TECHNICAL
REPORT

0697

FITTING THE RATCLIFF DIFFUSION MODEL TO EXPERIMENTAL DATA

VANDEKERCKHOVE, J. and F. TUERLINCKX

IAP STATISTICS
NETWORK

INTERUNIVERSITY ATTRACTION POLE

http://www.stat.ucl.ac.be/IAP

Title: Fitting the Ratcliff diffusion model to experimental data

Author names: Joachim Vandekerckhove and Francis Tuerlinckx

Author affiliations: University of Leuven

Contact address:

Joachim Vandekerckhove

Research Group Quantitative Psychology and Personality Psychology

Department of Psychology

Tiensestraat 102

B-3000 Leuven

Belgium

Ph: +32 16 32 61 18

E-mail: joachim.vandekerckhove@psy.kuleuven.be

Running head: Fitting the Ratcliff diffusion model

Keywords: decision making, mathematical modeling, reaction time analysis, statistics

Abstract

Many experiments in psychology yield both reaction time and accuracy data. However, no off-the-shelf methods yet exist for the statistical analysis of such data. One particularly successful model has been the diffusion process, but using it is difficult in practice because ad-hoc statistical methods and custom-written software are needed to process specific data sets. We present a general method for performing diffusion model analyses on experimental data. By implementing design matrices, a wide range of across-condition restrictions can be imposed on model parameters, in a flexible way. It becomes possible to fit models with parameters regressed onto predictors. We present an easy-to-use software tool that helps perform diffusion model analyses. The algorithm makes use of several techniques that boost its efficiency, some of which we briefly discuss.

Introduction

Mental chronometry, the study of psychological processes through observed response times, is one of the most prevalent approaches in cognitive psychology. As early as 1868, Donders (1868/1968) used reaction time measurements in order to investigate differences between mental processes. Since then, reaction time studies have been used in perhaps all fields of cognitive science. Such is the importance of reaction time data to cognitive psychology that methods for analyzing them have become an object of study in their own right (e.g., Luce, 1986; Townsend, 1990).

Continuing this trend, considerable attention has been lent to the combination of reaction time and accuracy data (a ubiquitous combination often referred to as *two-choice response time data*). For the analysis of this type of data, several nonlinear statistical models have been developed, often with substantive interpretations attached to the parameters and underlying processes (e.g., the discrete random walk model; Laming, 1968; Link & Heath, 1975). A more advanced model – and the one that is at the heart of the present article – is the Ratcliff Diffusion Model (Ratcliff, 1978, 1981, 1985, 1987, 1988). The latter model, which will be described in detail in the next section, has performed remarkably well in the analysis of two-choice response time data. It has successfully been applied to experiments in many different fields, such as memory (Ratcliff, 1978, 1988), letter matching (Ratcliff, 1981), lexical decision (Ratcliff, Gomez, & McKoon, 2004), signal detection (Ratcliff

& Rouder, 1998; Ratcliff, Thapar, & McKoon, 2001; Ratcliff, Van Zandt, & McKoon, 1999), visual search (Strayer & Kramer, 1994), and perceptual judgment (Ratcliff, 2002; Ratcliff & Rouder, 2000; Thapar, Ratcliff, & McKoon, 2003). In particular, the Ratcliff Diffusion Model (RDM) succeeds in explaining characteristic aspects of two-choice response time data such as the occurrence of both fast and slow errors.

In spite of its advantages, the Ratcliff Diffusion Model has not yet become a popular or widely used method to analyze the numerous two-choice response time data. The reasons for this lack of dispersion have to do with numerical, statistical, and software issues. A first set of reasons concerns the fact that the model is prohibitively difficult to implement for applied researchers because of numerical difficulties. One has to deal with an infinite oscillating series in the expression for the cumulative distribution function (CDF) or probability density function (PDF; see Ratcliff & Tuerlinckx, 2002). Recently, Voss and Voss (2007) have proposed a method to circumvent the problem but their solution relies on the numerical solution of a partial differential equation. In addition, some of the parameters are allowed to vary from trial to trial and this leads to (partly) intractable integrals (Tuerlinckx & Ratcliff, 2002; Tuerlinckx, 2004). Once the CDF or PDF have been computed, the task of estimating the parameters requires some skill in the area of function optimization. In sum, some experience with numerical methods is needed to implement the model.

A second group of reasons that forestalled widespread use of the RDM is related to statistical issues. The type of data used to fit the diffusion model is rather complex. On each trial, there is a bivariate response, consisting of a reaction time and a choice. The latter is binary and the former is continuous, but non-normal (skewed and with a lower boundary possibly different from zero). The treatment of such data is not a trivial issue and traditional statistical methods suited for linear analysis (e.g., relying on means, computing R^2, etc.) fail. In addition, the reaction time measure is possibly cursed with outliers and contaminants.

The third category of reasons has to do with the fact that at the time of writing, there is no flexible or general software available for diffusion model analysis. In each of the studies cited above, fitting software was custom-written. However, researchers often collect data in a design that deviates from designs for which the previously developed software was written.

It is the goal of the current paper to make diffusion model analysis accessible to a more general public of researchers. To this end, we provide some general methods for fitting the Ratcliff Diffusion Model and other data-analytical strategies useful in the same practical context. Also, we provide some demonstration of a MATLAB tool that implements the methods we present (the *Diffusion Model Analysis Toolbox*; Vandekerckhove & Tuerlinckx, 2007a). For an introduction to the practical side of working with "DMAT", however, we refer the interested reader to the DMAT primer (Vandekerckhove & Tuerlinckx, 2007b).

In what follows, we start with a brief explanation of the Ratcliff Diffusion Model. Next, we outline a *design matrix method* that permits to impose substantive restrictions on the model's parameters. This very flexible technique facilitates fitting of the RDM and allows for the construction of models that can capture a variety of substantive hypotheses.

Next, we will discuss techniques related to the estimation of the parameters of the RDM (i.e., the construction and minimization of the loss function, and the handling of outliers). We then briefly introduce our diffusion model analysis toolbox for MATLAB ("DMAT").

Then we will show which statistical methods are needed for testing substantive hypotheses and comparing different models. We present results from simulation studies where properties of these statistical methods are investigated.

Finally, we demonstrate the use of our methods and software in two example applications.

The Ratcliff Diffusion Model

Parameters of the model

The diffusion process (see Figure 1) has been used to describe and model the decision component in simple two-choice tasks. In the model, it is assumed that an observer has a one-dimensional *internal representation of evidence*, where the two choice options reside at the extremes. When the observer is presented with a stimulus, information regarding it is *accumulated* *sequentially over time* until its total amount reaches the upper or lower bound, resulting in a response (*absorbing boundaries*). The decision time is defined as the time from the start of the process until the moment one of the absorbing boundaries is reached.

* Figure 1 about here *

The RDM has seven parameters. The first parameter is the *boundary separation,* denoted by *a*. If *a* is small, the process is expected to end sooner but it is more prone to error since random variability inherent to the decision process may cause it to end up at the wrong boundary. When *a* is high, both accuracy and expected reaction time will increase. The distance between the two absorbing boundaries therefore regulates the relation between speed and accuracy (the so-called *speed-accuracy trade-off*).

A second property of the model is the *starting point* of the information accumulation process, which is denoted as z_0 ($0 < z_0 < a$). This parameter introduces the possibility of *response bias in the decision process* because the process is more likely to end at the boundary closer to the starting point. We will assume z_0 to vary from trial to trial (Laming, 1968), according to a uniform distribution, with mean *z* ($0 < z < a$) and range s_z ($0 < s_z < \min(z, a - z)$). These two, *z* and s_z, are the second and third parameters of the RDM.

Furthermore, the information accumulation process can have a tendency to drift off to one of the two absorbing boundaries, depending on the *quality of the stimulus* presented. This information accumulation rate, or *drift rate*, is assumed to vary *within* a trial, following a Gaussian distribution with mean ξ and standard deviation *s*, but also *across* trials (Ratcliff, 1978), such that ξ follows a Gaussian distribution with mean *v* and standard deviation η. An experimental condition with non-ambiguous stimuli will lead to a large positive mean drift rate *v*, thus a high probability of hitting the upper boundary (indicating a correct response) in a short time. The standard deviation *s*, which indicates the volatility in drift rate in a single trial, is a non-identified parameter in the model, so we fix it to the arbitrary value 0.1 (which is a consensus value in the literature, e.g., Ratcliff et al., 1999). Thus, with *s* being and ξ a random variable, we add a fourth and fifth parameter to the model, namely the mean drift rate *v* and its intertrial standard deviation η.

Finally, another component of the model is the time needed to perform non-decision processes such as encoding of the stimulus, response preparation and execution of the motor response (Luce, 1986). We denote the non-decision part of the observed reaction time as t_{er}. This t_{er} is assumed to vary from trial to trial, according to a uniform distribution with mean T_{er} and range s_t. These two are the sixth and seventh parameters of the RDM.

Some notational conventions

In the preceding section, we have defined the seven key parameters of the diffusion model. In the remainder of the article, we will repeatedly capture all of these parameters in a *parameter vector* $\boldsymbol{\theta}_{(c)} = \left(a_{(c)}, T_{er(c)}, \eta_{(c)}, z_{(c)}, s_{z(c)}, s_{t(c)}, v_{(c)}\right)$, where the bracketed subscript (c) refers to the c^{th} condition in an experiment, and $c = 1, \cdots, C$. When working with different conditions in an experiment (and thus different parameter vectors), we will vertically concatenate the parameter vectors into a *parameter matrix* $\mathbf{P}$. Thus, if we have *C* conditions,

$$\mathbf{P} = \left\{ \begin{bmatrix} a_{(1)} \\ \vdots \\ a_{(C)} \end{bmatrix}, \begin{bmatrix} T_{er(1)} \\ \vdots \\ T_{er(C)} \end{bmatrix}, \begin{bmatrix} \eta_{(1)} \\ \vdots \\ \eta_{(C)} \end{bmatrix}, \begin{bmatrix} z_{(1)} \\ \vdots \\ z_{(C)} \end{bmatrix}, \begin{bmatrix} s_{z(1)} \\ \vdots \\ s_{z(C)} \end{bmatrix}, \begin{bmatrix} s_{t(1)} \\ \vdots \\ s_{t(C)} \end{bmatrix}, \begin{bmatrix} v_{(1)} \\ \vdots \\ v_{(C)} \end{bmatrix} \right\}$$

. A single column in such a parameter matrix then contains estimates of one specific parameter over conditions, and such a column vector will be denoted with a $\boldsymbol{\psi}$. For example, the nondecision time in condition *c* will be denoted as $T_{er(c)}$, which is the c^{th} element of $\boldsymbol{\psi}_{T_{er}}$ (the second column of $\mathbf{P}$), and the second element of $\boldsymbol{\theta}_{(c)}$ (the c^{th} row of $\mathbf{P}$). Figure 2 provides a graphical presentation of these matrices.

* Figure 2 about here *

The design matrix method

There are several reasons why a researcher might not be interested in fitting a model with all parameters free, as depicted in Figure 2. First, there is the issue of parsimony. Fitting the Ratcliff Diffusion Model to an experiment with C conditions would leave us with $7 \times C$ distinct parameters to estimate. Even if the number of conditions is moderate, for example $C = 5$, this leads to a large number of parameters to be estimated (i.c., 35 parameters to be estimated). Therefore, it seems that some reduction in the number of parameters is needed from a pragmatic point of view.

Secondly, and more importantly, in many situations one may want to impose substantive restrictions on the parameters, which will in effect lead to a reduction in the number of parameters. An obvious example of such a restriction is the requirement that a certain parameter equals a known constant. For example, it can be hypothesized that the range of nondecision time, s_t, equals zero for all conditions ($s_{t(c)} = 0$ for $c = 1, \cdots, C$). In this way, s_t has been dropped from the model (below it will be shown how it can be evaluated whether this restriction makes sense). Another popular substantive restriction in the context of the diffusion model is the requirement of a symmetric diffusion process ($z_{(c)} = a_{(c)}/2$ for $c = 1, \cdots, C$).

However, we can go a step further by carrying out a regression of the parameters onto a set of predictors. To explain this concept, assume that a researcher has set up a brightness discrimination task (Ratcliff & Rouder, 1998; see also Example 2 in this paper). Suppose furthermore that there are 33 levels of brightness defined by increasing the number of white pixels in each step with an equal number. For the moment, the focus will be on the drift rates. Not restricting the drifts in any way will lead to 33 drift parameters to be estimated. However, the researcher may want to test the hypothesis that the drift rate varies linearly with brightness level:

$$v_{(c)} = v^*_{(1)} + B_{(c)} v^*_{(2)}$$

where $B_{(c)}$ refers to the brightness level in condition c and $c = 1,\ldots, C$. In this example we have reduced the number of parameters to be estimated from 33 to 2. (Note also that we have introduced a new notation here: *basic* or *design parameters* are marked with a star.)

