Vertical and Orthogonal L_1 Linear Approximation: Analysis and Algorithms

Peter J. Yamamoto
School of Computer Science
McGill University, Montréal
December, 1988

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Master of Science.

Abstract

This thesis presents algorithms for approximating a set of n points by a line which minimizes the L_1 norm of vertical and orthogonal distances. The algorithms find exact solutions based upon geometric properties of the problems as opposed to approximate solutions based upon existing numerical techniques. The algorithmic complexity of these problems appears not to have been investigated. The first result is an $O(n)$ optimal time algorithm for the weighted vertical problem based upon a modified multidimensional search technique which extends the applicability of the basic technique to a wider class of problems. Second, an $O(n^{1.5}\log^2 n)$ algorithm is presented for the unweighted orthogonal problem, and an $O(n^2)$ algorithm is presented for the weighted orthogonal problem; both algorithms provide an interesting application of the (weighted) k-belts of an arrangement of lines. Also, an $\Omega(n \log n)$ lower bound for the orthogonal L_1 problem is shown under a certain model of computation.

Résumé

Ce mémoire présente des algorithmes pour l'approximation d'un ensemble de n points par une droite minimisant la norme L_1 des distances verticales et orthogonales. Les algorithmes trouvent des solutions exactes basées sur les propriétés géométriques des problèmes par opposition aux solutions approximatives basées sur des techniques numériques existantes. La complexité algorithmique de ces problèmes ne semble pas avoir été étudiée. Le premier résultat est un algorithme prenant un temps optimal $O(n)$ pour le problème vertical pondéré basé sur une modification d'une technique de recherche multi-dimensionelle qui rend possible l'application de la technique de base à une plus vaste classe de problèmes. En second lieu, un algorithme $O(n^{1.5}\log^2 n)$ est présenté pour le problème orthogonal non-pondéré ainsi qu'un algorithme $O(n^2)$ pour le problème orthogonal pondéré; les deux constituent une application intéressante des "k-rubans" (pondérés) d'un arrangement de droites. Egalement, une borne inférieure de $\Omega(n \log n)$ pour le problème orthogonal L_1 est démontrée avec un certain modèle de calcul.

Preface

This preface serves two purposes. First, it gives credit to where it is due, since the results presented in this thesis represent a collection of results of research over a period of over one year. Second, it serves to give thanks to those people who helped make the thesis possible in one way or another.

This thesis represents the results of research performed while I was attending Kyushu University. The research involved four people, Hiroshi Imai, Keiko Imai, Kenji Kato, and myself. Hiroshi Imai was the supervisor for Kenji Kato and myself and his contribution to all the results goes without mention (but with much credit). Hiroshi and Keiko Imai independently produced the $\Omega(n \log n)$ lower bound for the orthogonal L_1 problem presented in Section 6.4. Kenji Kato and myself worked on sorting out the details for the vertical L_1 algorithm. Kato gave the proof for the convexity of the L_1 norm. I independently made sure that the individual steps of the algorithm were correct by proofs based on the convexity of the L_1 norm, although at times we sat and tried to convince each other that a particular idea worked or did not work. Kato gave a result for determining the minimum along a nonvertical line in the (a, b)-plane by considering the problem in the (x, y)-plane [IKY87]. However, for the thesis I presented a solution which works in the (a, b)-plane (Section 5). The results on the two-dimensional orthogonal L_1 problem are due to Hirsohi Imai and myself. I thank Emo Welzl for supplying the worst case weighted median-belt example in Section 6 which implied that our algorithm was optimal in terms of worst case complexity. I also thank Jiří Matoušek and Micha Sharir for communicating their unpublished results. I thank Chris Howson for helping with the proofreading.

Speaking of credit, I must also thank sources of financial income which are very important for the student to be able to apply her/himself to their research. First, I acknowledge a research bursary from the FCAR (Fondations pour les Chercheurs de l'aide pour les Recherches) during my first year. Second, I acknowledge a generous scholarship from the Ministry of Education (Monbusho) of the Japanese government which made my life in a foreign world free from financial worries. I also owe thanks to my parents who supported me at other times.

I have two people to thank the most for my accomplishments: my parents. I have to thank two people who provided me with the opportunity to write this thesis: Professor Yasuhiro Kambayashi of Kyushu University, Japan, and Professor Monroe "Monte" Newborn of McGill University, Canada. I owe giri to Professor Hiroshi Imai, my supervisor, for almost everything related to this thesis; the exceptions are the mistakes, which are all mine. I also have to give thanks to Professor Kambayashi and Professor Imai for their education, help, and support in Japanese life, and apologies for causing any problems therein. They also deserve special thanks for the encouragement and support of my research.

I owe thanks to many of the students in Imai-ken, Kambayashi-ken, and Ushijima-ken. I would like to give particular thanks to Hirofumi Amano, Hiromi Aonuma, Masatoshi Arikawa, Tetsuya Furukawa, Tetsuro Kakeshita, Kenji Katoh, and Yoshida Sonoda, for their help in coping with everyday life in Japan. I thank all the students for their friendship and support. I also owe thanks to many other people who opened their hearts and homes to me during my stay. In particular, I thank Dr. Matsuoka for his help and encouragement.

On the other side of the ocean, I owe thanks to Gilles Pésant for help with the French translation of the abstract. I would also like to thank Lise Brisebois for her assistance while I was both at McGill and at Kyushu University.

Table of Contents

Table of Contents for Figures

1. Introduction

This thesis is concerned with a problem which is encountered in almost all of the applied sciences: *the line fitting problem.* There are many variations of the general problem and the terminology varies between different fields of research. Simply stated, the input is a set of n points, and the output is a line which *best fits* the set of points according to some given criterion. The origins of the problem are related to data reduction, representing the information from a large point set in the form of a line. Applications range from clustering to signal processing. While statisticians and econometricians have produced the most literature on the subject, the problem is also of fundamental importance in transportation science where it is known as the *linear facility location problem*, or simply the *location problem.*

In particular, this thesis is concerned with the computational complexity of the L_1 linear approximation problem, or simply the L_1 *problem.* In general, statisticians have not been concerned with the amount of time required to solve their problems; they are concerned with the theoretical aspects of their results. Shamos [S76] may have been the first to analyze the computational complexity of some statistical algorithms and suggest efficient alternative algorithms for traditional methods. The first result presented in this thesis provides an answer to one of the questions raised in [S76]: an optimal, $\Theta(n)$, time algorithm is given for the vertical L_1 linear approximation problem. The next two results, algorithms for the unweighted and weighted orthogonal L_1 linear approximation problem, provide a direct application for (and generalize) a relatively recent concept in combinatorial geometry: the (weighted) k-belt [EW86]. Finally, a lower bound is given for the orthogonal L_1 linear approximation problem.

1.1. The Linear Approximation Problem

The problem, in its general form, may be stated as follows (see **Figure 1.1**). The input to the problem consists of two parts. The first part is a set, S, of n points in the (x, y)-plane:

$$S = \{\, p_i : (x_i, y_i), i = 1, \ldots, n \,\}.$$

The interpretation of the coordinates varies according to the application. In the location problem, the points represent the location of *feeders* (i.e. cities); the problem is to determine a transportation facility, (usually modeled by a straight line called the *trunk line*) which minimizes a given transportation cost from the feeders to the trunk line (*feeder routes*). From a mathematical viewpoint, x is considered as the independent variable and y is considered as the dependent variable defined by an unknown function $y(x)$ such that

$$y_i = y(x_i).$$

The general linear problem is to approximate the function $y(x)$ by a linear combination (hence, the term *linear* approximation) of so called *basis functions*, $\phi_j(x)$, of the

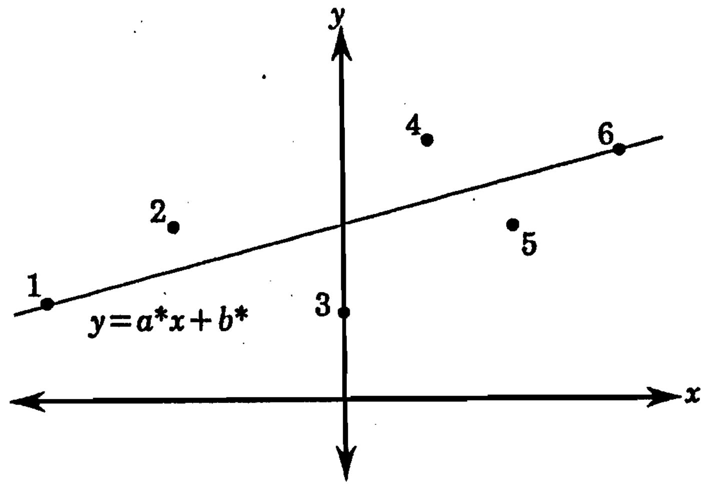


Figure 1.1. The line fitting problem.

independent variable, x,

$$y(x) \approx c_0\phi_0(x) + c_1\phi_1(x) + c_2\phi_2(x) + \cdots + c_m\phi_m(x),$$

where c_i represents the ith coefficient. In other words, the problem is to find values for the parameters, c_i, such that the approximating function best fits the data according to some given criterion. The second part defines the type of problem by specifying $\phi_j(x)$ and an error criterion, $E(c)$, where c represents the parameters c_i, $(i = 0, \ldots, m)$.

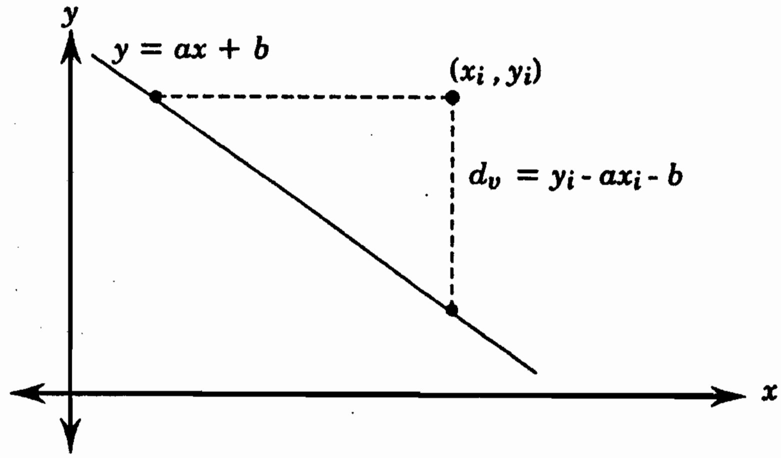
The goal of the linear approximation problem is to minimize the given error criterion with respect to the approximating function $\phi(x)$ and the set of points S. The linear approximation problem is characterized by the choice of the approximating function and the error criterion. Another characteristic of the problem is the weighting of the points p_i by corresponding weights w_i; the unweighted case corresponds to $w_i = 1$, $(i = 1, \ldots, n)$. Without loss of generality, we assume that all weights are positive valued, $w_i > 0$.

The most common choice of $\phi_j(x)$ is x^j, which implies that the data fits some polynomial curve. This thesis is concerned with approximating the data points by a polynomial of degree 1, a line $l : y = c_0 + c_1 x$; hence, the problem is called the *line fitting problem*. For convenience, let $a \equiv c_1$ and $b \equiv c_0$ represent the slope and y-intercept, respectively, of l. The problem may now be stated as the approximation of a set, S, of points by a line $y = ax + b$ according to a specified error criterion.

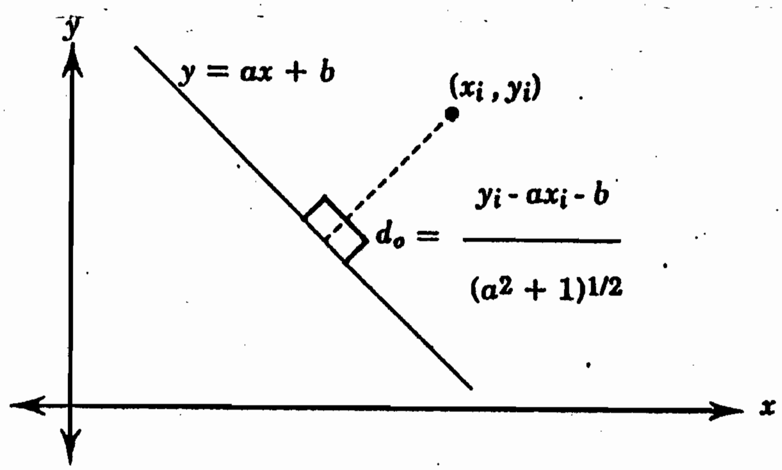
The most common choice of error criterion, or function, comes from the family of L_p norms:

$$E(c) \equiv L_p(c) = \left(\sum_{i=1}^{n} w_i |e_i(c)|^p \right)^{1/p},$$

where $e_i(c)$ represents the error from the ith point. The most common choices of $e_i(c)$, or $e_i(a, b)$, are the vertical, d_v, and orthogonal, d_o, distances from the point p_i to the line



(a)



(b)

Figure 1.2. (a) Vertical distance from a point to a line. (b) Orthogonal distance from a point to a line.

$y = ax + b$ (see **Figure 1.2**). In the weighted case, the weighting is taken into account by multiplying the (L_p adjusted) error from the ith point, $|e_i(c)|^p$, by the corresponding weight, w_i.

In particular, three values of p are usually considered in the linear approximation problem, $p = 1$, 2, and ∞:

$$L_1: \quad \sum_{i=1}^{n} w_i |e_i(a,b)|,$$

$$L_2: \quad \sum_{i=1}^{n} w_i |e_i(a,b)|^2$$

$$L_\infty: \quad \max_i \{|e_i(a,b)|\}.$$

The problem of minimizing a particular L_p norm, with respect to a point set and a line, will be called the *L_p method.* For example, minimizing the L_1 norm is simply called the (vertical or orthogonal) L_1 method. Several methods have adopted special names from the mathematical and statistical literature. Minimizing the L_2 norm with respect to vertical distances is the well known Least Squares method; note that the L_2 norm may be minimized without the pth root since the optimal lines are the same (but the optimal value may differ) [R64]. Minimizing the L_∞ norm, also known as the *Chebyshev norm*, with respect to vertical distances, is called the *Chebyshev method.* Unless otherwise specified, we assume that there are weights associated with the points.

In the statistical literature, the vertical distance (see **Figure 1.2(a)**),

$$e_i(a,b) = |y_i - (ax_i + b)|,$$

is almost exclusively used (the *vertical problem*). In the vertical problem, the independent variable, x, often represents time while the dependent variable, y, represents experimental measurements which may be subject to error. When the vertical distance is used as the error function, the optimal solutions to the three methods mentioned above have distinct characteristics which may determine the appropriate method for a particular application. Despite that fact, in practice, the Least Squares method has been almost exclusively used for the linear approximation problem.

Using the orthogonal, or Euclidean, distance (see **Figure 1.2(b)**),

$$e_i(a,b) = \frac{|y_i - (ax_i + b)|}{\sqrt{a^2+1}},$$

(the *orthogonal, or Euclidean, problem*) is of particular interest from the point of view of transportation science since the orthogonal distance represents the shortest path from a point (possibly representing a warehouse or city) to a line (representing a planned transportation route) in the location problem. In the location problem, the L_1 and L_∞ methods are the most practical of the three methods. The L_1 method minimizes the total distance from the points to the line which is often directly proportional to the cost of transporting goods along the route. The L_∞ method minimizes the maximum distance from the route to any point; such a criterion may be of interest, for example, in terms of locating emergency services. Use of the orthogonal distance in statistical applications represents error in both the independent and dependent variables.

1.2. Motivation

The L_2 norm has been well researched and a basic theory on L_2 approximation exists. As mentioned above, the Least Squares (vertical L_2) method enjoys the most popularity and use in the linear approximation problem. The Least Squares method may be solved in $\Theta(n)$ optimal time and, under certain assumptions about the distribution of points, the solution represents the intuitively *best* solution relative to the other two methods.

The orthogonal L_2 method has not received as much attention as the Least Squares method.

The vertical L_∞ method may be formulated as a convex linear programming problem and many techniques have been developed based on such a formulation. Megiddo [Me84] notes that the vertical L_∞ problem can be solved in $\Theta(n)$ optimal time by applying his multidimensional search technique developed for linear programming in linear time. Lee and Wu [LW86] considered the orthogonal L_∞ problem and some of its variations and developed optimal and efficient algorithms based on computational geometry techniques. The L_∞ method has been well researched from the practical viewpoint: efficient algorithms exist for the basic problem and its variations. The notable point about those algorithms ([Me84] and [LW86]) is that they are analytical: if a solution exists, then the algorithm will find the solution within a specified time bound. That property does not hold for many of the numerical techniques used to solve the problems.

The vertical L_1 problem may also be formulated as a convex linear programming problem and there exist many general and specialized techniques for solving the problem. The mathematical community has recently generated a renewed interest in L_1 approximation theory with the objective of promoting the practical use of the L_1 method. That research has resulted in many specialized algorithms; however, as noted in [BS80], the number of algorithms widely available for efficiently solving the L_1 method is very small and their performance may vary according to the distribution of the data points. Although both the L_1 and L_∞ problems may be formulated as convex linear programming problems, Megiddo's technique cannot be directly applied to the L_1 problem to produce an optimal algorithm. The orthogonal L_1 problem has appeared recently in the form of a location problem [MN80] and an $O(n^3)$ brute force algorithm was given.

Shamos [S76] was the only reference which provided an analysis of an algorithm for solving the vertical L_1 problem in $O(n^2)$ time and $O(n)$ space. [S76] notes that until a better algorithm is developed, iterative methods will remain preferable and also that no non-trivial lower bound is known. This thesis answers Shamos' problem by providing an optimal algorithm for the vertical L_1 problem based upon a modification of Megiddo's technique. The thesis also raises similar problems by providing efficient, but probably not optimal, algorithms for the unweighted and weighted orthogonal L_1 problems. All three algorithms find exact solutions by efficient search techniques as opposed to converging towards the solution by iterative techniques.

1.3. Contents and Summary of Results

In this thesis, the vertical and orthogonal L_1 problems are considered. First, a history of the L_1 problem is presented. Also, to provide a wider background, the histories of the L_2 and L_∞ problems are briefly discussed. Second, algorithms for three general cases of the L_1 problem are presented: the (weighted) vertical L_1 problem, the unweighted orthogonal L_1 problem, and the weighted orthogonal L_1 problem.

Section 2 presents the historical background. **Section 3** presents an analysis of the L_1, L_2, and L_∞ methods. **Section 4** provides geometric preliminaries applicable to the following sections. **Section 5** discusses the vertical L_1 method. The major result is an optimal, $\Theta(n)$, time algorithm for the vertical L_1 method in the plane. The algorithm is based on a modification of Megiddo's multidimensional search technique which may be applicable to similarly structured problems to produce optimal time algorithms. **Section 6** discusses the orthogonal L_1 method and relates the problem to (weighted) k-belts. The results are an $O(n^{1.5}\log^2 n)$ time and $O(n)$ space algorithm for the unweighted orthogonal L_1 method based on a sophisticated plane sweep, an $O(n^2)$ time and $O(n)$ space algorithm for the weighted orthogonal L_1 method based on a topological sweep algorithm, and an $\Omega(n \log n)$ lower bound (based on a particular computational model) for the orthogonal L_1 method in the plane. **Section 7**, the conclusion, summarizes the theoretical and practical aspects of the results presented. Open problems and future research are also mentioned.

2. History

This thesis is primarily concerned with the computational complexity of the L_1 linear approximation problem; however, since there is very little history in that area, a general history of the problem is given. This section summarizes the history of the L_1 linear approximation problem. Also, the histories of the L_2 and L_∞ problems are briefly discussed.

Three starting points for sources of historical information are Gentle's "Least absolute values estimation: An introduction" [G77] (an introduction to a special issue on L_1 estimation in *Communications in Statistics-Simulation and Computation*), Chvátal's "Linear Programming" [C83], and Dodge's "An introduction to L_1-norm based statistical data analysis" [D87b] (an introduction to three special issues on L_1 estimation in *Computational Statistics and data Analysis*). Harter provides an extensive historical account of linear model estimation based on Least Squares and alternative methods [H74a, H74b,H75a,H75b,H75c,H76].

2.1. The L_p Linear Approximation Problem

The linear approximation problem was motivated by the development of celestial mechanics in the eighteenth century in the form of approximations of data on star movements. Boscovitch gave a geometric method for solving the L_1 problem sometime between 1755 and 1757. The common method of solving the L_2 problem, the Least Squares method, dates back to the beginning of the nineteenth century although whether Gauss or Legendre deserves the credit for the invention of Least Squares remains unclear [St81]. Linear programming formulations for the L_1 and L_∞ problems and methods for solving them were proposed in the 1820's by J.B.J. Fourier. Wagner [W59] provides a summary of the linear programming approach up until 1959. Since 1959, many algorithms based on the linear programming formulation have been presented; however, most of those algorithms use a simplex method type approach to solve the problem.

Dodge notes that the Least Squares method has been widely used by statisticians for many years due to mathematical convenience, ease of computation, and since the method determines the most likely solution under strictly Gaussian parametric models. However, if the data points do not satisfy the Gaussian model, then the Least Squares solution may be intolerably skewed by *outliers* (also called noise or errors) in the data points (see next section). The term *robustness* was first used by Box to describe approximation methods which are not easily influenced by outliers in the data. The L_1 method is one of the most popular robust methods and provides a good alternative to the Least Squares method.

Presently, the linear approximation problem may be found in applications in almost all of the applied sciences. In particular, a large amount of the literature comes from statistics, econometrics, and biometrics, which consider the problem from a statistical point of view, and from operations research, transportation science, and geographical science, which consider the problem as a location problem. Shamos [S76] notes that the analysis of statistical algorithms remains largely ignored. In the last few years, however, several results from computational geometry have presented optimal or efficient algorithms and their analysis for the L_∞ and other statistical problems. Unfortunately, it seems that, to a large extent, the researchers from the different fields are not completely aware of each others work; for example, we found no reference to Morris and Norback's work [MN80,NM80,MN83] in the statistical literature even though their work appears in mathematical journals. Although we have not examined the L_∞ approximation literature as well as the L_1 approximation literature, Lee and Wu's [LW86] significant results on the L_∞ problem and its variations also appears to suffer from a lack of interdisciplinary communication. Even in articles which claim to be concerned with the complexity of statistical algorithms, we have not found a reference to Shamos' [S76] work.

2.2. The L_1 Method

An excellent source of references for the L_1 method is the above mentioned [G77]. The notable feature of the article is that the author divides the references into important aspects of L_1 approximation: history, robustness, nonuniqueness, basic theory, properties, and finally computational aspects of the problem. Dodge [D87b] also provides a well categorized list of references; although Dodge does not provide as many references as Gentle, he does provide a brief description of the contents of the referenced articles. Bloomfield and Steiger [BS80] present a brief history of algorithmic approaches to the vertical L_1 problem.

The earliest reference to linear approximation seems to be a geometric method for solving a special L_1 approximation problem proposed by R.J. Boscovitch sometime between 1755 and 1757. P.S. Laplace provided an analytic derivation of Boscovitch's method and further theoretical work on L_1 approximation was performed by C.F. Gauss. In the 1820's, J.B.J. Fourier outlined a linear programming formulation of L_1 and L_∞ approximation problems, and suggested a simplex method for solving them. However, L_1 approximation remained impractical until the advancement of linear programming and the computer code for its implementation.

Chvătal [C83] discusses the computation of best vertical L_1 and L_∞ approximations based on the paper by H.M. Wagner [W59]. [W59] provides a summary of the linear programming approach up until 1959. Since 1959, many algorithms based on the linear programming formulation have been presented; however, Chvătal notes that, during the period from 1974 to 1983, the best algorithm for solving the vertical L_1 problem seems to be the algorithm proposed by P.Bloomfield and W.L. Steiger [BS80].

While there existed many published algorithms for the vertical L_1 method by 1977, Gentle [G77] notes that one impediment to the applied statistician's use of L_1 approximation is the lack of such procedures in the statistical program packages. Ten years later, with many more published algorithms, a comparison of five "openly available" codes notes that two of the implementations were provided by the authors of the algorithms and one those failed to converge to a solution enough times to warrant its exclusion from the final survey [GSN88].

Even though the linear programming approaches tend to be iterative and manipulate large matrices, undesirable characteristics in terms of computational complexity, very few alternative approaches appear in the literature. Soliman, Christensen, and Rouhi [SoCR88] present a noniterative technique which appears simple to apply although they provide no computational complexity analysis. The lack of such analysis is typical of the statistical literature with one exception: Bloomfield and Steiger [BS80] provide a partial analysis of their algorithm versus ordinary Least Squares. Their final analysis rests on comparison of CPU times in which case they claim $O(n)$ complexity. One source of algorithms which addresses both the time and space complexity issues is computational geometry. [S76] provides an $O(n^2)$ time and $O(n)$ space algorithm for the vertical L_1 problem and notes that a lower bound remains an open problem. This thesis provides an optimal, $\Theta(n)$, time and $O(n)$ space algorithm thus closing that problem. The complexity of the orthogonal L_1 problem, however, remains an open problem.

The literature on the orthogonal L_1 problem is not nearly as extensive as that of the vertical L_1 problem. In the three special issues on L_1 estimation in *Computational Statistics and Data Analysis* [D87b,D88a,D88b], there is only one article on the orthogonal L_1 problem: [N88]. In that article, Nyquist references only two other results on orthogonal estimation. One of them, Wald [Wa40], was the earliest reference to the orthogonal L_1 problem that we found. A noticeable omission in [N88] is Späth and Watson's [SW87] "On orthogonal linear l_1 approximation" which characterizes best approximations, discusses robustness, and provides an algorithm and the results of its applications to some data sets.

One of the original references used in the research for this thesis did not come from the statistical literature: the article is "A Simple Approach to Linear Facility Location," by Morris and Norback [MN80]. Morris and Norback take a discrete geometric approach to solving the problem: since there exists an optimal solution to the L_1 method which is incident with two data points, their approach was (basicly) to investigate each of the $O(n^2)$ possible solutions. Their research led to at least two related papers [NM80,MN83] and provided some of the motivation for the research presented in this paper.

