

## ABSTRACT

Title of dissertation: Adaptive Constraint Reduction for  
Convex Quadratic Programming and  
Training Support Vector Machines

Jin Hyuk Jung, Doctor of Philosophy, 2008

Dissertation directed by: Professor Dianne P. O'Leary  
Department of Computer Science

Convex quadratic programming (CQP) is an optimization problem of minimizing a convex quadratic objective function subject to linear constraints. We propose an adaptive constraint reduction primal-dual interior-point algorithm for convex quadratic programming with many more constraints than variables. We reduce the computational effort by assembling the normal equation matrix with a subset of the constraints. Instead of the exact matrix, we compute an approximate matrix for a well chosen index set which includes indices of constraints that seem to be most critical. Starting with a large portion of the constraints, our proposed scheme excludes more unnecessary constraints at later iterations. We provide proofs for the global convergence and the quadratic local convergence rate of an affine scaling variant. A similar approach can be applied to Mehrotra's predictor-corrector type algorithms.

An example of CQP arises in training a linear support vector machine (SVM), which is a popular tool for pattern recognition. The difficulty in training a support

vector machine (SVM) lies in the typically vast number of patterns used for the training process. In this work, we propose an adaptive constraint reduction primal-dual interior-point method for training the linear SVM with  $l_1$  hinge loss. We reduce the computational effort by assembling the normal equation matrix with a subset of well-chosen patterns. Starting with a large portion of the patterns, our proposed scheme excludes more and more unnecessary patterns as the iteration proceeds. We extend our approach to training nonlinear SVMs through Gram matrix approximation methods. Promising numerical results are reported.

# Adaptive Constraint Reduction for Convex Quadratic Programming and Training Support Vector Machines

by

Jin Hyuk Jung

Dissertation submitted to the Faculty of the Graduate School of the  
University of Maryland, College Park in partial fulfillment  
of the requirements for the degree of  
Doctor of Philosophy  
2008

Advisory Committee:

Professor Dianne P. O'Leary, Chair/Advisor  
Professor Kyu Yong Choi  
Professor Hanan Samet  
Professor G. W. Stewart  
Professor André L. Tits

© Copyright by  
Jin Hyuk Jung  
2008

## Acknowledgements

I am very pleased to acknowledge individuals and institutions for their support during my research.

First of all, I owe my deepest appreciation to my academic advisor, Dianne P. O'Leary. Not only did she give encouragement, advice and support to me, but she also showed me the attitudes that a researcher, a teacher and a supervisor should have. Working with her has been a true privilege and a very special experience to me. Without her help, this thesis could not be completed.

I am in debt to André L. Tits for his time, advice and guidance. He has been virtually a co-advisor of my research.

Other committee members, Kyu Yong Choi, Hanan Samet and G. W. Stewart, deserve many thanks. They pleasantly accepted the tight examination schedule. They also gave plentiful advice and inspiring questions.

I would like to thank my financial supporters. I thank the IITA (Institute for Information Technology Advancement) and MIC (Ministry of Information and Communication), Republic of Korea, for an Information Technology Scholarship. I thank Jae-Kwang Bae for his personal financial support. I am very grateful to the US Department of Energy for supporting my research under grant DEFG0204ER25655.

I would like to thank my UMD (University of Maryland) friends and St. Andrew Kim Catholic Church friends. From the time when I struggled in settling my family

in Maryland, they have been an infinite source of support and joy.

Finally, my gratitude to my family members, especially my wife, Eun A Lee, is limitless. Their wordless sacrifice, care, support and love cannot be repaid.

# Table of Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Adaptive Constraint Reduction for Convex Quadratic Programming</b>	<b>5</b>
2.1	Related Work . . . . .	7
2.2	Solving Convex Quadratic Programming . . . . .	10
2.2.1	Standard Form . . . . .	10
2.2.2	Duality . . . . .	12
2.2.3	Optimality Conditions . . . . .	13
2.2.4	Primal-Dual Interior-Point Method . . . . .	17
2.2.5	Central Path . . . . .	19
2.2.6	Primal-Dual Path-Following Interior-Point Method . . . . .	21
2.2.7	Primal-Feasible Affine-Scaling PDIPM . . . . .	23
2.3	Adaptive Constraint Reduction . . . . .	25
2.3.1	Adaptive Constraint-Reduction for Primal-Feasible Affine-Scaling PDIPM . . . . .	25
2.3.2	Convergence of the Adaptive Constraint-Reduction Algorithm	31
2.4	Extension to Infeasible Problems . . . . .	33
2.5	Numerical Results . . . . .	46
2.5.1	Choosing $Q$ . . . . .	47

2.5.2	Scaling . . . . .	47
2.5.3	Data Fitting . . . . .	49
2.5.4	Random Problems . . . . .	55
2.6	Conclusions . . . . .	56
<b>3</b>	<b>Adaptive Constraint Reduction for Training Support Vector Machines</b>	<b>57</b>
3.1	Introduction to Support Vector Machines . . . . .	58
3.1.1	Data Representation, Classifier, Feature Space, and Kernel . . . . .	58
3.1.2	Separation Margin Maximization . . . . .	62
3.1.3	Dual Formulation and Support Vector . . . . .	65
3.2	Related Work . . . . .	67
3.3	Adaptive Constraint Reduction . . . . .	69
3.3.1	Primal-Dual Interior-Point Method . . . . .	69
3.3.2	Constraint Reduction . . . . .	73
3.3.3	Differences from Algorithm 1 . . . . .	81
3.3.4	Kernelization . . . . .	82
3.4	Numerical Results . . . . .	84
3.4.1	Linear SVM Examples . . . . .	85
3.4.2	Nonlinear SVM Examples . . . . .	92
3.4.3	Visualization of the Iterations . . . . .	96
3.5	Conclusion . . . . .	97
<b>4</b>	<b>Conclusions and Future Directions</b>	<b>102</b>
<b>A</b>	<b>Geometric Properties of CQP</b>	<b>107</b>

A.1	Definitions . . . . .	107
A.2	Properties of Convex Functions . . . . .	110
A.3	Geometric Properties of the Solution Set $\mathcal{F}_P^*$ . . . . .	111
<b>B</b>	<b>Convergence Proof for Constraint Reduced Affine-Scaling PDIPM</b>	<b>122</b>
B.1	Global Convergence Proof . . . . .	122
B.2	Local Rate of Convergence . . . . .	142
	<b>Bibliography</b>	<b>158</b>

## LIST OF FIGURES

2.1	Maximizing satisfaction. The filled area is the feasible region in which all the constraints (nutrient requirements) are satisfied. The dotted circles are contours of the objective function. The objective function (his satisfaction) is maximized on the feasible region at the point marked with a circle. . . . .	7
2.2	$i^{th}$ and $j^{th}$ constraints are active at the solution $\mathbf{x}^*$ . The gradient of the objective function $f(\mathbf{x})$ at $\mathbf{x}^*$ can be expressed by the nonnegative combination of $\mathbf{a}_i$ and $\mathbf{a}_j$ . . . . .	15
2.3	Two dimensional CQP examples . . . . .	25
2.4	Measured signal fit by various methods. Tested with $\bar{m} = 5000$ and $\bar{n} = 99$ . . . . .	52
2.5	Adaptive reduction is compared with nonadaptive reduction on the data-fitting problem (2.77). The horizontal axis is in log scale. Tested with $\bar{m} := 20000$ and $\bar{n} := 99$ . . . . .	54
2.6	Adaptive reduction is compared with nonadaptive reduction on the fully random problem (2.77). The horizontal axis is in log scale. . . .	55

3.1	By mapping the patterns in the input space $(x_1, x_2)$ to a higher dimensional feature space $(x_1^2, x_2^2, \sqrt{2}x_1x_2)$ , the SVM can find an ellipsoidal classifier in the original input space by finding a linear classifier in the feature space. . . . .	60
3.2	The learning machine is trained to find a hyperplane with maximal separation margin. The hyperplane can classify data according to the predetermined labels. Circles and squares denote positive and negative patterns, respectively. . . . .	63
3.3	Time and iteration count of adaptive reduction with balanced selection are compared to non-reduction algorithm. $q_U$ is set to $m$ (100%) for all cases. . . . .	88
3.4	The adaptive balanced and adaptive nonbalanced algorithms are compared, with the constraint choice based on $\Omega\mathbf{e}$ . . . . .	89
3.5	Letter. The adaptive and nonadaptive balanced algorithms are compared, with the constraint choice based on $\Omega\mathbf{e}$ . . . . .	90
3.6	The two constraint choices are applied to the adaptive balanced reduction. . . . .	91
3.7	Letter. Adaptive balanced reduction based on $\Omega\mathbf{e}$ constraint choice. . . . .	92
3.8	Timing results of algorithms for linear SVM training on adult data sets. . . . .	93
3.9	Gram matrix approximation on adult data sets . . . . .	99
3.10	Nonlinear SVM training on adult data sets . . . . .	100

3.11 Snapshots of finding a classifier using the adaptive reduction algorithm for a randomly generated toy problem in 2-dimensional input space with patterns eliminated intentionally around a hand-generated ellipsoid centered at the origin. The mapping associated with the second order homogeneous polynomial kernel is used to find the surface. The numbers below each figure indicate (iteration)/(number of patterns used).101

A.1 Geometries of the feasible set and contours of the objective function in a 2 dimensional space. Examples of unboundedness of the primal feasible set, the level set  $\{\mathbf{x} : f(\mathbf{x}) \leq f(\bar{\mathbf{x}})\}$  of the objective function, and their intersection  $\mathcal{F}_P^L(\bar{\mathbf{x}})$ . . . . . 117

B.1 Diagram for global convergence proof. Numbers in parentheses denote assumptions which each lemma, proposition, corollary, and theorem requires. Because Proposition B.2 guarantees the Algorithm 1 to generate an infinite sequence, it supports all other Lemmas, Propositions, Corollaries and Theorems. Since drawing all the outgoing arrows from Proposition B.2 may complicate this figure, we omit them. . . . . 123

B.2 Diagram for local convergence rate proof. Numbers in parentheses denote assumptions which each lemma, proposition, and theorem requires.142

## LIST OF TABLES

2.1	When $\bar{\mathbf{H}}$ is properly set, the model generated from the CQP (2.77) can be a compromise between that of LP (minimizing the maximal error) and least squares (minimizing sum of squared point-wise error). Mean square error was measured by $\ \bar{\mathbf{b}} - \bar{\mathbf{A}}\bar{\mathbf{x}}\ _2^2/\bar{m}$ . Tested with $\bar{m} := 5000$ and $\bar{n} := 99$ . . . . .	53
3.1	Classification of support vectors and nonsupport vectors. Here $s_i^*$ is the slack variable associated with the $i^{th}$ constraints in (3.15) and defined as $s_i^* := y_i(\langle \mathbf{w}^*, \mathbf{x}_i \rangle - \gamma^*) + \xi_i^* - 1$ . . . . .	67
3.2	Properties of the problems. ISD: Input space dimension. FSD: Feature space dimension using the map (3.65). SVs: support vectors. On-SVs: On-boundary support vectors. . . . .	86
3.3	Accuracy shown in percentage of correctly classified testing patterns. . . . .	94

# Chapter 1

## Introduction

Optimization is an essential part of our everyday life. Maximizing profit in running a business, maximizing capital gain in an investment, and finding a shortest path to a destination on a journey are instances of our optimization activities. Convex quadratic programming (CQP) has broad applications in modeling such activities.

The CQP is a problem of minimizing a convex quadratic objective function subject to linear constraints. A linear constraint can be either an equality or an inequality. In this dissertation, we are mainly concerned with the CQP with very large number of inequality constraints. An equality constraint can be easily transformed to two inequality constraints.

There are two widely used classes of algorithms, the active set methods and the interior-point methods (IPMs). In general, an active set method requires many more iterations than an IPM, whereas a single iteration for it is much cheaper than that for an IPM. As the problem size becomes larger, the iteration count of the active set

method soars, while that of the IPM increases rather slowly. It is reported that the active set methods are adequate for solving small to medium sized problems while the IPMs outperform the active set methods on large problems [NW00]. Since we are concerned with large problems, we use IPMs in this work.

As it will be explained later, a bulk of work in each iteration of a classical IPM lies in finding a search direction, requiring matrix computations involving every constraint. Thus the computational cost of the work increases as the number of constraints increases. However, a large portion of the constraints are not active at the solution. If we have a prior knowledge identifying such constraints, the true optimal solution can be found without them. Unfortunately we don't know which constraints are required before we start to solve the problem. It becomes clear, however, which constraints would be active at the solution as the iterate approaches it. This gives a good reason for adaptively eliminating more and more constraints that are unlikely to be active at the solution to find the search direction.

In this dissertation, we present an adaptive constraint reduction algorithm for CQP, extending a constraint reduction algorithm for LP [TAW06]. In addition, we propose an adaptive scheme for reducing the number of constraints involved in finding the search direction. In our new scheme for CQP, the size of the relevant constraint set is determined by how close the current point is to the solution.

Then, we apply adaptive constraint reduction to training support vector machines (SVMs). The SVM is a useful tool in automating pattern recognition tasks. Rec-

ognizing hand written characters or spoken words, discriminating edible mushrooms from poisonous ones, and determining fraud uses of credit cards are examples of pattern recognition tasks. Before being used in those tasks, the machine is trained with a set of training patterns represented by points in a certain space, each of which is assigned a predetermined class label. Through the training process, the machine builds a separating hyperplane in that space, with which it decides the class of a future input. The training process is modeled as a CQP problem, where each pattern corresponds to a constraint. The number of training patterns is often very large, but the hyperplane depends on a small number of the patterns. Thus training the SVM can benefit significantly from constraint reduction.

This dissertation is organized as follows. In Chapter 2, we first define the standard form of CQP. A standard framework of primal-dual interior-point methods is then discussed. An adaptive constraint reduction algorithm for the standard form is presented. We discuss the convergence of the constraint reduction applied to the standard form. We provide an extension to infeasible problems, which includes the CQP problem of the SVM training. We provide a constraint reduction guideline for the extension. Convergence is discussed for the extension. Demonstrations of our algorithm in solving data fitting problems and random problems are then presented.

In Chapter 3, we first introduce the linear SVM and its training. We apply an adaptively constraint reduced IPM directly to the SVM training and develop effective heuristics in selecting patterns. We extend our approach to the training of

nonlinear SVMs by the use of kernels. Demonstrations of our algorithm in training the SVM on several real life data sets are presented. Our algorithm is compared with well developed and widely used algorithms including sequential minimal optimization [Pla99] and SVM<sup>light</sup> [Joa99].

In Chapter 4, we summarize our current contributions and plans for future research. In Appendix A, we provide geometric properties of the CQP. In Appendix B, we provide full details of the convergence analysis for the constraint reduction algorithm for CQP.

## Chapter 2

### Adaptive Constraint Reduction for Convex Quadratic

#### Programming

Convex quadratic programming (CQP) is an optimization problem of minimizing a convex quadratic objective function subject to linear constraints. For a descriptive example, suppose that John wants to eat lunch. Today, he has two choices: foods A and B. After he eats A and B, he feels satisfaction (or utility) independently for A and B. However, since he gets satiated as he eats more, his marginal satisfaction (derivative of his satisfaction) decreases, say, linearly. Assume that, after he eats  $x_1$  grams of A and  $x_2$  grams of B, his total satisfaction is  $\int_0^{x_1} (500 - 2x)dx + \int_0^{x_2} (500 - 2x)dx$ . Note that this function decreases with with  $x_i > 250$ , i.e., his satisfaction decreases as he eats more. Meanwhile, A and B have different nutrients. Each gram of A has 0.002g of nutrient  $\alpha$  and 0.01g of nutrient  $\beta$ , whereas each gram of B has 0.004g of  $\alpha$  and 0.005g of  $\beta$ . He wants to take at least 1.8g of  $\alpha$  and 4.5g of  $\beta$ . Since he has sufficient money and their prices are cheap and the same, he doesn't care about the cost. What

amount of A and B he should eat so as to maximize his satisfaction?

He is a very smart person, so he formulates the following problem:

$$\begin{aligned}
 \max_{\mathbf{x}} f(x) = 500x_1 - x_1^2 + 500x_2 - x_2^2 & \quad : \text{Satisfaction} \\
 \text{s.t. } 0.002x_1 + 0.004x_2 \geq 1.8 & \quad : \text{At least } 1.8g \text{ of } \alpha, \\
 0.01x_1 + 0.005x_2 \geq 4.5 & \quad : \text{At least } 4.5g \text{ of } \beta, \\
 x_1, x_2 \geq 0. &
 \end{aligned} \tag{2.1}$$

In this problem, his satisfaction is the objective function and the nutrient requirements he sets are the constraints. This is an instance of convex quadratic programming. The objective function has polynomial terms of up to  $2^{nd}$  order and is concave (or convex if the function is negated to transform the problem to a minimization formulation), and the constraints are linear. Geometrically, contours of the objective function are ellipsoidal and the region formed by the constraints is polyhedral in the example as presented in Figure 2.1. As seen in the figure, maximal satisfaction is achieved by eating 300 grams of A and 300 grams of B.

In this chapter, we discuss the following. In section 2.1, we discuss previous approaches to solving large CQP problems. In section 2.2, we introduce our algorithm. First, we define a standard form of the convex quadratic programming. We identify the dual problem, and review necessary and sufficient conditions for optimality of the CQP. After introducing path-following interior-point methods, we introduce an primal-dual affine-scaling (PDAS) interior-point method (IPM) to which we apply the

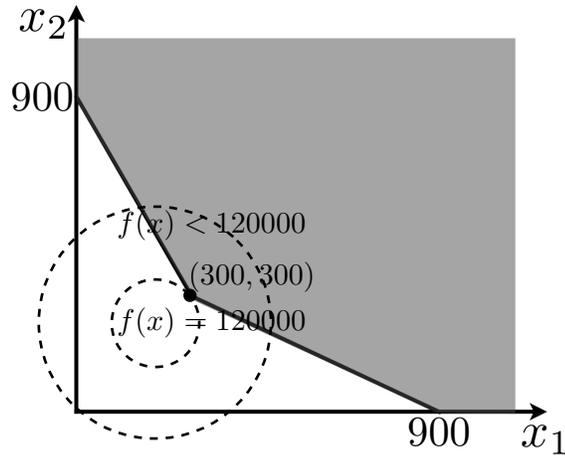


Figure 2.1: Maximizing satisfaction. The filled area is the feasible region in which all the constraints (nutrient requirements) are satisfied. The dotted circles are contours of the objective function. The objective function (his satisfaction) is maximized on the feasible region at the point marked with a circle.

constraint reduction. In Section 2.3, we present the constraint-reduction algorithm for CQP. The constraint reduced PDAS IPM for LP [TAW06] is adapted to CQP. In Section 2.4, an extension of the standard form is introduced. In Section 2.5, numerical results are presented. Concluding remarks are provided in Section 2.6.

## 2.1 Related Work

The example (2.1) has only four inequality constraints. In practice, the number of inequality constraints is often very large. However, a large portion of the constraints are not active at the solution and thus do not contribute much to deciding the search

direction of the later iterations of an IPM used to solve the problem. As will be explained later, since the major work in computing the search direction involves forming a matrix involving each constraint, computing the matrix without irrelevant constraints reduces the entire computational cost.

Reducing computational cost by finding search directions using only a fraction of the constraints has been actively studied. The most prominent approach is “column generation”. Ye [Ye90] used this approach with a “build-down” scheme for linear programming (LP), a special case of the CQP. He proposed a rule which can safely eliminate inequality constraints that will not be active at the optimum. The author applied the rule to Karmarkar’s method [Kar84] and the simplex method [Dan63]. Dantzig and Ye [DY91] proposed a “build-up” interior-point method of dual affine-scaling form. Starting from a strictly dual feasible point, it uses a subset of constraints to determine the search direction at each iteration. It accepts the direction if taking the direction violates no constraint. If some constraints are violated, it adds them to the set for determining the search direction and retries. Ye [Ye92] proposed a potential reduction algorithm allowing column generation for linear feasibility problems to which linear programs (LP) can be converted. Starting with a polytope including the feasible domain, at every iteration, the scheme builds a cutting plane for a violated inequality. Luo and Sun [LS98] proposed a similar scheme for convex quadratic feasibility problems to which CQP problems can be transformed. Tone [Ton93] proposed an active set strategy for the dual potential reduction algorithm proposed by

Ye [Ye91]. The strategy finds the search direction using constraints associated with small dual slack variables.

Another approach to reduce the computational time for finding a search direction is to use an iterative solver such as the preconditioned conjugate gradient method [Saa03, chap. 9] to solve the normal equations arising in the primal-dual interior-point method (PDIPM). Making a good preconditioner is the most critical part in guaranteeing the success of iterative solvers. Wang and O’Leary [WO00] used an adaptive preconditioner that approximates the LP normal equation matrix with a fraction of constraints. In their approach, once a preconditioner is formed, its Cholesky factor is updated or recomputed in the subsequent iterations.

The LP constraint-reduction algorithm of Tits et al. [TAW06] and Winternitz et al. [WNT07] chooses constraints from scratch rather than by building-up. Convergence was proven, and experiments demonstrated good performance. An attractive aspect of the constraint-reduction scheme considered in these papers is its easy applicability to the state of the art PDIPMs such as the variants of the Mehrotra’s predictor-corrector algorithm [Meh92, Wri97, NW00].

In this chapter, we present a constraint-reduction algorithm for CQP, inheriting the good properties of the constraint-reduction algorithm for LP [TAW06]. In addition, we propose an adaptive scheme for reducing the number of constraints involved in finding the search direction. Since it becomes more obvious which constraints would be active as the iterate gets closer to a solution, eliminating more prospec-

tively inactive constraints would not impair finding the search direction. In our new scheme for CQP, the size of the constraint set is determined by how close the current point is to the solution. Either the duality gap or a complementarity measure<sup>1</sup> provides a good criterion.

## 2.2 Solving Convex Quadratic Programming

### 2.2.1 Standard Form

A standard form of the CQP with inequality constraints can be stated as

$$\begin{aligned} \min_{\mathbf{x}} f(\mathbf{x}) &= \min_{\mathbf{x}} \frac{1}{2} \mathbf{x}^T \mathbf{H} \mathbf{x} + \mathbf{c}^T \mathbf{x}, \\ \text{s.t. } \mathbf{A} \mathbf{x} &\geq \mathbf{b}, \end{aligned} \tag{2.2}$$

where  $\mathbf{A} \in \mathbb{R}^{m \times n}$ ,  $\mathbf{H} \in \mathbb{R}^{n \times n}$ ,  $\mathbf{x} \in \mathbb{R}^n$ ,  $\mathbf{c} \in \mathbb{R}^n$ , and  $\mathbf{b} \in \mathbb{R}^m$ . The inequality constraints are frequently replaced with equality constraints by introducing additional slack variables. The equivalent standard form is

$$\begin{aligned} \min_{\mathbf{x}} f(\mathbf{x}) &= \min_{\mathbf{x}} \frac{1}{2} \mathbf{x}^T \mathbf{H} \mathbf{x} + \mathbf{c}^T \mathbf{x}, \\ \text{s.t. } \mathbf{A} \mathbf{x} - \mathbf{s} &= \mathbf{b}, \\ \mathbf{s} &\geq \mathbf{0}. \end{aligned} \tag{2.3}$$

The CQP is a special case of quadratic programming in that the objective function is convex quadratic. Geometric properties of the problem and the solution are

---

<sup>1</sup>This is often called the *duality measure*.

reviewed in Appendix A. The objective function is convex if and only if the Hessian matrix is symmetric and positive semidefinite, as shown in Lemma A.2. If the matrix is not symmetric, we can easily transform it to a symmetric matrix by replacing it with  $\frac{1}{2}(\mathbf{H} + \mathbf{H}^T)$ . It is trivial to show that the resulting objective function is the same as before. So, in the sequel, we assume that the Hessian matrix  $\mathbf{H}$  is always symmetric.

The set of points that satisfy the inequality constraints of (2.2) is said to be the primal feasible set:

$$\mathcal{F}_P := \{\mathbf{x} \in \mathbb{R}^n : \mathbf{A}\mathbf{x} \geq \mathbf{b}\}. \quad (2.4)$$

If a point strictly satisfies the inequality constraints, the point is said to be strictly primal feasible. We define the set of strictly primal feasible points as

$$\mathcal{F}_P^o := \{\mathbf{x} \in \mathbb{R}^n : \mathbf{A}\mathbf{x} > \mathbf{b}\}. \quad (2.5)$$

A point that solves the problem (2.2) is said to be a *minimizer*, an *optimal solution*, an *optimal point*, a *global optimum*, or a *solution*. The *primal solution set* is the set of optimal solutions:

$$\mathcal{F}_P^* := \{\mathbf{x}^* \in \mathcal{F}_P : f(\mathbf{x}^*) \leq f(\mathbf{x}), \forall \mathbf{x} \in \mathcal{F}_P\}. \quad (2.6)$$

Let  $M := \{1, \dots, m\}$ , and let  $\mathbf{a}_i^T$  be the  $i^{\text{th}}$  row of  $\mathbf{A} \in \mathbb{R}^{m \times n}$  with  $m \gg n$ . For an index set  $Q \subseteq M$ , let  $\mathbf{A}_Q$  be a submatrix of  $\mathbf{A}$  constructed by deleting rows  $\mathbf{a}_i^T$  for  $i \notin Q$ . The same notation  $\mathbf{v}_Q$  is applied to a column vector  $\mathbf{v} \in \mathbb{R}^m$ . Similarly,

for an  $m \times m$  matrix  $\mathbf{B}$ , we let  $\mathbf{B}_{Q^c}$  denote a submatrix of  $\mathbf{B}$  constructed by deleting both rows and columns indexed by  $i \notin Q$ . We use  $\mathbf{A}_Q^T$  and  $\mathbf{B}_Q^T$  to denote transpose of  $\mathbf{A}_Q$  and  $\mathbf{B}_Q$ , respectively. Horizontal concatenation of two matrices (or row vectors) with the same number of rows,  $\mathbf{H}$  and  $\mathbf{A}^T$  for instance, is denoted by  $[\mathbf{H}, \mathbf{A}^T]$ . We denote by  $\mathcal{N}(\mathbf{H})$  the nullspace of  $\mathbf{H}$ . The complement  $Q^c$  of an index set  $Q$  is defined as  $Q^c := M \setminus Q$ .

At a primal feasible point  $\mathbf{x} \in \mathcal{F}_P$ , an inequality constraint is said to be *active* if it holds as an equality. We define the *active set*  $\mathcal{A}(\mathbf{x})$ , the index set of active constraints at  $\mathbf{x} \in \mathcal{F}_P$ , as

$$\mathcal{A}(\mathbf{x}) := \{i \in M : \mathbf{a}_i^T \mathbf{x} = b_i\}, \quad (2.7)$$

and the *inactive set* as

$$\mathcal{A}(\mathbf{x})^c := \{i \in M : \mathbf{a}_i^T \mathbf{x} \neq b_i\}.$$

## 2.2.2 Duality

Every quadratic programming (QP) problem is associated with a dual problem defined by the same data with additional variables. The dual associated with the primal (2.2), which can be derived from the Lagrangian  $L(\mathbf{x}, \boldsymbol{\lambda}) := f(\mathbf{x}) - \boldsymbol{\lambda}^T (\mathbf{A}\mathbf{x} - \mathbf{b})$ , is as follows:

$$\begin{aligned} \max_{\mathbf{x}, \boldsymbol{\lambda}} f_D(\mathbf{x}, \boldsymbol{\lambda}) &= \max_{\mathbf{x}, \boldsymbol{\lambda}} -\frac{1}{2} \mathbf{x}^T \mathbf{H} \mathbf{x} + \mathbf{b}^T \boldsymbol{\lambda}, \\ \text{s.t. } \mathbf{H} \mathbf{x} + \mathbf{c} - \mathbf{A}^T \boldsymbol{\lambda} &= \mathbf{0}, \end{aligned} \quad (2.8)$$

$$\boldsymbol{\lambda} \geq \mathbf{0},$$

where  $\boldsymbol{\lambda} \in \mathbb{R}^m$  is called the Lagrange multipliers. The *duality gap* for a given pair  $(\mathbf{x}, \boldsymbol{\lambda})$  is  $f(\mathbf{x}) - f_D(\mathbf{x}, \boldsymbol{\lambda})$ , the difference between the primal and dual objective functions.

The primal-dual feasible set is defined as the set of points which satisfy the constraints of both the primal (2.3) and the dual (2.8):

$$\mathcal{F} := \{(\mathbf{x}, \mathbf{s}, \boldsymbol{\lambda}) : \mathbf{Ax} - \mathbf{s} = \mathbf{b}, \mathbf{Hx} + \mathbf{c} - \mathbf{A}^T\boldsymbol{\lambda} = \mathbf{0}, \mathbf{s} \geq \mathbf{0}, \boldsymbol{\lambda} \geq \mathbf{0}\}.$$

The relative interior of  $\mathcal{F}$  is defined as

$$\mathcal{F}^\circ := \{(\mathbf{x}, \mathbf{s}, \boldsymbol{\lambda}) : \mathbf{Ax} - \mathbf{s} = \mathbf{b}, \mathbf{Hx} + \mathbf{c} - \mathbf{A}^T\boldsymbol{\lambda} = \mathbf{0}, \mathbf{s} > \mathbf{0}, \boldsymbol{\lambda} > \mathbf{0}\}.$$

### 2.2.3 Optimality Conditions

We can obtain the the first order necessary conditions for the solution of the optimization problem (2.2) using the Karush-Kuhn-Tucker (KKT) conditions. For a proof of the necessity of the conditions, see Nocedal and Wright [NW00, Chapter 12] or Fletcher [Fle87, Chapter 9].

**Theorem 2.1** (KKT conditions). *If  $\mathbf{x}$  is an optimal solution of (2.2), then there exist  $\mathbf{s}$  and  $\boldsymbol{\lambda}$  such that*

$$\mathbf{Hx} + \mathbf{c} - \mathbf{A}^T\boldsymbol{\lambda} = \mathbf{0}, \tag{2.9}$$

$$\mathbf{Ax} - \mathbf{b} - \mathbf{s} = \mathbf{0}, \tag{2.10}$$

$$\mathbf{S}\boldsymbol{\lambda} = \mathbf{0}, \tag{2.11}$$

$$\mathbf{s}, \boldsymbol{\lambda} \geq \mathbf{0}, \quad (2.12)$$

where  $\mathbf{S} := \text{diag}(\mathbf{s})$ . Likewise, if  $(\mathbf{x}, \boldsymbol{\lambda})$  solves the dual (2.8), then there exists  $\mathbf{s}$  such that the conditions above hold as well. The points  $(\mathbf{x}, \mathbf{s}, \boldsymbol{\lambda})$  that satisfy the conditions are said to be the KKT points.

In the KKT conditions, (2.9) defines the necessary condition associated with the gradient of the objective function and the constraints. At an optimal solution (or point)  $\mathbf{x}$ , the gradient  $\mathbf{H}\mathbf{x} + \mathbf{c}$  can be expressed as a nonnegative combination of the gradients of active constraints. The boundary of a constraint  $\mathbf{a}_i^T \mathbf{x} \geq b_i$  forms a hyperplane in  $\mathbb{R}^n$  and  $\mathbf{a}_i$  is perpendicular to the hyperplane directed toward the inside of the feasible region. This implies that any direction from a KKT point toward the inside of the feasible region is ascending for the objective function. See Figure 2.2 for a geometrical interpretation. The second condition is the primal feasibility condition. The third condition (2.11), which is referred to as the complementarity condition, states that only the active constraints are involved in the first condition (2.9).

Indeed, the first order necessary conditions are sufficient for global optima in the convex quadratic programming case. We give a proof from Wright [Wri97, Appendix. A] here.

**Theorem 2.2** (Necessary and sufficient conditions for global optima).  *$\mathbf{x}^*$  is an optimal solution of (2.2) and  $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$  is that of (2.8) if and only if there exist  $\mathbf{s}^*$  such that  $(\mathbf{x}^*, \mathbf{s}^*, \boldsymbol{\lambda}^*)$  satisfies the KKT conditions.*

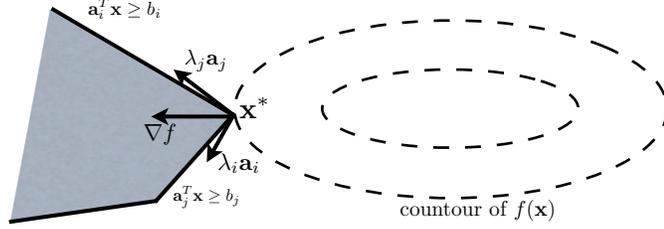


Figure 2.2:  $i^{\text{th}}$  and  $j^{\text{th}}$  constraints are active at the solution  $\mathbf{x}^*$ . The gradient of the objective function  $f(\mathbf{x})$  at  $\mathbf{x}^*$  can be expressed by the nonnegative combination of  $\mathbf{a}_i$  and  $\mathbf{a}_j$ .

*Proof.* Let us show that the KKT conditions are sufficient for the global optima, which can be proven by showing that, for any  $\bar{\mathbf{x}}$  in  $\mathcal{F}_P$ ,  $f(\mathbf{x}^*) \leq f(\bar{\mathbf{x}})$ . Since the objective function  $f(\mathbf{x})$  of the problem (2.2) is convex, we know that, for any feasible point  $\bar{\mathbf{x}}$  other than  $\mathbf{x}^*$ , for any  $\alpha \in (0, 1]$ , and for  $\mathbf{v} := \bar{\mathbf{x}} - \mathbf{x}^* \neq \mathbf{0}$ ,

$$\begin{aligned} f(\mathbf{x}^* + \alpha\mathbf{v}) &= f(\alpha\bar{\mathbf{x}} + (1 - \alpha)\mathbf{x}^*) \\ &\leq \alpha f(\bar{\mathbf{x}}) + (1 - \alpha)f(\mathbf{x}^*) \\ &= \alpha f(\mathbf{x}^* + \mathbf{v}) + (1 - \alpha)f(\mathbf{x}^*). \end{aligned}$$

Since  $\alpha > 0$ , it immediately follows that

$$\frac{f(\mathbf{x}^* + \alpha\mathbf{v}) - f(\mathbf{x}^*)}{\alpha} \leq f(\mathbf{x}^* + \mathbf{v}) - f(\mathbf{x}^*). \quad (2.13)$$

By taking the limit as  $\alpha \rightarrow 0$ , we know, from (2.13), that

$$f(\mathbf{x}^*) + \mathbf{v}^T \nabla f(\mathbf{x}^*) \leq f(\bar{\mathbf{x}}). \quad (2.14)$$

Now let us show that  $\mathbf{v}^T \nabla f(\mathbf{x}^*)$  (the scaled directional derivative of  $f$  at  $\mathbf{x}^*$  in the

direction toward  $\bar{\mathbf{x}}$ ) is nonnegative. Since  $\nabla f(\mathbf{x}) = \mathbf{H}\mathbf{x} + \mathbf{c}$ , from the first KKT condition (2.9) we get

$$\nabla f(\mathbf{x}^*) - \mathbf{A}^T \boldsymbol{\lambda} = \mathbf{0}. \quad (2.15)$$

So by multiplying by  $\mathbf{v}^T$  on both sides of (2.15), we obtain

$$\begin{aligned} \mathbf{v}^T \nabla f(\mathbf{x}^*) &= \boldsymbol{\lambda}^T \mathbf{A} \mathbf{v} \\ &= \sum_{i \in \mathcal{A}(\mathbf{x}^*)} \lambda_i \mathbf{a}_i^T \mathbf{v} + \sum_{i \notin \mathcal{A}(\mathbf{x}^*)} \lambda_i \mathbf{a}_i^T \mathbf{v}, \end{aligned}$$

from which, due to the complementarity condition (2.11), it immediately follows that

$$\mathbf{v}^T \nabla f(\mathbf{x}^*) = \sum_{i \in \mathcal{A}(\mathbf{x}^*)} \lambda_i \mathbf{a}_i^T \mathbf{v}. \quad (2.16)$$

Since  $\bar{\mathbf{x}} = \mathbf{x}^* + \mathbf{v}$  is feasible, it follows that, for all  $i \in \mathcal{A}(\mathbf{x}^*)$ ,

$$0 \leq \mathbf{a}_i^T (\mathbf{x}^* + \mathbf{v}) - b_i = \mathbf{a}_i^T \mathbf{v}.$$

From this and from (2.16) it follows that

$$\mathbf{v}^T \nabla f(\mathbf{x}^*) \geq 0. \quad (2.17)$$

Therefore, (2.14) and (2.17) yield

$$f(\mathbf{x}^*) \leq f(\bar{\mathbf{x}}), \forall \bar{\mathbf{x}} \in \mathcal{F}_P,$$

which implies that  $\mathbf{x}^*$  is a global optimum of (2.2).

It can be proven that  $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$  is a global optimum of (2.8) in the same way.  $\square$

Furthermore, if  $\mathbf{H}$  is positive definite, then the objective function is strictly convex. The contour of the objective function is ellipsoidal. This implies that the optimal solution is unique in this case.

## 2.2.4 Primal-Dual Interior-Point Method

In the previous section, it was shown that finding an optimal solution is equivalent to finding a KKT point that satisfies the first order necessary conditions (2.9)-(2.12). In other words, the problem is now finding a solution to the following nonlinear equation with nonnegativity constraints:

$$\mathbf{F}(\mathbf{x}, \mathbf{s}, \boldsymbol{\lambda}) := \begin{bmatrix} \mathbf{H}\mathbf{x} + \mathbf{c} - \mathbf{A}^T\boldsymbol{\lambda} \\ \mathbf{A}\mathbf{x} - \mathbf{b} - \mathbf{s} \\ \mathbf{S}\boldsymbol{\Lambda}\mathbf{e} \end{bmatrix} = \mathbf{0}, \quad (2.18)$$

$$\mathbf{s}, \boldsymbol{\lambda} \geq \mathbf{0}, \quad (2.19)$$

where  $\boldsymbol{\Lambda} := \text{diag}(\boldsymbol{\lambda})$  and  $\mathbf{e} := [1, \dots, 1]^T$ . The primal-dual interior-point method (PDIPM) uses a Newton-like method applied to the function (2.18) to generate a sequence of points  $(\mathbf{x}^k, \mathbf{s}^k, \boldsymbol{\lambda}^k)$  that strictly satisfy the non-negativity conditions (2.19) and converge to a KKT point.

Newton's method is a well known iterative algorithm for finding a root of a system of nonlinear equations. It builds the first order Taylor series approximation (or a linear model) of  $\mathbf{F}(\mathbf{x}, \mathbf{s}, \boldsymbol{\lambda})$  at the current point  $(\mathbf{x}^k, \mathbf{s}^k, \boldsymbol{\lambda}^k)$  and finds the search direction

$(\Delta \mathbf{x}^k, \Delta \mathbf{s}^k, \Delta \boldsymbol{\lambda}^k)$  by obtaining the solution to the linear approximation:

$$\mathbf{F}(\mathbf{x}^k, \mathbf{s}^k, \boldsymbol{\lambda}^k) + \mathbf{J}(\mathbf{x}^k, \mathbf{s}^k, \boldsymbol{\lambda}^k) \begin{bmatrix} \Delta \mathbf{x}^k \\ \Delta \mathbf{s}^k \\ \Delta \boldsymbol{\lambda}^k \end{bmatrix} = \mathbf{0},$$

where  $\mathbf{J}$  is the Jacobian matrix of  $\mathbf{F}$  at the current point.  $\mathbf{J}$  is defined as the matrix of the first order partial derivatives of  $\mathbf{F}$ ; in the case of (2.18),

$$\mathbf{J}(\mathbf{x}, \mathbf{s}, \boldsymbol{\lambda}) := \begin{bmatrix} \mathbf{H} & \mathbf{0} & \mathbf{A}^T \\ \mathbf{A} & -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Lambda} & \mathbf{S} \end{bmatrix}$$

Thus the search direction is obtained from the solution of the following system of equations:

$$\begin{bmatrix} \mathbf{H} & \mathbf{0} & \mathbf{A}^T \\ \mathbf{A} & -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Lambda} & \mathbf{S} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x}^k \\ \Delta \mathbf{s}^k \\ \Delta \boldsymbol{\lambda}^k \end{bmatrix} = \begin{bmatrix} -(\mathbf{H}\mathbf{x}^k + \mathbf{c} - \mathbf{A}^T \boldsymbol{\lambda}^k) \\ -(\mathbf{A}\mathbf{x}^k - \mathbf{b} - \mathbf{s}^k) \\ -\mathbf{S}^k \boldsymbol{\Lambda}^k \mathbf{e} \end{bmatrix}. \quad (2.20)$$

The direction  $(\Delta \mathbf{x}^k, \Delta \mathbf{s}^k, \Delta \boldsymbol{\lambda}^k)$  is said to be the full Newton step. Since taking the full step may violate the non-negativity constraints (2.19), a line search along the step for  $\alpha \in (0, 1]$  with

$$(\mathbf{x}^{k+1}, \mathbf{s}^{k+1}, \boldsymbol{\lambda}^{k+1}) := (\mathbf{x}^k, \mathbf{s}^k, \boldsymbol{\lambda}^k) + \alpha(\Delta \mathbf{x}^k, \Delta \mathbf{s}^k, \Delta \boldsymbol{\lambda}^k)$$

is often performed to keep  $\mathbf{s}^{k+1}$  and  $\boldsymbol{\lambda}^{k+1}$  positive. Some algorithms perform the line search for the dual variables  $\boldsymbol{\lambda}$  separately from the primal variables  $\mathbf{x}$  and  $\mathbf{s}$ . If the

current iterate is very close to the boundary of  $\mathcal{F}$ , the line search can result in very small  $\alpha$ . To avoid this trouble, many primal-dual interior-point methods intentionally keep the iterates from getting too close to the boundary of  $\mathcal{F}$  or bias  $\mathbf{s}^k$  and  $\boldsymbol{\lambda}^k$  toward the interior of nonnegative orthant.

### 2.2.5 Central Path

The primal central path can be derived from the following optimization problem with a logarithmic barrier function:

$$\min_{\mathbf{x}} f_B(\mathbf{x}) = \min_{\mathbf{x}} \frac{1}{2} \mathbf{x}^T \mathbf{H} \mathbf{x} + \mathbf{c}^T \mathbf{x} - \mu \sum_{i=1}^m \log(\mathbf{a}_i^T \mathbf{x} - b_i), \quad (2.21)$$

where  $\mu$  is the barrier parameter and  $\mu > 0$ . The logarithmic function is not defined if  $\mathcal{F}_P^o$  is empty, so we assume that  $\mathcal{F}_P^o$  is nonempty. The first two terms are the objective function of the standard form (2.2). The inequality constraints of (2.2) are arguments for the logarithmic function so that the new objective function diverges to infinity on the boundary of the feasible region  $\mathcal{F}_P$ . As a result, for  $\mu > 0$ , the barrier function forces the minimizer for (2.21) to be away from the boundary of  $\mathcal{F}_P$ . As  $\mu$  decreases, the effect of the barrier function wanes and the minimizer is allowed to approach closer to the boundary of  $\mathcal{F}_P$ .

The barrier function  $g(\mathbf{x}) := -\mu \sum_{i=1}^m \log(\mathbf{a}_i^T \mathbf{x} - b_i)$  is strictly convex on the orthogonal complement of the nullspace of  $\mathbf{A}$ , from which it follows that the objective function is strictly convex if the intersection of the two nullspaces of  $\mathbf{A}$  and  $\mathbf{H}$  is a

trivial set  $\{\mathbf{0}\}$ . For the time being, we assume  $\mathcal{N}(\mathbf{A}) \cap \mathcal{N}(\mathbf{H}) = \{\mathbf{0}\}$ . In fact, we can preprocess the optimization problem (2.2) so that the coefficient matrix  $\mathbf{A}$  has full column rank and the trivial nullspace  $\{\mathbf{0}\}$  [AA95].

For every  $\mu > 0$ , since  $f_B(\mathbf{x})$  of (2.21) is strictly convex and diverges to  $\infty$  as  $\mathbf{x}$  approaches the boundary of  $\mathcal{F}_P^o$ , the optimal solution  $\mathbf{x}(\mu)$  of the problem lies in  $\mathcal{F}_P^o$  and is unique. The trajectory  $\{\mathbf{x}(\mu) \text{ for } \mu \rightarrow 0\}$  is the *primal central path*. As  $\mu$  converges to 0, the solution of (2.21) converges to  $\mathcal{F}_P^*$  [Wri92, Theorem 8].

Using our slack variables, we see that (2.21) is equivalent to the following equality constrained optimization problem:

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{s}} \quad & \frac{1}{2} \mathbf{x}^T \mathbf{H} \mathbf{x} + \mathbf{c}^T \mathbf{x} - \mu \sum_{i=1}^m \log(s_i) \\ \text{s.t.} \quad & \mathbf{A} \mathbf{x} - \mathbf{s} = \mathbf{b}. \end{aligned} \tag{2.22}$$

With the vector  $\boldsymbol{\lambda}$  of Lagrange multipliers associated with the equality constraints of (2.22), we can obtain the first-order necessary (KKT) conditions for (2.22):

$$\mathbf{H} \mathbf{x} + \mathbf{c} - \mathbf{A}^T \boldsymbol{\lambda} = \mathbf{0}, \tag{2.23}$$

$$\mathbf{A} \mathbf{x} - \mathbf{b} - \mathbf{s} = \mathbf{0}, \tag{2.24}$$

$$\mathbf{S} \boldsymbol{\lambda} = \sigma \mu \mathbf{e}, \tag{2.25}$$

$$\mathbf{s}, \boldsymbol{\lambda} > \mathbf{0}. \tag{2.26}$$

These conditions are usually referred to as the *perturbed KKT conditions* in which the third condition (2.25) is perturbed from the complementarity condition (2.11). The positivity constraints (2.26) on  $\mathbf{s}$  and  $\boldsymbol{\lambda}$  are due to the logarithmic barrier function

and the third condition (2.25). The conditions are sufficient for a solution, because the objective function is convex. Assuming that  $\mathcal{N}(\mathbf{A}) \cap \mathcal{N}(\mathbf{H}) = \{\mathbf{0}\}$ , since the objective function of (2.21) is strictly convex,  $\mathbf{x}$  is uniquely defined, and  $\mathbf{s}$  and  $\boldsymbol{\lambda}$  are also uniquely defined (by (2.24) and (2.25)) for a given barrier parameter  $\mu$ . The trajectory,

$$\{(\mathbf{x}(\mu), \mathbf{s}(\mu), \boldsymbol{\lambda}(\mu)) \text{ where } (\mathbf{x}(\mu), \mathbf{s}(\mu), \boldsymbol{\lambda}(\mu)) \text{ satisfies (2.23) – (2.26) and } \mu \rightarrow 0\}$$

is the *primal-dual central path*.

## 2.2.6 Primal-Dual Path-Following Interior-Point Method

The primal-dual path-following method tries to stay within a certain neighborhood of the central path. The method approximately solves the perturbed-KKT system for a given  $\mu$ . Then it reduces  $\mu$  and repeats solving the system using the previous solution as the starting point for one step of Newton’s method, setting the barrier parameter as the complementarity measure:

$$\mu := \frac{\mathbf{s}^T \boldsymbol{\lambda}}{m}.$$

The Newton steps obtained from the perturbed KKT system (2.23)-(2.25) aim toward the primal-dual central-path, whereas the pure Newton steps (with  $\mu := 0$ ) aim directly toward the solution of the KKT system (2.9)-(2.11). Most primal-dual path-following algorithms balance those two aims using an additional parameter  $\sigma$ , the centering parameter, with  $\sigma \in [0, 1]$ . From the first-order Taylor series approximation

of the perturbed KKT system, the generic step equations for primal-dual methods are then obtained as follows:

$$\begin{bmatrix} \mathbf{H} & \mathbf{0} & \mathbf{A}^T \\ \mathbf{A} & -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Lambda} & \mathbf{S} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{s} \\ \Delta \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} -\mathbf{r}_c \\ -\mathbf{r}_b \\ -\mathbf{S}\boldsymbol{\lambda} + \sigma\mu\mathbf{e} \end{bmatrix},$$

where  $\mathbf{r}_c := \mathbf{H}\mathbf{x} + \mathbf{c} - \mathbf{A}^T\boldsymbol{\lambda}$  and  $\mathbf{r}_b := \mathbf{A}\mathbf{x} - \mathbf{b} - \mathbf{s}$ . The residuals  $\mathbf{r}_c$  and  $\mathbf{r}_b$  are  $\mathbf{0}$  in primal-dual feasible methods.

Various primal-dual path-following IPMs have been proposed. Monteiro and Adler proposed the short-step path-following (SPF) IPM for LP [MA89a] and extended the method to CQP in [MA89b]. They showed polynomial complexity bound of  $O(\sqrt{n} \log \epsilon)$  iterations, where  $\epsilon$  is the required accuracy. The SPF method keeps the iterates in the  $L_2$  neighborhood of the primal-dual central path. The long-step path-following methods are other variants using the  $L_\infty$  neighborhood. Kojima et al [KMY89] showed a complexity bound of  $O(n \log \epsilon)$  for a LPF method for linear complementarity problems to which CQP problems can be transformed.

Other variants are the predictor-corrector type methods which use the second order information of the Taylor series approximation. The methods are shown to be most effective in practice. Wright provided a variant of a Mehrotra-type predictor-corrector algorithm for LP [Meh92], with the same importance on the predictor and the corrector steps [Wri97]. However, the computational complexity of the variant has never been proven. Cartis [Car04] proposed the primal-dual second-order corrector for

LP, a variant of the Mehrotra's algorithm, which gives less importance to the corrector step. This variant has a polynomial complexity bound. Nocedal and Wright extended the algorithm to CQP [NW00, Chapter 16]. Y. Zhang and D. Zhang [ZZ95] showed a polynomial complexity bound for a variant of the Mehrotra-type methods, with less importance on the corrector step, for the horizontal linear complementarity problem which is a generalization of LP and CQP.