In general, we can write that drift rate in condition c can be decomposed into a weighted linear combination of M known predictor values:

$$v_{(c)} = \sum_{j=1}^{M} d_{cj} v^*_{(j)} \tag{1}$$

where d_{cj} is the value of the j^{th} predictor in condition c. In the aforementioned example, $M = 2$, $d_{c1} = 1$ and $d_{c2} = B_{(c)}$. Because we have C linear equations as in Equation (1) (one for each drift rate), we can make use of matrices and vectors to represent them all at once:

$$\boldsymbol{\psi}_{\mathrm{v}} = \begin{bmatrix} v_{(1)} \\ \vdots \\ v_{(c)} \\ \vdots \\ v_{(C)} \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^{M} d_{1j} v^{*}_{(j)} \\ \vdots \\ \sum_{j=1}^{M} d_{cj} v^{*}_{(j)} \\ \vdots \\ \sum_{j=1}^{M} d_{Cj} v^{*}_{(j)} \end{bmatrix} = \begin{bmatrix} d_{1j} & & \cdots & & d_{1M} \\ \vdots & \ddots & & & \vdots \\ d_{c1} & & d_{cj} & & d_{cM} \\ \vdots & & & \ddots & \vdots \\ d_{C1} & & \cdots & & d_{CM} \end{bmatrix} \times \begin{bmatrix} v^{*}_{(1)} \\ \vdots \\ v^{*}_{(c)} \\ \vdots \\ v^{*}_{(C)} \end{bmatrix} = \mathbf{D}_{v} \times \begin{bmatrix} v^{*}_{(1)} \\ \vdots \\ v^{*}_{(c)} \\ \vdots \\ v^{*}_{(C)} \end{bmatrix}$$

The *design matrix* $\mathbf{D}$ is a $C \times M$ matrix where each column represents a predictor (e.g., an intercept, an experimental treatment, a measured variable, etc.). The design matrix $\mathbf{D}$ is then multiplied with an $M \times 1$ *design parameter vector*, to recover a $C \times 1$ *model parameter vector* $\boldsymbol{\psi}$.

The idea of regressing the parameters onto a set of predictors can be applied to all parameters in the model and is by no means restricted to the drift rates. Because a different design matrix can be used for each parameter, **D** will be indexed with the parameter symbol in order to make it clear to which parameter the design corresponds. The entire parameter matrix $\mathbf{P}$ can be described in terms of only the seven (known) design matrices $\mathbf{D}$ and the seven design parameter vectors $\boldsymbol{\psi}$. The result is that, when fitting the model to the data, only the elements of the parameter vectors (as opposed to all the diffusion parameters) have to be estimated.

Two special and interesting cases of design matrices **D** are worth mentioning. The first special case is where **D** consists of a column of ones. This can be illustrated for the parameter T_{er} as follows:

$$\begin{bmatrix} T_{er(1)} \\ \vdots \\ T_{er(c)} \\ \vdots \\ T_{er(C)} \end{bmatrix} = \begin{bmatrix} 1 \\ \vdots \\ 1 \\ \vdots \\ 1 \end{bmatrix} \times T^{*}_{er(1)}$$

The result of this is that the *C* conditions have the same T_{er}. In a second special case, **D** equals the *C*×*C* identity matrix such that each of the *C* conditions has a different value for a certain parameter. In the case of an identity matrix as the design matrix, there is no restriction of parameters across conditions.

To illustrate the usefulness of the design matrix method, let us consider a final example. Suppose we want to fit a drift rate to the first condition and allow the drift rates of the other conditions to deviate from the first condition (but all in the same way: a "drift criterion change" from one condition to the next). This can be implemented by defining the design matrix

$$\mathbf{D}_v = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ \vdots & \vdots \\ 1 & 1 \end{bmatrix}$$

with $\boldsymbol{\psi}_v = \mathbf{D}_v \times \begin{bmatrix} v^{*}_{(1)} \\ v^{*}_{(2)} \end{bmatrix}$, and therefore $v_{(1)} = v^{*}_{(1)}$ and $v_{(c)} = v^{*}_{(1)} + v^{*}_{(2)}$ and for all $c \neq 1$ (see The SAS Institute, 1991, for some pointers on the construction of design matrices; Chapter 4).

In general, we formulate the parameter matrix $\mathbf{P} = \left\{ \mathbf{D}_a \times \mathbf{a}^{*}, \mathbf{D}_{T_{er}} \times \mathbf{T}^{*}_{\mathbf{er}}, \mathbf{D}_{\eta} \times \boldsymbol{\eta}^{*}, \mathbf{D}_z \times \mathbf{z}^{*}, \mathbf{D}_{s_z} \times \mathbf{s}^{*}_{\mathbf{z}}, \mathbf{D}_{s_t} \times \mathbf{s}^{*}_{\mathbf{t}}, \mathbf{D}_v \times \mathbf{v}^{*} \right\}$. Then, all the elements of $\mathbf{a}^*$, $\mathbf{T}^*_{\mathbf{er}}$, $\boldsymbol{\eta}^*$, $\mathbf{z}^*$, $\mathbf{s}^*_{\mathbf{z}}$, $\mathbf{s}^*_{\mathbf{t}}$, and $\mathbf{v}^*$ are the parameters over which we want to optimize the fit to data.

Creative use of design matrices allows one to impose substantive restrictions on parameter sets, and will enable researchers to test specific substantive hypotheses. Extending the diffusion model with the design matrix methodology, it becomes possible to build a kind of "analysis of variance/multiple regression"-diffusion model.

Using the design matrix method entails two restrictions, however. Firstly, only linear models can be represented by matrices. Secondly, only restrictions across conditions are possible, while restrictions across parameters (e.g., restricting *z* to be equal to *a*/2) requires a different strategy. Nonetheless, implementing restrictions using design matrices remains a very flexible and powerful tool.

Techniques for fitting the RDM

Finding the parameters of the Ratcliff Diffusion Model, given a data set, is something of a challenge. Before starting, several nontrivial choices need to be made, in particular regarding the objective function to use, how to deal with outliers and other contaminant reaction times, and the precise method of optimization. In this section, we discuss each of these choices, but for details we will refer the reader to Appendices A and B.

We will first briefly describe our outlier handling strategy. Then we will discuss the loss function and common difficulties encountered during optimization.

A crucial part of any algorithm to fit the diffusion model is the efficient computation of its cumulative distribution function (CDF). For this, we rely heavily on the methods described in Tuerlinckx (2004).

Outlier handling strategies

An important issue to consider when applying a statistical model to reaction time data is that of *contaminants* – data points that appear in the data sets but that are somehow not germane to the research question. A well-known class of contaminants is *outliers* (data points that are outside the range of normal observations), but other examples are random guesses (data from trials where the participant somehow missed the stimulus and guessed), delayed start-ups (where the participant was somehow inappropriately delayed in responding), and fast guesses (where the participant executed a response before having actually inspected the stimulus).

Each of these types of contaminants can severely muddy the data (Ratcliff, 1979; Ratcliff & Tuerlinckx, 2002; Ulrich & Miller, 1994), possibly resulting in biased parameter estimates and incorrect standard errors of estimation. A fitting procedure for a model such as this one should therefore always be equipped with a proper strategy for handling these contaminants. We opt for a combination of two methods: First, the data are preprocessed with an *exponentially weighted moving average* (EWMA) control method that gives the minimal reaction time necessary for inclusion in the data analysis and second, a mixture model is fitted to the data.

The EWMA method is used in a preprocessing step in order to filter out reaction times that are suspected of being fast guesses. The idea behind this method is that the identification of fast guesses is made possible because they tend to have a specific signature, being responses with a very short RT and chance level performance (whereas 'real' observations with low accuracy tend to be slow rather than fast). Working from these assumptions, we can formulate a statistical expectation for guesses, and then we can *find the minimal response time at which the data begin to deviate from this expectation*. This minimal response time is used as a lower cut-off value such that all observations with shorter RTs are censored. More technical detail regarding the method is provided in Appendix A.

The basic idea behind the *mixture model* (Ratcliff & Tuerlinckx, 2002) is that each trial has a probability $(1-\pi)\gamma$ of being a guess, a probability $(1-\pi)(1-\gamma)$ of being a 'delayed startup', and a remaining probability π of being an actual instance of a diffusion process. A trial can then be represented by the decision tree shown in Figure 3. The first step leads to either the diffusion process (with probability π) or to a contamination process (with probability $1-\pi$). A contamination process can in turn be a 'guess' (with probability γ) or a 'delayed startup' response (with probability $1-\gamma$). Note that this treatment adds two free parameters to the model (π and γ). Details concerning the component distributions and the mixture distribution are provided in Appendix A.

* Figure 3 about here *

The loss function

To estimate the best fitting parameters of the RDM (or the extended RDM), given a data set, we have to find the maximum of a likelihood function, or the minimum of some deviance function. For our loss function, we use a negative multinomial log-likelihood function (MLF). We opt for the MLF for several reasons, chiefly among them its computational tractability compared to continuous likelihood and robustness in the face of small measurement errors. Briefly, the loss is defined as -2 times the natural logarithm of the joint likelihood of observing the observed number of data points in each of a set of predefined "reaction time bins". We call this statistic Λ.

Details regarding Λ and its optimization are provided in Appendix B.

The Diffusion Model Analysis Toolbox

In one paper concerning the RDM, Wagenmakers, van der Maas, and Grasman (in press) note that for many experimental psychologists, the difficulty associated with fitting a diffusion model is "rather prohibitive." In an attempt to further popularize the diffusion model, we have developed a

MATLAB (version 2006a; ©1994 The MathWorks, Inc.) application, which should allow researchers with less technical background to use the diffusion model in practice. The program, which is called DMAT (for *Diffusion Model Analysis Toolbox*) can be freely downloaded from the website of the K.U.Leuven Research Group for Quantitative and Personality Psychology (http://ppw.kuleuven.be/okp/dmatoolbox).

Performance. In creating DMAT, we had two main goals in mind. The program should be (1) accurate and efficient and (2) user-friendly. We believe that we have achieved both goals to a satisfactory degree. Regarding accuracy and efficiency, DMAT performs well in simulations (see below) testing the recovery of model and design parameters from simulated data (biases are generally low and standard errors small), and on our desktop PCs, the algorithm typically converges in less than one minute's time. The program is developed to make use of all fitting and modeling strategies we have discussed above (and more).

Ease of use. Regarding flexibility and ease of use, we have added a graphical user interface (GUI) to accommodate users who are not familiar with MATLAB's text-based interface (though of course text mode is also available and offers more flexibility). Also, wherever we could, we have provided default settings that we believe will perform well in most cases, and we have written an instructional primer to the use of the toolbox (Vandekerckhove & Tuerlinckx, 2007b).

Statistical framework

Of course, finding the best fitting parameter set for the diffusion model given a dataset is more than a mere academic problem. At some point, end users of DMAT will want to draw substantive conclusions from their data. This is the realm of statistical inference or *model selection*. We distinguish between three types of tests: *testing a point estimate*, *comparing two nested models*, and *comparing non-nested models*.

Testing a point estimate

The simplest procedure that can be used tests whether a point estimate $\hat{\theta}$ of a parameter θ significantly deviates from some a-priori value θ_0. Under the null hypothesis that $\theta = \theta_0$, and under some regularity conditions, the Wald statistic $Z = \frac{\hat{\theta} - \theta_0}{\mathrm{se}_{\hat{\theta}}}$ (with $\mathrm{se}_{\hat{\theta}}$ the standard error of estimation) follows a standard normal distribution (Bishop, Fienberg, & Holland, 1975). One of the regularity conditions – the boundary condition – is noteworthy however. The Wald statistic should not be used if the test value θ_0 is at the boundary of the parameter space (Bishop et al., 1975, but see also Stram & Lee, 1994, 1995, for an adaptation of the reference distribution). As a consequence, it cannot be used to test the null hypothesis that, for example $\eta = 0$, since η is bounded at 0.

Testing the difference between two nested diffusion models

In many cases, however, it is more desirable to directly compare two model formulations against one another. For example, a researcher might want to test whether an experimental manipulation has some influence on drift rate. To that end, one could compare a model in which all drift rates are constrained to be equal to a model where they are free to vary over conditions (this would be analogous to an analysis-of-variance design). The former model – the null model – could be formulated as

$\mathbf{P}_0 = \left\{\mathbf{1_C}\times\mathbf{a}^*, \mathbf{1_C}\times\mathbf{T}^*_{\mathbf{er}}, \mathbf{1_C}\times\boldsymbol{\eta}^*, \mathbf{1_C}\times\mathbf{z}^*, \mathbf{1_C}\times\mathbf{s}^*_{\mathbf{z}}, \mathbf{1_C}\times\mathbf{s}^*_{\mathbf{t}}, \mathbf{1_C}\times\mathbf{v}^*\right\}$,

where $\mathbf{1_C}$ indicates a $C\times 1$ vector with all elements equal to 1 (*C* still being the number of conditions in the experiment). This model restricts all parameters to be equal across conditions, while the latter model – the effect model – could be this:

$\mathbf{P_E} = \left\{\mathbf{1_C}\times\mathbf{a}^*, \mathbf{1_C}\times\mathbf{T}^*_{\mathbf{er}}, \mathbf{1_C}\times\boldsymbol{\eta}^*, \mathbf{1_C}\times\mathbf{z}^*, \mathbf{1_C}\times\mathbf{s}^*_{\mathbf{z}}, \mathbf{1_C}\times\mathbf{s}^*_{\mathbf{t}}, \mathbf{I_{C\times C}}\times\mathbf{v}^*\right\}$, where the drift rates are now determined by the multiplication of the $C\times C$ identity matrix $\mathbf{I_{C\times C}}$ and the design parameter matrix $\mathbf{v}^*$. Thus, the restriction on drift rate *v* is now released and $C-1$ parameters have been added to the model (because $\mathbf{v}^*$ now contains *C* elements instead of 1).

