# Data Depth Contours - a Computational Geometry Perspective

Eynat Rafalin * Diane Souvaine*

## Abstract

**Data depth** is a statistical analysis method that is based on the shape of the data. *Depth contours* are nested contours that enclose regions with increasing depth. They provide powerful tools to visualize and compare data sets. Several contradicting definitions for depth contours exist in the statistical community. We provide a framework to analyze the competing notions which we term *cover* and *rank*. The important contribution of this paper is in analyzing inconsistencies and highlighting the open computational questions raised by the two approaches.

## 1 Introduction

A **data depth** measures how deep (or central) a given point $x \in \mathbb{R}^d$ is relative to $F$, a probability distribution in $\mathbb{R}^d$ or relative to a given data cloud. Data-depth provides *center-outward* orderings of points in Euclidean space of any dimension. It provides an alternative to classical statistics because no assumption about the underlying distribution is needed: data is analyzed according to the relative position of the data points (the 'shape') and deals with outliers. Some examples of data depth functions are *half-space depth* [6, 13] and *simplicial depth* [7]. The concept has attracted much recent attention in the computational geometry community.

**Depth contours** are nested contours that enclose regions with increasing depth. They were first introduced by Tukey as a data visualization tool for a two dimensional data [13] (half-space depth contours). Since then their use have been expanded for visualizing data sets (e.g. [11]) and quantifying and comparing data sets (e.g. [8]).

The statistical literature contains several contradicting definitions for computing depth contours. We term the two main approaches *cover* and *rank*. The *cover* approach received far greater attention from the computational geometry community but the *rank* method may prove to be more computationally effective.

Most depth functions are defined in respect to a probability distribution $F$, treating $\{X_1, .., X_n\}$ random observations from $F$ (we term this the **continuous case**). The **finite sample version** of the depth function results from replacing $F$ by $F_n$, the empirical distribution of the sample $\{X_1, .., X_n\}$. There is no debate regarding the definition of depth contours under the continuous case. However, translating to the finite sample case leads to different interpretation and the competing definitions of depth contours. We provide a framework to analyze the competing notions and highlight the discrepancies and open computational questions.

Section 2 provide the necessary background, including the definitions for the continuous and finite sample cases, under the framework we devised. Section 3 highlights the main differences between the two approaches and Section 4 contains important computational question raised by the competing definitions.

## 2 Depth Contours Background

Classical statistics most often focuses on the continuous case and depth contours were initially defined for this case. The discrepancy and competing definitions are caused because of different translations from the continuous case to the finite sample case.

### 2.1 The Continuous case

Let $D_F(x)$, $x \in \mathbb{R}^d$, be the value of a given depth function for point $x$ with respect to a probability distribution $F$.

The set $\{x \in \mathbb{R}^d : D_F(x) = t\}$ is called the **contour of depth** $t$. We usually refer to the *region enclosed by the contour* of depth $t$, the set $R_F(t) = \{x \in \mathbb{R}^d : D_F(x) > t\}$.

*Department of Computer Science, Tufts University, Medford, MA 02155. {erafalin, dls}@cs.tufts.edu

The $\alpha$ **central region,** $C_\alpha$ $(0 \leq \alpha \leq 1)$ is the smallest region enclosed by depth contours with probability $\alpha$. Under reasonable conditions $\partial C_\alpha = \{x \in \mathbb{R}^d : D_F(x) = t_\alpha\}$, where $P\{x \in \mathbb{R}^d : D_F(x) \leq t_\alpha\} = \alpha$. Under additional restrictions, $C_\alpha = R_F(t_\alpha)$, meaning that the contour of depth $\alpha$ is the $t_\alpha$th central region.

Desirable requirements for depth contours [4] are that for samples from certain classes of distributions, such as the elliptic ones, the depth contours should track the contours of the underlying model and that the contours should not be greatly influenced by outliers in the data set. Both approaches adhere to these requirements (for half-space depth see [3]).

## 2.2 The Finite Sample Case

Given $X = \{X_1, \cdots X_n\}$, let $D_n(x) = D(x)$ be the depth of $x \in \mathbb{R}^d$ with respect to the data set $X$. We wish to define the sample versions of the *contour of depth* $t$ and the *$\alpha$th central region.*

Several interpretations were suggested and debated by statisticians. We categorize the two main methods as the *cover* approach, most frequently studied by computational geometers, and the *rank* approach, which may produce contours more efficiently. Although lacking some valuable characteristics of the contours produced under the *cover* methods, the *rank*-method depth contours provide a reasonable, and perhaps less expensive, approximation.

### 2.2.1 Cover contours

The approach based on the notion of **cover** was first introduced by Tukey [13] for describing two dimensional half-space depth contours and has been discussed in [12, 11, 3, 4]. This approach defines the **sample version of the contour of depth** $t$ as the boundary of the set of points in $\mathbb{R}^d$ with depth $\geq t$. This approach constructs a contour of depth $t$ around all points (not necessarily from the original data set), that are of depth $k$. To compute the percentage of points of the original data set that are inside the contour one has to count the points of the data set that are inside the contour and compare to the total size of the data set.

