THE PENNSYLVANIA STATE UNIVERSITY
SCHREYER HONORS COLLEGE

DEPARTMENT OF MATHEMATICS

SOLUTION OF QUADRATIC PROGRAMMING USING DYKSTRA'S ALGORITHM

XIN JIN
SPRING 2022

A thesis
submitted in partial fulfillment
of the requirements
for baccalaureate degrees
in Mathematics
with honors in Mathematics

Reviewed and approved* by the following:

Ludmil Zikatanov
Professor of Mathematics
Honors Advisor
Thesis Supervisor

Xiantao Xu
Professor of Mathematics
Faculty Reader

*Signatures are on file in the Schreyer Honors College.

Abstract

This thesis regards the study of an efficient implementation of the method of alternating projections to semi-definite quadratic programming problems. Quadratic programming problems are ubiquitous. In canonical form they correspond to minimization of a quadratic form subject to linear inequality constraints. When the form is positive definite, a solution to the quadratic programming problem gives the closest point from a polyhedral set to a given point in space. Such solutions can also be used to approximate minimizers of nonlinear functions subject to nonlinear constraints. Our research focuses on Dykstra's cyclic projections method. We implemented the Dykstra's algorithm and compared the efficiency of Dykstra's algorithm with ellipsoid method.

Table of Contents

List of Figures

Acknowledgements

I would like to express the depth of my gratitude to my thesis advisor, Dr. Ludmil Zikatanov. Throughout my undergraduate career, he offered a great amount of useful guidance and dedication to support our study. It is because of his patience and passionate that I developed such an affinity for mathematics study.

Also, I would like to thank Zhengqi Liu for his help with the implementation of the python code for the ellipsoid method.

Finally, I would like to thank my family and friends, without whom I wouldn't have experienced such an meaningful and fulfilling undergraduate times.Your support means a lot to me.

Chapter 1

Introduction

1.1 Basis For Optimization

Optimization problem exists almost everywhere in our life. For example, in economic field, the producers would endeavour to maximize their revenue as well as minimize their input of the production, also known as factors of production, at the same time. The factors of production, including land, labor and capital, are scarce so that the company only have access to a limited amount of it. In mathematical optimization problem, these factors act as constraints that limit the companies from gaining more profits. Under this condition, the goal of the company to maximize their profit, which is considered as an objective function.

In general, mathematical optimization problem has the form[2]:

$$\begin{aligned} &\text{minimize} \quad f_0(x) \\ &\quad \text{subject to} \\ &\qquad f_i(x) \leq b_i, \quad i = 1, ..., m \end{aligned} \tag{1.1}$$

Here $x \in \mathbb{R}^n$ is the *optimization variable* of the problem, the function $f_0 : \mathbf{R^n} \to \mathbf{R}, i = 1, ..., m$, is the *objective function*, the functions $f_i : \mathbf{R^n} \to \mathbf{R}, i = 1, ..., m$, are the *constraint functions*, and the constants $b_i, i = 1, ..., m$ are the bounds, or limits for the constraint functions. We consider all the vector x which satisfy all the constraints functions to be our candidate solutions, and the set of x is called *feasible region*. And the value of the objective function on the feasible region are defined as *feasible values*. A vector x^* is called an optimal solution of the problem if it has the minimum objective value on the feasible region.

We consider a linear programming problem to be a problem which has the objective and constraint functions $f_0, ..., f_m$ in the form[4]:

$$\begin{aligned} &\text{minimize} \quad f_0(x) = c_1x_1 + c_2x_2 + ... + c_nx_n \\ &\quad \text{subject to} \\ &\qquad f_i(x) = a_{i1}x_1 + a_{i2}x_2 + ... + a_{in}x_n \leq b_i, \quad i = 1, ..., s \\ &\qquad f_i(x) = a_{i1}x_1 + a_{i2}x_2 + ... + a_{in}x_n \leq b_i, \quad i = s+1, ..., m \end{aligned}$$

An important application of optimization in production is to help human decision makers find out the best parameters that yield their wanting outcome while keeping the cost as reasonable as possible. One specific example can be *device sizing*[**?**] in electronic design, where engineers use mathematical optimization to find out the optimal width and length for the electronic devices in their circus. In this example, the variables represent the size of the devices while the constraints stand for the requirement that set up by the engineer to fit in certain operation time and total size of the circuit. One popular objective function is the total electric power that are used in the circuit, and the goal of this problem is to find the most power efficient design of the circuit that meet the

engineers requirement.

When the objective function or constraint functions are not linear, it is referred to as *nonlinear programming problem*. In this thesis we mainly focus on quadratic programming on a convex set, where the objective function is a quadratic function. Unlike linear programming problem, solving this kind of problem is far more challenging and the exact solutions are sometimes unable to get. Hence utilization of numerical methods are necessary to find the best approximation of the solution.[4]

Many methods have been developed in the past few decades, including interior-point method, active set method, and ellipsoid method.

Interior point method[4] is first discovered in 1960s, and further developed in mid-1980s, and solve the problem (1.1) by applying Newton's method to a sequence of equality constrained problems. Two branches of this methods include barrier method and primal-dual interior-point method, where the former is latter abandoned due to lack of efficiency in nonlinear programming problem compared to other alternatives.

Ellipsoid method[5] was first introduced in 1972 by Shor, Nemirovsky, and Yudin. It is an iterative method which generate a sequence of ellipsoids with uniformly decreasing volume and ensured to contain the minimizer. During each step, a cutting plane is applied on the center of the ellipsoid with its sub-gradient, which helps to decide which half of the ellipsoid containing the minimizer. An ellipsoid with smaller volume is then generated to contain that half of ellipsoid and the process is repeated. Although this method is theoretically efficient, in practice it can be very slow.

