On Algorithms for Simplicial Depth

Andrew Y. Cheng
Department of Industrial Engineering
Ming Ouyang
Department of Computer Science
Rutgers University
New Brunswick, New Jersey 08903

ABSTRACT

Simplicial depth is a way to measure how deep a point is among a set of points. Efficient algorithms to compute it are important to the usefulness of its applications, such as in multivariate analysis in statistics. A straightforward method takes $O(n^{d+1})$ time when the points are in d-dimensional space. We discuss an algorithm that takes $O(n^2)$ time when the points are in three-dimensional space, and we generalize it to four-dimensional space with a time complexity of $O(n^4)$. For spaces higher than four-dimensional, there are no known algorithms faster than the straightforward method.

1 Simplicial depth

A *simplex* in d-dimensional Euclidean space E^d is the set of points that are convex combinations of $d+1$ affinely independent points; that is, if the points are $p_1, p_2, \ldots, p_{d+1}$, they bring about the simplex

$$\{p : p = a_1p_1 + a_2p_2 + \ldots + a_{d+1}p_{d+1},\ a_i \geq 0 \text{ and } \sum a_i = 1\}.$$

A simplex is a line segment in E^1, a triangle in E^2, and a tetrahedron in E^3.

It is customary to say that a set of points in E^d are in *general position* if any $d+1$ points in the set are affinely independent. Let P be a set of n points in general position in E^d; take every $d+1$ distinct points from P and they uniquely identify a simplex in E^d; let S_P be the set of all such simplices; then, $|S_P| = \binom{n}{d+1}$. Let $\sigma(p, P)$ denote the *simplicial depth* of a point p with respect to a set P, defined by

$$\sigma(p, P) = |\{s \in S_P : p \in s\}|.$$

For example, if p is outside the convex hull of P, then $\sigma(p, P) = 0$, and if p is a vertex of the convex hull of P, then $\sigma(p, P) = \binom{n-1}{d}$. When p is in P, the following identity holds:

$$\sigma(p, P) = \sigma(p, P - \{p\}) + \left(\begin{array}{c} |P| - 1 \\ d \end{array} \right).$$

Boros and Füredi [1] have shown that for any set P of n points in E^2, there exists a point p such that $\sigma(p, P) = n^3/27 + O(n^2)$; furthermore, there is a set Q of n points such that for all points p in the plane, $\sigma(p, Q) < n^3/27 + n^2$. Liu [5] observes that simplicial depth is invariant under nonsingular affine transformations; we will use this property in the development of algorithms.

Simplicial depths have applications in statistics [2, 5, 6]; and efficient algorithms to compute them are of central concern. A straightforward method for d-dimensional simplicial depth is to generate all the simplices and then to count the number of containments; since there are $O(n^{d+1})$ simplices, it takes $O(n^{d+1})$ time in the real RAM model of computation [7]. However, in one-dimensional space, simplicial depth can be computed in $O(n)$ time by a partition algorithm; in two-dimensional space, simplicial depth can be computed in $O(n \lg n)$ time [3, 4, 8].

2 Computing simplicial depth in E^3

2.1 Counting tetrahedra not containing p

Rousseeuw and Ruts have briefly described an algorithm that computes simplicial depth in spaces higher than two-dimensional (page 523, [8]). We find it difficult to understand their argument; let us elaborate.

Their idea is as follows. For each p_i in P,

- let Π_i be the plane that passes through p and is orthogonal to $(p_i - p)$;
- project P to Π_i; let $\Pi_i(q)$ denote the image of q on Π_i; note that $\Pi_i(p_i) = \Pi_i(p)$.

The argument is that if a triangle $\triangle\Pi_i(p_j)\Pi_i(p_k)\Pi_i(p_l)$ does not contain p, then the tetrahedron $\triangle p_i p_j p_k p_l$ does not contain p. We can use the algorithm for two-dimensional simplicial depth in [8] to count the triangles in Π_i, $i = 1, 2, \ldots, n$, that do not contain p, and the sum of these counts corresponds to the number of tetrahedra that do not contain p.

