Algorithms for Bivariate Medians and a Fermat-Torricelli Problem for Lines

Greg Aloupis* Stefan Langerman† Michael Soss* Godfried Toussaint*

Abstract

Given a set S of n points in $\mathbb{R}^2$, the *Oja depth* of a point θ is the sum of the areas of all triangles formed by θ and two elements of S. A point in $\mathbb{R}^2$ with minimum depth is an *Oja median*. We show how an Oja median may be computed in $O(n \log^3 n)$ time. In addition, we present an algorithm for computing the *Fermat-Torricelli points of n lines* in $O(n)$ time. These points minimize the sum of weighted distances to the lines. Finally, we propose an algorithm which computes the *simplicial median* of S in $O(n^4)$ time. This median is a point in $\mathbb{R}^2$ which is contained in the most triangles formed by elements of S.

1 Introduction

The use of multidimensional medians as robust estimators of location has been studied extensively by computer scientists and statisticians. In general, a median is the location in $\mathbb{R}^d$ which maximizes a certain depth definition with respect to a data set. Applications include data description, multivariate confidence regions, p-values, quality indices, and control charts [RR96]. An introduction to the topic including definitions and properties of the most common medians is given in [Sma90]. For a recent account on the computational complexity of computing some of these medians, see [Alo01]. Below, we include descriptions for the Oja median [Oja83] and for the simplicial median of Liu [Liu90].

Let $S = \{s_1, \ldots, s_n\}$ be a set of data points in $\mathbb{R}^d$. The *Oja depth* of a point θ in $\mathbb{R}^d$ with respect to S is the cumulative volume of all simplices formed by θ and a subset of d elements from S. To find the Oja depth of θ in $\mathbb{R}^2$, we sum the areas of all triangles formed by θ and two points of S. The Oja median is any point in $\mathbb{R}^d$ with minimum Oja depth.

The *simplicial depth* of a point θ in $\mathbb{R}^d$ with respect to S is the number of closed simplices formed by $d+1$ elements of S that contain θ. To find the simplicial depth of θ in $\mathbb{R}^2$, we must find how many triangles formed by triples of points in S contain θ. The simplicial median is any point in $\mathbb{R}^d$ with maximum simplicial depth.

Rousseeuw and Ruts [RR96] presented an algorithm for computing bivariate simplicial depth in $O(n \log n)$ time. The same technique appeared independently by Khuller and Mitchell [KM89] and by Gil, Steiger and Wigderson [GSW92]. A matching lower bound was presented in [ACG+01]. Rousseeuw and Ruts mentioned that their algorithm leads to a straightforward computation of the simplicial median in $O(n^5 \log n)$ time since only the $O(n^4)$ intersection points of segments between pairs of data points need to be checked. They also briefly described how their methods can be used to speed up the algorithm of Niinimaa, Oja and Nyblom [NON92] for computing the Oja bivariate median from $O(n^6)$ to $O(n^5 \log n)$ time. This time complexity was improved by Aloupis, Soss and Toussaint [AST01, Alo01]. They proved that the Oja depth function is convex and that the minimum is on an intersection of lines formed by data points. The minimum depth on a line could be computed in $O(n^2)$ time, and in this time the side containing the median could be determined. This allowed the median to be found with $O(n)$ binary searches, in time $O(n^3 \log n)$.

Roy Barbara [Bar00] referred to the set of points with minimum sum of weighted distances to n given lines as the Fermat-Torricelli points of the n lines. Fermat and Torricelli are known for first solving the problem of finding a point which minimizes the sum of distances to three given points [GS98, dF34, LV19], and the generalized problem with n points also bears their name. For other generalizations of the Fermat-Torricelli problem, see [Dal01, KM97].

Barbara proved that a Fermat-Torricelli point of n lines may always be found on one of the intersections of the lines. He then suggested computing the weighted sum for each intersection point, which is straightforward to do in $O(n^3)$ time using $O(n)$ space.

In section 2 we propose an optimal $O(n)$ time algorithm to find the Fermat-Torricelli points of n lines. It is not difficult to see that this also leads to an $O(n^2)$ time algorithm for computing the Oja median of n points. We further reduce this complexity to $O(n \log^3 n)$ in section 3, with certain techniques developed recently by Langerman and Steiger [Lan01].

*School of Computer Science, McGill University. {athens,soss,godfried}@cs.mcgill.ca

†School of Computer Science, Rutgers University. lfalse@cs.rutgers.edu

Finally, in section 4 we present an algorithm for computing the simplicial median in $O(n^4)$ time and $O(n^2)$ space.

2 The Fermat-Torricelli Points of n Lines

As mentioned above, Barbara [Bar00] proved that it suffices to consider the intersections of n lines in order to find their Fermat-Torricelli point. Here, we provide a shorter proof:

Lemma 1 *One of the intersection points of a set L of n lines must be a Fermat-Torricelli point of L.*

Proof. Each cell formed by the arrangement of the n lines can be considered to be a linear program, where the feasible region is determined by the edges of the cell, and the function to be minimized is the weighted sum of distances to the n lines. Because this is a linear function, if a Fermat-Torricelli point is located within a given cell, one of the vertices of the cell must also be a Fermat-Torricelli point.