In this thesis, we demonstrate the interchangeability of solving quadratic programming problem with finding the best approximation of the solution generated from a polyhedral set in a norm defined by a symmetric matrix. To achieve this, we apply a memory efficient Dykstra algorithm that uses sparse matrices.

1.2 Convex Set and Some Important Examples

Before we start to explore the the *Dykstra's cyclic projection method*, we define the convex set where the quadratic programming problem in this thesis are based on. A set C is a *convex* set[2] if the line segment between any two points in C also lies in C, i.e., for a *convex set* C, any $x_1, x_2 \in C$ and any θ with $0 \leq \theta \leq 1$ we have:

$$\theta x_1 + (1 - \theta)x_2 \in C.$$

In the figure 1.1 below we show a few simple examples of 2-dimensional convex set and non-convex sets.

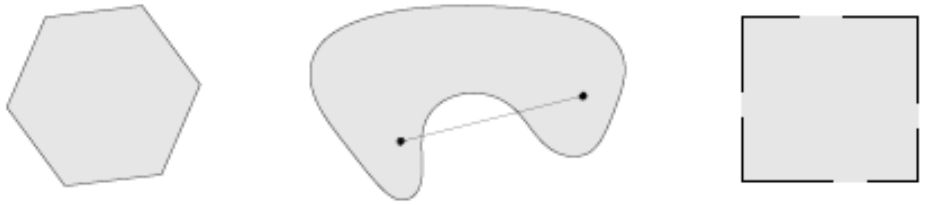


Figure 1.1: Examples of convex set and non-convex set[2]

The left one is a hexagon with closed boundary. We can see clearly it is a convex set because any two points in the set can form a line segment which also lies in the set. The middle one, therefore is obviously non-convex. The right one is a square while some part of its boundary points are missing. This is not a convex set because the line segments between two points on the same side of the square might be not completely in this square.

One important example of convex set is *hyperplanes* and *halfspaces*.[2]

Definition 1.2.1 *A hyperplane* is a set of the form

$$\{x \mid a^T x = b, \}$$

where $a \in \mathbb{R}^n$, and $b \in \mathbb{R}$.

In closer examine we can find that x is a solutions of a nontrivial linear equation, and geometrically the solution set of x can be interpreted as a hyperplane with a *normal vector* a. Two *halfspaces* are divided by one hyperplane in $\mathbb{R}^n$. Hence we get the definition of half-spaces.

Definition 1.2.2 A closed *half-space* is the solution set of the nontrivial linear inequality

$$\{x \mid a^T x \leq b\}$$

where $a \neq 0$.

Half-spaces are convex sets, which makes them a perfect place to work on in our quadratic programming problem.

An ellipsoid is also a convex set, defined as

$$\mathcal{E} = \{x \mid (x - x_c)^T p^(- 1)(x - x_c) \leq 1\}$$

where $P = P^T \geq 0$, i.e.,P is symmetric and positive definite. In the definition, the matrix P determines the size and orientation of the ellipsoid $\mathcal{E}$ while the square root of its eigenvalues represent the semi-axes of the ellipsoid. $x_c \in \mathbb{R}^n$ is the vector that represent the location of the center of the ellipsoid. Specifically, a ball is an ellipsoid with P of the form:

$$P = r^2 I$$

where I is an identity matrix.

1.3 Dykstra's Cyclic Projections Method

One general method that we will use in this thesis to find the best approximation from a convex set is Dykstra's cyclic projections method. This method is a variant of the alternating projections methods, and is theoretically powerful for computing the best approximations in a closed convex set $\mathbf{K}$, where $\mathbf{K}$ is the intersection of some finite number of closed convex sets. This method is highly efficient because it reduce the challenging problem into solving some easier smaller problems like finding the best approximation in each of the sets K_i. Here, the closed convex set K_i's can be half-spaces, hyperplanes, finite dimensional subspaces, or some cones.

Dykstra's cyclic projections method[6] was first proposed and proved by Richard L. Dykstra in the 1980s under the condition where X is Euclidean n-space[**?**] and all closed convex sets K_i's are convex cones. Its general convergence is proved later by Boyle and Dykstra in 1985. In 1997, by allowing the number of closed convex sets K_i's to be infinite and allowing the K_i's to be randomly ordered instead of the original cyclic ordered, this algorithm is developed into two directions.

Up to today, acceleration of the convergence rate of Dykstra's method is still unsolved. One method to accelerate a symmetric version of the alternating projections methods was proposed by Bauschke, Deutsch, Hundal, and Park in 2001. This method is proven to be faster, however there is no research that shows whether this method can be applied to Dykstra's method.

According to the survey conducted by Deutsch in 1992, Dykstra algorithm possesses many applications in various areas of mathematics research. Some main applications are listed: solving linear equations, probability and statistics, the Dirichlet problem, computing Bergman kernels, least change secant updates, approximating multivariate functions by sums of univariate ones, multi-grid methods, image restoration, conformal mapping and computed tomography.

In this thesis, we mainly focus on the application of Dykstra's algorithm of solving quadratic programming problems, which was first referenced by Deutsch in 1995.

Chapter 2

Quadratic Programming problems

2.1 Linear Transformation and SPD

A matrix is positive definite if it is symmetric and all its eigenvalues are positive. In this section, we first discuss the definition of symmetric and positive definite for a matrix. Here, a matrix represent a linear transformation on a given basis, although this thesis mainly focus on the numerical method instead of the linear transformation the matrix represent.

