UNIVERSITY OF CALIFORNIA SAN DIEGO

A Family of Sparsity-Promoting Gradient Descent Algorithms Based on Sparse Signal Recovery

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Electrical Engineering (Signal and Image Processing)

by

Ching-Hua Lee

Committee in charge:

Bhaskar D. Rao, Chair
Harinath Garudadri
William S. Hodgkiss
Truong Q. Nguyen
Piya Pal
Rayan Saab

2020

The dissertation of Ching-Hua Lee is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

_____

_____

_____

Chair

University of California San Diego

2020

DEDICATION

To my family and teachers.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

It is my pleasure to acknowledge the roles of several individuals who helped and supported me throughout my graduate studies, and to those who made my journey possible. First and foremost, I am deeply grateful to my Ph.D. advisors, Prof. Bhaskar D. Rao and Dr. Harinath Garudadri, for their guidance, patience, and support during my Ph.D. studies. I thank them for introducing me into the fields of sparse signal recovery and speech/audio processing, based on which this dissertation is developed. I would also like to thank them for teaching me how to recognize research ideas worth developing, how to describe complex engineering problems with concise mathematical expressions, and how to convert theoretical findings into practical applications. They guided me towards becoming an independent researcher.

I would also like to thank Prof. William S. Hodgkiss, Prof. Truong Q. Nguyen, Prof. Piya Pal, and Prof. Rayan Saab for serving on my committee. I have learned a lot from their lectures and feedbacks during my graduate studies. I also thank our collaborators, Prof. James M. Kates, Prof. fred harris and Prof. Arthur Boothroyd for providing insightful suggestions and valuable help and advice on various projects.

I have been fortunate to have some fellow graduate students who I collaborated with, including Swaroop Gadiyaram, Krishna C. Vastare, Louis Pisha, Igor Fedorov, Sung-En Chiu, Tharun Srikrishnan, Çağrı Yalçın, Gökçe Sarar, Dhiman Sengupta, Sean Hamilton, Jing Liu, Aditya Sant, Rohan Pote, Hitesh Khunti, Govind R. Gopal, David Ho, Alice Sokolova, Wenyu Zhang and Mingchao Liang. I have gained many valuable suggestions and insights from the discussions with them and have also learned a lot from them. Special thanks to Kuan-Lin Chen, who has not only been an excellent labmate but also an incredible friend, for the time and effort we have spent together on our collaboration works as well as life experiences.

Finally, I am deeply indebted to my parents for their unconditional love and endless encouragement throughout my life. Without their scarifies and support this would not have been possible. This dissertation is dedicated to them.

Chapter 2 is, in part, a reprint of the material as it appears in the two papers: C.-H. Lee, B. D. Rao, and H. Garudadri, "Proportionate adaptive filtering algorithms derived using an iterative reweighting framework," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2020 and C.-H. Lee, B. D. Rao, and H. Garudadri, "Proportionate adaptive filters based on minimizing diversity measures for promoting sparsity," in *53rd Asilomar Conference on Signals, Systems, and Computers (ACSSC)*, 2019. The dissertation author was the primary investigator and author of these papers.

Chapter 3 is, in part, a reprint of the material as it appears in the paper: C.-H. Lee, B. D. Rao, and H. Garudadri, "A sparse conjugate gradient adaptive filter," *IEEE Signal Processing Letters*, 2020. The dissertation author was the primary investigator and author of this paper.

Chapter 4 is, in part, a reprint of the material as it appears in the three papers: C.-H. Lee, K.-L. Chen, f. harris, B. D. Rao, and H. Garudadri, "On mitigating acoustic feedback in hearing aids with frequency warping by all-pass networks," in *20th Annual Conference of the International Speech Communication Association (Interspeech)*, 2019; C.-H. Lee, J. M. Kates, B. D. Rao, and H. Garudadri, "Speech quality and stable gain trade-offs in adaptive feedback cancellation for hearing aids," *Journal of the Acoustical Society of America Express Letters*, 2017; and C.-H. Lee, B. D. Rao, and H. Garudadri, "Sparsity promoting LMS for adaptive feedback cancellation," in *25th European Signal Processing Conference (EUSIPCO)*, 2017. The dissertation author was the primary investigator and author of these papers.

Chapter 5 is, in part, a reprint of the material as it appears in the two papers: C.-H. Lee, B. D. Rao, and H. Garudadri, "Weighted gradient descent algorithms for learning regularized models," *IEEE Transactions on Signal Processing*, under review and C.-H. Lee, I. Fedorov, B. D. Rao, and H. Garudadri, "SSGD: Sparsity-promoting stochastic gradient descent algorithm for unbiased DNN pruning," in *45th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2020. The dissertation author was the primary investigator and author of these papers.

VITA

2013	Bachelor of Science, National Taiwan University, Taiwan
2016	Master of Science, University of California San Diego, USA
2020	Doctor of Philosophy, University of California San Diego, USA

PUBLICATIONS

C.-H. Lee, B. D. Rao, and H. Garudadri, "Weighted gradient descent algorithms for learning regularized models," *IEEE Transactions on Signal Processing*, under review.

K.-L. Chen, **C.-H. Lee**, B. D. Rao, and H. Garudadri, "Interference-robust speech source localization using time-frequency weighted criteria," *46th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2021, under review.

C.-H. Lee, B. D. Rao, and H. Garudadri, "Proportionate adaptive filtering algorithms derived using an iterative reweighting framework," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2020.

H. Garudadri, **C.-H. Lee**, and B. D. Rao,"Sparsity-aware adaptive feedback cancellation," *United States Patent Application*, 2020.

C.-H. Lee, B. D. Rao, and H. Garudadri, "A sparse conjugate gradient adaptive filter," *IEEE Signal Processing Letters*, 2020.

K.-L. Chen, **C.-H. Lee**, B. D. Rao, and H. Garudadri, "Jointly leveraging decorrelation and sparsity for improved feedback cancellation in hearing aids," in *28th European Signal Processing Conference (EUSIPCO)*, 2020.

C.-H. Lee, I. Fedorov, B. D. Rao, and H. Garudadri, "SSGD: Sparsity-promoting stochastic gradient descent algorithm for unbiased DNN pruning," in *45th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2020.

L. Pisha, J. Warchall, T. Zubatiy, S. Hamilton, **C.-H. Lee**, G. Chockalingam, P. P. Mercier, R. Gupta, B. D. Rao, and H. Garudadri, "A wearable, extensible, open-source platform for hearing healthcare research," *IEEE Access*, 2019.

C.-H. Lee, B. D. Rao, and H. Garudadri, "Proportionate adaptive filters based on minimizing diversity measures for promoting sparsity," in *53rd Asilomar Conference on Signals, Systems, and Computers (ACSSC)*, 2019.

K.-L. Chen, **C.-H. Lee**, B. D. Rao, and H. Garudadri, "A generalized proportionate-type normalized subband adaptive filter," in *53rd Asilomar Conference on Signals, Systems, and Computers (ACSSC)*, 2019.

C.-H. Lee, K.-L. Chen, f. harris, B. D. Rao, and H. Garudadri, "On mitigating acoustic feedback in hearing aids with frequency warping by all-pass networks," in *20th Annual Conference of the International Speech Communication Association (Interspeech)*, 2019.

L. Pisha, S. Hamilton, D. Sengupta, **C.-H. Lee**, K. C. Vastare, T. Zubatiy, S. Luna, C. Yalcin, A. Grant, R. Gupta, G. Chockalingam, B. D. Rao, and H. Garudadri, "A wearable platform for research in augmented hearing," in *52nd Asilomar Conference on Signals, Systems, and Computers (ACSSC)*, 2018.

C.-H. Lee, B. D. Rao, and H. Garudadri, "Bone-conduction sensor assisted noise estimation for improved speech enhancement," in *19th Annual Conference of the International Speech Communication Association (Interspeech)*, 2018.

C.-H. Lee, J. M. Kates, B. D. Rao, and H. Garudadri, "Speech quality and stable gain trade-offs in adaptive feedback cancellation for hearing aids," *Journal of the Acoustical Society of America Express Letters*, 2017.

H. Garudadri, A. Boothroyd, **C.-H. Lee**, S. Gadiyaram, J. Bell, D. Sengupta, S. Hamiltonz, K. C. Vastare, R. Gupta, and B. D. Rao, "A realtime, open-source speech-processing platform for research in hearing loss compensation," in *51st Asilomar Conference on Signals, Systems, and Computers (ACSSC)*, 2017.

C.-H. Lee, B. D. Rao, and H. Garudadri, "Sparsity promoting LMS for adaptive feedback cancellation," in *25th European Signal Processing Conference (EUSIPCO)*, 2017.

ABSTRACT OF THE DISSERTATION

A Family of Sparsity-Promoting Gradient Descent Algorithms Based on Sparse Signal Recovery

by

Ching-Hua Lee

Doctor of Philosophy in Electrical Engineering (Signal and Image Processing)

University of California San Diego, 2020

Bhaskar D. Rao, Chair

Sparsity has played an important role in numerous signal processing systems. By leveraging sparse representations of signals, many batch estimation algorithms and methods that are efficient, robust, and effective for practical engineering problems have been developed. However, gradient descent-based approaches that are less computationally expensive have become essential to the development of modern machine learning systems, especially the deep neural networks (DNNs). This dissertation examines how we can incorporate sparsity principles into gradient-based learning algorithms, in both signal processing and machine learning applications, for improved estimation and optimization performance.

On the signal processing side, we study how to take advantage of sparsity in the system response for improving the convergence rate of the least mean square (LMS) family of adaptive filters, which are derived from using gradient descent on the mean square error objective function. Based on iterative reweighting sparse signal recovery (SSR) techniques, we propose a novel framework for deriving a class of sparsity-aware LMS algorithms by adopting an affine scaling transformation (AST) methodology in the algorithm design process. Sparsity-promoting LMS (SLMS) and Sparsity-promoting Normalized LMS (SNLMS) algorithms are introduced, which can take advantage of, though do not strictly enforce, the sparsity of the underlying system if it already exists for convergence speedup. In addition, the reweighting–AST framework is applied to the conjugate gradient (CG) class of adaptive algorithms, which in general demonstrate a much higher convergence rate than the LMS family. The resulting Sparsity-promoting CG (SCG) algorithm also demonstrates improved convergence characteristics for sparse system identification. Finally, the proposed algorithms are applied to the real-world problem of acoustic feedback reduction encountered in hearing aids.

On the machine learning side, we investigate how to exploit the SSR techniques in gradient-based optimization algorithms for learning compact representations in nonlinear estimation tasks, especially with overparameterized models. In particular, the reweighting–AST framework is utilized in the context of estimating a regularized solution exhibiting some desired properties such as sparsity without having to incorporate a regularization penalty. The resulting algorithms in general have a weighted gradient term in the update equation where the weighting matrix provides certain implicit regularization capabilities. We start by establishing a general framework that can possibly extend to various regularizers and then focus on the sparsity regularization aspect. As notable applications of nonlinear model sparsification, we propose i) Sparsity-promoting Stochastic Gradient Descent (SSGD) algorithms for DNN compression and ii) Sparsity-promoting Kernel LMS (SKLMS) and Sparsity-promoting Kernel NLMS (SKNLMS) algorithms for dictionary pruning in kernel methods.

Chapter 1

Introduction

This dissertation focuses on novel applications of sparse signal recovery (SSR) techniques to the family of gradient descent-based learning algorithms for the optimization of signal processing and machine learning systems. In this chapter, we briefly discuss the background and motivation of the work, followed by an overview of the remaining chapters in the dissertation.

1.1 SSR and Iterative Reweighting Methods

Sparsity has been an important attribute in many successful signal processing applications [1, 2, 3]. Specialized algorithms can exploit the parsimony in signals and systems to provide faster sampling rates in acquisition devices, more efficient digital communications, better model compression, and improved robustness to outliers, interference, and noise [4, 5, 6, 7, 8, 9, 10, 11]. Many of the algorithms are developed based on SSR techniques that search for a sparse solution to an underdetermined system of linear equations where there are infinitely many solutions.

More formally, the problem of SSR considers finding a solution to: $\mathbf{b} = \mathbf{A}\mathbf{x}$, where $\mathbf{A} \in \mathbb{R}^{N \times M}$ represents an overcomplete or redundant basis assuming $N < M$ and $\text{rank}(\mathbf{A}) = N$, $\mathbf{x} \in \mathbb{R}^{M}$ is the vector of unknown coefficients to be learned, and $\mathbf{b} \in \mathbb{R}^{N}$ is the measurement vector.

As the system has fewer equations than unknowns, it has infinitely many solutions. In the SSR problem, it is assumed that the solution of interest is sparse, i.e., only very few of its elements are nonzero. The sparse structure can thus be utilized as additional information to identify which of these candidate solutions is indeed the desirable one.

To approach a sparse solution, one possibility is to restrict the search space of possible solutions. It is natural to start from considering the search of the minimal ℓ_0 "norm"[1] solution:

$$\min_{\mathbf{x}} \quad \|\mathbf{x}\|_0, \qquad \text{s.t.} \quad \mathbf{b} = \mathbf{A}\mathbf{x}. \tag{1.1}$$

The ℓ_0 "norm" term serves as a measure of diversity (as opposite to sparsity) of the solution vector. Minimizing it is equivalent to looking for the sparsest (or least diverse) solution in terms of the count of nonzero elements.

To accommodate for measurement noise in practice, we instead consider the alternative regularized problem:

$$\min_{\mathbf{x}} \quad \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_0, \tag{1.2}$$

where $\lambda > 0$ is the weight for the ℓ_0 "norm" regularization penalty and is related to the measurement noise variance.

Unfortunately, finding the optimal solution in such a case is in general NP-hard [12, 1]. To allow for tractable computation, one usually resorts to approximations of the ℓ_0 "norm" penalty. For example, consider the following optimization problem instead:

$$\min_{\mathbf{x}} \quad \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda G(\mathbf{x}), \tag{1.3}$$

where $G(\cdot) : \mathbb{R}^M \mapsto \mathbb{R}$, usually referred to as the *general diversity measure*, is a function employed as a simpler alternative to the ℓ_0 "norm" that when minimized also encourages sparsity in its

[1] The ℓ_0 "norm" of a vector is defined as the number of its nonzero entries. The quotation marks are used to warn that it is not a proper norm.

argument. For $\mathbf{x} = [x_0, x_1, \ldots, x_{M-1}]^T$, a *separable* form is commonly used: $G(\mathbf{x}) = \sum_{i=0}^{M-1} g(x_i)$, where $g(\cdot)$ has the following properties [13]:

Property 1: $g(z)$ is symmetric, i.e., $g(z) = g(-z) = g(|z|)$;

Property 2: $g(|z|)$ is monotonically increasing with $|z|$;

Property 3: $g(0)$ is finite;

Property 4: $g(z)$ is concave in $|z|$ or z^2.

Any function that holds the above properties is a candidate for effective SSR algorithms. For example, popular choices include:

$$g(z) = |z|^p, \qquad 0 < p \leq 2 \qquad [14]$$

$$g(z) = \log(z^2 + \epsilon), \qquad \epsilon > 0 \qquad [15]$$

$$g(z) = \log(|z| + \epsilon), \qquad \epsilon > 0 \qquad [16]$$

However, the concave nature of the diversity measure function poses challenges to the optimization problem. Consequently, in practical situations there is a need for approximate methods that efficiently solve (1.3) in most cases. Many SSR algorithms rely on *iterative schemes* that produce more focal estimates as optimization progresses [13]. In every iteration, a new upper bound is created for the concave penalty $G(\mathbf{x})$ as a surrogate function, resulting in a simpler problem that can be solved more efficiently. This has led to the development of useful and effective reweighted norm minimization algorithms. Typically, ℓ_2 and ℓ_1 norms are selected because of their convex nature and the former because of the closed form solution at each iteration. We now discuss them in more detail.

1.1.1 Iterative Reweighted ℓ_2

To apply the reweighted ℓ_2 method, first note that the function $g(z)$ has to be concave in z^2 for Property 4; i.e., it satisfies:

$$g(z) = f(z^2), \tag{1.4}$$

where $f(z)$ is concave for $z \in \mathbb{R}_+$ (the positive orthant). Now, given an estimate $\mathbf{x}_k$ at iteration k, the solution estimate of the next iteration $\mathbf{x}_{k+1}$ is given by solving the weighted ℓ_2 norm minimization problem:

$$\mathbf{x}_{k+1} = \arg\min_{\mathbf{x}} \quad \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda \|\mathbf{W}_k^{-1}\mathbf{x}\|_2^2, \tag{1.5}$$

where $\mathbf{W}_k = \text{diag}\{w_{i,k}\}$ is a diagonal matrix with

$$w_{i,k} = \left(\frac{\mathrm{d}f(z)}{\mathrm{d}z}\bigg|_{z=x_{i,k}^2} \right)^{-\frac{1}{2}}, \tag{1.6}$$

where d denotes the differential operator.

This algorithm refines the estimate of $\mathbf{x}$ by iteratively solving the weighted ℓ_2 norm minimization problem (1.5) which has a closed form solution:

$$\mathbf{x}_{k+1} = \mathbf{W}_k^2 \mathbf{A}^T \left(\lambda \mathbf{I} + \mathbf{A}\mathbf{W}_k^2 \mathbf{A}^T \right)^{-1} \mathbf{b}, \tag{1.7}$$

until some convergence criterion is met.

One useful example of the diversity measure applicable to the reweighted ℓ_2 framework is the p-norm-like diversity measure [14]:

$$G(\mathbf{x}) = \sum_{i=0}^{M-1} |x_i|^p, \quad 0 < p \leq 2. \tag{1.8}$$

Since it satisfies Properties 1–4 (note that for Property 4 we have $|x_i|^p$ concave in x_i^2 for $0 < p \leq 2$), we can use (1.6) to obtain the update equation for $\mathbf{W}_k$ as:

$$w_{i,k} = \left(\frac{2}{p} |x_{i,k}|^{2-p} \right)^{\frac{1}{2}}. \tag{1.9}$$

This algorithm is known as the FOCUSS (FOcal Underdetermined System Solver) algorithm [14].

Another famous example is the log-sum penalty function used in [15]:

$$G(\mathbf{x}) = \sum_{i=0}^{M-l} \log(x_i^2 + \epsilon), \quad \epsilon > 0. \tag{1.10}$$

As it satisfies Properties 1–4 with the concavity of $\log(x_i^2 + \epsilon)$ in x_i^2 holding for Property 4, we can use (1.6) to obtain the update equation for $\mathbf{W}_k$ as:

$$w_{i,k} = \left(x_{i,k}^2 + \epsilon \right)^{\frac{1}{2}}. \tag{1.11}$$

Note that in practice, one would gradually decrease ϵ with increasing iteration number to obtain better performance of the algorithm as suggested by [15].

1.1.2 Iterative Reweighted ℓ_1

To apply the reweighted ℓ_1 method, first note that the function $g(z)$ has to be concave in $|z|$; i.e., it satisfies:

$$g(z) = f(|z|), \tag{1.12}$$

where $f(z)$ is concave for $z \in \mathbb{R}_+$. In this case, given an estimate $\mathbf{x}_k$ at iteration k, the solution estimate of the next iteration $\mathbf{x}_{k+1}$ is given by solving the weighted ℓ_1 norm minimization problem:

$$\mathbf{x}_{k+1} = \arg\min_{\mathbf{x}} \quad \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda \|\mathbf{W}_k^{-1}\mathbf{x}\|_1, \tag{1.13}$$

where $\mathbf{W}_k = \text{diag}\{w_{i,k}\}$ is now given differently as:

$$w_{i,k} = \left(\frac{\mathrm{d}f(z)}{\mathrm{d}z}\bigg|_{z=|x_{i,k}|} \right)^{-1}. \tag{1.14}$$

This algorithm refines the estimate of $\mathbf{x}$ by iteratively solving the weighted ℓ_1 norm minimization problem (1.13) until some convergence criterion is met. Note that in this case there is no closed form solution. However, (1.13) is still a tractable convex problem that can be efficiently solved by numerical programs (e.g., interior point method).

As an example, the p-norm-like diversity measure can also be utilized for the reweighted ℓ_1 framework with a different range of p:

$$G(\mathbf{x}) = \sum_{i=0}^{M-1} |x_i|^p, \quad 0 < p \leq 1. \tag{1.15}$$

Since it satisfies Properties 1–4 (note that for Property 4 we have $|x_i|^p$ concave in $|x_i|$ for $0 < p \leq 1$), the reweighted ℓ_1 scheme can be applied. Using (1.14) we obtain the update equation for $\mathbf{W}_k$ as:

$$w_{i,k} = \frac{1}{p} |x_{i,k}|^{1-p}. \tag{1.16}$$

Another widely seen diversity measure is the log-sum penalty proposed in [16]:

$$G(\mathbf{x}) = \sum_{i=0}^{M-l} \log(|x_i| + \epsilon), \quad \epsilon > 0. \tag{1.17}$$

As it satisfies Properties 1–4 with the concavity of $\log(|x_i| + \epsilon)$ in $|x_i|$ holding for Property 4, we can use (1.14) to obtain the update equation for $\mathbf{W}_k$ as:

$$w_{i,k} = |x_{i,k}| + \epsilon. \tag{1.18}$$

Note that in this case, a fixed ϵ is typically adopted as suggested in [16].

1.1.3 Discussions

The reweighting techniques actually belong to the more general class of majorization-minimization algorithms [17]. In each of the reweighting schemes, the weighted ℓ_2 or ℓ_1 norm term serves as an upper bound for $G(\mathbf{x})$ in every iteration. More specifically, the matrix $\mathbf{W}_k$, as a function of the current estimate $\mathbf{x}_k$, provides a majorizer w.r.t. $\mathbf{x}_k$ for the objective function to be minimized. This allows the iterative algorithms to produce more focal estimates as optimization progresses. Hopefully when the number of iterations is large enough, the optimal solution can be well approached or even achieved [13].

Note that (1.5) and (1.13) have assumed that $\mathbf{W}_k$ to be invertible, though it is not necessarily required as the final algorithm like (1.7) does not involve inverting $\mathbf{W}_k$. However, it is still favorable to have all the diagonal elements of $\mathbf{W}_k$ strictly positive for avoiding instability and algorithm stagnation. The positive definiteness of $\mathbf{W}_k$ can be shown to hold for a wide variety of diversity measures used in SSR. However, in cases where it is not (e.g., the $\mathbf{W}_k$ updates (1.9) and (1.16) using the p-norm-like function), practically some small regularization constant can be properly added to ensure the positive definiteness.

1.2 Gradient Descent-Based Learning and Optimization

Many learning and optimization problems rely on gradient descent-based algorithms as they are simple, effective, and suitable for stochastic estimation especially useful for tasks with large datasets, in which batch estimation algorithms like that in SSR are not easily applicable due to computational constraints. Typically, gradient-based algorithms learn the model parameters by

updating them in an iterative manner:

$$\begin{pmatrix}\text{updated}\\ \text{parameters}\end{pmatrix}=\begin{pmatrix}\text{old}\\ \text{parameters}\end{pmatrix}+\begin{pmatrix}\text{learning}\\ \text{rate}\end{pmatrix}\times\begin{pmatrix}\text{search}\\ \text{direction}\end{pmatrix}, \tag{1.19}$$

where the learning rate (also known as step size) determines how large a step is taken in each iteration along the search direction that is obtained based on the gradient information of the underlying objective function to be minimized.

Gradient-based algorithms have a long history in the signal processing domain and have been widely deployed since a few decades ago. In particular, the classic least mean square (LMS) algorithm [18], which was developed based on using a simple gradient descent on the mean square error objective, has motivated a bunch of LMS-type adaptive filtering algorithms [19, 20, 21] that are popular in many signal processing systems and applications. For estimating signals or systems with sparse structures, sparsity has naturally been leveraged to improve the convergence characteristics of the adaptive algorithms [22, 23, 24, 25]. The most well-known adaptive algorithms for exploiting sparsity may be the proportionate class of adaptive filters [26]. However, most of the proportionate algorithms were not developed based on any optimization criterion but were based on good heuristics [27, 28, 29]. Consequently, they might not be easy to generalize or adapt for different applications in a systematic way.

Furthermore, owing to the rise of deep learning since the introduction and success of AlexNet in 2012 [30], the gradient descent family of algorithms have attracted considerable attention due to their simplicity and effectiveness for optimizing deep neural networks (DNNs). In particular, the stochastic gradient descent (SGD) algorithm has been an essential optimization tool for DNNs, and many SGD variants have been proposed to achieved better learning outcomes, e.g., AdaGrad [31], Adam [32], RMSProp [33], to name a few. In the machine learning community, much effort has been invested into developing novel, effective SGD-type learning algorithms [31, 33, 32], or researching their capabilities of finding good models despite the highly nonlinear,

nonconvex objective function [34, 35, 36, 37, 38, 39, 40, 41]. However, there seems to have relatively less discussion on the aspect of exploiting the notion of sparsity with the optimization algorithms for deep learning applications.

1.3 Contributions of the Dissertation and Overview

In this dissertation, we present a general framework that incorporates sparsity into gradient descent-based learning algorithms for improving the optimization process or outcome for several machine learning and signal processing systems. Specifically, a family of sparsity-promoting algorithms are developed based on utilizing the iterative reweighting SSR techniques and incorporating an affine scaling transformation (AST) methodology [42] into the algorithm design process. We show that when the reweighting–AST framework is applied in the context of adaptive filtering, a general class of proportionate adaptive algorithms are obtained in a systematic way that achieve faster adaptation for identifying sparse systems. Based on the success in adaptive filters, we extend the framework to more general optimization problems where the models are not limited to be linear and are assumed to have multiple solutions that result in the same optimum. We show that for such scenarios as commonly encountered in deep learning, sparse models that are beneficial for pruning and compression purposes can be obtained using the developed algorithms.

This dissertation is organized as follows. An overview diagram is presented in Figure 1.1 to summarize the organization.

- Chapter 2 presents a novel reweighting–AST framework for developing a large class of adaptive filters that leverage the sparse nature of the system responses. The developed LMS-type sparse adaptive filters lie at the intersection of the proportionate-class [27, 28, 29, 26] and SSR-inspired [24, 25, 43, 44] adaptive algorithms and provides an interesting bridge. Under the framework, two new proportionate algorithms, namely, Sparsity-promoting LMS (SLMS) and Sparsity-promoting NLMS (SNLMS) are introduced, which permit incorporation of a broad

class of diversity measures that have proved effective for SSR. In this sense, our framework provides a systematic way of designing the proportionate factors for the algorithms in contrast to existing approaches that are mostly ad hoc.

- Chapter 3 extends the reweighting–AST framework developed in Chapter 2 to another class of adaptive filters, i.e., the conjugate gradient (CG) family of adaptive algorithms [45, 46, 47, 48, 49, 50], and devices a novel CG-type adaptive filter that we call Sparsity-promoting CG (SCG). The SCG algorithm generally has a faster convergence rate than the SLMS and SNLMS while with a higher computational complexity. When processing power is less of a constraint, the SCG may be a better choice.

- Chapter 4 discusses a real-world engineering problem where the proposed sparsity-promoting adaptive algorithms can be useful, i.e., the acoustic feedback reduction problem in hearing aids, where LMS-based adaptive feedback cancellation (AFC) algorithms are typically employed [51]. We show that the (quasi-) sparse nature of the acoustic feedback path impulse responses can be suitably leveraged to achieve better speech quality and higher system stable gain via our SLMS algorithm which offers control over the sparsity degree. Moreover, we introduce a novel decorrelation approach called "freping," which utilizes a network of all-pass filters to realize nonlinear frequency warping, to further enhance the AFC system on top of SLMS by mitigating the Nyquist stability criterion. The algorithms developed in this chapter have been implemented and run real-time on the Open Speech Platform [52, 53].

- Chapter 5 gives attention to the general optimization problem in machine learning where gradient descent plays an important role, e.g., in DNN optimization. Specifically, we discuss potential ways of leveraging the implicit regularization property of gradient descent to estimate a regularized model with desirable properties, e.g., sparsity, without explicitly using a regularization penalty. The reweighting–AST algorithmic framework developed in Chapter 2 turns out to be suitable for this purpose as well. While this time, instead of speeding up

convergence, the aim becomes to approach a desired solution when there are many. Based on the framework, we propose i) Sparsity-promoting SGD (SSGD) algorithm for neural network compression and ii) Sparsity-promoting Kernel LMS (SKLMS) and Sparsity-promoting Kernel NLMS (SKNLMS) algorithms for dictionary pruning in kernel methods.

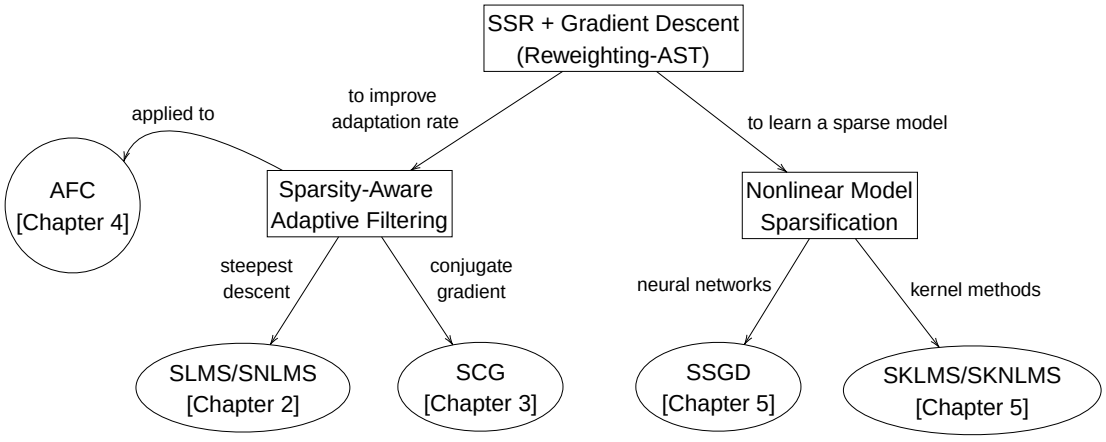


Figure 1.1: Overview of the dissertation.

1.4 Notation

Let $\mathbb{R}^M$ denote the M-dimensional real Euclidean space. $\mathbb{R}^{N\times M}$ denotes the set of $N\times M$ real matrices. $\mathbb{R}_+$ denotes the set of non-negative real numbers. Superscript ^{T} denotes the transpose of a vector or matrix. $\mathrm{E}[\cdot]$ denotes the mathematical expectation. Vectors and matrices are denoted by boldface lowercase and uppercase letters, respectively. Scalars are denoted by italics. For a vector $\mathbf{x} = [x_0, x_1, \ldots, x_{M-1}]^T \in \mathbb{R}^M$, the ℓ_p norm[2] (where $p > 0$) is defined as: $\|\mathbf{x}\|_p = (\sum_{i=0}^{M-1} |x_i|^p)^{1/p}$. We use $\mathrm{diag}\{x_i\}$ to denote the M-by-M diagonal matrix whose i-th diagonal element is x_i. We use $\mathrm{sgn}(\cdot)$ to denote the component-wise sign function. $\nabla_{\mathbf{x}}$ denotes

[2]Note that $\|\mathbf{x}\|_p$ for $0 < p < 1$ does not satisfy the required axioms for a norm and therefore it is not technically a norm. For exposition simplicity, since the range of p considered is from 0 to 2, we use the norm terminology to cover this range.

the gradient operator[3] w.r.t. $\mathbf{x}$. d denotes the differential operator. $\mathcal{R}(\mathbf{X})$, $\mathcal{N}(\mathbf{X})$, and rank($\mathbf{X}$) denote the range space, nullspace, and rank of a matrix $\mathbf{X}$, respectively. tr($\mathbf{X}$) denotes the trace of a square matrix $\mathbf{X}$. $\mathbf{I}$ denotes the identity matrix. $\mathbf{1}$ denotes the vector of all ones. $\mathbf{0}$ denotes the vector of all zeros. We use $\mathcal{N}(\cdot,\cdot)$ to denote the normal distribution with the first and second arguments being the mean and (co)variance, respectively.

[3]By abusing the notation we use $\nabla_{\mathbf{x}}$ also for the subgradient operator without explicitly noting it.

Chapter 2

Proportionate LMS-Type Adaptive Filters Derived Using Iterative Reweighting SSR Techniques

In this chapter, based on sparsity-promoting regularization techniques from the sparse signal recovery (SSR) area, least mean square (LMS)-type sparse adaptive filtering algorithms are derived. The approach mimics the iterative reweighted ℓ_2 and ℓ_1 SSR methods that majorize the regularized objective function during the optimization process. We show that the reweighting formulation naturally leads to an affine scaling transformation (AST) strategy, which effectively introduces a diagonal weighting on the gradient, giving rise to new algorithms that demonstrate improved convergence properties. Interestingly, setting the regularization coefficient to zero in the proposed AST-based framework leads to the Sparsity-promoting LMS (SLMS) and Sparsity-promoting Normalized LMS (SNLMS) algorithms, which exploit but do not strictly enforce the sparsity of the system response if it already exists. The SLMS and SNLMS realize proportionate adaptation for convergence speedup should sparsity be present in the underlying system response.

In this manner, we develop a new way for rigorously deriving a large class of proportionate algorithms, and also explain why they are useful in applications where the underlying systems admit certain sparsity.

2.1 Introduction

Adaptive filters [18, 19, 20, 21] have been an active research area over the past few decades for their capabilities of estimating and tracking time-varying systems. In several applications, the impulse responses (IRs) of the underlying systems to be identified are often sparse or compressible (quasi-sparse), i.e., only a small percentage of the IR components have a significant magnitude while the rest are zero or small. Examples include network and acoustic echo cancellation [27, 54, 23], hands-free mobile telephony [55], acoustic feedback reduction in hearing aids [56, 57, 58], and underwater acoustic communications [59], to mention a few. Designing adaptive filters that can exploit the sparse structure of the underlying system response for performance improvement over the conventional approaches, e.g., the least mean square (LMS) and normalized LMS (NLMS), is of great interest and importance especially for acoustic and speech applications. In this chapter, we utilize the iterative reweighted ℓ_2 and ℓ_1 algorithms that have been developed in the sparse signal recovery (SSR) area to minimize diversity measures as a starting point [13]. By incorporating an affine scaling transformation (AST) strategy [42, 60] into the algorithm design process, a new methodology for developing a large class of adaptive filters is presented that leverage the sparse nature of the system responses.

An early and influential work on identifying sparse IRs is the proportionate NLMS (PNLMS) algorithm proposed by Duttweiler [27] for acoustic echo cancellation. The main idea behind the approach is to update each filter coefficient using a step size proportional to the magnitude of the estimated coefficient, as opposed to the NLMS which assigns a uniform adaptation gain to all coefficients. Consequently, when the system is sparse, larger coefficients

are adapted using relatively large steps compared to the smaller ones with PNLMS. The overall convergence can thus be sped up by focusing on adjusting the significant coefficients, rather then treating them all equally as in NLMS. Although PNLMS was developed in an intuitive way, i.e., the equations used to calculate the proportionate factors that realize step-size control were not based on any optimization criterion but were based on good heuristics, it has motivated many new proportionate variants for sparse system identification. The proportionate class of algorithms represent an important subset among sparsity-aware adaptive filters.

The recent progress on SSR has led to a number of computational algorithms, e.g., [61, 14, 62, 63, 15, 16], among others. This makes available a plethora of approaches for systematically designing sparsity-aware adaptive algorithms that are a natural complement to the SSR batch estimation techniques. As a result, different from the proportionate approaches, another class of sparse adaptive filters have been introduced by utilizing sparsity-inducing regularization to speed up the adaptation of near-zero coefficients in sparse systems. This has led to several sparse adaptive filtering algorithms and even obtaining a general framework of adaptive filters that incorporate sparsity. SSR-motivated adaptive algorithms represent another important class of sparsity-aware adaptive filters. We now discuss a few works on the proportionate class followed by the SSR variants.

Several variations of the PNLMS have been proposed and [26] provides a good summary. Examples include the improved PNLMS (IPNLMS) [28], IPNLMS based on the ℓ_0 "norm"[1] (IPNLMS-ℓ_0) [29], etc. In [64], Martin et al. utilized a natural gradient framework to deduce adaptive filters having similar features to the PNLMS that can exploit the sparse structure. Rao and Song [65] and Jin [66] proposed a framework for promoting sparsity in adaptive filters based on minimizing diversity measures. The framework is quite general and encompasses a broad range of adaptive filtering algorithms having similarity with the PNLMS algorithm. Benesty et al. [67] derived the PNLMS from a different perspective by using a basis pursuit [63] formulation.

[1]The ℓ_0 "norm" of a vector is defined as the number of its nonzero entries. The quotation marks are used to warn that it is not a proper norm.

Following them, Liu and Grant [68] proposed a general framework of proportionate adaptive filters based on convex optimization and sparseness measures, which covers many traditional proportionate algorithms.

Several SSR-inspired algorithms have been introduced by integrating a sparsity-inducing regularizer into the original LMS objective function to accelerate the convergence of near-zero coefficients in sparse systems. For example, Chen et al. [24] proposed the zero-attracting LMS (ZA-LMS) derived by including the ℓ_1 norm penalty in the objective function. They also proposed the reweighted ZA-LMS (RZA-LMS) obtained by incorporating the log-sum penalty. Later, using the approximation of the ℓ_0 "norm" as a sparsity-inducing term, Gu et al. [25] proposed the ℓ_0-LMS that is capable of better estimating sparse systems. In [43], the authors utilized the p-norm-like penalty and considered the quantitative learning of the regularizer. Another work in this area is the new reweighted ℓ_1 norm penalized LMS algorithm proposed and studied in [44] for improving the ZA-LMS and RZA-LMS .

Recently, some works have considered both proportionate adaptation and sparsity-inducing regularization together. For example, [69] presents a modified PNLMS update equation with a zero attractor as in the ZA-LMS for all the taps, derived by introducing a carefully constructed ℓ_1 norm penalty in the PNLMS objective function. Other than the ℓ_1 norm, [70, 71] apply the ℓ_p norm penalty to the PNLMS cost function and derive ℓ_p-norm-constrained proportionate algorithms for improved broadband multipath channel estimation and active noise control. [72] encompasses a number of sparsity-aware adaptive filtering algorithms that go beyond the LMS and NLMS, including proportionate and regularization-based approaches. [73, 74] provide a general framework to combine proportionate updates and sparsity-inducing regularizers. In Section 2.3, we will derive algorithms whose update rules also consist of a proportionate term and another term due to regularization. However, our derivation follows a very different path from these previous works.

In this chapter, inspired by the conceptual similarity with SSR,[2] our goal is to add to this interesting body of work on adaptive filtering and sparsity. The contributions of this work are the following:

1. The sparsity-aware adaptive filters developed lie at the intersection of the proportionate-class and SSR-inspired adaptive algorithms and provide an interesting bridge. We start with the rigorous formulation of a regularization framework and derive novel sparse adaptive filtering algorithms. Specifically, based on diversity measure minimization in SSR, we adopt the iterative reweighted ℓ_2 and ℓ_1 approaches [13] and utilize an AST methodology [42, 60] in the algorithm development, naturally leading to a general class of proportionate adaptive filters. This is an unique feature of this work. The combination of AST and the reweighting frameworks contribute to the main innovation of our adaptive algorithm development framework.

2. Under the proposed framework, we introduce Sparsity-promoting LMS (SLMS) and Sparsity-promoting NLMS (SNLMS) algorithms that promote sparsity without having to bias the adaptation process by adopting $\lambda = 0$, where λ is the regularization coefficient associated with the sparsity penalty. This is not possible for the class of algorithms currently in existence that utilize a sparsity-inducing regularization penalty.[3] The SLMS and SNLMS can be viewed as realizing proportionate adaptation like the PNLMS class of algorithms [27]. Therefore, our framework provides theoretical support to existing proportionate algorithms which were mostly developed based on good heuristics rather than on optimization criteria, and paves the way for explaining why they are useful in circumstances where the channels to be estimated admit certain sparsity. More importantly, unlike most of them that design the proportionate factors heuristically, our SSR-motivated framework leads to a more systematic way of designing the

[2]This similarity has been noticed in [75] where sparse adaptive filtering techniques were utilized for solving the SSR problem. Here we take the opposite direction as we are interested in utilizing SSR techniques for assisting the adaptive filtering algorithms. Both cases exploit the connections between SSR and adaptive filtering but the objectives are different.

[3]The algorithms usually reduce to the standard LMS or NLMS algorithm if the regularization coefficient λ is set to zero.

factors, and permits incorporation of a broad class of diversity measures that have proved effective for SSR in our algorithms.

3. Compared to existing derivations of proportionate-type algorithms, using the proposed framework we derive the algorithms in a more natural way in terms of incorporating sparsity using a regularization framework. For instance, in some of the existing works modified objective functions were proposed that impose sparsity on the "change" of the filter rather than on the filter itself, e.g., [65, 66, 67, 68, 69]. However, since the assumption is that the filter itself is sparse, the motivation for enforcing sparsity on the "change" rather than on the filter is not clear and at best indirect. In contrast, we work with the general mean squared error (MSE) criterion in which sparsity can be directly imposed via regularization on the filter.

4. Steady-state analysis of the proposed algorithms is conducted and simulation results are provided to demonstrate the effectiveness of the proposed algorithms compared to existing approaches. Examples with the acoustic channel response measured on a real-world hearing aid device using speech input are also presented.

Organization of the Chapter: The rest of the chapter is organized as follows. Section 2.2 provides background on adaptive filters and iterative reweighting SSR algorithms. Section 2.3 derives adaptive filters that incorporate sparsity based on diversity measure minimization by utilizing the reweighted ℓ_2 and ℓ_1 frameworks together with the AST methodology. Section 2.4 introduces the SLMS and SNLMS that adopt $\lambda = 0$. Section 2.5 discusses the steady-state analysis. Section 2.6 presents simulation results. Section 2.7 concludes the chapter.

2.2 Adaptive Filtering and SSR

We provide some preliminaries of adaptive filters in the context of system identification and present several examples of existing sparsity-aware adaptive filtering algorithms. We also

discuss the iterative reweighting frameworks in SSR for developing our adaptive algorithms in later sections.

2.2.1 Adaptive Filters for System Identification

Let $\mathbf{h}_n = [h_{0,n}, h_{1,n}, \ldots, h_{M-1,n}]^T$ denote the adaptive filter of length M at discrete time instant n. Assume the IR of the underlying system is $\mathbf{h}^o = [h_0^o, h_1^o, \ldots, h_{M-1}^o]^T$, and the model for the observed or desired signal is $d_n = \mathbf{u}_n^T \mathbf{h}^o + v_n$. $\mathbf{u}_n = [u_n, u_{n-1}, \ldots, u_{n-M+1}]^T$ is the vector containing the M most recent samples of the input signal u_n and v_n is an additive noise signal. The output of the adaptive filter $\mathbf{u}_n^T \mathbf{h}_n$ is subtracted from d_n to obtain the error signal $e_n = d_n - \mathbf{u}_n^T \mathbf{h}_n$. The goal in general is to sequentially update the coefficients of $\mathbf{h}_n$ upon the arrival of a new data pair $(\mathbf{u}_n, d_n)$, such that eventually $\mathbf{h}_n = \mathbf{h}^o$; i.e., to identify the unknown system.

