bdots methodology

Abstract

The Bootstrapped Differences of Timeseries (bdots) was first introduced by Oleson (and others) as a method for controlling type I error in a composite of serially correlated tests of differences between two time series curves in the context of eye tracking data. This methodology was originally implemented in R by Seedorff 2018. Here, we revist the underlying methodology and suggest a new approach to identifying regions of statistically significant differences between (time series? functional time series?) as well as improving our estimate of group distributions in the context of the visual world paradigm.

1 Introduction

I am really starting to think that this paper can be very short.

I don't think that addressing the case of paired tests needs to be in the main, perhaps supplemental materials.

Introductions are tedious. The main takeaway of the original bdots paper begins with the understanding that we start with densly sampled time series (in the form of eyetracking data) that is averaged across time to create something that appears to be a nonlinear curve. It is unclear what effect this averaging has on the presupposed autocorrelation of the corresponding curves.

Nonlinear curves are fit to this data (parametric or otherwise), and comparisons are made between different groups across timepoints, allegedly resulting in a series of correlated tests. This curve fitting process is meant to smooth out the data (I read that somewhere) and adjust for any idiosyncrasies that may arise. (⇐ "The smoothing is important as individual subjects' curves can be fairly noisy despite inherent reality, which suggets gradual change" oleson 2018)

Imposing a functional structure on this aggregated data, parametric or otherwise, raises questions about the autocorrelated assumptions. Instead, we argue that systematic differnences between functional forms borrow from tools in functional data analysis. This, along with a proposed change to the bootstrapping algorithm presented in Oleson, leads to a more accurate estimate of group level distributions of curves (maybe earlier discuss the idea of comparing between groups) and more power to detect statistically significant differences in said curves.

There are two major changes that we propose to improve the original BDOTS (three if you count the lowercase stylization, "bdots”). First, we amend a step in the original bootstrapping algorithm to better reflect the between-subject variability of a group, leading to the generation of more accurate confidence intervals with drastically better coverage. Second, and most drastically, we propose leveraging the assumed functional form of the data in deriving tests and controlling the FWER when detecting regions of statistically significant differences.

Introduction (Attempt 2) The bdots package was originally created to implement a novel statistical methodology using the bootstrapped differences of time series to detect time-specific differnces between temporal nonlinear (parametric) curves. Motivated by the context of the visual world paradigm (I can elaborate), the bdots method sought to rectify the problem of identifying differences in functions in time without a priori specification of a time window. In other words, the problem was not simply to detect *if* two time series differ, but specifically at what time points they differ.

In the context of the VWP, the idea was that individual subjects have a probability of fixating on a particular object that changes in time, say, p_{it}. Based on empirical data, a (parametric) nonlinear curve is then fit to observed data, resulting in both a parameter estimate, θ_i, as well as an estimate of the covariance based on the resulting hessian matrix, V_i.

Often, we are interested in comparing groups rather than individual subjects. In this case, each individual θ_i can be considered an observation from a particular group distribution. Ultimately, then, we are interested in creating an estimate of the mean temporal curve from this group, comparing it against the mean curve of another. Said a different way – each group of interest is understood to have a distribution of nonlinear curves, with an associated mean curve as well as a degree of variability. Distributions of these group curves are estimated with a bootstrapping algorithm. Ultimately, our goal is to determine specific regions in time in which these mean curves differ in a statistically significant way, based off their mean estimates and the amount of observed variability.

The proposed method by Oleson comes down to performing a t-test between the distributions, one at each of a sequence of time points within the domain of observations (originally at each of the sampled time points of the VWP, consiting of 4ms samples between times of 0ms and 2000ms).

OR you know, maybe we can just paraphrase directly from Oleson:

"Here we develop a testing method to detect specific points in time at which two time series significantly differ. Our approach handles this task in three steps. First, we fit nonlinear curves to each participant's data. Second, for between-group comparisons, we bootstrap those fitted functions to estimate the mean of the group curves and the variance associated with these estimators. For within-subject comparisons, the bootstrapping is accomplished to assess the variation of the subject-specific curves. Third, we use these estimated curves along with the associated bootstrap measures of accuracy to make statistical comparisons at each point in time. Finally, we apply a novel family-wise error correction that is sensitive to the autocorrelation in the data to ensure appropriately conservative inference."