2.1.1 Symmetry

Here we define symmetry with a linear transformation A on basis V and a scalar product $(.,.)$.

Definition 2.1.1 A matrix A is symmetric in (.,.) *if and only if*

$$(Ax, y) = (x, Ay), \text{ for all } x, y \in V$$

Given a basis, we can write out the linear transformations on this basis by a matrix. Furthermore, given a representation, we can calculate the action of linear transformation on any vector.

2.1.2 Definitness

Definition 2.1.2 *A linear transformation is positive definite if and only if* $(Ax, x) > 0$ *for any nonzero* x *in* V*. Similarly, it is negative definite if and only if* $(Ax, x) < 0$*.*

Definition 2.1.3 *A linear transformation is semi-positive definite if and only if* $(Ax, x) \geq 0$ *for any nonzero* x *in* V*. Similarly, it is semi-negative definite if and only if* $(Ax, x) \leq 0$*.*

Alongside with these definition, we have indefinite linear transformation defined as symmetric matrix that are neither positive definite nor negative definite. In this thesis we mainly focus on SPD transformations.

2.1.3 SPD matrices

Definition 2.1.4 *A symmetric matrix* A *is defined as positive definite if either the following conditions hold:*

- *All eigenvalues of* A *are positive.*
- $\langle Ax, x\rangle > 0$ *for any* $x \in \mathbf{R}^n, \quad x \neq 0$

Note that in second condition:

$$(x, y)_A := \langle Ax, y\rangle , \quad \text{and} \quad \|x\|_A := \sqrt{\langle Ax, y\rangle}$$

And we have the property:

$$(A^{-1}x, y)_A = (x, A^{-1}y)_A = \langle x, y\rangle$$

This property can be easily shown using the symmetry of matrix A

$$\langle Ax, y\rangle = x^T A^T y = x^T Ay = \langle x, Ay\rangle$$

And furthermore we have:

$$(A^{-1}x, y)_A = (x, A^{-1}y)_A = \langle x, y\rangle$$

2.2 The Quadratic programming problem

In this thesis, we write quadratic programming problem as following: Find x^* such that

$$x^* = \arg\min_{x\in\mathbf{R}^n} f(x), \quad \text{where} \quad f(x) = \frac{1}{2}\langle Ax, x\rangle - \langle b, x\rangle , \quad x \geq c \tag{2.1}$$

Then we prove that by finding the best approximation on a convex polyhedral set we can find the solution to the quadratic programming problem.

Theorem 2.2.1 *Let x^* be the solution that solves the QPP, then x^* also solves the problem:*

$$x_* = \arg\min_{x\in\mathbb{R}^n} \|x - d\|_A^2, \quad x \geq c \tag{2.2}$$

Proof:

$$\begin{aligned}
f(x) &= \frac{1}{2}\langle Ax, x\rangle - \langle b, x\rangle \\
&= \frac{1}{2}[\langle Ax, x\rangle - 2\langle b, x\rangle] \\
&= \frac{1}{2}[\|x\|_A^2 - 2(A^{-1}b, x)_A] \\
&= \frac{1}{2}[\|x\|_A^2 - 2(A^{-1}b, x)_A + \|A^{-1}b\|_A^2 - \|A^{-1}b\|_A^2] \\
&= \frac{1}{2}\|x - A^{-1}b\|_A^2 - \frac{1}{2}\|A^{-1}b\|_A^2 \\
&= \frac{1}{2}\|x - d\|_A^2 - \frac{1}{2}\|d\|_A^2
\end{aligned}$$

Note that $(\cdot,\cdot)_A$ is linear with respect to both arguments. And in real vector space $(x,y)_A = (y,x)_A$. So we get the following proof for the 5th step:

$$\begin{aligned}\|x - A^{-1}b\|_A^2 &= (x - A^{-1}b, x - A^{-1}b)_A \\ &= (x, x - A^{-1}b)_A - (A^{-1}b, x - A^{-1}b)_A \\ &= (x,x)_A - (x, A^{-1}b)_A - (x, A^{-1}b)_A + (A^{-1}b, A^{-1}b)_A \\ &= \|x\|_A^2 - 2(x, A^{-1}b)_A + \|A^{-1}b\|_A^2\end{aligned}$$

Since $A^{-1}b$ is a constant with respect to x, we can conclude that: If x_* is a solution of (2.1), then x_* also solves the problem:

$$x_* = \arg\min_{x\in\mathbb{R}^n} \|x - d\|_A^2, \quad x \geq c \tag{2.3}$$

Using the result we get from (2.1) we can also get the following corollary:
Solving (2.1) is equivalent to finding x_* such that:

$$x_* = y_* + c, \quad where \quad y_* = \arg\min_{y\in\mathbb{R}^n} \|y - w\|_A^2, \quad y \geq 0 \tag{2.4}$$

And it is easily proved by letting $y = x - c, \quad w = d - c$, then

$$\|x - d\|_A^2 = \|x - c + c - d\|_A^2 = \|y - w\|_A^2 \tag{2.5}$$

After this process, the QQP (2.1) has been reduced to finding the unique element from the cone $y \geq 0$ which is closest to w in the length defined by A, where $w = d - c = (A^{-1}b - c)$.

