Stat

The ISI's Journal for the Rapid
Dissemination of Statistics Research

(wileyonlinelibrary.com) DOI: 10.1002/sta4.8

Exact fast computation of band depth for large functional datasets: How quickly can one million curves be ranked?

Ying Sun[a,*], Marc G. Genton[b] and Douglas W. Nychka[c]

Received 4 September 2012; Accepted 13 September 2012

Band depth is an important nonparametric measure that generalizes order statistics and makes univariate methods based on order statistics possible for functional data. However, the computational burden of band depth limits its applicability when large functional or image datasets are considered. This paper proposes an exact fast method to speed up the band depth computation when bands are defined by two curves. Remarkable computational gains are demonstrated through simulation studies comparing our proposal with the original computation and one existing approximate method. For example, we report an experiment where our method can rank one million curves, evaluated at fifty time points each, in 12.4 seconds with Matlab. Copyright © 2012 John Wiley & Sons, Ltd.

Keywords: approximate solution; band depth; exact solution; functional boxplot; functional data; large dataset; modified band depth

1 Introduction

Functional data analysis is an attractive approach to study complex data in statistics. Functional data are observed in many disciplines, such as electroencephalogram (EEG) tests for brain wave activity in medicine, spatio-temporal evolution of cells in biology, climate variables from monitoring networks or climate models in meteorology, chemical processes in engineering, to name but a few; see Ramsay et al. (2009) for many applications. Even longitudinal data can be viewed from a functional angle (Zhao et al., 2004). Many univariate methods have been extended to functional data. López-Pintado & Romo (2009) introduced a notion of band depth to generalize order statistics to functional data. It provides an ordering within a sample of functions, thus makes univariate methods based on order statistics possible for functional data. For example, the median function or a trimmed mean function can be defined for robust statistical analysis, whereas functional boxplots and adjusted functional boxplots (Sun & Genton, 2011; 2012a) can be constructed for visualization; see Zuo & Serfling (2000) for key properties that a depth measure should satisfy.

Data depth is an important concept to order functional data. In general, it allows for ordering a sample of functional data from the center outwards and, thus, introduces a measure to define the centrality or outlyingness of an observation. Indeed, one can compute the depth values of all the sample curves and order them according to decreasing depth values. Suppose each observation is a real function $y_i(t)$, $i = 1, \dots, n$, $t \in \mathcal{I}$, where $\mathcal{I}$ is an interval in $\mathbb{R}$. Let

[a]Department of Statistics, University of Chicago, Chicago, IL 60637, USA
[b]Department of Statistics, Texas A&M University, College Station, TX 77843-3143, USA
[c]Geophysical Statistics Project, National Center for Atmospheric Research, Boulder, CO 80307-3000, USA
*Email: sunwards@galton.uchicago.edu

$y_{[i]}(t)$ denote the sample curve associated with the ith largest band depth value. Then $y_{[1]}(t),\dots,y_{[n]}(t)$ can be viewed as order statistics, with $y_{[1]}(t)$ being the deepest (most central) curve or simply the median curve, and $y_{[n]}(t)$ being the most outlying curve. The implication is that a smaller rank is associated with a more central position with respect to the sample curves. The order statistics induced by depth start from the most central sample curve and move outwards in all directions.

More specifically, López-Pintado & Romo (2009) defined the band depth through a graph-based approach. The graph of a function $y(t)$ is the subset of the plane $G(y) = \{(t, y(t)) : t \in \mathcal{I}\}$. The band in $\mathbb{R}^2$ delimited by the curves $y_{i_1},\dots,y_{i_k}$ is $B\left(y_{i_1},\dots,y_{i_k}\right) = \{(t, x(t)) : t \in \mathcal{I}, \min_{r=1,\dots,k} y_{i_r}(t) \leqslant x(t) \leqslant \max_{r=1,\dots,k} y_{i_r}(t)\}$. Let J be the number of curves determining a band, where J is a fixed value with $2 \leqslant J \leqslant n$. If $Y_1(t),\dots,Y_n(t)$ are independent copies of the stochastic process $Y(t)$ generating the observations $y_1(t),\dots,y_n(t)$, the population version of the band depth for a given curve $y(t)$ with respect to the probability measure P is defined as $BD_J(y,P) = \sum_{j=2}^{J} BD^{(j)}(y,P) = \sum_{j=2}^{J} P\{G(y) \subset B(Y_1,\dots,Y_j)\}$, where $B(Y_1,\dots,Y_j)$ is a band delimited by j random curves. The sample version of $BD^{(j)}(y,P)$ is defined as

