IcETRAN 2014

Proceedings of papers

International Conference on Electrical, Electronic and Computing Engineering

Vrnjačka Banja, Serbia, June 2 – 5, 2014

ISBN 978-86-80509-70-9

Multivariate Medians and Halfspace Depth: Algorithms and Implementation

Milica Bogicevic, Milan Merkle

***Abstract*—We are considering the notions, properties and algorithms' implementations of data depth which represents the median of higher dimensional data. Our main objective is to present the snapshot of the data depth with respect to half-space depth, also known as location depth or Tukey depth. Although the problem is NP-hard, there are ways to compute nontrivial lower and upper bounds of the depth. Computation of Tukey depth is very demanding and even for low dimension dataset, it requires all one dimensional projections to be considered. This is the reason why implementations of particular algorithms represent a challenge, not only in order to calculate deepest data location, but also in order to visualize initial data set and its calculated results.**

***Keywords*—Algorithms; Data depth; Tukey depth; robustness; Multivariate median.**

I. Introduction

The motivation to generalize centrality of data is natural as far as there is necessity to analyze data and its behavior, found dependencies between observations and classify them. It helps to define an ordering and a version of ranks in multivariate data. Depending on how data is described and how its main properties are defined, results are different.

There are several functions for determining data depth. All depth functions measure the centrality of a point θ with respect to the data set or a probability distribution. Tukey's half-space depth is one of the most popular depth functions. Half-space depth of a point θ relative to the data set is defined as the smallest number of observations in any closed half-space with boundary through θ. The deepest location, i.e. the point with maximal half-space depth, is a generalization of median or "center" of the data.

Sometimes the mean is good enough to describe data centrality, but median is more robust since it less sensitive to outliers and heavy-tailed distributions. Robustness is one of the main median's properties. In order to illustrate how important it is, this simple example would be sufficient: it is enough to place one outlier to change the mean, but to change the median in one dimension up to one half of the data can be changed without affecting the median.

The Section II deals with properties of other properties of medians as well as definitions and notions in multivariate setup.

In Section III we present notions, meaning and construction of bivariate half-space median, introducing HALFMED algorithm.

In Section IV we present algorithm called DEEPLOC for approximation the deepest location or maximal half-space depth in higher dimension.

In Section V we present implementation and visualization of various half-space depth function using data generated from various distributions.

Milica Bogicevic – Faculty of Electrical Engineering, University of Belgrade, Bulevar Kralja Aleksandra 73, 11020 Belgrade, Serbia (e-mail: antomripmuk@ yahoo.com).

Milan Merkle – Faculty of Electrical Engineering, University of Belgrade, Bulevar Kralja Aleksandra 73, 11020 Belgrade, Serbia (e-mail: emerkle@etf.bg.ac.rs).

II. Half-space Depth Definitions and Properties

For multivariate data Tukey's half-space depth is one of the most popular depth functions in the literature. The Tukey median, i.e. the multivariate median associated with the half-space depth, is also a well-known measure of center for multivariate data.

There are several definitions of half-space depth and we consider some of them.

In p dimensions, the half-space location depth of a point θ relative to a data set is denoted as $ldepth(\theta; X_n)$. It is defined as smallest number of observation in any closed half-space with boundary through θ.

In the univariate setting ($p = 1$) , this definition becomes:

$$ldepth_1(\theta; X_n) = \min(\#\{x_i \le \theta\}, \#\{x_i \ge \theta\}). \qquad (1)$$

The multivariate $ldepth$ can be seen as the smallest univariate $ldepth$ of θ relative to any projection of the data set onto a direction u, since

$$\begin{aligned} ldepth(\theta; X_n) &= \min_{\|u\|=1} ldepth_1(u'\theta; u'X_n) \\ &= \min_{\|u\|=1} \#\{i; u'x_i \le u'\theta\} \end{aligned} \qquad (2)$$

In words, $ldepth(\theta; X_n)$ says how deep θ lies in the data cloud [4][2].

Another definition concerns the data distribution, i.e. given a probability distribution P defined in a multidimensional space X, a depth function tries to order data in X from the center of P to the outer of P. In other words, if data is moved towards the center of data cloud, then its depth increases and if the data is moved towards the outside, then its depth decreases.