### 2.2.7 Primal-Feasible Affine-Scaling PDIPM

In a primal-feasible affine-scaling primal-dual interior-point method, Newton's iteration is applied to the equalities in the KKT conditions (2.9)-(2.11) to get the solution of (2.2) with primal feasibility maintained. The search direction is obtained by solving the linear system (2.20) obtained from the first order Taylor series approximation at the current point  $(\mathbf{x}, \mathbf{s}, \boldsymbol{\lambda})$  to  $(\mathbf{x} + \Delta\mathbf{x}, \mathbf{s} + \Delta\mathbf{s}, \boldsymbol{\lambda} + \Delta\boldsymbol{\lambda})$ :

$$\mathbf{H}\Delta\mathbf{x} - \mathbf{A}^T\Delta\boldsymbol{\lambda} = -(\mathbf{H}\mathbf{x} + \mathbf{c} - \mathbf{A}^T\boldsymbol{\lambda}), \quad (2.27)$$

$$\mathbf{A}\Delta\mathbf{x} - \Delta\mathbf{s} = \mathbf{0}, \quad (2.28)$$

$$\mathbf{S}\Delta\boldsymbol{\lambda} + \boldsymbol{\Lambda}\Delta\mathbf{s} = -\mathbf{S}\boldsymbol{\lambda}. \quad (2.29)$$

The *augmented system* is (2.27) along with the equation obtained by substituting  $\mathbf{A}\Delta\mathbf{x}$  for  $\Delta\mathbf{s}$  in (2.29):

$$\begin{bmatrix} \mathbf{H} & -\mathbf{A}^T \\ \boldsymbol{\Lambda}\mathbf{A} & \mathbf{S} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x} \\ \Delta\boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} -(\mathbf{H}\mathbf{x} + \mathbf{c} - \mathbf{A}^T\boldsymbol{\lambda}) \\ -\mathbf{S}\boldsymbol{\lambda} \end{bmatrix}, \quad (2.30)$$

where we obtain  $\Delta \mathbf{s}$  from (2.28) by computing

$$\Delta \mathbf{s} = \mathbf{A} \Delta \mathbf{x}. \quad (2.31)$$

The matrices associated with the Newton system (2.27)-(2.29) and the augmented Newton system (2.30)-(2.31) will be referred to as Jacobian  $\mathbf{J}$  and augmented Jacobian  $\mathbf{J}_a$ , respectively:

$$\mathbf{J}(\mathbf{A}, \mathbf{s}, \boldsymbol{\lambda}) := \begin{bmatrix} \mathbf{H} & \mathbf{0} & -\mathbf{A}^T \\ \mathbf{A} & -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Lambda} & \mathbf{S} \end{bmatrix}, \text{ and } \mathbf{J}_a(\mathbf{A}, \mathbf{s}, \boldsymbol{\lambda}) := \begin{bmatrix} \mathbf{H} & -\mathbf{A}^T \\ \boldsymbol{\Lambda} \mathbf{A} & \mathbf{S} \end{bmatrix}. \quad (2.32)$$

When  $\mathbf{s} > 0$ ,  $\mathbf{S}$  is nonsingular, and the *normal equation* is obtained by eliminating  $\Delta \boldsymbol{\lambda}$  in (2.30):

$$(\mathbf{H} + \mathbf{A}^T \mathbf{S}^{-1} \boldsymbol{\Lambda} \mathbf{A}) \Delta \mathbf{x} = -(\mathbf{H} \mathbf{x} + \mathbf{c}), \quad (2.33)$$

where we obtain  $\Delta \mathbf{s}$  and  $\Delta \boldsymbol{\lambda}$  by computing

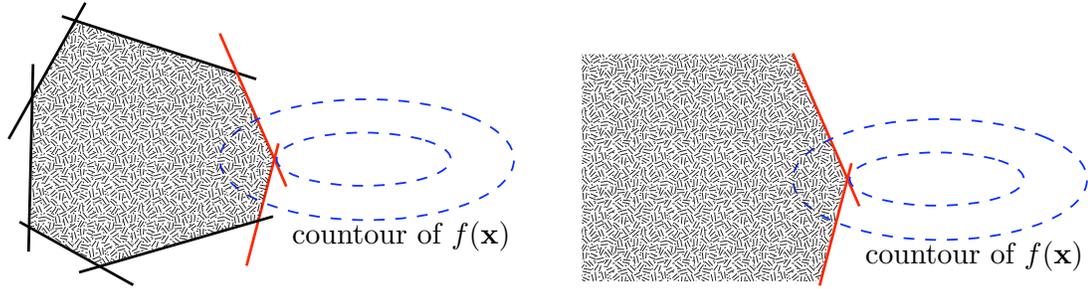
$$\Delta \mathbf{s} = \mathbf{A} \Delta \mathbf{x}, \quad (2.34)$$

$$\Delta \boldsymbol{\lambda} = -\boldsymbol{\lambda} - \mathbf{S}^{-1} \boldsymbol{\Lambda} \Delta \mathbf{s}. \quad (2.35)$$

The dominant work in (2.33)-(2.35) is forming the matrix

$$\mathbf{M} := \mathbf{H} + \mathbf{A}^T \mathbf{S}^{-1} \boldsymbol{\Lambda} \mathbf{A}$$

which requires approximately  $mn^2/2$  multiplications if  $\mathbf{A}$  is a dense matrix.



(a) Optimal solution is the point at which the two red lines meet.

(b) Without constraints other than the active ones, we still can get the same optimal solution.

Figure 2.3: Two dimensional CQP examples .

## 2.3 Adaptive Constraint Reduction

In this section, we present an adaptive constraint-reduction method based on the constraint-reduced dual-feasible primal-dual affine-scaling algorithm for LP proposed in [TAW06]. The primal (2.2) corresponds to the dual formulation of [TAW06].

### 2.3.1 Adaptive Constraint-Reduction for Primal-Feasible Affine-Scaling PDIPM

Assume for the time being that the solution is unique and strictly complementary, and we have prior knowledge of which constraints are active at the solution  $\mathbf{x}^* \in \mathcal{F}_P^*$ . So we have  $Q$  such that  $\mathcal{A}(\mathbf{x}^*) \subseteq Q$ . As illustrated in Figure 2.3, without other constraints, we still can get the same solution  $\mathbf{x}^*$  by solving the following reduced

minimization problem

$$\begin{aligned} \min_{\mathbf{x}} \quad & \frac{1}{2} \mathbf{x}^T \mathbf{H} \mathbf{x} + \mathbf{c}^T \mathbf{x}, \\ \text{s.t.} \quad & \mathbf{A}_Q \mathbf{x} \geq \mathbf{b}_Q. \end{aligned}$$

We would get a search direction by solving the reduced Newton system

$$\mathbf{H} \Delta \mathbf{x} - \mathbf{A}_Q^T \Delta \boldsymbol{\lambda}_Q = -(\mathbf{H} \mathbf{x} + \mathbf{c} - \mathbf{A}_Q^T \boldsymbol{\lambda}_Q), \quad (2.36)$$

$$\mathbf{A}_Q \Delta \mathbf{x} - \Delta \mathbf{s}_Q = \mathbf{0}, \quad (2.37)$$

$$\mathbf{S}_{Q^2} \Delta \boldsymbol{\lambda}_Q + \boldsymbol{\Lambda}_{Q^2} \Delta \mathbf{s}_Q = -\mathbf{S}_{Q^2} \boldsymbol{\lambda}_Q, \quad (2.38)$$

from which we derive the reduced normal equations

$$(\mathbf{H} + \mathbf{A}_Q^T \mathbf{S}_{Q^2}^{-1} \boldsymbol{\Lambda}_{Q^2} \mathbf{A}_Q) \Delta \mathbf{x} = -(\mathbf{H} \mathbf{x} + \mathbf{c}). \quad (2.39)$$

In reality, we do not know which constraints will be active until we get the solution. Instead we try to include in  $Q$  constraints that seem likely to be active. Thus,  $Q$  can vary iteration by iteration, and we need to update the entire vectors  $\mathbf{s}$  and  $\boldsymbol{\lambda}$ . We choose to do this using (2.34) and (2.35) for  $\Delta \mathbf{s}$  and  $\Delta \boldsymbol{\lambda}$ , thus maintaining the primal feasibility.

Intuitively, under the strict complementarity assumption, by considering condition (2.11) in the KKT conditions (2.9)-(2.12), we can notice that, if the  $i^{\text{th}}$  constraint is inactive at the solution, then  $s_i^{-1} \lambda_i = 0$  at the optimal solution. Therefore, rewriting the matrix in (2.33) in outer product summation form as

$$\mathbf{M} := \mathbf{H} + \sum_{i=1}^m s_i^{-1} \lambda_i \mathbf{a}_i \mathbf{a}_i^T,$$

it is not difficult to see that constraints that are inactive at the optimal solution make almost no contribution toward computing the matrix as the iterate approaches an optimal solution. This suggests that good search directions can be found without involving constraints that are unlikely to be active at the optimal solution.

So in summing the outer products, we only use the constraints which seem to be most active; that is, we use  $\mathbf{M}_{(Q)}$

$$\mathbf{M}_{(Q)} := \mathbf{H} + \sum_{i \in Q} s_i^{-1} \lambda_i \mathbf{a}_i \mathbf{a}_i^T$$

instead of  $\mathbf{M}$ , where  $Q$  contains indices of constraints potentially active at the solution. Then the cost of matrix formation reduces to  $|Q|n^2/2$  multiplications.

Now the most critical part is the selection of  $Q$ . Following [TAW06], we set  $Q$  to include indices of the  $q$  smallest components of  $\mathbf{Ax} - \mathbf{b}$ , breaking ties in an arbitrary way; i.e.,

$$Q \in \mathcal{Q}(\mathbf{Ax} - \mathbf{b}, q), \tag{2.40}$$

where

$$\begin{aligned} \mathcal{Q}(\mathbf{s}, q) := \{Q \subseteq M : \text{rank}([\mathbf{H}, \mathbf{A}_Q^T]) = n \text{ and } \exists Q' \subseteq Q \text{ s.t.} \\ |Q'| = q \text{ and } s_i \leq s_j, \forall i \in Q', \forall j \notin Q', \}, \end{aligned} \tag{2.41}$$

the set containing all possible candidates of  $Q$ . Notice  $[\mathbf{H}, \mathbf{A}_Q^T]$  has full rank if and only if  $\mathcal{N}(\mathbf{H}) \cap \mathcal{N}(\mathbf{A}_Q) = \{\mathbf{0}\}$ . Obtaining  $Q$  requires sorting ( $O(m \log m)$  operations),<sup>2</sup>

---

<sup>2</sup>The complexity can be reduced to  $O(m \log |Q|)$  by using a binary heap of size  $|Q|$  and extracting the  $|Q|$  indices corresponding to the  $|Q|$  smallest entries of  $\mathbf{Ax} - \mathbf{b}$ .

which is negligible additional work compared to the matrix formation.

To guarantee a successful iteration, we need to ensure that the matrix  $\mathbf{M}_{(Q)}$  is positive definite.

**Lemma 2.3.** *(Corresponds to Lemma 2 of [TAW06]) Let  $\boldsymbol{\lambda} > \mathbf{0}$ ,  $\mathbf{s} > \mathbf{0}$ , and  $Q \subseteq M$  such that  $\text{rank}([\mathbf{H}, \mathbf{A}_Q^T]) = n$ . Then  $\mathbf{M}_{(Q)} = \mathbf{H} + \mathbf{A}_Q^T \mathbf{S}_{Q^2}^{-1} \boldsymbol{\Lambda}_{Q^2} \mathbf{A}_Q$  is positive definite.*

*Proof.* If  $[\mathbf{H}, \mathbf{A}_Q^T]$  has full rank, then  $\mathcal{N}(\mathbf{H}) \cap \mathcal{N}(\mathbf{A}_Q) = \{\mathbf{0}\}$ . Since both  $\mathbf{H}$  and  $\mathbf{A}_Q^T \mathbf{S}_{Q^2}^{-1} \boldsymbol{\Lambda}_{Q^2} \mathbf{A}_Q$  are positive semidefinite, it immediately follows that their sum is positive definite.  $\square$

Although using a very small index set  $Q$  greatly reduces the cost of matrix assembly, it makes it more likely that  $Q$  misses important constraints in early iterations. As a result, the quality of the search direction could be impaired, particularly in early iterations, resulting in an increase in the iteration count. To keep the iteration count low, we use a large number of appropriately selected constraints in early iterations, but exclude more constraints in later iterations as the complementary measure  $\mu$  becomes smaller. Specifically, based on two user-selected parameters  $q_U$  (an upper bound for  $q$ ) and  $\beta$ , with  $n \leq q_U \leq m$  and  $\beta > 0$ , we set

$$q := \begin{cases} n & , \text{ if } \mu^\beta m \leq n, \\ \lceil \mu^\beta m \rceil & , \text{ if } n < \mu^\beta m \leq q_U, \\ q_U & , \text{ if } q_U < \mu^\beta m. \end{cases} \quad (2.42)$$

This leads to Algorithm 1, borrowed from [TZ94] with the addition of constraint reduction and a slight modification in (2.51) as in [TAW06]. Notice  $q$  is determined at each iteration.

---

**Algorithm 1** Primal-Feasible Primal-Dual Affine-Scaling Quadratic Programming

---

Algorithm

---

Parameters.  $\eta \in (0, 1)$ ,  $\beta \geq 0$ ,  $\lambda_{\max}$  and  $\underline{\lambda}$  satisfying  $\lambda_{\max} > \underline{\lambda} > 0$ ,  $q_U \in \{n, \dots, m\}$ ,  $tol > 0$ .

Data.  $\mathbf{x}^0 \in \mathcal{F}_P^o$  and  $\boldsymbol{\lambda}^0 > \mathbf{0}$ .

Set

$$\mathbf{s}^0 := \mathbf{A}\mathbf{x}^0 - \mathbf{b}. \quad (2.43)$$

**for**  $k = 0, \dots$  **do**

    Compute  $\mu := \mathbf{s}^{kT}\boldsymbol{\lambda}^k/m$ .

    Terminate if

$$\frac{\|\mathbf{H}\mathbf{x}^k + \mathbf{c} - \mathbf{A}^T\boldsymbol{\lambda}^k\|}{1 + \|\boldsymbol{\lambda}^k\|} \leq tol, \frac{\|\mathbf{A}\mathbf{x}^k - \mathbf{b} - \mathbf{s}^k\|}{1 + \|\mathbf{s}^k\|} \leq tol, \text{ and } \mu \leq tol, \quad (2.44)$$

    or if  $\mathbf{H}\mathbf{x}^k + \mathbf{c} = \mathbf{0}$ .

*Step 1.* Choose the index set:

    Pick  $q$  such that  $n \leq q \leq q_U$  using (2.42) and pick  $Q \in \mathcal{Q}(\mathbf{A}\mathbf{x}^k - \mathbf{b}, q)$ .

*Step 2.* Compute a feasible descent direction  $\Delta\mathbf{x}^k$ ,  $\Delta\mathbf{s}^k$ , and  $\Delta\boldsymbol{\lambda}^k$  satisfying the reduced normal equations (2.39) and (2.34)-(2.35).

Set

$$\tilde{\boldsymbol{\lambda}}^k := \boldsymbol{\lambda}^k + \Delta \boldsymbol{\lambda}. \quad (2.45)$$

Set

$$\tilde{\boldsymbol{\lambda}}_-^k := \min\{\tilde{\boldsymbol{\lambda}}, \mathbf{0}\}. \quad (2.46)$$

*Step 3.* Updates:

Compute the largest feasible primal step length.

$$\bar{\alpha}^k := \begin{cases} \infty & \text{if } \Delta \mathbf{s} \geq \mathbf{0}, \\ \min_i \left\{ -\frac{s_i}{\Delta s_i} \mid \Delta s_i < 0, i \in M \right\} & \text{otherwise.} \end{cases} \quad (2.47)$$

Set

$$\alpha^k := \begin{cases} \eta \bar{\alpha}^k & , \text{ if } \bar{\alpha}^k - \|\Delta \mathbf{x}^k\| \leq \eta \bar{\alpha}^k < 1, \\ \bar{\alpha}^k - \|\Delta \mathbf{x}^k\| & , \text{ if } \eta \bar{\alpha}^k < \bar{\alpha}^k - \|\Delta \mathbf{x}^k\| < 1, \\ 1 & , \text{ otherwise.} \end{cases} \quad (2.48)$$

Take the step

$$\mathbf{x}^{k+1} := \mathbf{x}^k + \alpha^k \Delta \mathbf{x}^k, \quad (2.49)$$

$$\mathbf{s}^{k+1} := \mathbf{s}^k + \alpha^k \Delta \mathbf{s}^k. \quad (2.50)$$

Notice  $\mathbf{s}^{k+1} = \mathbf{A}\mathbf{x}^{k+1} - \mathbf{b}$  and  $\mathbf{s}^{k+1} > \mathbf{0}$ , since, when  $\Delta \mathbf{s} \not\geq \mathbf{0}$ ,  $\alpha^k < \bar{\alpha}^k$  and  $\mathbf{s} + \bar{\alpha}^k \Delta \mathbf{s} \geq \mathbf{0}$ .

For  $i = 1, \dots, m$ , set

$$\lambda_i^{k+1} := \begin{cases} \|\Delta \mathbf{x}^k\|^2 + \|\tilde{\boldsymbol{\lambda}}_-^k\|^2, & \text{if } \tilde{\lambda}_i^k \leq \|\Delta \mathbf{x}^k\|^2 + \|\tilde{\boldsymbol{\lambda}}_-^k\|^2 \leq \underline{\lambda}, \\ \underline{\lambda} & \text{, if } \tilde{\lambda}_i^k \leq \underline{\lambda} < \|\Delta \mathbf{x}^k\|^2 + \|\tilde{\boldsymbol{\lambda}}_-^k\|^2, \\ \tilde{\lambda}_i^k & \text{, if } \min(\|\Delta \mathbf{x}^k\|^2 + \|\tilde{\boldsymbol{\lambda}}_-^k\|^2, \underline{\lambda}) < \tilde{\lambda}_i^k \leq \lambda_{\max}, \\ \lambda_{\max} & \text{, if } \lambda_{\max} < \tilde{\lambda}_i^k. \end{cases} \quad (2.51)$$

**end for**

---

In view of Lemma 2.3, the iteration is well defined and constructs an infinite sequence if the termination criteria are ignored.

### 2.3.2 Convergence of the Adaptive Constraint-Reduction Algorithm

For the global convergence proof of the algorithm, we will impose four assumptions. The first assumption guarantees that  $\mathbf{M}_{(Q)}$  is nonsingular, a sufficient condition for solving the reduced normal equations successfully.

**Assumption 2.1.**  $[\mathbf{H}, \mathbf{A}^T]$  has full row rank.

Under this assumption, there exists  $Q \subseteq M$  such that  $[\mathbf{H}, \mathbf{A}_Q^T]$  has full row rank. Therefore  $\mathcal{Q}(\mathbf{Ax} - \mathbf{b}, q)$  is not an empty set.

To guarantee that a starting point for Algorithm 1 and a solution for the problem exist, we make the following two assumptions.

**Assumption 2.2.**  $\mathcal{F}_P^o \neq \emptyset$ .

**Assumption 2.3.**  $\mathcal{F}_P^*$  is nonempty and bounded.

We impose a constraint qualification for uniqueness of the associated dual variables  $\lambda$ .

**Assumption 2.4.**  $\forall \mathbf{x} \in \mathcal{F}_P$ ,  $\{\mathbf{a}_i^T : i \in \mathcal{A}(\mathbf{x})\}$  is a linearly independent set.

If  $\{\mathbf{a}_i^T : i \in \mathcal{A}(\mathbf{x})\}$  is a linearly independent set, then  $\mathbf{A}_{\mathcal{A}(\mathbf{x})}$  has full row rank and  $|\mathcal{A}(\mathbf{x})| \leq n$ . Assumption 2.4 guarantees that  $\mathcal{A}(\mathbf{x}) \subseteq Q$  for any  $Q \in \mathcal{Q}(\mathbf{A}\mathbf{x} - \mathbf{b}, q)$  with  $q \geq n$  and  $\mathbf{x} \in \mathcal{F}_P$ . This is a key property of  $Q$  required for the convergence proof. Under these assumptions, we can prove convergence of the algorithm.

**Theorem 2.4.**  $\{\mathbf{x}^k\}$  converges to  $\mathcal{F}_P^*$ .

*Proof.* See Appendix B.1. □

To establish a q-quadratic local convergence rate, we will impose two more assumptions.

**Assumption 2.5.**  $\mathcal{F}_P^*$  is a singleton.

**Assumption 2.6.** The Lagrange multipliers  $\lambda^*$  associated with the optimal solution  $\mathbf{x}^*$  are strictly complementary to  $\mathbf{s}^* := \mathbf{A}^T \mathbf{x}^* - \mathbf{b}$ , i.e.,  $\lambda_i^* s_i^* = 0$  and  $\lambda_i^* + s_i^* > 0$  for all  $i \in M$ .

Notice Assumption 2.5 implies that  $\mathcal{N}(\mathbf{A}_{\mathcal{A}(\mathbf{x}^*)}) \cap \mathcal{N}(\mathbf{H}) = \{\mathbf{0}\}$ , or equivalently  $[\mathbf{H}, \mathbf{A}_{\mathcal{A}(\mathbf{x}^*)}]$  spans  $\mathbb{R}^n$  for an optimal solution  $\mathbf{x}^*$ .

With these additional assumptions, we can establish a rate of convergence.

**Theorem 2.5.** *Let  $\boldsymbol{\lambda}^*$  be the Lagrange multipliers associated with the optimal solution  $\mathbf{x}^*$ . If  $\lambda_i^* < \lambda_{\max}$  for all  $i \in M$ , then  $\{(\mathbf{x}^k, \boldsymbol{\lambda}^k)\}$  converges to the primal and dual optimal solution pair  $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$   $q$ -quadratically, i.e., there exist some nonnegative integer  $k'$  and some constant  $c$  such that, for all  $k \geq k'$ ,*

$$\begin{aligned} \|\mathbf{x}^{k+1} - \mathbf{x}^*\| &\leq c\|\mathbf{x}^k - \mathbf{x}^*\|^2, \text{ and} \\ \|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*\| &\leq c\|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^*\|^2. \end{aligned}$$

*Proof.* See Appendix B.2. □

## 2.4 Extension to Infeasible Problems

The minimization problem (2.2) can be infeasible, i.e., some constraints may conflict with others and, thus, no feasible solution exists. Examples include hard margin support vector machine (SVM) training [CV95, SS01, Bur98]. Despite the infeasibility we may still want to find a solution by allowing, but limiting, violation of constraints. By adding a nonnegative relaxation variable  $y_i$  to each constraint and a penalty for

violations, we obtain the extended problem:

$$\min_{\mathbf{x}, \mathbf{y}} f_E(\mathbf{x}, \mathbf{y}) = \min_{\mathbf{x}, \mathbf{y}} \frac{1}{2} \mathbf{x}^T \mathbf{H} \mathbf{x} + \mathbf{c}^T \mathbf{x} + \mathbf{d}^T \mathbf{y} \quad (2.52)$$

$$\text{s.t. } \mathbf{A} \mathbf{x} + \mathbf{y} \geq \mathbf{b}, \quad (2.53)$$

$$\mathbf{y} \geq \mathbf{0}, \quad (2.54)$$

where  $\mathbf{y} \in \mathbb{R}^m$ ,  $\mathbf{d} \in \mathbb{R}^m$ , and  $\mathbf{d} > \mathbf{0}$ . In (2.53)  $y_i$  is the deficit of a violated constraint and in (2.52)  $d_i$  keeps  $y_i$  from growing arbitrary large. We might use a small or large penalty parameter  $d_i$  for each constraint depending on its importance. If the original problem (2.2) is feasible and each component of  $\mathbf{d}$  is large enough in (2.52), then we obtain the same optimal solution  $\mathbf{x}^*$  from the original and extended problem (2.52)-(2.54).

We will see in Chapter 3 that the soft margin SVM [CV95, Bur98, SS01, JOT07] is derived from the hard margin SVM in this way. Other problems in the extended standard form (2.52)-(2.54) include support vector regression [SS01, chap. 1], data fitting with the  $\ell_1$  norm [Wat99], and the dual of equality constrained linear programming ( $\mathbf{H} = \mathbf{0}$ ) with lower and upper bounds on variables [Wri97, chap. 11]. In this section we investigate how to apply constraint reduction effectively to problems in this form.

We can convert the extended problem (2.52)-(2.54) to the standard form (2.2) by

defining  $\mathbf{z} := [\mathbf{x}^T, \mathbf{y}^T]^T$ ,  $\hat{\mathbf{b}} = [\mathbf{b}^T, \mathbf{0}^T]^T$ ,  $\hat{\mathbf{c}} = [\mathbf{c}^T, \mathbf{d}^T]^T$ ,

$$\hat{\mathbf{A}} := \begin{bmatrix} \mathbf{A} & \mathbf{I}_m \\ \mathbf{0} & \mathbf{I}_m \end{bmatrix}, \text{ and } \hat{\mathbf{H}} := \begin{bmatrix} \mathbf{H} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix},$$

where  $\mathbf{I}_m$  is the  $m \times m$  identity matrix,  $\hat{\mathbf{A}}$  is  $2m \times (m+n)$  and  $\hat{\mathbf{H}}$  is  $(m+n) \times (m+n)$ .

We can obtain the KKT conditions for the converted standard form by defining

$\mathbf{t} := [\mathbf{s}^T, \mathbf{w}^T]^T$ ,  $\boldsymbol{\phi} := [\boldsymbol{\lambda}^T, \boldsymbol{\pi}^T]^T$ ,  $\mathbf{Y} := \text{diag}(\mathbf{y})$ ,  $\mathbf{W} := \text{diag}(\mathbf{w})$ ,  $\boldsymbol{\Pi} := \text{diag}(\boldsymbol{\pi})$ ,

$\mathbf{T} := \text{diag}(\mathbf{t})$ , and  $\boldsymbol{\Phi} := \text{diag}(\boldsymbol{\phi})$ ,

$$\hat{\mathbf{H}}\mathbf{z} - \hat{\mathbf{A}}^T\boldsymbol{\phi} + \hat{\mathbf{c}} = \mathbf{0} \quad \leftrightarrow \quad \begin{bmatrix} \mathbf{H}\mathbf{x} - \mathbf{A}^T\boldsymbol{\lambda} + \mathbf{c} = \mathbf{0} \\ -\boldsymbol{\lambda} - \boldsymbol{\pi} + \mathbf{d} = \mathbf{0} \end{bmatrix}, \quad (2.55)$$

$$\hat{\mathbf{A}}\mathbf{z} - \hat{\mathbf{b}} - \mathbf{t} = \mathbf{0} \quad \leftrightarrow \quad \begin{bmatrix} \mathbf{A}\mathbf{x} + \mathbf{y} - \mathbf{b} - \mathbf{s} = \mathbf{0} \\ \mathbf{y} - \mathbf{w} = \mathbf{0} \end{bmatrix}, \quad (2.56)$$

$$\mathbf{T}\boldsymbol{\phi} = \mathbf{0} \quad \leftrightarrow \quad \begin{bmatrix} \mathbf{S}\boldsymbol{\lambda} = \mathbf{0} \\ \mathbf{W}\boldsymbol{\pi} = \mathbf{0} \end{bmatrix}. \quad (2.57)$$

We obtain the normal equations of size  $(m+n) \times (m+n)$ , as follows:

$$(\hat{\mathbf{H}} + \hat{\mathbf{A}}^T\mathbf{T}^{-1}\boldsymbol{\Phi}\hat{\mathbf{A}})\Delta\mathbf{z} = -(\hat{\mathbf{H}}\mathbf{z} + \hat{\mathbf{c}}), \quad (2.58)$$

where we compute the other variables by solving

$$\Delta\mathbf{t} = \hat{\mathbf{A}}\Delta\mathbf{z} \quad \leftrightarrow \quad \begin{bmatrix} \Delta\mathbf{s} \\ \Delta\mathbf{w} \end{bmatrix} = \begin{bmatrix} \mathbf{A}\Delta\mathbf{x} + \Delta\mathbf{y} \\ \Delta\mathbf{y} \end{bmatrix}, \quad (2.59)$$

$$\Delta\boldsymbol{\pi} = -\boldsymbol{\phi} - \mathbf{T}^{-1}\boldsymbol{\Phi}\Delta\mathbf{t} \quad \leftrightarrow \quad \begin{bmatrix} \Delta\boldsymbol{\lambda} \\ \Delta\boldsymbol{\pi} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{\lambda} - \mathbf{S}^{-1}\boldsymbol{\Lambda}\Delta\mathbf{s} \\ -\boldsymbol{\pi} - \mathbf{W}^{-1}\boldsymbol{\Pi}\Delta\mathbf{w} \end{bmatrix}. \quad (2.60)$$

Here we maintain the primal feasibility (2.56). From (2.56) and (2.59) we see that  $\mathbf{y} + \alpha\Delta\mathbf{y} = \mathbf{w} + \alpha\Delta\mathbf{w}$  and  $\mathbf{A}(\mathbf{x} + \alpha\Delta\mathbf{x}) + \mathbf{y} + \alpha\Delta\mathbf{y} - \mathbf{b} - \mathbf{s} - \alpha\Delta\mathbf{s} = \mathbf{0}$  for any  $\alpha \in \mathbb{R}$ . So  $\mathbf{w}$  and  $\Delta\mathbf{w}$  can be replaced by  $\mathbf{y}$  and  $\Delta\mathbf{y}$ .

Similarly to the reduced normal equations (2.39) for the original standard form (2.2), we can derive reduced normal equations of size  $(m+n) \times (m+n)$  with an index set, for  $\hat{q} \geq m+n$ ,

$$\hat{Q} \in \hat{Q}(\hat{\mathbf{A}}\mathbf{z} - \hat{\mathbf{b}}, \hat{q}), \quad (2.61)$$

where

$$\begin{aligned} \hat{Q}(\mathbf{t}, \hat{q}) &:= \{\hat{Q} \subseteq \{1, \dots, 2m\} : \text{rank}([\hat{\mathbf{H}}, \hat{\mathbf{A}}_{\hat{Q}}^T]) = m+n, \text{ and } \exists \hat{Q}' \subseteq \hat{Q} \text{ s.t.} \\ &|\hat{Q}'| = \hat{q}, \text{ and } t_i \leq t_j, \forall i \in \hat{Q}', \forall j \notin \hat{Q}'\}. \end{aligned} \quad (2.62)$$

So  $\hat{Q}$  includes the indices of  $\hat{q}$  most nearly active constraints in (2.53) and (2.54). This definition of  $\hat{Q}(\mathbf{t}, \hat{q})$  is consistent with (2.41). We define three index sets derived from  $\hat{Q}$ :

$$\begin{aligned} Q_1 &:= \hat{Q} \cap M, \\ Q_2 &:= \{i > 0 : m+i \in \hat{Q}\}, \text{ and} \\ Q_3 &:= Q_1 \cap Q_2. \end{aligned} \quad (2.63)$$

Notice  $Q_1 \cup Q_2 = M$ ; otherwise the rank of  $[\hat{\mathbf{H}}, \hat{\mathbf{A}}_{\hat{Q}}^T]$  is less than  $m + n$ . So it follows that

$$|Q_3| = |Q_1 \cap Q_2| = (|Q_1| + |Q_2|) - |Q_1 \cup Q_2| = |\hat{Q}| - m. \quad (2.64)$$

The indices of the nearly active constraints in (2.53) are contained in  $Q_1$ . The indices of the nearly active constraints in (2.54) are contained in  $Q_2$ .

We begin with the reduced normal equations obtained from (2.58):

$$\left( \hat{\mathbf{H}} + \hat{\mathbf{A}}_{\hat{Q}}^T \mathbf{T}_{\hat{Q}^2}^{-1} \boldsymbol{\Phi}_{\hat{Q}^2} \hat{\mathbf{A}}_{\hat{Q}} \right) \Delta \mathbf{z} = -(\hat{\mathbf{H}} \mathbf{z} + \hat{\mathbf{c}}). \quad (2.65)$$

This system of equations is of size  $(m + n) \times (m + n)$ , but well structured. It would cost  $O(|\hat{Q}|(m + n)^2)$  multiplications to naively form the matrix on the left hand side of (2.65) (if we do not exploit the structure of  $\hat{\mathbf{A}}$ ), and the gain we could achieve through the constraint reduction would not be impressive. The cost can be reduced to  $O(|Q_1|n^2)$  when the structure of  $\hat{\mathbf{A}}$  is considered. This will become clear at (2.67). However, by further exploiting the structure of the matrix, we can make the constraint reduction even more effective. Let us see how we can derive normal equations that can most benefit from the constraint reduction.

First we expand (2.65) to

$$\left( \begin{bmatrix} \mathbf{H} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{A}_{Q_1}^T & \mathbf{0} \\ \mathbf{I}_{Q_1}^T & \mathbf{I}_{Q_2}^T \end{bmatrix} \begin{bmatrix} \mathbf{S}_{Q_1^2}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_{Q_2^2}^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Lambda}_{Q_1^2} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Pi}_{Q_2^2} \end{bmatrix} \begin{bmatrix} \mathbf{A}_{Q_1} & \mathbf{I}_{Q_1} \\ \mathbf{0} & \mathbf{I}_{Q_2} \end{bmatrix} \right) \begin{bmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{y} \end{bmatrix} = - \begin{bmatrix} \mathbf{H} \mathbf{x} + \mathbf{c} \\ \mathbf{d} \end{bmatrix}, \quad (2.66)$$

where  $\mathbf{I}_Q$  is the  $|Q| \times m$  matrix obtained by including row  $i$  of the  $m \times m$  identity matrix for all  $i \in Q$ . Simple calculation in the left hand side of (2.66) yields

$$\begin{bmatrix} \mathbf{H} + \mathbf{A}_{Q_1}^T \mathbf{S}_{Q_1}^{-1} \mathbf{\Lambda}_{Q_1^2} \mathbf{A}_{Q_1} & \mathbf{A}_{Q_1}^T \mathbf{S}_{Q_1}^{-1} \mathbf{\Lambda}_{Q_1^2} \mathbf{I}_{Q_1} \\ \mathbf{I}_{Q_1}^T \mathbf{S}_{Q_1}^{-1} \mathbf{\Lambda}_{Q_1^2} \mathbf{A}_{Q_1} & \mathbf{I}_{Q_1}^T \mathbf{S}_{Q_1}^{-1} \mathbf{\Lambda}_{Q_1^2} \mathbf{I}_{Q_1} + \mathbf{I}_{Q_2}^T \mathbf{W}_{Q_2}^{-1} \mathbf{\Pi}_{Q_2^2} \mathbf{I}_{Q_2} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{y} \end{bmatrix} = - \begin{bmatrix} \mathbf{H} \mathbf{x} + \mathbf{c} \\ \mathbf{d} \end{bmatrix}. \quad (2.67)$$

Notice that  $\mathbf{I}_Q \mathbf{d} = \mathbf{d}_Q$ , and  $\mathbf{I}_Q \mathbf{I}_Q^T = \mathbf{I}_{Q^2}$ , where  $\mathbf{I}_{Q^2}$  is the  $|Q| \times |Q|$  identity matrix.

By multiplying  $\mathbf{I}_{Q_1}$  to the second block row in (2.67), we obtain

$$\begin{bmatrix} \mathbf{H} + \mathbf{A}_{Q_1}^T \mathbf{S}_{Q_1}^{-1} \mathbf{\Lambda}_{Q_1^2} \mathbf{A}_{Q_1} & \mathbf{A}_{Q_1}^T \mathbf{S}_{Q_1}^{-1} \mathbf{\Lambda}_{Q_1^2} \mathbf{I}_{Q_1} \\ \mathbf{S}_{Q_1}^{-1} \mathbf{\Lambda}_{Q_1^2} \mathbf{A}_{Q_1} & \mathbf{S}_{Q_1}^{-1} \mathbf{\Lambda}_{Q_1^2} \mathbf{I}_{Q_1} + \mathbf{I}_{Q_1} \mathbf{I}_{Q_2}^T \mathbf{W}_{Q_2}^{-1} \mathbf{\Pi}_{Q_2^2} \mathbf{I}_{Q_2} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{y} \end{bmatrix} = - \begin{bmatrix} \mathbf{H} \mathbf{x} + \mathbf{c} \\ \mathbf{d}_{Q_1} \end{bmatrix}. \quad (2.68)$$

Since  $\mathbf{I}_Q \mathbf{S} = \mathbf{S}_{Q^2} \mathbf{I}_Q$  for any  $m \times m$  diagonal matrix  $\mathbf{S}$ , by changing the second term of the (2, 2) entry in the left hand side matrix of (2.68), we derive

$$\begin{bmatrix} \mathbf{H} + \mathbf{A}_{Q_1}^T \mathbf{S}_{Q_1}^{-1} \mathbf{\Lambda}_{Q_1^2} \mathbf{A}_{Q_1} & \mathbf{A}_{Q_1}^T \mathbf{S}_{Q_1}^{-1} \mathbf{\Lambda}_{Q_1^2} \mathbf{I}_{Q_1} \\ \mathbf{S}_{Q_1}^{-1} \mathbf{\Lambda}_{Q_1^2} \mathbf{A}_{Q_1} & \mathbf{S}_{Q_1}^{-1} \mathbf{\Lambda}_{Q_1^2} \mathbf{I}_{Q_1} + (\mathbf{I}_{Q_2}^T \mathbf{W}_{Q_2}^{-1} \mathbf{\Pi}_{Q_2^2} \mathbf{I}_{Q_2})_{Q_1} \mathbf{I}_{Q_1} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{y} \end{bmatrix} = - \begin{bmatrix} \mathbf{H} \mathbf{x} + \mathbf{c} \\ \mathbf{d}_{Q_1} \end{bmatrix}. \quad (2.69)$$

Since  $\mathbf{I}_Q \Delta \mathbf{y} = \Delta \mathbf{y}_Q$ , we get the following system of equations:

$$\begin{bmatrix} \mathbf{H} + \mathbf{A}_{Q_1}^T \mathbf{S}_{Q_1}^{-1} \mathbf{\Lambda}_{Q_1^2} \mathbf{A}_{Q_1} & \mathbf{A}_{Q_1}^T \mathbf{S}_{Q_1}^{-1} \mathbf{\Lambda}_{Q_1^2} \\ \mathbf{S}_{Q_1}^{-1} \mathbf{\Lambda}_{Q_1^2} \mathbf{A}_{Q_1} & \mathbf{S}_{Q_1}^{-1} \mathbf{\Lambda}_{Q_1^2} + (\mathbf{I}_{Q_2}^T \mathbf{W}_{Q_2}^{-1} \mathbf{\Pi}_{Q_2^2} \mathbf{I}_{Q_2})_{Q_1} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{y}_{Q_1} \end{bmatrix} = - \begin{bmatrix} \mathbf{H} \mathbf{x} + \mathbf{c} \\ \mathbf{d}_{Q_1} \end{bmatrix}. \quad (2.70)$$

Since  $Q_1 \cup Q_2 = M$  and thus  $\mathbf{S}_{Q_1}^{-1} \mathbf{\Lambda}_{Q_1} + (\mathbf{I}_{Q_2}^T \mathbf{W}_{Q_2}^{-1} \mathbf{\Pi}_{Q_2} \mathbf{I}_{Q_2})_{Q_1}^2$  is nonsingular, with a block elimination to (2.70), we can derive a system of equations of size  $n \times n$ :

$$\begin{aligned} & \left( \mathbf{H} + \mathbf{A}_{Q_1}^T \left( \mathbf{S}_{Q_1}^{-1} \mathbf{\Lambda}_{Q_1} - \mathbf{S}_{Q_1}^{-1} \mathbf{\Lambda}_{Q_1} \left( \mathbf{S}_{Q_1}^{-1} \mathbf{\Lambda}_{Q_1} + (\mathbf{I}_{Q_2}^T \mathbf{W}_{Q_2}^{-1} \mathbf{\Pi}_{Q_2} \mathbf{I}_{Q_2})_{Q_1}^2 \right)^{-1} \mathbf{S}_{Q_1}^{-1} \mathbf{\Lambda}_{Q_1} \right) \mathbf{A}_{Q_1} \right) \Delta \mathbf{x} \\ & = -\mathbf{H}\mathbf{x} - \mathbf{c} + \mathbf{A}_{Q_1}^T \mathbf{S}_{Q_1}^{-1} \mathbf{\Lambda}_{Q_1} \left( \mathbf{S}_{Q_1}^{-1} \mathbf{\Lambda}_{Q_1} + (\mathbf{I}_{Q_2}^T \mathbf{W}_{Q_2}^{-1} \mathbf{\Pi}_{Q_2} \mathbf{I}_{Q_2})_{Q_1}^2 \right)^{-1} \mathbf{d}_{Q_1}. \end{aligned} \quad (2.71)$$

Now consider this diagonal matrix:

$$\mathbf{\Theta} := \mathbf{I}_{Q_1}^T \left( \mathbf{S}_{Q_1}^{-1} \mathbf{\Lambda}_{Q_1} - \mathbf{S}_{Q_1}^{-1} \mathbf{\Lambda}_{Q_1} \left( \mathbf{S}_{Q_1}^{-1} \mathbf{\Lambda}_{Q_1} + (\mathbf{I}_{Q_2}^T \mathbf{W}_{Q_2}^{-1} \mathbf{\Pi}_{Q_2} \mathbf{I}_{Q_2})_{Q_1}^2 \right)^{-1} \mathbf{S}_{Q_1}^{-1} \mathbf{\Lambda}_{Q_1} \right) \mathbf{I}_{Q_1}.$$

For  $i \notin Q_1$ ,  $\theta_{ii} = 0$ , for  $i \in Q_1 \cap Q_2 = Q_3$ ,

$$\theta_{ii} = \frac{\lambda_i}{s_i} - \frac{\lambda_i}{s_i} \left( \frac{\lambda_i}{s_i} + \frac{\pi_i}{w_i} \right)^{-1} \frac{\lambda_i}{s_i} = \left( \frac{s_i}{\lambda_i} + \frac{w_i}{\pi_i} \right)^{-1},$$

and for  $i \in Q_1 \setminus Q_2$ ,

$$\theta_{ii} = \frac{\lambda_i}{s_i} - \frac{\lambda_i}{s_i} \left( \frac{\lambda_i}{s_i} \right)^{-1} \frac{\lambda_i}{s_i} = 0.$$

Thus,  $\mathbf{\Theta} = \mathbf{I}_{Q_3}^T (\mathbf{S}\mathbf{\Lambda}^{-1} + \mathbf{W}\mathbf{\Pi}^{-1})_{Q_3}^{-1} \mathbf{I}_{Q_3}$ .

Therefore, considering  $\mathbf{A}_{Q_1}^T = \mathbf{A}_{Q_1}^T \mathbf{I}_{Q_1} \mathbf{I}_{Q_1}^T$  and  $\mathbf{A}_{Q_1}^T \mathbf{I}_{Q_1} \mathbf{I}_{Q_3}^T = \mathbf{A}_{Q_3}^T$ , we derive the following reduced normal equations from (2.71):

$$\begin{aligned} & \left( \mathbf{H} + \mathbf{A}_{Q_3}^T (\mathbf{S}\mathbf{\Lambda}^{-1} + \mathbf{W}\mathbf{\Pi}^{-1})_{Q_3}^{-1} \mathbf{A}_{Q_3} \right) \Delta \mathbf{x} \\ & = -\mathbf{H}\mathbf{x} - \mathbf{c} + \mathbf{A}_{Q_1}^T \mathbf{S}_{Q_1}^{-1} \mathbf{\Lambda}_{Q_1} \left( \mathbf{S}_{Q_1}^{-1} \mathbf{\Lambda}_{Q_1} + (\mathbf{I}_{Q_2}^T \mathbf{W}_{Q_2}^{-1} \mathbf{\Pi}_{Q_2} \mathbf{I}_{Q_2})_{Q_1}^2 \right)^{-1} \mathbf{d}_{Q_1}. \end{aligned} \quad (2.72)$$

These are the normal equations that we actually solve to get the  $\Delta \mathbf{x}$  part of  $\Delta \mathbf{z}$ .

Notice, in forming the normal matrix in the left hand side of (2.72), only the constraints (2.53) indexed by  $Q_3$  are involved. Thus the cost of forming the matrix is

only  $O(|Q_3|n^2)$  multiplications. Compare this cost with that of (2.65) or (2.67). Even if  $|\hat{Q}|$  and  $|Q_1|$  are large,  $|Q_3|$  can be small. For example, if  $|\hat{Q}|$  is  $m+n$  and  $|Q_1|$  is  $m$ , then, in view of (2.64),  $|Q_3|$  is only  $n$ . If the constraint reduction were not used, the matrix would be formed fully, costing  $O(mn^2)$  (i.e.,  $Q_3 = M$ ). The cost of forming the right hand side of (2.72) is  $O(|Q_1|n)$ . This cost is much less than that of the full formation of the left hand side matrix.

We get  $\Delta\mathbf{y}$  by solving

$$\Delta\mathbf{y} = (\mathbf{I}_{Q_1}^T \mathbf{S}_{Q_2}^{-1} \mathbf{A}_{Q_1} \mathbf{I}_{Q_1} + \mathbf{I}_{Q_2}^T \mathbf{W}_{Q_2}^{-1} \mathbf{\Pi}_{Q_2} \mathbf{I}_{Q_2})^{-1} (\mathbf{d} - \mathbf{I}_{Q_1}^T \mathbf{S}_{Q_2}^{-1} \mathbf{A}_{Q_2} \mathbf{A}_{Q_1} \Delta\mathbf{x}), \quad (2.73)$$

derived from (2.67). To obtain other variables we simply use (2.59) and (2.60). All these equations including (2.73) consist of matrix-vector products and operations with diagonal matrices, and the cost of each is at most  $O(mn)$ . Again, this cost is much less than the full formation of the normal matrix in (2.72).

Another way to obtain the reduced normal equations (2.72) is to apply block elimination to the following reduced Newton system:

$$\hat{\mathbf{H}}\Delta\mathbf{z} - \hat{\mathbf{A}}_{\hat{Q}}^T \Delta\boldsymbol{\pi}_{\hat{Q}} = -(\hat{\mathbf{H}}\mathbf{z} + \hat{\mathbf{c}}_{\hat{Q}} - \hat{\mathbf{A}}_{\hat{Q}}^T \boldsymbol{\phi}_{\hat{Q}}), \quad (2.74)$$

$$\hat{\mathbf{A}}_{\hat{Q}} \Delta\mathbf{z} - \Delta\mathbf{t}_{\hat{Q}} = \mathbf{0}, \quad (2.75)$$

$$\mathbf{T}_{\hat{Q}_2} \Delta\boldsymbol{\pi}_{\hat{Q}} + \boldsymbol{\Phi}_{\hat{Q}_2} \Delta\mathbf{t}_{\hat{Q}} = -\mathbf{T}_{\hat{Q}_2} \boldsymbol{\phi}_{\hat{Q}}. \quad (2.76)$$

Since (2.72) and (2.73) are derived from (2.65), the step  $\Delta\mathbf{z} := [\Delta\mathbf{x}^T, \Delta\mathbf{y}^T]^T$  generated from the reduced normal equations (2.72) and (2.73) is the same as the one

generated from the standard-form-like reduced normal equations (2.65), and satisfies the reduced Newton system above.

So we can define a constraint-reduced affine-scaling primal-dual interior-point algorithm for the extended problem (2.52)-(2.54), which actually solves the  $n$  by  $n$  normal equations (2.72), by using  $\hat{Q}$  and its by-products  $Q_1$ ,  $Q_2$  and  $Q_3$  as defined in (2.62) and (2.63); by solving (2.72), (2.73), (2.59), and (2.60); and by substituting  $\mathbf{z}$  for  $\mathbf{x}$ ,  $\mathbf{t}$  for  $\mathbf{s}$ , and  $\boldsymbol{\lambda}$  for  $\boldsymbol{\phi}$  in Algorithm 1, where, however,  $\mathbf{w}$  and  $\Delta\mathbf{w}$  do not need to be tracked as they are always the same as  $\mathbf{y}$  and  $\Delta\mathbf{y}$ . Therefore by imposing Assumptions 2.1-2.6 to the standard form converted from (2.52)-(2.54), we can extend the convergence analysis to the algorithm for the extended problem. Assumption 2.2 is not necessary because the problem (2.52)-(2.54) is always strictly feasible.

Here we state the modified assumptions for the convergence proof of the algorithm for the extended problem (2.52)-(2.54). For this we define

$$\begin{aligned}\tilde{\mathcal{F}}_P &:= \{(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{n+m} : \mathbf{A}\mathbf{x} + \mathbf{y} \geq \mathbf{b} \text{ and } \mathbf{y} \geq \mathbf{0}\}, \\ \tilde{\mathcal{F}}_P^o &:= \{(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{n+m} : \mathbf{A}\mathbf{x} + \mathbf{y} > \mathbf{b} \text{ and } \mathbf{y} > \mathbf{0}\}, \text{ and} \\ \tilde{\mathcal{F}}_P^* &:= \{(\mathbf{x}^*, \mathbf{y}^*) \in \tilde{\mathcal{F}}_P : f_E(\mathbf{x}^*, \mathbf{y}^*) \leq f_E(\mathbf{x}, \mathbf{y}), \forall (\mathbf{x}, \mathbf{y}) \in \tilde{\mathcal{F}}_P\},\end{aligned}$$

similarly to (2.4), (2.5) and (2.6). We define  $\hat{\mathcal{A}}(\mathbf{z}) \subseteq \{1, \dots, 2m\}$ , the index set of

active constraints at  $\mathbf{z} = [\mathbf{x}^T, \mathbf{y}^T]^T$ , similarly to (2.7). We also define its byproducts

$$\mathcal{A}_1(\mathbf{x}, \mathbf{y}) := \hat{\mathcal{A}}(\mathbf{z}) \cap M,$$

$$\mathcal{A}_2(\mathbf{x}, \mathbf{y}) := \{i > 0 : m + i \in \hat{\mathcal{A}}(\mathbf{z})\},$$

$$\mathcal{A}_3(\mathbf{x}, \mathbf{y}) := \mathcal{A}_1(\mathbf{x}, \mathbf{y}) \cap \mathcal{A}_2(\mathbf{x}, \mathbf{y}).$$

Clearly, under Assumption 2.1,  $[\hat{\mathbf{H}}, \hat{\mathbf{A}}^T]$  also has full row rank. Thus  $\hat{\mathcal{Q}}(\hat{\mathbf{A}}\mathbf{z} - \hat{\mathbf{b}})$  is not empty. Since, for  $\mathbf{T}^{-1}\boldsymbol{\phi} > \mathbf{0}$ ,  $\hat{\mathbf{H}} + \hat{\mathbf{A}}_{\hat{\mathcal{Q}}}^T \mathbf{T}_{\hat{\mathcal{Q}}_2}^{-1} \boldsymbol{\Phi}_{\hat{\mathcal{Q}}_2} \hat{\mathbf{A}}_{\hat{\mathcal{Q}}}$  in (2.65) is positive definite due to full row rank of  $[\hat{\mathbf{H}}, \hat{\mathbf{A}}_{\hat{\mathcal{Q}}}^T]$ ,  $\mathbf{H} + \mathbf{A}_{Q_3}^T (\mathbf{S}\boldsymbol{\Lambda}^{-1} + \mathbf{W}\boldsymbol{\Pi}^{-1})_{Q_3}^{-1} \mathbf{A}_{Q_3}$  in (2.72) should be positive definite as well. The following proposition supports this fact.

