

bdots methodology

Abstract

Gutting this entire thing and starting from scrayatch

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 revisit the underlying methodology and suggest a new approach to identifying regions of statistically significant differences between time series, or more generally, functions in time, as well as improving our estimate of group distributions in the context of the visual world paradigm. We also investigate the relationship between power, sample density and something else.

1 Introduction

2 Data Generation

In exploring the number of alternative methodologies, we have sought to construct simulated data under a number of differing assumptions.

2.1 Mean Structure

There are primarily two situations we consider here regarding the subject observations and their relation to the group.

The mean structure in simulating subject data will have profound implications for the observed variability.

2.1.1 Single Means

The first situation, which we call the “single mean” assumption (we can change this name its just in contrast to many means) goes like this: for a particular group of subjects, there is a single set of parameters, θ from

which all subject specific observations are derived. For the generating function f , we have that the subject specific observations are

$$y_{it} = f_{\theta}(t) + \epsilon_{it} \quad (1)$$

where the error structure may be either independent and normally distributed or maintain an autocorrelated structure. We will go into more detail on the error term later.

Under this assumption, all subjects have the same underlying parametric function with the only deviation between these subjects being their subject specific error terms. Under this assumption, we fit a nonlinear model to the data returning a parameter estimate and covariance matrix. It must be the case that we assume that the only relevant variability here is the subject specific variability, captured by the covariance matrix. That is, the difference between parameter mean estimates between subjects is of no interest (I actually don't even think here that is true, but that is essentially what is assumed in the Oleson bootstrapping method).

2.1.2 Many Means

The second situation is the many means assumption. Similar to the single means assumption, we assume that a group of subjects has a group mean, θ . In addition, however, we also assume variability, V in this group. This manifests itself by having subject specific parameters $\theta_i \sim N(\theta, \Sigma)$. From here, we assume that subject specific observations have the following structure:

$$y_{it} = f_{\theta_i}(t) + \epsilon_{it} \quad (2)$$

Similar to Equation 1, each individual subject has a mean structure around the generating function with an associated error; the difference here is that each subject's mean structure is dependent on their own subject-specific parameters. As such, not only must we estimate variability in our bootstrap based on the subject's estimated covariance matrix to capture within-subject variability, as was done in the single means method, we must also capture the *between* subject variability, an estimate of Σ , by bootstrapping the subjects themselves with replacement

2.2 Error Structure

We assume an error structure in the generation of our data. We will construct data with AR(1) correlation and without

2.2.1 Autocorrelated errors

In the case of autocorrelated errors, we assume

$$e_{it} \sim \rho e_{i,t-1} + w_{it} \quad (3)$$

where ρ is the autocorrelation parameter and

$$w_{it} \sim N(0, \sigma^2). \quad (4)$$

2.2.2 Independent errors

In the independent error case, we assume that the errors are independent and normally distributed

$$e_{it} \sim N(0, \sigma^2) \quad (5)$$

Not sure if this is worth discussing here or in some section detailing the actual construction of these estimates – what is actually done is we for a subject with N trials at each time point we have

$$y_{it} \sim \text{Bin}(N, f_{\theta_t}(t)) \quad (6)$$

This has the nice benefit of creating normally distributed errors with variances modulated by the number of trials.

3 Tables

This will go elsewhere, but for now we include it here just to be able to see it in one place. There are two sets of sets of simulations, each using 25 subjects, 100 trials in each condition. Consider the difference in the generating curves for each:

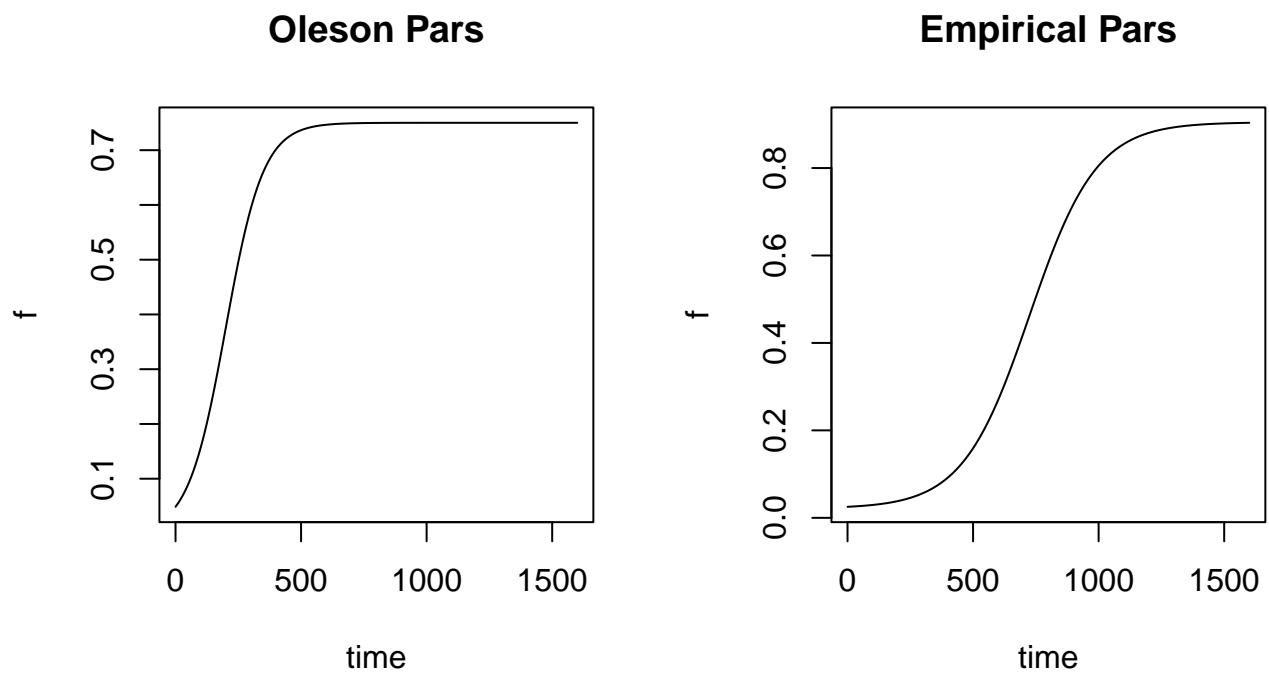


Figure 1: The generating curve used in finding TIE in Oleson 2017 against the one using empirically computed parameters

3.1 Using Oleson 2017 Parameters

In evaluating the logistic function in the original bdots paper, they examined a time period from 0 to 1600, sampled at 4ms intervals. The logistic curve they used had parameters: baseline = 0, peak = 0.75, slope = 0.0025, and crossover point = 200

First up we have the type I error rate

manymeans	ar1	bdotscor	Bad Bootstrap	Good Bootstrap	Permutation
FALSE	TRUE	TRUE	0.06	0.01	0.03
FALSE	TRUE	FALSE	0.76	0.01	0.01
FALSE	FALSE	TRUE	0.06	0.01	0.05
FALSE	FALSE	FALSE	0.05	0.02	0.11
TRUE	TRUE	TRUE	0.97	0.04	0.02
TRUE	TRUE	FALSE	1.00	0.03	0.00
TRUE	FALSE	TRUE	0.99	0.02	0.01
TRUE	FALSE	FALSE	1.00	0.02	0.01

Table 1: type 1 error rate using Oleson parameters

Then we have median per comparison error rate. So looking at error rate of each time slice

manymeans	ar1	bdotscor	Bad Bootstrap	Good Bootstrap	Permutation
FALSE	TRUE	TRUE	0.01	0.01	0.02
FALSE	TRUE	FALSE	0.32	0.00	0.00
FALSE	FALSE	TRUE	0.01	0.00	0.00
FALSE	FALSE	FALSE	0.01	0.00	0.02
TRUE	TRUE	TRUE	0.82	0.02	0.00
TRUE	TRUE	FALSE	0.92	0.02	0.00
TRUE	FALSE	TRUE	0.88	0.01	0.00
TRUE	FALSE	FALSE	0.91	0.00	0.00

Table 2: median per comparison error rate (why is this something considered?)

