点集合のL_1線形近似問題に対するアルゴリズム

山本ピーター*, 加藤研児*, 今井桂子†, 今井浩*

*九州大学工学部情報工学科

†九州工業大学情報科学センター

本論文では、平面上のn個の点を直線で近似する問題で、各点と直線とのEuclid距離の総和、すなわちL_1ノルムが最小になるような近似直線を求める問題を考える。この問題の計算複雑度は、L_1近似の最適解のある特徴付けにより$O(n^3)$であることは容易にわかるが、それより改善できるかどうかはわかっていなかった。本論文では、L_1ノルムで単に距離の総和をとる場合に対して、$O(n^{1.5}\log^2 n)$の手間のアルゴリズムを与え、また各点に重みが与えられ、L_1ノルムが各点の重みを掛けた距離の総和で定められる場合に対して、$O(n^2)$の手間のアルゴリズムを与える。一方、代数計算木モデルの下で、この問題の計算複雑度の下限が$\Omega(n \log n)$であることも示す。本論文で与えるアルゴリズムは、L_1近似問題と直線の配置構造での(重み付き)中間ベルト(median belt)を関連付けている。

Algorithms for L_1 Linear Approximation of Points

Peter YAMAMOTO*, Kenji KATO*, Keiko IMAI† and Hiroshi IMAI*

*Department of Computer Science and Communication Engineering
Kyushu University, Fukuoka 812, Japan

†Information Science Center, Kyushu Institute of Technology
IIzuka, Fukuoka 820, Japan

This paper presents algorithms for approximating a set of n points by a linear function, or a line, that minimizes the L_1 norm of orthogonal distances. The algorithmic complexity of the problem appears not to have been investigated, although O(n^3) naive algorithms can be easily obtained based on some simple characteristics of optimal L_1 solutions. In this paper, 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 problem. An $\Omega(n \log n)$ lower bound for the orthogonal L_1 problem is shown under a certain model of computation. Also, the complexity of solving the orthogonal L_1 problem is related to the construction of the k-belt of an arrangement of lines.

1. Introduction

Approximating a set of n points in the plane by a linear function, or a line, called the line-fitting problem, is a fundamental problem in scientific computing encountered in many fields, including statistics, econometrics, location theory, and signal-processing. Recently, the line-fitting problem and its variations have been considered from an algorithmic point of view in those areas. However, the algorithms presented are, for the most part, brute force and involve enumeration of all possible candidate solutions. Solutions to some of these problems, on the other hand, have inherent geometrical properties which have generated interest from the area of discrete and computational geometry.

The nature of the problem depends on three factors: the distance function used to measure the distance from a point to a line, the norm of the distance function used, and whether the distances are weighted by associating a weight to each point. The unweighted case corresponds to the situation where all weights are equal to one. Vertical or orthogonal distances are commonly used as the distance function (see **Figure 1.1**), although other measures, such as the rectangular distance, are also of interest.

The most frequently used norm is the L_2 norm which may be efficiently solved by the least squares method for both vertical and horizontal distances. However, the L_2 norm is not always the most appropriate criteria for a "best" fit. The two most popular alternatives are the L_1 and the L_∞ (or Chebyshev) norms; however, their use in practice has been limited due to the lack of efficient algorithms (see the conclusion for a brief historical note). The full paper provides more details and references concerning the applicability of the above three approaches to particular problems.

Megiddo [12] notes that by applying his multidimensional search technique to the vertical L_∞ problem, formulated as a linear programming problem, an exact solution may be found in linear time. Recently, Lee and Wu [10] provide optimal and efficient algorithms for the orthogonal L_∞ problem and some of its variations.

Optimal solutions to both the vertical and orthogonal L_1 problems are known in statistics as robust estimators: the solutions are not easily influenced (relative to solutions to the L_2 and L_∞ problems) by outliers, or noise, in the data (see **Figure 1.2**). For that reason, in applications in which the data is subject to error, for example, signal processing, the L_1 norms may be much more preferable than the more popular L_2 norms. Also, from the viewpoint of location problems, the orthogonal L_1 norm measures the total Euclidean distance of the points from the line.

This paper is concerned with the L_1 linear approximation problems in the plane. The problems may be stated as follows.

