Communications of the Association for Information Systems
CAIS

Bayesian Structural Equation Models for Cumulative Theory Building in Information Systems – A Brief Tutorial using BUGS and R

Joerg Evermann

Faculty of Business Administration, Memorial University of Newfoundland, Canada

jevermann@mun.ca

Mary Tate

School of Information Management, Victoria University of Wellington, New Zealand

Abstract:

Structural equation models (SEM) are frequently used in Information Systems (IS) to analyze and test theoretical propositions. As IS researchers frequently reuse measurement instruments and adapt or extend theories, it is not uncommon for a researcher to re-estimate regression relationships in their SEM that have been examined in previous studies. We advocate the use of Bayesian estimation of structural equation models as an aid to cumulative theory building; Bayesian statistics offer a statistically sound way to incorporate prior knowledge into SEM estimation, allowing researchers to keep a "running tally" of the best estimates of model parameters.

This tutorial on the application of Bayesian principles to SEM estimation discusses when and why the use of Bayesian estimation should be considered by IS researchers, presents an illustrative example using best practices and makes recommendations to guide IS researchers in the application of Bayesian SEM.

Keywords: quantitative analysis, structural equation models, Bayesian statistics, tutorial

Note: All scripts in the paper are available at http://staff.sim.vuw.ac.nz/mary-tate

Volume xx, Article x, pp. xx-xx, Month Year

The manuscript was received mm/dd/yyyy and was with the authors x months for y revisions. [Note: this box is used only for peer-reviewed papers.]

Communications of the Association for Information Systems

Communications of the Association for Information Systems

CAIS

Communications of the Association for Information Systems

Abstract:

I. INTRODUCTION

Theories are statements of causal relationships between constructs [Whetten, 1989; Gregor, 2006]. Constructs are imbued with meaning in part by their relationship with other constructs and their relationship with observations. In other words, besides the relationships specified in the “structural” model between one construct and another, the relationships in the “measurement” model (those between constructs and observations) are also theoretically interesting and important constituents of the theory.

Constructs are typically represented in statistical models as latent variables (SEM), composites (PLS), components (PCA) or common factors (EFA). These constructs are related to each other and to observed variables, which represent a construct's measures or indicators, by linear or non-linear relationships. The relationships are parameterized and the parameter values can be estimated using a range of statistical techniques.

IS researchers are encouraged to adapt and extend existing theories and measurement instruments in order to build cumulative knowledge. This advice frequently leads to situations where the same parameter value is estimated repeatedly. For example, there are a host of studies that build on or adapt some aspect of the Technology Acceptance Model (TAM), one of the most widely cited theories in IS. Between 2004 and 2011 (inclusive), we have identified 43 empirical studies in the top IS journals (MISQ, JMIS, ISR, JAIS, and ISJ) that reuse some of the TAM constructs and TAM indicators developed by Davis [1989] and Davis et al. [1989]. Given the extensive history of parameter estimation and consequently our knowledge of previously estimated values, researchers face the question of what to do with this prior knowledge. More importantly, as we show later in the paper (Table 6, Figure 1), the parameter estimates reported by these studies differ widely and the differences are statistically significant.

One option is to ignore previously estimated parameter values and only focus on the statistical significance of the parameters in the current study. This is the de-facto standard in IS research, but can lead to a situation where new estimates differ significantly, from previous estimates. Another alternative for the measurement model, but not for the structural model, is to simply omit the observed variable if it is particularly “badly behaved”. However, we agree with Evermann and Tate [2011] who argue that all data deserve an explanation and researchers should not omit data merely because it does not fit with pre-existing expectations. Ultimately, ignoring differences in parameter estimates can lead to measurement instability, if it occurs in the measurement model, or to divergent theoretical conclusions, if it occurs in the structural model. In other words, rather than building cumulative knowledge, we accumulate different parameter estimates without being able to reconcile them in a sound and systematic way.

In this tutorial we present a way to include our prior knowledge into the parameter estimation process, so that new estimates are based not only on the new data, but also on our existing knowledge about the likely values of the parameters. Bayesian statistical methods provide researchers with a statistically sound way of doing this. One can think of this as new studies updating our best estimates of the parameter values, in effect allowing us to keep a “running tally” of our model parameter estimates.

Structural equation models with latent variables are usually estimated in the IS literature either by means of covariance-based techniques (using software like LISREL, EQS, AMOS, Mplus, etc.) or by using partial-least squares approaches (with software like PLS-Graph, SmartPLS, WarpPLS, etc.), which are based on a frequentist concept of probability. Bayesian estimation provides a third alternative to these methods with some pragmatic advantages for researchers which are not offered by currently used methods. These include the ability to integrate prior knowledge or assumptions into our model estimation: Bayesian estimation can estimate missing values as part of the estimation process, rather than in a separate, prior step, as is done by imputation methods. It also provides the ability to explicitly model the missingness of MCAR and NMAR data. As part of the Bayesian estimation, latent variable scores are explicitly estimated. In fact, Bayesian estimation views a latent variable simply as one for which all value are missing. Especially for CFA (confirmatory factor analysis models), Bayesian estimation relaxes traditional model identification requirements, so that it is possible to estimate cross-loadings. Bayesian estimation also relaxes normality assumptions and allows the researcher to explicitly specify appropriate probability distributions. As a consequence, Bayesian estimation is naturally suited for ordinal data, such as from Likert scales, binary variables, and IRT (item response theory models). We discuss these and other advantages over existing methods in Section III. While Bayesian statistics itself are not new, there are few applications in the Information Systems literature. A search of the AIS electronic library (including CAIS, JAIS and AIS conference proceedings) with the keyword “Bayesian” showed a handful of Bayesian estimation of regression models that do not include latent variables, especially multi-level models, or the use of Bayesian networks in information systems engineering contexts. More specifically, we are aware of only one other paper in the IS literature that discusses a Bayesian approach in the context of structural equation modelling. Zheng and Pavlou [2009] offer a novel and effective method for inferring possible and plausible structural equation models from a given data set. However, their paper is very different from this tutorial in that it does not apply a Bayesian approach to the estimation of parameters in a structural equation models. Existing introductory texts on Bayesian methods [e.g. Congdon, 2006; Gelman et al., 2004] typically focus on regression models, especially multi-level regression models, that do not include latent variables. Given the extent of structural equation models (SEM) in information systems, this tutorial is specific to the use of Bayesian estimation for SEM.

The remainder of this tutorial is structured as follows. To establish some basic terminology, we first introduce the Bayesian principle of conditional probabilities on which all of Bayesian statistics is founded. To help researchers decide when Bayesian estimation may be appropriate, we then discuss some of the advantages and drawbacks of Bayesian statistics. The next section then provides an introductory example for the reader to become familiar with model specification and estimation in the Bayesian approach. We use an example from the Technology Acceptance Model (TAM) for illustration purposes. Following this, the main section of this tutorial presents a general procedure for Bayesian estimation and uses an in-depth example to guide the reader through best-practices of estimation and diagnostics. Our conclusion focuses on specific recommendations to researchers who wish to use Bayesian structural equation models.

II. BAYESIAN PRINCIPLES

In this section, we introduce the basic idea of Bayesian statistical models and focus on conceptual understanding of the principles. We show how Bayesian statistics differs from the traditional frequentist perspective and focuses on different goals and interpretations.

Conditional Probabilities

Bayesian statistics are based on Bayes' principle of conditional probabilities. In its simplest form, this can be written as follows:

$$p(\theta \mid x)p(x) = p(x \mid \theta)p(\theta)$$

In this equation, $p(\theta| x)$ is the *posterior probability* that the model parameter θ takes on a certain value, conditional on the observation of data x. The term $p(x |\theta)$ represents the *probability* of observing data x conditional on the value of model parameter θ (i.e. the likelihood of x). The term $p(\theta)$ is the *prior probability* of the values of model parameter θ and the term $p(x)$ is the probability of observing the data x not conditioned on any parameter θ.

In general, the terms θ and x are sets (vectors) of model parameters and observations, for example, θ represents all loadings, latent covariances, and error covariances in a structural equation model (and also the latent variables themselves, as we shall see below). The data x includes all observed variables in a structural equation model.

We do not need to consider $p(x)$ as this probability is not parameterized in terms of θ and therefore has no bearing on the estimation of the values for θ. The above equation can therefore be rewritten as a proportionality statement:

$$p(\theta \mid x) \propto p(x \mid \theta)\, p(\theta)$$

The second form of Bayes' principle shows that our belief about the probability of parameter values after observing certain data (posterior belief) depends on our prior belief about the probability of parameter values and the probability of the observed data under that prior probability. In other words, the posterior beliefs are an update of the prior beliefs after observation of data. For specific Bayesian models, the researcher assumes a probability distribution for $p(x \mid \theta)$ based on theoretical considerations and the distribution of $p(\theta)$ reflects the existing, prior knowledge about parameter values.

Bayesian Inferences

In the traditional frequentist approach to statistical inference, the probability of an event is interpreted as the relative frequency of an event given an infinite sequence of samples from an identical (i.e. fixed) probability distribution. This notion is made explicit in Null-hypothesis significance testing (NHST), where the researcher asks how likely it is to observe the estimated parameter values (i.e.the data), if a Null-hypothesis (which defines the assumed sampling distribution) were true. If this likelihood is below a certain threshold α (e.g. 0.05), the researcher rejects the Null-hypothesis. In other words, the focus in the frequentist paradigm is on $p(x |\theta)$ (more specifically on $p(x |\theta_0)$), not on $p(\theta|x)$ as in the Bayesian approach. In the frequentist approach, the data is treated as random by assuming that it is a random sample from a hypothetical probability distribution; the model parameters are assumed as fixed, e.g. in the form of a Null-Hypothesis that fixes $\theta = 0$. Importantly, because the p-value in NHST is derived under the assumption that the Null hypothesis is true, in rejecting the Null hypothesis researchers lose the ability to make any statements about the probability of the observed effect (or any effect, including the Null effect) [Zyphur and Oswald, 2013]. The only statement is admits is that the Null hypothesis is unlikely. Given that point hypotheses are very unlikely to be strictly true, this outcome is not very satisfying [Zyphur and Oswald, 2013].

In contrast, the Bayesian approach focuses directly on the probability of an effect, i.e. on the probability of observing the estimated parameters given the data, i.e. on $p(\theta |x)$. Further, in addition to the sampling uncertainty of the data, the Bayesian approach also treats the model parameters as uncertain, i.e. assumed as following a probability distribution, namely the prior distribution $p(\theta)$. This more realistic treatment allows the model to make a statement about the probability of the obvserved effect, rather than simply rejecting an (unrealistic) Null-hypothesis.

This difference in interpretation is evident in the reporting of Bayesian analyses. Whereas the frequentist researcher provides the p-value to show whether the Null-hypothesis should be rejected, the Bayesian provides a point estimate for the probability of the observed effect given the data $(p(\theta|x)$, as either the mean or mode of the posterior probability distribution. Additionally, Bayesian researchers report *credibility intervals* (e.g. the 2.5% and 97.5% percentile) around this point estimate to show the credible range of the parameter value given the observed data. While these credibility intervals can be used for significance testing in the same way as a confidence interval in NHST, this is not the main goal of Bayesian analysis.

III. WHEN TO USE BAYESIAN ESTIMATION OF STRUCTURAL EQUATION MODELS

While we have motivated this paper by appealing to our desire for integrating prior knowledge into our model estimation, Bayesian estimation of structural equation models offers other advantages as well.

- Integration of prior knowledge into the estimation process

  In contrast to covariance-based or partial least squares methods, the Bayesian approach can explicitly incorporate prior knowledge of parameter values into the estimation [Kruschke et al., 2012; Scheines et al., 1999]. Prior knowledge is specified by the probability distribution of model parameters. The mean and variance of these prior distributions reflect our “point beliefs” and the certainty about or the precision of our prior knowledge.

- Integrated treatment of missing values

  In contrast to missing value imputation prior to model estimation, Bayesian estimation allows missing values to be estimated as part of the estimation of the overall model [Asparouhov and Muthen, 2010a; Lunn et al., 2013]. Hence, missing value estimation is able to use the model structure, rather than relying only on sample information, such as when using the EM algorithm. This covers MCAR[1] (missing completely at random) (missing at random) and MAR data. Moreover, the flexibility of Bayesian models allows the researcher to also specify a mechanism to model the missingness, covering NMAR (not missing at random) data [Lee, 2007; Lunn et al., 2013; Song and Lee, 2008; 2012].

- Explicit estimation of latent variable scores

  Latent variables are explicitly modeled and estimated in Bayesian statistics. In fact, the treatment of latent variables differs little from the treatment of missing values, and one can view a latent variable as one for which all value are missing. Conceptually, missing values and latent variables are closely related in Bayesian estimation [Asparouhov and Muthen, 2010b, Lee, 2007; Song and Lee, 2008].

- Relaxation of model identification requirements

  Traditional estimation methods require a model to be identified. For example, it is impossible in covariance-based methods to estimate a CFA (confirmatory factor analysis) model in which all cross-loadings are free parameters. Bayesian estimation allows researchers to estimate non-identified models if the prior parameter distributions sufficiently constrains their values. For example, it is possible to estimate CFA models with cross-loadings that are expected to be approximately zero, but are allowed to vary somewhat around these values. Such models are argued to be more appropriate in expressing a researcher’s theoretical expectations about cross-loadings [Asparouhov and Muthen, 2010a; Scheines et al., 1999; Muthen and Asparouhov, 2012].

- Accuracy at small sample sizes and no reliance on asymptotic (large sample) validity of estimates

  Covariance-based methods make assumptions about the asymptotic distribution of parameter estimates and test statistics, which are strictly only valid for very large samples. Partial least squares techniques make no such assumptions for the test statistics, but the “consistency at large” theorem means that PLS estimates are only unbiased for very large sample. In contrast, Bayesian estimation does not make such large sample, asymptotic assumptions for the distribution of model parameter and variable estimates [Asparouhov and Muthen, 2010a; Kruschke et al., 2012; Rupp et al., 2004; Scheines et al., 1999]. Moreover, Bayesian

[1] Missing completely at random denotes data whose probability of missing does not depend on observed or unobserved data. Missing at random denotes data whose probability of missing depends on the observed data. MCAR and MAR data are called “ignorable” because they do not provide any information on the data.

estimates have been noted as more accurate for small sample sizes than maximum-likelihood (ML) estimates [Asparouhov and Muthen, 2010a].

- Relaxation of normality assumptions

  Especially covariance-based methods make assumptions about the (multivariate-)normal distribution of variables to arrive at well-defined test statistics. Because the probability distributions for different variables are explicitly modeled in Bayesian estimation, it is possible to assign other than normal distributions, if these are more appropriate [Scheines et al., 1999], either based on prior knowledge or theoretical considerations. However, for the estimation to remain possible, the distributions that can be modeled are often restricted to so-called conjugate distribution (see below).

