Journal of Memory and Language 102 (2018) 55–67

Contents lists available at ScienceDirect

Journal of Memory and Language

journal homepage: www.elsevier.com/locate/jml

Detecting when timeseries differ: Using the Bootstrapped Differences of Timeseries (BDOTS) to analyze Visual World Paradigm data (and more)

Michael Seedorff[a,*], Jacob Oleson[a], Bob McMurray[b,c,d]

[a] Dept. of Biostatistics, University of Iowa, United States
[b] Dept. of Psychological and Brain Sciences, University of Iowa, United States
[c] Dept. of Communication Sciences and Disorders, University of Iowa, United States
[d] Dept. of Linguistics, University of Iowa, United States

ARTICLE INFO

Keywords:
Statistical methods
Timeseries analysis
Visual world paradigm
Family-wise error

ABSTRACT

In the last decades, major advances in the language sciences have been built on real-time measures of language and cognitive processing, measures like mouse-tracking, event related potentials and eye-tracking in the visual world paradigm. These measures yield densely sampled timeseries that can be highly revealing of the dynamics of cognitive processing. However, despite these methodological advances, existing statistical approaches for timeseries analyses have often lagged behind. Here, we present a new statistical approach, the Bootstrapped Differences of Timeseries (BDOTS), that can estimate the precise timewindow at which two timeseries differ. BDOTS makes minimal assumptions about the error distribution, uses a custom family-wise error correction, and can flexibly be adapted to a variety of applications. This manuscript presents the theoretical basis of this approach, describes implementational issues (in the associated R package), and illustrates this technique with an analysis of an existing dataset. Pitfalls and hazards are also discussed, along with suggestions for reporting in the literature.

Introduction

A fundamental problem in psycholinguistics is time. Language unfolds over time and the cognitive processing has its own dynamics. Consequently, our understanding of language is built in part on methods that assess cognitive processing as it unfolds over time. Such methods include Event Related Potentials or ERPs (Osterhout, McLaughlin, & Bersick, 1997; Swaab, Ledoux, Camblin, & Boudewyn, 2012), Magnetic Encephalography or MEG (Frye, Rezaie, & Papanicolaou, 2009; Schmidt & Roberts, 2009), eye-tracking in the Visual World Paradigm (Allopenna, Magnuson, & Tanenhaus, 1998; Tanenhaus, Spivey-Knowlton, Eberhard, & Sedivy, 1995) and mouse-tracking (Spivey, Grosjean, & Knoblich, 2005).

While the behavioral, methodological and theoretical grounding of such approaches has developed substantially, our ability to statistically analyze such time series has lagged. Standard statistical techniques are appropriate for detecting *that* there was a difference between two timeseries in a particular timewindow, and can compare other aspects of the timeseries such as the slope or asymptotes. However, it is much harder to identify with precision (and appropriate statistical certainty) that there is a difference in two timeseries *when the timewindow cannot be specified in advance* (much less isolate the time at which a difference occurs). This paper presents a new method designed to address this goal. We focus on the analysis of typical time series data from the Visual World Paradigm (VWP: Tanenhaus et al., 1995), building on recent analytic approaches developed for such data (Farris-Trimble & McMurray, 2013; McMurray, Samelson, Lee, & Tomblin, 2010; Scheepers, Keller, & Lapata, 2008). However, this approach will likely be useful with any sorts of timeseries data in which the timeseries can be captured as some parameterized function.

A formal presentation of this approach can be found in Oleson, Cavanaugh, McMurray, and Brown (2017). They demonstrate its efficacy with a retro-active analysis of a prior study (Farris-Trimble, McMurray, Cigrand, & Tomblin, 2014) and with a series of Monte-Carlo analyses that evaluated the likelihood of falsely detecting an effect (alpha) and of correctly detecting an effect (power). The present manuscript expands on this in several ways. First, we offer a conceptual overview of this method for the psycholinguistic community, with a special emphasis on statistical issues most prominent in psycholinguistics, and we discuss specific issues in using and reporting this technique, which were not addressed in this prior study. Additionally, and most importantly, we present a newly developed R package—BDOTS—that implements the technique and adds a number of features (both in terms of usability, and in terms of the underlying

[*] Corresponding author at: 145 N. Riverside Dr., Iowa City, IA 52242, United States.
E-mail address: Michael-seedorff@uiowa.edu (M. Seedorff).

https://doi.org/10.1016/j.jml.2018.05.004
Received 1 August 2016; Received in revised form 7 May 2018
Available online 25 May 2018

analytic technique) that had not been developed in the earlier statistical/theoretical work.

We start by briefly describing the VWP and the type of data it usually generates. We then describe existing techniques for such data. Next, we present our own approach, starting with a conceptual overview, and documenting how to use this method step-by-step to illustrate a number of sophisticated features in the R package. Finally, we end with a discussion of the innovations and limitations of this method and with guidelines for appropriate use and reporting.

The Visual World Paradigm

In typical instantiations of the Visual World Paradigm (VWP), participants hear spoken instruction to manipulate or select one of several visual objects either on a computer screen or in the real world (see, Salverda, Brown, & Tanenhaus, 2011, for a review). Objects represent possible competing interpretations of the auditory signal that will be briefly considered before being ruled out. To complete the task, participants must typically fixate the objects (e.g., they need to know where the object is before they can click on it) and, consequently, their fixations to each object at any given point in time—fixations which are typically initiated before the overt response—indicate something about how strongly they are considering that interpretation.

In classic versions of this paradigm used to study single word recognition (Allopenna et al., 1998; McMurray et al., 2010), participants might hear a word like *lizard* while viewing a screen that contains a *liver* (a cohort, which overlaps with the target word at onset), a *wizard* (a rhyme, which overlaps at offset) and a *necklace* (which is phonologically unrelated; see Fig. 1A). Here fixations to each object can be considered an estimate of how strongly that class of items is being considered (how active it is). Fig. 1B shows a typical pattern of fixations. Within about 200 milliseconds, participants fixate both the target and cohort. This delay corresponds to the amount of time it takes to plan and launch an eye-movement (Viviani, 1989); consequently, these early fixations are driven by only the earliest portion of the stimulus (*li-*) and are directed to both objects. Several hundred milliseconds later, more of the word is heard, and the target starts to diverge from the cohort. However, as the complete word unfolds, emerging overlap with the rhyme object may lead to some partial fixations. By the end of the timecourse, typically only one object (the target) is being considered.

These timeseries are typically constructed by averaging within timeslices, across trials. By averaging across many trials, within small time slices, we can compute an estimate of how the probability of fixating a given object gradually changes over time. While such data are popularly termed "proportions of fixations" (or some variant), they are really the proportion of *trials* on which the participant is fixating an object at a specific time.

There are a variety of approaches for examining such timeseries statistically. But before we discuss them it is important to describe a number of key features of the underlying data—many of which are glossed over by existing approaches. First, underlying these densely sampled timeseries is an auto-correlational structure. People cannot move their eyes every 4 msec (a typical sampling window for many eyetrackers) – fixations usually last 200–300 msec. The smooth gradual changes in the function derive from averaging many series of ballistic saccades, and that means that adjacent timewindows are almost always in part the product of the same physiological events.

Second, as is typical in most psycholinguistic paradigms, these functions are a product of sampling from multiple random factors (Baayen, Davidson, & Bates, 2008; Clark, 1973) – typically subject and item (words), but often other things like talker. Finally, this data is probabilistic, but it does not derive from a binomial distribution. Rather at a fundamental level it is a multinomially distributed (since there are typically more than two options on the screen).

As we describe shortly most existing analytic approaches for VWP data ignore one or more of these factors in the interest of tractability, and the proposed approach does not overcome all of them. Nonetheless it is important to raise these issues both as a part of evaluating the existing approaches, and in developing and understanding our proposed approach.

Existing Analytic Approaches for VWP and other timeseries data

The present project developed a statistical tool to (1) detect differences in two timeseries (as in the VWP) when the timewindow is unknown in advance and (2) to offer a precise characterization of the timewindow in which a difference occurs. We now review the existing techniques for analysis with an eye toward their ability to accomplish these goals.

Area under the curve

The earliest and still most widely used method of analysis for VWP data is the area under the curve (AUC) approach. Here, the dynamics of the timeseries are typically discarded, and in each condition, the

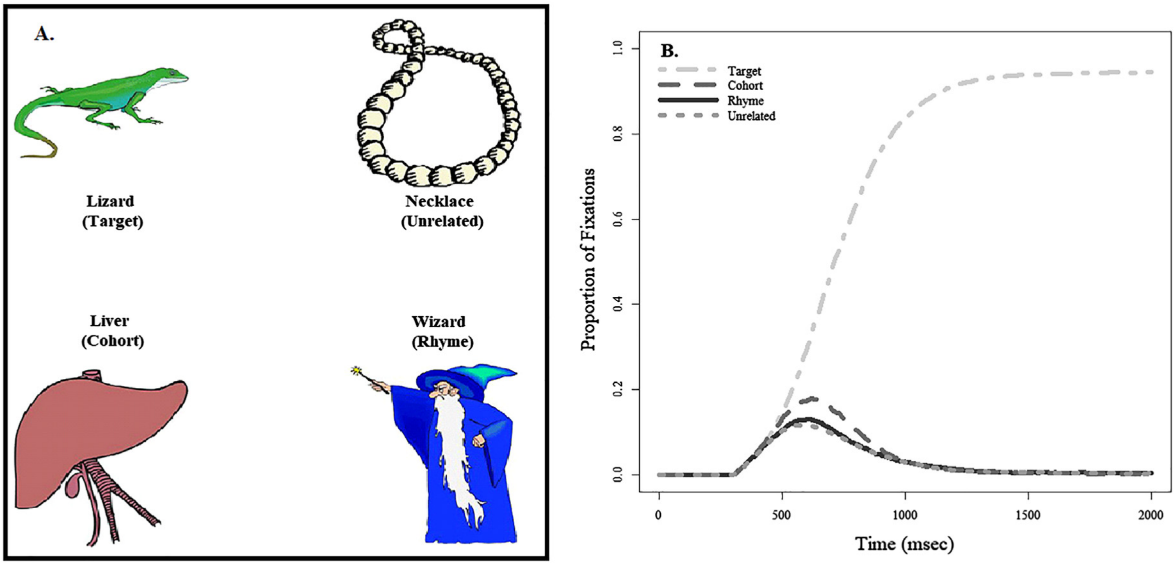

Fig. 1. (A) Sample display showing those same four competitors. Note the text is not present on the screen. (B) Looks to the four visual referents (target, cohort, rhyme, unrelated) as a function of time in a standard version of the VWP.