Problem 1.1. The Vertical L_1 Problem. *Given a set, S, of n points, $p_i \equiv (x_i, y_i)$ $(i = 1, \ldots, n)$, 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:*

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

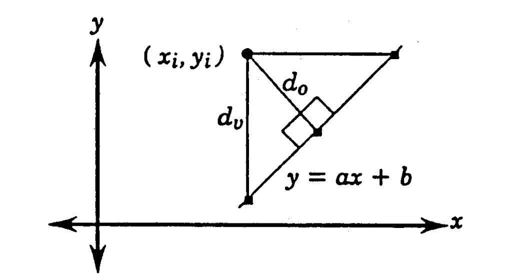

Figure 1.1. Vertical, d_v, and orthogonal, d_o,distances.

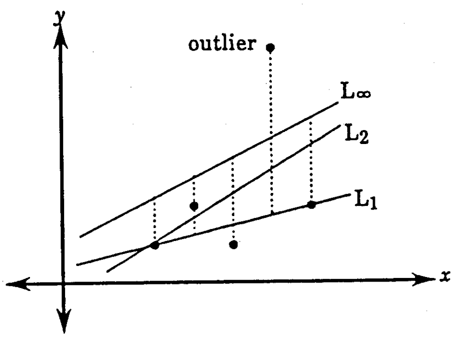

Figure 1.2. Effect of outliers on optimal solutions (vertical L_1 norm).

Problem 1.2. The Orthogonal L_1 Problem. *Given a set, S, of n points, $p_i \equiv (x_i, y_i)$ $(i = 1, \ldots, n)$, 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:*

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

The complexity of the vertical and orthogonal L_1 problems appears not to have been investigated from an algorithmic point of view before our work in [8]. The problems differ only by the divisor $\sqrt{a^2+1}$ which converts the vertical distance into the orthogonal distance; however, the divisor is an important factor in determining the complexity of the problems. The vertical L_1 norm is a convex function, whereas, due to the divisor, the orthogonal L_1 norm is not convex. Based on the convexity of the vertical L_1 norm, this paper presents a linear time algorithm for the weighted vertical L_1 problem; however, the results in this paper indicate that algorithms for the orthogonal L_1 problems lie in a different complexity class.

The unweighted vertical L_1 problem may be solved by general numerical approximation methods, such as the method of descent, or it may be formulated as a linear programming problem in higher dimensions [2]. [8] notes an $O(n^3)$ naive algorithm for finding an exact solution to the weighted vertical L_1 problem based on characteristic properties of optimal solutions.

In the case of the unweighted orthogonal L_1 problem, [15] presents a numerical algorithm which corresponds to a concave quadratic programming algorithm. [13] presents an $O(n^3)$ naive algorithm for finding an exact solution to the weighted orthogonal L_1 problem based on characteristic properties of optimal solutions.

Efficient algorithms for these problems have been proposed by Yamamoto, Kato, Imai and Imai [17]. Specifically, that paper gives an $O(n)$-time algorithm for the (weighted) vertical L_1 problem, using the so-called pruning technique, an $O(n^{1.5}\log^2 n)$-time algorithm for the unweighted orthogonal L_1 problem, based on an algorithm for constructing the median belt, and an $O(n^2)$-time algorithm for the weighted orthogonal L_1 problem, by means of the topological sweep of [5]. This paper describes the algorithms for the orthogonal problem as well as the lower bound proof to that problem in detail, following [17].

2. Preliminaries

This section defines the basic notation and concepts used in this paper. All of the algorithms make use of a point-line, or dual, transformation which is defined below. Some general notation is also introduced.

The transformation from the original, or (x, y)-, plane into the dual plane may be performed as follows (see **Figure 2.1**). The point (x', y') and the line $y = a'x + b'$ in the (x, y)-plane are mapped to the line $b = -x'a + y'$ and the point (a', b'), respectively, in the (a, b)-plane.

The following notation is used for describing the relative position of a point with respect to a line. A point (x', y') lies *above*, *on*, or *below*, the line $y = a'x+b'$ if $y' - (a'x' + b')$ is, respectively, greater than, equal to, or less than, zero. If the line is defined by $x = a'$ (a vertical line), then (x', y') lies to the *right* of, *on*, or to the *left* of, the line $x = a'$ if $x' - a'$ is, respectively, greater than, equal to, or less than, zero. The definitions also hold for a point (a', b') and a line $(b = x'a + y')$ in the (a, b)-plane. The terms right and left are also used when considering the one-dimensional space defined by the points on a line (the line, vertical or not, can be thought of as a horizontal axis).

The duality preserves the vertical distance (hence, also the above-below) and incidence relations between points and lines. A vertical line in one plane is conceptually mapped to a point at infinity in the other plane.