We need some terminology to explain the difficulty we have encountered. Assume the tetrahedron $\triangle p_i p_j p_k p_l$ does not contain p; there are four planes in E^3 defined by any three vertices of the tetrahedron; these planes divide the exterior of the tetrahedron into 14 cells:

- four cells adjacent to the faces of the tetrahedron,
- six cells adjacent to the edges of the tetrahedron, and
- four cells adjacent to the vertices of the tetrahedron.

If p is in a cell adjacent to a face of the tetrahedron, for example, the triangle $\triangle p_i p_j p_k$, then p_l is "invisible" to p (blocked by the tetrahedron). When the tetrahedron is projected to the plane Π_l, the triangle $\triangle\Pi_l(p_i)\Pi_l(p_j)\Pi_l(p_k)$ in fact *does* contain p. But when the tetrahedron is projected to the other three planes, Π_i, Π_j, and Π_k, the corresponding triangles do *not* contain p. Therefore, the tetrahedron is counted three times by the above method.

Similarly, if p is in a cell adjacent to a vertex of the tetrahedron, the tetrahedron is counted three times.

However, if p is in a cell adjacent to an edge of the tetrahedron, the tetrahedron will be counted *four* times. We can not find a way to account for this discrepancy.

2.2 Counting tetrahedra containing p

Gil, Steiger, and Wigderson [3] have described an algorithm that computes simplicial depth in three-dimensional space; however, there is a slight flaw. We describe a modification.

Let P be a set of n points in general position in E^3, and let p be a point in general position with P. Lemmas 1 and 2 are from [3].

Lemma 1. *Let p_i' be any point on the ray from p through p_i; the tetrahedron $\triangle p_i p_j p_k p_l$ contains p if and only if the tetrahedron $\triangle p_i' p_j p_k p_l$ contains p.*

Let θ_i be the point where the ray from the origin with the direction $(p_i - p)$ intersects the unit sphere about the origin; let $\hat{\theta}_i$ be the point where the ray from the origin with the direction $(p - p_i)$ intersects the unit sphere about the origin; that is, $\hat{\theta}_i$ is *antipodal* to θ_i. By Lemma 1, the tetrahedron $\triangle p_i p_j p_k p_l$ contains p if and only if the tetrahedron $\triangle \theta_i \theta_j \theta_k \theta_l$ contains the origin. Assuming θ_i, θ_j, and θ_k are not antipodal to one another, the *spherical triangle* $\triangle_s \theta_i \theta_j \theta_k$ is the area on the surface of the unit sphere bounded by the short arcs of the great circles passing any two points in $\{\theta_i \theta_j \theta_k\}$.

Lemma 2. *The tetrahedron $\triangle \theta_i \theta_j \theta_k \theta_l$ contains the origin if and only if the spherical triangle $\triangle_s \theta_i \theta_j \theta_k$ contains $\hat{\theta}_l$.*

Let P' be $\{\theta_1, \theta_2, \ldots, \theta_n\}$. By Lemmas 1 and 2, we can count, for $i = 1, \ldots, n$, the number of spherical triangles with vetices in $P' - \{\theta_i\}$ that contain $\hat{\theta}_i$, and $\sigma(p, P)$ is equal to the sum of these counts. Later we will describe a relabeling of the points in P'. With the new labeling in effect, let Π_l be a plane that contains the origin and separates θ_l from θ_i, θ_j, and θ_k; and let Π_l' be a plane that is parallel to Π_l, but is at some distance from the origin. The *radial projection* of a point q ($q \neq O$) onto Π_l', denoted by $\Pi_l'(q)$, is the intersection of Π_l' and the line $\overline{Oq}$; note that the image of a radially projected spherical triangle is a triangle in the plane, and $\Pi_l'(q) = \Pi_l'(\hat{q})$. Trivially,

Lemma 3. *The spherical triangle $\triangle_s \theta_i \theta_j \theta_k$ contains $\hat{\theta}_l$ if and only if, in the plane Π_l', the triangle $\triangle \Pi_l'(\theta_i) \Pi_l'(\theta_j) \Pi_l'(\theta_k)$ contains $\Pi_l'(\hat{\theta}_l)$.*