experimenter computes the average proportion of fixations within some fixed timewindow. The timewindow is arbitrary, but researchers often choose windows based on properties of the stimulus, prior experiments, or the overall shape of the function (e.g., when it reaches an asymptote). Nonetheless, the justification of the specific timewindow (on theoretical or empirical grounds) is an important methodological issue raised by this technique. The concern is that this offers too many researcher degrees of freedom, leading to spurious significant effects (see Bakker, van Dijk, & Wicherts, 2012; Simmons, Nelson, & Simonsohn, 2011, for broader discussion of the issue of researcher degrees of freedom). However, when the time-window is well justified and specified in advance, AUC is still useful. It is straightforward and has the added bonus of being able to account for multiple random effects using separate item and subject analyses, or with mixed models including both effects.

AUC has little to offer with respect to the problem posed here. As this approach require the timewindow to be specified in advance, a timewindow that is not of specific interest a priori cannot be analyzed. Moreover, it is not advisable to use multiple timewindows (though it is occasionally used for lack of better options, e.g., Watson, Tanenhaus, & Gunlogson, 2008), as the likely possibility that one or more fixations span both windows means that the two windows are not mathematically independent (though see, McMurray, Tanenhaus, Aslin, & Spivey, 2003). Thus, AUC may accurately detect *that* there was an effect, but it cannot do so unless the timewindow of interest can be defined in advance. Moreover, it has few options for detecting when an effect occurs (within that timewindow).

Saccade/fixation components

A second, and less common, option is to identify specific components of the fixation record for analysis. For example, the duration of a particular fixation (McMurray, Aslin, Tanenhaus, Spivey, & Subik, 2008), the latency to look at an object (McMurray, Tanenhaus, & Aslin, 2009), or the raw number of fixations (Blumenfeld & Marian, 2007; Ju & Luce, 2004). This overcomes some of the issues of using averaging over discrete trials that come up with AUC metrics; and such measures may show a more Gaussian distribution, making them more appropriate for linear models. However, they are also highly selective and focused, and therefore can have insufficient theoretical motivation for selecting particular measures. For example, increased activation for a competitor could have differential results both on the likelihood of fixating it in the first place, and the likelihood of staying longer once the participant is there. By emphasizing only one component, crucial effects may be missed; though if one were to examine all possible components, the converse problem of multiple comparisons and the more serious issue of p-hacking comes into play. A serious problem then is that currently there are no linking functions available for identifying such components raising the possibility of a fishing expedition. Perhaps more problematically, an underlying experimental difference may exert weak effects on both the likelihood of fixating and the likelihood of staying, but these effects are not robust enough to be seen individually, making it difficult to detect. In terms of our primary question, it is possible to develop such measures to detect effects at various times, since these measures generally respect the discrete series of events, and autocorrelation structure can be incorporated into such models.

Onset detection

A number of recent studies have investigated the time at which various properties of the stimulus affect the fixation record (McMurray, Clayards, Tanenhaus, & Aslin, 2008; Reinisch & Sjerps, 2013; Toscano & McMurray, 2012). In many ways, this complements the goals of the present project. In onset detection approaches, the primary concern is when an effect begins relative to the onset of some other effect (or that same effect in another condition). In contrast, we ask here whether there is an effect at all, and then seek to identify the timewindow over which it can be observed.

The analytic approaches that have been adapted to this problem largely assume that there is an effect (somewhere in the timecourse) and focus on detecting its onset. These approaches borrow from the ERP literature (Miller, Patterson, & Ulrich, 1998). Here, the dominant paradigm is to compute some measure of the effect of some condition at each point in time – for example for a condition with two levels, this could be a difference; for a multi-level condition, it could be a regression slope. The onset is then detected by setting an arbitrary threshold relative to the maximum (e.g., when the effect crosses 20% of its maximum) and then computing the time (for each subject) at which each effect crosses this threshold.

This procedure can easily be derailed if the timecourse of the effect is fluctuates over time, thus it requires a lot of data per subject per condition to achieve smooth timecourses. Consequently, as in the ERP literature, data are often jackknifed prior to onset detection. This means that the timecourse function is averaged across all participants except one, the onset is obtained, and then the process is repeated excluding the next subject (with appropriately more conservative test statistic). This is a flexible procedure that can be extended in various ways. For example, there are more robust ways of detecting the onset of effects (Mordkoff & Gianaros, 2000), and researchers can tie onset measurements to precise theoretical notions like immediacy and competition (Farris-Trimble et al., 2014).

In terms of the general concerns regarding analysis of VWP data, onset detection techniques have limitations and some strengths. They avoid the issue of multinomial responses, by collapsing the multinomial data into a continuous time to ask when effects occur. However, they presume that there is an effect at all whose onset latency can be measured, and researchers rarely compute omnibus-style tests first to detect an effect prior to computing its onset. When this is done at all, it is typically done with AUC approaches (with caveats described above). Even if the presence of an effect is known, this technique does not offer a unified approach for the question of when effects occur because it is limited, at this point, to onset detection. While it might be adaptable to offset detection as well, it is not clear how it could handle situations in which effects do not offset (e.g., they persist throughout the trial) or cases in which there are multiple onsets and offsets of an effect. Perhaps more importantly, it contains no way of evaluating the significance of the effect itself (in addition to difference in its timing).

Timeseries analysis

An increasingly popular approach to VWP data is to fit some typically nonlinear function of time to visualizations of the data as in Fig. 1. The parameters of the functions can then be used as descriptors of how the data change over time. These can be analyzed either with traditional sorts of models (e.g., GLM), or the functions can be integrated into mixed effects models. The first such approach (Mirman, Dixon, & Magnuson, 2008) proposed to use orthogonal polynomials as the basis of the analyses. These are highly advantageous because they can be fit with linear methods, and they can easily be integrated into mixed models which capture both subject- and item-effects. However, at the same time, polynomial functions do not offer a great parameterization of the typical shape of VWP data. For example, the looks to the target shown in Fig. 1A would require approximately 6–7 polynomial terms to capture, and the parameters themselves do not describe any meaningful aspect of the data. Thus, these functions offer license to say that there was an effect on the timecourse, but may not offer any more precise description. Perhaps more problematically, to overcome this, researchers often choose narrow timewindows where the function can be described with lower order polynomials (cubics, quadratics), but this of course introduces researcher degrees of freedom.

An alternative approach is to use nonlinear functions to capture more meaningful aspects of variation in these timeseries (Farris-Trimble & McMurray, 2013; McMurray, Clayards, et al., 2008; Scheepers et al., 2008). For example, target fixations can be fit by a four-parameter logistic function that captures the asymptotes, slope and cross-over point. These parameters offer more meaningful descriptors of the data, and the functions can often better approximate the data (than very high order polynomials). There are concerns about whether these parameters can be treated as independent (e.g., do slope estimates differ when the asymptotes are close together), but as we discuss below these can often be solved with different parameterizations of the function. These functions can typically be fit easily to individual participant data (see https://osf.io/4atgv/), and they can be jackknifed as well (Apfelbaum, Blumstein, & McMurray, 2011). However, they have not yet been integrated into a mixed effects framework which would enable simultaneous subject- and item-effects. Instead researchers typically estimate the parameters of the functions and analyze those with more traditional, GLM approaches. This ignores within-subject variability when computing the significance of effects. However, the lack of multi-level approaches with such functions is a failure of implementation, not a fundamental limitation.

Neither the polynomial nor nonlinear approach entirely deals with the probabilistic or multinomial nature of the data; however transformations like the empirical logistic could be applied prior to fitting the functions (which could improve the quality of polynomial fits), or to the relevant parameters after fitting (e.g., Farris-Trimble et al., 2014), and prior to analysis. It is also important to note that nonlinear functions that are good representations of subject averaged curves, may not necessarily be accurate representations of the true trial-by-trial response curves (Estes, 1956). Thus inference is required at the subject-averaged response level, rather than an individual trial response level. However, this is relevant to all analysis approaches of VWP data that base model fitting on the aggregate of a subject's trials. Interpretation at the trial level is possible when modeling individual trial responses (Vandeberg, Bouwmeester, Bocanegra, & Zwaan, 2013).

Both the polynomial and nonlinear approaches offer a reasonable way to characterize the timecourse of fixations to individual objects, and comparing the parameters of the function across conditions can permit researchers to make inferences about how they differ (e.g., two conditions peak at different heights, show different rates of growth, etc.). But they do not, in most cases, permit direct inferences about when conditions differ, and sometimes they can be ambiguous. For example, in Farris-Trimble et al. (2014), participants heard a target word while fixations to cohorts and rhymes were monitored. Here, there was a clear difference between 500 and 1000 msec (see their Fig. 1), but parametrically this would be described both as a difference in the slope (rate of change) and the crossover (the time at which the curve crosses 50%). In this case, it is possible that neither parameter by itself would be significant, or that both would be. However, this basic observation about *when* the difference may be significant cannot be captured.

One could compute confidence intervals around such functions to answer this question. However, systematic methods for doing this have not yet been developed, and it could be particularly difficult for nonlinear functions. This approach also has a large issue with significant family-wise error as it essentially amounts to repeated t-tests across time slices. The present project overcomes these limitations with new ways to compute confidence intervals for nonlinear functions, and a procedure for controlling the family-wise error.