If these two models are nested (i.e., the null model can be reached by implementing restrictions in the effect model, which is the case in this example), the resulting fit measures, $\Lambda_{\mathbf{P_0}}$ and $\Lambda_{\mathbf{P_E}}$, can be directly compared. Their difference follows a chi-square distribution with a number of degrees of freedom equal to the difference in number of parameters between the effect model and the null model: $\Delta\Lambda = \Lambda_{\mathbf{P_0}} - \Lambda_{\mathbf{P_E}} \sim \chi^2_{\Delta df}$, with in the case of this example $\Delta df = C - 1$.

Note, however, that the same boundary condition applies here as for the Wald statistic: if the parameter set $\mathbf{P_0}$ is at an edge of the parameter space where $\mathbf{P_E}$ is not, this statistic should not be used. Also, as always in null hypothesis significance testing, researchers should take care that their null model is a credible model – wildly inaccurate null models will always yield significant $\Delta\Lambda$ statistics.

Finally, it should be remarked that the distribution of $\Delta\Lambda$ assumes that fixed reaction time bins were used (see Appendix B). In the case where percentile-based bins were used, the chi-square assumption may not hold[1].

Comparing non-nested models

If two models are *not* nested, then their fit can still be compared by assessing the difference in *information criteria* such as the Small Sample Akaike Information Criterion (AICc; Hurvich & Tsai, 1989; Sugiura, 1978) or the Bayesian Information Criterion (BIC; Hoeting, Madigan, Raftery, & Volinski, 1999; Schwarz, 1978). In both cases, the model with lower value on the criterion is preferred. The two measures are defined as $\text{AICc} = \Lambda + d\left[2N/\left(N-d-1\right)\right]$ and $\text{BIC} = \Lambda + d\log(N)$, where *N* is the total number of data points and d indicates the number of free parameters in the model in question. Of course, these criteria can equally validly be applied to nested models.

DMAT is specifically built for comparisons across models like we describe here. From the start, the user is asked to provide a *queue* of models, each of which can then be fit to the data set and then compared.

Simulations

To test our algorithm, we performed many Monte Carlo simulations, of which we report only a selected few. Throughout, we use six standard parameter sets (A through F), which are reported in Table 1. Note that in these parameter sets, there are four conditions, across which all parameters are identical, except for drift rate, which systematically varies (there are ten columns; six with the parameters that are constant, and one for each condition's drift rate). We borrow these parameter sets from Ratcliff and Tuerlinckx (2002). Our simulation method was based on the rejection method described in Tuerlinckx, Maris, Ratcliff, and De Boeck (2001).

* Table 1 about here *

Asymptotic parameter recovery

As a first test of our algorithm, we used it to estimate RDM parameters, given exactly the expected distributions under many different parameter sets, including the ones in Table 1 (that is, we input the exact numbers of observations that each reaction time bin would have, given a certain parameter set). In each case, and for each objective function DMAT allows, the algorithm returned the exact parameter values to the requested accuracy.

Preasymptotic parameter recovery

As a second test of our algorithm, we performed a series of simple simulations to investigate biases and standard errors of the parameter estimates. We define the *relative bias* of each parameter as $\frac{\overline{\hat{\theta}}-\theta}{\theta}\times 100\%$, and the *standard error* as $\sqrt{\frac{1}{R-1}\sum_{j}^{R}\left(\hat{\theta}_j-\overline{\hat{\theta}}\right)^2}$, with *R* the number of replications, and $\hat{\theta}$ and $\overline{\hat{\theta}}$ respectively the estimate and the mean estimate of some parameter θ.

From each of the six parameter sets shown in Table 1, we generated 100 datasets with 250 data points in each condition (without outliers). We used DMAT to find parameter estimates and calculated relative biases and standard errors within each parameter set. As can be seen from Table 2, the simulation parameters are generally well recovered. The *a*, T_{er}, and *z* estimates tend to be within 10% of their simulation values. The relative bias of the *v* estimates are slightly larger for large values of the "true" *v*. As already seen in Ratcliff and Tuerlinckx (2002), the variance parameters tend to be slightly more difficult to estimate, in particular when they are small. The mixture model parameter π is very well estimated, but with only 250 data points per data set and only 5% of them in the contaminant distribution, estimates for γ are unavoidably unstable.

* Table 2 about here *

Outlier handling strategies

To test the outlier treatment procedure applied by our algorithm (see Appendix A), we performed four more series of simple simulations to evaluate the combined EWMA / mixture model approach. In each series of simulations, we again generated 100 datasets from each of the six parameter sets shown in Table 1, with 250 data points in each condition.

We employed a simple two-by-two design: we either added outliers to the simulated data or did not, and we either enabled the outlier treatment or did not. When we did add outliers, 2.5% were fast guesses (RT were draws from a uniform distribution between 200 and 400ms and accuracy was about 50%) and an additional 2.5% were delayed startups (RT draws from a uniform distribution between 500 and 3000ms, but with accuracy as expected under the diffusion model). We then estimated the parameters for each dataset with DMAT and compared parameter recovery. In Table 3, the results are shown for parameter set A. As can be seen, if the data set did contain outliers, and they are not treated by the algorithm, estimation biases increase dramatically, to over 100% for some drift values. When the combined EWMA / mixture model method is applied, relative biases return to the same magnitude as in the condition where no outliers existed.

To conserve space, we do not report results for the other parameter sets here, but as it turns out, our outlier treatment succeeds in alleviating the influence of outliers and contaminants on parameter estimates: biases and standard errors of the parameters that the adapted algorithm returned from the contaminated data set are closer to those of the parameters that the original algorithm returns from a "clean" data set, and they are lower than those from the original algorithm on the contaminated data set.

* Table 3 about here *

Power analyses

Power analysis 1. In another series of simulations, we evaluated the sensitivity (statistical rejection power) of our algorithm and associated inferential statistics. From each of the six parameter sets shown in Table 1, we again generated 100 datasets with 250 data points in each condition. Then we allowed DMAT to find the best parameter estimates, imposing two different design models. In the first design, we allowed for no parameter changes across conditions:

$\mathbf{P}_1 = \left\{\mathbf{1}_\mathbf{C} \times \mathbf{a}^*, \mathbf{1}_\mathbf{C} \times \mathbf{T}^*_{\mathbf{er}}, \mathbf{1}_\mathbf{C} \times \boldsymbol{\eta}^*, \mathbf{1}_\mathbf{C} \times \mathbf{z}^*, \mathbf{1}_\mathbf{C} \times \mathbf{s}^*_\mathbf{z}, \mathbf{1}_\mathbf{C} \times \mathbf{s}^*_\mathbf{t}, \mathbf{1}_\mathbf{C} \times \boldsymbol{\nu}^*, \mathbf{1}_\mathbf{C} \times \boldsymbol{\pi}^*, \mathbf{1}_\mathbf{C} \times \boldsymbol{\gamma}^*\right\}$ (note that we apply a design matrix to the mixture model parameters here).

In the second design, we allowed drift rates to vary across conditions. $\mathbf{P}_2 = \left\{\mathbf{1}_\mathrm{C} \times \mathbf{a}^*, \mathbf{1}_\mathrm{C} \times \mathbf{T}^*_\mathbf{er}, \mathbf{1}_\mathrm{C} \times \boldsymbol{\eta}^*, \mathbf{1}_\mathrm{C} \times \mathbf{z}^*, \mathbf{1}_\mathrm{C} \times \mathbf{s}^*_\mathbf{z}, \mathbf{1}_\mathrm{C} \times \mathbf{s}^*_\mathbf{t}, \mathbf{I}_{\mathrm{C}\times\mathrm{C}} \times \mathbf{v}^*, \mathbf{1}_\mathrm{C} \times \boldsymbol{\pi}^*, \mathbf{1}_\mathrm{C} \times \boldsymbol{\gamma}^*\right\}$.

Then, we computed the test statistic $\Delta\Lambda = \Lambda_{\mathbf{P}_1} - \Lambda_{\mathbf{P}_2} \sim \chi^2_{df=3}$ and its significance level. If our algorithm has sufficient sensitivity (statistical power), a large proportion of these statistics should be larger than the critical chi-square value. Failing to reject a null hypothesis that is truly false would be a Type II error. In fact, as it turns out, in 99% of these cases did a significant result emerge at all significance levels (down to $\alpha = 10^{-6}$). In only four cases (once in parameter set B and thrice in C) was the null hypothesis not rejected at the $\alpha = .01$ level. It can be concluded that, at least for these parameter sets and this sample size, this analysis has very high power – it is not overly conservative.

Type I error. However, the previous result begs the question of *selectivity*: Is it possible that the analysis would yield significant results, even where none were really present? To test this possibility, we repeated the same simulations, but now we changed the simulation parameters such that all drift rates were now constant across conditions. The drift rates were -.25, -.15, -.05, .05, .15, and .25 for parameter sets A through F, respectively. We repeated the same analyses as before, but now we would expect few of the test statistics to reach significance (under the null hypothesis, the *p*-values should follow a uniform distribution between 0 and 1). However, depending on the significance level α we select, sometimes the null hypothesis will be falsely rejected (Type I error). Figure 4 shows the proportion of such false rejections as a function of the threshold value α. As we can see, the proportion of false rejections is indeed approximately equal to α. However, a small trend upward of the diagonal is visible, indicating that our algorithm might be slightly too liberal in testing, and that it may be prudent to test at more restrictive significance levels in practical settings.

* Figure 4 about here *

Power analysis 2. Following up on the previous simulations, we set up a new series to determine how well DMAT is able to detect small differences in parameters. To that end, we simulated data sets with 250 data points in each of two conditions. In both conditions, all parameters were equal (and taken from parameter set A in Table 1), except for drift rate. Drift rate was always zero in the first condition, and in the second condition it was either also 0, or .02, .04, .06, .08, or .10. With each of those values for drift rate in the second condition, we generated 10,000 data sets and allowed DMAT to recover the parameter estimates, once with a model allowing no differences across the two conditions and once allowing drift rate to differ between the conditions. Then we calculated the $\Delta\Lambda$ statistic and computed associated *p*-values (under a χ^2-distribution with one degree of freedom). Figure 5 shows the proportion of rejected null hypotheses as a function of the threshold value α. There it can be seen that if the true drift rate in the second condition is 0.1, there are very many rejections of H_0, even with very conservative (small) values for α. Table 4 shows the proportion of rejections of H_0 for traditional values of α. At the traditional value of $\alpha = .05$, a drift rate of 0.1 is detected more than 96.3% of the time, but at a 6.1% risk of getting a "false alarm". If α is very conservative, there are no false alarms, but the test becomes much less sensitive, detecting a drift rate difference of 0.1 in only 18.5% of cases.

* Figure 5 about here *

* Table 4 about here*

Application examples

To demonstrate the use of DMAT, we obtained two data sets containing both accuracy and reaction time data (Experiment 3 in Vandekerckhove, Panis, & Wagemans, 2007, and Experiment 1 from Ratcliff & Rouder, 1998; used with permission). We will briefly describe the variables of interest, but for substantive details, the interested reader is referred to Vandekerckhove et al. (2007) and Ratcliff and Rouder (1998).

Example 1: An incomplete factorial ANOVA design

The experiment by Vandekerckhove et al. (2007) is in the domain of visual shape perception and change detection in particular. The basic effect of interest is that if observers are shown a succession of two 2D shapes which are different in only one vertex (an angle or curvature extreme), this difference is easier to detect if it is adding or removing a concavity than if it is adding or removing a convexity (Barenholtz, Cohen, Feldman, & Singh, 2003). The substantive research question in this experiment is: does the effect occur when the change is not adding or removing a *new* vertex, but increasing or decreasing an *existing one*? The paradigm was a two-interval forced choice (2IFC) task.

In the experiment, three variables are manipulated: (a) *change*: was there any difference between the two shapes? (b) *quality*: did the number of vertices change? (c) *type*: if there was a change, was it in a concavity (curvature with negative sign) or in a convexity (positive sign). As is obvious from variables (b) and (c), this is not a fully crossed design (properties of the change cannot be manipulated if there was none; as a result each 'change' condition had 24 data points, but each 'no-change' condition had 192). Table 5 lists all the conditions between which we would want to differentiate. Because the manipulations are all intended to affect the quality of the stimulus, we expect changes in drift rate, but not in any other variable. Writing the design as we do in Table 5 simplifies construction of a design matrix: The complete design matrix is simply the last three columns in the table, plus one column with ones for a grand mean (intercept).