$$BD_n^{(j)}(y) = \binom{n}{j}^{-1} \sum_{1 \leqslant i_1 < i_2 < \cdots < i_j \leqslant n} I\left\{G(y) \subseteq B\left(y_{i_1},\dots,y_{i_j}\right)\right\}, \tag{1}$$

where $I\{\cdot\}$ denotes the indicator function. Then, the sample band depth of a curve $y(t)$ is

$$BD_{n,J}(y) = \sum_{j=2}^{J} BD_n^{(j)}(y). \tag{2}$$

When curves are very irregular, few bands will completely contain a curve, and then many sample curves will have the same band depth value as implied by the indicator function involved in (2), which results in a poorly defined ranking. For example, the band defined by $J = 2$ curves that cross at one point does not contain any other curve with probability 1, hence it does not contribute to the band depth value. Another example is given by Brownian motion in continuous time, for which the band depth is identically zero with probability 1 for all J, for any n, and any time period $[0, T]$. To solve this problem, López-Pintado & Romo (2009) proposed a more flexible definition, the modified band depth (MBD), by measuring the proportion of time that a curve $y(t)$ is in the band:

$$MBD_n^{(j)}(y) = \binom{n}{j}^{-1} \sum_{1 \leqslant i_1 < i_2 < \cdots < i_j \leqslant n} \lambda_r\left\{A\left(y; y_{i_1},\dots,y_{i_j}\right)\right\}, \tag{3}$$

where $A_j(y) \equiv A\left(y; y_{i_1},\dots,y_{i_j}\right) \equiv \left\{t \in \mathcal{I} : \min_{r=i_1,\dots,i_j} y_r(t) \leqslant y(t) \leqslant \max_{r=i_1,\dots,i_j} y_r(t)\right\}$ and $\lambda_r(y) = \lambda(A_j(y))/\lambda(\mathcal{I})$, if λ is the Lebesgue measure on $\mathcal{I}$. Then, the sample modified band depth of a curve $y(t)$ is

$$MBD_{n,J}(y) = \sum_{j=2}^{J} MBD_n^{(j)}(y). \tag{4}$$

If $y(t)$ is always inside the band, the modified band depth degenerates to the band depth in (2).

From the definitions in (1) and (3), we can see that the computational cost of calculating the depth values in a sample with size n grows with the sample size at rate $\binom{n}{j}$, $2 \leqslant j \leqslant J$. Nowadays, with the advancement of technology, massive amounts of data are often observed at a large number of spatial locations in geophysical and environmental sciences. For example, the weather station monitoring network provided by the Institute for Mathematics Applied to Geosciences has $n = 11,918$ stations for the coterminous United States reporting monthly temperature and precipitation. If the number of curves in the sample of interest is large, the computational burden limits the applicability of the band depth,

(wileyonlinelibrary.com) DOI: 10.1002/sta4.8

especially when iterative procedures are involved in the analysis, for example such as the bootstrap in nonparametric testing for functional data (Li & Liu, 2004), the clustering and classification methods (Jornsten, 2004) and the functional median polish for functional ANOVA (Sun & Genton, 2012b). Therefore, from a practical perspective, it is important to devise fast and efficient computations of data depth. López-Pintado & Jornsten (2007) have proposed a resampling method to speed up the band depth computation which only provides an approximate solution. In this paper, we instead propose an exact fast method to alleviate the computational burden for $J = 2$.

The rest of our paper is organized as follows. The proposed exact fast method is described in Section 2. Section 3 provides the comparisons between our exact fast method and the approximate resampling method through simulation studies. A discussion is given in Section 4 and pseudo code is provided in the Appendix.

2 An exact fast method

The computational cost of calculating the band depth depends on the sample size n and j, the number of curves determining a band ($2 \leqslant j \leqslant J$). Since the order of curves induced by band depth is very stable in J, a small value of J is preferable to avoid computational issues. However, for large datasets, the computation is still very intensive even for $J = 2$. To solve this problem, a resampling method has been proposed by López-Pintado & Jornsten (2007), which computes the band depth in the following way:

S1. Randomly assign n sample curves into K blocks $B_1, \dots, B_K$, with each block of size $\sim n/K$.
S2. Compute depth values of each curve $y(t)$ with respect to each group B_i, denoted as $D(y|B_i)$, $i = 1, \dots, K$.
S3. The depth value of a curve $y(t)$ is $D(y) = \frac{1}{K}\sum_{k=1}^{K} D(y|B_k)$.

Suppose $n/K = m$. The computational cost is at rate $K \times \binom{m}{2}$, which is smaller than $\binom{n}{2}$ for large n and $K > 1$. Computational gains are then achieved. However, this is an approximate method that saves the computational time by sacrificing the accuracy, so a trade-off between the accuracy (block size) and the computational time is important.

