical dotted lines in each plot indicate the true positions of the time delays $\Delta t = T_s$ and $\Delta t = 8.75T_s$, respectively.

Figure. 3.12 shows the results of estimating the room impulse response using several different methods. All the estimates were performed over a range from $-10T_s$ to $+10T_s$, with discrete time increments of $0.25T_s$. Because the speech source signal has limited bandwidth, the cross-correlation in Fig 3.12(a) shows only a broad main lobe, resulting in poor time delay resolution. Due to the multipath reflection, the peak of the cross-correlation function estimates neither of the time delays present in the room impulse response. The phase alignment transform (PHAT) performs better than simple cross-correlation as shown in Fig. 3.12(b). PHAT prewhitens the signals before cross-correlation [37], but again since the signals are not broadband, there are still some errors in the time delay estimation. PHAT also significantly degrades in performance with the presence of any ambient noise. Even worse results arise from linear deconvolution as shown in Fig. 3.12(c). The ill-conditioning of the source correlation matrix $\mathbf{\Phi}^T\mathbf{\Phi}$ causes the calculation of the pseudo-inverse and the resulting estimates of the room impulse response to fluctuate wildly. The dramatic effect of nonnegativity constraints in regularizing the deconvolution problem is displayed in Fig. 3.12(d). Nonnegative deconvolution is able to precisely resolve the room impulse response from the given acoustic signals, including the multipath time delays and amplitudes of the filter coefficients.

We have seen the significance of nonnegative constraint for deriving sparse nonnegative filter solutions when a microphone signal is noiseless. However, when it is coupled with ambient noise, filter estimates will not be so clean anymore. In order to improve the robustness of the nonnegative deconvolution to ambient noise, we propose to enforce l_1-norm regularization on the filter coefficients, namely

$$\begin{aligned} \mathbf{w}^* &= \arg\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{y} - \mathbf{\Phi}\mathbf{w}\|_2^2 + \sum_{j=1}^{M} \lambda_j' w_j \\ &\text{subject to : } \mathbf{w} \geq 0, \end{aligned} \tag{3.75}$$

and the sparsity regularization parameters $\boldsymbol{\lambda}' = [\lambda'_1; \lambda'_2; ...; \lambda'_M]$ are to be learned in a Bayesian framework described in the following. When the nonnegative least squares is used for nonnegative deconvolution and it is formulated in a Bayesian framework for inferring the optimal l_1-norm sparsity regularization parameters, we named the resultant algorithm as *Bayesian Regularization And Nonnegative Deconvolution* (BRAND) algorithm. The Bayesian framework for Eq. 3.76 is similar to the one for ordinary least squares presented in Section 3.1. To make the presentation concise, we would only emphasize the difference between them.

3.2.1 Bayesian framework

The probabilistic model for Eq. 3.76 assumes that, the data vector $\mathbf{y}$ is corrupted by I.I.D. zero mean Gaussian noise

$$P(\mathbf{y}|\mathbf{w}, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{N/2}} \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{y} - \boldsymbol{\Phi}\mathbf{w}\|^2\right), \tag{3.76}$$

and the filter coefficients are governed by a factorized independent exponential distribution,

$$P(\mathbf{w}|\boldsymbol{\lambda}) = \prod_{j=1}^{M} \lambda_j \exp\{-\lambda_j|w_j|\}, \quad w_j \geq 0, j = 1, 2, ..., M, \tag{3.77}$$

where $\boldsymbol{\lambda} = [\lambda_1; \lambda_2; ...; \lambda_M]$ and each parameter λ_j describes the slope of the j^{th} exponential distribution, as shown in Fig. 3.13. Then, similar to the formulation in Section 3.1, the optimal parameters of σ^2 and $\boldsymbol{\lambda}$ are found by maximizing the marginal likelihood, and the resulting EM type update rule is

$$\lambda_j \longleftarrow \frac{1}{\int_0^{+\infty} d\mathbf{w}\, w_j Q(\mathbf{w})} \quad j = 1, 2, ..., M, \tag{3.78}$$

$$\text{and} \quad \sigma^2 \longleftarrow \frac{1}{N}\int_0^{+\infty} d\mathbf{w}\, \|\mathbf{y} - \boldsymbol{\Phi}\mathbf{w}\|^2 Q(\mathbf{w}), \tag{3.79}$$

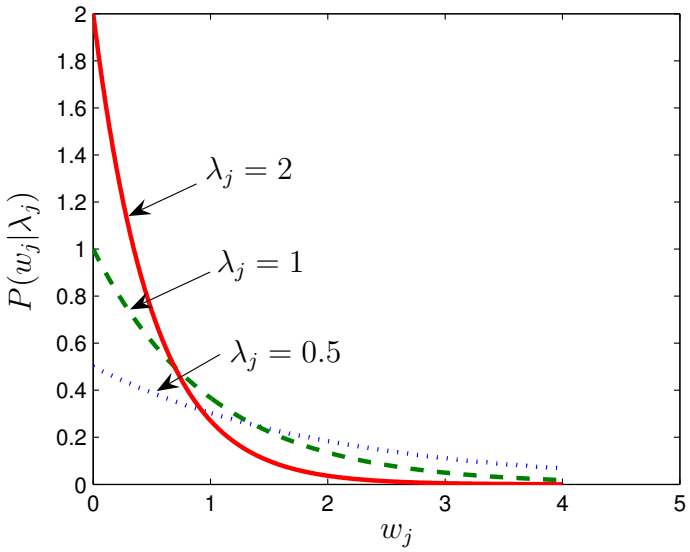


Figure 3.13: Exponential distribution $P(w_j|\lambda_j) = \lambda_j \exp\{-\lambda_j|w_j|\}, w_j \geq 0$.

.

where the expectations are taken over the distribution

$$Q(\mathbf{w}) = \frac{1}{\mathcal{Z}_w} \exp[-F(\mathbf{w})] \tag{3.80}$$

with normalization constant $\mathcal{Z}_w = \int_0^{+\infty} d\mathbf{w} \ \exp[-F(\mathbf{w})]$, and

$$F(\mathbf{w}) = \frac{1}{2\sigma^2}\|\mathbf{y} - \mathbf{\Phi}\mathbf{w}\|^2 + \sum_{j=1}^{M} \lambda_j w_j, \quad \mathbf{w} \geq \mathbf{0}. \tag{3.81}$$

Note that $\mathbf{w}$ here is with nonnegative constraint. In order to evaluate the integrals in Eqs. 3.78 and 3.79 given the current estimate of σ^2 and $\boldsymbol{\lambda}$, we again need to approximate the $Q(\mathbf{w})$ distribution around its mode $\mathbf{w}^{MP}$ that minimizes $F(\mathbf{w})$. The minimization is a nonnegative quadratic programming problem, and it can be solved by multiplicative update algorithm, Merhotra predictor-corrector primal-dual interior method, or projected gradient descent algorithm that were described in Section 3.1.2.

After the $\mathbf{w}^{MP}$ is derived, we again choose to approximate $Q(\mathbf{w})$ as $Q(\mathbf{w}) \approx$

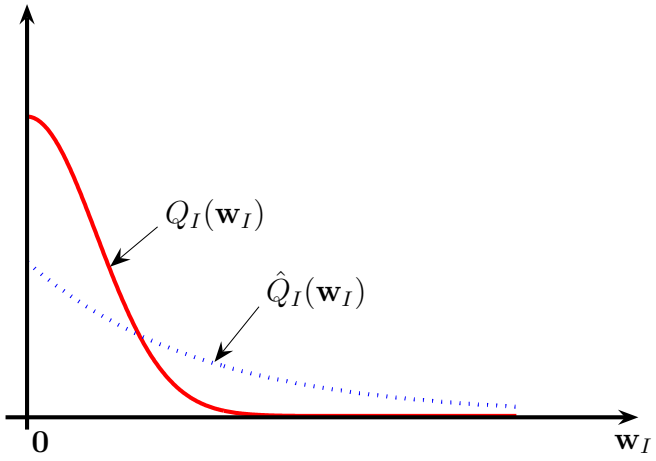


Figure 3.14: $\hat{Q}_I(\mathbf{w}_I)$ for approximating $Q_I(\mathbf{w}_I),\ \mathbf{w}_I \geq 0$. $Q_I(\mathbf{w}_I)$ has its mode at 0.

$Q_I(\mathbf{w}_I)Q_J(\mathbf{w}_J)$ where $I = \{i : w_i^{MP} = 0\}$, $J = \{j : w_j^{MP} \neq 0\}$, and $Q_J(\mathbf{w}_J)$ is a joint Gaussian distribution with mean $\mathbf{w}_J^{MP}$ and covariance $[(\frac{1}{\sigma^2}\mathbf{\Phi}^T\mathbf{\Phi})_{JJ}]^{-1}$. For $Q_I(\mathbf{w}_I)$, we approximate it with factorized exponential distribution (as shown in Fig. 3.14), namely

$$Q_I(\mathbf{w}_I) \approx \hat{Q}_I(\mathbf{w}_I) = \prod_{i \in I} \hat{Q}_i(w_i), \tag{3.82}$$

with

$$\hat{Q}_i(w_i) = \frac{1}{\mu_i} e^{-w_i/\mu_i}, \quad w_i \geq 0, \tag{3.83}$$

where $\boldsymbol{\mu} = \{\mu_i\}_{i \in I}$ are variational parameters that can be found by minimizing the KL divergence $D_{KL}[\hat{Q}_I(\mathbf{w}_I) \| Q_I(\mathbf{w}_I)]$. Using the following statistics of $\hat{Q}_i(\mathbf{w}_i)$ distribution,

$$\langle w_i \rangle = \mu_i; \tag{3.84}$$

$$\langle w_i w_j \rangle = \mu_i \mu_j \quad i \neq j; \tag{3.85}$$

$$\langle w_i^2 \rangle = 2\mu_i^2, \tag{3.86}$$

it can be shown that minimizing the KL divergence is to solving the following minimization

problem

$$\boldsymbol{\mu}^* = \arg\min_{\boldsymbol{\mu}\geq 0} \frac{1}{2}\boldsymbol{\mu}^T\hat{\mathbf{A}}\boldsymbol{\mu} + \hat{\mathbf{b}}^T\boldsymbol{\mu} - \sum_i \ln \mu_i, \tag{3.87}$$

where $\hat{\mathbf{b}} = (\mathbf{A}\mathbf{w}^{MP} + \mathbf{b} + \boldsymbol{\lambda})_I$, and $\hat{\mathbf{A}} = \mathbf{A}_{II} + \text{diag}(\mathbf{A}_{II})$, where $\mathbf{A}_{II}$ is the sub-matrix of $\mathbf{A} = \frac{1}{\sigma^2}\boldsymbol{\Phi}^T\boldsymbol{\Phi}$, and $\text{diag}(\mathbf{A}_{II})$ denoting the diagonal matrix whose diagonal is the diagonal of $\mathbf{A}_{II}$. The minimization in Eq. 3.87 can be solved by either Newton's methods or the multiplicative updated shown in Eq. 3.56.