2.3 Karush-Kuhn-Tucker Conditions

Here we introduce Karush-Kuhn-Tucker (KKT) Condition and the linear constrained problem (LCP) where they are applied to.
The linear constrained problem is defined as:
Finding the vectors $x, r \in \mathbb{R}^n$ such that

$$Ax - r = b, \quad x \geq c, \quad , r \geq 0, \quad \langle r, (x - c)\rangle = 0$$

Here, $A \in \mathbb{R}^{n\times n}$ is the matrix representation of the coefficient in the constraints, $b \in \mathbb{R}^n$ is the vector of constant terms in the constraint functions, $c \in \mathbb{R}^n$, and $r \in \mathbb{R}^n$ is a vector of slack variables that can be easily eliminated. Then, we consider the following constrained minimization

problem. $i \in \mathcal{I}$ and $j \in \mathcal{E}$ are indices of inequality or equality constraints.

$$\begin{aligned} &\min F(x) \\ &g_i(x) \geq 0 \quad for \quad i \in J, \\ &g_j(x) = 0 \quad for \quad j \in E \end{aligned}$$

To solve the minimization problem above, the following KKT conditions are are necessary:

$$\begin{aligned} \nabla_x[F(x) - \langle r, g(x)\rangle] &= 0, \\ r_k g_k(x) &= 0, \quad k \in \mathcal{E} \cup \mathcal{I} \\ g_i(x) \geq 0, \quad i \in \mathcal{I}, \quad g_j(x) &= 0, \quad j \in \mathcal{E} \end{aligned}$$

The last condition is complementary, which implies that either constraint i is active, $r_i^* = 0$, or both. The inactive inequalities constraints has Lagrange multiplier r_i equal to zero[7], so we can omit the terms whose indices $i \notin A(x^*)$ from the first condition and rewrite the KKT conditions as the following:

$$0 = \delta_x \mathcal{L}(x^*, r^*) = \delta F(x^*) - \sum_{i \in A(x^*)} r_i^* g_i^*(x^*)$$

From the background knowledge above we can then come to our main result.

Lemma 2.3.1 *With the KKT condition the QPP given in (2.1): Find x^* such that*

$$x_* = \arg\min_{x \in \mathbb{R}^n} f(x), \quad where \quad f(x) = \frac{1}{2}\langle Ax, x\rangle - \langle b, x\rangle, x \geq c$$

is equivalent to a linear constrained problem.

Proof: We apply the KKT conditions to (2.1)

$$\begin{aligned} F(x) &= f(x) \\ g_i(x) &= x_i - c_i, \quad i = 1, \ldots, n \\ \mathcal{E} &= \emptyset, \\ \nabla_x f(x) &= b - Ax \end{aligned}$$

Then, the optimization problem have the following KKT conditions:

$$Ax - b - r = 0, \quad or \quad Ax - r = b,$$

$$\mathrm{diag}(r)(x - c) = 0, \quad r \geq 0, \quad x - c \geq 0$$

$\mathrm{diag}(z)$ is a diagonal matrix with elements $z_1, \ldots, z_n$ on the diagonal and all other elements are

zero. In the second condition $r \geq 0, x - c \geq 0$, then we have:

$$\langle r, x - c \rangle = \sum_{i=1}^{n} r_i(x_i - c_i) = 0 \quad is \quad equavalent \quad to \quad r_k(x_k - c_k) = 0, \quad k = 1, \ldots, n$$

Hence, QPP (2.1) is equivalent to finding (x, r) such that $Ax - r = b, \langle r, x - c \rangle = 0, r \geq 0, x - c \geq 0$. And this is the claim of lemma 2.3.1. In this proof we have shown that KKT conditions for the QPP are the linear constrained problem. And using change of variable $y = x - c$, we can get the formulation: Find (x, r) such that

$$Ay - r = b - Ac, \quad \langle r, y \rangle = 0, r \geq 0, \quad y \geq 0.$$

Chapter 3

Dykstra's Algorithm in QPP

3.1 Dykstra's method

3.1.1 Convert QPP to Best Approximation Problem

From now on, we consider the matrix A in which our QPP is defined to be a SPD matrix, then we rewrite $\langle b, x\rangle$ as

$$\langle b, x\rangle = \langle x, b\rangle = \left\langle x, A(A^{-1}b)\right\rangle \tag{3.1}$$

Then we substitute (3.1) in QPP (2.1) and get $f = \frac{1}{2}\langle Ax, x\rangle - \langle x, A(A^{-1}b)\rangle$. And a new function $\tilde{f}$ is defined as[1]:

$$\tilde{f} = \frac{1}{2}\langle Ax, x\rangle - \left\langle x, A^{-1}b\right\rangle + \frac{1}{2}\left\langle A^{-1}b, A^{-1}b\right\rangle_A \tag{3.2}$$

By observation, we find that the minimizer of $\tilde{f}$ is the same as the minimizer of f, so we claim that

$$x^* = \arg\min_x f(x) = \arg\min_x \tilde{f}$$

Proof:

$$\begin{aligned}\tilde{f} &= \frac{1}{2}\langle Ax, x\rangle - \left\langle x, A^{-1}b\right\rangle + \frac{1}{2}\left\langle A^{-1}b, A^{-1}b\right\rangle_A \\ &= \frac{1}{2}\|x - A^{-1}b\|_A^2 \\ &= f(x) + \frac{1}{2}\|A^{-1}b\|^2\end{aligned} \tag{3.3}$$

Since $\tilde{f} - f = \frac{1}{2}\|A^{-1}b\|^2$, which is independent of x, we have shown that the minimizer of f and the minimizer of $\tilde{f}$ have the same value. And with this proven, we can restate our QPP as the following form:

$$x^* = \arg\min_{x\geq c} \|x - A^{-1}b\|_A^2 \tag{3.4}$$

This equation means finding the best approximation in A-norm to the vector $A^{-1}b$ from a convex set, and hence we can apply Dysktra Algorithm to it.