The value of the orthogonal L_1 norm will be referred to as

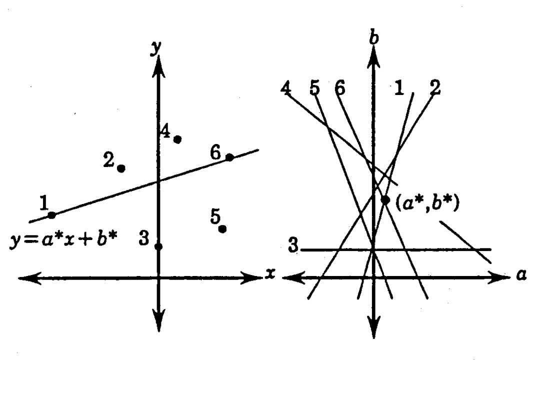

Figure 2.1. Primal-Dual Transformation

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

The absolute value signs may be eliminated by considering the value of the functions for fixed values of the parameters a and b. For fixed a and b, define sets $I_A(a, b)$ and $I_B(a, b)$ of indices by:

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

and

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

Note that the above notation may be interpreted as the set of indices of data points (x_i, y_i) above and below a line defined by (a, b) in the (x, y)-plane, or as the corresponding set of indices of data lines defined by $(-x_i, y_i)$ above and below a point (a, b) in the (a, b)-plane.

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

$$\begin{aligned} D_o(a,b) &= \sum_{i=1}^{n} w_i \frac{|y_i - (ax_i + b)|}{\sqrt{a^2+1}} \\ &= \Bigg(\sum_{i\in I_A(a,b)} w_i\,(y_i - (ax_i + b)) \\ &\quad - \sum_{i\in I_B(a,b)} w_i\,(y_i - (ax_i + b)) \Bigg) \Big/ \sqrt{a^2+1} \\ &= \frac{a(X_B - X_A) + b(W_B - W_A) + Y_A - Y_B}{\sqrt{a^2+1}} \end{aligned}$$

where

$$W_A = \sum_{i\in I_A} w_i, \quad X_A = \sum_{i\in I_A} w_i x_i, \quad Y_A = \sum_{i\in I_A} w_i y_i,$$
$$W_B = \sum_{i\in I_B} w_i, \quad X_B = \sum_{i\in I_B} w_i x_i, \quad Y_B = \sum_{i\in I_B} w_i y_i.$$

Hence, given the values of W_A, X_A, Y_A, W_B, X_B and Y_B, the value of the objective function $D_o(a, b)$ can be computed in constant time.

Suppose the values of the above variables are computed for some fixed parameter values a' and b'. If the next computation is for new values, a'' and b'', such that the above-below relationships of the data points and the lines defined by $y = a'x + b'$ and $y = a''x + b''$ do not change, then the values of those variables also do not change. Hence, the objective function for the parameters a'' and b'' can be computed in constant time. The above representation scheme is used in all the algorithms for maintaining the contribution of the data points to the objective function.

3. Orthogonal L_1 Problem

Both the unweighted and weighted orthogonal L_1 problems are solved by considering the problem in the dual plane. However, since $D_o(a, b)$ lacks the convexity of the vertical L_1 norm, there seems to be no obvious method for approaching the problem in a similar fashion used to solve the vertical L_1 problem [17].

First, properties of the orthogonal L_1 problem are given, and naive algorithms based on those properties are proposed. Second, the unweighted orthogonal L_1 problem is considered and details of the complexity of computing the solution are given. Third, the weighted orthogonal L_1 problem is considered and a brute force algorithm proposed; also, areas of improvement and related problems are described. Finally, an $\Omega(n \log n)$ lower bound under the algebraic computation tree model of Ben-Or [1] is shown.

Morris and Norback [13] present two characteristic properties of an optimal orthogonal L_1 problem solution.

Lemma 3.1. *There is an approximate line to the point set S which minimizes the orthogonal L_1 norm and which passes through two points of S.*

Lemma 3.1 suggests an $O(n^3)$ brute force approach to solving the problem. For each of the possible $\binom{n}{2}$ pairs of points, evaluate the function for a line which passes through the pair.

Lemma 3.2. *The sum of the weights on the optimal approximation line is greater than the difference of the sums of weights on either side of the line.*

Both **Lemma 3.1** and **Lemma 3.2** also hold in d-dimensions and can be solved by an $O(n^d)$-time and $O(n)$-space algorithm. An idea is to reduce the whole problem into $\binom{n}{d-2}$ subproblems, each of which can be solved in $O(n^2)$ time by utilizing the topological sweep of [5].