The most classic adaptive filtering algorithms are the LMS and NLMS [18, 19, 20], which can be derived based on minimizing the MSE objective function:

$$\min_{\mathbf{h}} \quad J(\mathbf{h}) \triangleq \mathrm{E}\left[e_n^2\right] = \mathrm{E}\left[\left(d_n - \mathbf{u}_n^T \mathbf{h}\right)^2\right]. \tag{2.1}$$

The method of steepest descent (gradient descent) for optimizing (2.1) suggests the following recursion for updating the filter coefficients [19]:

$$\mathbf{h}_{n+1} = \mathbf{h}_n - \frac{\mu}{2} \nabla_{\mathbf{h}} J(\mathbf{h}_n), \tag{2.2}$$

where $\mu > 0$ is the step size. To develop adaptive algorithms, in practice the gradient $\nabla_{\mathbf{h}} J(\mathbf{h}_n) = -2\mathrm{E}[\mathbf{u}_n e_n]$ is replaced by the instantaneous estimate $-2\mathbf{u}_n e_n$, i.e., the stochastic gradient [19, 20], leading to the standard LMS algorithm:

$$\mathbf{h}_{n+1} = \mathbf{h}_n + \mu \mathbf{u}_n e_n. \tag{2.3}$$

The normalized version of (2.3), i.e., the NLMS algorithm, can be derived based on the *principle of minimum disturbance* [19]. Alternatively, it can be obtained by performing *exact line search* for the optimal step size for each iteration [49]. Then, practically, a scaling factor $\tilde{\mu} > 0$ is introduced to exercise control over the adaptation[4] and a small regularization constant $\delta > 0$ is also employed to avoid division by zero [19], leading to the standard NLMS algorithm:

$$\mathbf{h}_{n+1} = \mathbf{h}_n + \frac{\tilde{\mu}\mathbf{u}_n e_n}{\mathbf{u}_n^T \mathbf{u}_n + \delta}. \tag{2.4}$$

Sparsity-Aware Adaptive Filtering Algorithms

When the underlying system response is sparse, a class of algorithms realizing proportionate adaptation [26] are able to take advantage of the structural sparsity. A typical update rule with proportionate adaptation is:

$$\mathbf{h}_{n+1} = \mathbf{h}_n + \mu\mathbf{\Gamma}_n\mathbf{u}_n e_n, \tag{2.5}$$

or the normalized version:

$$\mathbf{h}_{n+1} = \mathbf{h}_n + \frac{\tilde{\mu}\mathbf{\Gamma}_n\mathbf{u}_n e_n}{\mathbf{u}_n^T\mathbf{\Gamma}_n\mathbf{u}_n + \delta}, \tag{2.6}$$

where

$$\mathbf{\Gamma}_n = \mathrm{diag}\{\gamma_{0,n}, \gamma_{1,n}, \ldots, \gamma_{M-1,n}\} \tag{2.7}$$

is an M-by-M diagonal matrix assigning different weights to the step sizes for different filter taps, referred to as the *proportionate matrix*. It redistributes the adaptation gains among all coefficients and emphasizes the large ones in order to speed up their convergence. Typically, at the n-th

[4]Formally, $\tilde{\mu}$ is called the *normalized* step size. For brevity, we still refer to it as the *step size* but keep in mind that it does not have the same significance as the μ in (2.3). Note that it is also common in the literature that the same notation of the step size is shared for both LMS and NLMS without explicit distinction.

iteration the diagonal entries are computed as:

$$\gamma_{i,n} = \frac{\pi_{i,n}}{\sum_{j=0}^{M-1} \pi_{j,n}}, \tag{2.8}$$

$\forall i = 0, 1, \ldots, M-1$, where $\pi_{i,n}$ is algorithm-dependent and examples of such algorithms include the PNLMS [27], IPNLMS [28], IPNLMS-ℓ_0 [29], etc. In general, if the estimated filter coefficients $h_{i,n}$ are sparse, the resulting $\pi_{i,n}$ (thus $\gamma_{i,n}$) will also tend to be sparsely distributed (with positive values).

Another class of algorithms, inspired by developments in the SSR area, take sparsity into account using a regularization-based approach, e.g., [24, 25, 43, 44]. The algorithms are obtained by adding a sparsity-inducing term $G(\mathbf{h})$ to the MSE objective function:

$$\min_{\mathbf{h}} \quad J^G(\mathbf{h}) \triangleq J(\mathbf{h}) + \lambda G(\mathbf{h}), \tag{2.9}$$

where $\lambda > 0$ is the regularization coefficient. By simply applying (stochastic) gradient descent[5] on (2.9):

$$\mathbf{h}_{n+1} = \mathbf{h}_n + \mu \mathbf{u}_n e_n - \frac{\mu\lambda}{2} \nabla_{\mathbf{h}} G(\mathbf{h}_n), \tag{2.10}$$

various algorithms can be obtained with different sparsity-inducing functions $G(\cdot)$. Examples include the ZA-LMS [24], RZA-LMS [24], and ℓ_0-LMS [25, 76].

2.2.2 Iterative Reweighting Algorithms in SSR

The optimization of (2.9) is actually an SSR problem. The sparsity regularization term $G(\cdot)$ represents the *general diversity measure* that when minimized encourages sparsity in its argument. A *separable* function of the form $G(\mathbf{h}) = \sum_{i=0}^{M-1} g(h_i)$ is commonly used, where $g(\cdot)$ has the following properties [13]:

[5] By abusing the terminology we implicitly use "gradient" also for subgradient whenever appropriate.

Property 1: $g(z)$ is symmetric, i.e., $g(z) = g(-z) = g(|z|)$;

Property 2: $g(|z|)$ is monotonically increasing with $|z|$;

Property 3: $g(0)$ is finite;

Property 4: $g(z)$ is concave in $|z|$ or z^2.

Any function that holds the above properties is a candidate for effective SSR algorithm development.

The concave nature of the regularization penalty $G(\mathbf{h})$ poses challenges to the diversity measure minimization problem (2.9). The iterative reweighted ℓ_2 [14, 15] and ℓ_1 [16] methods are popular batch estimation algorithms for solving such minimization problems in SSR. By introducing a weighted ℓ_2 or ℓ_1 norm term as an upper bound for the diversity measure term in each iteration, they form and solve a convex optimization problem at each step to approach the optimal solution [13]. Specifically, instead of (2.9), at iteration n the reweighted ℓ_2 framework suggests solving:

$$\min_{\mathbf{h}} \quad J_n^{\ell_2}(\mathbf{h}) \triangleq J(\mathbf{h}) + \lambda \|\mathbf{W}_n^{-1}\mathbf{h}\|_2^2, \tag{2.11}$$

and the reweighted ℓ_1 framework suggests solving:

$$\min_{\mathbf{h}} \quad J_n^{\ell_1}(\mathbf{h}) \triangleq J(\mathbf{h}) + \lambda \|\mathbf{W}_n^{-1}\mathbf{h}\|_1, \tag{2.12}$$

where $\mathbf{W}_n = \text{diag}\{w_{i,n}\}$ is positive definite[6] and each $w_{i,n}$ is computed based on the current estimate $h_{i,n}$, depending on which framework (reweighted ℓ_2 or ℓ_1) and diversity measure (choice of $G(\cdot)$) are used.

To elaborate, for using the reweighted ℓ_2 (2.11), the diversity measure function $g(z)$ has to be concave in z^2 for Property 4; i.e., it satisfies $g(z) = f(z^2)$, where $f(z)$ is concave for $z \in \mathbb{R}_+$.

[6]The positive definiteness can be shown to hold for a wide variety of diversity measures used in SSR. In cases where it is not, the positive definiteness can still be ensured by utilizing some small regularization constant.

Based on [13], we have $w_{i,n}$ given as:

$$w_{i,n} = \left(\left. \frac{\mathrm{d}f(z)}{\mathrm{d}z} \right|_{z=h_{i,n}^2} \right)^{-\frac{1}{2}}. \tag{2.13}$$

For using the reweighted ℓ_1 (2.12), $g(z)$ has to be concave in $|z|$ for Property 4; i.e., it satisfies $g(z) = f(|z|)$, where $f(z)$ is concave for $z \in \mathbb{R}_+$. In this case, $w_{i,n}$ is given as:

$$w_{i,n} = \left(\left. \frac{\mathrm{d}f(z)}{\mathrm{d}z} \right|_{z=|h_{i,n}|} \right)^{-1}, \tag{2.14}$$

To utilize the reweighting frameworks, we first choose an appropriate diversity measure $G(\mathbf{h})$ and then use (2.13) or (2.14) to obtain the corresponding update form of $\mathbf{W}_n$. Several examples will be presented is Section 2.4.2.

2.3 Proposed Framework for Incorporating Sparsity in Adaptive Filters

Our framework for developing sparse adaptive filters is also based on (2.9). However, we will be deriving algorithms in a different way rather than using a simple gradient descent as is typically done in existing regularization-based adaptive filtering approaches, e.g., (2.10). Our novel derivation consists of two stages: i) adapting the iterative reweighting frameworks [13] popular in SSR to the adaptive filtering setting, followed by ii) incorporating the AST strategy [42, 60] from the optimization literature to obtain new adaptive filtering algorithms.

2.3.1 Reweighting Methods for Adaptive Filtering

The reweighting methods introduced in Section 2.2.2 actually belong to the more general class of majorization-minimization (MM) algorithms [17]. In each iteration n, the weighted ℓ_2 or

ℓ_1 norm term majorizes $G(\mathbf{h})$ at the current estimate $\mathbf{h}_n$, thereby providing a surrogate function (or majorizer) $J_n^{\ell_2}(\mathbf{h})$ or $J_n^{\ell_1}(\mathbf{h})$ for the regularized objective function $J^G(\mathbf{h})$. Sequentially minimizing the surrogate functions allows the algorithm to produce more focal estimates as optimization progresses. Hopefully when the number of iterations is large enough, the optimal solution can be well approached or even achieved [13].

In SSR, it is typical that the surrogate function is *exactly* minimized in each iteration n. For the purpose of developing adaptive filtering algorithms, here we consider performing *only one step of gradient descent per iteration*. In this sense, it corresponds to the *generalized* MM [77] where one does not need to minimize the majorizer but only to assure that it decreases in every iteration. Indeed, the MM viewpoint provides an interesting observation of using gradient descent for optimizing (2.9) and the reweighting formulations (2.11) and (2.12), as stated in the following proposition:

Proposition 2.1. *For any point* $\mathbf{h}_n$ *at which* $G(\mathbf{h})$ *is differentiable, the gradient vector of the surrogate function* $J_n^{\ell_2}(\mathbf{h})$ *or* $J_n^{\ell_1}(\mathbf{h})$ *evaluated at* $\mathbf{h}_n$ *coincides with that of the regularized objective function* $J^G(\mathbf{h})$*, i.e.,* $\nabla_{\mathbf{h}} J_n^{\ell_2}(\mathbf{h}_n) = \nabla_{\mathbf{h}} J^G(\mathbf{h}_n)$ *for the reweighted* ℓ_2 *case and* $\nabla_{\mathbf{h}} J_n^{\ell_1}(\mathbf{h}_n) = \nabla_{\mathbf{h}} J^G(\mathbf{h}_n)$ *for the reweighted* ℓ_1 *case.*

Proof: Since the surrogate function majorizes $J^G(\mathbf{h})$ at $\mathbf{h}_n$, the tangent plane (supporting hyperplane) of the majorizer coincides with that of $J^G(\mathbf{h})$ at $\mathbf{h}_n$. Consequently, the gradient vectors are the same at $\mathbf{h}_n$.

The observation in Proposition 2.1 implies that, if the gradient descent (when using a fixed step size) is utilized for optimization,[7] then adopting the reweighting frameworks (2.11) and (2.12) will be equivalent to directly working on (2.9) and lead to the existing regularization-based algorithms such as the ZA-LMS. In the following, we introduce the AST strategy naturally suggested by the reweighting frameworks, leading to new algorithms markedly different from

[7]For a point at which $G(\mathbf{h})$ is non-differentiable, this can still hold by properly choosing the subgradients.

those obtained by directly optimizing (2.9) with gradient descent.

2.3.2 AST-Based Adaptive Filtering Algorithms

The reweighting frameworks (2.11) and (2.12) naturally suggest the following reparameterization in terms of the (affinely) scaled variable $\mathbf{q}$:

$$\mathbf{q} \triangleq \mathbf{W}_n^{-1}\mathbf{h}. \tag{2.15}$$

This step can be interpreted as the AST commonly employed by the interior point approach to solving linear and nonlinear programming problems [42], where $\mathbf{W}_n$ is used as the *scaling matrix*. It is pre-calculated and treated as a given matrix at iteration n to perform a change of coordinates (variables) [78] from $\mathbf{h}$ to $\mathbf{q}$, acting as a scaling technique in gradient descent methods [79]. In the optimization literature, AST-based methods transform the original problem into an equivalent one, favorably positioning the current point at the center of the feasible region for expediting the optimization process [60]. While we do not claim this argument is rigorous in the context of adaptive filtering, where the convergence behavior is hard to characterize due to the nonlinear nature of the update equations and the long term dependency on the data, the numerical results appear to support this observation of enjoying the benefits of AST for convergence speedup.

Now we apply (2.15) to reparameterize the objective functions $J_n^{\ell_2}(\mathbf{h})$ and $J_n^{\ell_1}(\mathbf{h})$ and perform minimization w.r.t. $\mathbf{q}$, that is:

$$\min_{\mathbf{q}} \quad \tilde{J}_n^{\ell_2}(\mathbf{q}) \triangleq J_n^{\ell_2}(\mathbf{W}_n\mathbf{q}) = J(\mathbf{W}_n\mathbf{q}) + \lambda\|\mathbf{q}\|_2^2 \tag{2.16}$$

and

$$\min_{\mathbf{q}} \quad \tilde{J}_n^{\ell_1}(\mathbf{q}) \triangleq J_n^{\ell_1}(\mathbf{W}_n\mathbf{q}) = J(\mathbf{W}_n\mathbf{q}) + \lambda\|\mathbf{q}\|_1, \tag{2.17}$$

for the reweighted ℓ_2 and ℓ_1 cases, respectively. A gradient descent procedure will then be applied.

The overall update process conceptually can be summarized as follows: i) given an $\mathbf{h}$ compute $\mathbf{W}_n$ followed by reparameterization $\mathbf{q}$ as (2.15). ii) Update $\mathbf{q}$ using a gradient descent algorithm. iii) Use this new $\mathbf{q}$ to obtain the updated $\mathbf{h}$. iv) Repeat Steps i)–iii) till convergence.

More formally, to proceed with gradient-based updates, following [49] we define the *a posteriori* AST variable at time n:

$$\mathbf{q}_{n|n} \triangleq \mathbf{W}_n^{-1}\mathbf{h}_n \tag{2.18}$$

and the *a priori* AST variable at time n:

$$\mathbf{q}_{n+1|n} \triangleq \mathbf{W}_n^{-1}\mathbf{h}_{n+1}. \tag{2.19}$$

The recursive update by using gradient descent in the $\mathbf{q}$ domain can be formulated as:

$$\mathbf{q}_{n+1|n} = \mathbf{q}_{n|n} - \frac{\mu}{2}\nabla_{\mathbf{q}}\tilde{J}_n^{\ell_2}(\mathbf{q}_{n|n}) \tag{2.20}$$

and

$$\mathbf{q}_{n+1|n} = \mathbf{q}_{n|n} - \frac{\mu}{2}\nabla_{\mathbf{q}}\tilde{J}_n^{\ell_1}(\mathbf{q}_{n|n}), \tag{2.21}$$

for optimizing (2.16) and (2.17), respectively.

Using the chain rule[8] and the AST relationships (2.15) and (2.18), we can write (2.20) and (2.21) respectively as:

$$\mathbf{q}_{n+1|n} = \mathbf{q}_{n|n} - \frac{\mu}{2}\mathbf{W}_n\nabla_{\mathbf{h}}J_n^{\ell_2}(\mathbf{h}_n) \tag{2.22}$$

and

$$\mathbf{q}_{n+1|n} = \mathbf{q}_{n|n} - \frac{\mu}{2}\mathbf{W}_n\nabla_{\mathbf{h}}J_n^{\ell_1}(\mathbf{h}_n). \tag{2.23}$$

Premultiplying $\mathbf{W}_n$ on both sides of (2.22) and (2.23) and noting the relationships (2.18) and (2.19), we transform the $\mathbf{q}$ domain updates (2.22) and (2.23) back to the $\mathbf{h}$ domain respectively

[8]Note that the chain rule here is basically $\nabla_{\mathbf{q}} = \mathbf{W}_n\nabla_{\mathbf{h}}$ as a result of the change of variables (2.15) for a given $\mathbf{W}_n$ at iteration n.

as:

$$\mathbf{h}_{n+1} = \mathbf{h}_n - \frac{\mu}{2}\mathbf{W}_n^2 \nabla_{\mathbf{h}} J_n^{\ell_2}(\mathbf{h}_n) \tag{2.24}$$

and

$$\mathbf{h}_{n+1} = \mathbf{h}_n - \frac{\mu}{2}\mathbf{W}_n^2 \nabla_{\mathbf{h}} J_n^{\ell_1}(\mathbf{h}_n). \tag{2.25}$$

By Proposition 2.1, we can replace $\nabla_{\mathbf{h}} J_n^{\ell_2}(\mathbf{h}_n)$ and $\nabla_{\mathbf{h}} J_n^{\ell_1}(\mathbf{h}_n)$ with $\nabla_{\mathbf{h}} J^G(\mathbf{h}_n)$. Thus, (2.24) and (2.25) can both be written as:

$$\mathbf{h}_{n+1} = \mathbf{h}_n - \frac{\mu}{2}\mathbf{W}_n^2 \nabla_{\mathbf{h}} J^G(\mathbf{h}_n). \tag{2.26}$$

Note that based on the aforementioned update process i)-iv), we can in fact directly apply (2.15) to reparameterize $J^G(\mathbf{h})$ to obtain (2.26) without going through the reweighting formulation, as long as the scaling matrix $\mathbf{W}_n$ is specified. In this sense, the reweighing methods essentially play the role of suggesting a suitable $\mathbf{W}_n$ that eventually becomes a diagonal *weighting matrix* $\mathbf{W}_n^2$ on the gradient $\nabla_{\mathbf{h}} J^G(\mathbf{h}_n)$ in the update rule. Hopefully, it alters the ordinary descent direction in such a way that leads to convergence improvement. We should also emphasize that the scaling matrix $\mathbf{W}_n$ suggested by (2.24) and (2.25) will in general be different for a given $G(\mathbf{h})$ despite the fact that both can be expressed as (2.26).

In practice, the following update rule is suggested over (2.26) for avoiding instability and slow convergence issues:

$$\mathbf{h}_{n+1} = \mathbf{h}_n - \frac{\mu}{2}\mathbf{S}_n \nabla_{\mathbf{h}} J^G(\mathbf{h}_n), \tag{2.27}$$

where

$$\mathbf{S}_n = \frac{\mathbf{W}_n^2}{\frac{1}{M}\mathrm{tr}\left(\mathbf{W}_n^2\right)}, \tag{2.28}$$

referred to as the *sparsity-promoting matrix*, is the normalized version of $\mathbf{W}_n^2$. As a fixed step size μ is used, performing normalization of the weighting matrix compensates for any arbitrary scaling

inherent in $\mathbf{W}_n^2$ that might cause instability (scaling too large) or slow convergence (scaling too small). Note that by (2.28) we always have $\mathrm{tr}(\mathbf{S}_n) = M$, aligned with the non-AST case (i.e., using the ordinary gradient descent) which essentially has $\mathbf{S}_n = \mathbf{I}$ whose trace is also M.

Finally, to obtain the adaptive algorithm, we follow the standard procedure of replacing $\nabla_{\mathbf{h}} J^G(\mathbf{h}_n) = -2\mathrm{E}[\mathbf{u}_n e_n] + \lambda \nabla_{\mathbf{h}} G(\mathbf{h}_n)$ in (2.27) with its instantaneous estimate $-2\mathbf{u}_n e_n + \lambda \nabla_{\mathbf{h}} G(\mathbf{h}_n)$, leading to:

$$\boxed{\mathbf{h}_{n+1} = \mathbf{h}_n + \mu \mathbf{S}_n \mathbf{u}_n e_n - \frac{\mu\lambda}{2} \mathbf{S}_n \nabla_{\mathbf{h}} G(\mathbf{h}_n).} \tag{2.29}$$

We see that there is a term with a diagonal weighting $\mathbf{S}_n$ on the LMS update vector $\mathbf{u}_n e_n$, similar to that in proportionate algorithms (2.5) and (2.6). We also see another term weighted by λ which is due to the introduction of the regularizer, like that of (2.10). Therefore, the AST framework leads to a more general algorithm comprised of proportionate adaptation and sparsity-inducing regularization. We thus refer to (2.29) as the *generalized sparse LMS* algorithm.

2.3.3 Discussions

It may seem at the first glance that applying the reweighting techniques to (2.9) straightforwardly leads to our algorithm. We stress that it is not true. If the AST (2.15) was not considered, adopting the reweighting schemes would still end up with an update rule like (2.10) according to Proposition 2.1, rather than the proposed (2.29). It is also worth mentioning that there is considerable difference between the proposed algorithm (2.29) and existing SSR algorithms based on (2.11) and (2.12) – the conventional SSR techniques are batch estimation methods for recovering the underlying sparse representation, while the proposed algorithm is specifically tailored for the adaptive filtering scenario. That being said, as gradient descent is adopted for optimization, we actually perform a gradual update of the filter coefficients in each iteration n, rather than looking for an exact minimizer of the surrogate function as is typically pursued in SSR. This enables the algorithm to track temporal variations and environmental changes. Certainly,

considering the gradient noise in real scenarios, it may post the issue of whether the algorithm is convergent. However, even the standard LMS and NLMS that are based on gradual updates, work well in many practical situations with gradient noise. In Section 2.6, experimental results will demonstrate that the proposed algorithm, like the LMS and NLMS, also behaves well when certain level of environmental noise is present.

Finally, the following theorem establishes the convergence of the **q** domain recursions (2.20) and (2.21) and their relationships to (2.9) to shed light on the convergence of the adaptive algorithm (2.29) developed based on them:

Theorem 2.1. *For the objective function $J^G(\mathbf{h})$ in (2.9) with the general diversity measure $G(\mathbf{h})$ satisfying Properties 1-4 in Section 2.2.2,[9] there exists a step size sequence $\{\mu_n\}_{n=0}^{\infty}$ such that each of the update recursions (2.20) and (2.21) monotonically converges to a local minimum (or saddle point) of (2.9) under a wide-sense stationary (WSS) environment, i.e., $\mathbf{u}_n$ and d_n are jointly WSS.*

Proof: See Appendix 2.8.1.

2.4 Sparsity-Promoting Algorithms Adopting $\lambda = 0$

An interesting situation arises when we consider the limiting case of $\lambda \to 0^+$ for the proposed framework. By setting $\lambda = 0$ in (2.29), we see the λ-weighted term due to regularization vanishes, leading to a simpler equation:

$$\boxed{\mathbf{h}_{n+1} = \mathbf{h}_n + \mu \mathbf{S}_n \mathbf{u}_n e_n.} \tag{2.30}$$

The main feature of (2.30) is that it is able to *promote* sparsity of the system (through $\mathbf{S}_n$) if it already exists while *not strictly enforcing* it (as $\lambda = 0$). This property shall become clearer in later discussions. We refer to the algorithm (2.30) as the Sparsity-promoting LMS (SLMS).

[9] Note that for Property 4, Theorem 2.1 holds for (2.20) of the reweigthed ℓ_2 framework if $g(z)$ is concave in z^2. On the other hand, it holds for (2.21) of the reweighted ℓ_1 framework if $g(z)$ is concave in $|z|$.

The normalized version of (2.30) can also be developed by performing exact line search for the optimal step size at iteration n just like that when deriving the NLMS:

$$\mu_n = \arg\min_{\mu} \left(d_n - \mathbf{u}_n^T \left(\mathbf{h}_n + \mu \mathbf{S}_n \mathbf{u}_n e_n \right) \right)^2 = \frac{1}{\mathbf{u}_n^T \mathbf{S}_n \mathbf{u}_n}. \tag{2.31}$$

Similar to the NLMS, we introduce $\tilde{\mu} > 0$ to exercise control over the adaptation and $\delta > 0$ to avoid division by zero, resulting in:

$$\boxed{\mathbf{h}_{n+1} = \mathbf{h}_n + \frac{\tilde{\mu} \mathbf{S}_n \mathbf{u}_n e_n}{\mathbf{u}_n^T \mathbf{S}_n \mathbf{u}_n + \delta}.} \tag{2.32}$$

We refer to the algorithm (2.32) as the Sparsity-promoting NLMS (SNLMS).

An obvious benefit of adopting $\lambda = 0$ is that the computation for the term due to regularization is no longer needed, and we do not have to tweak this coefficient anymore (which is typically not a trivial task in practice). Still, the SLMS and SNLMS have the ability to leverage sparsity owing to the diagonal weighting $\mathbf{S}_n$, which is similar to the proportionate matrix $\mathbf{\Gamma}_n$ in (2.5) and (2.6). Again, this is made possible due to the use of the AST (2.15), wherein the gradient descent update is performed w.r.t. the $\mathbf{q}$ variable rather than in the original $\mathbf{h}$ domain. Otherwise, we will end up with algorithms like (2.10) that reduce to the ordinary LMS/NLMS when using $\lambda = 0$.

The SLMS and SNLMS can in fact be viewed as a broader class of proportionate algorithms. Actually, with certain choices of diversity measures and corresponding parameters, we can have the PNLMS (approximately) as a special case. For example, as we will see in Section 2.4.2, using $p = 1$ in (2.34) for $\mathbf{W}_n$, the sparsity-promoting matrix $\mathbf{S}_n$ approximates the proportionate matrix $\mathbf{\Gamma}_n$ of the PNLMS. Indeed, one of the main advantages of the SLMS and SNLMS is their ability to incorporate flexible diversity measures. It allows the algorithms to fit the sparsity level of the system response by optimizing corresponding sparsity control parameters in a more informed manner due to the underlying connections to SSR. Furthermore, the derivations provide

theoretical support to the class of proportionate algorithms that were mostly motivated based on heuristics, explaining why they are useful in practical identification tasks with sparse channels, e.g., in acoustic echo/feedback cancellation, from an SSR viewpoint.

2.4.1 Interpretation of $\lambda = 0$ from Optimization Perspective

We further discuss the interpretation of using $\lambda = 0$ in our framework from an optimization perspective. Recall that the AST reparameterization (2.15) results in the optimization problems (2.16) and (2.17). Setting $\lambda = 0$ leads both to:

$$\min_{\mathbf{q}} J(\mathbf{W}_n \mathbf{q}). \tag{2.33}$$

This actually applies a change of coordinates to the unregularized problem (2.1) via (2.15). Since $\mathbf{W}_n$ is invertible, the problem of finding the $\mathbf{h}$ that minimizes $J(\mathbf{h})$ is equivalent to finding the $\mathbf{q}$ which minimizes $J(\mathbf{W}_n \mathbf{q})$. Therefore, the advantage of solving (2.33) is that the solution is guaranteed to also be a solution of (2.1), which is not true for (2.9) with $\lambda > 0$. Thus, the optimization is *unbiased* while promoting sparsity – it is able to take advantage of sparsity whereas without having to supplement a sparsity penalty that incurs bias to the MSE objective. As noted in [79], the performance of gradient-based methods is dependent on the parameterization – a new choice may substantially alter convergence characteristics. Introducing variable scalings may speed up convergence by altering the descent direction toward the optimum. In our case, solving (2.33) with appropriately selected $\mathbf{W}_n$ can expedite the adaptation procedure toward the optimum of (2.1).

This observation can also be illustrated by looking at (2.9) which indicates a trade-off between estimation quality, as reflected in the MSE objective function, and solution sparsity as controlled by λ. In the limiting case of $\lambda \to 0^+$, the objective function exerts diminishing impact on enforcing sparsity on the solution, meaning that eventually no sparse solution is favored

over other possible solutions. To elaborate, with $\lambda = 0$ and under a WSS environment, all the algorithms derived from (2.9) minimize the MSE and converge toward the Wiener-Hopf solution. However, not surprisingly, the path they take is different and depends on how the iterations are developed. If the Wiener-Hopf solution is sparse, then all will converge toward the same sparse solution asymptotically. Interestingly, the SLMS and SNLMS, because of their proportionate nature similar to the PNLMS-type algorithms, can take advantage of the sparsity and are capable of speeding up convergence without compromising estimation quality should sparsity be present. This observation will later be supported by experimental results in Section 2.6.2.

2.4.2 Example Diversity Measures and Corresponding $\mathbf{W}_n$

To illustrate the flexibility of the proposed framework, we provide example algorithms instantiated with popular diversity measures that have proved effective in SSR.

Consider the p-norm-like diversity measure with $g(h_i) = |h_i|^p$, $0 < p \leq 2$ for the reweighted ℓ_2 framework [14, 42]. Using (2.13) leads to the update form of $\mathbf{W}_n$:

$$w_{i,n} = \left(\frac{2}{p} \left(|h_{i,n}| + c \right)^{2-p} \right)^{\frac{1}{2}}. \tag{2.34}$$

Note that we have empirically added a small regularization constant $c > 0$ for avoiding algorithm stagnation and instability,[10] which also ensures the positive definiteness of $\mathbf{W}_n$ [80]. The parameter $p \in (0, 2]$ in (2.34) is responsible for controlling the sparsity degree, as the p-norm-like diversity measure is associated with super-Gaussian prior distributions. In general, a smaller p corresponds to a heavier-tailed distribution, encouraging stronger sparsity in the parameters. It is worth noting that using $p \to 1$ in (2.34) results in a proportionate factor close to that of the PNLMS. On the other hand, letting $p = 2$ recovers the standard LMS/NLMS.

The p-norm-like diversity measure can also be adopted in the reweighted ℓ_1 framework if

[10] We suggest that c be kept relatively small as compared to the amplitude of the filter coefficients so that it would not affect the convergence significantly.

$0 < p \leq 1$. Applying (2.14), we obtain the update form of $\mathbf{W}_n$ in this case:

$$w_{i,n} = \frac{1}{p}\left(|h_{i,n}| + c\right)^{1-p}. \tag{2.35}$$

Again, a small constant $c > 0$ is added. The sparsity control parameter of (2.35) is now $p \in (0, 1]$. In this case, using $p \to 0.5$ in (2.35) results in a proportionate factor close to that of the PNLMS, whereas letting $p = 1$ recovers the standard LMS/NLMS.

We can also consider the log-sum penalty with $g(h_i) = \log(h_i^2 + \epsilon)$, $\epsilon > 0$ for the reweighted ℓ_2 framework [15]. The function is readily amenable to the use of (2.13) to obtain the update form of $\mathbf{W}_n$ as:

$$w_{i,n} = \left(h_{i,n}^2 + \epsilon\right)^{\frac{1}{2}}. \tag{2.36}$$

Or consider the log-sum penalty with $g(h_i) = \log(|h_i| + \epsilon)$, $\epsilon > 0$ for the reweighted ℓ_1 framework [16]. Using (2.14), the update form of $\mathbf{W}_n$ becomes:

$$w_{i,n} = |h_{i,n}| + \epsilon. \tag{2.37}$$

The sparsity control parameter is $\epsilon > 0$ for the two log-sum penalty cases. From (2.36) and (2.37) we can see that ϵ controls how much proportionate adaptation is encouraged: as ϵ becomes smaller, the term $h_{i,n}^2$ or $|h_{i,n}|$ becomes more dominant. Consequently, they exhibit a stronger proportionate adaptation characteristic. On the contrary, as ϵ becomes larger, the influence of $h_{i,n}^2$ or $|h_{i,n}|$ reduces. Thus, the algorithm will approach the standard LMS/NLMS when $\epsilon \gg h_{i,n}^2$ or $\epsilon \gg |h_{i,n}|$. In practice, one can start from a large ϵ and reduce it to find a suitable value.

More example functions can be found in [81, 68], including $g(h_i) = \arctan(|h_i|/\epsilon)$, $\epsilon > 0$ also suggested in [16], which works for both the reweighted ℓ_2 and ℓ_1 frameworks. Note that different diversity measures can result in different computational complexity for calculating $\mathbf{W}_n$. Notably, for example, the p-norm-like function resulting in (2.34) or (2.35) might incur extra

computation for calculating the quantity to the power $2-p$ or $1-p$ for some p value (e.g., non-integer power).

Algorithm 1 summarizes the proposed SLMS and SNLMS algorithms.

Algorithm 1: The proposed SLMS and SNLMS adaptive filtering algorithms

Input: step size $\mu > 0$ (or $\tilde{\mu} > 0$), regularization constant $\delta > 0$, input signal $\mathbf{u}_n$, desired signal d_n, and the choice of the diversity measure
Output: estimated filter $\mathbf{h}_n$
Initialize: $\mathbf{h}_0$
for $n = 0, 1, 2, \ldots$ **do**
    Compute error signal: $e_n = d_n - \mathbf{u}_n^T \mathbf{h}_n$
    Compute scaling matrix: $\mathbf{W}_n$ according to the specified diversity measure (e.g., using (2.34), (2.35), (2.36), or (2.37))
    Compute sparsity-promoting matrix: $\mathbf{S}_n$ by (2.28)
    Update adaptive filter coefficients:
        * SLMS: $\mathbf{h}_{n+1} = \mathbf{h}_n + \mu \mathbf{S}_n \mathbf{u}_n e_n$
        * SNLMS: $\mathbf{h}_{n+1} = \mathbf{h}_n + \dfrac{\tilde{\mu} \mathbf{S}_n \mathbf{u}_n e_n}{\mathbf{u}_n^T \mathbf{S}_n \mathbf{u}_n + \delta}$
end for

2.4.3 Comparison to Existing Work on PNLMS-Type Algorithms

Note that in IPNLMS [28] and IPNLMS-ℓ_0 [29] there is also a parameter for fitting the sparsity degree, which was heuristically introduced to weight between proportionate and non-proportionate updates. However, this empirical parameter does not reflect the sparsity level of the underlying system directly. In our algorithms, we have the sparsity control parameters that play a similar role for fitting different sparsity levels. However, based on diversity measures in SSR, they have direct connections to the system sparsity, thereby offering a more intuitive parameter selection procedure. Our algorithms thus have the advantages of enjoying theoretical support and leveraging sparsity more straightforwardly.

In terms of algorithm derivations, PNLMS-type algorithms were mostly developed

from a constrained optimization problem following the principle of minimal disturbance, e.g., [65, 66, 67, 68, 69], in which modified objective functions have been proposed that impose sparsity on the "change" of the filter rather than on the filter itself. For example, [65, 66, 69] considered enforcing sparsity on the difference between the current and updated filters; [67, 68] imposed sparsity on the so-called *correctness component* as defined in [67] which also represents the change in the filter coefficients. However, since the assumption is that the filter itself is sparse rather than the difference between successive updates, the motivation to enforce sparsity on the "change" of the filter is less clear. Sparsity, in turn, does not seem to fit in straightforwardly under the commonly adopted constrained optimization framework. In contrast, we work with the general MSE criterion in which filter sparsity can be directly imposed via regularization, which is more straightforward and also makes intuitive sense.

2.5 Steady-State Performance Analysis

The signal model of system identification described in Section 2.2.1 is employed for performance analysis. We further assume the noise v_n is i.i.d. according to $\mathcal{N}(0, \sigma_v^2)$. We also introduce several other assumptions useful for simplifying analysis. Although these assumptions may seem restrictive, they make meaningful analysis possible without significant loss of insight and are also commonly adopted in the literature. We shall later see that these assumptions lead to theoretical results that are supported by experiments.

Assumption 1: *The input data vector* $\mathbf{u}_n$ *is independent of* $\mathbf{u}_k$ *for* $n \neq k$. *Furthermore,* $\mathbf{u}_n$ *is independent of* v_k *for all* n *and* k. In practice and from past experience in adaptive filters, this assumption simplifies the analysis and does lead to useful insights [19, 20], despite the fact that it does not in general hold true.

Assumption 2: *The input data vector obeys* $\mathbf{u}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$ *for all* n. This technical assumption facilitates the analysis by taking advantage of the useful results on Gaussian random

variables [21].

Assumption 3: *At steady-state, the diagonal matrix* $\mathbf{W}_n$ *in the update equations can be view as a fixed matrix.* As suggested in [27, 64], when the system is at steady-state and when the step size is sufficiently small, the coefficients converge in both mean and mean squared senses. Thus, replacing $\mathbf{W}_n$ by a fixed matrix becomes reasonable and convenient.

For convenience we shall consider the algorithm of the following form for performance analysis:

$$\mathbf{h}_{n+1} = \mathbf{h}_n + \mu \mathbf{S}\mathbf{u}_n e_n, \tag{2.38}$$

where $\mathbf{S} = \text{diag}\{s_i\}$ with $s_i > 0, \forall i = 0, 1, \ldots, M-1$.

For a fixed underlying system $\mathbf{h}^o$, define the steady-state excess MSE [21]:

$$J_{\text{ex}} \triangleq \lim_{n\to\infty} \text{E}\left[\left(\mathbf{u}_n^T\left(\mathbf{h}^o - \mathbf{h}_n\right)\right)^2\right]. \tag{2.39}$$

Under Assumption 1, we have the steady-state MSE:

$$J \triangleq \lim_{n\to\infty} \text{E}\left[e_n^2\right] = \sigma_v^2 + J_{\text{ex}}. \tag{2.40}$$

The following theorems characterize the steady-state behavior of (2.38):

Theorem 2.2 (Steady-state excess MSE)**.** *Under Assumptions 1-2 with a sufficiently small* μ *and assume* $\mathbf{R} = \sigma_u^2\mathbf{I}$*, for the adaptive filter (2.38), the steady-state excess MSE is given by:*

$$J_{ex} = \frac{\mu \sum_{i=0}^{M-1} \frac{\sigma_u^2 s_i}{2-2\mu\sigma_u^2 s_i}}{1 - \mu \sum_{i=0}^{M-1} \frac{\sigma_u^2 s_i}{2-2\mu\sigma_u^2 s_i}} \sigma_v^2. \tag{2.41}$$

Proof: See Appendix 2.8.2.

Theorem 2.3 (Convergence conditions)**.** *Under Assumptions 1-2 with a sufficiently small μ and assume $\mathbf{R} = \sigma_u^2\mathbf{I}$, for the adaptive filter (2.38):*

i) It converges in the mean sense if:

$$|\lambda_{\max}\{\mathbf{I} - \mu\sigma_u^2\mathbf{S}\}| < 1, \tag{2.42}$$

where $\lambda_{\max}\{\mathbf{X}\}$ denotes the largest eigenvalue of a square matrix $\mathbf{X}$ in magnitude.

ii) It converges in the mean squared sense if:

$$0 < \mu < \left(\sum_{i=0}^{M-1} \frac{\sigma_u^2 s_i}{2 - 2\mu\sigma_u^2 s_i}\right)^{-1}. \tag{2.43}$$

Proof: See Appendix 2.8.3.

2.5.1 Steady-State Performance of SLMS

Consider the case where Assumptions 1-3 are in position and $\mathbf{R} = \sigma_u^2\mathbf{I}$. For analyzing the proposed SLMS (2.30), first we need to recognize an appropriate $\mathbf{S}$ with regard to Assumption 3. A useful approximation at steady-state is to replace the occurrence of $\mathbf{h}_n$ by the true system $\mathbf{h}^o$; that is, to use $\mathbf{S} = \frac{\mathbf{W}^2}{\frac{1}{M}\mathrm{tr}(\mathbf{W}^2)}$, where $\mathbf{W} = \mathrm{diag}\{w_i\}$ with w_i given by (2.13) for the reweighted ℓ_2 case, or by (2.14) for the reweighted ℓ_1 case, both computed based on the corresponding true coefficient h_i^o. Now, since $\mathrm{tr}(\mathbf{S}) = M$, the excess MSE (2.41) can be approximated as:

$$J_{\mathrm{ex}} \approx \frac{\mu\sum_{i=0}^{M-1}\frac{\sigma_u^2 s_i}{2}}{1 - \mu\sum_{i=0}^{M-1}\frac{\sigma_u^2 s_i}{2}}\sigma_v^2 = \frac{\mu\frac{\sigma_u^2}{2}\mathrm{tr}(\mathbf{S})}{1 - \mu\frac{\sigma_u^2}{2}\mathrm{tr}(\mathbf{S})}\sigma_v^2 = \frac{\mu}{\frac{2}{M\sigma_u^2} - \mu}\sigma_v^2, \tag{2.44}$$

where for the approximation we assume a sufficiently small step size μ such that $2\mu\sigma_u^2 s_i \ll 2$, $\forall i = 0, 1, \ldots, M-1$.

Now, for the mean squared convergence condition, although the upper bound in (2.43) of Theorem 2.3 contains μ itself, after some inspection it is clear that the lowest stability limit on μ occurs when $\mathbf{S}$ has its diagonal elements nonzero at one tap position (with a value of M) and zero at all others [27]. With such an $\mathbf{S}$, it leads to:

$$0 < \mu < \frac{2}{3M\sigma_u^2}. \tag{2.45}$$

On the other hand, the largest stability limit is associated with a proportionate matrix assigning equal gains at each position [27], i.e., $\mathbf{S} = \text{diag}\{s_i\}$ with $s_i = 1$, $\forall i = 0, 1, \ldots, M-1$. With such an $\mathbf{S}$ we have:

$$0 < \mu < \frac{2}{(2+M)\sigma_u^2}. \tag{2.46}$$

For a large M, the largest stability limit can be approximated as $\frac{2}{M\sigma_u^2} = \frac{2}{\text{tr}(\mathbf{R})}$, which is also the stability limit of the LMS [21]. This result is not surprising since using an $\mathbf{S}$ that assigns uniform gains essentially becomes the LMS.

2.5.2 Steady-State Performance of SNLMS

Consider the case where Assumptions 1-3 are in position and $\mathbf{R} = \sigma_u^2\mathbf{I}$. For analyzing the proposed SNLMS (2.32), first we must identify a fixed $\mathbf{S}$ to approximate the term $\frac{\mathbf{S}_n}{\mathbf{u}_n^T\mathbf{S}_n\mathbf{u}_n}$ (where we have ignored δ), for which an exact characterization seems difficult, if at all possible, to obtain. However, if we fix $\mathbf{W}_n = \mathbf{W}$ at steady-state by Assumption 3, where $\mathbf{W}$ is again computed based on the true system $\mathbf{h}^o$, then we have:

$$\mathbf{S} = \frac{\frac{\mathbf{W}^2}{\frac{1}{M}\text{tr}(\mathbf{W}^2)}}{\mathbf{u}_n^T\left(\frac{\mathbf{W}^2}{\frac{1}{M}\text{tr}(\mathbf{W}^2)}\right)\mathbf{u}_n} = \frac{\mathbf{W}^2}{\mathbf{u}_n^T\mathbf{W}^2\mathbf{u}_n} \approx \frac{\mathbf{W}^2}{\sigma_u^2\text{tr}\left(\mathbf{W}^2\right)}, \tag{2.47}$$

with the approximation $\mathbf{u}_n^T\mathbf{W}^2\mathbf{u}_n \approx \mathrm{E}\left[\mathbf{u}_n^T\mathbf{W}^2\mathbf{u}_n\right] = \sigma_u^2\mathrm{tr}\left(\mathbf{W}^2\right)$ utilized. A useful fact of (2.47) is that $\mathrm{tr}(\mathbf{S}) = (\sigma_u^2)^{-1}$. We can thus use the following approximation for (2.41) to express the excess MSE (and replace μ by $\tilde{\mu}$):

$$\begin{aligned} J_{\mathrm{ex}} \approx & \frac{\tilde{\mu}\sum_{i=0}^{M-1}\frac{\sigma_u^2 s_i}{2}}{1-\tilde{\mu}\sum_{i=0}^{M-1}\frac{\sigma_u^2 s_i}{2}}\sigma_v^2 = \frac{\tilde{\mu}\sigma_u^2\sum_{i=0}^{M-1}s_i}{2-\tilde{\mu}\sigma_u^2\sum_{i=0}^{M-1}s_i}\sigma_v^2 = \frac{\tilde{\mu}\sigma_u^2\mathrm{tr}(\mathbf{S})}{2-\tilde{\mu}\sigma_u^2\mathrm{tr}(\mathbf{S})}\sigma_v^2 \\ = & \frac{\tilde{\mu}\sigma_u^2(\sigma_u^2)^{-1}}{2-\tilde{\mu}\sigma_u^2(\sigma_u^2)^{-1}}\sigma_v^2 = \frac{\tilde{\mu}}{2-\tilde{\mu}}\sigma_v^2, \end{aligned} \tag{2.48}$$

for $\tilde{\mu}$ sufficiently small such that $2\tilde{\mu}\sigma_u^2 s_i \ll 2, \forall i = 0, 1, \ldots, M-1$,

For the mean squared convergence condition, using the same argument as in the SLMS case for (2.45) and (2.46), we can obtain the lowest stability limit as:

$$0 < \tilde{\mu} < \frac{2}{3} \tag{2.49}$$

and the largest stability limit as:

$$0 < \tilde{\mu} < \frac{2}{1+\frac{2}{M}}. \tag{2.50}$$

For a large M, we have (2.50) approximately as $0 < \tilde{\mu} < 2$, which is the classic result of the NLMS [21].