2.3. The L_2 Method

Plackett [P72] details the dispute over the priority of the independent discovery of the Least Squares method by Gauss and by Legendre. Stigler [St81] provides circumstantial evidence in favor of Gauss but credits Legendre for "crystalizing the idea in a form that caught the mathematical public's eye." Legendre was the first to publish the method in 1805; he also included an example application. However, Gauss is credited with fully exploring the method.

As mentioned above, Harter provides an extensive historical account of the Least Squares method. Those articles form a five part chronological history of the Least squares method dating from 1632 to 1974; the sixth article is a summary and subject and author indices to the references. There are also several articles which compare the L_1 and L_2 methods, such as [F87a].

Almost all college students learn how to solve the Least Squares method in one form or another. For an exposition on the computational aspects of the Least Squares method, almost any numerical analysis book will contain the necessary information if not actually devoting a chapter to the method such as in [PFTV86]. [PFTV86] explains the Least Squares method as interpreted by statisticians and presents several solution techniques. For a statistical discussion, any college level statistics text book will contain a discussion of the Least Squares method. The amount of literature on the orthogonal L_2 problem is relatively limited compared to the amount of literature on the vertical L_2 problem.

2.4. The L_∞ Method

Rice [R64] provides a brief historical account of the Chebyshev norm. The Chebyshev norm was proposed by Laplace in 1799 and Fourier studied a similar problem in 1824. The name of the norm is derived from P.L. Tchebycheff (many different spellings appear in the literature) who was the first to extensively study the problem (from around 1850-1897). The problem remained stagnant during the period from 1915 until the introduction of high speed computers after the Second World War. Farebrother presents "The historical development of the L_1 and L_∞ estimation procedures" in [F87b]. Wagner [W59] summarizes the development of the linear programming formulation of the vertical L_∞ problem. Appa and Smith [AS73] discuss some theoretical properties of the L_1 and L_∞ methods and consider the applicability of those methods for use in econometric work. Rice [R64] provides a comprehensive account of the theory of Chebyshev approximation.

Megiddo [Me84] notes that the vertical L_∞ problem may be solved in $\Theta(n)$ optimal time since it can be formulated as a linear programming problem and solved by his multidimensional search technique. Lee and Wu consider the orthogonal L_∞ problem and some of its variations from a location problem point of view. They present an optimal $\Theta(n \log n)$ algorithm for the unweighted case and an $O(n^2 \log n)$ algorithm for the weighted case; they also provide optimal algorithms for two variations of the basic problem (the optimality is based upon the algebraic computational model of Ben-Or [B-O83]). Lee and Wu state that the motivation for their work was provided by Morris and Norback [NM80]. Surprisingly, Lee and Wu have no reference to the statistical nature of the basic problem; similarly, the statistical community appears surprisingly unaware of their work, considering the simplicity and efficiency of their algorithms. Houle and Roberts [HR88] and Hiroshi and Keiko Imai [II88], have independently developed an $O(n \log n)$ algorithm for the weighted orthogonal L_∞ problem in two dimensions. The algorithm is based on transforming the problem into a three dimensional convex hull problem and provides a significant improvement over Lee and Wu's result.

3. Analysis

This section discusses certain characteristics of the L_1, L_2, and L_∞ methods, although the emphasis of the discussion is on the L_1 method. The objective of the discussion is an analysis of the methods; note, however, that the term "analysis" may have a broad range of meanings. In the context of this thesis, we use the term to describe the process of analyzing the problem in the context of determining its computational complexity. Of course, such a process implies, for the most part, reviewing formal analyses of the problem. In the case of the L_p approximation methods, the analyses fall under two different categories: mathematical and statistical (strictly speaking, statistics is not considered a branch of mathematics). Note that these two viewpoints are very much different. From the viewpoint of mathematical analysis, Rice [R64] notes that a rather complete theory has evolved for L_1 approximation. In contrast, Dodge [D87a] points out that, although a great deal of research has been performed on L_1 approximation methods, there is a need to unify and communicate the theory.

There were two primary sources of information. Rice [R64] gives a very good (and widely referenced) mathematical analysis of the L_p problems. On the other hand, Press *et al* [PFTV86] give a practical, statistical analysis (practical in the sense that they are concerned with the computation of solutions as well as the theory). Since much of their discussions goes beyond the context of this thesis, we outline the contents of the discussions and then briefly state the relevant results. We then continue by summarizing how those results are used in analyzing the computational complexity of the methods. We also look at the applicability of the methods to certain problems.

3.1. Mathematical Analysis

Rice [R64] is primarily concerned with the approximation of continuous functions and only considers the vertical problems in detail (hence, unless otherwise specified, the discussion in this section refers to the vertical problem). The first step in the analysis is choosing an approximating function and an error criterion. Rice considers many types of approximating functions and hence goes beyond the context of this thesis. On the other hand, he points out some of the general factors involved in choosing an error function. The most important factor in choosing the error criterion is the application, although the choice of approximating function may also effect the selection. In particular, Rice compares the L_1, L_2, and L_∞ norms and discusses the existence and uniqueness of optimal approximations.

Rice observes that the theory of the L_2 and L_∞ approximation of a continuous function may be generalized to the approximation of a discrete set of points. On the other hand, the theory of the L_1 approximation of a continuous function is basicly different from the approximation of a discrete set of points. Even so, the analysis may be extended to the approximation of a finite point set and Rice observes that some of the analysis is actually simplified.

The basic difference between the L_1 approximation of a continuous function and a set of points is in the nature of the L_1 norm. Let

$$K = \{ (a,b,c) \mid D(a,b) \leq c \},$$

where $D(a,b)$ represents the value of the L_1 norm. Hence, the L_1 method is equivalent to the geometrical problem of finding the lowest point on K. In the approximation of a finite set of points, the set K is the intersection of hyperplanes and forms a polytope in R^{d+1}, where d is the number of parameters. Rice notes that it is intuitively clear that the lowest point lies at a corner of K: i.e., at an intersection of the hyperplanes.

Morris and Norback [MN80,MN83,NM80] also develop the same result from a slightly different viewpoint. First, [MN80] show that there exists an optimal solution to the weighted orthogonal L_1 problem in the plane which is incident to two of the given data points; we call that result the *incidence property* of an optimal solution. Later, [NM80] generalize the result to higher, d, dimensions and note that an optimal hyperplane exists which is incident to d of the given data points. The equivalence of the results lies in a fundamental theorem of linear programming which states that an optimal solution lies at a vertex of a convex polyhedron, the corner of K; that relationship is explained in **Section 5**.

Morris and Norback [MN80] also prove a second property called the *weighted median property* which states that the absolute difference of the sums of weights of points above and below an optimal line does not exceed the sum of the weights of points on the line. In the unweighted case, this just means that the optimal line divides the data points into about equal size groups above and below the line. Combining the two results provides a nice characterization of an optimal solution which can be exploited to develop efficient geometric algorithms.

3.2. Statistical Analysis

Press *et al* [PFTV86] are primarily concerned with numerical techniques for determining an optimal solution; however, they provide a very readable account of several basic but important statistical properties in linear approximation. Their discussion also goes beyond the context of this thesis, but, in contrast to the mathematical analysis above, they give very sound advice in understanding and avoiding the pitfalls of practical linear approximation.

They devote a large amount of discussion to the concept of the Least Squares method as a *maximum likelihood estimator* and consider the special case of the line fitting problem. They also present a brief discussion of robust estimators. The above two concepts are of interest to the three methods with respect to the application of a particular method to a particular problem. They also discuss more general approximation methods (for example, nonlinear approximation functions) and confidence limits which are beyond the context of this thesis.

Of particular importance to the statistical performance of any of the three methods is the distribution of the measurement errors of the given data points. In statistics, the concept of *maximum likelihood estimation* relates the probability of the data given the parameters to the *likelihood* of the parameters given the data. [PFTV86] gives a very readable account of maximum likelihood estimation in the context of the Least Squares method. Under a strictly Gaussian distribution model, the solution to the Least Squares method is a maximum likelihood estimator. While the Gaussian distribution may be found in numerous applications, if any error is present the Least Squares solution may be intolerably skewed. [BS80] notes that the L_1 method is a maximum likelihood estimator under a double exponential Laplace distribution model. The L_∞ distribution is a maximum likelihood estimator under a uniform distribution model.

Although many applications exist in which the distribution of the measurement errors is known a *priori*, the existence of errors in the data is natural from both a statistical or systematic point of view. Even though such errors may be well understood, different methods behave differently for points which exceed the expected error. The error may be greater than expected either because the actual distribution of the measurement errors is not the assumed distribution, or because there is actually an error in the data. Note that the latter case is present in many applications; a power surge or a miscopied data entry may introduce a data point whose error is far from the assumed distribution. Such points are called *outliers*, or *noise*.

A *robust estimator* is a method which is insensitive to deviations from the assumed population distribution [S78]. Robust estimation is important in problems in which a distribution model cannot be predicted or in which outliers are important. The robustness has important implications to both the statistical and the location problem applications. Since there is a great deal of material on the subject of robustness ([Hu64,Hu72,Hu73,Hu81,Hu87] provides a mathematical view of robustness) the purpose of the presentation here is to provide an example of the effects of outliers on the optimal solutions.

The vertical L_1 method is well known in statistics as a robust estimator: the solution to the L_1 method is not easily influenced by outliers in the data (see **Figure 3.1(a)**); for that reason, the L_1 solution is recognized as a good initial estimate for iterative techniques for other approximation methods. Intuitively, the L_2 method assumes that all the data points are significant; hence, the method is susceptible to the influence of outliers, or error, in the data (see **Figure 3.1(b)**). The L_∞ method minimizes the maximum distance from the points to the line and hence is the most effected by outliers (see **Figure 3.1(c)**).

When the distribution of the data points is not known, but is of importance, the solutions to all three methods may be computed and then compared to help determine some characteristics of the distribution. The L_1 method may be preferable to the Least Squares method in applications in which the distribution of the data points is not known but the presence of outliers is known.

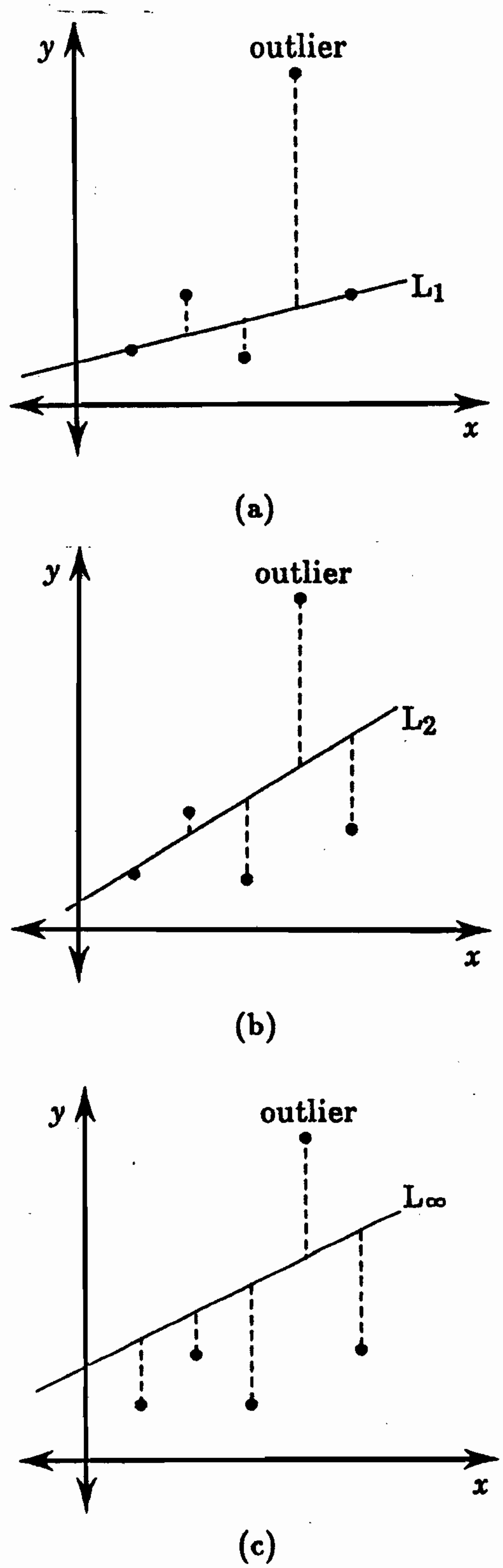


Figure 3.1. (a) Optimal L_1 line is not affected by outlier. (b) Optimal L_2 line is slightly affected by outlier. (c) Optimal L_∞ line is the most affected by the outlier.

3.3. Applications

The robustness of the method has different implications depending on the application. The location problem has an interesting interpretation of the effect of outliers. If the approximating line represents a major transportation route such as a railroad (as considered in [AS73]), then the L_1 method may produce the best answer since the solution will not sacrifice service to many points for the sake of one outlier. The L_1 method may be useful in evaluating the cost of a transportation network since the cost is often proportional to the total distance (the cost of traveling on the trunk line is considered negligible). The orthogonal L_1 method is of particular interest since it represents the minimum Euclidean distance from the points to the line. In the context of emergency services, the maximum distance from the route to a point may be minimized in order to guarantee response time for any point; such an assumption implies the use of the L_∞ method. While the above observations are based on mathematical properties of the solutions, the solutions may be counter intuitive; whether such solutions should be accepted depends on the application.

The importance of not blindly accepting a solution is illustrated by Appa and Smith [AS73] who give examples of data sets whose L_1 or L_∞ solutions are counterintuitive. They give two inputs for which the optimal L_1 solutions do not indicate downward and upward trends in the data, respectively, although such trends are visually perceptible. They note that, in general, in the vertical L_1 problem, any point not on the optimal line may be translated vertically without changing the optimal line as long as the point is not translated across the optimal line. They also note that a point which determines the optimal line may be translated vertically within a range which has both lower and upper bounds without affecting the optimality of the line.

The optimal solution to the L_∞ problem is known to be determined by points on the convex hull of the given points. Appa and Smith note that such a property makes the L_∞ method unattractive for work in econometrics since the optimal solution may depend on relatively few of the data points. They give an example in which all but one of the points lie on a straight line; hence, the convex hull forms a triangle with all but one point on a single edge. The optimal solution, however, does not lie on the edge with most of the points since it is influenced by the lone outlier; in fact, it lies on one of the other edges of the convex hull, thus intersecting only two of the points and giving a very false indication of the linear trend in the data.

3.4. Algorithms

As Rice notes, one of the most important points of a theory of approximation is the knowledge of some characteristic property of best approximations. He describes such properties as the method by which we may distinguish a "best" approximation from one which is merely good. Those properties, however, may also play a fundamental role in the computational complexity of the problem and in the development of algorithms.

The Least Squares method has the nicest characteristic property for a best, or optimal, solution: the formula for the optimal solution, or optimal parameters, may be explicitly given. Price *et al* give two different methods for computing the parameters. The first is an efficient algorithm for the fitting of a straight line which directly computes the values of the parameters. The second is an algorithm for the more general problem which, although less efficient, is considered a failsafe method.

The L_1 and L_∞ methods have received much attention from the algorithmic viewpoint. In terms of computational complexity, the formulation of the problem may be an important consideration. Both the vertical L_1 and vertical L_∞ problems may be formulated as convex linear programming problems, whereas the orthogonal formulations are quite different. The orthogonal L_1 method may be solved by a concave quadratic programming approach [SW87], while the orthogonal L_∞ problem lacks convexity, concavity, and differentiability [MN83]. On the other hand, optimal solutions to the L_1 and L_∞ methods may be geometrically characterized. The significance of such properties from the application side have not been mentioned in the literature although those properties have been exploited in the development of efficient algorithms.

The only difference between the vertical and orthogonal methods is the divisor in the distance function which converts the vertical to the orthogonal distance. However, that divisor destroys the convexity of the vertical problem, and the orthogonal L_1 and L_∞ methods seem to present more difficult problems than the corresponding vertical problems.

The vertical L_1 and L_∞ methods have a multitude of algorithms almost all based on a linear programming approach. However, very little analysis of the performance of those algorithms has been published [G77,GSN88]. In contrast, several analytical results have been produced based upon geometrical properties of the optimal solutions to the particular problem being solved.

Although both the vertical L_1 and vertical L_∞ methods may be formulated as convex linear programming problems, the L_∞ may be solved in $\Theta(n)$ optimal time by Megiddo's technique [Me83] while the L_1 method does not admit a direct application of the technique. The reason that the technique does not apply to the L_1 method is that it assumes the function to be optimized is dependent upon at most a constant number of the constraints. It is well known that at most three points determine the solution to the L_∞ method while the solution to the L_1 method is clearly dependent upon all the constraints due to the mini-sum nature of the problem. However, as illustrated in this thesis, the vertical L_1 problem may also be solved in $\Theta(n)$ optimal time.

There are several characterizations for the optimal orthogonal L_∞ solution; they may be summed up as follows. The optimal line to the L_1 linear approximation problem is at maximum weighted distance from at least three demand points [MN83]. Furthermore, for the unweighted problem, there are two points which determine an optimal solution and which are on the convex hull of the given set of points. Based on those properties, Lee and Wu [LW86] presented an optimal, $\Theta(n \log n)$, time algorithm for the unweighted orthogonal L_∞ problem. Recently, Houle and Roberts [HR88] and Imai and Imai [II88] have independently developed an $\Theta(n \log n)$ problem for the weighted problem which Lee and Wu conjectured to be a more difficult problem.

As mentioned above, the L_1 method has nice properties which characterize an optimal solution. Although Rice considers the vertical problem whereas Morris and Norback consider the orthogonal problem, we note that the incidence and weighted median properties mentioned above apply to both problems. The incidence property implies an $O(n^3)$ brute force algorithm which computes the value of the L_1 norm for a line determined by each pair of points. Although Morris and Norback tried to exploit the weighted median property to improve upon such an algorithm, in complexity terms their algorithm still took $O(n^3)$ time. This thesis takes further advantage of those properties to develop optimal and efficient algorithms for the L_1 methods.

The fundamental difference in the complexity of the vertical and orthogonal L_1 method is illustrated in the optimal $\Theta(n)$ time algorithm for the vertical L_1 problem as compared to the $\Omega(n \log n)$ lower bound for the orthogonal L_1 problem presented in this thesis. We also note that the weighted orthogonal L_1 problem seems to be harder than the unweighted problem, whereas introducing weights in the vertical L_1 problem does not change the computational complexity of the problem. We give an $O(n^{1.5} \log^2 n)$ time algorithm for the unweighted problem and an $O(n^2)$ time for the weighted problem.

Another feature of the geometric algorithms mentioned above is that they give the solution in terms of the data points, as opposed to the iterative techniques which improve an initial guess at each step. For example, the algorithms for the L_1 methods presented in this thesis always provide a solution which is incident to two data points. Similarly, the algorithm of Lee and Wu [LW86] determines a set of three data points from which the optimal solution may be computed for the unweighted L_∞ problem with a similar result for the weighted case. Another paper on linear approximation [ES88], although not L_p based, also applies computational geometry techniques to determine an answer which is dependent on a constant number of the original data points. Furthermore, recent techniques in computational geometry allow the algorithm to monitor the consistency of the intermediate computations [DS88,HHK88]. Since the answer can be given in terms of the input, there is essentially no error in the answer. The fact that optimal solutions determined by the geometric algorithms always satisfy such properties may be of definite interest from the point of view of the location problem. Whether such properties are of interest to statisticians remains to be seen.

4. Geometric Preliminaries

Computational geometry has provided the designer of algorithms with a very powerful set of tools, called *design paradigms* by [E87], which have proven to be useful in the design of algorithms for many different applications. In this thesis, all three algorithms use a paradigm called the *point-line*, or *dual*, *transformation*. As [E87] notes, the transformation of one problem into another cannot create anything new if the transformation maintains a one-to-one mapping; the power of the dual transformation lies in its ability to focus the designer's attention on different aspects of the problem, as will be demonstrated in the three algorithms presented in this thesis.

This section introduces the concept and terminology of the dual transformation and illustrates its application by introducing two other geometric paradigms used in the algorithms. The dual transformation and two of its properties are defined. The basic geometric concepts used in solving the vertical L_1 problem and the orthogonal L_1 problems are introduced. Although the purpose of this section is to introduce the geometric aspects of the problems rather than the problems themselves, some properties of the L_1 problems are introduced here to provide a motivation for the discussion.

4.1. Dual Transformation

Consider a set of points in the (x, y)-plane. The dual transformation, $\mathcal{D}$, maps a point, $p_i : (x_i, y_i)$, in the (x, y)-plane to a line in the (a, b)-plane (see **Figure 4.1**):

$$\mathcal{D}(p_i) \rightarrow l_i : b = -x_i a + y_i.$$

Similarly, $\mathcal{D}$ maps a line $l' : y = a'x + b'$ in the (x, y)-plane to a point in the (a, b)-plane:

$$\mathcal{D}(l') \rightarrow p' : (a', b').$$

The transformation also performs the same mapping from the (a, b)-plane to the (x, y)-plane, hence:

$$\mathcal{D}(l_i) \rightarrow p_i$$

and

$$\mathcal{D}(p') \rightarrow l'.$$

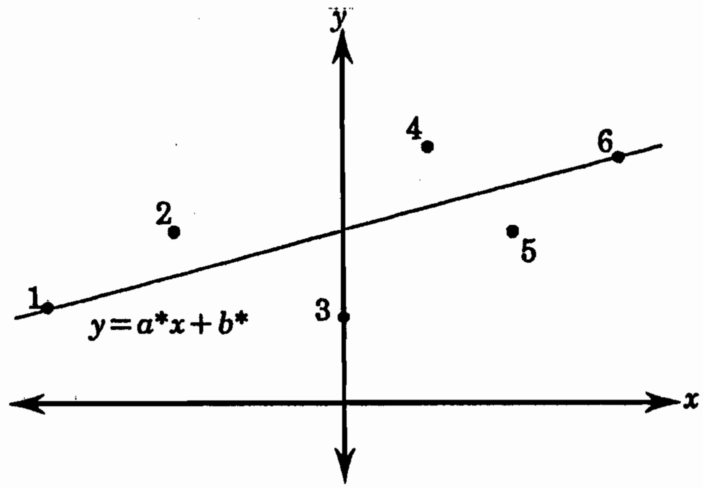
In the context of this thesis, the most notable properties of the dual transformation are the preservation of *above-below* and *incidence* relationships between the points and lines. Also note that the dual transformation always maps a point to a nonvertical line (assuming there are no points at infinity).

The point $p_i : (x_i, y_i)$ is *above* the line $l' : y = a'x + b'$ if

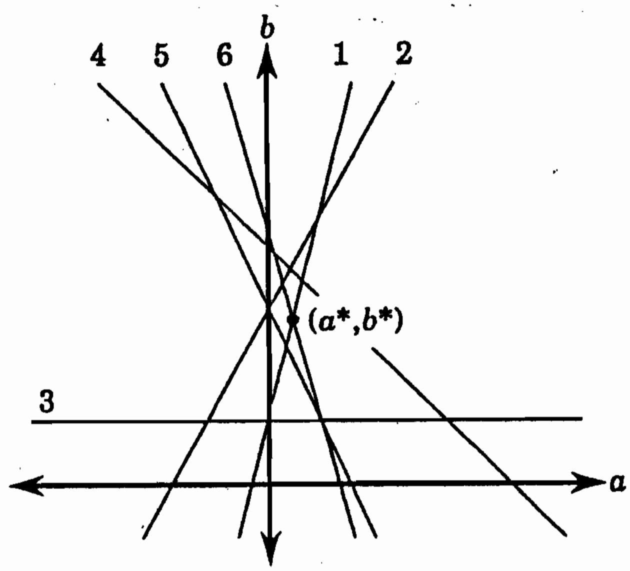
$$y_i - (a'x_i + b') > 0;$$

which may also be stated as the line is below the point. Similarly, the point p_i is *below* the line l' if

$$y_i - (a'x_i + b') < 0;$$



(a)



(b)

Figure 4.1. (a) Arrangement in the (x, y)-plane. (b) Dual arrangement in the (a, b)-plane.

which may also be stated as the line is above the point. If a point is neither above nor below the line, it must be on, or *incident to*, the line. The same terminology holds in either the (x, y)-plane or the (a, b)-plane.

We now show that if a point p_i is above the line l' in the (x, y)-plane, then the line $l_i = D(p_i)$ is above the point $p' = D(l')$ in the (a, b)-plane. If the point p_i is above the line l' in the (x, y)-plane, then

$$y_i - (a'x_i + b') > 0.$$

We wish to show that the line l_i is above the point p' in the (a,b)-plane. According to the definition, l_i is above the point p' if

$$b' - (-x_i a' + y_i) < 0,$$

which may be rewritten as

$$y_i - (a'x_i + b') > 0,$$

which is true from the assumption. Clearly, the reverse argument, from the (a,b)-plane to the (x,y)-plane, also holds (just reverse the variable names). Hence, the above-below relationship is preserved.

The point p_i is incident to the line l' if

$$y_i = a'x_i + b',$$

and vice versa. Assume that p_i is incident to l'. We wish to show that l_i is incident to the point p':

$$b' = -x_i a' + y_i$$

or

$$y_i = a'x_i + b',$$

which is true by the assumption. Hence the incidence relationship is preserved.

4.2. Arrangements

The original problem posed in this thesis is given in terms of n *data points*, $p_i : (x_i, y_i)$ $(i = 1, \ldots, n)$, and an *approximation line*, $l' : y = a'x + b'$, in the (x,y)-plane. After applying the dual transformation the problem is given in terms of n *data lines*, $l_i : b = -x_i a + y_i$ $(i = 1, \ldots, n)$, and an *approximation point*, $p' : (a', b')$, in the (a,b)-plane (see **Figure 4.1**). The (a,b)-plane has a nice interpretation in this application since it represents the parameter-space (slope and intercept) of the original problem (some applications do not have such a direct interpretation).

