AUGMENTED LAGRANGIAN BASED ALGORITHMS FOR CONVEX OPTIMIZATION PROBLEMS WITH NON-SEPARABLE ℓ_1-REGULARIZATION

GONG ZHENG

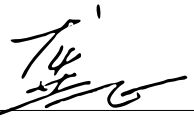
(B.Sc., NUS, Singapore)

A THESIS SUBMITTED
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
DEPARTMENT OF MATHEMATICS
NATIONAL UNIVERSITY OF SINGAPORE
2013

DECLARATION

I hereby declare that the thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.



Gong, Zheng

23 August, 2013

To my parents

Acknowledgements

The effort and time that my supervisor Professor Toh Kim-Chuan has spent on me throughout the five-year endeavor indubitably deserves more than a simple word "thanks". His guidance has been constantly ample in each stage of the preparation of this thesis, from mathematical proofs, algorithms design to numerical results analysis, and extends to paper writing. I have learned a lot from him, and this is not only limited to scientific ideas. His integrity and enthusiasm for research are communicative, and working with him has been a true pleasure for me.

My deepest gratitude also goes to Professor Shen Zuowei, my co-supervisor and perhaps more worthwhile to mention, my first guide to academic research. I always remember my first research project done with him as a third year undergraduate for his graduate course in Wavelets. It was challenging, yet motivating, and thus, led to where I am now. It has been my great fortune to have the opportunity to work with him again during my Ph.D. studies. The discussions in his office every Friday afternoon have been extremely inspiring and helpful.

I am equally indebted to Professor Sun Defeng, who has included me in his research seminar group and treated me as his own student. I have benefited greatly from the weekly seminar discussions throughout the five years, as well as his Conic Programming course. His deep understanding and great experience in optimization and nonsmooth analysis have been more than helpful in building up the theoretical aspect of this thesis. His kindness and generosity are exceptional. I feel very grateful and honored to be invited to his family parties almost every year.

It has been my privilege to be a member in both the optimization group and

the wavelets and signal processing group, which have provided me a great source of knowledge and friendship. Many thanks to Professor Zhao Gongyun, Zhao Xinyuan, Liu Yongjing, Wang Chengjing, Li Lu, Gao Yan, Ding Chao, Miao Weimin, Jiang Kaifeng, Wu Bin, Shi Dongjian, Yang Junfeng, Chen Caihua, Li Xudong and Du Mengyu in the optimization group; and Professor Ji Hui, Xu Yuhong, Hou Likun, Li Jia, Wang Kang, Bao Chenglong, Fan Zhitao, Wu Chunlin, Xie Peichu and Heinecke Andreas in the wavelets and signal processing group. Especially, Chao, Weimin, Kaifeng and Bin, I am sincerely grateful to your dedication for the weekly reading seminar of Convex Analysis, which lasted for more than two years and is absolutely the most memorable experience among all the others.

This acknowledgement will remain incomplete without expressing my gratitude to some of my other fellow colleagues and friends at NUS, in particular, Cai Yongyong, Ye Shengkui, Gao Bin, Ma Jiajun, Gao Rui, Zhang Yongchao, Cai Ruilun, Xue Hansong, Sun Xiang, Wang Fei, Jiao Qian, Shi Yan and Gu Weijia, for their friendship, (academic) discussions and of course, the (birthday) gatherings and chit-chats.

I am also thankful to the university and the department for providing me the full scholarship to complete the degree and the financial support for conference trips. Last but not least, thanks to all the administrative and IT staff for their consistent help during the past years.

Finally, they will not read this thesis, nor do they even read English, yet this thesis is dedicated to them, my parents, for their unfaltering love and support.

Gong, Zheng
August, 2013

Contents

Acknowledgements **vii**

Summary **xi**

1 Introduction **1**

1.1 Motivations and Related Methods 1

1.1.1 Sparse Structured Regression 2

1.1.2 Image Restoration 3

1.1.3 Limitations of the Existing First-order Methods 4

1.2 Contributions 5

1.3 Thesis Organization 6

2 Preliminaries **9**

2.1 Monotone Operators and The Proximal Point Algorithm 9

2.2 Basics of Nonsmooth Analysis 11

2.3 Tight Wavelet Frames 12

2.3.1 Tight Wavelet Frames Generated From MRA 15

2.3.2 Decomposition and Reconstruction Algorithms 16

3 A Semismooth Newton-CG Augmented Lagrangian Algorithm **19**

3.1 Reformulation of (1.1) 20

3.2 The General Augmented Lagrangian Framework 22
3.3 An Inexact Semismooth Newton Method for Solving (3.8) 23
3.4 Convergence of the Inexact SSNCG Method 26
3.5 The SSNAL Algorithm and Its Convergence 32
3.6 Extensions 38

4 First-order Methods **41**
4.1 Alternating Direction Method of Multipliers 41
4.2 Inexact Accelerated Proximal Gradient Method 44
4.3 Smoothing Accelerated Proximal Gradient Method 45

5 Applications of (1.1) in Statistics **49**
5.1 Sparse Structured Regression Models 49
5.2 Results on Random Generated Data 52
5.2.1 Fused Lasso 53
5.2.2 Clustered Lasso 54

6 Applications of (1.1) in Image Processing **61**
6.1 Image Restorations 61
6.2 Results on Image Restorations with Mixed Noises 63
6.2.1 Synthetic Image Denoising 65
6.2.2 Real Image Denoising 69
6.2.3 Image Deblurring with Mixed Noises 71
6.2.4 Stopping Criteria 75
6.3 Comparison with Other Models on Specified Noises 77
6.3.1 Denoising 78
6.3.2 Deblurring 86
6.3.3 Recovery from Images with Randomly Missing Pixels 87
6.4 Further Remarks 89
6.4.1 Reduced Model 89
6.4.2 ALM-APG versus ADMM 94

Bibliography **97**

Summary

This thesis is concerned with the problem of minimizing the sum of a convex function f and a non-separable ℓ_1-regularization term. The motivation for this work comes from recent interests in various high-dimensional sparse feature learning problems in statistics, as well as from problems in image processing. We present those problems under the unified framework of convex minimization with nonseparable ℓ_1-regularization, and propose an inexact semi-smooth Newton augmented Lagrangian (SSNAL) algorithm to solve an equivalent reformulation of the problem. Comprehensive results on the global convergence and local rate of convergence of the SSNAL algorithm are established, together with the characterization of the positive definiteness of the generalized Hessian of the objective function arising in each subproblem of the algorithm.

For the purpose of exposition and comparison, we also summarize/design three first-order methods to solve the problem under consideration, namely, the alternating direction method of multipliers (ADMM), the inexact accelerated proximal gradient (APG) method and the smoothing accelerated proximal gradient (SAPG) method.

Numerical experiments show that the SSNAL algorithm performs favourably in comparison to several state-of-the-art first-order algorithms for solving fused lasso problems, and outperforms the best available algorithms for clustered lasso problems.

With the available numerical methods, we propose a simple model to solve various image restoration problems in the presence of mixed or unknown noises. The proposed model essentially takes the weighted sum of ℓ_1 and ℓ_2-norm based distance

functions as the data fitting term and utilizes the sparsity prior of images in wavelet tight frame domain. Since a moderately accurate result is usually sufficient for image restoration problems, an augmented Lagrangian method (ALM) with the inner subproblem being solved by an accelerated proximal gradient (APG) algorithm is used to solve the proposed model.

The numerical simulation results show that the performance of the proposed model together with the numerical algorithm is surprisingly robust and efficient in solving several image restoration problems, including denoising, deblurring and inpainting, in the presence of both additive and non-additive noises or their mixtures. This single one-for-all fitting model does not depend on any prior knowledge of the noise. Thus, it has the potential of performing effectively in real color image denoising problems, where the noise type is difficult to model.

Chapter 1

Introduction

In this thesis, we focus on solving minimization problems of the following form:

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \quad f(\boldsymbol{x}) + \rho\|B\boldsymbol{x}\|_1, \tag{1.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a convex and twice continuously differentiable function, $B \in \mathbb{R}^{p\times n}$ is a given matrix, and ρ is a given positive parameter. For any $\boldsymbol{x} \in \mathbb{R}^n$, we denote its 2-norm by $\|\boldsymbol{x}\|$, and let $\|\boldsymbol{x}\|_1 = \sum_{i=1}^n |\boldsymbol{x}_i|$. We assume that objective function in (1.1) is coercive and *hence the optimal solution set of* (1.1) *is nonempty and bounded.*

1.1 Motivations and Related Methods

As the ℓ_1-norm regularization term encourages sparsity in the optimal solution, the special case of the problem (1.1) when $f(\boldsymbol{x}) = \frac{1}{2}\|A\boldsymbol{x} - \boldsymbol{b}\|^2$ and $B = I$, i.e.

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \ \frac{1}{2}\|A\boldsymbol{x} - \boldsymbol{b}\|^2 + \rho\|\boldsymbol{x}\|_1 \tag{1.2}$$

has drawn particular attention in both signal processing (basis pursuit [24]) and statistics (lasso [99]) communities since almost twenty years ago. Due to the separability of $\|\boldsymbol{x}\|_1$ and the simple structure of squared loss term, a great variety of algorithms have been designed to solve the problem (1.2). Ever since the compressed sensing theory in the context of signal processing has established the theoretical guarantee for stable recovery of the original sparse signal by solving (1.2) under

certain conditions [19, 35], the problem (1.2) has regained immense interest among the signal processing, statistics and optimization communities during the recent ten years.

Here we briefly describe some of the methods available for solving (1.2). These methods mainly fall into three broad categories. (1) first-order methods [6, 45, 53, 103, 106, 108], which are specifically designed to exploit the separability of $\|\boldsymbol{x}\|_1$ to ensure that a certain subproblem at each iteration admits an analytical solution. These methods have been very successful in solving large scale problems where A satisfies certain restricted isometry property [20], which ensures that the Hessian A^TA is well conditioned on the subspace corresponding to the non-zero components of the optimal $\boldsymbol{x}^*$; (2) homotopy-type methods [36, 38], which attempt to solve (1.2) by sequentially finding the break-points of the solution $\boldsymbol{x}(\rho)$ of (1.2) starting from the initial parameter value $\|A^T\boldsymbol{b}\|_\infty$ and ending with the desired target value. These methods rely on the property that each component of the solution $\boldsymbol{x}(\rho)$ of (1.2) is a piece-wise linear function; (3) inexact interior-point methods [24, 46, 60], which solve a convex quadratic programming reformulation of (1.1). The literature on algorithms for solving (1.2) is vast and here we only mention those that are known to be the most efficient. We refer the reader to the recent paper [46] for more details on the relative performance and merits of various algorithms. Numerical experiments have shown that first-order methods are generally quite efficient if one requires only a moderately accurate approximate solution for large scale problems. More recently, the authors in [9] have proposed an active-set method using the semismooth Newton framework to solve (1.2) by reformulating it as a bound constrained convex quadratic programming problem.

However, many applications require one to solve the general problem (1.1) where f is non-quadratic and/or the regularization term is non-separable, such as various extensions of the ℓ_1-norm lasso penalty and regression models with loss functions other than the least-squared loss; total variation (TV) regularized image restoration models, etc. Most of the algorithms mentioned in the last paragraph are specifically designed to exploit the special structure of (1.2), and as a result, they are either not applicable or become very inefficient when applied to (1.1).

1.1.1 Sparse Structured Regression

One of the main motivations for studying the problem (1.1) comes from high dimensional regression models with structured sparse regularizations, such as group lasso [107, 109], fused lasso [100], clustered lasso [78, 90], OSCAR [7], etc. In these statistical applications, $f(\boldsymbol{x})$ is the data fitting term (known as the loss function), and B is typically structured or sparse.

Efficient first-order algorithms that exploit the special structures of the corresponding regularization terms have been developed for different structured lasso problems. For example, proximal gradient methods have been designed in [5, 74] for non-overlapping grouped lasso problems, and coordinate descent methods [47] and accelerated proximal gradient based methods [65] have been proposed for fused lasso problems with quadratic loss function. Unfortunately, there are many more complex structured lasso problems such as overlapping grouped lasso, graph-guided fused lasso, clustered lasso etc, for which the aforementioned first-order algorithms are not applicable.

Although the problem (1.1) with a quadratic loss function can always be formulated as a second-order cone programming (SOCP) problem or a convex quadratic programming (QP) problem which are solvable by interior-point solvers such as [101] or [98], the high computational cost and limitation in the scale of the problem solvable usually prohibit one from doing so, especially when the problem is large.

1.1.2 Image Restoration

Image restoration is another major area that give rises to problems of the form (1.1), where f is typically the quadratic loss function.

In TV-regularized image restoration (original introduced by Rudin, Osher and Fatemi [88]), the regularization term is essentially the ℓ_1-norm of the first-order forward difference of $\boldsymbol{x}$ in the one-dimensional case, which is a non-separable ℓ_1-term similar to the fused lasso regularization term. With f being a quadratic loss function as in (1.2), the authors in [75] considered half-quadratic reformulations of (1.1) and applied alternating minimization methods to solve the reformulated problems. In [56, 102], the authors independently developed some alternating minimization algorithms for some types of TV image restoration problems. We should mention here that those alternating minimization methods only solve an approximate version (by

smoothing the TV-term) of the original problem (1.1), and hence the approximate solution obtained is at best moderately accurate for (1.1). More recently, [104] proposed to use the alternating direction method of multipliers (ADMM) to solve the original TV-regularized problem (1.1) with quadratic loss, and demonstrated very good numerical performance of the ADMM for such a problem.

In frame based image restoration, since the wavelet tight frame systems are redundant, the mapping from the image to its coefficients is not one-to-one, i.e., the representation of the image in the frame domain is not unique. Therefore, based on different assumptions, there are three formulations for the sparse approximation of the underlying image, namely, the analysis based approach, the synthesis based approach and the balanced approach. The analysis based approach proposed in [39,96] assumes that the coefficient vector can be sparsely approximated; therefore, it is formulated as the general problem (1.1) with a non-separable ℓ_1-regularization, where B is the framelet decomposition operator. The synthesis based approach introduced in [31, 41–44] and the balanced approach first used in [21, 22] assume that the underlying image is synthesized from some sparse coefficient vector via the framelet reconstruction operator; therefore, the models directly penalize the ℓ_1-norm of the coefficient vector, which leads to the special separable case (1.2). The proximal forward-backward splitting (PFBS) algorithm was first used to solve the synthesis based model in [29, 31, 41–44] (also known as the iterative shrinkage/thresholding (IST) algorithm), and the balanced model in [12–14, 18]. Later, a linearized Bregman algorithm was designed to solve the synthesis based model in [16], and an APG algorithm was proposed to solve the balanced model in [92], both of which demonstrated faster convergence than the PFBS (IST) algorithm. For the analysis based approach, where a non-separable ℓ_1 term is involved, the split Bregman iteration was used to develop a fast algorithm in [17]. It was later observed that the resulted split Bregman algorithm is equivalent to the ADMM mentioned previously.

1.1.3 Limitations of the Existing First-order Methods

To summarize, first-order methods have been very popular for structured convex minimization problems (especially those with the simple regularization term $\|\boldsymbol{x}\|_1$) arising from statistics, machine learning, and image processing. In those applications, the optimization models are used to serve as a guide to obtain a good feasible solution to the underlying application problems and the goal is not necessarily to

compute the optimal solutions of the optimization models. As a result, first-order methods are mostly adequate for many such application problems since the required accuracy (with respect to the optimization model) of the computed solution is rather modest. Even then, the efficiency of first-order methods are heavily dependent on the structures of the particular problem they are designed to exploit. To avoid having a multitude of first-order algorithms each catering to a particular problem structure, it is therefore desirable to design an algorithm which can efficiently be applied to (1.1), and its efficiency is not completely dictated by the particular problem structure on hand, while at the same time it is able to deliver a high accuracy solution when required.

For the general problem (1.1), so far there is no single unifying algorithmic framework that has been demonstrated to be efficient and robust for solving the problem. Although some general first-order methods (derived from the ADMM [37] and accelerated proximal gradient methods [73], [5]) are available for solving (1.1), their practical efficiency are highly dependent on the problem structure of (1.1), especially on the structure of the nonseparable ℓ_1-term $\|B\boldsymbol{x}\|_1$. One can also use the commonly employed strategy of approximating the non-smooth term $\|B\boldsymbol{x}\|_1$ by some smooth surrogates to approximately solve (1.1). Indeed, this has been done in [27], which proposed to use the accelerated proximal gradient method in [5] to solve smoothed surrogates of some structured lasso problems. But the efficiency of such an approach has yet to be demonstrated convincingly. A detailed discussion on those first-order methods will be given in Chapter 3.

Above all, the main purpose of this work is to design a unifying algorithmic framework (semismooth Newton augmented Lagrangian (SSNAL)) for solving (1.1), which does not depend heavily on the structure of $\|B\boldsymbol{x}\|_1$. Unlike first-order methods, our SSNAL based algorithm exploits second-order information of the problem to achieve high efficiency for computing accurate solutions of (1.1).

1.2 Contributions

The main contributions of this thesis are three-folds. First, we provide a unified algorithmic framework for a wide variety of ℓ_1-regularized (not necessarily separable) convex minimization problems that have been studied in the literature. The algorithm we developed is a semismooth Newton augmented Lagrangian (SSNAL)

method applied to (1.1), where the inner subproblem is solved by a semismooth Newton method for which the linear system in each iteration is solved by a preconditioned conjugate gradient method. An important feature of our algorithm is that its efficiency does not depend critically on the separability of the ℓ_1-term in contrast to many existing efficient methods. Also, unlike many existing algorithms which are designed only for quadratic loss functions, our algorithm can handle a wide variety of convex loss functions. Moreover, based the general convergence theory of the ALM [84, 85], we are able to provide comprehensive global and local convergence results for our algorithm. Second, our algorithm can solve (1.1) and its dual simultaneously, and hence there is a natural stopping criterion based on duality theory (or the KKT conditions). Third, our algorithm utilizes second-order information and hence it can obtain accurate solutions much more efficiently than first-order methods for (1.1) but at the same time it is competitive to state-of-the-art first-order algorithms (for which a high accuracy solution may not be achievable) for solving large scale problems. We evaluate our algorithm and compare its performance with state-of-the-art algorithms for solving the fused lasso and clustered lasso problems.

In addition, we propose a simple model for image restoration with mixed or unknown noises. While most of the existing methods for image restorations are designed specifically for a given type of noise, our model appears to be the first versatile model for handling image restoration with various mixed noises and unknown type of noises. This feature is particularly important for solving real life image restoration problems, since, under various constraints, images are always degraded with mixed noise and it is impossible to determine what type of noise is involved. The proposed model falls in the framework of the general non-separable ℓ_1-regularized problem (1.1). Since a moderately accurate solution is usually sufficient for image processing problems, we use an accelerated proximal gradient (APG) algorithm to solve the inner subproblem. The simulations on synthetic data show that our method is effective and robust in restoring images contaminated by additive Gaussian noise, Poisson noise, random-valued impulse noise, multiplicative Gamma noise and mixtures of these noises. Numerical results on real digital colour images are also given, which confirms the effectiveness and robustness of our method in removing unknown noises.

1.3 Thesis Organization

The rest of the thesis is organized as follows. In Chapter 2, we present some preliminaries that relate to the subsequent discussions. We first introduce the idea of monotone operators and the proximal point algorithm. The augmented Lagrangian method is essentially the dual application of the proximal point algorithm. Secondly, some basic concepts in nonsmooth analysis will be provided. The convergence of the SSNAL algorithm proposed here relies on the semismoothness of the projection operator (onto an ℓ_∞-ball). Finally, a brief introduction on tight wavelet frames will be given, which includes (1) the multiresolution analysis (MRA) based tight frames derived from the unitary extension principle; (2) the fast algorithms for framelet decomposition and reconstruction. All of the applications on image restoration problems presented in this thesis are, but not limited to, under the assumption that the images are sparse in the tight wavelet frame domain.

In Chapter 3, we first reformulate the original unconstrained problem (1.1) to an equivalent constrained one, and build up the general augmented Lagrangian framework. Then we propose an inexact semismooth Newton augmented Lagrangian (SSNAL) algorithm to solve this reformulated constrained problem. We also characterize the condition when the generalized Hessian of the objective function is positive definite, and provide the convergence analysis of the proposed SSNAL algorithm. Finally, the extensions of the SSNAL framework for solving some generalizations of (1.1) are described.

We summarize/design some first-order algorithms which are promising for solving the general problem (1.1) in Chapter 4. Although the computational efficiency of these first-order methods depends crucially on the problem structures of (1.1), our SSNAL algorithm can always capitalize on the strength (of rapid initial progress) of first-order methods for generating a good starting point to warm-start the algorithm.

Chapter 5 is devoted to the application of the SSNAL algorithm to solve the structured lasso problems of major concern among the statistics community. We first introduce the various sparse structured regression models and discuss how they can be fitted into our unified framework. The numerical performance of our SSNAL algorithm for fused lasso and clustered lasso problems on randomly generated data, as well as the comparison with other state-of-the-art algorithms is presented.

In Chapter 6, we propose a simple model for image restoration with mixed or unknown noises. The numerical results for various image restorations with mixed

noise and examples on noise removal of real digital colour images are presented. While there is no result for image restorations with such a wide range of mixed noise available in the literature as far as we are aware of, comparisons with some of the available models for removing noises such as single type of noise, mixed Poisson-Gaussian noise, and impulse noise mixed with Gaussian noise are given. Some additional remarks on our proposed model and numerical algorithm will be addressed.

Chapter 2

Preliminaries

In this chapter, we present some preliminaries that relate to the subsequent discussions. We first introduce the idea of monotone operators and the proximal point algorithm. The augmented Lagrangian method (ALM) is essentially the dual application of the proximal point algorithm. Secondly, some basic concepts in nonsmooth analysis will be provided. The convergence of the SSNAL algorithm proposed here relies on the semismoothness of the projection operator (onto an ℓ_∞-ball). Finally, a brief introduction on tight wavelet frames will be given, which includes (1) the multiresolution analysis (MRA) based tight frames derived from the unitary extension principle; (2) the fast algorithms for framelet decomposition and reconstruction. The proposed simple model for image restoration with mixed and unknown noises is based on, but not limited to, the assumption that the images are sparse in the tight wavelet frame domain.

2.1 Monotone Operators and The Proximal Point Algorithm

Let $\boldsymbol{H}$ be a real Hilbert space with inner product $\langle \cdot, \cdot \rangle$. A multifunction $\mathcal{T} : \boldsymbol{H} \to \boldsymbol{H}$ is said to be a *monotone operator* if

$$\langle z - z',\, w - w' \rangle \geq 0 \quad \text{whenever} \quad w \in \mathcal{T}(z),\ w' \in \mathcal{T}(z').$$

It is said to be *maximal monotone* if, in addition, the graph

$$G(\mathcal{T}) = \{(z, w) \in \boldsymbol{H} \times \boldsymbol{H} | w \in \mathcal{T}(z)\}$$

is not properly contained in the graph of any other monotone operator $\mathcal{T}' : \boldsymbol{H} \to \boldsymbol{H}$.

Such operators have been studied extensively for their important role in convex analysis. A fundamental problem is that of determining an element z such that $0 \in \mathcal{T}(z)$. For example, the subdifferential mapping ∂f of a proper closed convex function f is maximal monotone, and the inclusion $0 \in \partial f(z)$ means that $f(z) = \min f$. The problem is then one of minimization subject to implicit constraints.

A fundament algorithm for solving $0 \in \mathcal{T}(z)$ in the case of an arbitrary maximal monotone operator $\mathcal{T}$ is based on the fact that for each $z \in \boldsymbol{H}$ and $c > 0$ there is a unique $u \in \boldsymbol{H}$ such that $z - u \in c\mathcal{T}(u)$, i.e. $z \in (I + c\mathcal{T})(u)$ [70]. The operator $\mathcal{P} := (I + c\mathcal{T})^{-1}$ is therefore single-valued from all of $\boldsymbol{H}$ to $\boldsymbol{H}$. It is also *nonexpansive*:

$$\|\mathcal{P}(z) - \mathcal{P}(z')\| \leq \|z - z'\|, \tag{2.1}$$

and one has $\mathcal{P}(z) = z$ if and only if $0 \in \mathcal{T}(z)$. $\mathcal{P}$ is called the *proximal mapping* associated with $c\mathcal{T}$, following the terminology of Moreau [71] for the case of $\mathcal{T} = \partial f$.

The *proximal point algorithm* generates for any starting point z^0 a sequence $\{z^k\}$ in $\boldsymbol{H}$ by the approximate rule

$$z^{k+1} \approx \mathcal{P}_k(z^k), \quad \text{where} \quad \mathcal{P}_k = (I + c_k\mathcal{T})^{-1}. \tag{2.2}$$

Here $\{c_k\}$ is some sequence of positive real numbers. In the case of $\mathcal{T} = \partial f$, this procedure reduces to

$$z^{k+1} \approx \arg\min_z \left\{ f(z) + \frac{1}{2c_k}\|z - z^k\|^2 \right\}.$$

In [85], Rockafellar introduced the following two general criteria for the approximate calculation of $\mathcal{P}_k(z^k)$:

$$\|z^{k+1} - \mathcal{P}_k(z^k)\| \leq \varepsilon_k, \quad \sum_{k=0}^{\infty} \varepsilon_k < \infty, \tag{2.3}$$

$$\|z^{k+1} - \mathcal{P}_k(z^k)\| \leq \delta_k \|z^{k+1} - z^k\|, \quad \sum_{k=0}^{\infty} \delta_k < \infty. \tag{2.4}$$

He proved that under very mild assumptions that for any starting point z^0, the criterion (2.3) guarantees weak convergence of $\{z^k\}$ to a particular solution z^∞ to $0 \in \mathcal{T}(z)$. In general, the set of all such points z forms a closed convex set in $\boldsymbol{H}$, denoted by $\mathcal{T}^{-1}(0)$. If in addition, the criterion (2.4) is also satisfied and $\mathcal{T}^{-1}$ is Lipschitz continuous at 0, then it can be shown that the convergence is at least at a linear rate, where the modulus can be brought arbitrarily close to zero by taking c_k large enough. If $c_k \to \infty$, one has superlinear convergence.

Note that $\mathcal{T}^{-1}$ is *Lipschitz continuous* at 0 with modulus $a \geq 0$ if there is a unique solution $\bar{z}$ to $0 \in \mathcal{T}(z)$, i.e. $\mathcal{T}^{-1}(0) = \{\bar{z}\}$, and for some $\tau > 0$, we have

$$\|z - \bar{z}\| \leq a\|w\| \quad \text{whenever} \quad z \in \mathcal{T}^{-1}(w) \text{ and } \|w\| \leq \tau.$$

This assumption could be fulfilled very naturally in applications to convex programming, for instance, under certain standard second-order conditions characterizing a "nice" optimal solution (see [84] for detailed discussions).

There are three distinct types of applications of the proximal point algorithm in convex programming: (1) to $\mathcal{T} = \partial f$, where f is the objective function in the primal problem; (2) to $\mathcal{T} = -\partial g$, where g is the concave objective function in the dual problem, and (3) to the monotone operator corresponding to the convex-concave Lagrangian function. The augmented Lagrangian method that will be discussed further in Chapter 3 actually corresponds to the second application.

2.2 Basics of Nonsmooth Analysis

Let $\mathbf{X}$ and $\mathbf{Y}$ be two finite-dimensional real Hilbert spaces. Let $\mathcal{O}$ be an open set in $\mathbf{X}$ and $f : \mathcal{O} \subseteq \mathbf{X} \to \mathbf{Y}$ be a locally Lipschitz continuous function on the open set $\mathcal{O}$. Then f is almost everywhere F(réchet)-differentiable by Rademacher's

theorem. Let $\mathcal{D}_f$ denote the set of F-differentiable points of f in $\mathcal{O}$. Then the B(ouligand)-subdifferential of f at $x \in \mathcal{O}$, denoted by $\partial_B f(x)$, is

$$\partial_B f(x) := \left\{ \lim_{k\to\infty} f'(x^k) \mid x^k \in \mathcal{D}_f, x^k \to x \right\},$$

and the Clarke's generalized Jacobian [28] at x is the convex hull of $\partial_B f(x)$, i.e.

$$\partial f(x) = \operatorname{conv} \{\partial_B f(x)\}.$$

In addition, f is said to be directionally differentiable at x if for any $\Delta x \in \mathbf{X}$, the directional derivative of f at x along Δx, denoted by $f'(x; \Delta x)$ exists.

Definition 2.2.1. Let $f : \mathcal{O} \subseteq \mathbf{X} \to \mathbf{Y}$ be a locally Lipschitz continuous function on the open set $\mathcal{O}$. We say that f is *semismooth* at a point $x \in \mathcal{O}$ if

(i) f is directionally differentiable at x; and

(ii) for any $\Delta x \in \mathbf{X}$ and $V \in \partial f(x + \Delta x)$ with $\Delta x \to 0$,

$$f(x + \Delta x) - f(x) - V(\Delta x) = o(\|\Delta x\|). \tag{2.5}$$

Furthermore if (2.5) is replaced by

$$f(x + \Delta x) - f(x) - V(\Delta x) = O(\|\Delta x\|^2), \tag{2.6}$$

then f is said to be *strongly semismooth* at x.

Semismoothness was originally introduced by Mifflin [69] for functionals. Qi and Sun [81] extended the concept to vector valued functions.

2.3 Tight Wavelet Frames

We introduce the notion of tight wavelet frames in space $L_2(\mathbb{R})$, as well as some other basic concepts and notation. The space $L_2(\mathbb{R})$ is the set of all functions $f(x)$ satisfying $\|f\|_{L_2(\mathbb{R})} := (\int_{\mathbb{R}} |f(x)|^2 dx)^{1/2} < \infty$, and the space $\ell_2(\mathbb{Z})$ is the set of all sequences h defined on $\mathbb{Z}$ satisfying $\|h\|_{\ell_2(\mathbb{Z})} := (\sum_{k\in\mathbb{Z}} |h[k]|^2)^{1/2} < \infty$.

For any function $f \in L_2(\mathbb{R})$, the dyadic dilation operator D is defined by $Df(x) := \sqrt{2} f(2x)$ and the translation operator T is defined by $T_a f(x) := f(x - a)$

for $a \in \mathbb{R}$. Given $j \in \mathbb{Z}$, we have $T_a D^j = D^j T_{2^j a}$.

For given $\Psi := \{\psi_1, ..., \psi_r\} \subset L_2(\mathbb{R})$, define the wavelet system by

$$X(\Psi) := \{\psi_{\ell,j,k} : 1 \leq \ell \leq r;\ j,k \in \mathbb{Z}\},$$

where $\psi_{\ell,j,k} = D^j T_k \psi_\ell = 2^{j/2}\psi_\ell(2^j \cdot -k)$. The system $X(\Psi)$ is called a tight wavelet frame of $L_2(\mathbb{R})$ if

$$\|f\|^2_{L_2(\mathbb{R})} = \sum_{g \in X(\Psi)} |\langle f, g\rangle|^2$$

holds for all $f \in L_2(\mathbb{R})$, where $\langle \cdot, \cdot \rangle$ is the inner product in $L_2(\mathbb{R})$ and $\|\cdot\|_{L_2(\mathbb{R})} = \sqrt{\langle \cdot, \cdot \rangle}$. This is equivalent to $f = \sum_{g \in X(\Psi)} \langle f, g\rangle g$, for all $f \in L_2(\mathbb{R})$.

Note that when $X(\Psi)$ forms an orthonormal basis of $L_2(\mathbb{R})$, it is called an orthonormal wavelet basis. It is clear that an orthonormal basis is a tight frame.