3.1.2 Dykstra Algorithm

Let K is the intersection of K_i, where K_i are closed convex subsets of Hilbert space $X(i = 1, 2, ..., r)$. Assume $K \neq 0$. For $x \in X$, we need to compute $P_K(x)$ which is the projection of x on K.
Let j be natural numbers, and $[j]$ denote "$j \mod r$",[4] i.e.,

$$[j] := \{1, 2, ..., r\} \cap \{j - kr \mid k = 0, 1, ...\}$$

Hence, $[1] = 1, [2] = 2, ..., [r] = r, [r + 1] = 1, [r + 2] = 2, ..., [2r] = r, ...$

The Dykstra algorithm proceed as follows:
For a fixed $x \in X$
Initialize :

$$\begin{gathered} x_0 = x \\ e_{-(r-1)} = ... = e_{-1} = e_0 = 0 \\ \text{For} \quad j = 1, 2, ... \\ \textbf{Repeat} : \\ x_j = P_{K_j}(x_{j-1} + e_{j-r}) \end{gathered} \tag{3.5}$$

$$\begin{aligned} e_j &= x_{j-1} + e_{j-r} - x_j \\ &= x_{j-1} + e_{j-r} - P_{K_{[j]}}(x_{j-1} + e_{j-r}) \quad (j = 1, 2, \ldots) \end{aligned} \tag{3.6}$$

From (3.6) it is clear that the algorithm can be rewritten by letting $w_j = x_{j-1} + e_{j-r}$. The new equation we get is $e_j = w_i - P_{K_{[j]}}(w_j)$ and $x_j = w_j - e_j$. (3.7)

3.1.3 Two-Dimensional Example with Two Half-Spaces

In this section we will discuss an example of two-dimension cases with K_i are two half-spaces.

$$K = \bigcap_{i=1}^{2} K_i$$

$$K_1 : x_1 \geq a_1 t + c_1, \quad K_2 : x_2 \geq a_2 t + c_2$$

Here $x_1 \geq a_1 t + c_1$ and $K_2 : x_2 \geq a_2 t + c_2$ are two constraints for the QPP.
For a given $x = (x_1, x_2)$, $P_{K_{[j]}}(x)$ is the closest point from point x to the subset K_j in the $\|.\|_A$ norm. And here we take $A = I$.
Now we apply the Dykstra Algorithm to this example.

$$e_1 = w_1 - P_{K_{[1]}}(w_1) = x_0 - P_{K_{[1]}}(w_1) \quad x_1 = w_1 - e_1 = P_{K_{[1]}}(w_1) \tag{3.7}$$

$$e_2 = w_2 - P_{K_{[2]}}(w_2) = x_1 - P_{K_{[2]}}(w_2) \quad x_2 = w_2 - e_2 = P_{K_{[2]}}(w_2) \tag{3.8}$$

3.2 Example on Intersection of Half-Spaces

From the former theorems, we have concluded that the solution of a QPP problem is the same as the solution of a problem of finding the best approximation. Hence, we apply the Dykstra cyclic projection method to the best approximation problem which in the end give us the result of the QPP problem. Under the particular condition where K_i are half-spaces, we write the constraints $x_i > c_i$ to be

$$x \in \bigcap_{j=1}^{n} K_j \quad K_j = \{y = \mathbf{R}^n \mid y_j \geq c_j\}$$

Hence, given a vector $x \in \mathbf{R}^n$, we first find the projection from x to K_1, that is:
Find $y_1 \in K_1$ such that $\|y_1 - x\|_A$ is minimized.
Here we use an example where

- $x > c, \quad c \in \mathbf{R}^n$

- Either $x \in K_1$, which means $y_1 = x$; or $x \neq K_1$, which means the first coordinate of $y_1 = c_1$

We get the following figure (3.1) which illustrate our example.

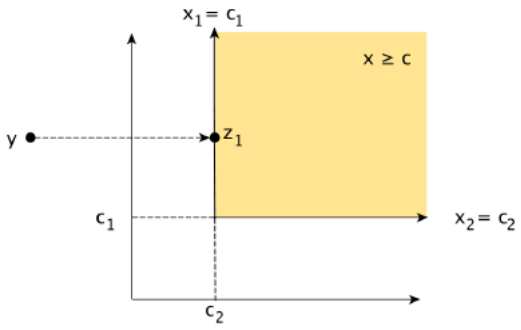


Figure 3.1: Best Approximation from a Half-Space $x_i \geq c_i; \quad A = I$[4]

In general case, we let the half-spaces for a given x to be:

$$K_j = \{y \in \mathbf{R} \mid (h_j, y) \leq c_j, \quad j = 1, ..., r\} \tag{3.9}$$

And the QPP problems is represented as finding a point $p \in \bigcap_{j=1}^{r} K_j$ such that

$$p = \arg\min\{\|x - q\|_A \mid \text{subject to } Hq \leq c\} \tag{3.10}$$

$H \in \mathbf{R^{n \times n}}$ here denote the matrix whose rows are given by the vector $h_j \in \mathbf{R}^n$, which determine the subspaces $K_j, \quad j = 1, ..., r$ In the former example, the constraints correspond to the case where $H = -I$ and $r = n$.[1]

Chapter 4

Ellipsoid Method and Implementation Comparison

4.1 Ellipsoid Method

In this section we mainly introduce an algorithm that is also used to solve quadratic programming problems, that is, the ellipsoid method. Now, let's consider a convex function $f : \mathbf{R}^n \to \mathbf{R}$. To minimize this function, we generate a sequence of ellipsoids $\mathcal{E}^{(k)} \in \mathbf{R}^n$ with decreasing volume and are guaranteed to contain a minimizing point of the function. To do this, we need to introduce sub-gradient.