To understand the advantage of considering the dual problem, we must first mention a property of the vertical and orthogonal L_1 methods: there exists an optimal line $y = a^*x + b^*$ (point (a^*, b^*)) which is incident to two data points (lines) in the (x,y)-plane ((a,b)-plane). Note that the combinatorial complexity of the problem is $O(n^2)$, the number of pairs of data points (or pairs of data lines). The advantage of considering the problem in the (a,b)-plane is that there are several efficient methods for searching the line arrangement for a point determined by the intersection of the lines, whereas there are no comparable methods for searching through the lines defined by pairs of points in a large point set.

4.3. Multidimensional Search Technique and the Vertical L_1 Problem

In the vertical L_1 problem, the search is performed efficiently by *eliminating* many lines with only a constant number of tests at each step and is based on the multidimensional search technique by Megiddo [Me83]. Since an optimal point exists which is determined by the intersection of two of the data lines, those data lines which do not determine an optimal solution may be eliminated. At each iteration, the test determines a constant factor, α, of the remaining lines which cannot determine an optimal (intersection) point.

Let $D(a,b)$ represent the value of the vertical L_1 norm at a point (a,b) with respect to n data lines in the (a,b)-plane. The following description is intended to illustrate the process of elimination. The discussion of the test used in this process is left to later sections.

The basic method for eliminating a constant factor of data lines proceeds as follows. Assume that there exists a test, which, given any line, l, in the (a,b)-plane reports the relative position (i.e. above, below, or on the line) of an optimal point (a^*,b^*); actually, we assume for reasons given below, that all optimal solutions lie on one side of a line. The test is applied to only two lines, l' and l'', in order to eliminate $1/8$ of the remaining lines.

(i) Determine the median, $-x_m$, of the slopes of the data lines.

(ii) Divide the data lines into pairs (l_i, l_j) such that l_i has slope less than the median and l_j has slope greater than the median. For each pair (l_i, l_j), compute their intersection point q_i.

(iii) Determine the point, $q_m : (a_m, b_m)$, with median a-coordinate among the $O(n/2)$ points $\{q_i\}$. Consider the vertical line, $l' : a = a_m$, which divides the points $\{q_i\}$ into approximately equal size groups of $O(n/4)$ points (see **Figure 4.2**). Test the line l'. If the minimum lies on l' then we are done; otherwise, suppose the test reports that an optimal solution exists to the right of l': $a^* > a_m$; then consider the set of approximately $n/4$ points

$$Q' = \{\, q_i \mid a_i < a_m \,\}$$

to the left of l'.

(iv) Determi[illegible] $b = -x_m a + c_m$, with median slope which divides the points in Q' into approximately equal size sets of $O(n/8)$ points (again, note that this can done in linear time with respect to the number of points in Q'). Test the line l''. Suppose the test reports that an optimal point (a^*,b^*) exists below l''. Let

$$Q'' = \{\, q_i \mid q_i \in Q', b_i > -x_m a_i + c_m \,\}$$

be the set of points above l''.

We now know that an optimal solution lies in the region R_1, formed by the intersection of the half-plane to the right of l' and the half-plane below l'' (see **Figure 4.3**). Furthermore, all the optimal solutions must lie in the region R_1. We know from the test

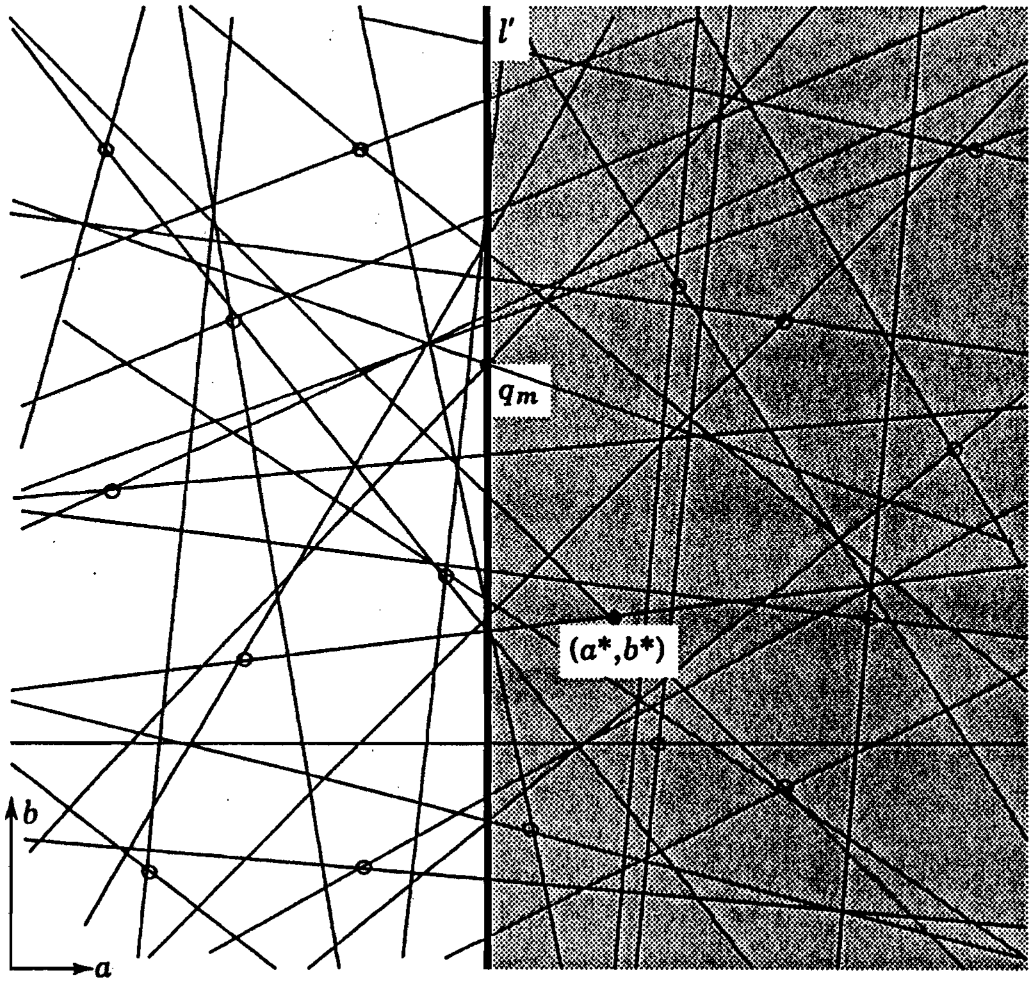


Figure 4.2. The first test line, l'.

that an optimal solution lies to one side of the test line. If another optimal solution lies on the other side, then by the convexity of $D(a,b)$ an optimal solution must lie on the line. Since the test did not find an optimal solution on the line, all optimal solutions must lie to one side of the line. Hence, all the optimal solutions lie in the region R_1 and any line which does not intersect R_1 cannot determine an optimal solution.

Consider the $O(n/8)$ points in Q''. In particular, consider the pairs of lines (l_s, l_t) which determine the points in Q''. One of each of those pairs of lines has slope greater than the median slope. A line with slope greater than the median slope and which is incident to a point in the region, R', to the left of l' and above l'', cannot intersect the region R_1. Hence, $O(n/8)$ data lines have been found which cannot determine an optimal point (a^*, b^*).

The process is repeated on the remaining lines until no more lines may be eliminated; the remaining lines determine the optimal point. Note that there are two major computations in the above algorithm: computing the median of a set of points along a line and testing a line to determine the relative position of an optimal solution. There

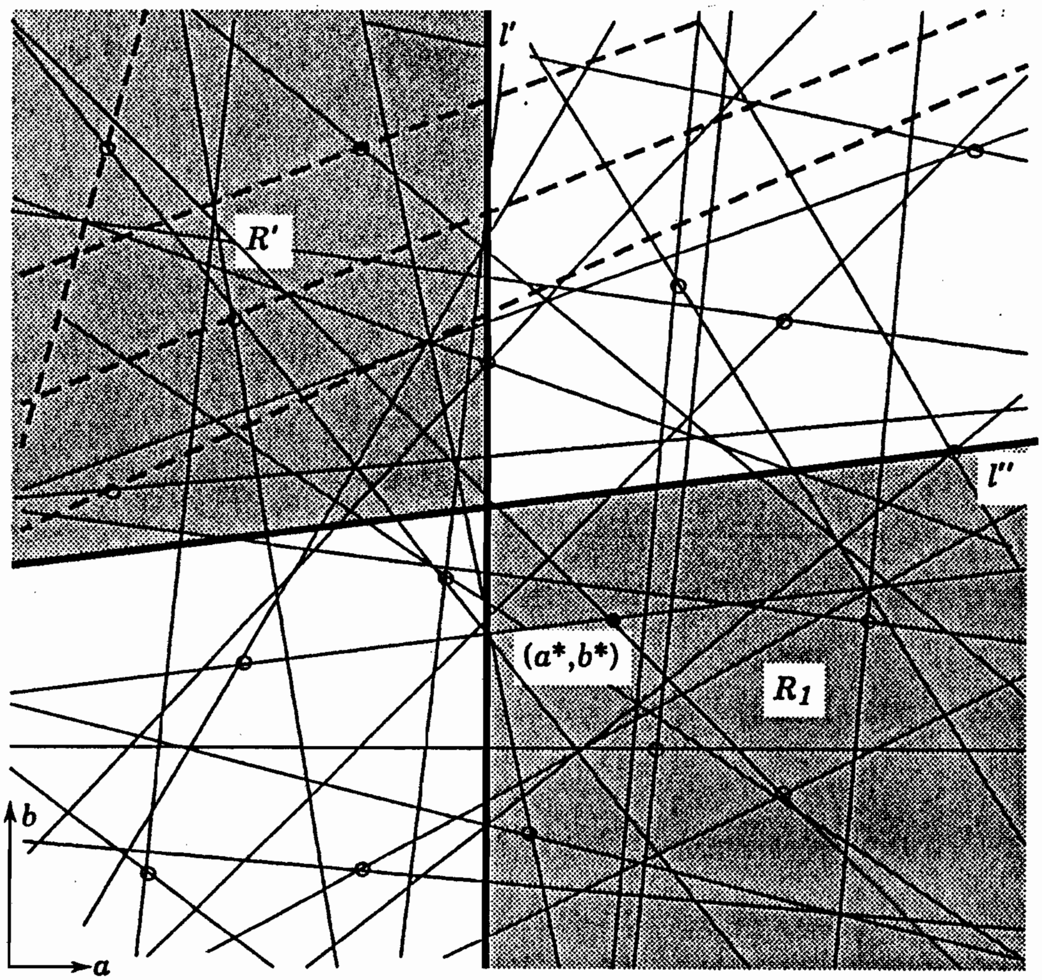


Figure 4.3. The situation after one iteration: the region R_1.

are several well known methods for computing the median point, or in general, selecting the kth element (see, for example, [AHU74]).

Hence, at each step a constant factor α, $0 < \alpha < 1$, of lines are eliminated, thus reducing the size of the problem for the next iteration. The crucial point in the complexity of the technique is the time required to perform the test of a line, which is performed twice for each iteration. Assuming that the test may be performed in linear time with respect to the number of remaining lines, let $T(n)$ represent the exact time to compute one iteration. $T(n)$ is $O(n)$ where the constant has been noted to be dependent on d, the number of dimensions, which is fixed [Me84].

Since the number of lines is reduced by a factor of α at each step, the time complexity is bounded by

$$T(n) + T(\alpha n) + T(\alpha^2 n) + T(\alpha^3 n) + \cdots + T(\alpha^{O(\log n)} n),$$

which is $O(n)$ total time, since each $T(m) < cm$ for some fixed $c > 0$ (as m becomes small the problem may be solved more efficiently by direct methods). Note that since the size of the problem is reduced by α at each step, there are only $O(\log n)$ iterations.

The above description is a direct application of Megiddo's technique. In Section 5, we show that a direct application of the technique does not result in a linear time algorithm. The difficulty lies in devising a test which reports an answer in linear time with respect to the remaining number of lines.

4.4. Plane Sweeps and The Orthogonal L_1 Problem

The major difference between the vertical and orthogonal problems is that the vertical L_1 norm is convex while the orthogonal L_1 norm is not. The lack of convexity makes devising a test for a line, which runs in linear time with respect to the number of remaining lines, very difficult. Instead, the approach we took was to investigate all possible solutions. Such an approach was investigated by Morris and Norback [MN80] who searched for *candidate lines* which satisfied two properties of an optimal line. The first property is the *incidence property* which is the same property used in the vertical L_1 problem. The second property, called the *weighted median property*, was used by Morris and Norback only to save some division operations. They did not investigate the complexity of their computations; in fact, their algorithm has the same time complexity, $O(n^3)$, as a brute force algorithm.

The weighted median property states that there exists a solution to the weighted orthogonal L_1 problem which satisfies

$$\left|\sum_{i\in A} w_i - \sum_{i\in B} w_i\right| < \sum_{i\in O} w_i,$$

where A, B, and O are the set of points (lines) above, below, and on, respectively, the optimal line (point) in the (x,y)-plane ((a,b)-plane). Lines (points) in the (x,y)-plane ((a,b)-plane) which satisfy both the incidence and weighted median property are called *candidate solutions*. Note that in the unweighted case, the weighted median property states that an optimal line (point) exists which divides the points (lines) into approximately equal size sets above and below the line (point).

Generally speaking, we take the same approach as Morris and Norback: we wish to find all candidate solutions and evaluate the value of $D(a,b)$ at those points in order to determine an optimal solution. However, rather than searching for candidate lines in the (x,y)-plane, we search for *candidate points* in the (a,b)-plane, the dual transformations of the candidate lines, by using efficient plane sweep techniques. The unweighted and weighted problems are considered separately since it seems easier to find the candidate points of the unweighted problem than those of the weighted problem.

4.4.1. k-graphs, k-belts and the Unweighted Orthogonal L_1 Problem

Let S be a set of n points in general position (no three collinear). For any two points

$p, q \in S$, the directed line $\overrightarrow{pq}$ has a certain number, $N(\overrightarrow{pq})$, of points of S on its *positive side*, that is, the open half-plane to the right of $\overrightarrow{pq}$. Erdös, Lovász, Simmons, and Strauss [ELSS73] consider the properties of *k-graphs*, G_k, of S whose edges are the segments $\overrightarrow{pq}$ with $N(\overrightarrow{pq}) = k$ $(k = 0, 1, \ldots, (n-2)/2)$ (see **Figure 4.4(a)**). Note that, for $k = (n-2)/2$, the lines $\overrightarrow{pq}$ are the candidate lines for the unweighted orthogonal L_1 problem. [ELSS73] provide an $\Omega(n \log k)$ lower bound and an $O(n\sqrt{k})$ upper bound for the number of edges of G_k. Hence the number of candidate solutions is in $\Omega(n \log n)$ and $O(n^{1.5})$.

The above complexity results were independently found by Edelsbrunner and Welzl [EW85] who studied so-called *k-sets*: sets of k points of S separated by a line from the rest of S, as defined above except that the line is not incident to any points in S. [ELSS73] conjecture that the upper bound is actually closer to the lower bound, thus providing motivation for an efficient search of the candidate vertices as a method for solving the orthogonal L_1 problem. However, neither of the above papers consider the efficient computation of the k-sets.

Edelsbrunner and Welzl [EW86] introduced the concept of the *k-belt* of an arrangement of lines. Consider a point p in the plane. Let $a(p)$, $b(p)$, and $o(p)$ represent the number of points above, below, and on p, respectively. The k-belt is defined as the set of all points such that $a(p) + o(p) \geq k$ and $b(p) + o(p) \geq k$, for $0 \leq k \leq \lceil n/2 \rceil$. For $k \geq 1$ the k-belt is bounded above and below by an unbounded polygonal chain. For k odd, note that the two boundaries of the $\lceil n/2 \rceil$-belt coincide (see **Figure 4.4(b)**). The chain is formed by *edges* which coincide with the lines of the arrangement and each edge is bounded by two *vertices* determined by the intersection of two of the lines in the arrangement. The chains are monotone with respect to the horizontal axis which means that a vertical line intersects the chain in exactly one point.

Although [EW85] do not consider, by definition, k-sets which are defined by lines which are incident to points of S, in [EW86], the vertices of the k-belt are actually the dual representation of the lines $\overrightarrow{pq}$ investigated in [ELSS73]. [EW86] provides an $O(b_k(n) \log^2 n)$ time algorithm for finding the vertices of the k-belt based upon a "sophisticated plane sweep" algorithm, where $b_k(n)$ is the number of vertices in the k-belt. The plane sweep may be considered sophisticated since it does not have to search through all of the $O(n^2)$ vertices of the arrangement; it is able to construct the k-belt, for fixed k, by directly computing the vertices which lie on the boundaries of the belt in left to right order.

In the unweighted L_1 problem we are interested in the $\lceil n/2 \rceil$-belt, referred to as the *median-belt*. Because of the dual correspondence between the lines of G_k and the vertices of the k-belt, we know that the vertices of the median-belt are the candidate points of the unweighted orthogonal problem. Hence, the sophisticated plane sweep algorithm may be used to find the median-belt vertices in $O(n^{1.5} \log^2 n)$ time. Based on that algorithm, we show how to efficiently compute the solution to the unweighted orthogonal L_1 problem.

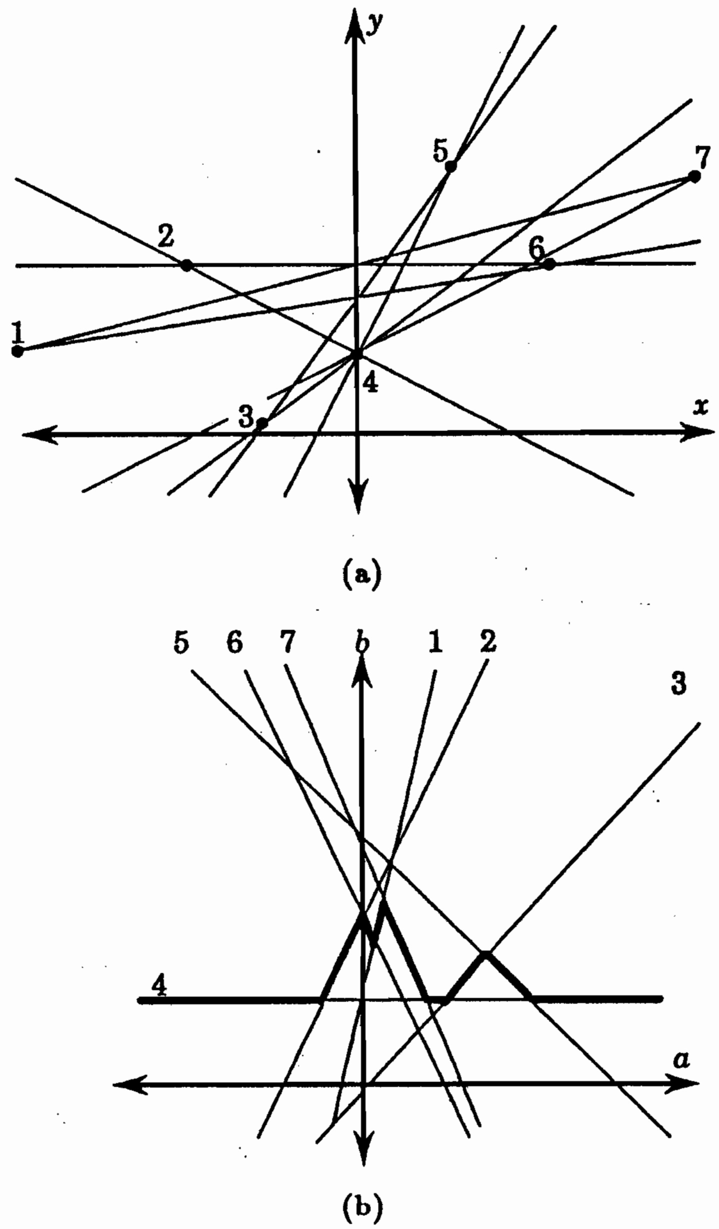


Figure 4.4. (a) The lines determined by the edges of the graph G_3. (b) The 3-belt of a set of 8 lines.

4.4.2. Weighted Median Belts and the Weighted Orthogonal L_1 Problem

The median-belt may be generalized to the *weighted median-belt*. The weighted median-belt is the set of points in the (a, b)-plane which satisfy the weighted median property: at each point on the belt, the sum of the weights of the data lines incident with the point is greater than the absolute difference of the sums of weights of the data lines above and below the point. Hence, the vertices of the weighted median-belt are the candidate points of the weighted orthogonal L_1 problem.

We are not aware of any previous work concerning weighted median-sets or weighted median-belts. In particular, we are interested in the determining the number of vertices in a weighted median-belt and the complexity of finding them. Those problems are answered in in **Section 6** where the weighted median-belt is investigated and then used to solve the weighted orthogonal L_1 problem.

5. Vertical L_1 Linear Approximation Problem

This section describes an optimal, $\Theta(n)$ time, algorithm for finding an exact solution to the weighted vertical L_1 linear approximation problem of a set of points in the plane (referred to as the *vertical L_1 problem*). Clearly, the algorithm is also optimal for the unweighted problem in which all weights, w_i, are set equal to 1. The algorithm is of interest for two reasons. First, the algorithm improves the previous upper bound of $O(n^3)$ for an algorithm which finds the exact solution. Second, the algorithm modifies a basic technique used in computational geometry, the multi-dimensional *pruning technique*, to suit a problem which does not admit a direct application of the basic technique. Furthermore, the modified technique is applicable to a wide class of similar problems.

5.1. Preliminaries

The vertical L_1 problem in the (x, y)-plane, referred to as the *primal problem*, and the corresponding problem in the (a, b)-plane, referred to as the *dual problem*, are formally stated (note that the use of the term *primal* is used for convenience; there is no relation to the *primal* and *dual* problems of mathematical programming). The convexity of the objective function, the vertical L_1 norm, is shown. A property of an optimal solution to the primal problem is noted and the corresponding dual property is shown.

5.1.1. The Primal Problem

The vertical L_1 problem may be formally stated as follows.

Problem 5.1. The Primal Problem. *Given a set, S, of n points, p_i : (x_i, y_i) $(i = 1, \ldots, n)$, in the (x, y)-plane, with corresponding weights, w_i, find a pair of values (a^*, b^*), for the parameters a and b, which solves the following mini-sum problem:*

$$\min_{a,b} \quad D(a, b) \equiv \sum_{i=1}^{n} w_i \left| y_i - (ax_i + b) \right| .$$

$D(a, b)$ will be referred to as the *objective function*. The goal of the vertical L_1 problem is to find parameter values a^* and b^* such that $D(a^*, b^*) \leq D(a, b)$, for all values of a and b.

Rice [R64] shows that the set of possible solutions to the (unweighted) vertical problem is convex by noting that $D(a, b)$ is convex. For the completeness of the argument, a definition of convexity is stated, and then $D(a, b)$ is shown to satisfy the definition of convexity.

Definition 5.1. Convex Function. *A function $F(x, y)$, defined for points in the (x, y)-plane, is said to be convex if, given λ, $\mu > 0$, $\lambda + \mu = 1$, then for*

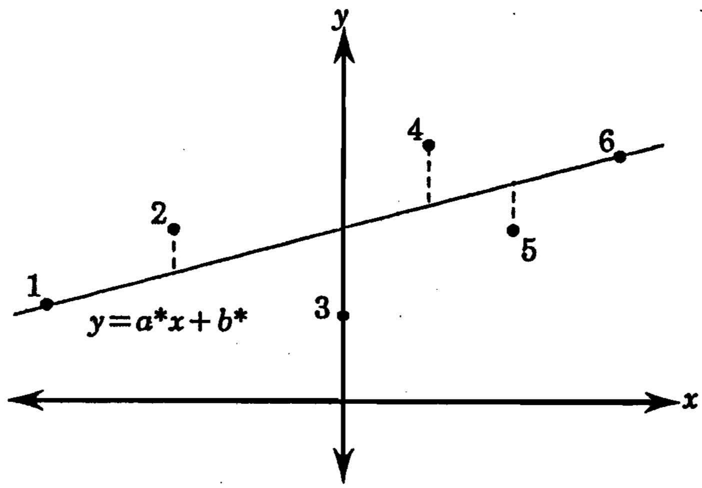


Figure 5.1. Vertical L_1 problem in the (x, y)-plane.

any two points (x_1, y_2), (x_2, y_2), the following is true:

$$F(\lambda(x_1, y_1) + \mu(x_2, y_2)) \le \lambda F(x_1, y_1) + \mu F(x_2, y_2),$$

where $\lambda(x_1, y_1) + \mu(x_2, y_2)$ is called the convex combination of the points (x_1, y_1) and (x_2, y_2).

$D(a, b)$ is shown to be convex by the definition of convexity.

Lemma 5.1. *$D(a, b)$ is convex.*

Proof: *For any θ $(0 \le \theta \le 1)$ and $(a_1, b_1), (a_2, b_2)$, $w_i > 0$,*

$$\begin{aligned}
&\theta D(a_1, b_1) + (1-\theta) D(a_2, b_2) - D(\theta a_1 + (1-\theta) a_2, \theta b_1 + (1-\theta) b_2) \\
&= \theta \sum_{i=1}^{n} w_i \left|\, y_i - (a_1 x_i + b_1) \,\right| + (1-\theta) \sum_{i=1}^{n} w_i \left|\, y_i - (a_2 x_i + b_2) \,\right| \\
&\quad - \sum_{i=1}^{n} w_i \left|\, y_i - \left[\, (\theta a_1 + (1-\theta) a_2)\, x_i + (\theta b_1 + (1-\theta) b_2) \,\right] \,\right| \\
&= \sum_{i=1}^{n} w_i \Big\{ \theta \left|\, y_i - (a_1 x_i + b_1) \,\right| + (1-\theta) \left|\, y_i - (a_2 x_i + b_2) \,\right| \\
&\qquad - \left|\, y_i - \left[\, (\theta a_1 + (1-\theta) a_2)\, x_i + (\theta b_1 + (1-\theta) b_2) \,\right] \,\right| \Big\} \\
&\ge 0
\end{aligned}$$

(Since $|A| + |B| \ge |A + B|$, for any A and B). □

$D(a, b)$ has a unique global minimum which is called the *optimal value* (see [R64] for a mathematical characterization of the L_1 approximation problem). A pair of values for the parameters a and b, (a^*, b^*), which attains the optimal value, called an *optimal solution*, always exists and is either unique or there are infinitely many such pairs.