To find an exact solution, we propose a fast method to compute the BD and MBD for $J = 2$, where the band depth in (2) becomes

$$BD_n^{(2)}(y) = \binom{n}{2}^{-1} \sum_{1 \leqslant i_1 < i_2 \leqslant n} I\left\{G(y) \subseteq B\left(y_{i_1}, y_{i_2}\right)\right\}, \tag{5}$$

and the modified band depth in (4) becomes

$$MBD_n^{(2)}(y) = \binom{n}{2}^{-1} \sum_{1 \leqslant i_1 < i_2 \leqslant n} \lambda_r \left\{A\left(y; y_{i_1}, y_{i_2}\right)\right\}. \tag{6}$$

To illustrate the exact fast method procedure, we consider an example (Figure 1) similar to the one in Sun & Genton (2011) with $n = 4$ curves on how to compute BD and MBD in practice. When $J = 2$, there are 6 possible bands delimited by 2 curves. For instance, the grey area in Figure 1 is the band delimited by $y_1(t)$ and $y_3(t)$ which completely contains the curve $y_2(t)$, but only partly contains $y_4(t)$. Moreover, a curve is also defined as "contained in a band" if it is on the border of the band. Then $BD(y_2) = 5/6 = 0.83$ since only the band delimited by $y_3(t)$ and $y_4(t)$ does not completely contain the curve $y_2(t)$ and $BD(y_4) = 3/6 = 0.5$ as it is only completely contained in the bands delimited by itself and another curve. Similarly, $BD(y_1) = 0.5$ and $BD(y_3) = 0.5$ can be computed.

From a programming perspective, there are two ways to implement this idea. One could use double loops to construct each possible band, then check which curves are contained in a certain band (López-Pintado & Jornsten, 2007; López-Pintado & Romo, 2009). In the first loop, the index i goes from 1 to $n - 1$, and in the second loop, the index j

(wileyonlinelibrary.com) DOI: 10.1002/sta4.8

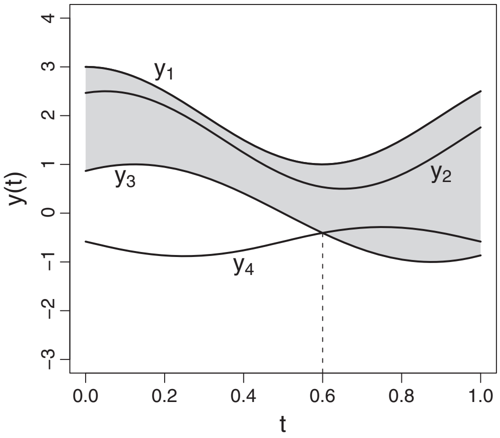

Figure 1. An example of BD and MBD computation: the grey area is the band delimited by $y_1(t)$ and $y_3(t)$. The curve $y_2(t)$ completely belongs to the band, but $y_4(t)$ only partly does.

goes from $i+1$ to n, with a total number of $\binom{n}{2}$ iterations. Then for the band delimited by the ith curve and the jth curve, all the n sample curves are checked to see which ones are contained in this band and which ones are not. The count number for each sample curve is accumulated respectively at each iteration. Let M be the $p \times n$ data matrix denoting n sample curves evaluated at p time points. The pseudo code is provided in Appendix A.1.

An alternative way that we propose is to check how many bands contain a certain sample curve without using loops based on a rank matrix. For the $p \times n$ data matrix M, we order the n data points for each row (time point) from the smallest value to the largest, and save their ranks $r_{ij} \in \{1, \ldots, n\}$ to the $p \times n$ rank matrix R. Then, it is clear that for the jth curve, $n_b = \min_{1 \leqslant i \leqslant p} r_{ij} - 1$ denotes the number of curves that are completely below the jth curve, and $n_a - \max_{1 \leqslant i \leqslant p} r_{ij}$ denotes the number of curves that are completely above the jth curve, $j = 1, \ldots, n$. This method has an absolute advantage over the former for large n and programming languages that are inefficient in loops since n_a and n_b can be computed by matrix operations. The pseudo code is provided in Appendix A.2.

In the example in Figure 1, $BD(y_2) = 5/6 = 0.83$ can be obtained by checking all the 6 bands and finding that 5 out of 6 completely contain $y_2(t)$ in the original method. For our exact fast method, the 5 bands are obtained by $n_a \times n_b + n - 1 = 1 \times 2 + 4 - 1 = 5$, where n_a is the number of curves completely above $y_2(t)$, n_b is the number of curves completely below $y_2(t)$ and $n-1$ is the number of bands with $y_2(t)$ on the border. When n is large, the exact fast method achieves remarkable computational gains.