The Fourier transform of a function $f \in L_1(\mathbb{R})$ is usually defined by

$$\widehat{f}(\omega) := \int_{\mathbb{R}} f(x) e^{-i\omega x} dx, \quad \omega \in \mathbb{R},$$

and then, the corresponding inverse is

$$f(x) = \frac{1}{2\pi} \int_{\mathbb{R}} \widehat{f}(\omega) e^{i\omega x} d\omega, \quad x \in \mathbb{R}.$$

They can be extended to more general functions, e.g. the functions in $L_2(\mathbb{R})$. Similarly, we can define the Fourier series for a sequence $h \in \ell_2(\mathbb{Z})$ by

$$\widehat{h}(\omega) := \sum_{k \in \mathbb{Z}} h[k] e^{-ik\omega}, \quad \omega \in \mathbb{R}.$$

To characterise the wavelet system $X(\Psi)$ to be a tight frame or even an orthonormal basis for $L_2(\mathbb{R})$ in terms of its generators Ψ, the dual Gramian analysis [86] is used in [87].

Theorem 2.3.1. The wavelet system $X(\Psi)$ is a tight frame of $L_2(\mathbb{R})$ if and only if the identities

$$\sum_{\psi \in \Psi} \sum_{k \in \mathbb{Z}} |\widehat{\psi}(2^k \xi)|^2 = 1; \quad \sum_{\psi \in \Psi} \sum_{k=0}^{\infty} \widehat{\psi}(2^k \xi) \overline{\widehat{\psi}(2^k(\xi + (2j+1)2\pi))} = 0,\ j = \mathbb{Z} \qquad (2.7)$$

hold for a.e. $\xi \in \mathbb{R}$. Furthermore, $X(\Psi)$ is an orthonormal basis of $L_2(\mathbb{R})$ if and only if (2.7) holds and $\|\psi\| = 1$ for all $\psi \in \Psi$.

Although Theorem 2.3.1 gives a complete characterization of the wavelet system $X(\Psi)$ being a tight frame of $L_2(\mathbb{R})$, it is only helpful for obtaining tight wavelet frame systems from bandlimited generators. In order to construct wavelet systems with compactly supported generators, here we adopt the multiresolution analysis (MRA) structure proposed in [8], which is more general than the original version proposed by Mallat and Meyer in [67, 68].

For a given function $\phi \in L_2(\mathbb{R})$, define the *shift-invariant subspace* $V \subset L_2(\mathbb{R})$ generated by ϕ as

$$V := \overline{\operatorname{span}\{\phi(\cdot - k),\ k \in \mathbb{Z}\}},$$

and denote V_n as its 2^n-dilation:

$$V_n = \overline{\operatorname{span}\{\phi(2^n \cdot -k),\ k \in \mathbb{Z}\}}, \quad n \in \mathbb{Z}.$$

We have $V = V_0$. A subspace $S \subset L_2(\mathbb{R})$ is called *translation-invariant* if for any $t \in \mathbb{R}$ and $f \in S$, we have $f(\cdot - t) \in S$. The subspace S is called *s-shift-invariant* if for any $k \in \mathbb{Z}$ and $f \in S$, we have $f(\cdot - sk) \in S$, and in particular if $s = 1$, we call S a *shift-invariant* subspace.

Now for a given sequence of subspaces $\{V_n\}_{n\in\mathbb{Z}}$, we say $\{V_n\}$ forms a *multiresolution analysis* (MRA) for $L_2(\mathbb{R})$, if the following conditions are satisfied:

$$V_n \subset V_{n+1},\ n \in \mathbb{Z};$$

$$\overline{\cup_n V_n} = L_2(\mathbb{R});$$

$$\cap_n V_n = \{0\}.$$

Then ϕ is called the *generator* of the MRA.

Finally, for any given $\phi \in L_2(\mathbb{R})$ that generates an MRA $\{V_n\}_n$, the *quasi-interpolatory operator* is defined as

$$P_n : f \mapsto \sum_{k\in\mathbb{Z}} \langle f, \phi_{n,k}\rangle \phi_{n,k}, \tag{2.8}$$

for any arbitrary $f \in L_2(\mathbb{R})$ with $\phi_{n,k} := D^n T_k \phi$.

The MRA generated tight wavelet frame systems is particularly useful in practice because it has fast decomposition and reconstruction algorithms. In the following, we first describe how tight wavelet frames are explicitly constructed based on MRA generated by a refinable function ϕ via the unitary extension principle (UEP) [87]; then, we provide the details of the decomposition and reconstruction algorithms for the MRA-based tight wavelet frames.

2.3.1 Tight Wavelet Frames Generated From MRA

We are interested in constructing compactly supported wavelet systems with finitely supported masks. Therefore, assume further that ϕ is a compactly supported refinable function. Note that a compactly supported function $\phi \in L_2(\mathbb{R})$ is refinable if it satisfies the refinement equation

$$\phi(x) = 2\sum_{k\in\mathbb{Z}} h_0[k]\phi(2x-k), \tag{2.9}$$

for some finitely supported sequence $h_0 \in \ell_2(\mathbb{Z})$. By taking the Fourier transform, equation (2.9) becomes

$$\widehat{\phi(2\cdot)} = \widehat{h_0}\widehat{\phi}, \quad \text{a.e. } \omega \in \mathbb{R}.$$

We call the sequence h_0 the refinement mask of ϕ and $\widehat{h_0}$ the refinement symbol of ϕ.

Let $\{V_n\}_{n\in\mathbb{Z}}$ be the MRA generated by the refinable function ϕ and the refinement mask h_0. Let $\Psi := \{\psi_1, ..., \psi_r\} \subset V_1$ be of the form

$$\psi_\ell(x) = 2\sum_{k\in\mathbb{Z}} h_\ell[k]\phi(2x-k). \tag{2.10}$$

The finitely supported sequences $h_1, ..., h_r$ are called wavelet masks, or the high pass filters of the system, and the refinement mask h_0 is called the low pass filter. In the Fourier domain, (2.10) can be written as

$$\widehat{\psi_\ell(2\cdot)} = \widehat{h_\ell}\widehat{\phi}, \quad \ell = 1, ..., r,$$

where $\widehat{h_1}, ..., \widehat{h_r}$ are 2π-periodic functions and are called wavelet symbols.

Theorem 2.3.2 (Unitary Extension Principle (UEP) [87])**.** Let $\phi \in L_2(\mathbb{R})$ be the compactly supported refinable function with its finitely supported refinement mask

h_0 satisfying $\widehat{h_0}(0) = 1$. Let $\{h_1, ..., h_r\}$ be a set of finitely supported sequences. Then the system $X(\Psi)$ where $\Psi = \{\psi_1, ..., \psi_r\}$ as defined in (2.10) forms a tight frame in $L_2(\mathbb{R})$ provided the equalities

$$\sum_{\ell=0}^{r} |\widehat{h_\ell}(\xi)|^2 = 1 \quad \text{and} \quad \sum_{\ell=0}^{r} \widehat{h_\ell}(\xi)\overline{\widehat{h_\ell}(\xi+\pi)} = 0 \tag{2.11}$$

hold for almost all $\xi \in [-\pi, \pi]$. Furthermore, assuming $r = 1$ and $\|\phi\|_{L_2(\mathbb{R})} = 1$, then $X(\Phi)$ is an orthonormal wavelet basis of $L_2(\mathbb{R})$.

Notice that the conditions in (2.11) can be written in terms of the sequences $\{h_0, ..., h_r\}$. The two conditions thus become

$$\sum_{\ell=0}^{r} \sum_{k\in\mathbb{Z}} \overline{h_\ell[k]} h_\ell[k-p] = \delta_{p,0}, \quad p \in \mathbb{Z},$$

where $\delta_{p,0} = 1$ when $p = 0$ and 0 otherwise, and

$$\sum_{\ell=0}^{r} \sum_{k\in\mathbb{Z}} (-1)^{k-p} \overline{h_\ell[k]} h_\ell[k-p] = 0, \quad p \in \mathbb{Z}.$$

The generators Ψ via the UEP is called framelet in [30]. The UEP provides the freedom to choose the number of the generators r in constructing the tight framelets, which allows one to construct tight framelets from, for examples, splines easily. In fact, [86] gives a systematic construction of tight wavelet frame systems from B-splines by using the UEP.

2.3.2 Decomposition and Reconstruction Algorithms

The decomposition and reconstruction algorithms for the MRA-based tight wavelet frames derived from the UEP are essentially the same as those of MRA-based orthonormal wavelets. Here, we assume that all masks used are finitely supported.

Since $P_L f = D^L P_0 D^{-L} f$, one may use $P_0 f \in V_0$ to approximate f without loss of generality. When a tight wavelet frame is used, the given data is considered to be sampled as local average $v[k] = \langle f, \phi(\cdot - k)\rangle$, which means

$$P_0 f = \sum_{k\in\mathbb{Z}} v[k]\phi(\cdot - k)$$

can be used to approximate the underlying function f.

Given the sequence $h_\ell = \{h_\ell[k]\}_{k\in\mathbb{Z}}$ for any $\ell = 0, 1, ..., r$, define an infinite matrix H_ℓ corresponding to h_ℓ as

$$\mathsf{H}_\ell := (\mathsf{H}_\ell[l,k]) := (\sqrt{2}\ \overline{h_\ell[k-2l]}),$$

where the (l,k)-th entry in H_ℓ is fully determined by the $(k-2l)$-th entry in h_ℓ. Then for any $v \in \ell_2(\mathbb{Z})$, we have

$$(\mathsf{H}_\ell v)[l] = \sqrt{2}\sum_{k\in\mathbb{Z}} \overline{h_\ell[k-2l]}v[k].$$

Similarly, the adjoint of H_ℓ, denote by H_ℓ^*, can be defined as

$$(\mathsf{H}_\ell^* v)[k] = \sqrt{2}\sum_{l\in\mathbb{Z}} h_\ell[k-2l]v[l].$$

Let us denote the downsampling operator as $\downarrow(\cdot)$, which is defined by

$$(\downarrow v)[k] = v[2k], \quad k \in \mathbb{Z};$$

and the upsampling operator as $\uparrow(\cdot)$, which is defined by

$$(\uparrow v)[k] = \begin{cases} v[k/2], & k \text{ even;} \\ 0, & k \text{ odd.} \end{cases}$$

Then, we have

$$\mathsf{H}_\ell v = \downarrow(\sqrt{2}\ \overline{h_\ell[-\cdot]} \otimes v) \quad \text{and} \quad \mathsf{H}_\ell^* v = \sqrt{2}h_\ell \otimes (\uparrow v).$$

The above notation based on convolution and (up)downsampling are the traditional notation used in the literature of wavelets.

It can be shown that

$$\sum_{\ell=0}^{r} \mathsf{H}_\ell^*\mathsf{H}_\ell = I,$$

which is the so-called perfect reconstruction property.

For multiple level decomposition, define W_L, $L < 0$ as a (rectangular) block

matrix:

$$W_L := [\mathsf{H}_0^L; \mathsf{H}_1\mathsf{H}_0^{L+1}; \ldots, \mathsf{H}_r\mathsf{H}_0^{L+1}; \ldots; \mathsf{H}_1; \ldots; \mathsf{H}_r]^T.$$

Then the reconstruction operator W_L^*, the adjoint operator of W_L, is given by

$$W_L^* = [\mathsf{H}_0^{*L}; \mathsf{H}_0^{*L+1}\mathsf{H}_1^*; \ldots, \mathsf{H}_0^{*L+1}\mathsf{H}_r^*; \ldots; \mathsf{H}_1^*; \ldots; \mathsf{H}_r^*]^T.$$

Similarly, we also have a multi-level perfect reconstruction formula $W_L^* W_L = I$.

The fast framelet decomposition and reconstruction algorithms are summarized as follows.

L-level Fast Framelet Decomposition and Reconstruction Algorithms
Given signal $v \in \mathbb{R}^N$ with N assumed to be an integer multiple of 2^L, $L \in \mathbb{Z}_+$. Denote $v_{0,0} = v$.

Decomposition: For each $j = 1, 2, \ldots, L$

(a) Obtain low frequency approximation to v at level j:

$$v_{0,j} =\downarrow (\sqrt{2}\ \overline{h_0[-\cdot]} \otimes v_{0,j-1});$$

(b) Obtain framelet coefficients of v at level j:

$$v_{\ell,j} =\downarrow (\sqrt{2}\ \overline{h_\ell[-\cdot]} \otimes v_{0,j-1}), \quad \ell = 1, 2, \ldots, r.$$

Reconstruction: For each $j = L, L-1, \ldots, 1$,

$$v_{0,j-1} = \sum_{\ell=0}^{r} \sqrt{2} h_\ell \otimes (\uparrow v_{\ell,j}).$$

Chapter 3

A Semismooth Newton-CG Augmented Lagrangian Algorithm

This chapter is devoted to designing and analysing augmented Lagrangian based algorithms to solve the general non-separable ℓ_1-regularized convex minimization problem (1.1). We first reformulate the original unconstrained problem to an equivalent constrained one, so that it is conducive for us to apply the augmented Lagrangian framework [84]. Then we propose an inexact semismooth Newton augmented Lagrangian (SSNAL) algorithm to solve this reformulated constrained problem. We also characterize the condition when the generalized Hessian of the objective function is positive definite, and subsequently provide the convergence analysis of the proposed SSNAL algorithm.

Notation

Throughout the discussions of the algorithms for solving (1.1) in this chapter and Chapter 4, we often use the following soft-thresholding and projection (onto an ℓ_∞-ball) functions defined by

$$\boldsymbol{s}_\nu : \mathbb{R}^n \to \mathbb{R}^n, \quad \boldsymbol{s}_\nu(\boldsymbol{x}) = \mathrm{sign}(\boldsymbol{x}) \circ \max\{|\boldsymbol{x}| - \nu, 0\} \tag{3.1}$$

$$\boldsymbol{\pi}_\nu : \mathbb{R}^n \to \mathbb{R}^n, \quad \boldsymbol{\pi}_\nu(\boldsymbol{x}) = \mathrm{sign}(\boldsymbol{x}) \circ \min\{|\boldsymbol{x}|, \nu\} \tag{3.2}$$

where $\nu \geq 0$ is a given parameter, and "$\circ$" denotes the componentwise product between two vectors; $\mathrm{sign}(\boldsymbol{x})$ and $|\boldsymbol{x}|$ are the vectors obtained from $\boldsymbol{x}$ by taking sign and absolute value of the components. Observe that $\boldsymbol{x} = \boldsymbol{s}_\nu(\boldsymbol{x}) + \boldsymbol{\pi}_\nu(\boldsymbol{x})$.

Define also the Huber function $\phi_\nu : \mathbb{R} \to \mathbb{R}$ by

$$\phi_\nu(t) = \begin{cases} \frac{1}{2}t^2 & \text{if } |t| \le \nu \\ \nu|t| - \frac{1}{2}\nu^2 & \text{if } |t| > \nu. \end{cases} \tag{3.3}$$

Note that $\phi_\nu(t)$ can be written compactly as

$$\phi_\nu(t) = \frac{1}{2}\left(t^2 - \boldsymbol{s}_\nu^2(t)\right).$$

It is not difficult to show that $\boldsymbol{s}_\nu(\boldsymbol{x})$ is the unique minimizer of the following proximal minimization problem:

$$\min\left\{\|\boldsymbol{y}\|_1 + \frac{1}{2\nu}\|\boldsymbol{y}-\boldsymbol{x}\|^2 \mid \boldsymbol{y} \in \mathbb{R}^n\right\} = \frac{1}{\nu}\sum_{i=1}^n \phi_\nu(\boldsymbol{x}_i) =: \Phi_\nu(\boldsymbol{x}) \tag{3.4}$$

where $\Phi_\nu(\boldsymbol{x})$ and $\boldsymbol{s}_\nu(\boldsymbol{x})$ are known as the Moreau-Yoshida regularization and proximal map of the function $\|\cdot\|_1$ with parameter ν, respectively.

3.1 Reformulation of (1.1)

It is clear that we can rewrite (1.1) as the following linearly constrained convex minimization problem by introducing an auxiliary variable $\boldsymbol{u}$:

$$\begin{array}{ll} \min_{\boldsymbol{x}\in\mathbb{R}^n, \boldsymbol{u}\in\mathbb{R}^p} & f(\boldsymbol{x}) + \rho\|\boldsymbol{u}\|_1 \\ \text{s.t.} & B\boldsymbol{x} - \boldsymbol{u} = 0. \end{array} \tag{P}$$

By direct calculation, we can show that the dual problem of (P) is given by:

$$\begin{aligned} &\max_{\boldsymbol{v}\in\mathbb{R}^p}\left\{g(\boldsymbol{v}) := \min_{\boldsymbol{x}\in\mathbb{R}^n}\left\{f(\boldsymbol{x}) + \langle B^T\boldsymbol{v},\, \boldsymbol{x}\rangle\right\} \mid \|\boldsymbol{v}\|_\infty \le \rho\right\} \\ =&\max_{\boldsymbol{v}\in\mathbb{R}^p}\left\{-f^*(-B^T\boldsymbol{v}) \mid \|\boldsymbol{v}\|_\infty \le \rho\right\} \qquad (3.5) \\ =&\max_{\boldsymbol{v}\in\mathbb{R}^p, \boldsymbol{x}\in\mathbb{R}^n}\left\{f(\boldsymbol{x}) + \langle B^T\boldsymbol{v},\, \boldsymbol{x}\rangle \mid \nabla f(\boldsymbol{x}) + B^T\boldsymbol{v} = 0,\ \|\boldsymbol{v}\|_\infty \le \rho\right\}, \qquad (\mathrm{D}) \end{aligned}$$

where f^* denotes the conjugate function of f defined by

$$f^*(\boldsymbol{y}) := \sup_{\boldsymbol{x}\in\mathbb{R}^n}\{\langle \boldsymbol{y},\, \boldsymbol{x}\rangle - f(\boldsymbol{x})\}.$$

Since the optimal value of (P) is finite and attained, and that the Slater condition holds for the convex problem (P), strong duality holds for (P) and (D) [3, Theorem 6.2.4], i.e., there exists $(\boldsymbol{x}^*, \boldsymbol{u}^*, \boldsymbol{v}^*)$ such that $(\boldsymbol{x}^*, \boldsymbol{u}^*)$ is optimal for (P) and $(\boldsymbol{x}^*, \boldsymbol{v}^*)$ is optimal for (D), and $\rho\|\boldsymbol{u}^*\|_1 = \langle B^T\boldsymbol{v}^*,\, \boldsymbol{x}^*\rangle$. Furthermore, $(\boldsymbol{x}^*, \boldsymbol{u}^*, \boldsymbol{v}^*)$ must satisfy the following optimality conditions for (P) and (D):

$$B\boldsymbol{x} - \boldsymbol{u} = 0,\ \nabla f(\boldsymbol{x}) + B^T\boldsymbol{v} = 0,\ \boldsymbol{v} \in \partial\rho\|\boldsymbol{u}\|_1, \tag{3.6}$$

where $\partial\rho\|\boldsymbol{u}\|_1$ is the subdifferential of $\rho\|\boldsymbol{u}\|_1$, and $\boldsymbol{v} \in \partial\rho\|\boldsymbol{u}\|_1$ means

$$\boldsymbol{v}_i \in \begin{cases} \{\rho\} & \text{if } \boldsymbol{u}_i > 0 \\ \{-\rho\} & \text{if } \boldsymbol{u}_i < 0 \\ [-\rho, \rho] & \text{if } \boldsymbol{u}_i = 0, \end{cases} \qquad i = 1, ..., p.$$

Note that the condition (3.6) is also the necessary and sufficient condition for $\boldsymbol{x}$ to be an optimal solution to (1.1). Based on (3.6), we can see that if B has full column rank, then for a sufficiently large parameter ρ, the problem (1.1) admits $\boldsymbol{x} = 0$ as the optimal solution. Specifically, let $\bar{\boldsymbol{v}} = B(B^TB)^{-1}\nabla f(0)$. It is easy to observe that if $\rho \geq \|\bar{\boldsymbol{v}}\|_\infty$, then $\boldsymbol{x} = 0, \boldsymbol{u} = 0, \boldsymbol{v} = \bar{\boldsymbol{v}}$ are the optimal solutions to (P) and (D).

The augmented Lagrangian method which we will design shortly is based on the following augmented Lagrangian function of (P) defined by: $\mathcal{L}_\sigma : \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$

$$\begin{aligned} \mathcal{L}_\sigma(\boldsymbol{x}, \boldsymbol{u}; \boldsymbol{v}) &= f(\boldsymbol{x}) + \rho\|\boldsymbol{u}\|_1 + \langle \boldsymbol{v},\, B(\boldsymbol{x}) - \boldsymbol{u}\rangle + \frac{\sigma}{2}\|B\boldsymbol{x} - \boldsymbol{u}\|^2 \\ &= f(\boldsymbol{x}) + \rho\|\boldsymbol{u}\|_1 + \frac{\sigma}{2}\|B(\boldsymbol{x}) - \boldsymbol{u} + \sigma^{-1}\boldsymbol{v}\|^2 - \frac{1}{2\sigma}\|\boldsymbol{v}\|^2 \end{aligned} \tag{3.7}$$

where $\sigma > 0$ is a given parameter. By virtue of [83, Theorem 3.2], we know that the optimal value of (P) is the same as the optimal value of the following maximization problem:

$$\max_{\boldsymbol{v}\in\mathbb{R}^p} \min_{\boldsymbol{x}\in\mathbb{R}^n, \boldsymbol{u}\in\mathbb{R}^p} \mathcal{L}_\sigma(\boldsymbol{x}, \boldsymbol{u}; \boldsymbol{v}).$$

3.2 The General Augmented Lagrangian Framework

One of the most popular methods to solve a convex problem like (P) is the Hestenes-Powell method of multipliers [54, 80], which is a special case of the augmented Lagrangian method (ALM) [84] when there are only equality constraints. The general framework of the ALM for solving (P) can be described as follows. Given $\boldsymbol{v}^0$, $\sigma_0 > 0$ and tolerance $\varepsilon > 0$, iterate the following steps:

$$(\boldsymbol{x}^{k+1}, \boldsymbol{u}^{k+1}) \approx \arg\min_{\boldsymbol{x}\in\mathbb{R}^n, \boldsymbol{u}\in\mathbb{R}^p} \mathcal{L}_{\sigma_k}(\boldsymbol{x}, \boldsymbol{u}; \boldsymbol{v}^k) \tag{3.8}$$

$$\boldsymbol{v}^{k+1} = \boldsymbol{v}^k + \sigma_k(B\boldsymbol{x}^{k+1} - \boldsymbol{u}^{k+1})$$

$$\text{If } \|(\boldsymbol{v}^k - \boldsymbol{v}^{k+1})/\sigma_k\| \le \varepsilon, \text{ stop; else update } \sigma_k \text{ such that } 0 < \sigma_k \uparrow \sigma_\infty \le \infty. \tag{3.9}$$

The convergence of the ALM for general convex optimization problems has been established in [84,85], where the theory is derived by interpreting the ALM applied to the primal problem (P) as a proximal point algorithm applied to the corresponding extended dual problem (3.5).

We should emphasize that the main task in each ALM iteration is to solve the minimization subproblem (3.8). And different strategies to solve the subproblem will lead to different variants of the ALM. In the next subsection, we will focus on designing an efficient inexact semismooth Newton algorithm (which exploits second-order information) to solve the inner subproblem (3.8). The convergence results, suitably adapted for (P), will also be provided accordingly.

One can of course use a variety of first-order methods to solve the subproblem (3.8), especially popular methods such as the gradient descent method, alternating direction methods, and accelerated proximal gradient method of Beck and Teboulle [5] (when ∇f is Lipschitz continuous). However, for the ALM to converge, the subproblem (3.8) must be solved to relatively high accuracy and first-order methods are typically not the most efficient ones for solving a problem to high accuracy. This weakness is especially disadvantageous because the problem in (3.8) must be solved repeatedly. Thus, this motivates us to design a semismooth Newton method for (3.8) which can achieve quadratic convergence under suitable constraint nondegeneracy conditions.

Note that based on the ALM framework, we can delineate the relation between various existing models/algorithms that have been used to approximately solve (1.1) for computational expediency, as we can see from the following remarks.

Remark 3.2.1. For the particular choice of setting the Lagrangian multiplier $\boldsymbol{v} = 0$, the value of the following minimization problem thus provides a lower bound for the optimal value of (P):

$$\min_{\boldsymbol{x}\in\mathbb{R}^n,\boldsymbol{u}\in\mathbb{R}^p} \mathcal{L}_\sigma(\boldsymbol{x},\boldsymbol{u};0) = \min_{\boldsymbol{x}\in\mathbb{R}^n,\boldsymbol{u}\in\mathbb{R}^p} \left\{ f(\boldsymbol{x}) + \frac{\sigma}{2}\|B\boldsymbol{x}-\boldsymbol{u}\|^2 + \rho\|\boldsymbol{u}\|_1 \right\} \tag{3.10}$$

$$= \min_{\boldsymbol{x}\in\mathbb{R}^n} \left\{ f(\boldsymbol{x}) + \sigma \sum_{i=1}^{p} \phi_{\rho/\sigma}((B\boldsymbol{x})_i) \right\}. \tag{3.11}$$

The problem (3.11) reduces to one that has been considered in [75] when $f(\boldsymbol{x}) = \frac{1}{2}\|A\boldsymbol{x}-\boldsymbol{b}\|^2$. The interpretation of (3.10) or (3.11) as a suboptimal approximation of (P) gives us an interesting view-point that it is perhaps not necessary to use exotic convex regularization terms such as the Huber functions considered in [75] but suffice to just use the term $\rho\|B\boldsymbol{x}\|_1$ in (1.1).

Remark 3.2.2. In the context of TV-norm image restoration, the problem (1.1) with $f(\boldsymbol{x}) = \frac{1}{2}\|A\boldsymbol{x}-\boldsymbol{b}\|^2$ and $\|B\boldsymbol{x}\|_1 = \|\boldsymbol{x}\|_{TV}$, is often approximated by the problem (3.10) for some suitably large parameter σ (see [102]), since it is well known that the solution $\boldsymbol{x}(\sigma)$ of (3.10) would converge to a solution of (1.1) when $\sigma \uparrow \infty$. But the problem (3.10) is exactly the subproblem in the zero-th iteration (with $\boldsymbol{v}^0 = 0$) of our ALM. Thus the approximation problem (3.10) solved in [102] is just one iteration of our ALM. In [102], an alternating minimization method is used to solve (3.10). We should also mention that while the parameter σ must be chosen to be relatively large in [102], it can be chosen to be a moderate constant for our ALM.

3.3 An Inexact Semismooth Newton Method for Solving (3.8)

In this section, we design a semismooth Newton method to solve the subproblem in (3.8). By minimizing $\mathcal{L}_\sigma(\boldsymbol{x},\boldsymbol{u};\boldsymbol{v}^k)$ with respect to $\boldsymbol{u}$ first and using (3.4), we get the

equivalent problem below:

$$\boldsymbol{x}^{k+1} \approx \operatorname{argmin}_{\boldsymbol{x}\in\mathbb{R}^n}\Big\{\psi(\boldsymbol{x}) := f(\boldsymbol{x}) + \sigma\sum_{i=1}^{p}\phi_{\rho/\sigma}(\boldsymbol{\eta}_i)\Big\} \tag{3.12}$$

where $\phi_\varepsilon(t)$ is the Huber function as defined in (3.3), and

$$\boldsymbol{\eta} := B\boldsymbol{x} + \sigma^{-1}\boldsymbol{v}^k. \tag{3.13}$$

Note that in $\boldsymbol{\eta}$ we have suppressed the index to show the dependence on k since it is fixed. Thus to solve (3.8), we can solve the problem (3.12) involving only the variable $\boldsymbol{x}$. Once we have computed the optimal solution $\boldsymbol{x}^{k+1}$ from (3.12), we can compute the optimal $\boldsymbol{u}$ by setting

$$\boldsymbol{u}^{k+1} = \boldsymbol{s}_{\rho/\sigma}(B\boldsymbol{x}^{k+1} + \sigma^{-1}\boldsymbol{v}^k). \tag{3.14}$$

From our assumption that the objective function in (1.1) is coercive, we can show that the function $\psi(\boldsymbol{x})$ is also coercive. Hence (3.12) has a minimizer, and a necessary and sufficient condition for optimality is given by:

$$0 = \nabla\psi(\boldsymbol{x}) = \nabla f(\boldsymbol{x}) + \sigma B^T[\phi'_{\rho/\sigma}(\boldsymbol{\eta}_1);\ldots;\phi'_{\rho/\sigma}(\boldsymbol{\eta}_p)] = \nabla f(\boldsymbol{x}) + \sigma B^T(\boldsymbol{\pi}_{\rho/\sigma}(\boldsymbol{\eta})). \tag{3.15}$$

Note that the objective function $\psi(\boldsymbol{x})$ in (3.12) is convex and smooth, but it is not necessarily twice continuously differentiable. Hence classical Newton method cannot be applied to (3.12). Fortunately, the gradient $\nabla\psi(\boldsymbol{x})$ is strongly semismooth for all $\boldsymbol{x} \in \mathbb{R}^n$ (since $\nabla f(\cdot)$ and $\boldsymbol{\pi}_{\rho/\sigma}(\cdot)$ are strongly semismooth), and we may apply a semismooth Newton method [81] to solve the nonlinear equation (3.15). The semismooth Newton method is a second-order method which can achieve quadratic convergence under suitable nondegeneracy conditions (more details will be given later).

In the following, we design an inexact semismooth Newton-CG (SSNCG) algorithm to solve the subproblem (3.12) based on the equation (3.15). At a current iterate $\boldsymbol{x}^j$, let $\boldsymbol{\eta}^j = B\boldsymbol{x}^j + \sigma^{-1}\boldsymbol{v}^k$. We compute the Newton direction for (3.12) from the following generalized Newton equation:

$$(\nabla^2 f(\boldsymbol{x}^j) + \sigma B^T\operatorname{diag}(\boldsymbol{w}^j)B)\Delta\boldsymbol{x} \;\approx\; -\nabla\psi(\boldsymbol{x}^j), \tag{3.16}$$

where $\mathrm{diag}(\boldsymbol{w}^j) \in \partial \boldsymbol{\pi}_{\rho/\sigma}(\boldsymbol{\eta}^j)$, and

$$\boldsymbol{w}_i^j = \begin{cases} 1 & \text{if } |\boldsymbol{\eta}_i^j| < \rho/\sigma \\ 0 & \text{otherwise.} \end{cases} \tag{3.17}$$

Note that for large scale problems where n is large, it is generally not possible or too expensive to solve (3.16) by a direct method, and iterative method such as the preconditioned conjugate gradient (PCG) method has to be employed.

Before we describe the inexact SSNCG algorithm for solving (3.12), we briefly discuss the generalized Hessian of ψ at a given $\boldsymbol{x} \in \mathbb{R}^n$ since it is required in the algorithm. Since $\nabla f(\cdot)$ and $\boldsymbol{\pi}_{\rho/\sigma_k}(\cdot)$ are locally Lipschitz continuous, the function $\nabla\psi(\cdot)$ is locally Lipschitz continuous on $\mathbb{R}^n$. By Rademacher's Theorem, $\nabla\psi$ is almost everywhere Fréchet-differentiable in $\mathbb{R}^n$, and the generalized Hessian of ψ at $\boldsymbol{x}$ is defined as

$$\partial^2\psi(\boldsymbol{x}) := \partial(\nabla\psi)(\boldsymbol{x}), \tag{3.18}$$

where $\partial(\nabla\psi)(\boldsymbol{x})$ is the Clarke generalized Jacobian of $\nabla\psi$ at $\boldsymbol{x}$ [28]. However, it is not easy to express $\partial^2\psi(\boldsymbol{x})$ exactly, and it is typically approximated by the following set:

$$\hat{\partial}^2\psi(\boldsymbol{x}) := \left\{\nabla^2 f(\boldsymbol{x}) + \sigma B^T D B \mid D \in \partial\boldsymbol{\pi}_{\rho/\sigma}(\boldsymbol{\eta})\right\} \tag{3.19}$$

where $\boldsymbol{\eta} := B\boldsymbol{x} + \sigma^{-1}\boldsymbol{v}^k$, and

$$\partial\boldsymbol{\pi}_{\rho/\sigma}(\boldsymbol{\eta}) = \left\{ \mathrm{diag}(\boldsymbol{w}) \;\middle|\; \begin{array}{ll} \boldsymbol{w}_i = 1 & \text{if } |\boldsymbol{\eta}_i| < \rho/\sigma \\ \boldsymbol{w}_i = 0 & \text{if } |\boldsymbol{\eta}_i| > \rho/\sigma \\ \boldsymbol{w}_i \in [0,1] & \text{otherwise} \end{array} \right\} \tag{3.20}$$

From [28, Corollary in p.75], it holds that for any $\mathbf{d} \in \mathbb{R}^n$, one has

$$\partial^2\psi(\boldsymbol{x})\mathbf{d} \subseteq \hat{\partial}^2\psi(\boldsymbol{x})\mathbf{d},$$

which means that if every element in $\hat{\partial}^2\psi(\boldsymbol{x})$ is positive definite, then so is every element in $\partial^2\psi(\boldsymbol{x})$.

