Bayesian Cultural Consensus Theory

Zita Oravecz[1]

Joachim Vandekerckhove[1]

William H. Batchelder[1]

[1]Department of Cognitive Sciences, University of California, Irvine;

zoravecz@uci.edu

joachim@uci.edu

whbatche@uci.edu

We gratefully acknowledge support from research grants from the Air Force Office of Scientific Research (AFOSR) and the Army Research Office (ARO) to Batchelder, PI.

Corresponding Author:

Professor William H. Batchelder,

Department of Cognitive Sciences

University of California

Irvine, CA 92697

whbatche@uci.edu

(949) 824-7271

Abstract

In this paper we present a Bayesian inference framework for Cultural Consensus Theory (CCT) models for dichotomous (True/False) response data, and we provide an associated, user friendly software package to carry out the inference. We believe that the time is ripe for Bayesian statistical inference to become the default choice in the field of CCT. Unfortunately, a lack of publications presenting a practical description of the Bayesian framework in the context of CCT models , as well as a dearth of accessible software to apply Bayesian inference to CCT data has prevented this from happening. In the present article, we provide an introduction to the Bayesian treatment of several CCT models for dichotomous (True/False) response data, with a focus on the various merits of Bayesian parameter estimation and interpretation of results. At the same time, we introduce the Bayesian Cultural Consensus Toolbox (BCCT): a user-friendly and comprehensive software package to fit these models to field data.

Keywords : Bayesian inference, Cultural Consensus Theory, Dichotomous Response Data

Introduction

Cultural Consensus Theory (CCT; Romney & Batchelder, 1999) refers to a family of models that enable researchers to learn about informants' shared cultural knowledge by taking advantage of modern techniques in signal detection theory and psychometrics. Since its publication in a seminal 1986 paper by (Romney, Weller, & Batchelder, 1986) in *American Anthropologist*, CCT has seen hundreds of applications. An overview of these studies, including possible related concerns can be found in Weller (2007). Most of these applications have concerned dichotomous (True/False) response data. Central to the success of CCT as an option for ethnographic research is the availability

of a software tool to perform statistical inference for CCT models. Such tools are currently available in two separate software packages, ANTHROPAC (Borgatti, 1996), and UCINET (Borgatti, Everett, & Freeman, 2002). Unfortunately these programs have several limitations in their ability to handle certain CCT models. Also, as they do not rely on the Bayesian inference framework, their output is limited in how it represents the knowledge about the model parameters that is obtained from the data.

The purpose of this paper is to present a Bayesian inference framework for Cultural Consensus Theory (CCT) models for dichotomous (True/False) response data, and to provide an associated, user-friendly software package to carry out the inference. Bayesian inference has been a mainstay in the field of statistics (Gelman, Carlin, Stern, & Rubin, 2004) for a long time, and in the last decade it has become increasing popular in the behavioral and social sciences reference (Gill, 2002; Jackman, 2009; Kruschke, 2011; Lee & Wagenmakers, 2012).

CCT investigates shared cultural knowledge and belief systems by administering questionnaires to informants who share this knowledge. Typically, several informants are asked multiple questions of a cultural subject and respond in terms of True/False. The standard CCT model for such data was introduced by Romney et al., 1986; there called the General Condorcet Model (GCM). The basic and still widely used (recent examples are Hopkins, 2011 and Miller, 2011) version of this model provides us with estimates of the informants' cultural competence (level of cultural knowledge) as well as with the culturally correct (consensus) answers. The standard CCT model is based on a formal mathematical model from signal detection theory (Macmillan & Creelman, 2005) that takes into account the possibility of a guessing mechanism as well. In particular, an informant is assumed to know the correct answer to a question with a certain probability called his or her 'competence,' and if the informant does not know the answer, they respond with a guess. It is well known in signal detection and psychometric theory that individuals vary in their guessing habits; however, computational limitations led Romney et al. (1986), to assume neutral guessing bias for all informants.

The GCM has been augmented in several ways. Batchelder and Romney (1988) ameliorate its

parameter estimation method and in most of the derivations a respondent specific guessing bias parameter is incorporated. However, both software packages currently used for CCT analysis (ANTHROPAC and UCINET) make the same simplifying assumption about the guessing bias to enable estimation of competencies and culturally correct answers. As a result, it has not been possible to investigate variability in guessing bias in ethnographic studies. A further model extension introduced by Batchelder and Romney (1988) allows item difficulty to be heterogeneous. This involves specifying cultural competence as a function of the respondent's ability and the item's difficulty by the Rasch model (Rasch, 1960), which is a central model in psychometric test theory.

Although Batchelder and Romney (1988) proposed an extended CCT model that can account for variability (heterogeneity) in respondent specific parameters and in item difficulties, a general estimation framework for such extended models was developed much later by Karabatsos and Batchelder (2003). They take advantage of the Bayesian framework (Gelman et al., 2004) and derive statistical inference for the most general GCM: one that allows heterogeneity in the competencies, informant abilities, and item difficulties, as well as for some constrained versions. Bayesian modeling provides a logical and coherent framework for drawing statistical inferences even for complex models like the GCM. As Bayesian statistical inference is directly based on probability theory, the resulting information about the model parameters takes the form of a probability distribution, from which one can obtain much more information about the competence parameters and the correct answers than the single point estimate provided by ANTHROPAC and UCINET. This property offers researchers an intuitively appealing way of dealing with uncertainty in model parameter estimates. In fact, the Bayesian principle is no stranger to those who are familiar with CCT analysis because the derivation of the answer key estimates is based on Bayes' Theorem (Romney et al., 1986). We will demonstrate how this simple and principled Bayesian logic can be applied in a straightforward manner to derive estimates of the GCM with the help of newly developed software that carries out the Bayesian CCT inference.

In spite of the attractive features of the Bayesian statistical framework, it has not gained much traction among ethnographers in general. This applies for CCT research specifically because the

Bayesian GCM models described in the paper of Karabatsos and Batchelder (2003) has so far not featured in any substantive ethnographic investigation. Likely one reason for that is the lack of practical exposure to the Bayesian framework. Also, the implementation of Bayesian parameter estimation routines for CCT has not yet appeared in the form of a user-friendly software package which could be easily accessible to a wide audience of ethnographers. Bayesian parameter estimation methods differ from those of the other estimation frameworks as they are based on iterative sampling schemes from probability distributions. As such, they are not implemented in common software packages or in the software packages typically used for CCT modeling like ANTHROPAC or UCINET.

In this paper we present a Bayesian inference framework from the viewpoint of CCT while also introducing an accompanying software package for conducting Bayesian inference for the GCM for dichotomous data. The software, called BCCT (Bayesian Cultural Consensus Toolbox), is freely accessible on the first author's website (bayesian.zitaoravecz.net) and features an easily comprehensible user interface, so that ethnographers can simply estimate model parameters after having read the instructions in this paper. The program offers a choice between several CCT models for dichotomous response data incorporating different degrees of variability in the leading parameters. We hope that this will facilitate the consideration of a greater variety of CCT models than those in ANTHROPAC and UCINET, while introducing ethnographers to Bayesian methods of statistical inference.

The structure of the rest of the paper is as follows. First, two sections are devoted to summarizing the current developments of CCT modeling, including an overview of the available software packages. Then we provide a pragmatic introduction to the Bayesian framework with a focus on CCT related problems. It is followed by a detailed, practically oriented section which guides the reader through the new software package and how to obtain it. After that we demonstrate the accuracy of parameters recovered by BCCT. Also, we relate BCCT to other software packages. Finally we draw some conclusions.

Cultural Consensus Theory

Cultural consensus theory assumes that cultural truth or shared knowledge resides in the agreement of the members of a culture. A typical dataset analyzed by CCT is collected from informants who presumably constitute a single culture. As part of a standard analysis, one should always investigate whether this condition holds, and a later section provides a way to do this in the context of Bayesian inference.

The type of data set that we analyze in this paper consist of 'true' and 'false' answers from N respondents on M items. Respondents are indexed with i, and items with k. Then the dataset is coded as:

$$Y_{ik} = \begin{cases} 1 & \text{if } i \text{ responds 'true' to item } k \\ 0 & \text{if } i \text{ responds 'false' to item } k \end{cases}$$

and exhibited as a N by M array, $\boldsymbol{Y} = (Y_{ik})_{N \times M}$.