To compute MBD, note that the curve $y_2(t)$ is always contained in the five bands, hence $MBD(y_2) = 0.83$, the same value as BD. In contrast, the curve $y_4(t)$ only belongs to the band in grey 40% of the time, thus $MBD(y_4) = (3 + 0.4 + 0.4)/6 = 0.63$ by definition. For the other two curves, $MBD(y_1) = 0.5$ and $MBD(y_3) = 0.7$. Similar to the computation of BD, to obtain the MBD of $y_4(t)$, for example, we only consider the combination of the curves above and below $y_4(t)$ for our exact fast method, but measuring the proportion of time that $y_4(t)$ is in the bands. The pseudo code is provided in Appendix A.3.

3 Simulations

In this section, we compare the performance of our exact fast method to that of the approximate resampling method for $J = 2$ in terms of computational time and accuracy. Intuitively, when the sample size n is fixed, the smaller each block is, the faster the computation. However, it is also necessary to keep the block size large enough in order to

(wileyonlinelibrary.com) DOI: 10.1002/sta4.8

Table I. The median curve index, averaged 50% central region range and elapsed time (in seconds) obtained by the exact fast method and the approximate resampling method for different values of K. Computations are done in R.

K n/K	1 1,000	5 200	10 100	20 50	50 20	100 10	Fast Exact Method
Median index	934	934	934	934	934	934	934
50% central region range	3.2763	3.2763	3.2763	3.2763	3.2466	3.2118	3.2763
Elapsed time (seconds)	1,189	239	120	54	20	10	0.02

ensure the accuracy of the band depth computation within each block. To illustrate this aspect as well as compare the two methods, we choose different values for the number K of blocks with $K = 1$ indicating the original exact band depth computation, and compare the median curve index and the 50% central region in the functional boxplot defined by Sun & Genton (2011).

The original sample curves are generated from an outlier model as in Sun & Genton (2011). The model is: $Y_i(t) = X_i(t) + c_i \sigma_i s$, where c_i is 1 with probability 0.1 and 0 with probability 0.9, $s = 6$ is a contamination size constant and σ_i is a sequence of random variables independent of c_i taking values 1 and -1 with probability 1/2. Here $X_i(t) = g(t) + e_i(t)$, $i = 1, \ldots, n$, with mean $g(t) = 4t$, $t \in [0, 1]$ and where $e_i(t)$ is a Gaussian stochastic process with zero mean and exponential covariance function $\gamma(s, t) = \exp\{-|t - s|\}$.

We generate $n = 1,000$ curves at $p = 50$ time points. The depth values are computed by MBD and all the computations are done by a single threaded application in R (R Development Core Team, 2012) on a 2.80Ghz Intel Xeon X5560 with 48GB of RAM. The results are summarized in Table I where the 50% central region range is an averaged value over all the time points.

Table I shows the performance of two methods compared to the original band depth computation, i.e. the case with $K = 1$. For the approximate resampling method, when the number of blocks is relatively small or reasonably large, for example, $K = 5, 10, 20$, the results are accurate with respect to the correct median curve and the averaged 50% central region range, but have less computational time. For instance, the computational time is reduced from 1,189 to 54 seconds when $K = 20$. The computational time can be further reduced when $K = 50, 100$, but with a loss of accuracy in the 50% central region range due to the small block sizes. For the exact fast method, it not surprisingly achieves the exact solution yet with a much shorter computational time, only 0.02 seconds. Such great computational gains will be even more prominent when the sample size n becomes larger.

4 Discussion

This paper proposed an exact fast method to speed up the computation of the band depth, a measure to order functional data. It greatly alleviates the computational burden and makes a wider application of the BD and MBD for large functional datasets possible. Compared to the approximate resampling method proposed by López-Pintado & Jornsten (2007), the exact fast method provides exact solutions for $J = 2$ and computes the depth values much faster as shown in Section 3, where it only costs 0.02 seconds in R for $n = 1,000$. Code in R and Matlab is available in the online supplement.

The exact fast method can also handle much larger datasets. For example, under the same simulation setting, we get the depth values in 0.2 seconds in R and 0.1 seconds in Matlab for $n = 10,000$ which is already impossible for the original computation to handle. For $n = 100,000$, it takes 1.6 seconds in R and 1.1 seconds in Matlab, whereas for $n = 1,000,000$, it takes 24.4 seconds in R and 12.4 seconds in Matlab. If n gets even larger, we can consider to

(wileyonlinelibrary.com) DOI: 10.1002/sta4.8

sacrifice some of the accuracy and combine the exact fast method with the approximate resampling method to further speed up the band depth computation.