* Table 5 about here*

The goal of this experiment (and thus the data-analysis) is twofold. Primarily, it was to find out whether the *type* variable contributes anything above and beyond the *quality* variable. Additionally, if *type* has an effect, we would want to know whether it is independent of *quality* (i.e., is there an interaction?). To this end, we defined a queue of five models, each an extended version of the former. In Model 1, we allowed for no effects of condition on any parameter:

$$\mathbf{P}_1 = \left\{\mathbf{1}_5 \times \mathbf{a}^*, \mathbf{1}_5 \times \mathbf{T}_{\mathbf{er}}^*, \mathbf{1}_5 \times \boldsymbol{\eta}^*, \mathbf{1}_5 \times \mathbf{z}^*, \mathbf{1}_5 \times \mathbf{s}_{\mathbf{z}}^*, \mathbf{1}_5 \times \mathbf{s}_{\mathbf{t}}^*, \mathbf{1}_5 \times \mathbf{v}^*, \mathbf{1}_5 \times \boldsymbol{\pi}^*, \mathbf{1}_5 \times \boldsymbol{\gamma}^*\right\}.$$

In Model 2, we let drift rates vary according to the design of the experiment, *without* the *type* variable, and in Model 3, we let drift rates vary according to the complete design of the experiment. These design matrices were $\mathbf{D}_{\mathbf{v2}} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & 1 & -1 \\ 1 & -1 & 0 \end{bmatrix}$ and $\mathbf{D}_{\mathbf{v3}} = \begin{bmatrix} 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & 1 \\ 1 & -1 & 0 & 0 \end{bmatrix}$, respectively. In Model 4, we allowed drift rates to vary freely across conditions: $\mathbf{D}_{\mathbf{v4}} = \mathbf{I}_{\mathbf{5\times5}}$, and finally, in Model 5, we allowed all diffusion parameters to vary freely across all five conditions:

$$\mathbf{P}_4 = \left\{\mathbf{I}_{5\times5} \times \mathbf{a}^*, \mathbf{I}_{5\times5} \times \mathbf{T}_{\mathbf{er}}^*, \mathbf{I}_{5\times5} \times \boldsymbol{\eta}^*, \mathbf{I}_{5\times5} \times \mathbf{z}^*, \mathbf{I}_{5\times5} \times \mathbf{s}_{\mathbf{z}}^*, \mathbf{I}_{5\times5} \times \mathbf{s}_{\mathbf{t}}^*, \mathbf{I}_{5\times5} \times \mathbf{v}^*, \mathbf{1}_5 \times \boldsymbol{\pi}^*, \mathbf{1}_5 \times \boldsymbol{\gamma}^*\right\}$$

.

Note that Model 1 has 9 free parameters, Model 2 has 11, Model 3 has 12, and Model 4 has 13, while Model 5 has 37. Note also that each model is nested in the previous. We ran this analysis for one participant in the experiment. Table 6 displays the fit statistics of each model in the queue, and shows the Λ and $\Delta\Lambda$ -statistics with their degrees of freedom.

* Table 6 about here*

From the table, we can conclude that the increase in model freedom from Model 1 to Model 2 was a success – the badness-of-fit measure significantly decreases (p = .0002). Going from Model 2 to Model 3 (adding the variable *type*) also significantly improved model fit (p = .0001). However, going to Model 4 (allowing deviation from the experimental design) was not a worthwhile step: allowing drift rates to vary freely does not significantly improve the fit of the model (p = .5434). Finally, while the step from Model 4 to Model 5 decreased the chi-square value, this decrease was again not significant when we take into consideration that 24 parameters had been added to the model (p = .9665). Considering this, and inspecting the *AICc* and *BIC* values, Model 3 earns our preference.

As for the substantive hypotheses, the finding that adding the *type* variable to the analysis significantly increased model fitness indicates that it influences drift rate, above and beyond the effect of the *quality* variable. Additionally, allowing for a more complex pattern than the experimental design (e.g., with interactions) did not lead to a better fit, indicating that for this participant, the experimental variables did not interact. All recovered parameters and their standard errors of estimation under Model 3 are given in Table 7.

* Table 7 about here *

However, results were not identical for all participants. Five out of ten showed the pattern discussed above. In two other cases, Model 3 did not provide a significantly better fit than Model 2, indicating no significant effect of *type*. In a further two cases, Model 4 *did* provide a significant improvement relative to Model 3, indicating interactions between experimental variables. In one final case, Model 4 performed better than Model 3, but Model 5 also performed better than Model 4, indicating influences on other parameters beside drift rate (with this participant, there seemed to be a large across-condition difference in boundary separation).

Example 2: A linear regression design

The experiment by Ratcliff and Rouder (1998) is a psychophysical study somewhat similar to the one in Example 1; in this case a brightness discrimination experiment. There were two manipulations of interest. Firstly, there was a speed-accuracy instruction (participants were either instructed to be fast or accurate) and secondly there was a manipulation of brightness, which increased with equal steps over 33 levels. However, because not all levels of the brightness variable had the same number of observations and in order to ensure a sufficient number of trials in each cell, we collected the five darkest and five brightest levels into one bigger level each, leaving 25 levels of the brightness variable. These two variables were completely crossed in a 2x25 design, yielding 50 conditions (we will say that conditions 1-25 have an 'accuracy' instruction and 26-50 have a 'speed' instruction). The task was a 2AFC procedure, whereby each subject was shown a stimulus and had to judge whether this stimulus was one from a 'bright' distribution or one from a 'dark' distribution. The number of trials varied across conditions from 61 to more than 200, with an average of about 150.

From the manipulations, we can expect two things. Firstly, we expect that the speed-accuracy instruction will have an effect on boundary separation. Secondly, we expect that as brightness of the stimulus increases, the drift towards the 'bright' response increases. Note that for this analysis, we change the interpretation of the model's upper and lower boundary. We will now say that a hit on the upper boundary leads to a 'bright' response, and on the lower leads to a 'dark' response. The drift rate is hence no longer a measure of ability to respond correctly, but of a tendency to respond 'bright' (and a negative drift rate can now reasonably occur, indicating a tendency to respond 'dark').

This new interpretation of participants' replies has an important consequence. Considering that the responses are not classified as 'correct' or 'incorrect', the assumption for the EWMA method that guesses are equally distributed across responses no longer holds. Accordingly, we switched off the EWMA preprocessing. For the same reason, the 'fast guesses' component of the mixture model is no longer a valid representation, thus we will assume that the weight γ is equal to zero.

To perform the analysis, we again defined a queue, again of five models, each a more complex version of the former. In Model 1, we allowed for no changes of parameters across conditions. In Model 2, we defined that there should be two different levels of the parameters *a*, *z*, and *s*$_z$: one for the conditions with accuracy instruction and one for those with speed instruction. To do this, we constructed the following design matrix for these parameters:

$\mathbf{D}_{\mathbf{a}} = \mathbf{D}_{\mathbf{z}} = \mathbf{D}_{\mathbf{s}_{\mathbf{z}}} = \begin{bmatrix} \mathbf{1}_{25} & \mathbf{0}_{25} \\ \mathbf{0}_{25} & \mathbf{1}_{25} \end{bmatrix}$, which has two columns with 25 ones and 25 zeros each. Additionally, we will allow *v* and η in this model to evolve linearly with the brightness manipulation, while allowing different regression slopes and intercepts for different speed-accuracy instructions:

$\mathbf{D}_{\eta} = \mathbf{D}_{\mathbf{v}} = \begin{bmatrix} \mathbf{1}_{25} & \mathbf{L} & \mathbf{0}_{25} & \mathbf{0}_{25} \\ \mathbf{0}_{25} & \mathbf{0}_{25} & \mathbf{1}_{25} & \mathbf{L} \end{bmatrix}$, where $\mathbf{L} = \begin{bmatrix} 3 \\ 6 \\ 7 \\ \vdots \\ 27 \\ 28 \\ 31 \end{bmatrix}$ represents the 25 brightness levels (with the first and last values adapted to reflect the average of the five groups that were pooled there). The other design matrices allow for no change across conditions:

$$\mathbf{P}_2 = \left\{ \mathbf{D}_{\mathbf{a}} \times \mathbf{a}^*, \mathbf{1}_{50} \times \mathbf{T}_{\mathbf{er}}^*, \mathbf{D}_{\eta} \times \boldsymbol{\eta}^*, \mathbf{D}_{\mathbf{z}} \times \mathbf{z}^*, \mathbf{D}_{\mathbf{s}_{\mathbf{z}}} \times \mathbf{s}_{\mathbf{z}}^*, \mathbf{1}_{50} \times \mathbf{s}_{\mathbf{t}}^*, \mathbf{D}_{\mathbf{v}} \times \mathbf{v}^*, \mathbf{1}_{50} \times \boldsymbol{\pi}^*, \mathbf{0}_{50} \times \boldsymbol{\gamma}^* \right\}$$

.

However, the restriction that drift rates should increase linearly with the brightness manipulation is hardly tenable, both on theoretical grounds (because performance has to be bounded somewhere) and due to opportunistic inspection of Ratcliff and Rouder's (1998) results. In fact, in their article, drift rate increases with brightness according to a sigmoid function. Thus, in Model 3, we add a quadratic, cubic and quartic component to the design, to emulate an s-shaped function. Now,

$\mathbf{D}_\eta = \mathbf{D}_v = \begin{bmatrix} \mathbf{1}_{25} & \mathbf{L} & \mathbf{L}^2 & \mathbf{L}^3 & \mathbf{L}^4 & \mathbf{0}_{25} & \mathbf{0}_{25} & \mathbf{0}_{25} & \mathbf{0}_{25} & \mathbf{0}_{25} \\ \mathbf{0}_{25} & \mathbf{0}_{25} & \mathbf{0}_{25} & \mathbf{0}_{25} & \mathbf{0}_{25} & \mathbf{1}_{25} & \mathbf{L} & \mathbf{L}^2 & \mathbf{L}^3 & \mathbf{L}^4 \end{bmatrix}$, where the exponents indicate the element-wise power function (i.e., each element of the vector **L** is taken to that power), which replaces the previous in the model construction:

$$\mathbf{P}_3 = \left\{\mathbf{D}_a \times \mathbf{a}^*, \mathbf{1}_{50} \times \mathbf{T}_{er}^*, \mathbf{D}_\eta \times \boldsymbol{\eta}^*, \mathbf{D}_z \times \mathbf{z}^*, \mathbf{D}_{s_z} \times \mathbf{s}_z^*, \mathbf{1}_{50} \times \mathbf{s}_t^*, \mathbf{D}_v \times \mathbf{v}^*, \mathbf{1}_{50} \times \boldsymbol{\pi}^*, \mathbf{0}_{50} \times \boldsymbol{\gamma}^*\right\}$$

.

In Model 4, we allowed drift rates and their intertrial variances to vary freely across conditions:

$$\mathbf{P}_4 = \left\{\mathbf{D}_a \times \mathbf{a}^*, \mathbf{1}_{50} \times \mathbf{T}_{er}^*, \mathbf{I}_{50\times50} \times \boldsymbol{\eta}^*, \mathbf{D}_z \times \mathbf{z}^*, \mathbf{D}_{s_z} \times \mathbf{s}_z^*, \mathbf{1}_{50} \times \mathbf{s}_t^*, \mathbf{I}_{50\times50} \times \mathbf{v}^*, \mathbf{1}_{50} \times \boldsymbol{\pi}^*, \mathbf{0}_{50} \times \boldsymbol{\gamma}^*\right\}$$

.

Finally, in Model 5, we allowed all parameters to vary freely across conditions.

These models have 8, 14, 20, 60, and 400 free parameters, respectively, and each model is again nested in the previous. As before, we ran these models for one participant. Table 8 displays the fit statistics of each model in the queue, and shows the Λ and $\Delta\Lambda$ -statistics with their degrees of freedom.

Figure 6 shows the drift rates according to models 2, 3, and 5 in function of brightness.

* Table 8 about here *

* Figure 6 about here *

As can be seen from the table, Model 2 outperforms Model 1 greatly, indicating significant deviations from the nil model. Model 3 also outperforms Model 2, indicating deviations from linearity (as is obvious from the figure as well). Moreover, Model 4 performs better than Model 3, indicating that the obtained fit to the drift rates still leaves to be desired (this, too, can be seen in the figure). Finally, Model 5 does not perform better than Model 4, indicating that across-condition trends in the diffusion parameters were sufficiently described by the design applied in Model 4. The *AICc* statistic (in Table 8) shows the same pattern. The *BIC* values, on the other hand, show a slight preference for Model 3.

All recovered parameters and their standard errors of estimation under Model 4 are given in Table 9.

* Table 9 about here *

Conclusions and recommendations

In the present paper, we investigated and further enhanced the practical applicability of the diffusion model for reaction time and accuracy data. For this, we explored four avenues of improvement. Regarding high numbers of parameters and interpretation of experimental effects, we suggested the use of design matrices in order to regress diffusion model parameters onto covariates from an experiment. Regarding the treatment of contaminant data points, we presented a compound method to curb their influence in the estimation of diffusion model parameters. This method proved a satisfactory remedy in our Monte Carlo simulations. We provided some details regarding the fitting algorithm. Simulation studies indicated good performance of this algorithm.

Regarding statistical inference, we propose $\Delta\Lambda$ as a generally applicable statistic. Simulations demonstrated proper small-sample behavior from this statistic.