Morris and Norback suggest using both properties to find *candidate solutions*, pairs of data points in the (x, y)-plane (or pairs of lines in the (a, b)-plane) which satisfy both properties. 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; if so, then compute the L_1 norm with respect to the *candidate line* defined by the pair of points. Clearly, that algorithm is not a great improvement over computing the L_1 norm for every possible pair. However, the approach does raise the question of how many candidate pairs there are. That question can be related to the number of k-sets as described below.

Given a set S of n points, a k-set of S is a subset S' such that S' contains k points and there exists a line, l_k, which separates S' from $S - S'$. The number of k-sets of a set of n points has been shown to have complexities of $\Omega(n \log k)$ and $O(nk^{1/2})$ [7].

In the unweighted problem, **Lemma 3.2** is just a cardinality problem: the difference in the number of points above and below the line must not exceed the number of points on the line. The number of candidate lines is related to the number of k-sets, or "median"-sets since $k = \lceil n/2 \rceil$, of a set of n points in the (x, y)-plane. Similarly, the number of candidate lines in the weighted problem is related to the number of "weighted" k-sets, or *weighted median-sets*, since the line must satisfy the second property which balances the weights equally on each side of the line.

The computation of k-sets is studied in [6]. In that paper, they introduce the notion of k-belts in the dual plane. Recall that a line l_k is represented in the dual plane as a point p_k; the k-belt is simply the set of points p_k, the dual transformations of all the lines l_k which determine a k-set in the (x, y)-plane.

For each point p in the dual plane, let $b(p)$, $o(p)$, and $a(p)$ denote the number of lines which lie below p, on p, and above p, respectively. Clearly $b(p) + o(p) + a(p) = n$, for all p. The k-belt, for $0 \leq k \leq \lceil n/2 \rceil$, of the arrangement of lines in the dual plane is defined as the set of points p in the dual plane such that $b(p) + o(p) \geq k$ and $a(p) + o(p) \geq k$. The 0-belt is the whole plane, and for $k \geq 1$, the k-belt is bounded above and below by an unbounded monotone polygonal chain. For n odd and $k = \lceil n/2 \rceil$, the two boundaries of the $\lceil n/2 \rceil$-belt, referred to as the *median*-belt, coincide. The bold line in **Figure 3.1(b)** represents the median-belt for the seven lines.

Hence, candidate lines in the (x, y)-plane, lines which define a k-set, are the vertices of the k-belt as seen in **Figure 3.1.** Both algorithms consider k-belts in the dual plane rather than k-sets in the (x, y)-plane as described below.

3.1. The Unweighted Problem

In the unweighted orthogonal L_1 problem, the dual transformations of the candidate lines correspond to the vertices of the $\lceil n/2 \rceil$-belt, or median-belt (see **Figure 3.1**). The algorithm presented below evaluates the function at each of those vertices in order to determine an optimal value (the existence is guaranteed by **Lemma 3.2**). The function evaluations can be performed in the same amount of time as it takes to construct the median-belt.

The construction of the k-belt of an arrangement, H, of n lines can be performed in $O(b_k(n) \log^2 n)$ time and $O(n)$ space, where $b_k(n)$ is the maximum number of vertices in the k-belt for all arrangements, H, of n lines [6]. In the unweighted orthogonal L_1 problem, $k = \lceil n/2 \rceil$ and $b_{\lceil n/2 \rceil}(n) = O(n^{1.5})$.

The algorithm sweeps the arrangement by a vertical line L from left to right. At $a = -\infty$, the data line, l_m,

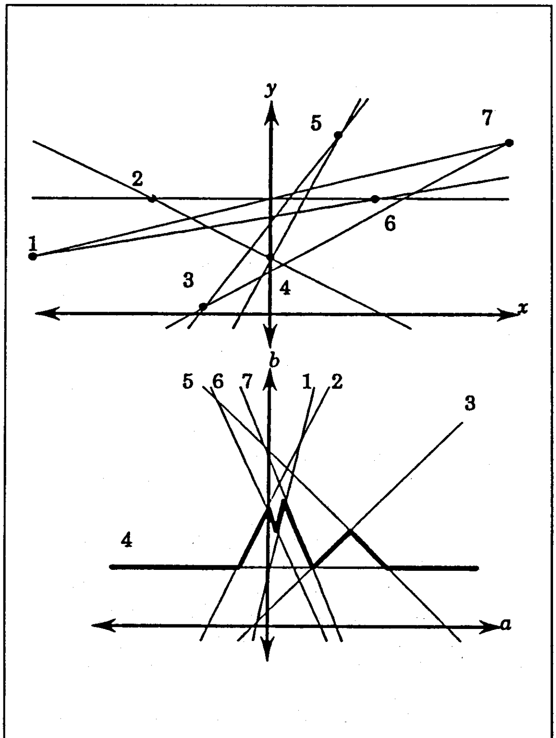

Figure 3.1. Candidates in the (x, y)- and (a, b)-planes.