2.6 Simulation Results

The proposed algorithms are evaluated using computer simulations in MATLAB. We consider three system IRs as shown in Figure 2.1 which represent different sparsity levels: quasi-sparse, sparse, and dispersive systems. The IR of the quasi-sparse system is an acoustic feedback path between the microphone and the loudspeaker of a hearing aid that was measured from a real-world scenario. It represents a typical IR of many practical system identification problems where certain degree of structural sparsity exists. The sparse and dispersive IRs were

artificially generated. Each of these IRs has 256 taps. We conducted experiments to obtain the MSE learning curves (i.e., the ensemble average of e_n^2 as a function of iteration n) for performance comparison. The ensemble averaging was performed over 1000 independent Monte Carlo runs for obtaining each curve. In all experiments, the adaptive filter coefficients were initialized with all zeros. For the input signal, we mainly consider two types of u_n for theoretical analysis: i) a zero mean, unit variance white Gaussian process and ii) a first order autoregressive (AR) process according to $u_n = \rho u_{n-1} + \eta_n$, where $\rho = 0.8$ and η_n is i.i.d. according to $\mathcal{N}(0,1)$. We also include results of speech inputs for demonstrating the algorithm performance with non-stationary signals. The system noise v_n is i.i.d. according to $\mathcal{N}(0, 0.01)$. Regarding the algorithms, when using (2.34) for updating $\mathbf{W}_n$, a small positive constant $c = 0.001$ was always used.

2.6.1 Comparison of Algorithms with and without AST

Figure 2.2 compares the proposed generalized sparse LMS (2.29), i.e., the AST-based approach, to some existing regularization-based algorithms of (2.10), i.e., the regular gradient descent without AST. Specifically, we use the p-norm-like penalty $\|\mathbf{h}\|_p^p$ with $p = 1$ and the log-sum penalty $\sum_{i=0}^{M-1} \log(|h_i| + \epsilon)$ with $\epsilon = 0.1$ as examples. These two choices of the sparsity-inducing function $\mathcal{I}(\mathbf{h})$ in (2.10) result in the ZA-LMS and RZA-LMS [24], respectively. We compare them with the corresponding AST-based algorithms obtained from (2.29), also adopting the two penalty functions for $G(\mathbf{h})$ that lead to (2.34) and (2.37) for computing $\mathbf{W}_n$, respectively. We set $\mu = 0.0025$ and $\lambda = 0.001$ in all cases and used the white Gaussian process input. Figure 2.2 (a) shows the results of identifying the sparse IR and Figure 2.2 (b) is the case of estimating the quasi-sparse IR. From the results we see that the AST strategy leads to algorithms (dotted lines) that demonstrate faster convergence than the existing approaches (solid lines).

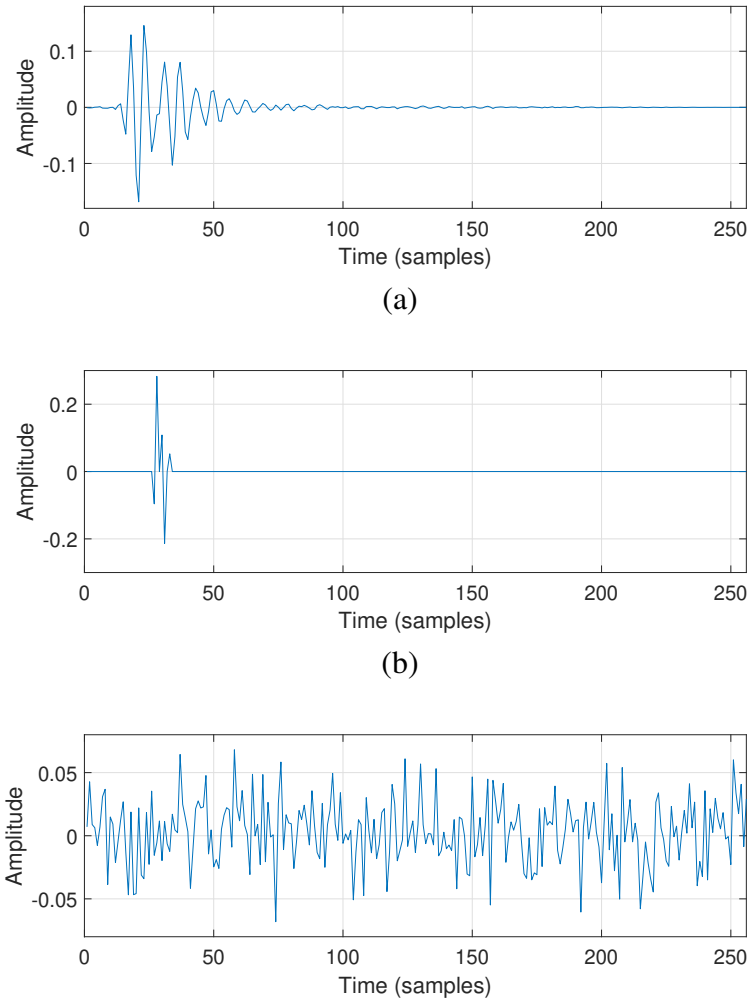
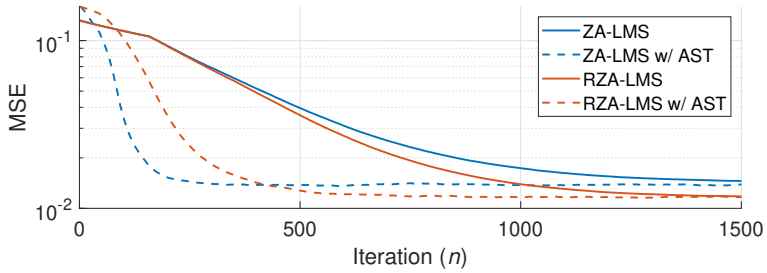


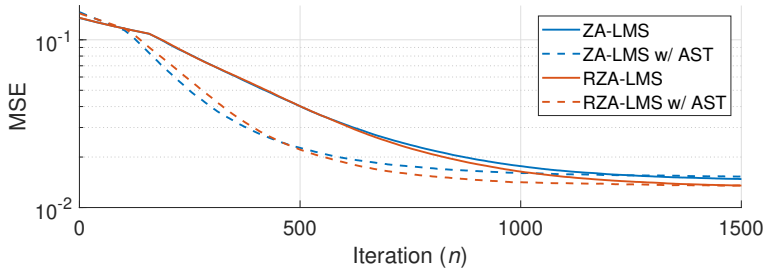
Figure 2.1: IRs of (a) quasi-sparse, (b) sparse, and (c) dispersive systems. The quasi-sparse IR is an acoustic feedback path of a hearing aid that was measured from a real-world scenario. The sparse and dispersive IRs were artificially generated.

2.6.2 Effect of Sparsity Control Parameter on SLMS and SNLMS

In this experiment we investigate the effect of the sparsity control parameter on the convergence of SLMS (2.30) and SNLMS (2.32). We use the p-norm-like diversity measure $\|\mathbf{h}\|_p^p$ within the reweighted ℓ_2 framework, i.e., using (2.34) for updating $\mathbf{W}_n$, for demonstration purposes. We study the cases of the sparsity control parameter $p = 1, 1.2, 1.5, 1.8, 2$. We also



(a)



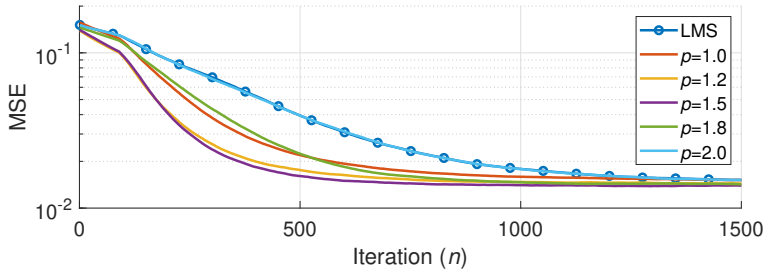
(b)

Figure 2.2: Comparison of algorithms with and without AST for identifying (a) sparse and (b) quasi-sparse IRs with white Gaussian process input. Solid lines are existing approaches as given by (2.10). Dotted lines are their corresponding AST-based algorithms given by (2.29). It can be seen that AST leads to improved performance.

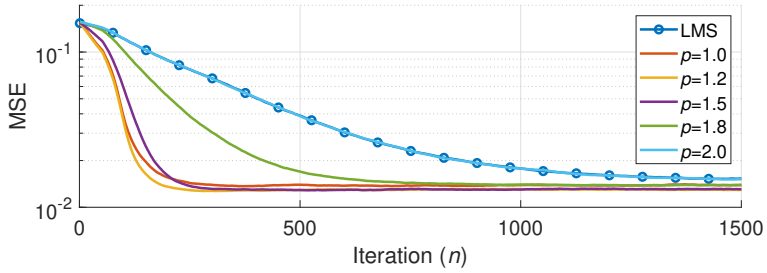
include the LMS (2.3) and NLMS (2.4) performance curves for reference. For LMS and SLMS we used $\mu = 0.0025$. For NLMS and SNLMS we used $\tilde{\mu} = 0.5$ and $\delta = 0.01$.

Figure 2.3 and Figure 2.4 show the resulting MSE curves for SLMS using the white Gaussian noise input and SNLMS using the AR process input, respectively. Recall that the proportionate factors of SLMS/SNLMS using (2.34) for $\mathbf{W}_n$ approximate that of the PNLMS when $p \to 1$, and regenerate the LMS/NLMS when $p = 2$, as has been discussed in Section 2.4.2. Therefore, the parameter p plays the role for fitting different sparsity levels and the selection of p can be crucial for obtaining optimal performance for IRs with different sparsity degrees. The results in both Figure 2.3 and Figure 2.4 suggest that for the quasi-sparse case, the fastest convergence is given by $p \in [1.2, 1.5]$, which seems reasonable in terms of finding a balance between PNLMS ($p \to 1$) and LMS/NLMS ($p = 2$). On the other hand, for the sparse system,

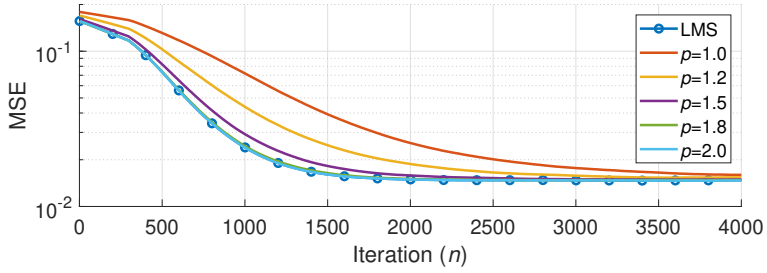
$p \in [1, 1.2]$ gives the best results, which is also reasonable since as the sparsity level increases, a more PNLMS-like algorithm can be more favorable. Finally, for the dispersive system we see that $p \in [1.8, 2]$ results in the fastest convergence and is comparable to, if not better than, the LMS and NLMS. This indicates that a more LMS/NLMS-like algorithm is preferable when the system IR is far from sparse. To conclude, the results show that the algorithms exploit the underlying system structure in the way we expect.



(a)

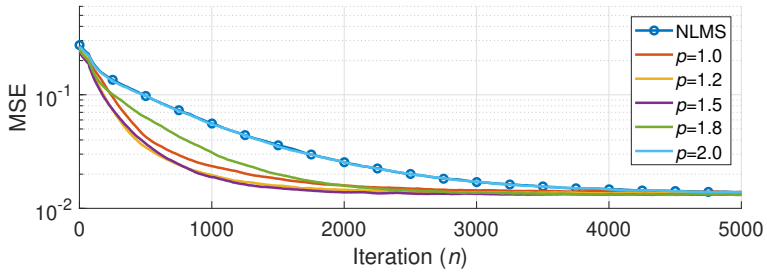


(b)

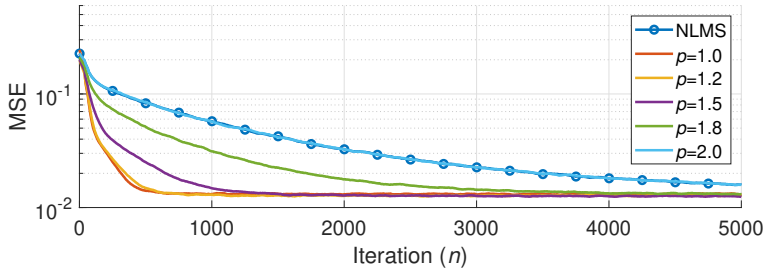


(c)

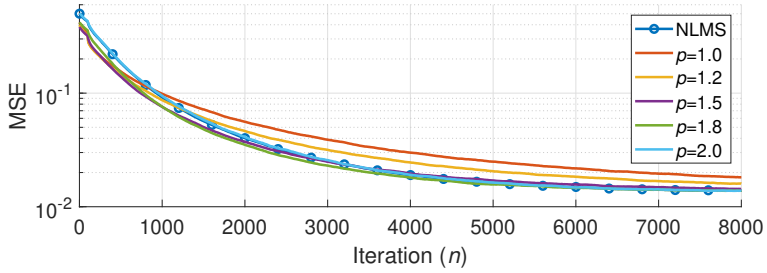
Figure 2.3: Effect of sparsity control parameter p on convergence of SLMS for (a) quasi-sparse, (b) sparse, and (c) dispersive IRs with white Gaussian process input. It can be seen that the optimal p value varies with the sparsity degree.



(a)



(b)



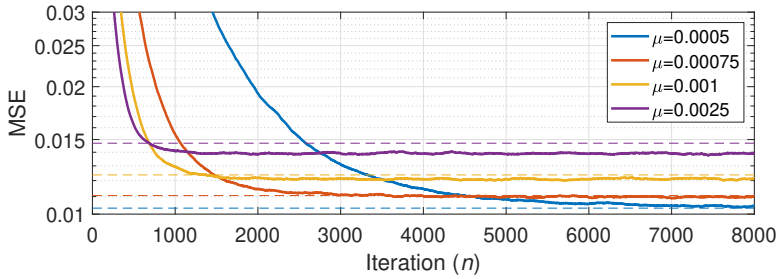
(c)

Figure 2.4: Effect of sparsity control parameter p on convergence of SNLMS for (a) quasi-sparse, (b) sparse, and (c) dispersive IRs with AR process input. In the colored input case here we have similar observations to the white input case of Figure 2.3.

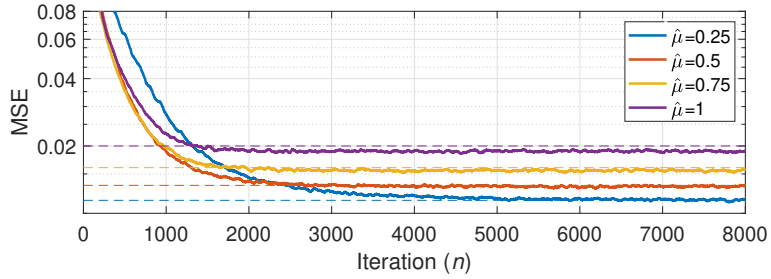
2.6.3 Effect of Step Size on SLMS and SNLMS

Figure 2.5 studies the effect of the step size on the convergence behavior of the SLMS and SNLMS. We again used (2.34) for updating $\mathbf{W}_n$. Figure 2.5 (a) shows the resulting MSE curves obtained by running the SLMS with $p = 1.2$ on the sparse IR with various μ values, using the white Gaussian noise input. Figure 2.5 (b) shows the resulting MSE curves obtained by running

the SNLMS with $p = 1.5$ on the quasi-sparse IR with various $\tilde{\mu}$ values, using the AR process input. The dotted lines indicate the theoretical steady-state MSE levels computed from (2.40) using (2.44) and (2.48) for SLMS and SNLMS, respectively. We can see that similar to the well-known trade-off in LMS and NLMS, a larger step size results in faster convergence while at the expense of steady-state performance. We also see that as the step size increases the theoretical prediction becomes less accurate; this is probably due to the approximation made based on the small step size assumption for arriving at (2.44) and (2.48). Nevertheless, the prediction agrees well with the stead-state MSE in most cases for a small step size. In addition, though several assumptions have been made to arrive at (2.40), (2.44) and (2.48), the results show that they predict reasonably well in the case of white input and even for correlated input.



(a)



(b)

Figure 2.5: Effect of step size μ or $\hat{\mu}$ on convergence of (a) SLMS for the sparse IR with white Gaussian process input and (b) SNLMS for the quasi-sparse IR with AR process input. Dotted lines indicate the theoretical steady-state MSE levels. It can be seen that the theoretical prediction agrees reasonably well with the experimental results especially for a small step size.

2.6.4 Comparison with Existing Algorithms

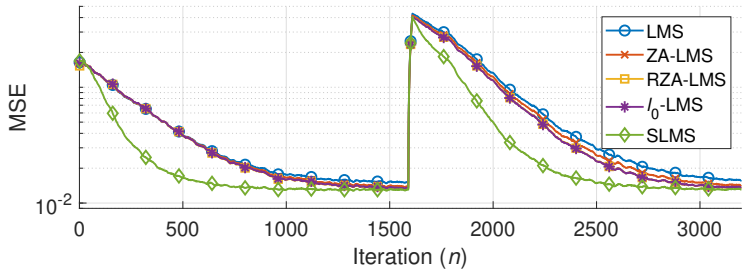
We compare the proposed SLMS and SNLMS using (2.34) for $\mathbf{W}_n$ with existing LMS-type and NLMS-type algorithms. To see how the algorithms behave in a changing environment, in each of the following experiments, a change in the underlying system was introduced by shifting the IR to the right by 16 samples in the middle of the adaptation process [82].

Figure 2.6 compares the LMS-type algorithms using the white Gaussian process input. Figure 2.6 (a) and Figure 2.6 (b) show the MSE curves obtained with the quasi-sparse and sparse IRs, respectively. For LMS we used $\mu = 0.0025$. For ZA-LMS, RZA-LMS, and ℓ_0-LMS we fixed $\mu = 0.0025$ and then experimentally optimized the remaining parameters to obtain the best performance. For SLMS we used $p = 1.5$ and $\mu = 0.002$ in the quasi-sparse case and $p = 1.2$ and $\mu = 0.0005$ in the sparse case. The results show that all the sparsity-aware algorithms outperform the LMS, with SLMS demonstrating the best result. Comparing Figure 2.6 (a) and Figure 2.6 (b), we also see that the benefit brought by existing sparsity-aware algorithms becomes limited when the system is less sparse, while the SLMS still provides significant improvement.

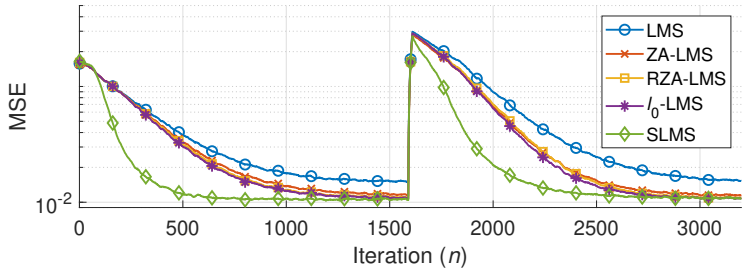
Figure 2.7 compares the NLMS-type algorithms using the AR process input. Figure 2.7 (a) and Figure 2.7 (b) show the MSE curves obtained with the quasi-sparse and sparse IRs, respectively. For all the algorithms we used $\tilde{\mu} = 0.5$. For NLMS we used $\delta = 0.01$. For PNLMS, IPNLMS, and IPNLMS-ℓ_0 we set $\delta = 0.01/M$ according to [82], and experimentally optimized the remaining parameters to obtain the best performance in each case. For SNLMS we used $p = 1.5$ in the quasi-sparse case and $p = 1.2$ in the sparse case. We used $\delta = 0.01$ for SNLMS, same as NLMS.[11] From the results we again observe the benefit of using sparsity-aware adaptation. In addition, the SNLMS demonstrates performance as good as, if not better than, the other proportionate algorithms.

Figure 2.8 considers a more practical scenario where we used a speech signal as the

[11] Due to the division by $\frac{1}{M}$ in (2.28) which is not present in (2.8) of existing PNLMS-type algorithms, the division by M is not needed for δ in SNLMS.



(a)

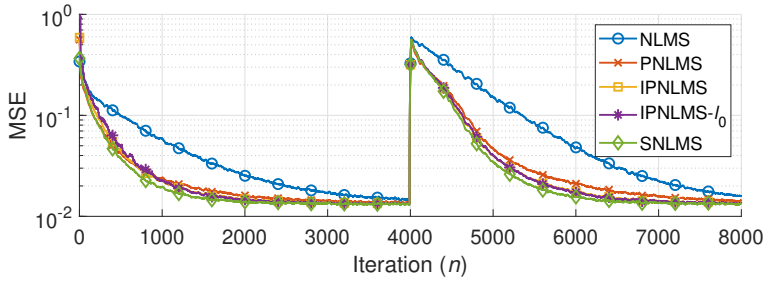


(b)

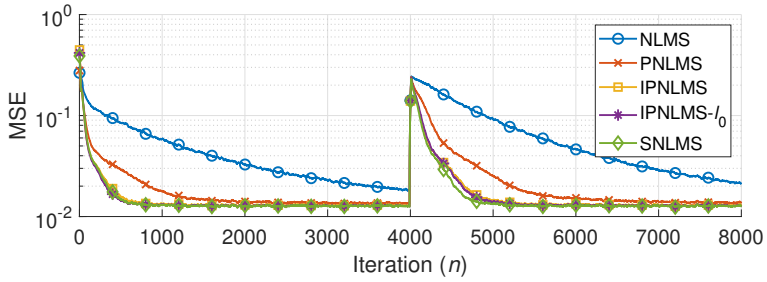
Figure 2.6: Comparison of LMS-type algorithms with white Gaussian process input on (a) quasi-sparse and (b) sparse IRs. One can see that the proposed SLMS outperforms all the other approaches in both cases.

input and the quasi-sparse IR, which represents an acoustic channel of practical interest, as the underlying system. The input signal-to-noise ration (SNR) was set to 20 dB using white Gaussian noise. For the SLMS and SNLMS we used $p = 1.5$ which is a suitable choice for quasi-sparse systems. For evaluation we compare the normalized misalignment $\|\mathbf{h}^o - \mathbf{h}_n\|_2^2 / \|\mathbf{h}^o\|_2^2$. In Figure 2.8 (a) we see that SLMS performs much better than the LMS, while the ℓ_0-LMS fails to provide any improvement. This may be due to the fact that existing regularization-based algorithms tend to enforce sparsity in a more aggressive manner as they work with $\lambda > 0$, and this may not be beneficial, if not harmful, when the underlying system is not truly sparse. In Figure 2.8 (b) we see that SNLMS demonstrates superior convergence behavior than the NLMS, and is also better than the IPNLMS and IPNLMS-ℓ_0.

Figure 2.9 shows the results for a noisier environment, i.e., 0 dB input SNR, for the same experimental setting of Figure 2.8 (only the step size parameters were further tuned due to the



(a)



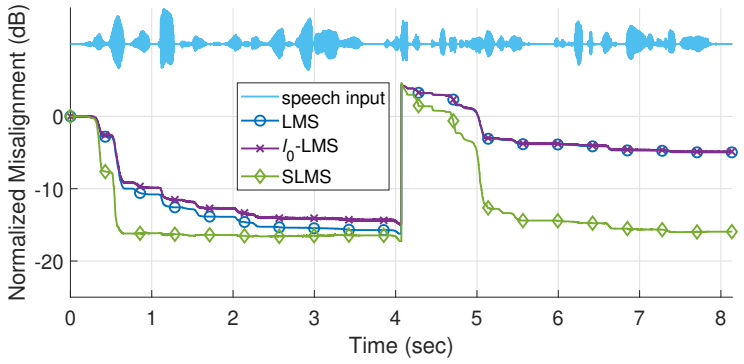
(b)

Figure 2.7: Comparison of NLMS-type algorithms with AR process input on (a) quasi-sparse and (b) sparse IRs. One can see that the proposed SNLMS performs better than all the other approaches in both cases.

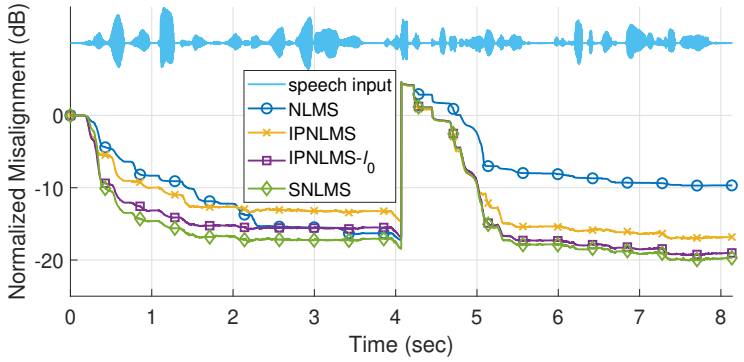
stronger noise). We see that in Figure 2.9 (a) the SLMS significantly outperforms the LMS, while the ℓ_0-LMS performs worse. The SNLMS in Figure 2.9 (b), on the other hands, still performs better than the NLMS, and is comparable to other proportionate algorithms. This indicates that our observation on the SLMS and SNLMS superiority may be robust to the noise condition.

2.7 Conclusion

In this chapter, we developed a mathematical framework for rigorously deriving adaptive filters that exploit the sparse structure of the underlying system response. We started with the regularized objective framework of SSR and developed algorithms that are of the proportionate type. As a result, the adaptive algorithms are quite general and can accommodate a range of regularization functions. The framework utilizes the AST methodology within the iterative



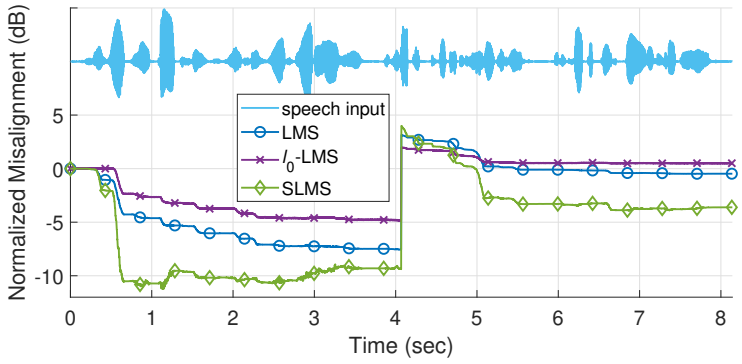
(a)



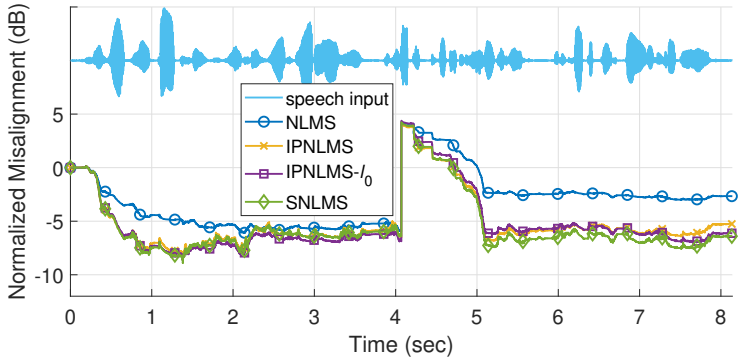
(b)

Figure 2.8: Comparison of (a) LMS-type and (b) NLMS-type algorithms for identifying the quasi-sparse acoustic channel response with speech input at 20 dB SNR. In can be seen that the SLMS and SNLMS perform the best in both cases.

reweighted ℓ_2 and ℓ_1 schemes, which is shown to be crucial for obtaining improved adaptive filtering performance over existing algorithms when gradient descent is concerned. We further introduced the SLMS and SNLMS by adopting a zero regularization coefficient, which take advantage of, though do not strictly enforce, the sparsity of the underlying system if it already exists. Note that the proposed framework is not limited to the algorithms that we have presented so far. Any other penalty function that satisfies the conditions imposed on the diversity measure can potentially be a good candidate for obtaining effective adaptive algorithms by utilizing the framework.



(a)



(b)

Figure 2.9: Comparison of (a) LMS-type and (b) NLMS-type algorithms for identifying the quasi-sparse acoustic channel response with speech input at 0 dB SNR. In the noisier setting here the SLMS and SNLMS perform comparably well, if not better than, the competing algorithms.

Acknowledgment

Chapter 2 is, in part, a reprint of the material as it appears in the two papers: C.-H. Lee, B. D. Rao, and H. Garudadri, "Proportionate adaptive filtering algorithms derived using an iterative reweighting framework," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2020 and C.-H. Lee, B. D. Rao, and H. Garudadri, "Proportionate adaptive filters based on minimizing diversity measures for promoting sparsity," in *53rd Asilomar Conference on Signals, Systems, and Computers (ACSSC)*, 2019. The dissertation author was the primary investigator and author of these papers. The work, in part, was supported by National Institutes

of Health/National Institute on Deafness and Other Communication Disorders under Grants R01DC015436 and R33DC015046 and National Science Foundation/Information and Intelligent Systems under Award 1838830.

2.8 Appendix

2.8.1 Proof of Theorem 2.1

The proof follows the idea in [83]. We wish to show that the regularized objective function $J^G(\mathbf{h})$ in (2.9) is decreased in each iteration when optimized via (2.20) and (2.21). Before proceeding, we need the following lemmas:

Lemma 2.1. *For the general diversity measure $G(\mathbf{h}) = \sum_{i=0}^{M-1} g(h_i)$ that satisfies Properties 1-4 in Section 2.2.2, with $g(z)$ being strictly concave in z^2 for Property 4, we have:*

$$G(\mathbf{h}_{n+1}) - G(\mathbf{h}_n) < \|\mathbf{W}_n^{-1}\mathbf{h}_{n+1}\|_2^2 - \|\mathbf{W}_n^{-1}\mathbf{h}_n\|_2^2, \tag{2.51}$$

where $\mathbf{W}_n = diag\{w_{i,n}\}$ with $w_{i,n}$ given by (2.13).

Proof: Since $g(z)$ is strictly concave in z^2, it satisfies $g(z) = f(z^2)$ where $f(z)$ is concave for $z \in \mathbb{R}_+$. Due to the concavity, we have the following inequality:

$$f(z_2) - f(z_1) < f'(z_1)(z_2 - z_1) \tag{2.52}$$

hold for some $z_1, z_2 \in \mathbb{R}_+$. Note that we use $f'(z_1)$ to denote the first order derivative of $f(z)$ w.r.t. z evaluated at $z = z_1$.

Substituting $z_1 = h_{i,n}^2$ and $z_2 = h_{i,n+1}^2$ into (2.52) gives:

$$f(h_{i,n+1}^2) - f(h_{i,n}^2) < f'(h_{i,n}^2)(h_{i,n+1}^2 - h_{i,n}^2). \tag{2.53}$$

Noting that $f(h_{i,n+1}^2) = g(h_{i,n+1})$ and $f(h_{i,n}^2) = g(h_{i,n})$, we have:

$$g(h_{i,n+1}) - g(h_{i,n}) < f'(h_{i,n}^2)(h_{i,n+1}^2 - h_{i,n}^2). \tag{2.54}$$

From (2.13) we have $f'(h_{i,n}^2) = w_{i,n}^{-2}$. Therefore,

$$g(h_{i,n+1}) - g(h_{i,n}) < w_{i,n}^{-2}(h_{i,n+1}^2 - h_{i,n}^2). \tag{2.55}$$

Summing over $i = 0, 1, \ldots, M-1$ on both sides of (2.55) justifies (2.51) of Lemma 2.1.

Lemma 2.2. *For the general diversity measure $G(\mathbf{h}) = \sum_{i=0}^{M-1} g(h_i)$ that satisfies Properties 1-4 in Section 2.2.2, with $g(z)$ being strictly concave in $|z|$ for Property 4, we have:*

$$G(\mathbf{h}_{n+1}) - G(\mathbf{h}_n) < \|\mathbf{W}_n^{-1}\mathbf{h}_{n+1}\|_1 - \|\mathbf{W}_n^{-1}\mathbf{h}_n\|_1, \tag{2.56}$$

where $\mathbf{W}_n = diag\{w_{i,n}\}$ with $w_{i,n}$ given by (2.14).

Proof: Since $g(z)$ is strictly concave in $|z|$, it satisfies $g(z) = f(|z|)$ where $f(z)$ is concave for $z \in \mathbb{R}_+$. Again, the inequality (2.52) holds due to the concavity of $f(z)$.

Substituting $z_1 = |h_{i,n}|$ and $z_2 = |h_{i,n+1}|$ into (2.52) gives:

$$f(|h_{i,n+1}|) - f(|h_{i,n}|) < f'(|h_{i,n}|)(|h_{i,n+1}| - |h_{i,n}|). \tag{2.57}$$

Noting that $f(|h_{i,n+1}|) = g(h_{i,n+1})$ and $f(|h_{i,n}|) = g(h_{i,n})$, we have:

$$g(h_{i,n+1}) - g(h_{i,n}) < f'(|h_{i,n}|)(|h_{i,n+1}| - |h_{i,n}|). \tag{2.58}$$

From (2.14) we have $f'(|h_{i,n}|) = w_{i,n}^{-1}$. Therefore,

$$g(h_{i,n+1}) - g(h_{i,n}) < w_{i,n}^{-1}(|h_{i,n+1}| - |h_{i,n}|). \tag{2.59}$$

Summing over $i = 0, 1, \ldots, M-1$ on both sides of (2.59), we have (2.56) of Lemma 2.2 justified.

Now we are ready to show that $J^G(\mathbf{h})$ decreases in each iteration by using the update recursions (2.20) and (2.21).

First, for the reweighted ℓ_2 framework with $J_n^{\ell_2}(\mathbf{q})$ in (2.16), we have:

$$\begin{aligned} J^G(\mathbf{h}_{n+1}) - J^G(\mathbf{h}_n) &= \left[J(\mathbf{h}_{n+1}) + \lambda G(\mathbf{h}_{n+1})\right] - \left[J(\mathbf{h}_n) + \lambda G(\mathbf{h}_n)\right] \\ &< \left[J(\mathbf{h}_{n+1}) + \lambda \|\mathbf{W}_n^{-1}\mathbf{h}_{n+1}\|_2^2\right] - \left[J(\mathbf{h}_n) + \lambda \|\mathbf{W}_n^{-1}\mathbf{h}_n\|_2^2\right] \\ &= \left[J(\mathbf{W}_n\mathbf{q}_{n+1|n}) + \lambda \|\mathbf{q}_{n+1|n}\|_2^2\right] - \left[J(\mathbf{W}_n\mathbf{q}_{n|n}) + \lambda \|\mathbf{q}_{n|n}\|_2^2\right] \\ &= J_n^{\ell_2}(\mathbf{q}_{n+1|n}) - J_n^{\ell_2}(\mathbf{q}_{n|n}). \end{aligned} \tag{2.60}$$

The inequality follows from Lemma 2.1. The AST relationships (2.18) and (2.19) are also utilized. As we perform optimization of (2.16) with gradient descent, we can have $J_n^{\ell_2}(\mathbf{q})$ decrease in each iteration n, i.e., $J_n^{\ell_2}(\mathbf{q}_{n+1|n}) - J_n^{\ell_2}(\mathbf{q}_{n|n}) < 0$, using some μ_n. Therefore, the choice of $\{\mu_n\}_{n=0}^{\infty}$ ensures the decrease in $J^G(\mathbf{h})$ according to (2.60), and the update recursion (2.20) monotonically converges to a local minimum (or saddle point) of (2.9) under a WSS environment.

On the other hand, for the reweighted ℓ_1 framework with $J_n^{\ell_1}(\mathbf{q})$ in (2.17), we have:

$$\begin{aligned} J^G(\mathbf{h}_{n+1}) - J^G(\mathbf{h}_n) &= \left[J(\mathbf{h}_{n+1}) + \lambda G(\mathbf{h}_{n+1})\right] - \left[J(\mathbf{h}_n) + \lambda G(\mathbf{h}_n)\right] \\ &< \left[J(\mathbf{h}_{n+1}) + \lambda \|\mathbf{W}_n^{-1}\mathbf{h}_{n+1}\|_1\right] - \left[J(\mathbf{h}_n) + \lambda \|\mathbf{W}_n^{-1}\mathbf{h}_n\|_1\right] \\ &= \left[J(\mathbf{W}_n\mathbf{q}_{n+1|n}) + \lambda \|\mathbf{q}_{n+1|n}\|_1\right] - \left[J(\mathbf{W}_n\mathbf{q}_{n|n}) + \lambda \|\mathbf{q}_{n|n}\|_1\right] \\ &= J_n^{\ell_1}(\mathbf{q}_{n+1|n}) - J_n^{\ell_1}(\mathbf{q}_{n|n}). \end{aligned} \tag{2.61}$$

The inequality follows from Lemma 2.2. The AST relationships (2.18) and (2.19) are also utilized.

Similar to the above argument of the reweighted ℓ_2 case, there exists a choice of $\{\mu_n\}_{n=0}^{\infty}$ that ensures the decrease in $J^G(\mathbf{h})$ according to (2.61), and the update recursion (2.21) monotonically converges to a local minimum (or saddle point) of (2.9) under a WSS environment.

2.8.2 Proof of Theorem 2.2

The proof follows the discussion in [21, 27, 66]. Substituting $e_n = d_n - \mathbf{u}_n^T \mathbf{h}_n$ into (2.38) we have:

$$\mathbf{h}_{n+1} = \mathbf{h}_n - \mu \mathbf{S}\mathbf{u}_n \mathbf{u}_n^T \mathbf{h}_n + \mu \mathbf{S}\mathbf{u}_n d_n. \tag{2.62}$$

Using the fact that $d_n = \mathbf{u}_n^T \mathbf{h}^o + v_n$, we have:

$$\mathbf{h}_{n+1} = \mathbf{h}_n + \mu \mathbf{S}\mathbf{u}_n \mathbf{u}_n^T \left(\mathbf{h}^o - \mathbf{h}_n\right) + \mu \mathbf{S}\mathbf{u}_n v_n. \tag{2.63}$$

Define the misalignment vector $\boldsymbol{\varepsilon}_n$ as:

$$\boldsymbol{\varepsilon}_n = \mathbf{h}^o - \mathbf{h}_n. \tag{2.64}$$

Then from (2.63) we have:

$$\boldsymbol{\varepsilon}_{n+1} = \left(\mathbf{I} - \mu \mathbf{S}\mathbf{u}_n \mathbf{u}_n^T\right) \boldsymbol{\varepsilon}_n - \mu \mathbf{S}\mathbf{u}_n v_n. \tag{2.65}$$

Next, based on (2.65) we have:

$$\boldsymbol{\varepsilon}_{n+1}\boldsymbol{\varepsilon}_{n+1}^T = \left(\mathbf{I} - \mu \mathbf{S}\mathbf{u}_n \mathbf{u}_n^T\right) \boldsymbol{\varepsilon}_n \boldsymbol{\varepsilon}_n^T \left(\mathbf{I} - \mu \mathbf{u}_n \mathbf{u}_n^T \mathbf{S}\right) + \mu^2 v_n^2 \mathbf{S}\mathbf{u}_n \mathbf{u}_n^T \mathbf{S} + \boldsymbol{\Xi}, \tag{2.66}$$

where $\boldsymbol{\Xi}$ represents the remaining cross terms whose expectations are zero.

Let $\boldsymbol{\Omega}_n = \mathrm{E}[\boldsymbol{\varepsilon}_n \boldsymbol{\varepsilon}_n^T]$. Taking expectation on both sides of (2.66) we have:

$$\boldsymbol{\Omega}_{n+1} = \underbrace{\mathrm{E}\left[\left(\mathbf{I} - \mu \mathbf{S}\mathbf{u}_n\mathbf{u}_n^T\right)\boldsymbol{\varepsilon}_n\boldsymbol{\varepsilon}_n^T\left(\mathbf{I} - \mu\mathbf{u}_n\mathbf{u}_n^T\mathbf{S}\right)\right]}_{\triangleq\boldsymbol{\Theta}} + \mu^2\sigma_v^2\mathbf{S}\mathbf{R}\mathbf{S}. \tag{2.67}$$

Note that:

$$\boldsymbol{\Theta} = \boldsymbol{\Omega}_n - \mu\mathbf{S}\mathbf{R}\boldsymbol{\Omega}_n - \mu\boldsymbol{\Omega}_n\mathbf{R}\mathbf{S} + \mu^2\mathbf{S}\mathrm{E}\left[\mathbf{u}_n\mathbf{u}_n^T\boldsymbol{\varepsilon}_n\boldsymbol{\varepsilon}_n^T\mathbf{u}_n\mathbf{u}_n^T\right]\mathbf{S}. \tag{2.68}$$

With Assumptions 1 and 2 it can be shown that [21]:

$$\mathrm{E}\left[\mathbf{u}_n\mathbf{u}_n^T\boldsymbol{\varepsilon}_n\boldsymbol{\varepsilon}_n^T\mathbf{u}_n\mathbf{u}_n^T\right] = 2\mathbf{R}\boldsymbol{\Omega}_n\mathbf{R} + \mathbf{R}\mathrm{tr}(\mathbf{R}\boldsymbol{\Omega}_n). \tag{2.69}$$

Thus,

$$\boldsymbol{\Theta} = \boldsymbol{\Omega}_n - \mu\mathbf{S}\mathbf{R}\boldsymbol{\Omega}_n - \mu\boldsymbol{\Omega}_n\mathbf{R}\mathbf{S} + 2\mu^2\mathbf{S}\mathbf{R}\boldsymbol{\Omega}_n\mathbf{R}\mathbf{S} + \mu^2\mathbf{S}\mathbf{R}\mathrm{tr}(\mathbf{R}\boldsymbol{\Omega}_n)\mathbf{S}. \tag{2.70}$$

Then, with $\mathbf{R} = \sigma_u^2\mathbf{I}$, in steady-state, i.e., $n \to \infty$,

$$\boldsymbol{\Omega}_\infty = \boldsymbol{\Omega}_\infty - \mu\sigma_u^2\mathbf{S}\boldsymbol{\Omega}_\infty - \mu\sigma_u^2\boldsymbol{\Omega}_\infty\mathbf{S} + 2\mu^2\sigma_u^4\mathbf{S}\boldsymbol{\Omega}_\infty\mathbf{S} + \mu^2\sigma_u^4\mathbf{S}\mathrm{tr}\left(\boldsymbol{\Omega}_\infty\right)\mathbf{S} + \mu^2\sigma_u^2\sigma_v^2\mathbf{S}^2. \tag{2.71}$$

This implies:

$$\boldsymbol{\omega}_\infty = \boldsymbol{\omega}_\infty - 2\mu\sigma_u^2\mathbf{S}\boldsymbol{\omega}_\infty + 2\mu^2\sigma_u^4\mathbf{S}^2\boldsymbol{\omega}_\infty + \mu^2\sigma_u^4\mathbf{s}^2\mathbf{1}^T\boldsymbol{\omega}_\infty + \mu^2\sigma_u^2\sigma_v^2\mathbf{s}^2, \tag{2.72}$$

where $\boldsymbol{\omega}_\infty$ and $\mathbf{s}$ are the vectors consisting of the diagonal elements of $\boldsymbol{\Omega}_\infty$ and $\mathbf{S}$, respectively, and $\mathbf{s}^2$ denotes the element-wise squares of the vector $\mathbf{s}$.

The steady-state excess MSE is then:

$$\begin{aligned} J_{\mathrm{ex}} &\triangleq \lim_{n\to\infty}\mathrm{E}\left[\left(\mathbf{u}_n^T\left(\mathbf{h}^o - \mathbf{h}_n\right)\right)^2\right] = \lim_{n\to\infty}\mathrm{E}\left[\left(\mathbf{u}_n^T\boldsymbol{\varepsilon}_n\right)^2\right] = \lim_{n\to\infty}\mathrm{tr}\left(\boldsymbol{\Omega}_n\mathrm{E}\left[\mathbf{u}_n\mathbf{u}_n^T\right]\right) \\ &= \sigma_u^2\mathrm{tr}\left(\boldsymbol{\Omega}_\infty\right) = \sigma_u^2\mathbf{1}^T\boldsymbol{\omega}_\infty. \end{aligned} \tag{2.73}$$

Using (2.73) for (2.72) and rearrange the equation, we have:

$$\omega_{i,\infty} = \frac{\mu^2\sigma_u^2 s_i^2 J_{\text{ex}} + \mu^2\sigma_u^2\sigma_v^2 s_i^2}{2\mu\sigma_u^2 s_i - 2\mu^2\sigma_u^4 s_i^2}. \tag{2.74}$$

Then it leads to:

$$J_{\text{ex}} = \sigma_u^2 \sum_{i=0}^{M-1} \omega_{i,\infty} = \sum_{i=0}^{M-1} \frac{\mu^2\sigma_u^2 s_i^2 J_{\text{ex}} + \mu^2\sigma_u^2\sigma_v^2 s_i^2}{2\mu s_i - 2\mu^2\sigma_u^2 s_i^2}, \tag{2.75}$$

which yields:

$$J_{\text{ex}} = \frac{\mu \sum_{i=0}^{M-1} \frac{\sigma_u^2 s_i}{2-2\mu\sigma_u^2 s_i}}{1-\mu \sum_{i=0}^{M-1} \frac{\sigma_u^2 s_i}{2-2\mu\sigma_u^2 s_i}} \sigma_v^2. \tag{2.76}$$

This justifies Theorem 2.2.