In the one dimensional case, the points are ordered by the

function:

$$x \to D_1(x,P) := \min\{P(-\infty,x], P[x,\infty)\}. \quad (3)$$

If $x \in \mathbb{R}^p$ then the half-space depth of point x with respect to P, $D_T(x,P)$, is the minimal probability that can be attained in the closest half-space that contains x [6][11], i.e.

$$D_T(x;P) = \inf_H \{P(H) : H \text{ a closed halfspace in } \mathbb{R}^d : x \in H\}. \quad (4)$$

The following definition considers half-space depth as a given set P of n points in $\mathbb{R}^d$ [7]. The half-space depth of a point $q \in \mathbb{R}^d$ is defined as:

$$\min\{|P \cap \gamma| : \text{over all halfspaces } \gamma \text{ containing } q\}. \quad (5)$$

All definitions (1), (2), (3), (4) and (5) are about the same depth function, half-space function. The differences are format and data dimensionality. For all of them the same properties can be applied [2]:

- Affine invariance – depth of a point $x \in \mathbb{R}^d$ should not be dependent on underlying coordinate system.
- Vanishing at infinity – the depth of point x should approach to zero as its norm approaches infinity, i.e. $D(x,P) \to 0, \|x\| \to 0$.
- Maximality at center – for a symmetric distribution the maximum value of depth should be attained at its center.
- Monotonicity relative to the deepest point – as a point $x \in \mathbb{R}^d$ moves from a deepest point along some fixed ray, its depth should decrease monotonically.

The term "center" at third point is used to denote point of symmetry – we can say that a random vector X is *half-space symmetric* around θ if $P(X \in H) \geq \frac{1}{2}$ for every closed half-space H containing θ.

In addition to these properties, one of the most the most cited is *breakdown property* [2]. In Section I we mentioned advantage of the median relative to the mean value. That advantage describes in short breakdown property. The breakdown value is a measure of the robustness of an estimator against outlying observations. It indicates the smallest fraction of incorrect observation in the sample that causes the estimator to "break down", or to take on values that are arbitrarily bad or meaningless. The higher the breakdown point of an estimator, the more robust it is. Half-space median has a breakdown point of at least $\frac{1}{p+1}$ in dimension p and the breakdown point can be as high as $\frac{1}{3}$. In contrast, various estimators that reject apparent outliers and afterwards calculate the mean of the remaining observation have breakdown point not larger than $\left\lceil \frac{n}{2} \right\rceil$ in dimension p.

Since the half-space median is represented as the deepest point, i.e. the point with the maximum half-space depth, half-space median is not unique point. The set of points of maximal depth is guaranteed to be a closed, bounded convex set and thanks to those three properties, half-space lies between $\left\lceil \frac{n}{p+1} \right\rceil$ and $\left\lfloor \frac{n}{2} \right\rfloor$ [2] as consequence of Helly's theorem. The maximal guaranteed depth in general is $\frac{d}{n+1}$. However, if we define depths as in (5), but over subfamily of halfspaces that are defined by tangents planes to a given convex cone, then the median property is preserved, that is, the maximal depth is at least $\frac{n}{2}$ [3] regardless of the shape of data cloud. The resulted median set in this case is not affine invariant in general. It is shown in [5] that for some symmetric distributions of data points the standard Tukey median has depth of $\frac{n}{2}$.

III. Bivariate Tukey Median - HALFMED

In the case of bivariate data instead of halfspace depth, the term will be halfplane depth.

The halfplane location depth of a point $\theta \in \mathbb{R}^2$ relative to the bivariate dataset $X = \{x_1, x_2, ..., x_n\}$ is the minimal number of observations in any closed halfplane that contains θ.

$$ldepth(\theta, X) = \min_H \#\{i; x_i \in H\}, \quad (6)$$

where H ranges over all closed halfplanes of which the boundary line passes through θ [1].