Nonparametric approaches based on clustering

Cluster based permutation testing (Bullmore et al., 1999; Maris & Oostenveld, 2007) offers another approach that has become increasingly popular for the VWP. These clustering techniques are intended to directly address the question of detecting if two curves differ when the time is unknown. They derive from methods like EEG and MEG which also generate autocorrelated timeseries. However, unlike VWP data, such curves do not often follow a parametric function, and there may be additional dimensions of auto-correlation (e.g., among adjacent sensors in space).

In cluster based permutation testing. This the researcher computes a test-statistic at all time points, and groups adjacent (significant) tests into a single cluster. This controls family wise error by replacing individual tests with a single test for the cluster (the average of the test-statistics within the cluster). Test statistics are computed by bootstrapping individual curves and computationally calculating a permutation test statistic. This avoids making assumptions as to the underlying timecourse curve for an individual, or their statistical distribution (which is potentially useful for VWP data which are not Gaussian distributed). Such an approach is currently implemented in the eyetrackingR package, making it available to researchers using the VWP, and it has been used to analyze visual world data for several studies (Barr, Jackson, & Phillips, 2014; Oakes, Baumgartner, Barrett, Messenger, & Luck, 2013; Weighall, Henderson, Barr, Cairney, & Gaskell, 2017; Wu, Barr, Gann, & Keysar, 2013).

The non-parametric approach is particularly advantageous for brain imaging data, which is variable and does not follow an easily defined curve. But it may not be ideal for applications like the VWP, in terms of power to detect differences between various groups. Visual world data (at least as applied to single-word recognition) tend to follow one of two types of parametric curves (described below in function implementation details of BDOTS); consequently, a non-parametric approach may lose some statistical power in the scenario where the parametric curve is a good fit to the true, underlying function for all subjects. Additionally, because this approach performs a permutation test to calculate a test statistic, computation time can become an issue in cases where there are a large number of total trials. Moreover, this approach offers no way to simultaneously model subject and item level variance, though these could be modeled separately with an F1/F2 analysis. But perhaps more importantly, this approach offers only "weak" control of family-wise error. Rather than controlling for error in many tests, it reduces the number of tests. Consequently, as Maris and Oostenveld admit, it may have reduced power for detecting second and third clusters with smaller effects. Importantly, no Monte-Carlo analyses have been run to systematically evaluate either Type 1 (Family-Wise) error or power, so it is unclear if reducing the number of tests by clustering accomplishes this goal. In that light, the clustering procedure has a number of free-parameters and it is unknown what effects they may have on the statistical properties. That said, the present approach shares with cluster-based permutation techniques the insight that non-parametric approaches may be useful for estimating variance, and that family-wise error may be controlled by attending to the fact that tests that are adjacent in time (or space) are highly correlated and therefore not independent.

Overview of BDOTS

Our procedure is termed the Bootstrapped Differences of Timeseries (BDOTS). A complete mathematical treatment is provided in (Oleson et al., 2017), and as part of developing this report, we have created an R package of the same name that is now available for download on the CRAN server. This R package transformed BDOTs from the primarily theoretical approach reported in that earlier manuscript to something usable by the psycholinguistics community. As we detail here, the package implements several changes to the underlying statistical method designed to make it more powerful. The remainder of this paper starts by describing the conceptual underpinnings of BDOTS. We then walk through an implementation to illustrate the range of options. Finally, we discuss some limitations and directions for future development.

BDOTS starts with the simplest approach: a comparison between two conditions at each time point. However, there are several issues that arise when attempting to run many t-tests on raw data. Variability in the data makes it possible to encounter inconsistent results (e.g. two curves that are significantly different at every point from 500 to 1000 msec except for at 800 and 644 msec). This makes for a difficult interpretation of the results. More importantly, given the dense time sampling of eye-tracking and/or ERP data, repeated test statistics can inflate family-wise type I error.

Additionally, there are often other factors of interest that cannot be addressed as a test of differences at individual time points, such as *when* the two groups peak or the time it takes to reach peak. These can be accomplished with curve fitting approaches that directly estimate these properties (Farris-Trimble & McMurray, 2013; Farris-Trimble et al., 2014), but it would be useful if this could be accomplished within the same statistical framework.

BDOTS addresses these problems in four parts. First, we use participant specific fitted curves to capture the shape of the functions. This helps to smooth the data (minimizing idiosyncratic patterns of significance). It also offers a parametric description of the functions, and the standard errors of the parameters take into account where this description may not fully account for the data. While BDOTs borrows these strengths of a parametric approach, our analysis does not depend a great deal on the specific parameter estimates or even the specific functions. Consequently, the dangers of overfitting are reduced since we are not trying to draw statistical inference on the basis of individual fits.

Second, we use a bootstrapping procedure to estimate the standard error of the mean at each time point. Third, we use these standard errors of the function to conduct 2 sample t-tests at every time point. Fourth, we control for family-wise error with a modified Bonferroni corrected significance level which takes advantage of the inherent autocorrelation of the test statistics to avoid being overly conservative.

This combines many strengths from the prior approaches. We leverage curve fitting and timeseries approaches to obtain accurate descriptions of the function (and its variance). As in permutation tests, we use non-parametric statistics to capture the error distribution without inappropriately assuming a Gaussian distribution. And our approach to family-wise error takes the insight from the cluster based approaches that adjacent tests are not independent, though rather than collapsing them we derive an explicit ("strong") alpha correction.

Conceptual implementation

The first part of our method estimates fitted curves for each subject. The actual function of the fitted curves doesn't matter, as these curves are only used to bootstrap the population-averaged curves and are subsequently thrown out. Consequently, any function that is a reasonable fit to the data could be used (e.g. polynomials, logistics, etc.). The current R package implements two common non-linear functions used in psycholinguistics. Along with estimates for the function parameters, we obtain subject-specific estimates of the covariance matrix for the set of parameters, which captures within-subject variance of the parameter estimates (trial-by-trial variance should be reflected in the smoothness of the subject-averaged curve and will have an effect on these estimates of the variance for subject-specific parameters). Note that if a subject's curve is not reasonably fit by the defined function, the standard deviation estimates for the parameters have the potential to be extremely large. Thus, as with all non-linear curve fitting, it is important to visually compare fitted curves to actual data, and the BDOTS R package has tools for enabling this.

Second, once we have estimated parameter values and standard errors for each subject, we draw random samples for each subjects' curve by drawing random values for the vector of parameters (from a distribution with the mean vector and variance matrix specified by the estimates from the fitter). We then use the functional definition of the curve to acquire an observation estimate at each time point for the resampled participant, a form of parametric bootstrapping. At each iteration of the bootstrap (at each random sample), we average all the sampled individual curves into a population average for each group. This provide an estimate of the bootstrapped mean difference and standard error between the groups at each time point to perform t-tests.

Notably, the specific functions used to fit the curves are discarded after the bootstrap step – all that is retained is the estimated difference and its standard errors. This allows any function with an adequate fit to be used and one can potentially use different functions for different conditions or for different subjects. Superficially this may appear problematic, but here the goal is not to make statistical inferences on the basis of individual fits; but rather to accurately characterize each subject's data, so that they can be resampled for boot-strapping.

The use of fitted curves provides three critical advantages over bootstrapping raw data. First, the SEs of the parameters of the fitted curves are sensitive to within-subject variance, and this can be used as part of the bootstrap. Second, the curves offer a form of smoothing eliminating the issue of obtaining small regions of insignificance (or significance). Finally, variation in the parameters can be mapped to the relevant psychological dimensions. For example two conditions which target fixations are simply delayed by 20 msec will appear as a small difference in the cross-over parameter (on the x- or time-axis) as opposed to a large, but short lived difference, in the proportion of fixations (the y-axis).

Finally, after bootstrapping, we calculate t-statistics at every observed time point (or, potentially, unobserved time points along the observed timecourse). The large number of tests performed for a typical dense timeseries demands some sort of family-wise correction. As described cluster-based approaches bypass this issue by simply collapsing comparisons into clusters. However, this is a somewhat weak form of control for family-wise error where a stronger approach may be needed.

In a traditional Bonferroni correction, the alpha is modified by dividing it by a constant value (the number of comparisons, C). However, with so many comparisons this will be overly conservative. A typical VWP timeseries may sample to 1500–2000 msec at a 4 msec sampling window, resulting in 375–500 comparisons—yielding a small α^* which is likely too small to have much power. However, a traditional Bonferroni correction also fails to take into consideration the autocorrelation one would expect when the underlying data is a continuous timeseries (not a set of truly independent tests). That is, in a timeseries with 500 points, there are not really 500 *independent* comparisons, since adjacent points will be related. Consequently, the probability of a type 1 error is greater than α/N and a Bonferroni type correction is likely to be far more conservative than necessary. Therefore, by estimating the autocorrelation between test statistics, we can use this to estimate an adjusted α^* value that is less conservative but still maintains an overall family-wise error level at a specified α.

Oleson et al. (2017) derive an estimate of α^* that captures these intuitions and this is implemented in BDOTS. They demonstrate that as the autocorrelation between test statistics increases, α^* increases to near .05, while a decreasing autocorrelation reduces α^* to the Bonferroni correction. An alternative solution to this problem is to use coarser adjustments like False Discovery Rate (FDR; Benjamini & Hochberg, 1985). However, there are some caveats to this: FDR has a less conservative correction of the test statistic but achieves this by accepting a certain proportion of Type I errors in the series of tests. This makes interpretation more difficult, particularly at the edges of the regions of difference where the test statistic will be closer to the threshold for significance (and the likelihood of a Type I error increases). Moreover, it may be more difficult to justify this approach if another solution is available. Importantly, this alpha correction is entirely independent of the bootstrap and curve fitting procedure, and can conceivably be used with any timeseries of statistical comparisons (and it can be called as an independent function within the BDOTS package to enable such use).

