A SPECTRAL PROXIMAL METHOD FOR SPARSE OPTIMISATION ON UNDERDETERMINED LINEAR SYSTEMS

GILLIAN WOO YI HAN

MASTER OF SCIENCE

LEE KONG CHIAN FACULTY OF ENGINEERING AND SCIENCE

UNIVERSITI TUNKU ABDUL RAHMAN

DECEMBER 2021

A SPECTRAL PROXIMAL METHOD FOR SPARSE OPTIMISATION ON UNDERDETERMINED LINEAR SYSTEMS

By

GILLIAN WOO YI HAN

A dissertation submitted to the
Department of Mathematical and Actuarial Sciences,
Lee Kong Chian Faculty of Engineering and Science,
Universiti Tunku Abdul Rahman,
in partial fulfillment of the requirements for the degree of
Master of Science

December 2021

ABSTRACT

A SPECTRAL PROXIMAL METHOD FOR SPARSE OPTIMISATION ON UNDERDETERMINED LINEAR SYSTEMS

Gillian Woo Yi Han

In this research, we will solve the l_0-norm sparse optimisation problem. This is a l_0-norm problem with an underdetermined system as its constraint. Using the Lagrangian method, this problem is transformed into an unconstrained optimisation problem. However, it cannot be solved by using the standard optimisation algorithm since l_0-norm is nonconvex and non-smooth. Hence, a new method, the spectral proximal method (SPM) has been proposed and applied to the l_0-norm unconstrained optimisation problem. This method is a combination of the proximal method and spectral gradient method. Based on previous research, the performance of the spectral gradient method is better than the other standard unconstrained optimisation methods due to the fact that the approximation of the full rank Hessian matrix is replaced by a diagonal matrix. Hence, the memory required $O(n)$ storage instead of $O(n^2)$ storage. Convergent analysis of this method is established. The efficiency of the proposed method with the existing version of proximal gradient methods as its benchmarks are compared using Python software on simulated datasets and also large real-world MNIST datasets. The results show that our proposed method is more robust and stable for finding sparse solution of the linear system.

ACKNOWLEDGMENTS

First and foremost, praises and thanks to the God, the Almighty, for His showers of blessings throughout my research work to complete the research successfully.

I would like to express my deep and sincere gratitude to my research supervisor, Dr. Sim Hong Seng for giving me the opportunity to do research and providing invaluable guidance throughout this research. His dynamism, vision, sincerity and motivation have deeply inspired me. He has taught me the methodology to carry out the research and to present the research works as clearly as possible. It was a great privilege and honor to work and study under his guidance. I am extremely grateful for what he has offered me.

Besides that, I would like to express my sincere gratitude to my co-supervisor, Dr. Goh Yong Kheng for the continuous support of my master study and research. His patience, motivation, enthusiasm, and immense knowledge has guided me in all the time of research and writing of this dissertation. I am extending my heartfelt thanks to his wife, family for their acceptance and patience during the discussion I had with him on research work and dissertation preparation.

I am extremely grateful to my parents, Woo Kok Woh and Loo Gek Ching, for their love, prayers, caring and sacrifices for educating and preparing me for my future. I am very much thankful to their continuing support to complete this research work. Also I express my thanks to my friends for their support and valuable prayers.

Finally, my thanks go to all the people who have supported me to complete the research work directly or indirectly.

APPROVAL SHEET

This dissertation entitled "**A SPECTRAL PROXIMAL METHOD FOR SPARSE OPTIMISATION ON UNDERDETERMINED LINEAR SYSTEMS**" was prepared by GILLIAN WOO YI HAN and submitted as partial fulfillment of the requirements for the degree of Master of Science at Universiti Tunku Abdul Rahman.

Approved by:

[illegible]

(Dr. SIM HONG SENG) Date: 21 December 2021
Assistant Professor/Supervisor
Department of Mathematical and Actuarial Sciences
Lee Kong Chian Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

[illegible]

(Dr. GOH YONG KHENG) Date: 21 December 2021
Associate Professor/Co-supervisor
Department of Mathematical and Actuarial Sciences
Lee Kong Chian Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

LEE KONG CHIAN FACULTY OF ENGINEERING AND SCIENCE

UNIVERSITI TUNKU ABDUL RAHMAN

Date: 21 December 2021

SUBMISSION OF DISSERTATION

It is hereby certified that **Gillian Woo Yi Han** (ID No: **19UEM06032**) has completed this dissertation entitled "A SPECTRAL PROXIMAL METHOD FOR SPARSE OPTIMISATION ON UNDERDETERMINED LINEAR SYSTEMS" under the supervision of Dr. Sim Hong Seng (Supervisor) from the Department of Mathematical and Actuarial Sciences, Lee Kong Chian Faculty of Engineering and Science and Dr. Goh Yong Kheng (Co-supervisor) from the Department of Mathematical and Actuarial Sciences, Lee Kong Chian Faculty of Engineering and Science.

I understand that the University will upload softcopy of my dissertation in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

[illegible]

Gillian Woo Yi Han

DECLARATION

I **GILLIAN WOO YI HAN** hereby declare that the dissertation is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UTAR or other institutions.

[illegible]

(GILLIAN WOO YI HAN)

Date: 21 December 2021

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

BB	Barzilai-Borwein method
MNIST	Modified National Institute of Standards and Technology
PBFGS	Proximal with Broyden-Fletcher-Goldfarb-Shann method
PSD	Proximal with steepest descent method
SPM	Spectral proximal method
VM-PG	Variable metric proximal gradient method

CHAPTER 1

INTRODUCTION

1.1 Introduction to Optimisation

Optimisation is a science that determines the optimum solution to a mathematically specified problem. It is a mathematical technique in applied mathematics, such as linear programming or system analysis, to maximize or minimize the value of a function of several variables subject to a set of constraints. It entails establishing a problem's optimality criterion, determining methods and algorithms, studying the methods' structure, and also finding the solutions by computer (Nocedal and Wright, 2006).

There are three fundamental components for optimisation problems. The first component is an objective function that must be minimized or maximized. The second component is a set of variables, which we refer to as vectors, x are quantities whose values can be changed to optimize the objective function value. The third fundamental component is a set of constraints that restricts the values of the variables. Different mathematical properties exist for optimisation problems. The continuous variable problems require a different strategy from the discrete or combinatorial variable problems.

The optimisation problem can be expressed mathematically as follows:

$$\begin{aligned} &\min f(x), \\ &\text{subject to } g_i(x) \leq b_i, \quad i = 1,\, 2,\, 3,\, \ldots,\, m, \end{aligned} \tag{1.1}$$

where $x = (x^{(1)},\, x^{(2)},\, \ldots,\, x^{(n)})$ is the problem variable. The objective function is defined as $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$, and $g_i(x) \leq b_i : \mathbb{R} \rightarrow \mathbb{R}$, $i = 1,\, 2,\, 3,\, \ldots,\, m$ are the constraints. The constraints can be either equal or unequal, and the constants $b_1,\, b_2,\, b_3,\, \ldots,\, b_m$ represent the boundaries or limitations of the constraints. If there is no constraint, the optimisation is called as unconstrained optimisation. The problem (1.1) is solved if the optimum point, x^* has the smallest objective function value among all vectors that satisfy the constraints (Boyd et al., 2004*a*).

Modelling is the process of determining a problem's objective function, variables, and constraints. The most essential stage in optimisation process is to construct an appropriate model. If the model is overly simple, it may not useful in real-world situations. It may, however, be costly to solve if it is too complex. To find the solution to the model, an optimisation algorithm will be chosen after the model has been created. There are a number of algorithms, so an appropriate algorithm customised to a particular type of optimisation issue or application must be chosen (Nocedal and Wright, 2006).

There are two important classes of optimisation, known as linear programming and nonlinear programming. The problems in linear class optimisation entail minimising or maximising a linear objective function where the variables are real numbers and the variables must meet the constraints, which are a set of linear equalities and inequalities.

Nonlinear programming uses real numbers as variables, while the objective function or some of the constraints are nonlinear functions. Optimisation is useful in many areas, such as statistics, aerodynamics, chemical engineering, and operational research (Niwattisaiwong and Suriya, 2018; Fei-Yue et al., 2019; Johansson et al., 2017; Stojaković et al., 2018; Yang et al., 2019; Huang and Li, 2012; Majozi et al., 2015; Agarana et al., 2016; Teplická et al., 2020).

1.1.1 Types of Optimisation

Continuous and Discrete Optimisation

Some models require the variables to be integer values. Discrete optimisation problems are models with discrete variables. Discrete optimisation problems not only contain integer and binary variables, but also more abstract variable objects like permutations of an ordered set. The variable x from discrete optimisation problem is drawn from a finite set, while the feasible set for the continuous optimisation problem is generally uncountably infinite. Models with continuous variables are continuous optimisation problems. Continuous optimisation problems are typically easier to solve than the discrete optimisation problems because the smoothness of the functions provides objective function and constraint information at a point x as a deduction of function's behaviour at points in the neighborhood of x. Continuous optimisation algorithms are frequently used to solve discrete optimisation problem because many discrete optimisation algorithms generate a sequence of continuous subproblems (Nocedal and Wright, 2006).

Constrained and Unconstrained Optimisation

An unconstrained optimisation problem arises directly in many practical applications. Natural constraints on the variables are safe to disregard since they have no effect on the solution or conflict with algorithms. Unconstrained problems can also occur when constrained optimisation problems are reformulated with a penalty component in the objective function that discourages constraint violations (Nocedal and Wright, 2006).

Constrained optimisation problems emerge from applications in which the variables are constrained in some way. The constraints on the variables may be simple limits, more general linear constraints, or nonlinear inequalities that indicate complicated connections among the variables. There are two types of constrained optimisation problems: linear and nonlinear. The objective function and all the constraints of a linear programming problem are linear functions of x. In disciplines of management, financial and economic, this sort of problem is frequently stated and addressed. A nonlinear programming problem is the problem with one or more of its constraints or objectives are nonlinear. They naturally appear in the physical sciences and engineering (Nocedal and Wright, 2006).

Global and Local Optimisation

In nonlinear optimisation problems, many optimisation algorithms seek only local solutions. A local minimum of a function is a point where the objective function is less than or equal to the value at all other possible nearby points, but may be larger than at a distant point. A global solution minimum is a point where the value of the function is less than or equal to the value at all possible points. For convex and linear problems, the local minimum is also the global minimum. For constrained and unconstrained nonlinear problems, the local minimum found might not be the global solution (Nocedal and Wright, 2006).

1.1.2 Gradient of a function

In n-dimensional space, a function's gradient is defined as a vector:

$$g = \nabla f = \begin{Bmatrix} \partial f / \partial x^{(1)} \\ \partial f / \partial x^{(2)} \\ \vdots \\ \partial f / \partial x^{(n)} \end{Bmatrix}. \tag{1.2}$$

In optimisation theory, the direction of the gradient is crucial . It is interpreted as the steepest ascent's direction. This is because if the gradient is non-zero, the function value grows fast as we go along the gradient direction from whatever position. However, the gradient direction is just a local property.

Due to this, the direction of steepest descent is represented by a negative gradient vector. Gradient-based optimisation methods are the method of finding search direction that directly or indirectly include a gradient vector. In comparison with the methods that do not apply the gradient vector in the search direction, gradient-based optimisation methods are anticipated to move relatively faster to the minimum point (Rao, 2009).

1.1.3 Hessian matrix

Let $f : \mathbb{R}^J \to \mathbb{R}$ has continuous second partial derivatives. Then, $H(x)$ is the Hessian of f at point x. It is a symmetric matrix and is equal to $\nabla^2 f(x)$.

$$H(x) = \left[\frac{\partial^2 f(x)}{\partial x^{(i)} \partial x^{(j)}} \right]. \tag{1.3}$$

From Taylor's theorem (*Theorem 1.1*), assume that $f(x)$ has continuous second partial derivatives. Then there are two points x and x^* in $\mathbb{R}^n$ giving

$$f(x) = f(x^*) + \nabla f(x^*)^T(x - x^*) + \frac{1}{2}(x - x^*)^T H(z)(x - x^*), \qquad (1.4)$$

where $z = \theta x^* + (1 - \theta)x$, for some scalar θ with $0 \leq \theta \leq 1$.

$\nabla f(x^*) = 0$ when f is twice differentiable and x^* is the critical point. The equation (1.4) gives

$$f(x) = f(x^*) + \frac{1}{2}(x - x^*)^T H(z)(x - x^*). \qquad (1.5)$$

$H(x)$ is positive definite in the neighbourhood of x^* if $H(x^*)$ is positive definite. Thus, from (1.5) implies that $f(x^*) \leq f(x)$ in some nearby neighbourhood of x^*, and therefore it is a local minimizer. Moreover, if $H(x)$ is positive definite and $f(x^*) \leq f(x)$ for every x with $Ax = b$, x^* is a global minimizer (Best, 2010).

If the Hessian matrix is positive definite, the point x^* is a relative minimum point. Similarly, if the Hessian matrix is negative definite, the point x^* is a relative maximum point (Rao, 2009).

1.1.4 Convexity

A fundamental concept in optimisation is convexity. When the straight line segment joining any two points in S falls fully within S, the set of $S \in \mathbb{R}^n$ is termed a convex set. When any $x \in S$, $y \in S$, and $\alpha \in [0,1]$ have $\alpha x + (1 - \alpha)y \in S$, the "convex" term can be applied to the set mathematically. When a function's domain S is a convex set and if $x \in S$

and $y \in S$, the function f is convex and the following relation holds:

$$f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha)f(y), \text{ for all } \alpha \in [0,1]. \tag{1.6}$$

Convex quadratic function $f(x) = x^T Hx$, where H is a symmetric positive semidefinite matrix, and convex linear function $f(x) = c^T x + \alpha$, for any constant vector $c \in \mathbb{R}^n$, are the examples of convex functions. If the inequality, $\leq$ in (1.6) is substituted with $<$, the function f is strictly convex. If f is a concave function, $-f$ is a convex function.

If the objective function φ is convex and $\mathscr{C}$ is a convex set, an optimisation problem in the form of

$$\begin{aligned} &\min_x \varphi(x) \\ &\text{such that } x \in \mathscr{C}, \end{aligned} \tag{1.7}$$

is convex . A convex problem's local minimum is necessarily a global minimum. Conversely, nonconvex problems might consist of sub-optimal local minima and they are recognized as the solutions for the problem. However, we cannot identify whether there exists any solution that will further minimize the objective function. As a consequence, the initial point for the iterative optimisation algorithm becomes pivotal in determining the quality of the solution (Antonello et al., 2018).

Most of the nonconvex problems are affected by the initialization issue. Random initialization utilising distributions generated by evaluating the available data is one of the techniques used (Theodoridis, 2015). There is no general rule exists for initialization and it is usually problem-dependent. To avoid this issue, many nonconvex optimisation problems are approximated by convex relaxation. By replacing nonconvex functions with convex functions that have the similar properties, nonconvex functions are relaxed. One of the most often used approaches is the l_1-regularization (LASSO) and it will be discussed further in Section 2.3.1.

1.1.5 Review of Minimizer

A global minimizer, x^* is defined as the point where the function, f attains least value or when $f(x^*) \leq f(x)$ for all $x \in \mathbb{R}^n$. The optimum solution for the function is a global minimizer, however, it usually takes a lot of time and required many iterations to obtain the solution. Therefore, most of the optimisation algorithms are focusing on finding local minimizers.

A local minimizer is a point that obtains the lowest value in a neighbourhood, $\mathcal{N}$ of x^* such that $f(x^*) \leq f(x)$ for all $x \in \mathcal{N}$. Weak local minimizer is another name for this point. The strict local minimizer is a point x^*that fulfils $f(x^*) < f(x)$ in a neighbourhood $\mathcal{N}$ of x^* for all $x \in \mathcal{N}$ and $x \neq x^*$. If the smooth function f is twice continuously differentiable, the Hessian, $\nabla^2 f(x^*)$ and the gradient, $\nabla f(x^*)$ can be utilised to find the local minimizer x^*.

Theorem 1.1 (Taylor's Theorem). *Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and $d \in \mathbb{R}^n$. Then*

$$f(x+d) = f(x) + \nabla f(x+td)^T d, \tag{1.8}$$

for some $t \in (0,1)$. Furthermore, if f is twice differentiable,

$$\nabla f(x+d) = \nabla f(x) + \int_0^1 \nabla^2 f(x+td)d\,dt, \tag{1.9}$$

and (1.8) can be rewritten as

$$f(x+d) = f(x) + \nabla f(x)^T d + \frac{1}{2} d^T \nabla^2 f(x+td)d, \tag{1.10}$$

for some $t \in (0,1)$.

Theorem (1.1) is the fundamental theorem in optimisation. Most of the definitions and theorems in optimisation are derived based on Theorem (1.1). Some of the useful conditions are listed below (Nocedal and Wright, 2006).

Theorem 1.2 (First-Order Necessary Conditions). *If f is a continuously differentiable function in an open neighborhood of x^* and x^* is a local minimizer, then its gradient $\nabla f(x^*) = 0$.*

If $\nabla f(x^*) = 0$, x^* is a stationary point. Theorem 1.2 shows that any local minimizer is a stationary point.

Theorem 1.3 (Second-Order Necessary Conditions). *If x^* is a local minimizer of function f and its Hessian $\nabla^2 f$ is continuous in an open neighborhood of x^*, then its gradient $\nabla f(x^*) = 0$ and Hessian $\nabla^2 f(x^*)$ is a positive semidefinite matrix.*

Theorem 1.4 (Second-Order Sufficient Conditions). *Suppose that $\nabla^2 f$ is continuous in an open neighborhood of x^*, $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is* *positive definite. Then x^* is a strict local minimizer of f.*

Theorem 1.5. *When f is convex, any local minimizer x^* is a global minimizer. In addition, if f is convex and differentiable, then any stationary point of f is a global minimizer.*

1.1.6 Overview Optimisation Algorithms

For solving unconstrained smooth optimisation problems, there are many efficient algorithms that have been established. The majority of optimisation algorithms are iterative. The iterative algorithms begin with an estimate for the variable, x_0. Starting from x_0, the algorithm will generate a sequence of improving approximate solutions, $\{x_k\}_{k=1}^{\infty}$ until it is terminated in m iterations when either there are no more progress can be made or when the approximate solution has reached a sufficient accuracy.

The following are characteristics of a good algorithm:

- Robustness. The algorithms should be able to solve most of the problems in their class with a reasonable starting point value.
- Efficiency. The algorithms should not require excessive storage and computational time.
- Accuracy. The algorithms should be able to discover a solution for an optimisation problem with a specified precision without being excessively sensitive to arithmetic rounding errors or data inaccuracies that arise when using computer software to perform the algorithm.

These characteristics, however, may be incompatible. For instance, a fast convergence rate method may need additional computer storage. A robust method, on the other hand, could have the slowest rate of convergence. Central concerns are the tradeoffs between each of the characteristics. The mathematical theory of optimisation is the basis for most algorithms (Nocedal and Wright, 2006).

1.1.7 Sensitivity Analysis

Condition Number

In the area of numerical analysis, the condition number is important. The sensitivity of the output to errors or modifications of the input is measured by the condition number. The error could be any uncertainty or round-off error. If the condition number $\mathcal{K}(A) = 10^k$, it implies that up to k decimal digits of precision may be lost (Cheney and Kincaid, 2012).

The forward problem is the computation of the output b given an input x for the linear system $Ax = b$. When the forward problem has a unique solution b that changes continuously depends on the input x, it is a well-posed problem. However, to estimate the approximate solution, x based on the output b, the inverse of the mapping A^{-1} needs to be computed. In most cases, inverse problems are ill-posed. This means that there is no assurance of a unique solution, which may or may not exist at all.

The condition number of A in the linear system $Ax = b$ can be calculated as:

$$\mathcal{K}([A]) = \frac{|\lambda_{max}(A)|}{|\lambda_{min}(A)|}, \tag{1.11}$$

where $\lambda_{\max}(A)$ and $\lambda_{\min}(A)$ are maximal and minimal (by moduli) eigenvalues of a square matrix A respectively. In non-square matrix A, λ

is its singular value. A matrix with high condition number is known as ill-conditioned while the matrix with low condition number is known to be well-conditioned.

A small change or error in the input, b may cause a large change to the output, x^*. It is known as ill-conditioning if the matrix A has a high conditioned number. When the estimations of x are substantially contaminated by noise or the model is inaccurate, the overfitting problem occurs (Antonello et al., 2018). Moreover, if matrix A is a singular matrix or when its determinant is 0, the solution does not exist.

The existence of the solution and the convergence of the iterative optimisation procedure are affected by the condition number of the matrix A in the linear system. The accuracy is defined by calculating the error in the least square sense, $\|r\| = \|b - Ax\|$. If the value of $\mathcal{K}(A)$ is near to 1, it indicates the matrix is well-conditioned. The iterative process will converge and the approximate solution, x^* will have good accuracy. However, if the value of $\mathcal{K}(A)$ is far from 1, it is an ill-conditioned matrix. The highly ill-conditioned matrix is not invertible and causes divergence for the iterative process (Pyzara et al., 2011).