**Proposition 2.6.** *Let  $\hat{\mathcal{Q}} \subseteq \{1, \dots, 2m\}$ . Define  $Q_1, Q_2$  and  $Q_3$  as in (2.63). Suppose that  $Q_1 \cup Q_2 = M$ . Then  $[\mathbf{H}, \mathbf{A}_{Q_3}^T]$  has full row rank if and only if  $[\hat{\mathbf{H}}, \hat{\mathbf{A}}_{\hat{\mathcal{Q}}}^T]$  does.*

*Proof.* Let us first prove the sufficiency. Suppose that  $[\mathbf{H}, \mathbf{A}_{Q_3}^T]$  does not have full row rank. Then, there exists some nonzero vector  $\mathbf{v} \in \mathbb{R}^n$  such that  $\mathbf{H}\mathbf{v} = \mathbf{0}$  and  $\mathbf{A}_{Q_3}\mathbf{v} = \mathbf{0}$ .

Let  $\mathbf{u} \in \mathbb{R}^m$  be such that  $\mathbf{u}_{Q_1 \setminus Q_3} = -\mathbf{A}_{Q_1 \setminus Q_3}\mathbf{v}$  and  $\mathbf{u}_{Q_2} = \mathbf{0}$ . Use  $\mathbf{I}_Q \mathbf{u} = \mathbf{u}_Q$  and use an  $(m + n + |\hat{\mathcal{Q}}|)$  by  $(m + n + |\hat{\mathcal{Q}}|)$  pivoting matrix  $\mathbf{P}$  such that

$$\mathbf{P} \begin{bmatrix} \hat{\mathbf{H}} \\ \hat{\mathbf{A}}_{\hat{\mathcal{Q}}} \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{H} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{A}_{Q_3} & \mathbf{I}_{Q_3} \\ \mathbf{A}_{Q_1 \setminus Q_3} & \mathbf{I}_{Q_1 \setminus Q_3} \\ \mathbf{0} & \mathbf{I}_{Q_2} \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{H}\mathbf{v} \\ \mathbf{0} \\ \mathbf{A}_{Q_3}\mathbf{v} + \mathbf{u}_{Q_3} \\ \mathbf{A}_{Q_1 \setminus Q_3}\mathbf{v} + \mathbf{u}_{Q_1 \setminus Q_3} \\ \mathbf{u}_{Q_2} \end{bmatrix}.$$

Then we know  $[\mathbf{v}^T, \mathbf{u}^T][\hat{\mathbf{H}}, \hat{\mathbf{A}}_{\hat{Q}}^T]\mathbf{P}^T = \mathbf{0}$ . Since  $[\mathbf{v}^T, \mathbf{u}^T]^T \neq \mathbf{0}$ , therefore  $[\hat{\mathbf{H}}, \hat{\mathbf{A}}_{\hat{Q}}^T]$  does not have full row rank.

Now let us consider the necessity. Let  $\mathbf{v} \in \mathbb{R}^n, \mathbf{u} \in \mathbb{R}^m$  be such that

$$[\hat{\mathbf{H}}, \hat{\mathbf{A}}_{\hat{Q}}^T]^T[\mathbf{v}^T, \mathbf{u}^T]^T = \mathbf{0}.$$

We show that  $[\mathbf{v}^T, \mathbf{u}^T]^T = \mathbf{0}$ , proving the claim. Indeed, there exists some pivoting matrix  $\mathbf{P}$  such that

$$\mathbf{P} \begin{bmatrix} \hat{\mathbf{H}} \\ \hat{\mathbf{A}}_{\hat{Q}} \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{H} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \\ \mathbf{A}_{Q_3} & \mathbf{I}_{Q_3} \\ \mathbf{A}_{Q_1 \setminus Q_3} & \mathbf{I}_{Q_1 \setminus Q_3} \\ \mathbf{0} & \mathbf{I}_{Q_3} \\ \mathbf{0} & \mathbf{I}_{Q_2 \setminus Q_3} \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{H}\mathbf{v} \\ \mathbf{0} \\ \mathbf{A}_{Q_3}\mathbf{v} + \mathbf{u}_{Q_3} \\ \mathbf{A}_{Q_1 \setminus Q_3}\mathbf{v} + \mathbf{u}_{Q_1 \setminus Q_3} \\ \mathbf{u}_{Q_3} \\ \mathbf{u}_{Q_2 \setminus Q_3} \end{bmatrix} = \mathbf{0}.$$

So it follows that  $\mathbf{H}\mathbf{v} = \mathbf{0}$  and  $\mathbf{A}_{Q_3}\mathbf{v} = \mathbf{0}$ . Since  $[\mathbf{H}, \mathbf{A}_{Q_3}^T]$  has full row rank, it follows that  $\mathbf{v} = \mathbf{0}$  and  $\mathbf{u} = \mathbf{0}$ .  $\square$

**Assumption 2.7.** For all  $\mathbf{z} = [\mathbf{x}^T, \mathbf{y}^T]^T$  in  $\tilde{\mathcal{F}}_P$ ,  $\{\mathbf{a}_i^T : i \in \mathcal{A}_3(\mathbf{x}, \mathbf{y})\}$  is a linearly independent set.

The following proposition will show the equivalence between Assumption 2.4 (with  $\mathbf{x}, \mathbf{a}_i, \mathcal{A}(\mathbf{x}), \mathcal{F}_P$  replaced by  $\mathbf{z}, \hat{\mathbf{a}}_i, \hat{\mathcal{A}}(\mathbf{z}), \tilde{\mathcal{F}}_P$ ) and Assumption 2.7.

**Proposition 2.7.**  $\{\mathbf{a}_i^T : i \in \mathcal{A}_3(\mathbf{x}, \mathbf{y})\}$  is a linearly independent set if and only if  $\{\hat{\mathbf{a}}_i^T : i \in \hat{\mathcal{A}}(\mathbf{z})\}$  is a linearly independent set.

*Proof.* For convenience, given  $\mathbf{z} = [\mathbf{x}^T, \mathbf{y}^T]^T$ , we let  $\hat{\mathcal{A}} := \hat{\mathcal{A}}(\mathbf{z})$ ,  $\mathcal{A}_1 := \mathcal{A}_1(\mathbf{x}, \mathbf{y})$ ,  $\mathcal{A}_2 := \mathcal{A}_2(\mathbf{x}, \mathbf{y})$  and  $\mathcal{A}_3 := \mathcal{A}_3(\mathbf{x}, \mathbf{y})$ . Let us first show the sufficiency. Suppose that  $\{\mathbf{a}_i^T : i \in \mathcal{A}_3\}$  is not a linearly independent set. This implies that  $\mathcal{N}(\mathbf{A}_{\mathcal{A}_3}^T) \neq \{\mathbf{0}\}$ . So there exists some nonzero vector  $\bar{\mathbf{u}} \in \mathbb{R}^{|\mathcal{A}_3|}$  such that  $\mathbf{A}_{\mathcal{A}_3}^T \bar{\mathbf{u}} = \mathbf{0}$ . Let  $\mathbf{u} \in \mathbb{R}^m$  be such that  $\mathbf{u}_{\mathcal{A}_3} = \bar{\mathbf{u}}$  and  $\mathbf{u}_{M \setminus \mathcal{A}_3} = \mathbf{0}$ . So we know that  $\mathbf{u}_{\mathcal{A}_1} \neq \mathbf{0}$  and  $\mathbf{A}_{\mathcal{A}_3}^T \mathbf{u}_{\mathcal{A}_3} = \mathbf{0}$ . Then, since  $\mathcal{A}_1 \subseteq M$ , it follows that  $\mathbf{A}_{\mathcal{A}_1}^T \mathbf{u}_{\mathcal{A}_1} = \mathbf{A}_{\mathcal{A}_1 \setminus \mathcal{A}_3}^T \mathbf{u}_{\mathcal{A}_1 \setminus \mathcal{A}_3} + \mathbf{A}_{\mathcal{A}_3}^T \mathbf{u}_{\mathcal{A}_3} = \mathbf{0}$ . Let  $\mathbf{v} := -\mathbf{u}$  and compute  $\hat{\mathbf{A}}_{\hat{\mathcal{A}}}^T [\mathbf{u}_{\mathcal{A}_1}^T, \mathbf{v}_{\mathcal{A}_2}^T]^T$ . Since  $\mathbf{u}_{\mathcal{A}_1 \setminus \mathcal{A}_3} = \mathbf{0}$ ,  $\mathbf{v}_{\mathcal{A}_2 \setminus \mathcal{A}_3} = \mathbf{0}$  and  $\mathbf{u}_{\mathcal{A}_3} = -\mathbf{v}_{\mathcal{A}_3}$ , it follows that

$$\hat{\mathbf{A}}_{\hat{\mathcal{A}}}^T \begin{bmatrix} \mathbf{u}_{\mathcal{A}_1} \\ \mathbf{v}_{\mathcal{A}_2} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{\mathcal{A}_1}^T & \mathbf{0} \\ \mathbf{I}_{\mathcal{A}_1}^T & \mathbf{I}_{\mathcal{A}_2}^T \end{bmatrix} \begin{bmatrix} \mathbf{u}_{\mathcal{A}_1} \\ \mathbf{v}_{\mathcal{A}_2} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{\mathcal{A}_1}^T \mathbf{u}_{\mathcal{A}_1} \\ \mathbf{I}_{\mathcal{A}_1}^T \mathbf{u}_{\mathcal{A}_1} + \mathbf{I}_{\mathcal{A}_2}^T \mathbf{v}_{\mathcal{A}_2} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}.$$

Since  $[\mathbf{u}_{\mathcal{A}_1}^T, \mathbf{v}_{\mathcal{A}_2}^T]^T \neq \mathbf{0}$ , this means that  $\{\hat{\mathbf{a}}_i^T : i \in \hat{\mathcal{A}}\}$  is not a linearly independent set.

Now let us prove the necessity. Assume  $\{\mathbf{a}_i^T : i \in \mathcal{A}_3\}$  is linearly independent. Let  $\mathbf{u}, \mathbf{v} \in \mathbb{R}^m$  be such that  $\hat{\mathbf{A}}_{\hat{\mathcal{A}}}^T [\mathbf{u}_{\mathcal{A}_1}^T, \mathbf{v}_{\mathcal{A}_2}^T]^T = \mathbf{0}$ . We show that  $[\mathbf{u}_{\mathcal{A}_1}^T, \mathbf{v}_{\mathcal{A}_2}^T]^T = \mathbf{0}$ , proving the claim. Indeed, since

$$\begin{aligned} \hat{\mathbf{A}}_{\hat{\mathcal{A}}}^T \begin{bmatrix} \mathbf{u}_{\mathcal{A}_1} \\ \mathbf{v}_{\mathcal{A}_2} \end{bmatrix} &= \begin{bmatrix} \mathbf{A}_{\mathcal{A}_1}^T & \mathbf{0} \\ \mathbf{I}_{\mathcal{A}_1}^T & \mathbf{I}_{\mathcal{A}_2}^T \end{bmatrix} \begin{bmatrix} \mathbf{u}_{\mathcal{A}_1} \\ \mathbf{v}_{\mathcal{A}_2} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{\mathcal{A}_1}^T \mathbf{u}_{\mathcal{A}_1} \\ \mathbf{I}_{\mathcal{A}_1}^T \mathbf{u}_{\mathcal{A}_1} + \mathbf{I}_{\mathcal{A}_2}^T \mathbf{v}_{\mathcal{A}_2} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{A}_{\mathcal{A}_1 \setminus \mathcal{A}_3}^T \mathbf{u}_{\mathcal{A}_1 \setminus \mathcal{A}_3} + \mathbf{A}_{\mathcal{A}_3}^T \mathbf{u}_{\mathcal{A}_3} \\ \mathbf{I}_{\mathcal{A}_1}^T \mathbf{u}_{\mathcal{A}_1} + \mathbf{I}_{\mathcal{A}_2}^T \mathbf{v}_{\mathcal{A}_2} \end{bmatrix}, \end{aligned}$$

we conclude that  $\mathbf{I}_{\mathcal{A}_1}^T \mathbf{u}_{\mathcal{A}_1} + \mathbf{I}_{\mathcal{A}_2}^T \mathbf{v}_{\mathcal{A}_2} = \mathbf{0}$ , from which it immediately follows that  $\mathbf{u}_{\mathcal{A}_1 \setminus \mathcal{A}_3} = \mathbf{0}$ ,  $\mathbf{v}_{\mathcal{A}_2 \setminus \mathcal{A}_3} = \mathbf{0}$ , and  $\mathbf{u}_{\mathcal{A}_3} = -\mathbf{v}_{\mathcal{A}_3}$ . Then since  $\mathbf{A}_{\mathcal{A}_1 \setminus \mathcal{A}_3}^T \mathbf{u}_{\mathcal{A}_1 \setminus \mathcal{A}_3} + \mathbf{A}_{\mathcal{A}_3}^T \mathbf{u}_{\mathcal{A}_3} =$

$\mathbf{A}_{\mathcal{A}_3}^T \mathbf{u}_{\mathcal{A}_3} = \mathbf{0}$ , it follows, from the linear independence of  $\{\mathbf{a}_i^T : i \in \mathcal{A}_3\}$ , that  $\mathbf{u}_{\mathcal{A}_3} = \mathbf{0}$ .

Thus  $[\mathbf{u}_{\mathcal{A}_1}^T, \mathbf{v}_{\mathcal{A}_2}^T]^T = \mathbf{0}$ .  $\square$

**Theorem 2.8.** *Let  $\{(\mathbf{x}^k, \mathbf{y}^k)\}$  (or  $\{\mathbf{z}^k\}$ ) be the sequence constructed by Algorithm 1 applied to the converted standard form. Under Assumptions 2.1, 2.3 (with  $\mathcal{F}_P^*$  replaced by  $\tilde{\mathcal{F}}_P^*$ ) and 2.7, the sequence  $(\mathbf{x}^k, \mathbf{y}^k)$  converges to  $\tilde{\mathcal{F}}_P^*$ .*

*Proof.* Due to Proposition 2.6 (by setting  $\hat{Q} := \{1, \dots, 2m\}$ ), Assumption 2.1 implies that  $[\hat{\mathbf{H}}, \hat{\mathbf{A}}^T]$  has full row rank.  $\tilde{\mathcal{F}}_P^o$  is trivially nonempty. Due to Proposition 2.7, Assumption 2.7 implies that  $\{\hat{\mathbf{a}}_i^T : i \in \hat{\mathcal{A}}(\mathbf{z})\}$  is a linearly independent set for all  $\mathbf{z} \in \tilde{\mathcal{F}}_P$ . Hence Assumptions 2.1-2.4 hold with  $\mathbf{H}$ ,  $\mathbf{A}$ ,  $\mathbf{a}_i$ ,  $\mathbf{x}$ ,  $\mathcal{A}(\mathbf{x})$ ,  $\mathcal{F}_P$ ,  $\mathcal{F}_P^o$ , and  $\mathcal{F}_P^*$  replaced by  $\hat{\mathbf{H}}$ ,  $\hat{\mathbf{A}}$ ,  $\hat{\mathbf{a}}_i$ ,  $\mathbf{z}$ ,  $\hat{\mathcal{A}}(\mathbf{z})$ ,  $\tilde{\mathcal{F}}_P$ ,  $\tilde{\mathcal{F}}_P^o$ , and  $\tilde{\mathcal{F}}_P^*$ . Since the set of index sets  $\hat{Q}(\hat{\mathbf{A}}\mathbf{z} - \hat{\mathbf{b}}, \hat{q})$  is consistent with  $Q(\mathbf{A}\mathbf{x} - \mathbf{b}, q)$ , the claim follows from Theorem 2.4.  $\square$

**Assumption 2.8.** *(Corresponds to Assumption 2.6) Strict complementarity holds at the optimal solution  $(\mathbf{x}^*, \mathbf{y}^*)$ .*

**Theorem 2.9.** *Let  $\{(\mathbf{x}^k, \mathbf{y}^k)\}$  (or  $\{\mathbf{z}^k\}$ ) be the sequence constructed by Algorithm 1 applied to the converted standard form. Suppose that Assumptions 2.1, 2.3 (with  $\mathcal{F}_P^*$  replaced by  $\tilde{\mathcal{F}}_P^*$ ), 2.7, 2.5 (with  $\mathcal{F}_P^*$  replaced by  $\tilde{\mathcal{F}}_P^*$ ), and 2.8 hold. If  $\lambda_i < \phi_{\max}$  and  $\pi_i < \phi_{\max}$  for all  $i \in M$ , then the sequence  $\{(\mathbf{x}^k, \mathbf{y}^k, \boldsymbol{\lambda}^k, \boldsymbol{\pi}^k)\}$  (or  $\{(\mathbf{z}^k, \boldsymbol{\phi}^k)\}$ ) converges to the primal-dual solution pair  $(\mathbf{x}^*, \mathbf{y}^*, \boldsymbol{\lambda}^*, \boldsymbol{\pi}^*)$  (or  $(\mathbf{z}^*, \boldsymbol{\phi}^*)$ )  $q$ -quadratically.*

*Proof.* Assumptions 2.1-2.6 hold with  $\mathbf{H}$ ,  $\mathbf{A}$ ,  $\mathbf{a}_i$ ,  $\mathbf{b}$ ,  $\mathbf{x}$ ,  $\mathcal{A}(\mathbf{x})$ ,  $\mathcal{F}_P$ ,  $\mathcal{F}_P^o$ ,  $\mathcal{F}_P^*$ ,  $\boldsymbol{\lambda}^*$ ,  $\mathbf{s}^*$ ,  $\mathbf{x}^*$ , and  $M$  replaced by  $\hat{\mathbf{H}}$ ,  $\hat{\mathbf{A}}$ ,  $\hat{\mathbf{a}}_i$ ,  $\hat{\mathbf{b}}$ ,  $\mathbf{z}$ ,  $\hat{\mathcal{A}}(\mathbf{z})$ ,  $\tilde{\mathcal{F}}_P$ ,  $\tilde{\mathcal{F}}_P^o$ ,  $\tilde{\mathcal{F}}_P^*$ ,  $\boldsymbol{\phi}^*$ ,  $\mathbf{t}^*$ ,  $\mathbf{z}^*$ , and  $\{1, \dots, 2m\}$ .

Then the claim follows directly from Theorem 2.5. □

It is possible to extend the convergence results to problems that also include an  $\ell_2$  penalty term in the objective function. A term  $\mathbf{y}^T \text{diag}(\mathbf{g})\mathbf{y}$  would then be added in the objective function (2.52), where  $\mathbf{g} \in \mathbb{R}^m$ ,  $\mathbf{d} \geq \mathbf{0}$ ,  $\mathbf{g} \geq \mathbf{0}$ , and  $\mathbf{d} + \mathbf{g} > \mathbf{0}$ . The same index set choice as defined in (2.62) can be used. By following the same steps, a constraint reduced algorithm can be devised. Problems of this type include support vector machine training with squared hinge loss [SS01, FM02].

## 2.5 Numerical Results

We implemented Algorithm 1 in MATLAB 7.1 R14 SP3 with a dense direct solver for the normal equations in order to concentrate on the action of the reduction. The algorithm was tested on a machine with an Intel Pentium IV 2.8GHz processor with 16 KB L1 cache, 1 MB L2 cache, 2.5 GB DDR2-400MHz configured as dual channel, and Hyper Threading enabled. The machine ran Windows XP SP2.

We set the algorithm to terminate when either convergence was detected or more than 200 iterations were performed. We set the parameters as  $\beta := 4$  for controlling reduction speed, and  $tol := 10^{-8}$ . We set  $\underline{\lambda} := 10^{-6}$ ,  $\lambda_{\max} := 10^{30}$ ,  $\eta := .98$ , and  $\theta := 10^2$ . We varied  $q_U$  to see how our algorithm would behave depending on it. Following [TAW06], we also enforced a “safeguard” on  $\mathbf{s}$ ,  $s_i := \max(10^{-14}, s_i)$ , for the purpose of assembling  $\mathbf{M}_{(Q)}$ . This keeps  $\mathbf{M}_{(Q)}$  from being too ill-conditioned.

In addition, when the Cholesky factorization routine `chol` failed to factor  $\mathbf{M}_{(Q)}$  for  $Q = M$  due to numerical difficulty, we used the Cholesky infinity factorization `cholinc` instead [Zha96].

### 2.5.1 Choosing $Q$

We had the algorithm choose a reasonably small  $Q \in \mathcal{Q}(\mathbf{s}, q)$  as follows. First  $\bar{Q}$  is taken to be the set of indices of some  $\bar{q} := q$  smallest slacks  $\mathbf{a}_i^T \mathbf{x} - b_i$ . Then, if  $\text{rank}([\mathbf{H}, \mathbf{A}_{\bar{Q}}]) = r < n$ ,  $\mathbf{M}_{\bar{Q}}$  becomes singular with rank  $r$ . With  $Q := \bar{Q}$ , solving the reduced normal equations (2.39) fails. In this event, we may calculate a unique Cholesky factor whose rows after the  $r$ th row are filled with zeros [Hou64, Hig90], and repeatedly perform a low rank update with the next most active constraints [WNT07]. However, for ease of implementation, we instead had the algorithm repeat doubling  $\bar{q}$  and choosing  $\bar{Q}$  as the set of indices of some  $\bar{q}$  smallest slacks until  $\mathbf{M}_{\bar{Q}}$  becomes nonsingular. Then we set  $Q := \bar{Q}$ . Since  $Q$  contains the indices of some  $q$  smallest slacks and  $\text{rank}([\mathbf{H}, \mathbf{A}_Q]) = n$ , we know that  $Q \in \mathcal{Q}(\mathbf{s}, q)$ .

### 2.5.2 Scaling

Rows or columns of the coefficient matrix  $\mathbf{A}$  are often associated with different measurement units. For instance, consider an optimization problem of eating food with various ingredients. Each constraint of the problem may restrict the consumption of the total amount of a nutrient measured in grams. Various nutrients are contained

in an ingredient, which might contain a relatively large amount of carbohydrate, protein and fat but a very small amount of minerals and vitamins. Then, values in some rows of the coefficient matrix are large while those in the others may be very small. Since the condition number of the matrix depends on scaling of rows, bad scaling can cause numerical instability. To balance the entries in the matrix, we may use different measurement units for each row.

Let us first consider scaling rows of the matrix  $\mathbf{A}$  in (2.2). Define  $\mathbf{D}$  to be an  $m \times m$  diagonal matrix with positive entries. By multiplying each constraint by  $d_{ii}$ , we see that the constraints  $\mathbf{Ax} \geq \mathbf{b}$  are equivalent to  $\mathbf{DAx} \geq \mathbf{Db}$ .

Now consider affine transformation of the problem space including scaling columns of the matrix  $\mathbf{DA}$ . Define  $\mathbf{P}$  to be a nonsingular  $n \times n$  matrix. Then the equivalent constraints are  $(\mathbf{DAP})(\mathbf{P}^{-1}\mathbf{x}) \geq \mathbf{Db}$ . If  $\mathbf{P}$  is diagonal, then it scales columns of  $\mathbf{A}$ . Furthermore, we may translate the origin of the problem space to a point  $\mathbf{v} \in \mathbb{R}^n$  before the linear transformation. Then the problem (2.2) is solved in terms of  $\bar{\mathbf{x}} := \mathbf{P}^{-1}(\mathbf{x} - \mathbf{v})$ . We obtain the following problem, which is equivalent to the original one up to a constant in the objective function:

$$\begin{aligned} \min_{\bar{\mathbf{x}}} \quad & \frac{1}{2} \bar{\mathbf{x}}^T (\mathbf{P}^T \mathbf{H} \mathbf{P}) \bar{\mathbf{x}} + (\mathbf{H}\mathbf{v} + \mathbf{c})^T \mathbf{P} \bar{\mathbf{x}}, \\ \text{s.t.} \quad & \mathbf{DAP} \bar{\mathbf{x}} \geq \mathbf{D}(\mathbf{b} - \mathbf{Av}). \end{aligned}$$

There is no optimal rule for scaling that fits every problem; insight into the specific problem is required.

When  $\mathbf{H} = \mathbf{0}$ , Algorithm 1 and the choice of initial guess can be modified to

make the iteration invariant to positive diagonal  $\mathbf{D}$ , nonsingular diagonal  $\mathbf{P}$ , and  $\mathbf{v}$  [TAW06]. The modification also identically applies to Algorithm 1 with  $\mathbf{H} \neq \mathbf{0}$ . Another way to make the iteration invariant under some types of scaling is to preprocess the problem, which we did in our numerical experiments. For this, we used the following heuristic. We normalized every row of  $\mathbf{A}$  to length 1, and scaled  $\mathbf{b}$  accordingly. In other words, we set  $d_{ii} := \frac{1}{\|\mathbf{a}_i\|}$  for  $i = 1, \dots, m$ . Since scaling columns of  $\mathbf{A}$  affects the normalized rows, we chose not to scale the columns. So we set  $\mathbf{P} := \mathbf{I}$  and  $\mathbf{v} := \mathbf{0}$ . Since every constraint is now assumed to be well scaled, the dual initial point was set as  $\boldsymbol{\lambda}^0 := \mathbf{e}$ . The Lagrange multipliers for the original problem can be recovered by using  $\boldsymbol{\lambda} := \mathbf{D}\boldsymbol{\lambda}$ . This preprocessing makes the iteration *mathematically* invariant (under exact arithmetic) to the action of  $\mathbf{D}$  and  $\mathbf{v}$  with a proper choice of a primal initial point.

### 2.5.3 Data Fitting

Data fitting is a problem of finding a model approximating time series data  $\bar{b}_1, \dots, \bar{b}_m$  measured at times  $t_1, \dots, t_m$ . We build a model with a set of basis functions  $\psi_1(t), \dots, \psi_n(t)$ :

$$u(t) := \sum_{j=1}^n \bar{x}_j \psi_j(t).$$

To find good coefficients  $\bar{x}_1, \dots, \bar{x}_n$ , we can use Chebyshev approximation, minimizing the maximal error of the model [Atk89, Wat99]. Let  $\bar{\mathbf{A}}$  be the  $m \times n$  matrix with

entries  $\bar{a}_{kj} := \psi_j(t_k)$ . If we form a vector  $\bar{\mathbf{b}}$  from the values  $\bar{b}_k$  and a vector  $\bar{\mathbf{x}}$  from the coefficients  $\bar{x}_j$ , then the maximal error can be written as

$$\max_{k \in \{1, \dots, \bar{m}\}} |\bar{b}_k - u(t_k)| = \|\bar{\mathbf{b}} - \bar{\mathbf{A}}\bar{\mathbf{x}}\|_\infty,$$

and we want to minimize this over all choices of  $\bar{\mathbf{x}}$ .

A model obtained by solving this *min-max* problem could be too sensitive to noise in the measurements. To reduce the sensitivity, we can utilize a regularization method, first introduced by Tikhonov [TA77]. By adding a regularization term in the objective function, we obtain the regularized min-max problem,

$$\min_{\bar{\mathbf{x}}} \|\bar{\mathbf{b}} - \bar{\mathbf{A}}\bar{\mathbf{x}}\|_\infty + \frac{1}{2}\alpha\|\bar{\mathbf{x}}\|_{\bar{\mathbf{H}}^{1/2}}^2,$$

where  $\bar{\mathbf{H}}^{1/2}$  is an  $\bar{n} \times \bar{n}$  symmetric positive semidefinite matrix,  $\|\bar{\mathbf{x}}\|_{\bar{\mathbf{H}}^{1/2}} := \sqrt{\bar{\mathbf{x}}^T \bar{\mathbf{H}} \bar{\mathbf{x}}}$  and  $\alpha$  is a scalar value. This can be transformed to the standard form:

$$\begin{aligned} \min_{\bar{\mathbf{x}}, t} \quad & t + \frac{1}{2}\alpha\|\bar{\mathbf{x}}\|_{\bar{\mathbf{H}}}^2 \\ \text{s.t.} \quad & \bar{\mathbf{A}}\bar{\mathbf{x}} - \bar{\mathbf{b}} \geq -t\mathbf{e} \\ & -\bar{\mathbf{A}}\bar{\mathbf{x}} + \bar{\mathbf{b}} \geq -t\mathbf{e}. \end{aligned} \tag{2.77}$$

If we define  $\mathbf{x} := [\bar{\mathbf{x}}^T, t]^T$ ,  $\mathbf{b} := [\bar{\mathbf{b}}^T, -\bar{\mathbf{b}}^T]^T$ ,  $\mathbf{c} := [0, \dots, 0, 1]^T$ ,  $m = 2\bar{m}$ , and  $n = \bar{n} + 1$ , then

$$\mathbf{A} = \begin{bmatrix} \bar{\mathbf{A}} & \mathbf{e} \\ -\bar{\mathbf{A}} & \mathbf{e} \end{bmatrix} \text{ and } \mathbf{H} = \begin{bmatrix} \alpha\bar{\mathbf{H}} & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix}.$$

We used the problem setting of [WNT07]. For convenience, we restate it here. For basis functions, we used cosine and sine functions: for  $j = 0, \dots, \bar{n}$ ,

$$\psi_j^c(t) := \cos(2j\pi t),$$

and, for  $j = 1, \dots, \bar{n}$ ,

$$\psi_j^s(t) := \sin(2j\pi t).$$

Then, we set the basis functions as

$$[\psi_1, \dots, \psi_{2\bar{n}+1}] := [\psi_0^c, \dots, \psi_{\bar{n}}^c, \psi_1^s, \dots, \psi_{\bar{n}}^s].$$

The sampling points were, for  $i = 1, \dots, \bar{m}$ ,

$$t_i := (i - 1)/\bar{m}.$$

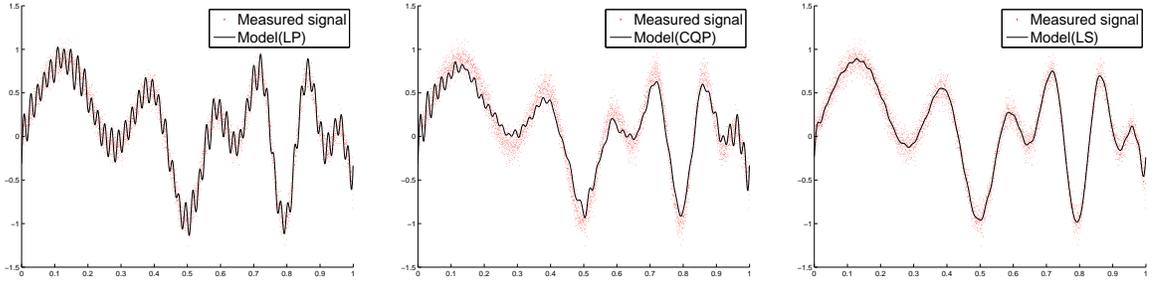
For observed data, we used the following signal function

$$g(t) := \sin(10t) \cos(25t^2),$$

and set

$$\bar{b}_i := g(t_i) + \epsilon_i, \text{ for } i = 1, \dots, \bar{m},$$

where  $\epsilon_i \sim N(0, .09)$  denotes independent random noise following normal distribution with 0 mean and 0.09 variance. For a strictly feasible initial point, we used  $\bar{\mathbf{x}}_0 := \mathbf{0}$  and  $t_0 := \|\bar{\mathbf{b}}\|_\infty + 1$ .



(a) Model is generated from an LP problem by setting  $\alpha = 0$  in (2.77).

(b) Model is generated from the CQP problem (2.77) with the diagonal components of  $\bar{\mathbf{H}}$  are proportional to the frequency of the corresponding basis function.

(c) Model is generated from a least squares (LS) problem,  $\min_{\bar{\mathbf{x}}} \|\bar{\mathbf{b}} - \bar{\mathbf{A}}\bar{\mathbf{x}}\|_2^2$ .

Figure 2.4: Measured signal fit by various methods. Tested with  $\bar{m} = 5000$  and  $\bar{n} = 99$ .

Without the regularization term, the model obtained from (2.77) tends to be too oscillatory as seen in Figure 2.4a. This tendency is caused by giving too much favor to basis functions with high frequency. To suppress high frequency components, we used a penalty weight proportional to the frequency of the basis functions. We let  $\bar{\mathbf{H}}$  be a diagonal matrix with  $\bar{h}_{jj} := 2(j - 1)\pi$  for  $j := 1, \dots, \bar{n} + 1$ , and  $\bar{h}_{jj} := 2(j - \bar{n} - 1)\pi$  for  $j = \bar{n} + 2, \dots, 2\bar{n} + 1$ . In our experiment, we set  $\alpha = 10^{-6}$  and  $\bar{n} = 99$ , resulting in  $n = 200$ .

We set  $\bar{m} := 5000$  ( $m = 10000$ ) for the comparison of models obtained from

	Regularized min-max (CQP)	Min-max (LP)	Least squares
Maximal error	0.328	0.326	0.491
Mean square error	0.0198	0.0217	0.00827

Table 2.1: When  $\bar{\mathbf{H}}$  is properly set, the model generated from the CQP (2.77) can be a compromise between that of LP (minimizing the maximal error) and least squares (minimizing sum of squared point-wise error). Mean square error was measured by  $\|\bar{\mathbf{b}} - \bar{\mathbf{A}}\bar{\mathbf{x}}\|_2^2/\bar{m}$ . Tested with  $\bar{m} := 5000$  and  $\bar{n} := 99$ .

linear programming (using  $\alpha := 0$ ), the regularized min-max (2.77), and least squares (using 2-norm instead of infinity norm) problems. Figure 2.4 shows the models. As illustrated in Table 2.1, the regularized min-max problem can be a compromise between the min-max and least squares problem.

We set  $\bar{m} := 20000$  ( $m = 40000$ ) for the comparison of our adaptive reduction and nonadaptive reduction which sets  $q := q_U$ . Timing and iteration counts on varying  $q_U$  are measured in Figure 2.5 with the horizontal axis in log scale. The nonadaptive algorithm with  $q_U := m$  corresponds to a standard PDAS IPM algorithm. When  $q_U$  is small (less than  $10^{-1}m$ ), adaptive shrinking of  $q$  does not take place until  $\mu$  becomes sufficiently small. Up to this point, both algorithms compute the same primal-dual iterates. Once  $\mu$  becomes sufficiently small, implying the iterate is close to the solution, shrinking the index set size does not affect the search direction as much as it does in early iterations, because the normal matrix is dominated by the

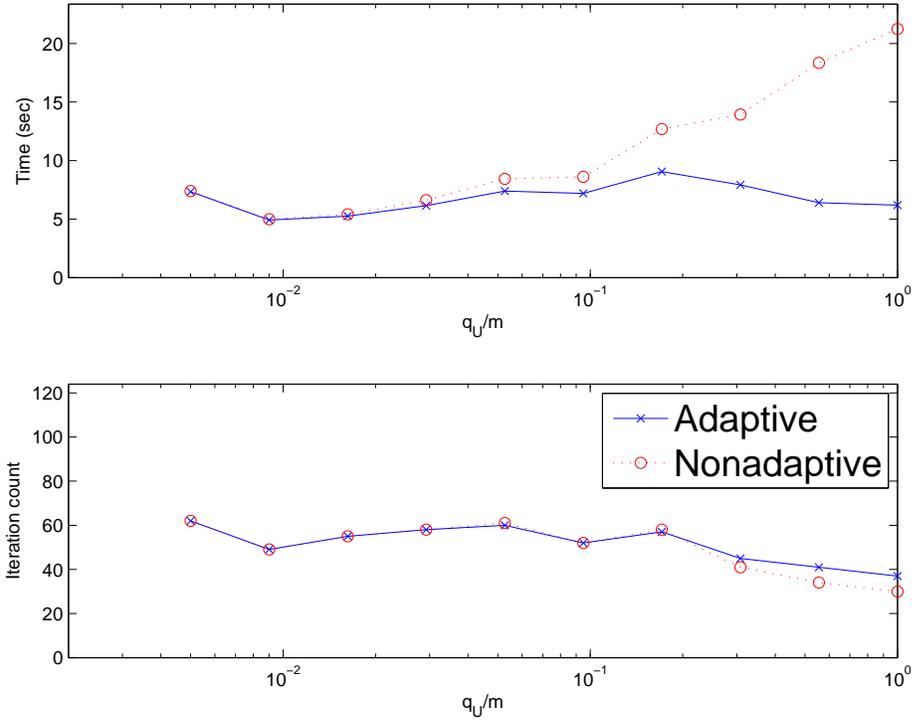


Figure 2.5: Adaptive reduction is compared with nonadaptive reduction on the data-fitting problem (2.77). The horizontal axis is in log scale. Tested with  $\bar{m} := 20000$  and  $\bar{n} := 99$ .

diverging  $\lambda_i/s_i^{-1}$ . On the other hand, when  $q_U$  is large, the adaptive algorithm may use fewer constraints than the nonadaptive even in early iterations. This affects the search direction at early iterations and may result in a different iteration count.

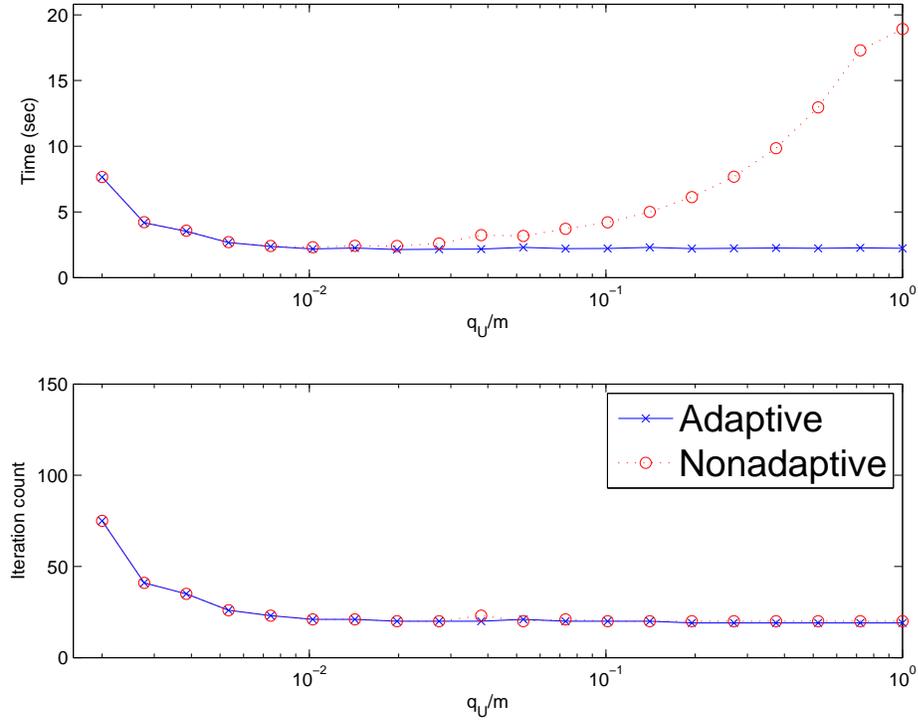


Figure 2.6: Adaptive reduction is compared with nonadaptive reduction on the fully random problem (2.77). The horizontal axis is in log scale.

### 2.5.4 Random Problems

We compared the adaptive reduction with the nonadaptive reduction on random problems of size  $m := 50000$  and  $n := 100$ . We generated  $\mathbf{A}$  and  $\mathbf{c}$  by taking random numbers drawn from a  $N(0, 1)$  distribution. Diagonal components of  $\mathbf{H}$  are taken from  $U(0, 1)$ , uniformly distributed random numbers in  $(0, 1)$ . We set  $\mathbf{s}^0$  by taking numbers from  $U(1, 2)$  and  $\mathbf{x}^0$  from  $U(0, 1)$ . We set  $\mathbf{b} := \mathbf{A}\mathbf{x}^0 - \mathbf{s}^0$ . This is a slight modification of the random problem in [TAW06], which uses different ranges of initial

$\mathbf{s}^0$ .

Timing and iteration counts are presented in Figure 2.6 with the horizontal axis in log scale. When  $q_U$  is very small (less than or equal to  $10^{-1}m$ ), adaptive shrinking of  $q$  takes place only for a few of the final iterations after the iterate is close enough to the solution. This prevents the adaptive reduction scheme from showing advantages for small  $q_U$ . It is noticeable that, for a wide range of  $q_U$ , the iteration count of both adaptive and nonadaptive reduction is near constant. In the same range of  $q_U$ , the timing of the adaptive reduction is near constant, while that of nonadaptive reduction decreases as  $q_U$  decreases.

## 2.6 Conclusions

We proposed an affine-scaling algorithm which significantly reduces computational effort in solving convex quadratic programming problems having many more constraints than variables.

We showed how the method can be applied to problems that explicitly include nonnegative slack variables such as the training of support vector machines with soft margins. We established convergence and a quadratic local convergence rate.

## Chapter 3

# Adaptive Constraint Reduction for Training Support Vector Machines

A baby learns from its own experience, or it may learn from what its parents teach. Then, the baby will cope with a new situation based on what it has learned. Babies are learning machines! We want to devise such a learning machine which learns from what we teach. The support vector machine (SVM) learns from a set of training patterns through a training process. Each pattern is associated with a predetermined class label which is assigned by humans. After the machine learns from the training patterns, the machine then can answer ‘yes’ or ‘no’ to new inputs.

In this chapter, we present an efficient algorithm for training of the machine. In section 3.1, we define the data representation and introduce the basics of training the support vector machine. We also formulate the training process as a CQP problem. In section 3.2, we review previous approaches to training the SVM on a large data set. In section 3.3, we present an interior-point method for training the machine.

A constraint-reduction mechanism is then applied to the method. Generalization to training nonlinear support vector machines is also provided. In section 3.4, promising numerical results are provided.

## 3.1 Introduction to Support Vector Machines

In this section, we introduce the support vector machine and its training.

### 3.1.1 Data Representation, Classifier, Feature Space, and Kernel

The training patterns are defined as

$$(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m) \in \mathcal{X} \times \{-1, +1\}, \quad (3.1)$$

where  $m$  is the number of training patterns,  $y_i$  is the known classification of the  $i^{\text{th}}$  pattern  $\mathbf{x}_i$ ,  $\mathcal{X}$  denotes the domain set, often called the *input space*, in which the patterns live, and  $\mathbf{y}$  will be used later for denoting  $[y_1, \dots, y_m]^T$ .

A *hyperplane classifier* or a *linear classifier* is a hyperplane,

$$\{\mathbf{x} : \langle \mathbf{w}, \mathbf{x} \rangle = \gamma\},$$

separating the negative patterns from the positive patterns. For an input or a pattern  $\mathbf{x} \in \mathcal{X}$ , the decision or prediction  $y$  of the classifier is

$$y := \text{sign}(\langle \mathbf{w}, \mathbf{x} \rangle - \gamma).$$

For ease of further discussion, we assume that  $\mathcal{X}$  is  $\mathbb{R}^l$ . This can be easily achieved by first preprocessing the training patterns.

To find a better classifier, we may want to map the training patterns into a vector space (probably of higher dimension) endowed with an inner product. This mapping is performed by a *feature map*  $\Phi(\cdot)$ :

$$\Phi : \mathcal{X} \rightarrow \mathcal{H} \tag{3.2}$$

$$\mathbf{x} \mapsto \Phi(\mathbf{x}), \tag{3.3}$$

where  $\mathcal{H}$  is a space endowed with some inner product  $\langle \cdot, \cdot \rangle_{\mathcal{H}}$  and the length or norm of a vector  $\mathbf{a} \in \mathcal{H}$  is defined as  $\|\mathbf{a}\|_{\mathcal{H}} := \sqrt{\langle \mathbf{a}, \mathbf{a} \rangle_{\mathcal{H}}}$ . The image space  $\mathcal{H}$  of the feature map is usually referred to as the *feature space*. A linear classifier determined in the feature space may induce a nonlinear classifier in the original input space; see Figure 3.1 for an example.

A symmetric and positive definite *kernel* is a function that measures the similarity of two vectors:

$$k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$$

$$(\mathbf{x}, \bar{\mathbf{x}}) \mapsto k(\mathbf{x}, \bar{\mathbf{x}}).$$

It is symmetric if

$$k(\mathbf{x}, \bar{\mathbf{x}}) = k(\bar{\mathbf{x}}, \mathbf{x}),$$

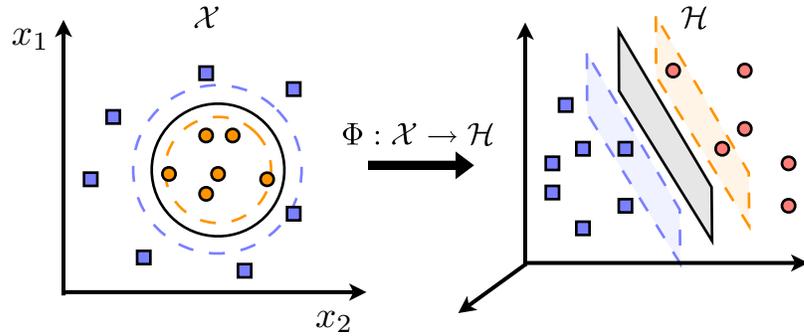


Figure 3.1: By mapping the patterns in the input space  $(x_1, x_2)$  to a higher dimensional feature space  $(x_1^2, x_2^2, \sqrt{2}x_1x_2)$ , the SVM can find an ellipsoidal classifier in the original input space by finding a linear classifier in the feature space.

for every  $\mathbf{x}, \bar{\mathbf{x}} \in \mathcal{X}$ . The kernel is positive definite if the *Gram* matrix  $\mathbf{K}$  whose  $i^{\text{th}}$  row and  $j^{\text{th}}$  column component  $k_{ij} := k(\mathbf{x}_i, \mathbf{x}_j)$ , induced from an arbitrary number of patterns  $\mathbf{x}_1, \dots, \mathbf{x}_m \in \mathcal{X}$  (not necessarily the training patterns), is positive semidefinite.<sup>1</sup>

Every symmetric and positive definite kernel is associated with the *reproducing kernel map*  $\Phi$  which is defined as

$$\begin{aligned} \Phi : \mathcal{X} &\rightarrow \mathcal{H} \\ \mathbf{x} &\mapsto k(\cdot, \mathbf{x}). \end{aligned} \tag{3.4}$$

Then the space spanned by the images of arbitrary patterns is the *reproducing kernel*

---

<sup>1</sup>Researchers in machine learning field often use “positive definite” to denote ‘positive semidefinite’ and “strictly positive definite” for ‘positive definite’.

*Hilbert space* (RKHS), a feature space corresponding to the reproducing kernel map:

$$\mathcal{H} := \left\{ f : f = \sum_{i=1}^{\infty} \alpha_i k(\cdot, \mathbf{x}_i), \forall i : \alpha_i \in \mathbb{R}, \text{ and } \mathbf{x}_i \in \mathcal{X} \right\}.$$

An inner product associated with the space is defined as

$$\langle f, g \rangle_{\mathcal{H}} := \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} \alpha_i \beta_j k(\mathbf{x}_i, \bar{\mathbf{x}}_j), \quad (3.5)$$

for  $f, g \in \mathcal{H}$ ,  $\mathbf{x}_i, \bar{\mathbf{x}}_j \in \mathcal{X}$ ,  $\alpha_i, \beta_j \in \mathbb{R}$ , and  $f := \sum_{i=1}^{\infty} \alpha_i k(\cdot, \mathbf{x}_i)$  and  $g := \sum_{j=1}^{\infty} \beta_j k(\cdot, \bar{\mathbf{x}}_j)$ .

It is not difficult to check that this definition leads to the *reproducing property*:

$\langle k(\cdot, \mathbf{x}_i), k(\cdot, \mathbf{x}_j) \rangle_{\mathcal{H}} = k(\mathbf{x}_i, \mathbf{x}_j)$ . This implies that the kernel evaluation between two

input patterns can replace the inner product between their images. This property

plays a central role in generalizing the linear SVM to nonlinear SVM. See [Bur98]

and [SS01] for more details.

Multiple feature spaces can be associated with a kernel. In addition to the RKHS

mapping (3.4), for instance, a feature map from  $\mathbb{R}^2$  to  $\mathbb{R}^3$  associated with the  $2^{nd}$

order *homogeneous polynomial kernel*  $k(\mathbf{x}, \bar{\mathbf{x}}) := (\mathbf{x}^T \bar{\mathbf{x}})^2$  can be defined as

$$\Phi : \mathbb{R}^2 \rightarrow \mathbb{R}^3, \quad (3.6)$$

$$[x_1, x_2]^T \mapsto [x_1^2, x_2^2, \sqrt{2}x_1x_2]^T,$$

so that the inner-product in the feature space is equivalent to the kernel evaluation

in the input space:

$$k(\mathbf{x}, \bar{\mathbf{x}}) = \langle \Phi(\mathbf{x}), \Phi(\bar{\mathbf{x}}) \rangle. \quad (3.7)$$

With the mappings such as (3.4) and (3.6), a linear classifier in the feature space is equivalent to some nonlinear classifier in the input space as depicted in Figure 3.1. Accordingly, we first limit our study to linear SVMs, and then extend it in section 3.3.4 to nonlinear SVMs. The numerical results of section 3.4 include nonlinear SVMs as well.

### 3.1.2 Separation Margin Maximization

In this section we consider finding a linear classifier where  $\mathbf{x}_i \in \mathbb{R}^n$  for  $i = 1, \dots, m$ . We assume that some map has already been applied so that the input space is the feature space:  $\mathcal{X} := \mathcal{H} := \mathbb{R}^n$  whose associated inner product is  $\langle \mathbf{x}_1, \mathbf{x}_2 \rangle := \mathbf{x}_1^T \mathbf{x}_2$ .