Graphically, in the (x, y)-plane (see **Figure 5.1**), an optimal solution, (a^*, b^*), defines an *optimal line*, $y = a^*x + b^*$, which minimizes $D(a, b)$ with respect to the given point set S. An *approximate solution* is any pair of values (a, b); likewise, an *approximate line* is any line defined by the pair of values (a, b). The following result characterizes an optimal solution to **Problem 5.1.**

Lemma 5.2. *There is an optimal line which is incident to two points of S.*

Barrodale and Roberts [BR70] provide a proof for **Lemma 5.2** for vertical L_1 approximation in $(n + 1)$-dimensions based on the properties of convex sets. We will give a proof for **Lemma 5.3**, the dual version of **Lemma 5.2.**

An approximate line which is incident to two points of S is called a *candidate line*, or in general, a *candidate solution*. Since there are $\binom{n}{2} = O(n^2)$ pairs of points, there are $O(n^2)$ candidate lines and one of the candidate lines is an optimal line due to **Lemma 5.2**. A brute force algorithm to determine the candidate lines can be easily derived. For each pair of points in S, compute $D(a, b)$ for the parameters a and b determined by a candidate line which passes through the pair of points, and choose the pair which minimizes $D(a, b)$. Since there are $O(n^2)$ pairs of points and for each pair it takes $O(n)$ time to compute $D(a, b)$, an upper bound for determining an optimal line is $O(n^2) * O(n) = O(n^3)$. However, by considering the vertical L_1 problem in the dual plane, an efficient search through the candidate solutions may be derived.

5.1.2. The Dual Problem

The point set S of the primal problem is transformed to a line set, H, of the dual problem by transforming each point (x_i, y_i) in S in the (x, y)-plane to a line $l_i : b = -x_i a + y_i$ in H in the (a, b)-plane (see **Section 4**). The corresponding dual problem of **Problem 5.1** may be formally stated as follows.

Problem 5.2. The Dual Problem. *Given a set, H, of n lines, $l_i : b = -x_i a + y_i (i = 1, \ldots, n)$, in the (a, b)-plane, with corresponding weights, w_i, find a pair of values (a^*, b^*), for the parameters a and b, which solves the following mini-sum problem:*

$$\min_{a,b} \quad D(a, b) \equiv \sum_{i=1}^{n} w_i \left| y_i - (a x_i + b) \right| .$$

Note that the objective function, $D(a, b)$, is the same in both the primal and the dual problems (in mathematical programming, the objective functions of the *primal* and *dual* problems are different).

Graphically, in the (a, b)-plane (see **Figure 5.2**), an optimal solution, (a^*, b^*), defines an *optimal point*, which minimizes the L_1 norm with respect to the set of

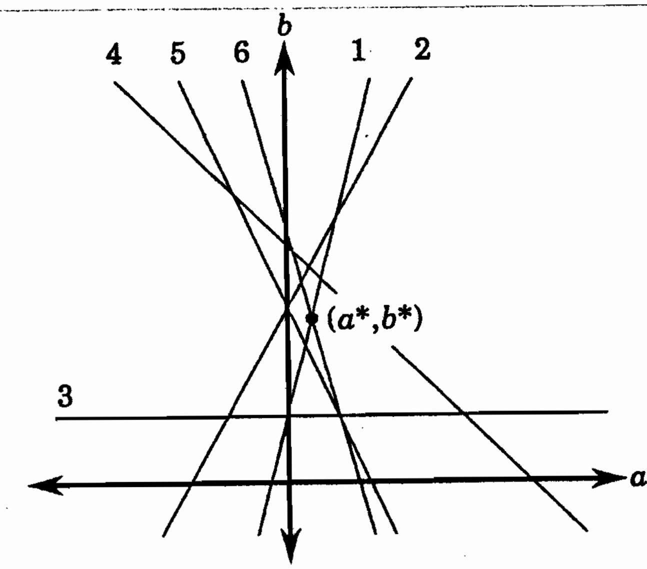


Figure 5.2. Vertical L_1 problem in the (a, b)-plane.

lines H. The following characterization of an optimal point in the (a, b)-plane, the dual version of **Lemma 5.2**, is the key to successfully applying the pruning technique.

> **Lemma 5.3.** *There is an optimal point which is incident to two lines in H.*
>
> **Proof:** *Consider the arrangement in the (a, b)-plane determined by the lines in H. Since $D(a, b)$ is convex* (**Lemma 5.1**) *and piecewise linear, the set of points $\{(a, b, c)|c \geq D(a, b)\}$, forms a convex polyhedron above the (a, b)-plane. By the Fundamental Theorem of Linear Optimization* [PFTV86], *the function $D(a, b)$ is minimized at an extreme point of the convex polyhedron. The extreme points are the vertices of the convex polyhedron. Since $D(a, b)$ is a piecewise linear function, the vertices of the polyhedron correspond to intersections of the lines in H. Hence, there exists an optimal point in the (a, b)-plane which lies at the intersection of two data lines in H.* □

A *candidate point* in the (a, b)-plane is a point determined by the intersection of two data lines. An $O(n^3)$ brute force algorithm to search through the candidate points may be derived from **Lemma 5.3**: perform a plane-sweep ($O(n^2)$ time, $O(n)$ space [EG86]), computing the value of the L_1 norm at each intersection point, or candidate point, and choose the candidate point which minimizes the L_1 norm. By utilizing another property of the vertical L_1 problem, the time complexity may be reduced to $O(n^2)$ as illustrated in the algorithm for the weighted orthogonal L_1 problem presented in **Section 6**.

5.2. $\Theta(n)$ Algorithm

Based on the result in **Lemma 5.3**, we will apply the pruning technique to the dual problem to eliminate data lines which are known not to determine an optimal intersection point; when no more lines may be eliminated we know that the remaining lines determine an optimal solution. As mentioned in Section 4 the important part of the technique is the test to determine the relative position of an optimal point with respect to a line. By using the convexity of $D(a,b)$, such a test may be derived based on the computation of the derivatives at the point which minimizes $D(a,b)$ on the given line l. Since the method for pruning the data lines is a direct application of the technique presented in [Me83], we focus upon the development of a test which reports the relative position of an optimal position with respect to the given line l in linear time with respect to the number of remaining data lines.

As mentioned above, the test utilizes the derivatives of $D(a,b)$ to determine an answer. The problem lies in the complexity of computing the derivatives. Since all of the data lines contribute to the value of the derivative at a given point, the time complexity of a basic test will require $\Omega(n)$ time even if some data lines have been "eliminated." In fact, the principles of the pruning technique are needed to produce an $O(n)$ time basic test as described below.

In order to produce an $O(n)$ total time algorithm, however, the test must be able to determine a result in time linear to the remaining number of constraints. The problem with the basic pruning technique is that it assumes information from eliminated lines is not needed in further computations. To overcome that problem a data structure, called R_0, is introduced which, in one sense, retains the relevant information about the eliminated lines. A method for exploiting that information in order to compute the derivatives in linear time with respect to the remaining data lines is given. The data structure R_0 is described and finally, a test is given which uses R_0 to test a line in the required time.

5.2.1. Maintaining $D(a,b)$

The analytic complexity of $D(a,b)$ is exploited in order to maintain $D(a,b)$ in a form such that, under certain conditions, the contribution of eliminated data lines can be accounted for in constant time. First, note that $D(a,b)$ is a convex piecewise linear function. Second, although $D(a,b)$ has points of nondifferentiability, the one-dimensional left-hand and right-hand side (*lhs* and *rhs*, respectively) derivatives of $D(a,b)$ may be computed at any point (a,b).

For fixed a and b, define sets $I^+(a,b)$ and $I^-(a,b)$ of indices by:

$$I^+(a,b) = \{\, i \mid y_i > ax_i + b \quad (i = 1,\ldots,n)\}$$

and

$$I^-(a,b) = \{\, i \mid y_i < ax_i + b \quad (i = 1,\ldots,n)\}\,.$$

The above notation may be interpreted as the set of indices of points (x_i, y_i) above and below, respectively, a line defined by (a,b) in the (x,y)-plane, or as the corresponding set of indices of lines defined by $(-x_i, y_i)$ above and below, respectively, a point (a, b) in the (a, b)-plane. The objective function $D(a, b)$ may then be written as follows.

$$\begin{aligned}
D(a,b) &= \sum_{i=1}^{n} w_i \left| y_i - (ax_i + b) \right| \\
&= \sum_{i \in I^+(a,b)} w_i \left(y_i - (ax_i + b) \right) \\
&\quad - \sum_{i \in I^-(a,b)} w_i \left(y_i - (ax_i + b) \right) \\
&= \sum_{i \in I^+(a,b)} w_i y_i - \sum_{i \in I^+(a,b)} w_i a x_i - \sum_{i \in I^+(a,b)} w_i b \\
&\quad - \sum_{i \in I^-(a,b)} w_i y_i + \sum_{i \in I^-(a,b)} w_i a x_i + \sum_{i \in I^-(a,b)} w_i b \\
&= Y^+ - aX^+ - bW^+ - Y^- + aX^- + bW^-,
\end{aligned}$$

where

$$W^+ = \sum_{i \in I^+(a,b)} w_i, \; X^+ = \sum_{i \in I^+(a,b)} w_i x_i, \; Y^+ = \sum_{i \in I^+(a,b)} w_i y_i,$$
$$W^- = \sum_{i \in I^-(a,b)} w_i, \; X^- = \sum_{i \in I^-(a,b)} w_i x_i, \; Y^- = \sum_{i \in I^-(a,b)} w_i y_i.$$

Similarly, the one-dimensional functions of $D(a, b)$ may be written as follows.

Fixing $a = \alpha$, the value of $D(a, b)$ may be regarded as a function of b:

$$D(b) \equiv D(\alpha, b) = \sum_{i=1}^{n} w_i \left| y_i - (\alpha x_i + b) \right|.$$

In the (x, y)-plane, the one-dimensional function, $D(b)$, corresponds to the value of $D(a, b)$ as a line with fixed slope α, $l : y = \alpha x + b$ is translated. In the (a, b)-plane, $D(b)$ corresponds to the value of $D(a, b)$ as a point (α, b) moves along the vertical line $l : a = \alpha$. $D(b)$ is a piecewise linear convex function of b, and at a differentiable point, (α, b),

$$\begin{aligned}
D'(b) \equiv \frac{dD(\alpha, b)}{db} &= - \sum_{i \in I^+(\alpha,b)} w_i + \sum_{i \in I^-(\alpha,b)} w_i \\
&= W^- - W^+
\end{aligned}$$

At a differentiable point on l in the (a, b)-plane, $D'(b)$ indicates whether $D(a, b)$ is increasing or decreasing along l. At a nondifferentiable point, (a', b'), on l (a point of intersection with a data line), the *lhs* and *rhs* derivatives may be computed in order to determine the local behaviour of $D(a, b)$. Since we are considering a one-dimensional problem, note that the *left-hand* side implies points on l with $b < b'$; similarly, the *right-hand* side implies points on l with $b > b'$.

Letting $b = \gamma a + \beta$, the value of $D(a,b)$ may be written as a function of a:

$$D(a) \equiv D(a, \gamma a + \beta) = \sum_{i=1}^{n} w_i \left| y_i - (a(x_i + \gamma) + \beta) \right| .$$

$D(a)$ corresponds to the value of $D(a,b)$ as a point moves along the nonvertical line, $b = \gamma a + \beta$, in the (a,b)-plane and to the value of $D(a,b)$ as a line is rotated about the point $(-\gamma, \beta)$ in the (x,y)-plane. $D(a)$ is a piecewise linear convex function of a, and at a differentiable point:

$$\begin{aligned} D'(a) \equiv \frac{dD(a, \gamma a + \beta)}{da} &= -\sum_{i \in I^+} w_i (x_i - \gamma) + \sum_{i \in I^-} w_i (x_i - \gamma) \\ &= X^- - X^+ + \gamma (W^+ - W^-), \end{aligned}$$

where $I^+ = I^+(a, \gamma a + \beta)$ and $I^- = I^-(a, \gamma a + \beta)$. In the (a,b)-plane, at a differentiable point on l, $D'(a)$ indicates whether $D(a,b)$ is increasing or decreasing along the line l. At a nondifferentiable point on l, the *lhs* and *rhs* derivatives may be computed in order to determine the local behaviour of $D(a,b)$.

The above representation will be used to maintain the contributions of eliminated data lines. Consider the problem in the (a,b)-plane. Assume that some data lines, E, have been eliminated by the pruning technique. Let the sets E^+ and E^- represent the set of indices of eliminated data lines above and below, respectively, a point (a,b), while I^+ and I^- represent the indices of the remaining data lines above and below, respectively, (a,b). Then $D(a,b)$, $D'(a)$, and $D'(b)$ may be written as follows.

$$\begin{aligned} D(a,b) = & \sum_{i \in I^+(a,b)} w_i (y_i - (a x_i + b)) - \sum_{i \in I^-(a,b)} w_i (y_i - (a x_i + b)) \\ & + Y'^+ - a X'^+ - b W'^+ - Y'^- + a X'^- + b W'^- \\ D'(a) = & -\sum_{i \in I^+} w_i (x_i - \gamma) + \sum_{i \in I^-} w_i (x_i - \gamma) \\ & + X'^- - X'^+ + \gamma (W'^+ - W'^-) \\ D'(b) = & -\sum_{i \in I^+(a,b)} w_i + \sum_{i \in I^-(a,b)} w_i \\ & + W'^- - W'^+, \end{aligned}$$

where

$$\begin{aligned} W'^+ &= \sum_{i \in E^+(a,b)} w_i, \; X'^+ = \sum_{i \in E^+(a,b)} x_i, \; Y'^+ = \sum_{i \in E^+(a,b)} y_i, \\ W'^- &= \sum_{i \in E^-(a,b)} w_i, \; X'^- = \sum_{i \in E^-(a,b)} x_i, \; Y'^- = \sum_{i \in E^-(a,b)} y_i. \end{aligned}$$

The advantage of considering the functions in this form is that, under certain conditions, the contributions of the eliminated constraints may be maintained by six variables.

Suppose that the set E of data lines has been eliminated and the six variables computed. The problem is to compute the next set of derivatives in linear time with respect to the remaining data lines. Note, however, that if the position of the eliminated data lines changes with respect to the next point of computation, then the contributions from eliminated points will have to be recomputed since we no longer know whether they are above or below the current point. The variables may be used to advantage by restricting the computations to a region whose relative position with respect to the eliminated lines does not change. The next section illustrates that such a restriction can be satisfied in the one-dimensional search along a line l for a point which minimizes $D(a,b)$ on l. The modified technique describes how such a region may be maintained in the (a,b)-plane and how the tests required by the pruning technique may be restricted to that region.

5.2.2. The Basic Test

The basis of the algorithm is a test which, given any line, l, and a set of data lines $H = \{ l_i : b = -x_i a + y_i, i = 1 \ldots n \}$, in the (a,b)-plane, answers:

Case 1. an optimal point, (a^*, b^*), lies on l, or

Case 2. (a^*, b^*) lies to the right of l, or

Case 3. (a^*, b^*) lies to the left of l.

The test result may be determined by exploiting the analytic behavior of $D(a,b)$. The test involves two steps. First, given a line l in the (a,b)-plane, find the point, (a', b'), which minimizes $D(a,b)$ on l. The convexity of $D(a,b)$ guarantees that such a point exists. Furthermore, the piecewise linearity of $D(a,b)$ guarantees that there exists such a point which lies at an intersection with one of the data lines, l_i. Second, compute the *lhs* and *rhs* one-dimensional derivatives along the data line l_i at (a', b') to determine the side which contains an optimal solution. Recall that the point (a', b') determines a left-hand side and a right-hand side on l_i; we say that a point lies to the left (right) of l with respect to l_i if the point lies on the same side of l as the left-hand (right-hand) side of l_i.

In the first step there are two cases two consider: first, the test line, $l : a = \alpha$, is vertical; second, the test line $l : b = \gamma a + \beta$, is nonvertical. In both cases, the principle of the pruning technique may be applied, although the first case is a very familiar one-dimensional problem. The second case illustrates the method of maintaining $D(a,b)$ described above in order to maintain linear time computation.

First, consider the case when l is vertical. In order to find the point which minimizes $D(a,b)$ on l, we set:

$$D'(b) = W^- - W^+ = 0.$$

The data line, l_i, which determines the intersection point with weighted median b-value minimizes $D(b)$ since it sets $D^{'}(b)$ closest to zero by balancing the weights equally above and below the point. This is a well known observation and avoids the use of computing derivatives to search for the minimum since there are $O(n)$ time weighted median finding algorithms [S76].

Second, consider the case when l is not vertical. Since we know that a minimizing point (a', b') lies at the intersection of one of the data lines l_i $(i = 1, \ldots, n)$ and l, the minimizing point may be found by a type of binary search with the moves decided by the derivatives at the intersection points. By computing the derivatives along l at the point with median a-coordinate, about half of the intersection points are "eliminated" at each step; hence, after $O(\log n)$ iterations the minimizing point is found. Note, however, that since the derivative requires $O(n)$ time to compute, the search would take a total time of $O(n \log n)$. The method for maintaining $D(a, b)$ described above will be used to compute the derivatives in linear time with respect to the number of remaining data lines thus providing an $O(n)$ linear time algorithm.

First, compute the intersection points, (a_i, b_i), of the data lines, l_i $(i = 1, \ldots, n)$, with the test line l. Find the intersection point with median a-coordinate, a_m. Compute the left-hand and right-hand side derivatives at the median point.

1. If the *rhs* derivative is negative, then $a' > a_m$.
2. If the *lhs* derivative is positive, then $a' < a_m$.
3. Otherwise, $a' = a_m$.

Hence, at each step, we can either eliminate half the points from the search or determine the minimizing point (a', b').

Suppose $a' < a_m$ (see **Figure 5.3**); then the data lines $E = \{ l_i : a_i > a_m \}$ may be eliminated from the search since they cannot determine (a', b'). Those data lines can be divided into two sets, A and B, according to their relative position with respect to l for $a < a_m$. A contains the eliminated data lines with slope less than l and B contains the rest; in other words, for $a < a_m$, the lines in A are above l and the lines in B are below l.

Let $H' = H \backslash E$. The value of the derivatives of $D(a, b)$ at a point on l to the left of a_m can be written as

$$\begin{aligned} D'(a) = &- \sum_{i \in I^+} w_i(x_i - \gamma) + \sum_{i \in I^-} w_i(x_i - \gamma) \\ &+ X'^- - X'^+ + \gamma(W'^+ - W'^-), \end{aligned}$$

where

$$\begin{aligned} W'^+ &= \sum_{i \in A} w_i, \ X'^+ = \sum_{i \in A} x_i, \\ W'^- &= \sum_{i \in B} w_i, \ X'^- = \sum_{i \in B} x_i, \end{aligned}$$

and $I^+ = I^+(a, \gamma a + \beta)$ and $I^- = I^-(a, \gamma a + \beta)$. Note, however, that the four variables, W'^+, X'^+, W'^-, and X'^-, are constant for points on l to the left of a_m; hence, the above procedure can be repeated in time proportional to the number of remaining data

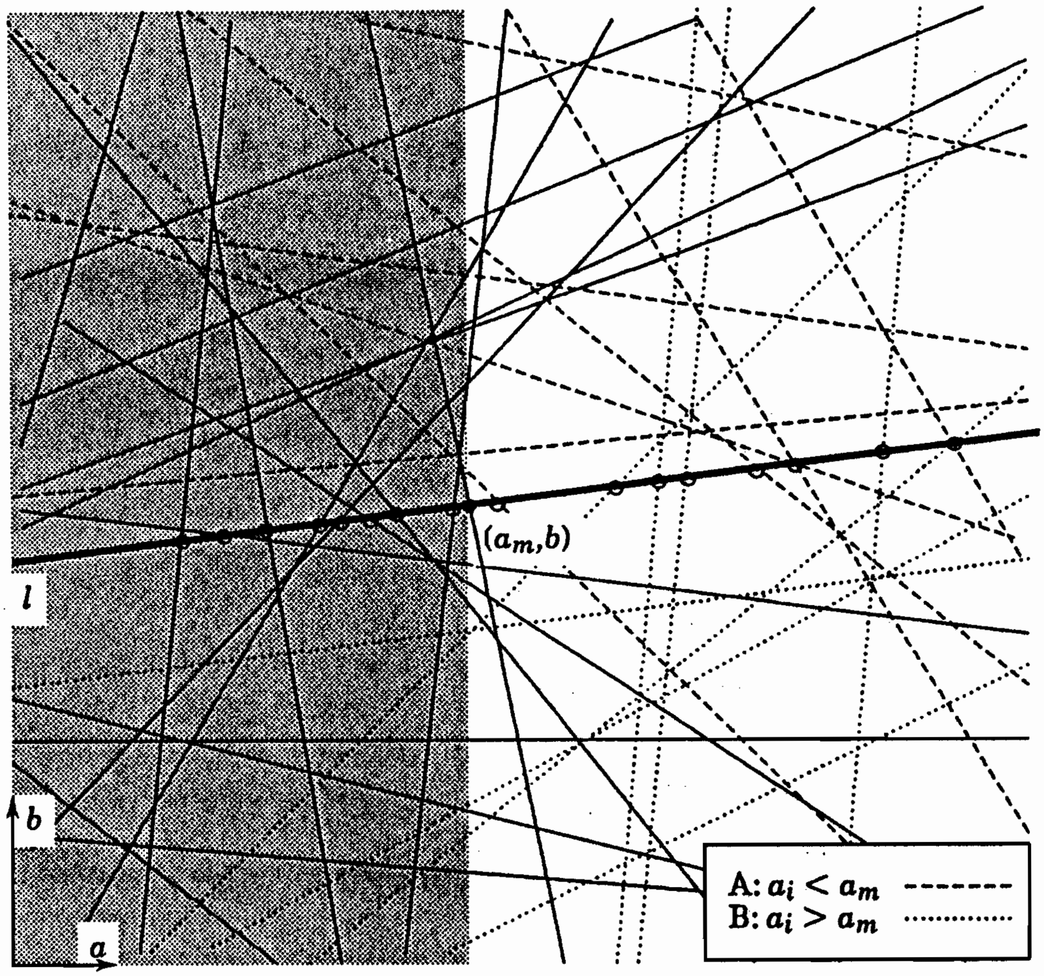


Figure 5.3. Minimizing $D(a, \gamma a + \beta)$.

lines (the lines which intersect l to the left of a_m) by accumulating the contributions of eliminated lines in the appropriate variables. Eventually, the point (a', b') is found and, as shown in **Section 4**, the total time spent is $O(n)$.

We may now assume we have found, in $O(n)$ time, a minimizing point, (a', b'), on l which lies at the intersection of a data line l_i. Consider the second step. Given the point (a', b'), the test result is determined as follows. Compute the *lhs* and *rhs* derivatives of $D(a, b)$ along l_i at (a', b').

1. If the *rhs* derivative is negative, then (a^*, b^*) lies to the right, with respect to l_i, of l.
2. If the *lhs* derivative is positive, then (a^*, b^*) lies to the left, with respect to l_i, of l.
3. Otherwise, (a', b') is (a^*, b^*).

The proofs for the above results are based on the convexity of $D(a, b)$.

Consider the first condition. If the *rhs* derivative is negative at (a', b'), then $D(a, b)$ decreases along l_i to the right of l at (a', b') and there exists a minimizing point (a'_i, b'_i) on l_i such that $D(a'_i, b'_i) < D(a', b')$ (see **Figure 5.3**). If an optimal point, (a^*, b^*), lies

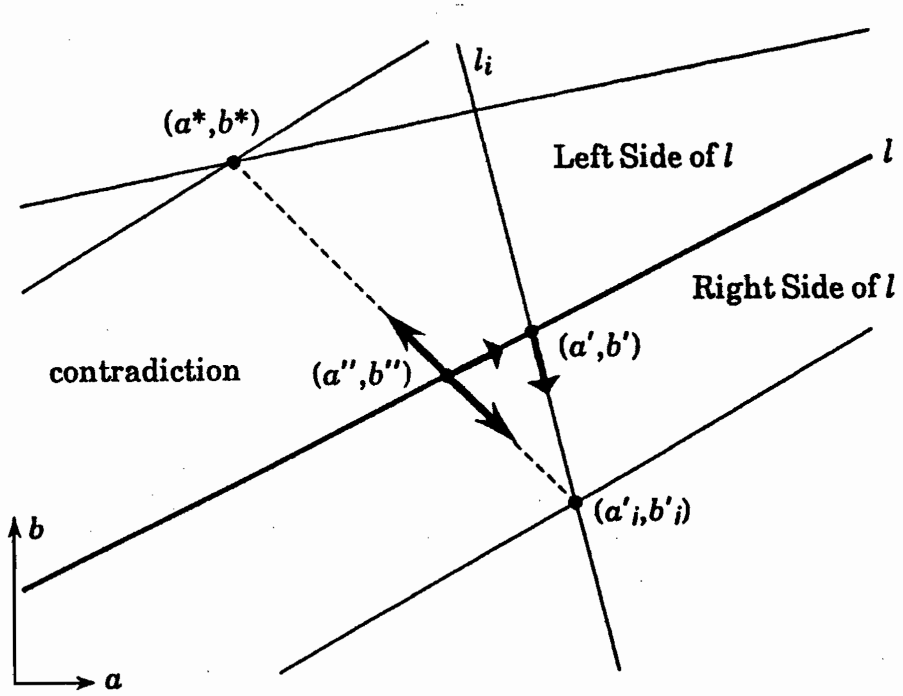


Figure 5.4. The arrows indicate the direction of decrease of the value of $D(a,b)$ along the line. If two arrows along the same line point away from a point known not to be an optimum solution then there exist two local minima; a contradiction to the convexity of $D(a,b)$.

to the left of l, then a line segment from (a^*,b^*) to (a'_i,b'_i) intersects l at some point (a'',b''). Since (a',b') is the minimum on l, $D(a',b') \le D(a'',b'')$. Hence, the relationship of the values of $D(a,b)$ at those points is $D(a^*,b^*) \le D(a'_i,b'_i) < D(a',b') \le D(a'',b'')$. However, that is a contradiction to the convexity of $D(a,b)$ since (a'',b'') is a convex combination of (a'_i,b'_i) and (a^*,b^*) which implies that there are two local minima. Hence, any optimal solution must lie to the right of l. The argument for the second condition is symmetrical to the above discussion. In the third condition, $(a,b,D(a',b'))$ is a supporting hyperplane for the convex polyhedron $(a,b,D(a,b))$. Note that, due to the convexity of $D(a,b)$, if an optimal solution does not lie on the test line l, then all the optimal solutions lie on one side of l. Also note that the line along which we compute the derivative at (a',b') need not be a data line; any line through (a',b') could be used to determine a decreasing direction.