The inexact SSNCG algorithm [110] we use to solve (3.12) is described as follows.

Semismooth Newton-CG (SSNCG) Algorithm

Step 0. Given $\boldsymbol{x}^0 \in \mathbb{R}^n$ and $\mu \in (0, 1/2), \bar{\gamma}, \delta, \tau_1, \tau_2 \in (0, 1)$, $\tau \in (0, 1]$. Set $j := 0$.

Step 1. Select $V_j \in \hat{\partial}^2\psi(\boldsymbol{x}^j)$ and compute

$$\epsilon_j := \tau_1 \min\{\tau_2, \|\nabla\psi(\boldsymbol{x}^j)\|\}, \quad \gamma_j := \min\{\bar{\gamma}, \|\nabla\psi(\boldsymbol{x}^j)\|^{1+\tau}\}.$$

Apply the PCG method to find an approximate solution $\Delta\boldsymbol{x}^j$ to

$$(V_j + \epsilon_j I)\Delta\boldsymbol{x} = -\nabla\psi(\boldsymbol{x}^j)$$

such that the residual satisfies the following condition:

$$\|(V_j + \epsilon_j I)\Delta\boldsymbol{x}^j + \nabla\psi(\boldsymbol{x}^j)\| \le \gamma_j. \tag{3.21}$$

Step 2. Let ℓ_j be the smallest nonnegative integer ℓ such that

$$\psi(\boldsymbol{x}^j + \delta^\ell\Delta\boldsymbol{x}^j) \le \psi(\boldsymbol{x}^j) + \mu\delta^\ell\langle\nabla\psi(\boldsymbol{x}^j), \Delta\boldsymbol{x}^j\rangle.$$

Set $\boldsymbol{x}^{j+1} = \boldsymbol{x}^j + \delta^{\ell_j}\Delta\boldsymbol{x}^j$.

Step 3. Replace j by $j + 1$ and go to Step 1.

3.4 Convergence of the Inexact SSNCG Method

The efficiency of the SSNCG algorithm for solving (3.12) depends on the positive definiteness of the generalized Hessian matrices of ψ. Thus before giving the convergence results for the SSNCG algorithm, we shall characterize the positive definiteness of the elements in $\hat{\partial}^2\psi(\boldsymbol{x})$.

By a direct calculation, we get the dual problem of (3.8):

$$\begin{aligned}
&\max_{\boldsymbol{s}\in\mathbb{R}^p}\left\{\min_{\boldsymbol{x}\in\mathbb{R}^n}\left\{f(\boldsymbol{x})+\langle B^T\boldsymbol{s},\,\boldsymbol{x}\rangle\right\}-\frac{1}{2\sigma}\|\boldsymbol{s}-\boldsymbol{v}\|^2 \mid \|\boldsymbol{s}\|_\infty\le\rho\right\}\\
=&\max_{\boldsymbol{s}\in\mathbb{R}^p}\left\{-f^*(-B^T\boldsymbol{s})-\frac{1}{2\sigma}\|\boldsymbol{s}-\boldsymbol{v}\|^2 \mid \|\boldsymbol{s}\|_\infty\le\rho\right\}\\
=&\max_{\boldsymbol{s}\in\mathbb{R}^p,\boldsymbol{x}\in\mathbb{R}^n}\left\{f(\boldsymbol{x})+\langle B^T\boldsymbol{s},\,\boldsymbol{x}\rangle-\frac{1}{2\sigma}\|\boldsymbol{s}-\boldsymbol{v}\|^2 \mid \nabla f(\boldsymbol{x})+B^T\boldsymbol{s}=0,\ \|\boldsymbol{s}\|_\infty\le\rho\right\}.
\end{aligned}\tag{3.22}$$

We can show that the objective function of (3.8) is coercive, and hence its optimal value is finite and attained. Furthermore, strong duality holds for (3.8) and (3.22), i.e., there exists a triple $(\hat{\boldsymbol{x}},\hat{\boldsymbol{u}},\hat{\boldsymbol{s}})$ such that $(\hat{\boldsymbol{x}},\hat{\boldsymbol{u}})$ is optimal for (3.8) and $(\hat{\boldsymbol{x}},\hat{\boldsymbol{s}})$ is optimal for (3.22). The triple $(\hat{\boldsymbol{x}},\hat{\boldsymbol{u}},\hat{\boldsymbol{s}})$ must satisfy the following optimality conditions:

$$\boldsymbol{\eta}:=B\boldsymbol{x}+\sigma^{-1}\boldsymbol{v},\ \boldsymbol{u}=\boldsymbol{\eta}-\sigma^{-1}\boldsymbol{s},\ \boldsymbol{s}=\sigma\,\boldsymbol{\pi}_{\rho/\sigma}(\boldsymbol{\eta}),\ \nabla f(\boldsymbol{x})+B^T\boldsymbol{s}=0.\tag{3.23}$$

From (3.23) and the definition of $\boldsymbol{\pi}_{\rho/\sigma}(\boldsymbol{\eta})$, it is obvious that $\|\boldsymbol{s}\|_\infty\le\rho$.

Let us denote the active set corresponding to the inequality constraints of (3.22) by

$$\hat{J}:=\{i \mid\ |\hat{\boldsymbol{s}}_i|=\rho,\ i=1,\dots,p\}.\tag{3.24}$$

Then, it is well known (cf. [76, Definition 12.1]) that the linear independence constraint qualification (LICQ) for (3.22) holds at $(\hat{\boldsymbol{x}},\hat{\boldsymbol{s}})$ if the following condition is satisfied:

$$\begin{bmatrix}\nabla^2 f(\hat{\boldsymbol{x}}) & 0\\ B_{\hat{J}} & \mathrm{diag}(\mathrm{sign}(\hat{\boldsymbol{s}}_{\hat{J}}))\\ \overline{B}_{\hat{J}} & 0\end{bmatrix}\ \text{has full column rank,}\tag{3.25}$$

where $B_{\hat{J}}$ is the submatrix formed by extracting rows of B with row-indices in $\hat{J}$ and $\overline{B}_{\hat{J}}$ is the remaining submatrix. It is interesting to note the following equivalent condition for the LICQ.

Lemma 3.4.1. The LICQ condition (3.25) can equivalently be stated as follows:

$$\mathrm{range}(\nabla^2 f(\hat{\boldsymbol{x}}))+\mathrm{range}(\overline{B}_{\hat{J}}^T)=\mathbb{R}^n.$$

Proof. It is well known that the matrix in (3.25) has full column rank if and only if its null space is trivial, which in turn is equivalent to

$$\text{range}\left(\begin{bmatrix} \nabla^2 f(\hat{\boldsymbol{x}}) & B_{\hat{J}}^T & \overline{B}_{\hat{J}}^T \\ 0 & \text{diag}(\text{sign}(\hat{\boldsymbol{s}}_{\hat{J}})) & 0 \end{bmatrix}\right) = \begin{pmatrix} \mathbb{R}^n \\ \mathbb{R}^{|\hat{J}|} \end{pmatrix}.$$

From here, we can readily show the required result. □

The following lemma will be needed in our subsequent analysis.

Lemma 3.4.2. For given $\boldsymbol{v} \in \mathbb{R}^p$ and $\rho > 0$. Let $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}}, \hat{\boldsymbol{s}})$ be a triple satisfying the KKT condition (3.23) and $\hat{\boldsymbol{\eta}} := B\hat{\boldsymbol{x}} + \sigma^{-1}\boldsymbol{v}$. Then, for $i = 1, \ldots, p$, the following results hold.

(i) If $|\hat{\boldsymbol{\eta}}_i| < \rho/\sigma$, then $\hat{\boldsymbol{u}}_i = 0$ and $|\hat{\boldsymbol{s}}_i| < \rho$;

(ii) If $|\hat{\boldsymbol{\eta}}_i| > \rho/\sigma$, then $\hat{\boldsymbol{u}}_i \neq 0$ and $|\hat{\boldsymbol{s}}_i| = \rho$;

(iii) If $|\hat{\boldsymbol{\eta}}_i| = \rho/\sigma$, then $\hat{\boldsymbol{u}}_i = 0$ and $|\hat{\boldsymbol{s}}_i| = \rho$.

Therefore, $|\hat{\boldsymbol{s}}_i| = \rho \Leftrightarrow \hat{\boldsymbol{\eta}}_i \geq \rho/\sigma$, and for the set $\hat{J}$ defined in (3.24), we have

$$\hat{J} = \{i \mid |\hat{\boldsymbol{\eta}}_i| \geq \rho/\sigma,\ i = 1, \ldots, p\}. \tag{3.26}$$

Proof. Since the proof of the first part of this lemma can be directly verified by (3.23), we omit it. The second part easily follows from the first part and (3.24). □

Proposition 3.1. *Suppose that $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}}, \hat{\boldsymbol{s}})$ is a triple satisfying the KKT condition (3.23). Let $\hat{\boldsymbol{\eta}} := B\hat{\boldsymbol{x}} + \sigma^{-1}\boldsymbol{v}$. Then the following statements are equivalent:*

(a) LICQ for (3.22) holds at $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{s}})$.

(b) Every $V \in \hat{\partial}^2\psi(\hat{\boldsymbol{x}})$ is symmetric positive definite.

(c) Let $V^0 = \nabla^2 f(\hat{\boldsymbol{x}}) + \sigma B^T \text{diag}(\boldsymbol{w}^0) B$, with $\boldsymbol{w}_i^0 = 1$ if $|\hat{\boldsymbol{\eta}}_i| < \rho/\sigma$ and $\boldsymbol{w}_i^0 = 0$ otherwise. It holds that $V^0 \in \hat{\partial}^2\psi(\hat{\boldsymbol{x}})$ is symmetric positive definite.

(d) $\nabla^2 f(\hat{\boldsymbol{x}})$ is positive definite on the null space $\text{Ker}(\overline{B}_{\hat{J}})$.

Proof. "(a) ⇒ (b)". Suppose for the purpose of contradiction that (b) does not hold. Then there exists some $V \in \hat{\partial}^2\psi(\hat{\boldsymbol{x}})$ such that V is not positive definite. By the

definition of $\hat{\partial}^2\psi(\hat{\boldsymbol{x}})$ in (3.19), there exists a $D \in \partial\boldsymbol{\pi}_{\rho/\sigma}(\hat{\boldsymbol{\eta}})$ such that $V = \nabla^2 f(\hat{\boldsymbol{x}}) + \sigma B^T DB$. Since $\nabla^2 f(\hat{\boldsymbol{x}})$ is symmetric positive semidefinite (because f is convex and twice continuously differentiable), we know that V is positive semidefinite, thus there exists $0 \neq \mathbf{d} \in \mathbb{R}^n$ such that

$$0 = \langle \mathbf{d},\, V\mathbf{d}\rangle = \langle \mathbf{d},\, \nabla^2 f(\hat{\boldsymbol{x}})\mathbf{d}\rangle + \sigma\langle \mathbf{d},\, (B^T DB)\mathbf{d}\rangle. \tag{3.27}$$

From (3.26) and (3.20), we obtain that $\langle \mathbf{d},\, (B^T DB)\mathbf{d}\rangle \geq \langle \overline{B}_{\hat{\jmath}}\mathbf{d},\, \overline{B}_{\hat{\jmath}}\mathbf{d}\rangle$, which, together with (3.27), yields

$$\langle \mathbf{d},\, \nabla^2 f(\hat{\boldsymbol{x}})\mathbf{d}\rangle = 0, \quad \langle \overline{B}_{\hat{\jmath}}\mathbf{d},\, \overline{B}_{\hat{\jmath}}\mathbf{d}\rangle = 0.$$

This implies that $\nabla^2 f(\hat{\boldsymbol{x}})\mathbf{d} = 0$ and $\overline{B}_{\hat{\jmath}}\mathbf{d} = 0$. Therefore, the linear system

$$\begin{bmatrix} \nabla^2 f(\hat{\boldsymbol{x}}) & 0 \\ B_{\hat{\jmath}} & \operatorname{diag}(\operatorname{sign}(\hat{\boldsymbol{s}}_{\hat{\jmath}})) \\ \overline{B}_{\hat{\jmath}} & 0 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{d}} \\ \boldsymbol{\alpha} \end{bmatrix} = 0 \tag{3.28}$$

has nonzero solution $(\hat{\mathbf{d}}, \boldsymbol{\alpha}) = (\mathbf{d}, -\operatorname{sign}(\hat{\boldsymbol{s}}_{\hat{\jmath}}) \circ (B_{\hat{\jmath}}\mathbf{d}))$, which contradicts that LICQ for (3.22) holds at $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{s}})$. The contradiction implies that (b) holds.

"(b) $\Rightarrow$ (c)". It is obvious since $V^0 \in \hat{\partial}^2\psi(\hat{\boldsymbol{x}})$.

"(c) $\Rightarrow$ (a)". By Lemma 3.4.2 and the definition of V^0, we can easily deduce that

$$V^0 = \nabla^2 f(\hat{\boldsymbol{x}}) + \sigma \overline{B}_{\hat{\jmath}}^T \overline{B}_{\hat{\jmath}}. \tag{3.29}$$

Now we show that (a) holds. Let $(\tilde{\mathbf{d}}, \tilde{\boldsymbol{\alpha}}) \in \mathbb{R}^n \times \mathbb{R}^{|\hat{\jmath}|}$ be a solution of linear system (3.28). Then, we have that

$$\nabla^2 f(\hat{\boldsymbol{x}})\tilde{\mathbf{d}} = 0, \quad \overline{B}_{\hat{\jmath}}\tilde{\mathbf{d}} = 0 \quad \text{and} \quad \tilde{\boldsymbol{\alpha}} = -\operatorname{sign}(\hat{\boldsymbol{s}}_{\hat{\jmath}}) \circ (B_{\hat{\jmath}}\tilde{\mathbf{d}}). \tag{3.30}$$

This implies that $\langle \tilde{\mathbf{d}},\, (\nabla^2 f(\hat{\boldsymbol{x}}) + \sigma \overline{B}_{\hat{\jmath}} \overline{B}_{\hat{\jmath}})\tilde{\mathbf{d}}\rangle = \langle \tilde{\mathbf{d}},\, V^0\tilde{\mathbf{d}}\rangle = 0$, and hence $\tilde{\mathbf{d}} = 0$ since V^0 is positive definite. From (3.30), it holds that $\tilde{\boldsymbol{\alpha}} = 0$. This shows that the condition (3.25) is satisfied and hence (a) holds.

"(c) $\Leftrightarrow$ (d)". By using (3.29), one can easily show that (c) is equivalent to (d). $\square$

Now we are ready to state the results on the rate of convergence of the SSNCG

algorithm.

Theorem 3.4.3. The SSNCG algorithm is well defined and any accumulation point of $\{\boldsymbol{x}^j\}$ generated by this algorithm is an optimal solution to the subproblem (3.12).

Moreover, since the optimal solution set of (3.12) is nonempty and bounded, the sequence $\{\boldsymbol{x}^j\}$ has an accumulation point $\hat{\boldsymbol{x}}$. Suppose that the LICQ for (3.22) holds at $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}}, \hat{\boldsymbol{s}})$, where $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}}, \hat{\boldsymbol{s}})$ satisfies the KKT condition (3.23). Then the sequence $\{\boldsymbol{x}^j\}$ converges to $\hat{\boldsymbol{x}}$ at least superlinearly, i.e.,

$$\|\boldsymbol{x}^{j+1} - \hat{\boldsymbol{x}}\| = O(\|\boldsymbol{x}^j - \hat{\boldsymbol{x}}\|^{1+\tau})$$

where the parameter $\tau \in (0, 1]$ is given in the SSNCG algorithm.

Proof. Since $V_j \in \hat{\partial}^2\psi(\boldsymbol{x}^j)$ is always positive semidefinite, the matrix $V_j + \epsilon_j I$ is positive definite as long as $\nabla\psi(\boldsymbol{x}^j) \neq 0$. Assuming $\nabla\psi(\boldsymbol{x}^j) \neq 0$ for any $j \geq 0$, by [110, Lemma 3.1], we have

$$\frac{1}{\lambda_{\max}(\widetilde{V}_j)} \leq \frac{\langle -\nabla\psi(\boldsymbol{x}^j), \Delta\boldsymbol{x}^j\rangle}{\|\nabla\psi(\boldsymbol{x}^j)\|^2} \leq \frac{1}{\lambda_{\min}(\widetilde{V}_j)} \tag{3.31}$$

where $\widetilde{V}_j := V_j + \epsilon_j I$ and $\lambda_{\max}(\widetilde{V}_j)$ and $\lambda_{\min}(\widetilde{V}_j)$ are the largest and smallest eigenvalues of $\widetilde{V}_j$ respectively. Therefore, the search direction $\Delta\boldsymbol{x}^j$ generated by the SSNCG algorithm is always a descent direction, and hence, the algorithm is well defined.

Moreover, since the optimal solution set of (3.12) is nonempty and bounded, the sequence $\{\boldsymbol{x}^j\}$ is bounded. Let $\hat{\boldsymbol{x}}$ be any accumulation point of $\{\boldsymbol{x}^j\}$. By using (3.31) and the fact that $\boldsymbol{\pi}_{\rho/\sigma}(\cdot)$ is Lipschitz continuous, it is easy to derive that $\nabla\psi(\hat{\boldsymbol{x}}) = 0$. By the convexity of $\psi(\cdot)$, $\hat{\boldsymbol{x}}$ is an optimal solution of (3.12).

Furthermore, since the LICQ for (3.22) is assumed to hold at $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}}, \hat{\boldsymbol{s}})$, where $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}}, \hat{\boldsymbol{s}})$ satisfies the KKT condition (3.23), $\hat{\boldsymbol{x}}$ is the unique optimal solution to (3.12) and the sequence $\{\boldsymbol{x}^j\}$ converges to $\hat{\boldsymbol{x}}$.

Finally, from Proposition 3.1, we know that for any $\widehat{V} \in \hat{\partial}^2\psi(\hat{\boldsymbol{x}})$ as defined in (3.19), there exists a $\widehat{D} \in \partial\boldsymbol{\pi}_{\rho/\sigma}(\hat{\boldsymbol{\eta}})$ such that

$$\widehat{V} = \nabla^2 f(\hat{\boldsymbol{x}}) + \sigma B^T \widehat{D} B \succ 0.$$

Similarly, for any $V_j,\ j \geq 0$, there exists a $D_j \in \partial\boldsymbol{\pi}_{\rho/\sigma}(\boldsymbol{\eta}^j)$, where $\boldsymbol{\eta}^j := B\boldsymbol{x}^j + \sigma^{-1}\boldsymbol{v}$,

such that

$$V_j = \nabla^2 f(\boldsymbol{x}^j) + \sigma B^T D_j B.$$

For all j sufficiently large, since $\widehat{V}$ is positive definite, $\{\|(V_j + \epsilon_j I)^{-1}\|\}$ is uniformly bounded, and since $\boldsymbol{\pi}_{\rho/\sigma}(\cdot)$ is strongly semismooth, we have

$$\begin{aligned}
&\|\boldsymbol{x}^j + \Delta\boldsymbol{x}^j - \hat{\boldsymbol{x}}\| = \|\boldsymbol{x}^j + (V_j + \epsilon_j I)^{-1}(((V_j + \epsilon_j I)\Delta\boldsymbol{x}^j + \nabla\psi(\boldsymbol{x}^j)) - \nabla\psi(\boldsymbol{x}^j)) - \hat{\boldsymbol{x}}\| \\
&\le \|\boldsymbol{x}^j - \hat{\boldsymbol{x}} - (V_j + \epsilon_j I)^{-1}\nabla\psi(\boldsymbol{x}^j)\| + \|(V_j + \epsilon_j I)^{-1}\|\|(V_j + \epsilon_j I)\Delta\boldsymbol{x}^j + \nabla\psi(\boldsymbol{x}^j)\| \\
&\le \|(V_j + \epsilon_j I)^{-1}\|\|(V_j + \epsilon_j I)(\boldsymbol{x}^j - \hat{\boldsymbol{x}}) - \nabla\psi(\boldsymbol{x}^j)\| + \gamma_j\|(V_j + \epsilon_j I)^{-1}\| \\
&\le \|(V_j + \epsilon_j I)^{-1}\|(\|\nabla\psi(\boldsymbol{x}^j) - \nabla\psi(\hat{\boldsymbol{x}}) - V_j(\boldsymbol{x}^j - \hat{\boldsymbol{x}})\| + \epsilon_j\|\boldsymbol{x}^j - \hat{\boldsymbol{x}}\| + \gamma_j) \\
&\le O(\|B^T\|\|\boldsymbol{\pi}_{\rho/\sigma}(\boldsymbol{\eta}^j) - \boldsymbol{\pi}_{\rho/\sigma}(\hat{\boldsymbol{\eta}}) - D_j(B(\boldsymbol{x}^j - \hat{\boldsymbol{x}}))\| + \tau_1\|\nabla\psi(\boldsymbol{x}^j)\|\|\boldsymbol{x}^j - \hat{\boldsymbol{x}}\| + \|\nabla\psi(\boldsymbol{x}^j)\|^{1+\tau}) \\
&\le O(\|B(\boldsymbol{x}^j - \hat{\boldsymbol{x}})\|^2 + \tau_1\|\nabla\psi(\boldsymbol{x}^j) - \nabla\psi(\hat{\boldsymbol{x}})\|\|\boldsymbol{x}^j - \hat{\boldsymbol{x}}\| + \|\nabla\psi(\boldsymbol{x}^j) - \nabla\psi(\hat{\boldsymbol{x}})\|^{1+\tau}) \\
&\le O(\|\boldsymbol{x}^j - \hat{\boldsymbol{x}}\|^2 + \tau_1\sigma\|B^T\|\|B\|\|\boldsymbol{x}^j - \hat{\boldsymbol{x}}\|^2 + (\sigma\|B^T\|\|B\|\|\boldsymbol{x}^j - \hat{\boldsymbol{x}}\|)^{1+\tau}) \\
&\le O(\|\boldsymbol{x}^j - \hat{\boldsymbol{x}}\|^{1+\tau}). \qquad (3.32)
\end{aligned}$$

This implies that for all j sufficiently large,

$$\boldsymbol{x}^j - \hat{\boldsymbol{x}} = -\Delta\boldsymbol{x}^j + O(\|\Delta\boldsymbol{x}^j\|^{1+\tau}) \quad \text{and} \quad \|\Delta\boldsymbol{x}^j\| \to 0. \qquad (3.33)$$

For each $j \ge 0$, let $R^j := (V_j + \epsilon_j I)\Delta\boldsymbol{x}^j + \nabla\psi(\boldsymbol{x}^j)$. Then, for all j sufficiently large,

$$\begin{aligned}
\langle R^j, \Delta\boldsymbol{x}^j\rangle &\le \gamma_j\|\Delta\boldsymbol{x}^j\| \le \|\nabla\psi(\boldsymbol{x}^j)\|^{1+\tau}\|\Delta\boldsymbol{x}^j\| \le \|\nabla\psi(\boldsymbol{x}^j) - \nabla\psi(\hat{\boldsymbol{x}})\|^{1+\tau}\|\Delta\boldsymbol{x}^j\| \\
&\le (\sigma\|B^T\|\|B\|\|\boldsymbol{x}^j - \hat{\boldsymbol{x}}\|)^{1+\tau}\|\Delta\boldsymbol{x}^j\| \le O(\|\Delta\boldsymbol{x}^j\|^{2+\tau}),
\end{aligned}$$

that is

$$-\langle\nabla\psi(\boldsymbol{x}^j), \Delta\boldsymbol{x}^j\rangle \ge \langle\Delta\boldsymbol{x}^j, (V_j + \epsilon_j I)\Delta\boldsymbol{x}^j\rangle + O(\|\Delta\boldsymbol{x}^j\|^{2+\tau}),$$

which, together with (3.33) and the fact that $\{\|(V_j+\epsilon_j I)^{-1}\|\}$ is uniformly bounded, implies that there exists a constant $\hat{\delta} > 0$ such that

$$-\langle\nabla\psi(\boldsymbol{x}^j), \Delta\boldsymbol{x}^j\rangle \ge \hat{\delta}\|\Delta\boldsymbol{x}^j\|^2 \quad \text{for all } j \text{ sufficiently large.}$$

Since $\nabla\psi(\cdot)$ is strongly semismooth at $\hat{\boldsymbol{x}}$, from [40, Theorem 3.2 & Remark 3.1] or [77], we know that for $\mu \in (0, 1/2)$, there exists an integer j_0 such that for any

$j \geq j_0$,

$$\psi(\boldsymbol{x}^j + \Delta\boldsymbol{x}^j) \leq \psi(\boldsymbol{x}^j) + \mu\langle\nabla\psi(\boldsymbol{x}^j),\, \Delta\boldsymbol{x}^j\rangle,$$

which means that for all $j \geq j_0$,

$$\boldsymbol{x}^{j+1} = \boldsymbol{x}^j + \Delta\boldsymbol{x}^j.$$

This, together with (3.32), completes the proof. □

Under the favourable condition stated in Proposition 3.1, the CG method for solving (3.16) can be expected to have reasonably fast convergence if the condition number of the coefficient matrix is not too large. Even then, it is still beneficial to have a preconditioner to accelerate the convergence of the PCG method. A simple and natural choice is the following diagonal preconditioner:

$$P_1 = \mathrm{diag}\Big(\nabla^2 f(\boldsymbol{x}^j) + \sigma B^T \mathrm{diag}(\boldsymbol{w}^j) B\Big).$$

If $\alpha I + \sigma B^T B$ can be inverted at moderate cost, another practical preconditioner would be

$$P_2 = \alpha I + \sigma B^T B$$

where α could be chosen as the mean of $\mathrm{diag}(\nabla^2 f(\boldsymbol{x}^j))$.

The above preconditioners are fairly general choices and may not be the most effective ones for a particular application problem. In practice, one may construct more sophisticated preconditioners for (3.16) based on the available problem structures.

3.5 The SSNAL Algorithm and Its Convergence

By adopting the general framework of the ALM to solve (P) with the subproblem (3.8) at each iteration solved by the SSNCG algorithm, we obtain our semismooth Newton based augmented Lagrangian method (for brevity, we call it the SSNAL algorithm), which is summarized in the following template.

SSNAL algorithm: Input $\boldsymbol{x}^0, \boldsymbol{v}^0 \in \mathbb{R}^n$, $\sigma_0 > 0$, iterate:

Step 1. Solve the following subproblem by using the SSNCG algorithm with $\boldsymbol{x}^k$ as the starting point:

$$\boldsymbol{x}^{k+1} \approx \arg\min\{\psi_k(\boldsymbol{x}) := \mathcal{L}_{\sigma_k}(\boldsymbol{x}; \boldsymbol{v}^k) \,:\, \boldsymbol{x} \in \mathbb{R}^n\}.$$

Step 2. Compute $\boldsymbol{\eta}^{k+1} = B\boldsymbol{x}^{k+1} + \sigma_k^{-1}\boldsymbol{v}^k$, $\boldsymbol{u}^{k+1} = \boldsymbol{s}_{\rho/\sigma_k}(\boldsymbol{\eta}^{k+1})$.

Step 3. Update $\boldsymbol{v}^{k+1} = \boldsymbol{v}^k + \sigma_k(B\boldsymbol{x}^{k+1} - \boldsymbol{u}^{k+1}) = \sigma_k \boldsymbol{\pi}_{\rho/\sigma_k}(\boldsymbol{\eta}^{k+1})$.

Step 4. If $\|\boldsymbol{r}^{k+1} := B(\boldsymbol{x}^{k+1}) - \boldsymbol{u}^{k+1} = \sigma_k^{-1}(\boldsymbol{v}^{k+1} - \boldsymbol{v}^k)\| \le \varepsilon$; stop; else; update σ_k; end.

Observe that the stopping condition for the outer iteration of the SSNAL algorithm corresponds to the primal feasibility condition of (P). Moreover, it is clear that $\|\boldsymbol{v}^{k+1}\|_\infty \le \rho$.

For the rest of this section, we will establish the global and local convergence of the SSNAL algorithm for solving the problem (P) and its dual (D), based on the general convergence theory of the ALM established in [84, 85].

We use the following stopping criteria suggested by Rockafellar [84, 85] to terminate the SSNCG algorithm for solving the subproblem (3.12):

$$\psi_k(\boldsymbol{x}^{k+1}) - \inf \psi_k \le \vartheta_k^2/(2\sigma_k), \quad \vartheta_k \ge 0, \sum_{k=0}^{\infty} \vartheta_k < \infty, \tag{A}$$

$$\psi_k(\boldsymbol{x}^{k+1}) - \inf \psi_k \le \delta_k^2/(2\sigma_k)\|\boldsymbol{v}^{k+1} - \boldsymbol{v}^k\|^2, \quad \delta_k \ge 0, \sum_{k=0}^{\infty} \delta_k < \infty, \tag{B}$$

$$\|\nabla \psi_k(\boldsymbol{x}^{k+1})\| \le (\delta_k'/\sigma_k)\|\boldsymbol{v}^{k+1} - \boldsymbol{v}^k\|, \quad 0 \le \delta_k' \to 0. \tag{B'}$$

First, we state the global convergence results of the SSNAL algorithm in the following theorem, which is obtained by specializing the results of [84, Theorem 4] to the problem (P).

Theorem 3.5.1 (Global convergence)**.** Let the SSNAL algorithm be executed with stopping criterion (A). Then the sequence $\{\boldsymbol{v}^k\}$ generated by the SSNAL algorithm is bounded and $\{\boldsymbol{v}^k\} \to \boldsymbol{v}^*$, where $\boldsymbol{v}^*$ is an optimal solution to (D), and $\{(\boldsymbol{x}^k, \boldsymbol{u}^k)\}$

is asymptotically minimizing for the problem (P) with

$$\|B\boldsymbol{x}^{k+1} - \boldsymbol{u}^{k+1}\| = \sigma_k^{-1}\|\boldsymbol{v}^{k+1} - \boldsymbol{v}^k\| \to 0 \tag{3.34}$$

$$f(\boldsymbol{x}^{k+1}) + \rho\|B\boldsymbol{x}^{k+1}\|_1 - \inf(\mathrm{P}) \le \frac{1}{2\sigma_k}\Big(\vartheta_k^2 + \|\boldsymbol{v}^k\|^2 - \|\boldsymbol{v}^{k+1}\|^2\Big). \tag{3.35}$$

Moreover, since the set $\{(\boldsymbol{x}, \boldsymbol{u}) \mid B\boldsymbol{x} - \boldsymbol{u} = 0, f(\boldsymbol{x}) + \rho\|\boldsymbol{u}\|_1 \le \alpha\}$ is nonempty and bounded for some scalar α, we have that the sequence $\{(\boldsymbol{x}^k, \boldsymbol{u}^k)\}$ is also bounded and any accumulation points $(\boldsymbol{x}^*, \boldsymbol{u}^*)$ of the sequence $\{(\boldsymbol{x}^k, \boldsymbol{u}^k)\}$ is an optimal solution to (P).