If the training patterns are strictly separable, then there will be infinitely many hyperplanes that can correctly classify the patterns; see Figure 3.2. To uniquely define a separating hyperplane, we seek one that maximizes the *separation margin*. The separation margin is defined to be the sum of the minimal distances from the hyperplane to the + patterns ( $y_i = 1$ ) and to the - patterns ( $y_i = -1$ ).

How can we train the machine so that it finds the hyperplane that maximizes the separation margin? Assuming that the patterns are separable and we already know the hyperplane with maximal margin, there would exist at least one point closest to the hyperplane in each class. We define the two hyperplanes parallel to the separating hyperplane that contain these two points. Then the distance between the parallel hyperplanes is the separation margin. The two parallel hyperplanes will be

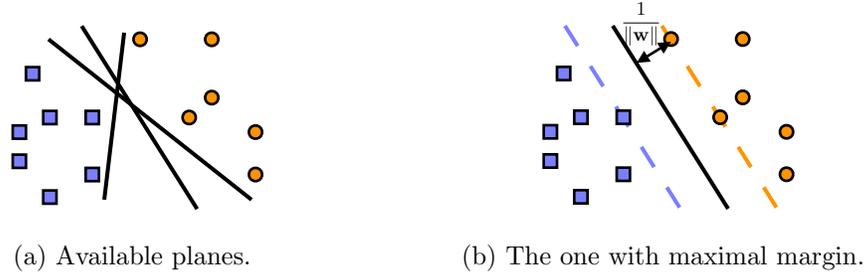


Figure 3.2: The learning machine is trained to find a hyperplane with maximal separation margin. The hyperplane can classify data according to the predetermined labels. Circles and squares denote positive and negative patterns, respectively.

referred to as the  $+$  and  $-$  class boundary plane, respectively.

Since the patterns are separable, there is no pattern in between the boundary hyperplanes. So, for all  $i \in \{1, \dots, m\}$ ,

$$\langle \mathbf{w}, \mathbf{x}_i \rangle - \gamma \geq y_i, \text{ if } y_i = +1, \tag{3.8}$$

$$\langle \mathbf{w}, \mathbf{x}_i \rangle - \gamma \leq y_i, \text{ if } y_i = -1, \tag{3.9}$$

or equivalently

$$y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle - \gamma) \geq 1, \tag{3.10}$$

where  $\mathbf{w} \in \mathbb{R}^n$  and  $\langle \mathbf{w}, \mathbf{x} \rangle := \mathbf{w}^T \mathbf{x}$ . So the boundaries of the half spaces defined by (3.8) and (3.9) are the  $+$  and  $-$  class boundary planes (or boundaries).

Since the distance between the boundary hyperplanes is  $\frac{2}{\|\mathbf{w}\|}$ , the problem, which is usually referred to as the *hard-margin SVM*, can now be modeled as an optimization

problem as follows:

$$\min_{\mathbf{w}, \gamma} \frac{1}{2} \|\mathbf{w}\|_2^2 \quad (3.11)$$

$$s.t. \quad \mathbf{Y}(\mathbf{X}\mathbf{w} - \mathbf{e}\gamma) \geq \mathbf{e}, \quad (3.12)$$

where  $\mathbf{X} := [\mathbf{x}_1, \dots, \mathbf{x}_m]^T \in \mathbb{R}^{m \times n}$ ,  $\mathbf{e} := [1, \dots, 1]^T$ , and  $\mathbf{Y} := \text{diag}(\mathbf{y})$  denotes the diagonal matrix version of  $\mathbf{y}$ . Typically  $m \gg n$ , and that is the case we consider here.

If the data are not separable, however, there is no solution to the optimization problem. To cope with this situation, we add a misclassification penalty in the objective function (3.11). By introducing nonnegative relaxation variables  $\boldsymbol{\xi}$  in order to tolerate misclassification, we get the relaxed constraints,

$$y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle - \gamma) \geq 1 - \xi_i. \quad (3.13)$$

After imposing an  $l_1$  penalty to the objective function (3.11), we get the primal formulation of a *soft margin SVM* proposed by Cortes and Vapnik [CV95]:

$$\min_{\mathbf{w}, \gamma, \boldsymbol{\xi}} \frac{1}{2} \|\mathbf{w}\|_2^2 + \boldsymbol{\tau}^T \boldsymbol{\xi} \quad (3.14)$$

$$s.t. \quad \mathbf{Y}(\mathbf{X}\mathbf{w} - \mathbf{e}\gamma) + \boldsymbol{\xi} \geq \mathbf{e}, \quad (3.15)$$

$$\boldsymbol{\xi} \geq \mathbf{0}, \quad (3.16)$$

where  $\boldsymbol{\tau}$  is an  $m$  dimensional vector of penalty parameters for the trade-off between the separation margin maximization and the empirical error minimization. This soft margin formulation is often preferred to the hard-margin formulation even when the

training patterns are strictly classifiable [Bur98]. Notice this formulation is a convex quadratic program with  $m$  nontrivial constraints (3.15),  $m$  trivial (bound) constraints (3.16),  $m$  trivial (relaxation) variables  $\boldsymbol{\xi}$ , and  $n$  nontrivial variables  $\mathbf{w}$ , where  $m \gg n$ . While  $\mathbf{w}$  is associated with the complicated coefficient matrix  $\mathbf{YX}$ ,  $\boldsymbol{\xi}$  is associated with the simple (identity) coefficient matrix in both sets of constraints, is not involved in the quadratic term, and, thus, can be trivially eliminated in the step equations which will be explained later. The variables  $\boldsymbol{\xi}$  make up for violations in the constraints (3.15).

### 3.1.3 Dual Formulation and Support Vector

Every optimization problem has an associated dual. The dual of (3.14)-(3.16) is

$$\max_{\boldsymbol{\alpha}} -\frac{1}{2}\boldsymbol{\alpha}^T\mathbf{YKY}\boldsymbol{\alpha} + \mathbf{e}^T\boldsymbol{\alpha} \quad (3.17)$$

$$\text{s.t. } \mathbf{y}^T\boldsymbol{\alpha} = 0, \quad (3.18)$$

$$\mathbf{0} \leq \boldsymbol{\alpha} \leq \boldsymbol{\tau}, \quad (3.19)$$

where the Gram matrix  $\mathbf{K} \in \mathbb{R}^{m \times m}$  is a symmetric and positive semidefinite matrix whose components are defined over the training patterns as

$$k_{ij} := \langle \mathbf{x}_i, \mathbf{x}_j \rangle, \quad (3.20)$$

and where  $\alpha_i$  is the dual variable associated with the  $i^{\text{th}}$  constraint in (3.15).

If  $\boldsymbol{\alpha}^*$  solves this dual problem, then the solution to the primal problem (3.14)-

(3.16) is

$$\mathbf{w}^* := \mathbf{X}^T \mathbf{Y} \boldsymbol{\alpha}^* = \sum_{i \in S} \alpha_i^* y_i \mathbf{x}_i, \text{ for } S := \{i : 0 < \alpha_i^*\}, \quad (3.21)$$

$$\gamma^* := \frac{1}{|S_{on}|} \sum_{i \in S_{on}} (\langle \mathbf{w}^*, \mathbf{x}_i \rangle - y_i), \text{ for } S_{on} := \{i : 0 < \alpha_i^* < \tau_i\}, \quad (3.22)$$

$$\xi_i^* := \max \{1 - y_i (\langle \mathbf{w}^*, \mathbf{x}_i \rangle - \gamma^*), 0\}, \text{ for } i = 1, \dots, m, \quad (3.23)$$

where  $\alpha_i^*$  and  $\tau_i$  are the  $i^{\text{th}}$  component of  $\boldsymbol{\alpha}^*$  and  $\boldsymbol{\tau}$ , and  $|S_{on}|$  denotes the size of  $S_{on}$ . The equation (3.22) for  $\gamma^*$  is obtained from (3.15). Since  $y_i (\langle \mathbf{w}_i^*, \mathbf{x}_i \rangle - \gamma^*) = 1$  for all  $i \in S_{on}$ , we know that  $\gamma^* = \langle \mathbf{w}^*, \mathbf{x}_i \rangle - y_i$ . We average for all  $i \in S_{on}$  as in (3.22) to have better accuracy [Bur98]. In view of (3.21), the Lagrange multiplier  $\alpha_i$  can be interpreted as the weight of the  $i^{\text{th}}$  pattern in defining the classifier.

*Support vectors* (SVs) are the patterns which contribute to defining the classifier. Therefore, they are associated with nonzero weight  $\alpha_i^*$ . Depending on whether the corresponding weight is equal to its upper bound  $\tau_i$ , the support vectors are divided into two groups [SBS99]. The *on-boundary* support vectors have weight strictly between the lower bound 0 and the upper bound  $\tau_i$ , and, geometrically, lie on their class boundary plane (i.e., both (3.15) and (3.16) are active). The *off-boundary* support vectors have the maximal allowable weight  $\alpha_i^* = \tau_i$  and lie on the wrong side of the class boundary plane (i.e., (3.15) is active but (3.16) is inactive).<sup>2</sup> We summarize this

---

<sup>2</sup>In the literature [SBS99], the terms in-bound and bound support vectors are used to denote on-boundary and off-boundary support vectors, respectively. The former terms are based on the bound constraints (3.19), whereas the latter terms are based on geometry.

Pattern type	$\alpha_i^*$	$s_i^*$	$\xi_i^*$
Off-boundary support vector	$\tau_i$	0	$(0, \infty)$
On-boundary support vector	$(0, \tau_i)$	0	0
Nonsupport vector	0	$(0, \infty)$	0

Table 3.1: Classification of support vectors and nonsupport vectors. Here  $s_i^*$  is the slack variable associated with the  $i^{th}$  constraints in (3.15) and defined as  $s_i^* := y_i(\langle \mathbf{w}^*, \mathbf{x}_i \rangle - \gamma^*) + \xi_i^* - 1$ .

classification in Table 3.1.

## 3.2 Related Work

Inspired by the fact that only a small portion of patterns contribute to forming the classifier, Osuna *et al.* [OFG97] proposed a decomposition algorithm for the dual SVM formulation. It first reduces the problem size by guessing the “active” or changeable (Lagrangian or dual) variables, a.k.a. the working set, and “nonactive” or fixed variables, where each variable is associated with a classification condition (or a primal constraint) for a pattern. The reduced problem is solved by an off-the-shelf quadratic programming (QP) solver (an active set method or an interior-point method). Then the fixed variables which violate the classification condition are

promoted to the working set by the “build-up” process. To keep the reduced problem size constant, the same number of variables in the working set are demoted to the non-working set by the “build-down” process. The rearrangement and the solution of the reduced QP are repeated until no violating variable is found in the non-working set. They showed the objective function value increases strictly at every iteration. Joachims [Joa99] further improved the algorithm of Osuna *et al.* by shrinking the problem size if possible and keeping a cache for kernel evaluations (or a submatrix of the Hessian). Platt [Pla99] proposed a sequential minimal optimization (SMO) algorithm which maintains a very small working set allowing only two variables to change. The subproblem of SMO can thus be solved analytically. See the four essays in [Hea98] for further discussion.

Recently primal-dual interior-point method (PDIPM) based algorithms were proposed. Ferris and Munson [FM02] considered training linear SVMs with  $l_1$  and  $l_2$  hinge loss. They efficiently applied the Sherman-Morrison-Woodbury (SMW) formula to solving the normal equations, the most expensive operation in an interior-point method (IPM). Gertz and Griffin [GG05] proposed a parallel direct solver and a preconditioned conjugate gradient solver tailored for the normal equations in training a SVM with  $l_1$  hinge loss.

Our focus is again on the normal equations. Like Osuna *et al.*, we reduce computational cost by filtering out unnecessary constraints or patterns in assembling the matrix for the normal equations. However, in contrast to the decomposition based

algorithms, we solve only *one* optimization problem, using constraint selection only to determine the search direction at each iteration of an IPM used for the training.

### 3.3 Adaptive Constraint Reduction

In this section we present a standard primal-dual interior-point method for training our SVM and then develop a way to improve the efficiency of the method by adaptively ignoring constraints.

#### 3.3.1 Primal-Dual Interior-Point Method

Since the soft margin formulation for the SVM (3.14)-(3.16) is a convex quadratic program, a solution to the formulation's KKT conditions is a global optimum and, thus, defines the separating hyperplane with maximal margin. Therefore, training the machine is equivalent to finding a solution to the KKT conditions (to the primal (3.14)-(3.16) and the dual (3.17)-(3.19)) [GG05]:

$$\mathbf{w} - \mathbf{X}^T \mathbf{Y} \boldsymbol{\alpha} = \mathbf{0}, \quad (3.24)$$

$$\mathbf{y}^T \boldsymbol{\alpha} = 0, \quad (3.25)$$

$$\boldsymbol{\tau} - \boldsymbol{\alpha} - \mathbf{u} = \mathbf{0}, \quad (3.26)$$

$$\mathbf{YXw} - \gamma \mathbf{y} + \boldsymbol{\xi} - \mathbf{e} - \mathbf{s} = \mathbf{0}, \quad (3.27)$$

$$\mathbf{S} \boldsymbol{\alpha} = \mathbf{0}, \quad (3.28)$$

$$\mathbf{U}\boldsymbol{\xi} = \mathbf{0}, \quad (3.29)$$

$$\mathbf{s}, \mathbf{u}, \boldsymbol{\alpha}, \boldsymbol{\xi} \geq \mathbf{0}, \quad (3.30)$$

where  $\mathbf{S} := \text{diag}(\mathbf{s})$ ,  $\mathbf{s}$  is a slack variable vector for the inequality constraints (3.15),  $\mathbf{U} = \text{diag}(\mathbf{u})$ , and  $\mathbf{u}$  is a slack for the upper bound constraints (3.19) or a vector of multipliers for the non-negativity constraints (3.16). Conditions (3.24)-(3.26) relate the gradient of the objective function to the constraints that are active at an optimal solution. The fourth condition is the primal feasibility condition. Conditions (3.28) and (3.29) enforce complementary slackness.

In order to find a solution, a Newton-like method can be applied to the KKT conditions with perturbations to the complementarity conditions (3.28) and (3.29). For the variant of the Mehrotra's Predictor Corrector (MPC) algorithm discussed in [Wri97] and [GG05], the search direction is obtained by solving the system of equations

$$\Delta \mathbf{w} - \mathbf{X}^T \mathbf{Y} \Delta \boldsymbol{\alpha} = -(\mathbf{w} - \mathbf{X}^T \mathbf{Y} \boldsymbol{\alpha}) \equiv -\mathbf{r}_w, \quad (3.31)$$

$$\mathbf{y}^T \Delta \boldsymbol{\alpha} = -\mathbf{y}^T \boldsymbol{\alpha} \equiv -r_\alpha, \quad (3.32)$$

$$-\Delta \boldsymbol{\alpha} - \Delta \mathbf{u} = -(\boldsymbol{\tau} - \boldsymbol{\alpha} - \mathbf{u}) \equiv -\mathbf{r}_u, \quad (3.33)$$

$$\mathbf{YX} \Delta \mathbf{w} - \mathbf{y} \Delta \gamma + \Delta \boldsymbol{\xi} - \Delta \mathbf{s} = -(\mathbf{YX} \mathbf{w} - \gamma \mathbf{y} + \boldsymbol{\xi} - \mathbf{e} - \mathbf{s}) \equiv -\mathbf{r}_s, \quad (3.34)$$

$$\mathbf{S} \Delta \boldsymbol{\alpha} + \text{diag}(\boldsymbol{\alpha}) \Delta \mathbf{s} = -\mathbf{r}_{sv}, \quad (3.35)$$

$$\text{diag}(\boldsymbol{\xi}) \Delta \mathbf{u} + \mathbf{U} \Delta \boldsymbol{\xi} = -\mathbf{r}_{\xi u}. \quad (3.36)$$

First, an affine-scaling (predictor) direction  $(\Delta \mathbf{w}^{\text{aff}}, \Delta \gamma^{\text{aff}}, \Delta \boldsymbol{\xi}^{\text{aff}}, \Delta \mathbf{s}^{\text{aff}}, \Delta \boldsymbol{\alpha}^{\text{aff}}, \Delta \mathbf{u}^{\text{aff}})$  is computed, by setting

$$\mathbf{r}_{\text{sv}} := \mathbf{S}\boldsymbol{\alpha}, \quad (3.37)$$

$$\mathbf{r}_{\boldsymbol{\xi}\mathbf{u}} := \mathbf{U}\boldsymbol{\xi}. \quad (3.38)$$

Then the combined affine-scaling and corrector step is obtained by setting

$$\mathbf{r}_{\text{sv}} := \mathbf{S}\boldsymbol{\alpha} - \sigma\mu\mathbf{e} + \Delta \mathbf{S}^{\text{aff}}\Delta \boldsymbol{\alpha}^{\text{aff}}, \quad (3.39)$$

$$\mathbf{r}_{\boldsymbol{\xi}\mathbf{u}} := \text{diag}(\boldsymbol{\xi})\mathbf{u} - \sigma\mu\mathbf{e} + \Delta \mathbf{U}^{\text{aff}}\Delta \boldsymbol{\xi}^{\text{aff}}, \quad (3.40)$$

where

$$\mu := \frac{\mathbf{s}^T \boldsymbol{\alpha} + \boldsymbol{\xi}^T \mathbf{u}}{2m} \quad (3.41)$$

is the complementarity measure;  $\sigma$  is the centering parameter;  $\Delta \mathbf{S}^{\text{aff}} := \text{diag}(\Delta \mathbf{s}^{\text{aff}})$ , and  $\Delta \mathbf{U}^{\text{aff}} := \text{diag}(\Delta \mathbf{u}^{\text{aff}})$ .

These equations can be reduced to the *normal equations*

$$\left( \mathbf{I} + \mathbf{X}^T \mathbf{Y} \boldsymbol{\Omega}^{-1} \mathbf{Y} \mathbf{X} - \frac{\bar{\mathbf{y}} \bar{\mathbf{y}}^T}{\bar{\mathbf{y}}^T \boldsymbol{\Omega}^{-1} \bar{\mathbf{y}}} \right) \Delta \mathbf{w} = -\bar{\mathbf{r}}_{\mathbf{w}} - \frac{1}{\bar{\mathbf{y}}^T \boldsymbol{\Omega}^{-1} \bar{\mathbf{y}}} \bar{r}_{\boldsymbol{\alpha}} \bar{\mathbf{y}}, \quad (3.42)$$

where

$$\boldsymbol{\Omega} := \text{diag}(\boldsymbol{\alpha})^{-1} \mathbf{S} + \mathbf{U}^{-1} \text{diag}(\boldsymbol{\xi}), \quad (3.43)$$

and  $\bar{\mathbf{y}} = \mathbf{X}^T \mathbf{Y} \boldsymbol{\Omega}^{-1} \mathbf{y} = \mathbf{X}^T \boldsymbol{\Omega}^{-1} \mathbf{e}$ . Then, we obtain  $\Delta \gamma$ ,  $\Delta \boldsymbol{\alpha}$ ,  $\Delta \boldsymbol{\xi}$ ,  $\Delta \mathbf{u}$ , and  $\Delta \mathbf{s}$  by solving

$$\Delta \gamma = \frac{1}{\bar{\mathbf{y}}^T \boldsymbol{\Omega}^{-1} \bar{\mathbf{y}}} \left( -\bar{r}_{\boldsymbol{\alpha}} + \bar{\mathbf{y}}^T \Delta \mathbf{w} \right), \quad (3.44)$$

$$\Delta \boldsymbol{\alpha} = -\boldsymbol{\Omega}^{-1}(\mathbf{r}_{\Omega} + \mathbf{YX}\Delta \mathbf{w} - \mathbf{y}\Delta \gamma), \quad (3.45)$$

$$\Delta \boldsymbol{\xi} = -\mathbf{U}^{-1}\text{diag}(\boldsymbol{\xi})(\bar{\mathbf{r}}_{\mathbf{u}} - \Delta \boldsymbol{\alpha}), \quad (3.46)$$

$$\Delta \mathbf{u} = -\text{diag}(\boldsymbol{\xi})^{-1}(\mathbf{r}_{\xi \mathbf{u}} + \mathbf{U}\Delta \boldsymbol{\xi}), \quad (3.47)$$

$$\Delta \mathbf{s} = -\text{diag}(\boldsymbol{\alpha})^{-1}(\mathbf{r}_{\mathbf{sv}} + \mathbf{S}\Delta \boldsymbol{\alpha}), \quad (3.48)$$

where  $\bar{\mathbf{r}}_{\mathbf{u}} := \mathbf{r}_{\mathbf{u}} + \text{diag}(\boldsymbol{\xi})^{-1}\mathbf{r}_{\xi \mathbf{u}}$ ,  $\mathbf{r}_{\Omega} := \mathbf{r}_{\mathbf{s}} + \text{diag}(\boldsymbol{\alpha})^{-1}\mathbf{r}_{\mathbf{sv}} - \mathbf{U}^{-1}\text{diag}(\boldsymbol{\xi})\bar{\mathbf{r}}_{\mathbf{u}}$ ,  $\bar{\mathbf{r}}_{\mathbf{w}} := \mathbf{r}_{\mathbf{w}} + \mathbf{X}^T\mathbf{Y}\boldsymbol{\Omega}^{-1}\mathbf{r}_{\Omega}$ , and  $\bar{r}_{\alpha} := r_{\alpha} - \mathbf{y}^T\boldsymbol{\Omega}^{-1}\mathbf{r}_{\Omega}$ . See [GG05] for detailed derivation.

In designing decomposition methods, it was common to use the dual formulation (3.17)-(3.19). This was done by discarding  $\mathbf{w}$ , thus removing (3.24) and replacing (3.27) with  $\mathbf{YKY}\boldsymbol{\alpha} - \gamma\mathbf{y} + \boldsymbol{\xi} - \mathbf{e} - \mathbf{s} = \mathbf{0}$ . Fine and Sheinberg [FS02] and Ferris and Munson [FM02] seem to have followed the tradition. They derived normal equations involving inversion of an  $m \times m$  matrix. They avoided the inversion through the SMW formula, reducing computational complexity from  $O(m^3)$  to  $O(mn^2)$ . Fine and Sheinberg used low rank Cholesky factorization with symmetric pivoting to approximate the Gram matrix. Ferris and Munson applied the IPM approach to various SVM formulations [FM02].

In contrast, Gertz and Griffin [GG05] derived the normal equations (3.42) by preserving  $\mathbf{w}$ . Their approach does not involve the SMW formula, and thus, does not suffer from numerical instability caused by it. Woodsend and Gondzio [WG07] derived normal equations for SVMs with  $l_1$  and  $l_2$  hinge loss, and SVM regression. They considered solving dual formulations in which  $\mathbf{w}$  is preserved. Their approach

with  $l_1$  hinge loss also results in the same KKT system and step equations as those of Gertz and Griffin. However, they applied a different sequence of block eliminations, resulting in different normal equations. Chapelle discussed relations between the primal and dual based approaches [Cha07]. He showed both approaches have the same computational complexity due to the SMW formula. Nevertheless, he argued that the primal based approach is superior because it directly attempts to maximize the separation margin.

Forming and solving the normal equations (3.42) is the most time consuming task in a step of the predictor-corrector algorithm, so we now focus on how to speed this process.

### 3.3.2 Constraint Reduction

In Chapter 2 we developed an algorithm for solving convex quadratic programming problems by replacing the matrix in the normal equations by an approximation to it. In this section we see how this idea can be applied to the SVM problem by using the matrix formed by setting small entries  $\omega_i^{-1}$  of  $\mathbf{\Omega}^{-1}$  to zero.

Since  $y_i = \pm 1$  and both  $\mathbf{Y}$  and  $\mathbf{\Omega}$  are diagonal, we know that  $\mathbf{Y}\mathbf{\Omega}^{-1}\mathbf{Y} = \mathbf{\Omega}$  and  $\mathbf{y}^T\mathbf{\Omega}^{-1}\mathbf{y} = \mathbf{e}^T\mathbf{\Omega}\mathbf{e}$ . Now consider the matrix of the normal equations (3.42),

$$\begin{aligned} \mathbf{M} &:= \mathbf{I} + \mathbf{X}^T\mathbf{Y}\mathbf{\Omega}^{-1}\mathbf{Y}\mathbf{X} - \frac{\bar{\mathbf{y}}\bar{\mathbf{y}}^T}{\mathbf{y}^T\mathbf{\Omega}^{-1}\mathbf{y}} \\ &= \mathbf{I} + \sum_{i=1}^m \omega_i^{-1}\mathbf{x}_i\mathbf{x}_i^T - \frac{(\sum_{i=1}^m \omega_i^{-1}\mathbf{x}_i)(\sum_{i=1}^m \omega_i^{-1}\mathbf{x}_i)^T}{\sum_{i=1}^m \omega_i^{-1}}, \end{aligned} \tag{3.49}$$

and the matrix

$$\mathbf{M}_{(Q)} := \mathbf{I} + \mathbf{X}_Q^T \mathbf{Y}_{Q^2} \mathbf{\Omega}_{Q^2}^{-1} \mathbf{Y}_{Q^2} \mathbf{X}_Q - \frac{\bar{\mathbf{y}}_{(Q)} \bar{\mathbf{y}}_{(Q)}^T}{\mathbf{y}_Q^T \mathbf{\Omega}_{Q^2}^{-1} \mathbf{y}_Q} \quad (3.50)$$

$$= \mathbf{I} + \sum_{i \in Q} \omega_i^{-1} \mathbf{x}_i \mathbf{x}_i^T - \frac{(\sum_{i \in Q} \omega_i^{-1} \mathbf{x}_i)(\sum_{i \in Q} \omega_i^{-1} \mathbf{x}_i)^T}{\sum_{i \in Q} \omega_i^{-1}}, \quad (3.51)$$

where  $\bar{\mathbf{y}}_{(Q)} := \mathbf{X}_Q^T \mathbf{Y}_{Q^2} \mathbf{\Omega}_{Q^2}^{-1} \mathbf{y}_Q$  and  $Q \subseteq M$ . We use the parenthesized subscript  $(Q)$  to denote that a vector or a matrix is a function of  $Q$ . We use  $\mathbf{Y}_{Q^2}$  to denote a submatrix of any  $m \times m$  matrix  $\mathbf{Y}$  with both rows and columns indexed by  $Q$  and, similarly,  $\mathbf{X}_Q$  for that of any matrix  $\mathbf{X}$  with rows indexed by  $Q$ . The same notation also applies to any  $m$  dimensional column vector.

If  $Q = \{1, \dots, m\}$ , then  $\mathbf{M}_{(Q)} = \mathbf{M}$ . If  $Q \subset \{1, \dots, m\}$ , then  $\mathbf{M}_{(Q)}$  is an approximation, accurate if the neglected terms are small relative to those included. Hence, the approximated matrix reduces the computational cost for the matrix assembly, which is the most expensive task, from  $O(mn^2)$  to  $O(|Q|n^2)$ , where  $|Q|$  denotes the size of  $Q$  and is expected to be significantly less than  $m$ .

How do we obtain a good approximation? We see that patterns associated with larger  $\omega_i^{-1}$  make a larger contribution to the matrix. Let's see which patterns these are. The quantity

$$\omega_i^{-1} = \frac{\alpha_i u_i}{s_i \alpha_i + \xi_i u_i}$$

becomes very large if both  $s_i$  and  $\xi_i$  are close to zero because  $u_i$  and  $\alpha_i$  sum to  $\tau_i$  and do not converge to zero due to the complementarity conditions. From (3.14)-(3.16)

we see that, in the optimal solution, only one of  $s_i$  and  $\xi_i$  should be nonzero. If either one is nonzero, in view of the complementary slackness (3.28)-(3.29), either  $\alpha_i$  or  $u_i$  is zero, and thus  $\omega_i^{-1}$  is zero in the optimal solution. Therefore, as seen in Table 3.1, the important terms in the summation in (3.49) are associated with the on-boundary support vectors. Identifying on-boundary support vectors is not possible until we find the maximal margin classifier and its class boundaries. At each iteration, however, we have intermediate values of  $\omega_i$ 's. So we find prospective on-boundary support vectors by choosing the patterns with small  $\omega_i$ .

As the intermediate classifier approaches the maximal margin classifier, it becomes clearer which patterns are more likely to be on-boundary support vectors. This enables us to adaptively reduce the index set size used in the summation of (3.49). To measure how close the intermediate classifier is to the optimal one, we can use the complementarity measure which converges to zero. We set the size  $q$  of our index set to be a value between two numbers  $q_L$  and  $q_U$ :

$$q := \max(q_L, \min(\lceil \rho m \rceil, q_U)), \quad (3.52)$$

where we define

$$\rho := \mu^{\frac{1}{\beta}}$$

to synchronize decrease of the index set size with that of the optimality measure. Here  $\beta > 0$  is a parameter for controlling rate of decrease as  $\mu$  converges to zero very fast. At the first iteration we randomly choose  $q_U$  indices because we have no

information about the classifier. Fast clustering algorithms may improve the initial selection [BC04].

As described in [GG05], in typical primal-dual interior-point methods, a growing  $\omega_i^{-1}$  diverges to infinity at an  $O(\mu^{-1})$  rate and a vanishing  $\omega_i^{-1}$  converges to zero at an  $O(\mu)$  rate. Therefore we can separate the two different types of  $\omega_i^{-1}$  using  $\sqrt{\mu}$ .

Having determined  $q$ , we now choose patterns. Based on our examination of  $\omega_i^{-1}$ , there are several reasonable choices. We define  $\mathcal{Q}(\mathbf{z}, q)$ , the set of all subsets of  $M := \{1, \dots, m\}$  that contain the indexes of the  $q$  smallest components of  $\mathbf{z}$ :

$$\mathcal{Q}(\mathbf{z}, q) := \{Q \mid Q \subseteq M, |Q| = q \text{ and } z_i \leq z_j \forall i \in Q, j \notin Q\}.$$

Then we have the following choices of patterns:

- $\mathcal{Q}(\mathbf{YX}\mathbf{w} - \gamma\mathbf{y} + \boldsymbol{\xi} - \mathbf{e}, q)$ : this is a set of sets having indices for  $q$  patterns  $\mathbf{x}_i$  whose “one-sided” distance to the boundary plane is smallest. When primal feasibility holds, this measures the slacks of the primal constraints (3.15). This choice is most intuitive because support vectors contribute to defining the classifier, which is the underlying idea of most decomposition based algorithms. For this rule, we define

$$q_L := |\{i : \alpha_i/s_i \geq \theta\sqrt{\mu} \text{ or } s_i \leq \sqrt{\mu}\}|,$$

where  $\theta$  is a prescribed parameter.

- $\mathcal{Q}(\Omega\mathbf{e}, q)$ : this is a set of sets with indices of the  $q$  smallest  $\omega_i$ . This choice reflects that the expression (3.49) for the matrix  $\mathbf{M}$  is dominated by the terms with large  $\omega^{-1}$ . Inspired by [GG05], we define the lower bound on the index set size  $q_L$  by counting the number of large  $\omega_i^{-1}$ :

$$q_L := |\{i : \omega_i^{-1} \geq \theta\sqrt{\mu}\}|, \quad (3.53)$$

where  $\theta$  is a prescribed parameter. The parameter  $q_L$  will eventually converge to the number of diverging  $\omega_i^{-1}$ , or equivalently, the number of on-boundary support vectors.

Any of these choices, however, may have an imbalance between the number of patterns chosen from the  $+$  and  $-$  classes. Not considering the class labels, we might unknowingly choose no pattern from one class, where on-boundary support vectors are typically found in both classes. To avoid this unfavorable situation, we might want to use a *balanced* choice of patterns, specifying the number  $q^+$  and  $q^-$  chosen from each class as

$$q^+ := \max\left(q_L^+, \min\left(\left\lceil \frac{\min([\rho m], q_U)}{2} \right\rceil, m^+\right)\right), \quad (3.54)$$

$$q^- := \max\left(q_L^-, \min\left(\left\lceil \frac{\min([\rho m], q_U)}{2} \right\rceil, m^-\right)\right), \quad (3.55)$$

where  $m^+$  and  $m^-$  are the number of  $+$  and  $-$  patterns, respectively. Then we adjust either  $q^+$  or  $q^-$  so that  $q^+ + q^- = q$ . The lower bounds,  $q_L^+$  and  $q_L^-$ , are determined

similarly to (3.53) for each class as

$$\begin{aligned} q_L^+ &:= |\{i : \omega_i^{-1} \geq \theta\sqrt{\mu} \text{ and } y_i = 1\}|, \\ q_L^- &:= |\{i : \omega_i^{-1} \geq \theta\sqrt{\mu} \text{ and } y_i = -1\}|. \end{aligned} \tag{3.56}$$

Now we define the set of the  $q^+/q^-$  smallest sets for each class:

$$\mathcal{Q}^+(\mathbf{z}, q^+) := \{Q \mid Q \subseteq M, |Q| = q^+ \text{ and } z_i \leq z_j \forall i \in Q, j \notin Q \text{ and } d_i = d_j = +1\},$$

$$\mathcal{Q}^-(\mathbf{z}, q^-) := \{Q \mid Q \subseteq M, |Q| = q^- \text{ and } z_i \leq z_j \forall i \in Q, j \notin Q \text{ and } d_i = d_j = -1\},$$

The set  $Q$  can be any of the union of any two elements, one in  $\mathcal{Q}^+(\mathbf{x}, q^+)$  and the other in  $\mathcal{Q}^-(\mathbf{x}, q^-)$ :

$$Q \in \mathcal{Q}(\mathbf{z}, q^+, q^-) := \{Q \mid Q = Q^+ \cup Q^-, Q^+ \in \mathcal{Q}^+(\mathbf{z}, q^+) \text{ and } Q^- \in \mathcal{Q}^-(\mathbf{z}, q^-)\}.$$

Having determined  $Q$ , we construct the reduced normal equation for one step of our interior-point method by assembling the matrix for the normal equation using a subset of the patterns:

$$\mathbf{M}_{(Q)}\Delta\mathbf{w} = -\bar{\mathbf{r}}_{\mathbf{w}} - \frac{1}{\mathbf{y}^T\boldsymbol{\Omega}^{-1}\mathbf{y}}\bar{\mathbf{r}}_{\boldsymbol{\alpha}}\bar{\mathbf{y}}. \tag{3.57}$$

Then we solve (3.44)-(3.48) for  $\Delta\boldsymbol{\gamma}$ ,  $\Delta\boldsymbol{\alpha}$ ,  $\Delta\boldsymbol{\xi}$ ,  $\Delta\mathbf{u}$ , and  $\Delta\mathbf{s}$ . Before we proceed, we have to ensure that the reduced matrix  $\mathbf{M}_{(Q)}$  is positive definite.

**Proposition 3.1.** *Assume that  $\omega_i > 0$  for all  $i \in Q$ . The matrix  $\mathbf{M}_{(Q)}$  is symmetric and positive definite.*

*Proof.* See Proposition 1 in [GG05]. □

The following proposition explains the asymptotic convergence of the reduced matrix to the unreduced one.

**Proposition 3.2.** *For  $q$  defined in (3.52) and for all  $Q \in \mathcal{Q}(\Omega\mathbf{e}, q)$ , there exists a positive constant  $C_{\mathbf{M}}$  satisfying  $\|\mathbf{M} - \mathbf{M}_{(Q)}\|_2 \leq C_{\mathbf{M}}\sqrt{\mu}$ .*

*Proof.* See Proposition 5 in [GG05]. □

We state in Algorithm 2 a variant of Mehrotra-type predictor-corrector algorithm with a constraint-reduction mechanism. This algorithm makes use of the reduced normal equations (3.57).

---

**Algorithm 2** Constraint reduced SVM (CRSVM)

---

*Parameters:*  $\beta > 0$ ,  $\boldsymbol{\tau} > \mathbf{0}$ ,  $\theta > 0$ , integer  $q_U$  satisfying  $q_U \leq m$ ,  $Bal \in \{false, true\}$ ,

$CC \in \{\text{'one-sided dist'}, \text{'omega'}\}$ .

Given a starting point  $(\mathbf{w}, \gamma, \boldsymbol{\xi}, \mathbf{s}, \boldsymbol{\alpha}, \mathbf{u})$  with  $(\boldsymbol{\xi}, \mathbf{s}, \boldsymbol{\alpha}, \mathbf{u}) > 0$ .

**for**  $k = 0, 1, 2, \dots$  **do**

Terminate if convergence is detected:

$$\frac{\max\{\|\mathbf{r}_{\mathbf{w}}\|_{\infty}, |r_{\boldsymbol{\alpha}}|, \|\mathbf{r}_{\mathbf{u}}\|_{\infty}, \|\mathbf{r}_{\mathbf{s}}\|_{\infty}\}}{\max\{\|A\|_{\infty}, \|\boldsymbol{\tau}\|_{\infty}, 1\}} \leq tol_r, \text{ and}$$

$$\mu \leq tol_{\mu},$$

or iteration count is reached.

**if**  $Bal$  is *true* **then**

Determine  $q^+$  and  $q^-$  to (3.54) and (3.55).

Pick  $Q$  from  $\mathcal{Q}(\mathbf{YX}\mathbf{w} - \gamma\mathbf{y} - \mathbf{e} + \boldsymbol{\xi}, q^+, q^-)$  if  $CC$  is ‘one-sided dist’ or from  $\mathcal{Q}(\boldsymbol{\Omega}\mathbf{e}, q^+, q^-)$  if  $CC$  is ‘omega’.

**else**

Determine  $q$  according to (3.52).

Pick  $Q$  from  $\mathcal{Q}(\mathbf{YX}\mathbf{w} - \gamma\mathbf{y} - \mathbf{e} + \boldsymbol{\xi}, q)$  if  $CC$  is ‘one-sided dist’ or from  $\mathcal{Q}(\boldsymbol{\Omega}\mathbf{e}, q)$  if  $CC$  is ‘omega’.

**end if**

Solve (3.57) and (3.44)-(3.48) for  $(\Delta\mathbf{w}^{\text{aff}}, \Delta\gamma^{\text{aff}}, \Delta\boldsymbol{\xi}^{\text{aff}}, \Delta\mathbf{s}^{\text{aff}}, \Delta\boldsymbol{\alpha}^{\text{aff}}, \Delta\mathbf{u}^{\text{aff}})$  using affine-scaling residuals from (3.37)-(3.38).

Determine predictor step length:

$$\alpha_{\text{aff}} := \max_{\alpha \in [0,1]} \{\alpha : (\boldsymbol{\xi}, \mathbf{s}, \boldsymbol{\alpha}, \mathbf{u}) + \alpha(\Delta\boldsymbol{\xi}^{\text{aff}}, \Delta\mathbf{s}^{\text{aff}}, \Delta\boldsymbol{\alpha}^{\text{aff}}, \Delta\mathbf{u}^{\text{aff}}) \geq 0\}. \quad (3.58)$$

$$\text{Set } \mu_{\text{aff}} := \frac{(\mathbf{s} + \alpha_{\text{aff}} \Delta\mathbf{s}^{\text{aff}})^T (\boldsymbol{\alpha} + \alpha_{\text{aff}} \Delta\boldsymbol{\alpha}^{\text{aff}}) + (\boldsymbol{\xi} + \alpha_{\text{aff}} \Delta\boldsymbol{\xi}^{\text{aff}})^T (\mathbf{u} + \alpha_{\text{aff}} \Delta\mathbf{u}^{\text{aff}})}{2m}.$$

$$\text{Set } \sigma := (\mu_{\text{aff}}/\mu)^3.$$

Solve (3.57) and (3.44)-(3.48) for  $(\Delta\mathbf{w}, \Delta\gamma, \Delta\boldsymbol{\xi}, \Delta\mathbf{s}, \Delta\boldsymbol{\alpha}, \Delta\mathbf{u})$  using combined step residuals from (3.39)-(3.40).

Determine the step length for the combined step:

$$\alpha := 0.99 \max_{\alpha \in [0,1]} \{\alpha : (\boldsymbol{\xi}, \mathbf{s}, \boldsymbol{\alpha}, \mathbf{u}) + \alpha(\Delta\boldsymbol{\xi}, \Delta\mathbf{s}, \Delta\boldsymbol{\alpha}, \Delta\mathbf{u}) \geq 0\}. \quad (3.59)$$

Set

$$(\mathbf{w}, \gamma, \boldsymbol{\xi}, \mathbf{s}, \boldsymbol{\alpha}, \mathbf{u}) := (\mathbf{w}, \gamma, \boldsymbol{\xi}, \mathbf{s}, \boldsymbol{\alpha}, \mathbf{u}) + \alpha(\Delta\mathbf{w}, \Delta\gamma, \Delta\boldsymbol{\xi}, \Delta\mathbf{s}, \Delta\boldsymbol{\alpha}, \Delta\mathbf{u}) \quad (3.60)$$

**end for**

---

When the matrix  $\mathbf{X}$  is sparse, the sum of the first two terms of  $\mathbf{M}_{(Q)}$  could result in a sparse matrix. However, adding the third term makes the matrix dense. So we obtain a sparse Cholesky factor for the sum of the first two terms with full pivoting. Then, to solve the normal equations (3.57), we apply the SMW formula to reflect the subtraction of the rank 1 matrix in  $\mathbf{M}_{(Q)}$ . For the dense matrix, we fully assemble  $\mathbf{M}_{(Q)}$  and obtain its dense Cholesky factor.

Winternitz *et al.* presented convergence results for a constraint-reduction algorithm of a MPC variant for LP [WNT07]. Extending the MPC variant and its convergence to CQP and SVM training is a topic for future research.

### 3.3.3 Differences from Algorithm 1

Algorithm 2 shows two significant differences from Algorithm 1 (if augmented to the extended problem). First, Algorithm 2 is based on an MPC variant, while Algorithm 1 is based on a primal-dual affine-scaling algorithm. Second, Algorithm 2 concentrates only on the approximation of the normal matrix, whereas Algorithm 1 generates a primal-dual step that satisfies the reduced Newton system (2.74)-(2.76). Due to this, the right hand side of the reduced normal equations (3.57) for the SVM training is

different from that (2.72) for the extended problem. While the right hand side of (3.57) is not reduced at all, that of (2.72) is. In addition, noting that  $\boldsymbol{\xi}$  in (3.14)-(3.16) corresponds to  $\mathbf{y}$  in (2.52)-(2.54), the equations (3.46) for obtaining  $\Delta\boldsymbol{\xi}$  is not reduced, but (2.73) for  $\Delta\mathbf{y}$  is reduced. Developing an algorithm without these differences and demonstration of its convergence is a future research topic.

### 3.3.4 Kernelization

As mentioned in section 3.1.1, due to the reproducing property of a symmetric and positive definite kernel,  $k(\mathbf{x}_i, \mathbf{x}_j)$  for two input patterns  $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^l$  can replace the inner product between their images  $\Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \in \mathcal{H}$ . So, in defining the Gram matrix (3.20), replacing  $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$  with  $k(\mathbf{x}_i, \mathbf{x}_j)$  is equivalent to mapping a point  $\mathbf{x}$  to an entity  $\Phi(\mathbf{x}) := k(\cdot, \mathbf{x})$  living in the corresponding RKHS. This is called the “*kernel trick*”. Using this trick and the dual formulation (3.17)-(3.19), we can train the SVM to find a classifier in a feature space that might be infinite dimensional (see [Bur98]).

If a data set has an enormous number of training patterns, building  $\mathbf{K}$  may not be feasible. In addition,  $\mathbf{K}$  is dense for many frequently employed kernels regardless of whether the input matrix is sparse. For instance, forming  $\mathbf{K}$  for 100,000 training patterns using the Gaussian kernel needs 80GB of storage space. Even worse is that a single iteration of an IPM could require factoring a  $m \times m$  matrix, costing  $O(m^3)$  arithmetic operations.

This issue has been tackled in several papers [SS01, FS02, Cha07] through a low

rank approximation to the Gram matrix,  $\mathbf{K} \approx \mathbf{V}\mathbf{G}^2\mathbf{V}^T$ , where  $\mathbf{G}$  is an  $n \times n$  symmetric and positive definite matrix and  $\mathbf{V}$  is an  $m \times n$  matrix for  $m \gg n$ . These include the truncated eigenvalue decomposition [GVL96, CSS06], low rank Cholesky factorization with symmetric pivoting [FS02, BJ05], Nyström method [WS01, DM05], and kernel PCA map [SS00, HZKM03]. The fast multipole method [YDD05, RYDG05] can be employed to compute the truncated eigenvalue decomposition.

Let's see how these results can be applied to the constraint reduction for training nonlinear SVMs. Consider an approximate dual with  $\mathbf{K}$  in (3.17)-(3.19) replaced by  $\mathbf{V}\mathbf{G}^2\mathbf{V}^T$ . What primal problem would induce the dual? We notice that using  $\mathbf{V}\mathbf{G}$  instead of  $\mathbf{X}$  in the primal (3.14)-(3.16) leads to the approximate dual. In other words, before we initiate Algorithm 2 we compute an approximation to  $\mathbf{K}$  from the original input  $\mathbf{X}$ . Then we pass  $\mathbf{V}\mathbf{G}$  as  $\mathbf{X}$  to Algorithm 2.

If  $\mathbf{G}$  is only readily available in its squared inverse form  $\mathbf{G}^{-2}$ , we could think of letting  $\bar{\mathbf{w}} := \mathbf{G}\mathbf{w}$ , which leads to the following problem:

$$\min_{\bar{\mathbf{w}}, \gamma, \xi} \frac{1}{2} \bar{\mathbf{w}}^T \mathbf{G}^{-2} \bar{\mathbf{w}} + \boldsymbol{\tau}^T \boldsymbol{\xi} \quad (3.61)$$

$$s.t. \quad \mathbf{Y}(\mathbf{X}\bar{\mathbf{w}} - \mathbf{e}\gamma) + \boldsymbol{\xi} \geq \mathbf{e}, \quad (3.62)$$

$$\boldsymbol{\xi} \geq \mathbf{0}, \quad (3.63)$$

where  $\mathbf{X} := \mathbf{V}$ . This formulation would be useful if obtaining  $\mathbf{G}$  is not desirable. For instance,  $\mathbf{G}^{-2} = \mathbf{K}_{B^2}$  for some index set  $B$  when the empirical kernel map [SMB<sup>+</sup>99, SS00] is employed. Applying the constraint reduction to this formulation

is straightforward. A simple change of  $\mathbf{M}_{(Q)}$  as

$$\mathbf{M}_{(Q)} := \mathbf{G}^{-2} + \mathbf{X}_Q^T \mathbf{Y}_{Q^2} \mathbf{\Omega}_{Q^2}^{-1} \mathbf{Y}_{Q^2} \mathbf{X}_Q - \frac{\bar{\mathbf{y}}_{(Q)} \bar{\mathbf{y}}_{(Q)}^T}{\mathbf{y}_Q^T \mathbf{\Omega}_{Q^2}^{-1} \mathbf{y}_Q}, \quad (3.64)$$

and the substitution of  $\bar{\mathbf{w}}$  for  $\mathbf{w}$ ,  $\Delta \bar{\mathbf{w}}$  for  $\Delta \mathbf{w}$ , and  $\mathbf{r}_{\bar{\mathbf{w}}} := \mathbf{G}^{-2} \bar{\mathbf{w}} - \mathbf{X}^T \mathbf{Y} \boldsymbol{\alpha}$  for  $\mathbf{r}_{\mathbf{w}}$  are all the required modifications.

### 3.4 Numerical Results

We implemented Algorithm 2 in MATLAB. We tested Algorithm 2 using MATLAB version R2007a on a machine running Windows XP SP2 with an Intel Pentium IV 2.8GHz processor with 16 KB L1 cache, 1 MB L2 cache, 2×1 GB DDR2-400MHz configured as dual channel, with Hyper Threading enabled.

Both  $tol_r$  and  $tol_\mu$  were set to  $10^{-8}$ . The iteration limit was set to 200. We set the parameters as  $\beta := 4$  to control the rate of decrease of  $q$ ,  $\theta := 10^2$  to determine  $q_L$ , and  $\tau_i := 1$  for  $i = 1, \dots, m$  to penalize misclassification. We vary  $q_U$  to see how Algorithm 2 reacts to it. The initial starting point was set as in [GG05]:

$$\mathbf{w} := \mathbf{0}, \gamma := 0, \boldsymbol{\xi} := \mathbf{s} := \boldsymbol{\alpha} := \mathbf{u} := 2\mathbf{e}.$$

We compared Algorithm 2 (CRSVM) to LIBSVM [CL01], and SVM<sup>light</sup> [Joa99]. We set their termination tolerance parameters as their default value  $10^{-3}$ .

### 3.4.1 Linear SVM Examples

We tested our implementation on problems mushroom, isolet, waveform, and letter, all taken from [GG05]. Except for the isolet problem, all inputs were mapped to higher dimensional feature space via the mapping associated with the second order polynomial kernel  $k(\mathbf{x}, \bar{\mathbf{x}}) := (\mathbf{x}^T \bar{\mathbf{x}} + 1)^2$  as in [GG05]. The mapping  $\Phi$  is defined as

$$\begin{aligned} \Phi : \mathbb{R}^l &\rightarrow \mathbb{R}^n \\ [x_1, \dots, x_l]^T &\mapsto \\ \sqrt{2} \left[ \frac{1}{\sqrt{2}}x_1^2, \dots, \frac{1}{\sqrt{2}}x_l^2, x_1x_2, \dots, x_1x_l, x_2x_3, \dots, x_2x_l, \dots, x_{l-1}x_l, x_1, \dots, x_l, \frac{1}{\sqrt{2}} \right]^T, \end{aligned} \quad (3.65)$$

where  $n := \binom{l+2}{2} = \binom{l}{2} + 2l + 1$ . The  $i^{\text{th}}$  row of  $\mathbf{X}$  is set to  $\Phi(\mathbf{x}_i)^T$ , where  $\mathbf{x}_i^T$  is the  $i^{\text{th}}$  training input. We also normalized the resulting matrix using

$$x_{ij} := \frac{x_{ij}}{\max_{kl} |x_{kl}|}, \quad (3.66)$$

as directed in [GG05]. Properties of the problems are summarized in Table 3.2.