If there is more than one data line which intersects l at the point (a',b'), then testing only one line is not sufficient; on the other hand, we cannot afford to test each line to determine a direction in which $D(a,b)$ decreases. As is seen from the previous discussion, we have only to find a line passing through (a',b') along which $D(a,b)$ decreases to one side in order to determine the relative position of (a^*,b^*). If no such line exists, (a',b') is a local minimum and, due to the convexity of $D(a,b)$, is also a global minimum. The result may be determined by considering three vertical test lines in the (a, b)-plane.

We first consider the line $l' : a = a'$, in the (a, b)-plane. We compute the partial derivatives at (a', b') along the vertical line l'; if $D(a, b)$ decreases along l' in one direction, then we are done. Otherwise, (a', b') is the minimizing point on l'. Note that (a', b') is the weighted median point of the intersection points of the data lines and the line l'.

Next we consider the two lines $l'^{+} : a = a' + \epsilon$ and $l'^{-} : a = a' - \epsilon$, where ϵ is a very small number. We choose ϵ small enough such that the relative positions of the data lines which lie above and below (a', b') are preserved with respect to the minimizing points on l'^{+} and l'^{-}. The choice of ϵ guarantees that, if an optimal point (a^*, b^*) lies to the right (left) of l', then it also lies to the right (left) of l'^{+} (l'^{-}). The proof may be based on the properties of the weighted median-belt, the set of all weighted median points: the weighted median-belt is a-monotone and the points of the belt lie on the data lines (see Section 6). Since an optimal solution (a^*, b^*) is a weighted median point with respect to the intersection points of the data lines and a vertical line $a = a^*$, and since we are looking for an optimal point which lies at the intersection of two data lines, we know that the closest possible optimal solution to (a', b') would have to lie at an intersection of one of the incident data lines with one of the nonincident data lines. Hence, the choice of ϵ guarantees that such a point does not lie in between l'^{+} and l'^{-}.

Consider the line l'^{+}. If we find the minimizing point on l'^{+} and consider the line, l'', incident to that point and (a', b'), then we can determine if an optimal line lies to the right, with respect to l'', of l as follows. Compute the lhs and rhs derivatives along l'' at (a', b').

1. If the lhs derivative is positive, then (a^*, b^*) lies to the left, with respect to l'', of l.
2. If the rhs derivative is negative, then (a^*, b^*) lies to the right, with respect to l'', of l.
3. Otherwise, test l'^{-}.

The argument for the first and second cases is based on the convexity of $D(a, b)$ as shown above. The first and second cases also hold for testing l'^{-}.

If we arrive at the third case in testing l'^{-}, then (a', b') must be the global minimum. Suppose (a', b') is not the global minimum; then we know that there is a line through (a', b') which decreases towards the optimal solution. But that line must pass through one of l'^{+} or l'^{-} at a point (a'', b''). Suppose it passes through l'^{+}. Since we have already found the minimizing point on l'^{+}, we know that the value of $D(a'', b'')$ is greater than or equal to the value at (a', b'). Since the previous cases have failed, we know that the value of $D(a, b)$ at (a', b'), the minimizing point on l', is less than at the minimizing point on l'^{+}. Hence, $D(a', b') < D(a'', b'')$, which contradicts the assertion that $D(a, b)$ is decreasing in the direction of (a'', b'').

We now show that the above testing may be done without specifying a specific ϵ. Recall that we need to consider an ϵ small enough such that, if (a', b') is not an optimal solution, then the optimal solution lies to the right (left) of l'^{+} (l'^{-}). First, we note that if (a^*, b^*) is an optimal solution which is incident to at least one data line, then it is the weighted median point of the intersection points of the data lines with the vertical line $a = a^*$. Second, note that the weighted median points form an a-monotone chain defined by line segments on the data lines with endpoints determined by the intersection of data lines as described in **Section 6**. Hence, the minimizing point on l'^+ (l'^-) must lie at the intersection with one of the lines incident to (a', b').

Note, however, that we do not need to determine the actual minimizing point on l'^+ or l'^-; we only need to determine the data line which determines the weighted median point and take the derivative along that line at (a', b'). In that case, we may consider the intersection points of the data lines incident at (a', b') with any line to the right (left) of l' and assume that the other lines intersect above or below those intersection points according to whether they intersected l' above or below, respectively, (a', b'). Now, the data line which determines the weighted median point at a vertical line within ϵ of l' may be determined in linear time.

5.2.3. The Region R_0

In this section we describe the basis of the *modified technique*. The objective of the modified technique is to somehow retain the information given by the tests in order that further calculations may be performed efficiently. In the basic technique, we determine a region which is not intersected by a constant factor of the remaining lines; those lines are then assumed to be eliminated. However, as we have already seen (in the one dimensional minimization problem), we may need some information from the "eliminated" lines. The modified technique does not directly provide that information; rather, it retains the information from the test result: the relative positions of all optimal points with respect to all eliminated lines is known. The application determines how to use that retained information. We motivate the discussion by considering the vertical L_1 problem, however, note that the technique may be used for a large class of problems.

Note that the method of finding the minimizing point on a nonvertical test line l described above is a simple example of the principle of the modified technique. In that application, each test not only eliminated a constant factor of lines but also determined a region (on the line l) which contained an optimal solution and whose relative position with respect to all the eliminated lines remained fixed. Hence, the contributions from eliminated lines to further computations remained fixed and the total time spent at each iteration was linear with respect to the number of remaining data lines. The same concept will be applied to the search for an optimal point (a^*, b^*) in the (a, b)-plane.

Recall the description of the basic pruning technique described in **Section 4**. The problem with the basic technique is that the computations occur at random locations depending on the pairing of the lines. Hence, the contribution from eliminated lines must be recomputed at each iteration. However, as seen by the one-dimensional example above, by controlling the region in which computations occur, the contributions of eliminated constraints to the computations may be accounted for in constant time. The

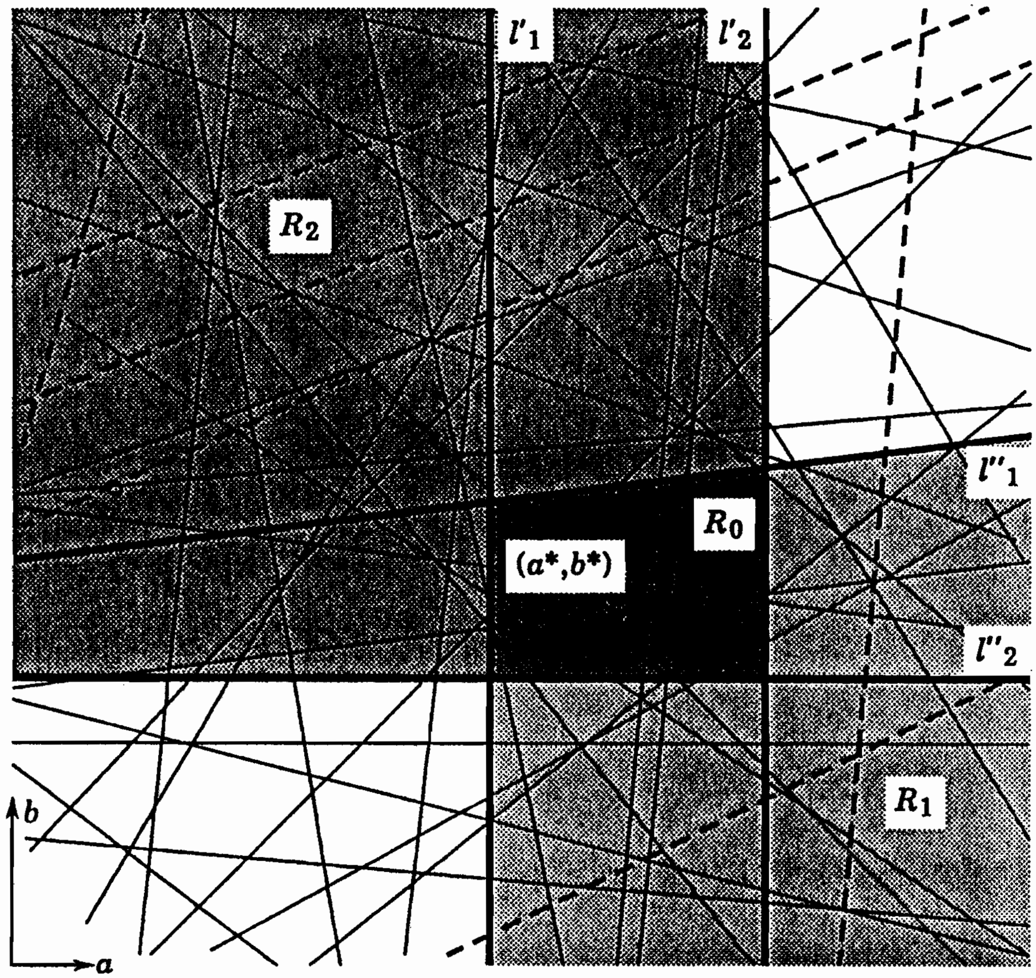


Figure 5.5. After a second iteration: the regions R_1 and R_2.

main characteristic of the modified technique is that the computations are performed within a specific region, R_0 which, in effect, maintains the necessary information from previous iterations.

Consider the situation after the first application of the pruning technique. The two lines l' and l'' divide the (a,b)-plane into four regions. One of those four regions, called R_1, is known to contain an optimal solution. The region, R', which is "opposite to" (does not share an edge with) R_1, is known to contain $n/8$ intersection points $P_{l''} = \{p''_s\}$, determined by the intersection of two data lines l_s and l_t, one of which, say, l_s, can be eliminated.

Let $E_1 = \{l_s\}$ represent the set of eliminated lines in the first application of the pruning technique. The relative position, with respect to R_1, of the lines in E_1 can be easily determined from the result of the test of the nonvertical test line. If the optimum solution lies above the nonvertical test line, then the eliminated lines lie below the region; otherwise, they lie above.

$D(a,b)$ may be written in terms of the remaining, or *active*, lines, $H' = H \backslash E_1$, and the eliminated lines, E_1, as follows. Suppose the lines in E_1 lie above the region R_1.

$$\begin{aligned} D(a,b) &= \sum_{i \in H'} w_i |y_i - ax_i - b| + \sum_{i \in E_1} w_i |y_i - ax_i - b| \\ &= \sum_{i \in H'} w_i |y_i - ax_i - b| + Y^+ - aX^+ - bW^+, \end{aligned}$$

where

$$W^+ = \sum_{i \in E_1} w_i, \; X^+ = \sum_{i \in E_1} w_i x_i, \; Y^+ = \sum_{i \in E_1} w_i y_i.$$

Hence, as long as the computations can be restricted within R_1, the region constructed by a second application of the pruning technique, R_2, can be constructed in linear, $O(|H'|)$, time with respect to the remaining constraints. But now there are two sets of eliminated constraints, E_1 and E_2 (for example, Figure 5.5); hence, further computations would have to be restricted to a region whose relative position with respect to both sets of eliminated constraints is known. The intersection of the two regions R_1 and R_2 is such a region.

Let $R_0 = \cap_{i=1,\ldots,m} R_i$, where m is the last iteration number. R_0 is initially set to the whole plane and hence contains all optimal solutions. Since each R_i contains all optimal solutions, $R_0 = \cap_{i=1,\ldots,m} R_i$ contains all optimal solutions as does the region R_{m+1}. Note that, since R_0 is the intersection of half-planes, R_0 is convex. Also, since at most two edges are added at each iteration and there are at most $O(\log n)$ iterations, R_0 has at most $O(\log n)$ edges and the total complexity of maintaining R_0 is $O(\log^2 n)$ [Pr79] which is negligible.

Let $E_0 = \cup_{i=1,\ldots,m} E_i$, where m is the last iteration number. Since the relative position of (the lines in) E_i with respect to R_i is the same as the relative position of E_i with respect to R_0, the relative position of all the lines in E_0 with respect to R_0 is known. Let $E_0 = E^+ \cup E^-$, where E^+ and E^- are the sets of eliminated lines above and below R_0, respectively. Let $H' = H \backslash E_0$. $D(a,b)$ may now be written in terms of the contributions of the active lines, H', and the eliminated lines, E_0.

$$\begin{aligned}
D(a,b) &= \sum_{i\in H'} w_i|y_i - ax_i - b| + \sum_{i\in E_0} w_i|y_i - ax_i - b| \\
&= \sum_{i\in H'} w_i|y_i - ax_i - b| + \sum_{i\in E+} w_iy_i - aw_ix_i - bw_i \\
&\quad - \sum_{i\in E^-} w_iy_i - aw_ix_i - bw_i \\
&= \sum_{i\in H'} w_i|y_i - ax_i - b| + \sum_{i\in E+} w_iy_i - \sum_{i\in E+} aw_ix_i - \sum_{i\in E+} bw_i \\
&\quad - \sum_{i\in E^-} w_iy_i + \sum_{i\in E^-} aw_ix_i + \sum_{i\in E^-} bw_i \\
&= \sum_{i\in H'} w_i|y_i - ax_i - b| + Y^+ - aX^+ - bW^+ - Y^- + aX^- + bW^-,
\end{aligned}$$

where

$$\begin{aligned}
W^+ &= \sum_{i\in E+} w_i,\ X^+ = \sum_{i\in E+} w_ix_i,\ Y^+ = \sum_{i\in E+} w_iy_i, \\
W^- &= \sum_{i\in E^-} w_i,\ X^- = \sum_{i\in E^-} w_ix_i,\ Y^- = \sum_{i\in E^-} w_iy_i.
\end{aligned}$$

W^+, X^+, Y^+, W^-, X^-, and Y^- may be updated after each iteration and saved for computations in the next iteration. Hence, the computations performed in the region R_0 may be computed in linear time with respect to the number of lines in H', since the contribution from eliminated lines may be accounted for in constant time.

5.2.4. The Modified Test

Next, a method for restricting the computations to the region R_0 is described. Recall that the problem is caused by the computations required by the testing of a line. The major problem is that, in the test described above, the minimizing point along the given test line must be determined; once determined, the one-dimensional derivatives are computed at that point to determine a result. However, the minimizing point may not lie inside the region R_0. The solution is to devise a new test which restricts the computations to the region R_0.

The modified test tests a line, l, against the region R_0 as follows. First, compute the intersection points, r_l and r_r, of l and the boundary of R_0. If l does not intersect R_0, then compute the position of R_0 with respect to l. Since R_0 contains all the optimal solutions, the position of R_0 determines which side of l an optimal solution exists (see **Figure 5.6(a)**). Otherwise, at least one of r_l and r_r exists; note that, since R_0 may be unbounded, only one of r_l and r_r may exist. Since it takes $O(\log n)$ to determine the

We must show that, in the sequence of elementary steps, the vertices of the weighted median-belt appear in a left to right order. The following lemma proves that the order of the vertices of the weighted median-belt will be encountered in left to right order.

Lemma 6.5. *The topological sweep of Edelsbrunner and Guibas* [EG86] *sweeps an arrangement of n lines, H, such that the vertices $p_1, p_2, \ldots, p_n$, $p_i : (a_i, b_i)$, of any monotone chain determined by the intersection of lines in the arrangement are encountered in left to right order, $a_i < a_{i+1} (i = 1, \ldots, n-1)$.*

Proof: *We consider the nondegenerate case in which all vertices are determined by only two lines (the reason for this is that the algorithm transforms degenerate, or multiple, intersections of $i > 2$ lines into $i - 1$ nondegenerate intersections). We will show that if a point p_i is to be swept by the next elementary step, then the point p_{i-1} must already have been swept by the arrangement, which implies that the points must be swept in left to right order. Suppose that edges c_i and c_{i+1} in the cut determine the next elementary step at vertex p_i of the monotone chain (see* **Figure 6.7***). One of them, say c_i, determines an edge of the monotone path with vertices p_{i-1} and p_i. Now consider the incident edges to the right of p_{i-1}. One of them must be c_i. Hence, c_i can only enter the cut after p_{i-1} has been swept. Since c_i is in the cut, p_{i-1} must have been swept. Hence, the vertices of the monotone chain must be swept in left to right order.* □

Next we show that we can use the topological sweep to find the weighted median-belt.

Note that the leftmost edge of the weighted median-belt can be determined in linear time by a weighted median selection algorithm (see [S76]). The weighted median-belt may be constructed by keeping track of the current edge of the belt. We start with the leftmost edge. We can detect when the sweep processes a median-belt vertex in constant time by checking each elementary step for the current edge. After a median-belt vertex is processed, there are only two choices for the next median-belt edge. Since the contributions of the lines above and below the current edge have already been computed and since only a constant factor of lines switch their relative positions, we can decide in constant time, which of the two edges is the next weighted median edge by checking if **Lemma 6.4** holds for either the line determined by c_i or c_j. Since there are $O(n^2)$ elementary steps we have the following result.

Lemma 6.6. *The vertices of the weighted median-belt may be found in optimal, $\Theta(n^2)$, time and $O(n)$ space by the topological sweep of* [EG86].

6.3.3. Computing the Optimal Solution

As the topological sweep proceeds, the optimal solution may be computed as follows. First, after the leftmost edge has been found, the six variables are initialized in linear time. Next, suppose that c_i and c_j in the cut determine the next elementary step and c_i

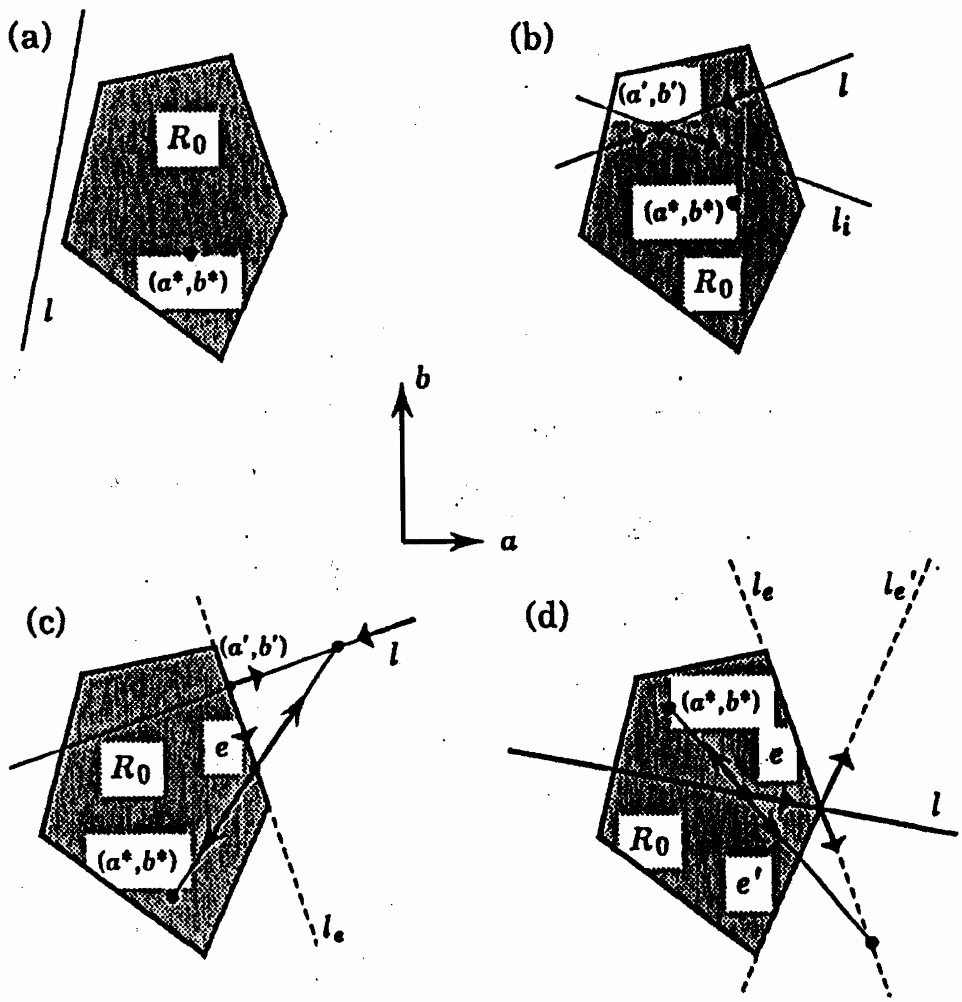


Figure 5.6. $D(a,b)$ decreases to side which contains an optimal solution.

intersections at each step, and there are $O(\log n)$ steps, the time required to determine the intersections does not factor in the total time complexity of the algorithm.

Compute the one-dimensional *lhs* and *rhs* derivatives along l at r_l and r_r. If the *rhs* derivative at r_l (if it exists) is negative and the *lhs* derivative at r_r (if it exists) is positive, then the minimizing point, (a', b'), on l lies inside the region R_0 and the basic test may be applied since the contributions from eliminated data lines are available in constant time. In **Figure 5.6**(b) the arrows indicate the direction in which $D(a, b)$ decreases along the line (the result of computing the derivatives). First the derivatives are computed along l at its intersection with the boundary of R_0. The minimum point, (a', b') is easily determined as stated above, and the derivative is computed along l_i (recall that (a', b') lies at the intersection with a data line, l_i) to determine the relative position of an optimal point (a^*, b^*).

Otherwise, consider the point r_r, the point r_l is handled in a similar fashion. If the *rhs*-derivative at r_r is positive, then the point, (a', b'), which minimizes $D(a, b)$ on l lies outside of R_0 to the right of r_r. In that case, the derivatives at (a', b') cannot be computed in the required time since the relative positions of the eliminated lines are not known at (a', b'). However, the side which does contain an optimal solution may still be determined as follows. Consider the edge of R_0 which intersects l at r_r (see **Figure 5.6(c)**); note that there would be two edges if r_r is a vertex (see **Figure 5.6(d)**). First, consider the case in which r_r is incident to only one edge e. Consider the line l_e which contains the edge e. Compute the derivatives along l_e at r_r.

Lemma 5.4. *An optimal solution lies on the side of l which has $D(a, b)$ decreasing from r_r along l_e. The side to which $D(a, b)$ decreases can be determined by computing the lhs and rhs derivatives at r_r along l_e. There are only two cases: if the lhs derivative is positive, then the optimal solution lies on the left-hand side, with respect to l_e, of l; otherwise, the rhs must be negative and the optimal solution lies on the right-hand side of l.*

Proof: *First, note that $D(a, b)$ can only decrease in one direction along l from r_r because of the convexity of $D(a, b)$: if $D(a, b)$ decreased in both directions (styled arrows in* **Figure 5.6(d)***), then there would be at least two local minima which contradicts the convexity of $D(a, b)$. Recall that the optimal point cannot lie on l since the minimizing point on l lies outside of R_0. Hence, there are only the two above mentioned cases to consider.*

The proof is identical for both cases. An optimal solution does exist and, furthermore, if there are more than one optimal solution, then they all lie on the same side of l; if not, there would have to be an optimal solution which lies on l, but we have already stated that that is not the case. Since $D(a, b)$ decreases along l_e to only one side of l, $D(a, b)$ must be increasing along l_e in the other direction. Let the increasing side denote the side of l on which $D(a, b)$ increases along l_e. Similarly, let the decreasing side denote the side of l on which $D(a, b)$ decreases along l_e.

The proof is essentially the same as for deciding the side in the basic test. Suppose an optimal solution, (a^*, b^*), lies on the increasing side (see **Figure 5.6(c)***). Consider the line segment, s, from that optimal solution to the minimizing point on l. The line segment intersects l_e at a point (a'', b'') on the increasing side. Consider the values of $D(a, b)$ at the points (a^*, b^*), (a'', b''), and (a', b'). The three points all lie on the line segment s but the values of $D(a, b)$ at the endpoints, (a^*, b^*) and (a', b'), are less than at the point (a'', b''), which is a contradiction to the convexity of $D(a, b)$; hence, the optimal solution lies to the decreasing side.* □

If r_r is a vertex of R_0, then a similar discussion applies. Let e and e' denote the edges of R_0 incident to r_r. Let l_e and $l_{e'}$ denote the lines obtained by extending the edges e and e', respectively. Consider the *lhs* and *rhs* derivatives at r_r along l_e and $l_{e'}$. $D(a, b)$ can only decrease to one side along l_e and $l_{e'}$. Suppose the contrary: $D(a,b)$ decreases along l_e to one side of l and $D(a,b)$ decreases along $l_{e'}$ to the opposite side of l. Consider the minimizing points on l_e and $l_{e'}$ which lie on opposite sides of l (see **Figure 5.6(d)**, the straight edge arrows indicate the direction in which those minimizing points supposedly lie). Consider an optimal point on the opposite side of either of those minimizing points. The line segment connecting the optimal point to the minimizing point on the opposite side of l intersects l at a point with greater $D(a,b)$ value than at the endpoints. But that is a contradiction to the convexity of $D(a,b)$ hence $D(a,b)$ decreases to only side along l_e and $l_{e'}$. Now the argument that the optimal solution lies on the decreasing side is the same as above.