- Easy extensibility to non-continuous observed data

  While some approaches exist to extend covariance analysis to ordinal data, this can be done more naturally and explicitly in Bayesian estimation [Asparouhov and Muthen 2010a; 2010b; Lee, 2007; Lee at el., 2010; Song et al., 2001]. This allows the easy expression of IRT (item-response-theory) models [Rupp et al., 2004] as well a more faithful representation of Likert scales or binary latent variables. Bayesian estimation has been shown to be more accurate than covariance-based methods for categorical data with missing variables [Asparouhov and Muthen, 2010a].

- Easy extensibility to multi-level models

  While multi-level structural equation models have not been used to great extent in the IS literature, they may be appropriate as organizational theories in IS may include individual-level, firm-level and industry-level constructs and relationships. Because the relationships between multiple levels of analysis are explicitly modeled and the estimation relies on iterative sampling of (relatively) simple distributions, it is possible to easily express multi-level statistical models in Bayesian approaches [Browne and Draper, 2006; Asparouhov and Muthen, 2010a; Song and Lee, 2008; Yuan and MacKinnon, 2009]. An easy way to model and estimate multi-level relationships may lead to more applications of these models in an IS context.

- Convergence with traditional methods

  Bayesian estimates of parameter values converge to those of traditional methods. Specifically, with increasing sample size, Bayesian estimates converge asymptotically on maximum-likelihood estimates [Lunn et al., 2013]. Intuitively, this expresses the increasing weight of evidence by the data over prior assumptions. Further, a non-informative prior distribution can be chosen to further reduce the effect of the prior distribution.

However, while Bayesian estimation has many advantages over traditional methods, it also has some drawbacks. The most important ones are the following:

- Large computational resource requirements

  Bayesian estimation uses an iterative method of sampling parameter estimates from posterior probability distributions. The computational requirements are generally larger than for covariance-based or partial-least-squares estimation. Further, because all latent variables in the model, including errors, are estimated during each iteration, the resulting data volume is significantly larger. However, with the increase in personal computer power in recent years, it is now feasible to estimate even complex models in a few seconds. Moreover, in some cases, Bayesian estimation is shown to be more computationally efficient than traditional estimation approaches [Asparouhov and Muthen, 2010a]

- Dependence of results on prior distributions (even uninformative ones)

  Even as Bayesian estimates are noted as more accurate than ML estimates for small samples, Bayesian results for small sample sizes may depend on the specified prior probability distributions of model parameters, especially and even for different uninformative distributions [Asparouhov and Muthen, 2010a]. While there are no guidelines as to which models are affected at which sample size, researchers are urged to check for prior assumption dependence by estimating the model with different prior knowledge assumptions [Asparouhov and Muthen, 2010a].

- Lack of overall model test (i.e. overidentification test as in covariance analysis)

  In covariance-analysis, the $\chi 2$ test of model fit (and its robust versions) provides an easy diagnostic tool to assess the fit of the estimated model with the sample data [Evermann and Tate, 2011]. There is no such statistical test for Bayesian structural equation models. However, the "posterior predictive p-value" (PPP) [Gelman et al., 1996; Scheines et al., 1999; Muthen and Asprouhov, 2012] has been argued to serve a similar role and might be used as a test of model fit: "The LRT [likelihood ratio test, i.e. $\chi 2$ test], appears to be more powerful than the PPP … but this is at the cost of incorrect type I error for small sample cases… On the other hand, the PPP is always reliable and for sufficiently large sample size has the same performance as the LRT" [Asparouhov and Muthen, 2010a, p. 31].

Recommendation: Use Bayesian analysis for

- non-standard models that are difficult to express in covariance or partial-least squares models (such as multi-level models, underidentified models, models with missing values and/or non-continuous variables)
- estimation that allows the use of prior knowledge about parameter values, and/or
- estimation from small sample sizes

What Bayesian Estimation is Not

Bayesian estimation can be related to other concepts in the research methods literature. First, Bayesian statistics is not a research methodology. The concept of a research method is broader and encompasses an underlying ontology and epistemology that guide the researcher in asking research questions, collecting data, analyzing data, and interpreting results. In contrast, Bayesian estimation, in its narrowest interpretation, is a statistical tool for data analysis. In a slightly broader interpretation, it also suggests a different interpretation of the results, differing from the frequentist notion of probability.

Bayesian estimation is not a method that is limited to survey research. Bayesian statistics are suitable for the analysis of other types of data [Congdon, 2006; Gelman et al. 2004] and it is up to the researcher to specify the appropriate statistical model. However, this tutorial is concerned only with structural equation models.

Bayesian analysis of structural equation models is not a new way of doing survey research. Recommendations for instrument design and data collection remain unaffected by the type of subsequent data analysis method. Bayesian estimation of SEM models also does not affect the notions of reliability or validity of measurement instruments. The substantive interpretation of the model and its estimated parameters, in terms of validity and reliability of indicators [e.g. Gefen et al., 2011] is based on the estimates of parameter values, and does not depend on the type of estimation as long as the estimation produces valid estimates (e.g. asymptotically unbiased estimates).

Bayesian estimation is not meta-analysis, nor an alternative to meta-analysis. Whereas meta-analysis is concerned only with a few important parameters and does not typically include new data, Bayesian estimation is concerned with all parameters of a model and requires a data set to analyze.

Finally, Bayesian estimation is not a "silver bullet" that fixes all shortcomings of existing methods. In fact, the advantages and disadvantages we have outlined should be used as guidelines by researchers to identify if Bayesian estimation is suitable, and whether it provides advantages over traditional methods in particular applications.

Relationship to Meta-Analysis

As can be seen from our discussion this far, Bayesian estimation, in that it allows researchers to synthesize prior estimates, is related to meta-analytic techniques. However, meta-analysis aims only to synthesize existing estimates, rather than to incorporate this existing knowledge into the estimation of a new model [King and He, 2005]. Meta-analysis is appropriate for synthesizing an existing corpus of studies, but is not a technique for model estimation. In contrast, Bayesian estimation is not suitable to synthesizing a set of existing studies, but is concerned with the estimation of particular model with a specific sample.

Meta-analyses are typically concerned with only a few model parameters of theoretical interest, whereas Bayesian analysis estimates all parameters in a model. For example, a meta-analysis of the Technology Acceptance Model [Ma and Liu, 2004], a model that we also use for illustration purposes later, only examines the structural relationships. On danger in this is what King and He [2005] call the "apples and oranges" issue, where researchers may aggregate results from studies with incommensurable measures. By estimating complete models including measurement relations, rather than focusing on a few structural relationships, Bayesian researchers are at least aware of the measurement model and can exclude studies with very different operationalizations of constructs.

Meta-analysis can also be used with structural equation models [Joseph et al., 2007]. Here too, the focus is typically on structural relationships, and again does not allow the estimation of a new model given the known information.

Meta-analyses can be conducted even if the models are very different from each other, as long as they contain the relationship(s) of interest. Similarly, prior estimates need not be available for all parameters in Bayesian model, as the Bayesian approach allows the use of uninformative priors when no such knowledge is available.

In summary, we view meta-analysis as a possible pre-cursor to Bayesian estimation. It provides the researcher with a systematic method to identify, collect, and aggregate the parameter estimates from different studies. Such systematically derived prior knowledge can then be modeled as part of the Bayesian structural equation model. Hence, for integration of prior studies, the researcher chooses a meta-analytic technique. If, in addition, a model is to be estimated with a new data set, a subsequent Bayesian approach can integrate the prior knowledge from the meta-analysis.

Recommendation:

- Meta-analysis is a valuable pre-cursor to Bayesian estimation
- Use the meta-analytic results to aggregate data from former studies for use in Bayesian estimation

IV. A SIMPLE ILLUSTRATION OF BAYESIAN ESTIMATION

We presented the basic principle of Bayesian statistics in Section II. This section illustrates how that principle is applied to the estimation of a simple linear regression model. The aim of the section is to familiarize the reader with Bayesian terminology and equip the reader with a basic understanding of Bayesian model specification and model estimation. While we illustrate the mathematical specification of the model and the different probabilities and likelihoods, we do not provide any derivations, which are conceptually simple but lengthy and somewhat tedious. They can be found in any good textbook, such as Congdon [2006] or Gelman et al. [2004] for regression models, and Lee [2007] or Song and Lee [2012] for structural equation models.

Consider a simple linear regression example including two observed variables. For example, in an application to the IS context, y_i might be the perceived usefulness in the Technology acceptance model (TAM), while x_i might represent the perceived ease of use of that technology[2].

$$y_i = \beta x_i + \varepsilon_i \qquad \text{(Equation 1)}$$

Further, we make the standard assumptions that the errors (residuals) are normally distributed with mean zero and variance σ^2:

$$\varepsilon_i \sim N(0, \sigma^2) \qquad \text{(Equation 2)}$$

Rewriting equations 1 and 2 in terms of probability distributions shows that the observations Y are normally distributed with mean $X\beta$ and variance σ^2:

$$Y \sim N(X\beta, \sigma^2) \qquad \text{(Equation 3)}$$

Here, Y and X are vectors of the y_i and x_i respectively. Thus, the likelihood function is the following normal density:

$$p(Y, X \mid \beta, \sigma^2) \propto (\sigma^2)^{-\frac{n}{2}} \exp\left(-\frac{1}{2\,\sigma^2}(Y - X\beta)^T (Y - X\beta)\right) \qquad \text{(Equation 4)}$$

We now need to specify our prior knowledge about the parameters β and σ^2 by specifying a distribution for the prior probability $p(\beta, \sigma^2)$. Assuming that the prior mean and variance are independent, we can simplify the prior distributions for easier specification:

$$p(\beta, \sigma^2) = p(\beta \mid \sigma^2) p(\sigma^2) \qquad \text{(Equation 5)}$$

Here, $p(\sigma^2)$ represents our prior knowledge about the errors (residuals) in the regression of attitude towards technology on perceived usefulness. For our TAM example we might come to have expectations based on e.g. the mean and range of reported r^2 values in published studies of perceived ease of use and perceived usefulness. Similarly, $p(\beta \mid \sigma^2)$ represents our prior knowledge about the regression coefficient in the regression of perceived usefulness on perceived ease of use. Again, we might come to have expectations based on the reported values in

[2] In the interest of presenting a running example, we use the TAM constructs here as observed variables, even though they should be modelled as latent variables, as we do in later sections. For this simple initial example, one could assume these variables as sum scores of their indicators.

prior studies on TAM. Alternatively, for either or both distributions, researchers might choose uninformative prior distributions, if no previous knowledge is available.

Conjugate Prior Distributions and Uninformative Priors

To make the estimation tractable, the prior probability is typically assumed to have a *conjugate distribution* to the likelihood function $p(x \mid \theta)$. This means that the product $p(x \mid \theta)\, p(\theta)$ (i.e. the posterior probability) is of the same distribution family as $p(\,\theta)$. Table 1 presents a list of frequently used conjugate prior distributions in Bayesian estimation.

For our case of a normal likelihood function (Equation 4), appropriate conjugate prior distributions are another normal distribution for $p(\beta \mid \sigma^2)$ and an inverse Gamma distribution for $p(\sigma^2)$:

$$p(\beta \mid \sigma^2) \sim N(\mu, \nu) \qquad \text{(Equation 6)}$$

$$p(\sigma^2) \sim invGamma\ (a, b) \qquad \text{(Equation 7)}$$

Each of these probability distributions has their own parameters, called *hyper-parameters*, which affect the mean and variance of the distribution (Table 1).

In our TAM example, the hyper-parameters μ and ν in Equation 6 represent our prior knowledge of the mean an variance for the regression coefficient β. As an example, from Table 1 we see that setting $\mu = 0.5$ and $\nu = 5$ for the hyper-parameters in Equation 6 will yield a mean of 0.5 which represents our prior “point belief” of β. The variance of the prior distribution of 5 represents our certainty (or uncertainty) about our prior “point belief”. For our particular example of the regression of perceived usefulness on perceived on ease of use, we look towards an existing meta-analytic study [Ma and Liu, 2004]. In that study, the authors analyzed 33 correlations between the two variables from 21 studies. With correlations being equal to standardized regression coefficients in a two-variable linear model we may use their point estimate of 0.50 for the hyper-parameter μ and their variance estimate of 0.038 for the hyper-parameter ν if we use standardized data.

Similarly, in our TAM example the parameters a and b in Equation 7 represent our prior knowledge of the means and variances for the variances of the regression errors ε_i in the regression of attitude on perceived usefulness. For example, from Table 1 we see that setting $a = 3$ and $b = 1$ yields a mean of 0.5 and a variance of 0.25 as our prior estimate of the error variances. Unfortunately, the meta-analysis by Ma and Liu [2004] does not provide any data on the r^2 for a simple regression of perceived usefulness on perceived ease of use. In cases like this, where there is no prior knowledge, or our prior knowledge is very uncertain, researchers can use *non-informative* distributions, e.g. an normal distribution with a very large variance, or a uniform distribution. The last column in Table 1 shows frequently used choices for uninformative prior distributions. In our TAM example, if we had no prior beliefs about the regression parameters of attitude on perceived usefulness, we might specify $\mu = 0$ and $\nu = 10^{10}$ which describes a zero-centered distribution with very large variance, i.e. it is essentially flat and provide no useful information about the parameter β that it describes. Similarly, if we had no prior knowledge about the error (residual) variance in the regression, we might choose an uninformative prior gamma distribution with $a = -1$ and $b = 0$ which yields a uniform density of 1.

With the choices of prior distributions in Equations 6 and 7, and the likelihood function as in Equation 4, one can analytically derive the form of the normal posterior probability distribution. Conceptually simple, the derivation is too space-consuming to show.

We emphasize that while conjugate priors are useful because they yield analytically derivable, closed form expressions for the posterior, and thus make estimation easier, the choice of conjugate priors is not a strict requirement. Even when a closed form expression of the posterior is not available, one can sample from it using Markov Chain Monte Carlo methods, and particularly the Gibbs sampler, presented next.