In order to calculate bivariate halfplane depth, new term is introduced: *depth region of depth k* which is considered as set D_k of points θ with $ldepth(\theta, X) \geq k$. In other words, D_k is the intersection of all closed halfplanes that contains at least $n-k+1$ observations. It is important to note that $D_1 \supset D_2 \supset D_3 ...$ The boundary of D_k is a convex polygon, called *contour of depth k*.

With this notion, halfplane median is defined as the θ with depth k^*. If D_{k^*} is not single point, halfplane median is defined as the center of gravity of D_{k^*} [2].

The basic idea of HALFMED algorithm uses the median property of maximal, e.g. minimal value. In the case of bivariate data set, halfplane median takes values between $\left\lceil \frac{n}{3} \right\rceil$ and $\left\lfloor \frac{n}{2} \right\rfloor$. Algorithm construct several regions D_k, where $\left\lceil \frac{n}{3} \right\rceil \leq k \leq \left\lfloor \frac{n}{2} \right\rfloor$ in order to find k^* for which $D_{k^*} \neq \varnothing$ and $D_{k^*+1} \neq \varnothing$.

1) The first step of the HALFMED algorithm is to test whether any two data points x_i and x_j coincide. In the same

step, algorithm assigns start value for $k \leftarrow \left\lfloor \frac{1}{2}(k^{lower} + k^{upper}) \right\rfloor$.

2) The second step constructs D_k. The main concept is that of a special k-divider, represented as a direct, oriented line passing through two observation and dividing on that way data set into two parts: the first part has $n-k-1$ observations lie strictly to its left and the second part has exactly $k-1$ observations lie strictly to its right. There are more than one special k-divider represented as L_{ij} lines drown between x_i and x_j points. Lines are sorted according to their angles α_{ij}, i.e. L_{ij} make angle with horizontal axis such that $0 \le \alpha_{ij} \le \pi$. This means that HALFMED starts from the projection of data points on the horizontal direction. D_k is the intersection of the closed halfplanes to the left of all L_{ij} that satisfies condition of special k-divider.

3) In step three, k value is updated in the following way:

If $D_k = \varnothing \Rightarrow k_{new}^{lower} \leftarrow k_{old}^{lower}$ and $k_{new}^{upper} \leftarrow k$ (previous).

If $D_k \neq \varnothing \Rightarrow k_{new}^{lower} \leftarrow k$ and $k_{new}^{upper} \leftarrow k_{old}^{upper}$.

If $k_{new}^{upper} - k_{new}^{lower} \ge 2 \Rightarrow k \leftarrow \left\lfloor \frac{1}{2}(k_{new}^{lower} + k_{new}^{upper}) \right\rfloor$ and return to step *2)*.

If $k_{new}^{upper} - k_{new}^{lower} = 1 \Rightarrow k^* = k_{new}^{lower}$, i.e. k^* is found as well as corresponding D_{k^*}.

4) If vertices of D_k are $\{y^1,...,y^m\}$, where $m \le n$, then the central point is

$$T^o = \frac{1}{m}\sum_{j=1}^{m} y^j. \tag{7}$$

If $m \le 3$ the halfplane median is equal to T^o. For $m \ge 4$ halfplane median is

$$T^* = \frac{\int xI(x \in D_{k^*})d\lambda(x)}{\lambda(D_{k^*})}, \tag{8}$$

where λ is usual measure of area.

The algorithm HALFMED takes $O(n^2 \log^2 n)$ time.

Few months later, algorithm for the location depth of a point relative to a three-dimensional data set was proposed [10] and it takes $O(n^2 \log n)$ time. The basic idea of this algorithm is to connect θ according from the definition (2) with one of the data points x_i, thus we get line L. A plane containing L is rotated around L in discrete steps. Whenever the plane containing L passes through some point x_j that is not on L, the points on both sides of plane is counted as required in (2). For $i = 1,...,n$ algorithm obtain all possible positions of a plain through θ in the point cloud $\{x_1,...,x_n\}$. Instead of counting number of points to the left and to the right for every plane and for every particular line L (which takes $O(n^2)$ operations), algorithm constructs another plane γ through θ orthogonal on L and then projects all observations on the plane γ and then algorithm similar to the HALFMED is used to count points on both sides of a discrete set of lines through θ. For more details see Appendix of [10].