Definition 4.1.1 *Vector $g^{(k)}$ is called sub-gradient at point $x^{(k)}$ if for any $x \in \mathcal{E}$*

$$f(x) - f(x^{(k)}) \geq g^{(k)}(x - x^{(k)})$$

And the set of all sub-gradient at $x^{(k)}$ is defined as $\partial f(x^{(k)})$.

Utilizing sub-gradient we can find a half-space that is guaranteed not to contain any of the minimizing point of the function, i.e., the cutting plane.
Suppose we have an ellipsoid $\mathcal{E}^{(k)}$ that is guaranteed to contain a minimizer of the function, the next step is for us to compute the sub-gradient $g^{(k}$ of the center of the ellipsoid $x^{(k)}$, then we can get a half ellipsoid

$$\mathcal{E}^{(k)} \cap \{z \mid g^{(k)T}(z - x^{(k)}) \leq 0\}$$

and this half ellipsoid must contains the minimizer of the function f. Now, we calculate the ellipsoid $\mathcal{E}$ with the minimum volume that contains every elements of the half ellipsoid. Hence, the new ellipsoid $\mathcal{E}^{(k)}$ is guaranteed to contain a minimizer of f as shown in the figure below.

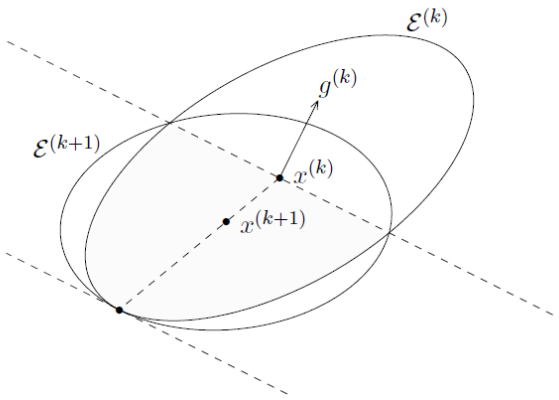


Figure 4.1: The right half ellipsoid contain a minimizer, and is enclosed by $\mathcal{E}^{(k+1)}$ which has the smallest volume and centered at $x^{(k+1)}$.[5]

The process described above is repeated. And due to the decreasing volume that the sequence of ellipsoid has, this method converges with a theoretically good complexity.
Now, we need to describe the algorithm more explicitly.

Definition 4.1.2 *An ellipsoid $\mathcal{E}^{(k)}$ can be written as*

$$\mathcal{E}^{(k)} = \{z \mid (z-x)^T P^{-1}(z-x) \leq 1\}.$$

[5] where $P \in S^n_{++}$ is the matrix that represent the size, shape and how tilted the ellipsoid is, and x represent the center of the ellipsoid. Furthermore, the square root of eigenvalues of P determines the lengths of the semi-axes of the ellipsoid, and the volume is given by:

$$\mathbf{Vol}(\mathcal{E}) = \beta_n \sqrt{\det P}$$

Here, β_n is defined as

$$\beta_n = \pi^{n/2}/\Gamma(n/2+1)$$

For $n > 1$, we have the half ellipsoid given by

$$\{z \mid (z-x)^T P^{-1}(z-x) \leq 1, g^T(z-x) \leq 0\}$$

And the ellipsoid with the minimum volume is

$$\mathcal{E}^+ = \{z \mid (z-x^+)^T (P^+)^{-1}(z-x^+) \leq 1\}$$

Here,

$$x^+ = x - \frac{1}{n+1} P\tilde{g}$$

is the new center of the new ellipsoid.

$$P^+ = \frac{n^2}{n^2-1}(P - \frac{2}{n+1} P\tilde{g}\tilde{g}^T P)$$

and

$$\tilde{g} = \frac{1}{g^T P g} g$$

is a normalized g.

4.2 Implementation of Ellipsoid Method

In this section, we state the basic ellipsoid algorithm that can be implemented to solve optimization problem with constraint.

Ellipsoid method [5]

Given an initial ellipsoid $\mathcal{E}^{(0)} = (P^{(0)}, x^{(0)})$ which contains feasible minimizers of f.

$k := 0$.

Repeat

If $f_i(x^{(k)}) \geq 0$ for some i, which means $x^{(k)}$ is feasible.

Compute the sub-gradient:

$g^{(k)} \in \partial f(x^{(k)})$.

Compute the normalization of the sub-gradient:

$\tilde{g} := \frac{1}{\sqrt{[g^{(k)}]^T P^{(k)} g^{(k)}}} g^{(k)}$.

$\alpha := \frac{f_i(x^{(k)})}{\sqrt{[g^{(k)}]^T P^{(k)} g^{(k)}}}$

Else, when $x^{(k)}$ is not feasible

Compute the sub-gradient:

$g^{(k)} \in \partial f(x^{(k)})$.

Compute the normalization of the sub-gradient:

$\tilde{g} := \frac{1}{\sqrt{[g^{(k)}]^T P^{(k)} g^{(k)}}} g^{(k)}$.