We have succeeded in implementing the above techniques and present DMAT, a software tool that allows the methods to be applied in practice. Finally, we gave two examples where we demonstrated the use of DMAT and the associated statistical analyses in practice.

References

Barenholtz, E., Cohen, E., Feldman, J., & Singh, M. (2003). Detection of change in shape: an advantage for concavities. *Cognition,* 89, 1-9.

Bishop, Y., Fienberg, S., & Holland, P. (1975). *Discrete Multivariate Analysis: Theory and Practice.* Cambridge: MIT Press.

Brown, S., & Heathcote, A. (2003). QMLE: Fast, robust and efficient estimation of distribution functions based on quantiles. *Behavior Research Methods, Instruments & Computers,* 35, 485-492.

Busemeyer, J.R., & Townsend, J.T. (1993). Decision field theory: A dynamic-cognitive approach to decision making in an uncertain environment. *Psychological Review,* 100, 432-459.

Chandra, M. (2001). *Statistical Quality Control.* New York: CRC Press.

De Valois, R.L. & De Valois, K.K. (1988). *Spatial vision.* New York: Oxford University Press.

Diederich, A. (1997). Dynamic stochastic models for decision making under time constraints. *Journal of Mathematical Psychology,* 41, 260-274.

Donders, F.C. (1969). On the speed of mental processes. *Acta Psychologica,* 30, 412–431 [Translation of: Die Schnelligkeit psychischer Processe, first published in 1868].

Glover, S., & Dixon, P. (2004). Likelihood ratios: A simple and flexible statistic for empirical psychologists. *Psychonomic Bulletin & Review,* 11, 791-806.

Heathcote, A., & Brown, S. (2004). Reply to Speckman and Rouder: A theoretical basis for QML. *Psychonomic Bulletin & Review,* 11, 577.

Heathcote, A., Brown, S., & Mewhort, D. (2002). Quantile maximum likelihood estimation of response time distributions. *Psychonomic Bulletin & Review,* 9, 394-401.

Hoeting, J.A., Madigan, D., Raftery, A.E., & Volinski, C.T. (1999). Bayesian model averaging: a tutorial (with discussion). *Statistical Science,* 14, 382-417.

Hurvich, C.M. & Tsai, C-L. (1989). Regression and time series model selection in small samples. *Biometrika,* 76, 297-307.

LaBerge, D. (1962). A recruitment theory of simple behavior. *Psychometrika,* 27, 375-396.

Laming, D. (1968). Information theory of choice reaction time. New York: Wiley.

Link, S. & Heath, R. (1975). A sequential theory of psychological discrimination. *Psychometrika,* 40, 77-105.

Luce, D. (1986). *Response times.* New York: Oxford University Press.

Nelder, J.A. & Mead, R. (1965). A Simplex Method for Function Minimization. *Computer Journal,* 7, 308-313.

Pike, A. (1966). Stochastic models of choice behavior: Response probabilities and latencies of finite Markov chain systems. *British Journal of Mathematical & Statistical Psychology,* 19, 15-32.

Pike, A. (1973). Response latency mechanisms for signal detection. *Psychological Review,* 80, 53-68.

Ratcliff, R. (1978). A theory of memory retrieval. *Psychological Review,* 85, 59-108.

Ratcliff, R. (1979). Group reaction time distributions and an analysis of distribution statistics. *Psychological Bulletin,* 86, 446-461.

Ratcliff, R. (1981). A theory of order relations in perceptual matching. *Psychological Review*, 88, 552-572.

Ratcliff, R. (1985). Theoretical interpretations of speed and accuracy of positive and negative responses. *Psychological Review*, 92, 212-225.

Ratcliff, R. (1987). More on the speed and accuracy of positive and negative responses. *Psychological Review*, 94, 277-280.

Ratcliff, R. (1988). Continuous versus discrete information processing: Modeling the accumulation of partial information. *Psychological Review*, 95, 238-255.

Ratcliff, R. (2002). A diffusion model account of reaction time and accuracy in a brightness discrimination task: Fitting real data and failing to fit fake but plausible data. *Psychonomic Bulletin & Review,* 9, 278–291.

Ratcliff, R., Gomez, P., & McKoon, G. (2004). A diffusion model account of the lexical-decision task. *Psychological Review*, 111, 159-182.

Ratcliff, R., & Rouder, J.N. (1998). Modeling response times for two-choice decisions. *Psychological Science,* 9, 347-356.

Ratcliff, R., & Rouder, J.N. (2000). A diffusion model account of masking in letter identification. *Journal of Experimental Psychology: Human Perception & Performance,* 26, 127-140.

Ratcliff, R., Thapar, A., & McKoon, G. (2004). A diffusion model analysis of the effects of aging on recognition memory. *Journal of Memory & Language*, 50, 408-424.

Ratcliff, R., & Tuerlinckx, F. (2002). Estimating parameters of the diffusion model: Approaches to dealing with contaminant reaction times and parameter variability. *Psychonomic Bulletin & Review,* 9, 438-481.

Ratcliff, R., Van Zandt, T., & McKoon, G. (1999). Connectionist and diffusion models of reaction time. *Psychological Review,* 106, 261-300.

Roberts, S. (1959). Control chart tests based on geometric moving averages, *Technometrics*, 1, 239-250.

Roe, R.M., Busemeyer, J.R., & Townsend, J.T. (2001). Multialternative decision field theory: A dynamic connectionist model of decisionmaking. *Psychological Review,* 108, 370-392.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics,* 6, 461-464.

Smith, P.L. (1995). Psychophysically principled models of visual simple reaction time. *Psychological Review,* 102, 567-591.

Smith, P.L., & Vickers, D. (1988). The accumulator model to two-choice discrimination. *Journal of Mathematical Psychology,* 32, 135-168.

Speckman, P., & Rouder, J. (2004). A comment on Heathcote, Brown, and Mewhort’s QMLE method for response time distributions. *Psychonomic Bulletin & Review,* 11, 574-576.

Stram, D.O., & Lee, J.W. (1994). Variance components testing in the longitudinal mixed effects model. *Biometrics,* 50, 1171–1177.

Stram, D.O. & Lee, J.W. (1995). Correction to "Variance components testing in the longitudinal mixed effects model." *Biometrics,* 51, 1196.

Strayer, D. L., & Kramer, A. F. (1994). Strategies and automaticity: I. Basic findings and conceptual framework. *Journal of Experimental Psychology: Learning, Memory, & Cognition,* 20, 318-341.

Sugiura, N. (1978). Further analysis of the data by Aikaike's information criterion and the finite corrections. *Communications in Statistics, Theory & Methods,* A7, 13-26.

Thapar, A., Ratcliff, R., & McKoon, G. (2003). A diffusion model analysis of the effects of aging on letter discrimination. *Psychology & Aging*, 18, 415-429.

Townsend, J. (1990). Serial vs. parallel processing: Sometimes they look like Tweedledum and Tweedledee but they can (and should) be distinguished. *Psychological Science,* 1, 46–54.

Townsend, J., & Ashby, F. (1983). *The stochastic modeling of elementary psychological processes.* Cambridge: Cambridge University Press.

Tuerlinckx, F. (2004). The efficient computation of the distribution function of the diffusion process. *Behavior Research Methods, Instruments, & Computers*, 36, 702-716.

Tuerlinckx, F., Maris, E., Ratcliff, R., & De Boeck, P. (2001). A comparison of four methods for simulating the diffusion process. *Behavior Research Methods, Instruments, & Computers,* 33, 443-456.

Ulrich, R., & Miller, J. (1994). Effects of truncation on reaction time analysis. *Journal of Experimental Psychology: General,* 123, 34-80.

Vandekerckhove, J., Panis, S., & Wagemans, J. (2007). On the nature of the concavity effect: The advantage for concavities is a compound of local and global effects. Manuscript under revision.

Vandekerckhove, J., & Tuerlinckx, F. (2007a). *The Diffusion Model Analysis Toolbox.* [Software and manual]. Retrieved from http://ppw.kuleuven.be/okp/dmatoolbox.

Vandekerckhove, J., & Tuerlinckx, F. (2007b). Diffusion Model Analysis with MATLAB: a DMAT primer. Conditionally accepted. *Behavior Research Methods.*

Vickers, D. (1970). *Decision processes in visual perception.* New York: Academic Press.

Wagenmakers, E.-J., van der Maas, H., & Grasman, R. (in press). An EZ-diffusion model for response time and accuracy. *Psychonomic Bulletin & Review*.

Authors' note

This research was supported by grants GOA/00/02-ZKA4511, GOA/2005/04-ZKB3312, and IUAP P5/24. The authors wish to thank Jeff Rouder for providing us with the data used in the second application example and Eric-Jan Wagenmakers, Robert Nosofsky and Andrew Heatcote for helpful comments on earlier drafts of this article.

Notes

1. We thank Andrew Heathcote for alerting us to this caveat. See Speckman and Rouder (2004) for a short discussion on the Quantile Probability Product estimator.

Tables

Table 1

Standard parameter sets used in the simulations. See text for details.

Parameter Set	a	T_{er}	η	z	s_z	s_t	v_1	v_2	v_3	v_4
A	0.08	0.30	0.08	0.04	0.02	0.02	0.40	0.25	0.10	0.00
B	0.08	0.30	0.16	0.04	0.02	0.02	0.40	0.25	0.10	0.00
C	0.16	0.30	0.08	0.08	0.02	0.02	0.30	0.20	0.10	0.00
D	0.16	0.30	0.16	0.08	0.02	0.02	0.30	0.20	0.10	0.00
E	0.16	0.30	0.08	0.08	0.10	0.10	0.30	0.20	0.10	0.00
F	0.16	0.30	0.16	0.08	0.10	0.10	0.30	0.20	0.10	0.00

Table 2

Recovery of simulation parameters by DMAT. Relative biases are in the upper half, standard errors (multiplied by 1,000) in the lower half

Parameter Set	a	T_{er}	η	z	s_z	s_t	v_1	v_2	v_3	v_4†	π‡	γ‡
A	-5	3	-60	-3	12	14	15	3	0	0	1	-55
B	-7	2	-48	-5	-38	-37	0	-7	-9	0	1	-13
C	-1	6	0	1	90	-10	20	10	6	0	2	-17
D	-3	9	9	-2	108	-1	25	15	11	0	0	8
E	3	2	4	3	4	-31	15	8	8	0	3	-21
F	-1	3	0	0	-2	-26	8	5	3	0	2	-4
A	4	8	46	2	18	16	72	32	18	16	29	347
B	4	6	70	2	17	13	51	34	26	22	24	407
C	11	29	48	5	31	51	113	61	31	6	61	472
D	19	28	91	9	45	46	191	89	47	18	52	418
E	13	21	48	7	29	42	71	45	24	8	29	464
F	20	23	80	10	44	42	90	61	33	12	38	436

†: Biases for v_4 are actual recovered values, not relative biases (since the true values were 0); ‡: True values for π and γ were .95 and .50, respectively.

Table 3

DMAT's recovery of simulation parameters from either "clean" or outlier-contaminated data (generated from parameter set A in Table 1), with our outlier correction method either enabled or disabled. Relative biases are in the upper half, standard errors (multiplied by 1,000) in the lower half

Outliers added	Outliers treated	a	T_{er}	η	z	s_z	s_t	v_1	v_2	v_3	v_4†	π†	γ†
No	No	2	1	4	3	25	-40	10	4	4	0		
Yes	No	62	3	513	55	471	191	112	110	166	0		
No	Yes	-5	3	-60	-3	12	14	15	3	0	0	0.96	0.22
Yes	Yes	0	2	-14	1	44	-3	6	2	0	0	0.94	0.05
No	No	4	7	64	2	21	13	71	40	23	21		
Yes	No	6	8	24	4	18	11	49	116	35	47		
No	Yes	4	8	46	2	18	16	72	32	18	16	29	347
Yes	Yes	3	8	63	2	20	23	60	37	25	16	6	90

†: Bias values for this parameter are actual recovered values, not relative biases.