The algorithm proceeds as follows. We need a point x in general position with $P' \cup \{O\}$; the line $\overline{Ox}$ is the *axis of rotation*. Let Λ be the plane that contains O and is orthogonal to the axis of rotation; project P' to Λ, and let $\alpha_0, \alpha_1, \ldots, \alpha_{n-1}$ be the polar angles of the projected images in ascending order, $0 \leq \alpha_i < 2\pi$; relabel the points in P' so that the indices start at 0 and θ_i corresponds to α_i. In what follows, we will assume "wrapping around" of indices (indices modulo n) and "wrapping around" of angles (adding 2π to negative angles, and subtracting 2π from angles that are equal to or greater than 2π).

On the plane Λ, let $\hat{\alpha}_i$ denote the point on the unit circle with the polar angle $\alpha_i + \pi$; that is, $\hat{\alpha}_i$ is *antipodal* to α_i. Suppose $\hat{\alpha}_i$ is angularly between α_j and α_{j+1}; let ϵ_i be the

minimum of $(\alpha_{i+1} - \alpha_i)$ and $(\alpha_{j+1} - \hat{\alpha}_i)$. Then, define the *dividing plane* Π_i to be the plane that contains both the axis of rotation and the line on Λ that goes through the origin and has the polar angle $\alpha_i + \epsilon_i/2$. Each dividing plane Π_i, $i = 0, 1, \ldots, n-1$, divides the unit sphere into two hemispheres, and the hemisphere containing $\hat{\theta}_i$ is designated the upper hemisphere; ϵ_i is chosen so that θ_i is put into the lower hemisphere as much as possible without moving other points into the upper hemisphere.

The algorithm moves from one dividing plane to the next. At each Π_i, it counts the spherical triangular containments of antipodal points on the upper hemisphere; by Lemma 3, it can instead count the triangular containments on the plane Π'_i. Suppose $\theta_i, \theta_{i+1}, \ldots, \theta_j$ are in the upper hemisphere of Π_{i-1}; after the algorithm is done with Π_{i-1}, it moves to Π_i. During this transition, θ_i moves to the lower hemisphere, and some points, $\theta_{j+1}, \theta_{j+2}, \ldots, \theta_{j+k}$, move to the upper hemisphere. The algorithm then counts, in the plane Π'_i,

(**a**) for each point in $\{\Pi'_i(\hat{\theta}_{j+k+1}), \Pi'_i(\hat{\theta}_{j+k+2}), \ldots, \Pi'_i(\hat{\theta}_{i-1})\}$ (the old antipodal points), the number of triangles containing it that have one or more of their vertices in $\{\Pi'_i(\theta_{j+1}), \ldots, \Pi'_i(\theta_{j+k})\}$ (the new primary points), while the rest of the vertices are in $\{\Pi'_i(\theta_{i+1}), \ldots, \Pi'_i(\theta_j)\}$ (the old primary points); and

(**b**) for $\Pi'_i(\hat{\theta}_i)$ (the new antipodal point), the number of triangles containing it that have all of their vertices in $\{\Pi'_i(\theta_{i+1}), \ldots, \Pi'_i(\theta_j), \Pi'_i(\theta_{j+1}), \ldots, \Pi'_i(\theta_{j+k})\}$ (all the primary points).

Then the algorithm moves on to Π_{i+1}. It does the above for all the n dividing planes.