Table 1: Typical conjugate prior distributions used in Bayesian estimation (choices for uninformative priors from [Asparouhov and Muthen, 2010b])

Likelihood function	Conjugate prior distribution	Mean	Variance	Example choices for uninformative prior distribution
Normal (with known variance)	Normal $N(\mu, \nu)$	μ	ν	$N(0, 10^{10})$
Normal (with known mean) (parameterized using mean and variance)	Inverse Gamma $IG(a, b)$	$\frac{b}{a-1}$	$\frac{b^2}{(a-1)^2(a-2)}$	$IG(0,0)$ $IG(-1,0)$ $IG(0.001, 0.001)$

Table 1: Typical conjugate prior distributions used in Bayesian estimation (choices for uninformative priors from [Asparouhov and Muthen, 2010b])

Likelihood function	Conjugate prior distribution	Mean	Variance	Example choices for uninformative prior distribution
Normal[3] (with known mean) (parameterized using mean and precision)	Gamma $G(a,b)$	$\frac{a}{b}$	$\frac{a}{b^2}$	$G(0.001, 0.001)$
Multivariate Normal (parameterized using means, covariances	Inverse Wishart[4] $IW(\Omega_{\mathrm{p}}, \mathrm{d})$	$\frac{\Omega}{d-p-1}$	Proportional to $\frac{1}{(d-p)(d-p-1)^2(d-p-3)}$	$IW(0,-p-1)$ $IW(0,0)$ $IW(I,p+1)$
Multivariate Normal (parameterized using means, inverse covariances	Wishart[5] $W(\Omega_{\mathrm{p}}, \mathrm{d})$	$d\Omega$		
Exponential, Gamma	Gamma $G(a,b)$	$\frac{a}{b}$	$\frac{a}{b^2}$	$G(0.001, 0.001)$
	Uniform[6] $U(a,b)$	$\frac{1}{2}(a+b)$	$\frac{1}{12}(b-a)^2$	$U(-10^{10}, 10^{10})$ $U(0, 10^{10})$

Bayesian Estimation with the Gibbs Sampler

Having developed our statistical model and found a solution for the posterior probability, we are now in a position to estimate the parameter values from this posterior distribution. This occurs by sampling values of individual parameters from the posterior distribution one parameter at a time, a process referred to as *Gibbs sampling,* a form of a technique called *Markov Chain Monte Carlo (MCMC)* sampling. Using our example, we have analytically determined the posterior probability distribution to be normal (because of the normal likelihood and the conjugate prior distribution). We now iteratively sample values from this normal distribution, e.g. first for β from

$$p(\beta \mid \sigma^2, Y, X, a, b, \bar{\beta}, S) \qquad \text{(Step 1)}$$

and then for σ^2 from

$$p(\sigma^2 \mid \beta, Y, X, a, b, \bar{\beta}, S) \qquad \text{(Step 2)}$$

Every iteration comprises these two steps. In the first iteration, a starting value for σ^2 is either specified by the researcher, sampled from the prior distribution, or is the default set by the estimation software. After the first step samples a value for β, this value becomes input to step 2 in that same iteration and allows sampling of a value for σ^2. These sampled values form the input for the next iteration of these two steps. The iterations continue until the sampled values are stable. In our simple example, each sampling step samples a single parameter. In many models, multiple parameters have a joint distribution, so that values for a set of parameters will be sampled in each step.

In practice, it is common to begin multiple of these sampling chains from different starting values to ensure convergence of samples on the posterior parameter estimate. Final parameter estimates are then computed as the mean of the sampled values after a “burn-in” period where stabilization occurs and whose samples are discarded. Typically, there may be up to 10,000 iterations in each of three Markov Chains, with burn-in periods of between 2,000 and 5,000. These numbers indicate the substantial computational requirements for Bayesian statistics, especially for complex structural equation models with dozens or hundreds of parameters.

[3] In some Bayesian literature, the normal distribution is parameterized as $N(\mu, \nu^{-1})$ where ν^{-1} is the inverse variance, called *precision*.

[4] For the inverse Wishart distribution, Ω_p is a positive definite matrix of size p. The variance is a complex formula not shown here, but can be influenced by the choice of d as shown in the table.

[5] For the Wishart distribution, Ω_p is a positive definite matrix of size p. The variance is a complex formula not shown here.

[6] The uniform distribution is often used as a “pseudo conjugate” prior and is an intuitive uninformative distribution.

OpenBUGS Model and Script

Easy to use software for Bayesian SEM has only been developed relatively recently, in the form of the WinBUGS and OpenBUGS software [Lunn et al., 2013], and inclusion of Bayesian analysis in popular SEM software packages like MPlus. In this tutorial, we focus on the use of open-source software OpenBUGS for estimating Bayesian models, and the R system to analyze the results. OpenBUGS is an open-source version of the commercial WinBUGS software ("Bayesian Inference Using Gibbs Sampling"), originally developed by the biostatistics unit at Cambridge University. Model definitions are fully interchangeable between the two. Another open-source software that is very similar to both WinBUGS and OpenBUGS is JAGS ("Just Another Gibbs Sampler"). OpenBUGS model definitions are also usable with JAGS, and OpenBUGS scripts can easily be translated to JAGS scripts. Lunn et al. [2013] provide an introduction to BUGS, its syntax and a comparison of the three BUGS implementations (WinBUGS, OpenBUGS, JAGS).

Table 2: OpenBUGS model definition for the introductory example

Line	Model
1	`model {`
2	`  for(i in 1:N) {`
3	`    mu[i] <- beta * x[i]`
4	`    y[i] ~ dnorm(mu[i],psi)`
5	`  }`
6	`  beta ~ dnorm(0.5, 5)`
7	`  psi ~ dgamma(3, 1)`
8	`}`

Table 2 shows how our introductory TAM example is defined as an OpenBUGS model. The model definition begins with the `model` keyword in line 1. Line 2 shows that each individual observation is defined separately. Lines 3 and 4 show the definition of the y_i in the same form as we used in Equation 3. In other words, `mu[i]` in line 3 represents the expected observation βx_i and line 4 mirrors Equation 3. Lines 6 and 7 set up the prior probability distributions for the two model parameters in the same form as we have done in Equations 6 and 7.

One important aspect of the OpenBUGS specification is that OpenBUGS parameterizes the normal distribution using the mean and precision (inverse variance), instead of the more typical mean and variance. The relationship between the two is simple:

$$x \sim Gamma(a,b) \quad \rightarrow \quad \frac{1}{x} \sim InverseGamma(a,b) \qquad \text{(Equation 8)}$$

Thus, the specification `dnorm(mu[i], psi)` on line 4 uses mean `mu[i]` and *precision* `psi`. Accordingly, instead of an inverse gamma for the prior distribution of the variance, as in Equation 7, we use a gamma prior distribution for the precision `psi` (line 8). As per Equation 8, the specification `dgamma(3,1)` in line 7 for the precision parameter (which yields a mean and a variance of 3, see Table 1) is equivalent to inverse gamma specification on the variance parameter (and yields a mean of 0.5 and variance of 0.25, see Table 1).

This simple example shows that the model definition in OpenBUGS is very explicit in the sense that it is analogous to the mathematical definition of the model derived earlier. This has the advantage of being very flexible. For example, we could easily specify hetero-skedastic models by introducing different `psi` parameters for different observations in lines 4 and 8 of Table 2. It is also easy to see how a regression intercept could be added to the model in line 3, with the addition of an appropriate prior specification later in the model. On the other hand, this explicit specification requires an understanding of the mathematical concepts in this section.

Having developed the OpenBUGS model specification, the model can be estimated with the OpenBUGS software, controlled via a script. This script is shown in Table 3. Line 1 is used to specify the working directory where the model and data files are found. Line 2 loads the model and performs a syntactic check. The model data file is loaded in line 3. The data file must also include values for all constants in the model, e.g. the number of observations N, which is used in line 2 in Table 2. Line 4 compiles the model for three MCMC sampling chains. Initial values are automatically generated in line 8. Lines 9 and 10 control for which of the model variables samples are to be collected. Line 11 sets up the computation of the Deviance Information Criterion (DIC), an important diagnostic tool. We discuss DIC and other diagnostics later. Finally, line 13 writes the sampled values in CODA format (a format that is suitable for later analysis using the R software) to the specified file. Lines 14 and 15 print summary statistics for the sampled variables and the DIC, respectively.

Table 3: OpenBUGS script to control the estimation

Line	Script
1	`modelSetWD('OpenBUGSExample')`
2	`modelCheck('model1.txt')`
3	`modelData('data1.txt')`
4	`modelCompile(3)`
8	`modelGenInits()`
9	`samplesSet('beta')`
10	`samplesSet('psi')`
11	`dicSet()`
12	`modelUpdate(5000, 1, 1, 'F')`
13	`samplesCoda('*', 'codaoutput')`
14	`samplesStats('*')`
15	`dicStats()`

Recommendation: Use the OpenBUGS software for Bayesian estimation because it is

- Flexible (not limited to certain types of models)
- Expressive (provides a wide range of probability distributions with which to model)
- Extendable (researchers can provide user-defined probability functions)
- Free and open-source (and integrates well with the popular R statistical environment)
- Cross-platform (works well in a heterogenous IT environment)
- Scriptable (rather than relying on graphical user interfaces, scripts can ensure replicability of results)

V. BEST-PRACTICE EXAMPLE: BAYESIAN ESTIMATION OF TAM CONSTRUCTS

The previous section presented a simple illustration of how the Bayesian principle can be applied to a linear regression problem. That section has provided us with Bayesian terminology and equipped use with a basic understanding of Bayesian model specification and model estimation. In this section, we illustrate Bayesian best practices using a full example. We follow the steps in Table 4, which are generic steps for every Bayesian estimation, whether structural equation model or others.

We use the Technology Acceptance Model (TAM) as an illustrative example also in this section because its constructs have been measured consistently using the same measurement items across multiple studies. Thus, it provides a rich set of prior knowledge about parameter estimates for us to use. TAM focuses on the relationship among three constructs, Perceived Ease of Use (PEoU), Perceived Usefulness (PU) and Behavioral Intention to use (BI). In this section, we focus on a CFA (confirmatory factor analysis) of perceived usefulness and behavioral intentions, due to the availability of data for these constructs. Our example uses the TAM data from Chin et al. [2008], which was also used in [Evermann and Tate, 2011].

Table 4: Recommended process steps for Bayesian model estimation

Step 1	Specify the statistical model
Step 2	Identify prior knowledge and distributional assumptions
Step 3	Estimate model
Step 4	Assess MCMC convergence
Step 5	Remove burn-in iterations and thin samples
Step 6	Evaluate model quality

Step 1: Specify the statistical model

The two constructs of interest in the TAM model are traditionally measured by six observed indicators each [Davis, 1989; Davis et al. 1989]. The main difference to our earlier regression model is the inclusion of latent variables, i.e. variables for which data is missing. Latent variables in a Bayesian model are treated in a similar way to parameter estimates: they are assigned a probability distribution and their values are estimated as part of the model estimation process.

Table 5: CFA model definition in OpenBUGS (part 1, the basic statistical model)

Line	Model definition
1	`model {`
2	`  for(i in 1:N){`
3	`    #measurement equation model`
4	`    for(j in 1:P){`
5	`      y[i,j]~dnorm(mu[i,j],errorprec[j])`
6	`    }`
7	`    mu[i,1]<-lam[1]*xi[i,1]`
8	`    mu[i,2]<-lam[2]*xi[i,1]`
9	`    mu[i,3]<-lam[3]*xi[i,1]`
10	`    mu[i,4]<-lam[4]*xi[i,1]`
11	`    mu[i,5]<-lam[5]*xi[i,1]`
12	`    mu[i,6]<-lam[6]*xi[i,1]`
13	`    mu[i,7]<-lam[7]*xi[i,2]`
14	`    mu[i,8]<-lam[8]*xi[i,2]`
15	`    mu[i,9]<-lam[9]*xi[i,2]`
16	`    mu[i,10]<-lam[10]*xi[i,2]`
17	`    mu[i,11]<-lam[11]*xi[i,2]`
18	`    mu[i,12]<-lam[12]*xi[i,2]`
19	`    #structural equation model`
20	`    xi[i,1:2]~dmnorm(u[1:2],latprec[1:2,1:2])`
21	`  } #end of i`

The basic structure of the model specification is similar to the earlier one (Table 2) and is shown in Table 5. The model definition begins on lines 1 and 2. Again, we specify each individual observation i of N total observations. Lines 4 through 6 of Table 5 are a generalization from a single dependent variable to P dependent variables. Both the sample size N as well as the number of dependent variables P will be defined in the data file. In our case of the TAM model, we have $P = 12$ observed variables, representing the 12 questionnaire items in the original TAM instrument. Similar to our earlier regression, we specify a normal likelihood for the observed variables with mean `mu[i,j]` and precision (inverse variance, see footnote 2) `errorprec[j]`. The error variance is the same for all observations, i.e. a homogenous sample/ a homoskedasticity assumption. We will specify the hyper-parameters in the next subsection.

Lines 7 – 18 define the mean of the variables in terms of the loading λ_i (`lam[1] – lam[12]`) and the latent variable that the item loads on, either ξ_1 or ξ_2 (`xi[i,1]` or `xi[i,2]`). These definitions cannot be moved into the "for" loop in line 4, because different items load on different latent variables. Finally, line 20 defines the likelihood for the two latent variables in terms of a multivariante normal distribution with means `u` and precision (inverse variance) `latprec`. Note that `u` is a vector of two quantities, whereas `latprec` is a 2x2 matrix of four quantities. We wlil define `u` as fixed, reflecting common practice to assume zero-centered variables, and will specify a prior distribution for the variance and covariance of the latent variables, reflecting common practice to estimate them.

An easy extension to this model is the inclusion of intercepts. In that case, the specification of e.g. line 7 would need to change to `mu[I, 1]<-lam[1]*xi[i,1] + alpha[1]` where `alpha[1]` represents the intercept. This requires the later specification of a prior distribution for the intercept and it might then also be appropriate to estimate the means of the latent variables, rather than fixing them to zero.

Another easy extension is the inclusion of cross-loadings. In that case, the specification of line 7 would need to change to `mu[i,j]<-lam[j,1]*xi[i,1]+lam[j,2]*xi[i,2]`. In this case, it is possible to include these definitions in the "for" loop of line 4.

While we did not have sufficient data on the TAM outcome variables and estimates for the structural coefficients of the TAM model, the above BUGS model is easily extended to a full SEM model. For example, a full structural model of TAM can be expressed using the following specification:

```
xi[i]~dnorm(mu[i],prec.xi)
nu[1,i]<-beta[1]*xi[i]
```

```
  nu[2,i]<-beta[2]*xi[i]+gamma*eta[1,i]
  eta[1,i]~dnorm(nu[1, i], prec.eta1)
  eta[2,i]~dnorm(nu[2, i], prec.eta2)
```

In this model, `xi` (ξ) represents the exogenous TAM latent variable PEoU, eta[1,] (η_1) represents the endogenous TAM variable PU and `eta[2,]` (η_2) represents the BI (behavioral intention construct). The indicator specifications are similar those in Table 5.

Given the explicit nature of the model specification, it is also easy to see how identity constraints can be imposed on the model. For example, to suggest that loadings on the first and second indicator are the same, one would only need to change line 8 to read `mu[i,2]<-lam[1]*xi[i,1]`.