with the median slope also has the median intersection with L. Let H_A be a set of data lines above l_m and similarly H_B a set of data lines below l_m with respect to the median intersection point. Hence, the sets I_A and I_B, as defined in **Section 2**, refer to the indices of the lines in H_A and H_B, respectively, and the variables used to maintain $D_o(a, b)$ (see **Section 2**) can be initialized in linear time.

The algorithm then sweeps the arrangement to find the next vertex on the belt which is the intersection of l_m and another data line l_A, where without loss of generality the line l_A is assumed to be in the set H_A. The value of $D_o(a, b)$ may be computed at the new vertex in constant time as follows. Subtract $w_A x_A$ from X_A and $w_A y_A$ from Y_A. Add $w_m x_m$ to X_A and $w_m y_m$ to Y_A. Those operations take constant time and $D_o(a, b)$ may be computed in constant time with the updated values. The algorithm keeps a pointer to the swept vertex which minimizes $D_o(a, b)$; clearly, such a pointer can be updated in constant time. Thus, each step of the plane sweep can be done without increasing the order of the complexity. Since $O(n^{1.5} \log^2 n)$ time is spent in constructing the median-belt, the following result is obtained.

Theorem 3.1. *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. Also, 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, the authors conjecture that a more efficient algorithm may be obtained by considering the particular properties of the median-belt.

3.2. The Weighted Problem

In the dual plane, the candidate solutions to the weighted orthogonal L_1 problem lie on the boundary of the *weighted median-belt* of the arrangement, the set of points which are the dual transformations of the weighted median lines in the (x, y)-plane. Hence, the complexity of finding a solution can be related to the number of vertices (due to **Lemma 3.2**), n_w, in a weighted median-belt.

The complexity of n_w 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), or any point on the boundary of the weighted median-belt whose incident boundary edges are distinct (*nondegenerate* vertices) (see **Figure 3.2(a),(b)**). Note that the nondegenerate vertices are a subset of the degenerate vertices.

Since the weighted median-belt is an x-monotone chain, the number of vertices can be bounded by results for monotone chains. The number of degenerate vertices is $\Omega(n^2)$ [16] (see **Figure 3.2(a)**). The number of nondegenerate vertices is $\Omega(n^{1.5})$ [14] (see **Figure 3.2(b)**, $n = \sqrt{k}$, shaded area has k^2 vertices); although an (unknown at the time of printing) improvement to this lower bound has been reportedly made by R. Cole and J. Matoušek [3].

Note, however, that not all x-monotone chains are weighted median-belts. In the case of the example for the degenerate bound, legal weight assignments (bold numbers) can be given to the lines, however, the particular example given for the nondegenerate bound cannot be assigned weights such that the chain becomes a weighted belt.

The geometry 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.

An algorithm based only on the properties given by **Lemma 3.1** and **Lemma 3.2** must check the objective function at all degenerate vertices. Hence, the complexity of applying the k-belt construction algorithm, mentioned above, to the weighted problem is in $O(n^2 \log^2 n)$. However, the weighted orthogonal L_1 problem can be solved in $O(n^2)$ time by the following "efficient" brute force algorithm.