After the variational parameters $\boldsymbol{\mu}^*$ are derived, the mean $\bar{\mathbf{w}}$ and covariance $\mathbf{C}$ of $\mathbf{w}$ under the approximated distribution $Q_J(\mathbf{w}_J)\hat{Q}_I(\mathbf{w}_I)$can be computed:

$$\bar{w_i} = \begin{cases} w_i^{ML} & \text{if} \quad i \in J \\ \mu_i{}^* & \text{if} \quad i \in I \end{cases}, \tag{3.88}$$

$$C_{ij} = \begin{cases} (\mathbf{A_{JJ}}^{-1})_{ij} & \text{if } i,j \in J \\ \delta_{ij}(\mu_i^{*2}) & \text{otherwise.} \end{cases} \tag{3.89}$$

And then, the update rule for $\boldsymbol{\lambda}$ and σ^2 in Eqs. 3.78 and 3.79 becomes:

$$\lambda_j \longleftarrow \frac{1}{\bar{w_j}}, \quad j = 1, 2, ..., M \tag{3.90}$$

$$\sigma^2 \longleftarrow \frac{1}{N}[(\mathbf{y} - \boldsymbol{\Phi}\bar{\mathbf{w}})^T(\mathbf{y} - \boldsymbol{\Phi}\bar{\mathbf{w}}) + \text{Tr}(\boldsymbol{\Phi}^T\boldsymbol{\Phi}\mathbf{C})]. \tag{3.91}$$

Similarly, we can also develop a Bayesian framework for inferring the optimal scalar regularization parameter λ' in the uniform l_1-norm regularized nonnegative least squares:

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{y} - \boldsymbol{\Phi}\mathbf{w}\|_2^2 + \lambda' \sum_{j=1}^{M} w_j \tag{3.92}$$
$$\text{subject to}: \ \mathbf{w} \geq 0.$$

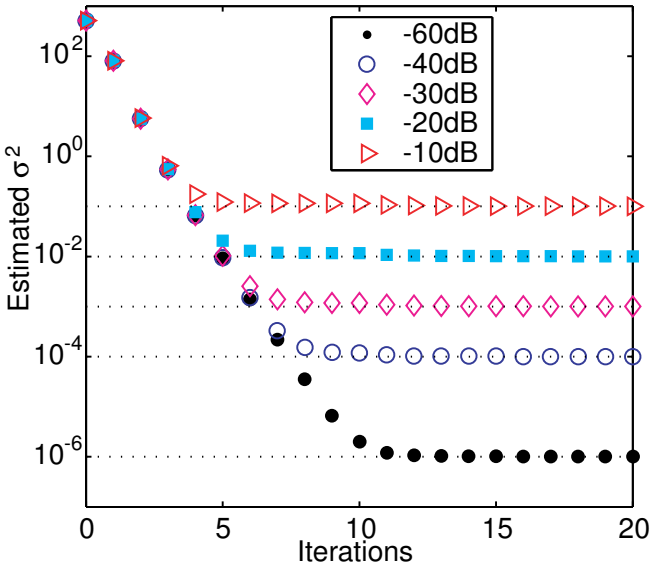


Figure 3.15: Estimate of σ^2 at each iteration of the BRAND algorithm for signals with varying levels of noise. Uniform regularization was employed for the first 10 iterations, followed by another 10 iterations of independent regularization.

3.2.2 A simulated time delay estimation example with noise

Now let's continue to look at the time delay estimation example that shown in Fig.3.11 and Fig. 3.12 but with microphone signal being corrupted by varying levels of Gaussian noise. To identify the filter given the source signal and noisy microphone observations, we employed the BRAND algorithm, which optimizes the optimization in Eq. 3.76 with the optimal regularization parameters $\boldsymbol{\lambda}' = \sigma^2\boldsymbol{\lambda}$ being inferred in the Bayseian framework presented in Section 3.2.1. Fig. 3.15 illustrates that BRAND is able to quickly and consistently estimate the true noise level σ^2 even with bad initial estimates.

Fig. 3.16 illustrates the need for the BRAND algorithm to infer the optimal setting of the regularization parameters. When the measured signal is contaminated with -10 dB ambient Gaussian white noise, different regularization strategies can lead to different filter estimates. With no regularization, the added noise causes the deconvolution solution to exhibit several small spurious peaks. However, manually setting too large of a regularization

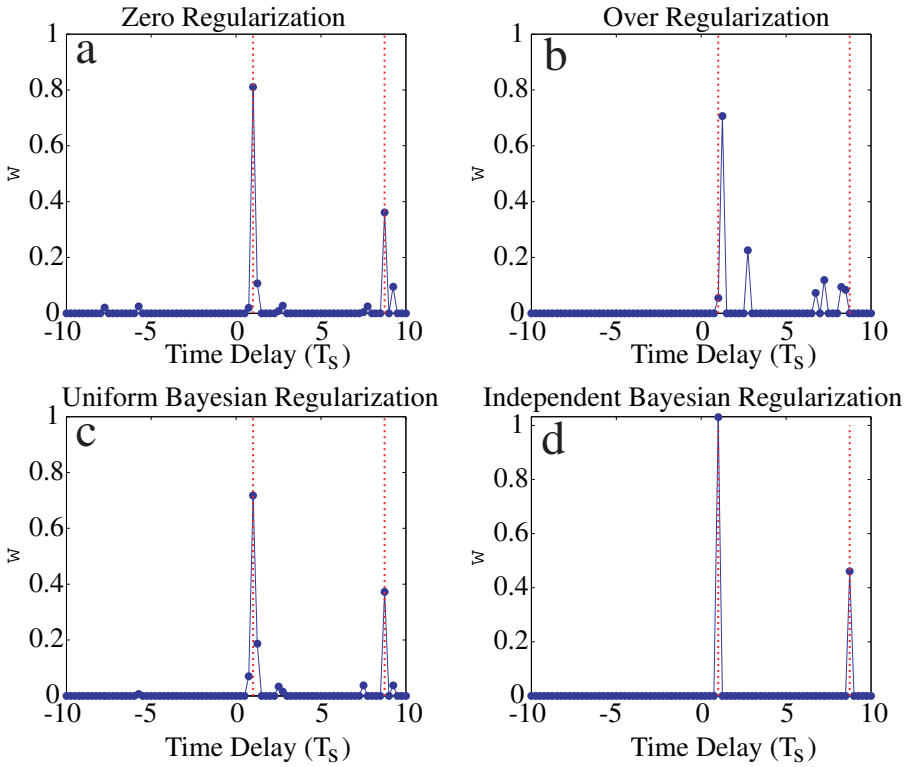


Figure 3.16: Nonnegative deconvolution results under different l_1-norm regularizations, when the measured signal is contaminated by -10 dB noise: a) zero regularization, b) manually set over-regularization, c) uniform Bayesian regularization, d) independent Bayesian regularization.

causes the time delay estimates to deviate from the true room impulse response structure. The uniform Bayesian regularization strategy is much better with regards to estimating the true sparse structure of the filter, but the absolute magnitudes of the filter coefficients are under-estimated. In contrast, the independent Bayesian regularization in BRAND leads to an almost perfect identification of the room impulse response, with the appropriate sparse structure and filter coefficients being optimally estimated.

3.3 l_1-norm sparse Bayesian learning for other problems

Besides l_1-norm regularized least squares problems, the Bayesian framework can also be utilized for learning the optimal regularization parameters in some other l_1-norm regularized problems, such as logistic regression and learning sparse Markov network. Here, we only describes those problems and leave most details for our future work.

Logistic regression is a popular probabilistic approach for classification. For K-class linear regression and given data $\{\mathbf{x}_i, y_i\}_{i=1}^N$ where $\mathbf{x}_i \in \mathbb{R}^M$ and $y_i \in \{1, 2, ..., K\}$, the task of logistic regression is to find weight vectors $\{\mathbf{w}_k\}_{k=1}^K$ that maximize the following log-likelihood

$$\mathcal{L}(\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_K) = \sum_{i=1}^{N} [\mathbf{w}_{y_i}^T \mathbf{x}_i - \ln \sum_{k=1}^{K} \exp(\mathbf{w}_k^T \mathbf{x}_i)], \tag{3.93}$$

where one of $\mathbf{w}_k$ may be set to zero because of solution degeneracy. Now, if the weight vectors are desired to be sparse, we can choose to enforce l_1-norm regularization on the weights, namely

$$\mathbf{w}_1^*, \mathbf{w}_2^*, ..., \mathbf{w}_K^* = \arg \min_{\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_K} \sum_{i=1}^{N} [-\mathbf{w}_{y_i}^T \mathbf{x}_i + \ln \sum_{k=1}^{K} \exp(\mathbf{w}_k^T \mathbf{x}_i)] + \sum_{k=1}^{K} \sum_{j=1}^{M} \lambda_{j,k} |w_{j,k}|, \tag{3.94}$$

where $w_{j,k}$ is the j^{th} element of $\mathbf{w}_k$, and $\{\boldsymbol{\lambda}_k = [\lambda_{1,k}; \lambda_{2,k}; ...; \lambda_{M,k}]\}_{k=1}^{M}$ are sparsity regularization parameters to be learned in a Bayesian framework. Given the regularization parameters, the optimization in Eq. 3.94 is convex with respect to the weight vectors, and it can be solved by a variety of algorithms [62] [38] [43].

To infer the optimal setting of the regularization parameters, the probabilistic model for Eq. 3.94 consists of data model

$$P(y_i|\mathbf{x}_i, \mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_K) = \frac{\exp(\mathbf{w}_{y_i}^T \mathbf{x}_i)}{\sum_{k=1}^{K} \exp(\mathbf{w}_k^T \mathbf{x}_i)} \tag{3.95}$$

and sparsity prior on the weights

$$P(\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_K|\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, ..., \boldsymbol{\lambda}_K) = \prod_{k=1}^{K} \prod_{j=1}^{M} \frac{\lambda_{j,k}}{2} \exp(-\lambda_{j,k}|w_{j,k}|). \tag{3.96}$$

Similar to the Bayesian formulation in Section 3.1, the optimal regularization parameters can be found by maximizing the marginal likelihood, and the maximization can be solved by the following EM update

$$\lambda_j \longleftarrow \frac{1}{\int_{-\infty}^{+\infty} d\mathbf{w}\, |w_j| Q(\mathbf{w})} \quad j = 1, 2, ..., M, \tag{3.97}$$

where the expectations are taken over the distribution

$$Q(\mathbf{w}) = \frac{1}{\mathcal{Z}_w} \exp[-F(\mathbf{w})] \tag{3.98}$$

where

$$F(\mathbf{w}) = \sum_{i=1}^{N} [-\mathbf{w}_{y_i}^T \mathbf{x}_i + \ln \sum_{k=1}^{K} \exp(\mathbf{w}_k^T \mathbf{x}_i)] + \sum_{k=1}^{K} \sum_{j=1}^{M} \lambda_{j,k}|w_{j,k}| \tag{3.99}$$

and $\mathcal{Z}_w = \int_{-\infty}^{+\infty} d\mathbf{w}\ \exp[-F(\mathbf{w})]$ is a normalization factor.

Therefore, learning the optimal regularization parameters is down to how to evaluate the integral in Eq. 3.97. This may be done by Markov chain Monte Carlo approach or variational approaches. It is worth noting that there exists nice quadratic upper bound to the soft-max function $\ln \sum_{k=1}^{K} \exp(\mathbf{w}_k^T \mathbf{x}_i)$ [9]. Therefore, the quadratic bound may replace the logistic data likelihood in $F(\mathbf{w})$, and we may proceed the variational approximation using the formulation developed in Section 3.1.