In addition to testing for differences in the timeseries, the bootstrap iterations can also calculate the significance of group-wise differences in the parameters. During the same resampling, we estimate the overall mean and standard error estimates of each parameter for each group. Using these values, t-tests can be performed that are more powerful than a typical t-test using only the subject-specific parameter estimates because we are able to incorporate the subject-specific standard errors of the parameter estimates.

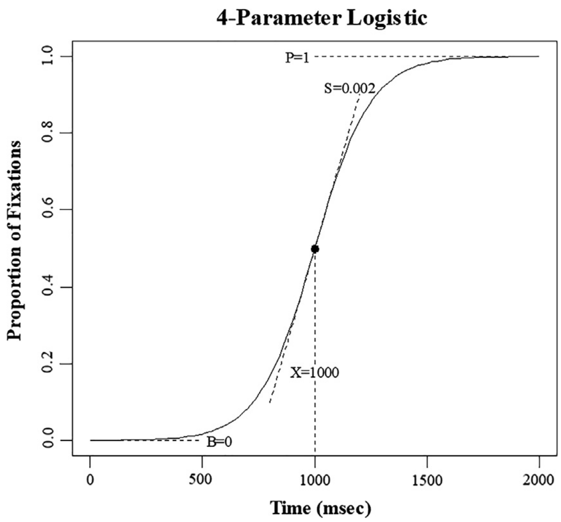

Fig. 2. A typical 4-parameter logistic with B = 0, P = 1, S = 0.002, and X = 1000.

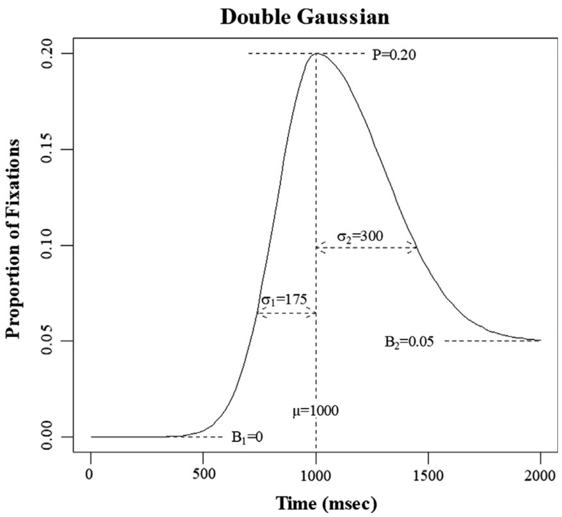

Fig. 3. A typical double Gaussian with $B_1 = 0$, $B_2 = 0.05$, $\sigma_1 = 175$, $\sigma_2 = 300$, $\mu = 1000$, and $P = 0.20$.

Implementation

We next describe BDOTS' implementation at two levels. First, as with any complex statistical tool experimenter decisions must be made, and several assumptions are made by the package. These details can matter greatly for outcomes (particularly with respect to non-linear curve fitting), and BDOTS users should aware of them. Second, along with these conceptual issues related to implementation we include R code detailing how to implement the package.

Step 1: Selecting a function

Currently BDOTS is implemented with two functions based on prior work (Farris-Trimble & McMurray, 2013; McMurray et al., 2010): a 4-parameter logistic and a double Gaussian. The 4-parameter logistic is a good approximation to an individual fixation curve for fixations to target objects where there is a constant period of low looks, followed by an "S" shaped increase in looks, and ending with a constant period of high looks (Fig. 2). We use the following parameterization of the logistic function where B_i describes the baseline for subject i; P_i describes the plateau value; X_i describes the crossover point; S_i describes the slope at the crossover point; and t defines time.

$$p_{it} = B_i + \frac{P_i - B_i}{1 + \exp\left(\frac{4S_i}{(P_i - B_i)}(X_i - t)\right)} \tag{1}$$

This particular parameterization has two advantages over traditional forms of the logistic. First, because the asymptotes are free parameters, it needs not asymptote at 0 and 1 as the more typical (two parameter) logistic functions do. Second, the slope S_i is divided by the difference between the asymptotes. Consequently, the slope reflects the derivative of the function at the midpoint *independent of the asymptotes*.

The double Gaussian (Eq. (2), Fig. 3) is a better approximation for looks to competitors. This curve begins with a period of low looks, followed by an increase in looks culminating at a peak, followed by a decrease and leveling off.

We use the following parameterization for the double Gaussian where μ describes the mean for each of the individual normal distributions; $\sigma_1{}^2$ describes the variance for the left-side normal distribution (essentially the onset slope); $\sigma_2{}^2$ describes the variance for the right-side normal distribution (the offset slope); P_i describes the peak, B_1 describes the baseline for the left-side normal distribution, and B_2 describes the baseline for the right-side Gaussian distribution.

$$p_{it} = \begin{cases} (P_i - B_{1i}) * \exp\left(\frac{(t-\mu_i)^2}{-2\sigma_{1i}^2}\right) + B_{1i} & if\ t \leqslant \mu_i \\ (P_i - B_{2i}) * \exp\left(\frac{(t-\mu_i)^2}{-2\sigma_{2i}^2}\right) + B_{2i} & if\ t > \mu_i \end{cases} \tag{2}$$

Difference of functions

One may also be interested in the amount of "drop-off" between two curves, as the relevant measure of interest. For example, researchers often use unrelated fixations as a baseline to control for differential levels of looking between groups (Fig. 4A). Consequently, the *difference* between cohort and unrelated fixations may be a useful estimate of cohort activity, and differences might further be compared between groups or conditions (Fig. 4B). BDOTS can handle this by fitting separate functions to the components of the difference curves, and then evaluating the difference curves between conditions.

Additional functions

Although there are only two functions (plus the difference curves) defined at the moment, it is possible to extend this methodology to any parametric function (we are currently working on polynomial and five-parameter logistic functions), and as open source software we encourage additional researchers to investigate new functions. The collection of viable functions may be limited, however, by the number of parameters that define it; a time series with high correlation and without a very large number of observations will have difficulty fitting a function with many parameters and results may be unstable.

Curve fitting and overparameterization

As with any non-linear curve fitting procedure one should be concerned about the potential for over-fitting the data. This is mitigated in a few ways in BDOTs. First, the two presented functions are generally appropriate fits to VWP data and have been used in many prior studies; the issue of selecting an overparameterized function that overfits specific individuals' responses is largely avoided by sticking to these more generic functions. Additionally, the amount of data collected in the VWP is large. Thus, the opportunity for overfitting with 4–6 parameter functions should be minimal (and reasonable polynomial curve fits would need even more parameters). Finally, ultimately inference is performed on the bootstrapped curves and the actual curve fits are discarded, so the specific function choice has little impact on the end results unless it is ill-fitting to individual subject curves. In this light, if functions are overfitting to specific subjects, the idiosyncratic properties of a given subject's data (that are likely to trigger overfitting) will

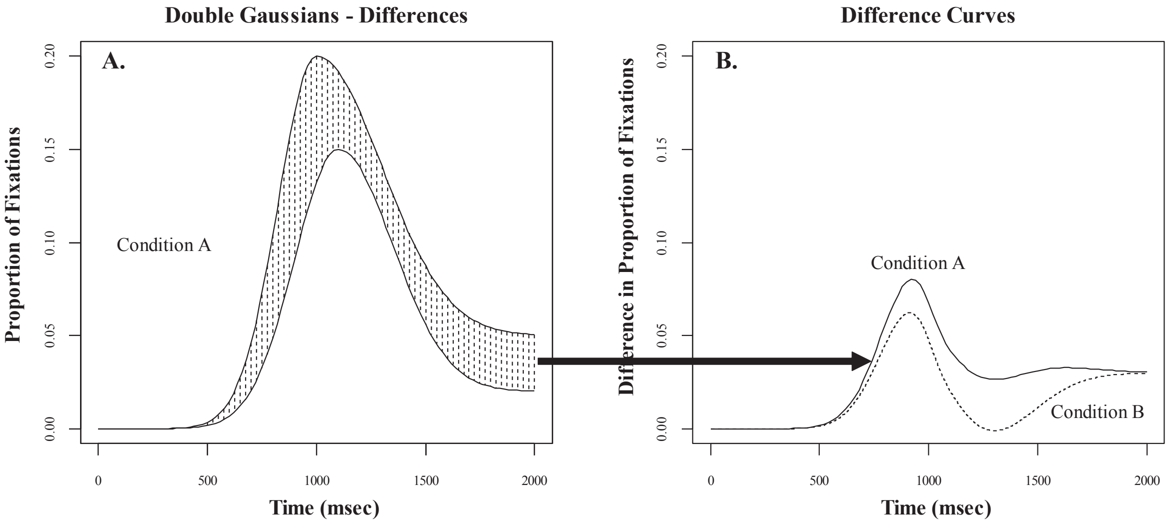

Fig. 4. (A) Two double Gaussian curves plotted against each other with the area of the differences highlighted. Each of these two curves is fit in the curve fitting method. (B) A plot of difference curves. These are the curves that interpretations will be made on when a difference of curves is requested.

result in much higher variability during the bootstrap portion making the estimates more conservative.

A worked example

For a worked example, we analyzed data from a study comparing the timecourse of spoken word recognition in cochlear implant (CI) users and normal hearing (NH) listeners (Farris-Trimble et al., 2014). Listeners heard an auditory stimulus in VWP eye-tracking experiment examining looks to target, cohort, rhyme, and unrelated candidates. The fundamental comparison here is between-subject: does the amount of looking to targets or competitors differ between populations of listeners. However, we point out that as implemented, BDOTS can also evaluate within-subject comparisons (e.g., cohort vs. unrelated).