1.2 Background of the Study

There has been an increase interest in the general field of sparsity in the past few years (Sharma et al., 2019; Li et al., 2020; Blanquero et al., 2020). In the fields of compressive sensing, image processing, machine learning, and statistics, sparse optimisation to underdetermined linear systems has become a popular study topic (Deng et al., 2013; Le Thi et al., 2015). An optimisation problem with zero-norm objective function or constraints is known as a l_0-norm sparse optimisation. l_0-norm or

$\| \, . \, \|_0$ denotes the zero-norm on $\mathbb{R}^n$, which indicates the number of nonzero elements in x. In optimisation problems, the l_0-norm plays an essential and critical role in modelling data sparsity and choosing representative variables. The norm of a mathematical object is referred to a quantity that measures the length or size of the vector in some sense (Nocedal and Wright, 2006). In some literature, some people disagree l_0-norm as a proper norm because it does not satisfy the property of a norm. $\|\lambda x\|_0 = \|x\|_0$ is obtained for every $x \in \mathbb{R}^n$ and $\lambda \neq 0$, indicating that l_0-norm is not being absolutely homogeneous and is not a norm (Le Thi et al., 2015). However, in this project, we will adopt it as a norm (Xu and Zhao, 2020).

In this research, we are motivated to propose an efficient, general purpose algorithmic approach and efficient implementations for the following nonconvex optimisation problem:

$$\begin{aligned} &\min \|x\|_0 \\ &\text{subject to } Ax = b, \end{aligned} \tag{1.12}$$

where $Ax = b$ is underdetermined system. Since l_0-norm is non-convex and non-smooth, the standard techniques for handling the smooth optimisation problem cannot solve it. We consider the constraint of problem (1.12), $Ax = b$ to be an underdetermined system. It is difficult to solve and requires large computational time when the system involves large dimensional dataset.

Since l_0-norm is known to be intractable due to its nonconvexity and discontinuity (Natarajan, 1995), l_1-norm convex regularization is a popular approach for replacing the l_0-norm (Candes, 2008). Because of its exact recovery property under certain conditions, the l_1-relaxation is quite common (Candes and Tao, 2005). However, l_1-regularizer does not always provide the true relevant variables (Candes et al., 2008). It penalizes the amplitude uniformly and thus, underestimates high-amplitude components of x. Non-convex l_0-norm regularization provides greater advantages than convex l_1-norm regularization in many cases (Bao et al., 2016; Zhang et al., 2012; Sun and Tao, 2014*a*; Trzasko and Manduca, 2008).

To this end, we propose an efficient algorithm to solve the l_0-norm sparse optimisation problem. In this project, the objective function, l_0-norm, and its constraint are minimized directly with the proposed method. Some proximal gradient algorithms such as the Newton-type method have been proposed previously (Antonello et al., 2018). However, the proximal quasi-Newton is demanding in memory due to the storage of the Hessian matrix. To fill in the gap, we propose a new proximal gradient method, known as spectral proximal method (SPM). SPM incorporates the spectral gradient method and proximal method. Sim et al. (2019) showed that the spectral gradient method has a better efficiency compared to the other standard optimisation methods. The SPM method is anticipated to be well-performed in solving sparse optimisation problems.

The objectives of this project are to:

- incorporate spectral gradient method and proximal method to solve the sparse optimisation problems.
- establish the convergence properties of the proposed method.
- develop an executable code using Python software to test the efficiency of the proposed method.

1.3 Structure of the Thesis

This thesis is structured as follows. Chapter 2 describes the literature review based on the existing research work. Some standard optimisation methods for solving unconstrained smooth optimisation problems and also the proximal method for solving non-smooth problems are highlighted. Spectral gradient method proposed by Sim et al. (2019) has been reviewed in order to incorporate with the proximal algorithm based on the ideas from the existing proximal gradient algorithms. The definition and some existing approaches of l_0-norm sparse optimisation are discussed.

In Chapter 3, a new proximal gradient method, the spectral proximal method (SPM) is proposed to solve the l_0-norm sparse optimisation problems. The convergent analysis is established and the scope of the experiments is listed. In Chapter 4, an executable programming code has been developed using Python to test the efficiency of the method. The performances between the spectral proximal method and some existing proximal gradient methods are compared and discussed. The simulated datasets and MNIST real-world dataset (LeCun et al., 1998) are used in the numerical experiments. Chapter 5 concludes the finding of the research and some possible future works are proposed.

CHAPTER 2

LITERATURE REVIEW

2.1 Gradient-Based Optimisation Techniques

The unconstrained optimisation problem is as follows:

$$\min_{x \in \mathbb{R}^n} f(x), \tag{2.1}$$

where f is a twice continuously differentiable function and gradient is denoted by $\nabla f(x)$. In this research, our interest is to solve large-scale sparse optimisation problems. In this case, due to the large dimensions of the datasets, the Hessian of f requires a large amount of storage or it will not be available.

The updating formula of the gradient method requires search direction, d_k and step length, α_k in every iteration. The updating formula is given as follows:

$$x_{k+1} = x_k + \alpha_k d_k, \tag{2.2}$$

Most gradient-based optimisation methods use the direction, d_k such that $d_k^T \nabla f_k < 0$. This is because it ensures that the function value of f will decrease. In most cases, the search direction is in the form of

$$d_k = -A_k^{-1} \nabla f_k, \tag{2.3}$$

where A_k is a non-singular symmetric matrix. A_k is the identity matrix, I in steepest descent. A_k is the actual Hessian matrix $\nabla^2 f(x_k)$ in Newton's technique, but A_k is an approximation to the Hessian matrix in Quasi-Newton methods. When A_k is positive definite, it ensures that

$$d_k^T \nabla f_k = -\nabla f_k^T A_k^{-1} \nabla f_k < 0. \tag{2.4}$$

Hence, d_k is the descent direction.

A tradeoff arises when determining the step length, α_k. This is due to the fact that we need a α_k that can substantially reduce the function value f, but does not require too much computational time. Differentiating the following minimization problem yields the optimal step length, α_k:

$$\min_{\alpha>0} f(x_k + \alpha d_k). \tag{2.5}$$

Problem (2.5) is solved exactly but it requires many function evaluations and gradient evaluations which will increase the computational time especially in large dimensional problems. Instead, we can generate a limited number of trial step lengths to determine the step length that is closest to the minimum of Problem (2.5). This is known as inexact step length. Practically, to reduce the cost, most of the optimisation problems are solved by using inexact step length (Nocedal and Wright, 2006).

2.1.1 Inexact Step Length

Inexact step length guarantees an adequate reduction in function value, f. The line search algorithm will attempt a few values of α iteratively and stop when certain termination conditions are satisfied. A simple condition could be imposed is $f(x_k + \alpha_k d_k) < f(x_k)$, which requires f to be decreased. However, this inequality does not preserve the decreasing in function value, f in every iteration. Thus, the iterations will fail to converge to the convex problem's minimizer. To overcome this, some conditions are given to assure that the function value, f, decreases sufficiently.

The Wolfe Conditions

Wolfe conditions are first published by Philip Wolfe in 1969 (Wolfe, 1969, 1971). Inexact line search requires the α_k to satisfy the *sufficient decrease* condition. This inequality is written as

$$f(x_k + \alpha d_k) \leq f(x_k) + c_1 \alpha \nabla f_k^T d_k, \tag{2.6}$$

where constant $c_1 \in (0,1)$. This inequality is also called as *Armijo condition* (Armijo, 1966). The right hand side is a linear function, and is denoted by $l(\alpha)$ and the left hand side is denoted by $\phi(\alpha)$. Although $c_1 \nabla f_k^T d_k$ is a negative slope, but for small positive values of α, the function $l(\cdot)$ is above the graph of ϕ because $c_1 \in (0,1)$. When $\phi(\alpha) \leq l(\alpha)$, α is considered acceptable.

The condition (2.6) is satisfied for all sufficiently small values of α. Hence, *curvature condition* is introduced to rule out unacceptably short steps. This condition requires α_k to satisfy

$$\nabla f(x_k + \alpha_k d_k)^T d_k \geq c_2 \nabla f_k^T d_k, \tag{2.7}$$

where constant $c_2 \in (c_1, 1)$ and c_1 is from (2.6). The left hand side is the derivative $\phi'(\alpha_k)$.

Condition (2.7) requires $\phi'(\alpha_k) \geq \phi'(0)$. This indicates that the f can be reduced significantly by moving along the direction, d_k if the slope $\phi'(\alpha_k)$ is strongly negative. Moreover, the function f is expected not to be reduced much in d_k if $\phi'(\alpha_k)$ is only slightly negative. In this case, the line search algorithm is terminated. Typically, when d_k is obtained from Newton or quasi-Newton method, c_2 is chosen to be 0.9 and when d_k is chosen by nonlinear conjugate gradient method, c_2 is 0.1 (Nocedal and Wright, 2006).

The α_k satisfies the *Wolfe conditions* if the sufficient decrease and curvature conditions hold. The Wolfe condition is restated as

$$f(x_k + \alpha d_k) \leq f(x_k) + c_1 \alpha \nabla f_k^T d_k, \tag{2.8a}$$

$$\nabla f(x_k + \alpha_k d_k)^T d_k \geq c_2 \nabla f_k^T d_k, \tag{2.8b}$$

with $0 < c_1 < c_2 < 1$.

Wolfe conditions can be satisfied by a step length that is not near to the minimizer of ϕ. Hence, by modifying the curvature condition, the $\phi'(\alpha_k)$ is restricted not to be too positive. This forces α_k to lie in the neighborhood of the critical point of ϕ. The step length, α_k is thus required to satisfy *strong Wolfe conditions*:

$$f(x_k + \alpha d_k) \leq f(x_k) + c_1 \alpha \nabla f_k^T d_k, \tag{2.9a}$$

$$|\nabla f(x_k + \alpha_k d_k)^T d_k| \leq c_2 |\nabla f_k^T d_k|, \tag{2.9b}$$

with $0 < c_1 < c_2 < 1$.

Lemma 2.1. *Suppose that $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable. Let d_k be a descent direction at x_k, and assume that f is bounded below along ray*

$\{x_k + \alpha d_k | \alpha > 0\}$. *Then if* $0 < c_1 < c_2 < 1$, *there exist intervals of step lengths satisfying the Wolfe conditions (2.8) and the strong wolfe conditions (2.9).*

From Lemma (2.1), if the function f is smooth and bounded below, then the step lengths that meet the Wolfe conditions exist. The proof can be found in Nocedal and Wright (2006). Wolfe conditions are scale-invariant and can be utilised in a broad range of line search methods, especially in quasi-Newton methods (Nocedal and Wright, 2006).

The Goldstein Conditions

The *Goldstein conditions*, like Wolfe conditions, choose an appropriate step length α that satistfies Armijo condition while not being too short. The Goldstein conditions are as follows:

$$f(x_k) + (1-c)\alpha_k \nabla f_k^T d_k \le f(x_k + \alpha_k d_k) \le f(x_k) + c\alpha \nabla f_k^T d_k, \qquad (2.10)$$

with $0 < c < 1/2$. The first inequalities is to control the step length from below and the second inequalities is the condition from (2.6).

These conditions have a disadvantage, because they may exclude all minimizers of ϕ by the first inequalities in (2.10). The convergence theories for Wolfe conditions and Goldstein conditions are quite similar (Nocedal and Wright, 2006).

Sufficient Decrease and Backtracking

It is not enough if we only apply the sufficient decrease condition in (2.6) for the line search algorithm. *Backtracking line search* can dispense condition (2.8b) by just using sufficient decrease condition to find an appropriate step length.

The following is the backtracking line search algorithm using the Armijo condition:

Algorithm 1: Backtracking line search with Armijo condtion (BTA)

```
1. Choose an initial step length, $\alpha > 0$, $\tau \in (0,1)$ and $c \in (0,1)$.
2. Check whether $f(x_k + \alpha d_k) \leq f(x_k) + c\alpha \nabla f_k^T d_k$ is satisfied.
3. If it is satisfied, terminate the algorithm and set $\alpha_k = \alpha$. If not
   satisfied, set $\alpha = \tau\alpha$, where $\tau \in [\tau_{lo}, \tau_{hi}]$, for some constants
   $0 < \tau_{lo} < \tau_{hi} < 1$ and repeat Step 2.
```

In the next few sections, some common optimisation methods will be discussed.

2.1.2 Steepest Descent Method

Steepest descent was proposed by Cauchy (1847). This method is the simplest and oldest gradient method for solving large-scale unconstrained optimisation. It approaches the minimum point by moving iteratively in the steepest descent directions from an initial point x_0. In every iteration, the new search direction is orthogonal to the previous (Rao, 2009; Nocedal and Wright, 2006).

The advantage of the steepest descent method is that it requires only the first derivative of the function, $-\nabla f(x_k)$. The algorithm can be implemented easily and only a low storage, $O(n)$ is required. However, this method moves in a zig-zag-like path along the negative direction of the gradient, $-\nabla f_k$ towards the local minimum point. It is relatively slow when close to the minimum because near the local minimization the gradient is nearly zero. It requires to run numerous iterations process which can take forever for a badly scaled system and causes slow convergence (Wang, 2008).

This method is critical for the advancement of optimisation theory, but it is too slow for most real-world problems. In general, any methods that have a descent direction (with $-\nabla f_k$, one that yields an angle that is strictly smaller than 90°) guarantee a decrease in function value f if the step length is sufficiently small (Nocedal and Wright, 2006). Hence, more sophisticated techniques like the conjugate gradient method and quasi-Newton methods are often utilised (Poisel, 2012).

2.1.3 Conjugate Gradient Method

For solving systems of linear equations, Hestenes and Stiefel (1952) developed the conjugate gradient method. It was established to enhance the steepest descent method's convergence properties. It is a conjugate directions approach that use the negative gradient, $-\nabla f$.

Definition 2.1 (Conjugate Directions). *Let $Q = [Q]$ be an $n \times n$ symmetric matrix. A finite set $[d_i]$, $i = 1, 2, \ldots, n$ is said to be Q-conjugate if*

$$d_i^T Q d_j = 0, \quad \forall i \neq j, \quad i = 1, 2, \ldots, n, \quad j = 1, 2, \ldots, n. \tag{2.11}$$

A special case of conjugate directions is called orthogonal directions. It is obtained when Q is an identity matrix, I and satisfy (2.11).

Definition 2.2 (Quadratically Convergent Method). *The minimization method which the minimum point of a quadratic function in n variables can be found in n steps if using exact arithmetic.*

Any method that involves conjugate directions is quadratically convergent. Because to this feature, the minimum point of a quadratic function can be determined in at most n steps. Given that a quadratic may approximate every general function pretty well when approaching the optimum point and hence, the optimum point should be able to be identified by any quadratically convergent technique within a limited number of iterations. The method, however, may require more than n steps of iterations for ill-conditioned quadratic problems. This is because of the cumulative effect of rounding errors. Despite this limitation, the conjugate gradient method outperforms the steepest descent method (Rao, 2009). The conjugate gradient method, on the other hand, is less efficient than Newton and quasi-Newton methods, but it has the benefit of not requiring any Hessian matrices to be stored (Nocedal and Wright, 2006).

2.1.4 Newton's Method

Newton's method is also called as the Newton-Raphson method in numerical analysis. It was first created by Newton for the purpose of solving nonlinear equations and improved by Raphson. A multivariate function $f(x)$ at $x = x_k$ in a quadratic approximation may be expressed as follows using Taylor's series expansion:

$$f \approx f_k + \nabla f_k^T (x - x_k) + \frac{1}{2}(x - x_k)^T H_k (x - x_k), \qquad (2.12)$$

where $H_k = H|_{x_k}$ is the Hessian matrix (second partial derivatives of f evaluated at x_k). By differentiating (2.12) and set $\nabla f = 0$ for the minimum of the function f:

$$\nabla f = \nabla f_k + H_k(x - x_k) = 0. \qquad (2.13)$$

If H_k is nonsingular, (2.13) may be arranged in the following way to provide an improved estimate, x_{k+1}:

$$x_{k+1} = x_k - H_k^{-1} \nabla f_k. \tag{2.14}$$

The iterative process given by (2.14) converges to the optimum solution x^* from any random point x_0 that is close to the x^*, if H_0 is nonsingular. Newton's method is a second-order method since it utilizes the objective function's second partial derivatives, H_k.

Newton's method finds the minimum in one iteration for quadratic functions. However, the Newton's iterative technique may diverge or converge to saddle points and maximum points for nonquadratic functions. This problem can be addressed by rewriting (2.14) to

$$x_{k+1} = x_k + \alpha_k^* d_k = x_k - \alpha_k^* H_k^{-1} \nabla f_k, \tag{2.15}$$

where α_k^* represents the step length in the direction $d_k = -H_k^{-1} \nabla f_k$. This modification brings many advantages as stated below:

- The minimum point can be found in lesser steps compared to (2.14).
- The minimum point can be obtained in all cases, whereas (2.14) might not converge in all cases.
- Convergence to the saddle point or maximal point is typically avoided using this method.

Although Newton's method can be seen as a powerful minimization method, it is not very applicable in practice, due to the following reasons:

- It needs to store a $n \times n$ Hessian matrix, H_k.

- Sometimes it is difficult and impossible to compute the Hessian matrix, H_k.
- In every iteration, it needs an inversion of the Hessian matrix, H_k and the evaluation of $H_k^{-1}\triangledown f_k$.

These reasons cause this method becomes not applicable to large-dimensional optimisation problems (Rao, 2009).

2.1.5 Quasi-Newton Methods

The computation of the Hessian matrix of the function is the main drawback of the Newton's method. To overcome this, quasi-Newton methods provide an alternative way to approximate the inverse of the true Hessian matrix by an approximation, B_k. In every step, B_k is updated by using only the gradient evaluation of the function (Nocedal and Wright, 2006; Rao, 2009).

Taylor's theorem (Equation 1.10) is modified by adding and removing the term $\triangledown^2 f(x)d$, and we obtain

$$\triangledown f(x+d) = \triangledown f(x) + \triangledown^2 f(x)d + \int_0^1 [\triangledown^2 f(x+td) - \triangledown^2 f(x)]d\, dt. \quad (2.16)$$

The final integral term can be written as $o(\|d\|)$ because $\triangledown f(.)$ is continuous. By setting $x = x_k$ and $d = x_{k+1} - x_k$, this leads to

$$\triangledown f_{k+1} = \triangledown f_k + \triangledown^2 f_k(x_{k+1} - x_k) + o(\|x_{k+1} - x_k\|). \quad (2.17)$$

When x_k and x_{k+1} are the neighborhood of the solution x^* and lie within which $\triangledown^2 f$ is positive definite,

$$\triangledown^2 f_k(x_{k+1} - x_k) \approx \triangledown f_{k+1} - \triangledown f_k. \quad (2.18)$$

The new Hessian approximation A_{k+1} is required to satisfy the secant equation:

$$A_{k+1}s_k = y_k, \tag{2.19}$$

where

$$s_k = x_{k+1} - x_k, \qquad\qquad y_k = \nabla f_{k+1} - \nabla f_k, \tag{2.20}$$

Equation (2.19) can be written as

$$s_k = B_{k+1}y_k, \tag{2.21}$$

where $B_{k+1} = A_{k+1}^{-1}$ is the approximation of the inverse Hessian matrix, H_{k+1}^{-1}. B_{k+1} must be symmetric and positive definite to fulfil this equation.

The general formula for updating B_{k+1} is

$$B_{k+1} = B_k + \triangle B_k, \tag{2.22}$$

where $\triangle B_k$ are the difference between consecutive approximations B_k and B_{k+1}, and in practice is generally in rank 1 or rank 2.

The rank 1 updating formula for B_{k+1} is given as

$$B_{k+1} = B_k + \triangle B_k = B_k + \frac{(s_k - B_k y_k)(s_k - B_k y_k)^T}{(s_k - B_k y_k)^T y_k}. \tag{2.23}$$

This equation is Broyden formula (Broyden, 1967). The initial matrix of B_0 must be positive definite and symmetric. Equation (2.23) ensures that B_{k+1} remains symmetric if B_k is symmetric. However, the positive definiteness of the B_{k+1} is not guaranteed eventhough B_k is positive definite. Thus, the iterative minimization process might break down, especially for non-quadratic functions (Rao, 2009).