We suggest two changes to this original statement. First, we propose an adjustment to the original bootstrapping algorithm that we show significantly improves coverage and better reflects the true distribution. This is necessary in estimating the amount of within-group variation for statistical comparisons at each point in time. Second, we take advantage of the functional form of the data to propose an alternative method for determining regions of difference with simple permutation testing, without the need of autocorrelation assumptions. This allows for a more powerful method while also reducing the number of assumptions made. We validate this method via simulation in which regions and magnitudes of differences between curves are known.

2 Proposed changes

More detail on the changes

2.1 Bootstrap

First, we begin with the original bootstrapping algorithm:

1. For each subject, fit the nonlinear function, speciying AR(1) autocorrelation structure for model errors. Assuing large sample normality, the sampling distribution of each estimator can be approximated by a noraml distribution with mean corresponding to the point estimate and standard deviation corresponding to the standard error

2. Using the approximate sampling distributions in (1.), randomly draw one bootstrap estimate for each of the model parameters on every subject

3. Once a bootstrap estimate has been collected for each parameter and for every subject, for each parameter, find the mean of the bootstrap esteiamtes across individuals

4. Use the mean estimates to determine the predicted population level curve, which provides the average population response at each time point

In other words, at each step b of the bootstrapping algorithm, we are asked to find

$$\theta_b = \frac{1}{n} \sum \hat{\theta}_{ib}$$

where

$$\hat{\theta}_{ib} \sim N(\theta_i, V_i)$$

Here, maybe, we can tie in the discussion of the many normal means problem, where $E(\hat{\theta}_i) = \theta_i$, but θ_i itself is drawn from a group distribution, (θ, Σ). This would be relevant in showing that

$$\overline{\theta}_b \sim N\left(\theta, \frac{1}{n^2} \sum V_i\right)$$

which is not a reflection of the true variance Σ.

At any rate, the only proposals we change here are

1. Fit nonlinear function without AR(1) assumption on model errors
2. At each bootstrap iteration, sample n subjects *with replacement*

This will make it such that the bootstrap estimates capture both within-subject and between-subject variability

2.2 Permutation

Here, I don't really want to go into all of the details of the autocorrelation argument as they're not entirely relevant. So I'll stick with this instead. The original bdots method used the old bootstrapping algorithm in order to make estimates of the group curve distributions, using the resulting means and variances at each time point to compute a t-statistic evalution the differences between the two curves. Nothing fancy here, it was simply

$$T(t) = \frac{\overline{p}_{1t} - \overline{p}_{2t}}{\sqrt{s_{1t}^2 + s_{2t}^2}}$$

Great. Now, the whole idea behind this methodology was that at subsequent time points, these t-statistics are going to be highly correlated, causing a bit of a pickle for controlling FWER. The assumptions made assume that the sequence of t statistics is correlated, withi

$$T_t = \rho T_t + \epsilon_t; \quad t = 1, \dots, N$$

where $\epsilon_t \sim N(0, (1-\rho^2))$, ρ being the autoregressive parameter estimated from the original step (1). This gave that

$$T_t | T_{t-1} \sim N$$

3 Methodology and Overview

It would seem as if it doesn't make much sense to try and mathematically argue why moving to an FDA domain would be better than the present argument for autocorrelation, outside of the fact that autocorrelation makes less sense once the data has been aggregated. Since bdots assumes a functional form (and indeed used a four-parameter logistic when originally arguing for this), we will again do so, albeit more systematically the the original. The "proof", then, will lie in the outcome of the simulations.

To that end, there are two proposed changes that we wish to make. First, we offer an adjustment to the bootstrapping algorithm that is able to better

To that end, there are two proposed changes that we wish to make. First, we offer and adjustment to the bootstrapping algorithm that is a little awkward to bring up because it really is just turning an existing algorithm that doesn't actually bootstrap into one that does. To do this, we will begin with a set of empirical eyetracking data collected (from bob), fit these curves using bdots, and use the collection of parameter estimates from the four parameter logistic to estimate the mean and covariance matrix for a group-level normal distribution. We can call this mean the true mean, θ, and the curve derived from these parameters f_θ.