Proof. The first part of this theorem follows from [85, Theorem 4], since the problem (P) satisfies the Slater condition. The second part also follows from [85, Theorem 4]. □

Next we shall state the results on the local rate of convergence of the SSNAL algorithm. To this end, we first introduce some relevant concepts from the paper [85].

Let $l(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}) : \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ be the ordinary Lagrangian for (P) in extended form:

$$l(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}) := \begin{cases} f(\boldsymbol{x}) + \rho\|\boldsymbol{u}\|_1 + \langle \boldsymbol{v},\, B\boldsymbol{x} - \boldsymbol{u}\rangle & \text{if } \|\boldsymbol{v}\|_\infty \le \rho \\ -\infty & \text{otherwise.} \end{cases}$$

Then, for any $(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}) \in \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^p$, we define the following maximal monotone operator:

$$\mathcal{T}_l(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}) := \{(\boldsymbol{y_x}, \boldsymbol{y_u}, \boldsymbol{z}) \in \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^p :\ (\boldsymbol{y_x}, \boldsymbol{y_u}, -\boldsymbol{z}) \in \partial l(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v})\}. \tag{3.36}$$

It follows that for any $(\boldsymbol{y_x}, \boldsymbol{y_u}, \boldsymbol{z}) \in \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^p$,

$$\mathcal{T}_l^{-1}(\boldsymbol{y_x}, \boldsymbol{y_u}, \boldsymbol{z}) = \arg\min_{(\boldsymbol{x},\boldsymbol{u})\in\mathbb{R}^n\times\mathbb{R}^p} \max_{\boldsymbol{v}\in\mathbb{R}^p}\{L(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}) - \langle(\boldsymbol{x}, \boldsymbol{u}),\, (\boldsymbol{y_x}, \boldsymbol{y_u})\rangle + \langle \boldsymbol{v},\, \boldsymbol{z}\rangle\}. \tag{3.37}$$

Then the essential objective function in (D) can be defined as

$$g(\boldsymbol{v}) = \inf_{(\boldsymbol{x},\boldsymbol{u})\in\mathbb{R}^n\times\mathbb{R}^p} l(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}),$$

which is a closed concave function on $\mathbb{R}^p$. Let $\mathcal{T}_g = -\partial g$. $\mathcal{T}_g$ is a maximal monotone

operator with

$$\mathcal{T}_g^{-1}(\boldsymbol{z}) = \arg\max_{\boldsymbol{v}\in\mathbb{R}^p}\{g(\boldsymbol{v}) + \langle \boldsymbol{v},\, \boldsymbol{z}\rangle\}.$$

From now onwards, we suppose that $(\boldsymbol{x}^*, \boldsymbol{u}^*, \boldsymbol{v}^*)$ is a KKT point of (P) and (D), i.e., a solution of (3.6). Since the objective function of (P) is not everywhere differentiable, we reformulate the problem (P) as the following problem:

$$\begin{aligned} \min \quad & f(\boldsymbol{x}) + \rho(\mathbf{e}^T\boldsymbol{w}) \\ \text{s.t.} \quad & B\boldsymbol{x} - \boldsymbol{u} = 0 \\ & \boldsymbol{u} - \boldsymbol{w} \le 0 \\ & -\boldsymbol{u} - \boldsymbol{w} \le 0, \end{aligned} \tag{3.38}$$

where $\mathbf{e} \in \mathbb{R}^p$ denotes the vector of all ones. The dual problem of (3.38) is the same as (D). It is clear that (P) and (3.38) are equivalent. Indeed we can readily show that if $(\boldsymbol{x}^*, \boldsymbol{u}^*)$ is an optimal solution to (P), then $(\boldsymbol{x}^*, \boldsymbol{u}^*, \boldsymbol{w}^*)$ with $\boldsymbol{w}^* = |\boldsymbol{u}^*|$ is an optimal solution to (3.38). Conversely, if $(\boldsymbol{x}^*, \boldsymbol{u}^*, \boldsymbol{w}^*)$ is an optimal solution to (3.38), then $(\boldsymbol{x}^*, \boldsymbol{u}^*)$ is an optimal solution to (P). In what follows, we say that the strong second-order optimality condition (LICQ, respectively) holds for (P) at $(\boldsymbol{x}^*, \boldsymbol{u}^*)$ to mean that the strong second-order optimality condition (LICQ, respectively) holds for (3.38) at $(\boldsymbol{x}^*, \boldsymbol{u}^*, \boldsymbol{w}^*)$.

Define the following three index sets by

$$J_0^p = \{i \mid \boldsymbol{u}_i^* = \boldsymbol{w}_i^* = 0\},\ J_+^p = \{i \mid \boldsymbol{u}_i^* = \boldsymbol{w}_i^* > 0\},\ J_-^p = \{i \mid \boldsymbol{u}_i^* = -\boldsymbol{w}_i^* < 0\}. \tag{3.39}$$

Note that the second constraint in (3.38) is active for $i \in J_0^p \cup J_+^p$, and the third constraint is active for $i \in J_0^p \cup J_-^p$. We denote by $\bar{J}^p$ the compliment of the index set $J_0^p \cup J_+^p \cup J_-^p$. Then the LICQ holds for (P) at $(\boldsymbol{x}^*, \boldsymbol{u}^*)$ if the following condition

is satisfied:

$$\left[\begin{array}{cccc|cc|cc} B_{J_0^p}^T & B_{J_+^p}^T & B_{J_-^p}^T & B_{\bar{J}^p}^T & 0 & 0 & 0 & 0 \\ \hline -I_{J_0^p} & 0 & 0 & 0 & I_{J_0^p} & 0 & -I_{J_0^p} & 0 \\ 0 & -I_{J_+^p} & 0 & 0 & 0 & I_{J_+^p} & 0 & 0 \\ 0 & 0 & -I_{J_-^p} & 0 & 0 & 0 & 0 & -I_{J_-^p} \\ 0 & 0 & 0 & -I_{\bar{J}^p} & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & -I_{J_0^p} & 0 & -I_{J_0^p} & 0 \\ 0 & 0 & 0 & 0 & 0 & -I_{J_+^p} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -I_{J_-^p} \end{array}\right] \text{ has full column rank,} \tag{3.40}$$

which is equivalent to the condition that the rows of B with indices in J_0^p are linearly independent.

The LICQ condition for (D) is analogous to that for (3.22). Following the idea in Lemma 3.4.2, define the following three index sets by

$$J_1^d = \{i \mid \boldsymbol{u}_i^* = 0, |\boldsymbol{v}_i^*| < \rho\},\ J_2^d = \{i \mid \boldsymbol{u}_i^* = 0, |\boldsymbol{v}_i^*| = \rho\},\ J_3^d = \{i \mid \boldsymbol{u}_i^* \neq 0, |\boldsymbol{v}_i^*| = \rho\}, \tag{3.41}$$

then by the same technique used for proving the equivalence of (a) and (d) in Proposition 3.1, one can easily get that the LICQ holds for (D) at $(\boldsymbol{x}^*, \boldsymbol{v}^*)$ is equivalent to

$$\nabla^2 f(\boldsymbol{x}^*) \text{ is positive definite on the null space } \mathrm{Ker}(B_{J_1^d}).$$

This is the same as the *strong second-order sufficient condition* [82] for optimality in (P).

In order to state our results on the rate of convergence of the SSNAL algorithm, we recall that $\mathcal{T}^{-1}$ is said to be Lipschitz continuous around the origin with modulus $a \geq 0$ if there exists a neighborhood $\mathcal{N}$ of the origin such that for any $\boldsymbol{z} \in \mathcal{N}$, there exists a unique solution $\bar{\boldsymbol{y}}$ to the inclusion problem: $\boldsymbol{y} \in \mathcal{T}^{-1}(\boldsymbol{z})$, and $\|\boldsymbol{y}_1 - \boldsymbol{y}_2\| \leq a\|\boldsymbol{z}_1 - \boldsymbol{z}_2\|$ whenever $\boldsymbol{z}_1, \boldsymbol{z}_2 \in \mathcal{N}$ and $\boldsymbol{y}_1 \in \mathcal{T}^{-1}(\boldsymbol{z}_1),\ \boldsymbol{y}_2 \in \mathcal{T}^{-1}(\boldsymbol{z}_2)$. It is easy to see that the Lipschitz continuity of $\mathcal{T}^{-1}$ around the origin implies the Lipschitz continuity of $\mathcal{T}^{-1}$ at the origin.

Theorem 3.5.2 (Local convergence)**.** Let the SSNAL algorithm be executed with stopping criterion (B). If the LICQ for (P) and (D) holds respectively at $(\boldsymbol{x}^*, \boldsymbol{u}^*)$

and $(\boldsymbol{x}^*, \boldsymbol{v}^*)$, then $(\boldsymbol{x}^k, \boldsymbol{u}^k, \boldsymbol{v}^k) \to (\boldsymbol{x}^*, \boldsymbol{u}^*, \boldsymbol{v}^*)$, where $(\boldsymbol{x}^*, \boldsymbol{u}^*, \boldsymbol{v}^*)$ is the unique point satisfying the KKT condition (3.6), and one has

$$\|\boldsymbol{v}^{k+1} - \boldsymbol{v}^*\| \leq \theta_k \|\boldsymbol{v}^k - \boldsymbol{v}^*\| \quad \text{for all } k \text{ sufficiently large,}$$

where $\theta_k = [a_g(a_g^2 + \sigma_k^2)^{-1/2} + \delta_k](1 - \delta_k)^{-1} \to \theta_\infty = a_g(a_g^2 + \sigma_\infty^2)^{-1/2} < 1$ with $\sigma_\infty := \sup_k\{\sigma_k\}$ and a_g being the Lipschitz constant of $\mathcal{T}_g^{-1}$ around the origin. Moreover, the conclusions of Theorem 3.5.1 about $\{(\boldsymbol{x}^k, \boldsymbol{u}^k)\}$ are valid with $\vartheta_k = \delta_k \|\boldsymbol{v}^{k+1} - \boldsymbol{v}^k\|$ in (3.34).

If in addition, one also has the stopping criterion (B'), then

$$\|(\boldsymbol{x}^k, \boldsymbol{u}^k) - (\boldsymbol{x}^*, \boldsymbol{u}^*)\| \leq \theta_k' \|\boldsymbol{v}^{k+1} - \boldsymbol{v}^k\| \quad \text{for all } k \text{ sufficiently large,}$$

where $\theta_k' = a_l(1 + \delta_k')/\sigma_k \to \theta_\infty' = a_l/\sigma_\infty$ with a_l being the Lipschitz constant of $\mathcal{T}_l^{-1}$ around the origin.

Proof. Since the LICQ for (D) holds at $(\boldsymbol{x}^*, \boldsymbol{v}^*)$, we have that the strong second-order sufficient condition for optimality in (P) holds. This together with the LICQ for (P) at $(\boldsymbol{x}^*, \boldsymbol{u}^*)$ implies the strong regularity of the solution to the KKT system (3.6) [82, Theorem 4.1]. Thus, $\mathcal{T}_l^{-1}$ is Lipschitz continuous around the origin and so is $\mathcal{T}_g^{-1}$. The rest of the conclusions directly follows from [85, Theorem 5]. □

The following result establishes a finite convergence property for the variable $\boldsymbol{u}_{J_1^d}$.

Theorem 3.5.3. Suppose the conditions of Theorem 3.5.2 are satisfied, and $\sigma_\infty < \infty$. Then the sequence $\{(\boldsymbol{x}^k, \boldsymbol{u}^k, \boldsymbol{v}^k)\}$ generated by the SSNAL algorithm satisfies $\boldsymbol{u}_i^k = \boldsymbol{u}_i^* = 0 \; \forall i \in J_1^d$, for all but a finite number of iterations.

Proof. First note that for $i \in J_1^d$, we have $|\boldsymbol{v}_i^*| < \rho$, and

$$0 = \boldsymbol{u}_i^* = (B\boldsymbol{x}^*)_i.$$

Let $\boldsymbol{\eta}_i^* = (B\boldsymbol{x}^*)_i + \sigma_\infty^{-1}\boldsymbol{v}_i^* = \sigma_\infty^{-1}\boldsymbol{v}_i^*$. Thus

$$\varepsilon := \min_{i \in J_1^*} \{\rho - \sigma_\infty |\boldsymbol{\eta}_i^*|\} > 0. \tag{3.42}$$

For all $i \in J_1^d$, we have that

$$\begin{aligned}
&|\sigma_k \boldsymbol{\eta}_i^{k+1} - \sigma_\infty \boldsymbol{\eta}_i^*| = |\boldsymbol{v}_i^k + \sigma_k (B\boldsymbol{x}^{k+1})_i - \boldsymbol{v}_i^*| \\
&\le |\boldsymbol{v}_i^k - \boldsymbol{v}_i^*| + \sigma_k |(B\boldsymbol{x}^{k+1})_i - (B\boldsymbol{x}^*)_i| \quad (\text{Note that } (B\boldsymbol{x}^*)_i = 0) \\
&\le |\boldsymbol{v}_i^k - \boldsymbol{v}_i^*| + \sigma_\infty |(B\boldsymbol{x}^{k+1})_i - (B\boldsymbol{x}^*)_i| \ < \ \varepsilon \quad \forall\, k \ge \bar{k}, \qquad (3.43)
\end{aligned}$$

where (3.43) holds for some fixed integer $\bar{k}$ because $\boldsymbol{v}^k \to \boldsymbol{v}^*$ and $B\boldsymbol{x}^k \to B\boldsymbol{x}^*$ as $k \to \infty$.

Thus, for all $k \ge \bar{k}$, we have that

$$|\sigma_k \boldsymbol{\eta}_i^{k+1}| \le |\sigma_k \boldsymbol{\eta}_i^{k+1} - \sigma_\infty \boldsymbol{\eta}_i^*| + \sigma_\infty |\boldsymbol{\eta}_i^*| \ < \ \varepsilon + \sigma_\infty |\boldsymbol{\eta}_i^*| \ \le \ \rho. \qquad (3.44)$$

By the definition of $\boldsymbol{u}_i^{k+1}$, we have

$$\boldsymbol{u}_i^{k+1} = \boldsymbol{s}_{\rho/\sigma_k}(\boldsymbol{\eta}_i^{k+1}) = 0 \quad \forall\, k > \bar{k}. \qquad (3.45)$$

This completes the proof. □

3.6 Extensions

Strictly speaking, when f is the Huber loss function in (1.1), it does not satisfy our requirement that ∇f is continuously differentiable. However, because ∇f is strongly semismooth, the SSNAL algorithm can be applied to solve such a problem without any difficulty (where instead of $\nabla^2 f(\boldsymbol{x})$, we pick an element of $\partial \nabla f(\boldsymbol{x})$), and most of the convergence theory results can be extended to such a case with appropriate modifications.

For the group lasso problem (possibly overlapping), it has the form

$$\min \Big\{ f(\boldsymbol{x}) + \rho \sum_{i=1}^{K} \|B_i(\boldsymbol{x})\| \Big\} \qquad (3.46)$$

where B_i denotes the matrix corresponding to extracting a subvector from $\boldsymbol{x}$. Although the SSNAL algorithm we developed here cannot be directly applied to solve (3.46), nevertheless it is not difficult to extend the ideas and algorithmic framework to solve this problem (even with general matrices B_i) by reformulation it to the

following form:

$$\min\Big\{f(\boldsymbol{x})+\rho\sum_{i=1}^{K}\|\boldsymbol{u}_i\| \mid B_i(\boldsymbol{x})-\boldsymbol{u}_i=0, i=1,\ldots,K\Big\}. \tag{3.47}$$

From the associated augmented Lagrangian function

$$\mathcal{L}_\sigma(\boldsymbol{x},\boldsymbol{u};\boldsymbol{v})=f(\boldsymbol{x})+\sum_{i=1}^{K}\Big(\rho\|\boldsymbol{u}_i\|+\frac{\sigma}{2}\|B_i(\boldsymbol{x})-\boldsymbol{u}_i+\sigma^{-1}\boldsymbol{v}_i\|^2-\frac{1}{2\sigma}\|\boldsymbol{v}_i\|^2\Big) \tag{3.48}$$

we see that the key step in designing the SSNAL algorithm for solving (3.46) is to solve the following minimization problem analytically:

$$\min_{\boldsymbol{u}}\mathcal{L}_\sigma(\boldsymbol{x}^{k+1},\boldsymbol{u};\boldsymbol{v}^k)\equiv\sum_{i=1}^{K}\min_{\boldsymbol{u}_i}\Big(\rho\|\boldsymbol{u}_i\|+\frac{\sigma}{2}\|\boldsymbol{u}_i-B_i(\boldsymbol{x}^{k+1})-\sigma^{-1}\boldsymbol{v}_i^k\|^2\Big). \tag{3.49}$$

Fortunately, analogous to the Moreau-Yoshida regularization and proximal map of $\|\boldsymbol{x}\|_1=\sum_{i=1}^n|x_i|$, we can compute the Moreau-Yoshida regularization of $\sum_{i=1}^n\|\boldsymbol{x}_i\|$ analytically as follows:

$$\min\Big\{\sum_{i=1}^{n}\|\boldsymbol{y}_i\|+\frac{1}{2\nu}\|\boldsymbol{y}-\boldsymbol{x}\|^2\Big\}=\sum_{i=1}^{n}\min\Big\{\|\boldsymbol{y}_i\|+\frac{1}{2\nu}\|\boldsymbol{y}_i-\boldsymbol{x}_i\|^2\Big\}=\frac{1}{\nu}\sum_{i=1}^{n}\phi_\nu(\|\boldsymbol{x}_i\|)$$

and the correspondingly proximal map (unique minimizer) is given by

$$\boldsymbol{s}_i^\nu(\boldsymbol{x}_i)=\operatorname{sign}(\boldsymbol{x}_i)\max\{\|\boldsymbol{x}_i\|-\nu,0\},\quad i=1,\ldots,n \tag{3.50}$$

where $\operatorname{sign}(\boldsymbol{x}_i)=\boldsymbol{x}_i/\|\boldsymbol{x}_i\|$ if $\boldsymbol{x}_i\neq 0$, and $\operatorname{sign}(\boldsymbol{x}_i)=0$ if $\boldsymbol{x}_i=0$. By considering the projection map onto the ball $\{\boldsymbol{x}_i\mid\|\boldsymbol{x}\|_i\leq\nu\}$, which is given by

$$\boldsymbol{\pi}_i^\nu(\boldsymbol{x}_i)=\operatorname{sign}(\boldsymbol{x}_i)\min\{\|\boldsymbol{x}_i\|,\nu\},\quad i=1,\ldots,n, \tag{3.51}$$

we have that $\boldsymbol{x}_i=\boldsymbol{s}_i^\nu(\boldsymbol{x}_i)+\boldsymbol{\pi}_i^\nu(\boldsymbol{x}_i)$.

To apply the SSNCG algorithm to solve a subproblem analogous to (3.8), we need to consider the optimality condition analogous to (3.15):

$$\nabla f(\boldsymbol{x})+\sigma\sum_{i=1}^{K}B_i^T(\boldsymbol{\pi}_i^{\rho/\sigma}(\boldsymbol{\eta}_i))=0 \tag{3.52}$$

where $\boldsymbol{\eta}_i = B_i\boldsymbol{x} + \sigma^{-1}\boldsymbol{v}_i$. Furthermore, the elements of the generalized Hessian of the objective function of the subproblem have the form:

$$\nabla^2 f(\boldsymbol{x}) + \sigma \sum_{i=1}^{K} B_i^T D_i B_i, \quad \text{with } D_i = \begin{cases} I & \text{if } \|\boldsymbol{\eta}_i\| < \rho/\sigma \\ \frac{\rho/\sigma}{\|\boldsymbol{\eta}_i\|}\Big(I - \frac{\boldsymbol{\eta}_i\boldsymbol{\eta}_i^T}{\|\boldsymbol{\eta}_i\|^2}\Big) & \text{if } \|\boldsymbol{\eta}_i\| \geq \rho/\sigma. \end{cases} \tag{3.53}$$

Note that the elements of $\partial\boldsymbol{\pi}_i^{\nu}(\boldsymbol{x})$ are given as follows:

$$\partial\boldsymbol{\pi}_i^{\nu}(\boldsymbol{x}_i) = \begin{cases} I & \text{if } \|\boldsymbol{x}_i\| < \nu \\ \frac{\nu}{\|\boldsymbol{x}_i\|}\Big(I - \frac{\boldsymbol{x}_i\boldsymbol{x}_i^T}{\|\boldsymbol{x}_i\|^2}\Big) & \text{if } \|\boldsymbol{x}_i\| > \nu \\ \Big\{I - t\frac{\boldsymbol{x}_i\boldsymbol{x}_i^T}{\|\boldsymbol{x}_i\|^2} \mid t \in [0,1]\Big\} & \text{if } \|\boldsymbol{x}_i\| = \nu. \end{cases} \tag{3.54}$$

Chapter 4

First-order Methods

In this chapter, we summarize/design three types of first-order methods which are either known to be efficient for solving some special cases of structured B, or are expected to be efficient based on established theoretical results. As we shall see later, the computational efficiency of these first-order methods depends crucially (much more so than the SSNAL algorithm) on the problem structures of (1.1). Thus their practical performance are very much problem dependent, where for some classes of problems they may have good efficiency, while for some others, they may even stagnate without delivering an acceptable solution. Nevertheless, our SSNAL algorithm can always capitalize on the strength (of rapid initial progress) of first-order methods for generating a good starting point to warm-start the algorithm.

4.1 Alternating Direction Method of Multipliers

We start by adapting the widely-used alternating direction method of multipliers (ADMM) [37, 48, 50] for solving (1.1). The ADMM can either be used as a stand alone algorithm for solving (1.1), or it can be used to generate a good starting point to warm-start a second-order algorithm so as to accelerate the onset of the quadratic convergence phase.

It is generally difficult to solve (3.8) simultaneously for the optimal solutions $\boldsymbol{x}^{k+1}$ and $\boldsymbol{u}^{k+1}$ because its objective function is not separable in the variables $\boldsymbol{x}$ and $\boldsymbol{u}$ due to the coupling term $\|\mathcal{B}(\boldsymbol{x}) - \boldsymbol{u} + \sigma^{-1}\boldsymbol{v}^k\|^2$ in (3.7). This has lead to a rich literature on designing alternating algorithms to alleviate such a difficulty in the ALM. The most common approach is the alternating direction method of multipliers

(ADMM) [37, 48, 50], which can be described as follows. Given $\boldsymbol{v}^0, \boldsymbol{u}^0 \in \mathbb{R}^p$, iterate the following steps:

$$\boldsymbol{x}^{k+1} \approx \arg\min_{\boldsymbol{x}\in\mathbb{R}^n} \left\{ \mathcal{L}_\sigma(\boldsymbol{x}, \boldsymbol{u}^k; \boldsymbol{v}^k) \equiv f(\boldsymbol{x}) + \frac{\sigma}{2}\|B\boldsymbol{x} - \boldsymbol{u}^k + \sigma^{-1}\boldsymbol{v}^k\|^2 \right\} \tag{4.1}$$

$$\boldsymbol{u}^{k+1} \approx \arg\min_{\boldsymbol{u}\in\mathbb{R}^p} \left\{ \mathcal{L}_\sigma(\boldsymbol{x}^{k+1}, \boldsymbol{u}; \boldsymbol{v}^k) \equiv \rho\|\boldsymbol{u}\|_1 + \frac{\sigma}{2}\|\boldsymbol{u} - B\boldsymbol{x}^{k+1} - \sigma^{-1}\boldsymbol{v}^k\|^2 \right\} \tag{4.2}$$

$$\boldsymbol{v}^{k+1} = \boldsymbol{v}^k + \kappa\sigma(B\boldsymbol{x}^{k+1} - \boldsymbol{u}^{k+1}) \tag{4.3}$$

where $\kappa \in (0, (1+\sqrt{5})/2)$ is a fixed parameter. (Note that in practice, κ should be chosen to be at least 1 for faster convergence.) Observe that the subproblem (4.1) is an unconstrained smooth convex minimization problem for which a variety of algorithms can be used to solve it. We will discuss in detail later on how we solve the subproblem (4.1).

The subproblem (4.2) is a nonsmooth problem because of the nonsmooth term $\|\boldsymbol{u}\|_1$, but the simplicity of the objective function actually allows one to compute the optimal solution analytically. From (3.4), we get

$$\boldsymbol{u}^{k+1} = \boldsymbol{s}_{\rho/\sigma}(\boldsymbol{\eta}^k), \quad \boldsymbol{\eta}^k = B(\boldsymbol{x}^{k+1}) + \sigma^{-1}\boldsymbol{v}^k. \tag{4.4}$$

The ADMM described in (4.1)–(4.3) can be traced back to the pioneering work of Glowinski and Marrocco [50] and Gabay and Mercier [48], who also established the convergence of the exact variant of the ADMM. The convergence of the inexact variant described in (4.1)–(4.3) can be found in [37], and we state the convergence result below for the convenience of the reader.

Theorem 4.1.1. Consider the convex program (P) where B is assumed to have full column rank. Suppose we are given $\boldsymbol{v}^0, \boldsymbol{u}^0 \in \mathbb{R}^p$, $\sigma > 0$, and summable sequences of nonnegative numbers $\{\mu_k\}_{k=0}^\infty, \{\nu_k\}_{k=0}^\infty$. Suppose $\{\boldsymbol{x}^k\}_{k=1}^\infty$ and $\{\boldsymbol{u}^k\}_{k=0}^\infty$ conform, for all k, to

$$\|\boldsymbol{x}^{k+1} - \arg\min_{\boldsymbol{x}} \left\{ f(\boldsymbol{x}) + \frac{\sigma}{2}\|B\boldsymbol{x} - \boldsymbol{u}^k + \sigma^{-1}\boldsymbol{v}^k\|^2 \right\}\| \le \mu_k \tag{4.5}$$

$$\|\boldsymbol{u}^{k+1} - \arg\min_{\boldsymbol{u}} \left\{ \rho\|\boldsymbol{u}\|_1 + \frac{\sigma}{2}\|\boldsymbol{u} - B\boldsymbol{x}^{k+1} - \sigma^{-1}\boldsymbol{v}^k\|^2 \right\}\| \le \nu_k \tag{4.6}$$

Then the sequence $\{\boldsymbol{x}^k\}$ would converge to a solution $\boldsymbol{x}^*$ of (P) and $\{\boldsymbol{v}^k\}$ would converge to a solution $\boldsymbol{v}^*$ of (D). Furthermore, $\{\boldsymbol{u}^k\}$ converges to $B\boldsymbol{x}^*$.

Proof. By [37, Theorem 8] and noting that (P) has a KKT point. □

Since the subproblem (4.2) can be solved exactly, the condition (4.6) is satisfied with $\nu_k = 0$. For the subproblem (4.1), the optimality condition that $\boldsymbol{x}^{k+1}$ must satisfy is given by

$$\nabla f(\boldsymbol{x}) + \sigma B^T B\boldsymbol{x} = \sigma B^T(\boldsymbol{u}^k - \sigma^{-1}\boldsymbol{v}^k). \tag{4.7}$$

There is a variety of methods which can be employed to solve (4.1), and the choice would depend on the property of $f(\boldsymbol{x})$ and also the dimension n of the problem. When $f(\boldsymbol{x}) = \frac{1}{2}\|A\boldsymbol{x} - \boldsymbol{b}\|^2$, with $\nabla f(\boldsymbol{x}) = A^T(A\boldsymbol{x} - \boldsymbol{b})$, $\boldsymbol{x}^{k+1}$ can be computed by just solving a linear system of equations with coefficient matrix $A^TA+\sigma B^TB$. When $f(\boldsymbol{x})$ is not a convex quadratic function but is twice continuously differentiable and n is moderate (say $n < 1000$), we propose to compute an $\boldsymbol{x}^{k+1}$ that satisfies the condition (4.5) by Newton's method with back-tracking linesearch while using $\boldsymbol{x}^k$ as the starting point. Since $\boldsymbol{x}^k$ is close to $\boldsymbol{x}^{k+1}$ when k is sufficiently large, we would expect that only a few Newton iterations are needed to solve (4.1). In the event when n is large and ∇f is Lipschitz continuous, one may consider using a fast first-order method, such as the FISTA method in [5], to solve (4.1). Generally speaking, when f is such that the subproblem (4.1) cannot be solved analytically like in the case of a quadratic loss function, ADMM would not be an efficient method for solving (1.1) since it generally requires a large number of iterations to converge.

Recall that when $f(\boldsymbol{x}) = \frac{1}{2}\|A\boldsymbol{x} - \boldsymbol{b}\|^2$, a linear system of normal equation of the form $(A^TA + \sigma B^TB)\boldsymbol{x} = \boldsymbol{h}$ has to be solved repeatedly with a different right-hand side vector $\boldsymbol{h}$. If one were to solve the linear system by an iterative solver such as the preconditioned conjugate gradient (PCG) method, then it is advisable to construct a highly effective preconditioner so as to cut down the number of PCG steps needed to solve the linear system even if the initial construction overhead may be high. A highly effective preconditioner would potentially lead to a much cheaper total cost when the linear system is solved repeatedly. Of course, the design of efficient preconditioners for the normal equation is problem dependent and it is beyond the scope of the topic here.

4.2 Inexact Accelerated Proximal Gradient Method

When the function f in (1.1) has Lipschitz continuous gradient (with Lipschitz constant L_f), one can attempt to use the accelerated proximal gradient (APG) method in [5] to solve (1.1). This has been done in [65] for the fused lasso problem [100], where the nonseparable ℓ_1-regularization term essentially consists of the simple separable ℓ_1 term and the fused lasso structure term (more details of which will be given in Chapter 5). In the context of image restoration where the term $\|B\boldsymbol{x}\|_1$ is the TV regularization, [4] has also proposed to use the APG method to solve the problem.

The APG method is appealing because of its powerful iteration complexity result which guarantees the delivery of an ε-optimal solution (in terms of the objective value) in at most $O(L_f/\sqrt{\varepsilon})$ iterations. The basic template of the inexact version of the APG algorithm is given below.

Inexact APG algorithm.
Given a tolerance $\varepsilon > 0$, choose $\boldsymbol{w}^0 = \boldsymbol{x}^0 \in \mathbb{R}^n$ and set $t_0 = 1$. Iterate the following steps:

Step 1. Let $g(\boldsymbol{x}) := \|B\boldsymbol{x}\|_1$ for convenience. Compute

$$\boldsymbol{x}^{k+1} \approx \arg\min_{\boldsymbol{x}\in\mathbb{R}^n} \Big\{ \frac{L_f}{2}\|\boldsymbol{x} - \big(\boldsymbol{w}^k - L_f^{-1}\nabla f(\boldsymbol{w}^k)\big)\|^2 + \rho g(\boldsymbol{x}) \Big\}. \tag{4.8}$$

Step 2. Set $t_{k+1} = (1 + \sqrt{1 + 4t_k^2})/2$.

Step 3. Set $\boldsymbol{w}^{k+1} = \boldsymbol{x}^{k+1} + \frac{t_k - 1}{t_{k+1}}(\boldsymbol{x}^{k+1} - \boldsymbol{x}^k)$.

For the APG method to be practically competitive in solving (1.1), it is crucial for the problem to have conducive structures so that the subproblem (4.8) at each iteration of the APG method can be solved efficiently. In other word, the practical efficiency of the APG method hinges crucially on whether one can compute the proximal map $P_\nu^g(\cdot)$ of $g(\cdot)$ efficiently for a given parameter $\nu > 0$. We should mention that unlike the original algorithm in [5], the solution $\boldsymbol{x}^{k+1}$ of (4.8) need not be computed exactly, and this allows the problem to be solved by an iterative algorithm when $P_\nu^g(\cdot)$ cannot be computed analytically. In [58], it has been shown that as long as $\boldsymbol{x}^{k+1}$ is computed with sufficient accuracy progressively, the $O(L_f/\sqrt{\varepsilon})$ iteration complexity result remains valid, as stated in the following theorem.