Finally, we note that, with the estimation of all loadings, latent variances and covariances, and error terms, the model is strictly not identified. However, as we see later, it is possible to estimate this model when the prior distributions sufficiently constrain the posterior parameter estimates. In fact, Muthen and Asparouhov [2012] recommend a model in which all cross-loadings are estimated but with small prior probabilities as more realistic and appropriate, given that in practice, cross-loadings are hardly ever exactly zero and the zero-constraint in covariance-based estimation leads to ill-fitting models that are still of practical interest. Moreover, Asparouhov and Muthen [2010a] have shown that the parameterization in which both latent variances and all loadings are estimated, as in the model in Table 5, provides considerable advantages in parameter accuracy, especially for small sample sizes and a large number of indicators.

Recommendation: For structural equation models

- Specify a model to estimate latent variances as well as all loadings.
- Estimate all cross-loadings with realistic small prior probabilities with sufficient precision (inverse variance) to ensure the model can be estimated.

Step 2: Identify prior knowledge and distributional assumptions

To identify previous estimates for the TAM model parameters, we focus on studies published in five IS journals, MISQ, JMIS, ISR, JAIS, and ISJ. Through the ISI web of science we identified papers in these journals that cite either Davis [1989] or Davis et al. [1989], revealing 263 papers. Of these, 43 are empirical papers that use at least some of the TAM indicators developed by Davis [1989] and Davis et al. [1989]. Figure 1 shows box-and-whisker plots of reported standardized loadings by items. That data is presented in Table 6 (all surveyed studies use 7-point scales). As many of the 43 studies do not use BI as outcome variable, we have compiled prior values only for the loadings of the PEoU and PU constructs. A more sophisticated meta-analysis may also use weighting by sample size when calculating the mean and variance; however, our focus is on the use of this data in Bayesian estimation. It is clear from the variance of the estimates reported in Table 6, as well as the diagram in Figure 1, that many parameter estimates reported in the literature are statistically significantly different from others, and from the mean. These outliers occur despite a certain “publication bias” from recommendations that parameter loadings should be greater than 0.7. Hence, significant differences in estimated loadings are quite likely to occur.

A researcher using PU and PEoU in a new study, and choosing to adopt the instrument pioneered by Davis [1989] and Davis et al. [1989], might be faced with the situation that, despite taking all reasonable precautions, her data does in fact show statistically significant differences to previously established values in Table 6. When the researcher is certain that her instrument does in fact measure the same construct (e.g. changes to the instrument have been ruled out, sample characteristics are comparable), the researcher may choose to use Bayesian statistics to estimate her model and thus interpret the knowledge from the newly collected sample in light of the prior knowledge about the parameter values.

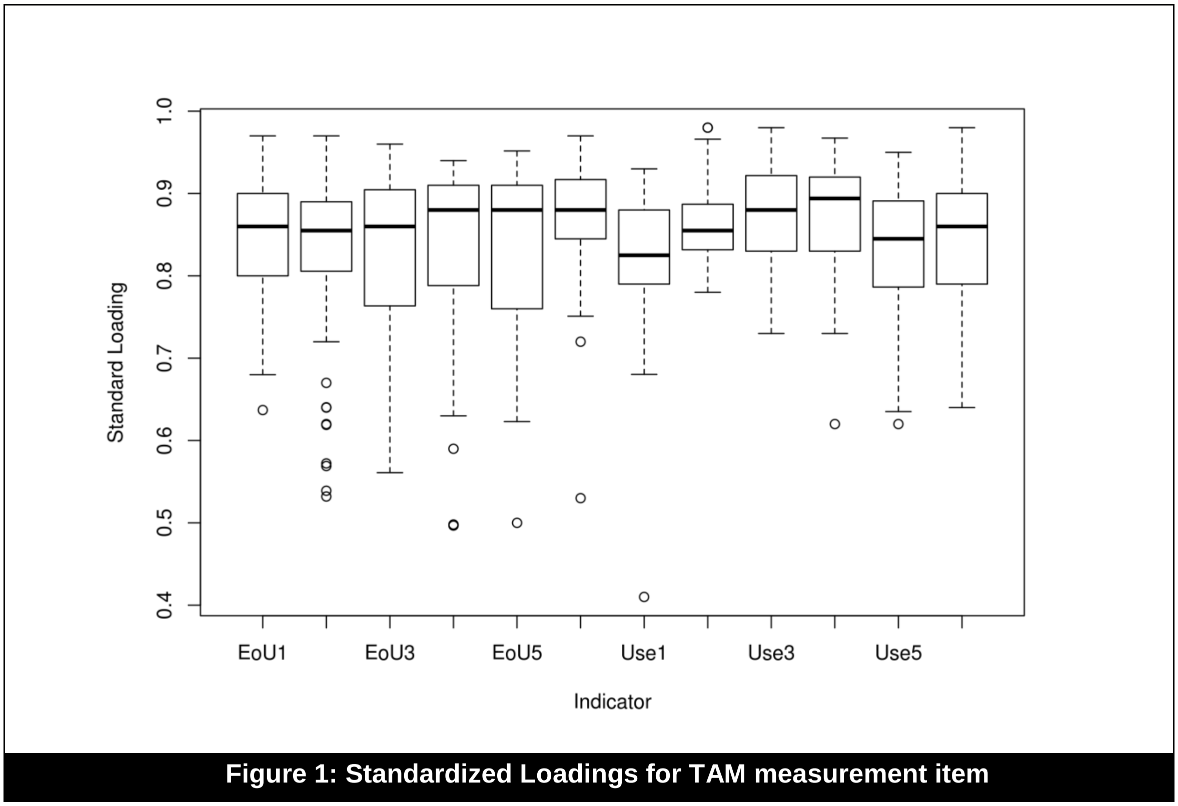


Figure 1: Standardized Loadings for TAM measurement item

Table 6: Standardized Loadings by TAM measurement item (S.E.M = Standard Error of Mean)

Item	Minimum	Median	Mean	Maximum	Variance	S.E.M.
PEoU1	.6370	.8600	.8432	.9700	.00586	.0095
PEoU2	.5320	.8550	.8202	.9700	.01261	.0154
PEoU3	.5610	.8600	.8327	.9600	.01028	.0135
PEoU4	.4967	.8800	.8217	.9400	.01526	.0211
PEoU5	.5000	.8800	.8344	.9517	.01260	.0158
PEoU6	.5300	.8800	.8682	.9700	.00562	.0092
PU1	.4100	.8250	.8199	.9300	.00743	.0127
PU2	.7800	.8550	.8652	.9800	.00298	.0105
PU3	.7300	.8800	.8724	.9800	.00421	.0087
PU4	.6200	.8940	.8728	.9673	.00451	.0124
PU5	.6200	.8450	.8309	.9500	.00711	.0124
PU6	.6400	.8600	.8429	.9800	.00622	.0100

With this prior knowledge, we can now continue the model specification in Table 7. Lines 24-35 specify normal prior distributions for the 12 item loadings. The prior mean is set to that calculated in Table 6. In OpenBUGS, the normal distribution is specified with the precision (inverse variance), rather than the variance, and we have used the inverse of the standard error of the mean as the precision for our prior belief. Bayesian estimation allows the researcher to “weight” the evidence provided by prior information. A higher precision gives relatively more weight to prior information, whereas a lower precision gives relatively more weight to the present data. Lines 36 and 37 specify the prior distribution of the precision estimate for the indicators, i.e. the inverse error variance. As for the simple regression example, because OpenBUGS parameterizes the normal distribution in terms of precision instead of variance, we specify a gamma prior distribution. Thus, the specification `dgamma(9.0, 4.0)` on line 37 yields a prior with mean of 2.25 and variance of 9/16 for the error precision (see Table 1), but a prior with mean of 0.5 and variance of 0.0357 for the error variance (see Table 1). We explicitly model the error variances (inverse precision) on line 38 as we require samples of this error variance for later analysis. Lines 40 to 47 specify the prior distribution for the variances and covariances of the two latent variables. Because of our assumption that these were normally distributed, we use the inverse Wishart distribution as conjugate prior of the multivariate normal distribution (Table 1). However, just as OpenBUGS parameterizes the normal distribution in terms of mean and inverse variance, the multivariate normal distribution is also parameterized as mean and inverse variance. Hence, instead of specifying a prior inverse Wishart distribution, we specify a prior Wishart distribution. (`dwish(…)` on line 41). The relationship between the two is simple, and analogous to Equation 8:

$$x \sim Wishart(\Psi, d) \quad \rightarrow \quad \frac{1}{x} \sim InverseWishart(\Psi^{-1}, d) \qquad \text{(Equation 9)}$$

However, to make matters confusing, the Wishart distribution in OpenBUGS is parameterized with the inverse of the Ψ matrix. In effect, this means that the matrix V supplied as parameter to `dwish(…)` on line 41 serves as our prior point belief about the variances and covariances of the latent variables. This matrix `V` is defined in lines 44 to 47. We have again modeled the latent covariance matrix explicitly as the inverse of the precision matrix on line 42, and, to make the subsequent model analysis easier still, we estimate the latent correlation directly in OpenBUGS (line 43).

Table 7: CFA model definition in OpenBUGS (part 2, specification of prior probabilities)

Line	Model definition
22	`#priors on loadings`
23	`lam[1]~dnorm(0.8432,105)`
24	`lam[2]~dnorm(0.8202,64)`
25	`lam[3]~dnorm(0.8327,74)`
26	`lam[4]~dnorm(0.8217,47)`
27	`lam[5]~dnorm(0.8344,63)`
28	`lam[6]~dnorm(0.8682,108)`
29	`lam[7]~dnorm(0.8199,78)`
30	`lam[8]~dnorm(0.8652,95)`
31	`lam[9]~dnorm(0.8724,114)`
32	`lam[10]~dnorm(0.8728,80)`
33	`lam[11]~dnorm(0.8309,80)`
34	`lam[12]~dnorm(0.8429,100)`
35	`#priors on errors`
36	`for(j in 1:P){`
37	`  errorprec[j]~dgamma(9.0, 4.0)`
38	`  errorvar[j]<-1/errorprec[j]`
39	`}`
40	`#priors on latent (co-)variances`
41	`latprec[1:2,1:2] ~ dwish(V[,], 5)`
42	`latcov[1:2,1:2] <- inverse(latprec[,])`
43	`latcor <- latcov[1,2]/(sqrt(latcov[1,1])*sqrt(latcov[2,2]))`
44	`V[1,1] <- 1`
45	`V[1,2] <- 0.5`
46	`V[2,1] <- V[1,2]`
47	`V[2,2] <- 1`
48	`} #end of model`

Recommendation: To specify prior probability distributions,

- Research the literature for previous estimates of model parameters
- Use the appropriate conjugate prior distribution for the type of assumed likelihood
- Use informative prior distributions when sufficient knowledge exists
- Use uninformative prior distributions when now previous knowledge exists. Such prior distributions should be “skeptical” in the sense that they reflect a null hypothesis of “no effect”, e.g. have a mean of zero for regression parameters.

Step 3: Estimate the Model

Once the statistical model with all prior probability distributions is specified, the model can be estimated using OpenBUGS or WinBUGS. This can be done interactively, but can also be scripted. For easy repeatability of the analysis, scripts are preferred. Table 8 shows the script to use for estimating our TAM model. It is similar in structure to the one used for the simple example earlier (Table 3).

Table 8: OpenBUGS script to control the estimation

Line	OpenBUGS script
1	`modelSetWD('/home/joerg/OpenBUGSExample')`
2	`modelCheck('model.txt')`
3	`modelData('data1.txt')`
4	`modelCompile(3)`
5	`modelGenInits()`
6	`samplesSet('lam')`
7	`samplesSet('latcov')`
8	`samplesSet('latcor')`
9	`samplesSet('errvar')`
10	`dicSet()`
11	`modelUpdate(5000, 1, 1, 'F')`
12	`samplesCoda('*', 'coda_output')`
13	`samplesStats('*')`
14	`dicStats()`

Lines 1-3 set the working directory, load and syntactically check the model definition, and load the data. The data file also needs to contain definitions for all fixed parameters that are not defined in the model file itself. For example, line 20 in Table 5 references a vector of value u that is not assigned a probability distribution or fixed in the model definition. Thus, OpenBUGS expects to find fixed values for u in the data file. Line 4 in Table 8 instructs OpenBUGS to set up the model with three MCMC sampling chains. Initial values are generated in line 5 for all three MCM chains. Lines 6 through 9 instruct OpenBUGS to keep samples of important model variables.

Note that we can sample any variables that we define in the model and for which no fixed values or data are provided. For example, the variable `latcor` was computed in line 43 of Table 7, and we can similarly compute other quantities of interest for sampling. More interestingly, if some data was missing completely at random (i.e. there is no missingness mechanism to be modelled), one can sample those values, e.g. by specifying `samplesSet(‘y[198,7]’)` to sample the value of the seventh indicator for case 198.

Line 10 sets up the computation of the DIC (deviance information criterion) for diagnostic purposes later. Line 11 then instructs OpenBUGS to update the model parameters with 5000 MCMC sampling iterations. Once this is completed, line 12 will save all sampled values to a set of files whose names begin with “coda_output”. Lines 13 and 14 instruct OpenBUGS to display on screen the sample statistics and the DIC statistics. For our example, this script took 102 seconds and produced the following output[7]:

```
OpenBUGS version 3.2.1 rev 781
type 'modelQuit()' to quit
OpenBUGS> OpenBUGS> model is syntactically correct
OpenBUGS> data loaded
OpenBUGS> model compiled
OpenBUGS> initial values generated, model initialized
OpenBUGS> monitor set
```

[7] Because this is a stochastic process, the exact values will differ a little from repetition to repetition.

```
OpenBUGS> monitor set
OpenBUGS> monitor set
OpenBUGS> monitor set
OpenBUGS> deviance set
OpenBUGS> 5000 updates took 50 s
OpenBUGS> CODA files written
OpenBUGS>
                         mean      sd        MC_error  val2.5pc  median    val97.5pc start     sample
 errvar[1]               0.6718    0.06247   6.69E-4   0.5572    0.6684    0.8023    1         15000
 errvar[2]               0.4839    0.04695   5.031E-4  0.3988    0.4816    0.5821    1         15000
 errvar[3]               0.3647    0.03699   4.285E-4  0.2977    0.3628    0.4428    1         15000
 errvar[4]               0.6458    0.05703   5.458E-4  0.5424    0.6431    0.7664    1         15000
 errvar[5]               0.501     0.04875   5.05E-4   0.4129    0.4982    0.6048    1         15000
 errvar[6]               0.3263    0.03591   4.991E-4  0.2623    0.3239    0.4031    1         15000
 errvar[7]               0.2985    0.02839   3.086E-4  0.2465    0.2969    0.3588    1         15000
 errvar[8]               0.3207    0.03024   3.046E-4  0.2667    0.3191    0.3846    1         15000
 errvar[9]               0.3637    0.03538   3.878E-4  0.3       0.3618    0.4386    1         15000
 errvar[10]              0.3174    0.0315    3.508E-4  0.2606    0.3157    0.3831    1         15000
 errvar[11]              0.5137    0.04801   5.114E-4  0.4265    0.5112    0.6144    1         15000
 errvar[12]              0.3376    0.03481   4.281E-4  0.2744    0.3357    0.4111    1         15000
 lam[1]                  0.9117    0.05841   0.00252   0.8024    0.9107    1.023     1         15000
 lam[2]                  0.8547    0.05657   0.002495  0.7491    0.8536    0.964     1         15000
 lam[3]                  0.8434    0.05501   0.002467  0.7409    0.8418    0.9504    1         15000
 lam[4]                  0.7158    0.05147   0.002116  0.6192    0.7149    0.8141    1         15000
 lam[5]                  0.8485    0.05629   0.002461  0.7439    0.8468    0.957     1         15000
 lam[6]                  0.9039    0.05673   0.002603  0.7967    0.9028    1.014     1         15000
 lam[7]                  0.7442    0.04686   0.00141   0.6544    0.7433    0.8375    1         15000
 lam[8]                  0.8007    0.04883   0.001468  0.7079    0.7999    0.8988    1         15000
 lam[9]                  0.878     0.05142   0.001544  0.7788    0.8772    0.98      1         15000
 lam[10]                 0.8712    0.05135   0.001594  0.7718    0.8706    0.9722    1         15000
 lam[11]                 0.9112    0.05623   0.001605  0.8015    0.9107    1.021     1         15000
 lam[12]                 0.9709    0.05483   0.001699  0.865     0.9702    1.078     1         15000
 latcor                  0.613     0.04007   4.101E-4  0.5294    0.6145    0.6867    1         15000
 latcov[1,1]             2.811     0.4158    0.01707   2.104     2.777     3.735     1         15000
 latcov[1,2]             1.098     0.1547    0.004086  0.8181    1.087     1.429     1         15000
 latcov[2,1]             1.098     0.1547    0.004086  0.8181    1.087     1.429     1         15000
 latcov[2,2]             1.146     0.1529    0.004557  0.8832    1.134     1.475     1         15000
OpenBUGS>  Dbar Dhat DIC pD
y 6584.0 6038.0 7130.0 545.7
total 6584.0 6038.0 7130.0 545.7
```

The output shows the mean, standard deviation, confidence intervals, and sample sizes for the sampled values. However, these values should not be relied upon or reported until the diagnostics in the next two steps are attended to.