Accordingly, the culturally correct answers to be estimated are denoted as:

$$Z_k = \begin{cases} 1 & \text{if item } k \text{ is 'true'} \\ 0 & \text{if item } k \text{ is 'false'} \end{cases}$$

and the entire answer key by $\boldsymbol{Z} = (Z_k)_{1 \times M}$. The GCM relies on a formal model called the two high-threshold model from signal detection theory (e.g., Macmillan & Creelman, 2005). The rationale behind it is that when a respondent is faced with a question, they either know the (culturally) correct answer with probability D_i or they do not. In the latter case they guess 'true' with a certain probability. Generally, the guessing probability is denoted g_i for informant i. This parameter is defined on the interval [0,1] and the closer g_i is to 1, the more likely it is that informant i will guess 'true'. However, the most basic model sets this parameter to 0.5 for all informants, as described in Romney et al. (1986) and used in ANTHROPAC and UCINET. According to this model, the

probability of respondent i answering 'true' to question k is defined as:

$$\begin{aligned} p(Y_{ik} = 1) &= Z_k[D_i + (1 - D_i)\,0.5] + (1 - Z_k)(1 - D_i)\,0.5 \\ &= Z_k D_i + (1 - D_i)\,0.5, \end{aligned} \tag{1}$$

where D_i is commonly labeled the "informant specific competence". As a first step towards extending the simplified model, the constraint on the guessing bias can be relaxed. Additionally, the competence parameter D_i can be indexed by item k to facilitate further extensions. These two simple modification of Equation 1 are represented as:

$$p(Y_{ik} = 1) = Z_k D_{ik} + (1 - D_{ik}) g_i. \tag{2}$$

The informant specific guessing bias g_i allows for respondents to differ in their response tendencies.

As a next step, we suppose that the questionnaire items aiming to explore some domain of shared cultural beliefs might have differential cultural saliency; that is, not equally difficult. Hence, we fit a Rasch measurement model to the competence parameter, as described in Karabatsos and Batchelder (2003). Consequently, the competence of an informant i for an item k can be modeled as the function of the informant's ability parameter θ_i and the item's difficulty δ_k:

$$D_{ik} = \frac{\theta_i(1 - \delta_k)}{\theta_i(1 - \delta_k) + \delta_k(1 - \theta_i)}. \tag{3}$$

In this specification of the Rasch model, θ_i and δ_k can take values between 0 and 1. As we can see from Equation 3, the probability of knowing the culturally correct answer (i.e., D_{ik}) increases with increasing ability (θ_i) and decreases with increasing item difficulty (δ_k).

As a summary, the extended GCM has 4 classes of parameters. Three of them, namely the informant-specific ability θ_i, the informant-specific guessing bias g_i and the item-specific difficulty δ_k parameters, can take values anywhere between 0 and 1. The fourth class, namely the answer key parameters Z_k can take only the values of 0 and 1.

With respect to the first three classes of parameters, constraints can be introduced. As mentioned above, the basic GCM assumes homogeneous item difficulty and, most often, neutral guessing bias. If the guessing bias is assumed to be neutral (consequently homogenous among respondents), then:

$$g_i = g = 0.5, \tag{4}$$

then informants are equally likely to guess 'true' or 'false'. We can constrain the item difficulty in the same manner to have a neutral, homogenous value across all items:

$$\delta_k = \delta = 0.5. \tag{5}$$

If we substitute 0.5 for δ_k into Equation 3, most of the terms cancel out and we are left only with $D_{ik} = \theta_i$, that is $D_i = \theta_i$. Therefore in models where we do not allow for heterogeneous item difficulty (i.e., $\delta = 0.5$), we keep the term 'competence' (for D). In models with heterogenous item difficulties, the term 'ability' will be used (for θ). By applying both constraints (Equations 4 and 5) above we arrive back to the basic version of the model as described in Equation 1.

Finally, the model can be further constrained by assuming that informants show no variability in their ability parameters, formally:

$$\theta_i = \theta. \tag{6}$$

Such a model constraint might be useful when dealing with a group of cultural experts, or informants that are otherwise very homogeneous in their competence. Also it was this constraint that allowed Batchelder and Romney (1988, Table 5) to construct conservative power tables of the minimum number of informants needed to achieve pre-specified levels of confidence in the accuracy of the estimated answer key. This simplification cannot be implemented in UCINET or ANTHROPAC. In fact, when we apply all three constraints together, as in equations 4, 5 and 6, we arrive to a

model in which the answer key items are simply determined by which response alternative is in the majority.

Equations 4, 5 and 6 describe constraints for three parameters. Therefore, combining the heterogeneous and constrained parameter settings leaves us with $2^3 = 8$ models. All of these models are implemented in the BCCT software package that we introduce later in this paper.

Statistical inference methods for CCT and available software packages

Before providing Bayesian inference for the GCM for dichotomous response data, it is useful to summarize the other published approaches to estimating the model. The earliest method for fitting CCT models deals with its basic version in which guessing bias and item difficulty are homogeneous (see Equation 1). As introduced by Romney et al. (1986), the *matching method* consists of composing a square symmetric matrix of informant by informant matching answer scores (that have been corrected for guessing bias). Then the minimum residual method of factor analysis by Comrey (1962) is applied to this matrix to obtain point estimates of the informants' competency parameters. As a second step the answer key is reconstructed by utilizing the recovered estimates of the competence parameters and applying Bayes' Theorem to obtain the posteriori probability estimates for the two answer categories for each answer key item. We will explicitly compare the formula used to recover the answer key as related to the general idea of Bayes' Theorem in the next section. Estimation routines for this model are implemented in ANTHROPAC. The program outputs informant-specific competencies and the answer key. Moreover it tests the single culture assumption by (1) calculating eigenvalue ratios provided by the factor analysis and (2) reporting possible negative competency estimates. In practice ethnographers applying the software generally consider the single culture assumption non-violated when the eigenvalue ratio (first and second) is higher than 3 and negative competence estimates do not occur. These indicators will be discussed later. ANTHROPAC runs under DOS and is free for download.

Another inference framework for the basic CCT model is derived by Batchelder and Romney (1988). They apply the *covariance method* for measuring similarities in the informants' re-

sponses. This method substitutes the informant-by-informant covariances over items for the corrected matches in the matches method, but otherwise proceeds similarly. The covariance method allows informants to have different guessing biases, but unlike the matching method it requires that the culturally correct answers split equally between 'True' and 'False'. Deriving the answer key estimates relies on Bayes' Theorem in the same manner as described before. The covariance method is implemented in UCINET, which is a Windows based commercial software package. Karabatsos and Batchelder (2003) formulate extended versions of the GCM described by Equation 2 and Equation 3 and derive statistical inference in the Bayesian framework. They allow for informant-specific bias as well as inhomogeneous items. They offer a program in S-PLUS for free download; however no user interface is provided.

Introduction to Bayesian parameter estimation from the perspective of CCT models

In this section we will describe the essence of the Bayesian approach to model inference that is becoming more and more popular in the social and behavioral sciences. The Bayesian framework (Gelman et al., 2004), while typically considered a "modern" method for statistical inference, has its roots in Georgian times. Thomas Bayes (1702 − 1761), an English mathematician and Presbyterian minister, first laid down its basic tenet – Bayes' Theorem – while decades later it was the well-known French mathematician Laplace who reproduced and publicized his findings. However, Bayesian statistics did not become widespread in the statistics community until the last half century, and only in the last decade or so has it become more popular in the social and behavioral sciences. Even nowadays, a majority of statistical methods for the social sciences are based on the classical statistical viewpoint; however, this is changing rapidly, e.g., Gill (2002); Jackman (2009); Kruschke (2011); Lee and Wagenmakers (2012). In the next paragraphs we argue that social science researchers should consider the Bayesian framework for drawing statistical inferences. We also demonstrate the application of the Bayesian framework to CCT models.

Let us consider the case where we want to draw inferences regarding some particular infor-

mant's competence parameter. The classical estimation methods as implemented in ANTHROPAC or UCINET would only provide us with a point estimate (particular number) for this parameter. However, with only a point estimate, we are not able to infer that this informant's competence is most likely (or 95% likely) to lie within a particular interval. Bayes' Theorem capitalizes directly on the axioms of probability theory and provides a more straightforward and natural way of interpreting parameter estimates. In particular, when deriving an estimate for a particular informant's competence parameter (D), the Bayesian framework provides us with a posterior distribution of the parameter values that reflects our knowledge of the parameter given the data. The posterior distribution permits us to make statements about the probability distribution of a parameter directly. For example, not only can we report a point estimate for the parameter such as the mean of the posterior distribution, but we can also infer that an informant's specific competence parameter lies in a certain region with probability 0.95. To illustrate this, see Figure 5, which is part of the output of our Bayesian inference program. It shows that for each parameter not only point estimates like the mean and median, but also a standard deviation of the posterior distribution and 2.5% and 97.5% percentiles are provided. These percentiles can simply be interpreted as the region that contains the parameter with probability 0.95. We will discuss Figure 5 in detail in a later section.