The rank 2 updating formula can be obtained by choosing $\triangle B_k$ as the total of two rank 1 updates, and is written as

$$B_{k+1} = B_k + \triangle B_k = B_k + \frac{s_k s_k^T}{s_k^T y_k} - \frac{(B_k y_k)(B_k y_k)^T}{(B_k y_k)^T y_k}. \tag{2.24}$$

Since

$$x_{k+1} = x_k + \alpha_k d_k, \tag{2.25}$$

where d_k is the search direction. We can rewrite $s_k = x_{k+1} - x_k$ as

$$s_k = \alpha_k d_k. \tag{2.26}$$

Hence, from (2.24), Davidon-Fletcher-Powell (DFP) formula (Davidon, 1959; Fletcher and Powell, 1963) can be expressed as

$$B_{k+1} = B_k + \frac{s_k s_k^T}{s_k^T y_k} - \frac{(B_k y_k)(B_k y_k)^T}{y_k^T B_k y_k}, \tag{2.27}$$

and Broyden-Fletcher-Goldfarb-Shanno (BFGS) formula (Broyden, 1970*a*,*b*; Fletcher, 1970; Goldfarb, 1970; Shanno, 1970) is expressed as

$$B_{k+1} = B_k + \left(1 + \frac{y_k^T B_k y_k}{y_k^T s_k}\right) \frac{s_k s_k^T}{s_k^T y_k} - \frac{(B_k y_k s_k^T) + (B_k y_k s_k^T)^T}{y_k^T s_k}. \tag{2.28}$$

Formulas (2.27) and (2.28) are from *Huang's family of updates* (Huang, 1970), which is a family of rank 2 updates. When the initial approximation B_k is positive definite and symmetric, the Rank 2 formula ensures that B_{k+1} to be symmetry and positive definite matrix. It is more robust than the rank 1 update formula in minimizing general nonlinear functions (Rao, 2009).

In DFP and BFGS methods, B_k remains positive definiteness if the optimal step length α_k^* are found accurately. However, if the α_k^* are not found accurately, the matrix B_k might become indefinite or even singular. As a result, the alternative is to regularly reset B_k as the identity matrix, I. According to numerical experience, the BFGS technique is less impacted by inaccuracies in determining α^* than the DFP method (Rao, 2009), and it shows superlinear convergence around x^* (Broyden et al., 1973).

2.2 Spectral Gradient Method

Previous researchers have developed some standard tools such as steepest descent (Cauchy, 1847), conjugate gradient (Fletcher and Reeves, 1964; Hestenes and Stiefel, 1952; Polyak, 1969) and quasi-Newton (DFP and BFGS) (Broyden, 1967; Davidon, 1959; Fletcher and Powell, 1963) which are popular in solving unconstrained optimisation problems. Steepest descent is simple but converges slowly with a "zigzags" form towards the minimum point. The conjugate gradient method is the most often used iterative approach for solving sparse systems of linear equations but converges much slower than the quasi-Newton method.

Quasi-newton methods use an approximation to the inverse of Hessian, B_k in place of the true inverse of Hessian, H_k^{-1}. The approximation, B_k must be positive definite and consists of the curvature information. In constructing this approximation, it requires an $O(n^2)$ storage memory which is costly and impractical for large-scale optimisation problems.

A pioneering paper by Barzilai and Borwein (1988) proposed a gradient method that the search direction, $d_k = -g_k$ and the steplength, α_k is chosen with a nonstandard strategy which is given as

$$\alpha_k = \frac{s_{k-1}^T y_{k-1}}{y_{k-1}^T y_{k-1}}, \tag{2.29}$$

or

$$\alpha_k = \frac{s_{k-1}^T s_{k-1}}{s_{k-1}^T y_{k-1}}. \tag{2.30}$$

The Barzilai-Borwein (BB) technique requires just $O(n)$ floating point operations for each iteration, and no line searches are required. This new steplength option needs less computing effort and substantially accelerates the convergence of the gradient method for quadratics. For minimizing general functions, BB method with nonmonotone line search strategy proven to be more efficient than standard conjugate gradient methods based on the numerical experiments showed in Raydan (1997). Hence, the original BB method has been extended to many variations.

The incorporation of the BB method with classical projected gradient strategies (Bertsekas, 1976; Goldstein, 1964; Levitin and Polyak, 1966) establised the spectral gradient method (Birgin et al., 2000, 2001, 2003, 2014). Sim et al. (2019) proposed an extension of the spectral gradient method to approximate the eigenvalues of the actual Hessian matrix. In the proposed spectral gradient method, the full rank matrix in the quasi-Newton method is replaced by the diagonal matrix. Therefore, it reduces the storage from $O(n^2)$ to $O(n)$ and thus the computational time.

In their paper, the step length α_k was chosen based on two line search strategies, namely, monotone and non-monotone. The non-monotone strategy was shown to be outperformed than monotone strategy. Sufficient descent search directions, d_k generated by the spectral gradient algorithm are proved to be independent of the line search methods under standard assumptions. This approach is well-suited to dealing with large-scale problems.

2.3 Sparse Optimisation

Big Data refers to massive datasets which consist of very large numbers of data samples or features. Large dimensional datasets are a difficult challenge for optimisation and machine learning researchers to solve in real-world applications. An underdetermined system is one in which the number of features exceeds the number of observations. Feature selection is involved to select the informative features and remove the irrelevant or redundant ones according to certain criteria (Zhao et al., 2010). This will reduce the computational complexity of the decision model and avoid the model overfitting problem. Thus, it will eventually improve the prediction accuracy, result interpretability, and computational run-time required by the models.

There has been a rise in interest in the general area of sparsity in recent years (Hastie et al., 2015; Wen et al., 2016; Narang et al., 2017; Gale et al., 2019; Sharma et al., 2019; Li et al., 2020; Blanquero et al., 2020). Finding sparse solutions to underdetermined linear systems has become a hot subject in the fields of statistics, compressive sensing, image processing, and machine learning in recent years (Deng et al., 2013;

Le Thi et al., 2015). For instance, in regression analysis and principal component analysis, a subset of informative variables is frequently required (Arthanari and Dodge, 1981; Bertsimas et al., 2016; Zou et al., 2006); Compressed sensing strives for a sparse representation of image or signal data (e.g., (Bruckstein et al., 2009; Donoho, 2006*a*)); A limited amount of invested assets in a portfolio is required for fund management (e.g., (Brodie et al., 2009; Takeda et al., 2013)); In bioinformatics, identifying relevant gene fragments is critical (e.g., (Shevade and Keerthi, 2003)).

Image and signal processing problems are generally stated as

$$Ax + \epsilon = b, \tag{2.31}$$

where A stands for the non-linear or linear operator, x stands for the observation data, while ϵ stands for the observation error or noise. It is difficult to solve problem (2.31) since it is frequently ill-posed and leads to an unknown error. We must impose certain constraints on the solution space, like the signals' prior sparsity for solving this ill-posed problem (Sun et al., 2020).

The model for a sparse optimisation problem is as follows:

$$\min \varphi(x) = \phi(x) + f(x), \tag{2.32}$$

where f is a smooth function and ϕ is possibly a nonsmooth function to recover sparse solutions of underdetermined linear systems. Smooth function is differentiable and its gradient, $\triangledown f$ is Lipschitz-continuous. A sparsity-inducing regularization function, ϕ is involved to find the sparse solution, x^*. It could be the nonsmooth function such as l_0-norm or l_1-norm. It is a nondifferentiable function, which prohibits the use of standard optimisation algorithms like nonlinear conjugate gradient, steepest descent, or quasi-Newton methods, which need the derivative of the function (Nocedal and Wright, 2006).

2.3.1 Current Trends for Solving Sparse Optimisation

During the last two decades, sparse optimisation becomes an active research topic. The original problem involves l_0-norm to promote the sparsity in the solution. The resulting optimisation problem is considered to be difficult due to the discontinuity and nonconvexity of the l_0-norm (Natarajan, 1995). To address such a non-convex optimisation issue, deterministic global optimisation methods (see, e.g., (Horst and Tuy, 2013)) can be used. However, except in small-scale cases, thorough application is impracticable since ensuring global optimality is typically prohibitively time-consuming. Local search methods that rely on approximation or relaxation via tractable convex optimisation are frequently used (Candes et al., 2006; Donoho, 2006*b*,*c*). The nonconvex problem can be approximated by the convex function that has similar properties.

Because l_1-norm would be a tight convex relaxation of l_0-norm, the l_1-norm convex approximation is a popular method to replace $\|x\|_0$ with the $\|x\|_1 = \sum_{i=1}^{n} |x^{(i)}|$ (Candes, 2008). The Least Absolute Shrinkage and Selection Operator (LASSO) is the l_1 regularization technique introduced by Tibshirani (1996) in the context of linear regression. Under proper assumptions, it was shown in Gribonval and Nielsen (2003) that the l_0-regularizer problem over a polyhedral set may be solved by getting the solution from the l_1-regularizer problem. Because of its precise recovery feature under certain conditions, the l_1-relaxation has a huge following in compressed sensing (Candes and Tao, 2005). When the operator A is a sensing matrix, the s-sparse signal x may be retrieved using the l_1 model if the operator A is assumed to have specific properties, like the restricted isometry property (RIP) (Candes, 2008). This l_1 model has been widely utilised in a variety of applications, including communications (Berger et al., 2010), radar systems (Patel et al., 2010; Yang et al., 2012), magnetic resonant imaging (MRI) (Lustig et al., 2007) and computed tomography(CT) (Chen et al., 2013).

However, the LASSO penalty is biased and inconsistent for variable selection in certain cases (Zou, 2006). l_1-regularizer can be a loose relaxation of the l_0-norm and does not always provide the true relevant variables (Candes et al., 2008). Because l_1-norm regularization penalizes the amplitude uniformly, it has a tendency to underestimate high-amplitude components of x, whereas l_0-norm penalises all nonzero entries equally. This might lead to failures in reconstruction with the least measurements (Chartrand and Staneva, 2008; Candes et al., 2008), resulting in unappealing blocky images (Sun and Tao, 2014*a*,*b*). This indicates that convex relaxation may give worse results than the results from the original nonconvex problem. The l_1-norm is well-known for not providing performance comparable to the l_0-norm when sparsity is encouraged. Improved approximations of the l_0-norm and matrix rank lead to better outcomes, according to several theoretical and practical studies in compressive sensing and low-rank matrix recovery (Sun et al., 2020).

Recently, researchers have investigated on the nonconvex continuous approaches to replace l_1-regularization. The non-convex l_0-norm based regularization outperforms the convex l_1-norm in the areas of image restoration (Bao et al., 2016; Zhang et al., 2014; Dong and Zhang, 2013; Zhang et al., 2013), bioluminescence (Zhang et al., 2012), CT (Sun and Tao, 2014*a*,*b*), and MRI reconstruction (Trzasko et al., 2007; Trzasko and Manduca, 2008). The l_0-norm term is approximated by sparsity-inducing penalty function which is the nonconvex continuous function, such as l_p-norm with $p < 0$ (Rao and Kreutz-Delgado, 1999) and $0 < p < 1$ (Fu, 1998), exponential concave function (Bradley and Mangasarian, 1998), Logarithmic function (Weston et al., 2003), Capped-l_1 (Peleg and Meir, 2008), and Smoothly Clipped Absolute Deviation (SCAD) (Fan and Li, 2001). Some algorithms have been developed based on these approximations for solving sparse optimisation problems. For instances, reweighted-l_2 algorithms (Zou and Li, 2008), reweighted-l_1 algorithms (Candes et al., 2008), Two-stage l_1 (Zhang, 2009), Adaptive Lasso (Zou, 2006), Local Linear Approximation (LLA) (Zou and Li, 2008), DCA algorithm (Difference of Convex functions Algorithm) (Neumann et al., 2005; Collobert et al., 2006; Le Thi et al., 2008; Gasso et al., 2009; Le Thi and Ouchani, 2009; Chen et al., 2010; Le et al., 2013; Le Thi et al., 2013; Ong and An, 2013; Le Thi and Nguyen, 2013; Guan and Gray, 2013), and Successive Linear Approximation algorithm (SLA) (Bradley and Mangasarian, 1998).

Besides that, the nonconvex reformulation method involves reformulating the l_0-regularized issue into a continuous nonconvex problem. The l_0-norm sparse optimisation problem for feature selection is recast as a linear program with equilibrium constraint (LPEC) (Mangasarian, 1996) in Support Vector Machine (SVM). However, it is not suitable for solving large-dimensional datasets. Meanwhile, the l_0-norm problem may be converted into an equivalent difference of two convex functions using an exact penalty approach (Thiao et al., 2008;

Dinh and Le Thi, 2014). The proximal gradient algorithm and the difference of convex algorithm (DCA) (Tono et al., 2017; Gotoh et al., 2018) are then used to solve the subproblems. This method is also used to solve the Sparse Eigenvalue issue, where the constraint is the l_0-norm (Thiao and Tao, 2010). The problem is written as

$$\max\left\{x^T A x : x^T x = 1, \|x\|_0 \leq k\right\}, \tag{2.33}$$

where $A \in \mathbb{R}^{n\times n}$ and k is just a number.

Some approaches, including the iterative hard thresholding (IHT) algorithm (Blumensath and Davies, 2008, 2009) and single best replacement (SBR) algorithm (Soussen et al., 2011), directly tackle the original l_0-norm problem. Besides that, some further development for the IHT such as proximal IHT (PIHT) (Lu, 2014), accelerated IHT (AIHT) (Blumensath, 2012), accelerated proximal IHT (APIHT) (Zhang and Zhang, 2017), and extrapolated proximal IHT (EPIHT) (Bao et al., 2016) have been proposed. The l_0-norm problem is not approximated by these techniques. There are some reasons that caused nonconvex optimisation becomes more and more popular:

- Convex relaxations may give unsatisfactory results or poorer solutions than the original nonconvex problem.
- Convex relaxations might produce a larger optimisation problem concerning the original nonconvex problem (Luo et al., 2010; Candes et al., 2015; Ling and Strohmer, 2015). This may be prohibitive in terms of computational power and memory storage.
- Sometimes, convex relaxations become impossible. For example, nonlinear mapping in optimisation.

Convex regularisation techniques are often simpler to solve, but the l_0-regularizer problem is difficult to solve. Nonconvex approximations yield better sparsity than convex relaxations. However, they are difficult to solve and cannot guarantee the local minimum obtained are global. To obtain a "good" local minimum in nonconvex optimisation, the problem needs to be initialized carefully.

Some issues in the existing approximation approaches have not yet been studied (Le Thi et al., 2015). The key problems to investigate in a large-scale problem are how to appropriately approximate the $l0$-norm in the optimisation problem and which approach should be utilised to solve the resultant problem. Researchers in optimisation and machine learning are continually challenged to come up with new models and approaches for sparse optimisation problems.

2.4 l_0-Norm Sparse Optimisation

In our research, we will solve for l_0-norm sparse optimisation problems. The l_0-norm is a key notion for modelling data sparsity, and it is vital in optimisation issues where representative variables must be chosen (Le Thi et al., 2015). It enables us to rebuild high-dimensional data using only a few samples (Deng et al., 2013). An optimisation problem incorporating the zero-norm in the objective function or constraints is known as l_0-norm sparse optimisation:

$$\min \|x\|_0 + f(x), \tag{2.34}$$

where the loss function, $f(x)$ is represented by the data fidelity term associated with (2.31). For instance, $f(x)$ can be the least-absolute (LA) loss function, $\|A(x) - b\|_1$ or the least square (LS) loss function,

$\|A(x) - b\|_2^2$.

The norm of a mathematical object is referred to a quantity that measures the length or size of the vector in some sense (Nocedal and Wright, 2006). The zero-norm on $\mathbb{R}^n$, often referred as the l_0-norm or $\|.\|_0$, is defined as follows:

$$\|x\|_0 := |\{i = 1, ..., n : x_i \neq 0\}|. \tag{2.35}$$

It denotes the number of nonzero elements in the vector, x. In some literature, some people disagree l_0-norm as a proper norm because it does not satisfy the property of a norm. One gets $\|\lambda x\|_0 = \|x\|_0$ for every $x \in \mathbb{R}^n$ and $\lambda \neq 0$, indicating that it is not absolutely homogeneous and hence, it is not a norm (Le Thi et al., 2015). However, in this research, we will adopt it as a norm (Xu and Zhao, 2020).

In this research, we propose to solve sparse optimisation problems involving l_0-norm and its constraint by minimizing directly with the proposed method, spectral proximal method (SPM). This method incorporates the spectral gradient method and the proximal method. This research idea is derived from the proximal gradient method.

2.5 Proximal Gradient Method

In the context of convex optimisation in Hilbert spaces, Martinet (1970) proposed the proximal method as a regularization method. Recently, the proximal algorithm was extended from convex optimisation to non-convex optimisation (Hare and Sagastizábal, 2009). Steepest descent method, conjugate gradient method, and Newton's method are standard tools for solving unconstrained smooth minimization problems of modest size, while proximal algorithms can be viewed as an analogous tool for handling non-smooth, constrained, large-scale, or distributed problems. Closed-form solutions are common for proximal mappings of the corresponding function. They can be effectively computed at a low-cost (Antonello et al., 2018) and are especially well-suited to large or high-dimensional dataset problems. Classical techniques' basis operations are low-level, consisting of linear algebra operations and the computation of gradients and Hessians, whereas proximal algorithms' base operation is evaluating a function's proximal operator (Parikh and Boyd, 2014).

First-order methods are algorithms that depend only on the function evaluation and the gradient evaluation. Due to large dimensional dataset problems arising in compressive sensing, first-order methods have become more popular for sparse recovery. Various gradient-based algorithms have been proposed in the area of sparse recovery (Becker et al., 2011; Figueiredo et al., 2007; Bioucas-Dias and Figueiredo, 2007; Hale et al., 2008; Van Den Berg and Friedlander, 2009; Yin et al., 2008; Figueiredo and Nowak, 2003; Wright et al., 2009; Fukushima and Mine, 1981). One of the most basic nontrivial proximal algorithms is the proximal gradient method or generalised gradient technique (Polson et al., 2015). Sometimes, it is also known as the forward-backward splitting method (Combettes and Wajs, 2005; Fukushima and Mine, 1981; Yamamoto et al., 2012). This algorithm can solve nonsmooth convex and nonconvex problems. A non-convex and non-smooth function is a hard problem in optimisation, it usually solves with tractable convex optimisation based on a regularization or relaxation. However, sparse optimisation which involved convex relaxations increase dimensionality (Luo et al., 2010) and may cause the problems to be computationally intractable. While proximal gradient methods treat the original nonconvex problem directly. It is independent of the smoothness and the convexity of the problem.

In problem (2.32), f is a smooth function, while ϕ is a possibly nonsmooth function. ϕ is the term to encourage the sparsity, such as l_0-norm or l_1-norm. The overall function φ can be minimized by the proximal gradient step. The proximal operator is defined in Park et al. (2020) as:

$$x_{k+1} = \operatorname{prox}_{\phi,P_k}(v_{k+1}) := \operatorname*{arg\,min}_{x\in\mathbb{R}^n} \left\{ \phi(x) + \frac{\lambda}{2}\|v_{k+1} - x\|^2_{P_k} \right\}, \qquad (2.36)$$

where λ is a positive scalar, $\|z\|^2_P = z^T P z$ for any $z \in \mathbb{R}^n$, $P \in \mathbb{R}^{n\times n}$ be any positive definite matrix. The scaled proximal mapping of ϕ respect to the metric P is represented as $\operatorname{prox}_{\phi,P_k}$.

Two basic steps, *Gradient step* (or *forward*) and *Proximal operator step* (or *backward*) are involved iteratively until it finds the approximate solution. The iterate x_k is forced to move towards the minimum of f by the gradient step and to be closer to the minimum of ϕ by the proximal step. Alternation of these two steps will ultimately lead the sum of these two functions to become minimum.

Various types of proximal gradient methods have been proposed. Recently, proximal Newton method and proximal quasi-Newton methods (Becker and Fadili, 2012; Lee et al., 2014; Karimi and Vavasis, 2017) become an active research topic due to these methods incorporate more information about the function without compromising the efficiency of the algorithms. The proximal operator step for the proximal quasi-Newton is defined as

$$x_{k+1} = \operatorname{prox}_{\phi,P_k}(v_{k+1}) = \operatorname{prox}_{\phi,P_k}(x_k - \alpha_k(B_k)\triangledown f(x_k)). \tag{2.37}$$

From the ideas of the proximal gradient method, we propose a new method, known as spectral proximal method (SPM) for solving the sparse optimisation problems. In the next section, the algorithm and the convergence analysis of SPM will be discussed. Besides that, the scope of the numerical experiments will also be given.

CHAPTER 3

METHODOLOGY

3.1 Background Concept

The sparse optimisation problem (3.1) usually arises in the areas of science and engineering. l_0-norm sparse optimisation is an optimisation problem with l_0-norm in the objective or constraints. The focus of this research is to develop an effective and general-purpose method for tackling sparse optimisation problems.