In our experiment, we compared our algorithms to the standard MPC algorithm, which uses all the constraints for every iteration. We experimented with several variants of our algorithms:

- Nonadaptive balanced constraint reduction, which uses fixed  $q^+$  and  $q^-$  throughout the iteration.
- Adaptive non-balanced constraint reduction, which determines  $q$  as in (3.52).

Name	ISD	FSD	Patterns (+/-)	SVs (+/-)	On-SVs (+/-)
mushroom	22	276	8124 (4208/ 3916)	2285 (1146/1139)	52 (31 / 21)
isolet	617	617	7797 (300 / 7497)	186 (74 / 112)	186 (74 / 112)
waveform	40	861	5000 (1692/ 3308)	1271 (633 / 638)	228 (110/ 118)
letter	letter	153	20000 (789 /19211)	543 (266 / 277)	40 (10 / 30)

Table 3.2: Properties of the problems. ISD: Input space dimension. FSD: Feature space dimension using the map (3.65). SVs: support vectors. On-SVs: On-boundary support vectors.

- Adaptive balanced constraint reduction, which determines  $q^+$  and  $q^-$  as in (3.54) and (3.55).

In all three reduction algorithms we selected patterns based on two choices, one-sided distance  $(\mathbf{YX}\mathbf{w} - \gamma\mathbf{y} + \boldsymbol{\xi} - \mathbf{e})$  and  $\boldsymbol{\Omega}\mathbf{e}$ , as explained in Section 3.3.2, resulting in 6 possible variations.

In comparing the reduction algorithm with the standard MPC, we used the two pattern choices with the adaptive balanced reduction. In comparing adaptive reduction with nonadaptive reduction, we used the balanced scheme with the constraint choice based on  $\boldsymbol{\Omega}\mathbf{e}$ . In comparing the two constraint choices, we used the adaptive balanced reduction.

In Figure 3.3a and 3.3b, the time and iteration count of the algorithm with the

two constraint choices and the balanced selection scheme are compared to those of the standard MPC. We set  $q_U := m$ ,  $Bal := true$ , and  $CC :=$  ‘one-sided dist’ or  $CC :=$  ‘omega’. Bar graphs are grouped for each problem. Figure 3.3b shows that the number of iterations for a problem is not much different for the algorithm variants. As a result, the reduction algorithms with any constraint choice are faster than the standard MPC algorithm, as seen in Figure 3.3a. In solving hard problems (mushroom and waveform for instance, which have very many support vectors), it is observed that the constraint choice based on  $\Omega e$  shows better performance than the other. This is because the number of on-boundary support vectors is nevertheless small in the hard cases as summarized in Table 3.2. Since both relative residual and gap are required to meet the termination criteria, they are within reasonable ranges in all cases, so we do not present them.

In Figure 3.4, the adaptive balanced reduction algorithm is compared with the adaptive nonbalanced algorithm over a range of  $q_U$ . In solving well balanced problems, the two algorithms show little difference as seen in Figure 3.4a. On the other hand, for problems such as isolet having a lot more patterns in one class than the other, balanced selection shows more stable results, especially for small values of  $q_U$ , as seen in Figure 3.4b. For training the machine for a data set with more than two classification labels, a one-class-versus-the-rest approach is frequently employed [SS01, chap. 7], so this problem characteristic is quite common.

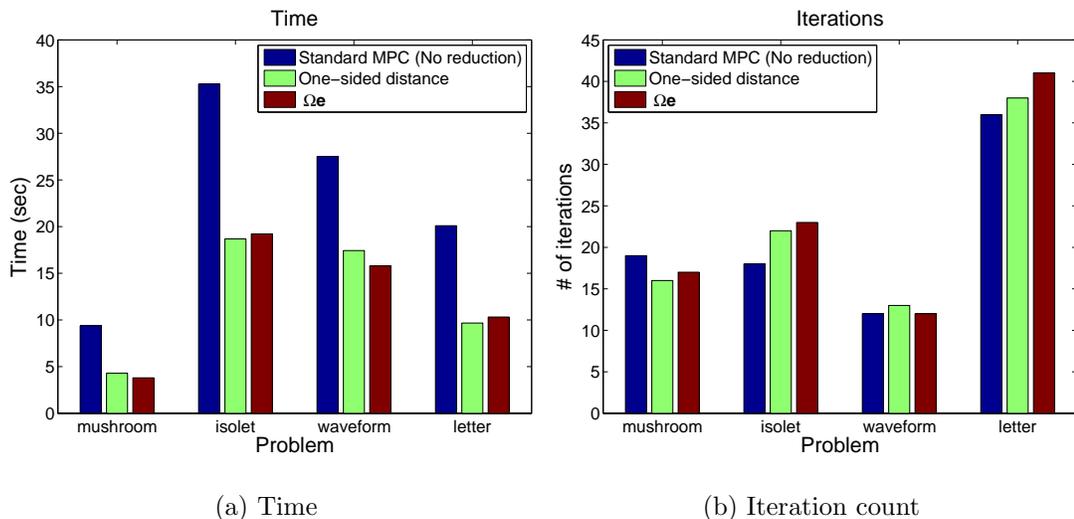
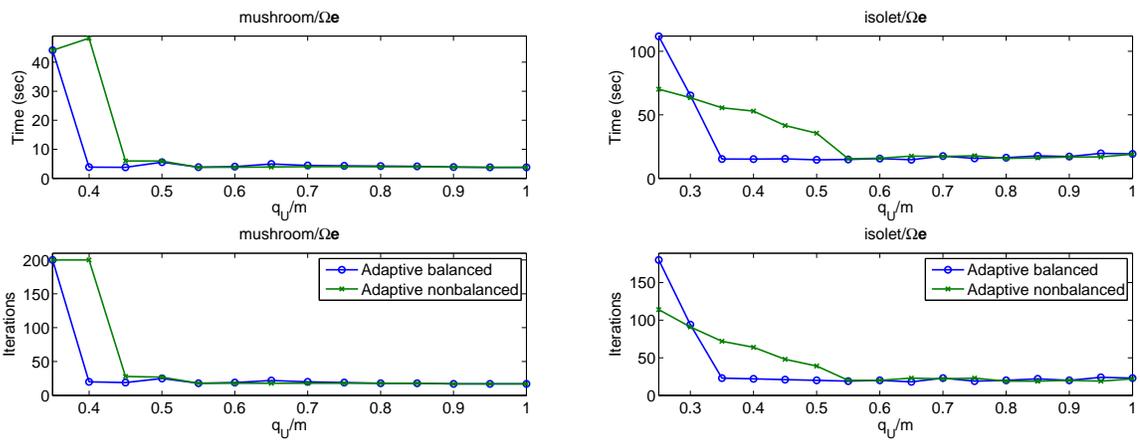


Figure 3.3: Time and iteration count of adaptive reduction with balanced selection are compared to non-reduction algorithm.  $q_U$  is set to  $m$  (100%) for all cases.

In Figure 3.5, the adaptive balanced reduction algorithm is compared with the nonadaptive balanced algorithm over a range of  $q_U$ . Observe that there is little difference in iteration counts among all of these algorithms. Similarly to nonadaptive constraint reduction for linear programming demonstrated in [TAW06], the number of iterations of adaptive and nonadaptive algorithms is almost invariant over a range of  $q_U$ , the upper bound of the index set size. The time taken to solve a problem decreases very slowly or remains almost invariant with the adaptive algorithm as  $q_U$  decreases over a range, whereas the nonadaptive algorithm is more expensive for large values of  $q_U$ .

In Figure 3.6, the two constraint choices based on one-sided distance and  $\Omega\mathbf{e}$  are



(a) Mushroom. In solving a well balanced problem, the two algorithms show little difference.

(b) Isolet. In solving a poorly balanced problem, the balanced algorithm shows better stability.

Figure 3.4: The adaptive balanced and adaptive nonbalanced algorithms are compared, with the constraint choice based on  $\Omega e$ .

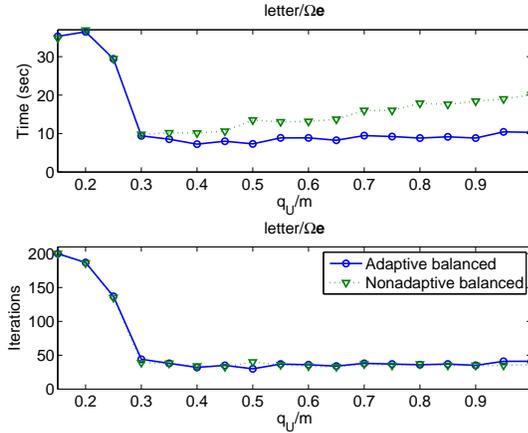
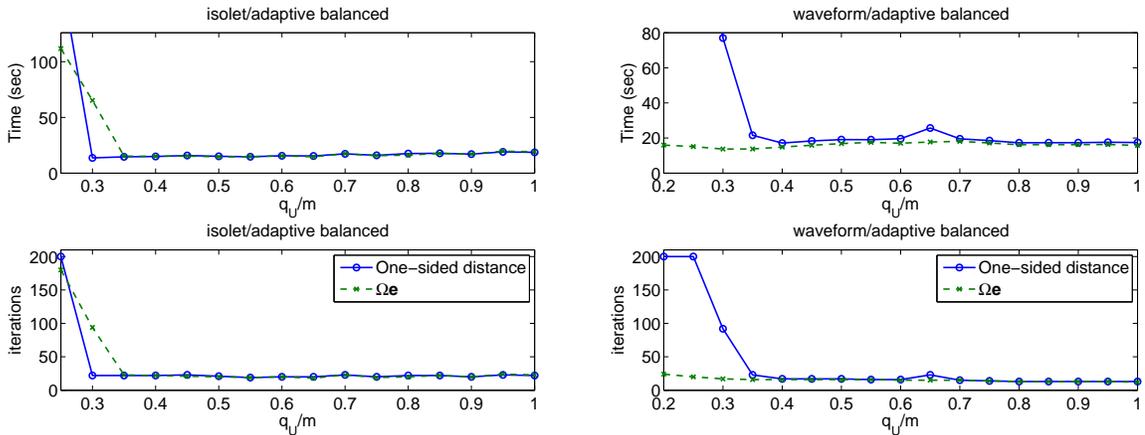


Figure 3.5: Letter. The adaptive and nonadaptive balanced algorithms are compared, with the constraint choice based on  $\Omega\mathbf{e}$ .

applied to the adaptive balanced reduction algorithm and are compared over a range of  $q_U$ . In solving easy problems having almost all support vectors on the boundary hyperplanes, it is hard to say which constraint choice is better than the other. For hard problems, since the  $\Omega\mathbf{e}$  based constraint choice is capable of filtering out more patterns at later iterations, it shows better performance.

In Figure 3.7, the number of patterns used and the complementarity measurement  $\mu$  are traced for every iteration. It is interesting that the graphs of  $\mu$  for the algorithm with high  $q_U$  values are quite close to each other. From these graphs we see that the search direction of the adaptive reduction algorithm is not as good as that of the non-reduction algorithm at early iterations. At later iterations, however, the search direction of the adaptive reduction algorithm is as good as or sometimes better than



(a) Isolet, an easy problem. The two constraint choices are applied to the adaptive balanced reduction.

(b) Waveform, a hard problem. The two constraint choices are applied to the adaptive balanced reduction.

Figure 3.6: The two constraint choices are applied to the adaptive balanced reduction.

that of the standard MPC algorithm.

We compared our algorithm (CRSVM) to LIBSVM [CL01] and SVM<sup>light</sup> [Joa99] on the adult problem of the UCI repository [AN07]. We obtained a formatted problem from the LIBSVM web page [CL01]. The problem consists of 9 sparse training sets with different numbers of sample patterns. Each training set has a corresponding testing set. For this comparison, we used the linear SVM. In other words, we gave the algorithms  $\mathbf{X}$  in a sparse format with no modification except the normalization (3.66). We used adaptive balanced constraint reduction, choosing patterns based on  $\Omega e$ . Figure 3.8 shows the timing results of algorithms on the data sets. Observe that the timing curve of our algorithm is close to linear, while those of LIBSVM and

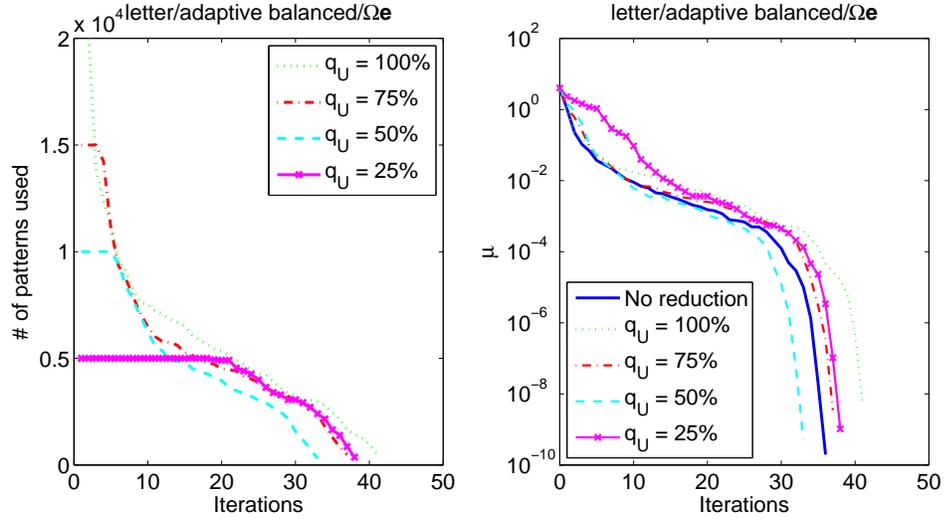


Figure 3.7: Letter. Adaptive balanced reduction based on  $\Omega \mathbf{e}$  constraint choice.

$\text{SVM}^{light}$  are between linear and cubic [Pla99].

### 3.4.2 Nonlinear SVM Examples

We compared our algorithm to LIBSVM and  $\text{SVM}^{light}$ . We used adaptive balanced constraint reduction, choosing patterns based on  $\Omega \mathbf{e}$ . We tested the algorithms on the adult problem of the UCI repository. For this comparison, we used a Gaussian kernel

$$k(\mathbf{x}, \bar{\mathbf{x}}) := \exp(-\|\mathbf{x} - \bar{\mathbf{x}}\|/(2\sigma^2)) \quad (3.67)$$

with  $\sigma := \sqrt{l/2}$ , where  $l := 123$  is the dimension of the input patterns.<sup>3</sup>

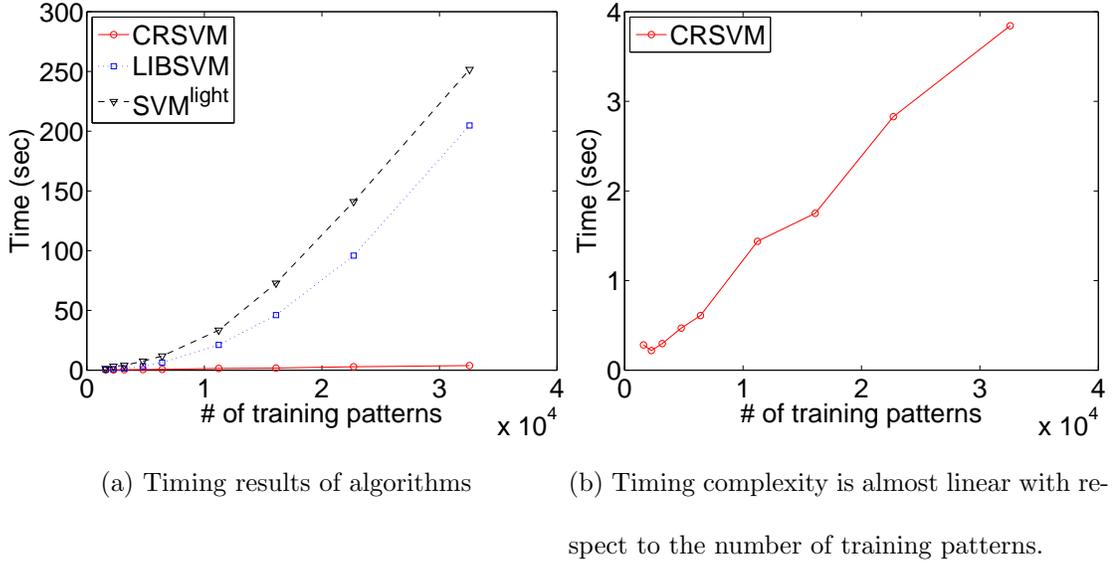


Figure 3.8: Timing results of algorithms for linear SVM training on adult data sets.

To approximate the Gram matrix  $\mathbf{K}$ , we used MATLAB’s `eigs` routine. It uses the Arnoldi iteration [LS96] to generate  $n$  eigenvalues  $\mathbf{\Lambda} := \text{diag}(\lambda_1, \dots, \lambda_n)$  and eigenvectors  $\mathbf{V} := [\mathbf{v}_1, \dots, \mathbf{v}_n]$ . Each iteration of `eigs` involves a matrix-vector product  $\mathbf{K}\mathbf{p}$ . Naive computation of  $\mathbf{K}\mathbf{p}$  costs  $O(m^2)$ , but by using the improved fast Gauss transform (IFGT) [YDD05, RYDG05] we approximate  $\mathbf{K}\mathbf{p}$  and reduce the cost to  $O(m)$ . We use EIGS to denote this approximation method. IFGT is written in C and its MATLAB interface is provided by its authors.

IFGT belongs to the fast multipole methods [GR87]. Given  $m$  source points  $\mathbf{x}_1, \dots, \mathbf{x}_m$  in  $\ell$  dimensions, a target point  $\mathbf{v}$  in  $\ell$  dimensions and  $m$  coefficients  $\mathbf{p} := [p_1, \dots, p_m]^T$ , IFGT uses clustering and truncated multivariate Taylor series

---

<sup>3</sup>This is the default setting of Gaussian kernel in LIBSVM.

Size	LIBSVM	SVMLight	CRSVM(+EIGS)	CRSVM(+CHOL)
1605	83.57	83.57	83.62	83.60
2265	83.94	83.94	83.93	83.95
3185	83.85	83.84	83.85	83.84
4781	83.97	83.97	83.97	83.97
6414	84.15	84.15	84.17	84.19
11220	84.17	84.18	84.21	84.21
16100	84.58	84.58	84.58	84.45
22696	85.01	-	84.82	84.98
32561	84.82	-	84.92	84.85

Table 3.3: Accuracy shown in percentage of correctly classified testing patterns.

to evaluate  $\hat{G}(\mathbf{v})$ , an approximation to the discrete Gauss transform

$$G(\mathbf{v}) := \sum_{j=1}^m p_j k(\mathbf{v}, \mathbf{x}_j),$$

where  $k(\cdot, \cdot)$  is the Gaussian kernel (3.67). If the source points are the input patterns used for the computation of the Gram matrix  $\mathbf{K}$ , then  $\hat{G}(\mathbf{x}_i)$  approximates the  $i^{\text{th}}$  component of  $\mathbf{Kp}$ . Thus  $[\hat{G}(\mathbf{x}_1), \dots, \hat{G}(\mathbf{x}_m)]^T$  is an approximation of  $\mathbf{Kp}$ .

In addition to EIGS, we implemented in MATLAB the low rank Cholesky factorization with symmetric pivoting [FS02]. It returns a rank  $n$  Cholesky factor  $\mathbf{L}$  and a pivoting matrix  $\mathbf{P}$  such that  $\mathbf{P}^T \mathbf{L} \mathbf{L}^T \mathbf{P} \approx \mathbf{K}$ . We refer to this approximation method

as CHOL.

Figure 3.9 shows results of the Gram matrix approximation on the adult data set.<sup>4</sup> Figure 3.9a and 3.9b show relative error of the low rank approximation to  $\mathbf{K}$  measured by  $\|\mathbf{K} - \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T\|_\infty/\|\mathbf{K}\|_\infty$  and  $\|\mathbf{K} - \mathbf{P}^T\mathbf{L}\mathbf{L}^T\mathbf{P}\|_\infty/\|\mathbf{K}\|_\infty$ . As illustrated in Figure 3.9a, with lower rank EIGS approximates  $\mathbf{K}$  better than CHOL. However, since EIGS uses IFGT to approximate  $\mathbf{K}\mathbf{p}$ , the tolerance to IFGT should be tightened as the rank is increased. In our experiments we set the IFGT tolerance to  $\min(0.5, 4/\sqrt{\text{rank}})$ . As depicted in Figure 3.9b, when the rank is fixed, errors in the Gram matrix approximation by CHOL are more affected by the number of training patterns. Figure 3.9c and 3.9d show time to approximate  $\mathbf{K}$ . In Figure 3.9a and 3.9c, EIGS and CHOL were tested on the set of 6414 training patterns. In Figure 3.9b and 3.9d, we requested a rank 64 approximation from EIGS and a rank 300 approximation from CHOL.

Figure 3.10a compares CRSVM with the other methods. Notice both LIBSVM and SVM<sup>light</sup> are implemented in the C language. We expect we can improve CRSVM and CHOL by implementing them in C. We requested 64 eigenvalues and eigenvectors from EIGS to form a rank 64 approximation to  $\mathbf{K}$ . We set CHOL to form a rank 300 approximation. Figure 3.10b shows separated timing results for the approximation and training. Table 3.3 shows accuracy of the classifier each algorithm generated. Accuracy denotes the percentage of correctly classified testing patterns. The classifiers

---

<sup>4</sup>IFGT supports dense input only.

were tested on testing data sets associated with the training set. Notice with a proper approximation, it is possible to get a classifier performing as well as the one trained with the exact matrix.

### 3.4.3 Visualization of the Iterations

To illustrate how our algorithm achieves efficiency, we made a two dimensional toy problem by generating 2000 uniformly distributed random points in  $[-1, 1] \times [-1, 1]$ . Then we set an intentional ellipsoidal separation gap and deleted patterns inside the gap, resulting in 1727 remaining patterns. Figure 3.11 shows snapshots of several iterations of the adaptive balanced reduction algorithm (with  $q_U := m$ ) in solving the problem. Patterns are chosen based on  $\mathbf{\Omega e}$ . To find an ellipsoidal classifier, the mapping (3.6) associated with the second order homogeneous polynomial kernel is used to map the problem's 2-dimensional input space to a 3-dimensional feature space. The dashed ellipsoids are the boundary curves (corresponding to boundary planes in the feature space). As the iteration count increases, the number of selected patterns decreases and only the on-boundary support vectors are chosen at the end, leading to significant time savings.

## 3.5 Conclusion

We presented an algorithm for training SVMs using a constraint reduced IPM with a direct solver for the normal equations. Significant time saving is reported for all problems. If we substituted an iterative solver, constraint reduction would reduce the cost of matrix-vector product.

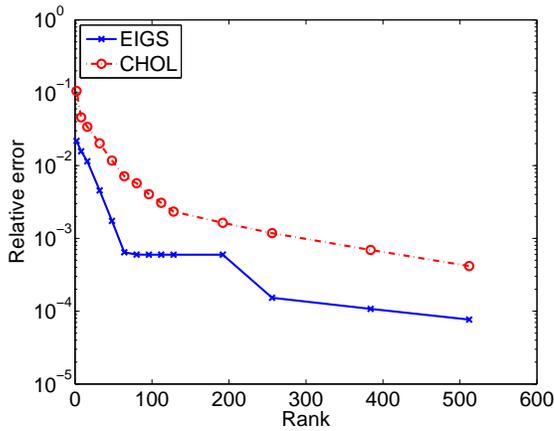
The  $\Omega\mathbf{e}$  constraint choice proved to be more effective than the one-sided distance, especially for hard problems which have many off-boundary support vectors. Other constraint choice heuristics can be used provided that they can include constraints which seem to be most active at the current point. Blending different constraint choices is also allowable. We also report that balanced selection is effective in training SVM for nonbalanced data sets.

We compared our algorithm to other popular algorithms including LIBSVM and SVM<sup>light</sup>. We showed potential of our algorithms on training linear SVMs. In training nonlinear SVMs, substantial time is consumed in calculating an approximation to the Gram matrix.

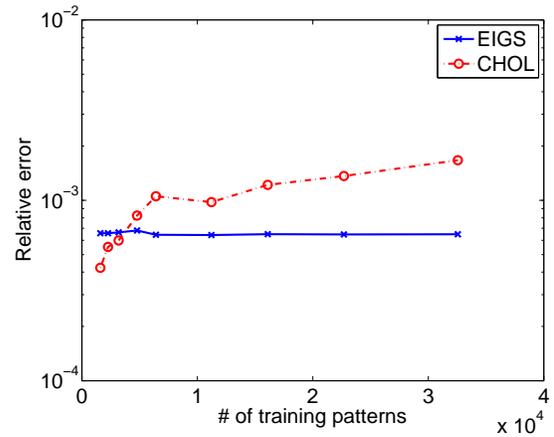
Snapshots of the 2D toy problem were presented to visualize how the adaptive reduction algorithm works. The algorithm acts as an adaptive filter for excluding unnecessary patterns.

Parallelization is a challenging topic for the reduction algorithm. Computation of the matrix arising in the normal equation requires the chosen constraints to be

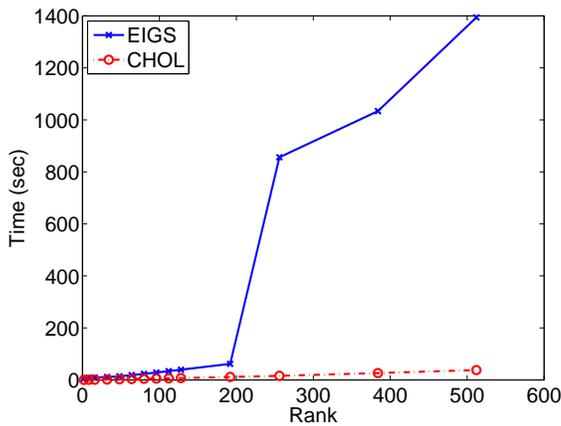
well distributed among multiple processors. This is problematic especially for later iterations when the chosen constraints might be concentrated on a small number of processors. For efficiency in essential support vector identification, we might want to distribute the patterns so that every processor has patterns at a wide range of distances from the separating hyperplane.



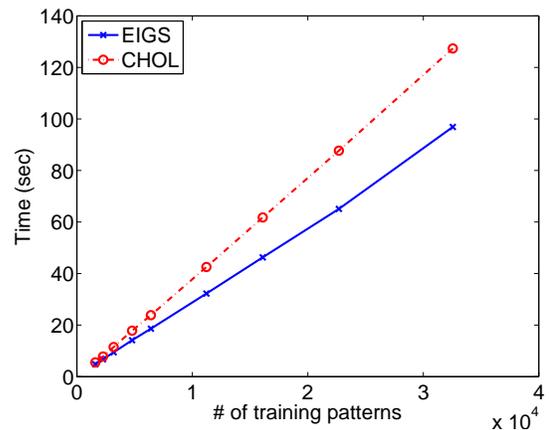
(a) Relative errors in the Gram matrix approximation. Tested on the training set with 6414 training patterns.



(b) Relative errors in the Gram matrix approximation. We requested rank 64 from EIGS and rank 300 from CHOL.

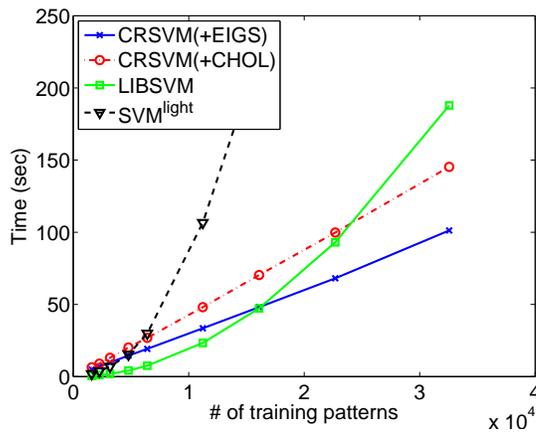


(c) Time to approximate the Gram matrix. Tested on the training set with 6414 training patterns.

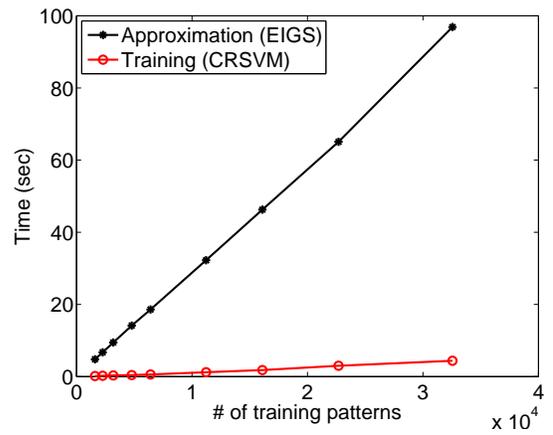


(d) Time to approximate the Gram matrix. We requested rank 64 from EIGS and rank 300 from CHOL.

Figure 3.9: Gram matrix approximation on adult data sets



(a) Timing results of algorithms on adult data sets. CRSVM result includes Gram matrix approximation.



(b) Time to approximate the Gram matrix and train SVMs with a rank 64 approximation through EIGS.

Figure 3.10: Nonlinear SVM training on adult data sets

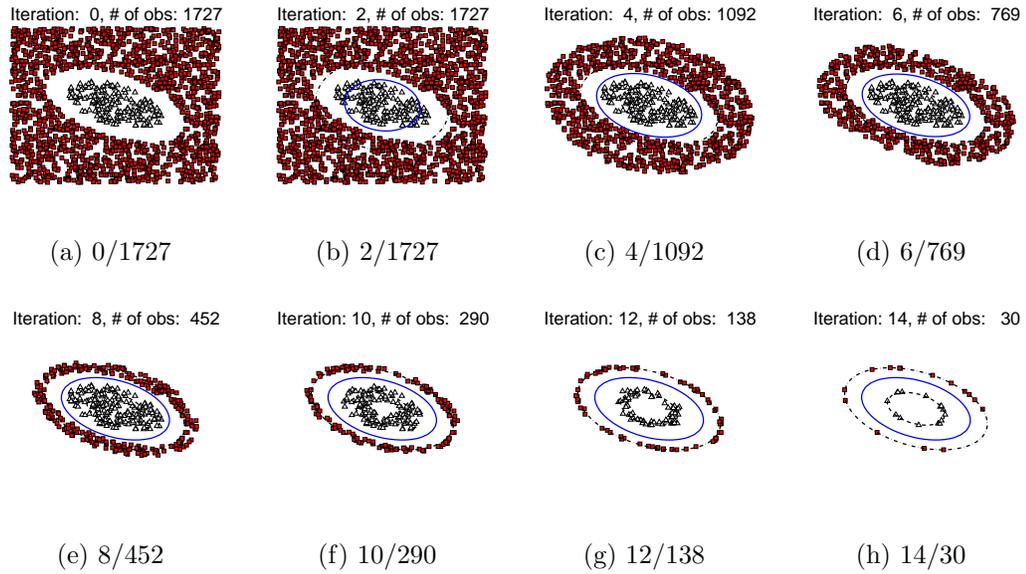


Figure 3.11: Snapshots of finding a classifier using the adaptive reduction algorithm for a randomly generated toy problem in 2-dimensional input space with patterns eliminated intentionally around a hand-generated ellipsoid centered at the origin. The mapping associated with the second order homogeneous polynomial kernel is used to find the surface. The numbers below each figure indicate (iteration)/(number of patterns used).

## Chapter 4

### Conclusions and Future Directions

In solving convex quadratic programming (CQP) with many inequality constraints, the primal-dual interior-point methods (PDIPMs) are frequently used due to their good performance in practice. The most expensive part in these methods is computing a search direction involving every constraint. Time for the computation is proportional to the number of constraints and this becomes the main bottleneck as the number of constraints increases.

In Chapter 2, we proposed a very effective mechanism for saving computational time in finding a search direction, without increasing the total iteration count. The mechanism is to assemble the matrix for the normal equations with a subset of the constraints while adaptively reducing the number of constraints involved as the iterate approaches the optimal solution. We first developed a primal-dual affine-scaling (PDAS) interior-point method (IPM) for the inequality constrained standard CQP form. We provided the convergence analysis for the method in Appendix B. We

discussed an extension to the standard form, to broaden applications of constraint reduction. Then we demonstrated the potential of our algorithm in solving large problems including data fitting.

Recently Winternitz et al. [WNTO07] proposed a convergence-proven constraint reduced Mehrotra’s predictor corrector (MPC) variant for linear programming (LP). They showed significant performance improvement over a standard MPC algorithm. Extending the constraint reduced MPC variant to the CQP is also possible.

However, it is still in question whether the constraint reduction algorithms [TAW06, WNTO07] have a guaranteed polynomial complexity bound. This issue is very critical in convincing practitioners. Devising constraint-reduced algorithms with polynomial complexity for LP and CQP is a topic for future research.

In Chapter 3, we applied constraint reduction to SVM training using an MPC variant [GG05]. We proposed several heuristics to save computational time. Adaptive balanced selection of training patterns was shown to be effective in training the SVM on an unbalanced data set. In practice, since the one-class-versus-the-other approach is frequently used for training the SVM on a data set with more than two classification labels, unbalanced data sets are quite common. The multicategory support vector machine is another approach to train the SVM on a data set with multiple class labels [LLW04]. The approach uses a multi-class objective function and constraints, thus generating multiple classifiers by solving a single CQP problem. Extending the constraint reduction to this SVM formula may be investigated in the future.

We visualized how the adaptive constraint reduction algorithm works in training the SVM. The proposed algorithm is realized as adaptive pattern filtering in the training process. Starting with many training patterns, the algorithm omits more and more unnecessary patterns in the normal matrix assembly as the intermediate classifier approaches the optimal one.

As we discussed in section 3.3.3, the constraint reduction algorithm for training the SVM is not exactly the same as that for solving the extended problem in section 2.4. Developing an algorithm common to all problem formulations and demonstrating its performance are future research topics.

We discussed generalization of the constraint reduction algorithm to the training of the nonlinear SVMs through the Gram matrix approximation. As demonstrated in section 3.4.2, the Gram matrix approximation needs much more time than the training process. To speed the approximation process, we used the improved fast Gaussian transform (IFGT) [YDD05, RYDG05], which belongs to the fast multipole methods. With the IFGT, matrix-vector products involving the Gram matrix can be approximated only for the Gaussian kernel. Fast multipole methods for other kernels may be investigated.

In Chapter 2 and 3, we demonstrated effectiveness of the constraint reduction algorithms when a direct solver is used for the normal equations. In using the direct solver, we did not consider a multiprocessor environment. It is becoming a norm that a CPU chip includes multiple processors. Another emerging multiprocessor is

the graphics processing unit (GPU), which is often regarded as a coprocessor helping CPU with parallel processing. A single GPU has many more processing units than a CPU. In [JO07b], we presented an IPM implementation which solves the normal equations with the use of matrix multiplication and decomposition routines working on a single GPU [JO07a]. Since the constraint reduction algorithms use matrix computation routines in BLAS (Basic Linear Algebra Subprograms) [BDD<sup>+</sup>02] and LAPACK (Linear Algebra PACKage) [ABD<sup>+</sup>90] to solve the normal equations, extending the constraint reduction algorithms to the single chip multiprocessor environment is rather straightforward.

However, developing an efficient constraint reduction algorithm can be challenging in a CPU or GPU cluster environment, where CPUs or GPUs are distributed among multiple machines connected through a network. Since the constraints are dynamically chosen and communication among machines is expensive, load balancing can be difficult. Thus, naive use of BLAS and LAPACK developed for a cluster system does not lead to the most efficient constraint reduction algorithm. Dynamic distribution of constraints should be considered in such environment.

The constraint reduction also reduces computational time required for matrix vector products in solving the normal equations when an iterative solver such as the preconditioned conjugate gradient method [Saa03, chap. 9] is used. To guarantee the success of an iterative solver, it is critical to use a good preconditioner. Since the reduced normal matrix (2.39) changes every iteration, the preconditioner

should change accordingly. An adaptive updating strategy is considered for a standard (non-reduced) MPC variant for LP in [WO00]. Some effort may be dedicated to incorporating an iterative method into the constraint reduction.

In conclusion, we provided convergence analysis of the constraint reduced PDAS IPM for a standard CQP problem and its extension. We presented an adaptive scheme which can significantly save computational time. Also we demonstrated the use of the adaptive constraint reduction algorithm for solving large CQP problems and training SVMs on large data sets.

## Appendix A

### Geometric Properties of Convex Quadratic Programming

This chapter explains geometric properties of the CQP to aid in understanding of the convergence proof in Appendix B.

#### A.1 Definitions

**Definition A.1.** *In  $\mathbb{R}^n$ , the open ball  $B$  of radius  $r > 0$  with center  $\mathbf{v}$  is defined as*

$$B(\mathbf{v}, r) := \{\mathbf{x} : \|\mathbf{x} - \mathbf{v}\| < r\}.$$

**Definition A.2.** *In  $\mathbb{R}^n$ , the closed ball  $\bar{B}$  of radius  $r > 0$  with center  $\mathbf{v}$  is defined as*

$$\bar{B}(\mathbf{v}, r) := \{\mathbf{x} : \|\mathbf{x} - \mathbf{v}\| \leq r\}.$$

In  $\mathbb{R}$ , the open ball and closed ball of radii  $r > 0$  centered at  $x$  are the intervals  $(x - r, x + r)$  and  $[x - r, x + r]$ , respectively.

**Definition A.3.** *For a set  $\mathcal{S} \subseteq \mathbb{R}^n$ , the closure of  $\mathcal{S}$ ,  $\bar{\mathcal{S}}$ , is the set of points  $\mathbf{x}$  such that the intersection of  $\mathcal{S}$  and any open ball centered at  $\mathbf{x}$  is nonempty.*

**Definition A.4.** For a set  $\mathcal{S} \subseteq \mathbb{R}^n$ , the interior of  $\mathcal{S}$ ,  $\text{int}(\mathcal{S})$ , is defined to be the set of points  $\mathbf{x}$ , for which there exists an open ball centered at  $\mathbf{x}$  that is contained in  $\mathcal{S}$ .

In  $\mathbb{R}^n$ , the closure of an open ball of radius  $r$  centered at  $\mathbf{x}$  is the closed ball of radius  $r$  centered at  $\mathbf{x}$ . Likewise, the interior of a closed ball of radius  $r$  centered at  $\mathbf{x}$  is the open ball of radius  $r$  centered at  $\mathbf{x}$ .

**Definition A.5.** A set  $\mathcal{S} \subseteq \mathbb{R}^n$  is a linear subspace of  $\mathbb{R}^n$  if and only if it satisfies

(i) For any two vectors  $\mathbf{v}$  and  $\mathbf{u}$  in  $\mathcal{S}$ , their sum  $\mathbf{v} + \mathbf{u}$  is also in  $\mathcal{S}$ .

(ii) For any vector  $\mathbf{v} \in \mathcal{S}$  and for any scalar  $\alpha \in \mathbb{R}$ , their product  $\alpha\mathbf{v}$  is also in  $\mathcal{S}$ .

From the definition above, it follows that the set  $\{\mathbf{0}\}$  containing only the origin is a linear subspace of  $\mathbb{R}^n$ .

**Definition A.6.** A set  $\mathcal{S}$  is an affine set of  $\mathbb{R}^n$  if and only if, for any two points in  $\mathcal{S}$ , any point in the (infinite) line passing through the points is contained in the set. In other words, for any points  $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{S}$  and for any scalar  $\alpha \in \mathbb{R}$ , the point  $\mathbf{x} := \alpha\mathbf{x}_1 + (1 - \alpha)\mathbf{x}_2$  is also in  $\mathcal{S}$ .

From the definition above, it follows that a set  $\mathcal{S}$  is an affine set if and only if  $\mathcal{S} - \mathbf{v} = \{\bar{\mathbf{x}} : \bar{\mathbf{x}} = \mathbf{x} - \mathbf{v}, \mathbf{x} \in \mathcal{S}\}$  is a linear subspace for any vector  $\mathbf{v}$  in  $\mathcal{S}$ . In addition, the set  $\{\mathbf{x}\}$  containing only a single point  $\mathbf{x} \in \mathbb{R}^n$  is an affine set of  $\mathbb{R}^n$ .

**Definition A.7.** For a set  $\mathcal{S} \in \mathbb{R}^n$ , the affine hull of  $\mathcal{S}$ ,  $\text{aff } \mathcal{S}$ , is defined to be the smallest affine set containing  $\mathcal{S}$ .

The affine hull of a set containing only one point in  $\mathbb{R}^n$  can be defined as the set itself.

**Definition A.8.** For a set  $\mathcal{S}$ , the relative interior of  $\mathcal{S}$ ,  $\text{ri } \mathcal{S}$ , is defined to be the interior of  $\mathcal{S}$  relative to  $\text{aff } \mathcal{S}$ , the affine hull of  $\mathcal{S}$ . In other words, for any point  $\mathbf{x}$  in  $\text{ri } \mathcal{S}$ , there exists some open ball  $B(\mathbf{x}, r)$  whose intersection with the affine hull of  $\mathcal{S}$  is contained in  $\mathcal{S}$ :

$$\forall \mathbf{x} \in \text{ri } \mathcal{S}, \exists r > 0 \text{ s.t. } B(\mathbf{x}, r) \cap \text{aff } \mathcal{S} \subseteq \mathcal{S}.$$

The relative interior of a set containing only a single point  $\mathbf{x} \in \mathbb{R}^n$  is the set itself, because the affine hull is the set itself which is contained in any open ball centered at the point.

The constraints of (2.2) form a polyhedron in  $\mathbb{R}^n$ . The set of all points in and on the polyhedron is a convex set, and we will minimize convex functions on convex sets.

**Definition A.9.** A set  $\mathcal{X}$  is convex if and only if, for any two points  $\mathbf{x}_1$  and  $\mathbf{x}_2$  in  $\mathcal{X}$ , any convex combination of the two points is also in  $\mathcal{X}$ .

$$\alpha \mathbf{x}_1 + (1 - \alpha) \mathbf{x}_2 \in \mathcal{X}, \forall \alpha \in [0, 1]. \quad (\text{A.1})$$

**Definition A.10.** A function  $f(\mathbf{x})$  defined on a convex domain set  $\mathcal{X}$  is convex if and only if, for any two points  $\mathbf{x}_1$  and  $\mathbf{x}_2$  in  $\mathcal{X}$  and for any  $\alpha \in [0, 1]$ ,

$$f(\alpha \mathbf{x}_1 + (1 - \alpha) \mathbf{x}_2) \leq \alpha f(\mathbf{x}_1) + (1 - \alpha) f(\mathbf{x}_2). \quad (\text{A.2})$$

## A.2 Properties of Convex Functions

Convex functions have the following property.

**Lemma A.1.** *The sum of convex functions is also convex.*

*Proof.* Let  $g(\mathbf{x})$  and  $h(\mathbf{x})$  be convex functions, and let  $f(\mathbf{x}) := g(\mathbf{x}) + h(\mathbf{x})$ . Now check whether  $f(x)$  satisfies the conditions of Definition A.10.

$$\begin{aligned} f(\alpha\mathbf{x}_1 + (1 - \alpha)\mathbf{x}_2) &= g(\alpha\mathbf{x}_1 + (1 - \alpha)\mathbf{x}_2) + h(\alpha\mathbf{x}_1 + (1 - \alpha)\mathbf{x}_2) \\ &\leq \alpha g(\mathbf{x}_1) + (1 - \alpha)g(\mathbf{x}_2) + \alpha h(\mathbf{x}_1) + (1 - \alpha)h(\mathbf{x}_2) \\ &= \alpha f(\mathbf{x}_1) + (1 - \alpha)f(\mathbf{x}_2). \end{aligned}$$

□

**Lemma A.2.** *The objective function  $f(\mathbf{x})$  of the CQP standard form (2.2) is convex if and only if the Hessian matrix is positive semidefinite.*

*Proof.* Let us check the definition of convexity by subtracting the left-hand side from the right-hand side in (A.2):

$$\begin{aligned} &\alpha f(\mathbf{x}_1) + (1 - \alpha)f(\mathbf{x}_2) - f(\alpha\mathbf{x}_1 + (1 - \alpha)\mathbf{x}_2) \\ &= \alpha\mathbf{x}_1^T \mathbf{H}\mathbf{x}_1 + (1 - \alpha)\mathbf{x}_2^T \mathbf{H}\mathbf{x}_2 + \alpha\mathbf{c}^T \mathbf{x}_1 + (1 - \alpha)\mathbf{c}^T \mathbf{x}_2 \\ &\quad - (\alpha\mathbf{x}_1 + (1 - \alpha)\mathbf{x}_2)^T \mathbf{H}(\alpha\mathbf{x}_1 + (1 - \alpha)\mathbf{x}_2) - \mathbf{c}^T(\alpha\mathbf{x}_1 + (1 - \alpha)\mathbf{x}_2) \\ &= \alpha(1 - \alpha)(\mathbf{x}_1 - \mathbf{x}_2)^T \mathbf{H}(\mathbf{x}_1 - \mathbf{x}_2), \end{aligned}$$

from which, since  $\alpha(1 - \alpha) \geq 0$ , and  $\mathbf{x}_1$  and  $\mathbf{x}_2$  are arbitrary, it follows that the Hessian is positive semidefinite if and only if the objective function is convex. □

### A.3 Geometric Properties of the Solution Set $\mathcal{F}_P^*$

**Lemma A.3.** *The line segment connecting two arbitrary points  $\mathbf{x}_1$  and  $\mathbf{x}_2$  in  $\mathcal{F}_P^*$  is parallel to the nullspace of  $\mathbf{H}$ . In other words,  $\mathbf{H}(\mathbf{x}_1 - \mathbf{x}_2) = \mathbf{0}$ .*

*Proof.* Let  $\mathbf{x}_1$  and  $\mathbf{x}_2$  be two arbitrary points in  $\mathcal{F}_P^*$ . Since the objective function is convex, we know that,  $\forall \hat{\mathbf{x}} \in \mathcal{F}_P, \forall \mathbf{x}^* \in \mathcal{F}_P^*, \forall \alpha \in (0, 1]$ , and for  $\mathbf{v} = \hat{\mathbf{x}} - \mathbf{x}^*$ ,

$$\begin{aligned} f(\mathbf{x}^* + \alpha\mathbf{v}) &= f(\alpha\hat{\mathbf{x}} + (1 - \alpha)\mathbf{x}^*) \\ &\leq \alpha f(\hat{\mathbf{x}}) + (1 - \alpha)f(\mathbf{x}^*) \\ &= \alpha f(\mathbf{x}^* + \mathbf{v}) + (1 - \alpha)f(\mathbf{x}^*) \\ &= \alpha f(\mathbf{x}^* + \mathbf{v}) - \alpha f(\mathbf{x}^*) + f(\mathbf{x}^*). \end{aligned}$$

Since  $\alpha > 0$  and  $f(\mathbf{x}^*) \leq f(\mathbf{x}^* + \alpha\mathbf{v})$ , it immediately follows that

$$0 \leq \frac{f(\mathbf{x}^* + \alpha\mathbf{v}) - f(\mathbf{x}^*)}{\alpha} \leq f(\mathbf{x}^* + \mathbf{v}) - f(\mathbf{x}^*). \quad (\text{A.3})$$

By taking the limit as  $\alpha \rightarrow 0$ , we know, from (A.3), that

$$f(\mathbf{x}^*) \leq f(\mathbf{x}^*) + \mathbf{v}^T \nabla f(\mathbf{x}^*) \leq f(\hat{\mathbf{x}}).$$

This implies that, since the objective function value is the same at any point in  $\mathcal{F}_P^*$ , for any two points  $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{F}_P^*$ , the directional derivatives from  $\mathbf{x}_1$  toward  $\mathbf{x}_2$  at  $\mathbf{x}_1$  and from  $\mathbf{x}_2$  toward  $\mathbf{x}_1$  at  $\mathbf{x}_2$  are 0, because

$$f(\mathbf{x}_1) \leq f(\mathbf{x}_1) + (\mathbf{x}_2 - \mathbf{x}_1)^T \nabla f(\mathbf{x}_1) \leq f(\mathbf{x}_2) = f(\mathbf{x}_1),$$

$$f(\mathbf{x}_2) \leq f(\mathbf{x}_2) + (\mathbf{x}_1 - \mathbf{x}_2)^T \nabla f(\mathbf{x}_2) \leq f(\mathbf{x}_1) = f(\mathbf{x}_2).$$

So we get

$$(\mathbf{x}_1 - \mathbf{x}_2)^T (\mathbf{H}\mathbf{x}_1 + \mathbf{c}) = 0, \quad (\text{A.4})$$

$$(\mathbf{x}_1 - \mathbf{x}_2)^T (\mathbf{H}\mathbf{x}_2 + \mathbf{c}) = 0. \quad (\text{A.5})$$

By subtracting (A.5) from (A.4), we get

$$(\mathbf{x}_1 - \mathbf{x}_2)^T \mathbf{H}(\mathbf{x}_1 - \mathbf{x}_2) = 0.$$

Since  $\mathbf{H}$  is positive semidefinite, it immediately follows that

$$\mathbf{H}\mathbf{x}_1 = \mathbf{H}\mathbf{x}_2.$$

This implies that the claim holds. □

**Lemma A.4.**  $\mathcal{F}_P^*$  is convex.

*Proof.* Let  $\mathbf{x}_1$  and  $\mathbf{x}_2$  be two arbitrary points in  $\mathcal{F}_P^*$ . For their convex combination  $\bar{\mathbf{x}} := \alpha\mathbf{x}_1 + (1 - \alpha)\mathbf{x}_2$ , with  $\alpha \in [0, 1]$ , the objective function value is

$$\begin{aligned} f(\bar{\mathbf{x}}) &= \frac{1}{2} \bar{\mathbf{x}}^T \mathbf{H}\bar{\mathbf{x}} + \mathbf{c}^T \bar{\mathbf{x}} \\ &= \frac{1}{2} \bar{\mathbf{x}}^T (\alpha\mathbf{H}\mathbf{x}_1 + (1 - \alpha)\mathbf{H}\mathbf{x}_2) + \mathbf{c}^T \bar{\mathbf{x}}. \end{aligned} \quad (\text{A.6})$$

Since  $\mathbf{H}\mathbf{x}_1 = \mathbf{H}\mathbf{x}_2$  by Lemma A.3, (A.6) can be reduced to

$$f(\bar{\mathbf{x}}) = \frac{1}{2} (\alpha\mathbf{x}_1 + (1 - \alpha)\mathbf{x}_2)^T \mathbf{H}\mathbf{x}_1 + \alpha\mathbf{c}^T \mathbf{x}_1 + (1 - \alpha)\mathbf{c}^T \mathbf{x}_2 \quad (\text{A.7})$$

$$= \alpha \left( \frac{1}{2} \mathbf{x}_1^T \mathbf{H} \mathbf{x}_1 + \mathbf{c}^T \mathbf{x}_1 \right) + (1 - \alpha) \left( \frac{1}{2} \mathbf{x}_1^T \mathbf{H} \mathbf{x}_2 + \mathbf{c}^T \mathbf{x}_2 \right) \quad (\text{A.8})$$

$$= \alpha \left( \frac{1}{2} \mathbf{x}_1^T \mathbf{H} \mathbf{x}_1 + \mathbf{c}^T \mathbf{x}_1 \right) + (1 - \alpha) \left( \frac{1}{2} \mathbf{x}_2^T \mathbf{H} \mathbf{x}_2 + \mathbf{c}^T \mathbf{x}_2 \right) \quad (\text{A.9})$$

$$= \alpha f(\mathbf{x}_1) + (1 - \alpha) f(\mathbf{x}_2) = f(\mathbf{x}_1) = f(\mathbf{x}_2). \quad (\text{A.10})$$

Since  $\mathcal{F}_P$  is convex,  $\bar{\mathbf{x}}$  is also in  $\mathcal{F}_P$ . Therefore  $\bar{\mathbf{x}}$  is also in  $\mathcal{F}_P^*$ , which implies that  $\mathcal{F}_P^*$  is convex. □

**Proposition A.5.** *The relative interior of a nonempty convex set  $\mathcal{C} \subseteq \mathbb{R}^n$  is nonempty and convex.*

*Proof.* See [Roc72, Chapter 6]. □

**Lemma A.6.** *The line segment connecting two arbitrary points in  $\mathcal{F}_P^*$  is orthogonal to  $\mathbf{c}$ .*

*Proof.* Let  $\mathbf{x}_1$  and  $\mathbf{x}_2$  be arbitrary two points in  $\mathcal{F}_P^*$ . Since  $\mathbf{H} \mathbf{x}_1 = \mathbf{H} \mathbf{x}_2$  by Lemma A.3, from the difference of the objective function values at  $\mathbf{x}_1$  and  $\mathbf{x}_2$ , we can obtain the following:

$$\begin{aligned} 0 = f(\mathbf{x}_1) - f(\mathbf{x}_2) &= \frac{1}{2} \mathbf{x}_1^T \mathbf{H} \mathbf{x}_1 + \mathbf{c}^T \mathbf{x}_1 - \frac{1}{2} \mathbf{x}_2^T \mathbf{H} \mathbf{x}_2 - \mathbf{c}^T \mathbf{x}_2 \\ &= \mathbf{c}^T (\mathbf{x}_1 - \mathbf{x}_2). \end{aligned}$$

This implies that the claim holds. □

Lemma A.3 and A.6 imply that, assuming  $\mathcal{F}_P^*$  is nonempty,  $\mathcal{F}_P^*$  is a singleton if the intersection of the nullspace of  $\mathbf{H}$  and  $\mathbf{c}^T$  is a trivial set  $\{\mathbf{0}\}$ , or equivalently if the columns of  $[\mathbf{H}, \mathbf{c}]$  span  $\mathbb{R}^n$ .