There are a number of reasons for performing this many iterations. First, each MCMC sampling chain typically requires a few hundred samples to converge to the proper posterior distribution. Hence, early samples must be discarded from the subsequent analysis of the estimation results. Second, because of the autocorrelation among samples, only every k-th sample should be considered to be independent and used for further analysis, i.e. there will be a degree of “thinning” of the samples. The number of remaining samples should be sufficient to provide a stable estimate of the posterior probability distributions. Gelman et al. [2004] recommend between 100 and 2000 samples be used for inferences, depending on model complexity and desired accuracy.

Recommendation: To estimate the model,

- Use at least 3 MCMC chains
- Use at least 5000 sampling iterations (possibly fewer for less complex models)

Optional: For repeatability of results,

- Set the random number generator seed in OpenBUGS (using `modelSetRN(…)`)
- Specify initial values, rather than generating them (using `modelInits(…)`)

Step 4: Assess MCMC Convergence

As any numerical, iterative algorithm, Bayesian estimation can suffer from convergence problems. Before interpreting the results of Bayesian estimation, it is therefore important to perform diagnostic evaluations. Two distinct checks are important. First, we need to check whether each sampling chain has converged. Second, we need to check whether the sampling chain has converged to the right value. Thus, the first issue is to assess intra-chain convergence, whereas the second can be assessed by examining inter-chain convergence.

To aid in this analysis we use the CODA package in the R statistical system [Plummer et al., 2006]. As part of our OpenBUGS estimation, we saved our MCMC samples to a set of files in CODA format (line12, Table 8). We read these files and analyze them using the R script shown in Table 9. Line 1 in Table 9 loads the “coda” package into the R workspace. Line 2 reads the coda format output that OpenBUGS has produced in the previous step (estimation).

Table 9: R Script for data analysis (Part 1: convergence diagnostics)

Line	R script
1	`library(coda)`
2	`mcmc.list <- read.openbugs('coda_output')`
3	`plot(mcmc.list)`
4	`geweke.diag(mcmc.list)`
5	`   Geweke.plot(mcmc.list)`
6	`heidel.diag(mcmc.list)`
7	`gelman.diag(mcmc.list)`
8	`gelman.plot(mcmc.list)`

Line 3 plots a sampling trace and a sampling density for every parameter that was sampled and is present in the coda file. These plots are useful for assessing both inter- and intra-chain convergence.

Figure 2 shows a properly converged solution for one parameter of the model. The trace plot on the left of the figure shows the sampled values for each of the three chains for the 5000 samples, while the density plot on the right shows the overall frequency of sampled values for the three chains. We can see that all three sampling chains converge on the same values and each of the three sampling chains has a stable average. The density plot in Figure 2 confirms this by showing an approximately normal distribution.

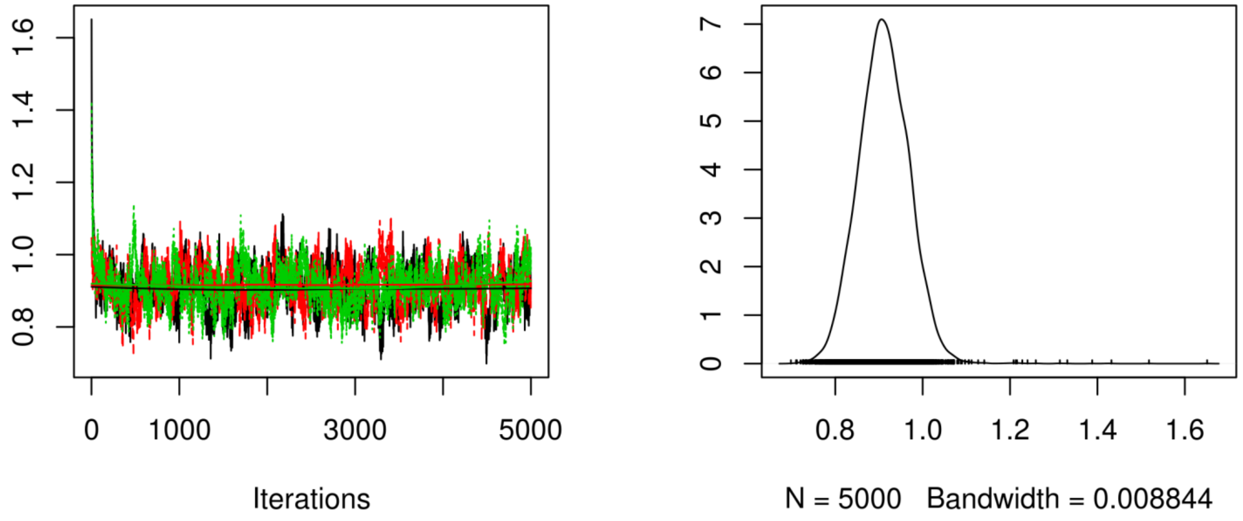


Figure 2: Trace plot and density plot for one parameter of the Bayesian CFA model showing a good solution

In contrast, Figure 3 shows a trace plot and a density plot for one parameter of the CFA model that shows convergence problems of the type that one of the chains produces stable values that differ from those of the other chains. We can see that one of the chains converged on a different value, which is also reflected in the bimodal density plot on the right of Figure 3. In this situation, the estimation should be re-run with different starting values for this parameter.

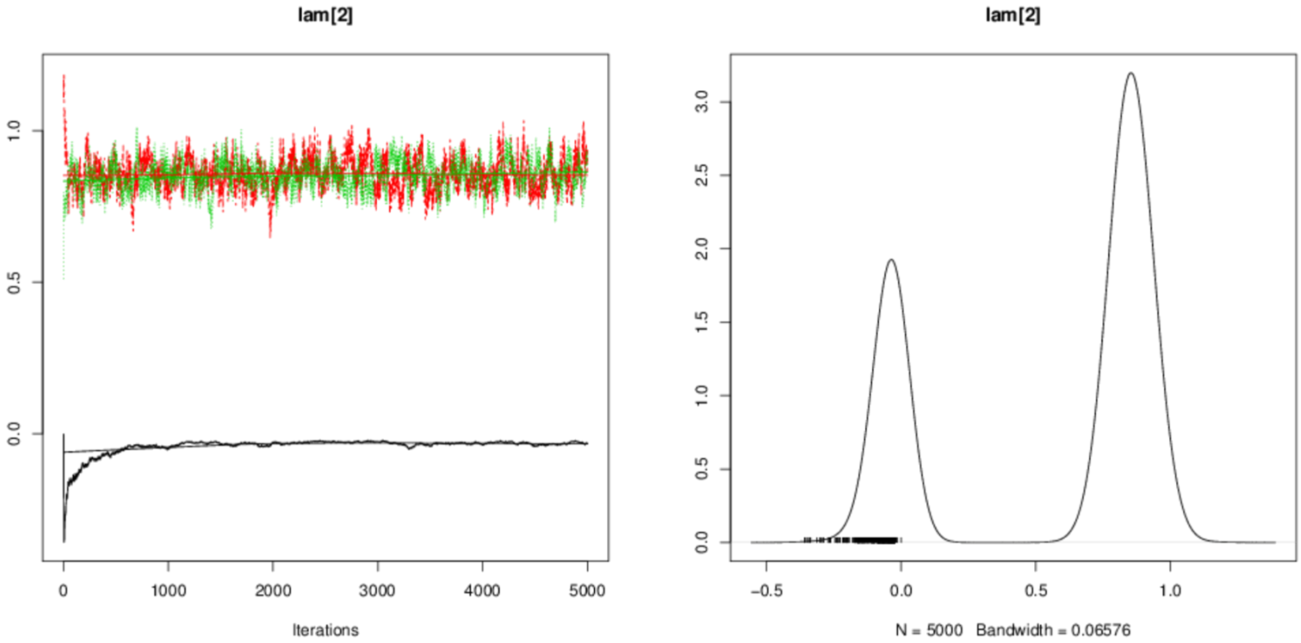


Figure 3: Trace plot and density plot for one parameter of the Bayesian CFA model showing non-convergence

The second issue is the convergence of each individual chain around a stable mean. Figure 4 below shows a trace plot and density plot for a situation where the individual chains did not converge. We can clearly see that the sampled values fluctuate wildly around their sliding-window average (solid lines in the trace plot).

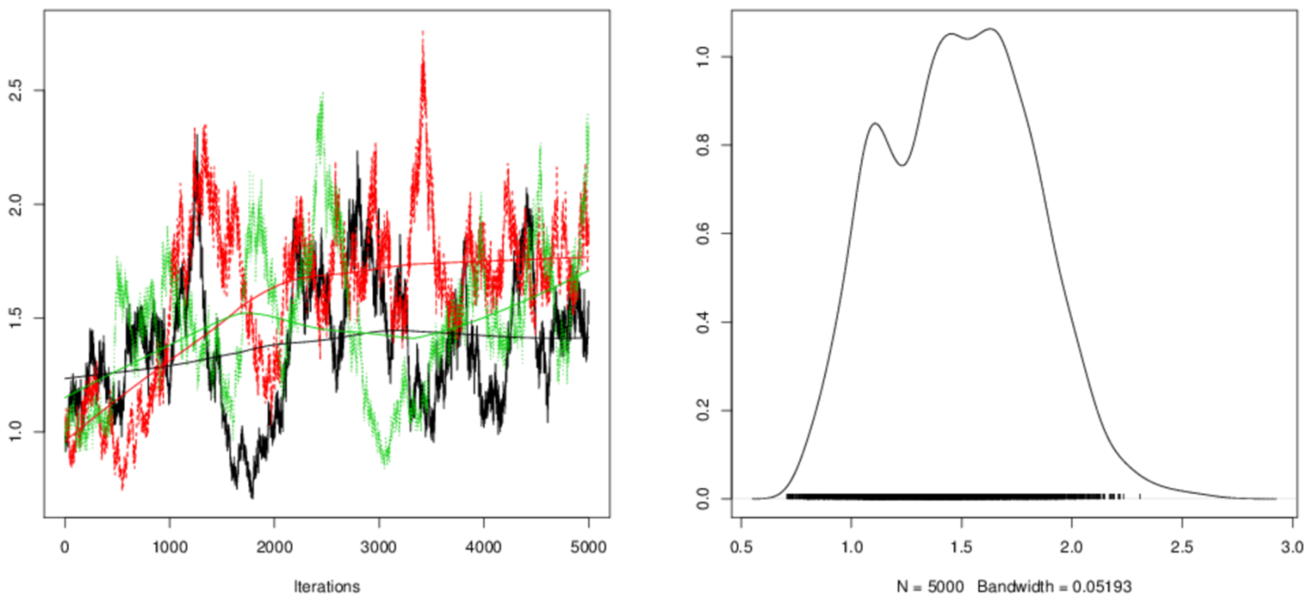


Figure 4: Trace plot and density plot for one parameter of the Bayesian CFA model showing non-convergence of the individual sampling chains.

Recommendation: Use trace and density plots for all parameters.

- Density plots should reflect the expected posterior distribution (based on the choice of likelihood and prior)
- Sampling means of each chain become stable
- Sampling means of all chains converge

More formally, a number of statistics can be computed to help identify convergence problems. For example, Geweke [1992] suggested testing the equality of means of the first 10% and the last 50% of the values in the sampling chain to assess the stability of the estimates. The test statistic is normally distributed and can be used for a z-test. Line 4 in Table 9 performs these tests on all sampled parameters and line 5 produces diagnostic plots as shown in Figure 5. The following is an example output (abbreviated and shown only for a single chain) that shows the z-distributed test statistics for our data:

```
[[1]]
Fraction in 1st window = 0.1
Fraction in 2nd window = 0.5
  errvar[1]   errvar[2]   errvar[3]   errvar[4]   errvar[5]   errvar[6]
    -0.7398     -0.6226      0.4811     -1.1590      0.7641      0.0302
...
     latcor latcov[1,1] latcov[1,2] latcov[2,1] latcov[2,2]
    -0.5470     -1.5084     -2.2907     -2.2907     -1.5105
```

Line 4 in our analysis script (Table 9) produces a plot like the one shown in Figure 5 for all sampled parameters. The plot shows the test statistics and the 95% confidence interval (1.96 standard deviations). For this plot, the first half of the Markov chain is divided into 20 segments, then Geweke's z-score is repeatedly calculated. The first z-score is calculated with all iterations in the chain, the second after discarding the first segment, the third after discarding the first two segments, and so on. The last z-score is calculated using only the samples in the second half of the chain. This diagnostic tool can show which part of the chain is different from the final part.