Update upper bound $f_{best}^{(k)}$

$\alpha := \frac{f(x^{(k)}) - f_{best}^{(k)}}{\sqrt{[g^{(k)}]^T P^{(k)} g^{(k)}}}$

Update the new center of the ellipsoid:

$x_{(k+1)} := x^{(k)} - \frac{1+n\alpha}{n+1} P^{(k)} \tilde{g}^{(k)}$

Update the matrix representation of the new ellipsoid:

$P^{(k+1)} := \frac{n^2}{n^2-1}(1-\alpha^2)(P^{(k)} - \frac{2(1+n\alpha)}{(n+1)(1+\alpha)} P^{(k)} \tilde{g} \tilde{g}^T P^{(k)})$

$k := k + 1$

Next, we prove the stopping criteria of ellipsoid method.

We keep record of the best point found, defined as

$$f_{best}^* = \min_{i=0,\ldots,k} f(x^{(j)})$$

And let $f^* = \lim_{k\to\infty} f_{best}^{(k)}$, the simple stopping criteria is

$$\sqrt{[g^{(k)}]^T P^{(k)} g^{(k)}} \leq \epsilon$$

Proof

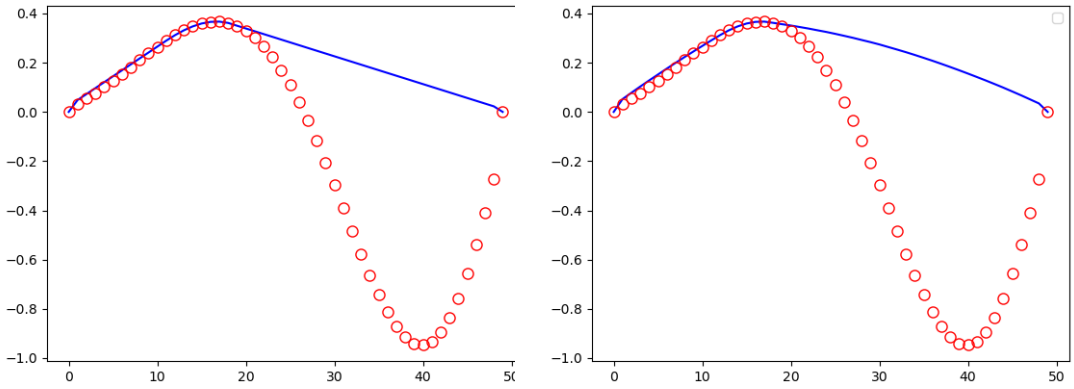
We have that for minimizer $x^* \in \mathcal{E}$

$$f^* = f(x^*) \geq f(x^{(k)}) + [g^{(k)}]^T (x^* - x^{(k)})$$

Hence,

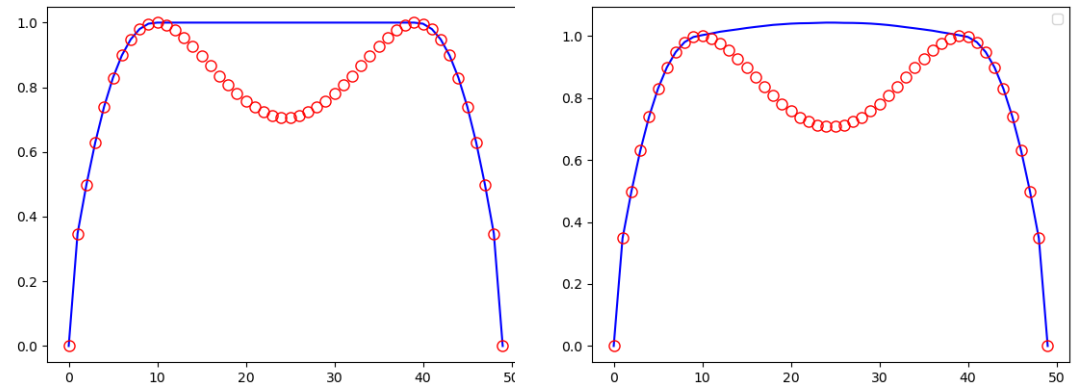
$$\begin{aligned} f(x)^{(k)} - f^* &\leq -g^{(k)T}(x^* - x^{(k)}) \\ &\leq \max_{x \in \mathcal{E}^{(k)}} -g^{(k)T}(z - x^{(k)} \\ &= \sqrt{[g^{(k)}]^T P^{(k)} g^{(k)}} \end{aligned}$$

4.3 Dykstra Algorithm and Ellipsoid Algorithm Comparison



(a) Result from Dykstra Algorithm (b) Result from Ellipsoid Method

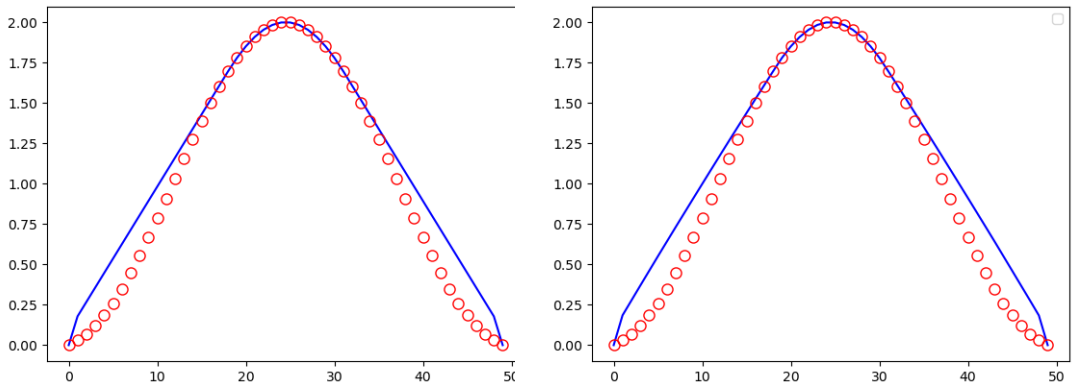
Figure 4.2: Example 1: $g(x) = \cos(5x^2) - 3\sin(\frac{5x^2(x-1}{4}) - (\cos(5) - 1)x - 1$