Another example of l_1-norm regularized problem is about fitting sparse Gaussian Markov model [6]. Given empirical data covariance matrix $\mathbf{S}$, the optimization for fitting the data with a sparse precision matrix (or the inverse of covariance matrix) $\mathbf{X}^*$ is

$$\mathbf{X}^* = \arg\min_{\mathbf{X}} -\log\det(\mathbf{X}) + \text{Tr}(\mathbf{S}^T\mathbf{X}) + \sum_{i,j:i\neq j} \lambda_{ij}|X_{ij}| \tag{3.100}$$

where $\mathbf{\Lambda} = \{\lambda_{ij}\}_{i,j:i\neq j}$ are to be learned in a Bayesian framework. Similarly, the l_1-norm sparse Bayesian learning may also be utilized to extend the existing work in [44] and [68] and discover the structures of sparse Markov networks under a Bayesian framework.

3.4 Discussion

We have described the l_1-norm sparse Bayesian learning for ordinary least squares and nonnegative least squares. Simulation results have demonstrated that the l_1-norm sparse Bayesian learning approach is able to accurately resolve the true sparse solutions even in very noisy data, and it provides much better performance than both the conventional l_1-norm sparsity regularization approaches and the l_2-norm Bayesian sparse learning (also known as relevance vector machine).

We also briefly mentioned the l_1-norm sparse Bayesian learning for other problems like logistic regression and finding sparse Markov network, and that is part of our ongoing and

future work.

With the powerful tool in our hands, we look for employing l_1-norm sparse Bayesian learning for solving challenging real problems. In the following two chapters, we show how the l_1-norm sparse Bayesian learning can be used for acoustic blind channel identification and provides dramatic improvement in both speech dereverberation and time difference of arrival estimation in reverberant environments compared to their conventional methods.

Chapter 4

Application I : blind channel identification for speech dereverberation

4.1 Introduction

Speech dereverberation, which may be viewed as a denoising technique, is crucial for many speech related applications, such as hands-free teleconferencing and automatic speech recognition. It is a challenging signal processing task and remains an open problem after more than three decades of research. Although many approaches [55] have been developed for speech dereverberation, blind channel identification (BCI) is believed to be the key to thoroughly solving the dereverberation problem. Most BCI approaches rely on source statistics (higher order statistics [35] or statistics of LPC coefficients [5]), or spatial difference among multiple channels [67] for resolving solution degeneracies due to the lack of knowledge of the source. The performance of these approaches depends on how well they model real acoustic systems (mainly sources and channels). The BCI approaches using source statistics need a long sequence of data to build up the statistics, and their performance often degrades significantly in real acoustic environments where acoustic systems

are time-varying and only approximately time-invariant during a short time window. Besides the data efficiency issue, there are some other difficulties in the BCI approaches using source statistics, for example, non-stationarity of a speech source, whitening side effect, and non-minimum phase of a filter [35]. In contrast, the BCI approaches exploiting channel spatial difference are blind to the source, and thus they avoid those difficulties arising in assuming source statistics. Unfortunately, these approaches are often too ill-conditioned to tolerate even a very small amount of ambient noise. In general, BCI for speech dereverberation is an active research area, and the main challenge is how to build an effective acoustic model that not only can resolve solution degeneracies due to the lack of knowledge of the source, but also robustly models real acoustic environments.

To address the challenge, we propose a *sparse acoustic room impulse response (RIR) model* for BCI, that is, an acoustic RIR can be modeled by a *sparse* FIR filter. The sparse RIR model is theoretically sound [4], and it has been shown to be useful for estimating RIRs in real acoustic environments when the source is given *a priori* [24]. In our proposal, the sparse RIR model is incorporated with channel spatial difference, resulting a *blind sparse channel identification* (BSCI) approach for a single-input multiple-output (SIMO) acoustic system. The BSCI approach aims to resolve some of the difficulties in conventional BCI approaches. It is blind to the source and therefore avoids the difficulties arising in assuming source statistics. Meanwhile, the BSCI approach is expected to be robust to ambient noise. It has been shown that, when the source is given *a priori* [47], the prior knowledge about sparse RIRs plays an important role in robustly estimating RIRs in noisy acoustic environments. Furthermore, the statistics describing the sparseness of RIRs are governed by acoustic room characteristics, and thus they are close to be stationary with respect to a specific room. This is advantageous in terms of both learning the statistics and applying them in channel identification.

Based on the cross relation formulation [67] of BCI, we develop a BSCI algorithm that

incorporates the sparse RIR model. Our choice for enforcing sparsity is l_1-norm regularization, as advocated in the previous chapters. In the context of BCI, two important issues need to be addressed when using l_1-norm regularization. First, the existing cross relation formulation for BCI is nonconvex, and directly enforcing l_1-norm regularization will result in an intractable optimization. Second, l_1-norm regularization parameters are critical for deriving correct solutions, and their improper setting may lead to totally irrelevant solutions. To address these two issues, we shows how to formulate the BCI of a SIMO system into a *convex* optimization, indeed an unconstrained least squares (LS) problem, which provides a flexible platform for incorporating l_1-norm regularization; it also shows how to infer the *optimal* l_1-norm regularization parameters directly from microphone observations under a Bayesian framework with some slight modifications from the one presented in Chapter 3.

We evaluate the proposed BSCI approach using both simulations and experiments in real acoustic environments. Simulation results illustrate the effectiveness of the proposed sparse RIR model in resolving solution degeneracies, and they show that the BSCI approach is able to robustly and accurately identify filters from noisy microphone observations. When applied to speech dereverberation in real acoustic environments, the BSCI approach yields source estimates with high fidelity to anechoic chamber measurements. All of these demonstrate that the BSCI approach has the potential for solving the difficult speech dereverberation problem.

4.2 Blind sparse channel identification (BSCI)

4.2.1 Previous work

Our BSCI approach is based on the cross relation formulation for blind SIMO channel identification [67]. In a one-speaker two-microphone system, the microphone signals at

time k can be written as:

$$x_i(k) = s(k) * h_i + n_i(k), \quad i = 1, 2, \tag{4.1}$$

where $*$ denotes linear convolution, $s(k)$ is a source signal, h_i represents the channel impulse response between the source and the ith microphone, and $n_i(k)$ is ambient noise. The cross relation formulation is based on a clever observation, $x_2(k) * h_1 = x_1(k) * h_2 = s(k) * h_1 * h_2$, if the microphone signals are noiseless [67]. Then, without requiring any knowledge from the source signal, the channel filters can be identified by minimizing the squared cross relation error. In matrix-vector form, the optimization can be written as

$$\mathbf{h}_1^*, \mathbf{h}_2^* = \underset{\|\mathbf{h}_1\|^2+\|\mathbf{h}_2\|^2=1}{\arg\min} \frac{1}{2}\|\mathbf{X}_2\mathbf{h}_1 - \mathbf{X}_1\mathbf{h}_2\|^2 \tag{4.2}$$

where $\mathbf{X}_i$ is the $(N + L - 1) \times L$ convolution Toeplitz matrix whose first row and first column are $[x_i(k - N + 1), 0, \ldots, 0]$ and $[x_i(k - N + 1), x_i(k - N + 2), ..., x_i(k), 0, \ldots, 0]^T$, respectively, N is the microphone signal length, L is the filter length, $\mathbf{h}_i (i = 1, 2)$ are $L \times 1$ vectors representing the filters, $\|\cdot\|$ denotes l_2-norm, and the constraint is to avoid the trivial zero solution. It is easy to see that the above optimization is a minimum eigenvalue problem, and it can be solved by eigenvalue decomposition. As shown in [67], the eigenvalue decomposition approach finds the true solution within a constant time delay and a constant scalar factor when 1) the system is noiseless; 2) the two filters are co-prime (namely, no common zeros); and 3) the system is sufficiently excited (i.e., the source needs to have enough frequency bands).

Unfortunately, the eigenvalue decomposition approach has not been demonstrated to be useful for speech dereverberation in real acoustic environments. This is because the conditions for finding true solutions are difficult to sustain. First, microphone signals in real acoustic environments are always immersed in excessive ambient noise (such as air-

conditioning noise), and thus the noiseless assumption is never true. Second, it requires precise information about filter order for the filters to be co-prime, however, the filter order itself is hard to compute accurately since the filters modeling RIRs are often thousands of taps long. As a result, eigenvalue decomposition approach is often ill-conditioned and very sensitive to even a very small amount of ambient noise.

Our proposed sparse RIR model aims to alleviate those difficulties. Under the sparse RIR model, sparsity regularization automatically determines filter order since surplus filter coefficients are forced to be zero. Furthermore, previous work [47] has demonstrated that, when the source is given *a priori*, sparsity regularization plays an important role in robustly estimating RIRs in noisy acoustic environments. In order to exploit the sparse RIR model, we first formulate the BCI using cross relation into a *convex* optimization, which will provide a flexible platform for enforcing l_1-norm sparsity regularization.

4.2.2 Convex formulation

The optimization in Eq. 4.2 is nonconvex because its domain, $\|\mathbf{h}_1\|^2 + \|\mathbf{h}_2\|^2 = 1$, is nonconvex. We propose to replace it with a *convex* singleton linear constraint, and the optimization becomes

$$\mathbf{h}_1^*, \mathbf{h}_2^* = \underset{h_1(l)=1}{\arg\min} \frac{1}{2}\|\mathbf{X}_2\mathbf{h}_1 - \mathbf{X}_1\mathbf{h}_2\|^2 \tag{4.3}$$

where $h_1(l)$ is the lth element of filter $\mathbf{h}_1$. It is easy to see that, when microphone signals are noiseless, the optimizations in Eqs. 4.2 and 4.3 yield equivalent solutions within a constant time delay and a constant scalar factor. Because the optimization is a minimization, $h_1(l)$ tends to align with the largest coefficient in filter $\mathbf{h}_1$, which normally is the coefficient corresponding to the direct path. Consequently, the singleton linear constraint removes two degrees of freedom in filter estimates: a constant time delay (by fixing l) and a constant

scalar factor [by fixing $h_1(l) = 1$]. The choice of l ($0 \leq l \leq L-1$) is arbitrary as long as the direct path in filter $\mathbf{h}_2$ is no more than l samples earlier than the one in filter $\mathbf{h}_1$.

The new formulation in Eq. 4.3 has many advantages. It is convex and indeed an unconstrained LS problem since the singleton linear constraint can be easily substituted into the objective function. Furthermore, the new LS formulation is more robust to ambient noise than the eigenvalue decomposition approach in Eq. 4.2. This can be better viewed in the frequency domain. Because the squared cross relation error (the objective function in Eqs. 4.2 and 4.3) is weighted in the frequency domain by the power spectrum density of a common source, the total filter energy constraint in Eq. 4.2 may be filled with less significant frequency bands which contribute little to the source and are weighted less in the objective function. As a result, the eigenvalue decomposition approach is very sensitive to noise. In contrast, the singleton linear constraint in Eq. 4.3 has much less coupling in filter energy allocation, and the new LS approach is more robust to ambient noise.