2.8.3 Proof of Theorem 2.3

Note that Assumption 1 ensures that $\mathbf{h}_n$, $\mathbf{u}_n$, and v_n are mutually independent. Thus, taking expectation of both sides of (2.65) gives:

$$\mathrm{E}\left[\boldsymbol{\varepsilon}_{n+1}\right] = (\mathbf{I} - \mu\mathbf{S}\mathbf{R})\mathrm{E}\left[\boldsymbol{\varepsilon}_n\right]. \tag{2.77}$$

Therefore, the following condition is sufficient for convergence in the mean sense [21]:

$$|\lambda_{\max}\{\mathbf{I} - \mu\mathbf{S}\mathbf{R}\}| < 1. \tag{2.78}$$

With $\mathbf{R} = \sigma_u^2\mathbf{I}$, Theorem 2.3-i) is justified.

From (2.76) we see, by requiring:

$$1 - \mu \sum_{i=0}^{M-1} \frac{\sigma_u^2 s_i}{2 - 2\mu\sigma_u^2 s_i} > 0, \tag{2.79}$$

we obtain the stability bound for μ as:

$$0 < \mu < \left(\sum_{i=0}^{M-1} \frac{\sigma_u^2 s_i}{2 - 2\mu\sigma_u^2 s_i} \right)^{-1}, \tag{2.80}$$

which justifies Theorem 2.3-ii).

Chapter 3

A Sparsity-Aware CG-Type Adaptive Filtering Algorithm

In this chapter, we propose a novel adaptive filter of the conjugate gradient (CG) type for online estimation of system responses that admit sparsity. Specifically, the Sparsity-promoting CG (SCG) algorithm is developed based on iterative reweighting methods popular in the sparse signal recovery area. We propose an affine scaling transformation strategy within the reweighting framework, leading to an algorithm that allows the usage of a zero sparsity regularization coefficient. As a result, it enables SCG to leverage the sparsity of the system response if it already exists, while not compromising the optimization process. Simulation results show that SCG demonstrates improved convergence and steady-state properties over existing methods.

3.1 Introduction

In many applications of adaptive filters [18, 19, 20, 21], the underlying system impulse responses (IRs) to be identified are often sparse or compressible (quasi-sparse), e.g., in acoustic echo and feedback cancellation [27, 54, 55, 23, 56, 57, 58]. Thus, designing adaptive filters that

can exploit the sparse structure of the system IR has been a research topic of great interest. Not so surprisingly, sparse signal recovery (SSR) techniques [61, 14, 62, 15, 16] that have proven successful in learning compact solutions to linear problems have fueled the trend of sparsity-aware adaptive filtering research.

Based on SSR, a great amount of work has focused on developing sparse variants of the least mean square (LMS) [24, 25, 76, 73, 43, 44, 72, 74, 69] and recursive least squares (RLS) [84, 85, 86, 87, 88, 89, 90, 91, 92] adaptive filters. The complexity of LMS-type algorithms scales only linearly, however, they suffer from slow convergence in the presence of strong signal correlation. In contrast, RLS-type algorithms in general have a faster convergence speed but with increased computational complexity [19, 20].

A more recent class of adaptive filters is the conjugate gradient (CG) method [45, 46, 47, 48, 49, 50]. The CG adaptive filter can be viewed as an alternative algorithm which inherits the virtues of LMS and RLS, while mitigating some of their drawbacks – it has a faster convergence rate and is less sensitive to signal correlation than LMS, while being numerically more stable than the conventional RLS [49]. Surprisingly, given its effectiveness, CG has yet to receive considerable attention in the adaptive filtering literature.

In this chapter, we propose a novel sparsity-aware CG adaptive filter that we call Sparsity-promoting CG (SCG). Starting by formulating an optimization problem with sparsity regularization, SCG is developed based on iterative reweighting methods [13] popular in SSR. Moreover, an affine scaling transformation (AST) [42, 60] strategy is utilized to allow the usage of $\lambda = 0$, where λ is the sparsity regularization coefficient. This leads to the algorithm developing a better path toward the optimum without biasing or compromising the optimization objective, and taking advantage of sparsity should it be present in the system IR. To our knowledge, it is the first study on sparsity-aware CG adaptive filtering.

Organization of the Chapter: The rest of the chapter is organized as follows. Section 3.2 provides background on CG-based adaptive filtering. Section 3.3 presents the CG adaptive filter

framework for incorporating sparsity based on iterative reweighting SSR techniques and an AST methodology, leading to the proposed SCG algorithm. Section 3.4 presents simulation results. Section 3.5 concludes the chapter.

3.2 CG-Based Adaptive Filtering

Let $\mathbf{h}_n = [h_{0,n}, h_{1,n}, ..., h_{M-1,n}]^T$ denote the adaptive filter of length M at discrete time instant n. Assume the IR of the underlying system is $\mathbf{h}^o = [h_0^o, h_1^o, ..., h_{M-1}^o]^T$, and the model for the observed or desired signal is $d_n = \mathbf{u}_n^T \mathbf{h}^o + v_n$, where $\mathbf{u}_n = [u_n, u_{n-1}, ..., u_{n-M+1}]^T$ is the vector containing the M most recent samples of the input signal u_n and v_n is an additive noise signal. The output of the adaptive filter $\mathbf{u}_n^T \mathbf{h}_n$ is subtracted from d_n to obtain the error signal $e_n = d_n - \mathbf{u}_n^T \mathbf{h}_n$. The goal in general is to sequentially update the coefficients of $\mathbf{h}_n$ upon the arrival of a new data pair $(\mathbf{u}_n, d_n)$, such that eventually $\mathbf{h}_n = \mathbf{h}^o$; i.e., to identify the unknown system.

To develop algorithms, we consider minimizing the objective function of the *weighted squared error* at time n:

$$\min_{\mathbf{h}} \quad J_n(\mathbf{h}) \triangleq \sum_{\tau=0}^{n} \gamma^{n-\tau} e_\tau^2 = \sum_{\tau=0}^{n} \gamma^{n-\tau} \left(d_\tau - \mathbf{u}_\tau^T \mathbf{h} \right)^2, \tag{3.1}$$

where $0 \ll \gamma \leq 1$ is called the *forgetting factor* [19]. Since $J_n(\mathbf{h})$ is convex and quadratic in $\mathbf{h}$, minimizing it corresponds to solving the linear equation: $\mathbf{R}_n \mathbf{h} = \mathbf{b}_n$, where

$$\mathbf{R}_n \triangleq \sum_{\tau=0}^{n} \gamma^{n-\tau} \mathbf{u}_\tau \mathbf{u}_\tau^T \qquad \text{and} \qquad \mathbf{b}_n \triangleq \sum_{\tau=0}^{n} \gamma^{n-\tau} \mathbf{u}_\tau d_\tau \tag{3.2}$$

are the correlation matrix estimate of $\mathbf{u}_n$ and the cross-correlation vector estimate between $\mathbf{u}_n$ and d_n, respectively. Note that both entities can be updated recursively as:

$$\mathbf{R}_n = \gamma \mathbf{R}_{n-1} + \mathbf{u}_n \mathbf{u}_n^T \qquad \text{and} \qquad \mathbf{b}_n = \gamma \mathbf{b}_{n-1} + \mathbf{u}_n d_n. \tag{3.3}$$

In adaptive filtering, we seek an algorithm of the form:

$$\mathbf{h}_{n+1} = \mathbf{h}_n + \mathbf{p}_n \alpha_n, \tag{3.4}$$

where $\mathbf{p}_n$ is the *search direction* and α_n is the *step size*. The standard CG in the optimization literature [60, 79] uses orthogonality, or *conjugacy*, of the search directions to simplify the steps necessary for convergence. In adaptive filtering, however, the sample-by-sample update of $\mathbf{R}_n$ and $\mathbf{b}_n$ causes a loss of the search direction conjugacy and in turn results in loss of convergence. Thus, some modifications are necessary to relax the standard conjugacy constraint, e.g., see [46, 47, 48, 49]. The authors of [49] have considered several methods under the umbrella of *Markov conjugacy*:

Definition 3.1. *A set of search directions* $\{\mathbf{p}_n\}$ *are said to be Markov conjugate w.r.t. symmetric matrices* $\{\mathbf{R}_n\}$ *if, at any iteration n, we have* $\mathbf{p}_n^T \mathbf{R}_n \mathbf{p}_{n-1} = 0$.

To develop CG algorithms for (3.1), first recognize that solving $\mathbf{R}_n \mathbf{h} = \mathbf{b}_n$ indirectly minimizes the quadratic function:

$$F_n(\mathbf{h}) \triangleq \frac{1}{2}\mathbf{h}^T \mathbf{R}_n \mathbf{h} - \mathbf{h}^T \mathbf{b}_n. \tag{3.5}$$

Utilizing the Markov conjugacy w.r.t. $\mathbf{R}_n$, the search direction is recursively updated according to:

$$\mathbf{p}_n = -\mathbf{g}_n + \mathbf{p}_{n-1}\beta_n, \tag{3.6}$$

where $\mathbf{g}_n$ denotes the gradient vector:

$$\mathbf{g}_n = \nabla_{\mathbf{h}} F_n(\mathbf{h}_n) = \mathbf{R}_n \mathbf{h}_n - \mathbf{b}_n, \tag{3.7}$$

and β_n is a scaling factor to sustain Markov conjugacy:

$$\beta_n = \frac{\mathbf{p}_{n-1}^T \mathbf{R}_n \mathbf{g}_n}{\mathbf{p}_{n-1}^T \mathbf{R}_n \mathbf{p}_{n-1}}, \tag{3.8}$$

obtained by premultiplying both sides of (3.6) with $\mathbf{p}_{n-1}^T \mathbf{R}_n$ and noticing that the right-hand side is zero by Definition 3.1.

Finally, the step size α_n is obtained via exact line search:

$$\alpha_n = \arg\min_{\alpha} F_n(\mathbf{h}_n + \mathbf{p}_n \alpha) = -\frac{\mathbf{p}_n^T \mathbf{g}_n}{\mathbf{p}_n^T \mathbf{R}_n \mathbf{p}_n}. \tag{3.9}$$

The above are summarized in Algorithm 2, which is essentially the *memory*-normalized LMS (m-NLMS),[1] proposed in [49].

Algorithm 2: The m-NLMS adaptive filtering algorithm proposed by Variddhisaï and Mandic in [49]

Input: γ, δ, $\mathbf{u}_n$, d_n
Output: $\mathbf{h}_n$
Initialize: $\mathbf{h}_0 = \mathbf{0}$, $\mathbf{R}_{-1} = \mathbf{0}$, $\mathbf{b}_{-1} = \mathbf{0}$, $\mathbf{p}_{-1} = \mathbf{0}$;
for $n = 0, 1, 2, \ldots$ **do**
  $\mathbf{R}_n = \gamma \mathbf{R}_{n-1} + \mathbf{u}_n \mathbf{u}_n^T$; ▷ corr. matrix update
  $\mathbf{b}_n = \gamma \mathbf{b}_{n-1} + \mathbf{u}_n d_n$; ▷ cross-corr. vector update
  $\mathbf{g}_n = \mathbf{R}_n \mathbf{h}_n - \mathbf{b}_n$; ▷ compute gradient vector
  $\beta_n = \frac{\mathbf{p}_{n-1}^T \mathbf{R}_n \mathbf{g}_n}{\mathbf{p}_{n-1}^T \mathbf{R}_n \mathbf{p}_{n-1} + \delta}$; ▷ for Markov conjugacy
  $\mathbf{p}_n = -\mathbf{g}_n + \mathbf{p}_{n-1} \beta_n$; ▷ search direction update
  $\alpha_n = -\frac{\mathbf{p}_n^T \mathbf{g}_n}{\mathbf{p}_n^T \mathbf{R}_n \mathbf{p}_n + \delta}$; ▷ compute step size
  $\mathbf{h}_{n+1} = \mathbf{h}_n + \mathbf{p}_n \alpha_n$; ▷ adaptive filter update
end for

[1] In [49], a recursive update for the gradient vector is further utilized to achieve more computational savings. For illustration purposes, we only outline the major steps here. Note that we have also introduced a small regularization constant $\delta > 0$ for preventing division by zero.

3.3 Proposed Sparsity-Aware CG Adaptive Filter Framework

To incorporate sparsity, we add a sparsity regularization term to the objective function in (3.1):

$$\min_{\mathbf{h}} \quad J_n(\mathbf{h}) + \lambda G(\mathbf{h}), \tag{3.10}$$

where $G(\cdot)$ represents the *general diversity measure* in SSR that induces sparsity in its argument and λ is the regularization coefficient. Commonly used is a *separable* form: $G(\mathbf{h}) = \sum_{i=0}^{M-1} g(h_i)$, where $g(\cdot)$ has the following properties [13]:

Property 1: $g(z)$ is symmetric, i.e., $g(z) = g(-z) = g(|z|)$;

Property 2: $g(|z|)$ is monotonically increasing with $|z|$;

Property 3: $g(0)$ is finite;

Property 4: $g(z)$ is strictly concave in $|z|$ or z^2.

Any function that holds the above properties is a candidate for effective SSR algorithms.

3.3.1 Iterative Reweighting Methods

In SSR, the iterative reweighted ℓ_2 [14, 15] and ℓ_1 [16] methods are popular batch estimation algorithms for solving diversity measure minimization problems. By introducing a weighted ℓ_2 or ℓ_1 norm term as an upper bound for the diversity measure term in each iteration, they form and solve for a new optimization problem to approach the optimal solution [13]. We adapt the reweighting methods to the online estimation setting here, where instead of (3.10), the following is suggested based on the reweighted ℓ_2 framework:

$$\min_{\mathbf{h}} \quad J_n(\mathbf{h}) + \lambda \|\mathbf{W}_n^{-1}\mathbf{h}\|_2^2, \tag{3.11}$$

or based on the reweighted ℓ_1 framework:

$$\min_{\mathbf{h}} \quad J_n(\mathbf{h}) + \lambda \|\mathbf{W}_n^{-1}\mathbf{h}\|_1, \tag{3.12}$$

where $\mathbf{W}_n = \text{diag}\{w_{i,n}\}$ is positive definite[2] and each $w_{i,n}$ is a function of the current estimate $h_{i,n}$ whose form depends on which framework and diversity measure are used.

To elaborate, for using the ℓ_2 framework (3.11), $g(z)$ has to be concave in z^2 for Property 4; i.e., it satisfies $g(z) = f(z^2)$, where $f(z)$ is concave for $z \in \mathbb{R}_+$. Based on [13], we have $w_{i,n}$ given as:

$$w_{i,n} = \left(\left. \frac{\mathrm{d}f(z)}{\mathrm{d}z} \right|_{z=h_{i,n}^2} \right)^{-\frac{1}{2}}. \tag{3.13}$$

For using the ℓ_1 framework (3.12), $g(z)$ has to be concave in $|z|$ for Property 4; i.e., it satisfies $g(z) = f(|z|)$, where $f(z)$ is concave for $z \in \mathbb{R}_+$. In this case, $w_{i,n}$ is given by:

$$w_{i,n} = \left(\left. \frac{\mathrm{d}f(z)}{\mathrm{d}z} \right|_{z=|h_{i,n}|} \right)^{-1}. \tag{3.14}$$

To utilize the reweighting scheme, we first choose an appropriate diversity measure $G(\mathbf{h})$ and then use (3.13) or (3.14) to obtain the update form of $w_{i,n}$. For example, consider the p-norm-like diversity measure [14, 42] with $g(h_i) = |h_i|^p$, $0 < p \leq 2$ for (3.11). Using (3.13) gives the $\mathbf{W}_n$ update [80]:

$$w_{i,n} = \left(\frac{2}{p} \left(|h_{i,n}| + c \right)^{2-p} \right)^{\frac{1}{2}}. \tag{3.15}$$

Note that we have added a small regularization constant $c > 0$ for stability purposes.[3] The p-norm-like diversity measure can also be adopted for (3.12) if $0 < p \leq 1$. In this case, we apply

[2]The positive definiteness can be shown to hold for a wide variety of diversity measures used in SSR. In cases where it is not, the positive definiteness can still be ensured by utilizing some small regularization constant.

[3]We suggest that c be kept relatively small as compared to the amplitude of the filter coefficients so that it would not affect the convergence significantly.

(3.14) to obtain the update equation for $\mathbf{W}_n$ [93]:

$$w_{i,n} = \frac{1}{p}\left(|h_{i,n}| + c\right)^{1-p}. \tag{3.16}$$

Again, a small constant $c > 0$ is added. In general, using a smaller p for (3.15) and (3.16) promotes stronger sparsity.

More options of the diversity measure $G(\mathbf{h})$ that have proved effective in SSR and the resulting $\mathbf{W}_n$ update forms can also be utilized as have been discussed in Section 2.4.2.

3.3.2 AST Methodology

We propose to reparameterize the problems (3.11) and (3.12) in terms of the (affinely) scaled variable $\mathbf{q}$:

$$\mathbf{q} \triangleq \mathbf{W}_n^{-1}\mathbf{h}, \tag{3.17}$$

in which $\mathbf{W}_n$ is used as the *scaling matrix*. This step can be interpreted as the AST commonly employed by the interior point approach to solving optimization problems [42, 60].

Now apply (3.17) to reparameterize the objective functions in (3.11) and (3.12) and perform minimization w.r.t. $\mathbf{q}$, that is:

$$\min_{\mathbf{q}} \quad J_n(\mathbf{W}_n\mathbf{q}) + \lambda\|\mathbf{q}\|_2^2 \tag{3.18}$$

and

$$\min_{\mathbf{q}} \quad J_n(\mathbf{W}_n\mathbf{q}) + \lambda\|\mathbf{q}\|_1 \tag{3.19}$$

for the reweighted ℓ_2 and ℓ_1 cases, respectively. Interestingly, if we set $\lambda = 0$, then both (3.18) and (3.19) become:

$$\min_{\mathbf{q}} \quad J_n(\mathbf{W}_n\mathbf{q}), \tag{3.20}$$

which actually applies a change of coordinates to the original problem (3.1) using (3.17). Since $\mathbf{W}_n$ is invertible, the problem of finding the $\mathbf{h}$ which minimizes $J_n(\mathbf{h})$ is equivalent to finding the $\mathbf{q}$ which minimizes $J_n(\mathbf{W}_n\mathbf{q})$. Therefore, the advantage of solving (3.20) is that the solution is guaranteed to also be a solution of (3.1), which is not true for (3.10) with $\lambda > 0$.

To proceed, define the *a posteriori* AST variable at time n:

$$\mathbf{q}_{n|n} \triangleq \mathbf{W}_n^{-1}\mathbf{h}_n \tag{3.21}$$

and the *a priori* AST variable at time n:

$$\mathbf{q}_{n+1|n} \triangleq \mathbf{W}_n^{-1}\mathbf{h}_{n+1}. \tag{3.22}$$

We can optimize (3.20) via CG recursion in the $\mathbf{q}$ domain:

$$\mathbf{q}_{n+1|n} = \mathbf{q}_{n|n} + \tilde{\mathbf{p}}_n\tilde{\alpha}_n. \tag{3.23}$$

For computing $\tilde{\mathbf{p}}_n$ and $\tilde{\alpha}_n$, note that since $J_n(\mathbf{W}_n\mathbf{q})$ is convex and quadratic in $\mathbf{q}$, minimizing it corresponds to solving the linear equation: $\mathbf{W}_n\mathbf{R}_n\mathbf{W}_n\mathbf{q} = \mathbf{W}_n\mathbf{b}_n$. Recognizing that it indirectly minimizes the quadratic function:

$$\tilde{F}_n(\mathbf{q}) \triangleq F_n(\mathbf{W}_n\mathbf{q}) = \frac{1}{2}\mathbf{q}^T\mathbf{W}_n\mathbf{R}_n\mathbf{W}_n\mathbf{q} - \mathbf{q}^T\mathbf{W}_n\mathbf{b}_n, \tag{3.24}$$

we can use Markov conjugacy w.r.t. $\mathbf{W}_n\mathbf{R}_n\mathbf{W}_n$ for the search directions. Thus, similar to (3.6), we update the search direction as: $\tilde{\mathbf{p}}_n = -\tilde{\mathbf{g}}_n + \tilde{\mathbf{p}}_{n-1}\tilde{\beta}_n$, where:

$$\tilde{\mathbf{g}}_n = \nabla_{\mathbf{q}}\tilde{F}_n(\mathbf{q}_{n|n}) = \mathbf{W}_n\nabla_{\mathbf{h}}F_n(\mathbf{h}_n) = \mathbf{W}_n\left(\mathbf{R}_n\mathbf{h}_n - \mathbf{b}_n\right) \tag{3.25}$$

by using the chain rule, (3.17), and (3.21), and

$$\tilde{\beta}_n = \frac{\tilde{\mathbf{p}}_{n-1}^T \mathbf{W}_n \mathbf{R}_n \mathbf{W}_n \tilde{\mathbf{g}}_n}{\tilde{\mathbf{p}}_{n-1}^T \mathbf{W}_n \mathbf{R}_n \mathbf{W}_n \tilde{\mathbf{p}}_{n-1}} \tag{3.26}$$

to sustain Markov conjugacy. For computing the step size, exact line search is performed:

$$\tilde{\alpha}_n = \arg\min_{\tilde{\alpha}} \tilde{F}_n(\mathbf{q}_{n|n} + \tilde{\mathbf{p}}_n \tilde{\alpha}) = -\frac{\tilde{\mathbf{p}}_n^T \tilde{\mathbf{g}}_n}{\tilde{\mathbf{p}}_n^T \mathbf{W}_n \mathbf{R}_n \mathbf{W}_n \tilde{\mathbf{p}}_n}. \tag{3.27}$$

Finally, noting that premultiplying both sides of the $\mathbf{q}$ domain update rule (3.23) by $\mathbf{W}_n$ and using the relationships (3.21) and (3.22), we have the equivalent update form in the $\mathbf{h}$ domain:

$$\mathbf{h}_{n+1} = \mathbf{h}_n + \mathbf{W}_n \tilde{\mathbf{p}}_n \tilde{\alpha}_n. \tag{3.28}$$

The above are summarized in Algorithm 3, which is the Sparsity-promoting CG (SCG) algorithm.

Algorithm 3: The proposed SCG adaptive filtering algorithm

Input: γ, δ, $\mathbf{u}_n$, d_n, choice of diversity measure
Output: $\mathbf{h}_n$
Initialize: $\mathbf{h}_0 = \mathbf{0}$, $\mathbf{R}_{-1} = \mathbf{0}$, $\mathbf{b}_{-1} = \mathbf{0}$, $\tilde{\mathbf{p}}_{-1} = \mathbf{0}$;
for $n = 0, 1, 2, \ldots$ **do**
  $\mathbf{R}_n = \gamma \mathbf{R}_{n-1} + \mathbf{u}_n \mathbf{u}_n^T$; ▷ corr. matrix update
  $\mathbf{b}_n = \gamma \mathbf{b}_{n-1} + \mathbf{u}_n d_n$; ▷ cross-corr. vector update
  Compute $\mathbf{W}_n$ according to the specified diversity measure (e.g., using (3.15) or (3.16));
  $\tilde{\mathbf{g}}_n = \mathbf{W}_n (\mathbf{R}_n \mathbf{h}_n - \mathbf{b}_n)$; ▷ compute gradient vector
  $\tilde{\beta}_n = \frac{\tilde{\mathbf{p}}_{n-1}^T \mathbf{W}_n \mathbf{R}_n \mathbf{W}_n \tilde{\mathbf{g}}_n}{\tilde{\mathbf{p}}_{n-1}^T \mathbf{W}_n \mathbf{R}_n \mathbf{W}_n \tilde{\mathbf{p}}_{n-1} + \delta}$; ▷ for Markov conjugacy
  $\tilde{\mathbf{p}}_n = -\tilde{\mathbf{g}}_n + \tilde{\mathbf{p}}_{n-1} \tilde{\beta}_n$; ▷ search direction update
  $\tilde{\alpha}_n = -\frac{\tilde{\mathbf{p}}_n^T \tilde{\mathbf{g}}_n}{\tilde{\mathbf{p}}_n^T \mathbf{W}_n \mathbf{R}_n \mathbf{W}_n \tilde{\mathbf{p}}_n + \delta}$; ▷ compute step size
  $\mathbf{h}_{n+1} = \mathbf{h}_n + \mathbf{W}_n \tilde{\mathbf{p}}_n \tilde{\alpha}_n$; ▷ adaptive filter update
end for

3.3.3 Discussion

By adopting $\lambda = 0$, SCG actually aims at solving (3.1), which is same for m-NLMS. Therefore, mathematically both SCG and m-NLMS should adapt toward the same optimum of (3.1). However, SCG utilizes $\mathbf{W}_n$ for leveraging sparsity. Effectively, as $w_{i,n}$ is typically a function of $|h_{i,n}|$, it tends to assign larger steps to coefficients with large magnitudes (e.g., see (3.28)). In this sense, it is similar to the proportionate NLMS (PNLMS) [27, 26], which redistributes the adaptation gains among all coefficients and emphasizes the large ones to speed up their convergence. The PNLMS has been widely used in sparse system identification. The SCG takes advantage of sparsity in a similar manner. Actually, if we use the instantaneous estimates $\mathbf{u}_n\mathbf{u}_n^T$ and $\mathbf{u}_n d_n$ for $\mathbf{R}_n$ and $\mathbf{b}_n$, respectively, and adopt the steepest descent by forcing $\tilde{\beta}_n = 0$, then Algorithm 3 reduces to a form similar to PNLMS. Note that in the preconditioned CG [94] there is also a matrix (preconditioner) used in a similar way as $\mathbf{W}_n$ for SCG, but the matrix is fixed though all iterations and its role is to reduce the condition number.

Complexity: SCG involves matrix-vector products and is thus in general of $\mathcal{O}(M^2)$ per-sample complexity, comparable to RLS but higher than LMS. Compared to the conventional RLS (without advanced techniques for improved stability [19]), SCG does not involve inverting the correlation matrix estimate and is thus numerically more stable like m-NLMS [49].

Convergence: We have the following theorem that establishes the convergence of SCG. While we do not prove it for a more general case, numerical results in Section 3.4 support the convergence of SCG in noisy, nonstationary environments.

Theorem 3.1. *Under a stationary environment assuming no measurement noise, SCG converges in the squared deviation sense.*

Proof: see Appendix 3.6.1.

3.4 Simulation Results

The proposed SCG algorithm is evaluated using MATLAB simulations. We set $\delta = 10^{-4}$ and used (3.15) for $\mathbf{W}_n$, setting $c = 0.001$. We compare the mean squared deviation (MSD) learning curves, i.e., the ensemble average of $\|\mathbf{h}^o - \mathbf{h}_n\|_2^2$ as a function of iteration n. The ensemble averaging was performed over 1000 independent Monte Carlo runs. In each run, the unknown channel IR with 100 taps was generated by randomly assigning the locations of the nonzeros. Each nonzero entry was drawn from $\mathcal{N}(0,1)$. The system noise $v_n \sim \mathcal{N}(0, 0.01)$. In all experiments, the adaptive filter coefficients were initialized with all zeros.

Figure 3.1 demonstrates the effect of the parameter p on SCG. The m-NLMS [49] is also compared. We used $\gamma = 0.95$ for both SCG and m-NLMS. The input was a zero mean, unit variance white Gaussian process. Figure 3.1 (a) considers the case of sparse systems with 5 nonzeros and Figure 3.1 (b) is the case of compressible systems with 20 nonzeros. The results indicate that SCG exploits the underlying system structure in the way we expect – a smaller p is favorable for a sparser system while a larger p is preferable for a less sparse system. In addition, with an appropriate $p < 2$, SCG can outperform m-NLMS should sparsity be present in the underlying system IR. Note that when $p = 2$, SCG is equivalent to m-NLMS according to (3.15), and the corresponding curves overlap with each other.

Next, we compare SCG with existing approaches for identifying sparse systems with 5 nonzeros. We set $p = 1$ for SCG and used $\gamma = 0.98$ for all algorithms. To see the behavior in a changing environment, we shifted the system IR to the right by 16 samples in the middle of the adaptation process.

Figure 3.2 compares SCG with several sparsity regularized RLS-type algorithms, namely, the ℓ_1-RLS [90, 91], ℓ_1-RRLS [91], and ℓ_0-RLS [90]. These algorithms utilize $\lambda > 0$ in order to incorporate sparsity. The input was a first order autoregressive process according to $u_n = \rho u_{n-1} + \eta_n$, where $\rho = 0.8$ and $\eta_n \sim \mathcal{N}(0,1)$. The standard RLS [19] is also compared. From the results we

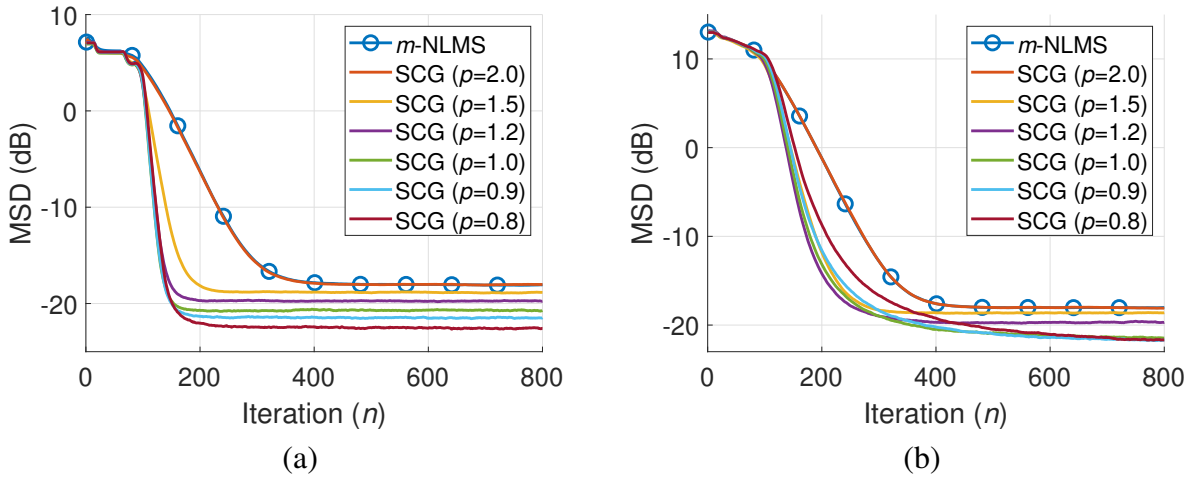


Figure 3.1: Effect of p on SCG for (a) sparse and (b) compressible systems using white Gaussian process as input.

see that SCG demonstrates fast convergence and achieves the lowest steady-state error. This could be attributed to $\lambda = 0$ in SCG which enables it to exploit the sparsity of the system IR if it already exists, while avoiding the potential bias incurred by strictly enforcing it.

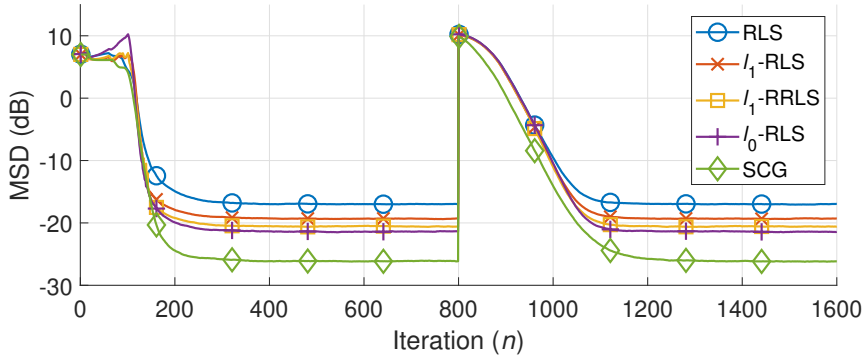


Figure 3.2: Comparison of SCG with sparsity regularized RLS-type algorithms using autoregressive process as input.

Figure 3.3 compares SCG with existing CG-type adaptive filtering algorithms, namely, the (standard) CG [45], modified CG [46], CG-CLF [47], and m-NLMS [49] algorithms. We used a speech signal as input and the signal-to-noise ratio was set to 20 dB. The normalized misalignment, i.e., $\|\mathbf{h}^o - \mathbf{h}_n\|_2^2 / \|\mathbf{h}^o\|_2^2$, was used for performance evaluation. It can be seen that SCG, by taking advantage of sparsity, demonstrates superior convergence performance over the

other CG-type algorithms that do not incorporate sparsity.

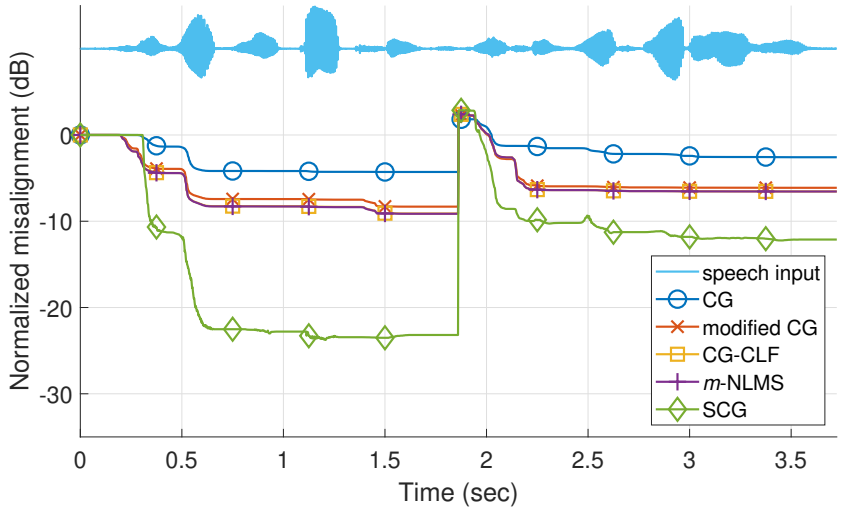


Figure 3.3: Comparison of SCG with existing CG adaptive filtering algorithms using speech as input.

3.5 Conclusion

In this chapter, we introduced SCG, a CG-based adaptive filter that leverages sparsity for improved adaptation when the underlying system IR is sparse. The SCG is derived by utilizing AST within the iterative reweighting SSR framework, which leads to admitting a zero regularization coefficient of the regularizer while promoting sparsity. Simulation results demonstrated that SCG is effective for system identification problems with sparse IRs.

Acknowledgment

Chapter 3 is, in part, a reprint of the material as it appears in the paper: C.-H. Lee, B. D. Rao, and H. Garudadri, "A sparse conjugate gradient adaptive filter," *IEEE Signal Processing Letters*, 2020. The dissertation author was the primary investigator and author of this paper. The work, in part, was supported by National Institutes of Health/National Institute on Deafness and

Other Communication Disorders under Grants R01DC015436 and R33DC015046 and National Science Foundation/Information and Intelligent Systems under Award 1838830.

3.6 Appendix

3.6.1 Proof of Theorem 3.1

The proof follows the idea in [48]. Define the instantaneous squared deviation as: $\boldsymbol{\varepsilon}_{n+1} \triangleq \|\mathbf{h}^o - \mathbf{h}_{n+1}\|_2^2$ and the difference of $\boldsymbol{\varepsilon}_{n+1}$ as:

$$\Delta\boldsymbol{\varepsilon}_{n+1} \triangleq \boldsymbol{\varepsilon}_{n+1} - \boldsymbol{\varepsilon}_n = \|\mathbf{h}^o - \mathbf{h}_{n+1}\|_2^2 - \|\mathbf{h}^o - \mathbf{h}_n\|_2^2. \tag{3.29}$$

Substituting the filter update rule (3.28) into (3.29), we have:

$$\Delta\boldsymbol{\varepsilon}_{n+1} = -2\tilde{\alpha}_n \tilde{\mathbf{p}}_n^T \mathbf{W}_n (\mathbf{h}^o - \mathbf{h}_n) + \tilde{\alpha}_n^2 \tilde{\mathbf{p}}_n^T \mathbf{W}_n^2 \tilde{\mathbf{p}}_n. \tag{3.30}$$

If we focus on the "homogeneous" case [95] that d_n is exactly given as $\mathbf{u}_n\mathbf{h}^o$, we have $\mathbf{b}_n = \mathbf{R}_n\mathbf{h}^o$. In this case, we can rewrite $\tilde{\alpha}_n$ in (3.27) using (3.25), leading to:

$$\tilde{\alpha}_n = \frac{\tilde{\mathbf{p}}_n^T \mathbf{W}_n \mathbf{R}_n (\mathbf{h}^o - \mathbf{h}_n)}{\tilde{\mathbf{p}}_n^T \mathbf{W}_n \mathbf{R}_n \mathbf{W}_n \tilde{\mathbf{p}}_n} \approx \frac{\tilde{\mathbf{p}}_n^T \mathbf{W}_n (\mathbf{h}^o - \mathbf{h}_n)}{\tilde{\mathbf{p}}_n^T \mathbf{W}_n^2 \tilde{\mathbf{p}}_n}, \tag{3.31}$$

where the approximation is used under the assumption of stationary input [48]. Substituting (3.31) into (3.30) we obtain:

$$\Delta\boldsymbol{\varepsilon}_{n+1} = -\frac{\left(\tilde{\mathbf{p}}_n^T \mathbf{W}_n (\mathbf{h}^o - \mathbf{h}_n)\right)^2}{\tilde{\mathbf{p}}_n^T \mathbf{W}_n^2 \tilde{\mathbf{p}}_n} \leq 0, \tag{3.32}$$

by noting the positive definiteness of $\mathbf{W}_n^2$. From (3.29), we have:

$$\lim_{n\to\infty} \boldsymbol{\varepsilon}_{n+1} = \lim_{n\to\infty} \boldsymbol{\varepsilon}_n + \lim_{n\to\infty} \Delta\boldsymbol{\varepsilon}_{n+1} = \boldsymbol{\varepsilon}_0 + \lim_{n\to\infty} \sum_{\tau=0}^{n} \Delta\boldsymbol{\varepsilon}_{\tau+1} \geq 0. \tag{3.33}$$

This implies:

$$\lim_{n\to\infty} \sum_{\tau=0}^{n} \Delta\boldsymbol{\varepsilon}_{\tau+1} \geq -\boldsymbol{\varepsilon}_0. \tag{3.34}$$

By (3.32), it indicates that the term $\Delta\boldsymbol{\varepsilon}_{n+1}$ is summable for infinite n. Therefore, we can conclude that $\lim_{n\to\infty} \Delta\varepsilon_{n+1} = 0$. From (3.32), it implies $\lim_{n\to\infty} \tilde{\mathbf{p}}_n^T \mathbf{W}_n(\mathbf{h}^o - \mathbf{h}_n) = 0$. Since $\mathbf{W}_n\tilde{\mathbf{p}}_n$ and $\mathbf{h}^o - \mathbf{h}_n$ are unlikely to be orthogonal, it can be concluded that the filter estimate $\mathbf{h}_n$ tends to the optimal $\mathbf{h}^o$ as $n \to \infty$.

Chapter 4

Improved Acoustic Feedback Reduction Using Novel Sparse Adaptive Filtering and Frequency Warping Techniques

In hearing aids (HAs), the acoustic coupling between the microphone and the receiver (i.e., the loudspeaker) results in the system becoming unstable under certain conditions and causes acoustic feedback artifacts commonly referred to as whistling or howling. Adaptive feedback cancellation (AFC) techniques have been the work horse for acoustic feedback reduction, where the feedback path is modeled as a finite-impulse-response filter whose filter coefficients are continuously adjusting to emulate the feedback path characteristics. Due to their simplicity and effectiveness, the least mean square (LMS) class of algorithms are commonly used in AFC. Furthermore, the sparse nature of the feedback path impulse response enables improved AFC by leveraging the sparsity. In this chapter, we apply the Sparsity-promoting LMS (SLMS) algorithm to the AFC problem in HAs and show that improvements in terms of speech quality and stable gain are possible by exploiting sparsity. Moreover, on top of the SLMS we introduce a frequency

warping technique that we call “freping” based on a novel use of all-pass networks to further improve the AFC by mitigating the Nyquist stability criterion.

4.1 Introduction

To compensate for mild to moderate hearing loss, commercial hearing aids (HAs) provide an average gain of $35-38$ dB. In the emerging form factors for advanced HAs and hearables, there is a significant acoustic coupling between the microphones and loudspeakers (called receivers in the telephony and HA communities). This acoustic coupling varies significantly based on surroundings (e.g. hats, scarves, hands, and walls that come in close proximity to the transducers) and can cause the system to become unstable, when the audio content includes characteristic frequencies of the system. This instability results in acoustic feedback artifacts such as brief “howling” that can be of immense annoyance to the HA users. As a result, acoustic feedback reduction continues to be a challenging problem due to the emerging form factors in advanced HAs and hearables.

Howling artifacts manifest when multiple factors collude to fulfill the magnitude and phase conditions of the Nyquist stability criterion (NSC) [96]. Adaptive feedback cancellation (AFC) has been the work horse for breaking NSC to avoid instabilities in many audio applications [51], including HAs [97, 98, 99]. Typically, the AFC deploys the least mean square (LMS) based approaches to mitigate the magnitude condition in NSC [100, 101, 56]. On the other hand, frequency shifting (FS) [102, 103, 104] and other ad hoc methods [105, 106, 107] mainly deal with the phase condition. In this chapter, we focus on spectral manipulations following LMS based approaches to break NSC in both magnitude and phase conditions.

In AFC, an adaptive filter is continuously adjusting to approximate the impulse response (IR) of the acoustic feedback path. In the adaptation stage, LMS algorithms [19] are the most widely used techniques due to computational simplicity and their effectiveness. However, the

conventional LMS suffers from slow convergence especially at the presence of correlated signals. As a result, it might fail to track the changes of the feedback path IR in a highly time-varying environment.

A natural question of interest is: can we further improve the convergence behavior of the LMS-based AFC from other aspects? Observing that typical feedback path IRs are (quasi-) sparse as shown in Figure 4.7, one might think of taking advantage of this sparseness for improvements. This can actually be carried out by the concept of proportionate adaptation that originated from the proportionate normalized LMS (PNLMS) algorithm [27]. The main idea behind proportionate adaptation is to update each filter coefficient independently of the others by assigning to the corresponding step size a weight in proportion to the magnitude of the estimated coefficient. In other words, it redistributes the adaptation gains among all coefficients and emphasizes the large ones in order to speed up their convergence.

However, the original PNLMS has the problem that it is more beneficial for systems with very sparse structures. For AFC application where the feedback path IRs are usually quasi-sparse, other proportionate-type LMS algorithms can be more suitable. For example, the improved PNLMS (IPNLMS) [28] and the IPNLMS-ℓ_0 [29] have the flexibility for identifying systems of different levels of sparsity. Attempts have been made to incorporate these proportionate algorithms into AFC [108, 109, 110] and improvements have been reported. However, these proportionate algorithms were not formally derived by minimizing any underlying objective functions so that their usage can be further optimized. Moreover, the parameters within these algorithms do not have direct connections to the sparsity degree of the underlying system they aim to identify.

In this chapter, we apply the Sparsity-promoting LMS (SLMS) algorithm to AFC which is found to be suitable for estimating the quasi-sparse feedback path IRs. The benefit of SLMS is brought by its direct connection to the system sparseness which provides a practical way of parameter selection, and it enjoys theoretical support, simpler parameter optimization, and more straightforward leverage of (quasi-) sparsity in acoustic feedback paths. In addition, on top of

SLMS we further propose a novel decorrelation algorithm called "freping," a portmanteau for frequency warping, to improve the AFC. We show that SLMS and freping, when utilized together, can achieve significant improvements for AFC in terms of speech quality and stable gain. The algorithms developed in this chapter have been implemented and run real-time on the Open Speech Platform [52, 53].