**Lemma A.7.** *All the points in the relative interior of  $\mathcal{F}_P^*$  are associated with the same active constraints.*

*Proof.* If  $\mathcal{F}_P^*$  is a singleton, then the claim is immediately true. So, in what follows, we assume  $\mathcal{F}_P^*$  is not a singleton. Since  $\mathcal{F}_P^*$  is not a singleton, there exist at least two different solutions  $\mathbf{x}_1$  and  $\mathbf{x}_2$ . Let  $\hat{\mathbf{x}}$  be a convex combination of  $\mathbf{x}_1$  and  $\mathbf{x}_2$  such that  $\hat{\mathbf{x}} = \alpha\mathbf{x}_1 + (1 - \alpha)\mathbf{x}_2$ , for some  $\alpha \in (0, 1)$ . By Lemma A.4,  $\hat{\mathbf{x}}$  is also in  $\mathcal{F}_P^*$ . For all  $i$  in  $\mathcal{A}(\hat{\mathbf{x}})$  and for all  $\alpha \in (0, 1)$ , it follows that

$$b_i = \mathbf{a}_i^T \hat{\mathbf{x}} = \alpha \mathbf{a}_i^T \mathbf{x}_1 + (1 - \alpha) \mathbf{a}_i^T \mathbf{x}_2.$$

Since  $\mathbf{a}_i^T \mathbf{x}_1 \geq b_i$  and  $\mathbf{a}_i^T \mathbf{x}_2 \geq b_i$ , it then follows that  $\mathbf{a}_i^T \mathbf{x}_1 = b_i$  and  $\mathbf{a}_i^T \mathbf{x}_2 = b_i$ . Thus

$$\mathcal{A}(\hat{\mathbf{x}}) \subseteq \mathcal{A}(\mathbf{x}_1) \cap \mathcal{A}(\mathbf{x}_2).$$

Now for all  $i \in \mathcal{A}(\mathbf{x}_1) \cap \mathcal{A}(\mathbf{x}_2)$ , it follows that

$$\mathbf{a}_i^T \hat{\mathbf{x}} = \alpha \mathbf{a}_i^T \mathbf{x}_1 + (1 - \alpha) \mathbf{a}_i^T \mathbf{x}_2 = b_i,$$

which yields

$$\mathcal{A}(\mathbf{x}_1) \cap \mathcal{A}(\mathbf{x}_2) \subseteq \mathcal{A}(\hat{\mathbf{x}}).$$

Thus  $\mathcal{A}(\hat{\mathbf{x}}) = \mathcal{A}(\mathbf{x}_1) \cap \mathcal{A}(\mathbf{x}_2)$ .

So, for any two points  $\bar{\mathbf{x}}$  and  $\hat{\mathbf{x}}$  in the relative interior of  $\mathcal{F}_P^*$ , we can pick two points  $\mathbf{x}_1$  and  $\mathbf{x}_2$  from  $\mathcal{F}_P^*$  so that  $\bar{\mathbf{x}}$  and  $\hat{\mathbf{x}}$  are in the interior of the line segment connecting  $\mathbf{x}_1$  and  $\mathbf{x}_2$ . Since  $\mathcal{A}(\bar{\mathbf{x}}) = \mathcal{A}(\mathbf{x}_1) \cap \mathcal{A}(\mathbf{x}_2)$  and  $\mathcal{A}(\hat{\mathbf{x}}) = \mathcal{A}(\mathbf{x}_1) \cap \mathcal{A}(\mathbf{x}_2)$ , it follows that  $\mathcal{A}(\bar{\mathbf{x}}) = \mathcal{A}(\hat{\mathbf{x}})$ .  $\square$

**Lemma A.8.** *Assuming that  $\mathcal{F}_P^*$  is nonempty,  $\mathcal{F}_P^*$  is a singleton (i.e, the solution  $\mathbf{x}^*$  is unique) if and only if, for all  $\mathbf{x}^* \in \mathcal{F}_P^*$ , the intersection of the nullspace of  $\mathbf{H}$  and the nullspace of  $\mathbf{A}_{\mathcal{A}(x^*)}$  is the trivial set  $\{\mathbf{0}\}$ .*

*Proof.* First, suppose  $\mathcal{F}_P^*$  is not a singleton. Since, in view of Lemma A.4 and Proposition A.5, the relative interior of  $\mathcal{F}_P^*$  is nonempty and convex, we can arbitrarily pick two different points  $\mathbf{x}_1$  and  $\mathbf{x}_2$  from the relative interior. In view of Lemma A.3, it follows that

$$\mathbf{x}_1 - \mathbf{x}_2 \in \mathcal{N}(\mathbf{H}). \quad (\text{A.11})$$

On the other hand, in view of Lemma A.7, it follows that  $\mathcal{A}(\mathbf{x}_1) = \mathcal{A}(\mathbf{x}_2) = \mathcal{A}^*$ , from which it follows that

$$\mathbf{A}_{\mathcal{A}^*}(\mathbf{x}_1 - \mathbf{x}_2) = \mathbf{b}_{\mathcal{A}^*} - \mathbf{b}_{\mathcal{A}^*} = \mathbf{0}. \quad (\text{A.12})$$

Thus, from (A.11) and (A.12), it follows that  $\mathbf{x}_1 - \mathbf{x}_2 \in \mathcal{N}(\mathbf{H}) \cap \mathcal{N}(\mathbf{A}_{\mathcal{A}^*})$ . So the intersection of the two nullspaces is nontrivial.

Now let us show the converse. Let  $\mathbf{s}^*$  and  $\boldsymbol{\lambda}^*$  be the slack and multiplier variables associated with the unique solution  $\mathbf{x}^*$  such that the KKT conditions (2.9)-(2.12) are satisfied at  $(\mathbf{x}^*, \mathbf{s}^*, \boldsymbol{\lambda}^*)$ . Suppose that the intersection of the two nullspaces is not  $\{\mathbf{0}\}$ ; then there exists a nonzero vector  $\mathbf{v}$  such that,

$$\mathbf{H}\mathbf{v} = \mathbf{0},$$

$$\mathbf{A}_{\mathcal{A}(\mathbf{x}^*)}\mathbf{v} = \mathbf{0}.$$

Thus, for any scalar  $\alpha$ ,

$$\mathbf{H}(\mathbf{x}^* + \alpha \mathbf{v}) = \mathbf{H}\mathbf{x}^*, \quad (\text{A.13})$$

$$\mathbf{A}_{\mathcal{A}(\mathbf{x}^*)}(\mathbf{x}^* + \alpha \mathbf{v}) = \mathbf{A}_{\mathcal{A}(\mathbf{x}^*)}\mathbf{x}^* = \mathbf{b}_{\mathcal{A}(\mathbf{x}^*)}. \quad (\text{A.14})$$

Equation (A.13) implies that  $\mathbf{H}(\mathbf{x}^* + \alpha \mathbf{v}) - \mathbf{A}^T \boldsymbol{\lambda}^* + \mathbf{c} = \mathbf{0}$  and (A.14) implies that  $\mathcal{A}(\mathbf{x}^*) \subseteq \mathcal{A}(\bar{\mathbf{x}}(\alpha))$  where  $\bar{\mathbf{x}}(\alpha) = \mathbf{x}^* + \alpha \mathbf{v}$ . Since  $\mathbf{A}_{\mathcal{A}(\mathbf{x}^*)^c} \mathbf{x}^* > \mathbf{b}_{\mathcal{A}(\mathbf{x}^*)^c}$ , there exists a sufficiently small  $\alpha > 0$  so that

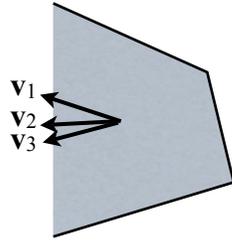
$$\mathbf{A}_{\mathcal{A}(\mathbf{x}^*)^c} \bar{\mathbf{x}}(\alpha) > \mathbf{b}_{\mathcal{A}(\mathbf{x}^*)^c},$$

which, together with (A.14), implies  $\bar{\mathbf{x}} \in \mathcal{F}_P$  and  $\mathcal{A}(\mathbf{x}^*) = \mathcal{A}(\bar{\mathbf{x}})$  for  $\bar{\mathbf{x}} = \bar{\mathbf{x}}(\alpha)$ . Now let  $\bar{\mathbf{s}} = \mathbf{A}\bar{\mathbf{x}} - \mathbf{b}$ . Since  $\mathcal{A}(\mathbf{x}^*) = \mathcal{A}(\bar{\mathbf{x}})$ ,  $\bar{s}_i = s_i^* = 0$  for  $i \in \mathcal{A}(\mathbf{x}^*)$ , and  $\lambda_i^* = 0$  for  $i \notin \mathcal{A}(\mathbf{x}^*)$ , it follows that

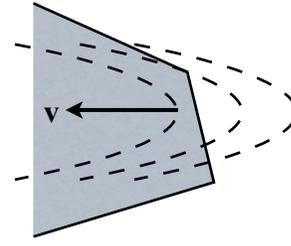
$$\bar{s}_i \lambda_i^* = 0, \quad \forall i \in \mathcal{A}(\mathbf{x}^*),$$

$$\bar{s}_i \lambda_i^* = 0, \quad \forall i \in \mathcal{A}(\mathbf{x}^*)^c.$$

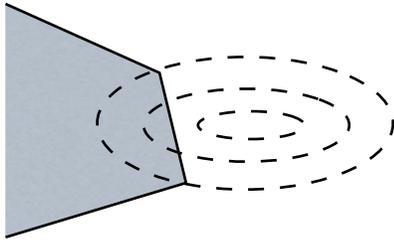
Thus, the KKT conditions are also satisfied at  $(\bar{\mathbf{x}}, \bar{\mathbf{s}}, \boldsymbol{\lambda}^*)$ , so  $\bar{\mathbf{x}} \in \mathcal{F}_P^*$ , and  $\mathcal{F}_P^*$  is not a singleton. □



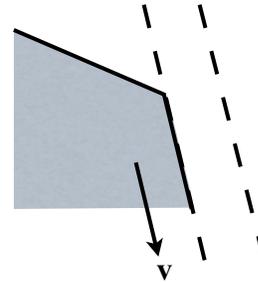
(a) The feasible set is unbounded along  $\mathbf{v}_1$ ,  $\mathbf{v}_2$  and  $\mathbf{v}_3$ , i.e.,  $\mathbf{A}\mathbf{v}_1 \geq 0$ ,  $\mathbf{A}\mathbf{v}_2 \geq 0$  and  $\mathbf{A}\mathbf{v}_3 \geq 0$ .



(b) The objective function has a recession direction  $\mathbf{v}$  along which the feasible set is unbounded. Thus, for any  $\bar{\mathbf{x}} \in \mathcal{F}_P$ , the feasible level set  $\mathcal{F}_P^L(\bar{\mathbf{x}})$  is unbounded along  $\mathbf{v}$  and no solution exists.



(c) Contours of the objective function are ellipsoidal if  $\mathcal{N}(\mathbf{H}) = \{\mathbf{0}\}$ . So, for any  $\bar{\mathbf{x}} \in \mathcal{F}_P$ , the feasible level set  $\mathcal{F}_P^L(\bar{\mathbf{x}})$  is bounded regardless of the boundedness of the primal feasible set  $\mathcal{F}_P$ .



(d) The feasible set is unbounded along  $\mathbf{v}$  and  $\mathbf{v} \in \mathcal{N}(\mathbf{H}) \cap \mathcal{N}(\mathbf{c}^T)$ . As a result,  $\mathcal{F}_P^L(\bar{\mathbf{x}})$  is bounded along  $\mathbf{v}$ .

Figure A.1: Geometries of the feasible set and contours of the objective function in a 2 dimensional space. Examples of unboundedness of the primal feasible set, the level set  $\{\mathbf{x} : f(\mathbf{x}) \leq f(\bar{\mathbf{x}})\}$  of the objective function, and their intersection  $\mathcal{F}_P^L(\bar{\mathbf{x}})$ .

Now we focus on conditions required for the boundedness of the solution set. We

first define the *feasible level set* as

$$\mathcal{F}_P^L(\bar{\mathbf{x}}) := \{\mathbf{x} : f(\mathbf{x}) \leq f(\bar{\mathbf{x}}) \text{ and } \mathbf{x} \in \mathcal{F}_P\}.$$

The feasible level set is the intersection of the level set  $\{\mathbf{x} : f(\mathbf{x}) \leq f(\bar{\mathbf{x}})\}$  and the primal feasible set  $\mathcal{F}_P$ . Assume that  $\mathcal{F}_P^o$  is not empty and let  $\bar{\mathbf{x}}$  be some strictly feasible point. If there exists a nonzero vector  $\mathbf{v}$  such that  $\mathbf{A}\mathbf{v} \geq \mathbf{0}$ , then the feasible set  $\mathcal{F}_P$  is unbounded along  $\mathbf{v}$  (see Figure A.1a). The feasible level set  $\mathcal{F}_P^L(\bar{\mathbf{x}})$  is unbounded if there exists  $\mathbf{v} \in \mathcal{N}(\mathbf{H}) \cap \mathcal{N}(\mathbf{c}^T)$  such that  $\mathbf{A}\mathbf{v} \geq \mathbf{0}$ . If the objective function has a recession direction  $\mathbf{v}$  which coincidentally satisfies  $\mathbf{A}\mathbf{v} \geq \mathbf{0}$ , no solution exists (see Figure A.1b).

Let us investigate the level set  $\{\mathbf{x} : f(\mathbf{x}) \leq f(\bar{\mathbf{x}})\}$  in detail. Geometrically, if  $\mathcal{N}(\mathbf{H}) = \{\mathbf{0}\}$ , then the level set  $\{\mathbf{x} : f(\mathbf{x}) \leq f(\bar{\mathbf{x}})\}$  is ellipsoidal and thus  $\mathcal{F}_P^L(\bar{\mathbf{x}})$  is always bounded (see Figure A.1c). If  $\mathcal{N}(\mathbf{H}) \neq \{\mathbf{0}\}$  and  $\mathcal{N}(\mathbf{H}) \cap \mathcal{N}(\mathbf{c}^T) = \{\mathbf{0}\}$ , the objective function has a recession direction  $\mathbf{v} \in \mathcal{N}(\mathbf{H})$  which is obtained by expressing  $-\mathbf{c} = \mathbf{v} + \mathbf{w}$  where  $\mathbf{w}$  is orthogonal to  $\mathcal{N}(\mathbf{H})$ . Then  $f(\mathbf{x} + \alpha\mathbf{v}) < f(\mathbf{x})$  for any positive scalar  $\alpha$  (see Figure A.1b). If  $\mathcal{N}(\mathbf{H}) \cap \mathcal{N}(\mathbf{c}^T) \neq \{\mathbf{0}\}$ , then the level set  $\{\mathbf{x} : f(\mathbf{x}) \leq f(\bar{\mathbf{x}})\}$  is unbounded in the direction of  $\forall \mathbf{v} \in \mathcal{N}(\mathbf{H}) \cap \mathcal{N}(\mathbf{c}^T)$  (see Figure A.1d). In the last case, the objective function may have a recession direction if  $\mathcal{N}(\mathbf{H}) \not\subseteq \mathcal{N}(\mathbf{c}^T)$ . And the recession direction is obtained in the same way as in the second case.

**Lemma A.9.** *Assume that  $\mathcal{F}_P$  is nonempty. For any  $\bar{\mathbf{x}} \in \mathcal{F}_P$ , the feasible level set*

$\mathcal{F}_P^L(\bar{\mathbf{x}})$  is unbounded if and only if there exists a nonzero vector  $\mathbf{v}$  such that  $\mathbf{A}\mathbf{v} \geq \mathbf{0}$ ,  $\mathbf{H}\mathbf{v} = \mathbf{0}$ , and  $\mathbf{c}^T\mathbf{v} \leq 0$ .

*Proof.* Let us show that the condition is sufficient. Since  $\mathbf{A}\mathbf{v} \geq \mathbf{0}$ , it follows, for any nonnegative scalar  $\alpha$ , that

$$\mathbf{A}(\bar{\mathbf{x}} + \alpha\mathbf{v}) \geq \mathbf{b}, \quad (\text{A.15})$$

which implies that  $\bar{\mathbf{x}} + \alpha\mathbf{v}$  is feasible. In addition, since  $\mathbf{v}$  is in the nullspace of  $\mathbf{H}$ , it follows, for any nonnegative scalar  $\alpha$ , that

$$\begin{aligned} f(\bar{\mathbf{x}} + \alpha\mathbf{v}) &= \frac{1}{2}(\bar{\mathbf{x}} + \alpha\mathbf{v})^T \mathbf{H}(\bar{\mathbf{x}} + \alpha\mathbf{v}) + \mathbf{c}^T(\bar{\mathbf{x}} + \alpha\mathbf{v}) \\ &= \frac{1}{2}\bar{\mathbf{x}}^T \mathbf{H}\bar{\mathbf{x}} + \mathbf{c}^T\bar{\mathbf{x}} + \alpha\mathbf{c}^T\mathbf{v} \\ &= f(\bar{\mathbf{x}}) + \alpha\mathbf{c}^T\mathbf{v} \\ &\leq f(\bar{\mathbf{x}}). \end{aligned} \quad (\text{A.16})$$

Therefore, by (A.15) and (A.16),  $\mathcal{F}_P^L(\bar{\mathbf{x}})$  is unbounded.

Now let us show that the condition is necessary. Suppose that for all nonzero  $\mathbf{v} \in \mathbb{R}^n$ ,  $\mathbf{A}\mathbf{v} \not\geq \mathbf{0}$ ,  $\mathbf{H}\mathbf{v} \neq \mathbf{0}$ , or  $\mathbf{c}^T\mathbf{v} > 0$ . We consider what happens if we take a step from a feasible point  $\bar{\mathbf{x}}$  along a direction  $\mathbf{v}$ . Let  $L(\bar{\mathbf{x}}) := \{\mathbf{x} : f(\mathbf{x}) \leq f(\bar{\mathbf{x}})\}$ . The feasible level set  $\mathcal{F}_P^L(\bar{\mathbf{x}})$  can be expressed as  $\mathcal{F}_P^L(\bar{\mathbf{x}}) = L(\bar{\mathbf{x}}) \cap \mathcal{F}_P$ . This proof will show that either  $L(\bar{\mathbf{x}})$  or  $\mathcal{F}_P$  is bounded along  $\mathbf{v}$ .

First, for any  $\mathbf{v}$  such that  $\mathbf{A}\mathbf{v} \not\geq \mathbf{0}$ , there exists positive  $\alpha$  so that  $\mathbf{A}(\bar{\mathbf{x}} + \alpha\mathbf{v}) \not\geq \mathbf{b}$ . This implies that  $\mathcal{F}_P$  is bounded along  $\mathbf{v}$ .

Second, for any nonzero  $\mathbf{v}$  such that  $\mathbf{H}\mathbf{v} \neq \mathbf{0}$ , since  $\mathbf{H}$  is positive semidefinite, it follows that

$$\frac{df(\bar{\mathbf{x}} + \alpha\mathbf{v})}{d\alpha} = \bar{\mathbf{x}}^T \mathbf{H}\mathbf{v} + \alpha \mathbf{v}^T \mathbf{H}\mathbf{v} + \mathbf{c}^T \mathbf{v}, \quad (\text{A.17})$$

and that

$$\frac{d^2 f(\bar{\mathbf{x}} + \alpha\mathbf{v})}{d\alpha^2} = \mathbf{v}^T \mathbf{H}\mathbf{v} > 0. \quad (\text{A.18})$$

So (A.17) and (A.18) imply that the objective function is strictly convex and goes to infinity as  $|\alpha| \rightarrow \infty$  (see Figure A.2a). Therefore, there exists  $\alpha > 0$  such that  $f(\bar{\mathbf{x}} + \alpha\mathbf{v}) > f(\bar{\mathbf{x}})$  and  $f(\bar{\mathbf{x}} - \alpha\mathbf{v}) > f(\bar{\mathbf{x}})$ . This implies that  $L(\bar{\mathbf{x}})$  is bounded along  $\mathbf{v}$  and  $-\mathbf{v}$ .

Third, for any nonzero  $\mathbf{v}$  such that  $\mathbf{c}^T \mathbf{v} > 0$ ,  $\mathbf{H}\mathbf{v} = \mathbf{0}$ , we get

$$f(\bar{\mathbf{x}} + \alpha\mathbf{v}) = f(\bar{\mathbf{x}}) + \alpha \mathbf{c}^T \mathbf{v} > f(\bar{\mathbf{x}}),$$

for  $\alpha > 0$  (see Figure A.2b). This implies that the objective function is linear and increases along  $\mathbf{v}$  and, thus,  $L(\bar{\mathbf{x}})$  is bounded along  $\mathbf{v}$ .  $\square$

**Lemma A.10.** *Assume that  $\mathcal{F}_P$  and  $\mathcal{F}_P^*$  are nonempty. The feasible level set  $\mathcal{F}_P^L(\bar{\mathbf{x}})$  is bounded for any  $\bar{\mathbf{x}} \in \mathcal{F}_P$  if and only if  $\mathcal{F}_P^*$  is bounded.*

*Proof.* First consider the sufficiency. Suppose that there exists a point  $\bar{\mathbf{x}} \in \mathcal{F}_P$ , for which the feasible level set  $\mathcal{F}_P^L(\bar{\mathbf{x}})$  is unbounded. Then, it follows from Lemma A.9 that there exists a nonzero vector  $\mathbf{v}$  such that  $\mathbf{A}\mathbf{v} \geq \mathbf{0}$ ,  $\mathbf{H}\mathbf{v} = \mathbf{0}$  and  $\mathbf{c}^T \mathbf{v} = 0$ . For an



(a) For a nonzero  $\mathbf{v} \in \mathbb{R}^n$ , if  $\mathbf{H}\mathbf{v} \neq 0$ , the objective function is a strictly convex quadratic function along  $\mathbf{v}$ .

(b) For a nonzero  $\mathbf{v} \in \mathbb{R}^n$ , if  $\mathbf{H}\mathbf{v} = 0$  and  $\mathbf{c}^T \mathbf{v} > 0$ , the objective function is linear and increases along  $\mathbf{v}$ .

Figure A.2

optimal solution  $\mathbf{x}^* \in \mathcal{F}_P^*$ , since  $f(\mathbf{x}^* + \alpha \mathbf{v}) = f(\mathbf{x}^*)$  and  $\mathbf{A}(\mathbf{x}^* + \mathbf{v}) = \mathbf{A}\mathbf{x}^*$ , it follows that  $\mathbf{x}^* + \alpha \mathbf{v} \in \mathcal{F}_P^*$ . Thus the primal solution set  $\mathcal{F}_P^*$  is unbounded.

Now consider the necessity. If  $\mathcal{F}_P^L(\bar{\mathbf{x}})$  is bounded for any  $\bar{\mathbf{x}} \in \mathcal{F}_P$ , then it is bounded for  $\bar{\mathbf{x}} \in \mathcal{F}_P^*$ . □

## Appendix B

### Convergence Proof for Constraint Reduced Primal Feasible Affine-Scaling Primal-Dual Interior-Point Method for CQP

The following proofs for global convergence and the local rate of convergence are adapted from the proofs provided in [TZ94] and [TAW06]. Many parts are identical to [TAW06] except for the action of the Hessian matrix.

#### B.1 Global Convergence Proof

Throughout this section, we use a superscript  $*$  to denote a limit point of a sequence, not necessarily the solution to (2.2). Dependencies among the lemmas, corollary, propositions, and theorem for the global convergence proof are presented in Figure B.1.

**Lemma B.1.** *(Corresponds to Lemma 1 of [TAW06]) For  $\mathbf{s}, \boldsymbol{\lambda} \geq \mathbf{0}$ ,  $\mathbf{J}(\mathbf{A}, \mathbf{s}, \boldsymbol{\lambda})$  is nonsingular if and only if*

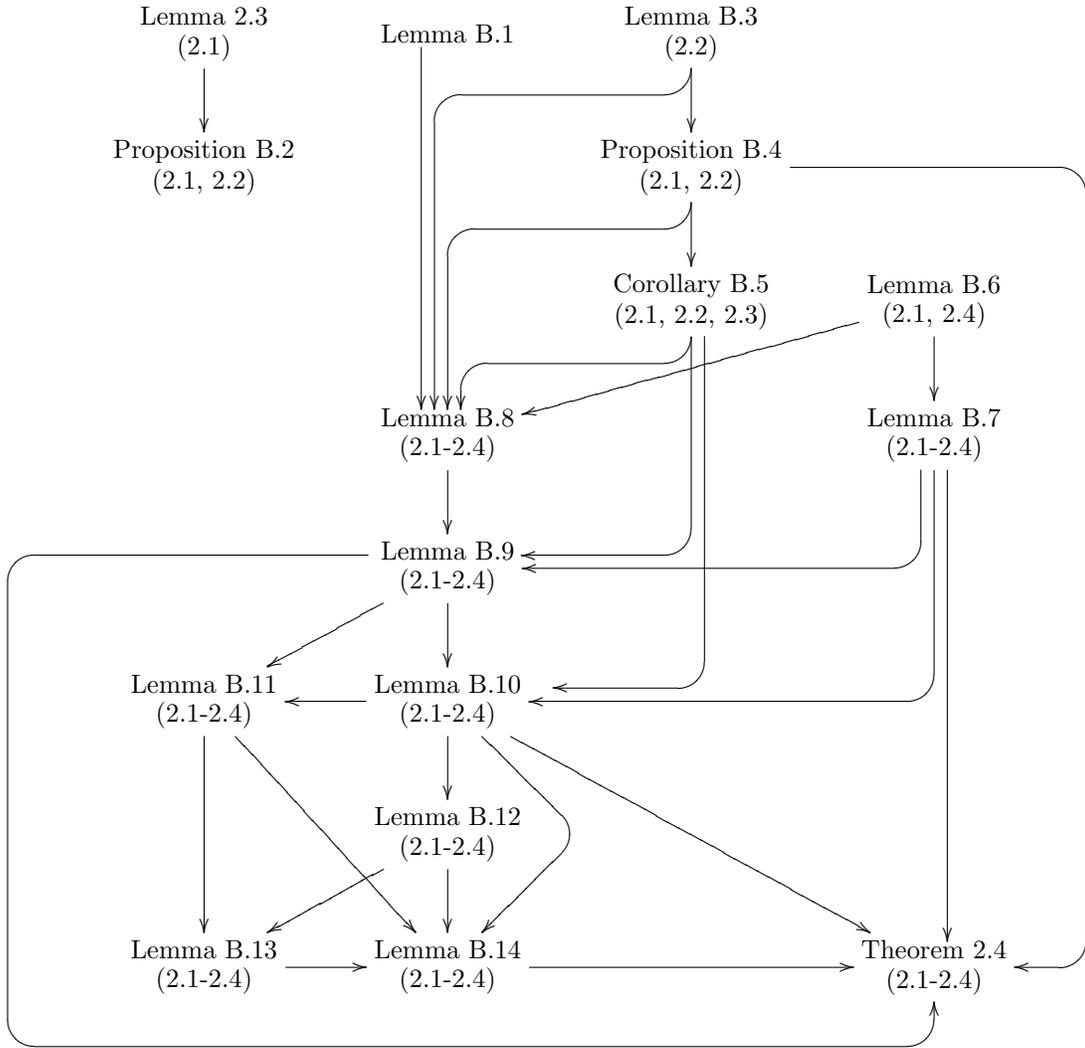


Figure B.1: Diagram for global convergence proof. Numbers in parentheses denote assumptions which each lemma, proposition, corollary, and theorem requires. Because Proposition B.2 guarantees the Algorithm 1 to generate an infinite sequence, it supports all other Lemmas, Propositions, Corollaries and Theorems. Since drawing all the outgoing arrows from Proposition B.2 may complicate this figure, we omit them.

(i)  $\forall i \in M : s_i + \lambda_i > 0$ ,

(ii) Rows of  $\mathbf{A}_{\{i:s_i=0\}}$  are linearly independent,

(iii)  $\mathbf{A}_{\{i:\lambda_i \neq 0\}}$  and  $\mathbf{H}$  share a trivial nullspace  $\{\mathbf{0}\}$ , i.e.,

$$\{\mathbf{x} : \mathbf{A}_{\{i:\lambda_i \neq 0\}}\mathbf{x} = \mathbf{0}\} \cap \{\mathbf{x} : \mathbf{H}\mathbf{x} = \mathbf{0}\} = \{\mathbf{0}\}. \quad (\text{B.1})$$

*Proof.* Suppose the three conditions hold. Let  $(\mathbf{u}^T, \mathbf{v}^T, \mathbf{w}^T)^T$  be in the nullspace of  $\mathbf{J}(\mathbf{A}, \mathbf{s}, \boldsymbol{\lambda})$ . Then it holds that

$$\mathbf{H}\mathbf{u} - \mathbf{A}^T\mathbf{w} = \mathbf{0}, \quad (\text{B.2})$$

$$\mathbf{A}\mathbf{u} - \mathbf{v} = \mathbf{0}, \quad (\text{B.3})$$

$$\boldsymbol{\Lambda}\mathbf{v} + \mathbf{S}\mathbf{w} = \mathbf{0}. \quad (\text{B.4})$$

Multiplying  $\mathbf{u}^T$  to (B.2) yields

$$\mathbf{u}^T\mathbf{H}\mathbf{u} - \mathbf{u}^T\mathbf{A}^T\mathbf{w} = 0, \quad (\text{B.5})$$

which, by (B.3) and the positive semidefiniteness of  $\mathbf{H}$ , yields

$$\mathbf{v}^T\mathbf{w} = \mathbf{u}^T\mathbf{H}\mathbf{u} \geq 0. \quad (\text{B.6})$$

By (B.4), for all  $i$  with  $\lambda_i > 0$ ,

$$v_i = -\frac{s_i}{\lambda_i}w_i. \quad (\text{B.7})$$

By (i), for all  $i$  with  $\lambda_i = 0$ , we have  $s_i > 0$ . By (B.4), this leads to  $s_i w_i = 0$  so

$$w_i = 0, \text{ for } i : \lambda_i = 0. \quad (\text{B.8})$$

Thus, from (B.6), (B.7), (B.8), and non-negativity of  $\mathbf{s}$  and  $\boldsymbol{\lambda}$ , it follows that

$$0 \leq \mathbf{u}^T \mathbf{H} \mathbf{u} = \mathbf{v}^T \mathbf{w} = \sum_{i:\lambda_i \neq 0} v_i w_i + \sum_{i:\lambda_i = 0} v_i w_i = \sum_{i:\lambda_i \neq 0} v_i w_i = - \sum_{i:\lambda_i \neq 0} \frac{s_i}{\lambda_i} w_i^2 \leq 0, \quad (\text{B.9})$$

so the all expressions in (B.9) vanish. In particular, again using the positive semidefiniteness of  $\mathbf{H}$ ,

$$\mathbf{H} \mathbf{u} = \mathbf{0}, \quad (\text{B.10})$$

and, using again (B.6) and nonnegativity of  $\mathbf{s}$  and  $\boldsymbol{\lambda}$ ,

$$v_i = -\frac{s_i w_i}{\lambda_i} = 0 \text{ for all } i \text{ such that } \lambda_i > 0. \quad (\text{B.11})$$

Together with (B.2), (B.10) leads to

$$\mathbf{A}^T \mathbf{w} = \mathbf{0}, \quad (\text{B.12})$$

and, by (B.11), it follows that

$$\boldsymbol{\Lambda} \mathbf{v} = \mathbf{0}, \quad (\text{B.13})$$

from which it immediately follows, by (B.4), that

$$\mathbf{S} \mathbf{w} = \mathbf{0}. \quad (\text{B.14})$$

Now (B.14) yields

$$w_i = 0 \text{ for all } i \text{ such that } s_i \neq 0, \quad (\text{B.15})$$

which, together with (B.12), yields

$$\mathbf{A}^T \mathbf{w} = \sum_{\{i:s_i \neq 0\}} w_i \mathbf{a}_i + \sum_{\{i:s_i=0\}} w_i \mathbf{a}_i = \sum_{\{i:s_i=0\}} w_i \mathbf{a}_i = \mathbf{0}. \quad (\text{B.16})$$

Since the rows of  $\mathbf{A}_{\{i:s_i=0\}}$  are assumed to be linearly independent, (B.15) and (B.16) yield  $\mathbf{w} = \mathbf{0}$ . Now (B.3) and (B.13) yield

$$\Lambda \mathbf{A} \mathbf{u} = \mathbf{0}. \quad (\text{B.17})$$

It then naturally follows that

$$\Lambda_{\{i:\lambda_i \neq 0\}} \mathbf{A}_{\{i:\lambda_i \neq 0\}} \mathbf{u} = \mathbf{0}, \quad (\text{B.18})$$

$$\mathbf{A}_{\{i:\lambda_i \neq 0\}} \mathbf{u} = \mathbf{0}. \quad (\text{B.19})$$

Since the intersection of the nullspace of  $\mathbf{A}_{\{i:\lambda_i \neq 0\}}$  and the nullspace of  $\mathbf{H}$  is  $\{\mathbf{0}\}$ ,  $\mathbf{u} = \mathbf{0}$  by (B.10) and (B.19). Now it's a direct consequence of (B.3) that  $\mathbf{v} = \mathbf{0}$ . So the zero vector is the only solution to (B.2)-(B.4), which implies that  $\mathbf{J}(\mathbf{A}, \mathbf{s}, \boldsymbol{\lambda})$  is nonsingular.

Now let us prove necessity. Inspecting the last  $m$  rows of  $\mathbf{J}(\mathbf{A}, \mathbf{s}, \boldsymbol{\lambda})$  shows that the first condition is necessary in order for  $\mathbf{J}(\mathbf{A}, \mathbf{s}, \boldsymbol{\lambda})$  to be nonsingular. Inspecting the last  $m$  columns shows that the second condition is necessary. Suppose the last condition doesn't hold. Then there must exist a nonzero vector  $\mathbf{u}$  which satisfies

both  $\mathbf{A}_{\{i:\lambda_i \neq 0\}} \mathbf{u} = \mathbf{0}$  and  $\mathbf{H}\mathbf{u} = \mathbf{0}$ . Now let  $\mathbf{w} := \mathbf{0}$  and  $\mathbf{v} := \mathbf{A}\mathbf{u}$ , then  $\mathbf{v}_{\{i:\lambda_i \neq 0\}} = \mathbf{A}_{\{i:\lambda_i \neq 0\}} \mathbf{u} = \mathbf{0}$  by the construction of  $\mathbf{u}$ , from which it immediately follows that  $\mathbf{A}\mathbf{v} = \mathbf{0}$ . Therefore (B.2)-(B.4) are satisfied with this configuration. Since  $\mathbf{u} \neq \mathbf{0}$ , this implies that  $\mathbf{J}(\mathbf{A}, \mathbf{s}, \boldsymbol{\lambda})$  is singular and, thus, a contradiction.  $\square$

For the following propositions, lemmas, corollaries, and theorems, unless explicitly stated, it is assumed that Assumptions 2.1, 2.2, 2.3, and 2.4 hold, although some of the earlier results do not require all of them.

**Proposition B.2.** *(Corresponds to Proposition 3 of [TAW06]) The points generated by the iteration of Algorithm 1 satisfy:*

- (i)  $\Delta \mathbf{x}^k \neq \mathbf{0}$  iff  $\mathbf{H}\mathbf{x}^k + \mathbf{c} \neq \mathbf{0}$ ,
- (ii)  $\alpha^k > 0$ ,
- (iii)  $\mathbf{s}^{k+1} = \mathbf{A}\mathbf{x}^{k+1} - \mathbf{b} > \mathbf{0}$  and  $\mathbf{x}^{k+1} \in \mathcal{F}_P^o$ ,
- (iv)  $\boldsymbol{\lambda}^{k+1} > \mathbf{0}$ .

*Proof.* The first claim is a direct consequence of Lemma 2.3 and (2.39). The second and the third claims are true due to (2.34), (2.48), (2.49), and (2.50). The fourth is true due to (2.51); specifically,  $\|\Delta \mathbf{x}^k\|^2 + \|\tilde{\boldsymbol{\lambda}}_-^k\|^2 > 0$ ,  $\underline{\lambda} > 0$ ,  $\lambda_{\max} > 0$ , and  $\tilde{\lambda}_i^k$  is taken only when  $0 < \min\{\|\Delta \mathbf{x}^k\|^2 + \|\tilde{\boldsymbol{\lambda}}_-^k\|^2, \underline{\lambda}\} \leq \tilde{\lambda}_i^k \leq \lambda_{\max}$ .  $\square$

From the initial point which is strictly primal feasible with  $\boldsymbol{\lambda}$  positive, the algorithm generates the next point which is also strictly primal feasible with  $\boldsymbol{\lambda}$  positive. Hence the iteration can be repeated. In other words, the sequence generated by Algorithm 1 is well defined and valid so that the interior-point method can generate infinite sequence of points unless a point  $\mathbf{x}^k$  such that  $\mathbf{H}\mathbf{x}^k + \mathbf{c} = \mathbf{0}$  is found, in which case  $\mathbf{x}^k$  is the solution of the unconstrained problem and also in  $\mathcal{F}_p^o$ . In the sequel it is assumed that Algorithm 1 generates an infinite sequence of primal-dual points.

**Lemma B.3.** *For any  $\mathbf{x} \in \mathcal{F}_p^o$ ,  $q \geq n$ ,  $Q \in \mathcal{Q}(\mathbf{A}\mathbf{x} - \mathbf{b}, q)$  and for every  $\tilde{\boldsymbol{\lambda}}$  and  $\Delta\mathbf{x}$  generated by Algorithm 1,  $\tilde{\boldsymbol{\lambda}}_Q^T \mathbf{A}_Q \Delta\mathbf{x} \leq 0$ , where the equality holds only when  $\tilde{\boldsymbol{\lambda}}_Q^T = \mathbf{0}$  and  $\mathbf{A}_Q \Delta\mathbf{x} = \mathbf{0}$ .*

*Proof.* From the definition of  $\tilde{\boldsymbol{\lambda}}$  and (2.34), we know that

$$\tilde{\boldsymbol{\lambda}}_Q^T \mathbf{A}_Q \Delta\mathbf{x} = \tilde{\boldsymbol{\lambda}}_Q^T \Delta\mathbf{s}_Q$$

From (2.35), the above equation continues to

$$\tilde{\boldsymbol{\lambda}}_Q^T \Delta\mathbf{s}_Q = -\tilde{\boldsymbol{\lambda}}_Q^T \mathbf{S}_{Q^2} \boldsymbol{\Lambda}_{Q^2}^{-1} (\boldsymbol{\lambda}_Q + \Delta\boldsymbol{\lambda}_Q) = -\tilde{\boldsymbol{\lambda}}_Q^T \mathbf{S}_{Q^2} \boldsymbol{\Lambda}_{Q^2}^{-1} \tilde{\boldsymbol{\lambda}}_Q \leq 0.$$

since  $\mathbf{S}_{Q^2}$  and  $\boldsymbol{\Lambda}_{Q^2}$  are diagonal and positive definite. Since  $\boldsymbol{\Lambda}_{Q^2}$  and  $\mathbf{S}_{Q^2}$  are non-singular, the equality holds only if  $\tilde{\boldsymbol{\lambda}}_Q = \mathbf{0}$ . In view of (2.35),  $\Delta\mathbf{s}_Q = \mathbf{0}$  if and only if  $\tilde{\boldsymbol{\lambda}}_Q = \mathbf{0}$ . Therefore, from (2.34), we conclude that  $\mathbf{A}_Q \Delta\mathbf{x} = \mathbf{0}$  if and only if  $\tilde{\boldsymbol{\lambda}}_Q = \mathbf{0}$ . □

As a first step in the proof of global convergence, we show that the objective function decreases monotonically on the sequence of points generated by Algorithm 1.

**Proposition B.4.** (Corresponds to Lemma 4 of [TAW06]) If  $\Delta \mathbf{x} \neq \mathbf{0}$ , then

(i)  $f(\mathbf{x} + \alpha \Delta \mathbf{x}) < f(\mathbf{x})$  for all  $\alpha \in (0, 2)$ ,

(ii)  $\frac{d}{d\alpha} f(\mathbf{x} + \alpha \Delta \mathbf{x}) < 0$  for all  $0 \leq \alpha < 1$ ,

(iii)  $f(\mathbf{x} + \alpha \Delta \mathbf{x}) < f(\mathbf{x} + \underline{\alpha} \Delta \mathbf{x})$  for all  $\alpha$  and  $\underline{\alpha}$  such that  $0 \leq \underline{\alpha} < \alpha \leq 1$ .

*Proof.* Since  $f$  is quadratic,  $f(\mathbf{x} + \alpha \Delta \mathbf{x})$  can be exactly expressed by the 2<sup>nd</sup> order Taylor expansion

$$\begin{aligned} f(\mathbf{x} + \alpha \Delta \mathbf{x}) &= f(\mathbf{x}) + \alpha \nabla f(\mathbf{x})^T \Delta \mathbf{x} + \frac{1}{2} \alpha^2 \Delta \mathbf{x}^T \mathbf{H} \Delta \mathbf{x} \\ &= f(\mathbf{x}) + \alpha \Delta \mathbf{x}^T (\mathbf{H} \mathbf{x} + \mathbf{c}) + \frac{1}{2} \alpha^2 \Delta \mathbf{x}^T \mathbf{H} \Delta \mathbf{x} \\ &= f(\mathbf{x}) + \alpha \Delta \mathbf{x}^T \left( -\mathbf{H} \Delta \mathbf{x} + \mathbf{A}_Q^T \tilde{\boldsymbol{\lambda}}_Q \right) + \frac{1}{2} \alpha^2 \Delta \mathbf{x}^T \mathbf{H} \Delta \mathbf{x} \quad (\text{by (2.36)}) \\ &= f(\mathbf{x}) - \alpha \left( 1 - \frac{1}{2} \alpha \right) \Delta \mathbf{x}^T \mathbf{H} \Delta \mathbf{x} + \alpha \Delta \mathbf{x}^T \mathbf{A}_Q^T \tilde{\boldsymbol{\lambda}}_Q. \end{aligned} \tag{B.20}$$

By Lemma B.3,  $\Delta \mathbf{x}^T \mathbf{A}_Q^T \tilde{\boldsymbol{\lambda}}_Q$  is nonpositive and, since  $\mathbf{H}$  is positive semidefinite, so is  $-\Delta \mathbf{x}^T \mathbf{H} \Delta \mathbf{x}$ . By Assumption 2.1 and Lemma B.3, they cannot be zero at the same time unless  $\Delta \mathbf{x} = \mathbf{0}$ . Since  $\alpha$  and  $1 - \frac{\alpha}{2}$  are both positive when  $\alpha \in (0, 2)$ , the first claim holds.

Now let us consider the second claim. From (B.20) we derive

$$\frac{d}{d\alpha} f(\mathbf{x} + \alpha \Delta \mathbf{x}) = -(1 - \alpha) \Delta \mathbf{x}^T \mathbf{H} \Delta \mathbf{x} + \tilde{\boldsymbol{\lambda}}_Q^T \mathbf{A}_Q \Delta \mathbf{x}. \tag{B.21}$$

Since  $-(1 - \alpha) < 0$  for all  $0 \leq \alpha < 1$ , the claim does hold.

Let us consider the third claim. Since  $\frac{d}{d\alpha} f(\mathbf{x} + \alpha \Delta \mathbf{x}) < 0$  for  $0 \leq \alpha < 1$ ,  $f(\mathbf{x} + \alpha \Delta \mathbf{x})$  strictly decreases with respect to  $\alpha \in [0, 1]$ . Then the claim immediately follows.  $\square$

**Corollary B.5.** *(Corresponds to Corollary of Proposition 3.1 [TZ94]) The sequence  $\{\mathbf{x}^k\}$  is bounded.*

*Proof.* Since, in view of Assumption 2.3,  $\mathcal{F}_P^*$  is bounded, the level set  $\{\mathbf{x} \in \mathcal{F}_P : f(\mathbf{x}) < f(\mathbf{x}^0)\}$  is bounded (Lemma A.10). As  $f(\mathbf{x}^k)$  decreases monotonically by Proposition B.4, the claim holds.  $\square$

A point  $\mathbf{x}$  is said to be *stationary* for (2.2) if it satisfies the KKT conditions without nonnegativity constraints on  $\boldsymbol{\lambda}$ , i.e.,

$$\mathbf{H}\mathbf{x} + \mathbf{c} - \mathbf{A}^T\boldsymbol{\lambda} = \mathbf{0}, \quad (\text{B.22})$$

$$\mathbf{A}\mathbf{x} - \mathbf{b} - \mathbf{s} = \mathbf{0}, \quad (\text{B.23})$$

$$\mathbf{S}\boldsymbol{\lambda} = \mathbf{0}, \quad (\text{B.24})$$

$$\mathbf{s} \geq \mathbf{0}. \quad (\text{B.25})$$

A stationary point  $\mathbf{x}$  is a solution to (2.2) if all the components of its associated multiplier vector  $\boldsymbol{\lambda}$  are nonnegative.

We proceed by showing that the sequence generated by Algorithm 1 approaches the set of stationary points. In the following lemma, it will be shown that, if the sequence converges to some point, the limit point is stationary and the sequence of modified Newton steps  $\{\Delta\mathbf{x}^k\}$  converges to  $\mathbf{0}$ .

**Lemma B.6.** *(Corresponds to Lemma 3.5 of [WNT007]) Suppose that  $\{\mathbf{x}^k\} \rightarrow \mathbf{x}^*$  on an infinite index set  $K$  and  $q^k \geq n$ . Then there exists  $k'$  such that  $\mathcal{A}(\mathbf{x}^*) \subseteq Q^k$  for all  $Q^k \in \mathcal{Q}(\mathbf{A}\mathbf{x}^k - \mathbf{b}, q^k)$  for all  $k \in K$  and  $k > k'$ .*

*Proof.* Under Assumption 2.4, it follows that  $|\mathcal{A}(\mathbf{x}^*)| \leq n$ . Since  $q^k \geq n$  and  $\{\mathbf{x}^k\} \rightarrow \mathbf{x}^*$  on  $K$ , the claim does hold by the definition of  $\mathcal{Q}(\mathbf{A}\mathbf{x}^k - \mathbf{b}, q^k)$  (2.41).  $\square$

In *Step 2* of Algorithm 1, by replacing  $\Delta \mathbf{s}$  in (2.35) with (2.34), we can rewrite (2.35) as

$$\tilde{\boldsymbol{\lambda}}^k = -(\mathbf{S}^k)^{-1} \boldsymbol{\Lambda}^k \mathbf{A} \Delta \mathbf{x}^k, \quad (\text{B.26})$$

or equivalently

$$\tilde{\lambda}_i^k = -\frac{\lambda_i^k}{s_i^k} \mathbf{a}_i^T \Delta \mathbf{x}^k. \quad (\text{B.27})$$

We use this modified form in the following lemmas.