Then, the BSCI approach is to incorporate the LS formulation with l_1-norm sparsity regularization, and the optimization becomes

$$\mathbf{h}_1^*, \mathbf{h}_2^* = \underset{h_1(l)=1}{\arg\min} \frac{1}{2}\|\mathbf{X}_2\mathbf{h}_1 - \mathbf{X}_1\mathbf{h}_2\|^2 + \lambda' \sum_{j=0}^{L-1}[|h_1(j)| + |h_2(j)|] \tag{4.4}$$

where λ' is a nonnegative scalar regularization parameter that balances the preference between the squared cross relation error and the sparseness of solutions described by their l_1-norm. The setting of λ' is critical for deriving appropriate solutions, and we will show how to compute its optimal setting in a Bayesian framework in Section 4.2.3. Given a λ', the optimization in Eq. 4.4 is *convex* and can be solved by various methods with guaranteed global convergence. We implemented the *Mehrotra predictor-corrector primal-dual interior point method* [69], which is known to yield better search directions than the Newton's method. Our implementation usually solves the optimization in Eq. 4.4 with extreme

accuracy (relative duality gap less than 10^{-14}) in less than 20 iterations.

4.2.3 Bayesian l_1-norm sparse learning for blind channel identification

The l_1-norm regularization parameter λ' in Eq. 4.4 is critical for deriving appropriately sparse solutions. How to determine its optimal setting is still an open research topic. We mentioned in Section 2.2.3 in Chapter 2 that λ' may be determined by cross-validation. However, it is not easy to obtain extra data for cross-validation in BCI since real acoustic environments are often time-varying. In this study, we develop a Bayesian framework for inferring the *optimal* regularization parameters for the BSCI formulation in Eq. 4.4. A similar Bayesian framework can be found in Section 3.1 in Chapter 3, where the source was assumed to be known *a priori*.

The optimization in Eq. 4.4 is a *maximum-a-posteriori* estimation under the following probabilistic assumptions

$$P\left(\mathbf{X}_2\mathbf{h}_1 - \mathbf{X}_1\mathbf{h}_2|\sigma^2, \mathbf{h}_1, \mathbf{h}_2\right) = \frac{1}{(2\pi\sigma^2)^{(N+L-1)/2}} \exp\left\{-\frac{1}{2\sigma^2}\|\mathbf{X}_2\mathbf{h}_1 - \mathbf{X}_1\mathbf{h}_2\|^2\right\} \quad (4.5)$$

$$P\left(\mathbf{h}_1, \mathbf{h}_2|\lambda\right) = \left(\frac{\lambda}{2}\right)^{2L} \exp\left\{-\lambda \sum_{j=0}^{L-1}[|h_1(j)| + |h_2(j)|]\right\} \quad (4.6)$$

where the cross relation error is an I.I.D. zero-mean Gaussian with variance σ^2, and the filter coefficients are governed by a Laplacian sparse prior with the scalar parameter λ. Then, the regularization parameter λ' in Eq. 4.4 can be written as

$$\lambda' = \sigma^2\lambda. \quad (4.7)$$

When the ambient noise [$n_1(k)$ and $n_2(k)$ in Eq. 4.1] is an I.I.D. zero-mean Gaussian

with variance σ_0^2, the parameter σ^2 can be approximately written as

$$\sigma^2 = \sigma_0^2(\|\mathbf{h}_1\|^2 + \|\mathbf{h}_2\|^2), \tag{4.8}$$

because $x_2(k) * h_1 - x_1(k) * h_2 = n_2(k) * h_1 - n_1(k) * h_2$. The above form of σ^2 is only an approximation because the cross relation error is temporally correlated through the convolution. Nevertheless, since the cross relation error is the result of the convolutive mixing, its distribution will be close to the Gaussian with its variance described by Eq. 4.8 according to the central limit theorem. We choose to estimate the ambient noise level (σ_0^2) directly from microphone observations via restricted maximum likelihood [33]:

$$\sigma_0^2 = \min_{\mathbf{s},\mathbf{h}_1,\mathbf{h}_2} \frac{1}{N-L-1} \sum_{i=1}^{2} \sum_{k=0}^{N-1} \|x_i(k) - s(k) * h_i\|^2 \tag{4.9}$$

where the denominator $N - L - 1$ (but not $2N$) accounts for the loss of the degrees of freedom during the optimization. The above minimization is solved by coordinate descent alternatively with respect to the source and the filters. It is initialized with the LS solution by Eq. 4.3 and often able to yield a good σ_0^2 estimate in a few iterations. Note that each iteration can be computed efficiently in the frequency domain. Meanwhile, the parameter λ can be computed by

$$\lambda = \frac{2L}{\sum_{j=0}^{L-1}[|h_1(j)| + |h_2(j)|]}, \tag{4.10}$$

as a result of finding the optimal Laplacian distribution given its sufficient statistics.

With the Eqs. 4.8 and 4.10, finding the optimal regularization parameters becomes computing the statistics of filters, $\|\mathbf{h}_1\|^2 + \|\mathbf{h}_2\|^2$ and $\sum_{j=0}^{L-1}[|h_1(j)| + |h_2(j)|]$. These statistics are closely related to acoustic room characteristics and may be computed from them if they are known *a priori*. For example, the reverberation time of a room defines how fast echoes decay -60 dB, and it can be used to compute the filter statistics. More gen-

erally, we choose to compute the statistics directly from microphone observations in the Baysian framework by maximizing the *marginal likelihood*, $P(\mathbf{X}_2\mathbf{h}_1 - \mathbf{X}_1\mathbf{h}_2|\sigma^2, \lambda) = \int_{h_1(l)=1} P(\mathbf{X}_2\mathbf{h}_1 - \mathbf{X}_1\mathbf{h}_2, \mathbf{h}_1, \mathbf{h}_2|\sigma^2, \lambda)\mathbf{dh}_1\mathbf{dh}_2$. The optimization is through Expectation-Maximization (EM) updates:

$$\sigma^2 \longleftarrow \sigma_0^2 \int_{h(l)=1} (\|\mathbf{h}_1\|^2 + \|\mathbf{h}_2\|^2) Q(\mathbf{h}_1, \mathbf{h}_2)\mathbf{dh}_1\mathbf{dh}_2 \tag{4.11}$$

$$\lambda \longleftarrow \frac{2L}{\int_{h(l)=1}(\sum_{j=0}^{L-1} |h_1(j)| + |h_2(j)|)Q(\mathbf{h}_1, \mathbf{h}_2)\mathbf{dh}_1\mathbf{dh}_2} \tag{4.12}$$

where $\mathbf{h}_1$ and $\mathbf{h}_2$ are treated as hidden variables, σ^2 and λ are parameters, and $Q(\mathbf{h}_1, \mathbf{h}_2) \propto \exp\{-\frac{1}{2\sigma^2}\|\mathbf{X}_2\mathbf{h}_1 - \mathbf{X}_1\mathbf{h}_2\|^2 - \lambda[\sum_{j=0}^{L-1} |h_1(j)| + |h_2(j)|]\}$ is the probability distribution of $\mathbf{h}_1$ and $\mathbf{h}_2$ given the current estimate of σ^2 and λ. The integrals in Eqs. 4.11 and 4.12 can be computed using the variational scheme described in Section 3.1 in Chapter 3. The EM updates often converge to a good estimate of σ^2 and λ in a few iterations. Moreover, since the filter statistics are relatively stationary for a specified room, the Bayesian inference may be carried out off-line and only once if the room conditions stay the same.

After the filters are identified by BCI approaches, the source can be computed by various methods [54]. We choose to estimate the source by the following optimization

$$\mathbf{s}^* = \arg\min_{\mathbf{s}} \sum_{i=1}^{2} \sum_{k=0}^{N-1} \|x_i(k) - s(k) * h_i\|^2, \tag{4.13}$$

which will yield maximum-likelihood (ML) estimation if the filter estimates are accurate.

4.3 Simulations and Experiments

4.3.1 Simulations

Simulations with artificial RIRs

We first employ a simulated example to illustrate the effectiveness of the proposed sparse RIR model for BCI. In the simulation, we used a speech sequence of 1024 samples (with 16 kHz sampling rate) as the source (s) and simulated two 16-sample FIR filters (h_1 and h_2). The filter h_1 had nonzero elements only at indices 0, 2, and 12 with amplitudes of 1, -0.7, and 0.5, respectively; the filter h_2 had nonzero elements only at indices 2, 6, 8, and 10 with amplitudes of 1, -0.6, 0.6, and 0.4, respectively. Notice that both h_1 and h_2 are sparse. Then the simulated microphone observations (x_1 and x_2) were computed by Eq. 4.1 with the ambient noise being real noise recorded in a classroom. The noise was scaled so that the signal-to-noise ratio (SNR) of the microphone signals was approximately 20 dB. Because a big portion of the noise (mainly air-conditioning noise) was at low frequency, the microphone observations were high-passed with a cut-off frequency of 100 Hz before they were fed to BCI algorithms. In the BSCI algorithm, the l_1-norm regularization parameters, σ^2 and λ, were estimated in the Bayesian framework using the update rules given in Eqs. 4.11 and 4.12.

Figure 4.1 shows the filters identified by different BCI approaches. Compared to the conventional eigenvalue decomposition method (Eq. 4.2), the new convex LS approach (Eq. 4.3) is more robust to ambient noise and yielded better filter estimates even though the estimates still seem to be convolved by a common filter. The proposed BSCI approach (Eq. 4.4) yielded filter estimates that are almost identical to the true ones. It is evident that the proposed sparse RIR model played a crucial role in robustly and accurately identifying filters in blind manners. The robustness and accuracy gained by the BSCI approach will

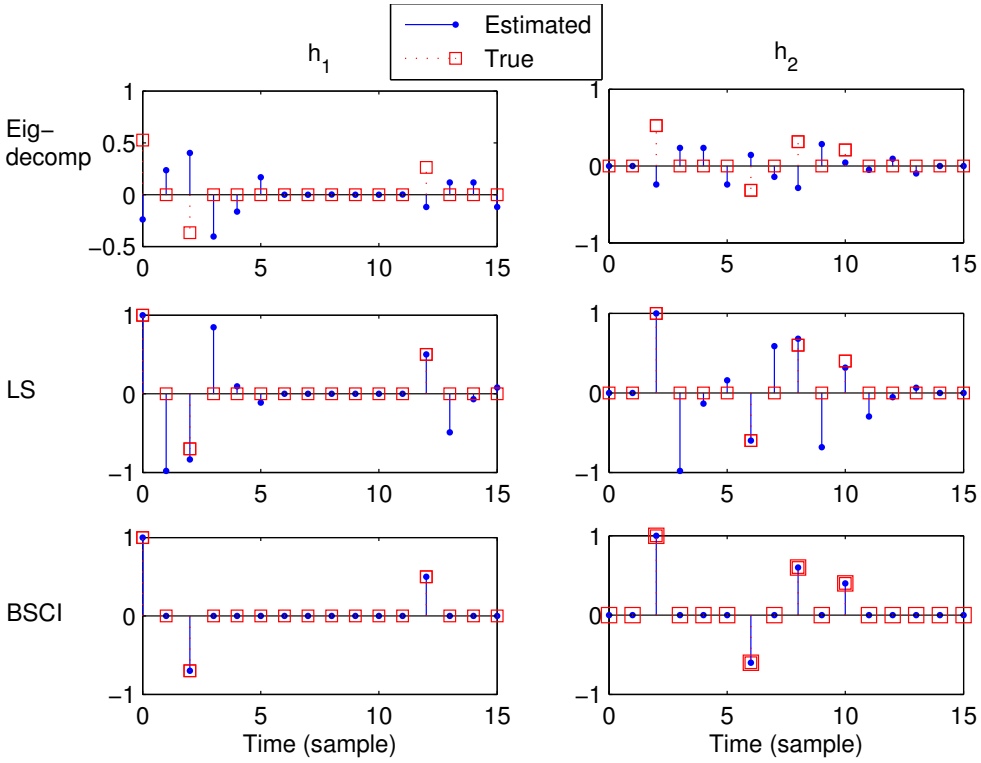


Figure 4.1: Identified filters by three different BCI approaches in a simulated example: the eigenvalue decomposition approach (denoted as eig-decomp) in Eq. 4.2, the LS approach in Eq. 4.3, and the blind sparse channel identification (BSCI) approach in Eq. 4.4. The solid-dot lines represent the estimated filters, and the dot-square lines indicate the true filters within a constant time delay and a constant scalar factor.

become essential when the filters are thousands of taps long in real acoustic environments.