IV. High Dimensional Deepest Location - DEEPLOC

The deepest location, i.e. θ with the maximal half-space depth is a multivariate generalization of the median. Similar as in Section III the notion of *ldepth regions* are used to defined depth of a point θ according to (2)

$$D_k = \{\theta \in \mathbb{R}^p; ldepth(\theta; X_n) \ge k\}. \tag{9}$$

For each *ldepth* k hold that $D_k \subseteq D_{k-1}$. The region D_1 is equal to convex hull X_n. The boundary of each region is called *ldepth contour*. The smallest *ldepth* region, D_{k^*}, represents set of all points with maximal *ldepth* k^* [4].

DEEPLOC starts from selected initial point and then takes carefully selected directions in order to increase the *ldepth*.

1) Initial point is selected as coordinate wise median $M_1 = (med_{i=1}^n x_{i1},...,med_{i=1}^n x_{ip})$.

2) Next step constructs m directions, $u \in \mathbb{R}^p$ with $\|u\| = 1$ which are randomly drawn from four classes of directions:

- The p coordinate axes
- Vectors connecting an observation with M_1
- Vectors connecting two observations
- Vectors perpendicular to a p-subset of observations.

3) The univariate *ldepth* of M_1 relative to the projection of X_n on each of these m directions is calculated and set U_{move} of directions u that yield the same lowest $\#\{i; u'x_i \le u'M_1\}$ is stored. Algorithm considers these as directions in which *ldepth* can still be improved. In order to do that, algorithm computes average

$$u_{move} = \frac{1}{U_{move}} \sum_{u \in U_{move}} u \tag{10}$$

4) Then, algorithm take step in the direction u_{move}. The *ldepth* attained by the deepest location relative to X_n must be at least $\left\lceil \frac{n}{p+1} \right\rceil$ [2]. If $ldepth_1(M_1; u'_{move} X_n) < \left\lceil \frac{n}{p+1} \right\rceil$ then algorithm takes a step large enough to reach a point M_2 which has univariate *ldepth* $\left\lceil \frac{n}{p+1} \right\rceil$ in direction u_{move}.

Otherwise, algorithm takes step such that the univariate *ldepth* of the resulting point M_2 in direction u_{move} is 1 unit larger than *ldepth* of M_1 in the same direction. Then,

algorithm repeats step 2) starting from M_2. Algorithm iterates until maximal halfspace depth becomes $\frac{n}{2}$. For detailed description, see Appendix given in [4].

Described algorithm has time complexity $O(kmn\log n + kpn + mp^3 + mpn)$, where k is the number of steps taken by the algorithm.

After taking all described steps, contours are determined. The Theorem 1. in [8] shows that empirical distribution of any dataset $X_n \subset \mathbb{R}^p$ is uniquely determined by its halfspace depth function, i.e. the list of contours $\{D_1,...,D_{k^*}\}$. This statement represents one more important property of halfspace depth function called *distributional property*.

Algorithm with minimum complexity is represented in [7]. The algorithm is randomized and requires $O(n\log n)$ expected time for n data points, when $p<3$. For $p\geq 3$ the expected time bound is $O(n^{p-1})$. This approach combines optimization randomized techniques and linear-programming-type problems. We describing algorithm in a few steps. For detailed description see [7].

At the first step, algorithm finds points with a depth at least k. In further description, author switched problem to dual space consisting of all *linear functionals* (or *linear forms* defined as *linear map* from a vector space to its field of scalars, where each point from initial dataset is represented as vector, i.e. $X=\{x_1,...,x_p\}$). By minimizing linear function that separates points with depth larger than k from the points that have depth less then k, the output of this approach is subset of points that have depth at least k.

Furthermore, *Cutting Lemma* is presented in order to optimize algorithm. Since the initial set is switched to a dual space, lemma proofs that it is possible to divide set of n hyperplanes (output of the first step) into a constant number of simplices such that each simplex intersects at most $\lceil \alpha n \rceil$ hyperplanes for some constant $\alpha<1$. On that way, the problem is divided into subproblems where each simplex represents a subproblem. The aim is to find maximal half-space depth in each simplex. The cutting lemma, also known as *cell decomposition lemma*, can be defined as follows: given n lines in the plane it is possible to divide it into $O(r^2)$ regions (even triangles) for any $1\leq r\leq n$ such that interior of any region is intersected by $O(\frac{n}{r})$ lines, i.e. there are $\frac{1}{r}$ (α) cuttings.