Organization of the Chapter: The rest of the chapter is organized as follows. Section 4.2 provides background on the AFC problem in HAs. Section 4.3 applies the SLMS to improve AFC by leveraging the sparsity nature of the acoustic feedback paths. Section 4.4 presents the freping technique for decorrelation in AFC. Section 4.5 discusses the trade-off between speech quality and stable gain in AFC, and introduces a novel AFC evaluation approach based on a quality metric. Section 4.6 presents simulation results. Section 4.7 concludes the chapter.

4.2 Acoustic Feedback Problem

In HAs, the output sound at the receiver can be picked up by the microphone due to the short distance between the two, resulting in acoustic feedback as illustrated in Figure 4.1. The acoustic feedback phenomenon not only degrades the audio quality of the HA output signal but also limits the amount of amplification, or the maximum stable gain (MSG), that a HA device can provide to the user. To overcome this problem many AFC techniques have been proposed for modern HAs [51]. In AFC, an adaptive finite impulse response (FIR) filter is used to emulate the feedback path IR, aiming at increasing the MSG while minimizing speech distortion.

4.2.1 AFC System

A typical AFC framework is depicted in Figure 4.2. The AFC filter $H_n(z)$, placed in parallel with the HA processing $G_n(z)$, is the transfer function of an M-tap adaptive filter $\mathbf{h}_n = [h_{0,n}, h_{1,n}, \ldots, h_{M-1,n}]^T$ that continuously adjusts its coefficients to capture the time-varying

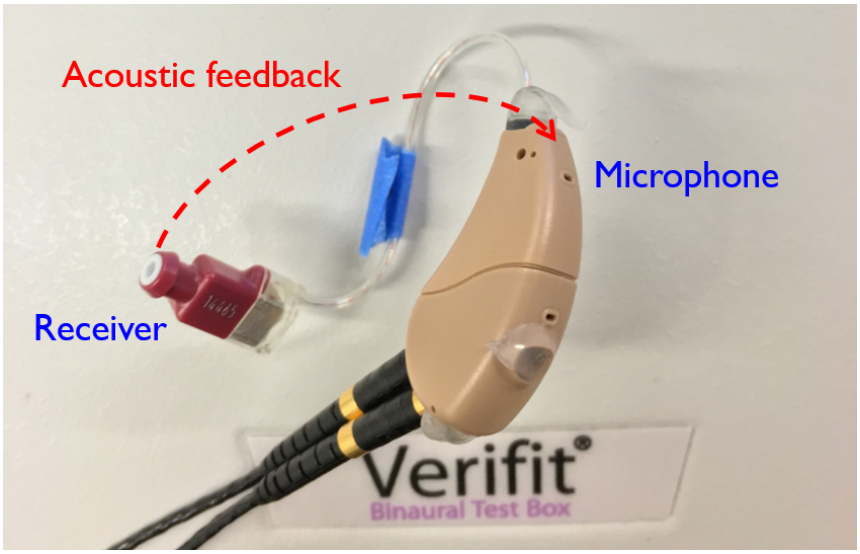


Figure 4.1: Illustration of acoustic feedback in HAs.

nature of the acoustic feedback path $F_n(z)$. d_n is the microphone input which contains the clean signal x_n and the feedback signal y_n caused by the HA output o_n passing through the feedback path. $\hat{y}_n$ is the feedback estimate. $e_n = d_n - \hat{y}_n$ is the feedback-compensated signal. $A_n(z)$ is a time-varying pre-filter to decorrelate the input and output signals based on the prediction error method (PEM) [97]. $B(z)$ is a band-limited filter to concentrate on the frequency region where oscillation is more likely to occur [100].

Typically, LMS-type algorithms are carried out for coefficient adaptation using the pre-filtered signals u_n^f and e_n^f to update the AFC filter $\mathbf{h}_n$ as:

$$\mathbf{h}(n+1) = \mathbf{h}_n + \frac{\mu}{M\hat{\sigma}_n^2 + \delta}\mathbf{u}_n^f e_n^f, \tag{4.1}$$

where $\mathbf{u}_n^f = [u_n^f, u^f(n-1), \ldots, u^f(n-M+1)]^T$, $\mu > 0$ is the step size parameter, $\delta > 0$ is a small constant to prevent division by zero, and

$$\hat{\sigma}_n^2 = \rho\hat{\sigma}_{n-1}^2 + (1-\rho)\left((u_n^f)^2 + (e_n^f)^2\right) \tag{4.2}$$

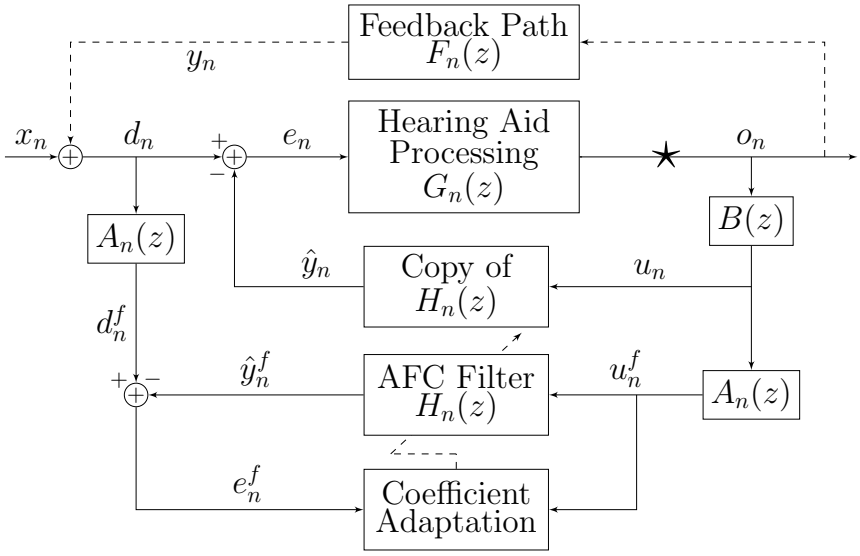


Figure 4.2: Block diagram of the AFC framework.

is the power estimate with a forgetting factor $0 < \rho \leq 1$. Using (4.2), the update rule (4.1) is actually the "modified" LMS using the sum method proposed in [111] to accommodate the time-varying nature of speech signals. It has been widely adopted in many AFC and speech processing systems [97, 100, 101].

4.2.2 Mitigating NSC

Without any feedback control mechanism, the frequency responses of the HA processing $G_n(e^{j\omega})$ and the feedback path $F_n(e^{j\omega})$ form a closed-loop system which exhibits instability that leads to howling. The NSC [96] states that the closed-loop system becomes unstable whenever the following magnitude and phase conditions are both fulfilled [98]:

$$\begin{cases} |G_n(e^{j\omega})F_n(e^{j\omega})| \geq 1, & \text{(magnitude condition)} \\ \angle G_n(e^{j\omega})F_n(e^{j\omega}) = 2\pi l, \quad l \in \mathbb{Z} & \text{(phase condition)} \end{cases}, \tag{4.3}$$

where $\mathbb{Z}$ denotes the set of integers.

When AFC in employed, it becomes:

$$\begin{cases} |G_n(e^{j\omega})(F_n(e^{j\omega}) - \hat{F}_n(e^{j\omega}))| \geq 1, \\ \angle G_n(e^{j\omega})(F_n(e^{j\omega}) - \hat{F}_n(e^{j\omega})) = 2\pi l, \quad l \in \mathbb{Z} \end{cases}, \tag{4.4}$$

where $\hat{F}_n(e^{j\omega}) = B(e^{j\omega})W_n(e^{j\omega})$ is the estimated feedback path frequency response. The AFC aims at minimizing $|F_n(e^{j\omega}) - \hat{F}_n(e^{j\omega})|$ to mitigate the magnitude condition.

4.3 Sparsity-Promoting LMS for AFC

Observing that typical feedback path IRs are sparse (to some degree) as, for example, the one shown in Figure 4.7, one might think of taking advantage of this sparseness for AFC improvements. Based on the iterative reweighting framework introduced in the previous chapters, we can leverage the sparsity of the feedback path IR via a sparsity-promoting matrix $\mathbf{S}_n$ to achieve faster convergence for improvement. Furthermore, to account for speech characteristics, we adopt the power estimate of the "modified" LMS (4.1), i.e., (4.2), leading to the "modified" SLMS update rule for AFC:

$$\mathbf{h}_{n+1} = \mathbf{h}_n + \frac{\mu}{M\hat{\sigma}_n^2 + \delta}\mathbf{S}_n\mathbf{u}_n^f e_n^f, \tag{4.5}$$

where $\mathbf{S}_n = \text{diag}\{s_{i,n}\}$ is an M-by-M diagonal matrix and the diagonal elements are updated according to:

$$s_{i,n} = \frac{w_{i,n}^2}{\frac{1}{M}\sum_{j=0}^{M-1} w_{j,n}^2}, \tag{4.6}$$

for $i = 0, 1, \ldots, M-1$ where $w_{i,n}$ depends on the reweighting framework and diversity measure used (for more details see Section 2.2.2). For example, using the p-norm-like diversity measure

$\|\mathbf{h}\|_p^p$ within the reweighted ℓ_2 framework we have:

$$w_{i,n} = \left(\frac{2}{p} \left(|h_{i,n}| + c \right)^{2-p} \right)^{\frac{1}{2}}, \tag{4.7}$$

where $p \in (0, 2]$ is the sparsity control parameter and $c > 0$ is a small positive constant to avoid stagnation of the algorithm. Note that the parameter p is influential: a sparse system would benefit more from a smaller p while for a dispersive system, p close to 2 would be more preferable. For the quasi-sparse feedback IRs like that in Figure 4.7 in AFC, we expect that the optimal p value would lie between 1 and 2.

4.4 Mitigating Acoustic Feedback with Frequency Warping by All-Pass Networks

It is well-known that the LMS-type algorithms widely used in AFC suffer from biased estimation due to signal correlation [112]. Consequently, the feedback path estimate can be erroneous if decorrelation is not carefully considered. Although the PEM-based pre-filter [97] has provided certain amount of decorrelation, further improvement is achievable by inserting additional signal processing into the forward path of the HA [104], usually placed at $\star$ as shown in Figure 4.2, to manipulate the HA output. As a result, quality degradation might be introduced by these decorrelation methods and thus there is generally the compromise between the sound quality and the decorrelation ability for AFC improvement. Existing methods for decorrelation include FS [102, 103, 104], phase modulation [105], time-varying all-pass filters to introduce phase shifts [106], linear predictive coding vocoder [107], to name a few.

Different from the previous works, we present a novel use of well-known all-pass filters in a network to perform frequency warping that we call "freping." Freping helps in breaking the NSC criterion and improves AFC further. This frequency warping is accomplished using an all-pass

network proposed by Oppenheim and Johnson [113], which realizes a nonlinear mapping of the frequency axis as controlled by a single warping parameter α. More formally, let $\omega = 2\pi(f/f_s)$ be the normalized angular frequency where f is the original frequency and f_s is the sampling rate. The nonlinear frequency mapping $\phi(\cdot)$ is according to [113]:

$$\hat{\omega} = \phi(\omega) = \omega + 2\arctan\left(\frac{\alpha\sin\omega}{1-\alpha\cos\omega}\right), \quad -1 < \alpha < 1, \tag{4.8}$$

where $\hat{\omega} = 2\pi(\hat{f}/f_s)$ and $\hat{f}$ is the warped frequency.

It can be shown that the mapping (4.8) between the original signal $v(n)$ and the frequency-warped signal $q(k)$ can be achieved by passing the time-reversed signal $v(-n)$ through a linear time-invariant system $T_k(z)$ given as:

$$T_k(z) = \begin{cases} \dfrac{(1-\alpha^2)z^{-1}}{(1-\alpha z^{-1})^2}\left(\dfrac{z^{-1}-\alpha}{1-\alpha z^{-1}}\right)^{k-1}, & k > 0 \\ \dfrac{1}{1-\alpha z^{-1}}, & k = 0 \end{cases}, \tag{4.9}$$

and taking the output of $T_k(z)$ at $n = 0$ as $q(k)$. It can thus be implemented as the network shown in Figure 4.3. The first two stages act as i) low-pass filters when α is positive and the network warps frequencies higher and ii) high-pass filters when α is negative and the network warps frequencies lower. The remaining stages realize the actual frequency warping based on the bilinear transformation [114]. Note that when $\alpha = 0$, it simply passes through the input without any spectral modifications.

The frequency-warped output is given by sampling $\widetilde{q}_k(n)$, the output signal at the k-th stage, along the cascade chain at $n = 0$, i.e., $q(k) = \widetilde{q}_k(0)$. In other words, the input sequence is first flipped and then passed through the network; the last sample of the output sequence at the k-th stage is taken as the k-th sample of the final frequency-warped sequence [115].

It is worth noting that in practice we need to truncate the signal for the all-pass network to

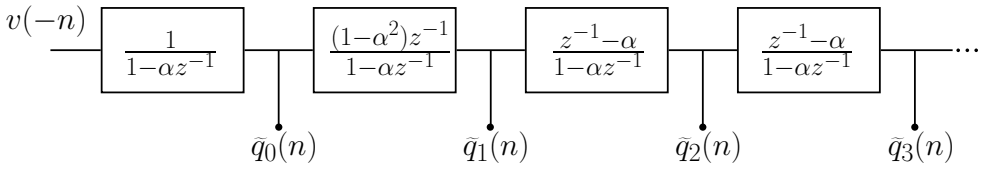


Figure 4.3: The all-pass network for frequency warping.

be realizable. Therefore, the warping performance will depend on other factors such as the length and the type of the window function used.

4.4.1 Freping: Real-Time Frequency Warping

The all-pass networks described above are adopted for real-time frequency manipulations as illustrated in Figure 4.4. The input signal is first divided into overlapping frames and windowed using a proper window function. Each windowed segment then goes through the all-pass network to perform frequency warping with a specified warping parameter α. Finally, the overlap-add method [116] is applied to produce the frequency-warped signal.

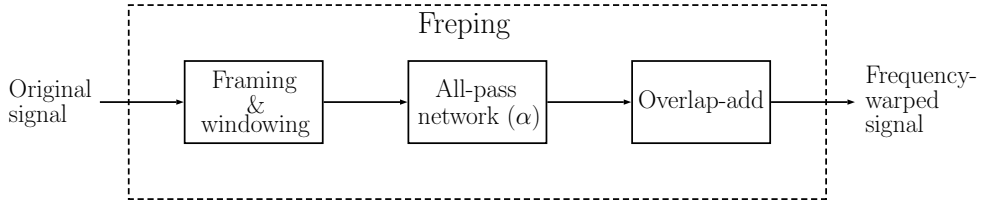


Figure 4.4: Short-time frequency warping using all-pass network.

To allow a more flexible way of manipulating spectral characteristics, we propose the multichannel freping as illustrated in Figure 4.5. The system utilizes a set of band-pass filters (BPFs) which divide the input signal into M frequency bands and a set of warping parameters $\boldsymbol{\alpha} = [\alpha_1, ..., \alpha_M]^T$. Each band goes through an independent all-pass network with the corresponding warping parameter. The output signals of all the frequency bands are summed up to produce the frequency-warped signal.

In many practical situations, it is convenient to reuse the multichannel compression modules [117] in HA processing for freping. For specific types of hearing loss (e.g. sloping,

cookie-bite, etc.), increasing the gain in higher frequency bands aids to fulfill the magnitude condition of NSC and freping hinders the phase condition to occur. Thus, freping provides a way for simultaneously optimizing the parameters of multichannel compression and frequency lowering [118] in HAs for individual hearing loss. In this work, we limit ourselves to negative values of α so that freping always shifts spectral content lower.

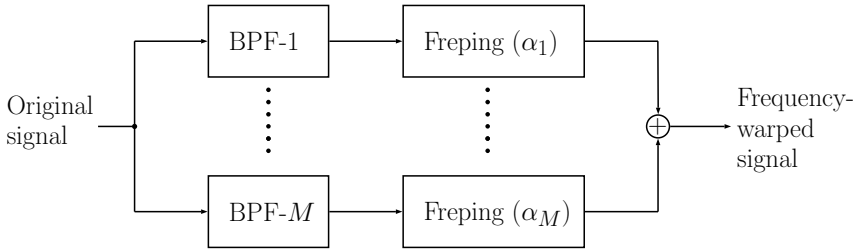


Figure 4.5: Multichannel freping.

Freping is an extreme version of FS [119] and it plays a similar role for decorrelation. It introduces nonlinear frequency shifts and the distortions appear to be perceptually benign based on informal subjective assessments. As instability is most likely to occur at the high-frequency region, it is reasonable to manipulate the high-frequency content while keeping the low-frequency region intact to avoid degradation in quality. By providing additional decorrelation, freping can reduce the AFC bias and thus a better feedback path estimate can be obtained, thereby improving the magnitude condition in NSC. On the other hand, freping also helps avoid the microphone and receiver signals from remaining continuously in phase with each other. This prevents the phase condition in NSC to hold at the same frequency at two consecutive instants. Consequently, the input and output sounds could not build up in amplitude as effectively. Therefore, the likelihood of instability is reduced.

4.5 Speech Quality and Stable Gain Trade-Offs

To quantify the AFC performance, one useful measure is the added stable gain (ASG) defined as the amount of additional MSG brought by AFC that the HA can still operate in the stable

state. We discuss the trade-off between speech quality and stable gain in AFC, by introducing a novel ASG estimation approach based on the hearing-aid speech quality index (HASQI) [120].

4.5.1 Quality Metric: HASQI

The impact of acoustic feedback on perceived speech quality is estimated using the HASQI version-2 speech quality metric [120]. The HASQI metric was trained on a large database of subject quality ratings, including nonlinear distortion and frequency response modifications that duplicated the resonance peaks typical of acoustic feedback. The metric was validated on data from a feedback cancellation experiment [121], and a value of 0.93 was found for the correlation coefficient between the subject ratings and the HASQI quality predictions [120]. In addition, recent papers have shown high degrees of correlation for perceptual metrics used to predict quality ratings for feedback cancellation in HAs [122, 112]. However, the idea of using HASQI as an objective metric for optimizing AFC is novel.

HASQI compares the processed HA signal to a reference signal. In this work, the reference signal is the unmodified computer audio file x_n, and the processed HA signal is the feedback-compensated signal e_n. Both the reference and processed signals are passed through a model of the auditory periphery. The auditory model includes auditory frequency analysis, the dynamic range compression mediated by the outer hair cells, two-tone suppression and the firing-rate adaptation present in the inner hair cell neural response. The metric compares the time-frequency envelope modulation, temporal fine structure, and long-term spectra between the processed and reference signals to produce the quality prediction. The HASQI model represents a distillation of listener ratings for a large number of linear filtering, noise, and distortion conditions. Since the metric was fit to these responses, the perceptual ratings are built into the predicted quality scores. In addition, HASQI has been validated by several perceptual quality experiments [123, 124, 125]. HASQI is therefore sensitive to changes in the speech spectrum introduced by acoustic feedback, whistling or ringing in the HA, and any nonlinear distortion introduced by the

feedback-cancellation processing.

4.5.2 Proposed HASQI-Based ASG Estimation Approach

For the purpose of estimating the ASG of the AFC algorithm, a uniform gain of the HA processing over all the sub-bands is applied. That is, we use $G_n(z) = gz^{-\tau}$, where g represents the gain of the HA and τ corresponds to the HA processing delay. The ASG by definition is given as the difference between the MSG of the system with the use of the AFC algorithm and that without the use of AFC (in dB):

$$ASG = MSG_{w/\,AFC} - MSG_{w/o\,AFC}. \tag{4.10}$$

To obtain the ASG estimate, we propose the following procedure:

i) Define a threshold $\theta \in (0, 1)$.

ii) Start from $g = 1$,

   a) Run the AFC algorithm on a given audio signal x_n and obtain the feedback-compensated signal e_n.

   b) Compute the HASQI of e_n using x_n as the reference signal. Record the obtained HASQI score.

   c) If the obtained score $\geq \theta$,

      Increase g by some small increment, e.g., $\Delta g = 0.1$, and then repeat from ii)-a).

      Else,

      Use the previous g value as the estimate of the MSG. Terminate.

iii) Perform ii) for both with AFC and without AFC cases to obtain $MSG_{w/\,AFC}$ and $MSG_{w/o\,AFC}$, respectively. Use (4.10) to obtain the ASG estimate (convert into dB first) of the AFC algorithm.

iv) Repeat ii) and iii) for multiple audio files. Average over the resulting ASG numbers to obtain the final ASG estimate.

We will also obtain a quality vs. gain curve once the above procedure has been done for a particular AFC algorithm with a given audio file. Typically, the quality score will decrease as the gain value increases. Figure 4.6 presents some example curves for various AFC algorithms. We can see that without AFC the curve falls rapidly as gain increases. On the other hand, the SLMS with decorrelation (here using PEM filters) retains the highest quality even for higher gain values. This demonstrates the importance of a good AFC system for HAs.

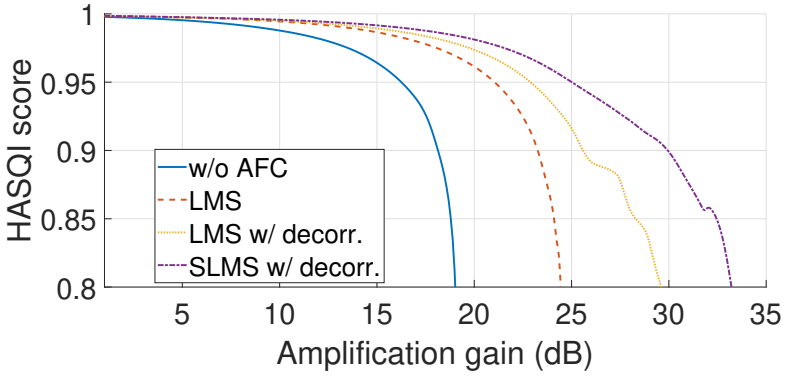


Figure 4.6: Block diagram of the AFC framework.

Once the score falls below the pre-defined threshold θ, the speech quality is considered unacceptable: we therefore consider the gain at which the system enters the unacceptable state as the MSG of the system. In our work, a HASQI score of 0.8 was used as the threshold for acceptable/unacceptable states. Table 4.1 presents some examples of the measured ASG values using the approach. The HASQI value of 0.8 is consistent with a high quality rating as reported for HA quality evaluations [123]. Because the data are simulation results with no other sources of noise or distortion, the maximum possible HASQI score is 1; a value of 0.8 thus represents a measurable degradation in signal quality. Nevertheless, the proposed methodology can still be

used for any value of HASQI. For example, a resource constrained HA may target lower speech quality to save power.

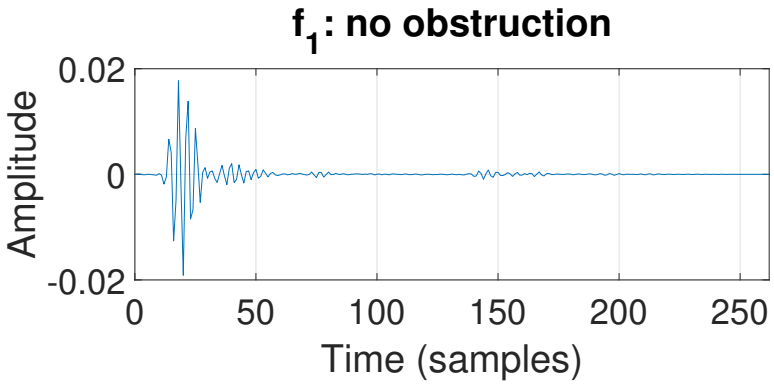
Table 4.1: Estimated ASG (in dB) of different AFC algorithms.

Input File	MSG w/o AFC	**ASG** LMS	**ASG** LMS w/ decorr.	**ASG** SLMS w/ decorr.
male 1	18.89	**5.35**	**10.74**	**13.60**
male 2	18.59	**5.38**	**7.73**	**10.63**
male 3	18.99	**5.41**	**10.64**	**14.21**
female 1	18.99	**5.25**	**9.57**	**12.09**
female 2	18.89	**5.46**	**9.96**	**12.61**
female 3	18.89	**5.35**	**7.60**	**10.59**
classical	18.99	**5.41**	**12.26**	**15.36**
jazz	19.08	**5.63**	**11.10**	**14.21**
choir	18.99	**5.36**	**11.82**	**14.72**
pop	18.79	**5.40**	**10.25**	**12.81**
Average	18.91	**5.40**	**10.17**	**13.08**

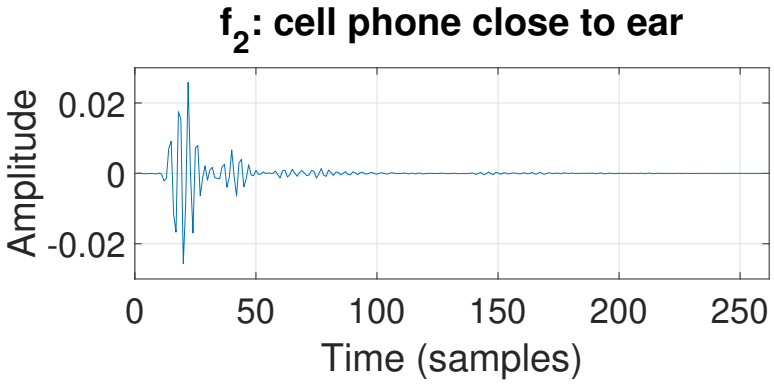
4.6 Simulation Results

4.6.1 SLMS

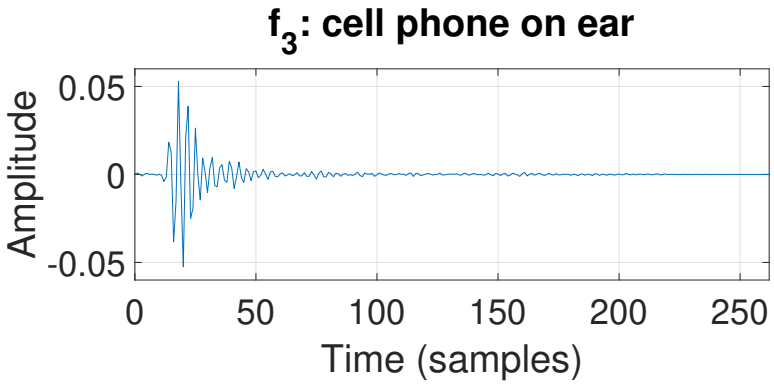
The proposed SLMS AFC algorithm (4.5) using (4.7) for promoting sparsity is evaluated using computer simulations in MATLAB at a sampling rate of 16 kHz. The PEM-AFC framework described in Figure 4.2.1 was adopted. The HA processing $G_n(z) = gz^{-\tau}$ with $g = 20$ and τ corresponding to a delay of 8 ms. The feedback path IRs were measured using a behind-the-ear HA with open fitting on a dummy head and truncated to a length of 263 samples as shown in Figure 4.7. The AFC filter length was $M = 100$ to cover the significant part of the IRs. The forgetting factor $\rho = 0.985$. The step size parameter $\mu = 0.005$. Small positive constants $\delta = c = 10^{-6}$. The band-limited filter $B(z) = 1 - 1.8z^{-1} + 0.81z^{-2}$ as used in [126]. The pre-filter $A_n(z)$ was an FIR filter of order 20 updated every 10 ms via linear prediction of e_n [127].



(a)



(b)



(c)

Figure 4.7: Measured acoustic feedback path IRs of (a) f_1: no obstruction, (b) f_2: with a cellphone close to the ear, and (c) f_3: with a cellphone right on the ear. Mind the different scales of the y-axis (Amplitude).

We ran the AFC system with the SLMS on 25 male and 25 female speech signals from TIMIT database and measured the corresponding HASQI of the feedback-compensated signal e_n. Average HASQI scores over the 50 speech files for different values of p are shown in Figure 4.8. We can see that the optimal p almost lies in the same range even as the feedback IR differs. This means, for a given HA device, if we have some rough knowledge about the sparsity degree of its feedback channel, the SLMS is robust since p is not very sensitive near the optimal point. From the results we found p around 1.5 to be a good choice.

We also compare the SLMS (using $p = 1.5$) with the LMS (4.1) and other proportionate algorithms:

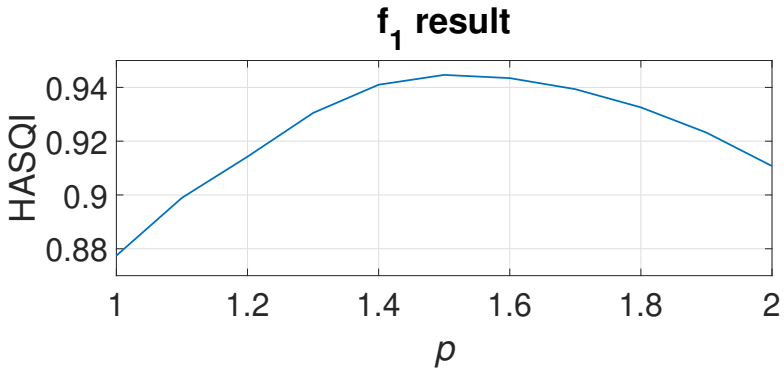
$$\mathbf{h}_{n+1} = \mathbf{h}_n + \frac{\mu}{M\hat{\sigma}_n^2 + \delta}\mathbf{\Gamma}_n \mathbf{u}_n^f e_n^f, \tag{4.11}$$

where $\mathbf{\Gamma}_n$ uses the proportionate matrices of the PNLMS [27], IPNLMS [28], and IPNLMS-ℓ_0 [67]. The algorithms were run on the speech dataset and the average HASQI were measured under 4 different feedback scenarios as shown in Figure 4.9. We see that the SLMS outperforms all the other ones, especially obvious under an adverse feedback situation such as the last two cases (about 0.25 HASQI improvement compared to the NLMS in the last case).

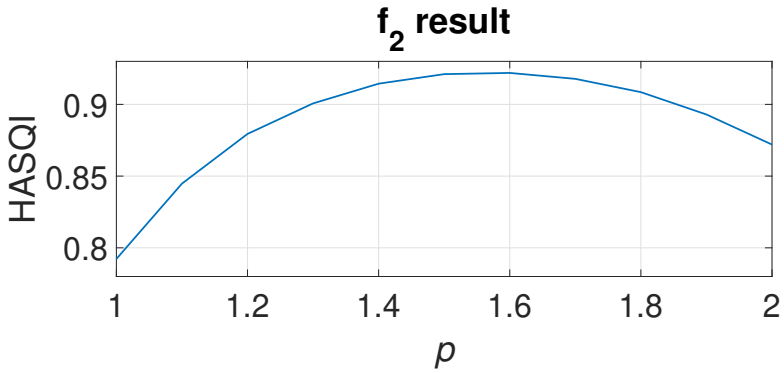
4.6.2 Freping

We evaluate the proposed freping system using computer simulations in MATLAB at a sampling rate of 16 kHz. We implemented a 6-band system using a set of BPFs with non-uniform bandwidth whose center frequencies are 250, 500, 1000, 2000, 4000, and 6000 Hz, respectively. Frames of 128 samples with 50% overlap were utilized. The Hann function was applied for windowing. 25 male and 25 female speech signals from TIMIT database were used for simulations.

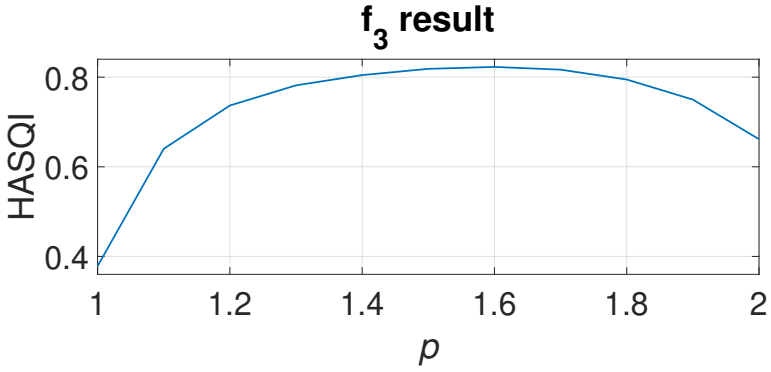
The PEM framework in Figure 4.2 is again considered. We study freping with $\boldsymbol{\alpha} = \alpha[0, 0, 0, 0, 1, 1]^T$ on top of the LMS (4.1) and the SLMS (4.5). As instability is most likely to occur at the high-frequency region in HAs, it is reasonable to manipulate the high-frequency



(a)



(b)



(c)

Figure 4.8: Effect of p on speech quality of SLMS for (a) f_1, (b) f_2, and (c) f_3.

content while keeping the low-frequency region intact to avoid degradation in quality. Based on informal subjective assessments, distortions due to freping are fairly benign. The experimental setup was as follows. The HA processing $G_n(z) = gz^{-\tau}$ where g is the HA gain and τ is the

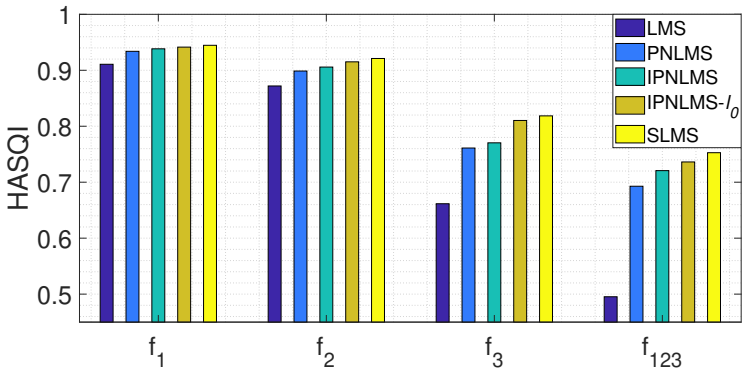


Figure 4.9: Comparison of speech quality for different feedback environments. The first three cases were fixed environments with f_1, f_2, and f_3. The last case f_{123} was the feedback path changing from f_1 to f_2 then f_3 at 1/3 and 2/3 of the input sequence, respectively.

sample delay chosen to have a total HA latency under 10 msec (from d_n to o_n). The feedback path IR of Figure 4.7 (c) was considered. For the AFC, we used $M = 100$, $\mu = 0.005$, $\rho = 0.985$, and $\delta = 10^{-6}$ for both LMS and SLMS. For the SLMS we used $p = 1.5$ and $c = 10^{-6}$ for (4.7). In all simulations, the AFC filter coefficients were initialized as all zeros.

Figure 4.10 presents example spectrograms of the feedback-compensated signal for several cases. We can see that freping effectively reduces the howling components present in the red boxes, resulting in improved quality.

We compare performance with an existing FS method based on the analytical representation of signal using the Hilbert transform [51, 102]. The amount of shift was set to 12 Hz, only applied to frequency region above 1.5 kHz as suggested by [103, 104]. Figure 4.11 demonstrates advantage of using freping by showing the average HASQI score over the 50 speech files for various gain settings. From the results we see that both the basic (LMS) and advanced (SLMS) AFC algorithms can benefit from freping. This indicates the ability of the proposed frequency warping method to further improve feedback reduction on top of many AFC approaches. Moreover, compared to the FS, freping demonstrates better performance under all the gain settings.

Finally, we compare the ASG for the cases of AFC, AFC with FS, and AFC with freping using the proposed HASQI-based ASG estimation approach, where a HASQI below $\theta = 0.8$ is

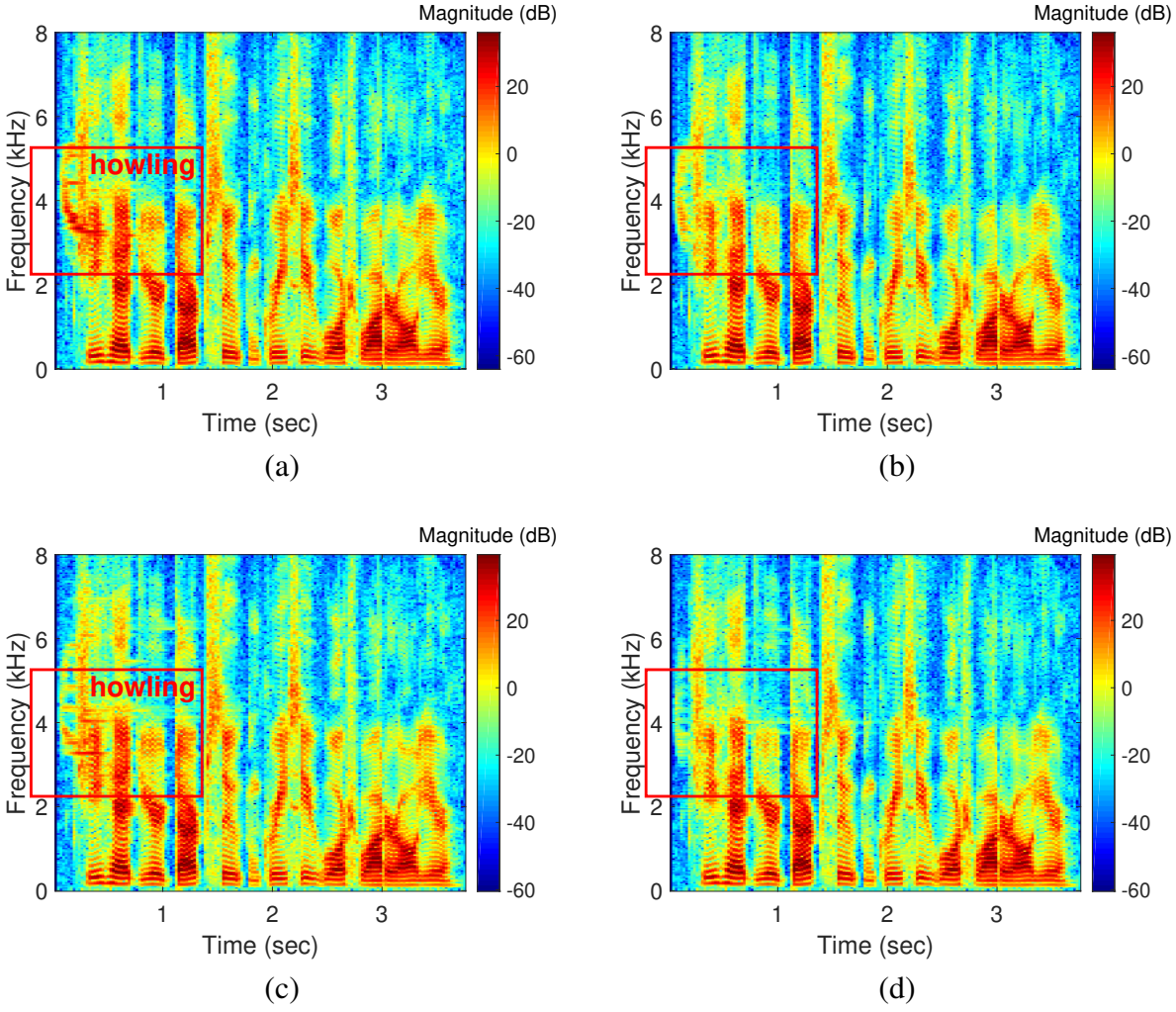


Figure 4.10: Spectrograms of feedback-compensated signal. The top row is for LMS with HA gain at 20 and the bottom row is for SLMS with HA gain at 30. Freping is disabled in the left column and enabled with $\alpha = -0.02$ in the right column. The HASQI scores are (a) 0.81, (b) 0.84, (c) 0.79, and (d) 0.82.

considered of unacceptable quality. The results are shown in Table 4.2, obtained from the average of 5 male and 5 female speech files. We can see that freping can improve the ASG on top of both the basic and advanced AFC algorithms. Compared to the FS, a higher ASG can be achieved by using freping.

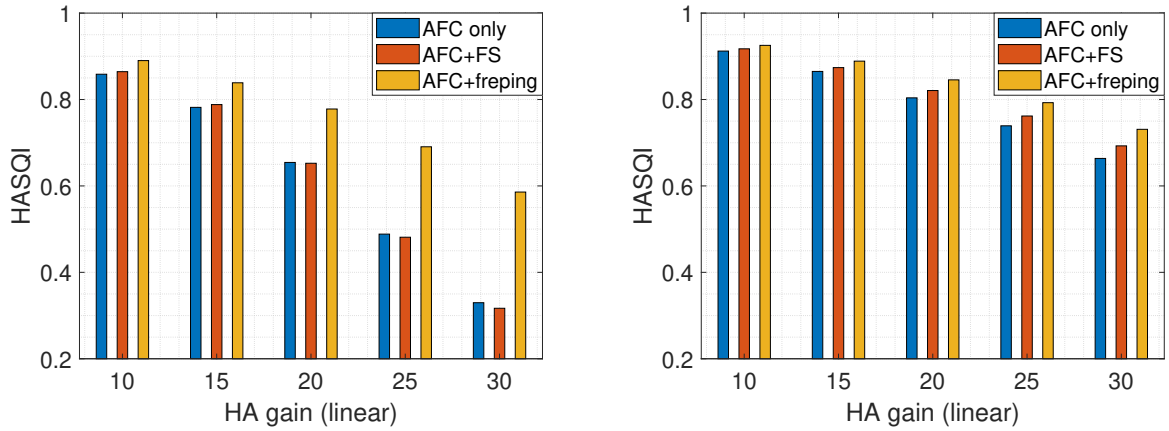


Figure 4.11: HASQI of feedback-compensated signal for AFC using (a) LMS and (b) SLMS. With freping, HASQI improvements of 0.65 to 0.78 and 0.66 to 0.73 can be seen for LMS with HA gain at 20 and SLMS with HA gain at 30, respectively.

Table 4.2: ASG (in dB) comparison.

AFC algorithm	AFC only	AFC+FS	AFC+freping
LMS	14.41	15.05	**16.90**
SLMS	17.87	18.47	**19.31**

4.7 Conlusion

In this chapter, we introduced the acoustic feedback problem associated with HAs, and applied the SLMS algorithm to improve AFC by leveraging the (quasi-) sparse structure of feedback path IRs. The SLMS has been shown to provide higher speech quality and stable gain compared to LMS in AFC. We further introduced freping, a frequency warping method that utilizes all-pass networks to decorrelate the signal for better feedback control with only negligible distortion incurred.

Acknowledgment

Chapter 4 is, in part, a reprint of the material as it appears in the three papers: C.-H. Lee, K.-L. Chen, f. harris, B. D. Rao, and H. Garudadri, "On mitigating acoustic feedback in hearing aids with frequency warping by all-pass networks," in *20th Annual Conference of the International Speech Communication Association (Interspeech)*, 2019; C.-H. Lee, J. M. Kates, B. D. Rao, and H. Garudadri, "Speech quality and stable gain trade-offs in adaptive feedback cancellation for hearing aids," *Journal of the Acoustical Society of America Express Letters*, 2017; and C.-H. Lee, B. D. Rao, and H. Garudadri, "Sparsity promoting LMS for adaptive feedback cancellation," in *25th European Signal Processing Conference (EUSIPCO)*, 2017. The dissertation author was the primary investigator and author of these papers. The work, in part, was supported by National Institutes of Health/National Institute on Deafness and Other Communication Disorders under Grants R01DC015436, R21DC015046, and R33DC015046.

Chapter 5

Weighted Gradient Descent Algorithms for Learning Regularized Models with Applications to Nonlinear Model Sparsification

In this chapter, we show that weighted gradient descent algorithms introduce implicit weighted norm regularization, which can be exploited for learning regularized models without having to deal with the complex and often overlooked task of selecting a weight for the regularization penalty. Specifically, we study a reparameterization framework that leads to learning algorithms wherein it is possible to set the regularization penalty to zero in a limiting manner, thereby minimizing the original unpenalized objective function. However, the resulting weighted gradient algorithm is able to capture the regularization information through the weighting matrix which can be iteration dependent in an implicit manner. The form of the matrix depends on the type of regularization considered, e.g., parameter sparsity, total variation, model complexity, etc. The

framework is thus general and it enables searching for solutions with some desirable properties without incorporating a regularization penalty. As a main application of the framework, we propose novel sparsity-promoting algorithms beneficial for i) deep neural network compression and ii) dictionary pruning in kernel methods. Simulation results are presented to demonstrate that the proposed algorithms find sparse models without incurring a regularization bias, and can be useful for learning compact representations in many nonlinear estimation problems.