This original study showed that in addition to differences in fixations to cohorts and rhymes, CI users also differed slightly from NH listeners in looking to unrelated objects (perhaps reflecting greater uncertainty). Thus, here we evaluate competitor fixations as a difference between cohorts (or rhymes) and unrelated fixations. This asks if that difference differs significantly between populations. That is, we are interested in the number of extra looks to the cohort object, above and beyond the looks to the unrelated (control) object between CI and NH populations. BDOTS asks specifically at which time points this difference curve differs.

Step 2: Fitting the functions

Curves are fit using a generalized nonlinear least squares estimation method, using the gnls function within the nlme R package (Pinheiro, Bates, DebRoy, Sarkar, & Team, 2015; ver 3.1-122). This finds parameter estimates for the function that minimize the sum of the squared errors (SSE), between the fitted and observed values. It starts from an initial set of parameters (described shortly). It then continually modifies these parameters as long as the SSE continues to decrease, until it reaches an optimal point where the SSE can be decreased no more.

With nonlinear estimation, one needs to provide some sort of naïve parameter estimates to start the process. While this is also a requirement in other common techniques (e.g. GLMs, GEEs, LMEs), in nonlinear approaches, the quality of final estimates can be more contingent upon decent starting values. It is possible that some starting parameters will be so far from optimal that the method finds a local minimum with very different parameter estimates than the global minimum. Thus, users of BDOTS should to be aware of how these naïve values are determined, as well as ways in which the starting parameters may be refined if there are issues with fit.

In the BDOTS R package, there are tools for doing this. To estimate starting parameter for the 4-parameter logistic, we estimate the initial baseline at the minimum observed value of the data, the plateau at the maximum observed value of the data. The initial crossover is estimated as the time-point at which the observed data (proportion of fixations) is nearest to halfway between the baseline and plateau. The initial slope is the observed slope between the 25th percentile and 75th percentile.

For the doubleGaussian function we estimate the peak and its location (μ) using the maximum observed value and the time point at which this maximum was observed. Starting baselines are estimated as the minimum values on each side of the estimated peak location. To estimate the onset and offset slopes, we compute the total area under half the curve, from the peak location to a specified time point, and find the timepoint at which the area was closest to 68.3% (e.g., within one standard deviation of the mean in a Gaussian distribution).

Once these initial parameters are estimated, curve fitting is conducted. In computing the “error” in the function (for purposes of minimizing the least squared error), we assume that the variance from adjacent points are correlated. We use an AR1 correlation structure which assumes that a time point is correlated to the time point immediately prior, with correlation decaying exponentially as time points get further apart.

$$p_t|p_{t-1} = F(t,\ldots) + N(\rho p_{t-1}, 1-\rho^2)$$

Here p_t is the output of either the output of the logistic or double Gaussian function (defined in *F[t, …]*) at time *t*, for a given subject. To this we add noise coming from a Gaussian distribution *N*() with a mean consisting of the autocorrelation, ρ, multiplied by the value at the prior time point, p_{t-1}, and a variance of $1 - \rho^2$.

There are several important reasons to assume autocorrelated errors in the curve fitting stage. First, the assumed parametric form is not expected to perfectly match the raw data, resulting in residuals that are not independent of each other. Second, this assumption yields smaller estimated variances for the parameter estimates. This provides more consistent results in the bootstrapping later, resulting in smaller confidence intervals and more sensitive statistical tests. Third, there are additional clear theoretical advantages to this assumption (over not assuming any correlation) as autocorrelated error is a key property of timeseries data.

BDOTS wraps the curve fitter (gnls) in separate functions that (1) estimate the appropriate starting parameters; (2) construct the objective function along with the autocorrelated error; and (3) fit the function. One thing to note: parameters are assumed to follow a multivariate normal distribution with some estimable mean vector and variance matrix. This is an improvement from Oleson et al. (2017), where an assumption of independence between parameters was used, and has the consequence of reducing standard error of the function, improving power. Below we provide an example of running the fitting procedure, taken from our R vignette (where the 4th column of the data set corresponds to the proportion of looks at each time point).

There are several problems that can arise during curve fitting. First, because it is an optimization procedure, it is possible for the parameter estimates to reach a local minimum that is different than the global minimum. The algorithm will converge and one will get parameter estimates, but it will be clear that the estimates are not very accurate (we will discuss this in the following paragraph). Second, in some cases, a given subject's curve doesn't conform to the specified function (e.g., for target fixations, they may look away from the target toward the end of the trial, creating a dip in the function prior to the asymptote). Because one poorly fit subject can greatly impact the overall result of these tests, it is important to verify and check all fits.

There are several ways to do this. We recommend examining R^2 values to evaluate goodness of fits ($R^2 > 0.95$ is generally seen as a good fit)[1] as well as a visual examination of observed data compared to the estimated curve for each subject. We have built easy to use tools for both into the R package (illustrated below).

When issues with individual fits arise, there are a few options. The most immediate option is to specify better starting parameters. This can be a good solution in cases where the optimization procedure has found a local minimum – e.g., where the data clearly conforms to the shape of the function it is just not being estimated properly. The next approach is to relax the assumption of autocorrelated errors, which eliminates a free parameter and simplifies the search. This comes with a disadvantage: the standard errors around the parameter estimates are increased (though only for that given fit). This can slightly decrease power for detecting effects in the bootstrap phase. Finally, alternative curve fitters (e.g., more constrained algorithms) may achieve better fits. Estimates from an alternative fitter can be used (without further fitting) during the refit stage (see help for the two .refit functions). If a good curve fit is still not achieved after these steps, the subject should likely be dropped from analysis, or a different analysis used. If a subject's data cannot be fit with autocorrelated error, the curve is refit with this assumption dropped.

Example

In the following, taken from our R package's vignette, we walk through this procedure.[2] A more extensive tutorial can be brought up in R, with the vignette function.

[R Input]

```
>vignette("bdots")
```

[R Input]

```
>data(ci)
>names(ci)[1]<- "Group"
>ci<- subset(ci, ci$LookType == "Cohort" | ci$LookType == "Unrelated")
>ci$Curve<- ifelse(ci$LookType == "Cohort", 1, 2)
>summary(ci)
```

[R Output]

```
                        Subject                      Time                   Fixations
   Group
 CI:28056             Min.  : 2.00              Min.  : 0              Min.  :0.000000
 NH:26052             1st Qu.: 36.00            1st Qu.: 500           1st Qu.:0.003448
                      Median : 62.00            Median :1000           Median :0.020725
                      Mean  : 57.91             Mean  :1000            Mean  :0.043573
                      3rd Qu.: 82.00            3rd Qu.:1500           3rd Qu.:0.066176
                      Max.  : 106.00            Max.  :2000            Max.  :0.270677


  LookType                                      Curve
 Cohort :27054                                  Min.  :1.0
    Rhyme  : 0                                  1st Qu.:1.0
    Target : 0                                  Median :1.5
Unrelated:27054                                 Mean  :1.5
                                                3rd Qu.:2.0
```

A summary of the data is shown above. Here, the critical comparison is on the Group variable and the numeric variable, *Curve,* stores whether the data point comes from the Cohort or unrelated curve. The data is then fit by running:

[R Input]

```
R>fits<- doubleGauss.fit(ci, col = 4, diffs = TRUE)
```

Here, the grouping variables include *Group*, *Subject*, *Time*, and *Curve*. This latter variable is only used when calculating a difference of curves for each group, specified by setting *diffs* = *TRUE* in the fitting method. The fitting method will look for these specific column names, conducting fits across *Time* for each group × subject × curve combination. *LookType* is a leftover variable from the dataset that corresponds to the *Curve* number. *Fixations* are designated in the fitting method by specifying their column in the data set (column 4) as the second parameter in the fitter. This will output:

[R Output]

[1] In general, Pearson's R (and variants of) are not good estimates of fit for non-linear functions (since they don't require an exact match of the function and the data; they just must be correlated). However in this case, they offer a more useful measure than better tools like least-squares error or BIC since the scale of R^2 is the same regardless of the number of samples (which affects BIC), and the scale of the Y-axis (which effects RMS error). Thus, as a quick and dirty way of evaluating fit, R^2 has some utility in the fact that it is scale free and intuitive to most people.

[2] Note that we are continually refining the way that functions are called in BDOTs to make it easier to perform complex analyses. Consequently the specific function calls referenced here may not work exactly as written in future version. See the vignette and help files with your version of BDOTs.

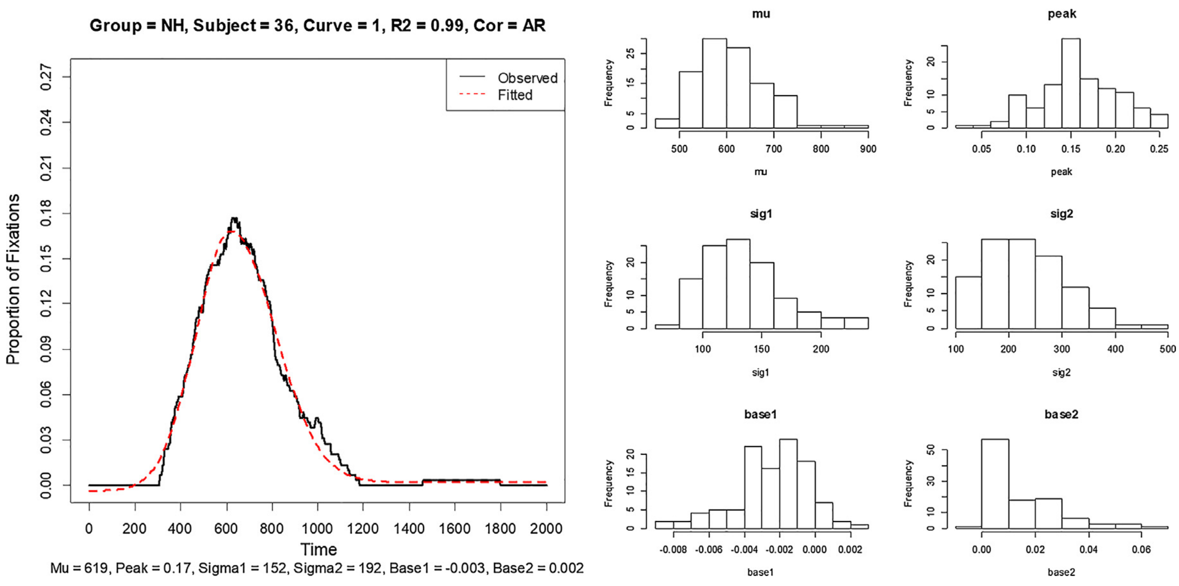

Fig. 5. (A) Sample output of subs.plot using data from the present example. (B) Sample output of ests.plot using data from the present example.

```
[1] “Group = NH, ID = 1, Subject = 36, Curve = 1, R2 = 0.99”
[1] “Group = NH, ID = 1, Subject = 36, Curve = 2, R2 =
   0.996”
[1] “Group = NH, ID = 2, Subject = 38, Curve = 1, R2 =
   0.995”
…
```