We will sample parameters from this distribution to create n additional subjects with binomial noise (to simulate fixations based on the current time/probability). These subjects will again be fit in bdots, and a comparison of the two algorithms will be made. What we plan to show is that ours has significantly better coverage in terms of both the bootstrapped parameters, as well as curve coverage.

Next, we argue that leveraging the assumption of a functional form to use permutation testing is superior than the current implementation assuming autocorrelation. To this end, we will use a two-parameter piecewise linear function, specifying the breakpoint and the slope, while the competitor/alternative curve will be constant. We will compare the permutation and autocorrelated approaches in each case to determine type I error rate and power.

4 Simulations

Simulations we ran to generally test two comparisons. First, we examined the differences between the original bdots algorithm with a modified form. In each case, we build a bootstrapped distribution and examine 90% quantiles to determine coverage of the true generating curve. Next, we compare bdots (with updated algorithm, since its better) with a permutation style test. Here, we are examining the power in detecting differences between the modified alpha method for controlling fwer and permtutations. I haven't done that simulation yet.

Each of these simulations was performed with 25 simulated participants in the visual world paradigm, each with a total of 10, 25, 50, 75, and 100 trials. Simulations were performed 100 times.

4.1 Setup and Data Generation

The creation of the simulated data was the same for each simulation, so it will be described in detail here. As we are simulating data for the visual world paradigm, we started with data collected empircally from (source dataset, the one in bdots). Twenty-eight experiment normal hearing participants were fit using the `bdotsFit` function to the four-parameter logistic curve (using looks to the Target). Fitted parameters for these subjects were extracted and use to construct an empirical group mean and empirical covariance matrix. These values were used as the normal parameters for our generating curve. That is, the empirical mean of these parameters, as well as the four parameter logistic function fit from this mean, represent our oracle curve. (I can report these values, I guess, if I rerun the simulation, but I don't have seeds in bdotsFit so this won't be practical).

For each of the 25 simulated subjects, we did the following:

1. First draw a set of parameters from $\theta_i \sim N(\theta, V_\theta)$, constrained for the baseline parameter, $b \geq 0$ and

for the peak parameter, $h \leq 1$

2. The drawn parameters were then used to create a subject specific generating curve f_i (f_{θ_i})
3. This is originally where I did the binomials to simulate looks based on probability at each time point. Patrick suggested normal distribution because he's unimaginative. That's fine, and I can redo it. I can also redo it with more than 100 simulations. At any rate, this is the step where we add noise.
4. THAT IS, for m trials, at each time point for each subject we drew $y_{it} \sim Bin(f_i(t), m)$ or rather the expectation of this

The final result was a data.frame with 25 subjects, each subject having 501 observed time points (12525x3 data.frame). This data.frame was used again in `bdotsFit` with the four-parameter logistic to parameter estimates for each of the subjects.

4.2 Simulations for Bootstrap

Our primary question here is if the resulting bootstrapped distribution appropriately covers the true generating curve, specified in the prior section. We compare the methodology used in the original bdots package, which we call bdots, with an updated methodology descriped in the first section, which we are calling bootstrap (granted, they are both bootstrapping algorithms). Simulations for each method are described in the following sections

4.2.1 bdots method

am i going into too much detail here? is this all necessary? do i need separate sections to detail this? because its pretty small

For the original bdots method, we simply took the object returned from `bdotsFit` and used it in the bootstrapping function, `bdotsBoot` with $B = 1000$ bootstraps. This function returns several things. First, we return a matrix of each of the bootstrapped mean parameters, in this case a matrix of size 1000×4. The function additionally returns a logistic function for each bootstrapped set of parameters evaluated at each time point, here a 1000×501 matrix, which is used