5.1 Introduction

Many signal processing and machine learning problems consider the empirical risk minimization (ERM):

$$\min_{\boldsymbol{\theta}} \quad J(\boldsymbol{\theta}) = \frac{1}{N} \sum_{n=0}^{N-1} L\left(h(\mathbf{x}_n; \boldsymbol{\theta}), \mathbf{y}_n\right), \tag{5.1}$$

where $\{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=0}^{N-1}$ is a training dataset of N input-output pairs, $\boldsymbol{\theta}$ is the parameter set of the model hypothesis $h(\cdot; \boldsymbol{\theta})$, $L(\cdot, \cdot)$ is the loss function, and $J(\boldsymbol{\theta})$ is called the empirical risk. Recently, *overparameterized* models that are equipped with many more parameters than statistically needed have become widely seen in practice, either due to the lack of guidance on the model size [128, 129], or because of potentially better generalization capabilities [130, 131, 132]. Overparameterization, in turn, leads to the problem having multiple solutions $\boldsymbol{\theta}$ that result in the same optimum. The question is how to approach, if not select, a particular solution exhibiting desirable properties among the many others. In this chapter, we study a novel algorithmic framework for developing learning algorithms that optimize (5.1) while implicitly finding regularized models that exhibit the desired properties.

To obtain a "good" solution, regularization techniques can be suitably employed by supplementing the objective function with an additional penalty as: $\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) + \lambda R(\boldsymbol{\theta})$, where $R(\cdot)$ is a regularizer function that outputs a scalar and $\lambda > 0$ is the regularization coefficient (weight). The regularizer embeds our prior knowledge about how a "good" solution should be like.

For example, when $\boldsymbol{\theta}$ is known to be sparse, then the regularizer can use some diversity measures of $\boldsymbol{\theta}$ to promote sparsity [14, 42, 15, 16]; if a simple model is desired, then the regularizer can use some capacity (complexity) measures [133, 134, 135, 136, 137]. The coefficient λ controls the trade-off between model fitting and the regularization penalty. However, the selection of λ is often non-trivial. Although regularization helps in restricting the search space for a desired solution, it often comes at the expense of model fitting due to *explicitly* using the regularizer as a penalty term – the solution does not minimize the original objective function but the penalized objective function which incurs a regularization bias.

Recently, there has been an increasing interest in studying *implicit* regularization of the gradient descent family of algorithms [35, 36, 37, 38, 39, 40, 138, 139, 41]. By "implicit" we mean that the learning algorithm still solves the unpenalized problem (5.1) while being aware of the designated regularizer, leading towards a specific minimizer if there are many [36]. For example, the gradient descent algorithm is known to be associated with the ℓ_2 norm regularization [37]. Despite a well-known result for linear regression, it is only recently that implicit regularization of gradient descent has received considerable attention for more general problems. For example in deep learning, due to the surprising observation that many highly overparameterized models learned by gradient-based algorithms do not overfit even without explicit regularization [130], studying implicit regularization properties for explaining generalizability has become an active research topic [35, 36, 37, 38, 39, 40, 41]. On the other hand, implicit regularization has recently been studied for sparse signal recovery problems in [138, 139], which have observed several advantages in using implicit over explicit regularization for incorporating sparsity.[1]

In this chapter, we follow this line of research to explore implicit regularization aspects of gradient-based algorithms. However, different from most of the prior work that focuses on generalization capabilities or computational advantages of complex models, we aim to get rid

[1] However, *early stopping* is required in [138, 139] for obtaining the regularized (sparse) models. This could be a potential limitation given that in the overparameterized regime, "double-descent" risk curves have been observed [140], which in turn would suggest a longer training time for achieving even better performance.

of the biasing effect that arises from explicit regularization schemes via a properly designed algorithmic framework. The main contributions of this work are as follows:

1. We propose a reparameterization scheme based on an affine scaling transformation (AST) methodology [42, 60] that enables setting λ to zero for the regularizer in a limiting manner, leading to learning algorithms that estimate regularized models without having to deal with the complex and often overlooked task of selecting a weight for the regularizer.[2] The algorithms in general have a weighted gradient term in the update equation, where the weighting matrix can be iteration dependent and the form of the matrix depends on the type of regularization considered.

2. We present design options of the weighting matrix with regularizers for parameter sparsity (and group sparsity), total variation, and model complexity to demonstrate flexibility of the framework. In this sense, we associate weighted gradient learning algorithms with implicit weighted norm regularization. In overparameterized models, the framework enables searching for solutions with some desirable properties without having to incorporate a regularization penalty, and thus can be useful to modern signal processing and machine learning systems utilizing large models.

3. Based on iterative reweighting techniques [13] popular in the sparse signal recovery (SSR) area, we develop sparsity-promoting weighted gradient algorithms for learning compact representations of nonlinear models. Specifically, we propose i) Sparsity-promoting Stochastic Gradient Descent (SSGD) algorithm for neural network compression and ii) Sparsity-promoting Kernel Least Mean Square (SKLMS) and Sparsity-promoting Kernel Normalized Least Mean Square (SKNLMS) algorithms for dictionary pruning in kernel methods.

[2]Our approach of advocating implicit regularization ($\lambda = 0$) is based on the recent observation that, in the overparameterized regime (e.g., in many deep networks), the learned models do not overfit even without explicit regularization ($\lambda > 0$), e.g., in [35, 36, 37, 38, 39, 40, 41]. In addition, implicit regularization has bee observed to be more advantageous than explicit regularization in some cases, e.g., [138, 139].

4. Simulation results are provided to demonstrate the capabilities of SSGD, SKLMS, and SKNLMS of learning sparse models, with sparsification performance compared to explicit regularization techniques.

Organization of the Chapter: The rest of the chapter is organized as follows. Section 5.2 discusses implicit regularization properties associated with gradient descent and weighted gradient descent in the linear regression setting. Section 5.3 presents a more generic framework of weighted gradient algorithms for the general ERM setting (5.1), taking into account an iteration dependent weighting matrix. Section 5.4 studies implicit sparsity regularization under the weighted gradient algorithmic framework, by utilizing popular reweighting techniques in SSR together with an AST methodology. Section 5.5 introduces sparsity-promoting weighted gradient algorithms for stochastic optimization of neural networks and online learning of kernel methods. Section 5.6 presents simulation results. Section 5.7 further discusses a complexity regularization example as an extension of the framework. Section 5.8 concludes the chapter.

5.2 Gradient Descent Algorithms for Linear Regression with Weighted Norm Regularization

Let us first consider for (5.1) the linear regression setting with N training data pairs $(\mathbf{x}_n, y_n) \in \mathbb{R}^M \times \mathbb{R}$, $n = 0, 1, \cdots, N-1$, parameter set $\boldsymbol{\theta} \in \mathbb{R}^M$, and model $h(\mathbf{x}_n; \boldsymbol{\theta}) = \mathbf{x}_n^T \boldsymbol{\theta}$. Assuming the squared loss $L(a, b) = \frac{1}{2}(a-b)^2$, for $a, b \in \mathbb{R}$, this amounts to the least squares (LS) problem:

$$\min_{\boldsymbol{\theta}} \quad J(\boldsymbol{\theta}) = \frac{1}{2N} \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|_2^2, \tag{5.2}$$

where $\mathbf{y} = [y_0, y_1, \ldots, y_{N-1}]^T \in \mathbb{R}^N$ is the measurement vector and $\mathbf{X} = [\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{N-1}]^T \in \mathbb{R}^{N \times M}$ is the input data matrix. We further assume $N < M$ and $\text{rank}(\mathbf{X}) = N$, resulting in (5.2) being an underdetermined system of equations having infinitely many solutions.

Consider a regularizer for (5.2) interpreted as the weighted ℓ_2 norm in terms of some invertible matrix $\mathbf{W}$:

$$\min_{\boldsymbol{\theta}} \quad J(\boldsymbol{\theta}) + \lambda \|\mathbf{W}^{-1}\boldsymbol{\theta}\|_2^2, \tag{5.3}$$

and $\mathbf{W}$ characterizes our prior knowledge about the desired properties of the solution. One example is total variation-like regularization which uses $\mathbf{W}^{-1} = \mathbf{D}$, where $\mathbf{D}$ is a finite-difference operator [141]. Note that for $\lambda > 0$, (5.3) is strictly convex in $\boldsymbol{\theta}$ and thus it has a unique minimizer $\boldsymbol{\theta}^*$.

In the limiting case as $\lambda \to 0^+$, the solution attempts to do data fitting with an eye towards seeking a regularized solution with minimal compromise. However, the problem can become highly ill-conditioned; optimization algorithms like gradient descent may get "stuck" and "never" achieves the (unique) limiting solution. Indeed, these converged solutions are potential global minima once λ becomes zero, which is when the problem has infinitely many solutions. In such a case, the optimization outcome may depend on the specific algorithm used to obtain the estimate. To see this, notice that when using the gradient descent scheme for optimizing (5.3), setting $\lambda = 0$ for the resulting update equation basically reduces to the gradient descent algorithm of the unconstrained data fitting problem (5.2):

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_t), \tag{5.4}$$

where t is the timestep and $\eta_t > 0$ is the learning rate. The following observations characterize the convergence of (5.4):

Proposition 5.1. *For the LS problem (5.2), starting from an initial $\boldsymbol{\theta}_0$ and with sufficiently small learning rates η_t, the gradient descent algorithm (5.4) converges to: $\boldsymbol{\theta}^*_{\text{gd}} = \boldsymbol{\theta}_{\min} + \mathcal{P}_{\mathcal{N}(\mathbf{X})}(\boldsymbol{\theta}_0)$, where $\boldsymbol{\theta}_{\min} = \arg\min_{\boldsymbol{\theta}} \|\boldsymbol{\theta}\|_2^2$ s.t. $\mathbf{y} = \mathbf{X}\boldsymbol{\theta}$ is the minimum ℓ_2 norm solution and $\mathcal{P}_{\mathcal{N}(\mathbf{X})}$ denotes the projector onto $\mathcal{N}(\mathbf{X})$.*

Proof: See Appendix 5.9.1.

Corollary 5.1. *Assuming the same conditions of Proposition 5.1, the gradient descent algorithm (5.4) converges to the minimum ℓ_2 norm solution $\boldsymbol{\theta}_{\min}$ if and only if $\boldsymbol{\theta}_0 \in \mathcal{R}(\mathbf{X}^T)$.*

Proof: See Appendix 5.9.2.

Remark 5.1. *Assuming the same conditions of Proposition 5.1, if $\boldsymbol{\theta}_0 \notin \mathcal{R}(\mathbf{X}^T)$ is close to the origin, we have $\boldsymbol{\theta}^*_{\text{gd}} \approx \boldsymbol{\theta}_{\min}$. This can simply be seen by noting the fact that $\boldsymbol{\theta}^*_{\text{gd}} = \arg\min_{\boldsymbol{\theta}} \|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|_2^2$ s.t. $\mathbf{y} = \mathbf{X}\boldsymbol{\theta}$.*

The above well-known results characterize the "implicit" regularization property of gradient descent associated with ℓ_2 norm: *without explicitly using it as a penalty, the gradient descent (5.4) (with proper initialization) finds the smallest ℓ_2 norm solution (or something close to it).*

Now consider the reparameterization in terms of the (affinely) transformed variable: $\mathbf{q} \triangleq \mathbf{W}^{-1}\boldsymbol{\theta}$. Reparameterizng the objective function in (5.3), the problem is equivalent to first solving:

$$\min_{\mathbf{q}} \quad J(\mathbf{W}\mathbf{q}) + \lambda \|\mathbf{q}\|_2^2 \tag{5.5}$$

and then obtaining the solution as $\boldsymbol{\theta}^* = \mathbf{W}\mathbf{q}^*$, where $\mathbf{q}^*$ is the minimizer of (5.5).

We may also optimize (5.5) by using gradient descent w.r.t. $\mathbf{q}$. Since we are interested in the limiting case, we set $\lambda = 0$ for the resulting gradient update equation, leading to:

$$\mathbf{q}_{t+1} = \mathbf{q}_t - \eta_t \nabla_{\mathbf{q}} J(\mathbf{W}\mathbf{q}_t). \tag{5.6}$$

Using the chain rule[3] and noting the relationship between $\mathbf{q}$ and $\boldsymbol{\theta}$, it can be shown that the

[3] This is basically $\nabla_{\mathbf{q}} = \mathbf{W}^T \nabla_{\boldsymbol{\theta}}$.

equivalent update form of (5.6) in the $\boldsymbol{\theta}$ domain is:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \mathbf{W}\mathbf{W}^T \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_t). \tag{5.7}$$

Comparing (5.7) to (5.4), we see that an additional weighting term $\mathbf{W}\mathbf{W}^T$ is introduced before the gradient of the ordinary objective $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_t)$. Adopting $\lambda = 0$, the weighted gradient descent algorithm (5.7) also optimizes the unpenalized problem (5.2) just like (5.4). However, the path it takes is different and may eventually converge to a different solution. As noted in [79], the performance of gradient-based methods is dependent on the parameterization – a new choice may substantially alter convergence characteristics. In the case of multiple solutions here, weighted gradients in (5.7) may push the algorithm toward a specific one. Indeed, we have the following results for (5.7):

Proposition 5.2. *For the LS problem (5.2), starting from an initial $\boldsymbol{\theta}_0$ and with sufficiently small learning rates η_t, the weighted gradient descent algorithm (5.7) converges to: $\boldsymbol{\theta}^*_{\text{wgd}} = \boldsymbol{\theta}_{\text{wmin}} + \mathbf{W}\mathcal{P}_{\mathcal{N}(\mathbf{X}\mathbf{W})}(\mathbf{W}^{-1}\boldsymbol{\theta}_0)$, where $\boldsymbol{\theta}_{\text{wmin}} = \arg\min_{\boldsymbol{\theta}} \|\mathbf{W}^{-1}\boldsymbol{\theta}\|_2^2$ s.t. $\mathbf{y} = \mathbf{X}\boldsymbol{\theta}$ is the minimum weighted ℓ_2 norm solution and $\mathcal{P}_{\mathcal{N}(\mathbf{X}\mathbf{W})}$ denotes the projector onto $\mathcal{N}(\mathbf{X}\mathbf{W})$.*

Proof: See Appendix 5.9.3.

Corollary 5.2. *Assuming the same conditions of Proposition 5.2, the weighted gradient descent (5.7) converges to the minimum weighted ℓ_2 norm solution $\boldsymbol{\theta}_{\text{wmin}}$ if and only if $\boldsymbol{\theta}_0 \in \mathcal{R}(\mathbf{W}\mathbf{W}^T\mathbf{X}^T)$.*

Proof: See Appendix 5.9.4.

Remark 5.2. *Assuming the same conditions of Proposition 5.2, if $\boldsymbol{\theta}_0 \notin \mathcal{R}(\mathbf{W}\mathbf{W}^T\mathbf{X}^T)$ is close to the origin, we have $\boldsymbol{\theta}^*_{\text{wgd}} \approx \boldsymbol{\theta}_{\text{wmin}}$. This can be simply seen by the fact that $\boldsymbol{\theta}^*_{\text{wgd}} = \arg\min_{\boldsymbol{\theta}} \|\mathbf{W}^{-1}(\boldsymbol{\theta} - \boldsymbol{\theta}_0)\|_2^2$ s.t. $\mathbf{y} = \mathbf{X}\boldsymbol{\theta}$.*

The above observations indicate that *weighted gradient descent (5.7) is implicitly associated with weighted ℓ_2 norm.* This is a natural extension of the well-known relation between gradient descent and ℓ_2 norm. To the best of our knowledge, we have not seen previous work that explicitly characterizes this property. Moreover, little attention has been paid on taking advantage of such implicit regularization, though weighted gradient algorithms are so simple and commonly seen in practice. The main contribution in this work is a generalization of this idea: by introducing a suitable $\mathbf{W}$ (maybe iteration dependent) that potentially characterizes the desired properties of the solution, we show that via weighted gradient algorithms it is possible to achieve a desirable solution (specified by $\mathbf{W}$) without having to penalize the problem (using a zero λ).

5.3 Weighted Gradient Learning Algorithms for Estimating Regularized Models

We aim to broaden the scope of the framework in Section 5.2 by considering the general ERM problem (5.1) and thereby going beyond the simple linear regression setting. Moreover, for the weighted ℓ_2 norm regularizer, we take into account an iteration dependent $\mathbf{W}_t$ for each timestep t, which is typically a function of the estimated parameters $\boldsymbol{\theta}_t$, instead of a fixed $\mathbf{W}$. Making it an iterative process, we allow $\mathbf{W}_t$ to also be refined as $\boldsymbol{\theta}_t$ gets updated. It is helpful in practice as a good $\mathbf{W}$ may not be trivial to find in advance. We further assume $\mathbf{W}_t$ symmetric positive definite for simplicity, and often a diagonal matrix for pragmatic reasons [79].

Now we have the penalized ERM problem at iteration t as:

$$\min_{\boldsymbol{\theta}} \quad J(\boldsymbol{\theta}) + \lambda \|\mathbf{W}_t^{-1}\boldsymbol{\theta}\|_2^2. \tag{5.8}$$

Based on the discussion in Section 5.2, (5.8) naturally leads to the following reparameterization

in terms of the (affinely) scaled variable:

$$\mathbf{q} \triangleq \mathbf{W}_t^{-1}\boldsymbol{\theta}. \tag{5.9}$$

Note that when applying (5.9), in each iteration the $\mathbf{W}_t$ is pre-calculated based on the estimate $\boldsymbol{\theta}_t$ and treated as a given matrix to perform a change of coordinates (variables) [78] from $\boldsymbol{\theta}$ to $\mathbf{q}$, acting as a scaling technique in gradient descent methods [79]. Viewing $\mathbf{W}_t$ as the *scaling matrix*, (5.9) can be interpreted as the AST commonly employed by interior point methods for solving linear and nonlinear programming problems [42]. In the optimization literature, AST-based methods transform the original problem into an equivalent one, favorably positioning the current point at the center of the feasible region to facilitate the optimization [60]. Typically, optimization problems concerned with the AST usually admit a unique solution and the main purpose is to speed up adaptation toward that particular solution. Differently, here we recognize the usage of AST for problems with multiple solutions with a different purpose: to guide the learning process toward a more desirable solution.

Following the discussion in Section 5.2, we reparameterize the penalized ERM objective in (5.8) using the AST (5.9), leading to the equivalent problem in the $\mathbf{q}$ domain:

$$\min_{\mathbf{q}} \quad J(\mathbf{W}_t\mathbf{q}) + \lambda\|\mathbf{q}\|_2^2. \tag{5.10}$$

A gradient descent procedure will then be applied. The overall update process conceptually can be summarized as follows: i) given a $\boldsymbol{\theta}$ compute $\mathbf{W}_t$ followed by reparameterization $\mathbf{q}$ as (5.9). ii) Update $\mathbf{q}$ using a gradient descent algorithm. iii) Use this new $\mathbf{q}$ to obtain the updated $\boldsymbol{\theta}$. iv) Repeat Steps i)–iii) till convergence.

More formally, to proceed with gradient-based updates, following [49] we define the *a*

posteriori AST variable at iteration t:

$$\mathbf{q}_{t|t} \triangleq \mathbf{W}_t^{-1}\boldsymbol{\theta}_t \tag{5.11}$$

and the *a priori* AST variable at iteration t:

$$\mathbf{q}_{t+1|t} \triangleq \mathbf{W}_t^{-1}\boldsymbol{\theta}_{t+1}. \tag{5.12}$$

The recursive update by using gradient descent for (5.10) in the $\mathbf{q}$ domain can be formulated in terms of the two AST variables as:

$$\mathbf{q}_{t+1|t} = \mathbf{q}_{t|t} - \eta_t \left(\nabla_{\mathbf{q}} J(\mathbf{W}_t \mathbf{q}_{t|t}) - 2\lambda \mathbf{q}_{t|t} \right). \tag{5.13}$$

Setting $\lambda = 0$ for (5.13) yields:

$$\mathbf{q}_{t+1|t} = \mathbf{q}_{t|t} - \eta_t \nabla_{\mathbf{q}} J(\mathbf{W}_t \mathbf{q}_{t|t}). \tag{5.14}$$

Using the chain rule[4] and the AST relationships (5.9) and (5.11), we can write (5.14) as:

$$\mathbf{q}_{t+1|t} = \mathbf{q}_{t|t} - \eta_t \mathbf{W}_t \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_t). \tag{5.15}$$

Premultiplying $\mathbf{W}_t$ on both sides of (5.15) and noting the relationships (5.11) and (5.12), we can transform the $\mathbf{q}$ domain update (5.15) back to the $\boldsymbol{\theta}$ domain as:

$$\boxed{\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \mathbf{W}_t^2 \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_t).} \tag{5.16}$$

This *weighted gradient algorithm can potentially learn regularized models ($\lambda \to 0^+$) without*

[4]Note that the chain rule here is basically $\nabla_{\mathbf{q}} = \mathbf{W}_t \nabla_{\boldsymbol{\theta}}$ as a result of the change of variables (5.9) for a given $\mathbf{W}_t$ at iteration t.

incurring a regularization penalty – even though adopting $\lambda = 0$, it still exhibits regularization properties through the *weighting matrix* $\mathbf{W}_t^2$ and thus exploits prior information. This is also revealed by our argument regarding implicit regularization of weighted gradient descent in Section 5.2, but now with a changing weighting matrix. However, with such a time-varying matrix analysis of convergence becomes nontrivial. Nevertheless, we have the following theorem that sheds light on the convergence of (5.16):

Theorem 5.1. *For the ERM problem (5.1), there exists a learning rate sequence* $\{\eta_t\}_{t=0}^{\infty}$ *such that, using* η_t *at iteration t, (5.16) monotonically converges to a local minimum (or saddle point) of the empirical risk objective.*

Proof: See Appendix 5.9.5.

Note that Theorem 5.1 applies not only to linear regression but the general problem setting in (5.1). In addition, even though with an iteration dependent $\mathbf{W}_t$, Theorem 5.1 indicates that (5.16) still solves (5.1). This is actually not surprising, since (5.16) optimizes (5.10) with a zero λ, meaning that it solves $\min_{\mathbf{q}} J(\mathbf{W}_t\mathbf{q})$ which is an equivalent problem to (5.1) given that $\mathbf{W}_t$ is invertible.

It is also worth mentioning the similarity of (5.16) with preconditioning methods [60, 79] by viewing the weighting matrix as the preconditioner. Traditionally, the preconditioner is fixed though all iterations and its role is mainly to reduce the condition number for convergence speed-up. Whereas in (5.16), the matrix is iteration dependent so that it adapts as time evolves, and its role is to help approach a desired solution. In this sense, algorithms like (5.16) can also be viewed as using an adaptive preconditioner $\mathbf{W}_t^2$ [142]. However, compared to the preconditioning viewpoint, our reparameterization framework better incorporates the implicit regularization property of the weighting matrix for flexible class of regularizers, thereby adding additional insights into the algorithms.

5.3.1 Practical Considerations for Constant Learning Rate

In some practical scenarios, a constant $\eta_t = \eta$ may be preferably used to avoid potentially expensive evaluation of the "best" learning rate by, e.g., line search methods. When this is the case, if the weighting matrix $\mathbf{W}_t^2$ scales arbitrarily, a fixed η may not be able to compensate for the scaling. This, in turn, may lead to instability (scaling too large) or slow convergence (scaling too small) issues as gradient-based algorithms are sensitive to the learning rate. To remedy it, in practice we can normalize the weighting matrix by some scalar α_t to compensate for arbitrary scalings, which corresponds to dividing the learning rate by α_t. Empirically, for example, for a diagonal weighting matrix $\mathbf{W}_t^2$ it is useful to normalize it such that the diagonal elements sum up to the number of parameters M, e.g., using $\alpha_t = \frac{1}{M}\sum_{i=0}^{M-1} w_{i,t}^2$. In other words, the following update rule is suggested over (5.16) when using a fixed η:

$$\boxed{\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_n - \eta \mathbf{S}_t \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_t),} \tag{5.17}$$

where

$$\boxed{\mathbf{S}_t = \frac{\mathbf{W}_t^2}{\frac{1}{M}\mathrm{tr}\left(\mathbf{W}_t^2\right)},} \tag{5.18}$$

is the normalized version of a diagonal $\mathbf{W}_t^2$. Later, we will see slightly modified $\mathbf{S}_t$ matrices for accommodating different algorithms as we propose.

5.4 Implicit Sparsity Regularization via Weighted Gradient Learning Algorithms

Sparsity has been an important attribute in many successful signal processing and machine learning systems especially for the last two decades. We study weighted gradient algorithms for

such aspects. To begin, we start with a sparsity-inducing penalty regularized ERM problem:

$$\min_{\boldsymbol{\theta}} \quad J(\boldsymbol{\theta}) + \lambda G(\boldsymbol{\theta}), \tag{5.19}$$

where $G(\cdot)$ represents the *general diversity measure* in SSR that when minimized encourages sparsity in its argument. For $\boldsymbol{\theta} \in \mathbb{R}^M$, a *separable* form is commonly used: $G(\boldsymbol{\theta}) = \sum_{i=0}^{M-1} g(\theta_i)$, where $g(\cdot)$ has the following properties [13]:

Property 1: $g(z)$ is symmetric, i.e., $g(z) = g(-z) = g(|z|)$;

Property 2: $g(|z|)$ is monotonically increasing with $|z|$;

Property 3: $g(0)$ is finite;

Property 4: $g(z)$ is concave in $|z|$ or z^2.

Any function that holds the above properties is a candidate for effective SSR algorithm development. One popular example is the *p-norm-like* diversity measure [14, 42] with $g(\cdot) = |\cdot|^p$, $p \in (0, 2]$, i.e., $G(\boldsymbol{\theta}) = \sum_{i=0}^{M-1} |\theta_i|^p$, which is associated with super-Gaussian prior distributions. In general, a smaller p corresponds to a heavier-tailed distribution, encouraging stronger sparsity in the parameters.

5.4.1 Iterative Reweighting Algorithms for SSR

The concave nature of the diversity measure penalty $G(\boldsymbol{\theta})$ poses challenges to the optimization of (5.19). For solving it, many algorithms rely on iterative reweighting schemes that produce more focal estimates as optimization progresses [13]. One popular method is the iterative reweighted ℓ_2 algorithm [14, 15] that introduces a weighted ℓ_2 norm term as an upper bound for $G(\boldsymbol{\theta})$ per iteration. Specifically, it suggests sequentially solving a series of weighted ℓ_2 penalized problems of (5.8), where in each iteration $\mathbf{W}_t = \text{diag}\{w_{i,t}\}$ is positive definite[5] and each

[5]The positive definiteness can be shown to hold for a wide variety of diversity measures used in SSR. In cases where it is not, the positive definiteness can still be ensured by utilizing some small regularization constant.

$w_{i,t}$ is computed based on $\theta_{i,t}$ as [13]:

$$w_{i,t} = \left(\frac{\mathrm{d}f(z)}{\mathrm{d}z}\bigg|_{z=\theta_{i,t}^2} \right)^{-\frac{1}{2}}, \tag{5.20}$$

where $g(z) = f(z^2)$ and $f(z)$ has to be concave for $z \in \mathbb{R}_+$, i.e., $g(z)$ is concave in z^2, for Property 4.

For example, choosing the p-norm-like diversity measure with $p \in (0, 2]$ and using (5.20) results in:

$$w_{i,t} = \left(\frac{2}{p} \left(|\theta_{i,t}| + c \right)^{2-p} \right)^{\frac{1}{2}}. \tag{5.21}$$

Note that a small regularization constant $c > 0$ is empirically added for avoiding algorithm stagnation and instability, which also ensures the nonsigularity of $\mathbf{W}_t$.

The reweighting method is actually based on the majorization-minimization (MM) framework [17], as the objective function in (5.8) serves as a *majorizer* of the objective function in (5.19) for every iteration. To approach a solution, one can sequentially minimize the majorizers as typically done in conventional SSR; or more generally, utilize gradient update schemes[6] since exact minimization of (5.8) may not be trivial.

5.4.2 Sparsity-Promoting Weighted Gradient Algorithm

For our discussion concerning the gradient update scheme, the reweighted ℓ_2 framework naturally suggests using the $\mathbf{W}_t$ given by (5.20) for the AST reparameterization (5.9), resulting in the weighted gradient algorithm (5.16) demonstrating sparsity-promoting characteristics.

To further see that (5.16) potentially converges toward a sparse solution with (5.20), let us again consider the LS problem setting of (5.2). According to Proposition 5.2, we can view (5.16) as aiming for the weighted ℓ_2 norm minimization: $\min_{\boldsymbol{\theta}} \|\mathbf{W}_t^{-1}\boldsymbol{\theta}\|_2^2$ s.t. $\mathbf{y} = \mathbf{X}\boldsymbol{\theta}$. From the

[6] It corresponds to the *generalized* MM [77] where one does not need to minimize the majorizer but only to assure that it decreases in every iteration.

MM viewpoint, sequentially solving the weighted ℓ_2 norm minimization problems approaches a solution to the diversity measure minimization: $\min_{\boldsymbol{\theta}} G(\boldsymbol{\theta})$ s.t. $\mathbf{y} = \mathbf{X}\boldsymbol{\theta}$. This reveals the intention of (5.16) to promote sparse solutions. Note that it is not to guarantee that (5.16) converges to a solution of the diversity measure minimization problem, since the minimum weighted ℓ_2 norm solution might not be exactly found in each iteration as we only carry out a gradient descent update. Nevertheless, this connection hints at the sparsity-promoting characteristics of (5.16) with (5.20) and will be supported by simulation results later in Section 5.6.

In terms of the general ERM setting (5.1), given the complication of the problem it is nontrivial to characterize the properties of the solution found by (5.16). Currently, the best we can do is really to "expect" the solution to be implicitly regularized. However, our empirical studies in Section 5.6 show that sparsity-promoting algorithms like (5.16) are capable of finding regularized models in practical problems.

5.4.3 Extensions

Incorporating Reweighted ℓ_1 Framework

In addition to the reweighted ℓ_2 framework, another approach popular in SSR is the reweighted ℓ_1 algorithm proposed in [16], which introduces a weighted ℓ_1 norm term as the majorizer for $G(\boldsymbol{\theta})$. In other words, it suggests that (5.19) be iteratively approached where in each iteration we solve:

$$\min_{\boldsymbol{\theta}} \quad J(\boldsymbol{\theta}) + \lambda \|\mathbf{W}_t^{-1}\boldsymbol{\theta}\|_1, \tag{5.22}$$

where the matrix $\mathbf{W}_t$ is now given as [13]:

$$w_{i,t} = \left(\left. \frac{\mathrm{d}f(z)}{\mathrm{d}z} \right|_{z=|\theta_{i,t}|} \right)^{-1}, \tag{5.23}$$

where $g(z) = f(|z|)$ and $f(z)$ has to be concave for $z \in \mathbb{R}_+$, i.e., $g(z)$ is concave in $|z|$, for Property 4.

As an example, the p-norm-like function can also be adopted in the reweighted ℓ_1 framework if $p \in (0, 1]$. Using (5.23), in this case we have:

$$w_{i,t} = \frac{1}{p}\left(|\theta_{i,t}| + c\right)^{1-p}, \tag{5.24}$$

where a small regularization constant $c > 0$ is again utilized.

Adopting the AST (5.9) with the $\mathbf{W}_t$ given by (5.23) for the reweighted ℓ_1 framework (5.22), one can also perform $\mathbf{q}$ domain gradient descent as we have done in the reweighted ℓ_2 case. If further setting $\lambda = 0$ in the resulting update equation, it will lead to the same form of the weighted gradient algorithm (5.16)! This indicates that gradient descent on the reweighted ℓ_1 (5.22) with AST reparameterization reduces to an implicit weighted ℓ_2 norm regularization algorithm when λ goes to zero, whilst with a differently defined $\mathbf{W}_t$. The benefit of incorporating the reweighted ℓ_1 framework is that a broader class of $\mathbf{W}_t$ for promoting sparsity is now possible, as both reweighted ℓ_2 and ℓ_1 frameworks can be considered for obtaining the weighting matrix. Table 5.1 presents several examples of the diversity measure $G(\boldsymbol{\theta})$ and the corresponding forms of $\mathbf{W}_t$. More example functions can be found in [81].

Table 5.1: Example diversity measures and corresponding update forms of $\mathbf{W}_t$.

Diversity measure type	$G(\boldsymbol{\theta})$ **function** $g(\theta_i) =$	**Parameter range**	**Reweighting framework**	$\mathbf{W}_t$ **update** $w_{i,t} =$
p-norm-like [14, 42]	$\lvert\theta_i\rvert^p$	$0 < p \leq 2$	reweighted ℓ_2	$\left(\frac{2}{p}\left(\lvert\theta_{i,t}\rvert + c\right)^{2-p}\right)^{\frac{1}{2}}$
p-norm-like [14, 42]	$\lvert\theta_i\rvert^p$	$0 < p \leq 1$	reweighted ℓ_1	$\frac{1}{p}\left(\lvert\theta_{i,t}\rvert + c\right)^{1-p}$
log-sum [15]	$\log(\theta_i^2 + \epsilon)$	$\epsilon > 0$	reweighted ℓ_2	$\left(\theta_{i,t}^2 + \epsilon\right)^{\frac{1}{2}}$
log-sum [16]	$\log(\lvert\theta_i\rvert + \epsilon)$	$\epsilon > 0$	reweighted ℓ_1	$\lvert\theta_{i,t}\rvert + \epsilon$
inverse tangent [16]	$\arctan(\lvert\theta_i\rvert/\epsilon)$	$\epsilon > 0$	reweighted ℓ_1	$\theta_{i,t}^2/\epsilon + \epsilon$

Group Sparsity Regularization

Structured sparsity are sometimes more preferable than the commonly seen unstructured sparsity, e.g., in neural network pruning [143]. It can be imposed by grouping parameters and encouraging sparsity among groups. Note that the sizes of groups need not be equal, e.g., in neural networks a group can be the parameters associated with a node, a filter, a channel, or even a layer.

To illustrate, let $\boldsymbol{\theta}^{(j)}$ denote the j-th group parameters and $|\boldsymbol{\theta}^{(j)}|$ denote its cardinality. As an example, we consider the group sparsity regularizer based on the p-norm-like diversity measure: $G(\boldsymbol{\theta}) = \sum_{j=0}^{J-1} \|\boldsymbol{\theta}^{(j)}\|_2^p$, $p \in (0, 2]$, where J is the number of groups. Based on the reweighted ℓ_2 framework in Section 5.4.1, the majorizer at timestep t is suggested as:

$$\sum_{j=0}^{J-1} (w_t^{(j)})^{-2} \|\boldsymbol{\theta}^{(j)}\|_2^2, \tag{5.25}$$

where

$$w_t^{(j)} = \left(\frac{2}{p} \left(\|\boldsymbol{\theta}^{(j)}\|_2 + c \right)^{2-p} \right)^{\frac{1}{2}}. \tag{5.26}$$

Note that (5.25) can be further written as $\sum_{i=0}^{M-1} w_{i,t}^{-2} \theta_i^2$, where $w_{i,t} = w_t^{(j)}$, $\forall \theta_i \in$ group j. This means that all the θ_i belonging to the j-th group share the same weight $w_t^{(j)}$. Consequently, (5.25) can also be expressed as $\|\mathbf{W}_t^{-1}\boldsymbol{\theta}\|_2^2$, where $\mathbf{W}_t = \text{diag}\{w_{i,t}\} = \text{diag}\{w_t^{(j)} \mathbf{I}_{|\boldsymbol{\theta}^{(j)}|}\}$ and $\mathbf{I}_{|\boldsymbol{\theta}^{(j)}|}$ is the $|\boldsymbol{\theta}^{(j)}| \times |\boldsymbol{\theta}^{(j)}|$ identity matrix. Then, the weighted gradient algorithm (5.16) utilizing such $\mathbf{W}_t$ can therefore promote group sparsity, in which all the parameters in a group share the same weight that is computed based on the ℓ_2 norm of the group. Following the same argument, similar algorithms can also be developed for the reweighed ℓ_1 class.

Total Variation Regularization

We demonstrate that the proposed framework can suitably incorporate the total variation regularizer widely used in image processing tasks [77]. For simplicity we only consider the one dimensional case here, where regularization is employed to help recover a piecewise constant signal $\boldsymbol{\theta} \in \mathbb{R}^M$ whose *variation*, i.e., the difference between consecutive samples, is sparse [141]. For such signals, a total variation-like regularizer $G(\tilde{\boldsymbol{\theta}})$ can be suitably used to impose sparsity on the *variation vector* $\tilde{\boldsymbol{\theta}} \triangleq \mathbf{D}\boldsymbol{\theta}$, where $\mathbf{D} \in \mathbb{R}^{M\times M}$ is a finite-difference operator consisting of -1's on the diagonal and 1's on the first upper off-diagonal. Based on the reweighted ℓ_2 or ℓ_1 framework, the majorizer $\|\mathbf{W}_t^{-1}\tilde{\boldsymbol{\theta}}\|_2^2$ or $\|\mathbf{W}_t^{-1}\tilde{\boldsymbol{\theta}}\|_1$ is suggested for $G(\tilde{\boldsymbol{\theta}})$, where $\mathbf{W}_t = \mathrm{diag}\{w_{i,t}\}$ is computed as (5.20) or (5.23) but with $\theta_{i,t}$ replaced by $\tilde{\theta}_{i,t}$. It naturally suggests the AST in this case:

$$\mathbf{q} \triangleq \mathbf{W}_t^{-1}\tilde{\boldsymbol{\theta}} = \mathbf{W}_t^{-1}\mathbf{D}\boldsymbol{\theta}. \tag{5.27}$$

This leads to having $\mathbf{W}_t^{-1}\mathbf{D}$ as the scaling matrix and eventually results in the weighted gradient algorithm:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \mathbf{D}^{-1}\mathbf{W}_t^2(\mathbf{D}^{-1})^T\nabla_{\boldsymbol{\theta}}J(\boldsymbol{\theta}_t) \tag{5.28}$$

for implicit total variation-like regularization.

5.5 Sparsity-Promoting Algorithms for Stochastic Optimization and Online Learning

When one has a large dataset or model, it would be preferable to carry out stochastic optimization or online learning schemes over batch updates, e.g., in deep neural network (DNN) training. Often a constant learning rate η may also be preferably used in such scenarios. In the following, we introduce stochastic/online variants of the sparsity-promoting weighted gradient

descent for finding compact nonlinear models in i) deep learning and ii) kernel methods.

5.5.1 Sparsity-Promoting Stochastic Optimization for DNNs

Consider the ERM problem (5.1) for the case where the hypothesis $h(\cdot;\boldsymbol{\theta})$ is a neural network with M trainable parameters (weights and biases). We treat $\boldsymbol{\theta} = [\theta_0, \theta_1, \ldots, \theta_{M-1}]^T$ as a vector consisting of all the parameters. In neural network training, the stochastic gradient descent (SGD) is widely used for learning the parameters:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \nabla_{\boldsymbol{\theta}} J_t(\boldsymbol{\theta}_t), \tag{5.29}$$

where $J_t(\boldsymbol{\theta})$ denotes the empirical risk computed only on a subset (mini-batch) of the training dataset given to the network at timestep t. Note that SGD is simply the gradient descent algorithm (5.4) using the stochastic gradient $\nabla_{\boldsymbol{\theta}} J_t(\boldsymbol{\theta}_t)$ and a constant learning rate η instead.

Using the $\mathbf{W}_t$ from the SSR reweighting methods, i.e., (5.20) or (5.23), together with the AST reparamterization (5.9), we can develop the weighted gradient version of SGD that promotes sparsity:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \mathbf{W}_t^2 \nabla_{\boldsymbol{\theta}} J_t(\boldsymbol{\theta}_t), \tag{5.30}$$

which essentially uses the stochastic gradient and a fixed learning rate for the weighted gradient algorithm (5.16). We refer to (5.30) as the Sparsity-promoting SGD (SSGD) algorithm for learning compact neural network models.

In practice, we find that normalizing the $\mathbf{W}_t^2$ term in (5.30) helps stabilize SSGD as a constant learning rate is used (see Section 5.3.1). We heuristically propose the practical SSGD update rule:

$$\boxed{\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \mathbf{S}_t \nabla_{\boldsymbol{\theta}} J_t(\boldsymbol{\theta}_t),} \tag{5.31}$$

where $\mathbf{S}_t = \mathrm{diag}\{s_{i,t}\}$, referred to as the *sparsity-promoting matrix*, is the (layer-wise) normalized

version of $\mathbf{W}_t^2$ computed according to:

$$\boxed{s_{i,t} = \frac{w_{i,t}^2}{\frac{1}{|\mathcal{I}^{(k)}|}\sum_{j\in\mathcal{I}^{(k)}} w_{j,t}^2}, \quad \text{for } i \in \mathcal{I}^{(k)},} \tag{5.32}$$

where $\mathcal{I}^{(k)}$ denotes the index set of the parameters of layer k and $|\mathcal{I}^{(k)}|$ is the cardinality of $\mathcal{I}^{(k)}$. Algorithm 4 summarizes the proposed SSGD algorithm which can be implemented using standard deep learning libraries without much effort. Later in Section 5.6.1, we show that SSGD is useful for DNN compression purposes.

Algorithm 4: SSGD for learning sparse DNN connections. $\mathbf{w}_t$ and $\mathbf{s}_t$ denote the vectors consisting of the diagonal elements of $\mathbf{W}_t$ and $\mathbf{S}_t$, respectively. $\odot$ denotes element-wise multiplication.

Input: learning rate $\eta > 0$, mini-batch of training data, and the choice of the diversity measure
Output: estimated model parameters $\boldsymbol{\theta}_t$
Initialize: $\boldsymbol{\theta}_0$
for $t = 0, 1, 2...$ **do**
  Compute gradient $\nabla_{\boldsymbol{\theta}} J_t(\boldsymbol{\theta}_t)$ via backpropagation
  Compute scaling factors: $\mathbf{w}_t$ according to the specified diversity measure (e.g., see Table 5.1)
  Compute sparsity-promoting factors: $\mathbf{s}_t$ by (5.32)
  Update parameters: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \cdot \mathbf{s}_t \odot \nabla_{\boldsymbol{\theta}} J_t(\boldsymbol{\theta}_t)$
end for

5.5.2 Sparsity-Promoting Online Learning for Kernel Methods

We demonstrate that the weighted gradient algorithmic framework can suitably extend to nonlinear estimation techniques using kernels. To begin, let the input space $\mathcal{X}$ be a compact subset of $\mathbb{R}^L$. We aim to estimate a nonlinear mapping $\phi(\cdot)$: $\mathcal{X} \mapsto \mathbb{R}$ with sequentially arriving input-output pairs $(\mathbf{x}_t, y_t) \in \mathcal{X} \times \mathbb{R}$, where t is the discrete time index. The function $\phi(\cdot)$ is modeled as an element of a *reproducing kernel Hilbert space*, $\mathcal{H}$, associated with a *Mercer kernel* $\kappa(\cdot,\cdot)$:

$\mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$, which satisfies $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \left\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \right\rangle_{\mathcal{H}}$, $\forall \mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}$, where $\langle \cdot, \cdot \rangle_{\mathcal{H}} \colon \mathcal{H} \times \mathcal{H} \mapsto \mathbb{R}$ is the inner product of $\mathcal{H}$ [144].