We have noted that all the computations may be performed in linear time with respect to the remaining, or active lines. Since only a constant number of lines have to be tested, we have the following result.

Lemma 5.5. *Given a line, l, in the (a,b)-plane, the Modified Test determines the relative position of an optimal solution in $O(H')$ time, where H' is the set of active lines.*

The algorithm for the weighted vertical L_1 linear approximation problem may be summarized as follows. For each step i:

(i) Determine region R_i and save the contributions from eliminated lines. If the optimal solution was found during the tests, stop and report.

(ii) Update $R_0 = R_0 \cap R_i$ and go back to step (i).

We showed that the time spent in step (i) is linear with respect to the number of remaining data lines. At each step, 1/8 of the remaining lines are eliminated. The total time spent for step (ii) is $O(\log^2 n)$ and hence is negligible since the linear time complexity of such an algorithm was illustrated in **Section 4**.

We formalize the result in the following theorem.

Theorem 5.1. *The weighted vertical L_1 linear approximation problem of a set of points in the plane may be solved in optimal, $\Theta(n)$, time and $O(n)$ space.*

5.3. Summary

The algorithm is notable for several reasons. First, the algorithm finds an exact solution to the problem as opposed to an approximate solution obtained by numerical approximation techniques. Second, the worst case complexity of the algorithm improves the previous bounds for finding an exact solution by two orders of magnitude. Third, the modification of the pruning technique is general enough to be applied to problems with similar structure in order to derive optimal algorithms.

The basic pruning technique has been applied to a variety of problems and has taken different forms according to the particular problem being solved; however, as far as the author knows, the above modified technique is the first time the basic technique has been applied in such a way to produce a linear time algorithm for a class of problems. While the above modification does not directly generalize to d-dimensions, the above concept has been generalized to maintaining a tetrahedral region to solve the vertical L_1 method in three dimensions [IK88]. That result may be generalized to higher dimensions by maintaining a simplex in d-dimensions.

6. Orthogonal L_1 Linear Approximation Problem

This section presents algorithms for the unweighted and weighted orthogonal L_1 linear approximation problems in the plane (referred to as the orthogonal L_1 problems). An $\Omega(n \log n)$ lower bound for the orthogonal L_1 problem is also shown. The algorithms find exact solutions to both the unweighted and weighted orthogonal L_1 problems, as opposed to approximate solutions derived by numerical approximation techniques. The results are of interest for at least three reasons. First, the algorithms improve upon previous $O(n^3)$ naive algorithms for finding an exact solution. Second, the unweighted orthogonal L_1 problem is shown to be an application of the k-belt of an arrangement of lines. Although [EW86] use the k-belt to solve several problems, none of the problems are as directly related to the notion of a k-belt as the orthogonal L_1 problem. We show how the k-belt algorithm of [EW86] may be used to efficiently solve the unweighted orthogonal L_1 problem. Third, the generalized concept of a weighted k-belt, or, more precisely, the weighted median-belt is introduced and it is shown that the unweighted k-belt algorithm is not optimal for finding the weighted median-belt. A basic technique in computational geometry, the topological sweep [EG86], is used to find the vertices of the weighted median-belt in optimal, $\Theta(n^2)$, time, and $O(n)$ space, and it is shown that the weighted orthogonal L_1 problem may be solved at the same time as the belt is computed.

6.1. Preliminaries

The primal problem and two characteristic properties of an optimal solution are given. Next, the dual problem is described and the dual version of the two properties are given. A method for maintaining $D(a, b)$ is described; the method is essentially the same technique which was described in **Section 5** except that the application is different.

6.1.1. The Primal Problem

The weighted orthogonal L_1 problem in the (x, y)-plane may be stated as follows.

> **Problem 6.1. The Primal Problem.** *Given a set, S, of n points, p_i : (x_i, y_i) ($i = 1, \ldots, n$), in the (x, y)-plane, with corresponding weights, w_i, find a pair of values (a^*, b^*), for the parameters a and b, which solves the following mini-sum problem:*

$$\min_{a,b} \quad D(a, b) \equiv \sum_{i=1}^{n} w_i \frac{|y_i - (ax_i + b)|}{\sqrt{a^2 + 1}}.$$

The unweighted primal problem corresponds to setting all the weights equal to one.

Morris and Norback [MN80] present two characteristic properties of an optimal solution to the primal problem. The first property is known as the *incidence property*.

Lemma 6.1. *There exists an optimal approximation line, $l^* : y = a^*x + b^*$, for the (weighted) orthogonal L_1 problem which is incident to two points of S.*

Lemma 6.1 suggests an $O(n^3)$ brute force approach to solving the primal problem. For each of the possible $\binom{n}{2} = O(n^2)$ pairs of points, compute the orthogonal L_1 norm for a line which passes through the pair. The pair which determines a line which minimizes the L_1 norm must be an optimal solution according to **Lemma 6.1**. Note, however, that **Lemma 6.1** does not characterize all optimal solutions.

The second property, known as the *(weighted) median-set property*, characterizes all optimal solutions.

Lemma 6.2. *The sum of the weights of points on an optimal approximation line, l^*, is greater than the absolute difference of the sums of weights of the points above and below l^*.*

$$|W^+ - W^-| < W^0,$$

where

$$W^+ = \sum_{i \in I^+} w_i,\ W^- = \sum_{i \in I^-} w_i,\ W^0 = \sum_{i \in I^0} w_i,$$

and

$$\begin{aligned} I^+ &= \{i \mid y_i - ax_i - b > 0\}, \\ I^- &= \{i \mid y_i - ax_i - b < 0\}, \\ I^0 &= \{i \mid y_i - ax_i - b = 0\}. \end{aligned}$$

Although **Lemma 6.2** characterizes all optimal solutions, it does not give much insight into finding lines which satisfy the weighted median-set property, or *weighted median lines*.

[MN80] suggest finding *candidate solutions*, pairs of data points in the (x, y)-plane which satisfy both properties, in order to solve the orthogonal L_1 problem. They suggest the brute force approach of inspecting each of the possible $\binom{n}{2}$ pairs to see if the line defined by the pair satisfies the second property ($O(n)$ time); if so, then compute the L_1 norm ($O(n)$ time) with respect to the *candidate line* defined by the pair of points. The advantage is that some division and multiplication operations may be avoided. In terms of asymptotic complexity, however, that algorithm is not a great improvement over computing the L_1 norm for every possible pair ($O(n^3)$ time) since just to check the second property for each pair takes $O(n^3)$ time.

[MN80] note that the number of pairs of points satisfying the second condition may be significantly smaller than $O(n^2)$, the total number of pairs. For example, in **Figure 6.1(a)**, there are only 8 candidate lines out of a total 21 possible lines satisfying only **Lemma 6.1**. However, they do not investigate or give references to any known bounds on the number of candidate lines. Their approach raises three questions. How many candidate lines are there? What is the complexity of finding the candidate lines?

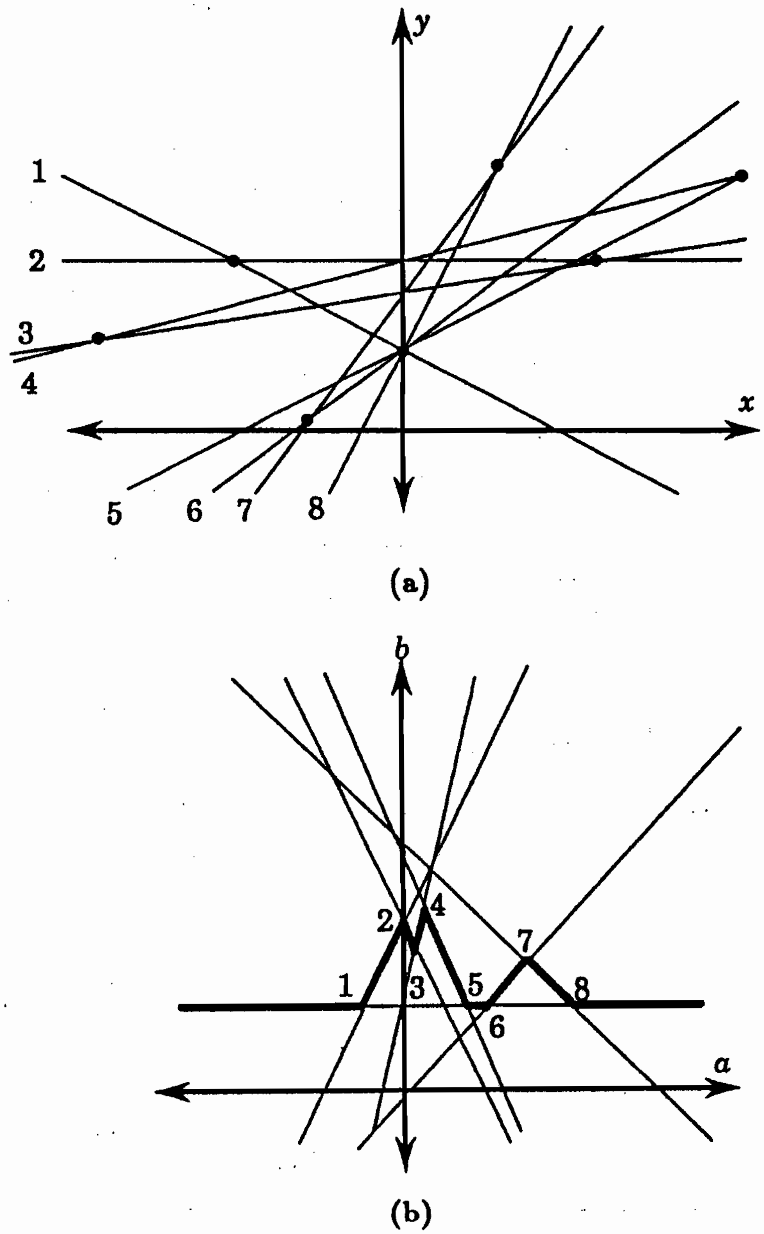


Figure 6.1. (a) Candidate lines. (b) Corresponding candidate vertices on median-belt.

What is the complexity of finding the candidate line which minimizes the orthogonal L_1 norm?

In the unweighted problem, **Lemma 6.2** is just a cardinality problem:

$$\big||I^+| - |I^-|\big| < |I^0|,$$

the absolute difference in the number of points above and below the line must not exceed the number of points on the line. Hence, candidate lines in the unweighted orthogonal L_1 problem correspond to the lines $\overrightarrow{pq}$ of k-graphs for $k = \lceil (n-2)/2 \rceil$, called the *median-lines.* (see **Figure 6.1(a)**). Hence, the answer to the first question, for the unweighted problem, is that there are $O(n^{1.5})$ candidate solutions due to [ELSS73] (see **Section 4**). Note, however, that [ELSS73] conjecture that the upper bound is actually closer to the lower bound of $\Omega(n \log n)$; hence, by considering only the candidate solutions we may be able to derive an efficient algorithm for the unweighted orthogonal L_1 problem.

In the weighted orthogonal L_1 problem, however, lines which satisfy **Lemma 6.2** are not as easy to count in the (x, y)-plane. By considering the counting problem in the dual plane, however, a worst case example may be easily illustrated. Furthermore, by considering both the unweighted and weighted orthogonal L_1 problems in the dual plane, the candidate points may be found by using basic techniques in computational geometry.

6.1.2. The Dual Problem

The weighted orthogonal L_1 problem in the (a, b)-plane may be stated as follows.

Problem 6.2. The Orthogonal L_1 Problem. *Given a set, H, of n lines, $l_i : b = -x_i a + y_i$ $(i = 1, \ldots, n)$, in the (a, b)-plane, with corresponding weights, w_i, find a pair of values (a^*, b^*), for the parameters a and b, which solves the following mini-sum problem:*

$$\min_{a,b} \quad D(a, b) \equiv \sum_{i=1}^{n} w_i \frac{|y_i - (ax_i + b)|}{\sqrt{a^2 + 1}}.$$

The unweighted primal problem corresponds to setting all the weights equal to one. As in the vertical L_1 problem, the objective function, the (orthogonal) L_1 norm, remains the same in the primal and dual problems. Note that, in the vertical L_1 problem the dual transformation preserves the geometry of the problem: the vertical distance from a data point to an approximate line in the (x, y)-plane is the same as the vertical distance from the transformed data line to an approximate point in the (a, b)-plane. However, in the orthogonal L_1 problem such a direct relationship does not hold. Nonetheless, the dual problem exhibits a geometry of the orthogonal L_1 problem not apparent in the (x, y)-plane.

The dual versions of **Lemma 6.1** and **Lemma 6.2** are now stated. The results hold in the dual plane since the dual transformation preserves the incidence and above-below relationships between points and lines.

Lemma 6.3. *There is an optimal point, (a^*, b^*), for the (weighted) orthogonal L_1 problem which is incident to two lines in H.*

Note that an $O(n^3)$ brute force algorithm may be devised for the dual problem based on **Lemma 6.3**: compute the value of the L_1 norm, $O(n)$ time, at each of the $O(n^2)$ intersection points of each pair of lines in H.

The second property is known as the *(weighted) median-belt property.*

Lemma 6.4. *The sum of the weights of lines incident to an optimal point (a^*, b^*) is greater than the absolute difference of the sums of weights of the lines above and below (a^*, b^*).*

$$|W^+ - W^-| < W^0,$$

where

$$W^+ = \sum_{i \in I^+} w_i,\ W^- = \sum_{i \in I^-} w_i,\ W^0 = \sum_{i \in I^0} w_i,$$

and I^+, I^-, and I^0 are as defined in **Lemma 6.2.**

In the unweighted case, *candidate points*, points which satisfy **Lemma 6.3** and **Lemma 6.4**, are the vertices of k-belts for $k = \lceil n/2 \rceil$, or median-belts, as defined in **Section 4**. The bold line in **Figure 6.1(b)** represents the median-belt for the seven data lines which are duals of the seven data points in **Figure 6.1(a)**; the vertices of the median-belt are the dual transformations of the eight candidate lines in the (x, y)-plane. Hence, we arrive at the same $\Omega(n \log n)$ lower bound and $O(n^{1.5})$ upper bound on the number of vertices (due to [EW86]) as in the primal problem (see **Section 4**). Furthermore, the vertices of the median-belt may be found in $O(n^{1.5} \log^2 n)$ time by the "sophisticated plane sweep algorithm" of [EW86].

In the weighted case, however, we will show that the number of vertices is $\Theta(n^2)$. We also show that the plane sweep algorithm of [EW86] may not be the best algorithm for computing the weighted median-belt by giving a method which has better worst case time complexity.

Finally, note that the (upper and lower chains of the) unweighted and weighted median-belts are a-monotone chains, (an orthogonal line from the a-axis intersects the belt in only one point). If the vertices of the belt are given in increasing a-coordinate, then the following method for maintaining $D(a, b)$ may be used to efficiently compute the value of the orthogonal L_1 norm at each vertex, or candidate solution, thereby determining an optimal solution.

6.1.3. Maintaining $D(a, b)$

We assume that the candidate points, $p_i : (a_i, b_i)$ $(i = 1, \ldots, n)$, are given in sorted order with increasing a-coordinates $(a_i < a_{i+1},\ i = 1, \ldots, (n-1))$. We will show that once the value of $D(a, b)$, the orthogonal L_1 norm, has been computed at a point p_i using the method presented below, then the value of $D(a, b)$ at the point p_{i+1} can be computed in constant time in the nondegenerate case.

$D(a,b)$ may be written as follows.

$$
\begin{aligned}
D(a,b) &= \sum_{i=1}^{n} w_i \frac{|y_i - (ax_i + b)|}{\sqrt{a^2+1}} \\
&= \sum_{i \in I^+(a,b)} w_i \frac{y_i - (ax_i + b)}{\sqrt{a^2+1}} - \sum_{i \in I^-(a,b)} w_i \frac{y_i - (ax_i + b)}{\sqrt{a^2+1}} \\
&= \frac{1}{\sqrt{a^2+1}} \left\{ \sum_{i \in I^+(a,b)} w_i y_i - a \sum_{i \in I^+(a,b)} w_i x_i - b \sum_{i \in I^+(a,b)} w_i \right. \\
&\qquad \left. - \sum_{i \in I^-(a,b)} w_i y_i + a \sum_{i \in I^-(a,b)} w_i x_i + b \sum_{i \in I^-(a,b)} w_i \right\} \\
&= \frac{a(X^- - X^+) + b(W^- - W^+) + Y^+ - Y^-}{\sqrt{a^2+1}},
\end{aligned}
$$

where

$$
\begin{aligned}
W^+ &= \sum_{i \in I^+} w_i, \quad X^+ = \sum_{i \in I^+} w_i x_i, \quad Y^+ = \sum_{i \in I^+} w_i y_i, \\
W^- &= \sum_{i \in I^-} w_i, \quad X^- = \sum_{i \in I^-} w_i x_i, \quad Y^- = \sum_{i \in I^-} w_i y_i,
\end{aligned}
$$

and

$$
\begin{aligned}
I^+(a,b) &= \{\, i \mid y_i > ax_i + b \quad (i = 1, \ldots, n) \}, \\
I^-(a,b) &= \{\, i \mid y_i < ax_i + b \quad (i = 1, \ldots, n) \}.
\end{aligned}
$$

Note that, by computing the values for the variables W^+, X^+, Y^+, W^-, X^-, and Y^-, the value of $D(a,b)$ can be computed in constant time for fixed I^+ and I^-. Also, and of more importance here, is that for small changes in the sets I^+ and I^- (a constant number of elements entering or leaving the set), the values of those variables may be updated in constant time.

In order to solve the orthogonal L_1 problems, the value at all the possible candidate solutions will be computed in order to determine an optimal solution. If those calculations are performed in a naive manner, then the computation time alone would be $O(|C|n)$, where C is the set of candidate solutions, which would exceed the cost of finding the candidate solutions. The above representation for $D(a,b)$ allows us to compute the values for all the candidate solutions with $O(n)$ preprocessing time, and then a constant time for each step of the algorithm.

Consider the situation at a candidate point p_i on the median-belt (the following discussion applies to both the weighted and unweighted problems although the weighted case is more general as will be described later). Suppose that the sets I^+ and I^- have been determined for p_i and the values for W^+, X^+, Y^+, W^-, X^-, and Y^- have been computed. Suppose that the point $p_i : (a_i, b_i)$ is determined by two of the data

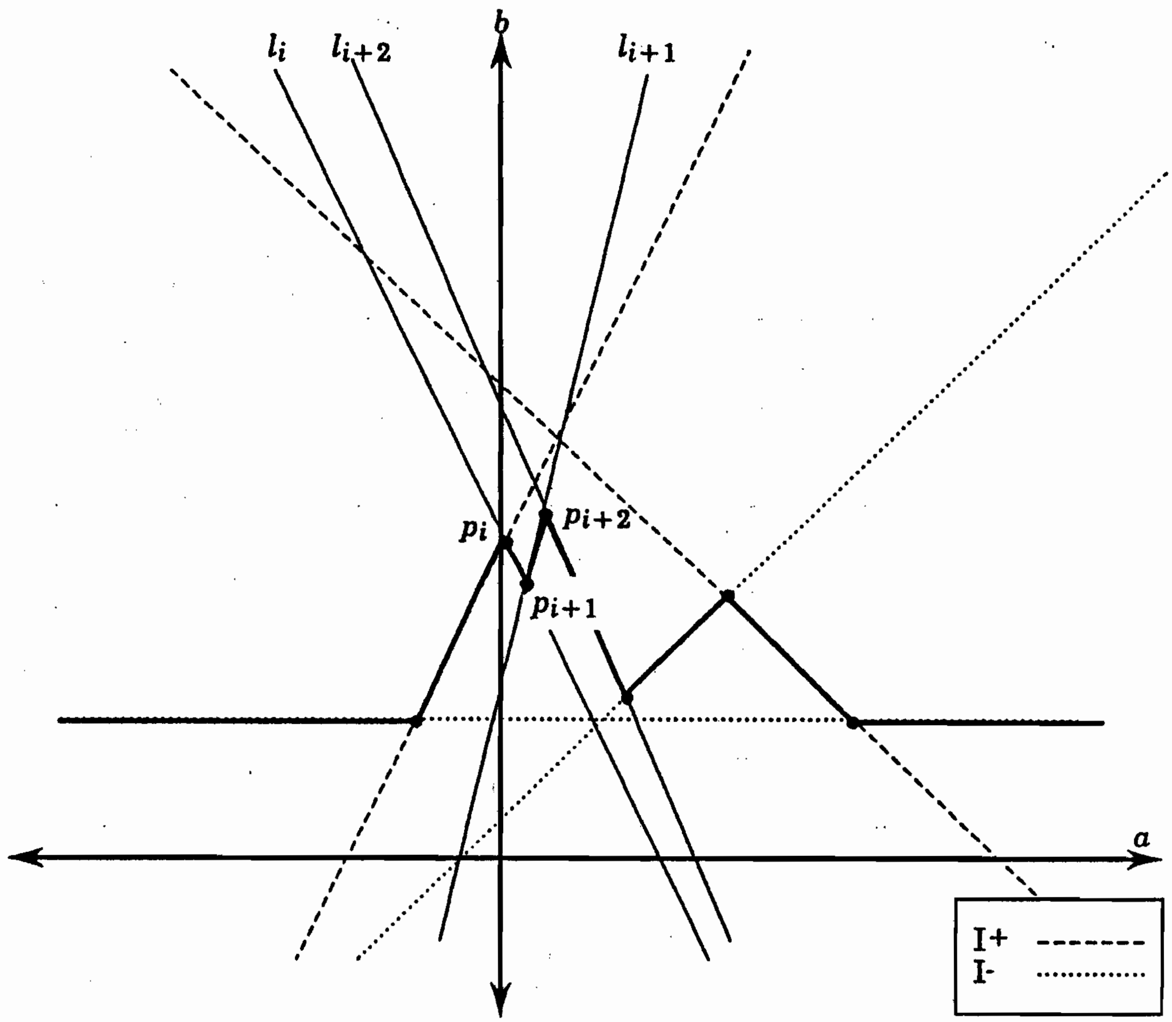


Figure 6.2. Maintaining $D(a,b)$.

lines l_i and l_{i+1}, where l_i determines the median-belt edge for $a_{i-1} < a < a_i$, and l_{i+1} determines the median-belt edge for $a_i < a < a_{i+1}$ (see **Figure 6.2**). Similarly, consider the candidate point p_{i+1} which is determined by the data lines l_{i+1} and l_{i+2}.

Note that, in the nondegenerate case, only a constant number of lines change their positions relative to the belt (and hence relative to I^+ and I^-) (see **Figure 6.2**). At p_i, l_{i+1} was removed from I^- without upsetting the balance since it is now incident to the candidate point. However, for $a_i < a < a_{i+1}$, l_i enters I^- to restore the balance. A similar exchange takes place at p_{i+1} except that l_{i+2} is removed from I^+ and l_{i+1} enters I^+.

The point of the above discussion is that between consecutive vertices of the median-belt the sets I^+ and I^- change by only a constant number of data lines in the nondegenerate case. The degenerate case is discussed in the descriptions of the algorithms. Since the sets of lines only change by a constant number of data lines, the variables

W^+, X^+, Y^+, W^-, X^-, and Y^- may be updated in constant time by simply adding or subtracting the contributions of the data lines which change sets. Hence, if the candidate points are processed in a sorted order with respect to the a-coordinate and given in terms of the incident data lines, then the value of the L_1 norm at any vertex may be computed in constant time given the values for the variables at the previous vertex. The preprocessing consists of initializing the variables at the leftmost point which can easily be done in $O(n)$ time.

6.2. The Unweighted Problem

In the unweighted orthogonal L_1 problem, the dual transformations of the candidate lines correspond to the vertices of the median-belt (see **Figure 6.1**). [EW86] present an $O(b_k(n)\log^2 n)$ time and $O(b_k(H))$ space algorithm for computing the k-belt, where $b_k(n)$ is the number of vertices in a k-belt of n-lines and $b_k(H)$ is the number of vertices in a k-belt for a given set H of lines. Conceptually, the algorithm sweeps a vertical line from left to right and reports only the vertices of interest; hence the sorted order of the candidate points is guaranteed. In our application, we are interested in the $k = \lceil n/2 \rceil$ or median-belt and since we do not need to store the whole k-belt the algorithm takes $O(n^{1.5}\log^2 n)$ time and $O(n)$ space. The details of the algorithm are not explained since the algorithm is applied directly (see [EW86] for the details). We show that the value of the L_1 norm can be computed while the algorithm proceeds, and that the time spent on such computations does not exceed the time complexity of the algorithm.

The algorithm sweeps an arrangement of lines by a vertical line L from left to right by maintaining two copies of a data structure which computes the intersection of n half-planes (determined by the data lines in H) in $O(n \log n)$ time [OL81]. Furthermore, the data structure allows the efficient insertion and deletion of lines in $O(\log^2 n)$ time and can report the adjacent edges of a given edge in constant time. The two data structure are used to maintain the sets of lines, H^+ and H^-, which lie above and below, respectively, a particular point on the median-belt. The intersection of the half-planes bounded below by lines in H^- is stored in one data structure, called $\mathcal{H}^-$ (hopefully, the double notation will not introduce any confusion) (see **Figure 6.3**); similarly, the intersection of the half-planes bounded above by lines in H^+ is stored in the other data structure, called $\mathcal{H}^+$.