The l_0-norm sparse optimisation model in our problem can be written as:

$$\min \|x\|_0 \tag{3.1a}$$

$$\text{subject to } Ax = b, \tag{3.1b}$$

where the constraint (3.1b) is usually an underdetermined system. The objective function, l_0-norm is the sparsity inducing regularization function. It denotes the number of nonzero components in x. Hence, the solution x^* should be sparse, which consists the minimum number of nonzero components in the solution, x.

Some issues occurred when solving l_0-norm sparse optimisation since the l_0-norm function is non-smooth and non-convex. The common optimisation methods for solving unconstrained smooth optimisation problems, including gradient descent, nonlinear conjugate gradient, and quasi-Newton methods, cannot be used (Nocedal and Wright, 2006).

The data fidelity function, $Ax = b$ is solved by using a least square method. Direct methods such as QR decomposition or Cholesky factorization are infeasible to solve when involving large linear system. Iterative techniques like steepest descent, quasi-Newton, and spectral gradient approaches are effective for addressing optimisation problems with high-dimensional data sets. However, the quasi-Newton method which stores a full rank matrix with $O(n^2)$ storage requires a larger computation time per iteration compared to the spectral gradient method.

The l_0-norm sparse optimisation problems can be solved by the ideas discussed in section (2.5). Previously, the proximal quasi-Newton method has been proposed by Chen and Fukushima (1999). However, the proximal quasi-Newton is demanding in memory due to the storage of the Hessian matrix. In the literature, no research has been done on the spectral proximal method. To fill in the gap, spectral proximal method (SPM) is proposed in this research by applying the spectral gradient method and proximal mapping alternatively to solve the problems.

3.2 Derivation and Algorithm of Spectral Gradient Method

To derive an updating scheme for the approximation to the eigenvalues of the Hessian matrix, H_k in the spectral gradient method, for any positive definite matrix H, the log-determinant norm is given as:

$$\phi(H) = \mathrm{tr}(H) - \ln(\det(H)). \tag{3.2}$$

It is required to be minimized while satisfying the weak secant equation which consists of the curvature information:

$$\min \operatorname{tr}(H_k) - \ln(\det(H_k)) \tag{3.3a}$$

$$\text{subject to } s_k^T H_k s_k = s_k^T y_k. \tag{3.3b}$$

Let $H_k = \operatorname{diag}(H_k^{(1)}, ..., H_k^{(n)})$, $s_k = x_k - x_{k-1}$, and $y_k = g_k - g_{k-1}$. Then, the minimization problem (3.3) becomes

$$\min \left(\sum_{i=1}^{n} H_k^{(i)}\right) - \ln\left(\prod_{i=1}^{n} H_k^{(i)}\right) \tag{3.4a}$$

$$\text{subject to } \left(\sum_{i=1}^{n} (s_k^{(i)})^2 H_k^{(i)}\right) - s_k^T y_k = 0. \tag{3.4b}$$

By Lagrangian method:

$$L(\alpha,\omega) = \left(\sum_{i=1}^{n} H_k^{(i)}\right) - \ln\left(\prod_{i=1}^{n} H_k^{(i)}\right) + \omega\left[\left(\sum_{i=1}^{n} (s_k^{(i)})^2 H_k^{(i)}\right) - s_k^T y_k\right], \tag{3.5}$$

where the constraint (3.4b) is connected by a Lagrange multiplier, ω. The partial derivations are set to zero after partially differentiating (3.5) with regard to each $H_k^{(i)}$, which yields:

$$H_k^{(i)} = \frac{1}{1 + \omega (s_k^{(i)})^2}, \; i = 1,\, 2,\, \ldots,\, n. \tag{3.6}$$

By substituting (3.6) into the constraint (3.3b) ,

$$F(\omega) = \sum_{i=1}^{n} \left(\frac{(s_k^{(i)})^2}{1 + \omega (s_k^{(i)})^2}\right) - s_k^T y_k. \tag{3.7}$$

Then, apply just one Newton-Raphson iteration with $\overline{\omega} = 0$. As a result, the equation (3.7) has a unique positive solution when $s_k^T s_k > s_k^T y_k$ and the Lagrange multiplier, ω_k may be approximated by

$$\begin{aligned}\omega_k &= \overline{\omega} - \frac{F(\overline{\omega})}{F'(\overline{\omega})} \\ &= \frac{s_k^T s_k - s_k^T y_k}{\sum_{i=1}^{n} (s_k^{(i)})^4}.\end{aligned} \tag{3.8}$$

When $\frac{s_k^T y_k}{s_k^T s_k} < 1$, we have $H_{k+1}^{(i)} > 0$, for all $i = 1, \ldots, n$ since $\omega > 0$. The ratio $\frac{s_k^T y_k}{s_k^T s_k}$ is exactly the Oren-Luenberger scaling (see (Luenberger and Ye, 1984)). Most quasi-Newton methods involve it to dampen the steepest descent direction. Thus, to ensure positive definiteness, the updating formula for H_k is as follows when these two events are combined:

$$H_k = \begin{cases} \text{diag}(H_k^{(i)}, \ldots, H_k^{(n)}), & \text{if } s_k^T s_k > s_k^T y_k \\ \frac{s_k^T y_k}{s_k^T s_k} I, & \text{otherwise,} \end{cases} \tag{3.9}$$

where $H_k^{(i)} = \frac{1}{1+\omega_k (s_k^{(i)})^2}$ and $\omega_k = \frac{s_k^T s_k - s_k^T y_k}{\sum_{i=1}^{n} (s_k^{(i)})^4}$.

Spectral gradient method retains a low per-step computation cost $O(n)$ while better satisfying the secant condition.

The algorithm is shown as below:

Algorithm 2: Spectral gradient method

```
1. Initialize parameters.
   Set $k = 0$; given an initial guessing point $x_0$, and a convergence
   tolerance $\epsilon$. $H_0^{-1}$ is an $n \times n$ identity matrix.
2. Compute the search direction $d_k$ as
       $d_k = -H_k^{-1} g_k,$        (3.10)
   where $H_k$ for $k \geq 1$ is defined by (3.9).
3. Compute the step length $\alpha_k$ based on Algorithm 1 in the direction
   $d_k$ and $x_{k+1} = x_k + \alpha_k d_k$.
4. Check stopping criterion.
   If $\|g_{k+1}\| \leq \epsilon$, stop. Otherwise, set $k = k + 1$ and go to Step 2.
```

The step length α_k can be computed by either monotone or non-monotone line search strategy.

3.3 Spectral Proximal Method

In order to solve the l_0-norm sparse optimisation problem, the spectral gradient method and proximal method have been reviewed. The constrained sparse optimisation problem (3.1) can be transformed into the Lagrangian form and formulated as (2.32):

$$\min \|x\|_0 + \frac{\lambda}{2}\|Ax - b\|_2^2. \tag{3.11}$$

The function (3.11) is minimized in term of x to obtain an estimate or optimal solution, x^*. $\|x\|_0$ is a sparsity inducing regularization function and $\frac{\lambda}{2}\|Ax - b\|_2^2$ is the data fidelity function. Instead of solving $Ax = b$

directly, we considered to minimize the residue or error between the model Ax and output b to find the least square solution. The l_2-norm regularization acts as the smoothing term.

The coefficient $\lambda \geq 0$ is a scalar that needs to be tuned properly. It balances the relative weight of the regularization function, $\|x\|_0$, and the data fidelity function, $\frac{\lambda}{2}\|Ax - b\|_2^2$. This prevents the data fidelity term from being too small, resulting in overfitting. Besides that, if $\lambda \to 0$, the regularization term dominates the cost function, resulting in the most sparse solution. In our numerical experiments, we set $\lambda = 1$.

The non-smooth function, l_0-norm in the problem (3.11) can be solved by using the proximal method. The solution to a non-convex and non-smooth function can be found using this method (Hare and Sagastizábal, 2009). It is well-suited and is very generally applicable for problems with large or high-dimensional data sets (Parikh and Boyd, 2014). On the other hand, the residue can be solved by applying the spectral gradient method. This method is proposed by Sim et al. (2019). From the results shown, the spectral gradient method outperformed a list of conjugate gradient methods since it needs less storage and less computational time.

In this research, the proximal method is modified and integrated with the spectral gradient method. These two methods are combined and applied alternatively to solve the l_0-norm sparse optimisation problems. This proposed method is called spectral proximal method (SPM). In this research, the step length, α_k is obtained by using Armijo condition in the backtracking line search strategy (see Algorithm 1).

The SPM algorithm is as follows:

Algorithm 3: Spectral proximal method

```
1. Initialize parameters.
   Set $k = 0$; give an initial guess for $x_0$, set a convergence tolerance $\epsilon$,
   and a sparsity control parameter $\mu > 0$. $H_0^{-1}$ is an $n \times n$ identity
   matrix.

2. Gradient step.
   Perform spectral gradient method (see Step 2 and Step 3 in
   Algorithm 2) to obtain $v_{k+1} = x_k - \alpha_k (H_k)^{-1} \nabla f(x_k)$.

3. Proximal operator step.
   Evaluate the proximal operator of $\|x\|_0$ at the intermediate point
   $v_{k+1}$. From this step, we get $x_{k+1} = \text{prox}_{\phi, P_k}(v_{k+1})$, where

   $$x_{k+1}^{(i)} = \begin{cases} 0, & \text{if } |v_{k+1}^{(i)}| < \sqrt{2\mu} \\ v_{k+1}^{(i)}, & \text{if } |v_{k+1}^{(i)}| = \sqrt{2\mu} \\ v_{k+1}^{(i)}, & \text{if } |v_{k+1}^{(i)}| > \sqrt{2\mu}. \end{cases} \quad (3.12)$$

   This is in absolute form, we can also change $|v_{k+1}^{(i)}|$ to without
   absolute form, $v_{k+1}^{(i)}$.

4. Check stopping criterion.
   If $\|g_{k+1}\| \le \epsilon$, stop. Otherwise, set $k = k + 1$ and go to Step 2.
```

We will write the proposed method as follows in our research:

Spectral proximal method which its proximal method is checked with $|v_{k+1}^{(i)}|$: Spectral proximal method with absolute;

Spectral proximal method which its proximal method is checked with $v_{k+1}^{(i)}$: Spectral proximal method without absolute.

The convergence properties of the proposed methods have been established based on standard assumptions provided by Byrd and Nocedal (1989). To compare the efficiency of the proposed method with the existing methods, an executable code is developed using Python software. The benchmarks of the comparison are in terms of the number of iterations, the number of function calls, and the computational time.

3.4 Convergence Analysis

$\varphi(x) = \phi(x) + f(x)$ is the objective function for a sparse optimisation problem. It is the product of a smooth function f and a nonsmooth function ϕ with efficient proximal mapping. By using the proximal gradient step, the function φ can be minimized. In Park et al. (2020), the convergence of variable metric proximal gradient method (VM-PG) was investigated.

For penalized non-quadratic problems, the convergence without line search might not be ensured by VM-PG with diagonal metric in (3.9), similar to many other BB techniques. As a result, in Algorithm 3, we employ the line search technique and utilise H_k, which is specified in (3.9), as an starting metric. In this research, $P_k = \frac{1}{\alpha} H_k$.

At x_k, the proximal mapping of ϕ is written as:

$$\begin{aligned}&\operatorname{prox}_{\phi,P_k}(x_k - (P_k)^{-1}\nabla f(x_k)) = \\ &\arg\min_x \left\{ \phi(x) + f(x_k) + \nabla f(x_k)^T (x - x^k) + \frac{1}{2}\|x - x_k\|^2_{P_k} \right\},\end{aligned} \tag{3.13}$$

since f is differentiable and its second order form may be estimated. In this case, $P = \mathrm{Diag}(p)$ (Chouzenoux et al., 2014).

Definition 3.1. *if $\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2$ holds for every $x, y \in \mathbb{R}^n$, then the differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is L-smooth. Moreover, if $\langle \nabla f(x) - \nabla f(y), x - y \rangle \geq m\|x - y\|_2^2$ holds for every $x, y \in \mathbb{R}^n$, f is m-strongly convex.*

G stands for the matrix of the second derivatives of f. The algorithm begins with x_0 as the starting point. Byrd and Nocedal (1989) provides the following assumptions, which we will use in our analysis:

Assumption 1. *i. The objective function f is twice continuously differentiable.*
ii. The level set $U = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$ is convex.
iii. There exist positive constants M_1 and M_2 such that

$$M_1\|z\|^2 \leq z^T G(x) z \leq M_2\|z\|^2, \tag{3.14}$$

for $\forall z \in \mathbb{R}^n$ and $\forall z \in U$. This indicates that there has a unique minimizer $x^* \in U$ for the objective function f.

The smooth part, f will be solved by spectral gradient method. The details of the proofing have been derived in Sim et al. (2019). In the lemma below, it states the boundness of $\|H_k\|$, where H_0 is considered to represent the identity matrix, I.

Lemma 3.1. *Let x_0 be an initial point such that f meets Assumption 1 and $H_0 = I$, where I is the $n \times n$ identity matrix. The sequence $\{\|H_k\|\}$ is then bounded by certain positive constants c_1 and c_2 for H_k defined by (3.9).*

$$c_1 = \min\left\{\frac{1}{1+n(1-M_1)}, M_1\right\} \leq H_{k+1}^{(i)} \leq \max\{1, M_2\} = c_2, \forall k \geq 0. \tag{3.15}$$

Since $P_k = \frac{1}{\alpha} H_k$ in Algorithm 3 and hence, $P_k > 0$.

The convergence analysis for Algorithm 3 is then shown, relying on the theorems presented in (Park et al., 2020). For sufficiently small steplength, $\alpha < \frac{1}{L}$, the classic proximal gradient method is guaranteed to converge when f is L-smooth. Even without knowledge of the Lipschitz, many line searches with backtracking techniques can still guarantee convergence (Boyd et al., 2004*b*; Beck and Teboulle, 2009).

The convergence analysis for the VM-PG algorithm is shown by assuming $P_k > 0$ and f is L-smooth.

Theorem 3.2. *$\varphi(x_k)$ converges to the optimal value φ^* for VM-PG in Algorithm 3, i.e., $\lim_{k\to\infty} \varphi(x_k) := \varphi^*$.*

Theorem 3.3. *With monotonic line search, the VM-PG with diagonal metric in Algorithm 3 satisfies,*

$$\min_{k=1,\,\dots,\,K} \|G_{P_k}(x_k)\|^2_{(P_k)^{-1}} \leq \frac{2(\varphi(x_0) - \varphi^*)}{K}, \tag{3.16}$$

where $G_{P_k}(x_k) \in \nabla f(x_k) + \partial\phi(x_k - (P_k)^{(-1)}\nabla f(x_k))$ and $G_{P_k}(x_k) = 0$ if $0 \in \partial\varphi(x_k)$.

Moreover, if f is m-strongly convex, we have

$$\|x_{k+1} - x^*\|^2_{P_k} \leq (1 - \frac{m}{p_k^{(\max)}})\|x_k - x^*\|^2_{P_k}, \tag{3.17}$$

where $p_k^{(\max)} = \max_i p_k^{(i)}$.

3.5 Scope of the Numerical Experiments

All the experiments in Chapter 4 are executed with an 8th Generation Intel Core i7 processor and 12 GB RAM. The executable code is developed using Python 3.7 software in order to compare the efficiency of the proposed method, spectral proximal method (SPM) with other types of proximal gradient methods: Proximal with steepest descent method (PSD) and proximal with Broyden-Fletcher-Goldfarb-Shann method (PBFGS).

We choose the initial point $x_0 \in \mathbb{R}^n$ as a vector of ones with n components. A is a $m \times n$ real matrix and $b \in \mathbb{R}^m$. The coefficient λ is set as 1. Tolerance, ϵ is set to be 10^{-4} and algorithm is terminated when the stopping criterion is met, $\|g_{k+1}\| \leq \epsilon$. We selected the value of the stopping criterion as a compromise between the demand for fast termination and the desire for an approximate solution. We also established a limit on the amount of iterations. The algorithm is forced to cease running when the number of iterations exceeds its upper limit, and the method is regarded to have failed to converge. For establishing an appropriate steplength, we exclusively utilised a backtracking line search with the Armijo condition (BTA) method in this research. In BTA, we used $c = 0.1$ and $\tau_{lo} = \tau_{hi} = 0.5$. The starting step length is set as 1.

In Section 4.1 and Section 4.2, the random matrices with dimension 7×10, 35×50, 70×100, 140×200 and 350×500 are generated. In every dimension, the condition number is set from 1 to 20. After that, 5 different seeds are generated for each particular dimension and condition number. In total, there are 500 random matrices are generated. These experiments will be tested on two different settings of the actual solution. Elements of the actual solution, x^* in the first and second problems are set within the interval of $[0,1]$ and $[-1,1]$ respectively. Besides that, for both problems, 10% of the elements in x^* are set to 0.

The backtracking line search with Armijo condition will be tested for a maximum of 15 iterations in all methods. The maximum number of iterations to run these experiments is set to be 5000. The parameter μ in the proximal operator of l_0-norm is set as 10^{-2}, 10^{-4} and 10^{-6} for each of the experiment. We assume that the efficiency's value for the failure run will be replaced by the maximum value from the particular efficiency times 60 to show that the solver s has failed to reach the convergence criterion on problem p.

In Section 4.1, the numerical results from 5 different seeds with the same dimension and condition number's random matrices are averaged and illustrated using the profiles of Dolan and Moré (Dolan and Moré, 2002). Performance profile or profiling graph is defined to evaluate and compare the performance measure $m_{p,s} > 0$ of the set of optimisation solvers $s \in \mathcal{S}$ on a set of problems $p \in \mathcal{P}$. The performance measure is obtained from each problem p and solver s. The number of iterations, number of function calls, and the computational time in seconds are the performance measures of interest. To obtain the performance ratio, we compare the performance of running solver s on problem p with the performance from the best solver on the problem p, that is,

$$r_{p,\, s} = \frac{m_{p,\, s}}{\min\{m_{p,\, s} : s \in \mathcal{S}\}}. \tag{3.18}$$

The overall assessment of the performance of the solver can be defined as the cumulative probability of the problems with the performance ratio $r_{p,s}$ within $\tau \in \mathbb{R}$. The performance profile of a solver s is:

$$P(\tau) = \frac{1}{n_p} size\{p \in \mathcal{P} : r_{p,\, s} \leq \tau\}. \tag{3.19}$$

where n_p is the total number of problems. The function $P(\tau)$ is the cumulative distribution for the performance ratio $r_{p,\, s}$. It is better to use solvers with high $P(\tau)$ values. The value of $P(1)$ is the probability of the particular solver will outperform the other solvers. The graphs for the performance profiles are plotted with base-10 logarithm scaling on the x-axis.

In Section 4.2, the relation between the efficiency and condition number is investigated. We compared the performance of the proposed method with the other methods on different condition numbers (from 1 to 20). This is to investigate how the condition number influences the performance of these methods. The performance metric for each of the efficiency is calculated, which is the probability of the particular method having the best performance among 25 sets of data for each of the condition numbers. The performance metric is the dependent variable and the condition number is the independent variable. The relations are illustrated in the graphs.

Section 4.3 is the application example using MNIST real-world datasets, which is on handwritten digits recognition (LeCun et al., 1998). It is commonly used in image processing and also machine learning fields (Platt, 1999). In our experiment, we used the least square (LR) loss, $\|Ax - b\|_2^2$ to estimate all labels ('0' - '9') and k number of images are randomly chosen from the training sets.

We compare our proposed method with other methods for these highly ill-conditioned data sets. The efficiencies we compared are based on the norm of gradient, residue, the cumulative computational time needed in seconds, and the cumulative number of function calls. To compare the number of function calls, the maximum number of iterations for the BTA algorithm in all methods is unified to 50. The maximum number of iterations is set at 10000 and μ is set at 10^{-6}. Other parameters remain the same as the previous experiments.

The results obtained are averaged among labels ('0'-'9'). To illustrate the trend and the movement of the efficiency for every increased iteration, the graphs are plotted with the number of iterations on the x-axis and the efficiency on the y-axis with base-10 logarithm scaling. Besides that, we also investigate the sparsity and the stability of the sparsity by using different proximal gradient methods in the last part of the next section.

CHAPTER 4

NUMERICAL RESULTS AND DISCUSSIONS

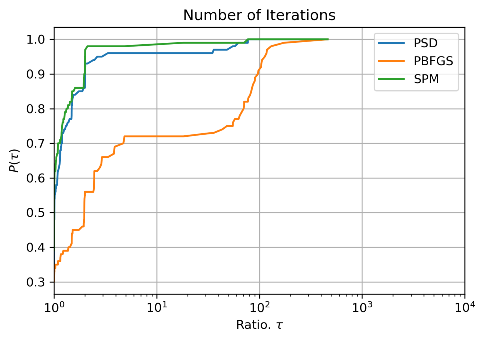
The numerical experiments are conducted based on the scope of numerical experiments described in Section 3.5. Simulated data in Section 4.1 and Section 4.2 are generated from the random matrices with condition numbers 1 until 20, while data in Section 4.3 comes from real-life MNIST datasets. In this chapter, we will compare the efficiency of the proposed method, SPM method with the PSD and the PBFGS methods. The graphs are plotted to illustrate the comparisons between the efficiency of these methods. Besides that, the sparsity of the solution for every experiment is shown in the tables. Sparsity is calculated as the percentage of zero elements contained in the approximate solution, x^*. For all the experiments, we used the BTA algorithm as our line search strategy.