Consider a training set of N input-output pairs $(\mathbf{x}_n, y_n) \in \mathcal{X} \times \mathbb{R}$, $n = 0, 1, \ldots, N-1$. To estimate $\phi(\cdot)$, the following nonlinear LS problem can be formulated:

$$\min_{\phi(\cdot) \in \mathcal{H}} \quad \mathcal{J}(\phi(\cdot)) = \frac{1}{2N} \sum_{n=0}^{N-1} \left(y_n - \phi(\mathbf{x}_n) \right)^2 . \tag{5.33}$$

The *representer theorem* [145] states that any optimal solution $\phi^*(\cdot) = \arg\min_{\phi(\cdot) \in \mathcal{H}} \mathcal{J}(\phi(\cdot))$ can be expressed as a *kernel expansion* in terms of available training data, i.e., $\phi^*(\cdot) = \sum_{i=0}^{N-1} \theta_i \kappa(\cdot, \mathbf{x}_i)$, where the elements of the set $\bar{\mathcal{D}} = \left\{ \kappa(\cdot, \mathbf{x}_i) \right\}_{i=0}^{N-1}$ form a *basis*, or *dictionary*, and the θ_i are the corresponding *expansion coefficients* of the dictionary elements. By virtue of the representer theorem, we have the equivalent linear LS problem [146]:

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^N} \quad J(\boldsymbol{\theta}) = \frac{1}{2N} \sum_{n=0}^{N-1} \left(y_n - \bar{\boldsymbol{\kappa}}_n^T \boldsymbol{\theta} \right)^2 , \tag{5.34}$$

where $\bar{\boldsymbol{\kappa}}_n = [\kappa(\mathbf{x}_n, \mathbf{x}_0), \kappa(\mathbf{x}_n, \mathbf{x}_1), \ldots, \kappa(\mathbf{x}_n, \mathbf{x}_{N-1})]^T \in \mathbb{R}^N$ is the *kernelized* input data vector formed by evaluating $\mathbf{x}_n$ over the entire dictionary $\bar{\mathcal{D}}$.

For online learning upon the arrival of $(\mathbf{x}_t, y_t)$, instead of (5.34) we consider the *instantaneous* squared error:

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^t} \quad J_t(\boldsymbol{\theta}) = \frac{1}{2} \left(y_t - \boldsymbol{\kappa}_t^T \boldsymbol{\theta} \right)^2 , \tag{5.35}$$

where $\boldsymbol{\kappa}_t = [\kappa(\mathbf{x}_t, \mathbf{x}_0), \kappa(\mathbf{x}_t, \mathbf{x}_1), \ldots, \kappa(\mathbf{x}_t, \mathbf{x}_t)]^T \in \mathbb{R}^{t+1}$ is given by evaluating $\mathbf{x}_t$ over the updated dictionary $\mathcal{D}_{t+1} = \left\{ \kappa(\cdot, \mathbf{x}_i) \right\}_{i=0}^{t}$. Note that the dictionary size gets increased by 1 when the new input $\mathbf{x}_t$ becomes available, i.e., $\mathcal{D}_{t+1} = \{ \mathcal{D}_t, \kappa(\cdot, \mathbf{x}_t) \}$. Thus, the final dictionary size is equal to the number of samples seen. It is likely that *redundancy* is exhibited when the dictionary becomes larger.

The kernel least mean square (KLMS) algorithm [146, 147, 148, 149] for updating the

coefficients $\boldsymbol{\theta}$ can be obtained by using the gradient descent update for (5.35), given as:

$$\boldsymbol{\theta}_{t+1} = \begin{bmatrix} \boldsymbol{\theta}_t \\ 0 \end{bmatrix} + \eta \boldsymbol{\kappa}_t e_t, \tag{5.36}$$

where $e_t = y_t - \boldsymbol{\kappa}_t^T [\boldsymbol{\theta}_t^T \, 0]^T$ is the instantaneous error and the term $\boldsymbol{\kappa}_t e_t$ is the (negative) gradient of $J_t(\boldsymbol{\theta})$. The 0 element inserted after $\boldsymbol{\theta}_t$ in (5.36) is to account for the new entry $\kappa(\cdot, \mathbf{x}_t)$ that has just been included to the dictionary.

To incorporate sparsity, we employ the diversity measure for (5.35): $\min_{\boldsymbol{\theta}} J_t(\boldsymbol{\theta}) + \lambda G(\boldsymbol{\theta})$. Then, the reweighted ℓ_2 or ℓ_1 framework in Section 5.4 suggests solving: $\min_{\boldsymbol{\theta}} J_t(\boldsymbol{\theta}) + \lambda ||\mathbf{W}_t^{-1}\boldsymbol{\theta}||_2^2$ or $\min_{\boldsymbol{\theta}} J_t(\boldsymbol{\theta}) + \lambda ||\mathbf{W}_t^{-1}\boldsymbol{\theta}||_1$ instead, where $\mathbf{W}_t$ is evaluated based on $\boldsymbol{\theta}_t$ as given by (5.20) or (5.23). Based on the discussion in Section 5.3, we can obtain the weighted gradient version of (5.36) for promoting sparsity:

$$\boldsymbol{\theta}_{t+1} = \begin{bmatrix} \boldsymbol{\theta}_t \\ 0 \end{bmatrix} + \eta \begin{bmatrix} \mathbf{W}_t^2 & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{\kappa}_t e_t, \tag{5.37}$$

where a diagonal weighting matrix $\mathbf{W}_t^2$ is present before the (negative) gradient $\boldsymbol{\kappa}_t e_t$, and to account for the newly included entry we simply assign a weight of 1. For stability purposes, practically we would normalize the weighting matrix to account for a fixed learning rate as discussed in Section 5.3.1. We heuristically suggest normalizing $\mathbf{W}_t^2$ by the mean value of its diagonal elements, leading to the following update equation instead of (5.37):

$$\boxed{\boldsymbol{\theta}_{t+1} = \begin{bmatrix} \boldsymbol{\theta}_t \\ 0 \end{bmatrix} + \eta \mathbf{S}_t \boldsymbol{\kappa}_t e_t,} \tag{5.38}$$

where

$$\boxed{\mathbf{S}_t = \begin{bmatrix} \frac{\mathbf{w}_t^2}{\frac{1}{t}\mathrm{tr}(\mathbf{w}_t^2)} & 0 \\ 0 & 1 \end{bmatrix},} \tag{5.39}$$

is the *sparsity-promoting matrix* responsible for encouraging sparsity in $\boldsymbol{\theta}$. We refer the algorithm (5.38) as the Sparsity-promoting KLMS (SKLMS).

The KLMS has a normalized version, i.e., the kernel normalized LMS (KNLMS) [147]:

$$\boldsymbol{\theta}_{t+1} = \begin{bmatrix} \boldsymbol{\theta}_t \\ 0 \end{bmatrix} + \frac{\tilde{\eta}\boldsymbol{\kappa}_t e_t}{\boldsymbol{\kappa}_t^T \boldsymbol{\kappa}_t + \delta}, \tag{5.40}$$

where $\tilde{\eta}$ is the "normalized" learning rate [19] and $\delta > 0$ is a regularization constant for avoiding singularity. The KNLMS can be derived by the projection method [147], or by performing exact line search for the optimal learning step using the instantaneous objective, similar to the NLMS for LMS [49] in the linear case. Adopting the latter approach we can similarly derive the normalized version of SKLMS:

$$\boxed{\boldsymbol{\theta}_{t+1} = \begin{bmatrix} \boldsymbol{\theta}_t \\ 0 \end{bmatrix} + \frac{\tilde{\eta}\mathbf{S}_t\boldsymbol{\kappa}_t e_t}{\boldsymbol{\kappa}_t^T \mathbf{S}_t \boldsymbol{\kappa}_t + \delta},} \tag{5.41}$$

where $\mathbf{S}_t$ is given by (5.39). We refer to the algorithm (5.41) as the Sparsity-promoting KNLMS (SKNLMS).

Algorithm 5 summarizes the proposed SKLMS and SKNLMS algorithms. Later in Section 5.6.2, we present experiments showing their capabilities of obtaining compact dictionaries in nonlinear estimation tasks using kernels.

Algorithm 5: SKLMS and SKNLMS for learning sparse representations in kernel methods.

Input: learning rate $\eta > 0$ (or $\tilde{\eta} > 0$), regularization constant $\delta > 0$, input-output pair $(\mathbf{x}_t, y_t)$, kernel function $\kappa(\cdot,\cdot)$, and the choice of the diversity measure
Output: expansion coefficients $\boldsymbol{\theta}_t$ and dictionary $\mathcal{D}_t$
Initialize: $\boldsymbol{\theta}_0$ and $\mathcal{D}_0$ as empty set
for $t = 0, 1, 2, \ldots$ **do**
    Update dictionary: $\mathcal{D}_{t+1} = \{\mathcal{D}_t, \kappa(\cdot, \mathbf{x}_t)\}$
    Compute $\boldsymbol{\kappa}_t = [\kappa(\mathbf{x}_t, \mathbf{x}_0), \kappa(\mathbf{x}_t, \mathbf{x}_1), \ldots, \kappa(\mathbf{x}_t, \mathbf{x}_t)]^T$ by evaluating $\mathbf{x}_t$ over $\mathcal{D}_{t+1}$
    Compute error: $e_t = y_t - \boldsymbol{\kappa}_t^T [\boldsymbol{\theta}_t^T \; 0]^T$
    Compute scaling matrix $\mathbf{W}_t$ according to the specified diversity measure (e.g., see Table 5.1)
    Compute sparsity-promoting matrix $\mathbf{S}_t$ as in (5.39)
    Update expansion coefficients:
        * SKLMS: $\boldsymbol{\theta}_{t+1} = \begin{bmatrix} \boldsymbol{\theta}_t \\ 0 \end{bmatrix} + \eta \mathbf{S}_t \boldsymbol{\kappa}_t e_t$
        * SKNLMS: $\boldsymbol{\theta}_{t+1} = \begin{bmatrix} \boldsymbol{\theta}_t \\ 0 \end{bmatrix} + \dfrac{\tilde{\eta} \mathbf{S}_t \boldsymbol{\kappa}_t e_t}{\boldsymbol{\kappa}_t^T \mathbf{S}_t \boldsymbol{\kappa}_t + \delta}$
end for

5.6 Simulation Results

5.6.1 SSGD

We evaluate the proposed SSGD algorithm for neural networks using the PyTorch [150] library. We first consider a simple multilayer perceptron (MLP) example for studying the effect of model size and initialization on the sparsification results. Then, examples of sparsifying more complex convolutional neural networks (CNNs) for realistic image classification tasks are presented. The rectified linear unit (ReLU) activation is used for all the models considered. For all the results, we use (5.21) for $\mathbf{W}_t$ in SSGD, setting $c = 0.001$. To visualize the sparsification performance, we measure the excess kurtosis of the parameters within each layer.[7]

[7]Distributions with excess kurtosis higher than 0 are called super-Gaussian, meaning that they have higher peaks at 0 and heavier tails compared to the Gaussian distribution, which has an excess kurtosis of 0. Excess kurtosis can

Effect of Model Size and Initialization

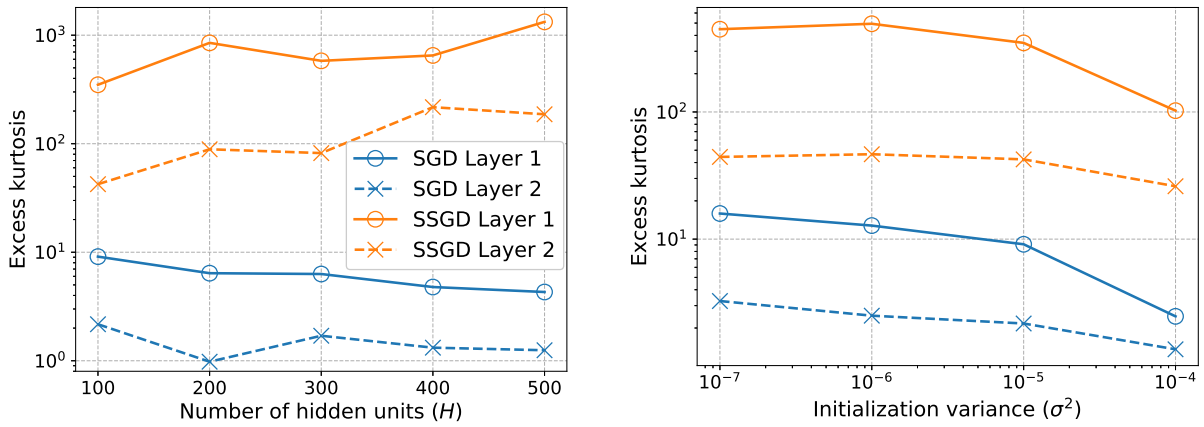
We study how the model size and initialization would affect the performance of SSGD using a simple MLP architecture. We consider the regression problem with $N = 10$ data pairs $(\mathbf{x}_i, y_i) \in \mathbb{R}^{20} \times \mathbb{R}$, where y_i is generated by a pre-defined one-hidden-layer MLP with 100 hidden units using $\mathbf{x}_i$ as input – a highly overparameterized scenario – where the pre-defined network parameters are drawn from i.i.d. Gaussian with zero mean and 0.01 variance, and the elements of $\mathbf{x}_i$ are drawn from i.i.d. standard Gaussian. The generated data pairs are used to train another one-hidden-layer MLP with H hidden units for comparing the learning performance. For reference, we also include the results of using SGD (5.29) for training. The same initialization is used for both SGD and SSGD in each case for fair comparison, where the initial parameters $\boldsymbol{\theta}_0$ are drawn from i.i.d. Gaussian with zero mean and σ^2 variance. H and σ^2 are to be specified in different experiments. A learning rate $\eta = 0.001$ and a batch size of 1 are used for all cases. The squared error loss is considered for optimization.

Figure 5.1 (a) investigates the effect of the model size on the sparsity-promoting performance. We train networks with different H to zero loss using SGD and SSGD with $p = 1$, and measure the excess kurtosis. We use $\sigma^2 = 10^{-5}$ for the initialization in each case. We see that SSGD consistently achieves higher sparsity than SGD. In addition, the first layer (Layer 1) is constantly sparser than the second layer (Layer 2) since it has much more parameters. Moreover, when the size increases, there is a trend of increasing sparsity for SSGD. This may be reasonable, as the degree of overparameterization increases with the model size, and thus there is more likely the presence of higher redundancy.

Figure 5.1 (b) studies the effect of initialization where we experiment with different σ^2. We use the network with $H = 100$ hidden units and train to zero loss to measure the excess kurtosis in each case. We again see that SSGD constantly finds a sparser solution than the SGD across different initialization variances. Notably, we see that when the initialization variance increases,

thus serve as a measure of sparsity (the higher, the sparser).

sparsity decreases to some degree. This is as expected, as indicated by Remark 5.2, when the initialization is closer to the origin, the regularization effect becomes more obvious and thus the algorithm should be finding a sparser model; otherwise, initialization would dominate. This indicates that as long as we keep the initialization close to the origin, which is not uncommon in practice, SSGD is able to attain a sparse network.



(a) (b)

Figure 5.1: Effect on sparsity-promoting performance of (a) model size and (b) initialization variance.

Sparsifying CNNs for Image classification

We consider two image classification tasks with different CNN architectures consisting of fully-connected (FC) and convolutional (CONV) layers, both using the cross-entropy loss for optimization:

- *CNN-1 on MNIST database [151]*: We define a model (referred to as CNN-1) that has 2 CONV layers (# input channels $\times$ # output channels: $1\times32-32\times64$) using 5×5 kernels followed by 3 FC layers (# input neurons $\times$ # output neurons: $2304\times128-128\times64-64\times10$) for this task. Max pooling is performed after each CONV layer.

- *CNN-2 on CIFAR-10 database [152]*: We define a more complicated model (referred to as CNN-2) with 6 CONV layers ($3\times64 - 64\times64 - 64\times128 - 128\times128 - 128\times256 - 256\times256$) using 3×3 kernels followed by 3 FC layers ($4096\times256 - 256\times128 - 128\times10$) for this task. Each of the CONV layer is followed by batch normalization [153] before activation. Max pooling is performed after the second, forth, and last CONV layers. Dropout [154] with a rate of 0.2 is applied to the first and second FC layers.

We compare SSGD with SGD (5.29). In addition, to compare with explicit regularization techniques, we also include the results of using SGD for optimizing an ℓ_1 regularized objective, which is equivalent to using the following update rule:

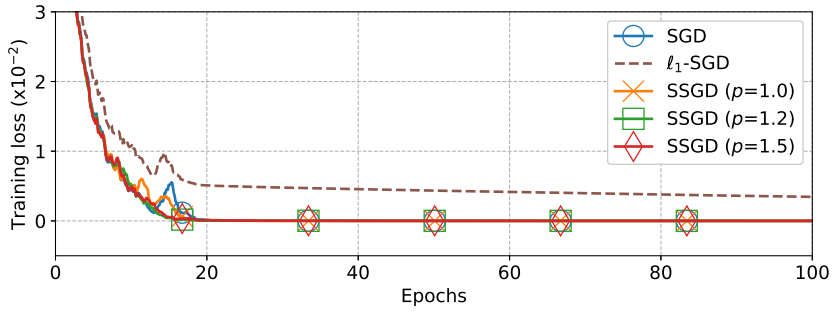
$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta\nabla_{\boldsymbol{\theta}} J_t(\boldsymbol{\theta}_t) - \eta\lambda\text{sgn}(\boldsymbol{\theta}_t). \tag{5.42}$$

We set $\lambda = 10^{-6}$ and 10^{-5} for CNN-1 and CNN-2, respectively, and refer to (5.42) as 'ℓ_1-SGD'. A learning rate $\eta = 0.1$ and a batch size of 64 are used for all the algorithms.

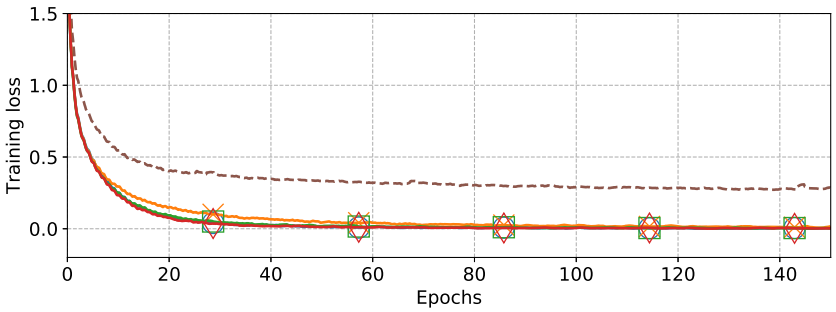
Figure 5.2 shows the training loss vs. epochs for SGD, ℓ_1-SGD, and SSGD with $p = 1.0$, 1.2, and 1.5. We train CNN-1 on MNIST for 100 epochs and CNN-2 on CIFAR-10 for 150 epochs. The same initialization is used among different algorithms for each model by using the default scheme of PyTorch. For reference, CNN-1 and CNN-2 achieve 99.27% and 85.21% test accuracy with normal SGD training, respectively. The results show that SSGD is able to converge toward the same loss as SGD, supporting the argument that SSGD finds solutions to the original unpenalized problem. The ℓ_1-SGD, however, ends up at a higher loss due to the bias induced by a nonzero λ for the ℓ_1 norm regularizer.

Figure 5.3 monitors the excess kurtosis vs. epochs for SSGD with various p values. We see that a smaller p leads to greater sparsity as expected. Note that when using (5.21) with $p = 2$, SSGD reduces to SGD, resulting in near 0 excess kurtosis.

Figure 5.4 confirms the observations in Figure 5.3 by comparing the weight distribution



(a)

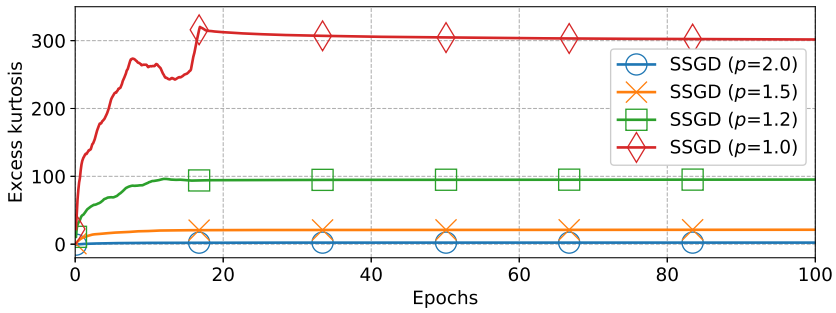


(b)

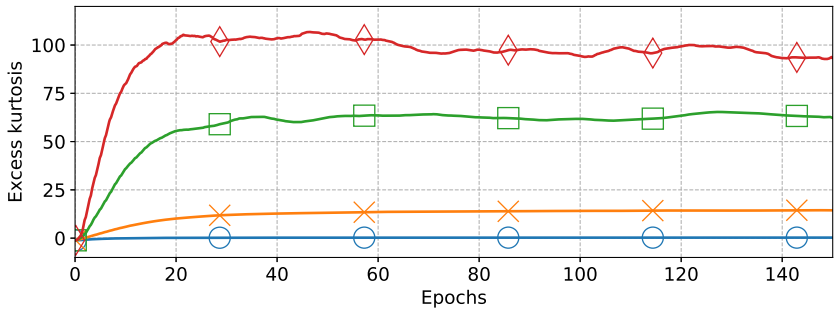
Figure 5.2: Training loss vs. epochs for (a) CNN-1 on MNIST and (b) CNN-2 on CIFAR-10.

densities of the models trained by SGD and SSGD (using $p = 1$). It can be seen that SSGD learns a heavier-tailed distribution with a higher peak at 0, meaning greater sparsity in the parameters. The sparsity is shown to be beneficial for DNN compression purposes as presented next.

**Application to DNN compression:* Han et al. [155] have proposed a 3-stage compression scheme: i) learning important connections, ii) pruning unimportant parameters by hard thresholding, and iii) fine-tuning the remaining ones. We adopt the same scheme, using SSGD in stage i). In [155], it is observed that, ℓ_1 regularization leads to sparser networks after stage i), but the network loses significant accuracy after stage ii), and is not able to recover from this accuracy drop even after stage iii). The authors posit that the discrepancy between using ℓ_1 regularization during stage i) and not using it during stage iii) leads to poor performance. SSGD circumvents such issues because it finds sparse solutions by optimizing the original unpenalized problem



(a)



(b)

Figure 5.3: Excess kurtosis vs. epochs for (a) first FC layer weights of CNN-1 and (b) last CONV layer weights of CNN-2.

directly, instead of switching between penalized and unpenalize problems like [155].

Figure 5.5 shows the test accuracy vs. % of nonzeros after pruning for different cases. We use the magnitude-based strategy from [155] to fix small weights to 0 in stage ii). As can be seen in Figure 5.5, after pruning (solid lines), accuracy drops with decreasing % of nonzeros (more aggressive pruning). SSGD (using $p = 1$) retains the highest accuracy after pruning in both Figure 5.5 (a) and Figure 5.5 (b). ℓ_1-SGD also maintains higher accuracy than SGD in Figure 5.5 (b). As both cases are sparsity-aware training, this supports the argument that sparsity is important for learning compact connectivity of models [156, 157]. Now, to regain accuracy, fine-tuning is necessary. Compared to the iterative process suggested in [155], one-shot pruning and retraining is more desirable [156]. In addition, the retraining period should also be kept short. Therefore, we fine-tune the pruned models once (CNN-1 for 35 epochs and CNN-2 for 50 epochs

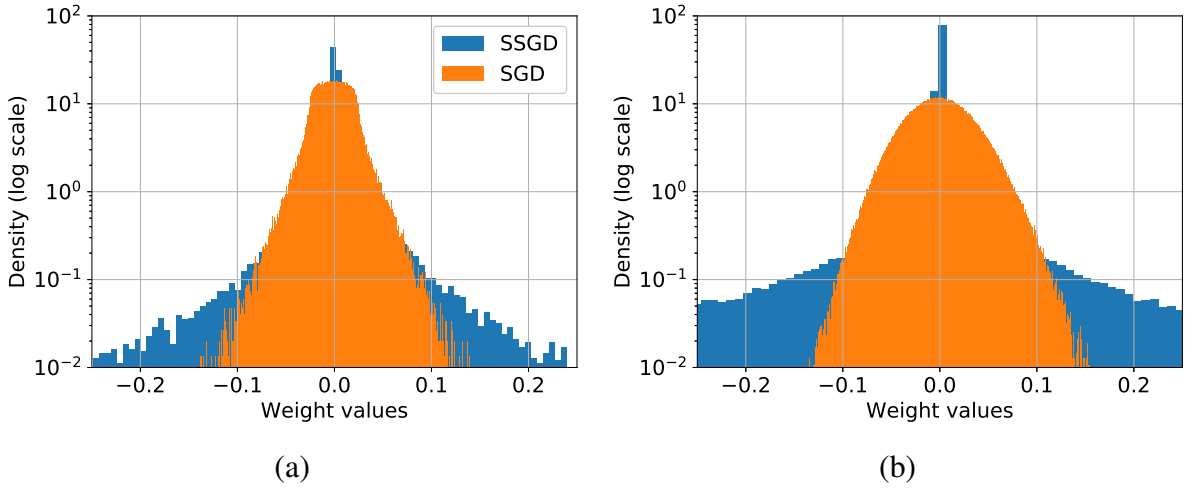
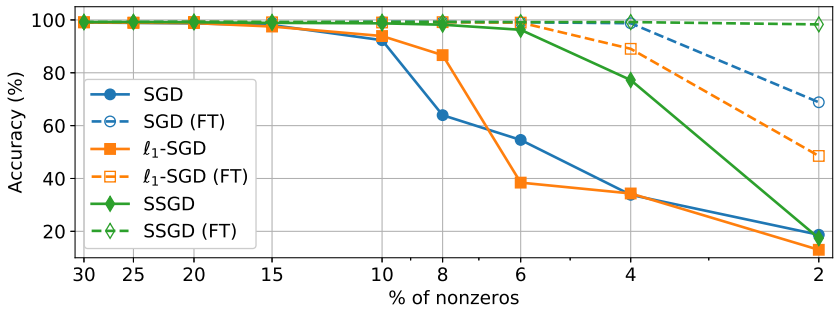


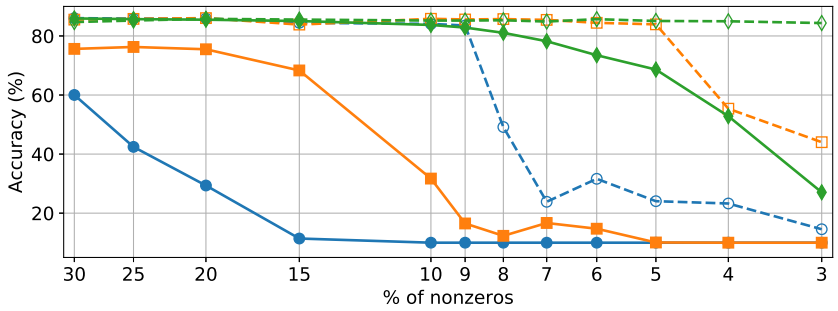
Figure 5.4: Distribution of (a) first FC layer weights of CNN-1 and (b) last CONV layer weights of CNN-2. Training with SSGD results in a super-Gaussian like distribution of parameters that has a higher peak at 0 and heavier tails than SGD.

only) by optimizing the unpenalized problem using the Adam optimizer [32] with a learning rate of 0.001. From Figure 5.5, we see that after fine-tuning (dashed lines, labeled with '(FT)'), accuracy can be regained to a certain degree for all cases. Note that ℓ_1-SGD is not necessarily better than the normal SGD after retraining, e.g., in Figure 5.5 (a). The proposed SSGD, on the other hand, achieves the highest accuracy after fine-tuning. This demonstrates that SSGD can learn better network connectivity in the training phase by leveraging the implicit sparsity regularization property, which happens to also be beneficial for avoiding possible issues due to change of optimization modes in the fine-tuning stage.

Table 5.2 compares the sparsification performance of the proposed SSGD-based approach to some recent pruning methods. We compare with [158], which also utilizes the iterative reweighting concept in their pruning framework. However, their method prunes a pre-trained network via log-sum minimization in a layer-by-layer fashion. Our approach, on the other hand, sparsifies all layers simultaneously during training. Moreover, we have a broader framework that covers the log-sum penalty as a special case. For comparison purposes, we adopt the same network architectures as in their paper, namely, a multi-layer perceptron on MNIST (referred to as



(a)



(b)

Figure 5.5: Test accuracy vs. % of nonzeros for (a) CNN-1 on MNIST and (b) CNN-2 on CIFAR-10. 'FT': fine-tuned.

MLP) which consists of 4 FC layers, and a CNN on CIRAF-10 (referred to as CNN-3) which consists of 2 CONV layers (each with batch normalization added before activation) followed by 3 FC layers. We also compare with Net-Trim [159], another pruning method also compared with in [158]. For the proposed method, we train the models with SSGD using $p = 1$. Then, we prune the models once and fine-tune using Adam. From the results, we can see that the proposed method achieves the highest sparsity with comparable, if not better, accuracy compared to existing methods.

5.6.2 SKLMS and SKNLMS

We present two examples of nonlinear estimation problems using computer simulations in MATLAB to demonstrate the proposed SKLMS and SKNLMS for obtaining a compact dictionary

Table 5.2: Comparison of sparsification results.

Model	Method	Accuracy	% of nonzeros
MLP	Original	98.62%	100.0
	Net-Trim [159]	97.70%	30.5
	Iter. Reweight. [158]	97.46%	14.8
	Proposed	98.39%	3.7
CNN-3	Original	77.44%	100.0
	Net-Trim [159]	75.92%	17.8
	Iter. Reweight. [158]	74.17%	7.9
	Proposed	74.54%	5.1

in kernel methods. Besides the KLMS (5.36) and KNLMS (5.40), we also compare to:

$$\boldsymbol{\theta}_{t+1} = \begin{bmatrix} \boldsymbol{\theta}_t \\ 0 \end{bmatrix} + \eta \boldsymbol{\kappa}_t e_t - \lambda \eta \text{sgn}(\begin{bmatrix} \boldsymbol{\theta}_t \\ 0 \end{bmatrix}) \tag{5.43}$$

and

$$\boldsymbol{\theta}_{t+1} = \begin{bmatrix} \boldsymbol{\theta}_t \\ 0 \end{bmatrix} + \frac{\tilde{\eta} \boldsymbol{\kappa}_t e_t}{\boldsymbol{\kappa}_t^T \boldsymbol{\kappa}_t + \delta} - \frac{\lambda \tilde{\eta}}{\boldsymbol{\kappa}_t^T \boldsymbol{\kappa}_t + \delta} \text{sgn}(\begin{bmatrix} \boldsymbol{\theta}_t \\ 0 \end{bmatrix}), \tag{5.44}$$

where the λ-weighted terms with the $\text{sgn}(\cdot)$ function are referred to as the *zero attractor* (ZA) [24]. The algorithms in (5.43) and (5.44) are thus referred to as the ZA-KLMS [160] and ZA-KNLMS, respectively, which represent methods that explicitly use $\lambda > 0$ for imposing sparsity – they can be derived from: $\min_{\boldsymbol{\theta}} J_t(\boldsymbol{\theta}) + \lambda \|\boldsymbol{\theta}\|_1$, where $J_t(\boldsymbol{\theta})$ is the instantaneous objective in (5.35).

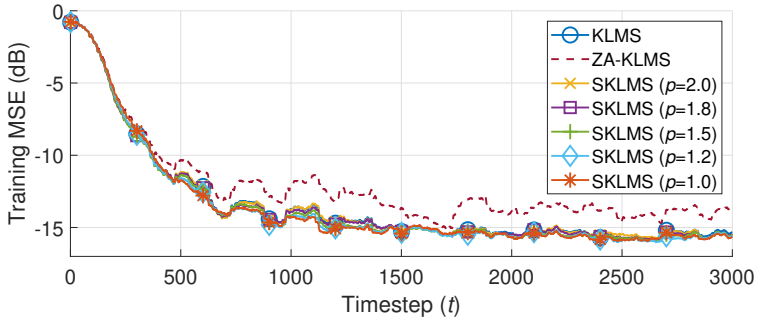
- *Mackey-Glass (MG) chaotic time series prediction:* We follow the experiment in [161] for predicting a MG chaotic time series u_t. A segment of 3000 samples is used for training and another segment of 2000 samples for testing. A time embedding $L = 12$ is used, i.e., $\mathbf{x}_t = [u_{t-1}, u_{t-2}, ..., u_{t-12}]^T$, to predict the present sample, i.e., $y_t = u_t$. The signal is corrupted by a white Gaussian noise with zero mean and 0.01 variance before processing. KLMS-type algorithms are compared in this task.

- *Nonlinear channel equalization:* We follow the experiment in [161] for nonlinear channel equalization. A binary signal $b_t \in \{-1, +1\}$ is fed into a nonlinear channel which is modeled as a Wiener system, where the output of the linear part $l_t = b_t + 0.5b_{t-1}$ is input to the nonlinear part that gives the system output $o_t = l_t - 0.9l_t^2$. The received signal $r_t = o_t + v_t$, where v_t is a white Gaussian noise with zero mean and 0.01 variance. A time embedding $L = 5$ and time lag $\tau = 2$ are used, i.e., $\mathbf{x}_t = [r_t, r_{t-1}, ..., r_{t-4}]^T$, to predict the delayed sample, i.e., $y_t = b_{t-2}$. We generate 10000 samples for training and another 5000 samples for testing. KNLMS-type algorithms are compared in this task.

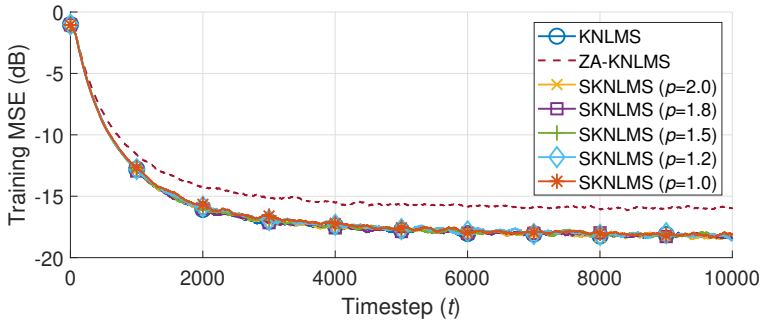
For all the algorithms we use the Gaussian kernel $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp(-a\|\mathbf{x}_i - \mathbf{x}_j\|_2^2)$ with $a = 1$. We use the same $\eta = 0.001$ and $\tilde{\eta} = 0.1$ for KLMS-type and KNLMS-type algorithms, respectively. For SKLMS and SKNLMS, we use (5.21) for $\mathbf{W}_t$, setting $c = 0.001$. For KNLMS-type algorithms we use $\delta = 0.01$. For ZA-KLMS and ZA-KNLMS we use $\lambda = 0.015$ and $\lambda = 0.002$, respectively, which are chosen such that they achieve the highest sparsity with minimal bias incurred. All the mean squared error (MSE) results are obtained by ensemble averaging over 100 Monte Carlo runs.

Figure 5.6 compares the training results of the two tasks. In both cases, one can see that SKLMS and SKNLMS converge to the same MSE level as KLMS and KNLMS, respectively, whereas ZA-KLMS and ZA-KNLMS result in a higher MSE – a consequence of trading off model fitting for regularization (sparsity). The results demonstrate that the proposed algorithms do not incur a regularization bias to the optimization.

Table 5.3 compares sparsity levels of the converged coefficients $\boldsymbol{\theta}$ learned by different algorithms in terms of the *sparseness* measure from [162] (range from 0 to 1, the higher the sparser). We see that for SKLMS and SKNLMS, a smaller p results in a higher degree of sparsity as expected. ZA-KLMS and ZA-KNLMS also yield sparser solutions than KLMS and KNLMS. However, the increased sparsity comes with increased MSE bias (see Figure 5.6). Moreover, also based on the ℓ_1 norm, SKLMS and SKNLMS with $p = 1$ achieve sparser solutions than ZA-KLMS and ZA-KNLMS.



(a)



(b)

Figure 5.6: Training results of (a) KLMS-type algorithms for MG chaotic time series prediction and (b) KNLMS-type algorithms for nonlinear channel equalization.

Table 5.3: Sparseness of learned expansion coefficients $\boldsymbol{\theta}$ of (a) KLMS-type algorithms for MG chaotic time series prediction and (b) KNLMS-type algorithms for nonlinear channel equalization.

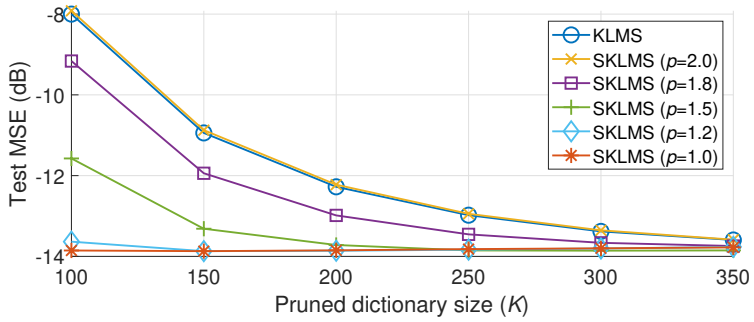
(a)

Algorithm	KLMS	ZA-KLMS	SKLMS				
			$p = 2.0$	$p = 1.8$	$p = 1.5$	$p = 1.2$	$p = 1.0$
Sparseness	0.59	0.64	0.59	0.63	0.72	0.80	0.85

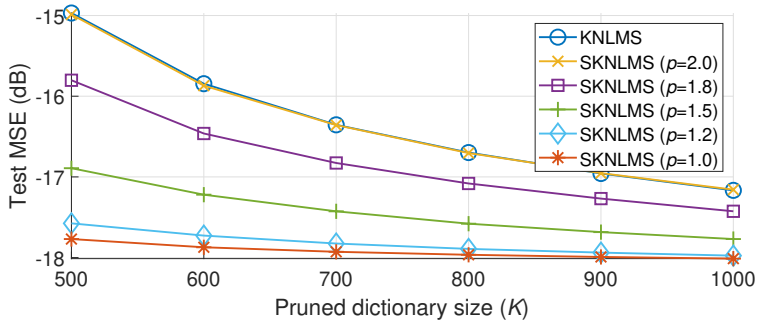
(b)

Algorithm	KNLMS	ZA-KNLMS	SKNLMS				
			$p = 2.0$	$p = 1.8$	$p = 1.5$	$p = 1.2$	$p = 1.0$
Sparseness	0.78	0.80	0.78	0.79	0.83	0.86	0.88

Figure 5.7 presents the pruning results, where we prune the learned models and see how they perform on the test data. Specifically, we remove elements of the final dictionary based on the *least magnitude criterion* [163] of the learned $\boldsymbol{\theta}$. We experiment with different pruned dictionary sizes K. It can be seen that SKLMS and SKNLMS using a small p can retain a relatively low error even for aggressive pruning (small K), as the learned $\boldsymbol{\theta}$ is sufficiently sparse.



(a)



(b)

Figure 5.7: Test results on pruning performance of (a) SKLMS for MG chaotic time series prediction and (b) SKNLMS for nonlinear channel equalization.

5.7 Implicit Complexity Regularization Using Fisher-Rao Norm Capacity Measure

The Fisher-Rao norm (FR norm) proposed in [137] acts as a capacity measure of neural network models, which can potentially be used as a regularizer for reduced model complexity. It is defined as: $\|\boldsymbol{\theta}\|_{\text{FR}} \triangleq \|\mathbf{F}^{\frac{1}{2}}(\boldsymbol{\theta})\boldsymbol{\theta}\|_2$, where $\mathbf{F}(\boldsymbol{\theta})$ is the *Fisher information matrix (FIM)*. The FR norm can also serve as a measure of *flatness* of minima, which is hypothesized to be related to generalization of deep nets [164], since the FIM approximates the Hessian at a minimum of the loss under certain conditions. Thus, a model with a smaller FR norm could be associated with better generalization capabilities.

If we make the modeling assumption of the joint probability that $p(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta}) = p(\mathbf{x})p(\mathbf{y}|\mathbf{x}; \boldsymbol{\theta})$, as we identify the loss function $L(h(\mathbf{x}; \boldsymbol{\theta}), \mathbf{y}) = -\log p(\mathbf{y}|\mathbf{x}; \boldsymbol{\theta})$, we have the FIM as $\mathbf{F}(\boldsymbol{\theta}) = \mathrm{E}_{\mathbf{x},\mathbf{y}}\left[\nabla_{\boldsymbol{\theta}} L(h(\mathbf{x}; \boldsymbol{\theta}), \mathbf{y}) \nabla_{\boldsymbol{\theta}} L(h(\mathbf{x}; \boldsymbol{\theta}), \mathbf{y})^T\right]$. However, the FIM may not be available in practice and has to be estimated. For neural networks, one possible alternative is the running sum estimate:

$$\mathbf{F}_t = \frac{1}{t+1} \sum_{\tau=0}^{t} \nabla_{\boldsymbol{\theta}} J_\tau(\boldsymbol{\theta}_\tau) \nabla_{\boldsymbol{\theta}} J_\tau(\boldsymbol{\theta}_\tau)^T, \tag{5.45}$$

where $J_\tau(\boldsymbol{\theta})$ is the empirical risk computed only on a subset (mini-batch) of the training data given to the network at timestep τ. Therefore, we can adopt $\|\mathbf{F}_t^{\frac{1}{2}}\boldsymbol{\theta}\|_2^2$ as the regularizer for capacity control purposes, which in turn suggests using $\mathbf{F}_t^{-1}$ as the weighting matrix, leading to the (stochastic) weighted gradient algorithm:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \mathbf{F}_t^{-1} \nabla_{\boldsymbol{\theta}} J_t(\boldsymbol{\theta}_t). \tag{5.46}$$

This actually corresponds to the (full matrix) *adaptive gradient algorithm* proposed in [31]. By using only the diagonal approximates of $\mathbf{F}_t$, the root of $\mathbf{F}_t$ instead, or introducing some empirical

gradient smoothing operations, (5.46) can be connected to many popular algorithms like AdaGrad [31], RMSProp [33], and Adam [32]. In this sense, we interpret the algorithms from an implicit regularization viewpoint for practicing capacity control. This possibly indicates their effectiveness in tending to a solution with a smaller model complexity, which could be associated with better generalization properties. Further studies with rigorous arguments and supporting experimental results are left for future research.

5.8 Conclusion

In this chapter, we studied a novel AST reparameterization scheme to associate weighted gradient descent with weighted norm regularization. We argued that by leveraging implicit regularization, through weighted gradient algorithms it is possible to obtain regularized models that exhibit desired properties without incorporating a regularization penalty, given that a suitable weighting matrix is provided. We presented weighting matrix examples for sparsity, group sparsity, total variation, and capacity control to demonstrate flexibility of the weighted gradient algorithmic framework. Utilizing the reweighting SSR techniques within the framework, we further introduced the SSGD, SKLMS, and SKNLMS algorithms for learning sparse representations in nonlinear estimation tasks.

Acknowledgment

Chapter 5 is, in part, a reprint of the material as it appears in the two papers: C.-H. Lee, B. D. Rao, and H. Garudadri, "Weighted gradient descent algorithms for learning regularized models," *IEEE Transactions on Signal Processing*, under review and C.-H. Lee, I. Fedorov, B. D. Rao, and H. Garudadri, "SSGD: Sparsity-promoting stochastic gradient descent algorithm for unbiased DNN pruning," in *45th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2020. The dissertation author was the primary investigator and

author of these papers. The work, in part, was supported by National Institutes of Health/National Institute on Deafness and Other Communication Disorders under Grants R01DC015436 and R33DC015046 and National Science Foundation/Information and Intelligent Systems under Award 1838830.

5.9 Appendix

5.9.1 Proof of Proposition 5.1

Despite the fact that Proposition 5.1 is well-known in the literature, we provide a proof here for completeness.

For the LS problem (5.2), using the gradient $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \frac{1}{N}\mathbf{X}^T(\mathbf{X}\boldsymbol{\theta} - \mathbf{y})$ for the gradient descent update (5.4) and starting with $\boldsymbol{\theta}_0$, we have:

$$\begin{aligned}\boldsymbol{\theta}_{t+1} =& \boldsymbol{\theta}_t - \eta_t \frac{1}{N}\mathbf{X}^T(\mathbf{X}\boldsymbol{\theta}_t - \mathbf{y}) \\ =& \boldsymbol{\theta}_0 - \sum_{\tau=1}^{t} \eta_\tau \frac{1}{N}\mathbf{X}^T(\mathbf{X}\boldsymbol{\theta}_\tau - \mathbf{y}) \\ =& \boldsymbol{\theta}_0 + \mathbf{X}^T\boldsymbol{\beta}, \quad \text{for some } \boldsymbol{\beta} \in \mathbb{R}^N.\end{aligned} \tag{5.47}$$

For sufficiently small learning rates, gradient descent converges to a solution $\boldsymbol{\theta}^*_{\text{gd}}$ when $t \to \infty$. According to (5.47), we must have $\boldsymbol{\theta}^*_{\text{gd}} = \boldsymbol{\theta}_0 + \mathbf{X}^T\boldsymbol{\beta}^*$, for some $\boldsymbol{\beta}^* \in \mathbb{R}^N$.