At $a = -\infty$, the data line, l_m, with the median slope also has the median intersection with L. Let H^+ be the set of data lines above l_m and similarly H^- the set of data lines below l_m with respect to the median intersection point on L. The sets I^+ and I^- refer to the indices of the lines in H^+ and H^-, respectively, and the variables used to maintain $D(a,b)$ can be initialized in linear time.

At each step the algorithm returns the line which determines the next edge of the median-belt in $O(\log^2 n)$ time as follows (see **Figure 6.3**). Suppose the current edge of the median-belt is determined by l_m. We wish to find the leftmost intersection of the data lines with l_m. First, l_m is inserted into $\mathcal{H}^+$ which may or may not cause a new

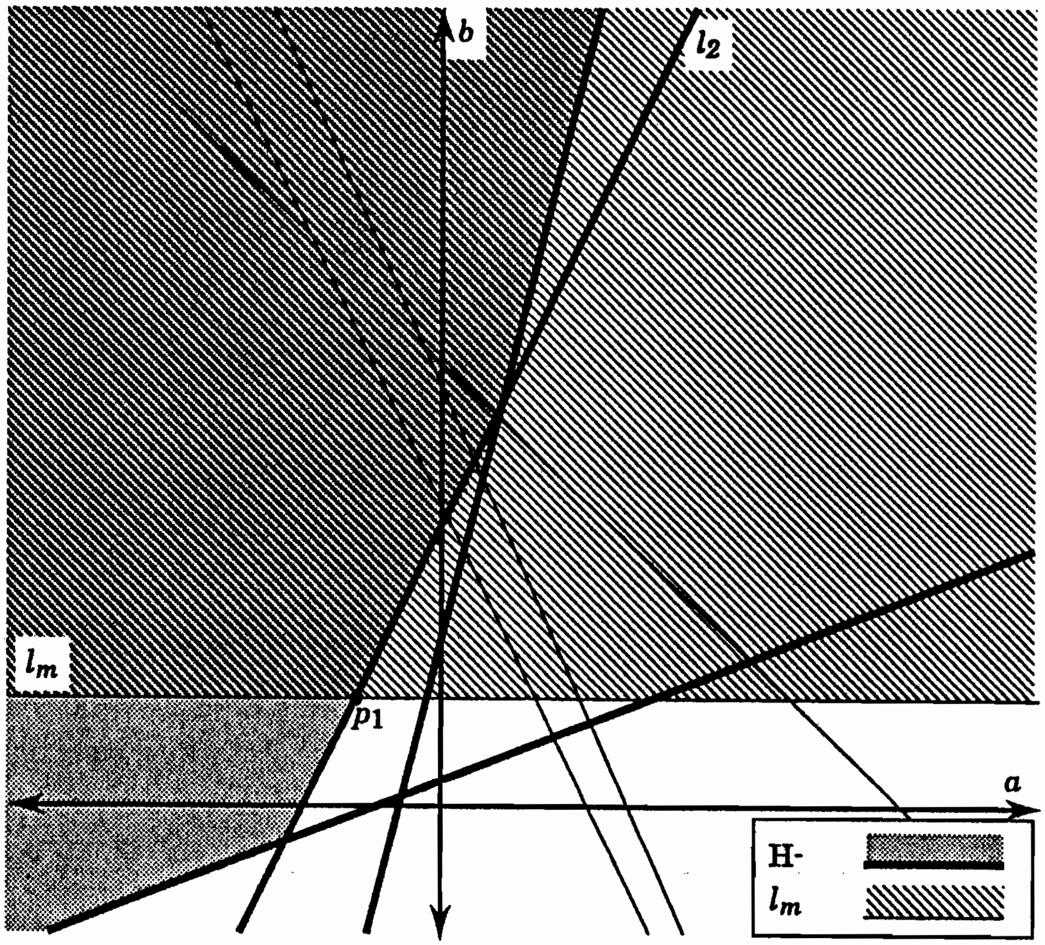


Figure 6.3. Determining the next median-belt edge.

edge; if it does create a new edge then the adjacent edge, which determines the leftmost intersection of the lines in $\mathcal{H}^+$ with l_m, can be found in constant time. Second, the identical operation is performed with l_m and $\mathcal{H}^-$ to determine the leftmost intersection of the lines in $\mathcal{H}^-$ and l_m. Suppose that, of those two intersection points, the leftmost point is caused by a line l_2 in $\mathcal{H}^-$. Then the next vertex, p_1, lies at the intersection of l_m and l_2. To reflect the median-belt as we sweep past p_1, l_m is left in $\mathcal{H}^-$ but deleted from $\mathcal{H}^+$, l_2 is deleted from $\mathcal{H}^-$, and l_2 now determines the current edge of the median-belt.

Now we show how the value of $D(a,b)$ is maintained at each step of the above algorithm. After determining l_m, the leftmost edge, suppose the algorithm finds that the next edge lies on the line $l_2 : b = -x_2a + y_2$; without loss of generality, assume that the line l_2 is is in the set H^+. Hence the first candidate point, p_1, lies at the vertex determined by the lines l_m and l_2. The value of $D(a,b)$ may be computed at p_1 in constant time as follows. Since l_2 is leaving H^+, subtract x_2 from X^+ and y_2 from Y^+. Those operations take constant time and $D(a,b)$ may be computed in constant time with the updated values. l_m must lie above the next candidate point so we add x_m to

X^+ and y_m to Y^+ and a similar process takes place for each subsequent line determined by the algorithm.

[EW86] show that degenerate vertices, vertices at which $i > 2$ data lines intersect, can be handled in $O(i \log^2 n)$ time which is less time than it would take to process those lines if they determined $\binom{i}{2}$ intersection points. Updating the variables in such a case would only take $O(i)$ time, hence the degenerate case poses no problem in terms of the time complexity of the algorithm.

Pointers to the data lines which determine a candidate point which minimizes $D(a, b)$ may be maintained in constant time at each step. Thus, an optimal solution may be found without increasing the complexity of the k-belt construction algorithm. Since $O(n^{1.5} \log^2 n)$ time is spent in constructing the median-belt, the following result is obtained.

Theorem 6.1. *The unweighted orthogonal L_1 linear approximation problem for n points can be solved in $O(n^{1.5} \log^2 n)$ time and $O(n)$ space.*

The above algorithm, although efficient, is still an exhaustive search of all the candidate solutions. Note that the algorithm is for constructing the general k-belt; for at least one value of k, $k = 1$, a more efficient algorithm, $O(n \log n)$ time, can be obtained by considering the particular problem. Hence, we conjecture that a more efficient algorithm may be obtained by considering the particular properties of the median-belt or by improving the bound on the number of vertices of median-belts.

6.3. The Weighted Problem

In the weighted orthogonal L_1 problem, the dual transformations of the candidate lines correspond to the vertices of the weighted median-belt. Hence, the complexity of finding a solution can be related to the number of vertices, n_w, on the boundary of the weighted median-belt (due to **Lemma 6.2**). Although the weighted median-belt is a straightforward generalization of the median-belt, we have not found any mention of it in the literature. In this section, we show that the weighted median-belt can have $\Theta(n^2)$ vertices. We give an $\Theta(n^2)$ time and $O(n)$ space algorithm for finding the weighted median-belt based on the topological sweep of [EG86] and show that it can also be used to solve the weighted orthogonal L_1 problem in the same time and space complexity.

6.3.1. The Weighted Median-Belt

First, we formally define the weighted median-belt.

Definition 6.1. The Weighted Median-Belt. *Given an arrangement of n lines in the (a, b)-plane, $l_i : b = -x_i a + y_i$ $(i = 1, \ldots, n)$, with associated weights, w_i, the weighted median-belt is defined as the set of all points, $p : (a', b')$, such that*

$$|W^+ - W^-| < W^0,$$

where

$$W^{+} = \sum_{i \in I^{+}(p)} w_i, \; W^{-} = \sum_{i \in I^{-}(p)} w_i, \; W^{0} = \sum_{i \in I^{0}(p)} w_i,$$

and

$$\begin{aligned} I^{+}(p) &= \{i \mid y_i - a'x_i - b' > 0\}, \\ I^{-}(p) &= \{i \mid y_i - a'x_i - b' < 0\}, \\ I^{0}(p) &= \{i \mid y_i - a'x_i - b' = 0\}. \end{aligned}$$

Although the weighted median-belt is simply derived from the median-belt by introducing weights to the lines, the complexities of the two belts appears to be fundamentally different.

The complexity of n_w, the number of vertices on the weighted median-belt, depends on how a vertex of the median-belt is defined. A vertex can be described as either any point on the boundary of the weighted median-belt which is incident to more than one line of the arrangement (called *degenerate* vertices) (see **Figure 6.4**), or any point on the boundary of the weighted median-belt whose incident boundary edges are distinct (*nondegenerate* vertices or *turns*) (see **Figure 6.5**). Note that the nondegenerate vertices are a subset of the degenerate vertices.

The geometries of the (unweighted) median-belt and of the weighted median-belt can be quite different. In the unweighted case, the median-belt only has nondegenerate vertices since the belt switches lines at every intersection. In the weighted case, however, the weighted median-belt may not switch at an intersection if the current edge has a relatively large weight.

Since the weighted median-belt is an a-monotone chain, we investigated whether lower bounds for monotone chains may be applied to the weighted median-belt. The number of degenerate vertices in a monotone chain is $\Omega(n^2)$ [We87] (see **Figure 6.4**). The number of nondegenerate vertices in a monotone chain is $\Omega(n^{5/3})$ [M87] (see **Figure 6.5**). Note, however, that not all monotone chains are weighted median-belts. In the examples above, legal weight assignments (bold numbers) can be given to the lines, however, there are examples in which no such assignments are possible (such as the example by Sharir with $\Omega(n^{3/2})$ nondegenerate vertices [Sh87]).

6.3.2. Constructing the Weighted Median Belt

Since there are $O(n^2)$ vertices on the weighted median-belt, the complexity of applying the sophisticated plane sweep algorithm to the weighted problem is in $O(n^2 \log^2 n)$. However, all $\binom{n}{2} = O(n^2)$ intersections of the arrangement of n lines may be determined by the $O(n^2)$ time and $O(n)$ space topological sweep of Edelsbrunner and Guibas [EG86].

Note that the plane sweep of [EG86] differs from the plane sweep algorithm used for the unweighted case [EW86]. The sophisticated plane sweep computes only points of interest in order from left to right, whereas the topological plane sweep of [EG86] reports all $O(n^2)$ vertices of the arrangement in an order defined by a topological sweep

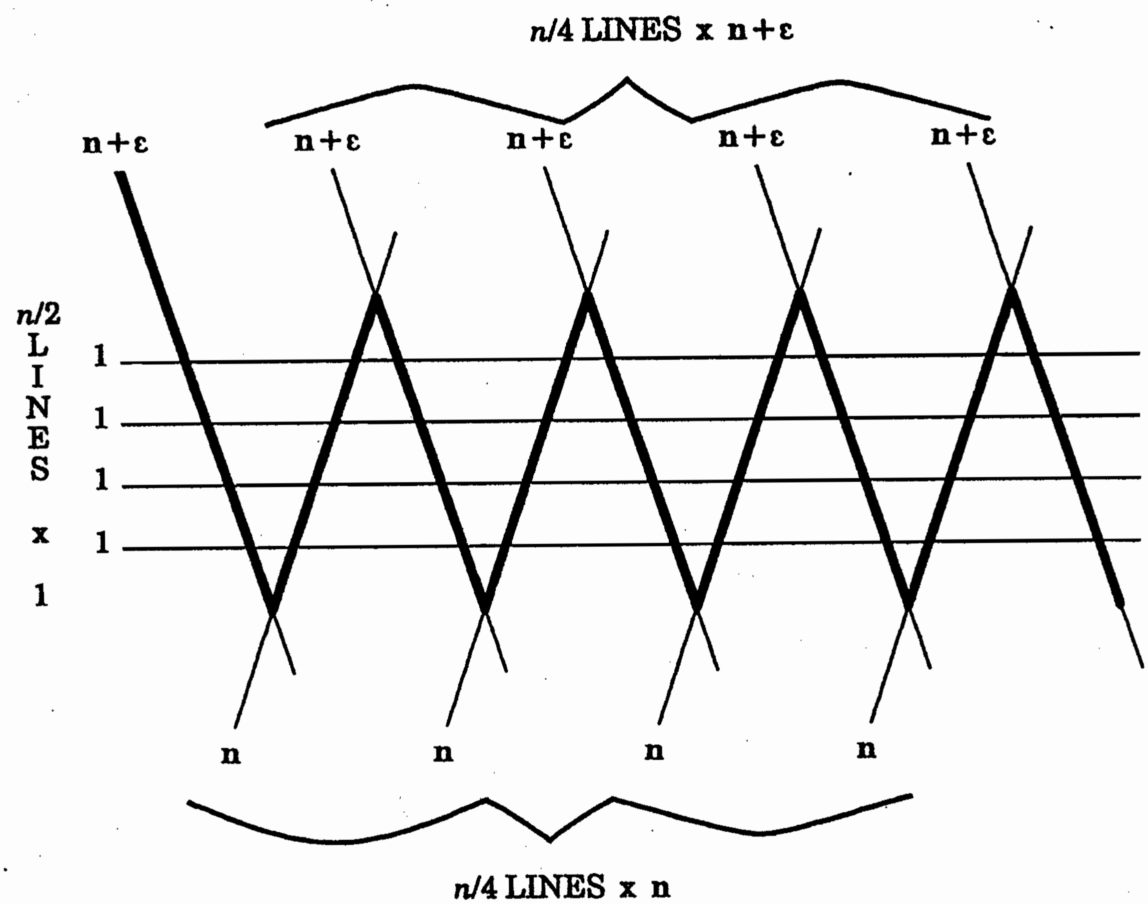
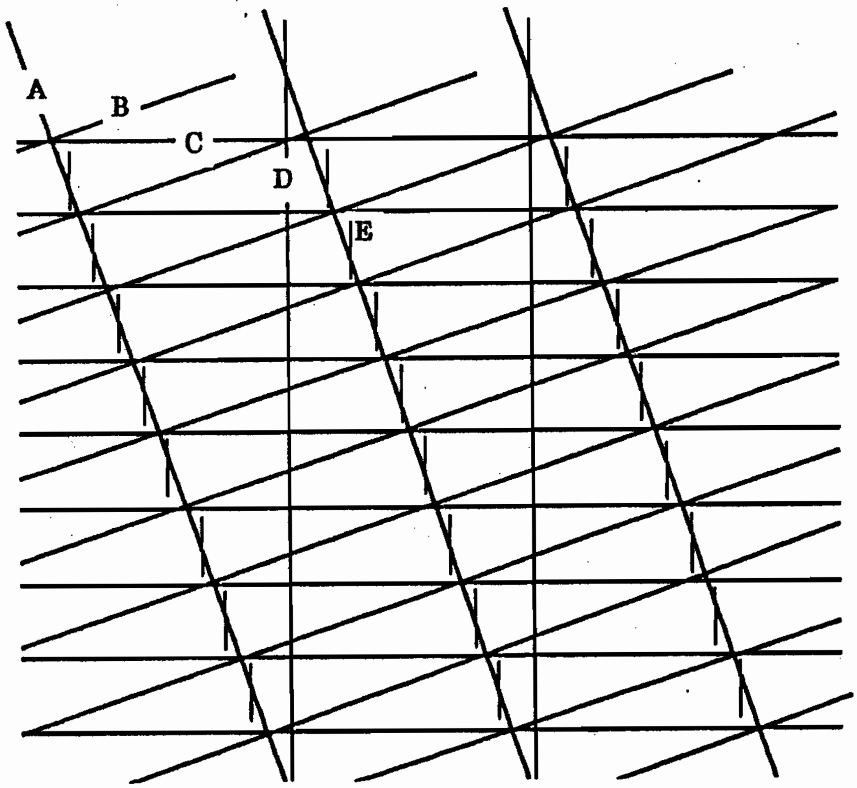


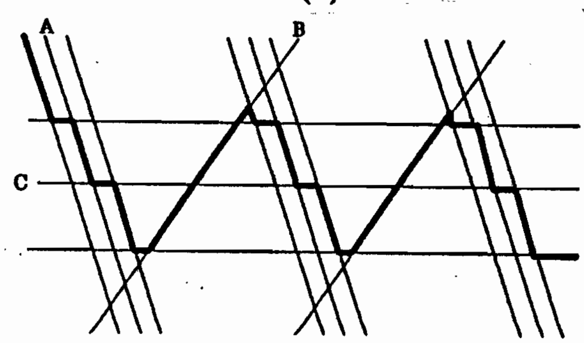
Figure 6.4. $\Omega(n^2)$ **Degenerate vertices.**

line from left to right. The topological sweep line is not a vertical line, and although it is only a conceptual line, it may be visualized by the bold line in **Figure 6.6**. In order to efficiently construct the weighted median-belt, we would like to determine the vertices of the belt in a left to right order as the plane sweep progresses; hence, we must show that the vertices of the weighted median-belt are found in a left to right order by the topological sweep.

The details of the plane sweep algorithm are beyond the scope of this thesis; however, we will need to introduce some of the concepts in order to show that the vertices will be found in a left to right order. Refer to **Figure 6.6** for the following discussion. First, we say that the lines in H form *regions* in the plane which are bounded by *edges* formed by line segments of the lines in H. The boundaries of the edges are points, called *vertices*, where the lines of H intersect. We say that a region A is above a region B if A and B have intersecting projections on the a-axis, and at each abscissa a of their intersection, all points of A are above all points of B. Note that there is exactly one region, $\mathcal{T}$, which is not below any other region; similarly, there is exactly one region, $\mathcal{B}$, which is not above any other region.



(a)



(b)

Direction	Bundles	Lines	Weight
A	m	m^2	a
B	$m^2 + m$	$m - 1$	$\mathbf{b} = m\mathbf{c}$
C	m^2	m	c
D	$m - 1$	1	$\mathbf{d} = m^2\mathbf{a}$
E	$m(m^2 - 1)$	1	$\mathbf{e} = (m^2 - 1)\mathbf{d}$

(c)

Figure 6.5. (a) An overview of the arrangement; thick lines are bundles and there are m^3, $m = \lfloor n^{1/3} \rfloor$, intersections of three bundles as shown in detail below. (b) Close up of a triple of bundles, or *fat vertex*, which has m^2 turns; hence, there are a total of $n^{5/3}$ turns. (c) General weighting for the lines such that the monotone path is a weighted median line.

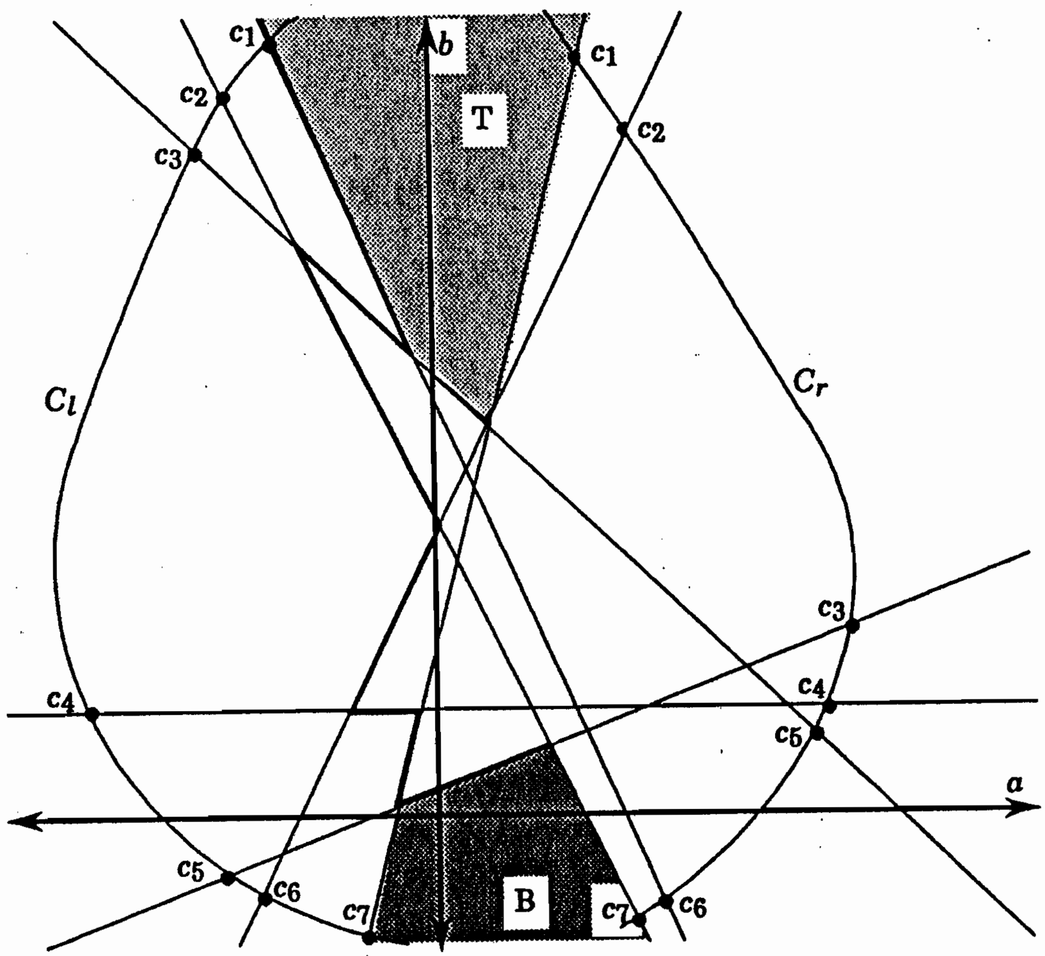


Figure 6.6. Topological sweep line: the leftmost, C_l and rightmost, C_r, cutsets.

The topological sweep line intersects the lines of H in the ordered set of edges, $c_1, c_2, \ldots, c_n$, which is called the *vertical cut* (see **Figure 6.6**). The edges are ordered such that for each i, $1 \leq i \leq (n-1)$, c_i and c_{i+1} are both incident upon the region R_i and c_i is above R_i and c_{i+1} is below R_i. Furthermore, c_1 is an edge of the region $\mathcal{T}$ and c_n is an edge of the region $\mathcal{B}$. Those two conditions imply that no two edges of the cut lie on the same line of H so that there is a one-to-one correspondence between the edges of the cut and the lines in H; also, the cut gives an ordering of the n lines in H. There is a leftmost cut, C_l, which contains all the leftmost edges of the lines, and a corresponding rightmost cut, C_r, which contains all the rightmost edges of the lines,

The sweep consists of pushing the cut from the leftmost cut to the rightmost cut by a sequence of *elementary steps*. An elementary step is performed when the topological line sweeps past a vertex. An elementary step is detected if two lines corresponding to adjacent edges in the cut, c_i and c_{i+1}, intersect to the right of the sweep line. Hence, sweeping over a vertex p_i with left incident edges c_i and c_{i+1} consists of transposing the order of the corresponding lines in the cut.

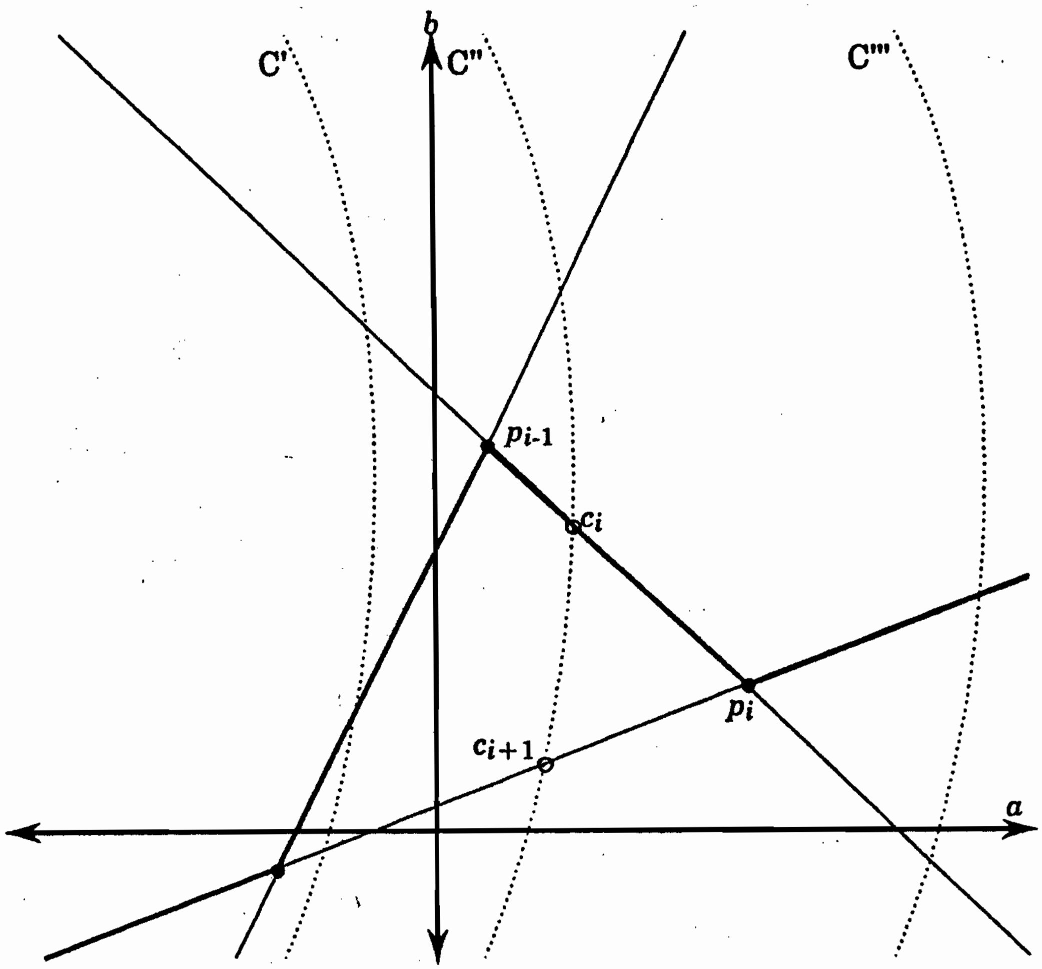


Figure 6.7. Elementary step determined by edges c_i and c_{i+1}; c_i is on the monotone path.

is the current edge of the median-belt. The intersection of the lines which determine c_i and c_j is a candidate point and we may compute the value for $D(a, b)$ in constant time since only the contribution from the line which determines c_j needs to be subtracted from the appropriate variables. Next, we determine, in constant time (since, again, the computations only involve adding or subtracting a constant number of values from the variables), which of c_i or c_j determines the next weighted median edge. Hence, the weighted median-belt and the value of $D(a, b)$ may be computed with $O(n)$ time plus a constant amount of time for each elementary step. Since there are $O(n^2)$ elementary steps, we obtain the following result.