4.1 Profiling and Benchmarking for Spectral Proximal Method

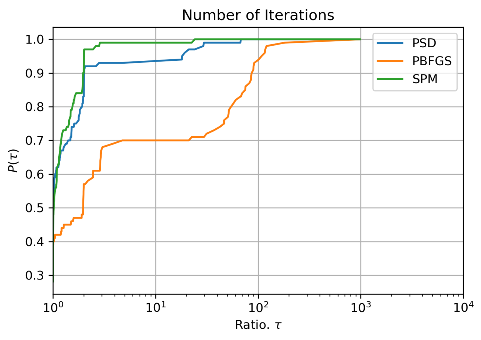
The profiling graphs of the proposed method and existing methods are shown below:

1. Test 1: when $x^* \in [0,1]$, and

   (a) $\mu = 10^{-2}$

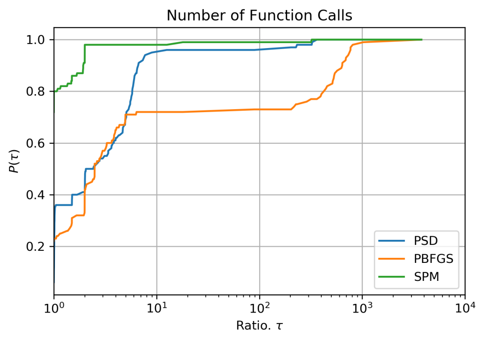


(a) Without absolute

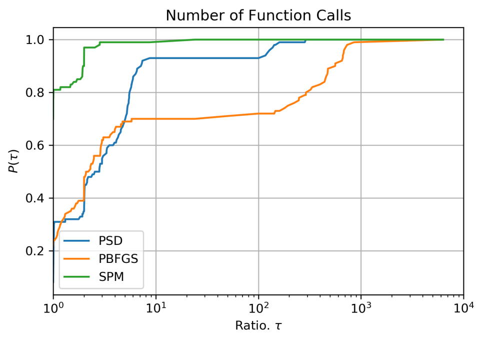


(b) With absolute

Figure 4.1: Profiling graph for number of iterations (test 1, $\mu = 10^{-2}$)

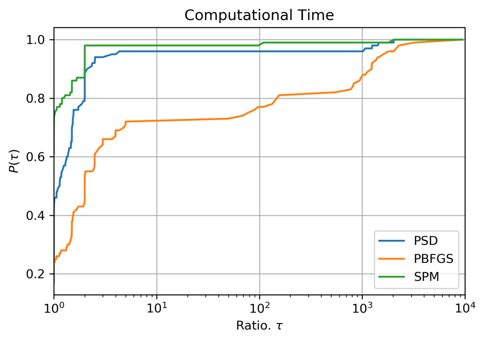


(a) Without absolute

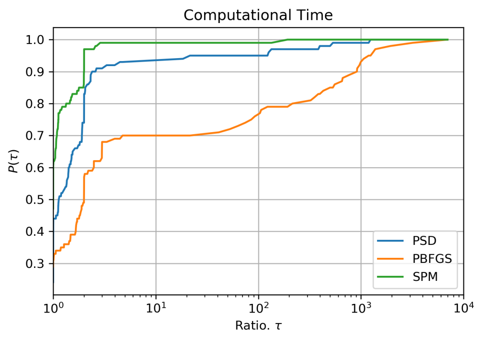


(b) With absolute

Figure 4.2: Profiling graph for number of function calls (test 1, $\mu = 10^{-2}$)



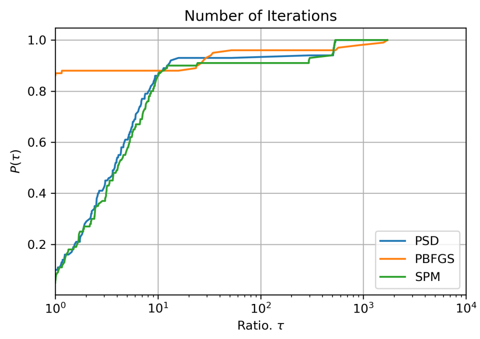
(a) Without absolute



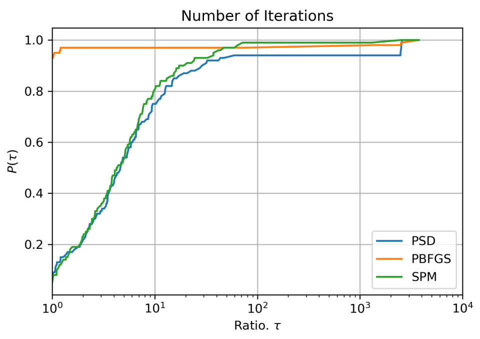
(b) With absolute

Figure 4.3: Profiling graph for computational time in seconds (test 1, $\mu = 10^{-2}$)

(b) $\mu = 10^{-4}$

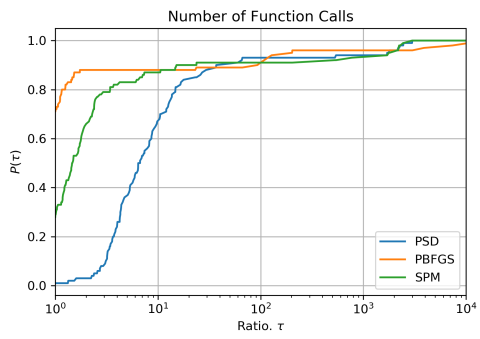


(a) Without absolute

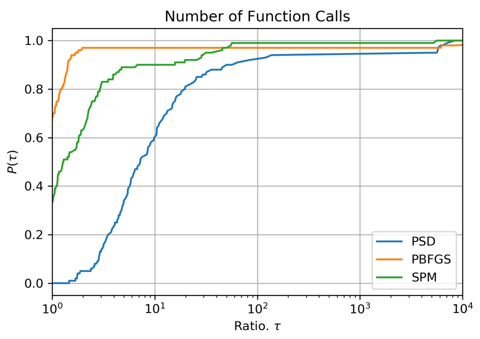


(b) With absolute

Figure 4.4: Profiling graph for number of iterations (test 1, $\mu = 10^{-4}$)

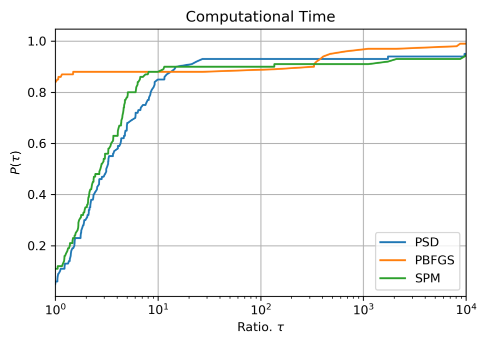


(a) Without absolute

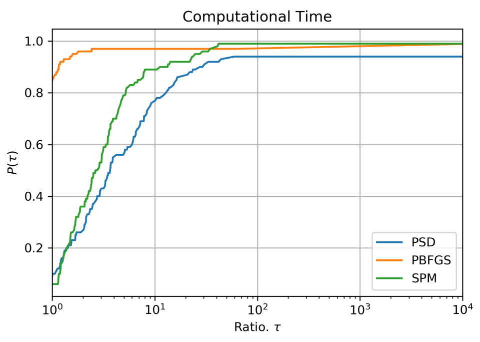


(b) With absolute

Figure 4.5: Profiling graph for number of function calls (test 1, $\mu = 10^{-4}$)



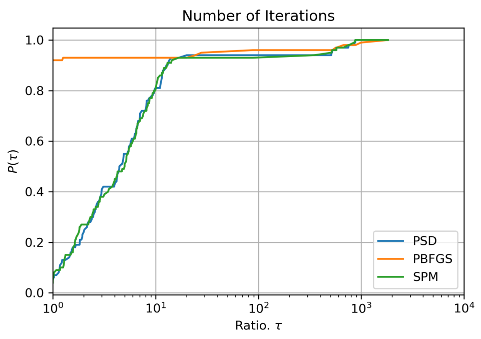
(a) Without absolute



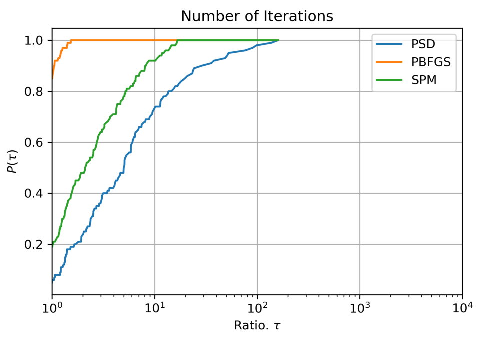
(b) With absolute

Figure 4.6: Profiling graph for computational time in seconds (test 1, $\mu = 10^{-4}$)

(c) $\mu = 10^{-6}$

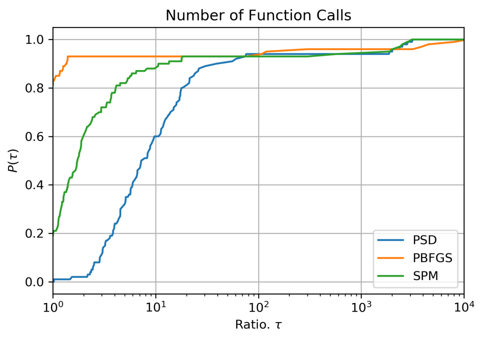


(a) Without absolute

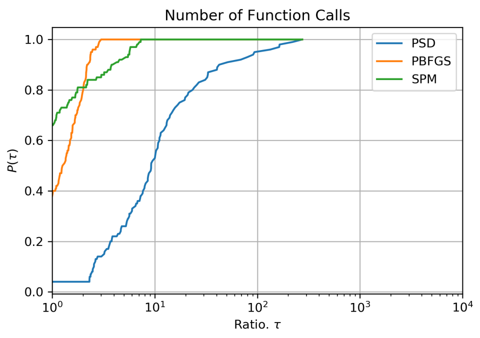


(b) With absolute

Figure 4.7: Profiling graph for number of iterations (test 1, $\mu = 10^{-6}$)

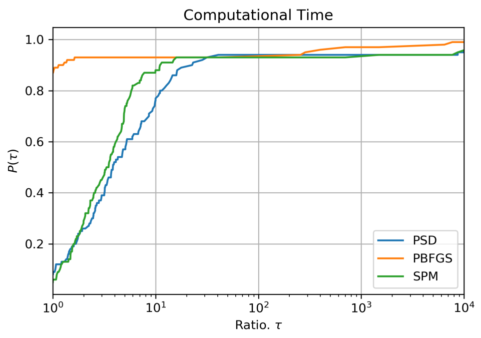


(a) Without absolute

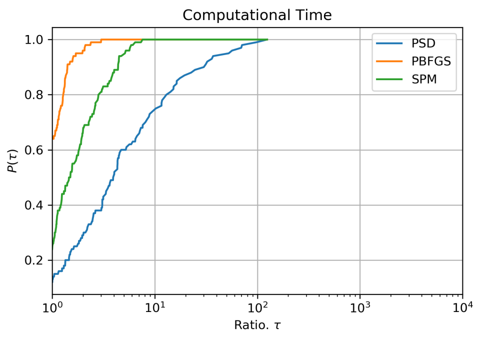


(b) With absolute

Figure 4.8: Profiling graph for number of function calls (test 1, $\mu = 10^{-6}$)



(a) Without absolute



(b) With absolute

Figure 4.9: Profiling graph for computational time in seconds (test 1, $\mu = 10^{-6}$)

In test 1, all components of the actual solution are within the interval [0,1]. It is worth noticing that, in the case with absolute, the PBFGS method gradually outperforms the SPM method when μ is getting smaller. When the sparsity of the solution is increased by setting the tuning parameter $\mu = 10^{-2}$, the results show that SPM method outperforms the others. Tables below show the average sparsity of the solution (%):

Table 4.1: Average sparsity of the solution (test 1, without absolute)

	PBFGS	PSD	SPM
$\mu = 10^{-2}$	16.73	13.60	13.25
$\mu = 10^{-4}$	4.92	6.68	6.68
$\mu = 10^{-6}$	3.74	5.52	5.55

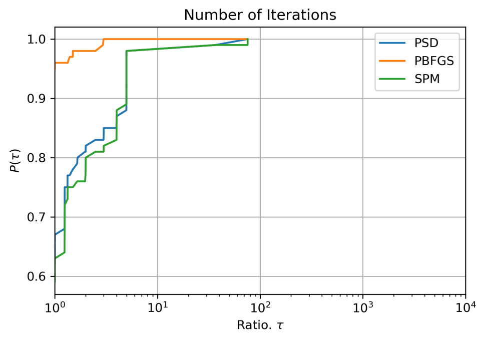
Table 4.2: Average sparsity of the solution (test 1, with absolute)

	PBFGS	PSD	SPM
$\mu = 10^{-2}$	14.80	12.05	11.83
$\mu = 10^{-4}$	1.43	2.64	2.33
$\mu = 10^{-6}$	0.05	0.17	0.09

The results show that the sparsities of the solution from all three methods are comparable. Overall, by considering the trade-off between the efficiency of the method and the solution's sparsity (%), SPM method has better performance in obtaining a desirable sparse solution.

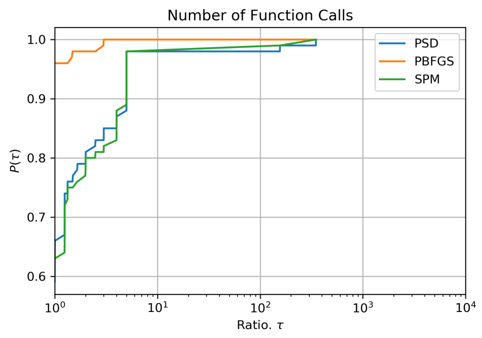
2. Test 2: when $x^* \in [-1,1]$, and

   (a) $\mu = 10^{-2}$



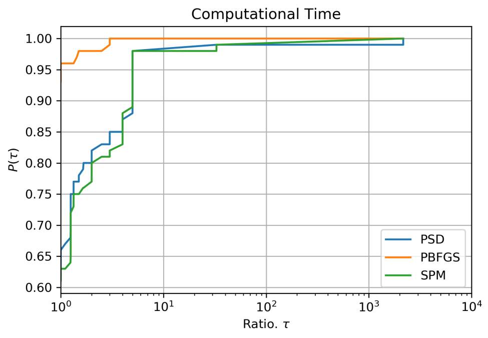
(a) With absolute

Figure 4.10: Profiling graph for number of iterations (test 2, $\mu = 10^{-2}$)



(a) With absolute

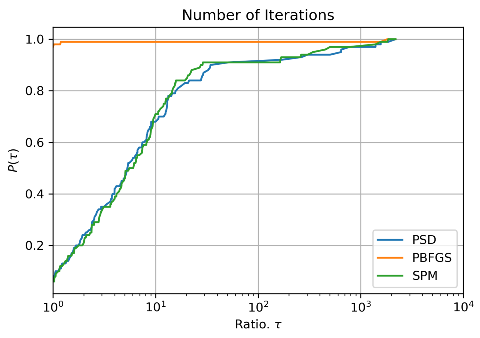
Figure 4.11: Profiling graph for number of function calls (test 2, $\mu = 10^{-2}$)



(a) With absolute

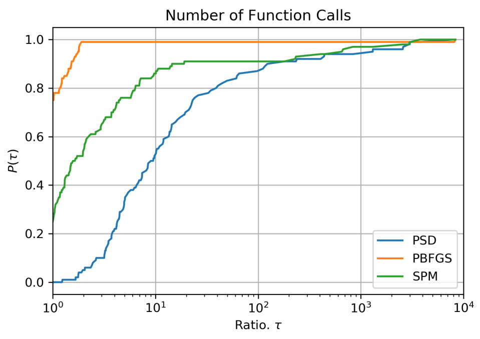
Figure 4.12: Profiling graph for computational time in seconds (test 2, $\mu = 10^{-2}$)

(b) $\mu = 10^{-4}$



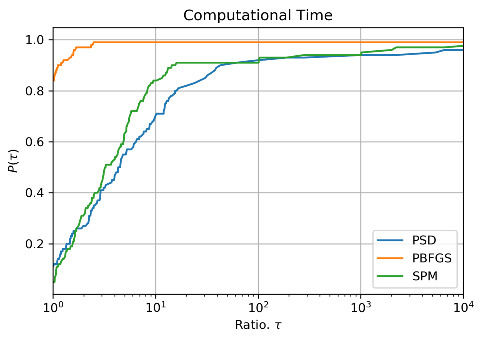
(a) With absolute

Figure 4.13: Profiling graph for number of iterations (test 2, $\mu = 10^{-4}$)



(a) With absolute

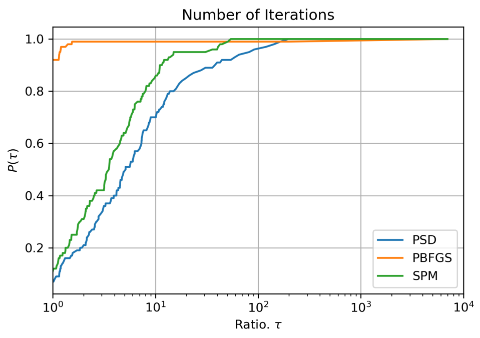
Figure 4.14: Profiling graph for number of function calls (test 2, $\mu = 10^{-4}$)



(a) With absolute

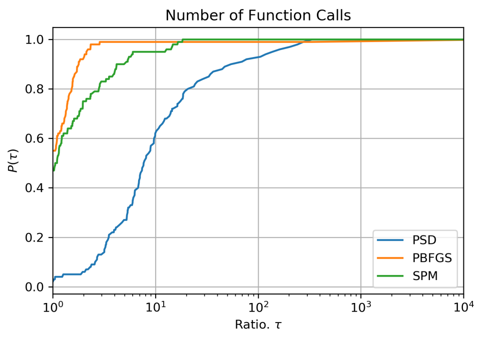
Figure 4.15: Profiling graph for computational time in seconds (test 2, $\mu = 10^{-4}$)

(c) $\mu = 10^{-6}$



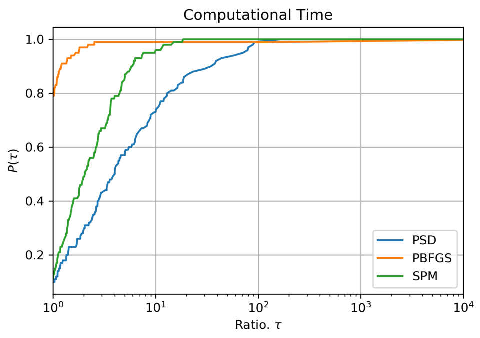
(a) With absolute

Figure 4.16: Profiling graph for number of iterations (test 2, $\mu = 10^{-6}$)



(a) With absolute

Figure 4.17: Profiling graph for number of function calls (test 2, $\mu = 10^{-6}$)



(a) With absolute

Figure 4.18: Profiling graph for computational time in seconds (test 2, $\mu = 10^{-6}$)

In test 2, all the components of the actual solution are within the interval $[-1,1]$. There is no result produced for the cases without absolute. This is because the proximal operator will change all the negative solutions to 0 and hence, the sparsity is too high and algorithm is unable to reach an approximate solution (the tolerance, ϵ is unachievable). While the results with absolute show that PBFGS has the best performance among the three methods since PBFGS method uses full rank matrix to approximate the Hessian matrix and hence, it consists more curvature information. Furthermore, the efficiency of the SPM method outperforms the PSD method because SPM method approximates the Hessian matrix with a diagonal matrix and PSD converges slowly in "zigzags" form towards the minimum point. To fulfill the objective in this research, the sparsity (%) is highly prioritized. Hence, table below shows the average sparsity of the solution (%):

Table 4.3: Average sparsity of the solution (test 2, with absolute)

	PBFGS	PSD	SPM
$\mu = 10^{-2}$	17.06	18.34	17.99
$\mu = 10^{-4}$	2.74	5.33	5.06
$\mu = 10^{-6}$	0.35	0.91	0.64

From the Table 4.3, it can be concluded that the average sparsity of the solution obtained by applying the PSD method and SPM method are comparable and exceed the one obtained from the PBFGS method.