An optimal solution is found by performing an

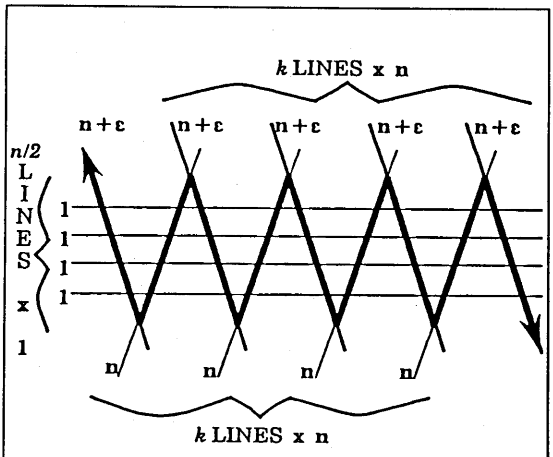

Figure 3.2.(a) $\Omega(n^2)$ Degenerate vertices

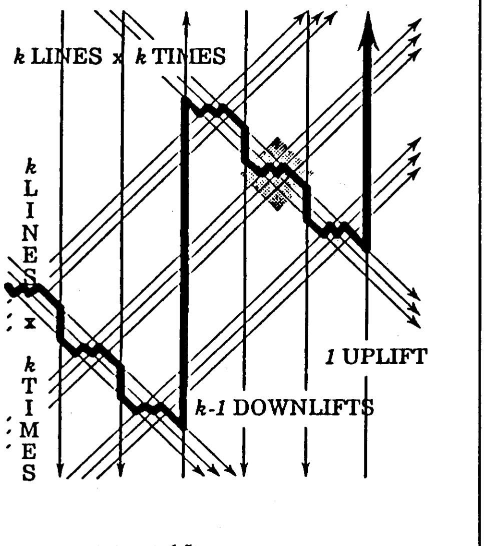

Figure 3.2.(b) $\Omega(n^{1.5})$ Non-degenerate vertices

$O(n^2)$ time, $O(n)$ space plane sweep as described in [5]. Note that this plane sweep differs from the plane sweep algorithm used for the unweighted case; the plane sweep used above only computes points of interest in order from left to right, whereas the plane sweep here reports all $O(n^2)$ vertices of the arrangement in an order defined by a topological sweep from left to right.

First, note that the leftmost edge of the weighted median-belt can be determined in linear time. Second, the plane sweep algorithm removes degenerate vertices in the arrangement, hence, each vertex is incident to only two lines and determining whether a switch to a new line should take place at a vertex of the belt can be decided in constant time by maintaining $D_o(a,b)$ as described in Section 2. Third, since the next event vertex in the sweep is identified by its incident edges (lines), the vertex can be tested to see if it is on the belt by simply comparing the current edge of the belt to the incident lines in constant time.

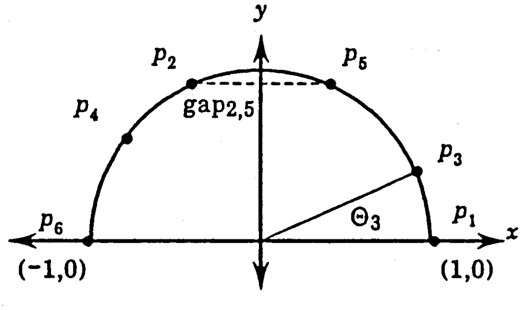

Figure 3.3. Uniform Gap Problem on Half Unit Circle

Since the plane sweep is performed in $O(n^2)$ time and the extra cost to maintain the weighted median-belt and the function $D_o(a,b)$ is $O(1)$ time for each vertex, an optimal solution may be found in $O(n^2)$ time. The algorithm is called an efficient brute force algorithm since 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 algorithm implies that the algorithm for constructing k-belts used in the unweighted problem is not, at least in one case, the most efficient algorithm for computing weighted k-belts.

3.3. $\Omega(n \log n)$ Lower Bound

This section provides an $\Omega(n \log n)$ lower bound for the orthogonal L_1 problem under the algebraic computation tree model [1]. (Due to space limitations not all the proofs are included in this paper). First, the lower bound of the complexity of a certain uniform gap problem of n points on a circle is shown to be in $\Omega(n \log n)$ under the algebraic computation tree model. Second, that uniform gap problem is reduced to the unweighted orthogonal L_1 problem in linear time, thus proving an $\Omega(n \log n)$ lower bound of this approximation problem.