The reason why we are able to make straightforward probability statements about parameter estimates is that in the Bayesian framework the knowledge about parameters is always described in terms of probability densities. The starting point in Bayesian inference is our prior knowledge, or the *prior distribution*, of the model parameters. In our example, we have to decide what we consider a reasonable distribution of the competence parameter of an informant. Most often, we will have no relevant information about particular respondents' ability parameters, so any a priori statement about their abilities must be necessarily vague. Indeed, this is in the spirit of goals of ethnographic research not to impose prior beliefs on the process of discovering cultural knowledge. Fortunately, there exist formal mathematical expressions of uncertainty for the Bayesian framework. For example, as the ability parameter in the CCT framework is, by definition, constrained to the

range [0,1], and if we have no a priori preference for any particular value, the probability density to express indifference between possible parameter values is the uniform distribution on the [0,1] interval. We can of course make the same assumption independently about every informant. In the same vein, it makes sense to express similar uncertainty about each participant's guessing bias parameter. Moreover, we would typically begin with uninformative prior distributions for the item difficulties and answer key items as well.[1] When we collect consensus data on a certain topic, that information is used to update all these initial prior distributions. The more data one acquires, the less influential the priors on the model parameters become.

Bayes' Theorem is the mathematical formalism used for updating prior knowledge regarding model parameters with the information brought by observed data. Knowledge about any particular parameter will still be expressed as a probability distribution. The probability distribution of a parameter conditional on the data (i.e., after having observed the data) is known as the posterior distribution mentioned previously. Let us start with the example of how to derive the answer key items when competencies and guessing bias are already known. As described in Romney et al. (1986) by using Bayes' Theorem, the probability that a particular answer key $\boldsymbol{Z}$ is correct is given by the following equation:

$$p(\boldsymbol{Z}|\boldsymbol{Y}) = \frac{p(\boldsymbol{Y}|\boldsymbol{Z})p(\boldsymbol{Z})}{p(\boldsymbol{Y})}, \tag{7}$$

where $\boldsymbol{Z}$ is the vector answer key Z_k and $\boldsymbol{Y}$ is the data, as defined above. On the left hand side $p(\boldsymbol{Z}|\boldsymbol{Y})$ is the posterior distribution of the possible answer keys. On the right hand side, $p(\boldsymbol{Y}|\boldsymbol{Z})$ is the *likelihood* of the data given the answer key, while $p(\boldsymbol{Z})$ expresses our prior knowledge of the answer key values. Most often, no relevant a priori information is at hand, so it is typically assumed that all consensus answer keys are apriori equally likely. The denominator is a numerical normalizing constant that expresses the probability of observing the data, averaged and weighted

[1] In principle, we could incorporate informative prior beliefs about model parameters, for example distributions centered around a certain value (in our setting, beta distributions would be a suitable choice). In case of a longitudinal setting for example, when the same informants are asked several times about the same topic, one might use information from one study to express prior information a subsequent study.

over all possible parameter values. The goal of Bayesian inference is to approximate the posterior distribution of the answer key values in order to draw inferences about it. Formulas for the posterior distribution of the answer keys are derived by Romney et al. (1986) and Equation 7 is essentially the same as Equation 24 in Batchelder and Romney (1988) accompanied with further derivation incorporating the competencies and guessing biases, and these results along with the constraint that $g = 0.5$ is implemented into the matching method in ANTHROPAC and UCINET.

The same logic can be applied to all parameters, not only the answer key. If we collect all model parameters in the vector $\boldsymbol{\xi}$, the joint posterior distribution of all the parameters can be expressed as:

$$p(\boldsymbol{\xi}|\boldsymbol{Y}) \propto p(\boldsymbol{Y}|\boldsymbol{\xi})p(\boldsymbol{\xi}),$$

where, as before, $\boldsymbol{Y}$ denotes the data. This formula is very similar to that in Equation 7, except that we no longer explicitly mention the normalizing constant in the denominator and replace the equality sign with the proportionality sign $\propto$. Computational statisticians have found numerical sampling ways to estimate the posterior distribution, even when we can describe it only up to a constant of proportionality (Gilks, Richardson, & Spiegelhalter, 1996; Robert & Casella, 2004). In practice, when we have estimated the joint posterior distribution of all the parameters, we often want to explore the marginal probability density of each parameter. Turning back to our example, let us say, for example, that we want to see how likely it is that the first informant's competence parameter (D_1) is smaller than a certain value, like 0.7. For that we explore the *marginal posterior density* of that competence parameter, which is simply proportional to our prior knowledge of the parameter multiplied by the probability of the data given the parameter (namely the likelihood function). Sometimes if the statistical model is very simple, this multiplication leads to a known distributional form of the posterior density, in which case the probability that the parameter is less than some particular value is given immediately by the posterior cumulative density function (CDF) at that value. However, in most cases such as ours the posterior distribution has a mathematical

form for which this probability is difficult or impossible to compute directly. The standard strategy in this common situation is to use modern computational tools, called Markov Chain Monte Carlo (MCMC) methods, to develop a sampler that can generate many random values from the posterior distribution, and use the sampled values to approximate the posterior distribution. An extensive literature exists (see e.g.,Robert & Casella, 2004; Gamerman & Lopes, 2006) regarding simulation procedures to draw samples from any kind of distribution. Also, there are available software packages that automate many of the steps required to run such a sampling procedure. However, these software programs require the computational power of a modern computer to do the sampling in a timely manner. Even with modern computers, computation time is a significant issue in Bayesian inference methods. Researchers might have to exercise patience by waiting a minute or so while their machine is carrying out Bayesian inference.

Once we have explored the posterior density of the competence parameter in question (i.e., we have generated an ample amount of samples from its posterior density), several statistics can be calculated. For example, measures of central tendency, such as the mean and median can easily be attained. Posterior standard deviations can also be calculated to express uncertainty around these point estimates. Also, Bayesian credibility intervals (BCIs, the Bayesian counterpart to classical confidence intervals) can be constructed. Such BCIs designate a region where the true parameter lies with a certain probability, typically with probability 0.95. In addition, if we want to evaluate how likely it is that the competence parameter is smaller than a given value, say 0.7, computing this likelihood from the posterior density simply amounts to counting the proportion of sampled values for the parameter that fell below 0.7. The straightforward way of interpreting uncertainty in parameter estimation can be especially appealing for the social sciences, where parameters are often associated with personality traits, attitudes, skills, behavioral properties and so on, where it makes particular sense to quantify uncertainty in inferences in an easily interpretable fashion. In summary, the Bayesian statistical framework offers a rational and coherent way of drawing inferences, yet it is still not as widespread as the classical inference framework in the social and behavioral sciences. There are several reasons for that. Most of all, as mentioned

above, in many cases Bayesian parameter estimation using computational MCMC samplers requires some magnitude of computational power, which has become available only a couple of decades ago. Second, until recently there have been only a few books and articles written especially for social scientists that describe the logic behind using MCMC samplers to estimate the posterior distribution. We deal with the second reason next. Although in theory the posterior distribution described above has nice and easily accessible properties, one has to make sure that the sampler is accurately tracking the correct posterior distribution on which the inferences are drawn. Any MCMC sampler is designed to sample values of the parameters in proportion to how probable they are from the posterior distribution. Unlike usual random sampling from a distribution that consists of independent draws from the distribution, an MCMC sampler draws correlated values from the posterior distribution, meaning that is there is autocorrelation between successive samples due to the fact the sampler satisfies the properties of a Markov chain. Nevertheless, if sufficiently many samples are obtained from a MCMC sampler, a binned frequency distribution of values closely approximates the desired posterior distribution. To assure this, the sampled values have to meet certain standards. We have developed an accessible, user-friendly, Bayesian inference software package for the GCM (dichotomous response data) that includes ways to assure that these standards are met. It is described in the next section; however, it is of value to briefly mention the standards and their associated statistical terms as follows.