Simulations with measured RIRs

Here we employ simulations using RIRs measured in real rooms to demonstrate the effectiveness of the proposed BSCI approach for speech dereverberation. Its performance is compared to the beamforming, the eigenvalue decomposition (Eq. 4.2), and the LS (Eq. 4.3) approaches. In the simulation, the source sequence (s) was a sentence of speech (approximately 1.5 seconds), and the filters (h_1 and h_2) were two measured RIRs from York MARDY database (http://www.commsp.ee.ic.ac.uk/ sap/mardy.htm) but down-sampled to 16 kHz (from originally 48 kHz). The original filters in the database were not sparse, but they had many tiny coefficients which were in the range of measurement uncertainty. To make the simulated filters sparse, we simply zeroed out those coefficients whose amplitudes were less than 2% of the maximum. Finally, we truncated the filters to have length of 2048 since there were very few nonzero coefficients after that. With the simulated source and filters, we then computed microphone observations using Eq. 4.1 with ambient noise being real noise recorded in a classroom. For testing the robustness of different BCI algorithms, the ambient noise was scaled to different levels so that the SNRs varied from 60 dB to 10 dB. Similar to the previous simulations, the simulated observations were high-passed with a cutoff frequency of 100 Hz before they were fed to different BCI algorithms. In the BSCI approach, the l_1-norm regularization parameters were iteratively computed using the updates in Eqs. 4.11 and 4.12. After filters were identified, the source was estimated using Eq. 4.13.

Because both filter and source estimates by BCI algorithms are within a constant time delay and a constant scalar factor, we use normalized correlation for evaluating the estimates. Let $\hat{\mathbf{s}}$ and $\mathbf{s_0}$ denote an estimated source and the true source, respectively, then the

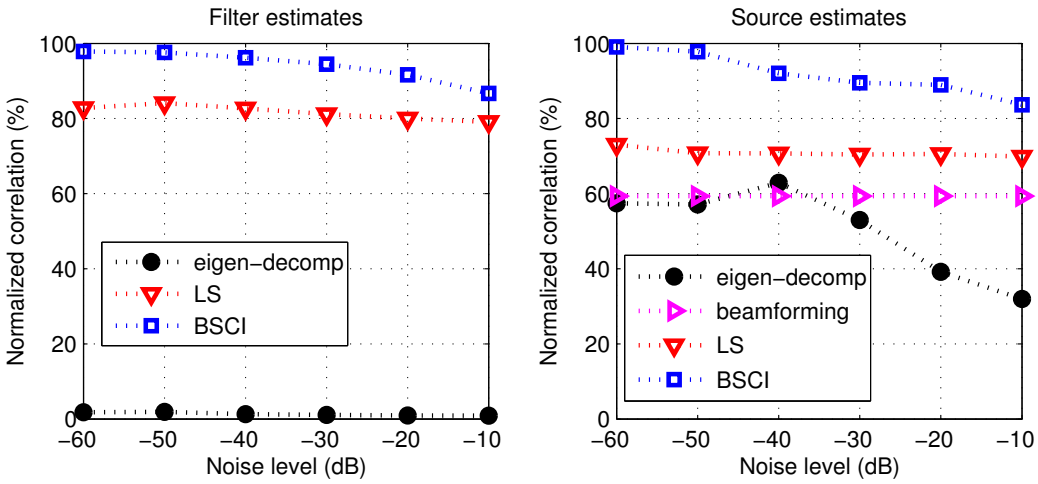


Figure 4.2: The simulation results using measured real RIRs. The normalized correlation (defined in Eq. 4.14) of the estimates were computed with respect to their true values. The filters were identified by three different approaches: the eigenvalue decomposition approach (denoted as eigen-decomp) in Eq. 4.2 , the LS approach in Eq. 4.3, and the blind sparse channel identification (BSCI) approach in Eq. 4.4. After the filters were identified, the source was estimated by Eq. 4.13. The source estimated by beamforming is also presented as a baseline reference.

normalized correlation $C(\hat{\mathbf{s}}, \mathbf{s}_0)$ is defined as

$$C(\hat{\mathbf{s}}, \mathbf{s}_0) = \max_m \frac{\sum_k \hat{s}(k-m)s_0(k)}{\|\hat{\mathbf{s}}\|\|\mathbf{s}_0\|} \tag{4.14}$$

where m and k are sample indices, and $\|\cdot\|$ denotes l_2-norm. It is easy to see that, the normalized correlation is between 0% and 100%: it is equal to 0% when the two signals are uncorrelated, and it is equal to 100% only when the two signal are identical within a constant time delay and a constant scalar factor. The definition in Eq. 4.14 is also applicable to the evaluation of filter estimates.

The simulation results are shown in Fig. 4.2. Similar to what we observed in the previous example, the convex LS approach (Eq. 4.3) shows significant improvement in both filter and source estimation compared to the eigenvalue decomposition approach (Eq. 4.2). In fact, the eigenvalue decomposition approach did not yield relevant results because it

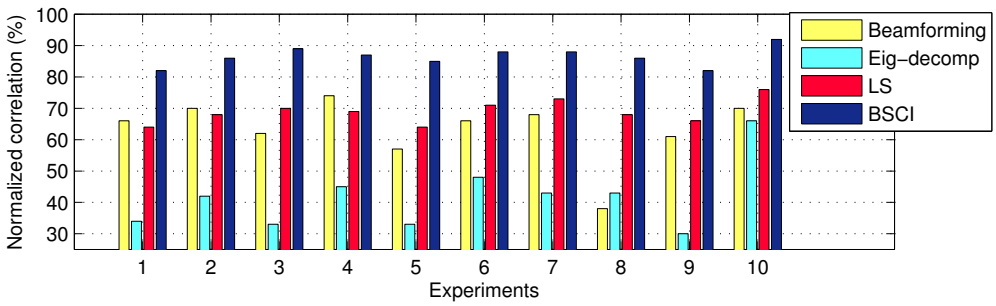


Figure 4.3: The source estimates of 10 experiments in real acoustic environments. The normalized correlation was with respect to their anechoic chamber measurement. The filters were identified by three different BCI approaches: the eigenvalue decomposition approach (denoted as eig-decomp) in Eq. 4.2, the LS approach in Eq. 4.3, and the blind sparse channel identification (BSCI) approach in Eq. 4.4. The beamforming results serve as the baseline performance for comparison.

was too ill-conditioned due to the long filters. The remarkable performance came from the BSCI approach, which incorporates the convex LS formulation with the sparse RIR model. In particular, the BSCI approach yielded higher than 90% normalized correlation in source estimates when SNR was better than 20 dB, and it yielded higher than 99% normalized correlation in the low noise limit. The performance of the canonical delay-and-sum beamforming is also presented as the baseline for all BCI algorithms.

4.3.2 Experiments

We also evaluated the proposed BSCI approach using signals recorded in real acoustic environments. We carried out 10 experiments in total in a reverberant room. In each experiment, a sentence of speech (approximately 1.5 seconds, and the same for all experiments) was played through a loudspeaker (NSW2-326-8A, Aura Sound) and recorded by a matched omnidirectional microphone pair (M30MP, Earthworks). The speaker-microphone positions (and thus RIRs) were different in different experiments. Because the recordings had a large amount of low-frequency noise, they were high-passed with a cutoff frequency of 100 Hz before they were fed to BCI algorithms. In the BSCI approach, the l_1-norm regulariza-

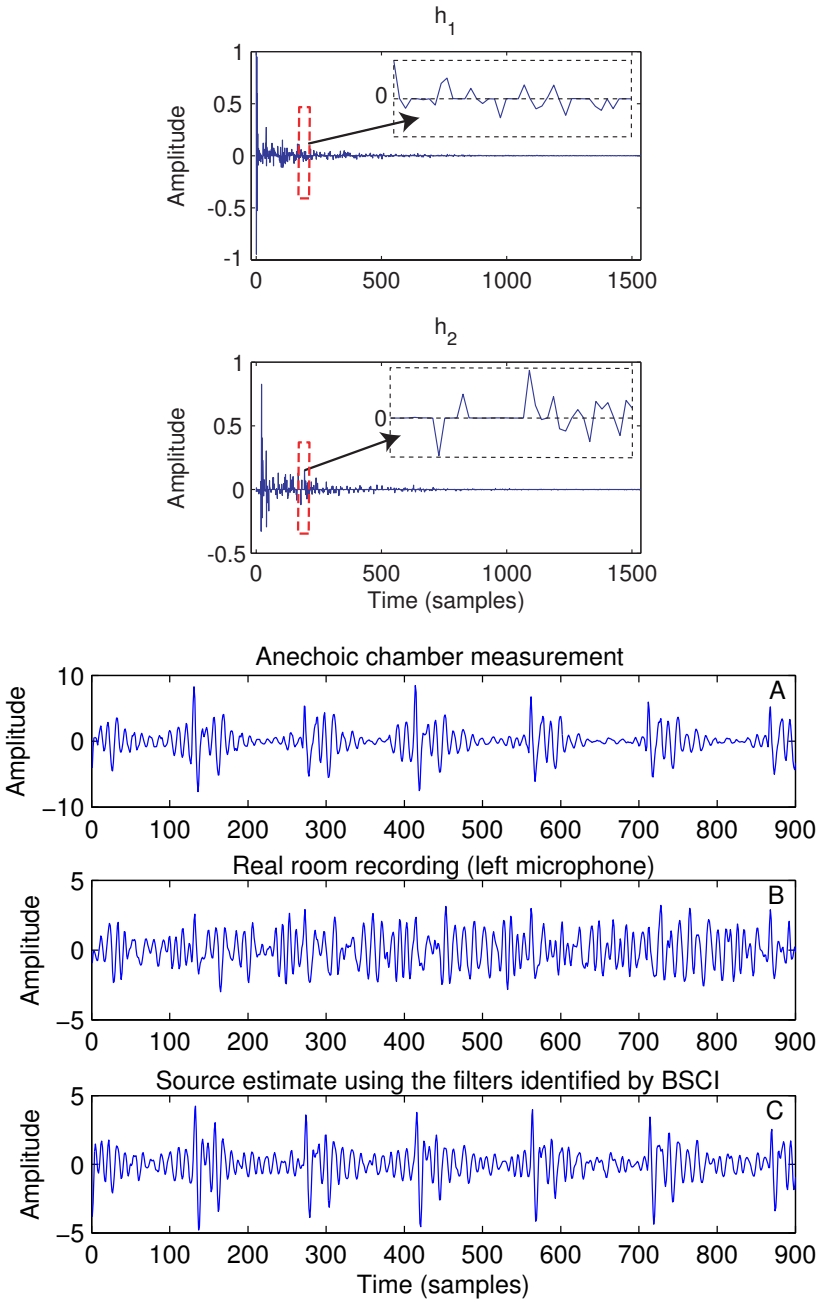


Figure 4.4: Results of Experiment 6 in Fig. 4.3. Top: the filters estimated by the proposed blind sparse channel identification (BSCI) approach. They are sparse as indicated by the enlarged segments. Bottom: a segment of source estimate (shown in C) using the BSCI approach. It is compared with its anechoic measurement (shown in A) and its microphone recording (shown in B).

tion parameters, σ^2 and λ, were iteratively computed using the updates in Eq. 4.11 and 4.12. After the filters were identified, the sources were computed using Eq. 4.13. We also had recordings in the anechoic chamber at Bell Labs using the same instruments and settings, and the anechoic measurement served as the approximated ground truth for evaluating the performance of different BCI approaches.