Table 4

Proportion of rejected null hypotheses for different levels of the α criterion, and with different 'real' effects in the data. Real differences in drift rate of 0.1 are detected in 96.28% of cases at the .05 significance level, but at that α-level there are also 6.07% false alarms

	Simulated difference in drift rate					
α	0	0.02	0.04	0.06	0.08	0.1
0.05	0.0607	0.1454	0.3778	0.6649	0.8769	0.9628
0.01	0.0151	0.0503	0.1848	0.4452	0.7297	0.9062
0.0001	0.0002	0.0023	0.0153	0.0819	0.2586	0.5388
0.00001	0	0.0009	0.0044	0.0285	0.1244	0.336
0.000001	0	0.0005	0.0007	0.0091	0.0536	0.1848

Table 5

Design of Experiment 3 in Vandekerckhove et al. (2007)

Condition	Change	Quality	Type
1	1	1	-1
2	1	1	1
3	1	-1	-1
4	1	-1	1
5	-1	0	0

Table 6

Fit statistics from the model queue for one participant (Example 1)

Model	Λ	*df*	$\Delta\Lambda$	Δdf	*p*	*AICc*	*BIC*
1	2414.32	9				2432.57	2471.64
2	2401.96	11	12.36	2	0.0002	2424.34	2472.03
3	2387.49	12	14.46	1	0.0001	2411.96	2463.93
4	2387.13	13	0.37	1	0.5434	2413.67	2469.93
5	2374.15	37	12.98	24	0.9665	2453.02	2609.83

Table 7

Recovered model parameters and associated standard errors (*SE*s) for Example 2, under Model 3, for one participant.

	a	T_{er}	η	z	s_z	s_t
Values	0.1712	0.3160	0.2315	0.0863	0.0003	0.0001
*SE*s	0.0277	0.0131	0.0582	0.0139	0.0538	0.0318

Table 8

Fit statistics from the model queue for one participant (Example 2)

Model	Λ	*df*	ΔΛ	*df*	*p*	*AICc*	*BIC*
1	35089.6	8				35105.6	35161.4
2	22764.5	14	12325.1	6	0.000	22792.5	22890.1
3	22467.1	20	297.37	6	0.000	22507.2	22646.6
4	22339.3	60	127.81	40	0.000	22460.2	22877.7
5	22125.7	400	213.59	340	1.000	22968.3	25715

Table 9

Recovered parameters and standard errors (*SE*s) for Example 2, under Model 4. Results for the speed-stressed condition are in the top two rows, results for the accuracy-stressed condition are in the bottom two rows

Condition		a	T_{er}	η	z	s_z	s_t
Speed	Values	0.0399	0.3058	0.0382	0.0178	0.0305	0.1470
Speed	*SE*s	0.0011	0.0016	0.0124	0.0008	0.0024	0.0025
Accuracy	Values	0.1717	0.3058	0.0382	0.0827	0.0239	0.1470
Accuracy	*SE*s	0.0035	0.0016	0.0124	0.0019	0.0152	0.0025

Appendix A: Outlier Treatment Methods

Because of the detrimental influence contaminants can have on the quality of parameter estimates, DMAT is equipped with several strategies for the treatment of outliers and other contaminants. Two of these treatments are somewhat non-standard and require some clarification.

Exponentially weighted moving average filter

The *exponentially weighted moving average* method (EWMA; Chandra, 2001; Roberts, 1951) is a statistical quality control method that can detect shifts in performance as reaction times increase. A cut-off threshold is set where the performance is judged to be above chance level.

The first step in the application of the method is sorting the reaction times from short to long. In effect, we will then look at our data set as if it described a binary process that unfolds (and changes) over time. As time progresses (i.e., RT increases), the process will start to shift away from its 'control state' (with 50% accuracy) and tend toward a biased process (with accuracy > 50%). The *control process* describes our expectation regarding fast guesses, which is straightforward: Guesses are draws from a Bernoulli process at chance level. Formally, if the s$^{\text{th}}$ observation (that is, the response $X_{(s)}$, corresponding to the s^{th} sorted reaction time $T_{(s)}$) is a guess, then $X_{(s)} \sim$ Bernoulli(0.5). (It should be noted that the control process should be a credible representation of fast guesses, otherwise this method will not work.)

To determine the minimal response time at which the system no longer follows this control process, we take the reaction times from all conditions (all RTs still sorted fast to slow), and then analyze their corresponding responses. Of these responses, we iteratively compute the EWMA statistic $c_s = \lambda x_s + (1 - \lambda) c_{s-1}$, where $x_s = 1$ if the response corresponding to the s^{th} sorted reaction time was correct and 0 otherwise, and $\lambda \in (0,1]$ is a *weight parameter* which controls how many of the last observations are used. If λ is 1 only the s^{th} observation is used and if λ approaches 0, all observations from the first to the s^{th} are weighted equally. We will then, at each iteration, calculate the upper control limit (*UCL*) of this process, and check if the EWMA statistic c_s exceeds this value.

In practice, some constants need to be defined. The first is the *in-control mean* of the process, which in this context represents the expected average performance of a fast guess. We denote this parameter c_0, and initialize it to 0.5. Second is the *in-control standard deviation* σ_0 (standard deviation of *X*), which is also equal to 0.5 (this follows from the properties of the Bernoulli distribution). A third constant for EWMA is the *weight parameter* λ. We choose $\lambda = 0.01$, thereby accounting for many previous data points. The final constant is the *width of the control limits* (in standard deviations). To ensure a sensitive test, we set *L* to 1.5 (a relatively low value).

Given these parameters, we now compute c_s and check if it is smaller than the upper control limit: $c_s < UCL_s = c_0 + L\sigma_0\sqrt{\frac{\lambda}{2-\lambda}\left[1-(1-\lambda)^{2s}\right]}$. If this inequality is true, then the process is judged to be within the limits of the control model, and we label observation *s* as a 'fast guess'. When the upper control limit is exceeded, we decide that the probability of giving a correct response significantly exceeds 0.5 from this reaction time on, and stop the iteration process. The reaction time at which the *UCL* was breached is then taken as the threshold, and all reaction times below it are censored.

The EWMA method is commonly illustrated with a *control chart*, which depicts the evolution of c_s in function of increasing reaction time. Figure A1 shows an example control chart, with the EWMA statistic indicated by a full line, the control state by a dotted line and the control limits by a shaded region around the control state. This control chart is based on data that were generated from the parameters shown in Table 1 (set A), with 250 data points in each condition, and 5% fast outliers added to the 200-400ms domain, uniformly distributed and with 50% accuracy. The EWMA algorithm returns a cut-off value of 322ms, which is reasonable considering that the diffusion process with these parameters starts around 300ms, but there are contaminants between 200 and 400ms.

* Figure A1 about here *

Mixture model approach

Ratcliff and Tuerlinckx (2002) use a *mixture model* approach to explicitly model contaminant reaction times. In that extended model, any observation has a probability π of having originated from a diffusion model, and a complementary probability $1-\pi$ of having originated from a contaminant distribution, which is uniform between the minimal and maximal reaction time recorded. A further extension of this mixture model can be made in which a distinction is made between a 'guesses' and a 'delayed startup' distribution.

The mixture distribution then consists of three components, which are easily represented as a decision tree with two levels (see Figure 3 in the main text). The 'guess' component, with total probability $(1-\pi)\gamma$, contains uniformly distributed response times with a 50% accuracy rate, thus representing guesses. The 'delayed startup' component, with total probability $(1-\pi)(1-\gamma)$, represents trials of an experiment in which a participant has in fact responded according to his or her ability, but due to some external circumstance (e.g., momentary distraction or pushing a wrong button) is slowed down in responding. Under this component distribution, the probability of a correct response is equal to that under the diffusion model, $\Pr(X = x \mid \boldsymbol{\theta})$. Reaction times for these contaminant components, however, are uniformly distributed over the entire observed range.

The CDF of this extended diffusion model is:

$$\begin{aligned} F_{X,T}(x,t,\boldsymbol{\theta}) = {} & \pi \mathrm{Diff}(x,t,\boldsymbol{\theta}) \\ & + (1-\pi)\gamma \tfrac{1}{2} U(t,T_{-},T_{+}) \qquad , \\ & + (1-\pi)(1-\gamma)\Pr(X = x \mid \boldsymbol{\theta}) U(t,T_{-},T_{+}) \end{aligned}$$

where $U\left(t, A, B\right)$ indicates the cumulative density function of a uniform distribution from *A* to *B*, evaluated at *t*. $\mathrm{Diff}_{X,T}(x,t,\boldsymbol{\theta})$ is the joint probability that the response equals *x* (*x* = 0 for an error and *x* = 1 for a correct response) and that the response is given at time *t* or before, under a Ratcliff Diffusion Model with parameter vector $\boldsymbol{\theta}$ (thus, $\mathrm{Diff}_{X,T}(x,t,\boldsymbol{\theta}) = \Pr(X = x, T \leq t \mid \boldsymbol{\theta})$). The exact formula for this joint probability is given in Ratcliff and Tuerlinckx (2002). If *t* becomes very large, then $\mathrm{Diff}_{X,T}(x,t,\boldsymbol{\theta})$ approaches $\Pr(X = x \mid \boldsymbol{\theta})$, which is the marginal probability of the response *x*. Further, T_{-} and T_{+} are the minimum and maximum of the assumed response time distributions for contaminants. Technically, T_{-} and T_{+} are parameters, but in the remainder of this paper we will not treat them as such. They are not included in the parameter estimation routine, but are directly estimated with the observed minimum and maximum response times (for each condition and each participant), respectively.

The parameter π is the probability of observing a 'good' data point (one that is a realization of a Ratcliff diffusion process) and γ is the probability that a given contaminant is a guess (as opposed to a delayed startup).

Appendix B: Minimizing the Multinomial Log-Likelihood Function

Loss function

DMAT uses a multinomial likelihood function (MLF), which expresses the likelihood of observing a certain proportion of responses in a given number of reaction time bins, and should therefore be maximized in order to find good parameter estimates.

To define *B* reaction time bins, we select *B-1* monotonically increasing bin edges $q_1,...,q_{B-1}$ and define $q_0 = 0$ and $q_B = +\infty$. The observed frequency in bin *b*, in condition *c*, for response *x*, is then simply

$O_{cxb} = \sum_{j=1}^{n_{cx}} I(q_{b-1} < t_{cxj} \leq q_b)$, with 1,…,*j*,…,$n_{cx}$ data points with response *x* in condition *c*, $1,...,b...,B$ bins, and $I(\cdot)$ the indicator function (which takes the value 1 if its argument is true and 0 otherwise). The predicted (or expected) proportions in bin *b* of condition *c*, for response *x*, are given by $P_{cxb} = F_{X,T}(x, q_b, \boldsymbol{\theta}_c) - F_{X,T}(x, q_{b-1}, \boldsymbol{\theta}_c)$, where $\boldsymbol{\theta}_c$ indicates the parameter vector for condition *c*, and $F_{X,T}$ is the CDF of the RDM (or of the extended diffusion model, as defined in Appendix A).

The MLF that needs to be maximized, for *C* conditions, is then defined as $L = \prod_{c=1}^{C} \prod_{x=0}^{1} \prod_{b=1}^{B} P_{cxb}^{O_{cxb}}$, or, equivalently, as

$$\Lambda = -2 \cdot \log(L) = -2 \cdot \sum_{c=1}^{C} \sum_{x=0}^{1} \sum_{b=1}^{B} O_{cxb} \log\left(P_{cxb}\right), \qquad \text{(B1)}$$

which has an optimum (in this case a minimum) at the same point in the parameter space as L, but is computationally more attractive. Note that as the number of bins *B* increases (to the point where each bin has at most one observation), Λ becomes equal to the continuous log-likelihood function (Heathcote & Brown, 2004). We will hereafter refer to Equation (B1) as 'the' multinomial (log)likelihood function (MLF). During parameter estimation this will be the loss function we will be minimizing.

An alternative to the MLF described above is the more common *chi-square loss function* (as described by Ratcliff & Tuerlinckx, 2002), which expresses a weighted discrepancy between predicted and observed frequencies of observations in a given number of data bins. Using the same notation as above, the chi-square loss function that needs to be minimized is defined as $\mathrm{X}^2 = \sum_{c=1}^{C}\sum_{x=0}^{1}\sum_{b=1}^{B}\frac{\left(O_{cxb} - n_{cx}\cdot P_{cxb}\right)^2}{n_{cx}\cdot P_{cxb}}$. DMAT allows the user to choose between these two, but the MLF is the default option.

Fixed bins versus percentiles.

Speckman and Rouder (2004), commenting on the grouped maximum likelihood method as implemented by Heathcote, Brown, and Mewhort (2002; these authors call it *quantile maximum likelihood estimation* [QMLE] or *quantile probability products* [QPP]; see Brown & Heathcote, 2003; Heathcote & Brown, 2004; Speckman & Rouder, 2004), highlighted another important consideration that we need to make. Speckman and Rouder note that Heathcote et al. (2002) made use of *data-dependent bin edges* (they selected as bin edges a series of order statistics: the 10th, 30th, 50th, 70th, and 90th percentiles, thus making the actual values of the bin edges dependent on the observed data). As a result of this, the *observed frequencies* in each of the six bins are fixed (to $N/10$, $N/5$, $N/5$, $N/5$, $N/5$, and $N/10$, respectively), and the *locations* of the bin boundaries are random variables. However, the true multinomial likelihood distribution (from which (B1) derives) is only appropriate if the boundaries are fixed and the frequencies are random. Even though Speckman and Rouder (2004) note that the percentile-based method seems to have good pre-asymptotic properties, it remains, in their words, "estimation without a safety net" (p.576).

For this reason, we have adapted the loss function to operate not on *percentile-based bins*, but on *fixed bins*. To find bin boundaries that would be appropriate, we calculated the 10th, 30th, 50th, 70th, and 90th percentiles of 1,000 simulated datasets, based on each of six typical parameter sets (see Table 1). We found that these percentiles were generally centered around 300ms, 360ms, 420ms, 520ms, and 800ms for correct responses, and 380ms, 470ms, 560ms, 700ms, and 1,000ms for error reaction times. Throughout, we use these values as bin boundaries when calculating the MLF. This resolves the issue of the "strange multinomial likelihood" that Speckman and Rouder (2004) reported.