The first standard that needs to be met by the MCMC sampler is that the entire parameter space must be sufficiently explored. For that, sampling schemes should start multiple times, each with different starting values in different regions of the space of possible parameter values, and the generated sample chains (sequences of draws from the posterior distribution of the parameters) should all end up conveying the same information about the posterior distribution. The second standard concerns the autocorrelation in the sequence of samples obtained from the MCMC sampler. It is important to make sure that all regions of the posterior density are well represented: that is, the regions of greater density should contribute more samples than sparser regions. If there is high autocorrelation in the sampled values (lag 1 autocorrelation; i.e., the correlation of one sampled

value with the next), there is a risk that some regions are oversampled, as the sampler moves too slowly in the parameter space. This leads to a decision about 'thinning' the sample by retaining only every k^{th} sample. We will describe the way that out software package handles multiple chains, autocorrelation, and thinning in a later section. There are several books written in an accessible fashion that describe how to carry out Bayesian parameter estimation (e.g., Kruschke, 2011; Lee & Wagenmakers, 2012). Researchers can decide to write their own script to sample from the posterior distribution or can rely on software packages with built in sampling algorithms to carry out these computations and draw inferences. Currently, there are two major, generic software packages that deal with Bayesian statistical inference, namely WinBUGS (Lunn, Thomas, Best, & Spiegelhalter, 2000) and JAGS (Plummer, 2011). The two programs have almost identical applicability areas, and are very similar in their usage.

Introducing the Bayesian Cultural Consensus Toolbox through an example dataset

We have written a software application called Bayesian Cultural Consensus Toolbox (BCCT) for conducting inference for various versions of the GCM model for dichotomous response data discussed in earlier sections. In this section we provide the user with all the information necessary to use the software. BCCT is a standalone application that works solely through a graphical user interface (similar to UCINET) and does not require any programming knowledge from the user. Through analyzing a dataset, in this section we demonstrate how straightforward it is to carry out statistical inference for several versions of the GCM. The interface allows the user to easily navigate among the different modeling choices. Moreover, it offers a straightforward way to explore the posterior distribution of the parameter estimates, as well as providing a way to deal with issues of model fit. A document that downloads with the software package guides the user step by step through the installation. BCCT uses two accompanying programs that extract automatically. One of them is JAGS which carries out the sampling for the different GCM models. It works in the background when the BCCT software is used and is not explicitly displayed in any way for the user.

Second, as the program for the interface and all supporting subprograms were written in MATLAB, and a MATLAB Compiler Runtime installs when the package for the software is extracted. The user does not need a MATLAB license.[2]

Reading in data, specifying the model, starting the estimation

Now we demonstrate how to use BCCT for making statistical inferences for GCM models. A flow diagram of the different steps of conducting Bayesian inference for the GCM with BCCT is included below in Figure 1.

We make use of a dataset that is published in Romney (1999). The dataset was collected by Weller (1984) and it has been analyzed by several other authors such as Weller (1984), Romney et al. (1986) and Batchelder and Anders (2012), however, only with models assuming item homogeneity. The original study measured knowledge about contagiousness of certain diseases in a Guatemalan population. The sample consists of 24 informants who provided dichotomous responses to 27 items about a disease being contagious or not.

BCCT reads the dataset from a text file with informants represented by rows and items as columns. The data should be coded as specified above, only 0 and 1 entries are accepted. The dataset analyzed in this example downloads with the program to provide an example. When BCCT opens the user can click on Browse and load the preferred dataset. Figure 2 displays a screenshot of BCCT when the contagion data set has been loaded.

Under the data path the program displays results from exploring the loaded data. This way the user can check whether the number of informants and items are correct. Moreover, the last box shows the ratio of the first and second eigenvalue based on the covariance matrix of the informants' answers as described in Batchelder and Romney (1988). The same diagnostic is built in UCINET and field researchers often check the basic model axiom of a single underlying culture with the help of this number. In practice, it has been assumed that a threshold of at least 3 is needed to

[2] As mentioned before, BCCT can be downloaded from bayesian.zitaoravecz.net. Also, all MATLAB scripts that constitute the program are automatically downloaded as part of the BCCT package, and can be found in the folder called 'Supplementary files'. Users who do have a MATLAB license can use these files to run BCCT in its native environment.