The performances of these methods are problem-dependent. Overall, the profiling graphs show that the PSD method requires more iterations than others. It moves in a zigzaging form whenever the point gets nearer to the optimum solution. Hence, it takes more steps and has slower convergence. Besides that, the SPM method uses the fewest function calls to get the desired solution. This is mainly due to the SPM demands fewer function calls in the backtracking line search strategy to get the optimum step size while the PBFGS method and PSD method need more function calls to achieve similar outcomes.

PBFGS is well known as not favorable for large dimension problems. Theoretically, the SPM method requires less computational time per iteration compared to the PBFGS method. This is because the Hessian matrix is estimated in the PBFGS method by a full rank matrix with storage memory $O(n^2)$, whereas in the SPM method, the eigenvalues of the Hessian matrix are approximated by a diagonal matrix with storage memory $O(n)$. Thus, the SPM method needs less computational time per iteration due to the requirement of the storage is smaller.

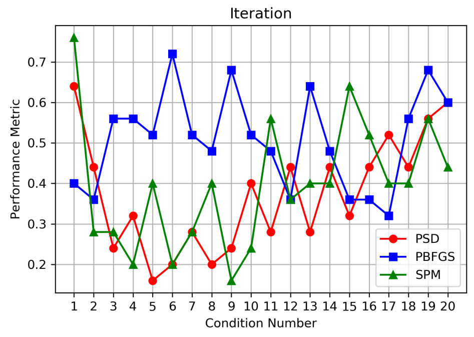
In general, the PBFGS method and SPM method have better efficiency. However, in terms of sparsity, the PSD method and SPM method obtained more stable and desirable results. Overall, in consideration of the efficiencies and the sparsity of the solutions, the results show that our proposed algorithms are more applicable in finding the sparse solution for a linear system.

4.2 Sensitivity Analysis on Condition Number

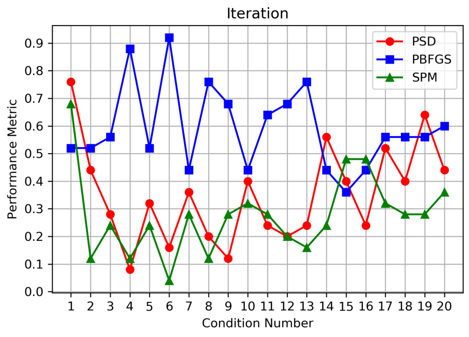
In this section, the same datasets from Section 4.1 are used and the influences of condition number on the performance of different variations of proximal gradient algorithms will be looked into. In order to compare the performance metric on different condition numbers for the proposed method with other proximal gradient methods, the results for these comparisons are shown in the graphs below:

1. Test 1: when $x^* \in [0,1]$, and

   (a) $\mu = 10^{-2}$

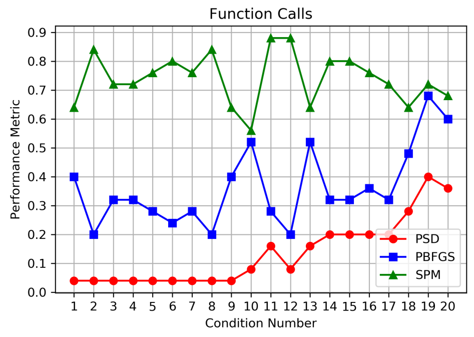


(a) Without absolute

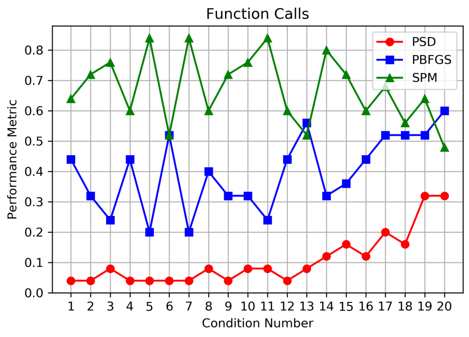


(b) With absolute

Figure 4.19: Sensitivity analysis for number of iterations (test 1, $\mu = 10^{-2}$)

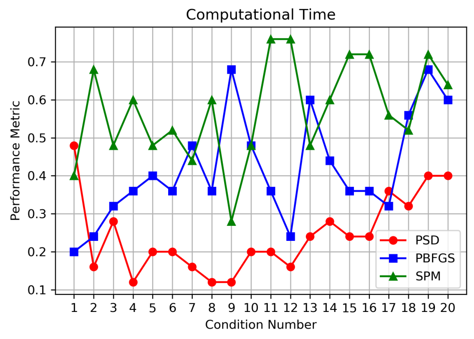


(a) Without absolute

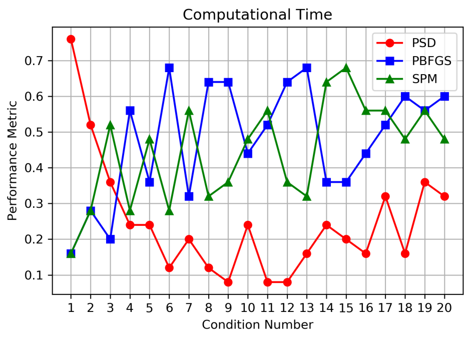


(b) With absolute

Figure 4.20: Sensitivity analysis for number of function calls (test 1, $\mu = 10^{-2}$)



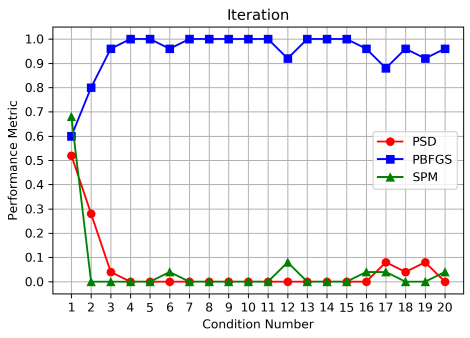
(a) Without absolute



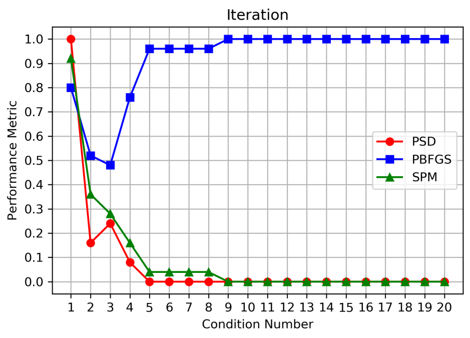
(b) With absolute

Figure 4.21: Sensitivity analysis for computational time in seconds (test 1, $\mu = 10^{-2}$)

(b) $\mu = 10^{-4}$

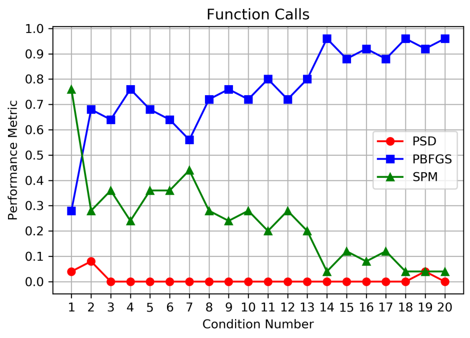


(a) Without absolute

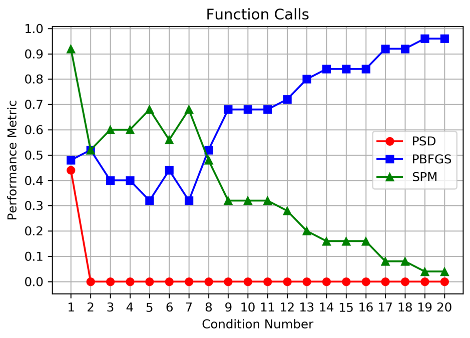


(b) With absolute

Figure 4.22: Sensitivity analysis for number of iterations (test 1, $\mu = 10^{-4}$)

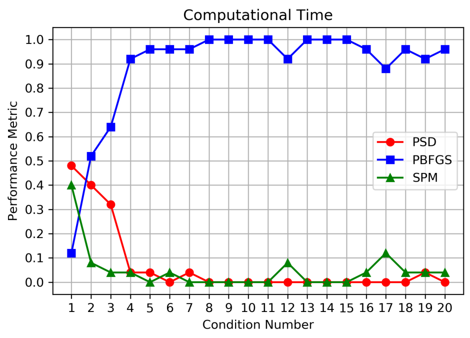


(a) Without absolute

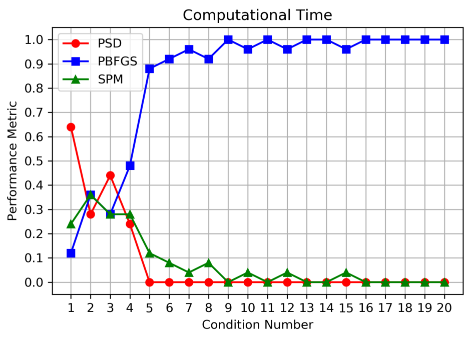


(b) With absolute

Figure 4.23: Sensitivity analysis for number of function calls (test 1, $\mu = 10^{-4}$)



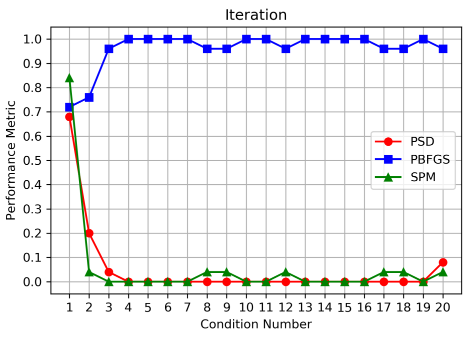
(a) Without absolute



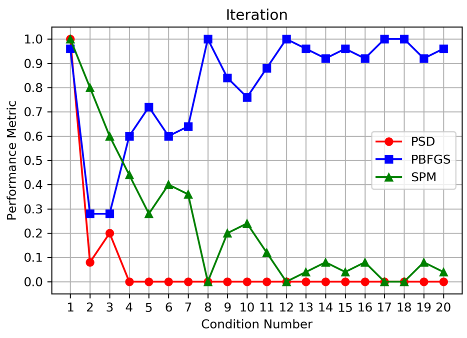
(b) With absolute

Figure 4.24: Sensitivity analysis for computational time in seconds (test 1, $\mu = 10^{-4}$)

(c) $\mu = 10^{-6}$

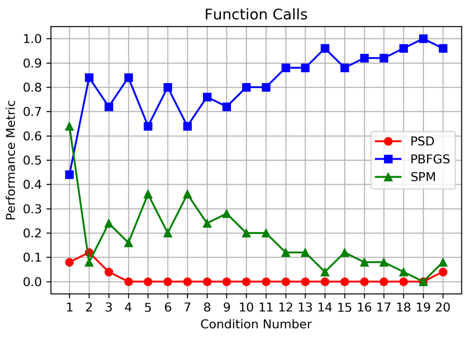


(a) Without absolute

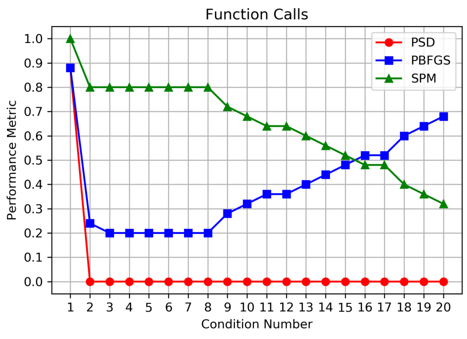


(b) With absolute

Figure 4.25: Sensitivity analysis for number of iterations (test 1, $\mu = 10^{-6}$)

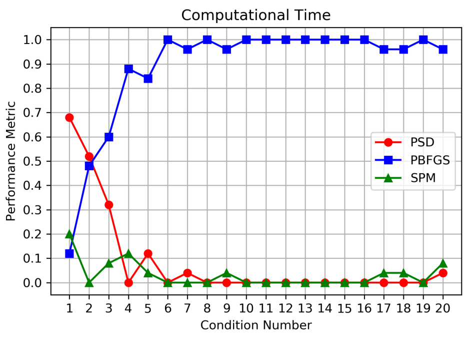


(a) Without absolute

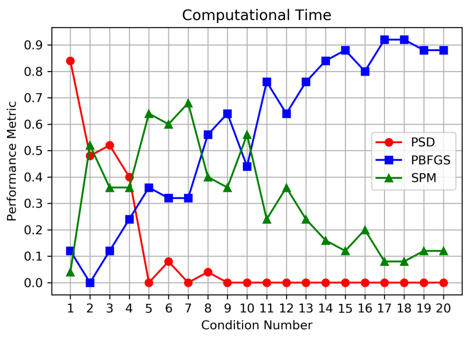


(b) With absolute

Figure 4.26: Sensitivity analysis for number of function calls (test 1, $\mu = 10^{-6}$)



(a) Without absolute

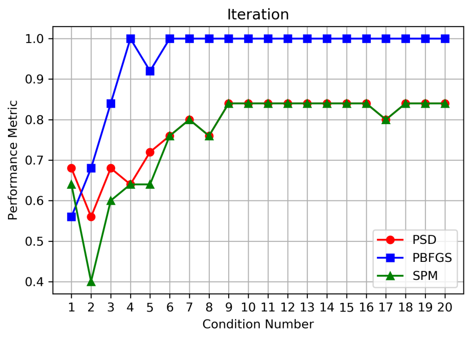


(b) With absolute

Figure 4.27: Sensitivity analysis for computational time in seconds (test 1, $\mu = 10^{-6}$)

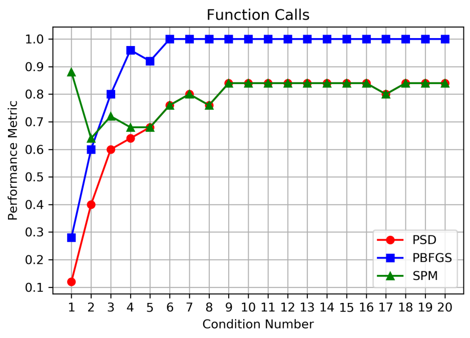
2. Test 2: when $x^* \in [-1, 1]$, and

   (a) $\mu = 10^{-2}$



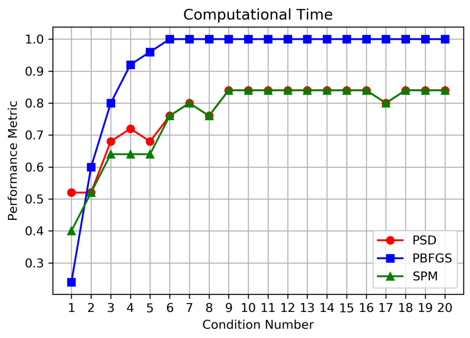
(a) With absolute

Figure 4.28: Sensitivity analysis for number of iterations (test 2, $\mu = 10^{-2}$)



(a) With absolute

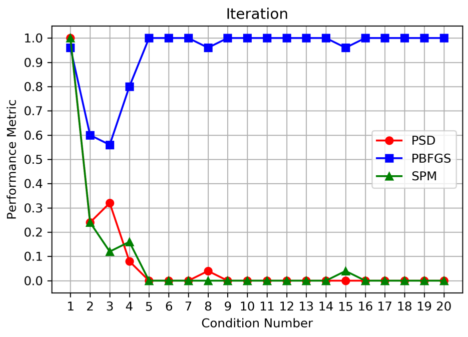
Figure 4.29: Sensitivity analysis for number of function calls (test 2, $\mu = 10^{-2}$)



(a) With absolute

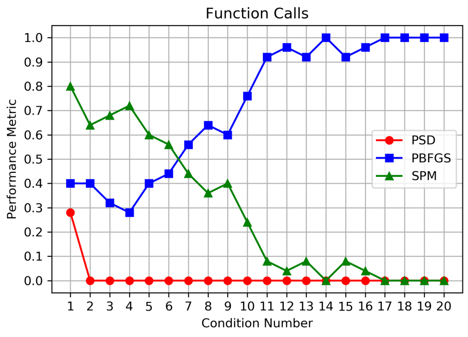
Figure 4.30: Sensitivity analysis for computational time in seconds (test 2, $\mu = 10^{-2}$)

(b) $\mu = 10^{-4}$



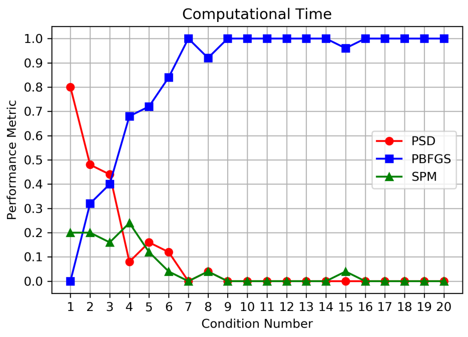
(a) With absolute

Figure 4.31: Sensitivity analysis for number of iterations (test 2, $\mu = 10^{-4}$)



(a) With absolute

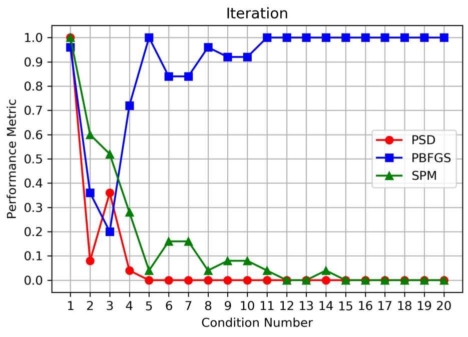
Figure 4.32: Sensitivity analysis for number of function calls (test 2, $\mu = 10^{-4}$)



(a) With absolute

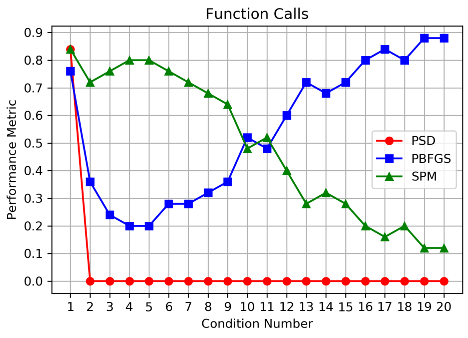
Figure 4.33: Sensitivity analysis for computational time in seconds (test 2, $\mu = 10^{-4}$)

(c) $\mu = 10^{-6}$



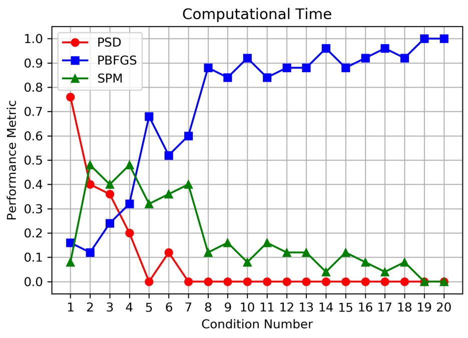
(a) With absolute

Figure 4.34: Sensitivity analysis for number of iterations (test 2, $\mu = 10^{-6}$)



(a) With absolute

Figure 4.35: Sensitivity analysis for number of function calls (test 2, $\mu = 10^{-6}$)



(a) With absolute

Figure 4.36: Sensitivity analysis for computational time in seconds (test 2, $\mu = 10^{-6}$)

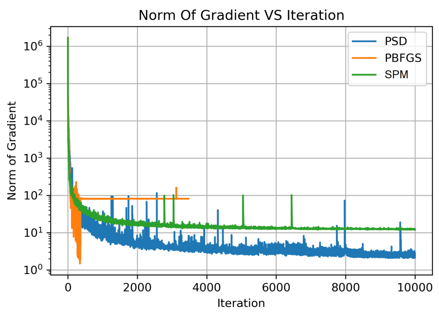
From the figures shown, the performances of the SPM method and PBFGS method are comparable when $\mu = 10^{-2}$. It is worth noting that figures from methods with absolute show that the SPM method outperforms the others in solving well-conditioned problems. The SPM method requires less time and fewer function calls in each iteration. While the PBFGS method approximates a full rank Hessian matrix which consists of more information requires fewer number of iterations to reach the minimum point. Hence, the total computational time and total function call required for the PBFGS method decline following the decrease in the total number of iterations.

4.3 Applications

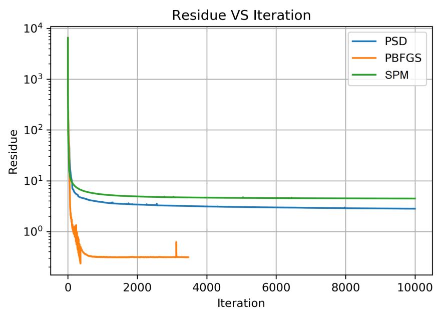
Finally, we evaluate the efficiency of the proposed method by applying it to a real-life problem. In this section, we compare the performance of the proposed method with existing methods on each iteration using ill-conditioned MNIST datasets. The dimension of matrix A is $k \times 784$, where k is the number of image samples chosen in the dataset. In this application, we not only apply the SPM method and other proximal gradient methods to the underdetermined cases, but also overdetermined cases. Therefore, k is chosen as 500, 1000, and 10000.