Lemma 4. *Each tetrahedron containing the origin will be counted twice by the algorithm.*

Proof. Since the point x is in general position with $P' \cup \{O\}$, for each tetrahedron containing the origin, the axis of rotation $\overline{Ox}$ intersects with two of its faces. For each of these two faces, its three vertices can not be separated from the other vertex of the tetrahedron by any of the dividing planes, and therefore, their projected images do not contain the antipodal of the other vertex of the tetrahedron. For each of the other two faces of the tetrahedron, it will be counted excatly once, as soon as

1. the other vertex of the tetrahedron is in the lower hemisphere, but one or more of its three vertices have just moved to the upper hemisphere so that all of them are in the upper hemisphere, or

2. the other vertex of the tetrahedron has just moved to the lower hemisphere, while the three vertices are all in the upper hemisphere. □

The above algorithm counts a spherical triangle containing an antipodal point when the configuration makes the transition to the upper hemisphere. The flaw of the algorithm in [3] is in that it treats Π_0 differently: it counts *all* containments in Π_0, not just the *new* ones as if rotating from Π_{n-1} to Π_0; some of the containments in Π_0 will move out and reappear later, and then they are counted again; therefore, some tetrahedra are counted three times. Except for this difference, the two algorithms work in the same way, and they take $O(n^2)$ time; see [3] for details.

3 Computing simplicial depth in E^4

We can easily adapt the algorithm in Section 2.2 to four-dimensional space. Lemmas 1, 2, and 3 can be generalized to E^4. Furthermore, we need two points, x_1 and x_2, in general position with $P' \cup \{O\}$; the plane defined by O, x_1, and x_2 is the *axis of rotation*. And Λ is the plane that contains O and is orthogonal to the axis of rotation. The *dividing hyperplane* Π_i is similarly defined: it contains O, x_1, x_2, and the points in Λ with the polar angle $\alpha_i + \epsilon_i/2$. The algorithm then proceeds the same way except that it counts tetrahedra in hyperplanes Π_i' that contain antipodal points, instead of counting triangles in planes.

Lemma 5. *Each four-dimensional simplex containing the origin will be counted twice by the algorithm.*

Proof. For each simplex containing the origin, the axis of rotation intersects with three of its faces, and thus each of these faces will not be separated from the other vertex of the simplex by any of the dividing hyperplanes. For each of the other two faces of the simplex, it will be counted exactly once. □

There are $O(n)$ antipodal points in step (**a**), and there is one antipodal point in step (**b**); in both steps, there are $O(n)$ primary points. Therefore the algorithm needs $O(n^2)$ time for each of the antipodal points in a dividing hyperplane, and $O(n^3)$ time for each of the n dividing hyperplanes, and $O(n^4)$ time in total, while the straightforward method takes $O(n^5)$ time.

The same idea can be used to compute simplicial depth in E^d, $d > 4$, where the problem is similarly turned into computing $O(n^2)$ simplicial depths in E^{d-1}. The resulting algorithm takes $O(n^{2d-4})$ time; however, the straightforward algorithm is at least as good.

Acknowledgment

We wish to thank Regina Liu for her having proposed the subject to us and for her support of our study. We also wish to thank Vašek Chvátal and William Steiger: their careful reading of and comments on an earlier version helped us improve the presentation. William Steiger's encouragement at the beginning and his guidance throughout this work helped us tremendously.

References

[1] E. Boros and Z. Füredi, "Triangles Covering the Centre of an n-set", *Geometriae Dedicata* **17** (1984), 69-77.

[2] A. Y. Cheng, "Control Chart Techniques for Complex Settings and Establishment of Threshold System", Ph.D. dissertation, Rutgers University, in preparation.

[3] J. Gil, W. Steiger, and A. Wigderson, "Geometric Medians", *Discrete Mathematics* **108** (1992), 37-51.

[4] S. Khuller and J. S. B. Mitchell, "On a Triangle Counting Problem", *Information Processing Letters* **33** (1989), 319-321.

[5] R. Y. Liu, "On a Notion of Data Depth Based on Random Simplicies", *The Annals of Statistics* **18** (1990), 405-414.

[6] R. Y. Liu and K. Singh, "A Quality Index Based on Data Depth and Multivariate Rank Tests", *Journal of American Statistical Association* **88** (1993), 252-260.

[7] F. P. Preparata and M. I. Shamos, "Computational Geometry", Springer-Verlag, New York, 1985.

[8] P. J. Rousseeuw and I. Ruts, "Bivariate Location Depth", *Applied Statistics* **45** (1996), 516-526.