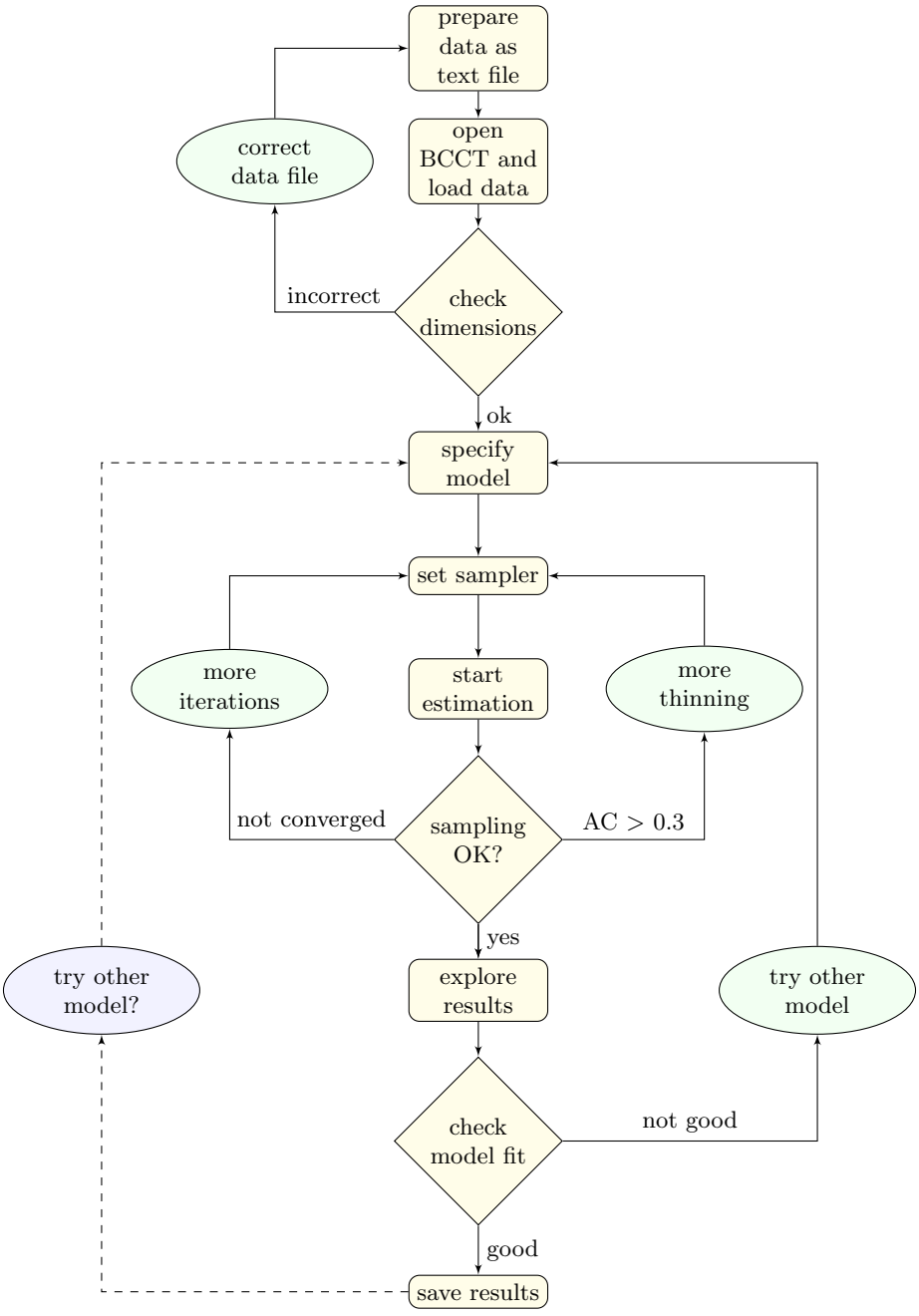


Figure 1. : Conducting Bayesian inference for the GCM with BCCT

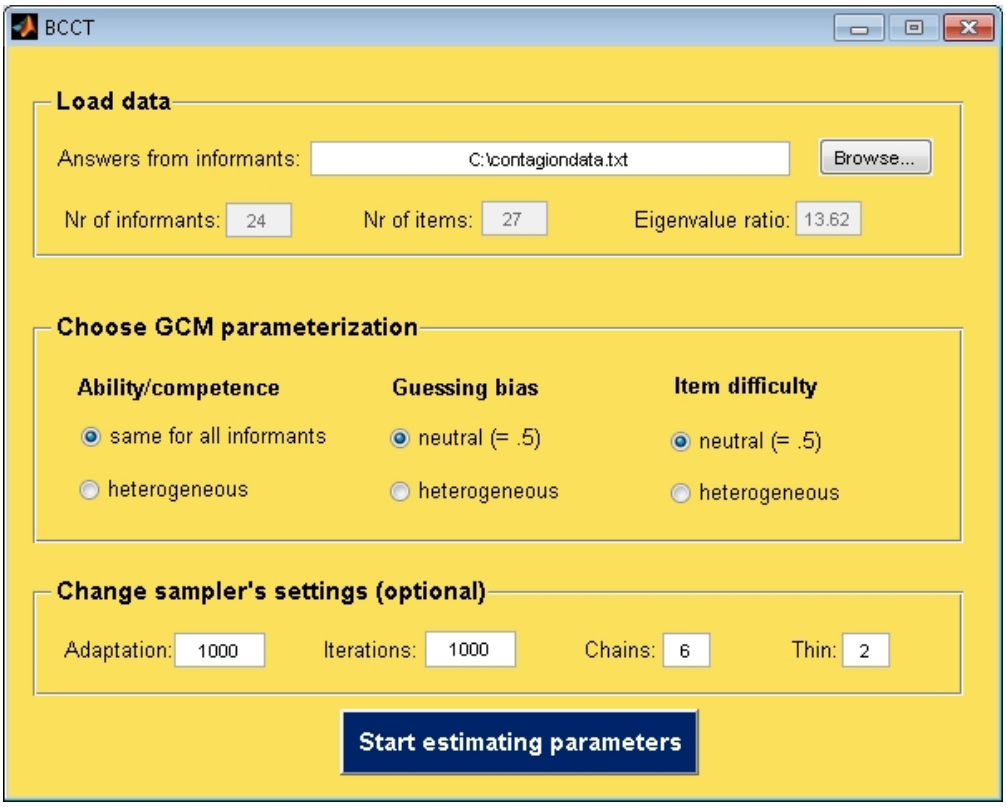


Figure 2. : Screenshot of BCCT after the dataset has been loaded

support the single culture assumption. However, the eigenvalue ratio can only indicate that the single culture assumption is most likely violated (if ratio is under 3), but it can never prove that this condition is met, even if it is above 3. For discussion on this problem see Hruschka, Sibley, Kalim, and Edmonds (2008). Here we offer this number only to provide an exploratory indicator of whether the dataset seems to violate the assumption. Later we show how to test the one culture assumption in the Bayesian framework as part of the posterior predictive model checking, following the guidelines of Batchelder and Anders (2012).

The second panel Figure 2 allows the user to specify which model is to be estimated. By default it is set to the simplest and most constrained model that assumes that all respondents share the same competence parameter (to be estimated), and that their guessing bias and item difficulties

are neutral. This way the model only estimates one competence parameter and k answer key variables. Alternative models can be set by relaxing any or all of these constraints by clicking on the 'heterogeneous' option.

The bottom panel Figure 2 contains the parameter settings for the sampler program that draws samples from the posterior distributions of the model parameters. The type of algorithms that it uses is automatically chosen by JAGS. The user has to decide only four options displayed in the bottom panel, however, most often the already pre-filled values should suffice, so the user can just leave those unmodified and click directly on the Start estimating parameters button. The reason for the options in this row is related to the standards for assuring the adequacy of the MCMC sampler discussed earlier. In the very first box, called Adaptation, we allocate a certain number of samples to be drawn but then discarded (not used to estimate the posterior distribution), so they will not count towards our final posterior samples. The reason for this adaptation period (most often called "Burn-in") is twofold. First of all, the first sampled values of each sample chain are highly determined by their random starting values, and therefore do not represent the posterior distribution accurately. Second, during some initial period of sampling we want the sampler program selected by JAGS to adapt its algorithms to find the most efficient one. For most CCT models the sampler needs around the first 1000 iterations to be discarded. In the second box, called Iterations, the number of final (useable) posterior samples for each chain is specified. It should be sufficiently large so that the sampler can explore the whole area of the conditional posterior densities of the parameters. Also, as mentioned above, this density exploration should be restarted with different initial values to further ensure the sufficient exploration of the entire parameter space, and in the box called Chains we define how many times the explorations are restarted. Our final posterior sample size is the product of these two. As a rule of thumb, 6 chains with a minimum of 1000 useable samples each should be more than sufficient in most cases. Finally, the very last box concerns the autocorrelation in the chains. Posterior inference should be based on chains that have little or no autocorrelation, so that we can be sure that each area of the posterior density is equally likely to be represented in our final posterior sample. The way the

Thin box controls the autocorrelation is the following: it keeps only every k^{th} of them, and discards everything between, this way decreasing correlation between the new neighbors. So if we put 1 in that box, that means that we do not thin, we keep every sample after the adaptation period. If we put 2 we discard every second sample and so on until we reach the number of useable iterations set in the Iteration box. The more we thin, the smaller the autocorrelation becomes. While for some models and some datasets probably no thinning is necessary, we will see that thinning needs to be increased in some cases, like in our forthcoming example.

Figure 3 shows how the model in which the competence and the guessing bias are set to heterogeneous, so that between informant variability is taken into account in these aspects. Also, in the bottom panel we changed the sampler setting parameters somewhat: we raised the thinning into 5. The screenshot represents a state of the program in which the estimation has already been started by the Start estimation button, the text of which is now changed into Estimating. We mention again that sampling might take several minutes, depending on the size of the dataset and the settings in the third row of boxes. The current example analysis for example took about one minute on a PC with a 3.20 GHz Xeon CPU and 12 GB RAM.

Exploring the results

Figure 4 displays the window which pops up after the sampling is finished, except that initially the lower panel would be empty until the user pushes the Perform checks button (which here is already changed to Checks performed).

The upper row provides us with some basic information about the quality of our posterior samples. The first piece of information reports on whether the chains appear to have converged to the same distribution. The way the program checks convergence is to calculate the $\hat{R}$ values (Gelman et al., 2004) for each parameter, which roughly equals the ratio of the between- and within-chain variances. A generally used rule of thumb by computational statisticians is implemented in the program namely that values less than 1.1 suggest convergence. The program checks this value for each model parameter and reports convergence only if all of them are less than 1.1. While in

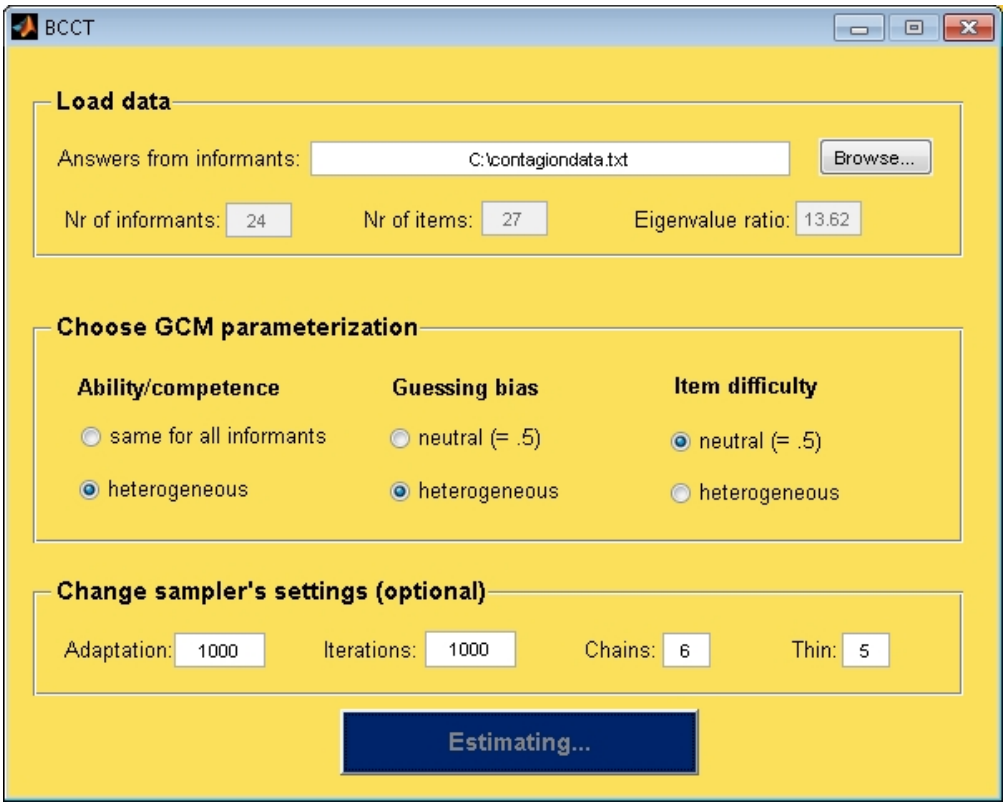


Figure 3. : Screenshot of BCCT after the model and the sampler options have been specified and the estimation has been started.