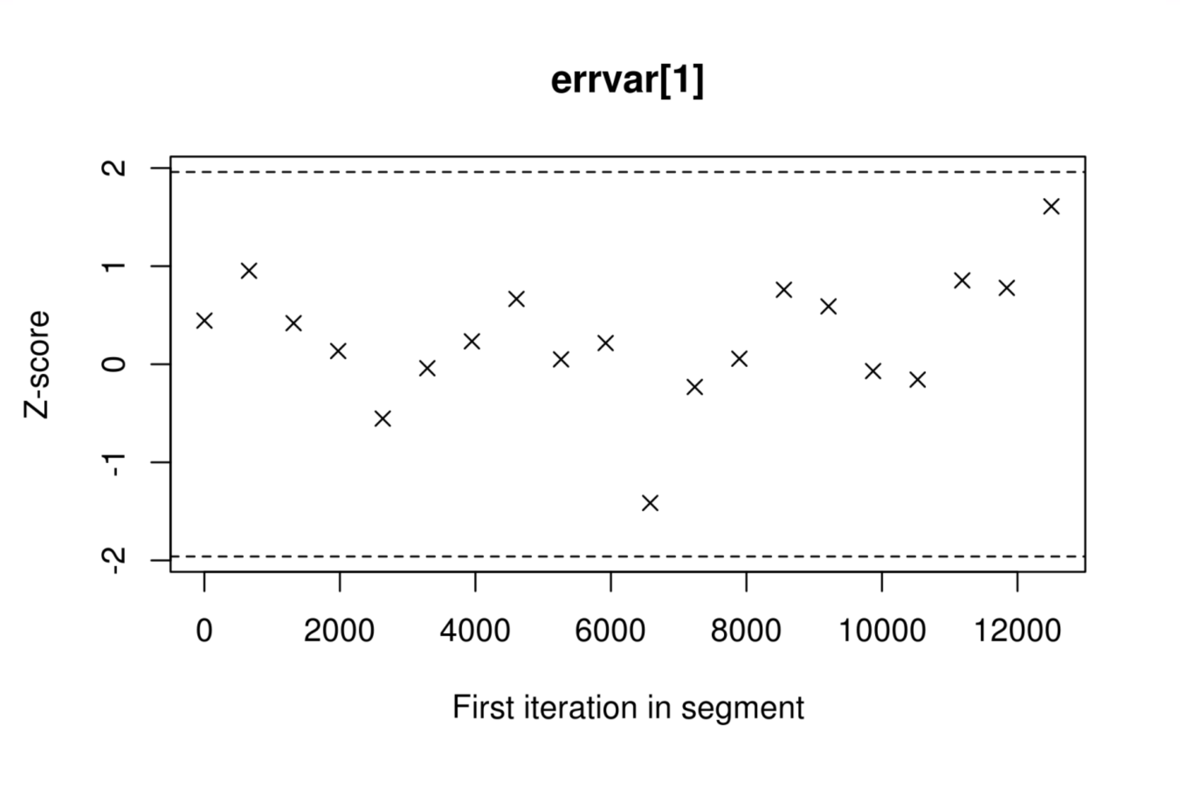


Figure 5: Plot of Geweke test statistics for a single parameter of a single Gibbs sampling chain

Another set of tests has been proposed by Heidelberger and Welch [1983]. The first uses the Cramer-von-Mises test to assess whether the sampled values come from a stationary distribution. As with Geweke's test, this test is also successively applied, first to the entire chain, then after discarding the first 10%, 20%, etc. of the chain. Line 6 in our analysis script (Table 9) performs these tests for all sampled parameters. The following is an example output (abbreviated and shown only for a single chain):

```
[[1]]
             Stationarity start     p-value
             test         iteration
errvar[1]    passed       1         0.659
errvar[2]    passed       1         0.670
...
latcov[2,1] passed        1         0.501
latcov[2,2] passed        1         0.541
             Halfwidth Mean  Halfwidth
             test
errvar[1]    passed    0.671 0.00202
errvar[2]    passed    0.483 0.00181
...
latcov[2,1] passed     1.107 0.01676
latcov[2,2] passed     1.148 0.01928
```

The reported start iteration is that iteration at the inclusion of which the stationarity test was passed. In our example, the stationarity test was passed even when the entire sample was used (start iteration equals one). The second test then takes the sampled values that are accepted by the stationarity test (in our case, the entire sample) and constructs a 95% confidence interval for the sampled value. It then compares the half-width of this interval to them mean and reports the difference between the two. The half-width of the confidence interval should coincide with the sample mean. In our example, both the stationarity and the half-width test are passed for all sampled parameters in all three chains.

In contrast to Geweke's and Heidelberger and Welch's tests, which examined the intra-chain convergence, Gelman's potential scale reduction factor (PSRF) assesses the inter-chain convergence [Gelman et al., 2004]. The PSRF is analogous to an ANOVA in that it compares the between- and within-sequence variances of parameter estimates. In the following expression for the PSRF, B and W refer to the between-chain and within-chain sampling variance, respectively, where n is the number of samples in each chain, and m is the number of chains:

$$B = \frac{n}{m-1}\sum_{j=1}^{m}\left(\bar{\psi}_{.j} - \bar{\psi}_{..}\right)^2 \qquad \text{(Equation 10)}$$

$$W = \frac{1}{m}\sum_{j=1}^{m}\left[\frac{1}{n-1}\sum_{i=1}^{n}\left(\bar{\psi}_{ij} - \bar{\psi}_{.j}\right)^2\right] \qquad \text{(Equation 11)}$$

$$PSRF = \sqrt{\frac{\frac{n-1}{n}W + \frac{1}{n}B}{W}} \qquad \text{(Equation 12)}$$

Line 7 in our analysis script (Table 9) calculates the PSRF for all sampled parameters, and Line 8 in our script produces plots of PSRF for all parameters, similar to the one shown in Figure 6 for a single parameter. The following is an example output of Gelman's diagnostic (abbreviated):

```
Potential scale reduction factors:
            Point est. 97.5% quantile
errvar[1]         1.00           1.00
errvar[2]         1.00           1.00
...
latcov[2,1]       1.00           1.00
latcov[2,2]       1.00           1.01

Multivariate psrf
1.02
```

The recommendation is for the PSRF to be less than 1.1 [Gelman et al., 2004] to indicate good convergence, which is true for all parameters in our example.

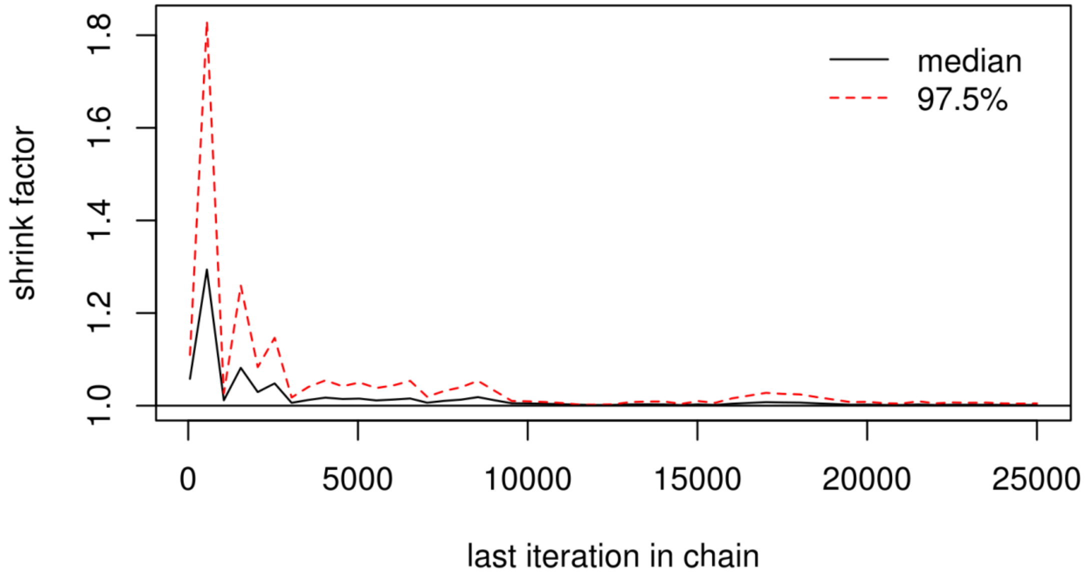


Figure 6: Plot of Gelman's PSR factor for one sampled parameter

Recommendation: To ensure intra- and inter-chain convergence,

- Geweke's z-statistics should be less than 1.96 for all parameters
- Heidelberger and Welch's stationarity tests should be passed (note the starting iteration) by all parameters
- Heidelberger and Welch's half-width test should be passed by all parameters
- Gelman's PSRF should be less than 1.1 for all parameters

Recommendation: If the diagnostics fail to support intra- or inter-chain convergence, go back to step 3 (Estimate the model) and

- Increase the number of MCMC iterations, and/or
- Manually set different starting values
- Repeat step 4 (Assess MCMC Convergence)

Step 5: Remove Burn-In Iterations and Thin Samples

Because of the iterative nature of MCMC sampling, the initial samples in each chain should be discarded prior to analysis. In many cases, researchers may discard the initial 10% or 20% or even half of the chain. This initial part of the chain where the sampled estimates are still converging is called the "burn-in" period. The results of the diagnostics in the previous subsections given an indication how many such "burn-in" samples should be discarded.

For example, both Geweke's as well as Heidelberger and Welch's diagnostics suggest that the entire chain might be usable. However, a look at Gelman's diagnostic (Figure 6) shows that convergence might not have been achieved for the first 500 iterations.

Table 10: R Script for data analysis (Part 2: Assessing auto-correlation, removing burn-in iterations and thinning the MCMC samples)

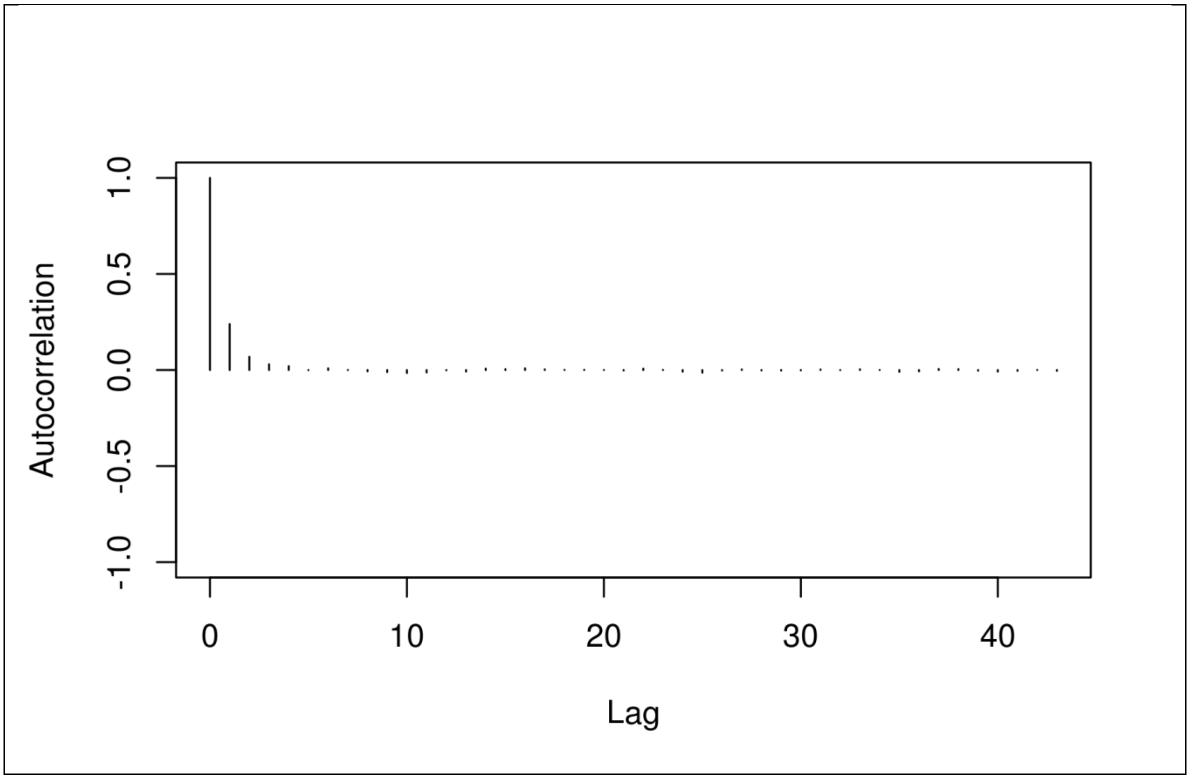
Line	R script
8	`autocorr.diag(mcmc.list)`
9	`autorcorr.plot(mcmc.list)`
10	`raftery.diag(mcmc.list, q=0.5, r=0.05)`
11	`thinned <- window(mcmc.list, 2000, 25000, 50)`

Because of the nature of the MCMC sampler (see earlier illustration), consecutive samples are not independent, and are likely correlated (“autocorrelation”). As we noted above, any inference on the parameter estimates assumes independence of observations. This can be achieved approximately by selecting only every k-th sample for analysis, where k is chosen to reduce the effect of autocorrelation (“thinning” the MCMC series). On the other hand, autocorrelation is in increasingly smaller problem the larger the sample becomes.

Line 8 in our analysis script (Table 10) computes the autocorrelation among sampled values for each sampled parameter at different lag distances. The following is an example output (abbreviated) for our data. It shows that for these parameters, a distance or lag of 5 is sufficient to significantly reduce autocorrelation. Line 9 in in our script produces a plot of these autocorrelation values, similar to the one shown in Figure 7.

```
         errvar[1]    errvar[2]    errvar[3]    errvar[4]   errvar[5]
Lag 0   1.00000000  1.000000000  1.000000000  1.000000000  1.00000000
Lag 1   0.23631085  0.280117179  0.351692755  0.158127378  0.28515151
Lag 5  -0.01243173  0.004290029  0.011596124  0.003318055 -0.01460019
Lag 10 -0.01634735 -0.010678489  0.006672367  0.004160701 -0.01899383
Lag 50  0.02435305  0.002701307 -0.002208589 -0.001444679  0.01539449
          lam[5]    lam[6]     lam[7]     lam[8]    lam[9]    lam[10]
Lag 0  1.0000000 1.0000000 1.00000000 1.00000000 1.0000000 1.00000000
Lag 1  0.8018296 0.8697411 0.60943472 0.61126568 0.6170946 0.65687820
Lag 5  0.6715677 0.7354921 0.41965701 0.41188918 0.4237928 0.46294914
Lag 10 0.5485456 0.6025749 0.28160973 0.26951662 0.2699685 0.29114718
Lag 50 0.1101499 0.1313006 0.01960327 0.03040368 0.0295588 0.02315531
```

From Figure 7 (top) we see that the autocorrelation with a lag of 3 is already quite small for one parameter. Retaining every 3rd sample of the remaining 4500 would leave us with 1500 retained samples (from each chain), well within the recommendation by Gelman et al. [2004], who recommend that between 100 and 2000 samples be used for inferences, depending on model complexity and desired accuracy. However, our data and Figure 7 (bottom) also indicate that the samples for other model parameters show significantly higher auto-correlation. In our second, extreme case, a lag of 50 is required to reduce autocorrelation to acceptable levels, retaining only 90 samples for that parameter.