The uniform gap problem on the unit circle is defined as:

Problem 3.1. Uniform Gap Problem on the (Half) Unit Circle. *Given $m+1$ points $p_1, p_2, \ldots, p_m$, p_{m+1} on the unit circle $x^2+y^2=1$ such that $p_1=(1,0)$, $p_{m+1}=(-1,0)$ and the y-coordinates of points p_i ($i=2,\ldots,m$) are positive* (see **Figure 3.3**), *the uniform gap problem on the unit circle answers whether the distances of every consecutive pair of points are equal when these $m+1$ points are arranged in increasing order of their polar angles.*

Denote the (x,y)-coordinates of point p_i by (x_i, y_i), where $x_1 = 1$, $x_{m+1} = -1$ and $y_i \geq 0$. For a permuta-

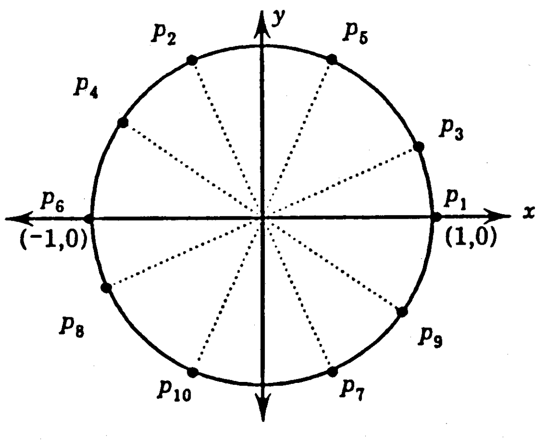

Figure 3.4. Transformed Input

tion σ on $\{2,\dots,m\}$, define $W_\sigma \subset \mathrm{R}^{m-1}$ by

$$\begin{aligned} W_\sigma = \Big\{ & (x_2,\dots,x_m) \in \mathrm{R}^{m-1} \mid \\ & 1 = x_1 > x_{\sigma(2)} > x_{\sigma(3)} > \cdots \\ & \quad > x_{\sigma(m)} > x_{m+1} = -1, \\ & (x_{\sigma(i+1)} - x_{\sigma(i)})^2 \\ & + \left(\sqrt{1 - x^2_{\sigma(i+1)}} - \sqrt{1 - x^2_{\sigma(i)}}\right)^2 = \epsilon^2 \Big\}, \end{aligned}$$

where $\sigma(1) = 1$ and $\sigma(m+1) = m+1$. Define $W \subset \mathrm{R}^{m-1}$ by $W = \bigcup_\sigma W_\sigma$ where the union is taken over all permutations σ on $\{2,\dots,m\}$. Then the answer for the uniform gap problem for points $p_1, p_2, \dots, p_{m+1}$ is yes if and only if $(x_2,\dots,x_m) \in W$, which implies that, by Ben-Or's theorem [1], 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. $\#W$ can be shown to be equal to $(m-1)!$, hence, the following result holds under the algebraic computational model of Ben-Or.

Lemma 3.3. *The complexity of the uniform gap problem on the 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 orthogonal linear L_1 approximation of points. Given the points $p_1, p_2, \dots, p_{m+1}$ of the uniform gap problem, assume that, without loss of generality, $0 = \theta_1 < \theta_2 < \cdots < \theta_m < \theta_{m+1} = \pi$, where θ_i denotes the polar angle of p_i. For each p_i $(i = 2,\dots,m)$, construct the point p_{i+m} on the unit circle whose polar angle θ_{i+m} is equal to $\theta_i + \pi$ (see **Figure 3.4**). The set, S, of $n = 2m$ points $p_1, p_2, \dots, p_n$ is then used as the input for the orthogonal L_1 problem. The transformation is then completed by showing the following result.

Lemma 3.4. *The minimum objective function value of the orthogonal L_1 linear approximation for the set S of points is at most $2\cot\frac{\pi}{n}$, and is $2\cot\frac{\pi}{n}$ if and only if the answer for the uniform gap problem of points $p_1, p_2, \dots, p_m, p_{m+1}$ is yes.*

The proof is developed as follows. For the orthogonal linear L_1 approximation problem, it is known that there is an optimal approximation line such that the line passes two points among the given points and $|N_A - N_B| < N_O$ where N_A, N_B and N_O are the numbers of points above, below, and on the line [13]. By the definition of the transformed problem, there is an optimal approximation line among the m lines l_i connecting points p_i and p_{i+m} $(i = 1,\dots,m)$. The function value of l_i, the summation of the orthogonal distances from points p_j $(j = 1,\dots,n)$ 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

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

We are considering to maximize (2) for $0 = \theta_1 < \theta_2 < \cdots < \theta_m < \pi$ and $\theta_{i+m} = \theta_i$ $(i = 1,\dots,m)$. However, this is an optimization problem of maximin type, and rather difficult to handle directly. Instead, we will consider to maximize

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

Here, 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 (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 $k = 1,\dots,m$, we have

$$0 < \theta_{i+k} - \theta_i < \pi, \sum_{i=1}^{m} (\theta_{i+k} - \theta_i) = k\pi.$$