our experience this $\hat{R}$ criterion has been accurate for indicating convergence (or lack of it) for the set of GCM models in the program, sometimes it can be reassuring to display posterior sample chains of some random parameters, and check whether the chains cover more or less the same area. Such plots can easily be generated by the first panel's Plot chain(s) button (see later). This is an especially useful technique if we suspect that some parameters might converge slowly. In case there is no convergence we can still look at the posterior samples to try to figure out that adding more samples would increase our chances of convergence. If we see that the chains are more or less in the same region, adding more iterations in the Iterations window and rerunning the model may solve the issue. However, when the model does not fit the data, for example because the one underlying culture assumption is not met, even very long chains might not reach convergence, as reported for

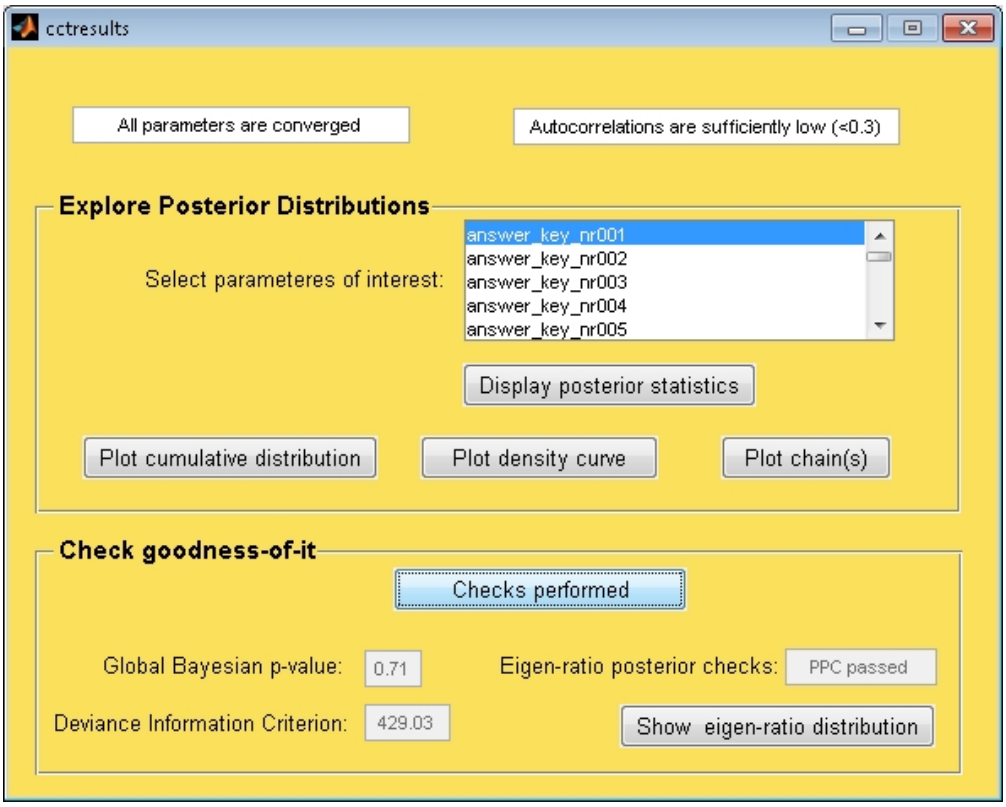
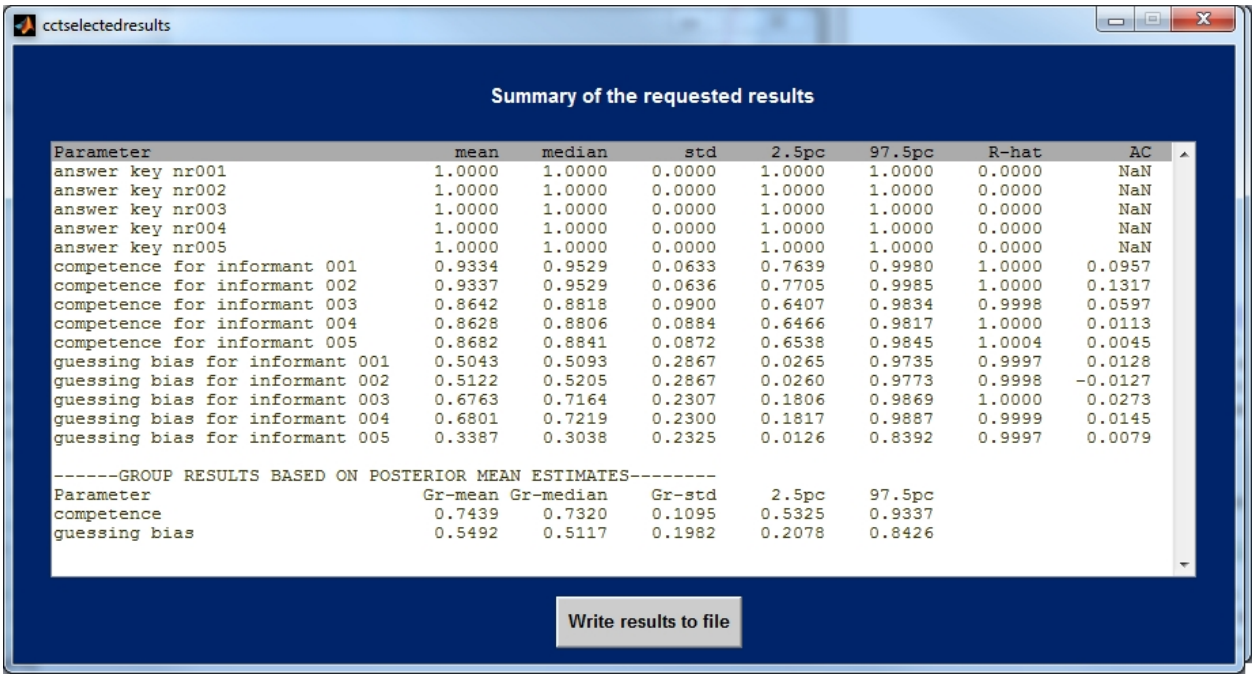


Figure 4. : Screenshot of BCCT's result summarizing window

example in Batchelder and Anders (2012).

The next textbox of the top line tells us about the autocorrelation. The program is set to the somewhat liberal level of 0.3 to decide about reporting a problem with the autocorrelations. However, the program can return information on the autocorrelations in the samples of any of the parameters (see next paragraph), hence users more concerned about autocorrelation can always check these values and rerun the analysis with an increased number of thinning to reduce autocorrelation. That strategy is also advised for cases when this textbox reports autocorrelations higher than 0.3. As mentioned above, we can decrease autocorrelation by increasing thinning (see box called Thin in Figure 3, lower right corner). Alternatively, instead of increasing thinning, one can add more chains.

The upper panel of Figure 4 contains options to explore the results. First of all, all estimated parameters, indexed by persons or items when applicable, are loaded into a textbox. The user can make multiple selections of parameters of interest to inspect the statistics of their posteriori density. Then by clicking on the 'Display posterior statistics' under this selection, a new window will pop out showing some summary statistics of the posterior samples. An example of this can be seen in Figure 5.



Parameter	mean	median	std	2.5pc	97.5pc	R-hat	AC
answer key nr001	1.0000	1.0000	0.0000	1.0000	1.0000	0.0000	NaN
answer key nr002	1.0000	1.0000	0.0000	1.0000	1.0000	0.0000	NaN
answer key nr003	1.0000	1.0000	0.0000	1.0000	1.0000	0.0000	NaN
answer key nr004	1.0000	1.0000	0.0000	1.0000	1.0000	0.0000	NaN
answer key nr005	1.0000	1.0000	0.0000	1.0000	1.0000	0.0000	NaN
competence for informant 001	0.9334	0.9529	0.0633	0.7639	0.9980	1.0000	0.0957
competence for informant 002	0.9337	0.9529	0.0636	0.7705	0.9985	1.0000	0.1317
competence for informant 003	0.8642	0.8818	0.0900	0.6407	0.9834	0.9998	0.0597
competence for informant 004	0.8628	0.8806	0.0884	0.6466	0.9817	1.0000	0.0113
competence for informant 005	0.8682	0.8841	0.0872	0.6538	0.9845	1.0004	0.0045
guessing bias for informant 001	0.5043	0.5093	0.2867	0.0265	0.9735	0.9997	0.0128
guessing bias for informant 002	0.5122	0.5205	0.2867	0.0260	0.9773	0.9998	-0.0127
guessing bias for informant 003	0.6763	0.7164	0.2307	0.1806	0.9869	1.0000	0.0273
guessing bias for informant 004	0.6801	0.7219	0.2300	0.1817	0.9887	0.9999	0.0145
guessing bias for informant 005	0.3387	0.3038	0.2325	0.0126	0.8392	0.9997	0.0079