[Number of curves fit using AR1 or non-AR1 assumptions as well as a categorization of the goodness of fit measure, R2]

```
########################################
############### FITS ###################
########################################
AR1, R2 >=0.95 103
AR1, 0.95 > R2 >=0.8 5
AR1, 0.8 > R2 0
Non-AR1, R2 >=0.95 0
Non-AR1, 0.95 > R2 >=0.8 0
Non-AR1, 0.8 > R2 0
No Fit - 0
########################################
```

The output of the function shows a summary of the number of fits of various sorts. This particular run was able to find fits for all 108 participants using the assumption of autocorrelated errors. 103 had R^2 greater than 0.95; and 5 had somewhat lower but acceptable R^2 between 0.8 and 0.95. If any datasets could not be fit with autocorrelated errors they appear in the Non-AR1 section (or in the No Fit section if they could not be fit at all). Note that the numbers reported here do not necessarily correspond to subjects – in a within subject design, for example, each condition will be fit separately.

Next, we check the quality of the curve fits.

[R Input – Plot subject curves and histograms of all parameter estimates – See Figs. 1, 2]

```
R> subs.plot(fits)
R> ests.plot(fits)
```

Subs.plot shows each participant's (or condition's) data superimposed on the fitted curve (Fig. 5A). Ests.plot shows histograms of the estimated parameters (Fig. 5B). This can be used to quickly check for outliers.

After the initial fits, if any subjects showed poor fits, they can be refit with different starting parameters or without the AR1 errors. In the initial fitting stage, BDOTS automatically relaxes the AR1 assumption and refits all curves that were completely unable to be fit using AR1 errors, but does not automatically relax this assumption if the curve fit succeeded but gave a poor fit. In this case, the data must be refit where the user must manually relax the AR1 assumption. Other subjects may simply need to be refit using different starting parameters – these can often be "eyeballed" from the plot (e.g., the upper and lower asymptotes which are easily seen) or by estimating them from other subjects or conditions.

If any conditions need to be refit (with specified starting values, or with a different AR1 assumption) that is done next. To do this, first create a matrix with the starting parameters for all of the subjects that need to be refit.

[R Input – Set up matrix for refitting method]

```
> refit.matrix <- matrix(NA, nrow = 2, ncol = 9)
> #ORDER OF COLUMNS sub group curve Mu Peak S1 S2 B1 B2
> refit.matrix[1,] <- c(30, “CI”, 2, 650, 0.15, 150,
   100, 0, 0.03)
> refit.matrix[2,] <- c(83, “CI”, 2, 700, 0.10, 150,
   100, 0, 0.01)
```

This holds the starting assumptions for subjects 13 and 23 (both in condition 2) using new estimated starting parameters (e.g., for subject 13, $\mu = 650$, $p = 15$, $\sigma_1 = 150$, $\sigma_2 = 100$, $B_1 = 0$ and $B_2 = .03$). The first three columns are grouping variables and the remaining columns are initial parameter estimates. It can also be helpful to read this matrix in from a text file. Next this matrix is sent to the refit function. Here, by setting cor = TRUE, it is possible to fit the data with assuming AR1 error. Cor can also be set to a matrix indicating this for each subject/condition.

[R Input – Refit curves while providing new estimated starting values and relaxing the correlation assumption]

```
> fits <- doubleGauss.refit(fits, cor = FALSE,
   info.matrix = refit.matrix)
```

[R Output – Large improvement in R^2]

```
Subject = 30, Group = CI, Curve = 2, Old R2 = 0.844
  New R2 = 0.954, Old AR1 = TRUE, New AR1 = FALSE
Subject = 83, Group = CI, Curve = 2, Old R2 = 0.884
```

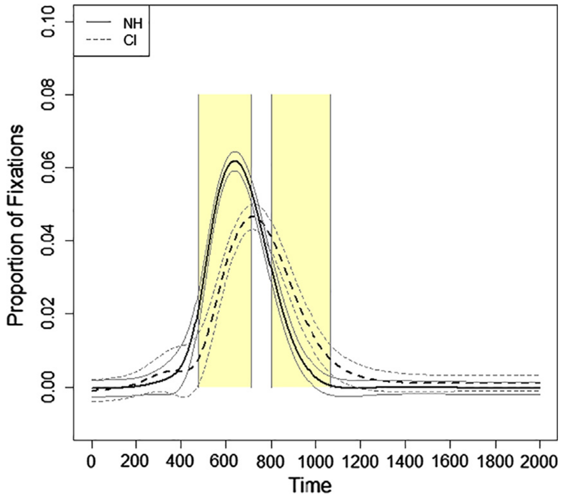

Fig. 6. The final results of the bootstrapping method with graphical parameters modified.

```
  New R2 = 0.954, Old AR1 = TRUE, New AR1 = FALSE
```

[R Input – Check updated summary of fits]

```
>printFits(fits)
```

[R Output]

```
########################################
############### FITS ###################
########################################
AR1, R2>=0.95 - 103
AR1, 0.95>R2>=0.8 - 3
AR1, 0.8>R2 - 0
Non-AR1, R2>=0.95 - 2
Non-AR1, 0.95>R2>=0.8 - 0
Non-AR1, 0.8>R2 - 0
No Fit - 0
########################################
```

Once good fits have been obtained, bootstrapping begins.

Step 3: Bootstrapping

After step 1, each subject will have estimates of each parameter and the corresponding variance matrix. During a bootstrap iteration, for each subject a new, random set of parameters is drawn from a multivariate normal distribution. These are used to calculate a new timecourse curve for each bootstrapped subject. Subsequently, group-averaged curves are created by averaging the individual bootstrapped curves within a group at each time point *t*. Finally, using all the group-averaged curves from the bootstrap iterations, an observed mean and standard deviation are calculated at every *t*. Note that the observed bootstrap standard deviation is actually an estimate of the group-averaged standard error.

In cases where we have a within-subject design there is a slight modification (in BDOTS users must set the *paired* variable to TRUE to signal this design). During the bootstrap iteration, for a specific subject, the random values are drawn from bivariate normal distributions (e.g. in the logistic curve, the peak parameter for both curves are drawn together). We use the parameter estimates from step 1 to calculate covariance estimates between the two curves (separate covariance estimates are created for each parameter).

Step 4: Testing and error correction

Finally, using these estimates of the bootstrapped mean and standard deviations, we conduct pairwise comparisons at each time to determine regions of difference in the curves. We start by performing t-tests at every time point (either between subjects or paired depending if subjects are matched between groups). A traditional t-statistic is calculated, assuming equal variance between groups, but potentially unequal sample sizes. Recall that we only have access to the bootstrapped SDs (group-specific SEs). Thus, the typical variance estimates in calculating the t-statistic are replaced with the standard error estimates multiplied by the size of the group.

Once the series of T statistics is obtained at each time point, a time series model is fit to the series of T statistics using an AR(1) correlation structure (assumes $T_t|T_{t-1} \sim N(\rho T_{t-1}, 1-\rho^2)$). Through this process, an estimate of the autoregression coefficient, $\hat{\rho}$, is obtained which describes the correlation between adjacent T statistics.

Finally, the probability of a family-wise Type I Error is calculated on the basis of both an assumed threshold for significance (α), and the observed autocorrelation, $\hat{\rho}$, computed in the previous step. The family-wise Type I Error is computed using a theoretical distribution which assumes that there is no true difference between groups, but that adjacent T statistics exhibit the observed level of autocorrelation (for details of how this is done and a partial mathematical derivation, see Oleson et al., 2017): This is computed for many values of α^* until one provides an overall *P(type I error)* that is equal to the desired α (typically .05). This α^* is then used as the individual α -level in all t-test calculations. The BDOTS R package will estimate this automatically.

In the following example, we take the output of the curve fitting, and send them to the bootstrapping function. This automatically performs the bootstrap, computes the series of t-tests, computes their autocorrelation and α^*, and then evaluates their significance. In cases where subjects are paired across groups, the bootstrap method requires the paired argument to be set to TRUE.

[R Input – Perform bootstrapping and error correction]

```
R>bootstraps<- doubleGauss.boot(fits, paired =
   FALSE)
```

[R Output – Significance in the [476, 708] and [804, 1064] regions. $\alpha^* = 0.003$ and $\hat{\rho}=0.9992$]

```
$alpha
  alpha alpha.adj rho.est
  0.0500 0.0030 0.9992
$significant
  [,1] [,2]
[1,] 476 708
[2,] 804 1064
```

It then outputs whatever regions of significance it finds. In this case, it finds two, between 476 and 708 msec and between 804 and 1064 msec. It will also generate a plot which shows the mean estimated timecourse functions, their bootstrapped confidence intervals, and with significant regions highlighted. These can be easily formatted using the replot function (below).