Figure 4.3 shows the source estimates in the 10 experiments in terms of their normalized correlation to the anechoic measurement. The performance of the proposed BSCI is compared with the beamforming, the eigenvalue decomposition (Eq. 4.2), and the convex LS (Eq. 4.3) approaches. The results of the 10 experiments unanimously support our previous findings in simulations. First, the convex LS approach yielded significantly better source estimates than the eigenvalue decomposition method. Second, the proposed BSCI approach, which incorporates the convex LS formulation with the sparse RIR model, yielded the most dramatic results, achieving 85% or higher of normalized correlation in source estimates in most experiments while the LS approach only obtained approximately 70% of normalized correlation.

Figure 4.4 shows one instance of filter and source estimates. The estimated filters have about 2000 zeros out of totally 3072 coefficients, and thus they are sparse. This observation experimentally validates our hypothesis of the sparse RIR models, namely, an acoustic RIR can be modeled by a sparse FIR filter. The source estimate shown in Fig. 4.4 vividly illustrates the convolution and dereverberation process. It only plots a small segment to reveal greater details. As we see, the anechoic measurement was clean and had clear harmonic structure; the signal recorded in the reverberant room was smeared by echoes during the convolution process; and then, the dereverberation using our BSCI approach deblurred the signal and recovered the underlying harmonic structure.

4.4 Discussion

We propose a *blind sparse channel identification* (BSCI) approach for speech dereverberation. It consists of three important components. The first is the *sparse RIR model*, which effectively resolves solution degeneracies and robustly models real acoustic environments. The second is the *convex formulation*, which guarantees global convergence of the proposed BSCI algorithm. And the third is the *Bayesian l_1-norm sparse learning* scheme that infers the optimal regularization parameters for deriving optimally sparse solutions. The results demonstrate that the proposed BSCI approach holds the potential to solve the speech dereverberation problem in real acoustic environments, which has been recognized as a very difficult problem in signal processing. The acoustic data used in this section are available at http://www.seas.upenn.edu/$\sim$linyuanq/Research.html.

Our future work includes side-by-side comparison between our BSCI approach and existing source statistics based BCI approaches. Our goal is to build a uniform framework that combines various prior knowledge about acoustic systems for best solving the speech dereverberation problem.

Chapter 5

Application II: blind sparse-nonnegative (BSN) channel identification for acoustic time-difference-of-arrival estimation

5.1 Introduction

Time delay estimation [17], which calculates the time-difference-of-arrival (TDOA) between signals received at different microphones, is essential for sound source localization using microphone arrays. The task of TDOA estimation is illustrated in Fig. 5.1. In terms of the underlying model for an acoustic room impulse response (RIR), the existing approaches for TDOA estimation can be classified into two categories: generalized cross-correlation (GCC) approaches and blind channel identification approaches. The GCC approaches approximate an acoustic RIR as a simple delta function, and the TDOA estimation is achieved by maximizing some weighted cross-correlation function with respect to a scalar time difference. An excellent review of this category of approaches can be found in [37]. The GCC approaches do not explicitly take multipath reflections into account and their per-

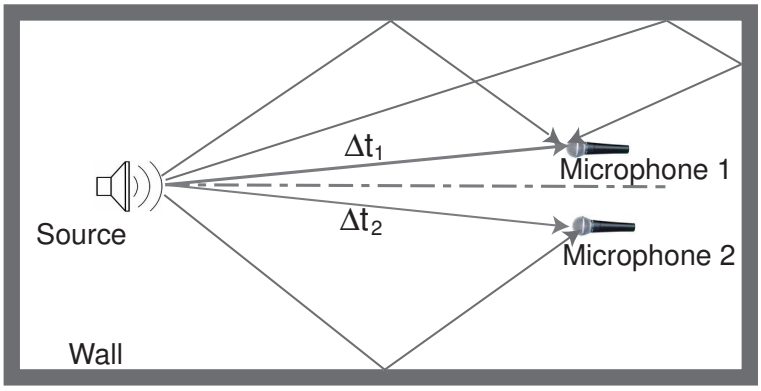


Figure 5.1: Illustration of a single-input two-output acoustic system. A microphone observation consists of a direct path signal, multipath reflections, and ambient noise. The task of TDOA estimation is to estimate the time difference of arrival between the two direct paths, $\Delta t_2 - \Delta t_1$.

formance in reverberant acoustic environments is limited due to the underlying unrealistic RIR model. In contrast, blind channel identification approaches [7] [20] model an acoustic RIR as an FIR filter that includes both a direct path and multipath reflections. In these approaches, after the modeling filters have been identified, the TDOA can be easily computed by examining the direct paths in the filters. By using a more realistic model, the blind channel identification approaches have been shown to be more effective than GCC approaches to reverberation. Unfortunately, blind channel identification approaches have been found to be sensitive to ambient noise. This is because blind channel identification needs to estimate a much more complex model having hundreds or even thousands of parameters (filter coefficients) and is often ill-conditioned due to the nature of blind estimation.

We propose to resolve the noise sensitivity issue in blind channel identification by exploiting prior knowledge about acoustic RIRs. According to many studies [4], an acoustic RIR can be modeled by an FIR filter, which is both *nonnegative* and *sparse* in theory. In practice, nonnegativity and sparsity may not be strictly satisfied due to effects such as low- or high-pass filtering in the propagation media or the imperfect frequency response of a microphone. However, when those effects are common to both channels, they can be viewed

as distortions to a common source. Therefore, the nonnegativity and sparsity assumption are reasonable for real acoustic environments if an acoustic system is appropriately constructed.

The nonnegativity and sparsity priors have been demonstrated to be effective in many signal processing tasks [18]. Our previous work [48] showed that these two priors provided dramatic regularization to the least-mean-square (LMS) problem for identifying acoustic RIRs and improved its robustness to ambient noise when the source was given *a priori*. We show that they play a critical role in *blind* acoustic channel identification for resolving ill-conditioned solutions, which may be caused by overestimating the filter length or insufficient excitation due to the band-limited nature of speech sources [67]. By making the problem better posed, the resulting blind sparse-nonnegative (BSN) channel identification approach is robust to ambient noise. Furthermore, the BSN channel identification approach also allows common preprocessing on the microphone observations to reduce the noise level. In contrast, conventional blind channel identification approaches prohibit preprocessing since they are not able to resolve the preprocessing filtering from filtering by a RIR.

5.2 Blind sparse-nonnegative (BSN) channel identification

The blind sparse-nonnegative (BSN) channel identification is based on the convex formulation in Eq. 4.3 In Chapter 4, which provides a flexible platform for incorporating the nonnegativity and sparsity priors. The optimization for *blind sparse-nonnegative (BSN)*

channel identification becomes

$$\begin{aligned} \mathbf{h}_1^*, \mathbf{h}_2^* &= \arg\min_{\mathbf{h}_1,\mathbf{h}_2} \frac{1}{2}\|\mathbf{X}_2\mathbf{h}_1 - \mathbf{X}_1\mathbf{h}_2\|^2 + \lambda' \sum_{j=0}^{L-1}[h_1(j) + h_2(j)] \\ &\text{subject to } \; h_1(0) = 1, \mathbf{h}_1 \geq 0, \mathbf{h}_2 \geq 0 \end{aligned} \tag{5.1}$$

where the second term is the l_1-norm of the filters, and λ' is the sparsity regularization parameter that balances the preference between the squared fitting error and the sparseness of the solution described by its l_1-norm. As we have described in Section 3.2 in Chapter 3, both nonnegative constraint and l_1-norm regularization are very useful for sparsity regularization. By combining both of them, the optimization in (5.1) is expected to resolve the ill-conditioning problem in blind channel identification and yield solutions that are robust to ambient noise.

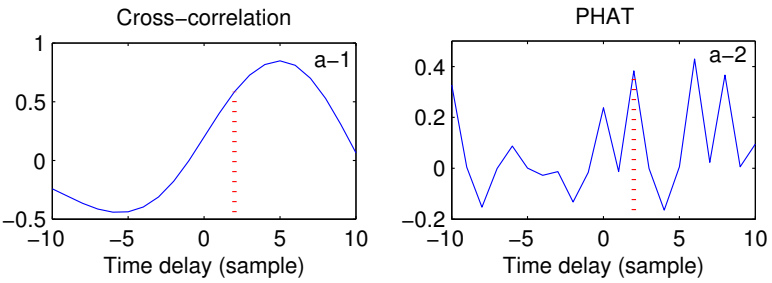
Given a sparsity regularization parameter λ', the optimization in Eq. 5.1 is a convex nonnegative quadratic programming (NNQP) problem, which can be solved by various methods such as multiplicative update, Merhotra primal-dual predictor-corrector interior point method and projected gradient descent, as described in Section 3.1.2 in Chapter 3. Another important issue in Eq. 5.1 is how to determine the regularization parameter λ', which controls the sparseness of solutions. Our Bayesian formulation in Chapter 3 showed that, the optimal regularization parameter λ' is equal to the product $\sigma^2\lambda$, where σ^2 describes the noise level and λ is the parameter describes the sparseness of filters. These two parameters can be determined by either *a priori* knowledge, or learned from observed microphone signals using a similar Bayesian approach described in Chapter 4.