Let $B^{(j)}$ denote a band delimited by j curves. The exact fast method we described is for $J = 2$, only considering bands $B^{(2)}$ in (5) and (6). For $J > 2$, $J = 3$ for instance, the method can be modified in a similar way by considering bands $B^{(3)}$ instead. If a band $B^{(2)}$ contains the ith sample curve, it is straightforward that the new band $B^{(3)}$ consisting of the band $B^{(2)}$ and one of any other curves also contains the ith sample curve. However, if a band $B^{(3)}$ contains the ith sample curve, the three curves do not necessarily include at least one completely above and one completely below the ith curve. Therefore, more special cases need to be considered. Since the order of curves induced by band depth is very stable in J, we only focused on $J = 2$ in this paper. However, further computational savings may be obtained by making use of the theory of partial ordering and directed acyclic graphs (Bang-Jensen & Gutin, 2008).

For spatio-temporal data, we have viewed the information as a temporal curve at each spatial location. An alternative would be to treat the dataset as a spatial surface at each time point. Sun & Genton (2011) proposed surface boxplots by defining a volume-based surface band depth for a surface S by counting the proportion of surface bands determined by J different surfaces ($2 \leqslant J \leqslant n$) in $\mathbb{R}^3$, containing S. Under this setting, our exact fast method is particularly useful for the spatial surfaces or images ordering due to the high resolution in space and time.

Appendix

A.1. Pseudo code for original computation of band depth

```
"count", "check" and "depth" are vectors of length n;
count=0;
for i from 1 to n-1
  for j from i+1 to n
    check[k] = if M[,k] is between M[,i] and M[,j], for each k=1,...,n;
    count = count + check;
  end
end
depth=count/nchoose2;
```

A.2. Pseudo code for exact fast computation of band depth

```
"n.a", "n.b" and "depth" are vectors of length n;
R[i,] = rank of M[i,], for each i=1,...,p;
n.a[j] = n-max(R[,j]), for each j=1,...,n;
n.b[j] = min(R[,j])-1, for each j=1,...,n;
depth = (n.a*n.b+n-1)/nchoose2;
```

A.3. Pseudo code for exact fast computation of modified band depth

```
"n.a", "n.b" and "match" are p by n matrices; "depth" is a vector of length n;
R[i,] = rank of M[i,], for each i=1,...,p;
n.a = n-R;
n.b = R-1;
match = n.a*n.b;
proportion = sum(match[,j])/p, for each j=1,...,n;
depth = (proportion+n-1)/nchoose2;
```

(wileyonlinelibrary.com) DOI: 10.1002/sta4.8

Acknowledgement

This research was partially supported by NSF grants DMS-1007504, DMS-1106494, and by Award No. KUS-C1-016-04 made by King Abdullah University of Science and Technology (KAUST).

References

Bang-Jensen, J & Gutin, G (2008), *Digraphs: Theory, Algorithms and Applications*, *Springer*, London.

Jornsten, R (2004), 'Clustering and classification via the L_1 data depth', *Journal of Multivariate Analysis*, **90**, 67–89.

Li, J & Liu, R (2004), 'New nonparametric tests of multivariate locations and scales using data depth', *Statistical Science*, **19**, 686–696.

López-Pintado, S & Jornsten, R (2007), 'Functional analysis via extensions of the band depth', *IMS Lecture Notes-Monograph Series, IMS*, **54**, 103–120.

López-Pintado, S & Romo, J (2009), 'On the concept of depth for functional data', *Journal of the American Statistical Association*, **104**, 718–734.

R Development Core Team (2012), R: A language and environment for statistical computing, R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.

Ramsay, J, Hooker, G & Graves, S (2009), *Functional Data Analysis with R and MATLAB*, *Springer*, New York.

Sun, Y & Genton, MG (2011), 'Functional boxplots', *Journal of Computational and Graphical Statistics*, **20**, 313–334.

Sun, Y & Genton, MG (2012a), 'Adjusted functional boxplots for spatio-temporal data visualization and outlier detection', *Environmetrics*, **23**, 54–64.

Sun, Y & Genton, MG (2012b), 'Functional median polish', *Journal of Agricultural, Biological, and Environmental Statistics*, **17**, 354–376.

Zhao, X, Marron, JS & Wells, MT (2004), 'The functional data analysis view of longitudinal data', *Statistica Sinica*, **14**, 789–808.

Zuo, Y & Serfling, R (2000), 'General notions of statistical depth function', *The Annals of Statistics*, **28**, 461–482.