**Lemma B.7.** *(Corresponds to Lemma 6 of [TAW06]) Suppose  $\{\mathbf{x}^k\}$  converges to some point  $\mathbf{x}^*$  on an infinite index set  $K$ . If  $\{\Delta \mathbf{x}^k\}$  converges to zero on  $K$ , then  $\mathbf{x}^*$  is stationary and  $\{\tilde{\boldsymbol{\lambda}}^k\}$  converges on  $K$  to  $\boldsymbol{\lambda}^*$ , which is the unique multiplier associated with  $\mathbf{x}^*$ .*

*Proof.* Suppose  $\{\Delta \mathbf{x}^k\} \rightarrow \mathbf{0}$  as  $k \rightarrow \infty, k \in K$ . Since  $\{\boldsymbol{\lambda}^k\}$  is bounded by construction of Algorithm 1 and  $\{s_i^k\}$  is bounded away from 0 for  $i \notin \mathcal{A}(\mathbf{x}^*)$ , it follows from (B.27) that

$$\forall i \in \mathcal{A}(\mathbf{x}^*)^c, \quad \{\tilde{\lambda}_i^k\} \rightarrow 0, \text{ as } k \rightarrow \infty, k \in K. \quad (\text{B.28})$$

We have shown the convergence of  $\boldsymbol{\lambda}_{\mathcal{A}(\mathbf{x}^*)^c}^k$  on  $K$  so far.

Now we need to show the convergence of  $\boldsymbol{\lambda}_{\mathcal{A}(\mathbf{x}^*)}$ . At iteration  $k$ , the system of equations (2.36) can be written as

$$\mathbf{H}\mathbf{x}^k + \mathbf{c} - \mathbf{A}_{Q^k}^T \tilde{\boldsymbol{\lambda}}_{Q^k}^k = -\mathbf{H}\Delta\mathbf{x}^k. \quad (\text{B.29})$$

By Lemma B.6, there exists  $k'$  such that  $\mathcal{A}(\mathbf{x}^*) \subseteq Q^k$  for  $k > k'$  and  $k \in K$ . Then, since  $\{\mathbf{x}^k\}$  converges to  $\mathbf{x}^*$  on  $K$  and  $\{\Delta\mathbf{x}^k\}$  converges to zero on  $K$  (by assumption), the equations (B.28) and (B.29) yield

$$\mathbf{H}\mathbf{x}^k + \mathbf{c} - \mathbf{A}_{\mathcal{A}(\mathbf{x}^*)}^T \tilde{\boldsymbol{\lambda}}_{\mathcal{A}(\mathbf{x}^*)}^k \rightarrow 0 \text{ as } k \rightarrow \infty, k \in K. \quad (\text{B.30})$$

Since the rows of  $\mathbf{A}_{\mathcal{A}(\mathbf{x}^*)}$  are linearly independent by Assumption 2.4, in view of (B.28), there exists a unique  $\boldsymbol{\lambda}^*$  to which  $\{\tilde{\boldsymbol{\lambda}}^k\}$  converges on  $K$ . By taking limits in (B.26), (B.28), and (B.30) and by using boundedness of  $\{\boldsymbol{\lambda}^k\}$  due to construction of Algorithm 1, it follows that

$$\mathbf{H}\mathbf{x}^* + \mathbf{c} - \mathbf{A}^T \boldsymbol{\lambda}^* = \mathbf{0}, \quad (\text{B.31})$$

$$\mathbf{S}^* \boldsymbol{\lambda}^* = \mathbf{0}. \quad (\text{B.32})$$

This implies that  $\mathbf{x}^*$  is stationary with the unique associated multiplier vector  $\boldsymbol{\lambda}^*$ .  $\square$

**Lemma B.8.** *(Corresponds to Lemma 7 of [TAW06]) Let  $K$  be an infinite index set such that*

$$\inf\{\|\Delta\mathbf{x}^{k-1}\|^2 + \|\tilde{\boldsymbol{\lambda}}_-^{k-1}\|^2 : k \in K\} > 0.$$

*Then  $\{\Delta\mathbf{x}^k\} \rightarrow \mathbf{0}$  as  $k \rightarrow \infty, k \in K$ .*

*Proof.* By contradiction. Suppose not. First, by (2.51) and by the condition that  $\inf\{\|\Delta\mathbf{x}^{k-1}\|^2 + \|\tilde{\boldsymbol{\lambda}}_-^{k-1}\|^2 : k \in K\}$  is greater than 0 (by the assumption on  $K$ ),  $\lambda_i^k$  ( $i \in M$ ) is bounded away from zero on  $K$ . Since  $\{\mathbf{x}^k\}$  (by Corollary B.5) and  $\{\boldsymbol{\lambda}^k\}$  (by construction) are bounded, we may conclude that there is a convergent subsequence  $\{\mathbf{x}^k\}$  and  $\{\boldsymbol{\lambda}^k\}$ . So there exists some infinite index set  $K' \subseteq K$ , a point  $(\mathbf{x}^*, \boldsymbol{\lambda}^*)$ , and an index set  $Q^*$  such that

$$\begin{aligned} \inf_{k \in K'} \|\Delta\mathbf{x}^k\| &> 0, \\ \{\mathbf{x}^k\} &\rightarrow \mathbf{x}^* \text{ as } k \rightarrow \infty, k \in K', \\ \{\boldsymbol{\lambda}^k\} &\rightarrow \boldsymbol{\lambda}^* > \mathbf{0} \text{ as } k \rightarrow \infty, k \in K', \end{aligned} \tag{B.33}$$

and

$$Q^k = Q^*, \forall k \in K'.$$

By Assumptions 2.1 and 2.4 and the fact that  $\boldsymbol{\lambda}^* > \mathbf{0}$ , Lemma B.1 tells us that  $\mathbf{J}(\mathbf{A}_{Q^*}, \mathbf{s}_{Q^*}^*, \boldsymbol{\lambda}_{Q^*}^*)$  is nonsingular. It then follows from (2.36)-(2.38) and continuity of  $\mathbf{J}(\mathbf{A}_{Q^*}, \mathbf{s}_{Q^*}, \boldsymbol{\lambda}_{Q^*})$  with respect to  $\mathbf{s}_{Q^*}$  and  $\boldsymbol{\lambda}_{Q^*}$  that, for some  $\Delta\mathbf{x}^* \neq \mathbf{0}$  and  $\tilde{\boldsymbol{\lambda}}^*$ ,

$$\{\Delta\mathbf{x}^k\} \rightarrow \Delta\mathbf{x}^*, \text{ as } k \rightarrow \infty, k \in K', \tag{B.34}$$

$$\{\tilde{\boldsymbol{\lambda}}_{Q^*}^k\} \rightarrow \tilde{\boldsymbol{\lambda}}_{Q^*}^*, \text{ as } k \rightarrow \infty, k \in K'. \tag{B.35}$$

Define  $\mathbf{s}^* := \mathbf{A}\mathbf{x}^* - \mathbf{b}$ . Since  $\mathbf{s}^k = \mathbf{A}\mathbf{x}^k - \mathbf{b}$  and  $\{\mathbf{x}^k\} \rightarrow \mathbf{x}^*$  on  $K'$ , we know that  $\{\mathbf{s}^k\} \rightarrow \mathbf{s}^*$  on  $K'$  and  $\mathbf{s}^* \geq \mathbf{0}$  by construction of Algorithm 1. In addition, since  $Q^k = Q^k \in \mathcal{Q}(\mathbf{A}\mathbf{x}^k - \mathbf{b}, q^k)$  for all  $k \in K'$  with  $q^k \geq n$ , it follows from Lemma B.6 that  $\mathcal{A}(\mathbf{x}^*) \subseteq Q^*$ . Therefore  $s_i^k$  is bounded away from zero for  $k \in K'$ ,  $i \notin Q^*$ .

Therefore, since  $\{\Delta \mathbf{x}^k\} \rightarrow \Delta \mathbf{x}^*$  as  $k \rightarrow \infty$  and since  $s_i^k$  is bounded away from zero for  $i \notin Q^*$  on  $K'$ , we know from (B.27) that

$$\forall i \notin Q^*, \quad \{\tilde{\lambda}_i^k\} \rightarrow \tilde{\lambda}_i^* \text{ as } k \rightarrow \infty, k \in K'. \quad (\text{B.36})$$

It then immediately follows from (B.35) and (B.36) that

$$\{\tilde{\boldsymbol{\lambda}}^k\} \rightarrow \tilde{\boldsymbol{\lambda}}^* \text{ as } k \rightarrow \infty, k \in K', \quad (\text{B.37})$$

implying  $\{\tilde{\boldsymbol{\lambda}}^k\}$  is bounded on  $K'$ .

Up to this point we have shown that  $\{\Delta \mathbf{x}^k\} \rightarrow \Delta \mathbf{x}^*$  and  $\{\tilde{\boldsymbol{\lambda}}^k\} \rightarrow \tilde{\boldsymbol{\lambda}}^*$  as  $k \rightarrow \infty$  on  $K'$ . With these facts, we will show that  $f(\mathbf{x}^k) \rightarrow \infty$  as  $k \rightarrow \infty$  on  $K'$ , contradicting Corollary B.5. For this we will first show that  $\alpha^k$  is bounded below on  $K'$ . Indeed, using (B.27), we can restate (2.47) as

$$\bar{\alpha}^k = \begin{cases} \infty & \text{if } \tilde{\boldsymbol{\lambda}} \leq \mathbf{0}, \\ \min_i \left\{ -\frac{s_i^k}{\Delta s_i^k} = \frac{\lambda_i^k}{\tilde{\lambda}_i^k} \mid \text{s.t. } \tilde{\lambda}_i > 0, i \in M \right\} & \text{otherwise.} \end{cases} \quad (\text{B.38})$$

Since  $\{\tilde{\boldsymbol{\lambda}}^k\}$  is bounded, and each  $\{\lambda_i^k\}$  is bounded away from zero for  $k \in K'$ , it follows that  $\bar{\alpha}^k$  is bounded away from zero on  $K'$  and, since  $\alpha^k \geq \eta \bar{\alpha}^k$  by (2.48), so is  $\alpha^k$ . That is, there exists  $\underline{\alpha} > 0$  such that  $\alpha^k > \underline{\alpha} \forall k \in K'$ .

We now combine the lower bound on  $\alpha$  with the monotonicity of the objective function we showed in Proposition B.4. From the second and third claim of Proposition B.4 (and the expansion of  $f(\mathbf{x}^k + \underline{\alpha} \Delta \mathbf{x}^k)$  similar to (B.20)) it immediately follows,

since  $\alpha^k > \underline{\alpha} > 0$  and  $\alpha^k \leq 1$  on  $K'$ , that

$$\begin{aligned} f(\mathbf{x}^k + \alpha^k \Delta \mathbf{x}^k) &< f(\mathbf{x}^k + \underline{\alpha} \Delta \mathbf{x}^k) \\ &= f(\mathbf{x}^k) - \underline{\alpha} \left(1 - \frac{1}{2} \underline{\alpha}\right) \Delta \mathbf{x}^{kT} \mathbf{H} \Delta \mathbf{x}^k + \underline{\alpha} \tilde{\boldsymbol{\lambda}}_{Q^k}^{*T} \mathbf{A}_{Q^k} \Delta \mathbf{x}^k. \end{aligned} \quad (\text{B.39})$$

Our focus from now is showing that  $\underline{\alpha} \left(1 - \frac{1}{2} \underline{\alpha}\right) \Delta \mathbf{x}^{kT} \mathbf{H} \Delta \mathbf{x}^k - \underline{\alpha} \tilde{\boldsymbol{\lambda}}_{Q^k}^{*T} \mathbf{A}_{Q^k} \Delta \mathbf{x}^k$  is bounded below on  $K'$ , which immediately implies that  $f(\mathbf{x}^k) \rightarrow -\infty$  on  $K'$ , since by Proposition B.4  $\{f(\mathbf{x}^k)\}$  is monotonically decreasing. Taking limits in (2.37)-(2.38) on  $K'$  yields

$$\mathbf{S}_{Q^*}^* \tilde{\boldsymbol{\lambda}}_{Q^*}^* = -\boldsymbol{\Lambda}_{Q^*}^* \mathbf{A}_{Q^*} \Delta \mathbf{x}^* \text{ as } k \rightarrow \infty, k \in K'. \quad (\text{B.40})$$

So, since  $\boldsymbol{\lambda}^* > \mathbf{0}$  by (B.33) and  $\mathbf{s}^* \geq \mathbf{0}$ , we know that  $\mathbf{S}_{Q^*}^* \tilde{\boldsymbol{\lambda}}_{Q^*}^* \geq \mathbf{0}$  and  $\mathbf{A}_{Q^*} \Delta \mathbf{x}^* \leq \mathbf{0}$ .

So, for  $i \in Q^*$ , it follows that  $\mathbf{a}_i^T \Delta \mathbf{x}^* < 0$  if  $\mathbf{a}_i^T \Delta \mathbf{x}^* \neq 0$ . Thus we know that

$$\tilde{\boldsymbol{\lambda}}_{Q^*}^{*T} \mathbf{A}_{Q^*} \Delta \mathbf{x}^* = \sum_{i \in Q^*} \tilde{\lambda}_i^* \mathbf{a}_i^T \Delta \mathbf{x}^* < 0 \text{ if } \mathbf{A}_{Q^*} \Delta \mathbf{x}^* \neq \mathbf{0}. \text{ From this, since } \mathcal{N}(\mathbf{A}_{Q^*}) \cap$$

$\mathcal{N}(\mathbf{H}) = \{\mathbf{0}\}$  under Assumption 2.1 and  $\mathbf{H}$  is positive semidefinite, we conclude

that  $\Delta \mathbf{x}^{*T} \mathbf{H} \Delta \mathbf{x}^* > 0$  or  $-\tilde{\boldsymbol{\lambda}}_{Q^*}^{*T} \mathbf{A}_{Q^*} \Delta \mathbf{x}^* > 0$ , similarly to Lemma B.3. So, since

$$\Delta \mathbf{x}^{kT} \mathbf{H} \Delta \mathbf{x}^k \rightarrow \Delta \mathbf{x}^{*T} \mathbf{H} \Delta \mathbf{x}^* \text{ and } -\tilde{\boldsymbol{\lambda}}_{Q^k}^{*T} \mathbf{A}_{Q^k} \Delta \mathbf{x}^k \rightarrow -\tilde{\boldsymbol{\lambda}}_{Q^*}^{*T} \mathbf{A}_{Q^*} \Delta \mathbf{x}^* \text{ as } k \rightarrow \infty \text{ on } K'$$

and  $0 < \underline{\alpha} < 1$ , there exists  $\delta > 0$  such that, for large enough  $k \in K'$ ,

$$\underline{\alpha} \left(1 - \frac{1}{2} \underline{\alpha}\right) \Delta \mathbf{x}^{kT} \mathbf{H} \Delta \mathbf{x}^k - \underline{\alpha} \tilde{\boldsymbol{\lambda}}_{Q^k}^{*T} \mathbf{A}_{Q^k} \Delta \mathbf{x}^k > \delta. \quad (\text{B.41})$$

Since  $f(\mathbf{x}^k)$  is monotonic decreasing by Proposition B.4, (B.39) yields that for some large  $k'$

$$f(\mathbf{x}^{k+1}) < f(\mathbf{x}^k) - \delta, \forall k \geq k' \text{ and } k \in K', \quad (\text{B.42})$$

which implies that  $f(\mathbf{x}^k) \rightarrow -\infty$  as  $k \rightarrow \infty$ . This contradicts the boundedness of  $\{\mathbf{x}^k\}$ .  $\square$

**Lemma B.9.** *(Corresponds to Lemma 8 of [TAW06]) Suppose  $\{\mathbf{x}^k\}$  is bounded away from  $\mathcal{F}_P^*$  on some infinite index set  $K$ . Then  $\{\Delta\mathbf{x}^k\}$  goes to  $\mathbf{0}$  on  $K$ .*

*Proof.* By contradiction. Suppose  $\{\Delta\mathbf{x}^k\}$  does not converge to zero as  $k \rightarrow \infty$ ,  $k \in K$ .

By Lemma B.8, there exists an infinite index set  $K' \subseteq K$  such that

$$\Delta\mathbf{x}^{k-1} \rightarrow \mathbf{0}, \text{ as } k \rightarrow \infty, k \in K', \quad (\text{B.43})$$

$$\tilde{\lambda}_-^{k-1} \rightarrow \mathbf{0}, \text{ as } k \rightarrow \infty, k \in K'. \quad (\text{B.44})$$

Since, in view of Corollary B.5,  $\{\mathbf{x}^k\}$  is bounded and, by the assumption, bounded away from  $\mathcal{F}_P^*$  on  $K'$ , without loss of generality, we can assume that  $\{\mathbf{x}^k\} \rightarrow \mathbf{x}^*$  for some  $\mathbf{x}^* \notin \mathcal{F}_P^*$  as  $k \rightarrow \infty$ ,  $k \in K'$ . Since  $\|\mathbf{x}^k - \mathbf{x}^{k-1}\| = \|\alpha^{k-1}\Delta\mathbf{x}^{k-1}\| \leq \|\Delta\mathbf{x}^{k-1}\|$ , it follows that  $\{\mathbf{x}^{k-1}\} \rightarrow \mathbf{x}^*$  as  $k \rightarrow \infty$ ,  $k \in K'$ . This implies, by Lemma B.7, that  $\mathbf{x}^*$  is stationary and  $\{\tilde{\lambda}^{k-1}\} \rightarrow \boldsymbol{\lambda}^*$  as  $k \rightarrow \infty$ ,  $k \in K'$ , where  $\boldsymbol{\lambda}^*$  is the corresponding multiplier vector. Since  $\tilde{\lambda}_-^{k-1} = \min\{\tilde{\lambda}^{k-1}, \mathbf{0}\}$ , it follows from (B.44) that  $\boldsymbol{\lambda}^* \geq \mathbf{0}$ , thus  $\mathbf{x}^* \in \mathcal{F}_P^*$ , a contradiction.  $\square$

We proceed by showing that  $\{\mathbf{x}^k\}$  approaches the set of stationary points.

**Proposition B.10.** *(Corresponds to Lemma 9 of [TAW06])  $\{\mathbf{x}^k\}$  approaches the set of stationary points of (2.2), i.e., for any  $\epsilon > 0$  there exists  $k'$  so that  $\mathbf{x}^k$  is  $\epsilon$ -close to a stationary point for  $k > k'$ .*

*Proof.* By contradiction. Suppose not. Then, since  $\{\mathbf{x}^k\}$  is bounded in view of Corollary B.5, there exists some infinite index set  $K$  and some non-stationary  $\mathbf{x}^*$  such that  $\mathbf{x}^k \rightarrow \mathbf{x}^*$  as  $k \rightarrow \infty$ ,  $k \in K$ . By Lemma B.7, thus, there exists an infinite index set  $K' \subset K$  such that  $\inf_{k \in K'} \|\Delta \mathbf{x}^k\| > 0$ , and this contradicts Lemma B.9.  $\square$

Now we are about to show that  $\{\mathbf{x}^k\}$  converges to  $\mathcal{F}_P^*$ , the solution set to (2.2). For showing this we define a set of all limit points of  $\{\mathbf{x}^k\}$ :

$$L := \{\mathbf{x} : \mathbf{x} \text{ is a limit point of } \{\mathbf{x}^k\}\}.$$

We will show that all the points in  $L$  are associated with the same multipliers in Lemma B.14. By Lemma B.10,  $L$  is a subset of the set of stationary points implying every  $\mathbf{x} \in L$  is a stationary point of (2.2). Since  $\{\mathbf{x}^k\}$  is bounded, so is  $L$ . Thus, as a limit set, it is closed and, thus, compact. The following lemmas help in proving Lemma B.14.

**Lemma B.11.** *(Corresponds to Lemma A.5 under Lemma 3.6 of [TZ94]) If  $\{\mathbf{x}^k\}$  is bounded away from  $\mathcal{F}_P^*$ , then  $L$  is connected.*

*Proof.* By contradiction. Suppose not. Since  $L$  is compact, there must exist nonempty sets  $D, E \subset \mathbb{R}^n$  such that  $L = D \cup E$ ,  $\bar{D} \cap E = D \cap \bar{E} = \emptyset$ , where  $\bar{D}$  is the closure of  $D$ . Since  $L$  is compact,  $D$  and  $E$  must be compact. Thus  $\delta := \min_{\mathbf{x} \in D, \mathbf{x}' \in E} \|\mathbf{x} - \mathbf{x}'\| > 0$ . By Lemma B.10 the entire sequence  $\{\mathbf{x}^k\}$  converges to  $L$ . A simple contradiction argument using the fact that  $\{\mathbf{x}^k\}$  is bounded shows that, for  $k$  large enough,

$\min_{\mathbf{x} \in L} \|\mathbf{x}^k - \mathbf{x}\| \leq \delta/3$ , i.e., either  $\min_{\mathbf{x} \in D} \|\mathbf{x}^k - \mathbf{x}\| \leq \delta/3$  or  $\min_{\mathbf{x} \in E} \|\mathbf{x}^k - \mathbf{x}\| \leq \delta/3$ . Moreover, since both  $D$  and  $E$  are nonempty (i.e., contain limit points of  $\{\mathbf{x}_k\}$ ), each of these situations occurs infinitely many times. Thus  $K := \{k : \min_{\mathbf{x} \in D} \|\mathbf{x}^k - \mathbf{x}\| \leq \delta/3, \min_{\mathbf{x} \in E} \|\mathbf{x}^{k+1} - \mathbf{x}\| \leq \delta/3\}$  is an infinite index set and  $\|\Delta \mathbf{x}^k\| \geq \delta/3 > 0$  for all  $k \in K$ . On the other hand since  $\{\mathbf{x}^k\}_{k \in K}$  is bounded and bounded away from  $\mathcal{F}_P^*$ , it has some limit point  $\mathbf{x}^* \notin \mathcal{F}_P^*$ . In view of Lemma B.9, this is a contradiction.  $\square$

**Lemma B.12.** *(Corresponds to Lemma A.3 under Lemma 3.6 of [TZ94])* Let  $\mathbf{x}, \mathbf{x}' \in L$  be such that  $\mathcal{A}(\mathbf{x}) = \mathcal{A}(\mathbf{x}')$ . Then  $\mathbf{H}(\mathbf{x} - \mathbf{x}') = \mathbf{0}$ .

*Proof.* Let  $\mathcal{A}^* := \mathcal{A}(\mathbf{x}) = \mathcal{A}(\mathbf{x}')$ . Since both  $\mathbf{x}$  and  $\mathbf{x}'$  are limit points of  $\{\mathbf{x}^k\}$ , they are stationary by Lemma B.10. Thus it follows that

$$\begin{aligned} \mathbf{H}\mathbf{x} + \mathbf{c} - \sum_{i \in \mathcal{A}^*} \lambda_i \mathbf{a}_i &= \mathbf{0}, \\ \mathbf{H}\mathbf{x}' + \mathbf{c} - \sum_{i \in \mathcal{A}^*} \lambda'_i \mathbf{a}_i &= \mathbf{0}, \end{aligned}$$

which implies that, for all  $\alpha \in (0, 1)$ ,

$$\mathbf{H}\mathbf{x}_\alpha + \mathbf{c} - \sum_{i \in \mathcal{A}^*} \lambda'_{\alpha,i} \mathbf{a}_i = \mathbf{0}, \tag{B.45}$$

where  $\mathbf{x}_\alpha := (1 - \alpha)\mathbf{x} + \alpha\mathbf{x}'$ ,  $\boldsymbol{\lambda}_\alpha := (1 - \alpha)\boldsymbol{\lambda} + \alpha\boldsymbol{\lambda}'$ . Now  $\mathbf{a}_i^T(\mathbf{x}' - \mathbf{x}) = \mathbf{0}$  for all  $i \in \mathcal{A}^*$

which, together with (B.45), implies that

$$(\mathbf{x}' - \mathbf{x})^T (\mathbf{H}\mathbf{x}_\alpha + \mathbf{c}) = 0, \forall \alpha \in (0, 1).$$

Since  $\mathbf{x}_\alpha = \mathbf{x} + \alpha(\mathbf{x}' - \mathbf{x})$ , we get, for all  $\alpha \in (0, 1)$ ,

$$0 = (\mathbf{x}' - \mathbf{x})^T (\mathbf{H}\mathbf{x} + \alpha\mathbf{H}(\mathbf{x}' - \mathbf{x}) + \mathbf{c})$$

$$= (\mathbf{x}' - \mathbf{x})^T (\mathbf{H}\mathbf{x} + c) + \alpha (\mathbf{x}' - \mathbf{x})^T \mathbf{H} (\mathbf{x}' - \mathbf{x}).$$

Thus  $(\mathbf{x}' - \mathbf{x})^T \mathbf{H} (\mathbf{x}' - \mathbf{x}) = 0$ . Since  $\mathbf{H}$  is positive semi-definite, the claim follows.  $\square$

**Lemma B.13.** *(Corresponds to Lemma A.4 under Lemma 3.6 of [TZ94]) If  $\{\mathbf{x}^k\}$  is bounded away from  $\mathcal{F}_P^*$ , then for all  $\mathbf{x}, \mathbf{x}' \in L$ ,  $\mathbf{H}(\mathbf{x}' - \mathbf{x}) = 0$ .*

*Proof.* Since there are only finitely many possible combinations of binding constraints, in view of Lemma B.12,  $L$  is a finite union of affine sets of the form  $L \cap (\mathbf{x} + \mathcal{N}(\mathbf{H}))$  with  $\mathbf{x} \in L$ .

Suppose that there are  $N$  such distinct affine sets  $A_i$  which are in the form  $L \cap (\mathbf{x} + \mathcal{N}(\mathbf{H}))$  with  $\mathbf{x} \in L$ . Then  $L = \cup_{i=1}^N A_i$ . Notice each  $A_i$  is a subset of an affine subspace  $\ell_i + \mathcal{N}(\mathbf{H})$  for  $\ell_i \in A_i$ . So, for any distinct  $i, j \in \{1, \dots, N\}$ ,  $A_i$  and  $A_j$  lie on either the same affine subspace ( $\ell_i + \mathcal{N}(\mathbf{H}) = \ell_j + \mathcal{N}(\mathbf{H})$ ) or parallel affine subspaces ( $\ell_i + \mathcal{N}(\mathbf{H}) \neq \ell_j + \mathcal{N}(\mathbf{H})$ ). However, since  $L$  is connected in view of Lemma B.11 and there are finitely many  $A_i$ 's, they can not lie on distinct parallel affine subspaces. Therefore all  $A_i$ 's lie on the same affine subspace which is parallel to  $\mathcal{N}(\mathbf{H})$ . This proves the claim.  $\square$

**Lemma B.14.** *(Corresponds to Lemma 3.6 of [TZ94]) Suppose  $\{\mathbf{x}^k\}$  is bounded away from  $\mathcal{F}_P^*$ . Let  $\mathbf{x}^*, \mathbf{x}'^* \in L$ . Let  $\boldsymbol{\lambda}$  and  $\boldsymbol{\lambda}'$  be the associated multiplier vectors. Then  $\boldsymbol{\lambda} = \boldsymbol{\lambda}'$ .*

*Proof.* In view of Lemma B.10, all points in  $L$  are stationary points of (2.2). Given any  $\mathbf{x} \in L$ , let  $\boldsymbol{\lambda}(\mathbf{x})$  be the multiplier vector associated with  $\mathbf{x}$  and let  $\mathcal{B}(\mathbf{x})$  be the

index set of “binding” constraints at  $\mathbf{x}$ , i.e.,

$$\mathcal{B}(\mathbf{x}) := \{i \in M : \lambda_i(\mathbf{x}) \neq 0\}. \quad (\text{B.46})$$

We first claim that, if  $\mathbf{x}, \mathbf{x}' \in L$  are such that  $\mathcal{B}(\mathbf{x}) = \mathcal{B}(\mathbf{x}')$ , then  $\boldsymbol{\lambda}(\mathbf{x}) = \boldsymbol{\lambda}(\mathbf{x}')$ .

Indeed, in view of (B.22), it follows from Lemma B.11 and Lemma B.13 that

$$\sum_{j \in \mathcal{B}(\mathbf{x})} \lambda_j(\mathbf{x}) \mathbf{a}_j^T = \sum_{j \in \mathcal{B}(\mathbf{x}')} \lambda_j(\mathbf{x}') \mathbf{a}_j^T.$$

Then the claim follows from linear independence of  $\{\mathbf{a}_j^T : j \in \mathcal{A}(\mathbf{x})\}$  under Assumption 2.4 and from the fact that  $\mathcal{B}(\mathbf{x}) \subseteq \mathcal{A}(\mathbf{x})$ .

To conclude the proof, we show that, for any  $\mathbf{x}, \mathbf{x}' \in L$ ,  $\mathcal{B}(\mathbf{x}) = \mathcal{B}(\mathbf{x}')$ . Let  $\bar{\mathbf{x}} \in L$  be arbitrary, and let  $D := \{\mathbf{x} \in L : \mathcal{B}(\mathbf{x}) = \mathcal{B}(\bar{\mathbf{x}})\}$  and  $E := \{\mathbf{x} \in L : \mathcal{B}(\mathbf{x}) \neq \mathcal{B}(\bar{\mathbf{x}})\}$ .

We show that both  $D$  and  $E$  are closed. Let  $\{\mathbf{y}^l\} \subseteq L$  be a sequence converging to some point  $\hat{\mathbf{x}}$  such that  $\mathcal{B}(\mathbf{y}^l) = \mathcal{B}$  for all  $l$  and for some  $\mathcal{B}$ . Notice, since  $L$  is closed,  $\hat{\mathbf{x}} \in L$ . It follows from the first part of this proof that all  $\mathbf{y}^l$  are associated with some common multiplier  $\boldsymbol{\lambda}$ , i.e.,  $\boldsymbol{\lambda}(\mathbf{y}^l) = \boldsymbol{\lambda}$ . Due to the complementarity condition of stationary points, it follows that  $s_j(\mathbf{y}^l) := \mathbf{a}_j^T \mathbf{y}^l - b_j = 0$  for all  $l$  and for all  $j$  such that  $\lambda_j \neq 0$ . Since  $\{\mathbf{y}^l\} \rightarrow \hat{\mathbf{x}}$ , we know that  $s_j(\hat{\mathbf{x}}) := \mathbf{a}_j^T \hat{\mathbf{x}} - b_j = 0$  for all  $j$  such that  $\lambda_j \neq 0$ . This means that  $\mathcal{B} \subseteq \mathcal{A}(\hat{\mathbf{x}})$ . Since  $\mathbf{H}\mathbf{y}^l = \mathbf{H}\mathbf{x}$  for any  $l$  (due to Lemma B.13), we know from (B.22) that

$$\sum_{j \in \mathcal{B}} \lambda_j \mathbf{a}_j^T = \mathbf{H}\mathbf{y}^l + \mathbf{c} = \mathbf{H}\hat{\mathbf{x}} + \mathbf{c} = \sum_{j \in \mathcal{A}(\hat{\mathbf{x}})} \lambda_j(\hat{\mathbf{x}}) \mathbf{a}_j^T, \forall l.$$

Therefore, from linear independence of active constraints under Assumption 2.4, we can conclude that  $\hat{\mathbf{x}}$  is also associated with the common multiplier  $\boldsymbol{\lambda}$ , i.e.,  $\boldsymbol{\lambda}(\hat{\mathbf{x}}) = \boldsymbol{\lambda}$ .

Thus, if  $\{\mathbf{y}^l\} \subseteq D$ , then  $\hat{\mathbf{x}} \in D$ . Likewise, if  $\{\mathbf{y}^l\} \subseteq E$ , then  $\hat{\mathbf{x}} \in E$ . Therefore both  $D$  and  $E$  are closed. Since  $D$  contains at least  $\bar{\mathbf{x}}$ , meaning that it is not empty, connectedness of  $L$  by Lemma B.11 implies that  $E$  is empty (otherwise  $D \cap E \neq \emptyset$ ). Thus all points in  $L$  are associated with the same multiplier.  $\square$

We are now ready to prove that  $\{\mathbf{x}^k\}$  converges to  $\mathcal{F}_P^*$ .

*Proof of Theorem 2.4.* By contradiction. Suppose that some limit point of  $\{\mathbf{x}^k\}$  is not in  $\mathcal{F}_P^*$ . Since  $f(\mathbf{x}^k)$  monotonically decreases in view of Proposition B.4 and since  $\mathbf{x}^k \in \mathcal{F}_P$ ,  $f$  takes on the same value at all limit points of  $\{\mathbf{x}^k\}$ . So  $\{\mathbf{x}^k\}$  is bounded away from  $\mathcal{F}_P^*$ . In view of Lemma B.9,  $\{\Delta \mathbf{x}\} \rightarrow \mathbf{0}$ . Let  $\boldsymbol{\lambda}^*$  be the common multiplier vector associated with all limit points of  $\{\mathbf{x}^k\}$  (see Lemma B.14). Lemma B.7 then implies that  $\{\tilde{\boldsymbol{\lambda}}^k\} \rightarrow \boldsymbol{\lambda}^*$ . Since  $\{\mathbf{x}^k\}$  is bounded away from  $\mathcal{F}_P^*$ , it follows that  $\boldsymbol{\lambda}^* \not\equiv \mathbf{0}$ . Let  $i_0 \in M$  be such that  $\lambda_{i_0}^* < 0$ . So  $\tilde{\lambda}_{i_0}^k < 0$  for all  $k$  large enough. Since  $\Delta s_{i_0}^k = \mathbf{a}_{i_0}^T \Delta \mathbf{x}$  by (2.35), we know from (B.27) that  $\Delta s_{i_0}^k = -(\lambda_{i_0}^k)^{-1} s_{i_0}^k \tilde{\lambda}_{i_0}^k$ . It then follows that  $\Delta s_{i_0}^k > 0$ , since  $\lambda_{i_0}^k > 0$  and  $s_{i_0}^k > 0$  by construction of Algorithm 1. Due to the strict feasibility of  $\mathbf{x}^k$ ,  $\alpha > 0$ . Since  $\mathbf{s}^{k+1} = \mathbf{s}^k + \alpha \Delta \mathbf{s}^k$ , for  $k$  large enough,

$$0 < s_{i_0}(\mathbf{x}^k) < s_{i_0}(\mathbf{x}^{k+1}) < \dots$$

On the other hand, in view of Lemma B.10, since all the limit points are stationary, we know that  $s_{i_0}^* = 0$  for  $i_0$  such that  $\lambda_{i_0}^* < 0$ . Therefore  $\{s_{i_0}^k\}$  converges to zero as  $\mathbf{x}^k$  converges to some point. This is a contradiction.  $\square$

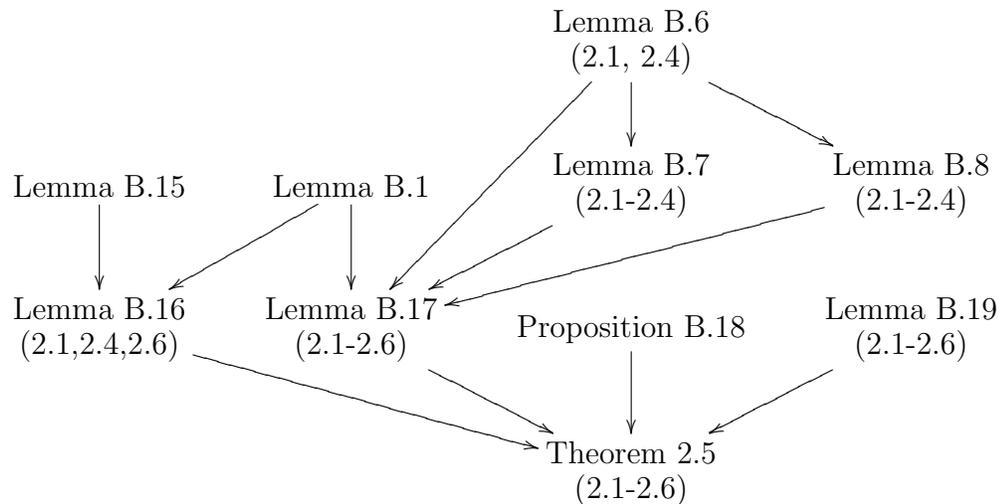


Figure B.2: Diagram for local convergence rate proof. Numbers in parentheses denote assumptions which each lemma, proposition, and theorem requires.

## B.2 Local Rate of Convergence

In this section, we will show that, under Assumptions 2.1-2.6,  $\{\mathbf{x}^k, \boldsymbol{\lambda}^k\}$  converges to the primal-dual solution  $\{\mathbf{x}^*, \boldsymbol{\lambda}^*\}$  q-quadratically. Figure B.2 illustrates the relations among the lemmas, proposition, and theorem in proving local convergence rate. For the following lemmas, propositions and theorems, suppose Assumptions 2.1-2.6 hold.

**Lemma B.15.** *(Corresponds to Lemma 1 of [TAW06])  $\mathbf{J}_a(\mathbf{A}, \mathbf{s}, \boldsymbol{\lambda})$  in (2.32) is non-singular if and only if  $\mathbf{J}(\mathbf{A}, \mathbf{s}, \boldsymbol{\lambda})$  is nonsingular.*

*Proof.* Assume that  $\mathbf{J}_a(\mathbf{A}, \mathbf{s}, \boldsymbol{\lambda})$  is singular. Then there exists a nonzero vector  $[\mathbf{u}^T, \mathbf{w}^T]^T \neq \mathbf{0}$  such that

$$\mathbf{H}\mathbf{u} - \mathbf{A}^T\mathbf{w} = \mathbf{0}, \quad (\text{B.47})$$

$$\Lambda \mathbf{A} \mathbf{u} + \mathbf{S} \mathbf{w} = \mathbf{0}. \quad (\text{B.48})$$

Now let  $\mathbf{v} := \mathbf{A} \mathbf{u}$ , then it immediately follows from (B.48) that

$$\Lambda \mathbf{v} + \mathbf{S} \mathbf{w} = \mathbf{0}, \quad (\text{B.49})$$

which implies that  $\mathbf{J}(\mathbf{A}, \mathbf{s}, \boldsymbol{\lambda})$  is also singular since  $[\mathbf{u}^T, \mathbf{v}^T, \mathbf{w}^T] \neq \mathbf{0}$ .

Now assume that  $\mathbf{J}(\mathbf{A}, \mathbf{s}, \boldsymbol{\lambda})$  is singular. Then there exists a nonzero vector  $[\mathbf{u}^T, \mathbf{v}^T, \mathbf{w}^T]^T \neq \mathbf{0}$  such that

$$\mathbf{H} \mathbf{u} - \mathbf{A}^T \mathbf{w} = \mathbf{0}, \quad (\text{B.50})$$

$$\mathbf{A} \mathbf{u} - \mathbf{v} = \mathbf{0}, \quad (\text{B.51})$$

$$\Lambda \mathbf{v} + \mathbf{S} \mathbf{w} = \mathbf{0}. \quad (\text{B.52})$$

Then  $\mathbf{u}$  and  $\mathbf{w}$  naturally satisfy (B.47) and (B.48). If  $\mathbf{u} = \mathbf{0}$  and  $\mathbf{w} = \mathbf{0}$  at the same time, then  $\mathbf{v}$  is also a zero vector, a contradiction. Thus  $J_a(\mathbf{A}, \mathbf{s}, \boldsymbol{\lambda})$  is also singular.  $\square$

From here on, we denote by  $\mathbf{x}^*$  the optimal solution to (2.2) (By Assumption 2.5, it exists and is unique) and by  $\boldsymbol{\lambda}^*$  its associated Lagrange multiplier. We define  $\mathbf{s}^* := \mathbf{A} \mathbf{x}^* - \mathbf{b}$ .

**Lemma B.16.** *(Corresponds to Lemma 13 of [TAW06]) If  $\mathcal{A}(\mathbf{x}^*) \subseteq Q$  then  $\mathbf{J}(\mathbf{A}_Q, \mathbf{s}_Q^*, \boldsymbol{\lambda}_Q^*)$  and  $\mathbf{J}_a(\mathbf{A}_Q, \mathbf{s}_Q^*, \boldsymbol{\lambda}_Q^*)$  are nonsingular.*

*Proof.* Let us verify the assumptions of Lemma B.1. First,  $\mathbf{s}_Q^* + \boldsymbol{\lambda}_Q^* > \mathbf{0}$  due to strict complementarity Assumption 2.6. Second, the rows of  $\mathbf{A}_{\mathcal{A}(\mathbf{x}^*)}$  are linearly independent by Assumption 2.4, and  $s_i^* = 0$  for  $i \in \mathcal{A}(\mathbf{x}^*) \subseteq Q$  and  $s_i^* > 0$  for  $i \notin \mathcal{A}(\mathbf{x}^*)$ . Third,  $\mathbf{A}_{\mathcal{A}(\mathbf{x}^*)}$  and  $\mathbf{H}$  share the trivial nullspace due to Assumptions 2.5 and 2.6 which implies that  $\{j : \lambda_j^* \neq 0\} = \mathcal{A}(\mathbf{x}^*)$ . Thus the conclusion follows from Lemmas B.1 and B.15.  $\square$

**Lemma B.17.** *(Corresponds to Lemma 14 of [TAW06]) Under our assumptions,*

(i)  $\{\Delta \mathbf{x}^k\} \rightarrow \mathbf{0}$ ,

(ii)  $\{\tilde{\boldsymbol{\lambda}}^k\} \rightarrow \boldsymbol{\lambda}^*$ ,

(iii) *If  $\lambda_i^* \leq \lambda_{\max}$  for all  $i \in M$ , then  $\{\boldsymbol{\lambda}^k\} \rightarrow \boldsymbol{\lambda}^*$ .*

*Proof.* Since  $\{\mathbf{x}^k\} \rightarrow \mathbf{x}^*$  (Theorem 2.4), the first claim immediately follows. The second claim follows by Lemma B.7 and the third claim by (2.51).  $\square$

The q-quadratic convergence will be shown using the following property of Newton's method, which is adapted from Proposition 3.10 of [TZ94].

**Proposition B.18.** *(Corresponds to Proposition 3.10 of [TZ94]) Let  $\Psi : \mathbb{R}^n \rightarrow \mathbb{R}^n$  be twice continuously differentiable and let  $\mathbf{t}^* \in \mathbb{R}^n$  be a zero of  $\Psi$ , i.e.,  $\Psi(\mathbf{t}^*) = \mathbf{0}$ . Suppose there exists  $\epsilon > 0$  such that  $\frac{\partial \Psi}{\partial \mathbf{t}}(\mathbf{t})$  is nonsingular for all  $\mathbf{t} \in B(\hat{\mathbf{s}}, \epsilon) := \{\mathbf{t} : \|\mathbf{t} - \mathbf{t}^*\| \leq \epsilon\}$ . Define  $\Delta^N \mathbf{t}$  to be the Newton increment at  $\mathbf{t}$ , i.e.,  $\Delta^N \mathbf{t} :=$*

$-(\frac{\partial \Psi}{\partial \mathbf{t}}(\mathbf{t}))^{-1} \Psi(\mathbf{t})$ . Then, given any  $c > 0$ , for all  $\mathbf{t} \in B(\mathbf{t}^*, \epsilon)$ , if  $\mathbf{t}^+ \in \mathbb{R}^n$  satisfies, for each  $i \in \{1, \dots, n\}$ , either

$$(i) |t_i^+ - t_i^*| \leq c \|\Delta^N \mathbf{t}\|^2$$

or

$$(ii) |t_i^+ - (t_i + \Delta^N t_i)| \leq c \|\Delta^N \mathbf{t}\|^2,$$

then there exists  $\nu > 0$  such that

$$\|\mathbf{t}^+ - \mathbf{t}^*\| \leq \nu \|\mathbf{t} - \mathbf{t}^*\|^2. \quad (\text{B.53})$$

*Proof.* See [TZ94]. □

To use this proposition, we write the first three conditions of the KKT system (2.9)-(2.11) as  $\Psi(\mathbf{x}, \boldsymbol{\lambda}) = 0$ , where

$$\Psi(\mathbf{x}, \boldsymbol{\lambda}) := \begin{bmatrix} \mathbf{H}\mathbf{x} - \mathbf{A}^T \boldsymbol{\lambda} + \mathbf{c} \\ \boldsymbol{\Lambda}(\mathbf{A}\mathbf{x} - \mathbf{b}) \end{bmatrix}. \quad (\text{B.54})$$

Then (2.30) is equivalent to the Newton direction for the solution of  $\Psi(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{0}$ , and  $\mathbf{J}_a(\mathbf{A}, \mathbf{A}\mathbf{x} - \mathbf{b}, \boldsymbol{\lambda})$  is the Jacobian of  $\Psi(\mathbf{x}, \boldsymbol{\lambda})$ . In other words,

$$\begin{bmatrix} \mathbf{H} & -\mathbf{A}^T \\ \boldsymbol{\Lambda}\mathbf{A} & \mathbf{S} \end{bmatrix} \begin{bmatrix} \Delta^N \mathbf{x} \\ \Delta^N \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} -\mathbf{H}\mathbf{x} - \mathbf{c} + \mathbf{A}^T \boldsymbol{\lambda} \\ -\boldsymbol{\Lambda}\mathbf{s} \end{bmatrix}. \quad (\text{B.55})$$

Although the direction generated by Algorithm 1 is not the same as the Newton direction  $\Delta^N \mathbf{t}$  of  $\Psi$ , Lemma B.19 will relate the two directions.

We use  $\mathbf{t}$  to denote the vector containing both  $\mathbf{x}$  and  $\boldsymbol{\lambda}$ , i.e.,  $\mathbf{t}^k := [\mathbf{x}^{kT}, \boldsymbol{\lambda}^{kT}]^T$ .

Also, we define a strictly feasible set  $E^o$  for  $\Psi$ :

$$E^o := \{\mathbf{t} : \mathbf{x} \in \mathcal{F}_P^o, \boldsymbol{\lambda} > 0\}. \quad (\text{B.56})$$

Hence,  $E^o \cap B(\mathbf{t}^*, \epsilon)$  denotes the set of strictly feasible points in a ball around  $\mathbf{t}^*$ . Given  $\mathbf{t} \in E^o$  and  $Q \in \mathcal{Q}(\mathbf{Ax} - \mathbf{b}, q)$ ,  $\Delta \mathbf{t} := [\Delta \mathbf{x}^T, \Delta \boldsymbol{\lambda}^T]^T$  denotes the composite direction at  $\mathbf{t}$  generated by Algorithm 1. Superscript + is attached to denote the quantities of the next iteration. We denote by  $[\Delta^N \mathbf{x}^T, \Delta^N \boldsymbol{\lambda}^T]^T \equiv \Delta^N \mathbf{t}$  the decomposed Newton direction for  $\Psi$  at  $\mathbf{t}$ .

**Lemma B.19.** *(Corresponds to Lemma 16 of [TAW06].) Let  $\epsilon$  be such that, for all  $\mathbf{t} \in E^o \cap B(\mathbf{t}^*, \epsilon)$  and for all  $Q \in \mathcal{Q}(\mathbf{Ax} - \mathbf{b}, q)$ ,  $\mathbf{J}_a(\mathbf{A}_Q, \mathbf{A}_Q \mathbf{x} - \mathbf{b}, \boldsymbol{\lambda}_Q)$  is nonsingular and  $\mathbf{A}_{Q^c} \mathbf{x} > \mathbf{b}_{Q^c}$ . Then there exists a positive constant  $\xi$  such that, for all  $\mathbf{t} \in E^o \cap B(\mathbf{t}^*, \epsilon)$  and for any  $Q \in \mathcal{Q}(\mathbf{Ax} - \mathbf{b}, q)$ ,*

$$\|\Delta \mathbf{t} - \Delta^N \mathbf{t}\| \leq \xi \|\mathbf{t} - \mathbf{t}^*\| \|\Delta^N \mathbf{t}\|.$$

*Proof.* Let  $\mathbf{t} := (\mathbf{x}^T, \boldsymbol{\lambda}^T)^T$  and let  $\mathbf{s} := \mathbf{Ax} - \mathbf{b}$ . By applying block elimination to the reduced Newton system (2.36)-(2.38), we can obtain an augmented reduced system:

$$\begin{bmatrix} \mathbf{H} & -\mathbf{A}_Q^T \\ \boldsymbol{\Lambda}_{Q^2} \mathbf{A}_Q & \mathbf{S}_{Q^2} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x} \\ \Delta \boldsymbol{\lambda}_Q \end{bmatrix} = \begin{bmatrix} -\mathbf{H}\mathbf{x} - \mathbf{c} + \mathbf{A}_Q^T \boldsymbol{\lambda}_Q \\ -\boldsymbol{\Lambda}_{Q^2} \mathbf{s}_Q \end{bmatrix}. \quad (\text{B.57})$$

The Newton direction  $\Delta^N \mathbf{t}$  for (B.54) satisfies (B.55). With a simple rearrangement of (B.55), we obtain

$$\begin{bmatrix} \mathbf{H} & -\mathbf{A}_Q^T & -\mathbf{A}_{Q^c}^T \\ \Lambda_{Q^2} \mathbf{A}_Q & \mathbf{S}_{Q^2} & \mathbf{0} \\ \Lambda_{Q^{c2}} \mathbf{A}_{Q^c} & \mathbf{0} & \mathbf{S}_{Q^{c2}} \end{bmatrix} \begin{bmatrix} \Delta^N \mathbf{x} \\ \Delta^N \boldsymbol{\lambda}_Q \\ \Delta^N \boldsymbol{\lambda}_{Q^c} \end{bmatrix} = \begin{bmatrix} -\mathbf{H}\mathbf{x} - \mathbf{c} + \mathbf{A}^T \boldsymbol{\lambda} \\ -\Lambda_{Q^2} \mathbf{s}_Q \\ -\Lambda_{Q^{c2}} \mathbf{s}_{Q^c} \end{bmatrix}. \quad (\text{B.58})$$

Multiplying  $\mathbf{A}_{Q^c} \mathbf{S}_{Q^{c2}}^{-1}$  to the third block row, adding the multiplied third block row to the first block row, and then eliminating  $\Delta^N \boldsymbol{\lambda}_{Q^c}$  from the system of equations above lead to

$$\begin{bmatrix} \mathbf{H} + \mathbf{A}_{Q^c}^T \mathbf{S}_{Q^{c2}}^{-1} \Lambda_{Q^{c2}} \mathbf{A}_{Q^c} & -\mathbf{A}_Q^T \\ \Lambda_{Q^2} \mathbf{A}_Q & \mathbf{S}_{Q^2} \end{bmatrix} \begin{bmatrix} \Delta^N \mathbf{x} \\ \Delta^N \boldsymbol{\lambda}_Q \end{bmatrix} = \begin{bmatrix} -\mathbf{H}\mathbf{x} - \mathbf{c} + \mathbf{A}_Q^T \boldsymbol{\lambda}_Q \\ -\Lambda_{Q^2} \mathbf{s}_Q \end{bmatrix}. \quad (\text{B.59})$$

Since (B.57) and (B.59) have the same right-hand side, equating the left hand side of (B.59) and (B.57) yields

$$\mathbf{J}_a(\mathbf{A}_Q, \mathbf{A}_Q \mathbf{x} - \mathbf{b}_Q, \boldsymbol{\lambda}_Q) \begin{bmatrix} \Delta \mathbf{x} - \Delta^N \mathbf{x} \\ \Delta \boldsymbol{\lambda}_Q - \Delta^N \boldsymbol{\lambda}_Q \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{Q^c}^T \mathbf{S}_{Q^{c2}}^{-1} \Lambda_{Q^{c2}} \mathbf{A}_{Q^c} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta^N \mathbf{x} \\ \Delta^N \boldsymbol{\lambda}_Q \end{bmatrix}. \quad (\text{B.60})$$

Then the nonsingularity of  $\mathbf{J}_a(\mathbf{A}_Q, \mathbf{A}_Q \mathbf{x} - \mathbf{b}_Q, \boldsymbol{\lambda}_Q)$  (by the assumption) leads to

$$\begin{bmatrix} \Delta \mathbf{x} - \Delta^N \mathbf{x} \\ \Delta \boldsymbol{\lambda}_Q - \Delta^N \boldsymbol{\lambda}_Q \end{bmatrix} = \mathbf{J}_a(\mathbf{A}_Q, \mathbf{A}_Q \mathbf{x} - \mathbf{b}_Q, \boldsymbol{\lambda}_Q)^{-1} \begin{bmatrix} \mathbf{A}_{Q^c}^T \mathbf{S}_{Q^{c2}}^{-1} \Lambda_{Q^{c2}} \mathbf{A}_{Q^c} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta^N \mathbf{x} \\ \Delta^N \boldsymbol{\lambda}_Q \end{bmatrix}. \quad (\text{B.61})$$

Since  $\mathcal{A}(\mathbf{x}^*) \cap Q^c = \emptyset$ , in view of strict complementarity,  $\boldsymbol{\lambda}_{Q^c}^* = \mathbf{0}$  and  $\boldsymbol{\Lambda}_{Q^{c2}}^* = \mathbf{0}$ .