There are two approaches to construct the **sample version of the $\alpha$ central region**:

- It can be built by enclosing all points that are of depth $D_\alpha(X_{[\lceil \alpha n \rceil]})$, which is the depth of the $\lceil \alpha n \rceil$ deepest sample point [3, 4]. The contour is constructed by sorting all points of the original set according to their depth (called the *order statistics*), breaking ties at random. A contour that encloses, for example, 75% of the points is created by taking the $\lceil .75n \rceil$ deepest data point $X_{[\lceil .75n \rceil]}$, and constructing the contour of depth $D(X_{[\lceil .75n \rceil]})$. According to this approach the depth contour can be a $d$ dimensional region in $\mathbb{R}^d$; we look for a $d-1$ dimensional contour. This inconsistency is solved by taking the outer boundary of this region as the contour. The contours converge to contours of the continuous case when $n \to \infty$ [3].
- Rousseeuw [12, 11, 10] considers the two adjacent contours $D_k$ and $D_{k-1}$ where $D(X_{[\alpha n]}) = k$ ($D_k$ encloses $< \alpha$ of the points, and $D_{k-1} > \alpha$). The $\alpha$th contour will be constructed *not* by extending the inner contour until it encloses $\lceil \alpha n \rceil$ of the points (as above), *but rather* by interpolating between the coverage of the two contours, according to the percentage of points inside each, using the deepest point as the center. This will create a contour that may have no common point with the data set.

In terms of computational complexity both approaches are similar, since to compute the $\alpha$ central hull, one has to compute a single $D(X_{[\lceil \alpha n \rceil]})$ contour or the two adjacent $D(X_{[\lceil \alpha n \rceil]})$ and $D(X_{[\lceil \alpha n \rceil - 1]})$ contours and interpolate.

Rousseeuw *et al.* advocate for this model based on the assumption that the points are random samples from a probability distribution and that the contours should represent the behavior of the distribution and not the specific points.

### 2.2.2 Rank Contours

A different approach, advocated for in [8], is based on **rank**. It defines the **sample $\alpha$th central region** as the convex hull containing the most central fraction of $\alpha$ sample points. The contour is constructed by sorting all points of the original set according to their depth. A contour that encloses, for example, 75% of the points is

created by taking the convex hull around data points $X_{[1]}, \cdots X_{[[.75n]]}$.

# 3 Comparing the Definitions

Under reasonable condition the depth contours for the *continuous* version are equivalent for *cover* and *rank* [8]. However, the contours for the sample version differ significantly. While the *rank* approach maintains the rank order of the sample points the *cover* approach mimics the continuous quantiles. A few differences:

- *Rank* contours can be constructed according to any depth function that calculates the depth of a data point, while contours according to *cover* require one to compute the depth of any other point $x \in \mathbb{R}^d$.
- Both approaches create depth contours that are nested. However, the *cover* method may produce contours that are not connected (if the depth function is not connected, e.g. simplicial depth in [1]). In addition, *rank* contours will always be convex, while the *cover* contours may not be. For the half-space function the *cover* contours are always convex [3].
- The main visual difference between the two approaches are the vertices of the contours. Every data point will be a vertex of (at least one) depth contour. However, the vertices of the *rank* contours will only be data points while the vertices of the *cover* contours can be any point from the data set. A vertex will be common to more then one *rank* contour while only degenerate data sets will create common vertices for *cover* contours.
- Degenerate data sets lead to contours with peculiar properties: If several points have the same depth values, under the *rank* approach they may appear on different depth contours, and thus the *random* order of equal depth points can create different sets of contours. To remedy this [8] require that equal depth points belong to the same sample $p$th level contour for some $p$. However in degenerate cases such as when all the points are on the convex hull of the set, this constructs only one valid contour.

  Take for example the contour that contains 50% of the data points, but assume a degenerate set that contains more then 50% of the data points on its convex hull. Different depth ordering of the data points of minimal depth (points on the convex hull) will create varying contours according to the *cover* approach (depending on which of the points on the convex hull is chosen to be in the 50% inner points). According to Donoho and Gasko's *cover* approach the 50% contour will be identical to the 100% contour since the depth of the $X_{[[n/2]]}$ point is identical to the depth of any point on the convex hull. Only Roussuew's *cover* approach will create different, non overlapping, contours for every percentile, since it interpolates between the $< 50\%$ contour and the $> 50\%$ contour which is the outer convex hull.

# 4 Computational questions For Half-space Depth Contours

In analyzing the open computational questions we concentrate on the half-space depth contours. However, most questions are valid (and open) under any other depth function.

**Half-space depth**[1] of a point $x$ relative to a set of points $\mathcal{S} = \{X_1, ..., X_n\}$ is the minimum number of points of $\mathcal{S}$ lying in any closed half-space determined by a line through $x$ [6, 13] The half-space depth is robust, affine invariant and does not rely on distance.