(a) Result from Dykstra Algorithm (b) Result from Ellipsoid Method

Figure 4.3: Example 2: $g(x) = \sin(3\pi x(1-x))$

When running both algorithms using the three examples above, ellipsoid method takes significantly larger iterations to simulate a similar result compared to Dykstra Algorithm. Hence we can reach an conclusion that under the conditions provided in this thesis where the QPP is represented with a SPD matrix, Dykstra is more efficient finding the solution of wanted Quadratic programming problems compared to Ellipsoid method.



(a) Result from Dykstra Algorithm

(b) Result from Ellipsoid Method

Figure 4.4: Example 3: $g(x) = \cos(\pi(2x-1)) + 1$

Chapter 5

Conclusion

In conclusion, in this thesis we studied the quadratic programming problem with SPD linear transformation, and proved that this kind of QPP is equivalent to finding the best approximation from a convex set. Therefore, we applied Dykstra's cyclic projection method to solving the QPP which is later proven quite efficient. Meanwhile, we also studied the classic ellipsoid method and compared the two algorithms by its running iteration. However, this thesis is limited by only discussed SPD matrix. In the future the we plan to explore the algorithm used to solve problems based on indefinite matrices.

Bibliography

[1] Frank Deutsch. Best approximation in inner product spaces. *CMS Books in Mathematics/Ouvrages de Math´ematiques de la SMC*, 7:14, Dec 2001.

[2] Stephen Boyd and Lieven Vandenberghe. Convex optimization. 7:21, Dec 2004.

[3] Richard L. Dykstra. An algorithm for restricted least squares regression. *Journal of the American Statistical Association*, 78:14, Dec 1983.

[4] Kristin Sickau. Efficient cyclic projection algorithms for quadratic progrmming problems, April 2020. Honors Thesis, Schreyer Honors College, The Pennsylvania State University.

[5] Stephen Boyd and Craig Barratt. Linear controller design: Limits of performance, 1991. Prentice-Hall Inc.

[6] Richard L. Dykstra. An algorithm for restricted least squares regression, 1991. J. Amer. Statist.Assoc.,78(384):837–842, 1983.

[7] Kazufumi Ito and Karl Kunisch. Lagrange multiplier approach to variational problems and applications, 2008. volume 15 of Advances in Design and Control. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA,.

[8] Jorge Nocedal and Stephen J. Wright. Numerical optimization., 2006. Springer Series in Operations Research and Financial Engineering. Springer, New York, second edition, 2006.

Xin Jin

Education Background

Pennsylvania State University — PA., U.S.

Bachelor of Science, Double Major in Computational Math and Data Science; Minor in statistics — Aug. 2019-May, 2022

Nanjing University of Information Science and Technology — Nanjing, China

Major in Math and Applied Math — Sept. 2017-June. 2019

Academic Research

Schreyer Honor College Honor Thesis: Solution of Indefinite Problems in Quadratic Programming — **PA., U.S.**

Advisor: Ludmil Zikatanov — April.2020-April.2021

- Completed an undergraduate honors thesis to test, compare and design novel approaches for the solution of indefinite quadratic programming problems
- Retrieved related information to study the theoretical foundation and application of ellipsoid method, interior point method and Dykstra algorithm to solve the proposed problems

Data-based Project: Building Hospital Database — **PA., U.S.**

Advisor: Dan Richert — Aug.2021-Dec.2021

- Brainstormed ideas about the structure of hospital database, and visualized database with entity relationship diagram
- Built the database covering physical database design, data modeling, relational model, logical database design by applying SQL query language

Professional Experiences

Inspur — **Shandong, China**

Intern Algorithm Engineer — May 2021- Aug. 2021

- Researched on the relevant contents of RASA conversational AI solutions and learned its structure and method of language acquisition, tested the main components of RASA
- Retrieved and learned the Baidu ERNIE papers
- Summarized the report of hyperspectral anti-counterfeiting investigation to support project leader's work

Honor & Awards

Merit Scholarship sponsored by Reading Academy, Nanjing University of Information Science and Technology — Feb. 2019

Excellent School Student Leader — Feb. 2019

Second Prize of Cheer-leading Group Contest — Oct. 2019

Scholarship sponsored by Nanjing University of Information Science and Technology — Apr. 2018

Leadership Experiences

Student Union President, Reading Academy, Nanjing University of Information Science and Technology — **Sept. 2017-Jul. 2019**

- Planned and organized school events, including orientation ceremony, Mid-Autumn Party, sports meeting, environment and art festival, etc
- Proposed and launched the proposal of study community to provide opportunity for peer students to learn each other by organizing major courses workshops, English corners, relevant academic seminars

Secretary General, Chinese Theater and Movie Society, Pennsylvania State University — **Sept. 2019**

Recruited club members, cast for dramas, organized script discussions and rehearsal

Coordinated and scheduled the performance arrangements, communicated with the author of drama to get the authorization

Other Information

Language Proficiency: Mandarin (Native), English (Proficient)

Software Skills: R Studio, Python, SQL, MATLAB