Here, too, DMAT allows users to choose between the default fixed bins, user-supplied fixed bins, or "data-snooped" percentile-based bins (for both the chi-square and the MLF loss functions).

Parameter space

A final choice that has to be made concerns the boundaries of the parameter space. Considering that some parameters of our diffusion model cannot take all possible real values, we may run into numerical difficulties during optimization. There are several possible strategies to avoid these problems, but in our algorithm we simply define that $\Lambda_{\boldsymbol{\theta}} = 10^{10}$ if the parameter set $\boldsymbol{\theta}$ is outside the allowable parameter space. This high penalty value ensures that our optimization algorithm will not stray outside the parameter space. However, it does introduce large discontinuities in the objective function, which we will need to take into account during the optimization phase.

Optimization

Minimizing the MLF as a function of the design parameters requires the use of an iterative optimization algorithm. Equation (B1) is a complicated nonlinear function of the data and the diffusion model parameters and it is not possible to find explicit formulas to obtain optimal parameter estimates from the data in one step.

To improve the optimization process, we make use of three strategies. First, we use a method of moments to produce a good *initial estimate* of the parameters (the *EZ-diff algorithm*; Wagenmakers et al., in press). Second, we have developed a defensive *generative algorithm* that is both efficient and accurate. Third, we incorporate a strategy for identifying and escaping suspected *local minima*.

Starting guesses. Starting an iterative optimization algorithm near to the optimum will decrease time to convergence. If the user of DMAT did not provide the algorithm with a starting guess, DMAT uses the *EZ-diff algorithm* (see below) to generate a plausible starting point for the first model in the queue. For subsequent models in a queue, DMAT uses the final estimates of the previous model as starting point (if necessary, a linear transformation or regression will be applied to ensure that this initial guess does not violate any restrictions of the present model).

The *EZ-diff algorithm* was introduced by Wagenmakers et al. (in press). Under the assumptions of no trial-to-trial variability ($\eta = s_t = s_z = 0$) and no participant bias ($z = a/2$)), they derived the following system of equations:

$$\begin{cases} v = \text{sign}(P_{corr} - \frac{1}{2}) \times s \times \left\{ \frac{\text{logit}(P_{corr})}{V_{RT}} \times \left[P_c^2 \times \text{logit}(P_{corr}) - P_{corr} \times \text{logit}(P_{corr}) + P_{corr} - \frac{1}{2} \right] \right\}^{\frac{1}{4}} \\ a = \frac{s^2}{v} \times \text{logit}(P_{corr}) \\ T_{er} = M_{RT} - \frac{a}{2v} \times \frac{1 - \exp(-va/s^2)}{1 + \exp(-va/s^2)} \end{cases}$$

,

where M_{RT} and V_{RT} are, respectively, the mean and variance of the correct responses' reaction time distribution, P_{corr} is the proportion correct answers, and $\text{logit}(P_{corr}) = \log\left[P_{corr}/(1 - P_{corr})\right]$. By convention, *s* is equal to 0.1. Given this system of equations, we are able to make reasonable guesses regarding the estimates for *v*, *a*, and T_{er}, separately for each condition. For the remaining parameters, we make the following (arbitrary) guesses: $\eta = 0.2$, $z = \frac{1}{2}a$, $s_z = \frac{9}{20}a$, and $s_t = \frac{9}{10}T_{er}$. As before, if necessary linear regression techniques will be applied to ensure that the starting point does not violate model restrictions (e.g., if the model is formulated such that a parameter should be in a linear relation with a covariate, the least-squares regression estimates of the parameter will be used as starting point).

Optimization algorithm. To locate the minimum of the MLF, we use an iterative optimization algorithm with several phases, each of which brings the process closer to the global minimum. The algorithm is based on the Nelder-Mead Simplex algorithm (NMS algorithm; Nelder & Mead, 1965), with a few adaptations.

The main part of the algorithm is a series of runs of the NMS algorithm. This algorithm has several interesting properties that we can use to our advantage. In particular, it is considered robust against discontinuities in the objective function, and it does not require that we provide analytical derivatives of the MLF (which would be time-intensive to compute). The NMS algorithm involves creating a so-called simplex (a shape with *N*+1 vertices enclosing an *N*-dimensional subspace), which is moved across the objective function in a "downward" direction. As the algorithm progresses and draws near to the minimum, the simplex shrinks more and more, until its vertices are sufficiently close to each other to be said to occupy a single point in the parameter space. In our algorithm, we allow a single NMS run to proceed for 200 steps, after which the size of the simplex shape is reset to its original size (the shrinkage is undone). We do this because we have observed that (due to numerical issues) our objective function sometimes converges in a local minimum. Resetting the simplex size allows the algorithm to escape from local minima.

Normally, we reset the simplex size three times, thus performing four runs with maximally 200 steps. When these are finished, we start a fifth, longer, NMS run with maximally 5,000 iterations. Usually, however, the last NMS run converges before that. DMAT users can change the number of NMS runs, as well as the maximum number of iterations allowed.

The final phase of the algorithm is a single quasi-Newton step, where the first and second derivatives of the objective are numerically approximated and used to find the local minimum near the point where the NMS run converged. If the NMS algorithm converged, this final phase does little to improve the accuracy of the parameter estimations. However, it is a convenient procedure that provides us with a numerical approximation to the Hessian matrix (the matrix of second derivatives) at the minimum, which is then used to verify that the solution point is in fact a minimum, and to calculate estimates of parameter standard errors.

Known local minima. The fact that the algorithm has converged to a minimum is still no guarantee that we have in fact found optimal parameter estimates. A better parameter set might still exist in a region that our algorithm has not visited. This is a very difficult problem and it is not possible in general to give strong guarantees about the optimality of a set of estimated parameters.

However, the authors have had much experience with fitting diffusion models and certain regularities in recovered local minima have presented themselves. In particular, local minima seem to exist near those boundaries of the parameter space where any of the variability parameters η, s_t, or s_z are zero (although it is of course possible that this is a true minimum for some data sets).

To handle these known local minima, we equip the generative algorithm with a strategy to identify these suspect parameter estimates, and let it perform a 'jump' from the edge of the parameter space to somewhere nearer the middle (in particular, s_z and s_t will be changed to half of their maximal value, and η will be, arbitrarily, reset to 0.2). After making this jump, the generative algorithm restarts from the NMS phase. We have experienced that with this *identify-and-jump strategy* the algorithm often succeeds in locating a better point in the parameter space.

List of figure captions

Figure 1. An illustration of the Ratcliff diffusion model.

Figure 2. An example parameter matrix. See text for details.

Figure 3. A decision tree structure illustrating the mixture model.

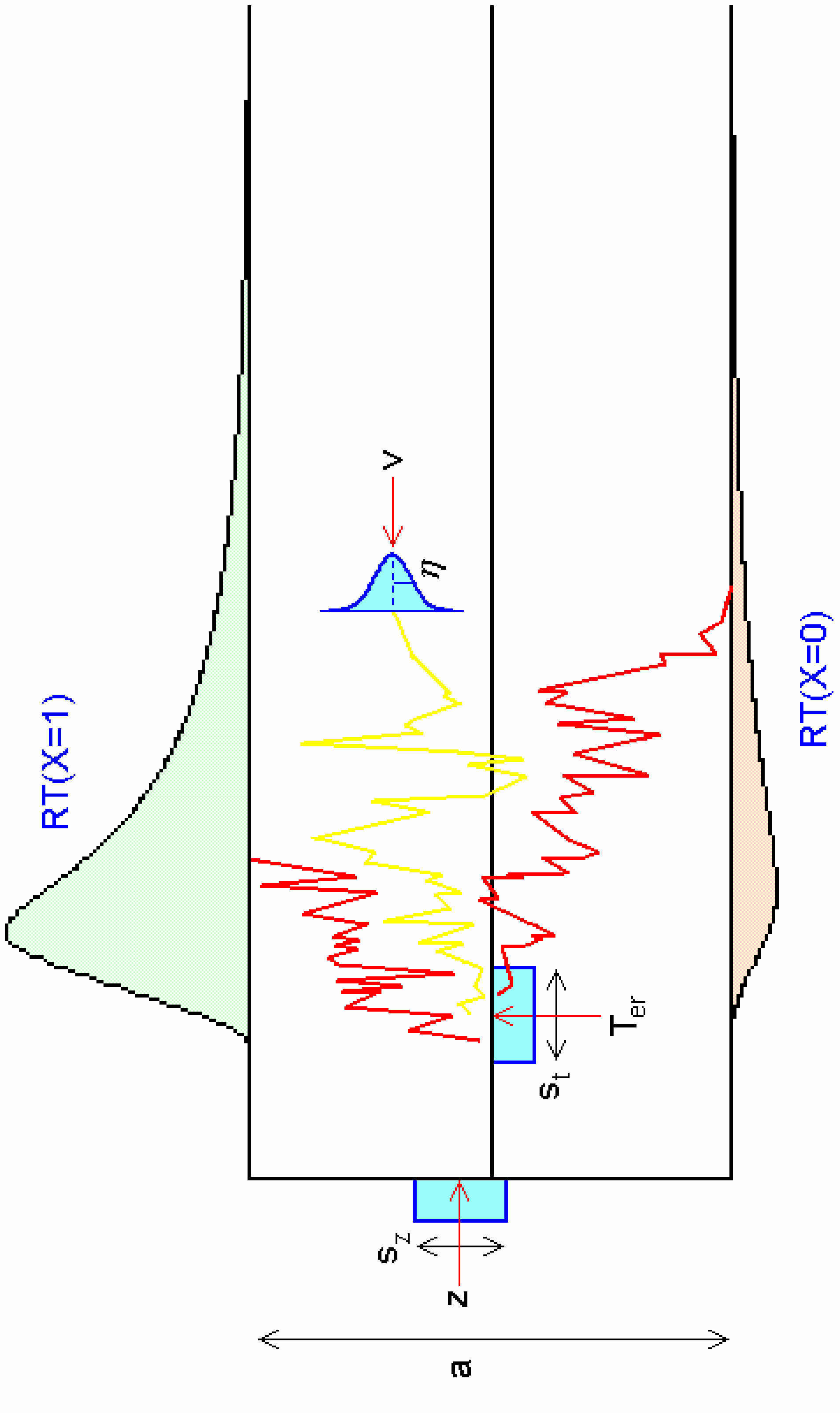
Figure 4. The selectivity of the DMAT algorithm is illustrated. Different lines indicate different parameter sets. The full black line on the diagonal is the statistical expectation of rejected nil hypotheses in function of the significance criterion. Lines above the diagonal indicate a liberal decision, while lines under it indicate a conservative decision. In general, DMAT produces selectivity curves that are close to the diagonal.

Figure 5. The sensitivity of DMAT is illustrated. Different lines indicate data sets with larger or smaller simulated differences in drift rate. As the simulated difference increases, sensitivity curves depart from the diagonal.

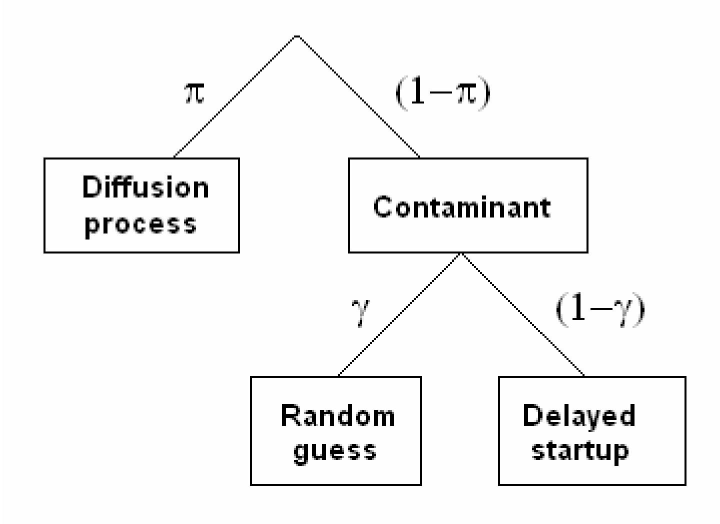
Figure 6. Drift rates of one participant in Experiment 1 of Ratcliff and Rouder (1998). Drifts recovered by Model 2 are shown as dashed lines, with the steeper line indicating the Speed condition. Drifts from Model 3 are full curves, and drifts from Model 5 are stars. As can be seen, Model 2 provides a poor fit, while Model 3 is

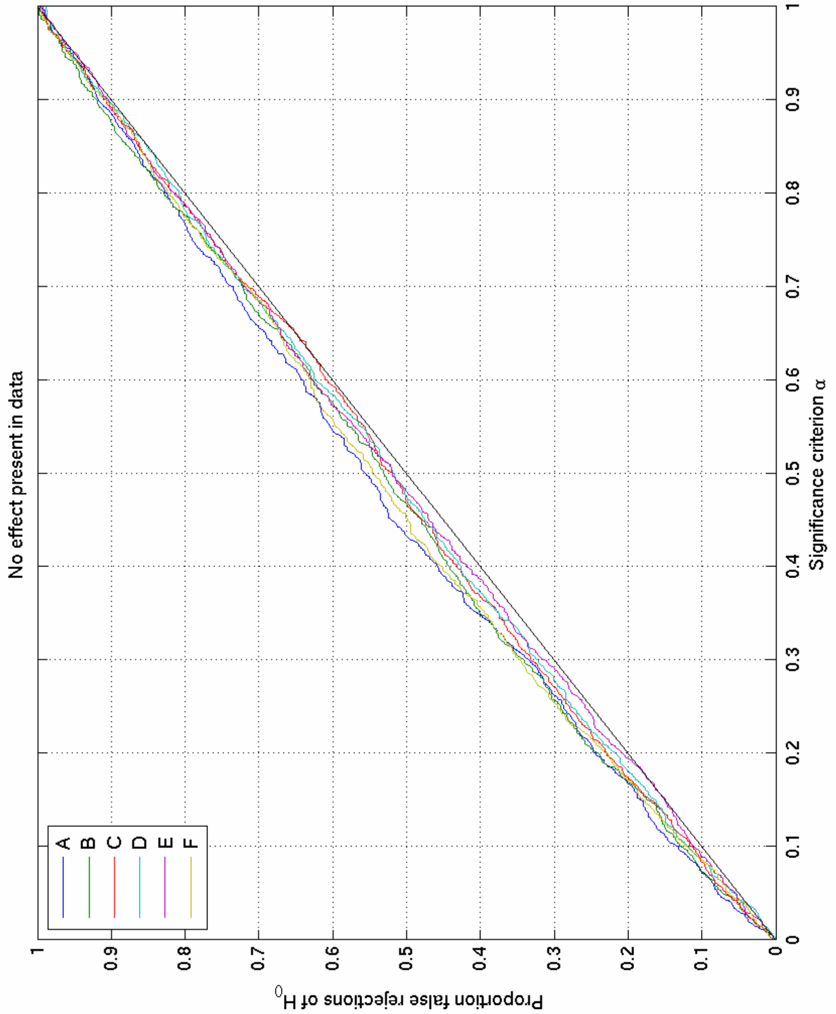
much closer to the separate drift rates, though with still some deviations left.