## 4.1 Computation of All Contours

An $\Theta(n^2)$ time implementation computes all the *cover* depth contours, the depth of all the data points [10] using duality and topological sweep. Since the size of the contours can be quadratic (e.g. if all data points are on the convex hull and every vertex of the contours set is unique), in non degenerate cases, this is the best we can hope for.

As for *rank* contours, the cost of storing the set of *all depth contours* can be quadratic (for example, if all data points are on the convex hull and hence adjacent contours differ by only one point). However, if we can order the data points

[1] In the literature this is sometimes called *location depth* or *Tukey depth*.

by their depth, we can use $O(n \log n)$ storage in order to construct the contours incrementally [5]. The construction will be output sensitive in the size of the set and the $k/n$th contour will be be constructed in $O(k \log n)$ time.

### 4.2 A Subset of the Depth contours

Instead of computing all contours is $O(n^2)$ (see Section 4.1), we would like to compute a contour set of constant size $c$ (e.g. the ones that contain $a_1\%, a_2\%, \cdots a_c\%$ of the data, or the single *Bag*) more efficiently (e.g. sub-quadratic). One approach to compute *cover* depth contours is to locate the $a_1 n \cdots a_c n$ deepest points and then compute their depth (in $O(n \log n)$ time). Next, using Matoušek's ideas [9] construct every one of the the contours in $O(n \log^4 n)$ time[2]. To the best of our knowledge finding the $a_i n$ deepest point can be done in no better then $O(n^2)$ time (since the point has to be from the data set, and not in $\mathbb{R}^2$ other solutions will not apply), dominating the complexity of the algorithm. A different approach is based on binary search, first constructing a contour of depth $k$ ($O(n \log^4 n)$ [9], as above), counting the number of data points enclosed by it, and binary searching to find the contour enclosing $c_i n$ of the data points. This adds a log factor to the cost of computing a single contour, but avoids finding the $c_i n$ deepest point. The above two approaches are theoretically correct, however, in practice only a quadratic time algorithm exists (for example [10]).

For *rank* based contours, once we sort the points according to their half-space depth, computing the contours can be done in $O(n \log n)$ using the incremental approach. However, to the best of our knowledge no sorting procedure exists. Similarly, if we identify the $c_i n$ deepest point and could partition based on its depth in sub-quadratic time, we would be able to construct the convex hull of the deepest set in $O(n \log n)$. To the best of our knowledge, no such efficient partitioning method exist.

To summarize, existing algorithms produce $O(n^2)$ time solutions. For improved time we need an efficient predicate that compares two data points and reports which one is deeper.

[2] Chan [2] claims this can be improved using his technique to $O(n \log^3 n)$ expected time.

## References


[1] M. Burr, E. Rafalin, and D. L. Souvaine. Simplicial depth: An improved definition, analysis, and efficiency for the sample case. Technical report 2003-28, DIMACS, 2003.

[2] T. M. Chan. An optimal randomized algorithm for maximum tukey depth. In *Proceedings of 15th ACM-SIAM Symposium on Discrete Algorithms (SODA04)*. ACM Press, 2004.

[3] D. L. Donoho and M. Gasko. Breakdown properties of location estimates based on halfspace depth and projected outlyingness. *Ann. Statist.*, 20(4):1803–1827, 1992.

[4] X. He and G. Wang. Convergence of depth contours for multivariate datasets. *Ann. Statist.*, 25(2):495–504, 1997.

[5] J. Hershberger and S. Suri. Applications of a semi-dynamic convex hull algorithm. *BIT*, 32:249–267, 1992.

[6] J. Hodges. A bivariate sign test. *The Annals of Mathematical Statistics*, 26:523–527, 1955.

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

[8] R. Liu, J. Parelius, and K. Singh. Multivariate analysis by data depth: descriptive statistics, graphics and inference. *The Annals of Statistics*, 27:783–858, 1999.

[9] J. Matoušek. Computing the center of planar point sets. *DIMACS Series in Disc. Math. and Theoretical Comp. Sci.*, 6:221–230, 1991.

[10] K. Miller, S. Ramaswami, P. Rousseeuw, T. Sellarés, D. Souvaine, I. Streinu, and A. Struyf. Efficient computation of location depth contours by methods of combinatorial geometry. *Statistics and Computing*, 13(2):153–162, 2003.

[11] P. J. Rousseeuw, I. Ruts, and J. W. Tukey. The bagplot: A bivariate boxplot. *The American Statistician*, 53:382–387, 1999.

[12] I. Ruts and P. J. Rousseeuw. Computing depth contours of bivariate point clouds. *Comp. Stat. and Data Analysis*, 23:153–168, 1996.

[13] J. W. Tukey. Mathematics and the picturing of data. In *Proc. of the Int. Cong. of Math. (Vancouver, B. C., 1974), Vol. 2*, pages 523–531. Canad. Math. Congress, Montreal, Que., 1975.