3.2 Using Empirical Starting parameters

Here we used parameters from Normal hearing subjects from one of Bob's papers. I should find out exactly which. But in this case, we have mean = 0.21, peak = 0.90, slope = 0.0016, and crossover point = 725. What is especially relevant here is the EXTREME difference in crossover points. Having it previously at 200? It will asymptote way before 1600 (the end point considered) giving far fewer places for any actual differences to be observed. That's absurd.

Whatever. Here are the TIE, broken into paired case and unpaired case. Paired case at the end

manymeans	ar1	bdotscore	Bad Bootstrap	Good Bootstrap	Permutation
FALSE	TRUE	TRUE	0.06	0.01	0.21
FALSE	TRUE	FALSE	0.87	0.08	0.18
FALSE	FALSE	TRUE	0.08	0.00	0.14
FALSE	FALSE	FALSE	0.15	0.02	0.21
TRUE	TRUE	TRUE	0.92	0.03	0.03
TRUE	TRUE	FALSE	0.96	0.02	0.04
TRUE	FALSE	TRUE	0.99	0.05	0.01
TRUE	FALSE	FALSE	1.00	0.05	0.03

Table 3: TIE for realistic parameters (unpaired)

manymeans	ar1	bdotscore	Bad Bootstrap	Good Bootstrap	Permutation
FALSE	TRUE	TRUE	0.12	0.02	0.03
FALSE	TRUE	FALSE	0.86	0.08	0.03
FALSE	FALSE	TRUE	0.09	0.01	0.01
FALSE	FALSE	FALSE	0.14	0.01	0.03
TRUE	TRUE	TRUE	0.49	0.02	0.01
TRUE	TRUE	FALSE	0.94	0.03	0.02
TRUE	FALSE	TRUE	0.72	0.02	0.00
TRUE	FALSE	FALSE	0.74	0.04	0.00

Table 4: TIE for realistic parameters (paired)

Then we again look at the median per slice comparison, though this (again) doesn't seem like an important metric if the goal is comparing fwer

manymmeans	ar1	bdotscore	Bad Bootstrap	Good Bootstrap	Permutation
FALSE	TRUE	TRUE	0.01	0.00	0.04
FALSE	TRUE	FALSE	0.31	0.00	0.04
FALSE	FALSE	TRUE	0.00	0.00	0.02
FALSE	FALSE	FALSE	0.00	0.00	0.03
TRUE	TRUE	TRUE	0.51	0.01	0.01
TRUE	TRUE	FALSE	0.76	0.01	0.00
TRUE	FALSE	TRUE	0.86	0.01	0.00
TRUE	FALSE	FALSE	0.81	0.01	0.00

Table 5: median per comparison error rate (unpaired)

manymmeans	ar1	bdotscore	Bad Bootstrap	Good Bootstrap	Permutation
FALSE	TRUE	TRUE	0.03	0.00	0.00
FALSE	TRUE	FALSE	0.26	0.00	0.00
FALSE	FALSE	TRUE	0.00	0.00	0.00
FALSE	FALSE	FALSE	0.01	0.00	0.00
TRUE	TRUE	TRUE	0.13	0.00	0.00
TRUE	TRUE	FALSE	0.52	0.02	0.00
TRUE	FALSE	TRUE	0.38	0.01	0.00
TRUE	FALSE	FALSE	0.44	0.01	0.00

Table 6: median per comparison error rate (paired)