Theorem 4.2.1. Let $\{\delta_k\}$ be a given summable sequence of nonnegative numbers. Suppose for each k, $\boldsymbol{x}^{k+1}$ is computed from (4.8) satisfying the condition that

$$\|L_f(\boldsymbol{x}^{k+1} - \boldsymbol{w}^k + L_f^{-1}\nabla f(\boldsymbol{w}^k)) + \rho \boldsymbol{z}^{k+1}\| \leq \sqrt{\frac{L_f}{2}}\,\frac{\delta_k}{t_k}$$

for some $\boldsymbol{z}^{k+1} \in \partial g(\boldsymbol{x}^{k+1})$. Then

$$f(\boldsymbol{x}^{k+1}) + \rho g(\boldsymbol{x}^{k+1}) - (f(\boldsymbol{x}^*) + \rho g(\boldsymbol{x}^*)) \leq \frac{4}{(k+1)^2}\Big(\sqrt{\frac{L_f}{2}}\|\boldsymbol{x}^0 - \boldsymbol{x}^*\| + \sum_{i=1}^{k}\delta_i\Big)^2.$$

Hence $\boldsymbol{x}^k$ is ε-optimal (in terms of the function value) if $k \geq O(\sqrt{L_f}/\sqrt{\varepsilon})$.

Proof. The result follows from [58, Theorem 2.1] with some adaptations to our problem (1.1). □

In both [65] and [4], for a given $\boldsymbol{x}$, the proximal map is computed via solving the dual of the proximal minimization problem: $\min_{\boldsymbol{y}}\{\frac{1}{2\nu}\|\boldsymbol{y} - \boldsymbol{x}\|^2 + g(\boldsymbol{y})\}$, which is given by

$$\frac{1}{2\nu}\|\boldsymbol{x}\|^2 - \min_{\boldsymbol{v}}\Big\{\frac{\nu}{2}\|B^T\boldsymbol{v} - \nu^{-1}\boldsymbol{x}\| \mid \|\boldsymbol{v}\|_\infty \leq 1\Big\}, \tag{4.9}$$

with $P_\nu^g(\boldsymbol{x}) = \boldsymbol{x} - \nu B^T\bar{\boldsymbol{v}}$, where $\bar{\boldsymbol{v}}$ is an optimal solution of the above dual problem. For the special case of the fused lasso regularization function, [65] is able to design an extremely efficient algorithm to compute $P_\nu^g(\boldsymbol{x})$ based on (4.9). However the algorithm is also very specialized, and does not appear to be applicable beyond the fused lasso function. For a general g, it appears that computing the proximal map $P_\nu^g(\boldsymbol{x})$ is expensive, either by solving (4.8) directly or via (4.9).

As we shall see in the next section, it is possible to overcome the bottleneck caused by the computation of the proximal map, but at the expense of worsening the iteration complexity guarantee from $O(1/\sqrt{\varepsilon})$ to $O(\sqrt{n}/\varepsilon)$.

4.3 Smoothing Accelerated Proximal Gradient Method

Recently, [27] proposed a smoothing proximal gradient (SPG method) to solve the problem stated in (5.1). This is a first-order method which essentially solves a smooth approximation of the original non-smooth problem (5.1) by using the well known accelerated proximal gradient (APG) method designed in [5]. The same idea

can of course be applied to (1.1). The key step is to approximate the nonsmooth function $\|\boldsymbol{x}\|_1$ by the following smoothed one:

$$h_\mu(\boldsymbol{x}) := \max_{\|\boldsymbol{y}\|_\infty \le 1} \Big\{ \langle \boldsymbol{y}, \boldsymbol{x} \rangle - \frac{\mu}{2}\|\boldsymbol{y}\|^2 \Big\} = \frac{1}{2\mu}\Big(\|\boldsymbol{x}\|^2 - \|\boldsymbol{s}_\mu(\boldsymbol{x})\|^2 \Big), \quad \boldsymbol{x} \in \mathbb{R}^n, \tag{4.10}$$

where $\mu > 0$ is the smoothing parameter, and $h_\mu(\boldsymbol{x}) \uparrow \|\boldsymbol{x}\|_1$ when $\mu \downarrow 0$. Indeed $h_\mu(\cdot)$ is a smooth approximation of $\|\boldsymbol{x}\|_1$ such that

$$\|\boldsymbol{x}\|_1 - \frac{n\mu}{2} \le h_\mu(\boldsymbol{x}) \le \|\boldsymbol{x}\|_1, \quad \forall\, \boldsymbol{x} \in \mathbb{R}^n \tag{4.11}$$

$$\nabla h_\mu(\boldsymbol{x}) = \frac{1}{\mu}(\boldsymbol{x} - \boldsymbol{s}_\mu(\boldsymbol{x})) = \frac{1}{\mu}\boldsymbol{\pi}_\mu(\boldsymbol{x}). \tag{4.12}$$

Based on the smooth function $h_\mu(\cdot)$, the problem (1.1) is approximated by the following smoothed problem:

$$\min_{\boldsymbol{x}} \; f(\boldsymbol{x}) + \rho h_\mu(B\boldsymbol{x}). \tag{4.13}$$

Note that in [27], which focuses on sparse structured lasso problems, whose regularization term consists of the simple separable ℓ_1-norm term and a structured lasso term, the authors only smoothed the structured lasso term, while keeping the simple nonsmooth ℓ_1 term intact. One can of course do the same for the problem (1.1) if the term $\|B\boldsymbol{x}\|_1$ has a similar structure. But for simplicity of exposition, here we have opted to smooth the entire term $\|B\boldsymbol{x}\|_1$.

To apply the APG method to solve (4.13), we need to assume that ∇f is Lipschitz continuous on $\mathbb{R}^n$ with Lipschitz constant L_f. Under this assumption, the objective function in (4.13) has Lipschitz continuous gradient with Lipschitz constant $L = L_f + \lambda_{\max}(B^TB)\rho/\mu$, where $\lambda_{\max}(B^TB)$ is the largest eigenvalue of B^TB. Note that each iteration of the APG method (with current iterate $\boldsymbol{x}^k$ and auxiliary iterate $\boldsymbol{w}^k$) requires the solution of the following subproblem:

$$\begin{aligned} \boldsymbol{x}^{k+1} &= \arg\min_{\boldsymbol{x}} \langle \nabla f(\boldsymbol{w}^k) + \rho B^T \nabla h_\mu(B\boldsymbol{w}^k),\, \boldsymbol{x} - \boldsymbol{w}^k \rangle + \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{w}^k\|^2 \\ &= \boldsymbol{w}^k - \frac{1}{L}\Big(\nabla f(\boldsymbol{w}^k) + \frac{\rho}{\mu} B^T \boldsymbol{\pi}_\mu(B\boldsymbol{w}^k) \Big). \end{aligned} \tag{4.14}$$

The template for the smoothing APG algorithm which can be used to solve (1.1) is given as follows.

Smoothing accelerated proximal gradient (SAPG) algorithm.
Given a tolerance $\varepsilon > 0$, set $\mu = \min\{\varepsilon/(n\rho), \rho\lambda_{\max}(B^TB)/L_f\}$. Choose $\boldsymbol{w}^0 = \boldsymbol{x}^0 \in \mathbb{R}^n$ and set $t_0 = 1$. Iterate the following steps:

Step 1. Compute $\boldsymbol{x}^{k+1} = \boldsymbol{w}^k - \frac{1}{L}\Big(\nabla f(\boldsymbol{w}^k) + \frac{\rho}{\mu}B^T\boldsymbol{\pi}_\mu(B\boldsymbol{w}^k)\Big)$.

Step 2. Set $t_{k+1} = (1 + \sqrt{1 + 4t_k^2})/2$.

Step 3. Set $\boldsymbol{w}^{k+1} = \boldsymbol{x}^{k+1} + \frac{t_k - 1}{t_{k+1}}(\boldsymbol{x}^{k+1} - \boldsymbol{x}^k)$.

By using the iteration complexity result for APG in [5] and the approximation result in (4.11), one can prove that the SAPG algorithm can compute an ε-optimal solution (in terms of the function value) for (1.1) in at most $O(\sqrt{n}/\varepsilon)$ iterations. The precise statement is given in the following theorem.

Theorem 4.3.1. Let $\varepsilon > 0$ be given. The solution $\boldsymbol{x}^k$ computed from the SAPG algorithm satisfies the following accuracy condition:

$$0 \le f(\boldsymbol{x}^k) + \rho\|B\boldsymbol{x}^k\|_1 - (f(\boldsymbol{x}^*) + \rho\|B\boldsymbol{x}^*\|_1) \le \varepsilon$$

when $k \ge O(\sqrt{n\rho^2\lambda_{\max}(B^TB)}/\varepsilon)$.

Proof. Let $\boldsymbol{x}^\mu$ be the optimal solution of (4.13). From (4.11), we have that

$$\begin{aligned} 0 \le \Delta^k &:= f(\boldsymbol{x}^k) + \rho\|B\boldsymbol{x}^k\|_1 - (f(\boldsymbol{x}^*) + \rho\|B\boldsymbol{x}^*\|_1) \\ &\le f(\boldsymbol{x}^k) + \rho h_\mu(B\boldsymbol{x}^k) + \frac{\mu n\rho}{2} - (f(\boldsymbol{x}^\mu) + \rho h_\mu(B\boldsymbol{x}^\mu)) \end{aligned}$$

where we have used the fact that

$$f(\boldsymbol{x}^*) + \rho\|B\boldsymbol{x}^*\|_1 \ge f(\boldsymbol{x}^*) + \rho h_\mu(B\boldsymbol{x}^*) \ge f(\boldsymbol{x}^\mu) + \rho h_\mu(B\boldsymbol{x}^\mu).$$

By the iteration complexity result in [5], we have that

$$f(\boldsymbol{x}^k) + \rho h_\mu(B\boldsymbol{x}^k) - (f(\boldsymbol{x}^\mu) + \rho h_\mu(B\boldsymbol{x}^\mu)) \le 2L\|\boldsymbol{x}^0 - \boldsymbol{x}^\mu\|^2/(k+1)^2$$

where $L = L_f + \lambda_{\max}(B^TB)\rho/\mu \le 2\lambda_{\max}(B^TB)\rho/\mu$. Thus

$$\Delta^k \le \frac{\mu n\rho}{2} + \frac{4\rho\lambda_{\max}(B^TB)\|\boldsymbol{x}^0 - \boldsymbol{x}^\mu\|^2}{\mu(k+1)^2} \le \varepsilon$$

if

$$k+1 \geq \frac{\sqrt{8n\rho^2\lambda_{\max}(B^TB)\|\boldsymbol{x}^0-\boldsymbol{x}^\mu\|^2}}{\varepsilon}$$

From here, the required result follows. □

Chapter 5

Applications of (1.1) in Statistics

5.1 Sparse Structured Regression Models

In this section, we list some typical sparse structured regression models that have drawn great interests from the statistics and machine learning communities.

Consider the linear regression of the data set $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$, where $\boldsymbol{x}_i \in \mathbb{R}^p$ are predictors and y_i are the responses. Let $X \in \mathbb{R}^{n\times p}$ denote the regression matrix of the n samples, and $\boldsymbol{y}$ be the observed response vector. Suppose $\boldsymbol{y} = X\boldsymbol{\beta}+\boldsymbol{\epsilon}$, and the noise is Gaussian, i.e., $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \sigma^2\mathbf{I})$. The main task is to recover $\boldsymbol{\beta}$ under certain sparsity assumption. It is well-known that the standard lasso regression [99] obtains a sparse estimation of $\boldsymbol{\beta}$ by solving the following convex optimization problem

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^p} f(\boldsymbol{\beta}) + \lambda\|\boldsymbol{\beta}\|_1,$$

where $f(\boldsymbol{\beta}) = \frac{1}{2}\|\boldsymbol{y} - X\boldsymbol{\beta}\|^2$ is the quadratic loss function, and the ℓ_1-regularization term promotes the zero-sparsity of the solution with the parameter λ that controls the sparsity level.

Although the standard lasso model has enjoyed great success in variable selection owing to the induced sparsity in $\boldsymbol{\beta}$, one obvious limitation is that it does not assume any structure among the input variables. Hence the selection may not be appropriate when it is applied to complex high-dimensional data. Therefore, different structural assumptions on the input variables have been introduced in later studies in order to capture the actual sparsity patterns, not necessary just on the individual components of $\boldsymbol{\beta}$ but some linear combinations of the components. For

this purpose, we introduce the following generic sparse regression framework with customizable sparse structures:

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^p} f(\boldsymbol{\beta}) + \lambda_1\Omega(\boldsymbol{\beta}) + \lambda_2\|\boldsymbol{\beta}\|_1, \tag{5.1}$$

where $\Omega(\boldsymbol{\beta})$ serves as the structured sparse constraints over the input variables. The scalars λ_1 and λ_2 are positive tuning parameters that control the structure and sparsity of the solution. Note that we can express (5.1) in the form (1.1) by setting $\|B(\boldsymbol{\beta})\|_1 = \Omega(\boldsymbol{\beta}) + \lambda_2/\lambda_1\|\boldsymbol{\beta}\|_1$.

Here, we mention several important choices of $\Omega(\boldsymbol{\beta})$ already introduced in the literature.

1. Elastic net [111]: $\Omega(\boldsymbol{\beta}) = \frac{1}{2}\|\boldsymbol{\beta}\|^2$.

2. Fused lasso [100]: $\Omega(\boldsymbol{\beta}) = \sum_{j=2}^{p} |\boldsymbol{\beta}_j - \boldsymbol{\beta}_{j-1}|$. A similar regularization term involving the second order finite difference of $\boldsymbol{\beta}$ also arises in the ℓ_1-version (studied in [59]) of the Hodrick-Prescott filtering problem [55] in macroeconomics to decompose a discrete time series into a cyclical component and a trend component.

3. Group lasso [107, 109]: $\Omega(\boldsymbol{\beta}) = \sum_{j=1}^{J} \|\vec{\boldsymbol{\beta}}_j\|$, where $\vec{\boldsymbol{\beta}}_j$ denotes a sub-vector of $\boldsymbol{\beta}$.

4. Graph-guided elastic net [95]. Such a regularization term is used in neuroimaging applications [52] where the regression problem has predictors constrained by an underlying graph structure. A brief description of the problem is given next. Let the pairwise relationship between the input variables be described by a graph $G = (V, \mathcal{E})$, where $V = \{1, ..., p\}$ denotes the variables of interests, and $\mathcal{E}$ denotes the set of edges among V. Moreover, let $w_{ij} \in \mathbb{R}$ denote the weight of the edge $(i, j) \in \mathcal{E}$, which corresponds to certain similarity measure between the variables i and j. Define the edge-vertex incidence matrix

$$\Lambda_{(i,j),k} = \begin{cases} |w_{ij}| & \text{if } k = i; \\ -w_{ij} & \text{if } k = j; \\ 0 & \text{otherwise,} \end{cases} \tag{5.2}$$

Then $\Omega(\boldsymbol{\beta}) = \frac{1}{2}\|\Lambda\boldsymbol{\beta}\|^2 = \sum_{(ij)\in\mathcal{E}} w_{ij}^2(\boldsymbol{\beta}_i - \text{sign}(w_{ij})\boldsymbol{\beta}_j)^2$.

5. Graph-guided fused lasso [26]: $\Omega(\boldsymbol{\beta}) = \|\Lambda\boldsymbol{\beta}\|_1$, where Λ is the edge-vertex incidence matrix defined as in (5.2).

6. Clustered lasso [78, 90]: $\Omega(\boldsymbol{\beta}) = \sum_{j=1}^{p}\sum_{k=1}^{j}|\boldsymbol{\beta}_j - \boldsymbol{\beta}_k|$. Such a regularization term is motivated by the desire to learn the group structure of the regression parameters $\{\boldsymbol{\beta}_i\}$.

7. OSCAR [7]: $\Omega(\boldsymbol{\beta}) = \sum_{k<j}\max\{|\boldsymbol{\beta}_j|, |\boldsymbol{\beta}_k|\} = \frac{1}{2}\sum_{j=1}^{p}\sum_{k=1}^{j}|\boldsymbol{\beta}_j+\boldsymbol{\beta}_k|+|\boldsymbol{\beta}_j-\boldsymbol{\beta}_k|$. Just like clustered lasso, the purpose here is to learn the group structure of the parameters $\{\boldsymbol{\beta}_i\}$.

In addition, the loss function f in (5.1) may not be limited to the quadratic loss function $\frac{1}{2}\|\boldsymbol{y} - X\boldsymbol{\beta}\|^2$. We also list some of the other commonly used loss functions.

1. Logistic loss function: $f(\boldsymbol{\beta}) = \sum_{i=1}^{n}\log(1 + \exp(-y_i\boldsymbol{\beta}^T\boldsymbol{x}_i))$, where $y_i \in \{\pm 1\}$. For the special case where $\|B\boldsymbol{x}\|_1 = \|\boldsymbol{x}\|_1$ in (1.1), various methods have been proposed to solve the problem where f is the logistic loss function. The recent paper [94] has given a nice summary of these methods. For example, [63] proposed a method that requires the solution of a bound constrained convex quadratic programming at each iteration; [49] proposed a method that is based on a (cyclic) coordinate descent method; [108] proposed a coordinate gradient descent method; [61] proposed an inexact interior-point method; [94] extended the fixed-point continuation (FPC) method [53]. However, it was observed in [94] that a pure FPC method is not efficient enough and a hybrid algorithm was proposed where it started with the FPC method and later switched to the Newton-type inexact interior-point method of [61].

2. Huber loss function: $f(\boldsymbol{\beta}) = \sum_{i=1}^{n}\phi_\varepsilon(y_i - \boldsymbol{\beta}^T\boldsymbol{x}_i)$, where $\phi_\varepsilon(t)$ is the Huber function defined as in (3.3). Note that unlike the quadratic loss and logistic loss functions, the Huber function is smooth but not twice continuously differentiable. The Huber loss function is commonly used for robust regression against sparse outliers. It is no coincidence that the Huber loss function is used since one can consider it as a way to remove sparse outliers in the responses $\{y_i\}$. Here we show that it arises naturally from the following regression model with sparse outliers detection:

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^p,\boldsymbol{\gamma}\in\mathbb{R}^n}\frac{1}{2}\|\boldsymbol{y} - X\boldsymbol{\beta} - \boldsymbol{\gamma}\|^2 + \varepsilon\|\boldsymbol{\gamma}\|_1 + \lambda_1\Omega(\boldsymbol{\beta}) + \lambda_2\|\boldsymbol{\beta}\|_1 \tag{5.3}$$

where $\boldsymbol{\gamma}$ attempts to capture the sparse outliers in the response vector. By minimizing with respect to $\boldsymbol{\gamma}$ first and using (3.4), it is easy to see that the resulting function is the Huber loss function.

3. Smoothed hinge loss function: $f(\boldsymbol{\beta}) = \sum_{i=1}^{n} \psi_\varepsilon(1 - y_i \boldsymbol{\beta}^T \boldsymbol{x}_i)$, where $\psi_\varepsilon(t) = (t + \sqrt{t^2 + \varepsilon^2})/2$ is a smoothing function of $\max\{t, 0\}$.

5.2 Results on Random Generated Data

In this section, we illustrate the efficiency and effectiveness of the proposed SSNAL algorithm for solving fused lasso and clustered lasso problems (5.1). We tested on random generated data, and compare the performance of the SSNAL method against the first-order methods (when applicable) presented in Chapter 4. We should mention that our focus here is to illustrate the efficiency of the proposed SSNAL algorithm for solving the large scale problems.

The loss function used is the quadratic loss, unless otherwise stated. The tuning parameters λ_1 and λ_2 in (5.1) were chosen based on numerical experience.

For fused lasso problems, we also compare with the very efficient APG based algorithm (called EFLA) recently proposed in [65]. The algorithm hinges on a novel subgradient finding algorithm for computing the proximal map associated with the fused lasso term, by cleverly exploiting the special structure of the fused lasso term. In our experiments, we use the software package SLEP (Sparse Learning with Efficient Projection) [64] wherein EFLA is implemented.

Note that for the EFLA and the SAPG algorithm, we stop the iterations whenever the relative change in $\boldsymbol{\beta}^k$ is less than 10^{-6}. The maximum number of iterations is capped at 20000 for the EFLA and SAPG algorithms.

For the ADMM and the SSNAL, the stopping condition is based on the KKT conditions (3.6), i.e., we terminate the algorithms when the primal and dual infeasibilities both fall below 10^{-6}. The maximum number of iterations is capped at 2000 for the ADMM and 50 the SSNAL algorithm.

For the SSNAL algorithm, the stopping condition for solving the subproblem (3.8) is also based on the KKT conditions (3.6). Although for ALM, the dual variable $\boldsymbol{v}$ is only updated in the outer iterations, to calculate the primal and dual infeasibilities within the inner iterations, we use the $\boldsymbol{v}$ computed from the latest

available $\boldsymbol{x}$ and $\boldsymbol{u}$. The iterations terminates when the primal and dual infeasibilities both fall below 10^{-6}, and the number of iterations is capped at 30.

All our algorithms are implemented in MATLAB (Version 7.14), and compiled on a MacBook Pro with Intel Core2 Duo 2.26GHz CPU and 4GB RAM.

The data used in this section are generated randomly from the following true model

$$\boldsymbol{y} = X\boldsymbol{\beta} + \sigma\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim N(\mathbf{0}, \mathbf{I}).$$

In all examples, the data consist of a training set and an independent test set, and we use the notation $\cdot/\cdot$ to describe the number of observations in the training and test sets respectively. Models were fitted on training data only, and the test error (the mean-squared error and the prediction error) were computed on the test data set.

5.2.1 Fused Lasso

We use the same data setting as in [105]. The true parameter vector $\boldsymbol{\beta}$ is generated according to

$$\beta_i = \begin{cases} 2 & i = 1, 2, ..., 20, 121, 122, ..., 125 \\ 3 & i = 41 \\ 1 & i = 71, 72, ..., 85 \\ 0 & \text{else.} \end{cases}$$

The noise level σ is chosen proportional to the size of $X\boldsymbol{\beta}$ relative to the size of noise; in this case, we use $\sigma = 0.1\|X\boldsymbol{\beta}\|/\|\boldsymbol{\epsilon}\|$. The observation numbers are 200/200.

In [105], the authors proposed a split Bregman method (which is equivalent to the ADMM in Chapter 4 for step-length equals to 1) to solve the fused lasso problems. It has been demonstrated to be more efficient than the standard convex optimization solvers, SQOPT and CVX (which is a MATLAB based modeling language relying on interior-point based solvers such as SDPT3 [101] and SeDuMi [98] to solve the resulting second-order cone programming problems). Therefore, here we only compare the performance of SSNAL against the ADMM, SAPG, and EFLA (in SLEP). The comparison results in terms of the CPU time, the mean-squared error and the prediction error are listed in Table 5.1.

The results show that EFLA is extremely efficient in dealing with the relatively easier problems, in the sense when there is no correlation between the predictors.

When the correlation $\hat{\rho}$ is nonzero, the performance of the ADMM, EFLA and SSNAL are comparable. The SSNAL shows its advantage when the problem size grows and when $\hat{\rho}$ increases. Note that for the case with 0-correlation among the predictors, the SAPG could not achieve a comparable accuracy as the other methods.

In Table 5.2, we list the relative residuals of the KKT conditions (3.6) for the ADMM and SSNAL used to solve (P). The columns under "pinf", "dinf" and "comp" in the table refer to the relative residuals of the three conditions in (3.6), respectively. Notice that the ADMM failed to achieve the required accuracy of 10^{-6} within 2000 iterations for only one case, where $\hat{\rho} = 0$ and $p = 40000$.

5.2.2 Clustered Lasso

In order to apply the proposed SSNAL algorithm to solve the clustered lasso problem (5.1), it is more convenient and efficient for us to express the structured sparse regularization term $\lambda_1\Omega(\boldsymbol{\beta})+\lambda_2\|\boldsymbol{\beta}\|_1$ using a linear map as follows. Given $W \in \mathbb{R}^{p\times p}$ such that $W_{ij} = W_{ji} > 0$ for all i, j, we define the linear map $\mathcal{B} : \mathbb{R}^p \to \mathbb{R}^{p\times p}$ by

$$(\mathcal{B}(\boldsymbol{\beta}))_{ij} = \begin{cases} W_{ij}(\boldsymbol{\beta}_i - \boldsymbol{\beta}_j) & \text{if } i \neq j \\ W_{ii}\boldsymbol{\beta}_i & \text{if } i = j. \end{cases} \tag{5.4}$$

Observe that the linear map $\mathcal{B}$ is injective, and $\mathcal{B}(\boldsymbol{\beta})$ can be written more compactly as $\mathcal{B}(\boldsymbol{\beta}) = W \circ (\boldsymbol{\beta}\mathbf{e}^T - \mathbf{e}\boldsymbol{\beta}^T + \mathrm{diag}(\boldsymbol{\beta}))$, where the operation $\circ$ denotes elementwise multiplication, and $\mathbf{e}$ is the vector of ones in $\mathbb{R}^p$. Note that the adjoint $\mathcal{B}^* : \mathbb{R}^{p\times p} \to \mathbb{R}^p$ of $\mathcal{B}$ is given by

$$\mathcal{B}^*(V) = (W \circ V)\mathbf{e} - (W \circ V^T)\mathbf{e} + \mathrm{diag}(V) \circ \mathrm{diag}(W),$$

and

$$\mathcal{B}^*\mathcal{B}(\boldsymbol{\beta}) = \Big(2(W \circ W)\mathbf{e} + \mathrm{diag}(W \circ W)\Big) \circ \boldsymbol{\beta} - 2(W \circ W)\boldsymbol{\beta}.$$

The examples presented in this subsection were mainly constructed based on the classic simulation scenarios used in [78, 90, 111]. Instead of using a specified noise level σ for each case, we adopt the same strategy as in last subsection and set $\sigma = 0.05\|X\boldsymbol{\beta}\|/\|\boldsymbol{\epsilon}\|$ for all examples. Moreover, since we want to focus on large-scale problems, we introduced a scaling variable k, i.e. with the specified parameter

Problem Size		$n = 200,\ p = 10000$			$n = 200,\ p = 20000$			$n = 200,\ p = 40000$		
$\hat{\rho}$	Method	Time	MSE	PE	Time	MSE	PE	Time	MSE	PE
0	ADMM	22.53	1.114e-1	1.481e-1	66.04	1.196e-1	1.517e-1	1527.47	1.232e-1	1.617e-1
	EFLA	8.08	1.116e-1	1.483e-1	17.77	1.196e-1	1.517e-1	72.66	1.235e-1	1.633e-1
	SAPG	150.49	1.238e-1	1.577e-1	306.91	1.397e-1	1.677e-1	600.75	1.565e-1	1.892e-1
	SSNAL	34.01	1.114e-1	1.482e-1	56.00	1.196e-1	1.517e-1	167.36	1.242e-1	1.639e-1
0.2	ADMM	24.08	2.341e-1	1.155e-1	64.47	2.368e-1	1.207e-1	180.79	2.281e-1	1.332e-1
	EFLA	14.41	2.343e-1	1.155e-1	34.89	2.352e-1	1.205e-1	122.74	2.304e-1	1.337e-1
	SAPG	151.22	2.342e-1	1.159e-1	299.32	2.368e-1	1.205e-1	597.39	2.289e-1	1.334e-1
	SSNAL	20.57	2.342e-1	1.155e-1	45.90	2.369e-1	1.207e-1	138.62	2.281e-1	1.332e-1
0.4	ADMM	26.16	3.649e-1	1.131e-1	70.73	3.680e-1	1.179e-1	188.33	3.529e-1	1.312e-1
	EFLA	21.94	3.659e-1	1.132e-1	44.37	3.746e-1	1.185e-1	181.14	3.570e-1	1.317e-1
	SAPG	152.49	3.650e-1	1.133e-1	325.00	3.678e-1	1.179e-1	658.23	3.533e-1	1.312e-1
	SSNAL	19.91	3.649e-1	1.131e-1	56.47	3.680e-1	1.179e-1	123.54	3.529e-1	1.312e-1
0.6	ADMM	22.66	4.874e-1	1.094e-1	62.07	4.909e-1	1.127e-1	147.61	4.767e-1	1.261e-1
	EFLA	19.77	4.872e-1	1.094e-1	71.18	4.914e-1	1.128e-1	156.76	4.756e-1	1.259e-1
	SAPG	153.57	4.876e-1	1.095e-1	315.32	4.910e-1	1.127e-1	581.88	4.770e-1	1.261e-1
	SSNAL	17.75	4.874e-1	1.094e-1	48.48	4.910e-1	1.127e-1	111.06	4.767e-1	1.261e-1
0.8	ADMM	22.74	6.827e-1	1.060e-1	55.84	6.766e-1	1.083e-1	232.62	6.143e-1	1.167e-1
	EFLA	30.93	6.851e-1	1.061e-1	66.34	6.816e-1	1.085e-1	315.00	6.130e-1	1.167e-1
	SAPG	153.62	6.828e-1	1.060e-1	306.53	6.768e-1	1.083e-1	643.25	6.144e-1	1.167e-1
	SSNAL	17.94	6.827e-1	1.060e-1	45.02	6.765e-1	1.083e-1	142.21	6.139e-1	1.167e-1

Table 5.1: Running time (CPU seconds), MSE and PE for **fused lasso** problems of various problem size p and n, and different constant correlation $\hat{\rho}$ between the predictors.

Problem Size		$n = 200,\ p = 10000$			$n = 200,\ p = 20000$			$n = 200,\ p = 40000$		
$\hat{\rho}$	Method	pinf	dinf	comp	pinf	dinf	comp	pinf	dinf	comp
0	ADMM	9.55e-7	9.77e-7	2.66e-15	9.80e-7	9.01e-7	1.20e-14	4.08e-3	4.43e-3	1.11e-15
	SSNAL	6.17e-7	5.73e-7	1.08e-13	6.44e-7	9.02e-7	1.68e-13	6.28e-7	7.23e-7	2.14e-13
0.2	ADMM	7.85e-7	9.87e-7	2.00e-15	8.11e-7	9.90e-7	1.78e-15	9.91e-7	8.99e-7	4.44e-16
	SSNAL	8.46e-7	8.04e-8	3.13e-14	9.82e-7	1.10e-7	2.99e-13	8.22e-7	2.60e-7	2.80e-13
0.4	ADMM	9.98e-7	9.29e-7	2.22e-16	9.85e-7	7.81e-7	6.66e-16	8.98e-7	6.84e-7	7.17e-14
	SSNAL	9.40e-7	2.96e-8	5.17e-14	8.78e-7	2.89e-7	2.61e-13	7.77e-7	7.28e-7	3.45e-13
0.6	ADMM	9.23e-7	9.51e-7	4.44e-16	9.70e-7	9.64e-7	4.44e-16	7.23e-7	9.97e-7	2.66e-15
	SSNAL	3.80e-7	4.31e-8	2.66e-14	3.64e-7	5.32e-8	1.39e-13	9.92e-7	4.69e-8	4.06e-14
0.8	ADMM	7.62e-7	9.94e-7	4.44e-16	9.85e-7	8.54e-7	6.66e-16	9.52e-7	9.25e-7	4.37e-13
	SSNAL	9.74e-7	1.64e-9	2.44e-15	2.39e-7	7.55e-9	2.03e-13	1.81e-7	2.73e-7	1.84e-12

Table 5.2: Infeasibility of the KKT conditions for **fused lasso** problems of various problem size p and n, and different constant correlation $\hat{\rho}$ between the predictors.

vector $\boldsymbol{\beta} \in \mathbb{R}^p$ in each case, the actual $\boldsymbol{\beta}$ is of size $p \times k$, and every component of $\boldsymbol{\beta}$ is repeated for k times consecutively. The corresponding number of observations is also scaled up to a factor of k.