[R Input – Modify graphical parameters in the plot – See Fig. 6]

```
R>replot(bootstraps, ylim = c(-0.01, 0.1),
   bucket.lim = c(0, 0.08))
```

Additional BDOTS features

Parallel processing

To speed computation, the curve fitting and bootstrapping steps of BDOTS have parallel implementations. For the parallel implementations, the individual fitting of the curves and the individual bootstrap sample estimates are sent to separate cores which can reduce the bootstrapping time substantially. These functions have an argument (*cores*) which defaults to 1 but can be changed to the appropriate number of cores for the system. A typical recommendation is one less than the number of physical cores available.

Saving bootstrapped data

There is an option within the package for outputting the bootstrapped data to a .csv file so that results may be analyzed or plotted in another program. For each group, this function outputs their estimated function, the mean of the raw data, and lower/upper confidence interval estimates at each time point, as well as the modified p-value for significance of group difference. This is implemented through the method *bdots.write.csv*.

```
R > bdots.write.csv(fits,bootstraps, “bdots.txt” =
    row.names = FALSE)
```

Family-wise error correction

The family-wise error correction (assuming auto-correlated test statistics) can be used with any timeseries of T-statistics, not just those generated by BDOTs. This is exposed as a function (tsmultcomp), which takes as arguments the number of tests being performed, the autocorrelation of the test statistics (which can be computed with ar()), the degrees of freedom, and the desired alpha. It outputs adjusted α^*.

Reporting

If using this method to analyze data, reported results should include the given α (the overall alpha assumed by the researcher), the calculated α^* (alpha.adj in the output), regions of significance, the estimated correlation of the T-statistics (rho.est in the output), and the degrees of freedom used for the T-tests (number of subjects minus 2 for a between subjects analysis or $N - 1$ for within subjects). If the estimated correlation of the T-statistics is small, then researchers may want to consider another analysis or correction method, as BDOTS may be conservative.

Information should be provided about the quality of the individual curve fits, including the number of subjects removed from each group due to poor fit, the number of Non-AR(1) fits, the number of AR(1) fits, as well as the number of curves defined as good fits ($R^2 > 0.95$), adequate fits ($0.95 > R^2 > 0.80$), and poor fits ($R^2 < 0.80$) in the AR (1) and non-AR(1) categories (available via print.fits()). Subjects removed from the analysis should have a short description as to the reason they were removed (e.g. their curve did not follow the general trend of the population or their curve could not be adequately fit). Additionally, a measure of the change in the group level eye-tracking curve should be reported (as well as a figure of the pre- and post-deletion curves if possible). Next, the function used for curve fitting (logistic vs. double Gaussian) and subject setup (between subject vs within subject) should be reported. Finally, regions of significance and the direction of the effect should be reported. The average fixations within these windows can be reported as descriptive statistics.

An example results section should look like the following: 40 subjects were fit using the double Gaussian function in a between-subject study design where each group was a single condition (difference of conditions). In the fitting stage, 37 curves had good fits with AR1 ($R^2 > =.95$), 1 curve had good fits without AR1, 1 curves had reasonable fits with AR1 ($R^2 > =.8$), and 1 subject (from the CI group) was dropped due to poor fitting in at least one of their curves. The average absolute change in the CI group curve by dropping the poor fitting subject was 0.001. In the bootstrapping stage, autocorrelation of the t-statistics was 0.998, the adjusted alpha was calculated to be 0.004. We found regions of significance at [350, 600] and [1000, 2000]. In both cases, the CI group exceeded the NH group (region 1: difference = .03; region 2: difference = .023).

Discussion

This paper introduced a new technique for analyzing differences among two population curves for timeseries data with high levels of autocorrelation. We focused on eye-tracking in the visual world paradigm as our prime example but the technique may be broadly useful in other domains as well (particularly as new functions are added to the package). BDOTS is not a complete solution, but it does offer some unique insight into timeseries data and a platform for further statistical development. Here we discuss it relative to existing approaches, before mentioning one additional caveat.

Comparison to other approaches

Prior approaches for analyzing such data would aggregate the fixation data over a specific timewindow (AUC analyses) and use these aggregates in more traditional analyses. In the best-case scenario, where this window is chosen before data collection begins, interpretation still suffers because the AUC is averaged across a window rather than at specific time points. The worst-case scenarios would be situations in which this window is chosen by looking at the data for areas of interest or, even worse, by testing different potential windows and choosing the one with the largest effect. These would clearly inflate Type I error. Our method allows for interpretation at each individual time point in addition to avoiding any need for choosing this arbitrary window to aggregate over. Additionally, we are able to provide confidence intervals and p-values corresponding to group differences at each time point.

Another approach has been to incorporate polynomial terms into a mixed effects model analysis. While this offers good fits to the data, interpretation of the differences are essentially reduced to stating that there exists some difference between the groups and little more. Our method allows for specification of a function of the user's choice (and polynomial functions will be an option in the near future) but this choice only affects the quality of curve fits and is mostly discarded later on when values are bootstrapped, so it is less dependent on the specific form of the function. Moreover, when using a mixed effects model to analyze this data, the use of group as a variable indicates that it will have a constant effect on the mean of the function (unless additional group × time and group × time^2 [etc] interaction terms are added). However, with our approach the size of the group difference can vary with time as the goal is not to model the effect of group as part of the function, but rather to use the functions to estimate its effect.

While our method does allow the relaxation of the assumption of a constant group effect, it does not allow for inclusion of covariates other than group. In principle this can be done by incorporating more complex test statistics at each point in time. However, in practice this may be difficult as separate good fits will need to be obtained for different subsets of the data, and this may be difficult as these subsets may be too small to obtain good fits.

It is also important to point out that BDOTs uses a true strong family-wise error correction. It neither admits some possibility of false-discovery (as an FDR based solution would) nor does it simply collapse nearby comparisons to avoid the issue as cluster-based permutation approaches do. As we have demonstrated (Oleson et al., 2017) this leads to robust power, while maintaining family-wise alpha. Moreover, while there are a number of researcher driven choices (e.g., the choice of the function), inferences are not made on the direct basis of these choices. As a result, in most cases, the consequences of overfitting the data are minimal (if anything, a poorly chosen function will yield less power). Consequently, BDOTs may be more robust to researcher d.f. than other approaches.

Finally, while BDOTS can detect an onset effect by looking for the first significant difference along the timecourse, we are unable to provide a confidence interval or test to compare onsets of two separate curves in the current framework. That is, BDOTS focuses on comparisons along the Y axis (degree of looking) not comparisons on the X axis (time). This may be possible by adapting existing techniques (McMurray, Clayards, et al., 2008) in the BDOTS framework. For example, once functions are estimated, one could estimate the time at which they cross a fixed threshold. The variance of this new time-estimate could then be estimated at the bootstrap step in order to compare the onset of two effects.

Dangers of overfitting

As always, with nonlinear fitters, it is important to be cautious about overfitting the available data. The largest danger lies within the selection of a non-parametric function after looking at the data, as a function can specifically be chosen that represents the observed curve well but may not a good fit to the true underlying trend. To combat this, it is important to make decisions about the parametric form of the function used for fitting (e.g. 4-parameter logistic or double Gaussian) prior to viewing of the data and to constrain choices by the general properties of the measure, and the history of work with this measure. Because the functions included in BDOTS (the 4-parameter logistic and double Gaussian) have relatively few parameters for the amount of data and are specified based on a history of timecourse curves in the VWP, the likelihood of overfitting based on over parameterization is small. However, if the amount of data were reduced by decreasing the period of time over which responses are observed or sampling less frequently, overfitting could result in overly optimistic fits and inflated Type I Error. It is important to justify future function implementations a priori, and not based on observed timecourse curves from a current study, or run the risk of overfitting.

Item effects

An ongoing issue in psycholinguistics is accounting for variance due to items (words/sentences) as well as more traditional variation due to subjects. This was traditionally solved by separate item and subject analyses (F1/F2 analyses: Clark, 1973), and more recently by mixed models that account for both simultaneously (Baayen et al., 2008). BDOTS does not yet have the ability to capture crossed random effects, and ongoing work should build on the tools presented here to develop such models. In the meantime, BDOTS can be used with separate item and subject analyses and an F1/F2 style approach to combining these analyses. This is not a perfect alternative to the incorporation of item effects, but is better than interpretations based on subject analyses alone.

Other caveats and future directions

While fixation curves generally fit quite well for typical participants (when sufficient repetitions are available), in some cases looking may not follow the same pattern between participants (e.g., see individual curves plotted in Yee, Blumstein, & Sedivy, 2008). One approach that could be explored is to use different functions to fit curves for each individual. The bootstrap does not care that all participants use the same function as long as it can generate the estimated fixations and their variance for that subjects. For some populations this may work well statistically. However, this raises concern as to what is truly being analyzed when different subjects exhibit functions of qualitatively different forms. Of course, the ultimate solution for poor fits is to remove them from the procedure. However, care must be taken to not systematically modify the overall group estimate by excluding a common group of subjects. And of course, with special populations this may not always be advisable. Such concerns may be mitigated by summarizing the change in group averages when subjects are dropped.

Finally, having T as the only choice for a sample statistic is limiting. Implementation of the Pearson correlation, R, should be straightforward as the distribution assumptions are similar to those of T. Ultimately, F would be of particular interest (detecting a difference between at least two groups within a larger collection of groups) but would require more theoretical work.

Conclusion

BDOTS provides a computationally efficient package for analyzing highly correlated temporal data in such a way that the true family-wise error rate can be maintained. BDOTS is able to take advantage of the functional definitions of the 4-parameter logistic and double Gaussian for eye-tracking data, but is general enough where additional functions could be defined for other types of timeseries data, and would provide the same benefits from being able to describe the curve in a parametric form. More importantly, in contrast to previous approaches BDOTS allows inference of group differences without assuming a particular time window, can identify the window of interest, and avoids the computationally expensive task of calculating permutation tests. Because the timewindow is no longer up to the researcher, it eliminates one opportunity for p-hacking.