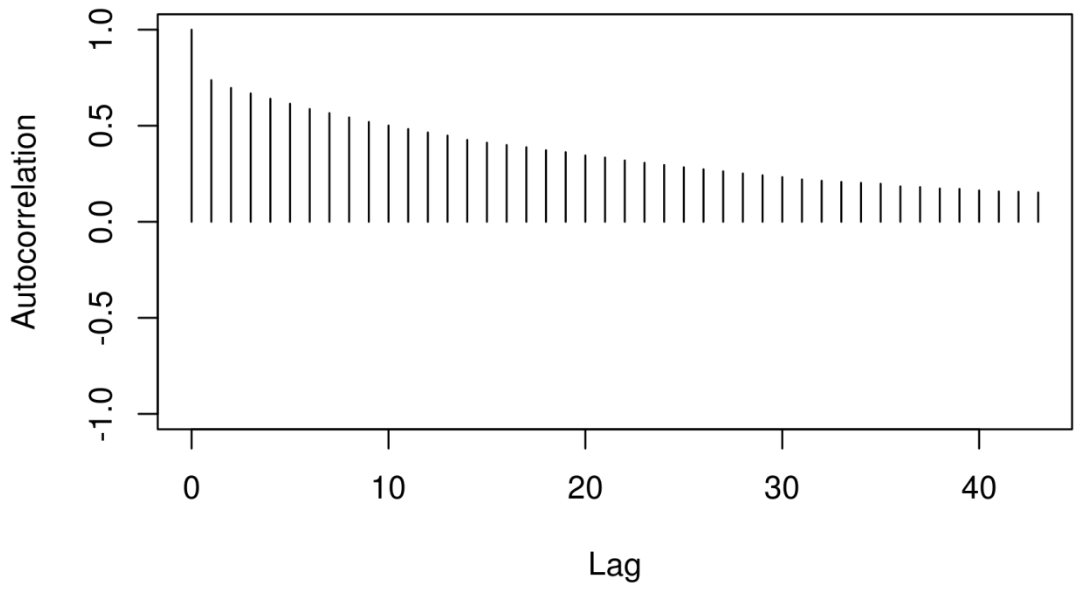


Figure 7: Autocorrelation plot for two sampled parameters

A more precise recommendation for the required sample size, given desired margin of errors for the parameter inference and the degree of autocorrelation in the samples, is provided by Raftery and Lewis [1992; 1995]. Line 10 in our analysis script (Table 10, continued from Table 9) computes this information. The parameter q describes the quantile to be estimated (here, the mean) and the parameter r describes the required accuracy or precision for the estimate. The following is an example output (abbreviated, and for a single chain only):

```
[[1]]
Quantile (q) = 0.5
Accuracy (r) = +/- 0.05
Probability (s) = 0.95

             Burn-in  Total Lower bound  Dependence
             (M)      (N)   (Nmin)       factor (I)
 errvar[1]   4        536   385           1.39
 errvar[2]   4        550   385           1.43
...
 lam[1]      63       8154  385          21.20
 lam[2]      72       8808  385          22.90
...
 latcov[2,1] 16       2168  385           5.63
 latcov[2,2] 30       3955  385          10.30
```

The first column (Burn-in M) shows the recommended number of burn-in iterations to discard, the second column (N) shows the recommended number of samples to retain. This estimate is based on an estimate of the autocorrelation in the last column (Dependence Factor I). The higher the autocorrelation (dependence factor), the larger the number of required samples. The third column (lower bound Nmin) gives the minimum required sample size when there is no autocorrelation. In our example, the high auto-correlation for some parameters leads to a very large required sample size.

As the output shows, it is possible to either "thin" the series sufficiently to reduce autocorrelation (keeping in mind the lower bound Nmin) or to estimate more samples but without thinning the series (keeping in mind the required Total N). Sometimes, both options must be chosen. While the decision how many "burn-in" samples to discard and how much to "thin" a MCMC series is within the discretion of the researcher, it is generally better to discard samples liberally. If the researcher feels that the remaining samples are insufficient for later analysis, the estimation should be repeated with more samples. While perhaps prohibitive even 10 years ago, given the increasing computing power available for Bayesian analysis, this kind of "trial-and-error" estimation is easily possible now.

For our example, we decided to re-estimate the model with 25,000 iterations, to discard (generously) the first 2,000 iterations and to thin (also generously) with a distance of 50, i.e. retain only every 50th sample. This leaves a total sample size of 460 for each of the three chains. The coda package provides a convenient function for selecting from, and thinning a set of MCMC samples, as shown in line 11 in Table 10.

Recommendation: After the model estimation,

- Based on the results from step 4 (Assess MCMC Convergence), discard (generously) the burn-in iterations from the sample.
- Assess autocorrelation within MCMC series
- Use Raftery and Lewis' method to identify required number of samples for desired accuracy

Then, either

- Thin the series to avoid autocorrelation, and/or
- Increase the sample size (MCMC iterations) and rerun the model estimation (step 3)

Step 6: Evaluate Model Quality

Bayesian structural equation modeling, unlike covariance based SEM analysis, does not offer a simple test of overall model fit like the χ2 test statistic [Evermann and Tate 2011]. Instead, the recommended way to assess the fit to data of a Bayesian model is to use the posterior-predictive p-value (PPP) [Asparouhov and Muthen, 2010a; Asparouhov and Muthen, 2010b; Gelman et al., 1996; Muthen and Asparouhov, 2012; Scheines et al., 1999]. The idea is to define a discrepancy function that represents the fit between data Y and model. It is common to use the traditional χ^2 fit function for this. The discrepancy function $f(Y,\theta)$ is calculated at each sampling iterations, based on the currently sampled values θ of the model parameters. At the same time, a new data set $\tilde{Y}$ is drawn from a multivariate-normal distribution that is based on the currently sampled values θ of the model parameters. Note that the new data set is sampled from the posterior distribution with sampling error. It is data that is predicted by the model and the posterior probability values. The discrepancy function $f(\tilde{Y},\theta)$ is then also evaluated using this new data set. The PPP value is defined as the probability that $f(Y,\theta) < f(\tilde{Y},\theta)$, i.e. that the original data fits the model better than the predicted data, formally:

$$PPP = p(f(Y,\theta) < f(\tilde{Y},\theta)$$

This probability is approximated as the proportion of sampling iterations for which $f(Y,\theta) < f(\tilde{Y},\theta)$. Low PPP values indicate that original sample data fits the model significantly worse than data that is predicted from the model. An excellent fit is characterized by a PPP of 0.5, i.e. the original data fits the model as well as data that is predicted from it. Muthen and Asparouhov [2012] suggest that a PPP of 0.05 is a reasonable indicator of acceptable fit. In simulation studies [Asparouhov and Muthen, 2010a], the PPP has been shown to perform with less bias than the classical χ^2 fit statistic for small sample sizes. However, at the same time, it was also less powerful to reject misspecified models for all sample sizes, although this effect diminishes as sample size increases.

Because the discrepancy function is specific to structural equation models, it is not available in OpenBUGS nor in any standard R package. Thus, we have implemented the χ^2 fit function for a CFA analysis in our analysis script (Table 11).

Line 1 in Table 11 loads the MASS package, which is required to draw a sample from a multivariate-normal distribution. Line 2 loads the original data set and lines 3 through 7 set up some variables needed for the later computation. In line 9, we move the thinned MCMC samples into an R data frame for easier access. Line 11 begins a loop over all MCMC samples so that we can assess the discrepancy functions. Based on the parameter estimates for that iteration (mean across all chains, line 13), we calculate the model matrices for the CFA model in lines 14 – 19. The model-implied covariance matrix is computed in line 21 for the CFA model. The expression for the general SEM can be found in any SEM textbook, e.g. [Bollen, 1989]. With this matrix in hand, we can compute the chi-square discrepancy function f in line 23 [Bollen, 1989]. Next, in line 25 we simulate data by drawing from a multivariate normal distribution with the model-implied covariance matrix and compute the covariance of the simulated (predicted) data (line 27). Again, we calculate the same chi-square discrepancy function, this time for the simulated/predicted data (line 29). Lines 30 to 35 provide some output and keep track of the differences in the discrepancy functions, as well as the PPP. Once the loop over all MCMC iterations is completed, line 38 computes the 95% confidence interval on the differences in discrepancy function, and line 40 outputs the PPP, approximated as the proportion of iterations for which the fit of the actual data to the model was better than the fit of the simulated/predicted data.

Table 11: R Script for data analysis (Part 3: Calculating the PPP)

Line	Script file
1	`library(MASS)`
2	`data <- read.csv('simulated.data.283.csv')`
3	`# set up some variables`
4	`n <- nrow(data) # Number of observations`
5	`S.f <- cov(data) # Covariance matrix of data`
6	`c <- 0 # Counter for PPP`
7	`diff <- matrix(0) # Store differences in fit`
8	`# Move thinned MCMC samples into data frame for easier access`
9	`d <- as.data.frame(as.matrix(thinned, iters=TRUE, chains=TRUE))`
10	`# Loop over all retained MCMC samples`
11	`for (l in seq(start(thinned), end(thinned), thin(thinned))) {`
12	`  # Calculate the mean parameter values over all chains`
13	`  m <- apply(d[d$ITER==l,], 2, mean)`
14	`  # Matrix ephat.mat is the error variance matrix`
15	`  ephat.mat <- diag(m[3:14], nrow=12, ncol=12)`
16	`  # Matrix phi.mat is the latent covariance matrix`
17	`  phi.mat <- matrix(m[28:31], nrow=2, ncol=2, byrow=TRUE)`
18	`  # Matrix lambda.mat is the loading matrix`
19	`  lambda.mat <- matrix(c(m[15:20], rep(0,12), m[21:26]), nrow=12, ncol=2, byrow=FALSE)`
20	`  # Calculate predicted covariance matrix`
21	`  Sigma.pred <- lambda.mat %*% phi.mat %*% t(lambda.mat) + ephat.mat`
22	`  # The chi-square discrepancy based on current model and actual covariance`
23	`  f <- (n - 1) * (log(det(Sigma.pred)) + sum(diag(solve(Sigma.pred) %*% S.f )) - log(det(S.f)) - 12)`
24	`  # Simulate data set from current model`
25	`  sim.data <- mvrnorm(n=n, mu=rep(0,12), Sigma=Sigma.pred, empirical=FALSE)`
26	`  # Calculate simulated data covariance matrix`
27	`  S.pred <- cov(sim.data)`
28	`  # The chi-square discrepancy based on current model and predicted data`
29	`  f.pred <- (nrow(sim.data) - 1) * (log(det(Sigma.pred)) + sum(diag(solve(Sigma.pred) %*% S.pred)) - log(det(S.pred)) - 12)`
30	`  # Some output to see what is going on`
31	`  cat('Iteration', l, ': f.pred = ', f.pred, ' // f = ', f, ' \n')`
32	`  # Keep track of the differences in discrepancy function values`
33	`  diff[ (l - start(thinned))/thin(thinned) + 1] <- abs(f.pred-f)`
34	`  # Keep track of whether model fits better to actual than to predicted data`
35	`  if (f < f.pred) c <- c + 1`
36	`}`
37	`# Report 95% confidence interval on fit differences`
38	`quantile(diff, c(0.05, 0.95))`
39	`# Report the PPP`
40	`c/length(diff)`

For our example, the 95% confidence interval of the difference in discrepancy values (across 461 MCMC iterations) was [35.134; 76,924] and the PPP was 0. While this indicates that the model does not fit, this results comes as no surprise as the original model, estimated using covariance analysis, also shows lack of fit [Evermann and Tate, 2011].

Another measure of model fit is the DIC (deviance information criterion) [Gelman et al., 2004; Spiegelhalter et al., 2002]. The deviance itself is defined in terms of the log-likelihood

$$D(x,\theta) = -2 \log p(x \mid \theta) \qquad \text{(Equation 13)}$$

Using the mean of the posterior distributions for each parameter $\hat{\theta}$, one can estimate an overall summary of the deviance as

$$D_{\hat{\theta}}(x) = D(x,\hat{\theta}) \qquad \text{(Equation 14)}$$

On the other hand, one can compute the deviance for each MCMC sample of the posterior θ_l, and then compute the mean of those:

$$\widehat{D(x)} = \frac{1}{L}\sum_{l=1}^{L} D(y,\theta_l) \qquad \text{(Equation 15)}$$

The DIC is then defined in terms of the difference between the two:

$$DIC = \ 2\,\widehat{D(x)} - D_{\hat{\theta}}(x) \qquad \text{(Equation 16)}$$

The DIC, similar to the better known Akaike information criterion (AIC), does not provide an absolute criterion of model fit, but is used to compare competing models. Specifically, the model with the lower DIC should be preferred. It can be used for hypothesis testing by comparing (nested or non-nested) models that embody the Null and alternate hypotheses. Lunn et al. [2013] suggest that differences in DIC between 5 and 10 are important. The DIC for our example model was 7129.0 and is shown at the end of the OpenBUGS script output, together with its components, as per equations 13 and 14:

```
  Dbar Dhat DIC pD
y 6584.0 6039.0 7129.0 545.2
```

Because the model quality and fit can always be improved with a more complex model, a measure of complexity should be used. The pD measure reported by OpenBUGS (in our example 545.2) is called the “effective number of parameters” and differs from the number of parameters as traditionally counted to reflect the fact that the prior distributions effectively acts to restrict the freedom of the model parameters [Lunn et al., 2013].

Results for small sample sizes may depend strongly on the specified prior probability distributions of model parameters. Perhaps counterintuitively, this is especially the case when different uninformative priors are used [Asparouhov and Muthen, 2010a]. However, this is because for small sample sizes, the likelihood plays a relatively smaller role in determining the posterior, so that different types of priors can exert their influence. While there are no guidelines as to which models are affected at which sample size, researchers should check for this “prior assumption dependence” by estimating the model with different uninformative priors [Asparouhov and Muthen, 2010a].

Recommendation:

- Use the PPP to assess model fit
- Use the DIC to compare alternative/competing models
- Especially for small sample sizes, re-estimate model with different uninformative priors (step 2)

Results

Only when the above steps of convergence assessment, thinning, and model quality evaluation are completed, should the results be reported. In our example, we can simply use “`summary(thinned)`” to get a summary of the parameter estimates in our thinned MCMC sample set.

To show the effect that the prior probability specifications have on the estimation results, we ran the Bayesian estimation with three different variances of the normal prior probability distributions for the loadings. These correspond to different degrees of certainty about the prior model parameter values. The initial estimation used the standard error of the mean (S.E.M.) of the published estimates from Table 1. This expresses relatively little certainty about the prior values. The second estimation used 1/10 times the standard error of the published estimates, expressing greater certainty about the published estimates, while the third estimation used 1/100 times the standard error of the published estimates, expressing even more certainty about the published estimates.