Theorem 6.2. *The weighted orthogonal L_1 linear approximation problem for a set of points may be found in $O(n^2)$ time and $O(n)$ space.*

The algorithm may be called an efficient brute force algorithm since, even though all the vertices are visited, the computations are performed efficiently, reducing the complexity of a pure brute force search by a factor of $O(n)$. Also note that the above

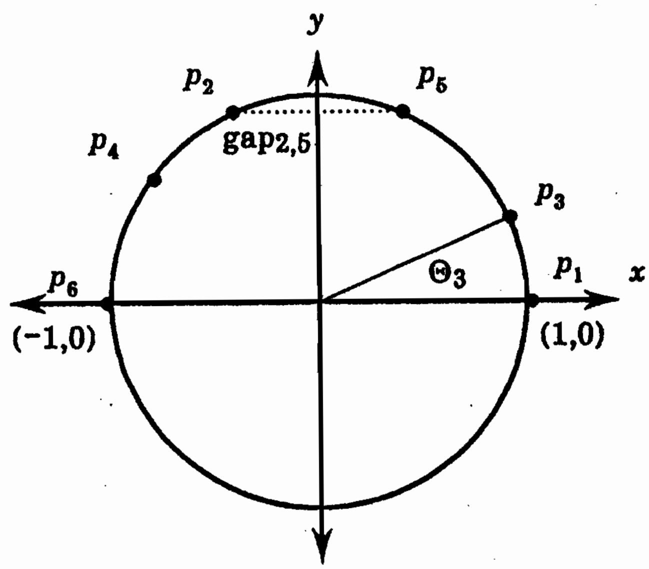


Figure 6.8. Uniform gap problem on the half unit circle.

uniformly placed under a given permutation σ. Define $W \subset \mathbf{R}^{m-1}$ by

$$W \equiv \bigcup_{\sigma} W_{\sigma},$$

where the union is taken over all permutations σ on $\{2, \ldots, m\}$. Hence, W represents the set of all possible inputs to the uniform gap on the half unit circle problem which have a yes answer.

The answer for the uniform gap problem for points $p_1, p_2, \ldots, p_{m+1}$ is yes if and only if $(x_2, \ldots, x_m) \in W$, which implies that by Ben-Or's theorem [B-O83] a lower bound of the complexity of the uniform gap problem under the algebraic computation tree is $\Omega(\log \#W)$, where $\#W$ is the number of connected components of W. Each W_σ consists of a single point in $\mathbf{R}^{m-1}$, and for two distinct permutations σ and σ', $W_\sigma \neq W_{\sigma'}$, so that the number $\#W$ of connected components of W is $(m-1)!$. Since $\log(m-1)! = \Omega(m \log m)$, we obtain the following.

Lemma 6.7. *Under the algebraic computation tree model, the complexity of the uniform gap problem on the half unit circle is $\Omega(m \log m)$.*

Next, the input for the uniform gap problem on a circle is transformed, in linear time, to an input for the L_1 problem. Given the points $p_1, p_2, \ldots, p_{m+1}$ of the uniform gap problem, and let θ_i denote the polar angle of p_i. For each p_i ($i = 2, \ldots, m$), construct the point p_{i+m} on the unit circle whose polar angle θ_{i+m} is equal to $\theta_i + \pi$ (see **Figure 6.9**). The set, S, of $n = 2m$ points $p_1, p_2, \ldots, p_n$ is then used as the input for the L_1 problem. The validity of the transformation is given by the following result.

Lemma 6.8. *The minimum objective function value of the orthogonal L_1 linear approximation for the set S of points is at most $2\cot(\pi/n)$, and is*

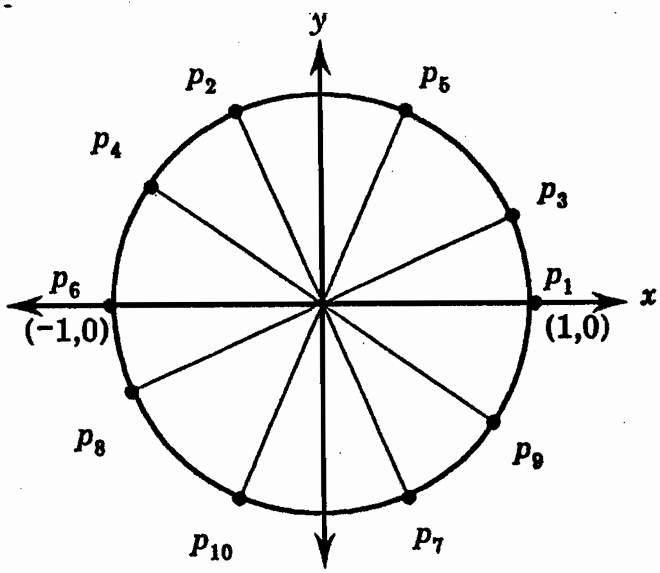


Figure 6.9. Transformed input of the UGP. for the L_1 problem.

$2\cot(\pi/n)$ if and only if the answer for the uniform gap problem of points $p_1, p_2, \ldots, p_m, p_{m+1}$ is yes.

The proof is developed as follows. We know from **Lemma 6.1** that there is an optimal approximation line such that the line passes two points among the given points and from **Lemma 6.2** that $|N^+ - N^-| < N^0$ where N^+, N^- and N^0 are the numbers of points above, below, and on the line, respectively.

By the symmetry of the transformed problem, there is an optimal approximation line, l_i, among the m lines connecting points p_i and p_{i+m} $(i = 1, \ldots, m)$. Without loss of generality, assume that $0 = \theta_1 < \theta_2 < \cdots < \theta_m < \theta_{m+1} = \pi$, where θ_i denotes the polar angle of p_i. There is no loss of generality since we are interested in showing the value of the minimum of the L_1 norm, not how to compute that value (that is done by the algorithm). The assumption simplifies the expression for $D(a, b)$ and allows us to derive a minimum.

The value for the orthogonal L_1 norm with respect to the line l_i, the summation of the orthogonal distances from points p_j $(j = 1, \ldots, n)$ to l_i, is given by

$$\sum_{j=1}^{n} |\sin(\theta_j - \theta_i)|. \tag{1}$$

Hence, the minimum function value of the orthogonal linear L_1 approximation for S is given by

$$\min_{i=1,\ldots,m} \sum_{j=1}^{n} |\sin(\theta_j - \theta_i)|. \tag{2}$$

Recall that we want to show that the maximum of (2) is $2\cot \pi/n$. We wish to maximize (2) for $0 = \theta_1 < \theta_2 < \cdots < \theta_m < \pi$ and $\theta_{i+m} = \theta_i$ $(i = 1, \ldots, m)$; however, that is an optimization problem of maxi-min type which is rather difficult to solve directly.

Observe that the function values for lines l_i are the same when the set S of points are uniformly placed on the circle. Hence, if

$$\sum_{i=1}^{m}\sum_{j=1}^{n} |\sin(\theta_j - \theta_i)| \tag{3}$$

is maximized when and only when the set S of points are uniformly placed, then (2) is maximized only in the same uniform case. Let us prove that (3) is maximized when and only when the set S of points are uniformly placed.

(3) is further expressed as

$$\begin{aligned}\sum_{i=1}^{m}\sum_{j=1}^{n} |\sin(\theta_j - \theta_i)| &= 2\sum_{i=1}^{m}\sum_{j=i+1}^{i+m} \sin(\theta_j - \theta_i)\\ &= 2\sum_{k=1}^{m}\left(\sum_{i=1}^{m} \sin(\theta_{i+k} - \theta_i)\right).\end{aligned}$$

For any $k = 1, \ldots, m$, we know that

$$0 < \theta_{i+k} - \theta_i < \pi$$

and

$$\sum_{i=1}^{m}(\theta_{i+k} - \theta_i) = k\pi.$$

Since $\sin x$ is strictly concave on the interval $[0, \pi]$, we know

$$\sum_{i=1}^{m} \sin(\theta_{i+k} - \theta_i) \le m \sin\frac{k\pi}{m}$$

and the equality holds if and only if all $\theta_{i+k} - \theta_i$ $(i = 1, \ldots, m)$ are equal. All $\theta_{i+k} - \theta_i$ $(i = 1, \ldots, m)$ are equal for any $k = 1, \ldots, m$ if and only if all $\theta_{i+1} - \theta_i$ $(i = 1, \ldots, m)$ are equal. Hence, (3) is maximized when and only when the set S of points are located uniformly on the circle.

When the set S of points are placed uniformly, the function value of the orthogonal L_1 norm is expressed by

$$\begin{aligned}\sum_{j=1}^{n} |\sin\frac{j\pi}{m}| &= 2\sum_{j=1}^{m} \sin\frac{j\pi}{m}\\ &= 2\cot\frac{\pi}{2m} = 2\cot\frac{\pi}{n}\end{aligned}$$

(the trigonemetric equality comes from [Mo]). Thus, we have shown **Lemma 6.8.**

Using the above lemmas, the following result provides a lower bound for the orthogonal L_1 problem.

Theorem 6.3. *Under the algebraic computation tree model, the complexity of the orthogonal L_1 linear approximation of n points is $\Omega(n \log n)$.*

Proof: *As mentioned above, we show that the uniform gap problem on the half unit circle is linear-time reducible to the L_1 problem. Given $m+1$ input points for the uniform gap problem, we consider the set S of n points as above, and solve the orthogonal linear L_1 approximation problem for this set S. Then, we compare the minimum function value of this problem with $2\cot(\pi/n)$, and answer yes or no to the uniform gap problem if they are equal or not, respectively. The validity of this algorithm follows from* **Lemma 6.8.**

*As to the complexity of the transformation, since n is a power of 2 we can compute $\cot(\pi/n)$ in $O(\log n)$ time using the operations of the algebraic computation tree (recall the half-angle identities, apply recursively). Hence, the problem transformation can be done in $O(n)$ total time. Since the uniform gap problem on the half unit circle has an $\Omega(n \log n)$ lower bound (***Lemma 6.7***), we obtain an $\Omega(n \log n)$ lower bound for the orthogonal L_1 linear approximation problem via transformability.* □

This result is mainly of interest to the unweighted orthogonal problem since the actual bound for the algorithm presented above is in $O(b_k(n) \log^2 n)$, where $b_k(n)$ is the number of k-sets of n points ($k = \lceil n/2 \rceil$). As mentioned above, the bounds for $b_k(n)$ are in $\Omega(n \log k)$ and $O(nk^{1.5})$ ($\Omega(n \log n)$ and $O(n^2)$, respectively, for $k = \lceil n/2 \rceil$), but Erdős, Lovász, Simmons, and Strauss [ELSS73] conjecture that the upper bound is actually closer to the lower bound of $\Omega(n \log k)$ ($\Omega(n \log n)$ for $k = \lceil n/2 \rceil$).

6.5. Summary

We have shown that the unweighted orthogonal L_1 method may be solved in $O(n^{1.5} \log^2 n)$ time and $O(n)$ space and the weighted orthogonal L_1 method may be solved in $O(n^2)$ time and $O(n)$ space. Both algorithms use plane sweep methods to determine exact solutions as opposed to iterative techniques which converge to optimal solutions from an initial guess. Both algorithms improve upon previous results for algorithms which find the exact solution. The results are a direct application of the concept of k-belts. We also introduced the concept of the weighted median-belt and showed that it can be found in optimal $\Theta(n^2)$ time.

The question remains as to the tightness of the $\Omega(n \log n)$ lower bound. Note that both algorithms are based on k-belt algorithms. While the k-belt algorithms provide an efficient method for solving the orthogonal L_1 problem, the method is a direct, or brute force type, approach and we hope that by further investigating the properties of the orthogonal L_1 method, more efficient solutions may be found.

Further research includes the investigation of the multidimensional orthogonal L_1 linear approximation problem, the fitting of a hyperplane to a set of points in $\mathbf{R}^d$. Both **Lemma 6.1** and **Lemma 6.2** generalize to d-dimensions and the d-dimensional orthogonal L_1 method can be solved by an $O(n^d)$ time and $O(n)$ space algorithm by applying the algorithms above to all the combinations of $(d-2)$ hyperplanes.

7. Conclusion

Several results were given concerning the computation and analysis of three L_1 linear approximation problems of a set of points by a line. An optimal $\Theta(n)$ time and $O(n)$ space algorithm was given for the (weighted) vertical L_1 problem. An $O(n^{1.5}\log^2 n)$ time and $O(n)$ space algorithm was given for the unweighted orthogonal L_1 problem. An $O(n^2)$ time and $O(n)$ space algorithm was given for the weighted orthogonal L_1 problem. Finally, an $\Omega(n \log n)$ lower bound was given for the orthogonal L_1 problem. The results are significant from theoretical, practical, and historical viewpoints.

The result for the vertical L_1 problem is of theoretical interest since the modification of the multidimensional search technique extends the basic technique to a larger class of problems. The results for the unweighted and weighted orthogonal L_1 algorithm are of interest from a theoretical viewpoint since they relate the complexities of the problems to k-sets and k-belts and introduce the concept of the weighted median-belt. The geometric and combinatorial complexities of k-sets and k-belts is an open problem in combinatorial geometry. We showed that the combinatorial complexity of the weighted median-belt is $\Theta(n^2)$ and gave an optimal, $\Theta(n^2)$, time and $O(n)$ space algorithm for searching through the vertices. The results presented here motivate further improvements on the bounds for the number of vertices in a median-belt. Another open problem is an improved algorithm for constructing the median-belt.

The results are of practical interest since the algorithms provide efficient algorithms for solving the most popular forms of the L_1 approximation problem. The results are of particular interest for the linear facility problem and the linear regression problem since the algorithms provide practical and efficient alternatives to the currently used methods (for example, the L_2 and L_∞ approximations).

The results are of historical interest since this is the last of the three most popular L_p approximation problems ($p = 1, 2, \infty$) to succumb to efficient (analytical) algorithms. [C83] notes that alternative criteria to the L_2 norm have been investigated since the mid-1750s when R.J. Boscovitch proposed a geometric method for solving a special case of the L_1 approximation problem. Interestingly enough, the efficient algorithms for both the L_1 and the L_∞ problems have been derived from applying basic paradigms used in computational geometry.

Continuing research includes the L_1 problem in higher dimensions, which is of particular interest to econometricians since they often consider the linear and non-linear L_p problems in higher dimensions [AS73]. We are also investigating the combinatorial complexity of the median-belt. Also, the algorithms will be implemented in order to determine their applicability in practical applications.

My hope is that the results on L_1 approximation presented in this thesis do not get categorized as yet another algorithm for L_1 approximation. While the results are of theoretical and practical interest in the field of computational geometry, we hope that these results will find their way into the mathematical literature and point out that the potential for "applications which has a long history in economy, but not probably known in other fields" [D87a] has already materialized.

References

[AHU74] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, The Design and Analysis of Computer Algorithms, Addison-Wesley, Reading, MA, 1974.

[AS73] Gautam Appa and Cyril Smith, On L_1 and Chebyshev estimation, *Mathematical Programming*, Vol. 5, 73–87, 1973.

[B-O83] Michael Ben-Or, Lower bounds for algebraic computation trees (Preliminary Report), *Proceedings of the 15th ACM Annual Symposium on Theory of Computing*, 80–86, 1983.

[BR70] I. Barrodale and F.D.K. Roberts, Applications of mathematical programming to L_p approximation, *Nonlinear Programming*, Academic Press, New York, 447–464. 1970.

[BS80] Peter Bloomfield and William Steiger, Least absolute deviations curve-fitting, *SIAM J. Sci. Comput.*, Vol. 1, No. 2, June, 1980.

[C83] Vašek Chvátal, *Linear Programming*, W.H. Freeman and Company, New York, 1983.

[D87a] Yadolah Dodge, *Statistical Data Analysis Based on the L_1 norm and related Methods*, Y. Dodge, ed., North-Holland, Amsterdam, 1987.

[D87b] Yadolah Dodge, An introduction to L_1-norm based statistical data analysis, *Computational Statistics & Data Analysis*, Vol. 5, No. 4, 239–253, September, 1987.

[D88a] Yadolah Dodge, guest ed., Second Special Issue on Statistical data Analysis Based on the L_1 Norm and Related Methods, *Computational Statistics & Data Analysis*, Vol. 6, No. 3, 1988.

[D88b] Yadolah Dodge, guest ed., Third Special Issue on Statistical data Analysis Based on the L_1 Norm and Related Methods, *Computational Statistics & Data Analysis*, Vol. 6, No. 4, 1988.

[DS88] David Dobkin and Deborah Silver, Recipes for geometry & numerical analysis-Part I: An empirical study, *Proceedings of the Fourth Annual Symposium on Computational Geometry*, Urbana-Champaign, Illinois, June 6-8, ACM, 93–105, 1988. [E87] Herbert Edelsbrunner, Algorithms in Combinatorial Geometry, Springer-Verlag, 1987.

[EG86] Herbert Edelsbrunner and Leonidas J. Guibas, Topologically sweeping an arrangement, Digital Systems Research Center Report, April, 1986, (a version of this paper also appears in *Proceedings of the 1986 ACM Symposium on Theory of Computing*, Berkeley, May, 1986).

[ELSS73] P. Erdos, L. Lovasz, A. Simmons, and E.G. Straus, Dissection graphs of planar point sets, *A Survey of Combinatorial Theory*, J.N. Srivastava et al., eds., North Holland Publishing Company, 139–149, 1973.

[ES88] Herbert Edelsbrunner and Diane L. Souvaine, Computing the median-of-squares regression lines and guided topological sweep, Technical Report No. UIUCDCS-R-88-1438, Dept. of Computer Science, University of Illinois at Urbana-Champaign, June, 1988.

[EW85] Herbert Edelsbrunner and Emo Welzl, On the number of line separations of a finite set in the plane, *Journal of Combinatorial Theory*, Series A, Vol. 38, 15–29, 1985.

[EW86] Herbert Edelsbrunner and Emo Welzl, Constructing belts in two-dimensional arrangements with applications, *SIAM Journal of Computing*, Volume 15, Number 1, 15–29, February, 1986.

[F87a] R.W. Farebrother, The historical development of the L_1 and L_2 estimation procedures, in Y. Dodge, ed., *Statistical Data Analysis Based on the L_1-Norm and Related Methods*, North-Holland, Amsterdam, 1987.

[F87b] R.W. Farebrother, The historical development of the L_1 and L_∞ estimation procedures, in Y. Dodge, ed., *Statistical Data Analysis Based on the L_1-Norm and Related Methods*, North-Holland, Amsterdam, 1987.

[G77] James E. Gentle, Least absolute values estimation: An introduction, *Communications in Statistics-Simulation and Computation*, Vol. B6, No. 4, 313–328, 1977.

[GSN88] James E. Gentle, V.A. Sposito, and Subhash C. Narula, Algorithms for unconstrained L_1 simple linear regression, *Computational Statistics & Data Analysis*, Vol. 6, No. 4, 335–339, 1988.

[H74a] H.L. Harter, The method of least squares and some alternatives I, *Int. Stat. Rev.*, Volume 42, 1974.

[H74b] H.L. Harter, The method of least squares and some alternatives II, *Int. Stat. Rev.*, Volume 42, 1974.

[H75a] H.L. Harter, The method of least squares and some alternatives III, *Int. Stat. Rev.*, Volume 43, 1975.

[H75b] H.L. Harter, The method of least squares and some alternatives IV, *Int. Stat. Rev.*, Volume 43, 1975.

[H75c] H.L. Harter, The method of least squares and some alternatives V, *Int. Stat. Rev.*, Volume 43, 1975.

[H76] H.L. Harter, The method of least squares and some alternatives VI, *Int. Stat. Rev.*, Volume 44, 1976.

[HHK88] Christoph M. Hoffman, John E. Hopcroft, and Michael S. Karasick, Towards implementing robust geometric computations, *Proceedings of the Fourth Annual Symposium on Computational Geometry*, Urbana-Champaign, Illinois, June 6-8, ACM, 106–117, 1988.

[HR88] Michael Houle and Jean-Marc Robert, personal communication, August 1988.

[Hu64] P.J. Huber, Robust estimation of a location parameter, *Ann. Math. Statist.*, Volume 35, 73–101, 1964.

[Hu72] P.J. Huber, Robust statistics: A review, *Ann. Math. Statist.*, Volume 43, 1041–1067, 1972.

[Hu73] P.J. Huber, Robust regression: Asymptotics, conjectures, and Monte Carlo, *Ann. Statist. 1*, 1041–1067, 1973.

[Hu81] P.J. Huber, Robust Statistics, John Wiley, New York, 1981.

[Hu87] P.J. Huber, The place of the L_1-norm in robust estimation, in Y. Dodge, ed., *Statistical Data Analysis Based on the L_1-Norm and Related Methods*, North-Holland, Amsterdam, 1987.

[II88] Hiroshi Imai and Keiko Imai, personal commmunication, August, 1988.

[IK88] Hiroshi Imai and Kenji Katoh, personal communication, 1988.

[LW86] D.T. Lee and Y.F. Wu, Geometric complexity of some location problems, *Algorithmica*, Volume 1, 193–212, 1986.

[M87] Jiří Matoušek, personal communication, 1987.

[MB72] Ross D. MacKinnon and G.M. Barber, A new approach to network generation and map representation: the linear case of the location-allocation problem, *Geographical Analysis*, Volume 4, 156–168, 1972.

[Me83] N. Megiddo, Linear-time algorithms for linear programming in $\mathbf{R}^3$ and related problems, *SIAM Journal on Computing*, Vol. 12, No. 4, 759–776, November, 1983.

[Me84] N. Megiddo, Linear programming in linear time when the dimension is fixed, *Journal of the Association for Computing Machinery*, Vol. 31, No. 1, 114–127, January, 1984.

[MN80] James G. Morris and John P. Norback, A simple approach to linear facility location, *Transportation Science*, Vol. 14, No. 1, 1–8, February, 1980.

[MN83] James G. Morris and John P. Norback, Linear facility location - solving extensions of the basic problem, *European Journal of Operational Research*, Vol. 12, 90–94, 1983.

[Mo] S. Moriguti, Mathematical Formulae (in Japanese), Iwanumi, Tokyo, (reference communicated by Hiroshi Imai).

[N88] Hans Nyquist, Least orthogonal absolute deviations, *Computational Statistics & Data Analysis*, Vol. 6, No. 4, 361–367, 1988.

[NM80] John P. Norback and James G. Morris, Short communication: fitting hyperplanes by minimizing orthogornal deviations, *Mathematical Programming*, Vol. 19, 102–105, 1980.

[OL81] Mark H. Overmars and Jan van Leeuwen, Maintenance of configurations in the plane, *Journal of Computer and System Sciences*, Vol. 23, 166–204, 1981.

[P72] R.L. Plackett, The discovery of the method of least squares, *Biometrika*, Volume 59, pp. 239–251, 1972.

[PFTV86] W.H. Press, B.P. Flannery, S.A. Teulosky, and W.T. Vetterling, Numerical Recipes, Cambridge University Press, Cambridge, MA, 1986.

[Pr79] F.P. Preparata, An optimal real-time algorithm for planar convex hulls, *Communications of the ACM*, Volume 22, 402–405, 1979.

[R64] J. Rice, *The Approximation of Functions, Volume 1: The Linear Theory*, Addison-Wesley, Reading, MA, 1964.

[S76] Michael Ian Shamos, *Geometry and statistics: Problems at the interface*, in Algorithms and Complexity, New Directions and Recent Results, J.F. Traub, ed., Academic Press, Inc., 251–280, 1976.

[S78] Michel Ian Shamos, Computational Geometry, Ph.D. Thesis, Yale University, 1978.

[Sh87] Micha Sharir, personal communication from Emo Welzl, 1987.

[SoCR88] S.A. Soliman, G.S. Christensen, and A. Rouhi, A new technique for curve fitting based on minimum absolute deviations, *Computational Statistics & Data Analysis*, Vol. 6, No. 4, 341–351, 1988.

[St73] Stephen M. Stigler, Studies in the history of probability and statistics. XXXII, Laplace, Fisher, and the discovery of the concept of sufficiency, *Biometrika*, Vol. 60, No. 3, 439–445, 1973. [St81] Stephen M. Stigler, Gauss and the invention of least squares, *The Annals of Statistics*, Vol. 9, No. 3, 465–474, 1981.

[SW87] H. Spath and G.A. Watson, On orthogonal linear L_1 approximation, *Numerische Mathematik*, Vol. 51, 531–543, 1987.

[W59] Harvey M. Wagner, Linear programming techniques for regression analysis, *Journal of the American Statistical Association*, Vol. 54, 206–212, 1959.

[Wa40] A. Wald, The fitting of straight lines if both variables are subject to error, *Ann. Math. Statist.*, Volume 11, 285–300, 1940.

[We87] Emo Welzl, personal communication, 1987.