□

Fact 1 *The sum of two or more linear functions is linear.*

Fact 2 *The sum of two or more piecewise-linear functions is piecewise-linear.*

Fact 3 *The sum of two or more convex functions is convex.*

Let $D_L(p)$ denote the function whose value at a point p in the plane is the sum of weighted distances from p to a set L of n lines. In other words we have

$$D_L(p) = \sum_{\ell \in L} D_\ell(p)$$

where $D_\ell(p)$ is the distance from p to a line ℓ.

Lemma 2 *The function D_L is piecewise-linear and convex, and the linear pieces correspond to the cells of the arrangement of L.*

Proof. $D_L(p)$ is a sum of piecewise-linear convex functions, so the proof is by facts 1–3.

□

Lemma 3 *The minimum value of D_L along any line ℓ may be found in $O(n)$ time. In the same time bound, we can determine which side of ℓ contains a Fermat-Torricelli point.*

Proof. Compute the intersection points of ℓ with all lines of L in $O(n)$ time. Adjust the weight of each line according to the angle that it makes with ℓ. Thus for any point on ℓ, D_L becomes a weighted sum of distances to all intersection points on ℓ. In other words, the point p with minimum value of D_L along ℓ is equal to the weighted univariate median of n points. Since D_L is convex, by computing a gradient vector at p in $O(n)$ time, we can determine which side of ℓ contains the minimum of D_L.

□

We say that two lines ℓ_1,ℓ_2 form an *equitable partition* of a set L of n lines in $\mathbb{R}^2$ if each quadrant determined by ℓ_1 and ℓ_2 is guaranteed not to intersect roughly a quarter of the lines in L. Langerman and Steiger [Lan01] have shown that an equitable partition always exists and may be computed in $O(n)$ time.

Let K be the set of lines for which we know on which side the minimum lies. Let U be the complement of K. We can compute a partial gradient belonging to the lines in K. This partial gradient may be updated in constant time for each new line introduced to K. Thus we can use only $O(|U|)$ time to determine which side of a line contains the global minimum.

Algorithm FT

1. Compute an equitable partition $\{\ell_A,\ell_B\}$ of U.

2. Determine which quadrant formed by the lines ℓ_A and ℓ_B contains a Fermat-Torricelli point.

3. Determine which lines in U do not intersect the quadrant.

4. Add to K the lines found in step 3, and update the gradient vector representing the distance to all lines in K.

5. Go to step 1 if $|U|$ is still large. Otherwise evaluate the function D_L on each intersection point of the lines in U.

A constant fraction of the lines is pruned by each iteration in $O(|U|)$ time. Thus we have the following theorem:

Theorem 4 *Algorithm FT computes a Fermat-Torricelli point of n lines in $O(n)$ time.*

Since the Oja depth function can be described as a weighted distance function to $O(n^2)$ lines, we can see that in $O(n^2)$ time we can obtain the Oja median.

3 The Oja Median

Suppose $S = \{s_1, s_2, \ldots, s_n\}$ is a set of data points in $\mathbb{R}^2$, L is the set of lines formed by all pairs of points in S, and A is the arrangement of L.

Let $f_S(p) : \mathbb{R}^2 \to \mathbb{R}$ be a function for which:

- The minimum of f_S is on a vertex of A.
- Given a point a, in time $T(n)$ we can find a half-plane H determined by a line through a where $f_S(q) \geq f_S(a)$ for all $q \in H$.

Langerman and Steiger [Lan01] have proved the following theorem:

Theorem 5 *The minimum $f^\star$ of f_S and a vertex x of A with $f_S(x) = f^\star$ may be computed in $O(T(n)\log^2 n + n\log^3 n)$ time.*

In the full version, we prove that the bivariate Oja depth function satisfies the properties of f_S, with $T(n) = O(n \log n)$. Therefore we obtain the following theorem:

Theorem 6 *The bivariate Oja median may be computed in $O(n\log^3 n)$ time.*

4 The Simplicial Median

Let S be a data set of n points in $\mathbb{R}^2$, and I be the set of line *segments* formed between every pair of data in S. To simplify this description, assume that S is in general position, in the sense that no three points are collinear. This assumption is dealt with in the full version of this paper.

Algorithm *Simp-Med*

1. Calculate the simplicial depth of every data point in S.
2. Compute the number of points which are strictly to one side of each segment in I (each side separately).
3. For every segment ℓ in I,
   (a) compute and sort the intersection points of all other segments with ℓ, if there are any.
   (b) let MAX be the greatest depth among the endpoints of ℓ.
   (c) if there exist other intersection points on ℓ, calculate the simplicial depth d of the intersection point adjacent to MAX. Update MAX (if $MAX < d$, MAX$\leftarrow$ d).
   (d) continue through the sorted list: every time an intersecting line is left behind, subtract from d the number of points strictly behind the line. Every time a line is encountered, add to d the number of points strictly ahead of that line. Update MAX after encountering each intersection.
4. Exit with the greatest MAX value found over all segments, and the associated intersection point.