Acknowledgements

This research was funded in part by the National Institutes of Health – United States grant numbers DC000242 and DC008089. We would like to thank Ashley Farris-Trimble for helpful discussions on curve fitting. We would also like to thank the reviewers for their insightful comments which helped to improve the paper.

References

Allopenna, P., Magnuson, J. S., & Tanenhaus, M. K. (1998). Tracking the time course of spoken word recognition using eye-movements: Evidence for continuous mapping models. *Journal of Memory and Language, 38*(4), 419–439.

Apfelbaum, K. S., Blumstein, S. E., & McMurray, B. (2011). Semantic priming is affected by real-time phonological competition: Evidence for continuous cascading systems. *Psychonomic Bulletin and Review, 18*(1), 141–149.

Baayen, R. H., Davidson, D. J., & Bates, D. M. (2008). Mixed-effects modeling with crossed random effects for subjects and items. *Journal of Memory and Language, 59*(4), 390–412. http://dx.doi.org/10.1016/j.jml.2007.12.005.

Bakker, M., van Dijk, A., & Wicherts, J. M. (2012). The rules of the game called psychological science. *Perspectives on Psychological Science, 7*(6), 543–554. http://dx.doi.org/10.1177/1745691612459060.

Barr, D. J., Jackson, L., & Phillips, I. (2014). Using a voice to put a name to a face: The psycholinguistics of proper name comprehension. *Journal of Experimental Psychology: General, 143*(1), 404.

Benjamini, Y., & Hochberg, Y. (1985). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society B, 85*, 289–300.

Blumenfeld, H. K., & Marian, V. (2007). Constraints on parallel activation in bilingual spoken language processing: Examining proficiency and lexical status using eye-tracking. *Language and Cognitive Processes, 22*(5), 633–660. http://dx.doi.org/10.1080/01690960601000746.

Bullmore, E. T., Suckling, J., Overmeyer, S., Rabe-Hesketh, S., Taylor, E., & Brammer, M. J. (1999). Global, voxel, and cluster tests, by theory and permutation, for a difference between two groups of structural MR images of the brain. *IEEE Transactions on Medical Imaging, 18*(1), 32–42.

Clark, H. H. (1973). The language-as-fixed-effect fallacy: A critique of language statistics in psychological research. *Journal of Verbal Learning and Verbal Behavior, 12*(4), 335–359. http://dx.doi.org/10.1016/s0022-5371(73)80014-3.

Estes, W. K. (1956). The problem of inference from curves based on group data. *Psychological Bulletin, 53*(2), 134.

Farris-Trimble, A., & McMurray, B. (2013). Test-retest reliability of eye tracking in the visual world paradigm for the study of real-time spoken word recognition. *Journal of Speech Language and Hearing Research, 56*, 1328–1345.

Farris-Trimble, A., McMurray, B., Cigrand, N., & Tomblin, J. B. (2014). The process of spoken word recognition in the face of signal degradation: Cochlear implant users and normal-hearing listeners. *Journal of Experimental Psychology: Human Perception and Performance, 40*(1), 308–327.

Frye, R. E., Rezaie, R., & Papanicolaou, A. C. (2009). Functional neuroimaging of language using magnetoencephalography. *Physics of Life Reviews, 6*(1), 1–10. http://dx.doi.org/10.1016/j.plrev.2008.08.001.

Ju, M., & Luce, P. A. (2004). Falling on sensitive ears: Constraints on bilingual lexical activation. *Psychological Science, 15*(5), 314–318. http://dx.doi.org/10.1111/j.0956-7976.2004.00675.x.

Maris, E., & Oostenveld, R. (2007). Nonparametric statistical testing of EEG-and MEG-data. *Journal of Neuroscience Methods, 164*(1), 177–190.

McMurray, B., Aslin, R. N., Tanenhaus, M. K., Spivey, M. J., & Subik, D. (2008). Gradient sensitivity to within-category variation in words and syllables. *Journal of Experimental Psychology, Human Perception and Performance, 34*(6), 1609–1631.

McMurray, B., Clayards, M., Tanenhaus, M. K., & Aslin, R. N. (2008). Tracking the time course of phonetic cue integration during spoken word recognition. *Psychonomic Bulletin and Review, 15*(6), 1064–1071.

McMurray, B., Samelson, V. S., Lee, S. H., & Tomblin, J. B. (2010). Individual differences in online spoken word recognition: Implications for SLI. *Cognitive Psychology, 60*(1), 1–39.

McMurray, B., Tanenhaus, M. K., & Aslin, R. N. (2009). Within-category VOT affects recovery from "lexical" garden paths: Evidence against phoneme-level inhibition. *Journal of Memory and Language, 60*(1), 65–91.

McMurray, B., Tanenhaus, M. K., Aslin, R. N., & Spivey, M. J. (2003). Probabilistic constraint satisfaction at the lexical/phonetic interface: Evidence for gradient effects of within-category VOT on lexical access. *Journal of Psycholinguistic Research, 32*(1), 77–97.

Miller, J. O., Patterson, T., & Ulrich, R. (1998). Jackknife-based method for measuring LRP onset latency differences. *Psychophysiology, 35*, 99–115.

Mirman, D., Dixon, J., & Magnuson, J. S. (2008). Statistical and computational models of the visual world paradigm: Growth curves and individual differences. *Journal of Memory and Language, 59*(4), 475–494.

Mordkoff, J. T., & Gianaros, P. J. (2000). Detecting the onset of the lateralized readiness potential: A comparison of available methods and procedures. *Psychophysiology, 37*(3), 347–360. http://dx.doi.org/10.1111/1469-8986.3730347.

Oakes, L. M., Baumgartner, H. A., Barrett, F. S., Messenger, I. M., & Luck, S. J. (2013). Developmental changes in visual short-term memory in infancy: Evidence from eye-tracking. *Frontiers in Psychology, 4*, 697.

Oleson, J. J., Cavanaugh, J. E., McMurray, B., & Brown, G. (2017). Detecting time-specific differences between temporal nonlinear curves: Analyzing data from the visual world paradigm. *Statistical Methods in Medical Research, 26*(6), 2708–2725.

Osterhout, L., McLaughlin, J., & Bersick, M. (1997). Event-related brain potentials and human language. *Trends in Cognitive Sciences, 1*(6), 203–209. http://dx.doi.org/10.1016/S1364-6613(97)01073-5.

Pinheiro, J., Bates, D., DebRoy, S., Sarkar, D., & Team, R. C. (2015). *nlme: Linear and nonlinear mixed effects models*. Retrieved from < http://CRAN.R-project.org/package=nlme > .

Reinisch, E., & Sjerps, M. J. (2013). The uptake of spectral and temporal cues in vowel perception is rapidly influenced by context. *Journal of Phonetics, 41*(2), 101–116.

Salverda, A. P., Brown, M., & Tanenhaus, M. K. (2011). A goal-based perspective on eye movements in visual world studies. *Acta Psychologica, 137*(2), 172–180.

Scheepers, C., Keller, F., & Lapata, M. (2008). Evidence for serial coercion: A time course analysis using the visual-world paradigm. *Cognitive Psychology, 56*(1), 1–29.

Schmidt, G. L., & Roberts, T. P. (2009). Second language research using magnetoencephalography: A review. *Second Language Research, 25*(1), 135–166.

Simmons, J. P., Nelson, L. D., & Simonsohn, U. (2011). False-positive psychology: Undisclosed flexibility in data collection and analysis allows presenting anything as significant. *Psychological Science, 22*(11), 1359–1366. http://dx.doi.org/10.1177/0956797611417632.

Spivey, M. J., Grosjean, M., & Knoblich, G. (2005). Continuous attraction toward phonological competitors. *Proceedings of the National Academy of Sciences of the United States of America, 102*(29), 10393–10398. http://dx.doi.org/10.1073/pnas.0503903102.

Swaab, T. Y., Ledoux, K., Camblin, C. C., & Boudewyn, M. A. (2012). Language-related ERP components. In *Oxford handbook of event-related potential components* (pp. 397–440).

Tanenhaus, M. K., Spivey-Knowlton, M. J., Eberhard, K. M., & Sedivy, J. C. (1995). Integration of visual and linguistic information in spoken language comprehension. *Science, 268*, 1632–1634.

Toscano, J. C., & McMurray, B. (2012). Online integration of acoustic cues to voicing: Natural vs. synthetic speech. *Attention, Perception & Psychophysics, 74*(6), 1284–1301.

Vandeberg, L., Bouwmeester, S., Bocanegra, B. R., & Zwaan, R. A. (2013). Detecting cognitive interactions through eye movement transitions. *Journal of Memory and Language, 69*(3), 445–460.

Viviani, P. (1989). Eye movements in visual search: Cognitive, perceptual and motor control aspects. *Reviews of Oculomotor Research, 4*, 353–393.

Watson, D. G., Tanenhaus, M. K., & Gunlogson, C. A. (2008). Interpreting pitch accents in online comprehension: H^* vs. $L+H^*$. *Cognitive Science, 32*(7), 1232–1244. http://dx.doi.org/10.1080/03640210802138755.

Weighall, A., Henderson, L.-M., Barr, D., Cairney, S. A., & Gaskell, M. G. (2017). Eye-tracking the time-course of novel word learning and lexical competition in adults and children. *Brain and Language, 167*, 13–27.

Wu, S., Barr, D. J., Gann, T. M., & Keysar, B. (2013). How culture influences perspective taking: Differences in correction, not integration. *Frontiers in Human Neuroscience, 7*, 822.

Yee, E., Blumstein, S. E., & Sedivy, J. C. (2008). Lexical-semantic activation in Broca's and Wernicke's aphasia: Evidence from eye movements. *Journal of Cognitive Neuroscience, 20*(4), 592–612.