------GROUP RESULTS BASED ON POSTERIOR MEAN ESTIMATES--------

Parameter	Gr-mean	Gr-median	Gr-std	2.5pc	97.5pc
competence	0.7439	0.7320	0.1095	0.5325	0.9337
guessing bias	0.5492	0.5117	0.1982	0.2078	0.8426

Figure 5. : Screenshot of the BCCT window displaying summary of posterior statistics on selected parameters

Each line corresponds to the statistics of one of the model parameters named in the first column. The second and third columns show the posterior mean and median, which can be thought of as point estimates of the parameters. The fourth column displays the uncertainty in the parameter estimates in terms of posterior standard deviations. The fifth and sixth columns concern the spread of the posterior density by specifying its 2.5^{th} and 97.5^{th} percentiles, or in other words the 95% Bayesian credibility interval. Then the next column shows the exact $\hat{R}$ values on which the

convergence check (see above) is based. Finally, the last column reports the level of autocorrelation. The results displayed in this table can be written out into a standard text file by clicking the button below the table. Looking at the first line for a competence parameter in Figure 5, we can see that a possible point estimate for the competence parameter of informant 001 is 0.93. Also, its 95% BCI lies above 0.7, so it is very likely that this informant's competence level is above 0.7. For deriving how probably that is exactly, we go back to the main window, select our parameter of interest and push Plot cumulative distribution button. It then displays its cumulative posterior density curve, as it is shown in Figure 6. Since this plot is interactive, by pointing at the particular parameter value on the x-axis, the plot shows us its place on the curve and displays numerically how likely it is that the parameter value is lower than that value. For $D_1 < 0.7$ it shows 0.0065, so it is rather unlikely that this informant has low competence level.

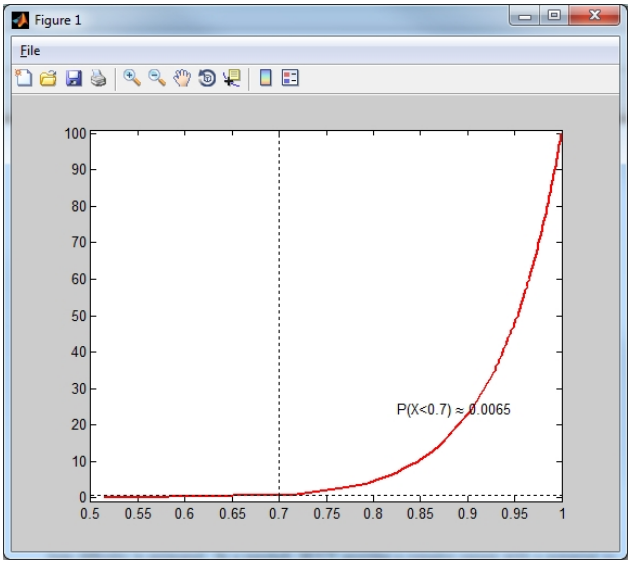


Figure 6. : Screenshot of the cumulative distribution plot generated by BCCT

The parameter selection in Figure 5 is only an example, the user can set all of the parameters or any subset of them. However, at the end of the displayed table, there is always some summary about informant and/or item specific parameters. More specifically, based on the point estimates

(posterior means) of the competence (or ability), guessing bias and item difficulty (if applicable) parameters, summary statistics such as mean, median, standard deviation, and 2.5^{th} and 97.5^{th} percentiles are calculated. This way the user can have an idea about the average level of these parameters in the sample (mean, median), as well as about how much variability there is among informants and items (standard deviation, percentiles). These quantities would be the usual sorts of statistics about the competence level of a group of informants that one would publish in a research paper. For example, we see that the mean competence level is 0.74 and the standard deviation is 0.11. In the case of the guessing bias, the mean is 0.55, which is close to the neutral guessing value of 0.50 assumed in earlier analysis of the contagion data by ANTHROPAC. However, the standard deviation of the guessing bias is 0.20, which suggests that there is a lot of heterogeneity in guessing bias in the group of informants. Certainly, one can look not only at the point estimates but the BCIs as well, and can calculate for example how many informants are most likely to have competency above 0.7, or how many of them will most likely always answer "False" etc. Moreover, although it is not displayed in the current example, ethnographers might also find it interesting to compare item difficulty levels for the different questions of the questionnaire, when the GCM with item difficulty is estimated. In a nutshell, BCCT provides a complex output with a potential to investigate various research questions.

The Plot density curve button opens a plot of the smoothed posterior distribution of the selected parameter. Finally the Plot chain(s) button generates plots showing all the sample chains for each of selected parameters.

Checking model fit

The Perform check button of the lower panel (already pushed in Figure 4) carries out several goodness-of-fit tests. As the tests may take a few minutes, this option is not executed automatically, but only at the user's request. The lower panel of Figure 4 shows the results of these checks for the example data set.

Bayesian p-value. The posterior p-value reported in that box represents a global fit statistic of the current CCT model to the data. Calculation of this value is based on Bernoulli deviance functions (Karabatsos & Batchelder, 2003) of the raw data and replicated data sets based on the posterior distributions of the model parameters (set by a maximum 1000 samples in BCCT). As this statistic summarizes how likely it is that the current data occurred under the model assumptions, low p-values (under 0.05) indicate a poor fit.

Posterior predictive check of the single culture assumption. This option calculates the first and second eigen-value ratio of the data and the same ratio for generated datasets based on the posterior distribution of the model parameters. As has been described before, the usual rule of thumb for supporting the single culture assumption of the model is that this ratio is above 3. This rule is actually questionable, as the ratio is very dependent on the overall level of competence on the group of informants. On the other hand, the Bayesian postpredictive test used by BCCT is based on the ratio of the first to second eigenvalues. The posterior predictive check of the eigenvalues has recently been introduced by Batchelder and Anders (2012) and the BCCT program follows the calculations outlined in that paper to get these eigenvalue-ratios. Then it checks whether the data eigenvalue ratio is within the 95% posterior predictive distribution of the model based eigenvalue ratios. If not, it returns an error message saying: 'PPC failed', otherwise it returns 'PPC passed'. By pushing the button under the statistic we can see the smoothed posterior predictive distribution of these eigenvalues with the data's eigenvalue ratio represented by a straight line, and this plot enables the user to exactly where the data eigenvalue ratio is with respect to the distribution of simulated values.

Deviance Information Criterion. While the Bayesian p-value and the eigen-ratio test provide goodness-of-fit tests in the absolute sense (the model either fits or does not), with the Deviance Information Criterion (DIC; Spiegelhalter, Best, Carlin, & van der Linde, 2002) one can evaluate the relative goodness of fit of different models. DIC takes into account two important features of the model: the complexity (based on the number of parameters) and the fit (measured by a

Table 1:: Results of the simulation study with BCCT.

	Ability	Guessing bias	Item difficulty
'True' value outside 95% BCI	1.6 %	5.6 %	6.4 %
Average Absolute Difference	0.0881 (0.0190)	0.0884 (0.0151)	0.1156 (0.0111)

deviance statistic). Naturally, more complex models fit better, therefore DIC penalizes based on the effective number of parameters in the model to arrive to a fair measure of fit. The model with the lowest DIC is supposed to be the best fitting one. For further discussion of DIC in the context of CCT modeling please consult (Karabatsos & Batchelder, 2003).

Testing BCCT

Here we provide a small simulation study to demonstrate the parameter recovery accuracy of BCCT. Five datasets – 25 informants and 50 items each – were generated based on the all heterogeneous model. 'True' parameter values were drawn from a uniform distribution between 0.1 and 0.99 for the ability, guessing bias and item difficulty parameters. The answer key was sampled from a Bernoulli distribution with probability 0.5. In the BCCT window the competence, guessing bias and item difficulty were set to "heterogeneous". The results are based on converged chains with a good level of autocorrelation in each analysis. Based on posterior samples, two measures were calculated. First, we checked whether the 95% posterior BCIs of the estimates of abilities, guessing biases and item difficulties contained the simulated 'true' parameter values. We calculated the proportion of times when it was outside of this region over persons (for ability and guessing bias) and items (item difficulty) for each run. The first row of Table 1 shows these percentages averaged over the 5 runs. Second, we calculated Averaged absolute distances (AAD) between simulated and estimated values, averaged again over persons and items. The second row displays the mean and standard deviations of these values over the runs.

Table 2:: Results of the simulation study with UCINET and BCCT.