5.3 Results

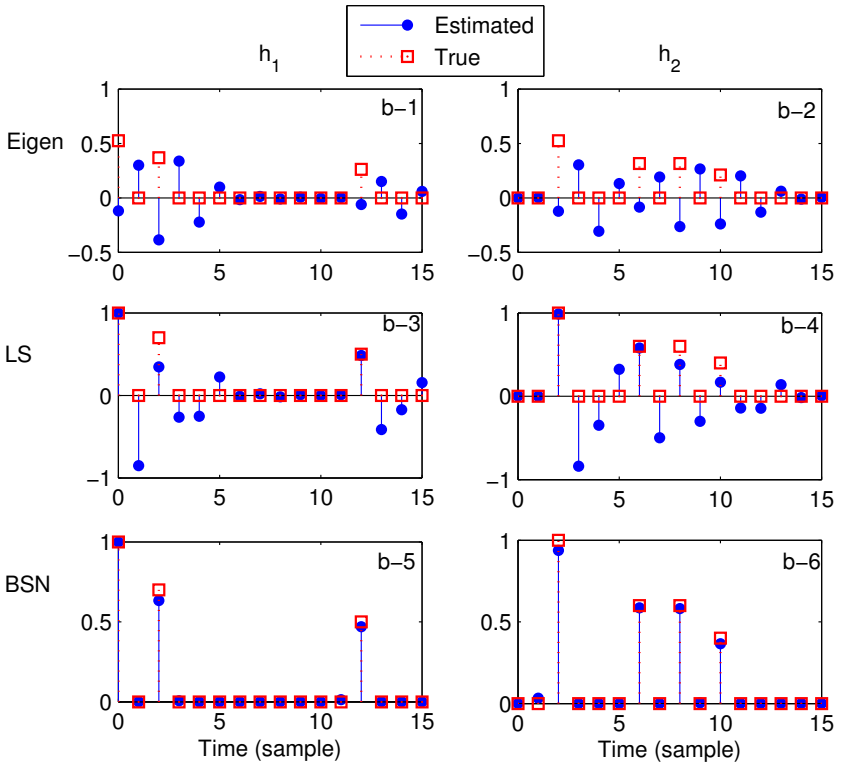
5.3.1 A simulated example

Here we first provide a toy example to illustrate the the advantage of the proposed BSN channel identification approach for TDOA estimation in comparison with other existing approaches. In the simulation, the source (s) is a speech segment of 4096 samples with sampling rate of 16 kHz, and both of the two FIR filters (h_1 and h_2) are 16 samples long. If we use $j = 0, 1, ...15$ to index the filter coefficients, filter h_1 has nonzero elements only at j=0, 2, and 12 with amplitudes of 1, 0.7, and 0.5, respectively; filter h_2 has nonzero elements only at j=2, 6, 8, and 10 with amplitudes of 1, 0.6, 0.6 and 0.4, respectively. Notice that both filters are nonnegative and sparse. Then, the simulated microphone observations (x_i) were computed according to Eq. 4.1 where the ambient noise (n_i) was real noise recorded in a conference room. The noise was scaled so that the signal-to-noise ratio (SNR) of the microphone signals was 15 dB. The simulated microphone signals were then highpassed with a cut-off frequency of 300 Hz to reduce the low frequency noise before they were fed to different algorithms for TDOA estimation.

The simulation results are shown in Fig. 5.2. The traditional cross-correlation approach [Fig. 5.2 (a-1)] has low temporal resolution, and multipath reflections often cause a peak shift in the cross-correlation function. Consequently, this approach performs poorly in reverberant environments. The phase transform (PHAT) approach [Fig. 5.2 (a-2)] improves the temporal resolution by pre-whitening the microphone signals, however, its performance is still limited by the underlying oversimplified RIR model. The simulation results of blind channel identification approaches are shown in Fig. 5.2 (b), illustrating strong advantages of our new formulation of blind channel identification presented in Section 5.2. As shown in Fig. 5.2 (b), the LS formulation in Eq. 4.3 in Chapter 4 is more robust to ambient noise than the conventional eigenvalue decomposition approach in Eq. 4.2 in Chapter 4. Moveover, the



(a) GCC approaches. In each figure, the solid line describes the GCC function between two microphone signals, and the vertical dot line indicates the true time delay. The traditional cross-correlation is on the left and the phase transform (PHAT) is on the right.



(b) Blind channel identification approaches. The three rows from top to bottom are the identified filters respectively by eigenvalue decomposition approach (Eq. 4.2), LS approach (Eq. 4.3) and the BSN channel identification approach (Eq. 5.1). The left and right columns represent the identified filters associated with channel 1 and channel 2, respectively. In each figure, the dot-solid line describes the identified filters, and the square-dot line indicates the true filters up to a constant time delay and a constant scalar factor.

Figure 5.2: Results of GCC approaches and blind channel identification approaches for TDOA estimation.

sparsity and nonnegativity prior knowledge helps to resolve the degeneracy in blind channel identification and yields dramatic improvement in filter estimates. The filter estimation accuracy gained by the BSN channel identification approach will become critical when the filters are thousands of taps long, as in typical real acoustic environments.

5.3.2 Performance comparison using real room recordings

Now we evaluate the performance of the proposed BSN channel identification approach for TDOA estimation in real environments. The experimental setup is illustrated in Fig. 5.3. Prerecorded speech sequences were played through a loudspeaker located at one end of the room and recorded by a matched omnidirectional microphone pair (SP-CMC-8, Sound Professionals) located at the other end of the room. We recorded two data sets: one set had the loudspeaker in the middle (see Position 1 in Fig. 5.3), and the other had the loudspeaker about 75 cm away from the middle (see Position 2 in Fig. 5.3). At each speaker position, 100 speech sentences (50 by a male speaker and 50 by a female speaker) were played and recorded with a sampling rate of 16 kHz. In our evaluation, we divided the recordings into segments of 4096 samples, and discarded those silent segments which contained no speech signals. Then, we treated each segment independently and performed TDOA algorithms on each of them. Since a large portion of the ambient noise was at low frequency (such as air-conditioning noise), the recorded signals were highpassed with a cut-off frequency of 300 Hz before they were fed to TDOA estimation algorithms. For the BSN channel identification approach, the filter length was 2048.

As shown in Fig. 5.4, the proposed BSN channel identification approach yielded consistent TDOA estimates at both Position 1 and Position 2, even though Position 2 is difficult for TDOA estimation since the loudspeaker was close to the wall and the wall reflections were very strong. In contrast, the PHAT approach had good estimates only at position 1 but not position 2. The cross-correlation approach did not yield satisfactory estimates

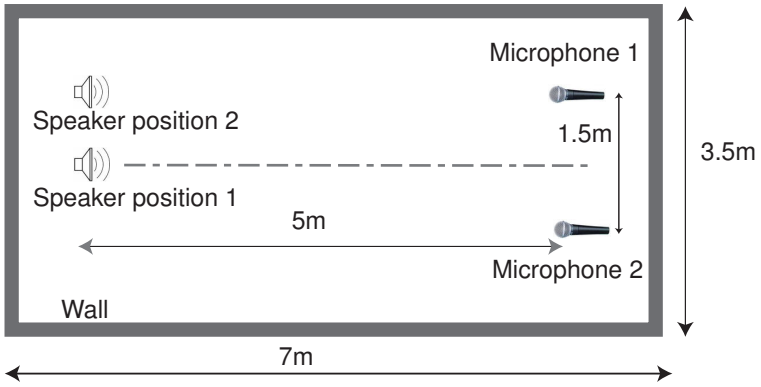


Figure 5.3: The loudspeaker-microphone positions in a conference room during recording. The dot-dash line indicates the center line of the room.

at either positions and almost completely failed at position 2. As for other blind channel identification approaches, the batch-mode eigenvalue decomposition (in Eq. **??**) and the LS (in Eq. 4.3), they were not able to yield competitive results simply because there were not enough frequency components in a short 4096-sample frame for estimating filters of length 2048. The BSN channel identification approach overcomes the difficulty by exploiting knowledge about the nonnegativity and sparsity of the RIRs.

5.4 Discussion

We have developed a blind sparse-nonnegative (BSN) channel identification approach for TDOA estimation, which exploits prior knowledge about an acoustic RIR, namely, an acoustic RIR can be modeled by a sparse-nonnegative FIR filter. The BSN channel identification is formulated as an l_1-norm regularized nonnegative LS problem, which is convex and can be solved efficiently with guaranteed global convergence. Both simulation and experimental results in real acoustic environments demonstrate the effectiveness of the BSN channel identification approach for TDOA estimation.

Although modeling an acoustic RIR as a sparse-nonnegative FIR filter is demonstrated to be effective for TDOA estimation, how accurate the modeling is in real acoustic envi-

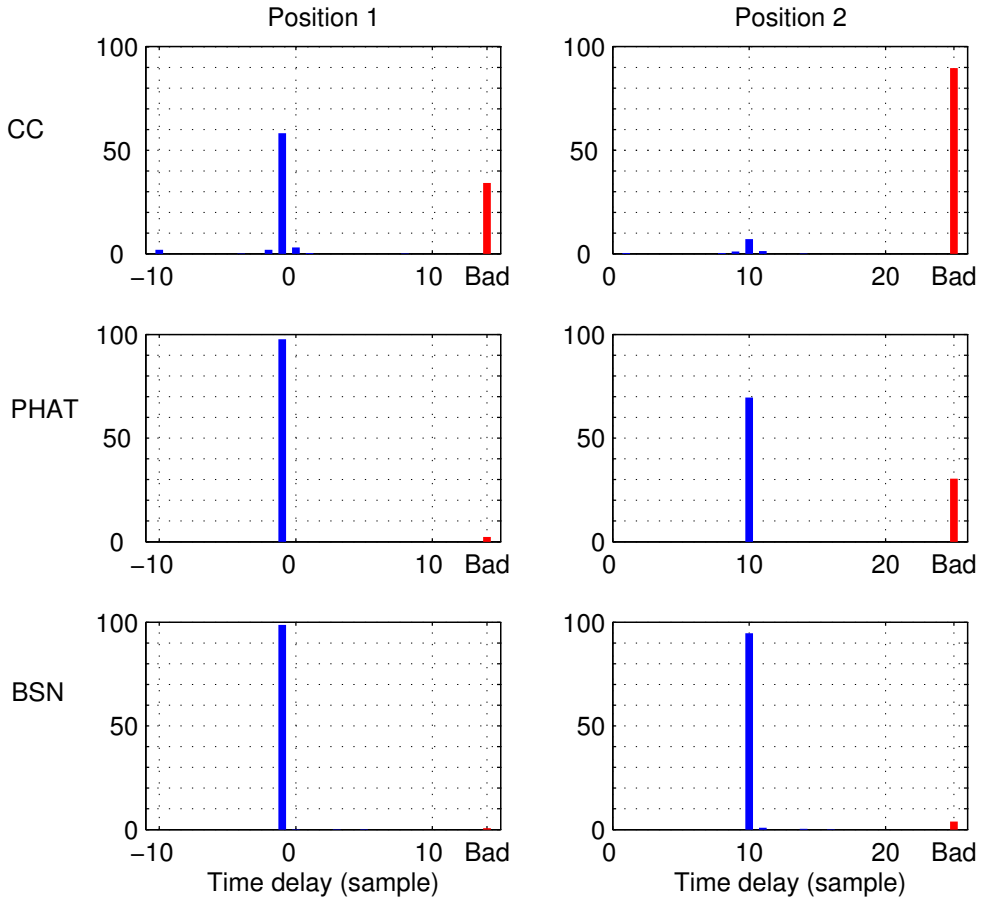


Figure 5.4: Histogram in percentage of TDOA estimates using three different approaches: the cross-correlation (CC) approach, the phase transform (PHAT) approach, and the BSN channel identification approach. The left and right column describes the TDOA estimation results when the speaker was at Position 1 and Position 2, respectively. The bad estimates are those that are more than 10 samples away from the true values (-1 for Position 1, and 10 for Position 2).

ronments remains an open problem. TDOA estimation is relatively immune to moderate modeling inaccuracy since it only requires information about the direct path but not the whole filter. Nevertheless, we believe exploiting prior knowledge about RIRs is crucial for blind channel identification to resolve its underlying degeneracy and become robust to ambient noise.

Our future work is to develop an adaptive algorithm for BSN channel identification. We expect the resulting adaptive algorithm would outperform the adaptive eigenvalue decomposition (AED) algorithm [7], which has been shown to be not only computationally efficient, but also effective in dealing with reverberation.