Figures below show the performance of these methods in the cases with different k numbers of image samples. The norm of gradient, residue, number of function calls, and total computational time in seconds are all compared between these methods. Here, the residue plotted is calculated as $\|Ax - b\|_2$.

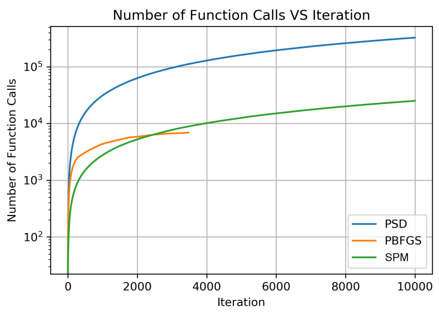
1. When $k = 500$:



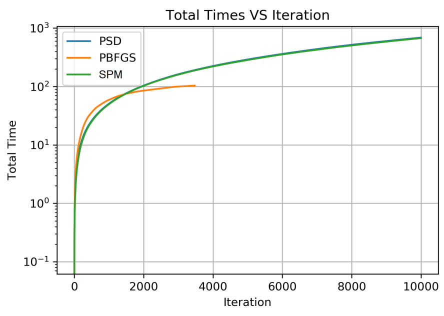
(a) Norm of gradient



(b) Residue



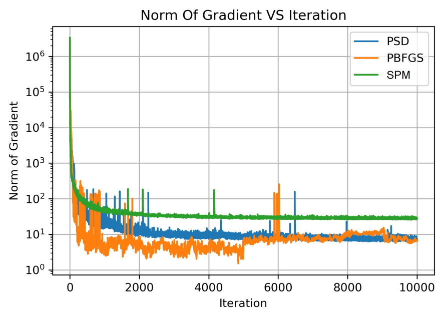
(c) Number of function calls



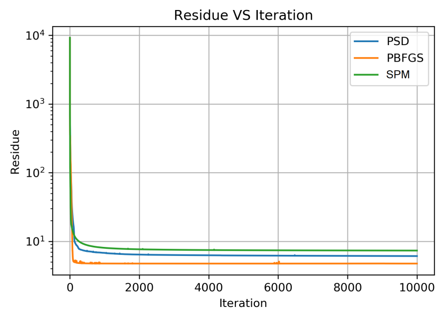
(d) Total times

Figure 4.37: Evolutions of the efficiencies with respect to the number of iterations for k = 500 (MNIST)

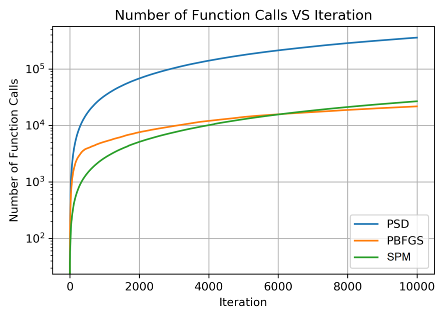
2. When $k = 1000$:



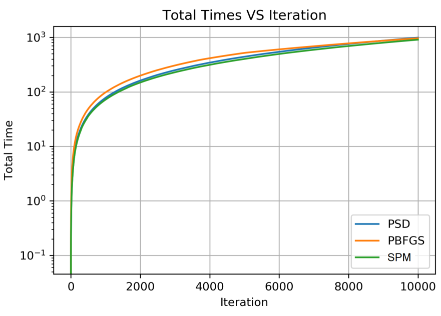
(a) Norm of gradient



(b) Residue



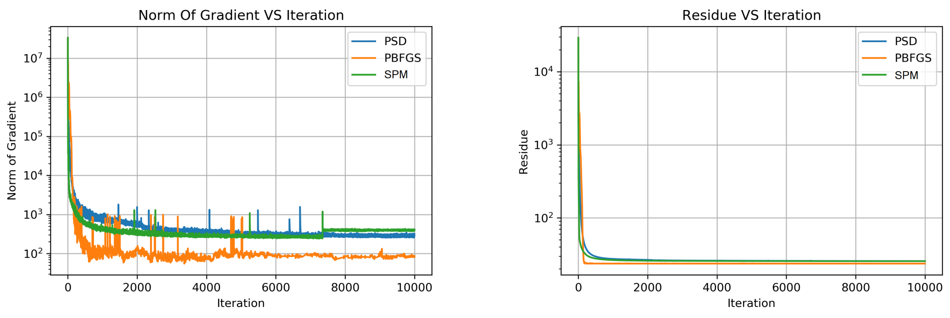
(c) Number of function calls



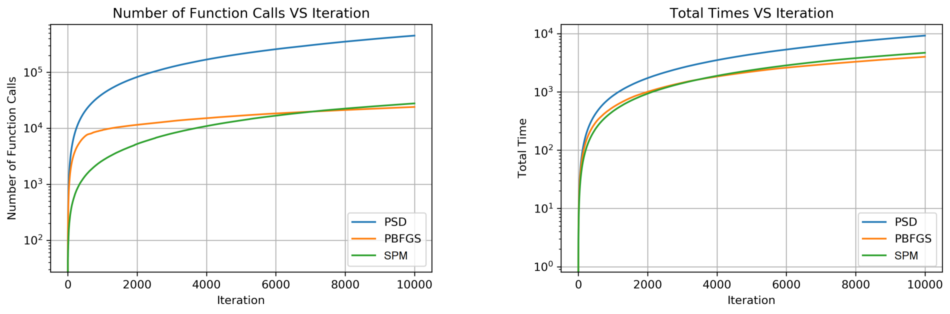
(d) Total times

Figure 4.38: Evolutions of the efficiencies with respect to the number of iterations for k = 1000 (MNIST)

3. When $k = 10000$:



(a) **Norm of gradient** (b) **Residue**



(c) **Number of function calls** (d) **Total times**

Figure 4.39: Evolutions of the efficiencies with respect to the number of iterations for k = 10000 (MNIST)

From the figures shown, the PBFGS algorithm outperforms the other proximal gradient algorithms. Although PBFGS requires a fewer iterations and computational times than the SPM algorithm, the sparsity of the solution is undesirable. In this research, our main focus is to achieve high sparsity in the approximate solution, which necessitates a low tolerance in order to obtain a satisfactory sparse solution.

Sparsity for each label is calculated as the percentage of zero elements contained in the solution. We average them among labels ('0'-'9'). The table below shows the average sparsity of the solution (%):

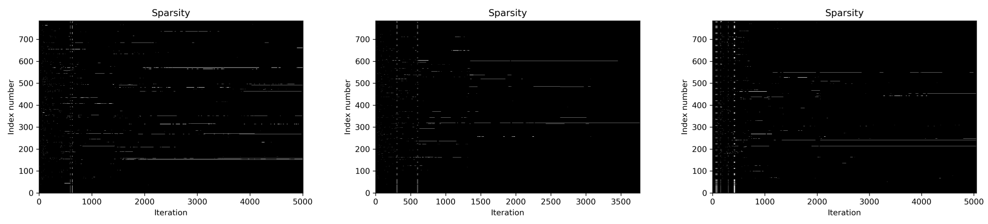
Table 4.4: Average sparsity of the solution (MNIST)

	PBFGS	PSD	SPM
k=500	3.13	31.96	56.14
k=1000	0.83	37.29	59.38
k=10000	2.27	66.31	62.80

Table 4.4 shows that the solutions of PSD and SPM have achieved a desirable sparsity (in most of the cases, SPM has a better sparsity than PSD). The performance of SPM and PSD in terms of convergence rate and computational times is very similar for low accuracies of the solution, as shown in the figures. However, PSD requires more function calls in the backtracking line-search procedure than SPM. In fact, these results are problem-dependent, the performance of the algorithms should always be verified empirically in the specific application.

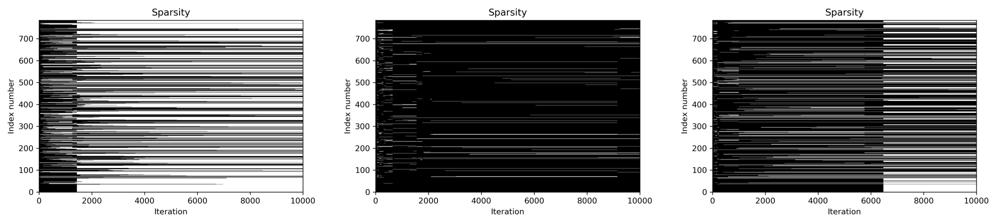
The stability for the sparsity of the solution is also important. To verify the stability for the sparsity of these proximal gradient methods, the figures below are plotted to indicate the sparsity for the solution, x_k at every k^{th} iteration. The x-axis is the k^{th} iteration number and the y-axis represents the elements of the solution, x_k at every k^{th} iteration. The black block indicates the nonzero element and the white block indicates the zero element.

Here, we only show the graphs from labels ('1', '5' and '9') with $k = 1000$:



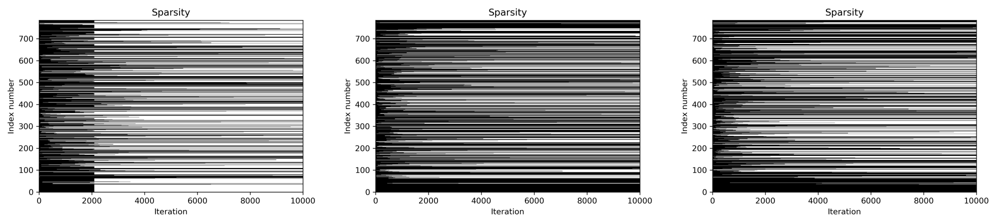
(a) Label '1' (b) Label '5' (c) Label '9'

Figure 4.40: Sparsity of the solutions obtained from the PBFGS method with respect to the number of iterations for k =1000 (MNIST)



(a) Label '1' (b) Label '5' (c) Label '9'

Figure 4.41: Sparsity of the solutions obtained from the PSD method with respect to the number of iterations for k =1000 (MNIST)



(a) Label '1' (b) Label '5' (c) Label '9'

Figure 4.42: Sparsity of the solutions obtained from the SPM method with respect to the number of iterations for k =1000 (MNIST)

From these figures, we can see that the sparsity of the SPM method are the highest. In most cases, SPM is a robust method for sparse optimisation because it can produce higher sparsity and there is less random switching of the zero elements when the iterative optimisation procedure performs.

CHAPTER 5

CONCLUSION AND FUTURE WORKS

5.1 Conclusion

The l_0-norm sparse optimisation involves minimizing l_0-norm which is non-convex and non-smooth. Spectral gradient method was proposed by Sim et al. (2019). This method generally requires less storage, $O(n)$, and less CPU time. A new method called spectral proximal method (SPM) was proposed in this research which incorporating the spectral gradient method with the proximal method for solving a large-scale sparse optimisation problem.

The numerical experiments conducted on the simulated datasets showed that the efficiency for the methods was problem-dependent. Although the PBFGS method performed better than SPM and PSD in some problems, but the solutions were not sparse. The proposed method, SPM showed that it has better efficiency than the PSD method in finding the sparse solution for a linear system. Furthermore, the solutions obtained from the SPM method were more stable and sparser.

In real-life applications, these methods have been tested on highly ill-conditioned MNIST datasets. The results showed that the SPM method was robust in finding the sparse solution for underdetermined and also overdetermined linear system problems. Hence, we could conclude that this proposed method can be used to solve large-scale and ill-conditioned problems.

5.2 Future Works

In this research, this method was only applied to least square regression problems, however, it can also be considered in solving logistic regression problems. Moreover, the algorithm of the methods will be terminated when $\|g_{k+1}\| \leq \epsilon$, without concerning the sparsity of the solution. As an alternative, the percentage of the sparsity of the solution should also be considered as the stopping criterion. Further improvement of this method might include integrating alternative backtracking line search algorithms into the algorithm, which could improve its efficiency even further. We can apply strong Wolfe condition for monotone line search (Wolfe, 1969, 1971). Moreover, a non-monotone line search strategy can also be applied onto the SPM method. From the results shown in Sim et al. (2019), the spectral gradient method with non-monotone line search is performed better than with monotone line search.

Besides that, the tuning parameters λ and μ can be further investigated. The coefficient $\lambda \geq 0$ will give weight to the regularization term. It balances the weight between the data fidelity term and the regularization term in the sparse optimisation problem. It needs careful tuning to avoid data fidelity term becomes too small and leads to overfitting problem. Components of x^* can avoid becoming excessively large by minimizing the sparsity-inducing regularization term, $\|x\|_0$, and hence ensuring numerical stability (Antonello et al., 2018). It can be tuned to a lower value if more weightage is required on the sparsity of the solution. Conversely, λ can be tuned to a higher value when the problem is to emphasize the accuracy of the solution. The tuning parameter μ for the proximal operator also can be tuned to other values based on different kinds of problems. It controls the sparsity of the solution.

This proposed method is useful for supervised learning. Most investors in the actual world prefer to invest in a limited number of stocks. It can generate a sparse and stable portfolio model in portfolio optimisation. In a portfolio selection problem, the l_0-norm functions as a regularisation term for the objective function. This penalty encourages sparse portfolios with only a few active positions (Dai and Wen, 2018).

Our proposed method can also be further applied to work on machine learning, neural network, and image processing fields. Currently, there is a trend in building deeper and larger neural networks. The number of training parameters is of order of ten of thousands and requires large memory storage. By applying SPM method, it will be potentially reduced the memory cost in this kind of deep neural networks. Hence, we are interesting to study the efficiency of this method in the neural network. Besides that, for image processing, inverse problems are involved. The number of pixels in the modern camera is large, hence, the SPM method can be used to reduce the memory cost in the sparse optimisation while the cost function tries to minimize the difference between the inverse model and the image.

BIBLIOGRAPHY

Agarana, M., Anake, T. and Okagbue, H. (2016), Optimization of urban rail transportation in emerging countries using operational research techniques, *Applied Mathematics* **7**, 1116–1123.

Antonello, N., Stella, L., Patrinos, P. and van Waterschoot, T. (2018), Proximal gradient algorithms: Applications in signal processing, *arXiv preprint arXiv:1803.01621* .

Armijo, L. (1966), Minimization of functions having lipschitz continuous first partial derivatives, *Pacific Journal of mathematics* **16**(1), 1–3.

Arthanari, T. and Dodge, Y. (1981), *Mathematical programming in statistics*, Vol. 341, Wiley New York.

Bao, C., Dong, B., Hou, L., Shen, Z., Zhang, X. and Zhang, X. (2016), Image restoration by minimizing zero norm of wavelet frame coefficients, *Inverse problems* **32**(11), 115004.

Barzilai, J. and Borwein, J. M. (1988), Two-point step size gradient methods, *IMA journal of numerical analysis* **8**(1), 141–148.

Beck, A. and Teboulle, M. (2009), A fast iterative shrinkage-thresholding algorithm for linear inverse problems, *SIAM journal on imaging sciences* **2**(1), 183–202.

Becker, S., Bobin, J. and Candès, E. J. (2011), Nesta: A fast and accurate first-order method for sparse recovery, *SIAM Journal on Imaging Sciences* **4**(1), 1–39.

Becker, S. and Fadili, J. (2012), A quasi-newton proximal splitting method, *Advances in neural information processing systems* **25**, 2618–2626.

Berger, C. R., Wang, Z., Huang, J. and Zhou, S. (2010), Application of compressive sensing to sparse channel estimation, *IEEE Communications Magazine* **48**(11), 164–174.

Bertsekas, D. P. (1976), On the goldstein-levitin-polyak gradient projection method, *IEEE Transactions on automatic control* **21**(2), 174–184.

Bertsimas, D., King, A. and Mazumder, R. (2016), Best subset selection via a modern optimization lens, *Annals of statistics* **44**(2), 813–852.

Best, M. J. (2010), *Portfolio optimization*, CRC Press.

Bioucas-Dias, J. M. and Figueiredo, M. A. (2007), A new twist: Two-step iterative shrinkage/thresholding algorithms for image restoration, *IEEE Transactions on Image processing* **16**(12), 2992–3004.

Birgin, E. G., Martínez, J. M. and Raydan, M. (2000), Nonmonotone spectral projected gradient methods on convex sets, *SIAM Journal on Optimization* **10**(4), 1196–1211.

Birgin, E. G., Martínez, J. M. and Raydan, M. (2001), Algorithm 813: Spg—software for convex-constrained optimization, *ACM Transactions on Mathematical Software (TOMS)* **27**(3), 340–349.

Birgin, E. G., Martínez, J. M. and Raydan, M. (2003), Inexact spectral projected gradient methods on convex sets, *IMA Journal of Numerical Analysis* **23**(4), 539–559.

Birgin, E. G., Martínez, J. M. and Raydan, M. (2014), Spectral projected gradient methods: review and perspectives, *Journal of Statistical Software* **60**(1), 1–21.

Blanquero, R., Carrizosa, E., Molero-Río, C. and Morales, D. R. (2020), Sparsity in optimal randomized classification trees, *European Journal of Operational Research* **284**(1), 255–272.

Blumensath, T. (2012), Accelerated iterative hard thresholding, *Signal Processing* **92**(3), 752–756.

Blumensath, T. and Davies, M. E. (2008), Iterative thresholding for sparse approximations, *Journal of Fourier analysis and Applications* **14**(5-6), 629–654.

Blumensath, T. and Davies, M. E. (2009), Iterative hard thresholding for compressed sensing, *Applied and computational harmonic analysis* **27**(3), 265–274.

Boyd, S., Boyd, S. P. and Vandenberghe, L. (2004*a*), *Convex optimization*, Cambridge university press.

Boyd, S., Boyd, S. P. and Vandenberghe, L. (2004*b*), *Convex optimization*, Cambridge university press.

Bradley, P. S. and Mangasarian, O. L. (1998), Feature selection via concave minimization and support vector machines, *in* 'ICML', Vol. 98, pp. 82–90.

Brodie, J., Daubechies, I., De Mol, C., Giannone, D. and Loris, I. (2009), Sparse and stable markowitz portfolios, *Proceedings of the National Academy of Sciences* **106**(30), 12267–12272.

Broyden, C. G. (1967), Quasi-newton methods and their application to function minimisation, *Mathematics of Computation* **21**(99), 368–381.

Broyden, C. G. (1970*a*), The convergence of a class of double-rank minimization algorithms 1. general considerations, *IMA Journal of Applied Mathematics* **6**(1), 76–90.

Broyden, C. G. (1970*b*), The convergence of a class of double-rank minimization algorithms: 2. the new algorithm, *IMA journal of applied mathematics* **6**(3), 222–231.

Broyden, C. G., Dennis Jr, J. E. and Moré, J. J. (1973), On the local and superlinear convergence of quasi-newton methods, *IMA Journal of Applied Mathematics* **12**(3), 223–245.

Bruckstein, A. M., Donoho, D. L. and Elad, M. (2009), From sparse solutions of systems of equations to sparse modeling of signals and images, *SIAM review* **51**(1), 34–81.

Byrd, R. H. and Nocedal, J. (1989), A tool for the analysis of quasi-newton methods with application to unconstrained minimization, *SIAM Journal on Numerical Analysis* **26**(3), 727–739.

Candes, E. J. (2008), The restricted isometry property and its implications for compressed sensing, *Comptes rendus mathematique* **346**(9-10), 589–592.

Candes, E. J., Eldar, Y. C., Strohmer, T. and Voroninski, V. (2015), Phase retrieval via matrix completion, *SIAM review* **57**(2), 225–251.

Candes, E. J., Romberg, J. K. and Tao, T. (2006), Stable signal recovery from incomplete and inaccurate measurements, *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences* **59**(8), 1207–1223.

Candes, E. J. and Tao, T. (2005), Decoding by linear programming, *IEEE transactions on information theory* **51**(12), 4203–4215.

Candes, E. J., Wakin, M. B. and Boyd, S. P. (2008), Enhancing sparsity by reweighted l_1 minimization, *Journal of Fourier analysis and applications* **14**(5-6), 877–905.

Cauchy, A. (1847), Méthode générale pour la résolution des systemes d'équations simultanées, *Comp. Rend. Sci. Paris* **25**(1847), 536–538.

Chartrand, R. and Staneva, V. (2008), Restricted isometry properties and nonconvex compressive sensing, *Inverse Problems* **24**(3), 035020.

Chen, X. and Fukushima, M. (1999), Proximal quasi-newton methods for nondifferentiable convex optimization, *Mathematical Programming* **85**(2), 313–334.

Chen, X., Xu, F. and Ye, Y. (2010), Lower bound theory of nonzero entries in solutions of l_2-l_p minimization, *SIAM Journal on Scientific Computing* **32**(5), 2832–2852.

Chen, Z., Jin, X., Li, L. and Wang, G. (2013), A limited-angle ct reconstruction method based on anisotropic tv minimization, *Physics in Medicine & Biology* **58**(7), 2119.

Cheney, E. W. and Kincaid, D. R. (2012), *Numerical mathematics and computing*, Cengage Learning.