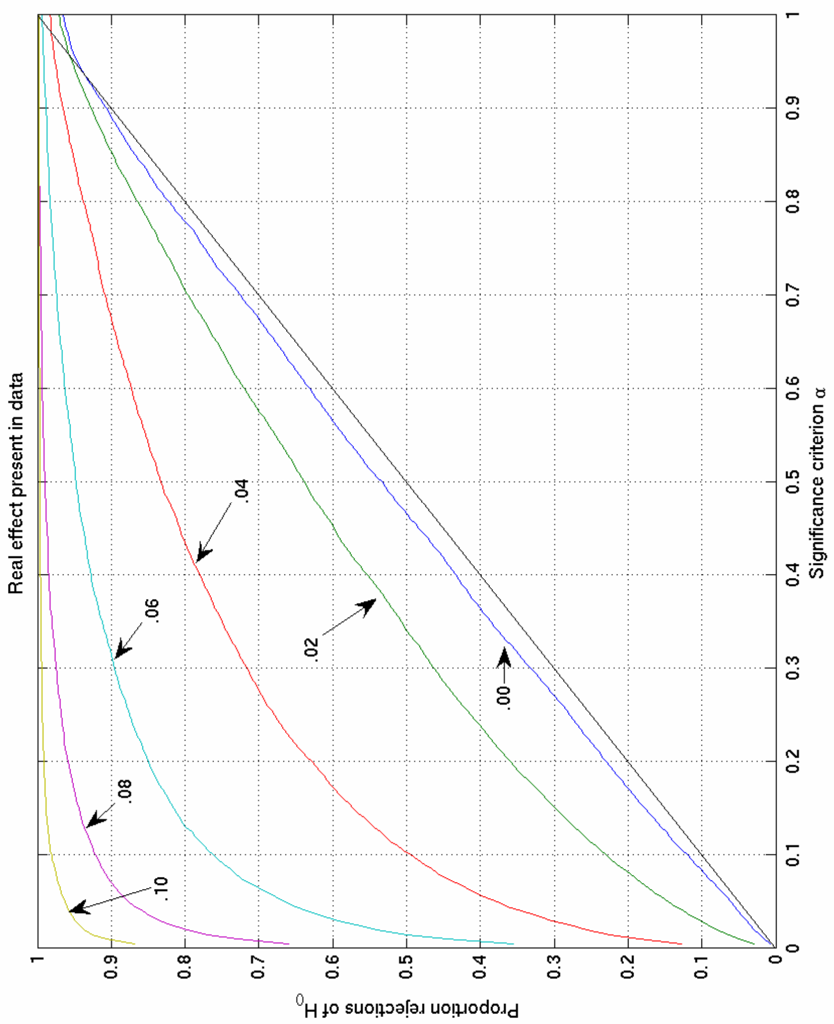
Figure A1. An EWMA control chart showing guessing for reaction times lower than approximately 322ms. See text for details.

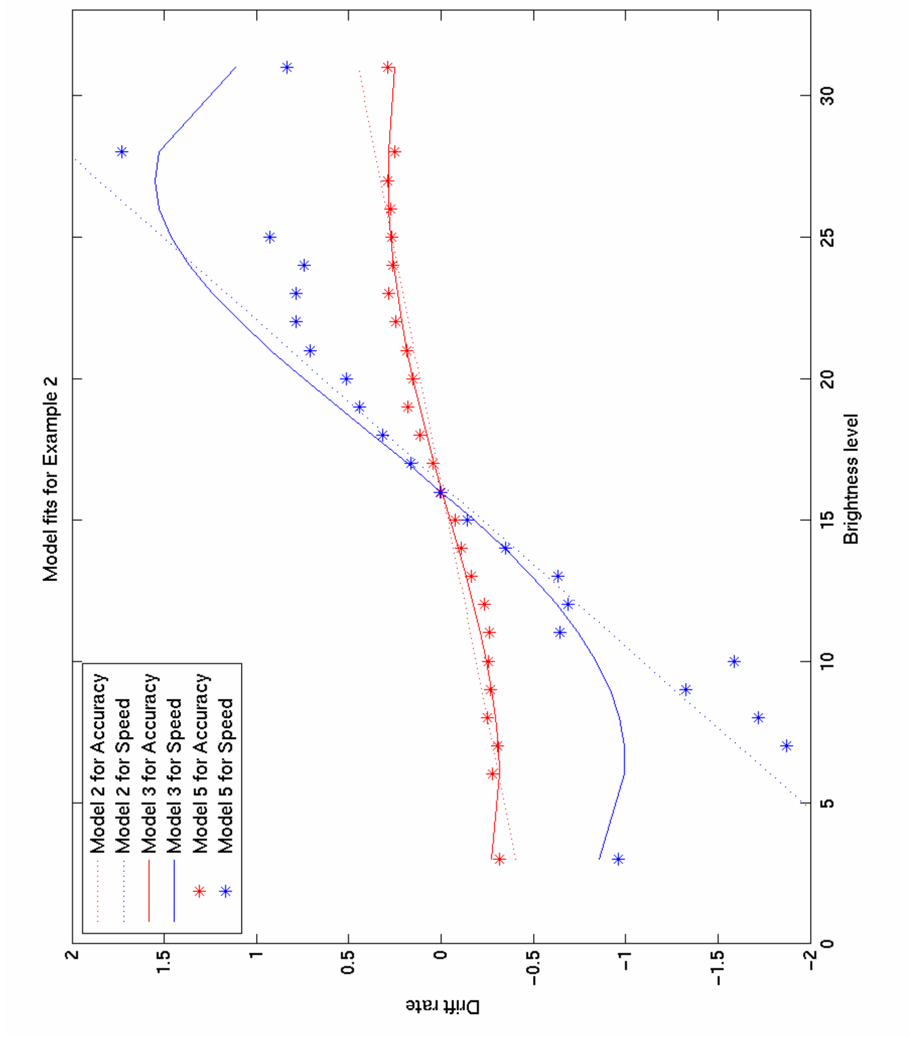

RT(X=1)
v
η
T_{er}
s_t
RT(X=0)
s_z
z
a

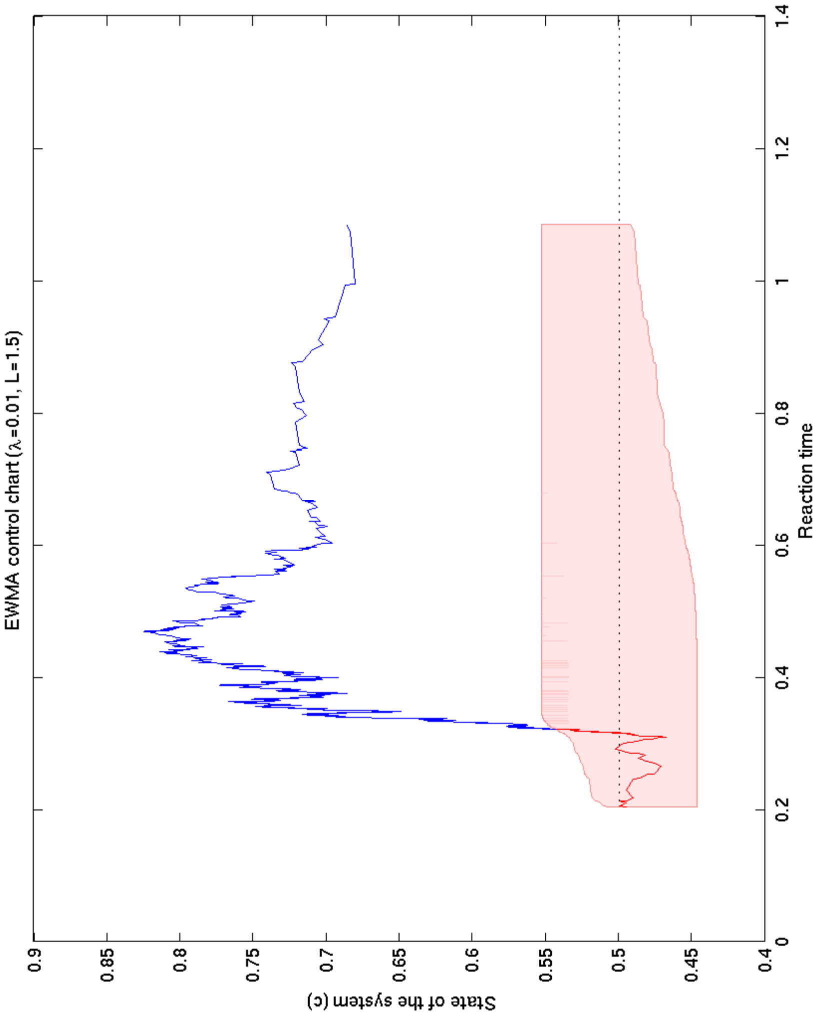
$$
\begin{array}{cc}
 & \begin{array}{ccccccc} \boldsymbol{\Psi}_{\mathbf{a}} & \boldsymbol{\Psi}_{\mathbf{T_{er}}} & \boldsymbol{\Psi}_{\boldsymbol{\eta}} & \boldsymbol{\Psi}_{\mathbf{z}} & \boldsymbol{\Psi}_{\mathbf{s_z}} & \boldsymbol{\Psi}_{\mathbf{s_t}} & \boldsymbol{\Psi}_{\mathbf{v}} \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \end{array} \\
\begin{array}{c} \boldsymbol{\theta}_{(\mathbf{1})} \rightarrow \\ \boldsymbol{\theta}_{(\mathbf{2})} \rightarrow \\ \\ \boldsymbol{\theta}_{(c)} \rightarrow \\ \\ \boldsymbol{\theta}_{(\mathbf{C})} \rightarrow \end{array} &
\left[\begin{array}{ccccccc}
a_{(1)} & T_{er(1)} & \eta_{(1)} & z_{(1)} & s_{z(1)} & s_{t(1)} & v_{(1)} \\
a_{(2)} & T_{er(2)} & \eta_{(2)} & z_{(2)} & s_{z(2)} & s_{t(2)} & v_{(2)} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
a_{(c)} & T_{er(c)} & \eta_{(c)} & z_{(c)} & s_{z(c)} & s_{t(c)} & v_{(c)} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
a_{(C)} & T_{er(C)} & \eta_{(C)} & z_{(C)} & s_{z(C)} & s_{t(C)} & v_{(C)}
\end{array}\right]
\end{array}
$$


π
(1−π)
Diffusion process
Contaminant
γ
(1−γ)
Random guess
Delayed startup


No effect present in data
Proportion false rejections of H_0
Significance criterion α
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
A
B
C
D
E
F


Real effect present in data
Proportion rejections of H_0
Significance criterion α
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
.00
.02
.04
.06
.08
.10


Model fits for Example 2
Drift rate
Brightness level
2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
0
5
10
15
20
25
30
Model 2 for Accuracy
Model 2 for Speed
Model 3 for Accuracy
Model 3 for Speed
Model 5 for Accuracy
Model 5 for Speed


EWMA control chart (λ=0.01, L=1.5)
State of the system (c)
0.4
0.45
0.5
0.55
0.6
0.65
0.7
0.75
0.8
0.85
0.9
Reaction time
0
0.2
0.4
0.6
0.8
1
1.2
1.4