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

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

and the equality holds if and only if all $\theta_{i+k} - \theta_i$ $(i = 1,\dots,m)$ are equal. All $\theta_{i+k} - \theta_i$ $(i = 1,\dots,m)$ are equal for any $k = 1,\dots,m$ if and only if all $\theta_{i+1} - \theta_i$ $(i = 1,\dots,m)$ 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 linear L_1 approximation is expressed by

$$\sum_{j=1}^{n} |\sin \frac{j\pi}{m}| = 2 \sum_{j=1}^{m} \sin \frac{j\pi}{m} = 2 \cot \frac{\pi}{2m} = 2 \cot \frac{\pi}{n}.$$

Thus, we have shown **Lemma 3.3.**

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

Theorem 3.2. *Under the algebraic computation tree model, the complexity of the orthogonal L_1 linear approximation of n points is $\Omega(n \log n)$.*

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 Erdös, Lovász, Simmons, and Strauss [7] conjecture that the upper bound is actually closer to the lower bound of $\Omega(n \log k)$ ($\Omega(n \log n)$ for $k = \lceil n/2 \rceil$).

4. Conclusion

Several results were given concerning the computation and analysis of the orthogonal L_1 linear approximation problems. The results are significant from theoretical, practical, and historical viewpoints.

The results are of theoretical interest in discrete and computational geometry, since a nice connection is demonstrated between the orthogonal L_1 problem and the (weighted) median belt, thus having raised some questions about the complexity of computing median-belts and weighted median-belts.

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 algorithms. [2] 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.

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. Recently, Kato and Imai [9] have succeeded in devising a linear-time algorithm for the vertical L_1 problem in the three-dimensional space. It is conjectured that the result can be extended to higher dimensions. It is a challenging problem to improve the complexity of the algorithms for the orthogonal L_1 linear problem described in this paper.

Acknowledgment

This study was carried out in part under the ISM Cooperative Research Program (88-ISM·CRP-13) of the Institute of Statistical Mathematics.

References

[1] M. Ben-Or: Lower bounds for algebraic computation trees. *Proc. 15th ACM Symp. on Theory of Computing*, 1983, 80–86.

[2] V. Chvátal: *Linear programming.* Freeman Press, 1983.

[3] R. Cole and J. Matousek, result communicated by E. Welzl.

[4] R. Cole, M. Sharir and Chee K. Yap, On k-Hulls and related problems, *SIAM J. Comput.*, 16 1 (1987), 61–77.

[5] H. Edelsbrunner and L.J. Guibas: Topologically sweeping an arrangement, Digital Systems Research Center, April, 1986.

[6] H. Edelsbrunner and E. Welzl: Constructing belts in two-dimensional arrangements with applications. *SIAM J. Comput.*, 15 (1986), 271–284,

[7] P. Erdös, L. Lovász, A. Simmons and E. G. Straus: Dissection graphs of planar point sets. In "*A Survey of Combinatorial Theory*" (J. N. Srivastava et al., eds.), North-Holland, 1973, 139–149.

[8] H. Imai, K. Kato, and P. Yamamoto: A linear time algorithm for linear L_1 approximation of points, *Tech. Rep. CSCE-87-C30*, Dept. Comp. Sci. and Comm. Eng., Kyushu University, August 1987.

[9] K. Kato and H. Imai: A linear-time algorithm for planar L_1 linear approximation of points, in preparation.

[10] D.T. Lee and Y.F. Wu, Geometric complexity of some location problems, *Algorithmica*, 1 (1986), 193–211.

[11] N. Megiddo: Linear time algorithms for linear programming in $\mathbf{R}^3$ and related problems. *SIAM J. COMP.*, 12 (1983), 759–776.

[12] N. Megiddo: Linear programming in linear time when the dimension is fixed. *J. ACM*, 31 (1984), 114–127.

[13] J. G. Morris and J. P. Norback: A simple approach to linear facility location. *Transportation Science*, 14 (1980), 1–8.

[14] M. Sharir, result communicated by E. Welzl.

[15] H. Späth and G.A. Watson: On orthogonal linear L_1 approximation, *Numer. Math.*, 51 (1987), 531–543.

[16] E. Welzl, personal communication.

[17] P. Yamamoto, K. Kato, K. Imai and H. Imai: Algorithms for Orthogonal and Vertical L_1 Linear Approximation of Points, *Proc. 4th ACM Symp. on Computational Geometry*, Urbana, Illinois, 1988, 352–361.