In case we have multiple segments intersecting the current line at the same point in step 3d, we may form a temporary list of these segments. When the last one is encountered, we process the list: First we add the points to the right of each line, then we subtract all the points to the left of each line.

Note that step 3a may be omitted if we first compute the arrangement of I. Thus algorithm *Simp-Med* computes the bivariate simplicial median of n points in $O(n^4 \log n)$ time and $O(n^2)$ space or alternatively in $O(n^4)$ time and space.

Instead of performing step 3 in algorithm *Simp-Med*, we can perform a topological sweep, which takes $O(n^2)$ time and $O(n)$ space for n lines [EG89]. Instead of processing each segment sequentially, we essentially process all $O(n^2)$ segments in parallel. Every time the sweep curve encounters an intersection point on some segment, we process the point in the same way as step 3d. Thus we have the following theorem:

Theorem 7 *The simplicial median of n points may be computed in $O(n^4)$ time and $O(n^2)$ space.*

5 Conclusion

We have reduced the time complexity of calculating the Oja and simplicial medians. The Oja median may be computed in $O(n\log^3 n)$ time using $O(n)$ space. The simplicial median may be computed in $O(n^4)$ time using $O(n^2)$ space. This complements the algorithm of Rousseeuw and Ruts [RR96] which computes each median in $O(n^5 \log n)$ time and $O(n)$ space.

We have also presented an optimal algorithm for computing the Fermat-Torricelli points of n lines in $O(n)$ time, thus improving the algorithm suggested by Barbara which takes $\Omega(n^3)$ time.

Acknowledgements

This research was started at the Workshop on Applications of Computational Geometry in Statistics, organized by Godfried Toussaint at the Bellairs Research Institute of McGill University in Barbados, February 1999. We thank all the participants for helpful discussions.

References

[ACG$^+$01] G. Aloupis, C. Cortes, F. Gomez, M. Soss, and G. Toussaint. Lower bounds for computing statistical depth. Technical Report SOCS-01.1, School of Computer Science, McGill University, February 2001.

[Alo01] G. Aloupis. On computing geometric estimators of location. M.Sc. thesis, McGill University, 2001.

[AST01] G. Aloupis, M. Soss, and G. Toussaint. On the computation of the bivariate median and a Fermat-Torricelli problem. Technical Report SOCS-01.2, School of Computer Science, McGill University, February 2001.

[Bar00] R. Barbara. The Fermat-Torricelli points of n lines. *Mathematical Gazette*, 84:24–29, 2000.

[Dal01] Leoni Dalla. A note on the Fermat-Torricelli point of a d-simplex. *J. geom.*, 70:38–43, 2001.

[dF34] P. de Fermat. Abhandlungen über maxima und minima. *Ostwalds Klassiker der exacten Wissenschaften, M. Miller (ed)*, 238, 1934.

[EG89] H. Edelsbrunner and L. Guibas. Topologically sweeping an arrangement. *Journal of Computer and System Sciences*, 38:165–194, 1989.

[GS98] C. Groß and T. Strempel. On generalizations of conics and on a generalization of the Fermat-Torricelli problem. *American Mathematical Monthly*, 105(8):732–743, 1998.

[GSW92] J. Gill, W. Steiger, and A. Wigderson. Geometric medians. *Discrete Mathematics*, 108:37–51, 1992.

[KM89] S. Khuller and J. Mitchell. On a triangle counting problem. *Information Processing Letters*, 33:319–321, 1989.

[KM97] Y. Kupitz and H. Martini. Geometric aspects of the generalized Fermat-Torricelli problem. *Intuitive Geometry, Bolyai Society Math Studies*, 6:55–127, 1997.

[Lan01] S. Langerman. *Algorithms and Data Structures in Computational Geometry.* Ph.D. thesis, Department of Computer Science, Rutgers University, New Brunswick, 2001.

[Liu90] R. Liu. On a notion of data depth based upon random simplices. *The Annals of Statistics*, 18:405–414, 1990.

[LV19] G. Loria and G. Vassura, editors. *Opere di Evangelista Torricelli*, volume 1. Faenza, 1919.

[NON92] A. Niinimaa, H. Oja, and J. Nyblom. Algorithm AS 277: The Oja bivariate median. *Applied Statistics*, 41:611–617, 1992.

[Oja83] H. Oja. Descriptive statistics for multivariate distributions. *Statistics and Probability Letters*, 1:327–332, 1983.

[RR96] P. Rousseeuw and I. Ruts. Bivariate location depth. *Applied Statistics*, 45:516–526, 1996.

[Sma90] C. Small. A survey of multidimensional medians. *International Statistical Review*, 58:263–277, 1990.