Sample size	UCINET	BCCT
50 informants - 100 items	0.0657 (0.0131)	0.0615 (0.0089)
25 informants - 50 items	0.0984 (0.0106)	0.0855 (0.0085)

Ideally, the 'true' values should lie outside of the 95% credibility interval about 5% of the time. As we can see, all three parameter classes seem to meet this standard. Also, the AADs suggest that even with a moderately sized dataset the estimates are rather accurate. With respect to the answer keys, on average over the 5 runs only 1.8 (std = 0.8) items were incorrectly recovered. These incorrectly recovered items were all rather difficult: their average item difficulty level was 0.89 (std = 0.03). As the average of the simulated ability parameters were only slightly above 0.5, these results are coherent with our expectations based on the simulation settings.

Relating the BCCT to UCINET

We also set up a simulation study for comparing UCINET and BCCT. Ten datasets were simulated based on the CCT model in which informants' competencies were drawn from a uniform distribution between 0.1 and 0.99, the guessing bias was fixed for all informants on a neutral level (0.5) and item homogeneity was assumed. The answer key was sampled from a Bernoulli distribution with probability 0.5. The first 5 simulated datasets were of size 50 informants and 100 items, while the last 5 were of half of that size. CCT analyses were done by UCINET and BCCT. In case of BCCT, the competence was set to 'heterogeneous', guessing bias and item difficulty to "neutral" in order to match the assumptions of the comparison software. Averaged absolute distances were calculated between the simulated and the estimated competence parameters. Table 2 contains the mean of these AADs over the runs, their standard deviations in brackets. The answer key items were always recovered correctly by both programs.

As we can see both programs do rather well in recovering the simulated parameters, the Bayesian method does only slightly, although consistently better (it was more accurate in each of these runs). One reason for that could be that when drawing inferences in the Bayesian framework, negative competency estimates cannot occur, as they are not in line with the specification of the prior distributions. UCINET reported negative competency estimates for almost all runs. According to model assumptions underlying CCT, however, negative competencies are not allowed, therefore UCINET also warned that the basic CCT assumption might have been violated (eigenvalue-ratio was above 3 in all of these cases). Since we worked with simulated data based on the same answer key for all informants, it is unlikely that the single culture assumption was violated, especially because we even tried to reduce the possibility of simulation errors that could lead to negative estimates by sampling competency parameters larger than 0.1. Such inconsistencies in the parameter estimates based on the classical framework suggest that the Bayesian statistical framework is more disciplined and coherent.

Conclusions

CCT has been used extensively in field research in the last couple of decades. We believe that new developments in the field of statistical theory and computer science can help us improve upon the range of models that researchers can exploit for studying various questions of cultural consensus. Therefore we provided an introduction into the Bayesian modeling framework that is gaining more and more traction as a method for statistical inference in several fields. While we consider the current encounter with Bayesian statistics sufficient for ethnographers to deal with Bayesian CCT inference, we encourage readers to further investigate the possibilities of the Bayesian framework.

The Bayesian approach allows one to examine properties of the posterior distribution of a parameter rather than just a point estimate. In addition, in cases with a small number of informants the extra flexibility in allowing heterogeneity in the guessing process and in the item difficulty may contribute to better reconstruction of the answer key than the classical approach to estimation. Finally the matching and covariance methods in ANTHROPAC and UCINET are specific to di-

chotomous true/false or multiple choice response data. The Bayesian approach to inference can be developed for any CCT model, although for more complex models more complex MCMC samplers may need to be developed (e.g., Batchelder, Strashny, & Romney, 2010).

The BCCT software has been recently developed and although it has been thoroughly tested it might still run into unexpected errors. We encourage all users to send e-mails to the first author about problems as they arise, so that these issues can be mended. In turn updates for BCCT will be available for download.

We intend the current software package to be a starting point towards developing a large variety of user-friendly BCCT toolboxes for CCT models for questionnaires requiring other types of responses, e.g. Likert, ranking, matching, and continuous response scales. Based on feedback from users more complex BCCT packages will be written to enable researchers to investigate even more complex CCT modeling strategies, including hierarchical models and covariate analysis.

References

Batchelder, W. H., & Anders, R. (2012). Cultural consensus theory: Comparing different concepts of cultural truth. *Journal of Mathematical Psychology*. (Under revision)

Batchelder, W. H., & Romney, A. K. (1988). Test theory without an answer key. *Psychometrika*, *53*, 71-92.

Batchelder, W. H., Strashny, A., & Romney, A. K. (2010). Cultrual consensus theory: Aggregating continuous responses in a finite interval. In S. K. Chai, J. J. Salemo, & P. L. Mabry (Eds.), *Social computing, behavioral modeling and prediction* (pp. 98–107). New York: Springer.

Borgatti, S. P. (1996). *Anthropac 4.0.* Natick, MA: Analytic Technologies.

Borgatti, S. P., Everett, M. G., & Freeman, L. C. (2002). *UCINET for Windows: Software for social network analysis*. Harvard, MA: Analytic Technologies.

Comrey, A. L. (1962). The minimum residual method of factor analysis. *Psychological Reports*, *11*, 15-18.

Gamerman, D., & Lopes, H. F. (2006). *Markov chain Monte Carlo: Stochastic simulation for Bayesian inference (2nd ed.)*. Boca Raton, FL: Chapman & Hall/CRC.

Gelman, A., Carlin, J., Stern, H., & Rubin, D. (2004). *Bayesian data analysis*. New York: Chapman & Hall.

Gilks, W., Richardson, S., & Spiegelhalter, D. (1996). *Markov chain monte carlo in practice.* London: Chapman & Hall.

Gill, J. (2002). *Bayesian methods: A social and behavioral sciences approach.* Boca Raton, FL: Chapman & Hall/CRC.

Hopkins, A. (2011). Use of network centrality measures to explain individual levels of herbal remedy cultural competence among the Yucatec Maya in Tabi, Mexico. *Field Methods*, *23*(3), 307-328.

Hruschka, D. J., Sibley, L. M., Kalim, N., & Edmonds, J. K. (2008). When there is more than one answer key: Cultural theories of postpartum hemorrhage in Matlab, Bangladesh. *Field Methods*, *20(4)*, 315–327.

Jackman, S. (2009). *Bayesian analysis for the Social Sciences.* New York: John Wiley & Sons.

Karabatsos, G., & Batchelder, W. H. (2003). Markov chain estimation methods for test theory without an answer key. *Psychometrika*, *68*, 373-389.

Kruschke, J. K. (2011). *Doing Bayesian data analysis : A tutorial with R and BUGS.* New York: Academic Press.

Lee, M., & Wagenmakers, E. (2012). *Bayesian Cognitive Modeling: A Practical Course.* (Forthcoming.)

Lunn, D. J., Thomas, A., Best, N., & Spiegelhalter, D. (2000). WinBUGS- a Bayesian modeling framework: Concepts, structure, and extensibility. *Statistics and Computing*, *10*, 325-337.

Macmillan, N. A., & Creelman, C. D. (2005). *Detection theory: A users guide (second ed.).* Mahwah, N.J.: Erlbaum.

Miller, E. (2011). Maternal health and knowledge and infant health outcomes in the Ariaal people of northern Kenya. *Social Science & Medicine*, *73*(8), 1266 - 1274.

Plummer, M. (2011). *Rjags: Bayesian graphic models using MCMC.* (R package version 2.2.0-3 (http://CRAN.R-project.org/package=rjags))

Rasch, G. (1960). *Probabilistic models for some intelligent and attainment tests.* Copenhagen, Denmark: Danish Institute for Educational Research.

Robert, C. P., & Casella, G. (2004). *Monte Carlo statistical methods.* New York: Springer.

Romney, A. K. (1999). Cultural consensus as a statistical model. *Current Anthropology*, *40*, 103-115.

Romney, A. K., & Batchelder, W. H. (1999). Cultural consensus theory. In R. Wilson & F. Keil (Eds.), *The MIT encyclopedia of the cognitive sciences* (p. 208-209). Cambridge, MA.: The MIT Press.

Romney, A. K., Weller, S. C., & Batchelder, W. H. (1986). Culture as Consensus: A theory of culture and informant accuracy. *American Anthropologist*, *88*, 313-338.

Spiegelhalter, D. J., Best, N. G., Carlin, B. P., & van der Linde, A. (2002). Bayesian measures of model complexity and fit (with discussion). *Journal of the Royal Statistical Society, Series B*, *6*, 583–640.

Weller, S. C. (1984). Cross-cultural concept of illness: Variation and validation. *American Anthropologist*, *86*, 341-351.

Weller, S. C. (2007). Cultural consensus theory: Applications and frequently asked questions. *Field Methods*, *19*, 339-368.