After dividing the space on the subspaces (subproblems or simplices), the next step is maximizing k in each simplex.

This algorithm combines two techniques: in the first step, dataset is pruned and this is close to the *prune-and-search* technique described in details in [17]. The cutting lemma in the context of finding maximal half-space depth is most similar to the *divide-and-conquer* technique represented in [18].

V. Implementation and Visualization

For the purpose of demonstrating the DEEPLOC algorithm, we have implemented following:

- Multivariate normal random generator for generating test dataset in Java programming language
- Multidimensional implementation of DEEPLOC algorithm in Java programming language
- Web interface for visualizing both generated random dataset and results of DEEPLOC algorithm over the selected pairs of variables (dimensions)

Random dataset from multivariate normal distribution was generated so we could also test DEEPLOC algorithm implementation on neutral random dataset, however, this implementation could be used against any other dataset.

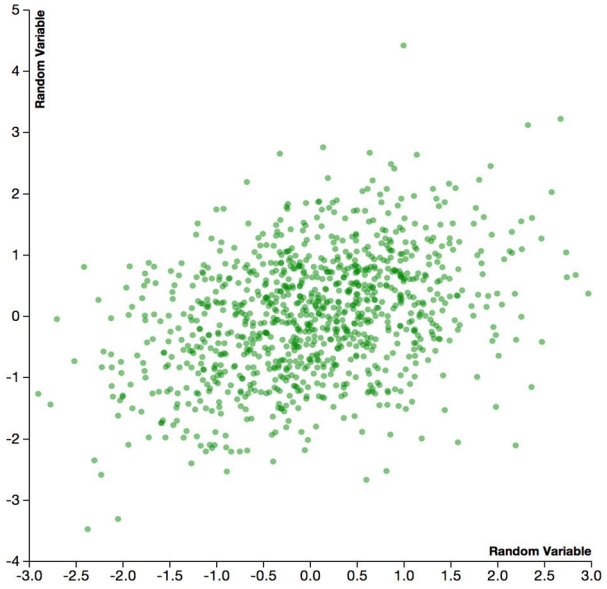

Figure 1. Bivariate normal random dataset visualization example

Dataset from multivariate normal distribution was generated using the random vectors from uniform distribution using the inverse transform sampling. Vectors from uniform distribution were generated using the combined multiple recursive generator algorithm (CMRG) [13] implemented in SSJ Java library using COLT library for fast matrix operations.

DEEPLOC algorithm was also implemented in Java language and can be used with either random vectors generated using the web application, or use dataset stored in database. We have been using simple model stored in MongoDB that can be used to store both datasets and end results of DEEPLOC algorithm. Application is further extensible to support any other depth algorithm and will be used to further explore options to efficiently find data depth on multi-dimensional data.

Both data generator and DEEPLOC algorithm implementation are exposed as RESTful web services and since it's completely decoupled from web application they can be used independently for generating various

distributions and interfacing data depth calculation from other applications using RESTful HTTP interface and utilizing simple JSON messages that contain the data/vectors.

We have developed web application encouraged by John Tukey's PRIME-9 [14], and as he has said: "Picturing of data is the extreme case. Why do we use pictures? Most crucially to see behavior we had not explicitly anticipated as possible—for what pictures are best at is revealing the unanticipated; crucially, often as a way of making it easier to perceive and understand things that would otherwise be painfully complex. These are the important uses of pictures." [9]

Our web application was implemented using AngularJS and simple Ruby Sinatra backend for routing the requests to adequate services and charts have been generated using D3.js library.