1. The first setting is specified by the parameter vector

$$\boldsymbol{\beta} = (3, 1.5, 0, 0, 0, 2, 0, 0)^T.$$

   The correlation between the i-th and the j-th predictor is

$$\text{corr}(i, j) = 0.9^{|i-j|}, \ \forall \ i, j \in \{1, ..., 8\}. \tag{5.5}$$

   The observation numbers are 20/20.

2. In this setting, we have $p = 20$ predictors. The parameter vector is structured into blocks:

$$\boldsymbol{\beta} = (\underbrace{0, ..., 0}_{5}, \underbrace{2, ..., 2}_{5}, \underbrace{0, ..., 0}_{5}, \underbrace{2, ..., 2}_{5})^T.$$

   The correlation between two predictors X_i and X_j is given by $\text{corr}(i, j) = 0.3$. The observation numbers are 50/50.

3. This setting consists of $p = 20$ predictors. The parameter vector is given by

$$\boldsymbol{\beta} = (5, 5, 5, 2, 2, 2, 10, 10, 10, \underbrace{0, ..., 0}_{11})^T.$$

   Within each of the first three blocks of 3 variables, the correlation between the two predictors is 0.9, but there is no correlation between these blocks. The observation numbers are 50/50.

4. The fourth setting consists of $p = 13$ predictors. The parameter vector is structured into many small clusters:

$$\boldsymbol{\beta} = (0, 0, -1.5, -1.5, -2, -2, 0, 0, 1, 1, 4, 4, 4)^T.$$

   The correlation between the i-th and the j-th predictor is

$$\text{corr}(i, j) = 0.5^{|i-j|}, \ \forall \ i, j \in \{1, ..., 13\}. \tag{5.6}$$

The observation numbers are 20/20.

5. The fifth setting is the same as the fourth one, but with a higher correlation between the predictors: $\text{corr}(i,j) = 0.9^{|i-j|},\ \forall\ i,j \in \{1,...,13\}$.

6. In the last setting, we have $p = 16$ predictors. The parameter vectors is structured such that big clusters coexist with small ones:

$$\boldsymbol{\beta} = (\underbrace{0,...,0}_{3},\underbrace{4,...,4}_{5},\underbrace{-4,...,-4}_{5},2,2,-1)^T.$$

The predictors are possibly negatively correlated: $\text{corr}(i,j) = (-1)^{(i-j)}0.8$. The observation numbers are 30/30.

For all six examples, we tested on three different problem sizes, namely $k = 1$, which is the same size as they were designed in [78, 90, 111], and $k = 10, 100$ for our purpose of testing on large-scale problems. The comparison results in terms of the CPU time, the mean-squared error and the prediction error are listed in Table 5.3. The results again confirm the advantage of the SSNAL when the problem size is large, especially when $k = 100$. For small size problems, all three algorithms are comparable. Note that in Example 3, the ADMM did not achieve a similar accuracy as the other two methods.

Moreover, a comparison of the ADMM and the SSNAL in achieving the KKT conditions for solving (P), as presented in Table 5.4, shows that for clustered lasso problems where the sparse structure of the predictors is more complicated than that of the fused lasso, the ADMM generally could not achieve the required accuracy within 2000 iterations, whereas the SSNAL is more robust in this sense.

Problem Size		$k = 1$			$k = 10$			$k = 100$		
e.g.	Method	Time	MSE	PE	Time	MSE	PE	Time	MSE	PE
1	ADMM	0.77	1.30e-1	5.93e-2	1.78	1.26e-1	5.44e-2	148.16	7.53e-2	5.11e-2
	SPG	1.60	1.29e-1	5.90e-2	1.81	1.27e-1	5.44e-2	53.21	7.52e-2	5.11e-2
	SSNAL	0.84	1.30e-1	5.93e-2	1.27	1.26e-1	5.44e-2	85.03	7.53e-2	5.11e-2
2	ADMM	1.15	5.22e-2	5.91e-2	8.22	1.72e-1	5.71e-2	872.96	5.29e-1	5.55e-2
	SPG	1.41	5.19e-2	5.91e-2	5.23	1.72e-1	5.71e-2	658.76	5.29e-1	5.55e-2
	SSNAL	0.93	5.22e-2	5.91e-2	2.30	1.72e-1	5.71e-2	3393.43	5.29e-1	5.55e-2
3	ADMM	1.55	9.09e-2	6.68e-2	8.47	8.16e-2	5.48e-2	873.30	3.39e-2	5.26e-2
	SPG	1.26	9.09e-2	6.69e-2	1.64	5.04e-2	5.50e-2	2314.38	1.79e-2	5.26e-2
	SSNAL	0.95	9.09e-2	6.68e-2	3.60	4.59e-2	5.49e-2	1277.52	1.79e-2	5.26e-2
4	ADMM	0.75	1.45e-1	1.07e-1	3.78	9.09e-2	7.79e-2	318.54	1.01e-1	8.03e-2
	SPG	1.51	1.45e-1	1.07e-1	3.80	9.10e-2	7.81e-2	281.89	1.01e-1	8.03e-2
	SSNAL	0.87	1.45e-1	1.07e-1	1.50	9.09e-2	7.79e-2	179.06	1.01e-1	8.03e-2
5	ADMM	0.82	1.81e-1	7.55e-2	2.29	1.14e-1	5.62e-2	319.01	9.11e-2	5.59e-2
	SPG	1.36	1.83e-1	7.54e-2	2.11	1.14e-1	5.62e-2	128.79	9.13e-2	5.59e-2
	SSNAL	0.87	1.81e-1	7.55e-2	1.69	1.14e-1	5.62e-2	363.76	9.11e-2	5.59e-2
6	ADMM	0.82	7.86e-2	9.44e-2	5.31	8.06e-2	8.95e-2	516.75	8.63e-2	1.02e-1
	SPG	2.33	7.92e-2	9.44e-2	6.80	8.04e-2	8.94e-2	996.52	8.59e-2	1.01e-1
	SSNAL	0.92	7.86e-2	9.44e-2	1.78	8.06e-2	8.95e-2	191.70	8.63e-2	1.02e-1

Table 5.3: Running time (CPU seconds), MSE and PE for **clustered lasso** problems of various size $p \times k$ and n, and different correlations between the predictors.

Problem Size		$k = 1$			$k = 10$			$k = 100$		
e.g.	Method	pinf	dinf	comp	pinf	dinf	comp	pinf	dinf	comp
1	ADMM	6.99e-7	9.58e-10	2.42e-7	6.30e-7	3.36e-10	9.87e-7	1.01e-4	5.75e-10	1.77e-4
	SSNAL	9.35e-7	1.50e-10	1.11e-14	6.88e-7	1.90e-8	7.97e-13	2.39e-7	1.10e-7	2.65e-10
2	ADMM	9.99e-7	7.18e-11	8.95e-7	3.47e-5	5.84e-12	6.09e-5	3.65e-3	1.92e-11	1.26e-2
	SSNAL	6.08e-7	1.51e-8	7.53e-14	7.13e-7	4.39e-8	2.27e-10	6.08e-7	9.89e-7	9.83e-7
3	ADMM	1.00e-6	1.10e-10	5.69e-7	4.00e-2	4.57e-8	2.95e-1	1.83e-1	3.76e-10	7.44e-1
	SSNAL	4.94e-7	2.98e-8	2.66e-12	4.61e-7	1.07e-7	2.48e-11	9.99e-8	7.01e-8	1.15e-8
4	ADMM	3.11e-7	1.33e-9	9.32e-7	9.99e-7	9.85e-11	9.73e-7	2.29e-4	2.31e-10	1.37e-4
	SSNAL	6.83e-7	6.24e-9	2.53e-14	4.00e-7	2.24e-8	1.27e-11	5.57e-7	1.25e-7	4.70e-10
5	ADMM	6.22e-7	8.16e-9	8.54e-7	7.87e-7	1.61e-10	9.85e-7	4.51e-4	1.16e-9	5.08e-4
	SSNAL	9.26e-7	7.19e-10	2.00e-15	9.99e-7	1.58e-7	7.94e-13	9.87e-7	5.56e-7	3.82e-10
6	ADMM	5.83e-7	1.77e-9	9.32e-7	7.64e-6	8.66e-10	5.09e-6	1.34e-4	6.04e-10	1.23e-4
	SSNAL	9.88e-7	3.53e-8	9.77e-14	5.95e-7	7.22e-8	2.10e-11	6.62e-7	7.24e-8	1.57e-9

Table 5.4: Infeasibility of the KKT conditions for **clustered lasso** problems of various size $p \times k$ and n, and different correlations between the predictors.

Chapter 6

Applications of (1.1) in Image Processing

In this chapter, we propose a simple model for image restoration with mixed or unknown noises. The proposed model falls in the framework of the general non-separable ℓ_1-regularized problem (1.1). Since a moderately accurate solution is usually sufficient for image processing problems, we use an accelerated proximal gradient (APG) algorithm to solve the inner subproblem. The chapter is organized as follows. First, we will give a brief introduction on image restorations models. Then we will present the numerical results for various image restorations with mixed noise and examples on noise removal of real digital colour images. Comparisons with some of the available models for removing noises such as single type of noise, mixed Poisson-Gaussian noise, and impulse noise mixed with Gaussian noise are also given. Finally, some additional remarks on our proposed model and numerical algorithm will be addressed.

6.1 Image Restorations

Image restoration is often formulated as an inverse problem, which amounts to recovering an unknown true image u from a degraded image (or a measurement) b given by

$$b \approx Hu, \tag{6.1}$$

where H is a linear operator denoting the degradation operations and Hu is also possibly contaminated by random noises. Such noises can be the additive noise (e.g. Gaussian noise, impulse noise, in which case $b = Hu + \epsilon$), or Poisson noise and

other multiplicative noise. A typical H can be the identity operator, a convolution or a projection, and the corresponding inverse problems are known as denoising, deblurring and inpainting.

To restore u, one of the most popular approach is to solve a minimization problem of the following form

$$\min_u R_1(u) + R_2(u) \tag{6.2}$$

where $R_1(u)$ denotes certain data fitting term derived according to the assumed noise type and $R_2(u)$ is a regularization term that imposes the prior on u. The latter is necessary due to the ill-posedness of the inverse problem.

Typically, the regularization term $R_2(u)$ is determined by the prior assumptions imposed on the underlying solutions. One of the assumptions commonly used is the sparsity of the underlying solutions in some transformed domain. Such transforms can be gradient (e.g. the total variation (TV) model), wavelet tight frames, local cosine transforms etc. Since the TV model is closely related to the wavelet tight frame model (see [15]) and since wavelet tight frame model has proven to be efficient in image restorations (see e.g. [33,91]), we use the sparsity in the wavelet tight frame transform domain as the prior assumption on the underlying solutions. For this, we will take $R_2(u) = \|Wu\|_1$, where W is the wavelet tight frame transform.

Generally, the choice of the data fitting term $R_1(u)$ depends on the specified noise distribution. For example, the ℓ_2-norm based distance function

$$R_1(u) = \|Hu - b\|_2^2 \tag{6.3}$$

is used for additive Gaussian noise, which averages out the noise. When image is corrupted by impulse noise, a typical choice of $R_1(u)$ is the ℓ_1-norm distance function

$$R_1(u) = \|Hu - b\|_1, \tag{6.4}$$

which leads to the median of the data. Note that the impulse noise can be considered as outliers, and a median approximation is a good choice since it is robust to outliers. For Poisson noise and multiplicative Gamma noise, the Bayesian maximum a posterior (MAP) likelihood estimation approach gives rise to the Csiszár's

I-divergence of Hu from b [62, 89, 97] as the data fitting term, i.e.

$$R_1(u) = \sum_{i=1}^{n} \left((Hu)_i - b_i \log(Hu)_i \right). \tag{6.5}$$

However, since the noise in an image seldom appears from a single distribution, nor could one determine the noise type in reality, a specific data fitting term for a given noise may not work for mixed or unknown noises. We aim to find a simple model which can remove mixed type of noises as well as unknown type of noises, while at the same time also works comparably well with the model whose data fitting term is designed for a specific given noise type.

The data fitting term $R_1(u)$ suggested here is surprisingly simple. It is the weighted sum of ℓ_1-norm and ℓ_2-norm based distances, which leads to the following model:

$$\min_{u \in \mathbb{R}^n} \lambda_1 \|Hu - b\|_1 + \frac{\lambda_2}{2} \|Hu - b\|_2^2 + \rho \|Wu\|_1, \tag{6.6}$$

where λ_1, λ_2 and ρ are nonnegative parameters. Note that the parameter λ_1 in (6.6) is always fixed as 1 in our numerical simulations, but we keep it in (6.6) for ease of discussion. While the model (6.6) looks too simple, the numerical simulation results show that this model together with the proposed numerical algorithm can efficiently remove various mixed noises and unknown noises. When it is applied to remove a given type of noise, such as additive Gaussian noise, impulse noise, Poisson noise and multiplicative Gamma noise, it performs as well as those models whose data fitting terms are chosen according to the statistical distribution of the noise.

6.2 Results on Image Restorations with Mixed Noises

This section is mainly devoted to numerical simulation of image restorations in the presence of mixed noise. It is easy to see that the proposed model (6.6) falls in the framework of the general non-separable ℓ_1-regularized problem (1.1) as shown below. By letting $z_1 = Hu - b$ and $z_2 = Wu$, (6.6) can be reformulated in the form

(P) as follows:

$$\begin{aligned} \min \quad & \frac{\lambda_2}{2}\|Hu-b\|_2^2+\lambda_1\|z_1\|_1+\rho\|z_2\|_1 \\ \text{s.t.} \quad & Hu-z_1=b \\ & Wu-z_2=0. \end{aligned}$$

Here we use an accelerated proximal gradient (APG) algorithm to solve the inner subproblem because for image processing problems, the optimization model usually serves as a guide that leads to a sparse approximate optimal solution and it is not necessary to solve the model with high accuracy. We show the efficiency, effectiveness and robustness of model (6.6) by applying the ALM-APG algorithm to restore images with a wide range of mixed noise. Specifically, we consider image restorations with the mixture of Gaussian, Poisson, and impulse noises. To the best of our knowledge, there is no discussion of image restorations with such a wide range of mixed noises in the current literature. It should be noted that video denoising for the mixture of these three noises has been considered in [57]. However, the method given there is not applicable for the image case, since it relies heavily on the temporal direction information. In addition, some denoising examples of real life digital color images are also presented to show that the proposed method is capable of removing unknown noises.

The performance on restorations of synthetic data is measured by the PSNR value defined as

$$\text{PSNR}=10\log_{10}\left(\frac{I_{\max}^2}{\text{MSE}}\right), \tag{6.7}$$

where $I_{\max}$ is the maximum intensity of the original image. In the presence of Poisson noise, the maximum intensity of the original noise free image is varied in order to create images with different levels of Poisson noise.

The tight wavelet frame transform W used here is generated from piecewise linear B-spline constructed via the unitary extension principle [87]. The wavelet transform W and its inverse related to the wavelet frame system via the tight wavelet frame decomposition and reconstruction operators have been briefly introduced in Chapter 2. More details can be found in the survey paper [91] and the long note [33] and the references therein.

6.2.1 Synthetic Image Denoising

We start with the simpler case of the mixed noise removal. The key point for the noise removal here is that we do not need to know a priori what kind of noises contaminate the image. The aim is to remove noises in real images, and examples will be given the next subsection.

The mixed noise considered in this section includes additive Gaussian noise, Poisson noise and impulse noise. The Poisson noise in this experiment was generated using the "poissrnd" function in MATLAB with the input image scaled to the maximum intensity ($\mathrm{I_{max}}$) as specified in each experiment. For the impulse noise, we only consider the random-valued impulse noise, because a pixel contaminated by such an impulse noise is not as distinctively an outlier as that contaminated by the salt-and-pepper noise, and consequently is more difficult to be detected. The random-valued impulse noise is defined as follows: with probability r, the pixel value u_{ij} is altered to be a uniform random number in the interval between the minimum intensity and the maximum intensity of the image. For all cases, impulse noise was the last to be added.

First, we consider the case when images' maximum intensity is not rescaled. The results in terms of PSNR using model (6.6) are summarized in Table 6.1 and Table 6.2, and some of the restored images are presented in Figure 6.1 and Figure 6.2. The results show that the model together with the proposed ALM-APG algorithm presented in this paper are effective in removing random-valued impulse noise mixed with Poisson noise and Gaussian noise. In the case when all three types of noises are involved, we consider both cases of generating Poisson noise before and after adding Gaussian noise. It turns out that the performance of our method is robust regardless the intrinsic distribution of the image noise.

Second, we conduct a more extensive test of our method in removing mixed noise of the three types as previously discussed, where the Poisson noise is generated from noise-free images rescaled to the maximum intensity ranging from 120 to 1. In the cases when both Gaussian noise and Poisson noise are involved, Gaussian noise is added after Poisson noise with standard deviation $\sigma = \mathrm{I_{max}}/10$. The results in terms of PSNR are summarized in Table 6.3. The results show that the simple model (6.6) together with the proposed ALM-APG algorithm is, in general, effective in removing mixed impulse noise, Poisson noise and Gaussian noise at different levels.

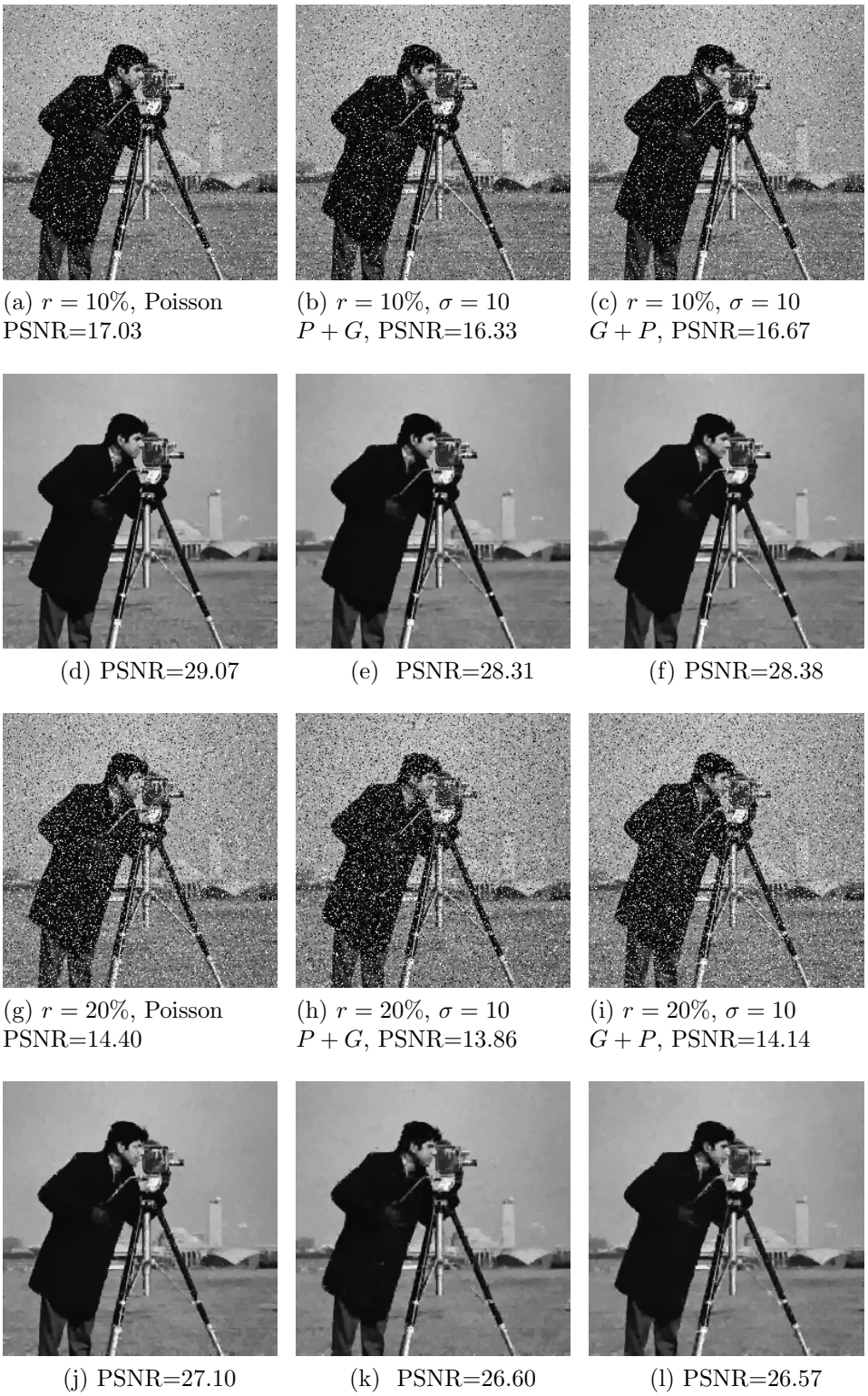
The choice of parameters were $\lambda_1 = 1$, $\lambda_2 = 0.01 \sim 0.1$ and $\rho = 1 \sim 2$. The

Random-valued impulse noise (r)	10%		20%		40%	
Gaussian noise (σ)	0	10	0	10	0	10
Poisson+Gaussian	29.07	28.31	27.10	26.60	23.46	23.68
Gaussian+Poisson	28.38		26.57		23.35	

Table 6.1: Denoising results (PSNR) for the image "Cameraman", in the presence of random-valued impulse noise, Gaussian noise and Poisson noise at image peak intensity of 255.

Image	Baboon		Boat		Bridge		Barbara512	
Random-valued impulse noise (r)	10%	20%	10%	20%	10%	20%	10%	20%
Poisson+Gaussian	25.26	24.08	27.60	26.17	26.38	25.23	27.20	25.58
Gaussian+Poisson	25.23	24.13	27.57	26.19	26.26	25.06	27.24	25.54

Table 6.2: Denoising results (PSNR) for various testing images, in the presence of random-valued impulse noise, Gaussian noise with standard deviation $\sigma = 10$ and Poisson noise at image peak intensity of 255.



(a) $r = 10\%$, Poisson PSNR=17.03
(b) $r = 10\%$, $\sigma = 10$ $P + G$, PSNR=16.33
(c) $r = 10\%$, $\sigma = 10$ $G + P$, PSNR=16.67
(d) PSNR=29.07
(e) PSNR=28.31
(f) PSNR=28.38
(g) $r = 20\%$, Poisson PSNR=14.40
(h) $r = 20\%$, $\sigma = 10$ $P + G$, PSNR=13.86
(i) $r = 20\%$, $\sigma = 10$ $G + P$, PSNR=14.14
(j) PSNR=27.10
(k) PSNR=26.60
(l) PSNR=26.57

Figure 6.1: Denoising results for the image "Cameraman", in the presence of random-valued impulse noise, Poisson noise and Gaussian noise. The first and third rows are the noisy images, and the second and fourth rows are the corresponding denoised images.

(a) $r = 10\%$, $\sigma = 10$ $P + G$, PSNR=18.37 (b) PSNR=25.26 (c) $r = 10\%$, $\sigma = 10$ $P + G$, PSNR=16.71 (d) PSNR=27.20

(e) $r = 20\%$, $\sigma = 10$ $P + G$, PSNR=14.87 (f) PSNR=26.17 (g) $r = 20\%$, $\sigma = 10$ $P + G$, PSNR=13.97 (h) PSNR=25.23

Figure 6.2: Denoising results for various testing images, in the presence of random-valued impulse noise, Poisson noise and Gaussian noise.

value of ρ depends on the general noisiness of the image: the noisier the image is, the larger ρ shall be chosen; however, the value of λ_2 mainly depends on the impulse noise level, where smaller value is preferred when the impulse noise level gets higher.

6.2.2 Real Image Denoising

In digital color photos, there are two most significant sources of noise: the photon shot noise due to the random striking of the photons on the image sensor, and the leakage current due to the additional electrical signal generated by the semiconductor when converting energy from photons to electrical energy. Besides, interpolation of the captured partial color data to complete the RGB channels, quantization and artifacts caused by JPEG format and the build-in sharpening, denoising functions in cameras etc, make it difficult or rather impossible to model the noise. Therefore, real image denoising problems are much more challenging than those of synthetic data.

The main difficulty behind the noise removal for real images is that there is no prior knowledge of the noise and its statistical distribution, which itself is the result of a mixture of different noises. Hence, models based on a specific type of noise distribution is hard to be effective. Since our model does not assume any prior statistical distribution of the noise, it has the potential to perform well in real image denoising. Here, we show the promise of the method via a few examples.

Most digital color images are in the RGB color space. It is known that due to the uneven distribution of the noise in each channel, by denoising each channel separately, one tends to excessively denoise the blue channel, which can lead to undesirable color artifacts. A standard practice is to transform the RGB color space to YCrCb color space (linear transformation) or LAB color space (nonlinear transformation), both of which separate the luminance and chrominance. However, the luminance resulted from both transformations is still contaminated by the noise from the blue channel, and if a substantial denoising process is performed on the luminance channel, the quality of the denoised image can be adversely affected. Chan *et al.* proposed a modified YCrCb (m-YCrCb) color space [23], which is more effective since the luminance channel does not contain any information from the blue channel. More precisely, the m-YCrCb color space is obtained from the RGB color

Image		Cameraman		Baboon		Boat		Bridge		Barbara512	
Gaussian noise (σ)		0	12	0	12	0	12	0	12	0	12
$I_{max} = 120$	r=10%	28.13	26.08	24.87	23.18	27.34	25.14	26.08	24.02	26.75	24.59
	r=20%	26.31	24.96	23.82	22.66	25.92	24.44	24.93	23.41	25.20	23.87
	r=40%	23.38	22.87	22.29	21.71	23.39	22.72	22.64	21.81	23.36	22.66
Gaussian noise (σ)		0	6	0	6	0	6	0	6	0	6
$I_{max} = 60$	r=10%	27.32	25.78	23.86	22.90	26.16	24.82	24.93	23.63	25.48	24.14
	r=20%	25.97	24.63	23.20	22.39	25.10	24.13	24.23	23.11	24.48	23.58
	r=40%	23.06	22.48	21.97	21.49	22.87	22.49	22.15	21.52	22.90	22.42
Gaussian noise (σ)		0	3	0	3	0	3	0	3	0	3
$I_{max} = 30$	r=10%	25.83	24.97	22.95	22.39	24.83	24.19	23.89	23.16	24.21	23.60
	r=20%	24.97	24.15	22.50	22.02	24.09	23.54	23.20	22.55	23.69	23.20
	r=40%	22.38	22.15	21.50	21.13	22.35	21.89	21.29	21.04	22.17	22.10
Gaussian noise (σ)		0	1	0	1	0	1	0	1	0	1
$I_{max} = 10$	r=10%	23.69	23.43	21.73	21.46	22.92	22.59	22.19	21.98	22.84	22.64
	r=20%	23.09	22.77	21.44	21.27	22.48	22.23	21.79	21.46	22.51	22.34
Gaussian noise (σ)		0	0.5	0	0.5	0	0.5	0	0.5	0	0.5
$I_{max} = 5$	r=10%	22.39	22.09	21.12	20.92	21.83	21.60	21.08	21.01	22.15	21.92
	r=20%	21.90	21.56	20.81	20.58	21.32	21.26	20.76	20.66	21.78	21.68
Gaussian noise (σ)		0	0.1	0	0.1	0	0.1	0	0.1	0	0.1
$I_{max} = 1$	r=10%	19.55	19.51	19.61	19.60	19.22	19.44	18.63	18.76	19.99	19.95
	r=20%	19.03	18.82	19.53	19.51	19.00	19.05	18.45	18.37	19.69	19.67

Table 6.3: Denoising results (PSNR) for various testing images, in the presence of random-valued impulse noise, Gaussian noise and mixed Poisson-Gaussian noise.

space via following linear transformation:

$$\begin{aligned} Y_m &= 0.666G + 0.334R; \\ Cr_m &= 0.666(R - G)/1.6; \\ Cb_m &= (B - 0.666G - 0.334R)/2. \end{aligned} \tag{6.8}$$

In [23], the authors use the multiscale total variational (MTV) method for denoising. Here, we adopted their m-YCrCb transformation (6.8) and apply our model (6.6) to each of the transformed channel, as numerical simulation showed that its performance is superior to a direct application of our method to each of the RGB channel. We present both of the denoised result in [23] and ours in Figure 6.3.

One can observe that while there is still visible noise remains in the denoised image using MTV (6.3b), our model provides a clearer result (6.3c) with equally sharpe edges and details. The improvement in blue channel is not that significant, but one can still notice the difference in the shadow part.

Besides the example used in [23], we also present some other examples in Figure 6.4 to 6.6. The image "M83" is a testing image for real image denoising in Matlab; the image "books" was taken from a students' office under fluorescent lighting and the image "toys" was taken from a room without artificial lighting. All results show that our model is effective in removing unknown type of noise in real color images. The parameters used for all the examples given here were $\lambda_1 = 1$, $\lambda_2 = 0.1$ and $\rho = 0.08, 0.1, 1$ for the three channels respectively, except for the image "toys", where $\rho = 3$ for the luminance channel, since it is of full size and is much noisier than the other examples.

6.2.3 Image Deblurring with Mixed Noises

This subsection is devoted to the harder problem of image deblurring in the presence of mixed noise, where H in (6.1) is a convolution operator.

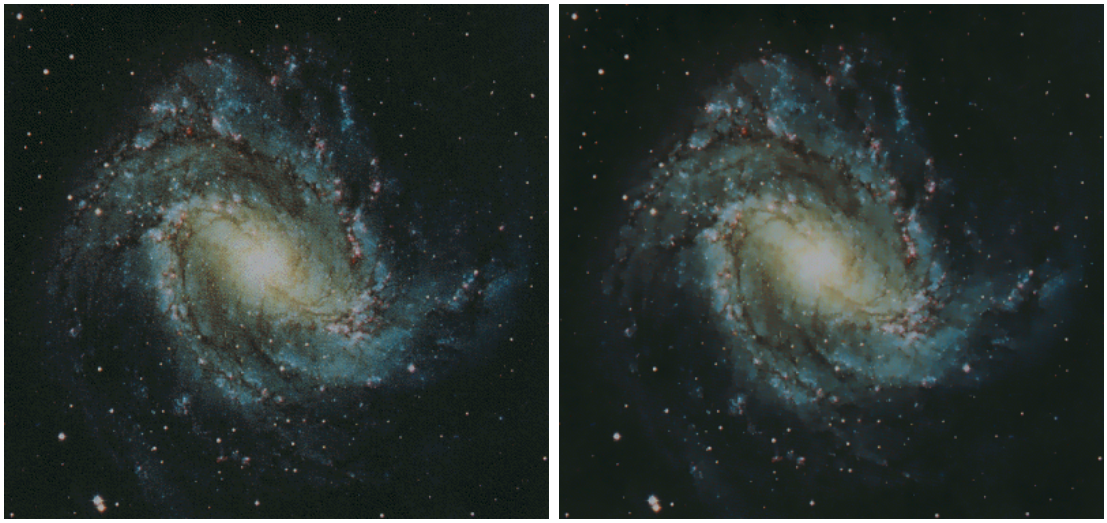
In this experiment, the image's maximum intensity is not rescaled and the blurring kernel is the "disk, 3" kernel generated by the MATLAB command "fspecial".

Table 6.4 summarizes the PSNR results of the deblurring problems in the presence of additive Gaussian noise, Poisson noise, and random-valued impulse noise. Some of the restored images are presented in Figure 6.7. The results again confirm

(a) Original noisy image (b) Denoised result using MTV [23] (c) Denoised result using proposed model

(d) (Blue channel of the original image (e) Denoised blue channel using MTV [23](f) Denoised blue channel using proposed model

Figure 6.3: Real image denoising 1 (image size: 256×256)



(a) Original noisy "M83" (b) Denoised "M83" using the proposed model

Figure 6.4: Real image denoising 2 (image size: 400×378)

(a) Original noisy "books" (b) Denoised "books" using the proposed model

Figure 6.5: Real image denoising 3 (image size: 624×624)

Image	Baboon		Goldhill		Cameraman	
Random-valued impulse noise (r)	10%	20%	10%	20%	10%	20%
Poisson+Gaussian	22.34	22.22	25.30	25.00	24.10	23.82
Gaussian+Poisson	22.33	22.22	25.24	24.95	24.12	23.85

Table 6.4: Deblurring results (PSNR) for various testing images with blurring kernel "disk, 3", in the presence of random-valued impulse noise, Gaussian noise with standard deviation $\sigma = 10$ and Poisson noise at image peak intensity of 255.