Chapter 6

Conclusion

We have proposed the l_1-norm sparse Bayesian learning for finding the *optimally* sparse solution in a given problem, and it is described in details in Chapter 3 using least squares problems as the examples. The *optimal sparseness* of solutions is defined in a Bayesian sense and inferred by learning directly from data. Our simulation results show that the l_1-norm sparse Bayesian learning is able to accurately resolve the true sparse structures even in very noisy data. Furthermore, our experimental results also demonstrate that the l_1-norm sparse Bayesian learning has the potential for solving very hard problems in signal processing, speech dereverberation and time difference of arrival (TDOA) estimation in reverberant real acoustic environments.

We have discussed our ongoing and future work in the respective chapters. We believe the l_1-norm sparse Bayesian learning provides an very powerful tool of dimensionality reduction for dealing with today's exponentially growing and heterogeneous data. We expect that it will play an important role in both data compression and knowledge discovery.

Bibliography

[1] A data compression website. In *http://www.data-compression.com/index.shtml*.

[2] High-dimensional data analysis: The curses and blessings of dimensionality. In *http://www-stat.stanford.edu/ donoho/Lectures/AMS2000/AMS2000.html*.

[3] JPEG standard (JPEG ISO/IEC 10918-1 ITU-T Recommendation T.81). *http://www.w3.org/Graphics/JPEG/itu-t81.pdf*.

[4] J. B. Allen and D. A. Berkley. Image method for efficiently simulating small-room acoustics. *J. Acoustical Society America*, 65:943–950, 1979.

[5] H. Attias, J. C. Platt, A. Acero, and L. Deng. Speech denoising and dereverberation using probabilistic models. In *NIPS 13*, 2000.

[6] Onureena Banerjee, Laurent El Ghaoui, Alexandre dAspremont, and Georges Natsoulis. Convex optimization techniques for fitting sparse gaussian graphical models. In *ICML*, 2006.

[7] J. Benesty. Adaptive eigenvalue decomposition algorithm for passive acoustic source localization. *J. Acoust. Soc. Am.,*, 107(1):384–391, 2000.

[8] Dimitri P. Bertsekas. *Nonlinear programming*. Athena Scientific, 2003.

[9] Guillaume Bouchard. Efficient bounds for the softmax function and applications to approximate inference in hybrid models, 2007.

[10] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[11] Atul Butte. The use and analysis of microarray data. *Nature Reviews Drug Discovery*, 2002.

[12] E. J. Candès. Compressive sampling. *Proceedings of the International Congress of Mathematicians*, pages 1433–1452, 2006.

[13] E. J. Candès, M. Wakin, and S. Boyd. Enhancing sparsity by reweighted l_1 minimization. *Technical Report, California Institute of Technology*, 2007.

[14] Emmanuel Candès and Justin Romberg. l_1-magic : Recovery of sparse signals. In *http://www.acm.caltech.edu/l1magic/downloads/l1magic.pdf*, 2005.

[15] Emmanuel Candes, Justin Romberg, and Terence Tao]. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52:489 – 509, 2006.

[16] Emmanuel Candes and Terence Tao. The dantzig selector: Statistical estimation when p is much larger than n. *The Annals of Statistics*, 35:2313–2351, 2007.

[17] Jingdong Chen, Jacob Benesty, and Yiteng (Arden) Huang. Time delay estimation in room acoustic environments: An overview. *EURASIP Journal on Applied Signal Processing*, 2006:Article ID 26503, 19 pages, 2006.

[18] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM J. Scientific Computing*, 20(1):33–61, 1998.

[19] Adrian Dobra, Chris Hans, Beatrix Jones, Joseph R. Nevins, Guang Yao, and Mike West. Sparse graphical models for exploring gene expression data. *Journal of Multivariate Analysis*, 90(1):196–212, 2004.

[20] Simon Doclo and Marc Moonen. Robust adaptive time delay estimation for speaker localization in noisy and reverberant acoustic environments. *EURASIP Journal on Applied Signal Processing*, 2003(11):1110–1124, 2003.

[21] David Donoho and Michael Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via $l^{(1)}$ minimization. *Proceedings of the National Academy of Sciences of the United States of America*, 100(5):2197–2202, 2003.

[22] David L. Donoho and Michael Elad. On the stability of the basis pursuit in the presence of noise. *Signal Processing*, 86:511 – 532, 2006.

[23] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern classification*. Wiley, New York, 2001.

[24] D. L. Duttweiler. Proportionate normalized least-mean-squares adaptation in echo cancelers. *IEEE Trans. Speech Audio Processing*, 8:508–518, 2000.

[25] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *Annals of Statistics*, 32:407–499, 2004.

[26] Jianqing Fan and Runze Li. Statistical challenges with high dimensionality: Feature selection in knowledge discovery, 2006.

[27] Usama M. Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery: an overview. pages 1–34, 1996.

[28] D. Foresee and M. Hagan. Gauss-Newton approximation to Bayesian regularization. In *Proceedings of the 1997 International Joint Conference on Neural Networks*, pages 1930–1935, 1997.

[29] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. In *http://www-stat.stanford.edu/ tibs/glasso/index.html*, 2007.

[30] George-Otbon Glentis, Kostas Berberidis, and Sergios Theodoridis. Efficient least squares adaptive algorithms for FIR transversal filtering. *IEEE Signal Processing Magazine*, 16(4):13–41, 1999.

[31] I.F. Gorodnitsky and B. D. Rao. Sparse signal reconstruction from limited data using focuss: are-weighted minimum norm algorithm. *IEEE Transactions on Signal Processing*, 45(3):600–616, 1997.

[32] Isabelle Guyon and Andr Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[33] D.A. Harville. Maximum likelihood approaches to variance component estimation and to related problems. *J. American Statistical Association*, 72:320–338, 1977.

[34] Patrik O. Hoyer. Non-negative sparse coding. In *Proc. IEEE Workshop on Neural Networks for Signal Processing*, 2002.

[35] A. Hyvarinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. New York, NY: John Wiley and Sons, 2001.

[36] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. A method for large-scale l1-regularized least squares. *IEEE Journal on Selected Topics in Signal Processing*, 4:606–617, 2007.

[37] C. H. Knapp and G. C. Carter. The generalized correlation method for estimation of time delay. *IEEE Trans. ASSP*, 24(4):320–327, 1976.

[38] K. Koh, S.-J. Kim, and S. Boyd. An interior-point method for large-scale l1-regularized logistic regression. *Journal of Machine Learning Research*, 2007.

[39] H. Krim and M. Viberg. Two decades of array signal processing research: the parametric approach. *IEEE Signal Processing Magazine*, 13(4):67–94, 1996.

[40] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.

[41] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In Todd K. Leen, Thomas G. Dietterich, and Volker Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 556–562. MIT Press, 2000.

[42] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y. Ng. Efficient sparse coding algorithms. In *Proceedings of the Neural Information Processing Systems (NIPS) 19*, 2007.

[43] S.-I. Lee, H. Lee, P. Abbeel, and A.Y. Ng. Efficient l1 regularized logistic regression. In *Proceedings of the Twenty-first National Conference on Artificial Intelligence (AAAI-06)*, pages 1–9, 2006.

[44] Su-In Lee, Varun Ganapathi, and Daphne Koller. Efficient structure learning of markov networks using l1-regularization. In *Proceedings of Neural Information Processing Systems (NIPS 19)*, 2007.

[45] Michael S. Lewicki and Terrence J. Sejnowski. Learning overcomplete representations. *Neural Computation*, 12:337–365, 2000.

[46] Chih-Jen Lin. Projected gradient methods for nonnegative matrix factorization. *Neural Computation*, 19:2756–2779, 2007.

[47] Y. Lin and D. D. Lee. Bayesian l_1-norm sparse learning. In *Proc. ICASSP*, 2006.

[48] Y. Lin and D. D. Lee. Bayesian Regularization And Nonnegative Deconvolution (BRAND) for room impulse response estimation. *IEEE Trans. Signal Processing*, 54(3):839–847, 2006.

[49] Yuanqing Lin and Daniel D. Lee. Bayesian regularization and nonnegative deconvolution for time delay estimation. In *Advances in Neural Information Processing Systems*, 2005.

[50] Yuanqing Lin, Paul Vernaza, Jihun Ham, , and Daniel D. Lee. Cooperative relative robot localization with audible acoustic sensing. In *IEEE/ISJ International Conference on Intelligent Robotics and Systems*, pages 662–667, 2005.

[51] L.B. Lucy. An iterative technique for the rectification of observed distributions. *Astronomical Journal*, 79:745–754, 1974.

[52] D. J. C. MacKay. Bayesian interpolation. *Neural Computation*, 4:415–447, 1992.

[53] D. M. Malioutov, M. Cetin, and A. S. Willsky. Homotopy continuation for sparse signal representation. In *Proc. ICASSP*, 2005.

[54] M. Miyoshi and Y. Kaneda. Inverse filtering of room acoustics. *IEEE Trans. Acoustics, Speech, and Signal Processing*, 36(2):145–152, 1988.

[55] T. Nakatani, M. Miyoshi, and K. Kinoshita. One microphone blind dereverberation based on quasi-periodicity of speech signals. In *NIPS 16*. 2004.

[56] B.A. Olshausen and D.J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for nature images. *Nature*, 381:607–609, 1996.

[57] Bruno A Olshausen and David J Field. Sparse coding of sensory inputs. *Current Opinion in Neurobiology*, 2004.

[58] Alan V. Oppenheim and Ronald W. Schafer. *Discrete-time signal processing*. Prentice Hall, 1998.

[59] M. R. Osborne, Brett Presnell, and B. A. Turlach. A new approach to variable selection in least squares problems. *IMA Journal of Numerical Analysis*, 20(3):389–403, 2000.

[60] Simon Perkins, Kevin Lacker, and James Theiler. Grafting: fast, incremental feature selection by gradient descent in function space. *Journal of Machine Learning Research*, 3:1333 – 1356, 2003.

[61] Bernhard Schlkopf and Alex Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

[62] F. Sha, Y. Park, and L. K. Saul. Multiplicative updates for l1-regularized linear and logistic regression. In *Proc. of Intelligent Data Analysis (IDA-2007)*.

[63] Fei Sha, Yuanqing Lin, Lawrence K. Saul, and Daniel D. Lee. Multiplicative updates for nonnegative quadratic programming. *Neural Comput.*, 19(8):2004–2031, 2007.

[64] Evan C. Smith and Michael S. Lewicki. Efficient auditory coding. *Nature*, 439:978–982, 2006.

[65] Robert Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Socity, Series B, (Methodological)*, 58(1):267–288, 1996.

[66] M. E. Tipping. Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244, 2001.

[67] L. Tong, G. Xu, and T. Kailath. Blind identification and equalization based on second-order statistics: A time domain approach. *IEEE Trans. Information Theory*, 40(2):340–349, 1994.

[68] M. J. Wainwright, P. Ravikumar, and J. Lafferty. High-dimensional graphical model selection using ℓ_1-regularized logistic regression. In *Advances in Neural Information Processing Systems*. 2006.

[69] S. J. Wright. *Primal-Dual Interior Point Methods*. Philadelphia, PA: SIAM, 1997.

[70] W. J. Zangwill. *Nonlinear Programming: a unified approach*. Prentice-Hall, 1969.