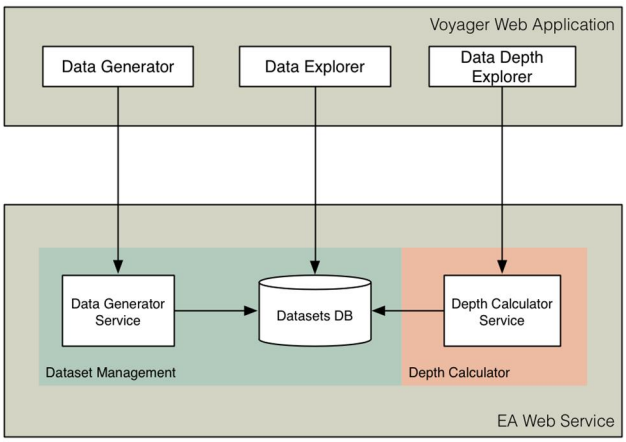

Figure 2. Diagram of demo application infrastructure

Top level architecture diagram of demo application can be seen on Figure 2. together with relationships between general components of demo application.

VI. Summary

In this article we have listed all known half-space definitions as well as their explanations. Furthermore, we have represented some of main half-space properties that are used widely in order to proof any other property. Also, we have explained some of half-space algorithms: HALFMED, algorithm for finding half-space depth in three dimensions, DEEPLOC and optimal randomizes algorithm for finding half-space depth in higher fixed dimensions. At the end, we're implemented DEEPLOC with intention to:

- Dataset management interface – for easier importing and management of external datasets; datasets other than those generated with Data Generator interface have to be imported manually into MongoDB database
- Data Generator support for various distributions – random vectors are currently being generated only from multivariate normal distribution, while we might want to test depth algorithm(s) on other distributions
- Depth Calculator support for various algorithms – we have currently implemented only DEEPLOC algorithm, but same framework could be used to test various other depth function algorithms and compare both efficiency and quality of results.

References

[1] P. J. Rousseeuw and I. Ruts (1998), "Constructing the Bivariate Tukey Median," *Statist. Sinica* **8** 827-839

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

[3] M. Merkle (2010), "Jensen's inequality for multivariate medians", J. Math. Anal. Appl., **370** 258-269

[4] A. Struyf and P. J. Rousseeuw (1999), "High-dimensional computation of the deepest location," *Comp. Statist. & Data Anal.*, **34** 415-426

[5] S. Dutta, A. K. Ghosh and P. Chaudhuri (2011), "Some intriguing properties of Tukey's half-space depth," *Bernoulli* **17** 1420-1434

[6] J.A. Cuesta-Albertos and A. Nieto-Reyes (2008), "The random Tukey depth" *Comp. Statist. & Data Anal,* **52** 4979-4988

[7] T. M. Chan, "An Optimal Randomized Algorithm for Maximum Tukey Depth", *NSERC Research Grant*

[8] A. Struyf and P. J. Rousseeuw (1998), "Half-space depth and regression depth characterize the empirical distribution"

[9] J. W. Tukey (1974), "Mathematics and Picturing of Data", *International Congress of Mathematicians,* Vancouver

[10] P. J. Rousseeuw and A. Struyf (1998), "Computing location depth and regression depth in higher dimensions", *Statist. and Comp.* **8** 193-203

[11] R. Serfling (2000), "Depth Functions in Nonparametric Multivariate Inference", *DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 72, NFS Grant DMS-0103698*

[12] S. Krishnan, N. H. Mustafa and S. Venkatasubramanian, "Statistical Data Depth and the Graphics Hardware", *DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 72, NFS Grants CCR-00-86013, EIA-98-70724, EIA-99-72879, EIA-01-31905 and CCR-02-04118*

[13] P. L'Ecuyer (1998), "Good Parameters and Implementations for Combined Multiple Recursive Random Number Generators"

[14] J. W. Tukey (1972), "PRIME-9", http://stat-graphics.org/movies/prim9.html

[15] J. Matousek (2000), "Computing the center of planar point set", In Computational Geometry: Papers from the DIMACS Special Year, 221-230

[16] O. Vencalek (2011), "Concept of Data Depth and Its Applications", Mathematica **50** 111-119

[17] T. M. Chan (1999), "Geometric Application of a Randomized Optimization Technique", Discrete Comput. Geom. **22** 547-57

[18] B. Chazelle, "Cutting Hyperplanes for Devide-and-Conquer", Grant CCR-9002352