Chouzenoux, E., Pesquet, J.-C. and Repetti, A. (2014), Variable metric forward–backward algorithm for minimizing the sum of a differentiable function and a convex function, *Journal of Optimization Theory and Applications* **162**(1), 107–132.

Collobert, R., Sinz, F., Weston, J. and Bottou, L. (2006), Trading convexity for scalability, *in* 'Proceedings of the 23rd international conference on Machine learning', pp. 201–208.

Combettes, P. L. and Wajs, V. R. (2005), Signal recovery by proximal forward-backward splitting, *Multiscale Modeling & Simulation* **4**(4), 1168–1200.

Dai, Z. and Wen, F. (2018), A generalized approach to sparse and stable portfolio optimization problem, *Journal of Industrial & Management Optimization* **14**(4), 1651.

Davidon, W. C. (1959), Variable metric method for minimization (revised), Technical report, Technical Report ANL-5990, Argonne National Laboratory, USA.

Deng, W., Yin, W. and Zhang, Y. (2013), Group sparse optimization by alternating direction method., *in* 'Proceedings of SPIE - The International Society for Optical Engineering', Vol. 8858, Department of Computational and Applied Mathematics, Rice University.

Dinh, T. P. and Le Thi, H. A. (2014), Recent advances in dc programming and dca, *in* 'Transactions on computational intelligence XIII', Springer, pp. 1–37.

Dolan, E. D. and Moré, J. J. (2002), Benchmarking optimization software with performance profiles, *Mathematical programming* **91**(2), 201–213.

Dong, B. and Zhang, Y. (2013), An efficient algorithm for l_0 minimization in wavelet frame based image restoration, *Journal of Scientific Computing* **54**(2), 350–368.

Donoho, D. L. (2006*a*), Compressed sensing, *IEEE Transactions on information theory* **52**(4), 1289–1306.

Donoho, D. L. (2006*b*), For most large underdetermined systems of equations, the minimal l_1-norm near-solution approximates the sparsest near-solution, *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences* **59**(7), 907–934.

Donoho, D. L. (2006*c*), For most large underdetermined systems of linear equations the minimal l_1-norm solution is also the sparsest solution, *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences* **59**(6), 797–829.

Fan, J. and Li, R. (2001), Variable selection via nonconcave penalized likelihood and its oracle properties, *Journal of the American statistical Association* **96**(456), 1348–1360.

Fei-Yue, K., Si-Jing, C., Xue-Fen, W. and Yi, Y. (2019), Parallel algorithmic optimization for enhanced quasi-maximum likelihood decoding on gpu, *in* '2019 IEEE 2nd International Conference on Electronics and Communication Engineering (ICECE)', IEEE, pp. 42–46.

Figueiredo, M. A. and Nowak, R. D. (2003), An em algorithm for wavelet-based image restoration, *IEEE Transactions on Image Processing* **12**(8), 906–916.

Figueiredo, M. A., Nowak, R. D. and Wright, S. J. (2007), Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems, *IEEE Journal of selected topics in signal processing* **1**(4), 586–597.

Fletcher, R. (1970), A new approach to variable metric algorithms, *The computer journal* **13**(3), 317–322.

Fletcher, R. and Powell, M. J. (1963), A rapidly convergent descent method for minimization, *The computer journal* **6**(2), 163–168.

Fletcher, R. and Reeves, C. M. (1964), Function minimization by conjugate gradients, *The computer journal* **7**(2), 149–154.

Fu, W. J. (1998), Penalized regressions: the bridge versus the lasso, *Journal of computational and graphical statistics* **7**(3), 397–416.

Fukushima, M. and Mine, H. (1981), A generalized proximal point algorithm for certain non-convex minimization problems, *International Journal of Systems Science* **12**(8), 989–1000.

Gale, T., Elsen, E. and Hooker, S. (2019), The state of sparsity in deep neural networks, *arXiv preprint arXiv:1902.09574* .

Gasso, G., Rakotomamonjy, A. and Canu, S. (2009), Recovering sparse signals with a certain family of nonconvex penalties and dc programming, *IEEE Transactions on Signal Processing* **57**(12), 4686–4698.

Goldfarb, D. (1970), A family of variable-metric methods derived by variational means, *Mathematics of computation* **24**(109), 23–26.

Goldstein, A. A. (1964), Convex programming in hilbert space, *Bulletin of the American Mathematical Society* **70**(5), 709–710.

Gotoh, J.-y., Takeda, A. and Tono, K. (2018), Dc formulations and algorithms for sparse optimization problems, *Mathematical Programming* **169**(1), 141–176.

Gribonval, R. and Nielsen, M. (2003), Sparse representations in unions of bases, *IEEE transactions on Information theory* **49**(12), 3320–3325.

Guan, W. and Gray, A. (2013), Sparse high-dimensional fractional-norm support vector machine via dc programming, *Computational Statistics & Data Analysis* **67**, 136–148.

Hale, E. T., Yin, W. and Zhang, Y. (2008), Fixed-point continuation for l_1-minimization: Methodology and convergence, *SIAM Journal on Optimization* **19**(3), 1107–1130.

Hare, W. and Sagastizábal, C. (2009), Computing proximal points of nonconvex functions, *Mathematical Programming* **116**(1-2), 221–258.

Hastie, T., Tibshirani, R. and Wainwright, M. (2015), *Statistical learning with sparsity: the lasso and generalizations*, CRC press.

Hestenes, M. R. and Stiefel, E. (1952), *Methods of conjugate gradients for solving linear systems*, Vol. 49, NBS Washington, DC.

Horst, R. and Tuy, H. (2013), *Global optimization: Deterministic approaches*, Springer Science & Business Media.

Huang, H.-Y. (1970), Unified approach to quadratically convergent algorithms for function minimization, *Journal of Optimization Theory and Applications* **5**(6), 405–423.

Huang, J. and Li, H. (2012), A novel real-time optimization methodology for chemical plants, *Chinese Journal of Chemical Engineering* **20**(6), 1059 – 1066.

Johansson, C., Derelöv, M. and Ölvander, J. (2017), How to use an optimization-based method capable of balancing safety, reliability, and weight in an aircraft design process, *Nuclear Engineering and Technology* **49**(2), 404–410.

Karimi, S. and Vavasis, S. (2017), Imro: A proximal quasi-newton method for solving l_1-regularized least squares problems, *SIAM Journal on Optimization* **27**(2), 583–615.

Le, H. M., Le Thi, H. A. and Nguyen, M. C. (2013), Dca based algorithms for feature selection in semi-supervised support vector machines, *in* 'International Workshop on Machine Learning and Data Mining in Pattern Recognition', Springer, pp. 528–542.

Le Thi, H. A., Dinh, T. P., Le, H. M. and Vo, X. T. (2015), Dc approximation approaches for sparse optimization, *European Journal of Operational Research* **244**(1), 26–46.

Le Thi, H. A., Le, H. M. and Dinh, T. P. (2008), A dc programming approach for feature selection in support vector machines learning, *Advances in Data Analysis and Classification* **2**(3), 259–278.

Le Thi, H. A. and Nguyen, M. C. (2013), Efficient algorithms for feature selection in multi-class support vector machine, *in* 'Advanced Computational Methods for Knowledge Engineering', Springer, pp. 41–52.

Le Thi, H. A. and Ouchani, S. (2009), Gene selection for cancer classification using dca, journal of frontiers of computer science and technology, *Journal of Frontiers of Computer Science and Technology* **3**(6), 612–620.

Le Thi, H. A., Thi, B. T. N. and Le, H. M. (2013), Sparse signal recovery by difference of convex functions algorithms, *in* 'Asian Conference on Intelligent Information and Database Systems', Springer, pp. 387–397.

LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P. (1998), Gradient-based learning applied to document recognition, *Proceedings of the IEEE* **86**(11), 2278–2324.

Lee, J. D., Sun, Y. and Saunders, M. A. (2014), Proximal newton-type methods for minimizing composite functions, *SIAM Journal on Optimization* **24**(3), 1420–1443.

Levitin, E. S. and Polyak, B. T. (1966), Constrained minimization methods, *USSR Computational mathematics and mathematical physics* **6**(5), 1–50.

Li, Y., Gu, S., Mayer, C., Gool, L. V. and Timofte, R. (2020), Group sparsity: The hinge between filter pruning and decomposition for network compression, *in* 'Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition', pp. 8018–8027.

Ling, S. and Strohmer, T. (2015), Self-calibration and biconvex compressive sensing, *Inverse Problems* **31**(11), 115002.

Lu, Z. (2014), Iterative hard thresholding methods for l_0 regularized convex cone programming, *Mathematical Programming* **147**(1-2), 125–154.

Luenberger, D. G. and Ye, Y. (1984), *Linear and nonlinear programming*, Vol. 2, Springer.

Luo, Z.-Q., Ma, W.-K., So, A. M.-C., Ye, Y. and Zhang, S. (2010), Semidefinite relaxation of quadratic optimization problems, *IEEE Signal Processing Magazine* **27**(3), 20–34.

Lustig, M., Donoho, D. and Pauly, J. M. (2007), Sparse mri: The application of compressed sensing for rapid mr imaging, *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine* **58**(6), 1182–1195.

Majozi, T., Seid, E. R. and Lee, J.-Y. (2015), *Synthesis, design, and resource optimization in batch chemical plants*, CRC press.

Mangasarian, O. (1996), Machine learning via polyhedral concave minimization, *in* 'Applied Mathematics and Parallel Computing', Springer, pp. 175–188.

Martinet, B. (1970), Régularisation d'inéquations variationnelles par approximations successives. rev. française informat, *Recherche Opérationnelle* **4**, 154–158.

Narang, S., Elsen, E., Diamos, G. and Sengupta, S. (2017), Exploring sparsity in recurrent neural networks, *arXiv preprint arXiv:1704.05119* .

Natarajan, B. K. (1995), Sparse approximate solutions to linear systems, *SIAM journal on computing* **24**(2), 227–234.

Neumann, J., Schnörr, C. and Steidl, G. (2005), Combined svm-based feature selection and classification, *Machine learning* **61**(1-3), 129–150.

Niwattisaiwong, S. and Suriya, K. (2018), Optimal initial values in maximum likelihood estimation of logistic regression models, *International Journal of Intelligent Technologies & Applied Statistics* **11**(2).

Nocedal, J. and Wright, S. (2006), *Numerical optimization*, Springer Science & Business Media.

Ong, C. S. and An, L. T. H. (2013), Learning sparse classifiers with difference of convex functions algorithms, *Optimization Methods and Software* **28**(4), 830–854.

Parikh, N. and Boyd, S. (2014), Proximal algorithms, *Foundations and Trends in optimization* **1**(3), 127–239.

Park, Y., Dhar, S., Boyd, S. and Shah, M. (2020), Variable metric proximal gradient method with diagonal barzilai-borwein stepsize, *in* 'ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)', IEEE, pp. 3597–3601.

Patel, V. M., Easley, G. R., Healy Jr, D. M. and Chellappa, R. (2010), Compressed synthetic aperture radar, *IEEE Journal of selected topics in signal processing* **4**(2), 244–254.

Peleg, D. and Meir, R. (2008), A bilinear formulation for vector sparsity optimization, *Signal Processing* **88**(2), 375–389.

Platt, J. C. (1999), Using analytic qp and sparseness to speed training of support vector machines, *in* 'Advances in neural information processing systems', pp. 557–563.

Poisel, R. A. (2012), *Electronic Warfare Target Location Methods, Second Edition*, Artech House.

Polson, N. G., Scott, J. G. and Willard, B. T. (2015), Proximal algorithms in statistics and machine learning, *Statistical Science* **30**(4), 559–581.

Polyak, B. T. (1969), The conjugate gradient method in extremal problems, *USSR Computational Mathematics and Mathematical Physics* **9**(4), 94–112.

Pyzara, A., Bylina, B. and Bylina, J. (2011), The influence of a matrix condition number on iterative methods' convergence, *in* '2011 Federated Conference on Computer Science and Information Systems (FedCSIS)', IEEE, pp. 459–464.

Rao, B. D. and Kreutz-Delgado, K. (1999), An affine scaling methodology for best basis selection, *IEEE Transactions on signal processing* **47**(1), 187–200.

Rao, S. S. (2009), *Engineering Optimization: Theory and Practice*, John Wiley & Sons, Inc.

Raydan, M. (1997), The barzilai and borwein gradient method for the large scale unconstrained minimization problem, *SIAM Journal on Optimization* **7**(1), 26–33.

Shanno, D. F. (1970), Conditioning of quasi-newton methods for function minimization, *Mathematics of computation* **24**(111), 647–656.

Sharma, A., Bhadauria, S. S. and Gupta, R. (2019), Image compression and sparsity measurement by using multilevel and different wavelet functions, pp. 517–520.

Shevade, S. K. and Keerthi, S. S. (2003), A simple and efficient algorithm for gene selection using sparse logistic regression, *Bioinformatics* **19**(17), 2246–2253.

Sim, H. S., Leong, W. J. and Chen, C. Y. (2019), Gradient method with multiple damping for large-scale unconstrained optimization, *Optimization Letters* **13**(3), 617–632.

Soussen, C., Idier, J., Brie, D. and Duan, J. (2011), From bernoulli–gaussian deconvolution to sparse signal restoration, *IEEE Transactions on Signal Processing* **59**(10), 4572–4584.

Stojaković, P., Velimirović, K. and Rašuo, B. (2018), Power optimization of a single propeller airplane take-off run on the basis of lateral maneuver limitations, *Aerospace Science and Technology* **72**, 553–563.

Sun, Y., Tan, X., Li, X., Lei, L. and Kuang, G. (2020), Sparse optimization problem with s-difference regularization, *Signal Processing* **168**, 107369.

Sun, Y. and Tao, J. (2014*a*), Few views image reconstruction using alternating direction method via l_0-norm minimization, *International Journal of Imaging Systems and Technology* **3**(24), 215–223.

Sun, Y. and Tao, J. (2014*b*), Image reconstruction from few views by l0-norm optimization, *arXiv preprint arXiv:1401.1882* .

Takeda, A., Niranjan, M., Gotoh, J.-y. and Kawahara, Y. (2013), Simultaneous pursuit of out-of-sample performance and sparsity in index tracking portfolios, *Computational Management Science* **10**(1), 21–49.

Teplická, K., Hurná, S., Kádárová, J. and Čulková, K. (2020), Competitiveness increasing in mining companies through application of operation research methods, *TEM Journal* **9**(1), 393–401.

Theodoridis, S. (2015), *Machine learning: a Bayesian and optimization perspective*, Academic press.

Thiao, M., Dinh, T. P. and Le Thi, H. A. (2008), Dc programming approach for a class of nonconvex programs involving l_0 norm, *in* 'International Conference on Modelling, Computation and Optimization in Information Systems and Management Sciences', Springer, pp. 348–357.

Thiao, M. and Tao, P. D. (2010), A dc programming approach for sparse eigenvalue problem, *in* 'ICML 2010 - Proceedings, 27th International Conference on Machine Learning', pp. 1063–1070.

Tibshirani, R. (1996), Regression shrinkage and selection via the lasso, *Journal of the Royal Statistical Society: Series B (Methodological)* **58**(1), 267–288.

Tono, K., Takeda, A. and Gotoh, J.-y. (2017), Efficient dc algorithm for constrained sparse optimization, *arXiv preprint arXiv:1701.08498* .

Trzasko, J. and Manduca, A. (2008), Highly undersampled magnetic resonance image reconstruction via homotopic l_0-minimization, *IEEE Transactions on Medical imaging* **28**(1), 106–121.

Trzasko, J., Manduca, A. and Borisch, E. (2007), Sparse mri reconstruction via multiscale l0-continuation, *in* '2007 IEEE/SP 14th Workshop on Statistical Signal Processing', IEEE, pp. 176–180.

Van Den Berg, E. and Friedlander, M. P. (2009), Probing the pareto frontier for basis pursuit solutions, *SIAM Journal on Scientific Computing* **31**(2), 890–912.

Wang, X. (2008), Method of steepest descent and its applications, *IEEE Microwave and Wireless Components Letters* **12**, 24–26.

Wen, W., Wu, C., Wang, Y., Chen, Y. and Li, H. (2016), Learning structured sparsity in deep neural networks, *arXiv preprint arXiv:1608.03665* .

Weston, J., Elisseeff, A., Schölkopf, B. and Tipping, M. (2003), Use of the zero-norm with linear models and kernel methods, *Journal of machine learning research* **3**(Mar), 1439–1461.

Wolfe, P. (1969), Convergence conditions for ascent methods, *SIAM review* **11**(2), 226–235.

Wolfe, P. (1971), Convergence conditions for ascent methods. ii: Some corrections, *SIAM review* **13**(2), 185–188.

Wright, S. J., Nowak, R. D. and Figueiredo, M. A. (2009), Sparse reconstruction by separable approximation, *IEEE Transactions on signal processing* **57**(7), 2479–2493.

Xu, J. and Zhao, Y.-B. (2020), Stability analysis of a class of sparse optimization problems, *Optimization Methods and Software* pp. 1–19.

Yamamoto, T., Yamagishi, M. and Yamada, I. (2012), Adaptive proximal forward-backward splitting for sparse system identification under impulsive noise, *in* '2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)', IEEE, pp. 2620–2624.

Yang, J., Thompson, J., Huang, X., Jin, T. and Zhou, Z. (2012), Random-frequency sar imaging based on compressed sensing, *IEEE Transactions on Geoscience and Remote Sensing* **51**(2), 983–994.

Yang, Y., Li, H. and Yu, C. (2019), Energy optimization analysis of turbofan engine for more-electric civil aircraft, *IEEE Access* **8**, 34008–34018.

Yin, W., Osher, S., Goldfarb, D. and Darbon, J. (2008), Bregman iterative algorithms for l_1-minimization with applications to compressed sensing, *SIAM Journal on Imaging sciences* **1**(1), 143–168.

Zhang, J., Zhao, C., Zhao, D. and Gao, W. (2014), Image compressive sensing recovery using adaptively learned sparsifying basis via l0 minimization, *Signal Processing* **103**, 114–126.

Zhang, T. (2009), Some sharp performance bounds for least squares regression with l1 regularization, *The Annals of Statistics* **37**(5A), 2109–2144.

Zhang, X., Lu, Y. and Chan, T. (2012), A novel sparsity reconstruction method from poisson data for 3d bioluminescence tomography, *Journal of scientific computing* **50**(3), 519–535.

Zhang, X. and Zhang, X. (2017), An accelerated proximal iterative hard thresholding method for l_0 minimization, *arXiv preprint arXiv:1709.01668* .

Zhang, Y., Dong, B. and Lu, Z. (2013), l_0 minimization for wavelet frame based image restoration, *Mathematics of Computation* **82**(282), 995–1015.

Zhao, Z., Morstatter, F., Sharma, S., Alelyani, S., Anand, A. and Liu, H. (2010), Advancing feature selection research, *ASU feature selection repository* pp. 1–28.

Zou, H. (2006), The adaptive lasso and its oracle properties, *Journal of the American statistical association* **101**(476), 1418–1429.

Zou, H., Hastie, T. and Tibshirani, R. (2006), Sparse principal component analysis, *Journal of computational and graphical statistics* **15**(2), 265–286.

Zou, H. and Li, R. (2008), One-step sparse estimates in nonconcave penalized likelihood models, *Annals of statistics* **36**(4), 1509.

APPENDIX A

RESEARCH DATA AND PYTHON CODE

For data reproducibility, we declare that our data for all the experiments in Chapter 4 can be accessible through the following links.

- Section 4.1: Profiling and Benchmarking for Spectral Proximal Method
  `https://drive.google.com/drive/folders/`
  `1oXGJ-WjDZ-MnNJDfsW0_y2SH4QorkDx5?usp=sharing`

- Section 4.2: Influences of Condition Number
  `https://drive.google.com/drive/folders/`
  `1sPgro2mzEsSStMnm-bCr_vECwYZ-cux8?usp=sharing`

- Section 4.3: Applications (MNIST)
  `https://drive.google.com/drive/folders/`
  `1G4RxMOHg1hKXxS45iiI---bVJ62QPfRh?usp=sharing`

APPENDIX B

FIGURES FOR SPARSITY OF THE SOLUTIONS (MNIST)

Results of the sparsity of the solutions with respect to the number of iterations for all labels in Section 4.3 are provided through the links below.

- $k = 500$:
  `https://drive.google.com/drive/folders/1tTIHi9a0hEOqGU5HfzEoEQNcXsjUDfYH?usp=sharing`
- $k = 1000$:
  `https://drive.google.com/drive/folders/1AyZR8VZm1GENjBDHvJqGEMJqOOeyacno?usp=sharing`
- $k = 10000$:
  `https://drive.google.com/drive/folders/1ECA0YML-FyYWRiAwK-Ru_OtWjBoc-L2r?usp=sharing`