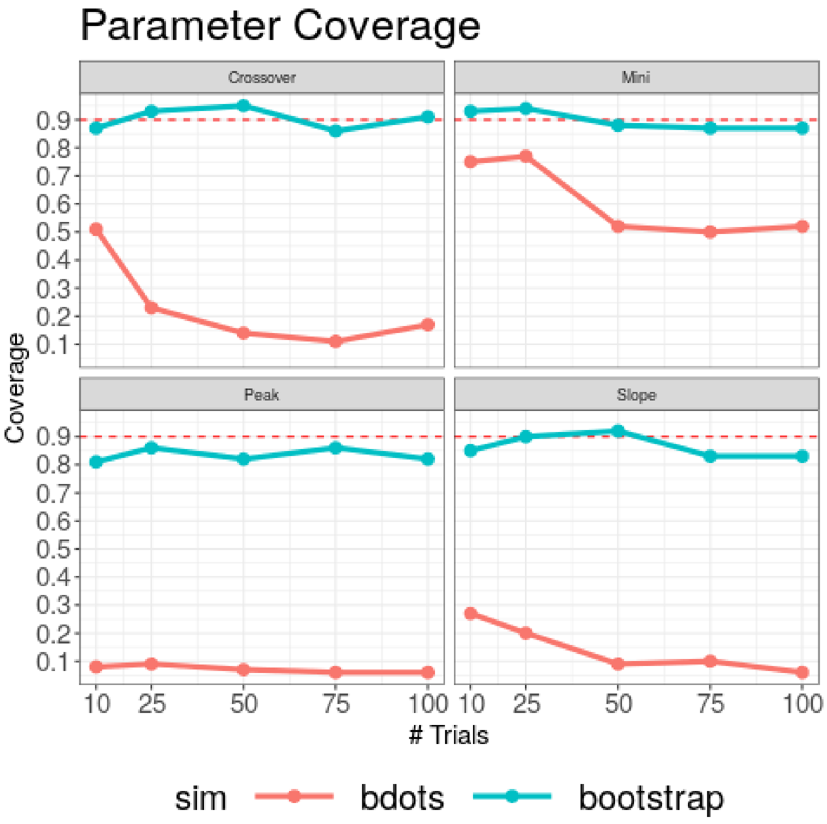

Figure 1: Coverage of generating parameter values

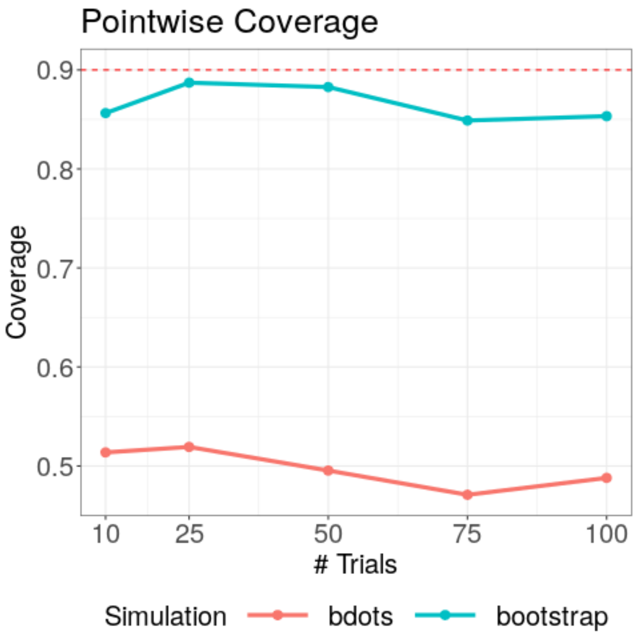

Figure 2: Coverage of generating parameter values

4.2.2 bootstrap method

4.2.3 Results

4.3 Simulations for Region Testing

Well, realizing now the writeup for this will be different since we are using different curve. We are going to write simulation so that we can explore a variety of conditions. It's going to be a piecewise linear function with a crossover point, and they will differ in the slope of the second part

piecewise function makes the most sense here. maybe also the double humped rectangles?

5 Discussion

Wow this was neat

6 Conclusion

Life is a distrupting burst of consciousness, bookended by eternal blackness and despair. Why are we even here? Why do we try at all?