So, there exists some positive constant  $c_0$  independent of  $\mathbf{t}$  such that

$$\|\boldsymbol{\Lambda}_{Q^{c2}}\| = \|\boldsymbol{\Lambda}_{Q^{c2}} - \boldsymbol{\Lambda}_{Q^c}^*\| \leq c_0 \|\boldsymbol{\lambda}_{Q^c} - \boldsymbol{\lambda}_{Q^c}^*\| \leq c_0 \|\mathbf{t} - \mathbf{t}^*\|. \quad (\text{B.62})$$

By taking the norm of both sides of (B.61) and by using (B.62), we obtain

$$\begin{aligned} & \left\| \begin{array}{c} \Delta \mathbf{x} - \Delta^N \mathbf{x} \\ \Delta \boldsymbol{\lambda}_Q - \Delta^N \boldsymbol{\lambda}_Q \end{array} \right\| \\ & \leq c_0 \|\mathbf{J}_a(\mathbf{A}_Q, \mathbf{A}_Q \mathbf{x} - \mathbf{b}_Q, \boldsymbol{\lambda}_Q)^{-1}\| \|\mathbf{A}_{Q^c}^T\| \|\mathbf{S}_{Q^{c2}}^{-1}\| \|\mathbf{A}_{Q^c}\| \|\mathbf{t} - \mathbf{t}^*\| \left\| \begin{array}{c} \Delta^N \mathbf{x} \\ \Delta^N \boldsymbol{\lambda}_Q \end{array} \right\|. \end{aligned} \quad (\text{B.63})$$

In addition, it follows from (B.57) and (B.58), by taking  $Q^c$  components of  $\Delta^N \boldsymbol{\lambda}$  and  $\Delta \boldsymbol{\lambda}$ , that

$$\Delta \boldsymbol{\lambda}_{Q^c} - \Delta^N \boldsymbol{\lambda}_{Q^c} = -\mathbf{S}_{Q^{c2}}^{-1} \boldsymbol{\Lambda}_{Q^{c2}} \mathbf{A}_{Q^c} (\Delta \mathbf{x} - \Delta^N \mathbf{x}). \quad (\text{B.64})$$

Then taking the norm yields, due to (B.62),

$$\|\Delta \boldsymbol{\lambda}_{Q^c} - \Delta^N \boldsymbol{\lambda}_{Q^c}\| \leq c_0 \|\mathbf{S}_{Q^{c2}}^{-1}\| \|\mathbf{t} - \mathbf{t}^*\| \|\mathbf{A}_{Q^c}\| \|\Delta \mathbf{x} - \Delta^N \mathbf{x}\|. \quad (\text{B.65})$$

Finally, for  $\mathbf{t} \in E^o \cap B(\mathbf{t}^*, \epsilon)$  and a fixed  $Q$ , since  $\mathbf{J}_a(\mathbf{A}_Q, \mathbf{A}_Q \mathbf{x} - \mathbf{b}_Q, \boldsymbol{\lambda}_Q)$  and  $\mathbf{S}_{Q^{c2}}$  are nonsingular and continuous,  $\|\mathbf{J}_a(\mathbf{A}_Q, \mathbf{A}_Q \mathbf{x} - \mathbf{b}_Q, \boldsymbol{\lambda}_Q)^{-1}\|$  and  $\|\mathbf{S}_{Q^{c2}}^{-1}\|$  are bounded above. In addition, since the number of possible candidates for  $Q$  and  $Q^c$  is finite,  $\|\mathbf{J}_a(\mathbf{A}_Q, \mathbf{A}_Q \mathbf{x} - \mathbf{b}_Q, \boldsymbol{\lambda}_Q)^{-1}\|$ ,  $\|\mathbf{S}_{Q^{c2}}^{-1}\|$ ,  $\|\mathbf{A}_{Q^c}\|$  and  $\|\mathbf{A}_{Q^c}^T\|$  are bounded above for any  $Q$ .

So, for some positive constant  $c_1$  independent of  $\mathbf{t}$ , we can derive from (B.63)

$$\left\| \begin{array}{c} \Delta \mathbf{x} - \Delta^N \mathbf{x} \\ \Delta \boldsymbol{\lambda}_Q - \Delta^N \boldsymbol{\lambda}_Q \end{array} \right\| \leq c_1 \|\mathbf{t} - \mathbf{t}^*\| \left\| \begin{array}{c} \Delta^N \mathbf{x} \\ \Delta^N \boldsymbol{\lambda}_Q \end{array} \right\| \leq c_1 \|\mathbf{t} - \mathbf{t}^*\| \|\Delta^N \mathbf{t}\|. \quad (\text{B.66})$$

Moreover, since  $\mathbf{t}$  is bounded, there exist some positive constants  $c_2$  and  $c_3$  independent of  $\mathbf{t}$  such that, by (B.65) and (B.66),

$$\begin{aligned} \|\Delta \boldsymbol{\lambda}_{Q^c} - \Delta^N \boldsymbol{\lambda}_{Q^c}\| &\leq c_2 \|\mathbf{t} - \mathbf{t}^*\| \|\Delta \mathbf{x} - \Delta^N \mathbf{x}\| \\ &\leq c_2 \|\mathbf{t} - \mathbf{t}^*\| \left\| \begin{array}{c} \Delta \mathbf{x} - \Delta^N \mathbf{x} \\ \Delta \boldsymbol{\lambda}_Q - \Delta^N \boldsymbol{\lambda}_Q \end{array} \right\| \\ &\leq c_2 c_1 \|\mathbf{t} - \mathbf{t}^*\|^2 \|\Delta^N \mathbf{t}\| \\ &\leq c_3 \|\mathbf{t} - \mathbf{t}^*\| \|\Delta^N \mathbf{t}\| \end{aligned} \quad (\text{B.67})$$

Therefore combining (B.66) and (B.67) using the fact that  $\|[\mathbf{x}_1^T, \mathbf{x}_2^T]^T\| \leq \|\mathbf{x}_1\| + \|\mathbf{x}_2\|$  proves the claim.  $\square$

We now establish the q-quadratic rate of convergence.

*Proof of Theorem 2.5.* We confine our interest to  $\mathbf{t}$  strictly feasible and close enough to  $\mathbf{t}^*$  to make use of Lemma B.19. So we assume  $\mathbf{t}$  exists inside a ball of radii  $\epsilon$ :  $\|\mathbf{t} - \mathbf{t}^*\| \leq \epsilon$ . We are interested in taking one step from  $\mathbf{t}$  to  $\mathbf{t}^+$  according to Algorithm 1. Note that we will frequently use strict complementarity Assumption 2.6 and the triangle inequality  $\|\mathbf{v} + \mathbf{u}\| \leq \|\mathbf{v}\| + \|\mathbf{u}\|$  for any same dimensional vectors  $\mathbf{u}$  and  $\mathbf{v}$ . Also notice  $\|\mathbf{u}\| \leq \|(\mathbf{u}^T, \mathbf{w}^T)^T\|$ ,  $(\|\mathbf{u}\| + \|\mathbf{w}\|)^2 \leq 2(\|\mathbf{u}\|^2 + \|\mathbf{w}\|^2)$ , and  $|u_i| \leq \|\mathbf{u}\|$  for any vectors  $\mathbf{u}$  and  $\mathbf{w}$  and for any  $p$ -norm ( $p = 1, 2, \dots, \infty$ ).

Let  $\mathbf{s} := \mathbf{A}\mathbf{x} - \mathbf{b}$  and  $\mathbf{s}^* := \mathbf{A}\mathbf{x}^* - \mathbf{b}$ . Let  $Q$  be fixed and  $\mathcal{A}(\mathbf{x}^*) \subseteq Q$ . Then by the strict complementarity Assumption 2.6, it follows that  $\mathbf{S}_{Q^c}^* \boldsymbol{\lambda}_Q^* = \mathbf{0}$  and  $\boldsymbol{\lambda}_{Q^c}^* = \mathbf{0}$ . Since  $\mathbf{H}\mathbf{x}^* + \mathbf{c} = \mathbf{A}^T \boldsymbol{\lambda}^*$  by the first KKT condition (2.9), it follows from the definition of  $\mathbf{J}$  (2.32) that

$$\mathbf{J}(\mathbf{A}_Q, \mathbf{s}_Q^*, \boldsymbol{\lambda}_Q^*) \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \boldsymbol{\lambda}_Q^* \end{bmatrix} = \begin{bmatrix} -\mathbf{H}\mathbf{x}^* - \mathbf{c} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}.$$

Also we know from (2.32) and (2.36)-(2.38) that

$$\mathbf{J}(\mathbf{A}_Q, \mathbf{s}_Q, \boldsymbol{\lambda}_Q) \begin{bmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{s}_Q \\ \tilde{\boldsymbol{\lambda}}_Q \end{bmatrix} = \begin{bmatrix} -\mathbf{H}\mathbf{x} - \mathbf{c} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}.$$

So  $\mathbf{J}(\mathbf{A}_Q, \mathbf{s}_Q, \boldsymbol{\lambda}_Q)$  is continuous and nonsingular (due to Lemma B.1 under Assumption 2.1),  $\mathbf{J}(\mathbf{A}_Q, \mathbf{s}_Q^*, \boldsymbol{\lambda}_Q^*)$  is nonsingular (by Lemma B.16),  $-\mathbf{H}\mathbf{x} - \mathbf{c}$  is continuous,  $\mathbf{s}_{Q^c} \rightarrow \mathbf{s}_{Q^c}^* > \mathbf{0}$  as  $\mathbf{t} \rightarrow \mathbf{t}^*$ ,  $\mathbf{s}$  and  $\boldsymbol{\lambda}$  are strictly complementary, and

$$\tilde{\boldsymbol{\lambda}}_{Q^c} = -\mathbf{S}_{Q^c}^{-1} \mathbf{A}_{Q^c} \mathbf{A}_Q \Delta \mathbf{x}$$

by (B.26). It follows that

$$\Delta \mathbf{x} \rightarrow \mathbf{0} \text{ as } \mathbf{t} \rightarrow \mathbf{t}^* \tag{B.68}$$

$$\tilde{\boldsymbol{\lambda}} \rightarrow \boldsymbol{\lambda}^* \text{ as } \mathbf{t} \rightarrow \mathbf{t}^*. \tag{B.69}$$

Notice this holds for any  $Q \supseteq \mathcal{A}(\mathbf{x}^*)$ .

Now we will first investigate components of  $\boldsymbol{\lambda}$  for  $i \in \mathcal{A}(\mathbf{x}^*)$ . Note that for  $\mathbf{t}$  strictly feasible and close enough to  $\mathbf{t}^*$ , we can assume, without loss of generality, that  $\mathcal{A}(\mathbf{x}^*) \subseteq \mathcal{Q} \in \mathcal{Q}(\mathbf{Ax} - \mathbf{b}, q)$  for  $q \geq n$ . So, since  $\tilde{\boldsymbol{\lambda}}_{\mathcal{A}(\mathbf{x}^*)} \rightarrow \boldsymbol{\lambda}_{\mathcal{A}(\mathbf{x}^*)}^* > \mathbf{0}$  and  $\tilde{\boldsymbol{\lambda}}_{\mathcal{A}(\mathbf{x}^*)^c} \rightarrow \boldsymbol{\lambda}_{\mathcal{A}(\mathbf{x}^*)^c}^* = \mathbf{0}$  as  $\mathbf{t} \rightarrow \mathbf{t}^*$  by strict complementarity, it holds, by (2.46), that  $(\tilde{\boldsymbol{\lambda}}_-)_i = \min(0, \tilde{\lambda}_i) = 0$  for  $i \in \mathcal{A}(\mathbf{x}^*)$  and for  $\mathbf{t}$  close enough to  $\mathbf{t}^*$ , and that  $(\tilde{\boldsymbol{\lambda}}_-)_i = \min(0, \tilde{\lambda}_i) \rightarrow 0$  for  $i \in \mathcal{A}(\mathbf{x}^*)^c$  as  $\mathbf{t} \rightarrow \mathbf{t}^*$ . From this, we know, for  $\mathbf{t}$  strictly feasible and close enough to  $\mathbf{t}^*$ , that

$$\|\Delta \mathbf{x}\|^2 + \|\tilde{\boldsymbol{\lambda}}_-\|^2 < \tilde{\lambda}_i, \forall i \in \mathcal{A}(\mathbf{x}^*). \quad (\text{B.70})$$

Since  $\lambda_i^* < \lambda_{\max}$ , it follows from (B.70) and the update rule (2.51) that

$$\boldsymbol{\lambda}_{\mathcal{A}(\mathbf{x}^*)}^+ = \tilde{\boldsymbol{\lambda}}_{\mathcal{A}(\mathbf{x}^*)} = \boldsymbol{\lambda}_{\mathcal{A}(\mathbf{x}^*)} + \Delta \boldsymbol{\lambda}_{\mathcal{A}(\mathbf{x}^*)}, \quad (\text{B.71})$$

which results in

$$\boldsymbol{\lambda}_{\mathcal{A}(\mathbf{x}^*)}^+ - (\boldsymbol{\lambda}_{\mathcal{A}(\mathbf{x}^*)} + \Delta^N \boldsymbol{\lambda}_{\mathcal{A}(\mathbf{x}^*)}) = \Delta \boldsymbol{\lambda}_{\mathcal{A}(\mathbf{x}^*)} - \Delta^N \boldsymbol{\lambda}_{\mathcal{A}(\mathbf{x}^*)}. \quad (\text{B.72})$$

By Lemma B.19, thus, (B.72) yields

$$\begin{aligned} \|\boldsymbol{\lambda}_{\mathcal{A}(\mathbf{x}^*)}^+ - (\boldsymbol{\lambda}_{\mathcal{A}(\mathbf{x}^*)} + \Delta^N \boldsymbol{\lambda}_{\mathcal{A}(\mathbf{x}^*)})\| &= \|\Delta \boldsymbol{\lambda}_{\mathcal{A}(\mathbf{x}^*)} - \Delta^N \boldsymbol{\lambda}_{\mathcal{A}(\mathbf{x}^*)}\| \\ &\leq \|\Delta \mathbf{t} - \Delta^N \mathbf{t}\| \\ &\leq \xi \|\mathbf{t} - \mathbf{t}^*\| \|\Delta^N \mathbf{t}\|. \end{aligned} \quad (\text{B.73})$$

This leads to

$$\|\boldsymbol{\lambda}_{\mathcal{A}(\mathbf{x}^*)}^+ - \boldsymbol{\lambda}_{\mathcal{A}(\mathbf{x}^*)}^*\| = \|\boldsymbol{\lambda}_{\mathcal{A}(\mathbf{x}^*)}^+ - (\boldsymbol{\lambda}_{\mathcal{A}(\mathbf{x}^*)} + \Delta^N \boldsymbol{\lambda}_{\mathcal{A}(\mathbf{x}^*)}) + (\boldsymbol{\lambda}_{\mathcal{A}(\mathbf{x}^*)} + \Delta^N \boldsymbol{\lambda}_{\mathcal{A}(\mathbf{x}^*)} - \boldsymbol{\lambda}_{\mathcal{A}(\mathbf{x}^*)}^*)\|$$

$$\begin{aligned}
&\leq \|\boldsymbol{\lambda}_{\mathcal{A}(\mathbf{x}^*)}^+ - (\boldsymbol{\lambda}_{\mathcal{A}(\mathbf{x}^*)} + \Delta^N \boldsymbol{\lambda}_{\mathcal{A}(\mathbf{x}^*)})\| + \|\boldsymbol{\lambda}_{\mathcal{A}(\mathbf{x}^*)} + \Delta^N \boldsymbol{\lambda}_{\mathcal{A}(\mathbf{x}^*)} - \boldsymbol{\lambda}_{\mathcal{A}(\mathbf{x}^*)}^*\| \\
&\leq \xi \|\mathbf{t} - \mathbf{t}^*\| \|\Delta^N \mathbf{t}\| + \|\boldsymbol{\lambda}_{\mathcal{A}(\mathbf{x}^*)} + \Delta^N \boldsymbol{\lambda}_{\mathcal{A}(\mathbf{x}^*)} - \boldsymbol{\lambda}_{\mathcal{A}(\mathbf{x}^*)}^*\|. \tag{B.74}
\end{aligned}$$

Here, let  $\hat{\mathbf{t}}^+ := \mathbf{t} + \Delta^N \mathbf{t}$ . Then, since  $|\hat{t}_i^+ - (t_i + \Delta^N t_i)| = 0 \leq \|\Delta^N \mathbf{t}\|$ , in view of Proposition B.18, we know that, for some positive constant  $c_1$  independent of  $\mathbf{t}$ ,

$$\|\hat{\mathbf{t}}^+ - \mathbf{t}^*\| \leq c_1 \|\mathbf{t} - \mathbf{t}^*\|^2, \tag{B.75}$$

which immediately yields

$$\|\boldsymbol{\lambda}_{\mathcal{A}(\mathbf{x}^*)} + \Delta^N \boldsymbol{\lambda}_{\mathcal{A}(\mathbf{x}^*)} - \boldsymbol{\lambda}_{\mathcal{A}(\mathbf{x}^*)}^*\| = \|\hat{\boldsymbol{\lambda}}_{\mathcal{A}(\mathbf{x}^*)}^+ - \boldsymbol{\lambda}_{\mathcal{A}(\mathbf{x}^*)}^*\| \leq \|\hat{\mathbf{t}}^+ - \mathbf{t}^*\| \leq c_1 \|\mathbf{t} - \mathbf{t}^*\|^2. \tag{B.76}$$

So (B.74) and (B.76) result in, for  $c_2 = \max(\xi, c_1)$  independent of  $\mathbf{t}$ ,

$$\|\boldsymbol{\lambda}_{\mathcal{A}(\mathbf{x}^*)}^+ - \boldsymbol{\lambda}_{\mathcal{A}(\mathbf{x}^*)}^*\| \leq c_2 \|\mathbf{t} - \mathbf{t}^*\| \|\Delta^N \mathbf{t}\| + c_2 \|\mathbf{t} - \mathbf{t}^*\|^2. \tag{B.77}$$

From this point, we will closely look at the other components of  $\boldsymbol{\lambda}$ ,  $\boldsymbol{\lambda}^+$  and  $\boldsymbol{\lambda}^*$ , i.e.,  $i \notin \mathcal{A}(\mathbf{x}^*)$ . Since  $\lambda_i \rightarrow 0$ ,  $\tilde{\lambda}_i \rightarrow 0$ ,  $\Delta \lambda_i \rightarrow 0$  and  $\Delta \mathbf{x} \rightarrow \mathbf{0}$  as  $\mathbf{t} \rightarrow \mathbf{t}^*$ , it holds from the dual update rule (2.51), for  $\mathbf{t}$  strictly feasible and close enough to  $\mathbf{t}^*$ , that either

$$\lambda_i^+ = \lambda_i + \Delta \lambda_i (\equiv \tilde{\lambda}_i) \tag{B.78}$$

or,

$$\lambda_i^+ = \|\Delta \mathbf{x}\|^2 + \|\tilde{\boldsymbol{\lambda}}_-\|^2. \tag{B.79}$$

Here the first case (B.78) again yields, as we did for (B.71),

$$|\lambda_i^+ - \lambda_i^*| \leq c_2 \|\mathbf{t} - \mathbf{t}^*\| \|\Delta^N \mathbf{t}\| + c_2 \|\mathbf{t} - \mathbf{t}^*\|^2. \quad (\text{B.80})$$

Consider the second case (B.79). For  $i$  such that  $\tilde{\lambda}_i \geq 0$ , we know that  $|(\tilde{\lambda}_-)_i| = 0$ . For  $i$  such that  $\tilde{\lambda}_i = \lambda_i + \Delta\lambda_i < 0$ , since  $\lambda_i > 0$ , we know that  $|(\tilde{\lambda}_-)_i| = |\lambda_i + \Delta\lambda_i| < |\Delta\lambda_i|$ .

Thus we conclude from the second case that

$$|\lambda_i^+ - \lambda_i^*| = |\lambda_i^+| = \|\Delta \mathbf{x}\|^2 + \|\tilde{\lambda}_-\|^2 \leq \|\Delta \mathbf{x}\|^2 + \|\Delta \lambda\|^2 \leq \|\Delta \mathbf{t}\|^2. \quad (\text{B.81})$$

This immediately yields, by Lemma B.19,

$$|\lambda_i^+ - \lambda_i^*| \leq \|\Delta \mathbf{t}\|^2 = \|\Delta \mathbf{t} - \Delta^N \mathbf{t} + \Delta^N \mathbf{t}\|^2 \leq (\|\Delta \mathbf{t} - \Delta^N \mathbf{t}\| + \|\Delta^N \mathbf{t}\|)^2 \quad (\text{B.82})$$

$$\leq 2\|\Delta \mathbf{t} - \Delta^N \mathbf{t}\|^2 + 2\|\Delta^N \mathbf{t}\|^2 \leq 2\xi^2 \|\mathbf{t} - \mathbf{t}^*\|^2 \|\Delta^N \mathbf{t}\|^2 + 2\|\Delta^N \mathbf{t}\|^2, \quad (\text{B.83})$$

Therefore, by combining (B.80) and (B.83) we obtain, for  $i \notin \mathcal{A}(\mathbf{x}^*)$ ,

$$|\lambda_i^+ - \lambda_i^*| \leq \max(c_2 \|\mathbf{t} - \mathbf{t}^*\| \|\Delta^N \mathbf{t}\| + c_2 \|\mathbf{t} - \mathbf{t}^*\|^2, 2\xi^2 \|\mathbf{t} - \mathbf{t}^*\|^2 \|\Delta^N \mathbf{t}\|^2 + 2\|\Delta^N \mathbf{t}\|^2). \quad (\text{B.84})$$

Since  $\|\mathbf{t} - \mathbf{t}^*\| \leq \epsilon$ , we know from (B.84), for a constant  $c_3 := \max\{c_2, 2\xi^2\epsilon^2 + 2\}$  independent of  $\mathbf{t}$  within the ball, that

$$|\lambda_i^+ - \lambda_i^*| \leq \max(c_3 \|\mathbf{t} - \mathbf{t}^*\| \|\Delta^N \mathbf{t}\| + c_3 \|\mathbf{t} - \mathbf{t}^*\|^2, c_3 \|\Delta^N \mathbf{t}\|^2), \quad (\text{B.85})$$

for  $i \notin \mathcal{A}(\mathbf{x}^*)$ .

Now let us investigate the  $\mathbf{x}$  component of  $\mathbf{t}$ . For this, we first obtain the lower bound of the step size  $\alpha$  and then bound the step in (B.87).

We will consider three cases: (i)  $\mathcal{A}(\mathbf{x}^*) \neq \emptyset$  and  $\Delta \mathbf{s} \not\geq \mathbf{0}$ , (ii)  $\mathcal{A}(\mathbf{x}^*) \neq \emptyset$  and  $\Delta \mathbf{s} \geq \mathbf{0}$ , and (iii)  $\mathcal{A}(\mathbf{x}^*) = \emptyset$ . Consider the first case, so  $\mathcal{A}(\mathbf{x}^*) \neq \emptyset$  and there exists some  $i$  such that  $\Delta s_i = \mathbf{a}_i^T \Delta \mathbf{x} < 0$ . Observing (2.47), we notice that indices corresponding to  $\Delta s_i \geq 0$  have no effect in deciding lower bounds on  $\bar{\alpha}$  and  $\alpha$ . So we focus on indices associated with  $\Delta s_i < 0$ .

It follows from (2.34) and (2.35) that

$$\frac{s_i}{\mathbf{a}_i^T \Delta \mathbf{x}} = -\frac{s_i}{\Delta s_i} = \frac{\lambda_i}{\tilde{\lambda}_i}, \text{ for } i : \Delta s_i < 0.$$

Thus, for all  $i$  such that  $i \notin \mathcal{A}(\mathbf{x}^*)$  and  $\Delta s_i < 0$ , since  $s_i^* > 0$  and  $\Delta s_i = \mathbf{a}_i^T \Delta \mathbf{x}$  converges to 0 as  $\mathbf{t} \rightarrow \mathbf{t}^*$  by (B.68), it follows that

$$\frac{\lambda_i}{\tilde{\lambda}_i} = -\frac{s_i}{\Delta s_i} \rightarrow \infty \text{ as } \mathbf{t} \rightarrow \mathbf{t}^*,$$

implying  $i$  such that  $i \notin \mathcal{A}(\mathbf{x}^*)$  and  $\Delta s_i < 0$  has no effect in determining lower bounds on  $\bar{\alpha}$  and  $\alpha$  if  $\mathbf{t}$  is close enough to  $\mathbf{t}^*$ . Thus indices  $i$  such that  $i \in \mathcal{A}(\mathbf{x}^*)$  and  $\Delta s_i < 0$  affect the lower bound on  $\bar{\alpha}$  and  $\alpha$  when  $\mathbf{t}$  is close enough to  $\mathbf{t}^*$ . Next we investigate these indices.

Assume there exist some indices  $i$  such that  $i \in \mathcal{A}(\mathbf{x}^*)$  and  $\Delta s_i < 0$ . We can rewrite (2.47) as

$$\bar{\alpha} = \min_{i \in \mathcal{A}(\mathbf{x}^*) : \Delta s_i < 0} \frac{\lambda_i}{\tilde{\lambda}_i}.$$

For  $i \in \mathcal{A}(\mathbf{x}^*)$ , since  $\lambda_i \rightarrow \lambda_i^* > 0$  (by the definition of  $\mathbf{t}$  and  $\mathbf{t}^*$ ) and  $\tilde{\lambda}_i \rightarrow \lambda_i^*$  by (B.69) as  $\mathbf{t} \rightarrow \mathbf{t}^*$ , we know that  $\bar{\alpha} \rightarrow 1$  as  $\mathbf{t} \rightarrow \mathbf{t}^*$ . Now define  $j$  as the index that

determines  $\bar{\alpha}$ . Then we can restate (2.48) as

$$\alpha = \min\left\{1, \frac{\lambda_j}{\tilde{\lambda}_j} - \|\Delta\mathbf{x}\|\right\},$$

for  $\mathbf{t}$  strictly feasible close enough to  $\mathbf{t}^*$ . So it holds that

$$|1 - \alpha| \leq \left|1 - \frac{\lambda_j}{\tilde{\lambda}_j} + \|\Delta\mathbf{x}\|\right| \leq \|\Delta\mathbf{x}\| + \left|\frac{\tilde{\lambda}_j - \lambda_j}{\tilde{\lambda}_j}\right|. \quad (\text{B.86})$$

Thus

$$\begin{aligned} \|\mathbf{x}^+ - (\mathbf{x} + \Delta\mathbf{x})\| &= \|\alpha\Delta\mathbf{x} - \Delta\mathbf{x}\| = |1 - \alpha|\|\Delta\mathbf{x}\| \\ &\leq \left(\|\Delta\mathbf{x}\| + \left|\frac{\tilde{\lambda}_j - \lambda_j}{\tilde{\lambda}_j}\right|\right)\|\Delta\mathbf{x}\|, \end{aligned}$$

which implies, by Lemma B.19 and by the fact that  $\tilde{\lambda}_j \rightarrow \lambda_j^* > 0$  as  $\mathbf{t} \rightarrow \mathbf{t}^*$  and  $|\tilde{\lambda}_j - \lambda_j| = |\Delta\lambda_j| \leq \|\Delta\boldsymbol{\lambda}\|$ , the existence of some positive constant  $c_4$  independent of  $\mathbf{t}$  such that

$$\begin{aligned} \|\mathbf{x}^+ - (\mathbf{x} + \Delta\mathbf{x})\| &\leq (c_4\|\Delta\mathbf{x}\| + c_4\|\Delta\boldsymbol{\lambda}\|)\|\Delta\mathbf{x}\| \\ &\leq c_4\sqrt{2}\|\Delta\mathbf{t}\|^2 \\ &\leq c_4\sqrt{2}(\|\Delta\mathbf{t} - \Delta^N\mathbf{t}\| + \|\Delta^N\mathbf{t}\|)^2 \\ &\leq c_4\sqrt{2}(\xi\|\mathbf{t} - \mathbf{t}^*\| + \|\Delta^N\mathbf{t}\|)^2. \end{aligned} \quad (\text{B.87})$$

This also holds when there's no index  $i$  satisfying both  $i \in \mathcal{A}(\mathbf{x}^*)$  and  $\Delta s_i < 0$ , since  $\alpha = 1$  and  $\mathbf{x}^+ - (\mathbf{x} + \Delta\mathbf{x}) = \mathbf{0}$ .

Let us consider the other cases. In case (ii),  $\mathcal{A}(\mathbf{x}^*) \neq \emptyset$  and  $\Delta\mathbf{s} \geq \mathbf{0}$ , so  $\bar{\alpha} = \infty$  and  $\alpha = 1$  by (2.47) and (2.48). In case (iii),  $\mathcal{A}(\mathbf{x}^*) = \emptyset$ , so  $s_i^* > 0$  for all  $i \in M$ . So,

since  $\Delta s_i = \mathbf{a}_i^T \Delta \mathbf{x}$  converges to 0 as  $\mathbf{t} \rightarrow \mathbf{t}^*$  by (B.68) and  $s_i$  is bounded away from 0 for all  $i$ ,  $\bar{\alpha} \rightarrow \infty$  as  $\mathbf{t} \rightarrow \mathbf{t}^*$  by (2.47), meaning  $\bar{\alpha}$  becomes large enough to make  $\alpha = 1$  for  $\mathbf{t}$  strictly feasible and close enough to  $\mathbf{t}^*$ . In these two cases, since  $\alpha = 1$  for  $\mathbf{t}$  strictly feasible and close enough to  $\mathbf{t}^*$ , the same inequality as (B.87) naturally holds, because its left-hand side becomes 0.

Therefore, by (B.87), by Lemma B.19 and by the triangle inequality, it follows that

$$\begin{aligned} \|\mathbf{x}^+ - (\mathbf{x} + \Delta^N \mathbf{x})\| &= \|\mathbf{x}^+ - (\mathbf{x} + \Delta \mathbf{x}) + (\Delta \mathbf{x} - \Delta^N \mathbf{x})\| \\ &\leq \|\mathbf{x}^+ - (\mathbf{x} + \Delta \mathbf{x})\| + \|\Delta \mathbf{x} - \Delta^N \mathbf{x}\| \\ &\leq c_4 \sqrt{2} (\xi \|\mathbf{t} - \mathbf{t}^*\| \|\Delta^N \mathbf{t}\| + \|\Delta^N \mathbf{t}\|)^2 + \xi \|\mathbf{t} - \mathbf{t}^*\| \|\Delta^N \mathbf{t}\|. \end{aligned} \quad (\text{B.88})$$

Now (B.88) leads to

$$\begin{aligned} \|\mathbf{x}^+ - \mathbf{x}^*\| &= \|\mathbf{x}^+ - (\mathbf{x} + \Delta^N \mathbf{x}) + (\mathbf{x} + \Delta^N \mathbf{x}) - \mathbf{x}^*\| \\ &\leq c_4 \sqrt{2} (\xi \|\mathbf{t} - \mathbf{t}^*\| \|\Delta^N \mathbf{t}\| + \|\Delta^N \mathbf{t}\|)^2 + \xi \|\mathbf{t} - \mathbf{t}^*\| \|\Delta^N \mathbf{t}\| + \|(\mathbf{x} + \Delta^N \mathbf{x}) - \mathbf{x}^*\| \\ &\leq c_4 \sqrt{2} (\xi \|\mathbf{t} - \mathbf{t}^*\| \|\Delta^N \mathbf{t}\| + \|\Delta^N \mathbf{t}\|)^2 + \xi \|\mathbf{t} - \mathbf{t}^*\| \|\Delta^N \mathbf{t}\| + c_1 \|\mathbf{t} - \mathbf{t}^*\|^2, \end{aligned} \quad (\text{B.89})$$

because  $\|(\mathbf{x} + \Delta^N \mathbf{x}) - \mathbf{x}^*\| \leq \|(\mathbf{t} + \Delta^N \mathbf{t}) - \mathbf{t}^*\|$  and every component of  $\hat{\mathbf{t}}^+ := \mathbf{t} + \Delta^N \mathbf{t}$  satisfies condition (ii) of Proposition B.18 leading to (B.75). Since  $\|\mathbf{t} - \mathbf{t}^*\| \leq \epsilon$ , for some constant  $c_5 := \max\{c_4 \sqrt{2} (\xi \epsilon + 1)^2, \xi, c_1\}$ ,

$$\|\mathbf{x}^+ - \mathbf{x}^*\| \leq c_5 \|\Delta^N \mathbf{t}\|^2 + c_5 \|\mathbf{t} - \mathbf{t}^*\| \|\Delta^N \mathbf{t}\| + c_5 \|\mathbf{t} - \mathbf{t}^*\|^2. \quad (\text{B.90})$$

We are ready to complete this proof. First consider the case when  $\mathbf{t}$  satisfies  $\|\Delta^N \mathbf{t}\| \leq \|\mathbf{t} - \mathbf{t}^*\|$ . Then since  $\|\mathbf{t} - \mathbf{t}^*\| \leq \epsilon$ , it follows from (B.77), (B.85), and (B.90) that, for some positive constant  $c_6$  independent of  $\mathbf{t}$  within the ball,

$$\begin{aligned}
\|\mathbf{t}^+ - \mathbf{t}^*\| &\leq \|\boldsymbol{\lambda}_{\mathcal{A}(\mathbf{x}^*)}^+ - \boldsymbol{\lambda}_{\mathcal{A}(\mathbf{x}^*)}^*\| + \sum_{i \notin \mathcal{A}(\mathbf{x}^*)} |\lambda_i^+ - \lambda^*| + \|\mathbf{x}^+ - \mathbf{x}^*\| \\
&\leq 2c_2 \|\mathbf{t} - \mathbf{t}^*\|^2 + 2c_3 |\mathcal{A}(\mathbf{x}^*)^c| \|\mathbf{t} - \mathbf{t}^*\|^2 + 3c_5 \|\mathbf{t} - \mathbf{t}^*\|^2 \\
&\leq c_6 \|\mathbf{t} - \mathbf{t}^*\|^2,
\end{aligned} \tag{B.91}$$

which yields the required result.

On the other hand, when  $\|\mathbf{t} - \mathbf{t}^*\| \leq \|\Delta^N \mathbf{t}\|$ , since  $\|\Delta^N \mathbf{t}\|$  is bounded, (B.77), (B.85), and (B.90) yield

$$\begin{aligned}
\|\mathbf{t}^+ - \mathbf{t}^*\| &\leq \|\boldsymbol{\lambda}_{\mathcal{A}(\mathbf{x}^*)}^+ - \boldsymbol{\lambda}_{\mathcal{A}(\mathbf{x}^*)}^*\| + \sum_{i \notin \mathcal{A}(\mathbf{x}^*)} |\lambda_i^+ - \lambda^*| + \|\mathbf{x}^+ - \mathbf{x}^*\| \\
&\leq 2c_2 \|\Delta^N \mathbf{t}\|^2 + 2c_3 |\mathcal{A}(\mathbf{x}^*)^c| \|\Delta^N \mathbf{t}\|^2 + 3c_5 \|\Delta^N \mathbf{t}\|^2 \\
&\leq c_6 \|\Delta^N \mathbf{t}\|^2.
\end{aligned} \tag{B.92}$$

Since, in view of (B.92) every component of  $\mathbf{t}^+$  satisfies condition (i) of Proposition B.18, it holds, for some constant  $c_7$ , that

$$\|\mathbf{t}^+ - \mathbf{t}^*\| \leq c_7 \|\mathbf{t} - \mathbf{t}^*\|^2, \tag{B.93}$$

which, together with (B.91), proves the q-quadratic convergence.  $\square$

## BIBLIOGRAPHY

- [AA95] Erling D. Andersen and Knud D. Andersen. Presolving in linear programming. *Mathematical Programming*, 71(2):221–245, 1995.
- [ABD<sup>+</sup>90] E. Anderson, Z. Bai, J. Dongarra, A. Greenbaum, A. McKenney, J. Du Croz, S. Hammarling, J. Demmel, C. Bischof, and D. Sorensen. LAPACK: a portable linear algebra library for high-performance computers. In *Supercomputing '90: Proceedings of the 1990 ACM/IEEE conference on Supercomputing*, pages 2–11, Washington, DC, USA, 1990. IEEE Computer Society.
- [AN07] A. Asuncion and D.J. Newman. UCI machine learning repository, 2007.
- [Atk89] Kendall E. Atkinson. *An Introduction to Numerical Analysis*. John Wiley & Sons, Washington, DC, USA, 1989.
- [BC04] Daniel Boley and Dongwei Cao. Training support vector machines using adaptive clustering. In *Proceedings of the SIAM Conference on Data Mining*, 2004.
- [BDD<sup>+</sup>02] L. Susan Blackford, James Demmel, Jack Dongarra, Iain Duff, Sven Hammarling, Greg Henry, Michael Heroux, Linda Kaufman, Andrew Lumsdaine, Antoine Petitet, Roldan Pozo, Karin Remington, and R. Clint

- Whaley. An updated set of Basic Linear Algebra Subprograms (BLAS). *ACM Transactions on Mathematical Software*, 28(2):135–151, June 2002.
- [BJ05] Francis R. Bach and Michael I. Jordan. Predictive low-rank decomposition for kernel methods. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 33–40, New York, NY, USA, 2005. ACM Press.
- [Bur98] Chris Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.
- [Car04] Coralia Cartis. *On Interior Point Methods for Linear Programming*. PhD thesis, University of Cambridge, 2004.
- [Cha07] Olivier Chapelle. Training a support vector machine in the primal. *Neural Computation*, 19(5):1155–1178, 2007.
- [CL01] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [CSS06] Tat-Jun Chin, Konrad Schindler, and David Suter. Incremental kernel SVD for face recognition with image sets. In *FGR '06: Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition*

- (*FGR06*), pages 461–466, Washington, DC, USA, 2006. IEEE Computer Society.
- [CV95] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, Sep 1995.
- [Dan63] G. B. Dantzig. *Linear Programming and Extensions*. Princeton University Press, Princeton, NJ, USA, 1963.
- [DM05] Petros Drineas and Michael W. Mahoney. On the Nystrom method for approximating a Gram matrix for improved kernel-based learning, 2005.
- [DY91] G. B. Dantzig and Y. Ye. A build-up interior-point method for linear programming : Affine scaling form. Technical report, University of Iowa, Iowa City, IA, USA, July 1991.
- [Fle87] R. Fletcher. *Practical Methods of Optimization*. Wiley-Interscience, New York, NY, USA, 2nd edition, 1987.
- [FM02] Michael C. Ferris and Todd S. Munson. Interior-point methods for massive support vector machines. *SIAM Journal on Optimization*, 13(3):783–804, 2002.
- [FS02] Shai Fine and Katya Scheinberg. Efficient SVM training using low-rank kernel representations. *Journal of Machine Learning Research*, 2:243–264, 2002.

- [GG05] E. Michael Gertz and Joshua D. Griffin. Support vector machine classifiers for large data sets. Preprint, October 2005.
- [GR87] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *Journal of Computational Physics*, 73:325–348, 1987.
- [GVL96] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, MD, USA, 3rd edition, 1996.
- [Hea98] Marti A. Hearst. Trends and controversies: Support vector machines. *IEEE Intelligent Systems*, 13(4):18–28, 1998.
- [Hig90] Nicholas J. Higham. Analysis of the Cholesky decomposition of a semi-definite matrix. In M. G. Cox and S. J. Hammarling, editors, *Reliable Numerical Computation*, pages 161–185. Oxford University Press, Oxford, UK, 1990.
- [Hou64] Alston S. Householder. *The Theory of Matrices in Numerical Analysis*. Blaisdell, New York, NY, USA, 1964. Reprinted by Dover, New York, 1975.
- [HZKM03] Stefan Harmeling, Andreas Ziehe, Motoaki Kawanabe, and Klaus-Robert Müller. Kernel-based nonlinear blind source separation. *Neural Computation*, 15(5):1089–1124, 2003.

- [JO07a] Jin Hyuk Jung and Dianne P. O’Leary. Exploiting structure of symmetric or triangular matrices on a GPU. In *Proceedings of First Workshop on GPGPU*, Boston, MA, USA, October 2007.
- [JO07b] Jin Hyuk Jung and Dianne P. O’Leary. Implementing an interior point method for linear programs on a CPU-GPU system. *Submitted for publication*, 2007.
- [Joa99] Thorsten Joachims. Making large-scale SVM learning practical. In Bernhard Schölkopf, Chris Burges, and Alex Smola, editors, *Advances in Kernel Methods: Support Vector Learning*, pages 169–184. MIT Press, Cambridge, MA, USA, 1999.
- [JOT07] Jin Hyuk Jung, Dianne P. O’Leary, and André L. Tits. Adaptive constraint reduction for support vector machines. In preparation, 2007.
- [Kar84] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4):373–395, 1984.
- [KMY89] Masakazu Kojima, Shinji Mizuno, and Akiko Yoshise. A polynomial-time algorithm for a class of linear complementary problems. *Mathematical Programming*, 44(1):1–26, 1989.
- [LLW04] Yoonkyung Lee, Yi Lin, and Grace Wahba. Multicategory support vector machines: Theory and application to the classification of microarray

data and satellite radiance data. *Journal of the American Statistical Association*, 99:67–81, January 2004.

[LS96] R. B. Lehoucq and D. C. Sorensen. Deflation techniques for an implicitly restarted Arnoldi iteration. *SIAM Journal on Matrix Analysis and Applications*, 17(4):789–821, 1996.

[LS98] Zhi-Quan Luo and Jie Sun. An analytic center based column generation algorithm for convex quadratic feasibility problems. *SIAM Journal on Optimization*, 9(1):217–235, 1998.

[MA89a] Renato D.C. Monteiro and Ilan Adler. Interior path following primal-dual algorithms. Part I: Linear programming. *Mathematical Programming*, 44(1):27–41, 1989.

[MA89b] Renato D.C. Monteiro and Ilan Adler. Interior path following primal-dual algorithms. Part II: Convex quadratic programming. *Mathematical Programming*, 44(1):43–66, 1989.

[Meh92] Sanjay Mehrotra. On the implementation of a primal-dual interior point method. *SIAM Journal on Optimization*, 2(4):575–601, November 1992.

[NW00] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, New York, NY, USA, 2000.

- [OFG97] Edgar Osuna, Robert Freund, and Federico Girosi. An improved training algorithm for support vector machines. In *Proceedings of IEEE Workshop on Neural Networks and Signal Processing*, Piscataway, NJ, USA, 1997. IEEE Press.
- [Pla99] John C. Platt. Fast training of support vector machines using sequential minimal optimization. In Bernhard Schölkopf, Chris Burges, and Alex Smola, editors, *Advances in Kernel Methods: Support Vector Learning*, pages 185–208. MIT Press, Cambridge, MA, USA, 1999.
- [Roc72] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, NJ, USA, 1972.
- [RYDG05] V. C. Raykar, C. Yang, R. Duraiswami, and N. Gumerov. Fast computation of sums of Gaussians in high dimensions. Technical Report CS-TR-4767, Department of Computer Science, University of Maryland, College Park, 2005. [http://www.umiacs.umd.edu/~vikas/Software/IFGT/IFGT\\_code.htm](http://www.umiacs.umd.edu/~vikas/Software/IFGT/IFGT_code.htm).
- [Saa03] Yousef Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, Philadelphia, PA, USA, 2nd edition, 2003.
- [SBS99] Bernhard Schölkopf, Chris Burges, and Alex Smola. Introduction to support vector learning. In Bernhard Schölkopf, Chris Burges, and Alex

- Smola, editors, *Advances in Kernel Methods: Support Vector Learning*, pages 1–16. MIT Press, Cambridge, MA, USA, 1999.
- [SMB<sup>+</sup>99] Bernhard Schölkopf, Sebastian Mika, Chris J. C. Burges, Philipp Knirsch, Klaus-Robert Müller, Gunnar Rätsch, and Alex J. Smola. Input space versus feature space in kernel-based methods. *IEEE Transactions on Neural Networks*, 5(10):1000–1017, 1999.
- [SS00] Alex J. Smola and Bernhard Schölkopf. Sparse greedy matrix approximation for machine learning. In *Proceedings of 17th International Conference on Machine Learning*, pages 911–918. Morgan Kaufmann, San Francisco, CA, 2000.
- [SS01] Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.
- [TA77] Andrey N. Tikhonov and Vasilii Y. Arsenin. *Solutions of Ill-Posed Problems*. John Wiley & Sons, Washington, DC, USA, 1977.
- [TAW06] André L. Tits, P.-A. Absil, and William P. Woessner. Constraint reduction for linear programs with many inequality constraints. *SIAM Journal on Optimization*, 17(1):119–146, 2006.

- [Ton93] Kaoru Tone. An active-set strategy in an interior point method for linear programming. *Mathematical Programming*, 59(3):345–360, 1993.
- [TZ94] A. L. Tits and J. L. Zhou. A simple, quadratically convergent interior point algorithm for linear programming and convex quadratic programming. In W. W. Hager, D. W. Hearn, and P. M. Pardalos, editors, *Large Scale Optimization: State of the Art*, pages 411–427. Kluwer Academic Publishers, 1994.
- [Wat99] G. A. Watson. Choice of norms for data fitting and function approximation. *Acta Numerica*, pages 337–376, January 1999.
- [WG07] Kristian Woodsend and Jacek Gondzio. Exploiting separability in large-scale support vector machine training. Technical Report MS-07-002, School of Mathematics, University of Edinburgh The Kings Buildings, Edinburgh, EH9 3JZ, UK, August 2007. [http://www.optimization-online.org/DB\\_HTML/2007/08/1750.html](http://www.optimization-online.org/DB_HTML/2007/08/1750.html).
- [WNT07] Luke Winternitz, Stacey Nicholls, André L. Tits, and Dianne P. O’Leary. Analysis of a constraint reduced variant of Mehrotra’s predictor-corrector algorithm. *Submitted for publication*, 2007.
- [WO00] Weichung Wang and Dianne P. O’Leary. Adaptive use of iterative meth-

- ods in predictor-corrector interior point methods for linear programming. *Numerical Algorithms*, 25(1–4):387–406, 2000.
- [Wri92] Margaret H. Wright. Interior methods for constrained optimization. *Acta Numerica*, pages 341–407, 1992.
- [Wri97] Stephen J. Wright. *Primal-Dual Interior-Point Methods*. SIAM, Philadelphia, PA, USA, 1997.
- [WS01] Christopher K. I. Williams and Matthias Seeger. Using the Nyström method to speed up kernel machines. In T. Leen, T. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 682–688. MIT Press, 2001.
- [YDD05] Changjiang Yang, Ramani Duraiswami, and Larry Davis. Efficient kernel machines using the improved fast Gauss transform. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1561–1568. MIT Press, Cambridge, MA, USA, 2005.
- [Ye90] Y. Ye. A “build-down” scheme for linear programming. *Mathematical Programming*, 46(1):61–72, 1990.
- [Ye91] Yinyu Ye. An  $O(n^3L)$  potential reduction algorithm for linear programming. *Mathematical Programming*, 50:239–258, 1991.

- [Ye92] Yinyu Ye. A potential reduction algorithm allowing column generation. *SIAM Journal on Optimization*, 2(1):7–20, February 1992.
- [Zha96] Y. Zhang. Solving large-scale linear programs by interior-point methods under the MATLAB environment. Technical Report 96–01, Department of Mathematics and Statistics, University of Maryland Baltimore County, Baltimore, MD, USA, 1996.
- [ZZ95] Yin Zhang and Detong Zhang. On polynomiality of the Mehrotra-type predictor-corrector interior-point algorithms. *Mathematical Programming*, 68(3):303–318, 1995.