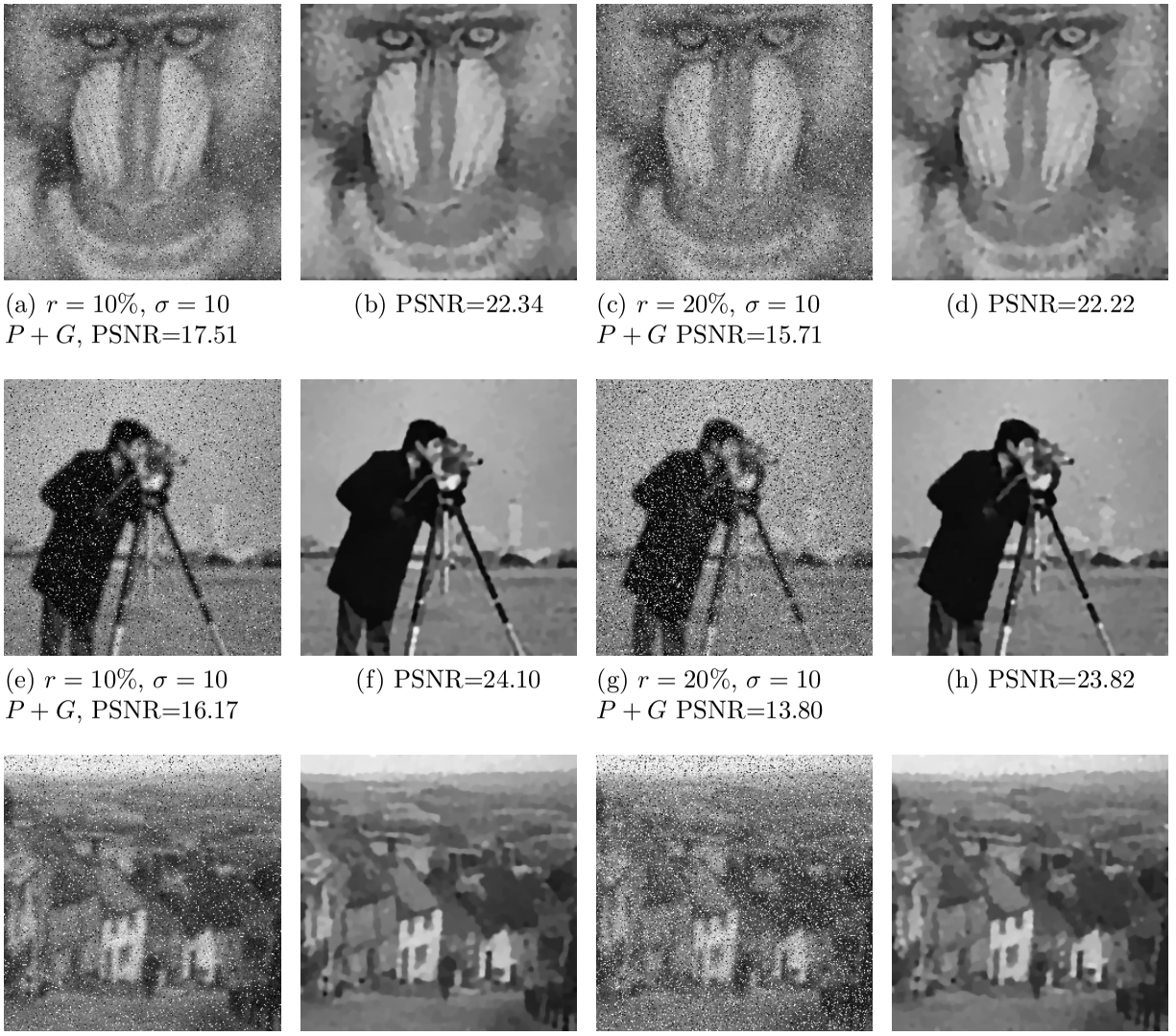
the effectiveness and robustness of our method in dealing with mixed noises.

For all cases, the parameters were set to $\lambda_1 = 1,\ \lambda_2 = 0.1,\ \rho = 0.2$.

6.2.4 Stopping Criteria

In our numerical experiments, the APG algorithm for the inner subproblem is stopped when either the relative norm of the gradient or the relative difference of u is smaller than 10^{-5} and 10^{-6} respectively, and for efficiency purpose, the number of iterations is capped at 50. For the outer ALM iterations, because of multiple degradation factors, there is no generic stopping criterion based on image residue (such as terminating the algorithm when the residue is about the noise level for the pure Gaussian noise case) which can be adopted here. On the other hand, it is far too expensive to terminate the ALM based on the convergence criterion of the dual variable in Step 3 of the ALM, which is in fact not necessary either, since for image restoration, one seeks a moderately accurate sparse solution to (6.6) rather than an optimal solution. Therefore, we simply pre-set the number of outer ALM iterations based on the degradation level of the observed image b. In the cases when the image is very badly damaged by multiple factors, e.g. Poisson noise at low image peak intensity plus blurry effect etc., we terminate the algorithm in 2 to 3 outer iterations; whereas in the cases when only a small percentage of the image pixels are damaged, e.g. with only low level of random-valued impulse noise etc., more iterations help to regain the missing information and we set it to 20 or more. Aside from the two extremes, most of the examples given in the paper are the results obtained after 5 to 7 outer iterations. As suggested in [16], a post-processing procedure by passing through a bilateral filter can remove artifacts effectively. To further improve the results produced by the algorithm, we build in this post-processing procedure to reduce the possible artifacts.

While the non-smoothness of the model usually makes it difficult to design an



(a) $r = 10\%$, $\sigma = 10$ $P + G$, PSNR=17.51 (b) PSNR=22.34 (c) $r = 20\%$, $\sigma = 10$ $P + G$ PSNR=15.71 (d) PSNR=22.22

(e) $r = 10\%$, $\sigma = 10$ $P + G$, PSNR=16.17 (f) PSNR=24.10 (g) $r = 20\%$, $\sigma = 10$ $P + G$ PSNR=13.80 (h) PSNR=23.82

(i) $r = 10\%$, $\sigma = 10$ $P + G$, PSNR=16.36 (j) PSNR=25.30 (k) $r = 20\%$, $\sigma = 10$ $P + G$, PSNR=13.95 (l) PSNR=25.00

Figure 6.7: Deblurring results for various testing images with blurring kernel "disk, 3", in the presence of random-valued impulse noise, Poisson noise and Gaussian noise.

efficient solver, we mention that on average, it takes about 200 inner iterations in total for our proposed algorithm to converge to a fairly good restored image, which is about 20 to 30 seconds (including the bilateral filtering) for an image of size 256×256 on a MacBook Pro with Intel Core2 Duo 2.26GHz CPU and 4GB RAM. In our numerical simulations, all algorithms are implemented in MATLAB (Version 7.14).

6.3 Comparison with Other Models on Specified Noises

As it has been shown in last section, our proposed model is effective and robust in image restoration with a wide range of mixed noise and unknown noise. We may not expect an image restoration model/algorithm to perform comparably well with those models/algorithms that are designed for a particular noise type. However, as we shall show in this section, for many specified noises, such as additive Gaussian noise, Poisson noise, mixed Poisson-Gaussian noise, multiplicative Gamma noise etc, the results produced by our method are comparable to, and sometimes, even better than those specialized algorithms. Nevertheless, the purpose of this section is to show that the model (6.6) together with the algorithm given here is versatile in its ability to handle various types of single noise and mixed noise. This robustness is particularly important because in the situations where the noise type is not known a priori, our method has a good chance to produce a reasonable result, while those methods tailored to a specific noise may no longer be reliable.

Note that we do not make comparisons with some of the well known noise removal cases, e.g. Gaussian noise removal, because they are well studied cases and the standard quality of the restored results in terms of PSNR is commonly known. In the case when we compare with other methods, we state the simulation results from the original papers, and the results from our algorithm are generated from images with the same noise level.

Gaussian noise (σ)	10	20	30	40
Lena	33.70	30.01	27.85	26.76
Cameraman	33.72	29.91	27.64	26.29
Mixed Gaussian noise (σ_m)	$[2, 5, 10]$	$[4, 10, 20]$	$[6, 15, 30]$	$[8, 20, 40]$
Lena	32.91	29.23	27.37	26.20
Cameraman	32.76	29.29	27.21	25.91

Table 6.5: Denoising results (PSNR) for the images "Lena" and "Cameraman", in the presence of single Gaussian noise and a mixture of Gaussian noises with different standard deviations.

6.3.1 Denoising

Gaussian noise and mixed Gaussian noises

First, we consider the additive mixed Gaussian noises with standard deviation $\sigma_m = [2, 5, 10]$, $[4, 10, 20]$, $[6, 15, 30]$ and $[8, 20, 40]$. For comparison purpose, we also give denoising results for single Gaussian noise with standard deviations $\sigma = 10$, 20, 30 and 40. The results in terms of PSNR values for the images "Lena" and "Cameraman" are listed in Table 6.5. For mixed Gaussian noises, although there are no available results for comparison, one can observe that the PSNR values are only slightly lower than that of the single Gaussian noise cases with $\sigma = \max\{\sigma_m\}$. Note that the denoising results for single Gaussian noise cases are obtained by using the same set of parameters for the mixed cases, and are comparable to the results obtained from existing regularization models.

The parameters used in this experiment were $\lambda_1 = 1$, $\lambda_2 = 0.1 \sim 0.5$ and $\rho = 0.4 \sim 0.7$. Based on the experience, a smaller λ_2 and a larger ρ should be chosen as the noise level increases.

Poisson noise and mixed Poisson-Gaussian noise

In the second experiment, we consider Poisson noise at the image peak intensity ($\mathrm{I_{max}}$) ranging from 120 to 1, as well as their mixture with a Gaussian noise with standard deviation $\sigma = \mathrm{I_{max}}/10$. The results are presented in Figure 6.8 and 6.9, and a comparison of PSNR values with other methods is summarized in Table 6.6, where we compared our results with two different approaches: the first is the Anscombe variance-stabilizing transform (VST) [1] followed by a white Gaussian noise denoiser [79] and the second is the PURE-LET model proposed in [66], whose fidelity term

and minimization algorithm were specially designed for denoising mixed Poisson-Gaussian noise based on the statistical analysis of the Poisson and Gaussian random variables. All PSNR results are directly extracted from [66]. The VST used in our experiment is the generalized Anscombe transform [72] defined as follows

$$u_{VST} = 2\sqrt{u + 3/8 + \sigma^2}. \tag{6.9}$$

Note that we provide both results for our model with and without the preprocessing step using VST only for comparison purpose.

The results listed in Table 6.6 show that when Poisson noise is at moderate levels, i.e. $\mathrm{I_{max}} = 120, 60, 30$, even without a VST preprocessing step, our method outperforms the VST plus a state-of-the-art multiresolusion-based Gaussian noise reduction algorithm, which consists of a multivariate estimator resulting from a Bayesian least-squares (BLS) optimization, assuming Gaussian scale mixtures (GSM) as a prior for neighborhoods of coefficients at adjacent positions and scales. For the PURE-LET approach, it optimizes a linear expansion of threshold (LET) by relying on a purely data-adaptive unbiased estimate of the mean-squared error, derived from a Poisson-Gaussian unbiased risk estimate (PURE) [66]. We listed two cases here. For the first case, the LET spans on the transformed domain with Haar undecimated wavelet transform (UWT) only, and for the second case, it spans on both UWT and block discrete cosine transform (BDCT). In the case where the image is corrupted by Poisson noise only, UWT PURE-LET produced slightly better results than our method without VST, while with the VST, our results are even comparable to those of UWT/BDCT PURE-LET. We note that our method is based on piecewise linear B-spline wavelet tight frame and it can easily be extended to a model with two systems including a local DCT, which generally can produce better results, especially for images with rich textures, as demonstrated in [32, 92]. In the case where both Poisson and Gaussian noises are present, the VST preprocessing step becomes redundant for our method, and our results are slightly better than those of UWT/BDCT PURE-LET. This shows that our model is especially effective in removing mixed noises.

In the cases when Poisson noise is at extremely high levels, the variance reduction procedure (6.9) becomes ineffective, and in fact, could be inappropriate. This is because, as studied in [72], the variance of the stabilized Poisson data is approximately equal to 1 irrespective of the mean value of the original data, and for Poisson

Peak intensity (I_{max})		120		60		30	
Gaussian noise (σ)		0	12	0	6	0	3
VST+BLS-GSM		30.85	27.56	29.13	27.02	27.54	26.19
UWT PURE-LET		31.03	27.68	29.29	27.14	27.67	26.32
UWT/BDCT PURE-LET		31.35	27.92	29.58	27.37	27.93	26.53
Model (6.6)	without VST	31.01	28.06	29.23	27.61	27.44	26.58
	with VST	31.42	27.99	29.72	27.54	27.56	26.58
Peak intensity (I_{max})		10		5		1	
Gaussian noise (σ)		0	1	0	0.5	0	0.1
VST+BLS-GSM		24.63	24.43	22.50	22.58	14.44	14.63
UWT PURE-LET		25.10	24.56	23.50	23.22	20.44	20.42
UWT/BDCT PURE-LET		25.29	24.74	23.65	23.36	20.48	20.44
Model (6.6)	without VST	25.02	24.60	23.44	23.11	20.09	19.98
	with VST	23.53	24.01	21.64	21.30	8.71	12.07

Table 6.6: Denoising results (PSNR) for the image "Cameraman", in the presence of single Poisson noise and mixed Poisson-Gaussian noise.

parameter values under 10, the VST looses control over the bias; therefore, it tends to underestimate the pixel values when I_{max} is less than 10. However, as listed in Table 6.6, the results of our method without using VST are still comparable to those of UWT PURE-LET.

The parameters used in the experiments were $\lambda_1 = 1$, $\lambda_2 = 0.1$ and $\rho = 0.2 \sim 1.2$. The parameter ρ increases as I_{max} decreases.

Random-valued impulse noise and Gaussian noise

The third example we show is to remove outliers mixed with Gaussian noise. This problem has been considered in [32], where the authors proposed a model that separates the outliers while estimating u

$$\min_{u,v} \frac{1}{2}\|Hu + v - b\|_2^2 + \lambda_1\|Wu\|_1 + \lambda_2\|v\|_1, \tag{6.10}$$

where H is the identity operator for the denoising case here. In this model, a new variable v is introduced to explicitly represent the outliers in b and the ℓ_1-regularization on v is based on the assumption that the outliers are sparse. The experiments in [32] show that the model (6.10) outperforms the available two-phase approaches (including two-phase methods with pre-detection using the adaptive center-weighted median filter (ACWMF) [25] or ROLD detection [34]), as well as the

(a) $I_{max} = 120$, $\sigma = 0$
PSNR=24.11

(b) Denoised with VST
PSNR=31.42

(c) Denoised without VST
PSNR=31.01

(d) $I_{max} = 60$, $\sigma = 0$
PSNR=21.10

(e) Denoised with VST
PSNR=29.72

(f) Denoised without VST
PSNR=29.23

(g) $I_{max} = 30$, $\sigma = 0$
PSNR=18.05

(h) Denoised with VST
PSNR=27.56

(i) Denoised without VST
PSNR=27.44

Figure 6.8: Denoising results for the image "Cameraman", in the presence of Poisson only.

(a) $I_{max} = 120$, $\sigma = 12$
PSNR=18.56

(b) Denoised with VST
PSNR=27.99

(c) Denoised without VST
PSNR=28.06

(d) $I_{max} = 60$, $\sigma = 6$
PSNR=17.53

(e) Denoised with VST
PSNR=27.54

(f) Denoised without VST
PSNR=27.61

(g) $I_{max} = 30$, $\sigma = 3$
PSNR=15.92

(h) Denoised with VST
PSNR=26.58

(i) Denoised without VST
PSNR=26.58

Figure 6.9: Denoising results for the image "Cameraman", in the presence of both Poisson and Gaussian noises.

model with only the ℓ_1-norm based distance (6.4) as the data fitting term solved using a split Bregman method. Since it is shown in [32] that (6.10) compares favorably against all the above mentioned models, here we only take the results from (6.10) for comparison, and all comparing PSNR values are directly extracted from [32].

Table 6.7 and Table 6.8 summarize the results in terms of PSNR values, from which we can see that our method significantly outperforms the outliers model (6.10), e.g. a gain of 1 to 2dB in the PSNR values for the images "Baboon" and "Bridge". Some of the restored images are presented in Figure 6.10. All parameters used in this experiment were the same as in Section 6.2.1, based on the level of random-valued impulse noise r.

Multiplicative Gamma noise and mixed Gamma-Gaussian noise

Finally, we test our model on the image "House" with multiplicative Gamma noise and its mixture with a Gaussian noise. For the multiplicative Gamma noise, Aubert and Aujol [2] introduced a non-convex data fitting term based on the MAP likelihood estimation approach, i.e.

$$R_1(u) = \sum_{i=1}^{n} \left(\log u_i + \frac{b_i}{u_i} \right). \tag{6.11}$$

Subsequently, several models with an equivalent data fitting term are proposed in order to overcome the numerical difficulties arisen from the non-convexity of (6.11), such as the exponential model [93], the I-divegence model [97] and the m-V model [108].

We listed the results in Table 6.9, where the comparing models are essentially all based on the MAP model (6.11) with a TV regularization: the exponential model [93] uses a logarithm transformation to overcome the non-convexity of (6.11); the m-V model [108] uses an m-th root transformation to achieve the same purpose; and the I-divergence model [97], whose solution is shown to be theoretically equivalent to that of the exponential model. All PSNR values are extracted from [108].

In this experiment, the noisy images were generated by multiplying a Gamma random variable with shape parameter L and scale parameter $1/L$ to the original noise free image and then, adding a Gaussian noise with standard deviation σ.

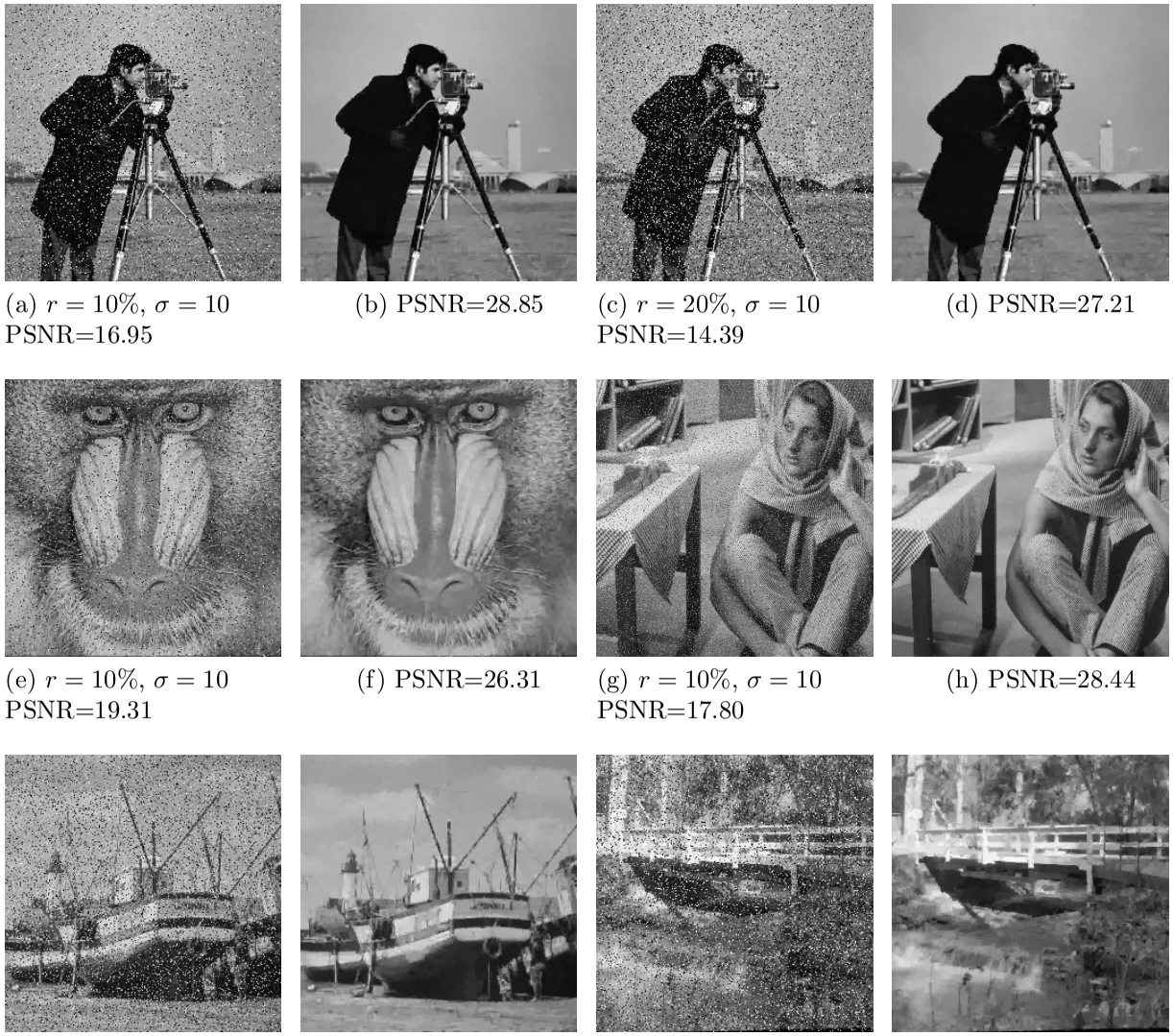
Table 6.9 shows that the results from our model are noticeably better than the others in terms of PSNR and the performance of our model is stable even with an

Random-valued impulse noise (r)	10%		20%		40%	
Gaussian noise (σ)	0	10	0	10	0	10
Model outlier [32]	30.3	28.4	27.4	26.6	23.6	23.3
Model (6.6)	30.83	28.85	27.57	27.21	24.10	24.17

Table 6.7: Denoising results (PSNR) for the image "Cameraman", in the presence of random-valued impulse noise and Gaussian noise.

Image	Baboon		Boat		Bridge		Barbara512	
Random-valued impulse noise (r)	10%	20%	10%	20%	10%	20%	10%	20%
Model outlier [32]	25.1	23.5	28.3	26.4	25.4	23.7	27.9	26.0
Model (6.6)	26.31	24.69	28.67	26.83	27.36	25.83	28.44	26.35

Table 6.8: Denoising results (PSNR) for various testing images, in the presence of random-valued impulse noise and Gaussian noise with standard deviation $\sigma = 10$.



(a) $r = 10\%$, $\sigma = 10$ PSNR=16.95 (b) PSNR=28.85 (c) $r = 20\%$, $\sigma = 10$ PSNR=14.39 (d) PSNR=27.21

(e) $r = 10\%$, $\sigma = 10$ PSNR=19.31 (f) PSNR=26.31 (g) $r = 10\%$, $\sigma = 10$ PSNR=17.80 (h) PSNR=28.44

(i) $r = 20\%$, $\sigma = 10$ PSNR=15.52 (j) PSNR=26.83 (k) $r = 20\%$, $\sigma = 10$ PSNR=14.81 (l) PSNR=25.83

Figure 6.10: Denoising results for various testing images, in the presence of random-valued impulse noise and Gaussian noise.

Speckle	$L = 1$			$L = 3$		
Gaussian noise (σ)	0	10	20	0	10	20
I-div model	22.5			24.7		
exp model	22.4			24.6		
m-V model	22.6			25.1		
Model (6.6)	22.99	22.88	22.95	25.77	25.69	25.55

Table 6.9: Denoising results (PSNR) for the image "House", in the presence of speckle and a mixture of speckle and Gaussian noise.

additive Gaussian noise.

The parameters used were $\lambda_1 = 1,\ \lambda_2 = 0.01$, and $\rho = 0.7, 0.5$ for $L = 1, 3$ respectively.

6.3.2 Deblurring

Poisson noise and mixed Poisson-Gaussian noise

First, we consider the deblurring problems in the presence of Poisson noise and its mixture with Gaussian noise. We use the following 4 blurring kernels generated by the MATLAB command "fspecial": "disk, 3" kernel; "15×30 motion" kernel; "15×15 Gaussian" kernel with standard deviation 2; and "9×9 average" kernel. We vary the image's maximum intensity from 120 to 30 and the Gaussian noise level correspondingly as in Section 6.3.1. Note that we did not consider the cases for $\mathrm{I}_{\max}$ below 30 because blurred images are less likely to be extremely noisy.

Table 6.10 summarizes the PSNR values we obtained for the deblurring problems with Poisson and Gaussian noises. As far as we are aware of, no such thorough testing have been done in the literature. Note that deblurring in the presence of Poisson noise has been considered in [89]; however, the maximum intensity are scaled up to thousands, thus, the noisiness is much weaker than the cases given here.

The parameters used in this experiment were $\lambda_1 = 1,\ \lambda_2 = 0.01,\ \rho = 0.1$ for $\mathrm{I}_{\max} = 120, \sigma = 0$, and $\rho = 0.15$ for all the other cases.

Random-valued impulse noise and Gaussian noise

In the second experiment, we consider the problem of deblurring the images contaminated by both impulse noise and Gaussian noise. Note that two-phase methods

Peak intensity (I_{max})		120		60		30	
Gaussian noise (σ)		0	12	0	6	0	3
Cameraman	disk, 3	24.29	23.36	23.91	23.24	23.43	22.84
Cameraman	motion, 15, 30	22.89	21.74	22.28	21.56	21.69	21.27
Cameraman	Gaussian, 15, 2	23.77	22.87	23.30	22.79	22.86	22.51
Cameraman	average, 9	22.76	21.89	22.30	21.95	21.89	21.62
Lena	disk, 3	26.31	24.89	25.66	24.81	24.93	24.31
Lena	motion, 15, 30	24.24	22.99	23.57	22.72	22.85	22.39
Lena	Gaussian, 15, 2	25.74	24.51	25.24	24.24	24.47	23.94
Lena	average, 9	24.65	23.44	23.97	23.27	23.55	23.10

Table 6.10: Deblurring results (PSNR) for the images "Cameraman" and "Lena", in the presence of Poisson noise and mixed Poisson-Gaussian noise.

Image	Baboon		Goldhill		Cameraman	
Random-valued impulse noise (r)	10%	20%	10%	20%	10%	20%
Model outlier [32]	21.2	21.1	25.7	21.4	24.2	24.0
Model (6.6)	22.62	22.49	25.65	25.48	24.48	24.24

Table 6.11: Deblurring results (PSNR) for various testing images with blurring kernel "disk, 3", in the presence of random-valued impulse noise, Gaussian noise with standard deviation $\sigma = 10$.

with pre-detection of the impulse noise using median filters have also been used to solve this problem [10, 11]. However, since it is shown in [32] that the outlier model (6.10) outperforms the two-phase methods, we again compare our results with that of the model (6.10) only.

The blurring kernel used in this experiment is the "disk, 3" kernel, and the image's maximum intensity is not rescaled.

The results in terms of PSNR are summarized in Table 6.11, and some of the restored images are presented in Figure 6.11. We can see that our method generally produces better results, especially for the image "Baboon", there is a gain of more than 1dB in terms of PSNR.

The setting of parameters was the same as in Section 6.2.3.

(a) $r = 10\%$, $\sigma = 10$ PSNR=18.28 (b) $r = 10\%$, $\sigma = 10$ PSNR=16.66 (c) $r = 10\%$, $\sigma = 10$ PSNR=17.67

(d) PSNR=22.62 (e) PSNR=24.58 (f) PSNR=25.65

Figure 6.11: Deblurring results for various testing images with blurring kernel "disk, 3", in the presence of random-valued impulse noise and Gaussian noise. The first row are blurred noisy images and the second row are the corresponding restored ones.

	Without mask			With mask		
Percentage of pixels missing	10%	20%	30%	50%	70%	90%
Lena	31.81	27.87	24.59	31.33	28.04	23.46
Peppers	33.77	30.04	25.85	30.80	26.20	22.43
Cameraman	29.34	25.56	22.11	28.60	25.34	21.55

Table 6.12: Recovery results (PSNR) for several testing images with different percentage of missing pixels.

6.3.3 Recovery from Images with Randomly Missing Pixels

From the results presented in previous sections, we have already seen that our method is robust to outliers. To further demonstrate this, we present in this section the recovery results from images with randomly missing pixels. We consider both cases when the positions of the missing pixels are known and unknown; in other words, one can refer them as the inpainting and the blind inpainting problems respectively.

Table 6.12 shows that without knowing which pixels are missing, our method is able to recover the original image to a moderate extent with at most 30% missing pixels; however, knowing their exact positions allows us to increase this number to 90%. The visual results for the image "Peppers" are presented in Figure 6.12.

The parameters used in this experiment were $\lambda_1 = 1,\ \lambda_2 = 0.01,\ \rho = 1.2, 1.5, 1.8$ for the recovery of missing $10\%,\ 20\%,\ 30\%$ pixels without mask, and $\lambda_1 = 1,\ \lambda_2 = 0.1,\ \rho = 0.05, 0.1, 0.15$ for the recovery of missing $50\%,\ 70\%,\ 90\%$ pixels with mask.

6.4 Further Remarks

In this section, we propose an even simpler model with only the ℓ_1-norm based distance function as the data fitting term. The reduced model works perfectly well as (6.6) mainly due to the ALM and the special feature of image restoration problems that one only seeks a good sparse approximated solution to the original model. In addition, we present comparisons of our proposed ALM-APG algorithm with the widely used alternating direction method of multipliers (ADMM) algorithm.

(a) 10% pixels missing PSNR=15.59

(b) 20% pixels missing PSNR=12.60

(c) 50% pixels missing PSNR=8.62

(d) 70% pixels missing PSNR=7.10

(e) Recovered without mask PSNR=33.77

(f) Recovered without mask PSNR=30.04

(g) Recovered with mask PSNR=30.80

(h) Recovered with mask PSNR=26.20

Figure 6.12: Recovery results (PSNR) for the image "Peppers" with different percentage of missing pixels. The first row are the images with missing pixels, and the second row are the corresponding recovered ones.

Gaussian noise (σ)	10	20	30	40
Lena	33.67	29.96	27.81	26.75
Cameraman	33.62	29.84	27.61	26.24
Mixed Gaussian noise (σ_m)	$[2, 5, 10]$	$[4, 10, 20]$	$[6, 15, 30]$	$[8, 20, 40]$
Lena	32.88	29.24	27.35	26.21
Cameraman	32.67	29.28	27.20	25.91

Table 6.13: Denoising results (PSNR) for the images "Lena" and "Cameraman", in the presence of single Gaussian noise and a mixture of Gaussian noises with different standard deviations using model (6.12).

Peak intensity (I_{max})		120		60		30	
Gaussian noise (σ)		0	12	0	6	0	3
Model (6.12)	without VST	30.99	28.10	29.19	27.60	27.44	26.53
	with VST	31.40	27.97	29.69	27.52	27.77	26.49
Peak intensity (I_{max})		10		5		1	
Gaussian noise (σ)		0	1	0	0.5	0	0.1
Model (6.12)	without VST	25.02	24.56	23.35	22.99	19.95	19.94
	with VST	23.58	24.12	22.13	21.57	9.34	13.79

Table 6.14: Denoising results (PSNR) for the image "Cameraman", in the presence of single Poisson noise and mixed Poisson-Gaussian noise using model (6.12).