Due to convexity, $\boldsymbol{\theta}^*_{\text{gd}}$ must be a global minimizer. As we assume $\mathbf{y} \in \mathcal{R}(\mathbf{X})$, the global optimum must be zero, i.e., $J(\boldsymbol{\theta}^*_{\text{gd}}) = \frac{1}{2N}\|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}^*_{\text{gd}}\|_2^2 = 0$. Thus, we have:

$$\begin{aligned}\mathbf{X}\boldsymbol{\theta}^*_{\text{gd}} = \mathbf{y} \quad &\Rightarrow \quad \mathbf{X}(\boldsymbol{\theta}_0 + \mathbf{X}^T\boldsymbol{\beta}^*) = \mathbf{y} \\ &\Rightarrow \quad \boldsymbol{\beta}^* = (\mathbf{X}\mathbf{X}^T)^{-1}(\mathbf{y} - \mathbf{X}\boldsymbol{\theta}_0).\end{aligned} \tag{5.48}$$

Therefore, we have the minimizer:

$$\begin{aligned}\boldsymbol{\theta}^*_{\text{gd}} =& \boldsymbol{\theta}_0 + \mathbf{X}^T\boldsymbol{\beta}^* \\ =& \boldsymbol{\theta}_0 + \mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}(\mathbf{y} - \mathbf{X}\boldsymbol{\theta}_0) \\ =& \mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{y} + (\mathbf{I} - \mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X})\boldsymbol{\theta}_0.\end{aligned} \tag{5.49}$$

Note that $\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{y} = \arg\min_{\boldsymbol{\theta}} \|\boldsymbol{\theta}\|_2^2$ s.t. $\mathbf{y} = \mathbf{X}\boldsymbol{\theta}$ is the minimum ℓ_2 norm solution $\boldsymbol{\theta}_{\text{min}}$ and $\mathbf{I} - \mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}$ is the the projection matrix onto $\mathcal{N}(\mathbf{X})$. Therefore, we have the converged solution:

$$\boldsymbol{\theta}^*_{\text{gd}} = \boldsymbol{\theta}_{\text{min}} + \mathcal{P}_{\mathcal{N}(\mathbf{X})}(\boldsymbol{\theta}_0). \tag{5.50}$$

5.9.2 Proof of Corollary 5.1

Let $\boldsymbol{\theta}_0 \in \mathcal{R}(\mathbf{X}^T)$. Then we have $\boldsymbol{\theta}_0 \perp \mathcal{N}(\mathbf{X})$, as $\mathcal{R}(\mathbf{X}^T)$ and $\mathcal{N}(\mathbf{X})$ are orthogonal complements. This implies the projection $\mathcal{P}_{\mathcal{N}(\mathbf{X})}(\boldsymbol{\theta}_0) = \mathbf{0}$. Thus, by Proposition 5.1, we have $\boldsymbol{\theta}^*_{\text{gd}} = \boldsymbol{\theta}_{\text{min}}$.

Let $\boldsymbol{\theta}^*_{\text{gd}} = \boldsymbol{\theta}_{\text{min}}$. Then from Proposition 5.1 we know that $\mathcal{P}_{\mathcal{N}(\mathbf{X})}(\boldsymbol{\theta}_0) = \mathbf{0}$. This indicates $\boldsymbol{\theta}_0 \perp \mathcal{N}(\mathbf{X})$ and thus $\boldsymbol{\theta} \in \mathcal{R}(\mathbf{X}^T)$.

Therefore, $\boldsymbol{\theta}^*_{\text{gd}} = \boldsymbol{\theta}_{\text{min}}$ if and only if $\boldsymbol{\theta}_0 \in \mathcal{R}(\mathbf{X}^T)$.

5.9.3 Proof of Proposition 5.2

First note that the $\mathbf{q}$ domain update rule (5.6) is the gradient descent update for solving:

$$\min_{\mathbf{q}} \quad J(\mathbf{W}\mathbf{q}) = \frac{1}{2N}\|\mathbf{y} - \mathbf{X}\mathbf{W}\mathbf{q}\|_2^2. \tag{5.51}$$

Based on the discussion in Proposition 5.1, starting from $\mathbf{q}_0$ and with sufficiently small η_t, we have (5.6) converge to:

$$\mathbf{q}^*_{\text{gd}} = \mathbf{q}_{\text{min}} + \mathcal{P}_{\mathcal{N}(\mathbf{XW})}(\mathbf{q}_0), \tag{5.52}$$

where $\mathbf{q}_{\text{min}} = \mathbf{W}^T\mathbf{X}^T(\mathbf{XWW}^T\mathbf{X}^T)^{-1}\mathbf{y} = \arg\min_{\mathbf{q}} \|\mathbf{q}\|_2^2 \quad \text{s.t. } \mathbf{y} = \mathbf{XWq}$.

Since the the weighted gradient descent (5.7) is the equivalent $\boldsymbol{\theta}$ domain update of (5.6), we have it converge to $\boldsymbol{\theta}^*_{\text{wgd}} = \mathbf{Wq}^*_{\text{gd}}$. Using (5.52), we have:

$$\begin{aligned}\boldsymbol{\theta}^*_{\text{wgd}} =& \mathbf{Wq}_{\text{min}} + \mathbf{W}\mathcal{P}_{\mathcal{N}(\mathbf{XW})}(\mathbf{q}_0) \\ =& \mathbf{WW}^T\mathbf{X}^T(\mathbf{XWW}^T\mathbf{X}^T)^{-1}\mathbf{y} + \mathbf{W}\mathcal{P}_{\mathcal{N}(\mathbf{XW})}(\mathbf{q}_0).\end{aligned} \tag{5.53}$$

By noting the fact that $\mathbf{WW}^T\mathbf{X}^T(\mathbf{XWW}^T\mathbf{X}^T)^{-1}\mathbf{y} = \arg\min_{\boldsymbol{\theta}} \|\mathbf{W}^{-1}\boldsymbol{\theta}\|_2^2 \quad \text{s.t. } \mathbf{y} = \mathbf{X}\boldsymbol{\theta}$ is the minimum weighted ℓ_2 norm solution $\boldsymbol{\theta}_{\text{wmin}}$ and $\mathbf{q}_0 = \mathbf{W}^{-1}\boldsymbol{\theta}_0$, we have:

$$\boldsymbol{\theta}^*_{\text{wgd}} = \boldsymbol{\theta}_{\text{wmin}} + \mathbf{W}\mathcal{P}_{\mathcal{N}(\mathbf{XW})}(\mathbf{W}^{-1}\boldsymbol{\theta}_0). \tag{5.54}$$

5.9.4 Proof of Corollary 5.2

Let $\boldsymbol{\theta}_0 \in \mathcal{R}(\mathbf{WW}^T\mathbf{X}^T)$. Then $\boldsymbol{\theta}_0 = \mathbf{WW}^T\mathbf{X}^T\boldsymbol{\beta}_0$, for some $\boldsymbol{\beta}_0 \in \mathbb{R}^N$. Since $\mathbf{q}_0 = \mathbf{W}^{-1}\boldsymbol{\theta}_0 = \mathbf{W}^T\mathbf{X}^T\boldsymbol{\beta}_0$, we have $\mathbf{q}_0 \in \mathcal{R}(\mathbf{W}^T\mathbf{X}^T)$ and thus $\mathbf{q}_0 \perp \mathcal{N}(\mathbf{XW})$. Therefore, the projection $\mathcal{P}_{\mathcal{N}(\mathbf{XW})}(\mathbf{q}_0) = \mathbf{0}$. By Proposition 5.2, we thus have $\boldsymbol{\theta}^*_{\text{wgd}} = \boldsymbol{\theta}_{\text{wmin}}$.

Let $\boldsymbol{\theta}^*_{\text{wgd}} = \boldsymbol{\theta}_{\text{wmin}}$. From Proposition 5.2, we know that $\mathbf{W}\mathcal{P}_{\mathcal{N}(\mathbf{XW})}(\mathbf{W}^{-1}\boldsymbol{\theta}_0) = \mathbf{0}$. Since $\mathbf{W}$ is nonsingular, we must have $\mathcal{P}_{\mathcal{N}(\mathbf{XW})}(\mathbf{W}^{-1}\boldsymbol{\theta}_0) = \mathbf{0}$ and therefore $\mathbf{W}^{-1}\boldsymbol{\theta}_0 \perp \mathcal{N}(\mathbf{XW})$. This implies $\mathbf{W}^{-1}\boldsymbol{\theta}_0 \in \mathcal{R}(\mathbf{W}^T\mathbf{X}^T)$ and thus $\mathbf{W}^{-1}\boldsymbol{\theta}_0 = \mathbf{W}^T\mathbf{X}^T\boldsymbol{\beta}_0$, for some $\boldsymbol{\beta}_0 \in \mathbb{R}^N$. This suggests $\boldsymbol{\theta}_0 = \mathbf{WW}^T\mathbf{X}^T\boldsymbol{\beta}_0$, for some $\boldsymbol{\beta}_0 \in \mathbb{R}^N$, i.e., $\boldsymbol{\theta}_0 \in \mathcal{R}(\mathbf{WW}^T\mathbf{X}^T)$.

Therefore, $\boldsymbol{\theta}^*_{\text{wgd}} = \boldsymbol{\theta}_{\text{wmin}}$ if and only if $\boldsymbol{\theta}_0 \in \mathcal{R}(\mathbf{WW}^T\mathbf{X}^T)$.

5.9.5 Proof of Theorem 5.1

We prove that there exists *at least* a sequence $\{\eta_t\}_{t=0}^{\infty}$ such that the weighted gradient algorithm (5.16) monotonically converges to a local minimum (or saddle point) of the ERM problem (5.1), by showing that $J(\boldsymbol{\theta})$ is decreased at each iteration.

First we note that the $\mathbf{q}$ domain update rule (5.14) is the gradient descent update for solving:

$$\min_{\mathbf{q}} \quad J(\mathbf{W}_t\mathbf{q}). \tag{5.55}$$

Therefore, there exists an η_t for (5.14) such that

$$J(\mathbf{W}_t\mathbf{q}_{t+1|t}) - J(\mathbf{W}_t\mathbf{q}_{t|t}) < 0, \tag{5.56}$$

for each t. Since (5.16) is equivalent to (5.14), this means that there exists an η_t for (5.16) such that (5.56) holds for each t. By the AST relationships (5.11) and (5.12), we have:

$$J(\boldsymbol{\theta}_{t+1}) - J(\boldsymbol{\theta}_t) = J(\mathbf{W}_t\mathbf{q}_{t+1|t}) - J(\mathbf{W}_t\mathbf{q}_{t|t}) < 0, \tag{5.57}$$

for using (5.16) with η_t for iteration t. This means that there exists a sequence $\{\eta_t\}_{t=0}^{\infty}$ for (5.16) such that $J(\boldsymbol{\theta})$ is decreased at each iteration. Thus, it monotonically converges to a local minimum (or saddle point) of the empirical risk.

Another way to interpret the above is that the correction term $\mathbf{W}_t^2\nabla_{\boldsymbol{\theta}}J(\boldsymbol{\theta}_t)$ in (5.16) is a *descent direction* for (5.1). Just as in gradient descent (5.4) the term $\nabla_{\boldsymbol{\theta}}J(\boldsymbol{\theta}_t)$ is a descent direction for (5.1) and thus it monotonically converges to a local minimum (or saddle point) for some sequence of learning rates, the same argument applies for (5.16).

Chapter 6

Conclusions

This dissertation presented a family of gradient descent algorithms which employ SSR techniques for promoting sparsity to achieve improved convergence characteristics or obtain desirable solutions. In the following, we review the main contributions delineated in this dissertation.

In Chapter 2, we developed a mathematical framework that utilizes an AST methodology within the iterative reweighted ℓ_2 and ℓ_1 frameworks for deriving LMS and NLMS adaptive filtering algorithms of the proportionate type. In particular, we introduced the SLMS and SNLMS algorithms by adopting a zero regularization coefficient in our framework, which takes advantage of, though do not strictly enforce, the sparsity of the underlying system if it already exists. Unlike most of the existing proportionate algorithms that design the proportionate factors heuristically, our SSR-motivated framework leads to a more systematic way of designing the factors, and permits incorporation of a broad class of diversity measures that have proved effective for SSR in our algorithms.

In Chapter 3, we utilized the reweighting and AST strategies in the context of CG-type adaptive filtering and developed the SCG algorithm. To our knowledge, it is the first work on

sparsity-aware CG adaptive filtering. The SCG in general has a much higher convergence rate than the LMS-type algorithms while having a higher computational complexity. Compared to other existing adaptive algorithms with the same order of complexity but not leveraging sparsity (e.g., the m-NLMS and other CG-type adaptive filters), SCG also demonstrates superior convergence characteristics for identifying sparse systems.

In Chapter 4, we presented an important engineering application of the SLMS, i.e., the AFC problem in HAs. We showed that the SLMS is suitable for improving AFC by leveraging the "quasi-" sparse structure of feedback path IRs, given that it has the flexibility to incorporate different degrees of sparsity. To further improve AFC, we also introduced "freping," a frequency warping method that utilizes a network of all-pass filters to decorrelate the signal and mitigate the NSC for better feedback reduction with negligible distortion incurred. Finally, to quantify the trade-off between speech quality and stable gain performance of AFC systems, we proposed an off-line, HASQI-based ASG estimation approach. The approach was utilized to verify that the proposed SLMS+freping AFC system outperforms existing methods in the literature.

In Chapter 5, we turned our attention to model optimization of the empirical risk minimization problem over a given dataset, and studied a novel AST-based reparameterization scheme to associate weighted gradient descent with weighted norm regularization. We argued that by leveraging implicit regularization, through weighted gradient algorithms it is possible to obtain regularized models that exhibit desirable properties without having to deal with the task of selecting a weight for the regularization penalty. We presented weighting matrix examples for incorporating various regularizers to demonstrate flexibility of the weighted gradient algorithmic framework. In particular, two sparsity regularization applications utilizing the reweighting SSR techniques were presented, namely, i) the SSGD algorithm for neural network compression in deep learning and ii) the SKLMS and SKNLMS algorithms for dictionary pruning in kernel methods. The proposed algorithms were shown to be capable of learning sparse representations in several nonlinear estimation tasks without explicitly incorporating a regularization penalty.

Bibliography

[1] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*, Berlin, Germany: Springer, 2010.

[2] Y. C. Eldar and G. Kutyniok, *Compressed Sensing: Theory and Applications*, Cambridge, U.K.: Cambridge University Press, 2012.

[3] S. Foucart and H. Rauhut, "An invitation to compressive sensing," in *A Mathematical Introduction to Compressive Sensing*, pp. 1–39. Berlin, Germany: Springer, 2013.

[4] I. F. Gorodnitsky, J. S. George, and B. D. Rao, "Neuromagnetic source imaging with FOCUSS: A recursive weighted minimum norm algorithm," *Electroencephalography and Clinical Neurophysiology*, vol. 95, no. 4, pp. 231–251, 1995.

[5] S. F. Cotter and B. D. Rao, "Sparse channel estimation via matching pursuit with application to equalization," *IEEE Transactions on Communications*, vol. 50, no. 3, pp. 374–377, 2002.

[6] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[7] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[8] M. Lustig, D. Donoho, and J. M. Pauly, "Sparse MRI: The application of compressed sensing for rapid MR imaging," *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, vol. 58, no. 6, pp. 1182–1195, 2007.

[9] Z. Zhang, T.-P. Jung, S. Makeig, and B. D. Rao, "Compressed sensing of EEG for wireless telemonitoring with low energy consumption and inexpensive hardware," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 1, pp. 221–224, 2012.

[10] J. Liu, P. C. Cosman, and B. D. Rao, "Robust linear regression via ℓ_0 regularization," *IEEE Transactions on Signal Processing*, vol. 66, no. 3, pp. 698–713, 2017.

[11] Y. Ding and B. D. Rao, "Dictionary learning-based sparse channel representation and estimation for FDD massive MIMO systems," *IEEE Transactions on Wireless Communications*, vol. 17, no. 8, pp. 5437–5451, 2018.

[12] D. Ge, X. Jiang, and Y. Ye, "A note on the complexity of l_p minimization," *Mathematical Programming*, vol. 129, no. 2, pp. 285–299, 2011.

[13] D. Wipf and S. Nagarajan, "Iterative reweighted ℓ_1 and ℓ_2 methods for finding sparse solutions," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 317–329, 2010.

[14] I. F. Gorodnitsky and B. D. Rao, "Sparse signal reconstruction from limited data using FOCUSS: A re-weighted minimum norm algorithm," *IEEE Transactions on Signal Processing*, vol. 45, no. 3, pp. 600–616, 1997.

[15] R. Chartrand and W. Yin, "Iteratively reweighted algorithms for compressive sensing," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2008, pp. 3869–3872.

[16] E. J. Candès, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5, pp. 877–905, 2008.

[17] Y. Sun, P. Babu, and D. P. Palomar, "Majorization-minimization algorithms in signal processing, communications, and machine learning," *IEEE Transactions on Signal Processing*, vol. 65, no. 3, pp. 794–816, 2017.

[18] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*, Upper Saddle River, NJ, USA: Pearson, 1985.

[19] S. Haykin, *Adaptive Filter Theory*, 5th edition, Upper Saddle River, NJ, USA: Pearson, 2013.

[20] A. H. Sayed, *Adaptive Filters*, Hoboken, NJ, USA: John Wiley & Sons, 2011.

[21] D. G. Manolakis, V. K. Ingle, and S. M. Kogon, *Statistical and Adaptive Signal Processing: Spectral Estimation, Signal Modeling, Adaptive Filtering, and Array Processing*, New York, NY, USA: McGraw-Hill, 2000.

[22] Y. Huang, J. Benesty, and J. Chen, "Sparse adaptive filters," *Acoustic MIMO Signal Processing*, pp. 59–84, 2006.

[23] C. Paleologu, J. Benesty, and S. Ciochină, "Sparse adaptive filters for echo cancellation," *Synthesis Lectures on Speech and Audio Processing*, vol. 6, no. 1, pp. 1–124, 2010.

[24] Y. Chen, Y. Gu, and A. O. Hero, "Sparse LMS for system identification," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2009, pp. 3125–3128.

[25] Y. Gu, J. Jin, and S. Mei, "l_0 norm constraint LMS algorithm for sparse system identification," *IEEE Signal Processing Letters*, vol. 16, no. 9, pp. 774–777, 2009.

[26] K. Wagner and M. Doroslovački, *Proportionate-Type Normalized Least Mean Square Algorithms*, Hoboken, NJ, USA: John Wiley & Sons, 2013.

[27] D. L. Duttweiler, "Proportionate normalized least-mean-squares adaptation in echo cancelers," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 5, pp. 508–518, 2000.

[28] J. Benesty and S. L. Gay, "An improved PNLMS algorithm," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2002, pp. 1881–1884.

[29] C. Paleologu, J. Benesty, and S. Ciochină, "An improved proportionate NLMS algorithm based on the l_0 norm," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2010, pp. 309–312.

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[31] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of International Conference on Learning Representations*, 2014.

[33] T. Tieleman and G. Hinton, "Lecture 6.5-RMSProp: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural Networks for Machine Learning*, vol. 4, no. 2, pp. 26–31, 2012.

[34] B. Neyshabur, R. Tomioka, and N. Srebro, "In search of the real inductive bias: On the role of implicit regularization in deep learning," in *Proceedings of International Conference on Learning Representations Workshops*, 2014.

[35] B. Neyshabur, R. Tomioka, R. Salakhutdinov, and N. Srebro, "Geometry of optimization and implicit regularization in deep learning," *arXiv preprint arXiv:1705.03071*, 2017.

[36] S. Gunasekar, B. E. Woodworth, S. Bhojanapalli, B. Neyshabur, and N. Srebro, "Implicit regularization in matrix factorization," in *Advances in Neural Information Processing Systems*, 2017, pp. 6151–6159.

[37] A. C. Wilson, R. Roelofs, M. Stern, N. Srebro, and B. Recht, "The marginal value of adaptive gradient methods in machine learning," in *Advances in Neural Information Processing Systems*, 2017, pp. 4148–4158.

[38] N. Azizan and B. Hassibi, "Stochastic gradient/mirror descent: Minimax optimality and implicit regularization," in *Proceedings of International Conference on Learning Representations*, 2018.

[39] S. Gunasekar, J. Lee, D. Soudry, and N. Srebro, "Characterizing implicit bias in terms of optimization geometry," in *Proceedings of International Conference on Machine Learning*, 2018, pp. 1832–1841.

[40] Y. Li and Y. Liang, "Learning overparameterized neural networks via stochastic gradient descent on structured data," in *Advances in Neural Information Processing Systems*, 2018, pp. 8157–8166.

[41] N. Azizan, S. Lale, and B. Hassibi, "A study of generalization of stochastic mirror descent algorithms on overparameterized nonlinear models," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2020, pp. 3132–3136.

[42] B. D. Rao and K. Kreutz-Delgado, "An affine scaling methodology for best basis selection," *IEEE Transactions on Signal Processing*, vol. 47, no. 1, pp. 187–200, 1999.

[43] F. Y. Wu and F. Tong, "Gradient optimization p-norm-like constraint LMS algorithm for sparse system estimation," *Signal Processing*, vol. 93, no. 4, pp. 967–971, 2013.

[44] O. Taheri and S. A. Vorobyov, "Reweighted l_1-norm penalized LMS for sparse channel estimation and its analysis," *Signal Processing*, vol. 104, pp. 70–79, 2014.

[45] G. K. Boray and M. D. Srinath, "Conjugate gradient techniques for adaptive filtering," *IEEE Transactions on Circuits and Systems–I: Fundamental Theory and Applications*, vol. 39, no. 1, pp. 1–10, 1992.

[46] P. S. Chang and A. N. Willson, Jr., "Analysis of conjugate gradient algorithms for adaptive filtering," *IEEE Transactions on Signal Processing*, vol. 48, no. 2, pp. 409–418, 2000.

[47] O. Diene and A. Bhaya, "Adaptive filtering algorithms designed using control Lyapunov functions," *IEEE Signal Processing Letters*, vol. 13, no. 4, pp. 224–227, 2006.

[48] S. Zhao, Z. Man, and S. Khoo, "A generalized data windowing scheme for adaptive conjugate gradient algorithms," *Signal Processing*, vol. 89, no. 5, pp. 894–900, 2009.

[49] T. Variddhisaï and D. P. Mandic, "On an RLS-like LMS adaptive filter," in *Proceedings of IEEE International Conference on Digital Signal Processing*, 2017, pp. 1–5.

[50] P. S. R. Diniz, M. O. K. Mendonça, J. O. Ferreira, and T. N. Ferreira, "Data-selective conjugage gradient algorithm," in *Proceedings of European Signal Processing Conference*, 2018, pp. 707–711.

[51] T. van Waterschoot and M. Moonen, "Fifty years of acoustic feedback control: State of the art and future challenges," *Proceedings of the IEEE*, vol. 99, no. 2, pp. 288–327, 2011.

[52] Open Speech Platform, *http://openspeechplatform.ucsd.edu/*, University of California, San Diego, 2020.

[53] L. Pisha, J. Warchall, T. Zubatiy, S. Hamilton, C.-H. Lee, G. Chockalingam, P. P. Mercier, R. Gupta, B. D. Rao, and H. Garudadri, "A wearable, extensible, open-source platform for hearing healthcare research," *IEEE Access*, vol. 7, pp. 162083–162101, 2019.

[54] J. Benesty, T. Gänsler, D. R. Morgan, M. M. Sondhi, and S. L. Gay, *Advances in Network and Acoustic Echo Cancellation*, Berlin, Germany: Springer, 2001.

[55] E. Hänsler and G. Schmidt, *Topics in Acoustic Echo and Noise Control: Selected Methods for the Cancellation of Acoustical Echoes, the Reduction of Background Noise, and Speech Processing*, Berlin, Germany: Springer, 2006.

[56] L. T. T. Tran, H. Schepker, S. Doclo, H. H. Dam, and S. Nordholm, "Proportionate NLMS for adaptive feedback control in hearing aids," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2017, pp. 211–215.

[57] C.-H. Lee, B. D. Rao, and H. Garudadri, "Sparsity promoting LMS for adaptive feedback cancellation," in *Proceedings of European Signal Processing Conference*, 2017, pp. 226–230.

[58] C.-H. Lee, J. M. Kates, B. D. Rao, and H. Garudadri, "Speech quality and stable gain trade-offs in adaptive feedback cancellation for hearing aids," *Journal of the Acoustical Society of America*, vol. 142, no. 4, pp. EL388–EL394, 2017.

[59] M. Kocic, D. Brady, and M. Stojanovic, "Sparse equalization for real-time digital underwater acoustic communications," in *Proceedings of MTS/IEEE OCEANS*, 1995, vol. 3, pp. 1417–1422.

[60] S. G. Nash and A. Sofer, *Linear and Nonlinear Programming*, New York, NY, USA: McGraw-Hill, 1996.

[61] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[62] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *Journal of Machine Learning Research*, vol. 1, no. Jun, pp. 211–244, 2001.

[63] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Review*, vol. 43, no. 1, pp. 129–159, 2001.

[64] R. K. Martin, W. A. Sethares, R. C. Williamson, and C. R. Johnson, "Exploiting sparsity in adaptive filters," *IEEE Transactions on Signal Processing*, vol. 50, no. 8, pp. 1883–1894, 2002.

[65] B. D. Rao and B. Song, "Adaptive filtering algorithms for promoting sparsity," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2003, pp. 361–364.

[66] Y. Jin, *Algorithm Development for Sparse Signal Recovery and Performance Limits Using Multiple-User Information Theory*, Ph.D. dissertation, University of California, San Diego, 2011.

[67] J. Benesty, C. Paleologu, and S. Ciochină, "Proportionate adaptive filters from a basis pursuit perspective," *IEEE Signal Processing Letters*, vol. 17, no. 12, pp. 985–988, 2010.

[68] J. Liu and S. L. Grant, "A generalized proportionate adaptive algorithm based on convex optimization," in *Proceedings of IEEE China Summit and International Conference on Signal and Information Processing*, 2014, pp. 748–752.

[69] R. L. Das and M. Chakraborty, "Improving the performance of the PNLMS algorithm using l_1 norm regularization," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 7, pp. 1280–1290, 2016.

[70] Y. Li and M. Hamamura, "An improved proportionate normalized least-mean-square algorithm for broadband multipath channel estimation," *The Scientific World Journal*, vol. 2014, 2014.

[71] F. Albu, I. Caciula, Y. Li, and Y. Wang, "The l_p-norm proportionate normalized least mean square algorithm for active noise control," in *Proceedings of International Conference on System Theory, Control, and Computing*, 2017, pp. 401–405.

[72] M. V. S. Lima, T. N. Ferreira, W. A. Martins, and P. S. R. Diniz, "Sparsity-aware data-selective adaptive filters," *IEEE Transactions on Signal Processing*, vol. 62, no. 17, pp. 4557–4572, 2014.

[73] K. Pelekanakis and M. Chitre, "New sparse adaptive algorithms based on the natural gradient and the l_0-norm," *IEEE Journal of Oceanic Engineering*, vol. 38, no. 2, pp. 323–332, 2013.

[74] T. N. Ferreira, M. V. S. Lima, P. S. R. Diniz, and W. A. Martins, "Low-complexity proportionate algorithms with sparsity-promoting penalties," in *Proceedings of IEEE International Symposium on Circuits and Systems*, 2016, pp. 253–256.

[75] J. Jin, Y. Gu, and S. Mei, "A stochastic gradient approach on compressive sensing signal reconstruction based on adaptive filtering framework," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 409–420, 2010.

[76] G. Su, J. Jin, Y. Gu, and J. Wang, "Performance analysis of l_0 norm constraint least mean square algorithm," *IEEE Transactions on Signal Processing*, vol. 60, no. 5, pp. 2223–2235, 2012.

[77] J. P. Oliveira, J. M. Bioucas-Dias, and M. A. T. Figueiredo, "Adaptive total variation image deblurring: A majorization–minimization approach," *Signal Processing*, vol. 89, no. 9, pp. 1683–1693, 2009.

[78] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge, U.K.: Cambridge University Press, 2004.

[79] D. G. Luenberger and Y. Ye, *Linear and Nonlinear Programming*, 4th edition, Berlin, Germany: Springer, 2016.

[80] C.-H. Lee, B. D. Rao, and H. Garudadri, "Proportionate adaptive filters based on minimizing diversity measures for promoting sparsity," in *Proceedings of Asilomar Conference on Signals, Systems, and Computers*, 2019, pp. 769–773.

[81] L. Chen and Y. Gu, "From least squares to sparse: A non-convex approach with guarantee," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2013, pp. 5875–5879.

[82] J. Benesty, C. Paleologu, and S. Ciochină, "On regularization in adaptive filtering," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 6, pp. 1734–1742, 2010.

[83] B. D. Rao, K. Engan, S. F. Cotter, J. Palmer, and K. Kreutz-Delgado, "Subset selection in noise based on diversity measure minimization," *IEEE Transactions on Signal Processing*, vol. 51, no. 3, pp. 760–770, 2003.

[84] D. Angelosante, J. A. Bazerque, and G. B. Giannakis, "Online adaptive estimation of sparse signals: Where RLS meets the ℓ_1-norm," *IEEE Transactions on Signal Processing*, vol. 58, no. 7, pp. 3436–3447, 2010.

[85] B. Babadi, N. Kalouptsidis, and V. Tarokh, "SPARLS: The sparse RLS algorithm," *IEEE Transactions on Signal Processing*, vol. 58, no. 8, pp. 4013–4025, 2010.

[86] N. Kalouptsidis, G. Mileounis, B. Babadi, and V. Tarokh, "Adaptive algorithms for sparse system identification," *Signal Processing*, vol. 91, no. 8, pp. 1910–1919, 2011.

[87] B. Dumitrescu, A. Onose, P. Helin, and I. Tabus, "Greedy sparse RLS," *IEEE Transactions on Signal Processing*, vol. 60, no. 5, pp. 2194–2207, 2012.

[88] Y. V. Zakharov and V. H. Nascimento, "DCD-RLS adaptive filters with penalties for sparse identification," *IEEE Transactions on Signal Processing*, vol. 61, no. 12, pp. 3198–3213, 2013.

[89] X. Hong, J. Gao, and S. Chen, "Zero-attracting recursive least squares algorithms," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 1, pp. 213–221, 2016.

[90] E. M. Eksioglu and A. K. Tanc, "RLS algorithm with convex regularization," *IEEE Signal Processing Letters*, vol. 18, no. 8, pp. 470–473, 2011.

[91] E. M. Eksioglu, "Sparsity regularised recursive least squares adaptive filtering," *IET Signal Processing*, vol. 5, no. 5, pp. 480–487, 2011.

[92] H. Yazdanpanah and P. S. R. Diniz, "Recursive least-squares algorithms for sparse system modeling," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2017, pp. 3879–3883.

[93] C.-H. Lee, I. Fedorov, B. D. Rao, and H. Garudadri, "SSGD: Sparsity-promoting stochastic gradient descent algorithm for unbiased DNN pruning," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2020, pp. 3869–3872.

[94] J. R. Shewchuk, *An Introduction to the Conjugate Gradient Method without the Agonizing Pain*, 1994.

[95] R. M. Johnstone, C. R. Johnson Jr., R. R. Bitmead, and B. D. O. Anderson, "Exponential convergence of recursive least squares with exponential forgetting factor," in *Proceedings of IEEE Conference on Decision and Control*, 1982, pp. 994–997.

[96] H. Nyquist, "Regeneration theory," *Bell System Technical Journal*, vol. 11, no. 1, pp. 126–147, 1932.

[97] A. Spriet, S. Doclo, M. Moonen, and J. Wouters, "Feedback control in hearing aids," *Springer Handbook of Speech Processing*, pp. 979–1000, 2008.

[98] M. Guo, *Analysis, design, and evaluation of acoustic feedback cancellation systems for hearing aids*, Ph.D. dissertation, Aalborg University, 2012.

[99] C. R. C. Nakagawa, S. Nordholm, and W.-Y. Yan, "New insights into optimal acoustic feedback cancellation," *IEEE Signal Processing Letters*, vol. 20, no. 9, pp. 869–872, 2013.

[100] H.-F. Chi, S. X. Gao, S. D. Soli, and A. Alwan, "Band-limited feedback cancellation with a modified filtered-X LMS algorithm for hearing aids," *Speech Communication*, vol. 39, no. 1-2, pp. 147–161, 2003.

[101] H. Schepker, L. T. T. Tran, S. Nordholm, and S. Doclo, "Improving adaptive feedback cancellation in hearing aids using an affine combination of filters," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2016, pp. 231–235.

[102] T. van Waterschoot and M. Moonen, "Assessing the acoustic feedback control performance of adaptive feedback cancellation in sound reinforcement systems," in *Proceedings of European Signal Processing Conference*, 2009, pp. 1997–2001.

[103] F. Strasser and H. Puder, "Sub-band feedback cancellation with variable step sizes for music signals in hearing aids," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2014, pp. 8207–8211.

[104] F. Strasser and H. Puder, "Adaptive feedback cancellation for realistic hearing aid applications," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 12, pp. 2322–2333, 2015.

[105] M. Guo, S. H. Jensen, J. Jensen, and S. L. Grant, "On the use of a phase modulation method for decorrelation in acoustic feedback cancellation," in *Proceedings of European Signal Processing Conference*, 2012, pp. 2000–2004.

[106] C. Boukis, D. P. Mandic, and A. G. Constantinides, "Toward bias minimization in acoustic feedback cancellation systems," *Journal of the Acoustical Society of America*, vol. 121, no. 3, pp. 1529–1537, 2007.

[107] G. Ma, F. Gran, F. Jacobsen, and F. T. Agerkvist, "Adaptive feedback cancellation with band-limited LPC vocoder in digital hearing aids," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 677–687, 2011.

[108] H. Shen and L. Zhang, "A new variable step-size algorithm on acoustic feedback suppression for digital hearing aids," in *Proceedings of International Conference on Audio, Language and Image Processing*, 2014, pp. 171–175.

[109] F. Albu, C. R. C. Nakagawa, and S. Nordholm, "Proportionate algorithms for two-microphone active feedback cancellation," in *Proceedings of European Signal Processing Conference*, 2015, pp. 290–294.

[110] Vasundhara, G. Panda, and N. B. Puhan, "A VSS sparseness controlled algorithm for feedback suppression in hearing aids," in *Proceedings of IEEE International Symposium on Signal Processing and Information Technology*, 2015, pp. 151–156.

[111] J. E. Greenberg, "Modified LMS algorithms for speech processing with an adaptive noise canceller," *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 4, pp. 338–351, 1998.

[112] M. Guo, S. H. Jensen, and J. Jensen, "Evaluation of state-of-the-art acoustic feedback cancellation systems for hearing aids," *Journal of the Audio Engineering Society*, vol. 61, no. 3, pp. 125–137, 2013.

[113] A. V. Oppenheim and D. H. Johnson, "Discrete representation of signals," *Proceedings of the IEEE*, vol. 60, no. 6, pp. 681–691, 1972.

[114] C. Braccini and A. Oppenheim, "Unequal bandwidth spectral analysis using digital frequency warping," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 22, no. 4, pp. 236–244, 1974.

[115] W. P. M. Allen, D. G. Bailey, S. N. Demidenko, and V. Piuri, "Analysis and application of digital spectral warping in analog and mixed-signal testing," *IEEE Transactions on Reliability*, vol. 52, no. 4, pp. 444–457, 2003.

[116] J. B. Allen, "Short term spectral analysis, synthesis, and modification by discrete Fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 25, no. 3, pp. 235–238, 1977.

[117] J. M. Kates, "Principles of digital dynamic-range compression," *Trends in Amplification*, vol. 9, no. 2, pp. 45–76, 2005.

[118] A. Simpson, "Frequency-lowering devices for managing high-frequency hearing loss: A review," *Trends in Amplification*, vol. 13, no. 2, pp. 87–106, 2009.

[119] H. Dillon, *Hearing aids*, 2nd edition, Turramurra, Australia: Boomerang Press, 2008.

[120] J. M. Kates and K. H. Arehart, "The hearing-aid speech quality index (HASQI) version 2," *Journal of the Audio Engineering Society*, vol. 62, no. 3, pp. 99–117, 2014.

[121] J. Bondy, A. Dittberner, M. Coughlin, and B. Whitmer, "Assessing sound quality of feedback algorithms with auditory models," in *International Symposium on Auditory and Audiological Research*, 2007.

[122] A. J. Manders, D. M. Simpson, and S. L. Bell, "Objective prediction of the sound quality of music processed by an adaptive feedback canceller," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 6, pp. 1734–1745, 2012.

[123] D. Suelzle, V. Parsa, and T. H. Falk, "On a reference-free speech quality estimator for hearing aids," *Journal of the Acoustical Society of America*, vol. 133, no. 5, pp. EL412–EL418, 2013.

[124] A. A. Kressner, D. V. Anderson, and C. J. Rozell, "Evaluating the generalization of the hearing aid speech quality index (HASQI)," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 2, pp. 407–415, 2013.

[125] P. Kendrick, I. R. Jackson, F. F. Li, T. J. Cox, and B. M. Fazenda, "Perceived audio quality of sounds degraded by non-linear distortions and single-ended assessment using HASQI," *Journal of the Audio Engineering Society*, vol. 63, no. 9, pp. 689–712, 2015.

[126] J. Hellgren, "Analysis of feedback cancellation in hearing aids with Filtered-X LMS and the direct method of closed loop identification," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 2, pp. 119–131, 2002.

[127] S. Gazor and T. Liu, "Adaptive filtering with decorrelation for coloured AR environments," *IEE Proceedings – Vision, Image and Signal Processing*, vol. 152, no. 6, pp. 806–818, 2005.

[128] J. M. Alvarez and M. Salzmann, "Learning the number of neurons in deep networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 2270–2278.

[129] S. Scardapane, D. Comminiello, A. Hussain, and A. Uncini, "Group sparse regularization for deep neural networks," *Neurocomputing*, vol. 241, pp. 81–89, 2017.

[130] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," in *Proceedings of International Conference on Learning Representations*, 2017.

[131] B. Neyshabur, Z. Li, S. Bhojanapalli, Y. LeCun, and N. Srebro, "The role of over-parametrization in generalization of neural networks," in *Proceedings of International Conference on Learning Representations*, 2019.

[132] Z. Allen-Zhu, Y. Li, and Y. Liang, "Learning and generalization in overparameterized neural networks, going beyond two layers," in *Advances in Neural Information Processing Systems*, 2019, pp. 6155–6166.

[133] A. Krogh and J. A. Hertz, "A simple weight decay can improve generalization," in *Advances in Neural Information Processing Systems*, 1992, pp. 950–957.

[134] B. Neyshabur, R. R. Salakhutdinov, and N. Srebro, "Path-SGD: Path-normalized optimization in deep neural networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 2422–2430.

[135] B. Neyshabur, R. Tomioka, and N. Srebro, "Norm-based capacity control in neural networks," in *Proceedings of Conference on Learning Theory*, 2015, pp. 1376–1401.

[136] P. L. Bartlett, D. J. Foster, and M. J. Telgarsky, "Spectrally-normalized margin bounds for neural networks," in *Advances in Neural Information Processing Systems*, 2017, pp. 6240–6249.

[137] T. Liang, T. Poggio, A. Rakhlin, and J. Stokes, "Fisher-Rao metric, geometry, and complexity of neural networks," in *Proceedings of International Conference on Artificial Intelligence and Statistics*, 2019, pp. 888–896.

[138] T. Vaskevicius, V. Kanade, and P. Rebeschini, "Implicit regularization for optimal sparse recovery," in *Advances in Neural Information Processing Systems*, 2019, pp. 2968–2979.

[139] P. Zhao, Y. Yang, and Q.-C. He, "Implicit regularization via hadamard product over-parametrization in high-dimensional linear regression," *arXiv preprint arXiv:1903.09367*, 2019.

[140] M. Belkin, D. Hsu, S. Ma, and S. Mandal, "Reconciling modern machine-learning practice and the classical bias–variance trade-off," *Proceedings of the National Academy of Sciences*, vol. 116, no. 32, pp. 15849–15854, 2019.

[141] F. I. Karahanoğlu, I. Bayram, and D. Van De Ville, "A signal process. approach to generalized 1-D total variation," *IEEE Transactions on Signal Processing*, vol. 59, no. 11, pp. 5265–5274, 2011.

[142] M. Staib, S. J. Reddi, S. Kale, S. Kumar, and S. Sra, "Escaping saddle points with adaptive gradient methods," in *Proceedings of International Conference on Machine Learning*, 2019, pp. 5956–5965.

[143] W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li, "Learning structured sparsity in deep neural networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 2074–2082.

[144] N. Aronszajn, "Theory of reproducing kernels," *Transactions of the American Mathematical Society*, vol. 68, no. 3, pp. 337–404, 1950.

[145] B. Schölkopf, R. Herbrich, and A. J. Smola, "A generalized representer theorem," in *Proceedings of International Conference on Computational Learning Theory*, 2001, pp. 416–426.

[146] W. Gao, J. Chen, C. Richard, and J. Huang, "Online dictionary learning for kernel LMS," *IEEE Transactions on Signal Processing*, vol. 62, no. 11, pp. 2765–2777, 2014.

[147] C. Richard, J. C. M. Bermudez, and P. Honeine, "Online prediction of time series data with kernels," *IEEE Transactions on Signal Processing*, vol. 57, no. 3, pp. 1058–1067, 2008.

[148] W. Liu, P. P. Pokharel, and J. C. Principe, "The kernel least-mean-square algorithm," *IEEE Transactions on Signal Processing*, vol. 56, no. 2, pp. 543–554, 2008.

[149] W. D. Parreira, J. C. M. Bermudez, C. Richard, and J.-Y. Tourneret, "Stochastic behavior analysis of the Gaussian kernel least-mean-square algorithm," *IEEE Transactions on Signal Processing*, vol. 60, no. 5, pp. 2208–2222, 2012.

[150] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *Advances in Neural Information Processing Systems Workshops*, 2017.

[151] Y. LeCun, C. Cortes, and E. J. Burges, *The MNIST Database of Handwritten Digits*, 1998.

[152] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Tech. Rep., University of Toronto, 2009.

[153] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of International Conference on Machine Learning*, 2015, pp. 448–456.

[154] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[155] S. Han, J. Pool, J. Tran, and W. J. Dally, "Learning both weights and connections for efficient neural networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 1135–1143.

[156] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, "Pruning filters for efficient ConvNets," in *Proceedings of International Conference on Learning Representations*, 2017.

[157] Y. He, J. Lin, Z. Liu, H. Wang, L.-J. Li, and S. Han, "AMC: AutoML for model compression and acceleration on mobile devices," in *Proceedings of European Conference on Computer Vision*, 2018, pp. 784–800.

[158] T. Jiang, X. Yang, Y. Shi, and H. Wang, "Layer-wise deep neural network pruning via iteratively reweighted optimization," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2019, pp. 5606–5610.

[159] A. Aghasi, A. Abdi, N. Nguyen, and J. Romberg, "Net-Trim: Convex pruning of deep neural networks with performance guarantee," in *Advances in Neural Information Processing Systems*, 2017, pp. 3177–3186.

[160] W. Gao and J. Chen, "Transient performance analysis of zero-attracting gaussian kernel LMS algorithm with pre-tuned dictionary," *IEEE Access*, vol. 7, pp. 135770–135779, 2019.

[161] W. Liu, J. C. Principe, and S. Haykin, *Kernel adaptive filtering: a comprehensive introduction*, vol. 57, Hoboken, NJ, USA: John Wiley & Sons, 2011.

[162] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *Journal of Machine Learning Research*, vol. 5, no. Nov, pp. 1457–1469, 2004.

[163] J. A. K. Suykens, J. De Brabanter, L. Lukas, and J. Vandewalle, "Weighted least squares support vector machines: robustness and sparse approximation," *Neurocomputing*, vol. 48, no. 1-4, pp. 85–105, 2002.

[164] S. Hochreiter and J. Schmidhuber, "Flat minima," *Neural Computation*, vol. 9, no. 1, pp. 1–42, 1997.