6.4.1 Reduced Model

From the numerical experiments presented the previous two sections, the parameter λ_2 is much smaller than $\lambda_1 = 1$ in most of the cases, especially when impulse noise is involved. In fact, for the cases with only Gaussian noise, where the ℓ_2-norm distance function (6.3) is the best choice based on statistical analysis among all the possible data fitting terms, reducing λ_2 to 0 in the experiments in subsection 6.3.1 does not vary the results too much. Table 6.13 summarizes the PSNR results for the same denoising problems as presented in Table 6.5. Note that except λ_2, all the other parameters used were the same for the two experiments.

Moreover, for denoising of mixed Poisson-Gaussian noise in subsection 6.3.1, where λ_2 is also chosen to be relatively larger than the cases with impulse noise, the results with $\lambda_2 = 0$, as listed in Table 6.14, are still very close to those presented in Table 6.6.

The above observations suggest that we can reduce the model (6.6) to the following even simpler version:

$$\min_{u\in\mathbb{R}^n} \ \lambda_1\|Hu-b\|_1+\rho\|Wu\|_1. \tag{6.12}$$

This reduction will not reduce the effectiveness of the model (6.6) because the ℓ_2-norm distance term is implicitly built in by the proposed ALM. This is readily shown via the augmented Lagrangian functions of the corresponding two models.

If we write $y=\begin{pmatrix}y_1\\y_2\end{pmatrix}$ and $z=\begin{pmatrix}z_1\\z_2\end{pmatrix}$, then the augmented Lagrangian function of (6.6) associated with $\sigma>0$ can be written as

$$\begin{aligned}\mathcal{L}_{2,\sigma}(u,z_1,z_2;y_1,y_2)=&\frac{\lambda_2}{2}\|Hu-b\|^2+\lambda_1\|z_1\|_1+\rho\|z_2\|_1+\langle y_1,b-Hu-z_1\rangle\\&+\frac{\sigma}{2}\|b-Hu-z_1\|^2+\langle y_2,-Wu-z_2\rangle+\frac{\sigma}{2}\|Wu+z_2\|^2\\=&\frac{\lambda_2+\sigma}{2}\|Hu-b\|^2+\lambda_1\|z_1\|_1+\rho\|z_2\|_1+\langle y_1,b-Hu-z_1\rangle\\&+\frac{\sigma}{2}\|z_1\|^2+\sigma\langle Hu-b,z_1\rangle+\langle y_2,-Wu-z_2\rangle+\frac{\sigma}{2}\|Wu+z_2\|^2.\end{aligned}$$

The augmented Lagrangian function of (6.12) can be obtained simply by setting $\lambda_2=0$ in the above expression:

$$\begin{aligned}\mathcal{L}_{1,\sigma}(u,z_1,z_2;y_1,y_2)=&\frac{\sigma}{2}\|Hu-b\|^2+\lambda_1\|z_1\|_1+\rho\|z_2\|_1+\langle y_1,b-Hu-z_1\rangle\\&+\frac{\sigma}{2}\|z_1\|^2+\sigma\langle Hu-b,z_1\rangle+\langle y_2,-Wu-z_2\rangle+\frac{\sigma}{2}\|Wu+z_2\|^2.\end{aligned}$$

Obviously, $\mathcal{L}_{1,\sigma}$ also contains the ℓ_2-norm distance term $\|Hu-b\|^2$. Although there is a difference of $\lambda_2/2$ in the weight, we note that the relative difference is insignificant in our numerical experiments as σ is usually $10\sim 50$ times larger than λ_2. As a result, the necessity of having the ℓ_2 fitting term in the model (6.6) is often obscured by the augmented Lagrangian based method we use to solve the reduced model (6.12). (We give a more detailed explanation in the next remark.)

If instead of using an ALM, one uses a subgradient method to solve (6.12), then the importance of the ℓ_2 fitting term would become clearer since the subgradient method does not involve an implicit ℓ_2 fitting term. To verify this claim, we used a subgradient method to denoised the 256×256 "Cameraman" image contaminated by

a mixed Poisson noise (with $I_{\max} = 255$) and Gaussian noise (with $\sigma = 10$) based on the reduced model (6.12) with $\lambda_1 = 1, \rho = 1$. The best PSNR value obtained by the subgradient method (running for 200 iterations) is 28.80, whereas the corresponding PSNR value obtained by ALM-APG when solving (6.12) is 30.46.

We should mention that by running more subgradient (or ALM-APG) iterations can be counter-productive in terms of reducing the PSNR value. Though by performing more iterations, we get a better approximate minimizer (for having a lower objective value) for the model (6.12), but it does not necessarily give a better recovered solution for the original ill-posed image restoration problem. One needs to remember that the model (6.12) is only a regularization model for an ill-posed problem, which is only meant to be used as a guide to find a good recovered solution for the ill-posed problem, and there is no guarantee that a minimizer of the model would provide a superior recovered solution. Similar remarks also apply to the model (6.6).

Remark 6.1. Observe that the ALM automatically builds in an ℓ_2 data fitting term in each iteration, thus the solution path generated by the proposed ALM when solving the reduced model (6.12) is, therefore, similar to that generated by the ALM for solving the original model (6.6) when the parameter λ_2 is much smaller than σ. More specifically, because the approximate solutions $(u^{k+1}, z_1^{k+1}, z_2^{k+1})$ and $(\bar{u}^{k+1}, \bar{z}_1^{k+1}, \bar{z}_2^{k+1})$ for the models (6.6) and (6.12) are generated from minimizing $\mathcal{L}_{2,\sigma}(u, z_1, z_2; y_1^k, y_2^k)$ and $\mathcal{L}_{1,\sigma}(u, z_1, z_2; \bar{y}_1^k, \bar{y}_2^k)$ respectively, the small relative difference between $\mathcal{L}_{2,\sigma}$ and $\mathcal{L}_{1,\sigma}$ would imply that both models tend to produce similar results when $(y_1^k, y_2^k) \approx (\bar{y}_1^k, \bar{y}_2^k)$. Thus when the ALMs applied to (6.6) and (6.12) both use the same starting iterate $(y_1^0, y_2^0) = (\bar{y}_1^0, \bar{y}_2^0)$ and the same sequence of parameters $\{\sigma_k \gg \lambda_2\}$, the iterates for the first few outer iterations of the ALMs would have the property that $(u^{k+1}, z_1^{k+1}, z_2^{k+1}) \approx (\bar{u}^{k+1}, \bar{z}_1^{k+1}, \bar{z}_2^{k+1})$, and $(y_1^{k+1}, y_2^{k+1}) \approx (\bar{y}_1^{k+1}, \bar{y}_2^{k+1})$. This does not contradict the fact that different optimal solutions for the two models are expected at convergence since eventually (y_1^k, y_2^k) and $(\bar{y}_1^k, \bar{y}_2^k)$ would become different enough for the ALMs to converge to different optimal solutions. Nor does it contradict the less prominent results of (6.12) presented in [33], because different numerical algorithms generate different solution paths and subsequently, lead to different approximate solutions.

Recall that all minimization problems with ℓ_1-regularization terms in image restorations are regularization models for ill-posed problems. The optimization models are used to serve as a guide to obtain a good sparse solution in a certain

transformed domain, and the solver is usually terminated when a good sparse approximate solution is obtained instead of running to full convergence to an optimal solution. For our ALMs applied to (6.6) and (6.12), because we only need less than two dozens outer iterations to get good feasible solutions, the solutions obtained for both models would be quite similar based on the explanation given in the previous paragraph. Thus it is not surprising that the results obtained in Tables 6.13 and 6.14 are quite close to those in Tables 6.5 and 6.6, respectively.

6.4.2 ALM-APG versus ADMM

As it has been discussed in Chapter 4, to tackle the non-separable ℓ_1-term in (P), most of the existing fast algorithms for image restoration problems are variants of the augmented Lagrangian method (ALM) with the inner subproblem being solved by an alternating direction method of multipliers (ADMM), such as the split Bregman algorithm [17, 51]. The difference between the ALM-APG algorithm used here and the split Bregman algorithm is that by expressing $\boldsymbol{u}$ in terms of $\boldsymbol{x}$ as in (3.14) and substituting it to the objective function, we essentially solve an inner subproblem with only one variable $\boldsymbol{x}$, which therefore, can be solved to a moderate accuracy before updating the Lagrangian dual variable $\boldsymbol{v}$.

Here we also provide some numerical results from the model solved by the ADMM for comparison. The parameters used in the ADMM have been tuned and the total number of the ADMM iterations is capped at 200, because additional iterations do not make significant progress. The denoising results in terms of PSNR are summarized in Table 6.15 and Table 6.16, and in Table 6.15, we also listed the CPU time cost. The time recorded is in seconds and includes the bilateral filtering, which takes around 5s. In the cases with Poisson noise, images' maximum intensity were not rescaled and Gaussian noise was added after Poisson noise. The results show that the ALM-APG algorithm generally performs better than the ADMM. Since in [32], the authors also have considered the reduced model (6.12) solving by a split Bregman algorithm for denoising of mixed Gaussian noise and random-valued impulse noise, we include their results in the table for reference purpose.

Both the ADMM and the ALM-APG algorithm used in this chapter are first-order methods. However, it is well-known that for many problems, the ADMM algorithm may converge to a low accurate solution very fast but slows down drastically in subsequent iterations; whereas, the proposed ALM-APG algorithm does not

suffer from stagnation in all the numerical experiments conducted in this chapter.

Random-valued impulse noise (r)		10%		20%		40%	
Gaussian noise (σ)		0	10	0	10	0	10
without Poisson noise		PSNR (Time)					
Model (6.6)	ADMM	30.30 (32.8)	28.20 (30.6)	27.36 (32.5)	26.77 (30.0)	23.95 (32.2)	23.82 (30.0)
	ALM-APG	30.83 (33.0)	28.85 (20.5)	27.57 (33.4)	27.21 (21.5)	24.10 (16.2)	24.17 (21.3)
Model (6.12)	ADMM	30.30 (32.3)	28.20 (30.2)	27.36 (32.9)	26.77 (30.0)	23.95 (32.0)	23.83 (30.5)
	ALM-APG	30.82 (32.7)	28.89 (20.6)	27.58 (33.8)	27.21 (20.9)	24.10 (16.4)	24.17 (21.2)
	Split Bregman [32]	29.9	27.5	27.1	26.0	23.1	22.9
with Poisson noise		PSNR (Time)					
Model (6.6)	ADMM	28.26 (30.0)	27.08 (29.8)	26.72 (30.0)	25.99 (30.0)	23.27 (33.7)	23.40 (30.5)
	ALM-APG	29.07 (20.1)	28.31 (17.6)	27.10 (20.9)	26.60 (18.8)	23.46 (19.2)	23.68 (18.7)
Model (6.12)	ADMM	28.25 (30.2)	27.09 (30.4)	26.72 (30.1)	26.00 (29.9)	23.28 (29.8)	23.41 (29.8)
	ALM-APG	29.11 (20.4)	28.36 (17.8)	27.10 (20.9)	26.60 (18.2)	23.48 (19.6)	23.69 (18.8)

Table 6.15: Denoising results (PSNR) for the image "Cameraman", in the presence of Gaussian noise, Poisson noise and random-valued impulse noise using different models and numerical algorithms.

Image		Baboon		Boat		Bridge		Barbara512	
Random-valued impulse noise (r)		10%	20%	10%	20%	10%	20%	10%	20%
without Poisson noise									
Model (6.6)	ADMM	25.97	24.68	27.99	26.25	27.11	25.41	27.76	25.93
	ALM-APG	26.31	24.69	28.67	26.83	27.36	25.83	28.44	26.35
Model (6.12)	ADMM	25.97	24.69	27.99	26.26	27.11	25.42	27.76	25.94
	ALM-APG	26.32	24.69	28.68	26.83	27.37	25.83	28.45	26.34
	Split Bregman [32]	24.5	23.2	27.6	26.1	25.0	23.4	27.0	25.5
with Poisson noise									
Model (6.6)	ADMM	24.92	23.75	26.38	24.88	25.85	24.37	26.25	24.78
	ALM-APG	25.26	24.08	27.60	26.17	26.38	25.23	27.20	25.58
Model (6.12)	ADMM	24.92	23.76	26.38	24.90	25.85	24.38	26.26	24.80
	ALM-APG	25.26	24.08	27.60	26.17	26.38	25.24	27.21	25.57

Table 6.16: Denoising results (PSNR) for various testing images, in the presence of Poisson noise, Gaussian noise with standard deviation $\sigma = 10$ and random-valued impulse noise using different models and numerical algorithms..

Bibliography

[1] F. J. Anscombe, *The transformation of Poisson, binomial and negative-binomial data*, Biometrika, 35 (1948), pp. 246–254.

[2] G. Aubert and J.-F. Aujol, *A variational approach to remove multiplicative noise*, SIAM Journal on Applied Mathematics, 68 (2008), pp. 925–946.

[3] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Algorithm*, John Wiley & Sons, Inc., 2006.

[4] A. Beck and M. Teboulle, *Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems*, IEEE Trans. Image Proc., 18 (2009), pp. 2419–2434.

[5] ——, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM Journal on Imaging Sciences, 2 (2009), pp. 183–202.

[6] E. V. D. Berg and M. P. Friedlander, *Probing the pareto frontier for basis pursuit solutions*, SIAM Journal of Scientific Computing, 31 (2008), pp. 890–912.

[7] H. Bondell and B. Reich, *Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with oscar*, Biometrics, 64 (2008), pp. 115–123.

[8] C. D. Boor, R. A. DeVore, and A. Ron, *On the construction of multivariate (pre)wavelets*, Constructive Approximation, 9 (1993), pp. 123–166.

[9] R. H. Byrd, G. M. Chin, J. Nocedal, and F. Oztoprak, *A family of second-order methods for convex ℓ_1-regularized optimization*, tech. rep., Northwestern University, June 2012.

[10] J.-F. Cai, R. Chan, and M. Nikolova, *Two-phase approach for deblurring images corrupted by impulse plus Gaussian noise*, Inverse Problems and Imaging, 2 (2008), pp. 187–204.

[11] J.-F. Cai, R. H. Chan, and M. Nikolova, *Fast two-phase image deblurring under impulse noise*, Journal of Mathematical Imaging and Vision, 36 (2010), pp. 46–53.

[12] J.-F. Cai, R. H. Chan, L. Shen, and Z. Shen, *Restoration of chopped and nodded images by framelets*, SIAM Journal on Scientific Computing, 30 (2008), pp. 1205–1227.

[13] ——, *Convergence analysis of tight framelet approach for missing data recovery*, Advances in Comupational Mathematics, 31 (2009), pp. 87–113.

[14] J.-F. Cai, R. H. Chan, and Z. Shen, *A framelet-based image inpainting algorithm*, Applied and Computational Harmonic Analysis, 24 (2008), pp. 131–149.

[15] J.-F. Cai, B. Dong, S. Osher, and Z. Shen, *Image restoration: total variation, wavelet frames, and beyond*, Journal of the American Mathematical Society, 25 (2012), pp. 1033–1089.

[16] J.-F. Cai, S. Osher, and Z. Shen, *Linearized Bregman iteration for frame based image deblurring*, SIAM Journal on Imaging Sciences, 2 (2009), pp. 226–252.

[17] ——, *Split Bregman methods and frame based image restoration*, Multiscale Modeling and Simulations: A SIAM Interdisciplinary Journal, 8 (2009), pp. 337–369.

[18] J.-F. Cai and Z. Shen, *Framelet based deconvolution*, Journal of Computational Mathematics, 28 (2010), pp. 289–308.

[19] E. Candès, J. Romberg, and T. Tao, *Stable signal recovery from incomplete and inaccurate measurements*, Communications on Pure and Applied Mathematics, 59 (2006), pp. 1207–1223.

[20] E. J. Candès and T. Tao, *Decoding by linear programming*, IEEE Trans. Inform. Theory, 51 (2005), pp. 4203–4215.

[21] R. H. Chan, T. F. Chan, L. Shen, and Z. Shen, *Wavelet algorithms for high-resolution image reconstruction*, SIAM Journal on Scientific Computing, 24 (2003), pp. 1408–1432.

[22] R. H. Chan, S. D. Riemenschneider, L. Shen, and Z. Shen, *Tight frame: an efficient way for high-resolution image reconstruction*, Applied and Computational Harmonic Analysis, 17 (2004), pp. 91–115.

[23] T. Chan, Y. Wang, and H. Zhou, *Denoising natural color photos in digital photography.* Preprint, 2007.

[24] S. Chen, D. L. Donoho, and M. Saunders, *Atomic decomposition by basis pursuit*, SIAM Journal on Scientific Computing, 20 (1998), pp. 33–61.

[25] T. Chen and H. R. Wu, *Adaptive impulse detection using center-weighted median filters*, IEEE Signal Processing Letters, 8 (2001), pp. 1–3.

[26] X. Chen, S. Kim, Q. Lin, J. G. Carbonell, and E. P. Xing, *Graph-structured multi-task regression and an efficient optimization method for general fused lasso*, arXiv preprint arXiv:1005.3579, (2010).

[27] X. Chen, Q. Lin, S. Kim, J. G. Carbonell, and E. P. Xing, *Smoothing proximal gradient method for general structured sparse regression*, Annals of Applied Statistics, 6 (2012), pp. 719–752.

[28] F. H. Clarke, *Optimization and Nonsmooth Analysis*, Wiley, New York, 1983.

[29] I. Daubechies, M. D. Friese, and C. D. Mol, *An iterative thresholding algorithm for linear inverse problems with a sparsity constraint*, Communications on Pure and Applied Mathematics, 57 (2004), pp. 1413–1457.

[30] I. Daubechies, B. Han, A. Ron, and Z. Shen, *Framelets: MRA-based constructions of wavelet frames*, Applied and Computational Harmonic Analysis, 14 (2003), pp. 1–46.

[31] I. Daubechies, G. Teschke, and L. Vese, *Iteratively solving linear inverse problems under general convex constraints*, Inverse Problems and Imaging, 1 (2007), pp. 29–46.

[32] B. Dong, H. Ji, J. Li, Z. Shen, and Y. Xu, *Wavelet frame based blind image inpainting*, Applied and Computational Harmonic Analysis, 32 (2012), pp. 268–279.

[33] B. Dong and Z. Shen, *MRA-based wavelet frames and applications*, IAS/-Park City Mathematics Series, 19 (2010). Summer Program on "The Mathematics of Image Processing".

[34] Y. Dong, R. F. Chan, and S. Xu, *A detection statistic for random-valued impulse noise*, IEEE Transactions on Image Processing, 16 (2007), pp. 1112–1120.

[35] D. Donoho, M. Elad, and V. Temlyakov, *Stable recovery of sparse overcomplete representations in the presence of noise*, IEEE Transactions on Information Theory, 52 (2006), pp. 6–18.

[36] D. L. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck, *Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit*, IEEE Trans. Inform. Theory, 58 (2012), pp. 1094–1121.

[37] J. Eckstein and D. P. Bertsekas, *On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators*, Mathematical Programming, 55 (1992), pp. 293–318.

[38] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, *Least angle regression*, Ann. Statist., 32 (2004), pp. 407–499.

[39] M. Elad, J.-L. Starck, P. Querre, and D. L. Donoho, *Simultaneous cartoon and texture image inpaiting using morphological component analysis (MCA)*, Applied and Computational Harmonic Analysis, 19 (2005), pp. 340–358.

[40] F. Facchinei, *Minimization of SC^1 functions and the Maratos effect*, Operation Research Letters, 17 (1995), pp. 131–137.

[41] M. Fadili and J.-L. Starck, *Sparse representations and bayesian image inpainting*, in SPARS'05, vol. I, Rennes, France, November 2005.

[42] M. Fadili, J.-L. Starck, and F. Murtagh, *Inpainting and zooming using sparse representations*, The Computer Journal, 52 (2009), pp. 64–79.

[43] M. Figueiredo and R. Nowak, *An EM algorithm for wavelet-based image restoration*, IEEE Transactions on Image Processing, (2003), pp. 906–916.

[44] ——, *A bound optimization approach to wavelet-based image deconvolution*, in IEEE ICIP'05, 2005.

[45] M. Figueiredo, R. Nowak, and S. J., *Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems*, IEEE Journal of Selected Topics in Signal Processing, 1 (2007), pp. 586–597.

[46] K. Fountoulakis, J. Gondzio, and P. Zhlobich, *Matrix-free interior point method for compressed sensing problems*, Mathematical Programming Computation, (2013), p. to appear.

[47] J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani, *Pathwise coordinate optimization*, Annals of Applied Statistics, 1 (2007), pp. 302–332.

[48] D. Gabay and B. Mercier, *A dual algorithm for the solution of nonlinear variational problems via finite element approximation*, Computers & Mathematics with Applications, 2 (1976), pp. 17–40.

[49] A. Genkin, D. Lewis, , and D. Madigan, *Large-scale bayesian logistic regression for text categorization*, Technometrics, 49 (2007), pp. 291–304.

[50] R. Glowinski and A. Marrocco, *Sur l'approximation par éléments finis d'ordre un, et la resolution par pénalisation-dualité d'une classe de problèmes de Dirichlet non-linéaires*, Revue Française d'Automatique, Informatique, Recherche Opérationnelle. Analyse Numérique, 9 (1975), pp. 41–76.

[51] T. Goldstein and S. Osher, *The split Bregman algorithm for L1 regularized problems*, SIAM Journal on Imaging Sciences, 2 (2009), pp. 323–343.

[52] L. Grosenick, B. Klingenberg, K. Katovich, B. Knutson, and J. Taylor, *Interpretable whole-brain prediction analysis with graphnet*, Neuroimage, 72 (2013), pp. 304–321.

[53] E. T. Hale, W. Yin, and Y. Zhang, *Fixed-point continuation for ℓ_1-minimization: methodology and convergence*, SIAM J. Optimization, 19 (2008).

[54] M. Hestenes, *Multiplier and gradient methods*, J. Optimization Theory and Applications, 4 (1969), pp. 303–320.

[55] R. Hodrick and E. Prescott, *Postwar u.s. business cycles: an empirical investigation*, Journal of Money, Credit, and Banking, 29 (1997), pp. 1–16.

[56] Y. Huang, M. Ng, and Y. Wen, *A fast total variation minimization method for image restoration*, SIAM J. Multiscale Modeling and Simulation, 7 (2008), pp. 774–795.

[57] H. Ji, C. Liu, Z. Shen, and Y. Xu, *Robust video denoising using low rank matrix completion*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, 2010.

[58] K. Jiang, D. Sun, and K. Toh, *An inexact accelerated proximal gradient method for large scale linearly constrained convex sdp*, SIAM J. Optimization, 22 (2012), pp. 1042–1064.

[59] S.-J. Kim, K. Koh, S. Boyd, and D. Gorinevsky, *ℓ_1 trend filtering*, SIAM Review, 51 (2009), pp. 339–360.

[60] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, *An interior-point method for large scale ℓ_1-regularized least squares*, IEEE Journal of Selected Topics in Signal Processing, 1 (2007), pp. 606–617.

[61] K. Koh, S.-J. Kim, and S. Boyd, *An interior-point method for large-scale ℓ_1-regularized logistic regression*, Journal of Machine Learning Research, 8 (2007), pp. 1519–1555.

[62] T. Le, R. Chartrand, and T. J. Asaki, *A variational approach to reconstructing images corrupted by poisson noise*, Journal of Mathematical Imaging and Vision, 27 (2007), pp. 257–263.

[63] S. Lee, H. Lee, P. Abbeel, and A. Ng, *Efficient ℓ_1-regularized logistic regression*, In National Conference on Artificial Intelligence (AAAI), (2006).

[64] J. Liu, S. Ji, and J. Ye, SLEP*: Sparse Learning with Efficient Projections*, Arizona State University, 2009.

[65] J. Liu, L. Yuan, and J. Ye, *An efficient algorithm for a class of fused lasso problems*, in Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2010, pp. 323–332.

[66] F. Luisier, *Image denoising in mixed Poisson-Gaussian noise*, IEEE Transactions on Image Processing, 20 (2011), pp. 696–707.

[67] S. G. Mallat, *Multiresolution approximations and wavelet orthonormal bases of* $L_2(\mathbb{R})$, Transactions of the American Mathematical Society, 315 (1989), pp. 69–87.

[68] Y. Meyer, *Wavelets and Operators*, vol. 1, Cambridge Studies in Advanced Mathematics, 1992.

[69] R. Mifflin, *Semismooth and semiconvex functions in constrained optimization*, SIAM J. Control and Optimization, 15 (1977), pp. 957–972.

[70] G. J. Minty, *Monotone (nonlinear) operators in Hilbert space*, Duke Math. J., 29 (1962), pp. 341–346.

[71] J. J. Moreau, *Proximité et dualité dans un espace Hilbertien*, Bull. Soc. Math. France, 93 (1965), pp. 273–299.

[72] F. Murtagh, J.-L. Starck, and A. Bijaoui, *Image restoration with noise suppression using a multiresolution support*, Astron. Astrophys. Supplement Series, 112 (1995), pp. 179–189.

[73] Y. Nesterov, *Smooth minimization of non-smooth functions*, Math. Program., 103 (2005), pp. 127–152.

[74] Y. Nesterov, *Gradient methods for minimizing composite objective function*, ECORE Discussion paper, (2007).

[75] M. Nikolova and M. K. Ng, *Analysis of half-quadratic minimization methods for signal and image recovery*, SIAM J. Imaging Sciences, 27 (2005), pp. 937–966.

[76] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer Series in Operations Research and Financial Engineering, Springer, New York, 2nd ed., 2006.

[77] J. S. Pang and L. Qi, *A globally convergent Newton method of convex SC^1 minimization problems*, Journal of Optimization Theory and Application, 85 (1995), pp. 633–648.

[78] S. Petry, C. Flexeder, and G. Tutz, *Pairwise fused lasso*, tech. rep., University of Munich, 2011.

[79] J. Portilla, V. Strela, M. J. Wainwright, and E. P. Simoncelli, *Image denoising using scale mixtures of Gaussians in the wavelet domain*, IEEE Transactions on Image Processing, 12 (2003), pp. 1338–1351.

[80] M. Powell, *A method for nonlinear constraints in minimization problems*, in In R. Fletcher, editor, Optimization, Academic Press, 1969, pp. 283–298.

[81] L. Qi and J. Sun, *A nonsmooth version of Newton's method*, Mathematical Programming, 58 (1993), pp. 353–367.

[82] S. M. Robinson, *Strongly regular generalized equations*, Mathematics of Operations Research, 5 (1980), pp. 43–62.

[83] R. T. Rockafellar, *A dual approach to solving nonlinear programming problems by unconstrained optimization*, Math. Programming, 5 (1973), pp. 354–373.

[84] R. T. Rockafellar, *Augmented Lagrangians and applications of the proximal point algorithm in convex programming*, Mathematics of Operations Research, 1 (1976), pp. 97–116.

[85] ——, *Monotone operators and the proximal point algorithm*, SIAM Journal on Control and Optimization, 14 (1976), pp. 877–898.

[86] A. Ron and Z. Shen, *Frames and stable bases for shift-invariant subspaces of $L_2(\mathbb{R}^d)$*, Canadian J. Math., 47 (1995), pp. 1051–1094.

[87] ——, *Affine systems in $L_2(\mathbb{R}^d)$: the analysis of the analysis operator*, Journal of Functional Analysis, 148 (1997), pp. 408–447.

[88] L. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Physica D, 60 (1992), pp. 259–268.

[89] S. Setzer, G. Steidl, and T. Teuber, *Deblurring poissonian images by split bregman techniques*, Journal of Visual Communication and Image Representation, 21 (2010), pp. 193–199.

[90] Y. She, *Sparse regression with exact clustering*, Electronic Journal of Statistics, 4 (2010), pp. 1055–1096.

[91] Z. Shen, *Wavelet frames and image restorations*, in Proceedings of the International Congress of Mathematicians, R. Bhatia, ed., vol. IV, Hyderabad, India, 2010, Hindustan Book Agency, pp. 2834–2863.

[92] Z. Shen, K.-C. Toh, and S. Yun, *An accelerated proximal gradient algorithm for frame based image restoration via the balanced approach*, SIAM Journal on Imaging Sciences, 4 (2011), pp. 573–596.

[93] J. Shi and S. Osher, *A nonlinear inverse scale space method for a convex multiplicative noise model*, SIAM Journal on Imaging Sciences, 1 (2008), pp. 294–321.

[94] J. Shi, W. Yin, S. Osher, and P. Sajda, *A fast hybrid algorithm for large-scale ℓ_1-regularized logistic regression*, J. Machine Learning Research, 11 (2010), pp. 713–741.

[95] M. Slawski, W. Castell, and G. Tutz, *Feature selection guided by structural information*, Annals of Applied Statistics, 4 (2010), pp. 1056–1080.

[96] J.-L. Starck, M. Elad, and D. L. Donoho, *Image decomposition via the combination of sparse representations and a variational approach*, IEEE Trans. Image Proc., 14 (2005), pp. 1570–1582.

[97] G. Steidl and T. Teuber, *Removing multiplicative noise by Douglas-Rachford splitting*, Journal of Mathematical Imaging and Vision, 36 (2010), pp. 168–184.

[98] J. F. Sturm, *Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones*, Optimization Methods and Software, 11/12 (1999), pp. 625–653.

[99] R. Tibshirani, *Regression shrinkage and selection via the lasso*, Journal of the Royal Statistical Society. Series B (Methodological), (1996), pp. 267–288.

[100] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, *Sparsity and smoothness via the fused lasso*, J. Royal Statist. Soc. B, 67 (2005), pp. 91–108.

[101] K.-C. Toh, M. J. Todd, and R. H. Tütüncü, *SDPT3-a MATLAB software package for semidefinite programming, version 1.3*, Optimization Methods and Software, 11/12 (1999), pp. 545–581.

[102] Y. Wang, J. Yang, W. Yin, and Y. Zhang, *A new alternating minimization algorithm for total variation image reconstruction*, SIAM J. Imaging Sciences, (2008), pp. 248–272.

[103] Z. Wen, W. Yin, D. Goldfarb, and Y. Zhang, *A fast algorithm for sparse reconstruction based on shrinkage, subspace optimization and continuation*, SIAM J. Scientific Computing, 32 (2010), pp. 1809–1831.

[104] Y. Xiao, J. Yang, and X. Yuan, *Alternating algorithms for total variation image reconstruction from random projections*, Inverse Problems and Imaging, 6 (2012), pp. 547–563.

[105] G.-B. Ye and X. Xie, *Split Bregman method for large scale fuesd lasso*, Computational Statistics and Data Analysis, 55 (2011), pp. 1552–1569.

[106] W. Yin, S. Osher, D. Goldfarb, and J. Darbon, *Bregman iterative algorithm for ℓ_1 minimization with applications to compressed sensing*, SIAM Journal on Imaging Sciences, 1 (2008), pp. 143–168.

[107] M. Yuan and Y. Lin, *Model selection and estimation in regression with grouped variables*, J. Royal Statist. Soc. B, 68 (2006), pp. 49–67.

[108] S. Yun and K.-C. Toh, *A coordinate gradient descent method for l1-regularized convex minimization*, Computational Optimization and Applications, 48 (2011), pp. 273–307.

[109] P. Zhao and B. Yu, *Grouped and hierarchical model selection through composite absolute penalties*, tech. rep., University of California, Berkeley, 2006.

[110] X. Y. Zhao, D. Sun, and K.-C. Toh, *A Newton-CG augmented Lagrangian method for semidefinite programming*, SIAM Journal on Optimization, 20 (2010), pp. 1737–1765.

[111] H. Zou and T. Hastie, *Regularization and variable selection via the elastic net*, J. Royal Statist. Soc. B, 67 (2005), pp. 301–320.

AUGMENTED LAGRANGIAN BASED ALGORITHMS FOR CONVEX OPTIMIZATION PROBLEMS WITH NON-SEPARABLE ℓ_1-REGULARIZATION

GONG ZHENG

NATIONAL UNIVERSITY OF SINGAPORE
2013

Augmented Lagrangian based algorithms for convex optimization problems with non-separable ℓ_1-regularization Gong Zheng 2013