The results of the ML and Bayesian estimations are shown and compared in Table 12. The table shows that the Bayesian estimates and standard errors are of the same order of magnitude as the traditional ML estimates (column "Bayesian 1" in Table 12, variance of prior probability distribution on factor loadings equals the standard error of the mean from Table 1). In the Bayesian perspective, the ML estimates could be viewed as posterior estimates with an uninformative prior distribution because they make no use of existing information about parameter distributions, only of the sample data. When the certainty of the prior information is increased (column "Bayesian 2" in Table 12, variance of prior probability distribution on factor loadings equals 1/10 the standard error of the mean from Table 1) estimates for most parameters tend to be closer to the prior estimates, showing the influence of prior information on the estimates. When the certainty of the prior estimates is further increased (column "Bayesian 3" in Table 12, variance of prior probability distribution on factor loadings equals 1/100 the standard error of the mean from Table 1), the estimates tend to be still closer to the prior estimates. Table 12 also shows that the standard errors for the estimate are smaller when the prior means are more certain, and larger when the prior means are less certain.

Table 12: CFA model loadings for different estimation methods

Param.	ML Estimate		Prior Estimates (From Table 1)		Bayesian 1		Bayesian 2 (greater certainty)		Bayesian 3 (much greater certainty)	
	Est.	Std. Err.	Est.	Std. Err.	Est. (Mean)	Std. Dev.	Est. (Mean)	Std. Dev.	Est. (Mean)	Std. Dev.
Eou1	.881	.0468	.8432	.0095	.9143	.05294	.8723	.02285	.8489	.009182
Eou2	.899	.0462	.8202	.0154	.8553	.05065	.8313	.02444	.8237	.011295
Eou3	.922	.0453	.8327	.0135	.8455	.04804	.8274	.02209	.8316	.010390
Eou4	.827	.0486	.8217	.0211	.7185	.04655	.7317	.02744	.7967	.013656
Eou5	.895	.0463	.8344	.0158	.8507	.05082	.8318	.02483	.8341	.011489
Eou6	.939	.0446	.8682	.0092	.9066	.05026	.8809	.02110	.8721	.008987
Use1	.828	.0390	.8199	.0127	.7454	.04697	.7701	.02564	.8110	.010786
Use2	.837	.0406	.8652	.0105	.7986	.05012	.8233	.02523	.8580	.010057
Use3	.845	.0498	.8724	.0087	.8761	.05162	.8696	.02375	.8722	.008967
Use4	.861	.0466	.8728	.0124	.8705	.05182	.8655	.02611	.8721	.010731
Use5	.809	.0595	.8309	.0124	.9113	.05822	.8634	.02887	.8363	.010998
Use6	.880	.0594	.8429	.0100	.9685	.05537	.8998	.02550	.8520	.009949
Phi1,2	.612	.0403	NA	NA	.6130	.03902	.6130	.03900	.6137	.038984

Using Bayesian estimation methods, it is thus possible to build cumulative evidence of model parameter estimates and to incorporate prior knowledge in a statistically sound way. This allows researchers to keep a "running tally" of the best estimates of model parameters.

The substantive interpretation of the model and its estimated parameters, in terms of validity and reliability of indicators, the adequacy of the structural model in terms of explanatory value, etc. are the next steps a researcher needs to attend to. However, the fact that Bayesian estimation of the model was used has no effect on these and existing guidelines [e.g. Gefen et al., 2011] remain largely applicable.

In terms of reporting of results, our key recommendation is to report the choice of prior distribution. Because there is no "correct" prior and the prior can have a potentially strong effect on the results, especially at small sample sizes, researchers at the very least need to report *and justify* their choice of prior if it is an informative prior. At best, researchers report results of different models for different priors, including a "skeptical" one that represents a "no-effect" hypothesis [Lunn et al., 2013] . Researchers should also report the different diagnostics and the decisions based on them, like increasing the sample size, thinning the MCMC series, or discarding burn-in iterations. Finally, researchers should not only recommend the estimated mean or mode of the posterior distribution, but also credibility intervals, e.g. the 2.5% and 97.5% bounds. In case of severely skewed posteriors, researchers may want to include a plot of the distribution, as in Figure 2.

Recommendation:

- Report and justify the choice of informative prior distributions
- Report all diagnostics and the estimation decisions based on them
- Report the estimated mean or mode of the posterior distribution and credibility intervals for important parameters.

VI. CONCLUSION

Our specific contribution with this tutorial is to present a collection of best practices for Bayesian estimation and diagnostics to Information Systems researchers. These best practices are summarized in Figure 8 and Table 13. We used the OpenBUGS and R software packages for their flexibility and expressiveness in modeling, their wide-ranging support for diagnostics and their easy availability. This tutorial provides detailed instructions on how best practices can be instantiated with these software packages.

Our tutorial shows that Bayesian statistics is not inherently more difficult to apply than traditional methods. The often-cited computational requirement is no longer an impediment, and expressing a SEM in Bayesian terms is made easier with the availability of expressive software such as OpenBUGS. Given this, we believe that IS researchers should add Bayesian methods to the arsenal of tools used to evaluate their theories. We agree with Rupp et al. [2004] who suggest that "the appropriate question that a contemporary psychometrician should ask is not whether to Bayes but instead when to Bayes" (pg. 447). In other words, there are situations when a Bayesian approach is just one possible method, and other situations when it should be the preferred method. We hope that this tutorial will provide guidance to IS researchers to recognize when and how to use a Bayesian approach to structural equation modeling.

We believe that any research discipline is interested in meaningfully accumulating knowledge, rather than merely piling up results. Thus, the ability to reconcile different parameter estimates that occur when our theoretical and measurement models are reused and re-estimated, is important. From this perspective, Bayesian estimation is a tool to integrate our existing knowledge into the estimation and provide updated knowledge, in effect keeping a "running tally" of our best knowledge of model parameters. In the larger picture, Bayesian methods allow researchers to pay increasing attention to the parameter estimates produced by their models. They are, after all, part of the theory that is being proposed. Only by paying such attention to parameter estimates can we successfully refine our theories and build truly cumulative knowledge.

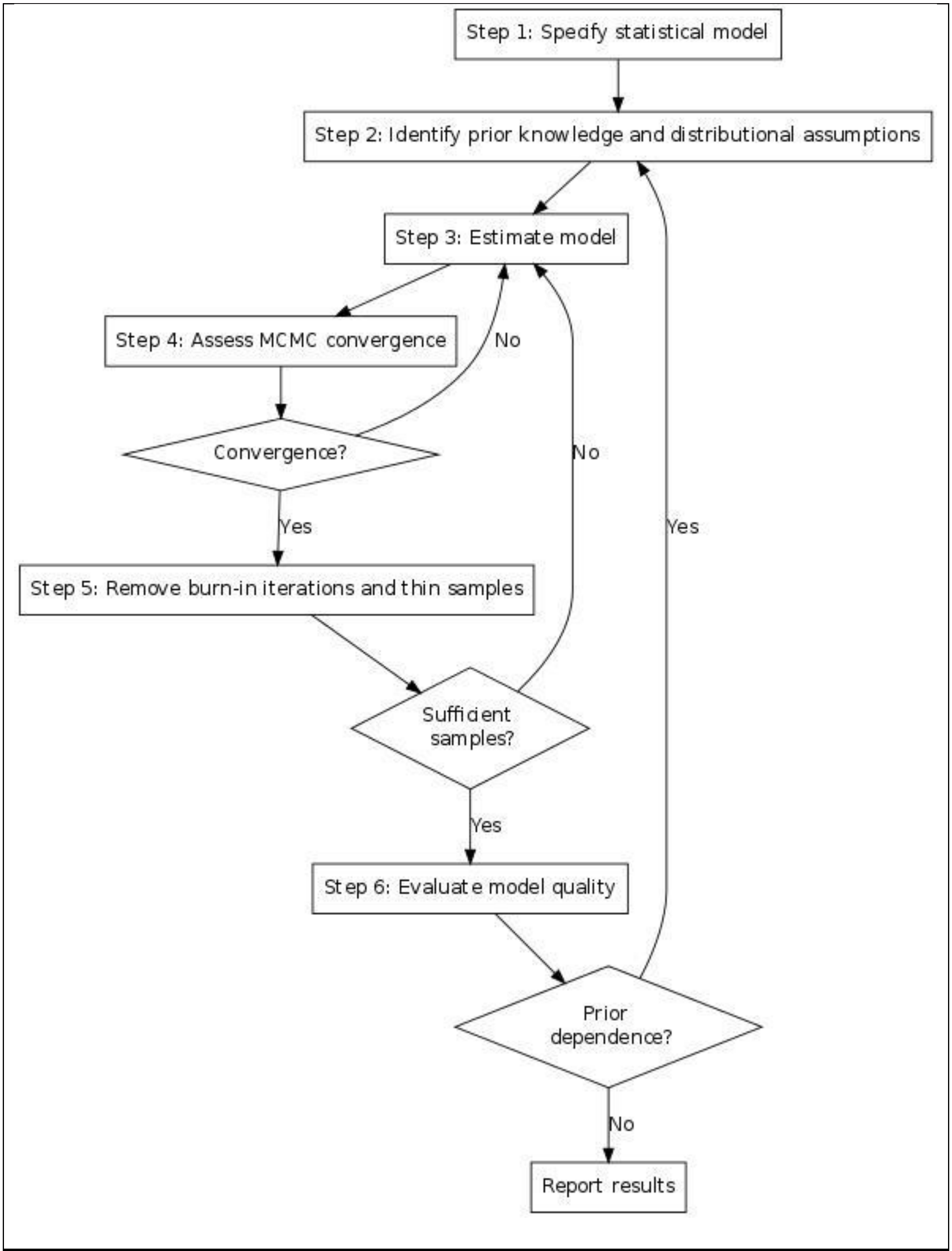

Step 1: Specify statistical model
Step 2: Identify prior knowledge and distributional assumptions
Step 3: Estimate model
Step 4: Assess MCMC convergence
No
Convergence?
No
Yes
Yes
Step 5: Remove burn-in iterations and thin samples
Sufficient samples?
Yes
Step 6: Evaluate model quality
Prior dependence?
No
Report results

Figure 8: Recommended process for Bayesian estimation of SEM

Table 13: Summary of recommended best practices

Pre-Study		
	Recommended:	Use Bayesian analysis for • non-standard models that are difficult to express in covariance or partial-least squares models (such as multi-level models, under-identified models, models with missing values and/or non-continuous variables) • estimation that allows the use of prior knowledge about parameter values, and/or estimation from small sample sizes
	Recommended:	• Meta-analysis is a valuable pre-cursor to Bayesian estimation. Use the meta-analytic results to aggregate data from former studies for use in Bayesian estimation
	Recommended:	Use the OpenBUGS software for Bayesian estimation because it is • Flexible • Expressive • Extendable • Free and open-source • Cross-platform • Scriptable
Step 1	Specify the basic statistical model	
	Recommended:	Specify a model to estimate latent variances as well as all loadings.
	Recommended:	Estimate all cross-loadings with realistic small prior probabilities with sufficient precision (inverse variance) to ensure the model can be estimated.
Step 2	Identify prior knowledge and distributional assumptions	
	Option 1:	Research literature for prior estimates
	Option 2:	Theoretically motivate informative prior distributions
	Option 3:	Use non-informative prior distributions. Such prior distributions should be “skeptical” in the sense that they reflect a null hypothesis of “no effect”, e.g. have a mean of zero for regression parameters.
Step 3	Estimate model	
	Recommended:	Use at least 3 MCMC chains
	Recommended:	Use at least 5000 sampling iterations
	Optional:	Set random number seed in OpenBUGS (using `modelSetRN(…)`) and specify initial values (using `modelInits(…)`) for repeatability
Step 4	Assess MCMC convergence	
	Recommended:	Visually assess convergence of MCMC chains using trace plots, sampling means of all chains should converge
	Recommended:	Visually assess posterior sampling distribution using density plots, should be similar to expected theoretical posterior distribution
	Recommended:	Use Geweke’s [1992] test for convergence, the z-statistic should be less than 1.96 for all parameters
	Recommended:	Use Heidelberger and Welch [1983] tests for stationarity. Stationarity and half-width tests should be passed by all parameters.
	Recommended:	Use Gelman’s Potential Scale Reduction (PSR) criterion for convergence, ensure all PSR values are less than 1.1.
Step 5	Remove burn-in iterations and thin samples	
	Recommended:	Discard the “burn-in” iterations from the sample, based on the results from step 4.
	Recommended:	Assess autocorrelation of samples within MCMC series
	Recommended:	Use Raftery and Lewis’ [1992; 1995] method to estimate required sample size
	Option 1:	Thin the series to reduce auto-correlation

Table 13: Summary of recommended best practices

	Option 2:	Increase the number of sampling iterations.
Step 5	Evaluate model quality	
	Recommended:	Assess posterior-predictive probability (PPP)
	Optional:	Compare different models using Deviance Information Criterion (DIC)
	Optional:	Perform sensitivity analysis to assess the impact of prior distribution dependence (especially for non-informative priors with small samples)
Reporting		
	Recommended:	Report and justify the choice of informative prior distributions
	Recommended:	Report all diagnostics and the estimation decisions based on them.
	Recommended:	Report the estimated mean or mode of the posterior distribution and credibility intervals for important parameters.

REFERENCES

Asparouhov, T. and Muthén, B. (2010a) “Bayesian Analysis using MPlus: Technical Implementation,” Version 3, September 29th, 2010. http://www.statmodel.com

Asparouhov, T. and Muthén, B. (2010b) “Bayesian Analysis of Latent Variable Models using MPlus,” Version 4, September 29th, 2010. http://www.statmodel.com

Bollen, K. (1989). *Structural Equations with Latent Variables.* John Wiley and Sons: Chichester, UK.

Brown, W.J. and Draper, D. (2006) “A Comparison of Bayesian and Likelihood-Based Methods for Fitting Multi-Level Models,” *Bayesian Analysis,* 1(3), 473-514.

Chin, W. W., Johnson, N., and Schwarz, A. (2008). “A fast form approach to measuring technology acceptance and other constructs,” *MIS Quarterly*, (32)4, pp. 687–703.

Congdon, P. (2006). *Bayesian Statistical Modelling* 2nd ed. Chichester, England: John Wiley & Sons, Ltd.

Davis, F. (1989). “Perceived usefulness, perceived ease of use, and user acceptance of information technology,” *MIS Quarterly*, (13), pp. 319–340.

Davis, F., Bagozzi, R., and Warshaw, P. (1989). “User acceptance of computer technology: A comparison of two theoretical models,” *Management Science*, (35), pp. 982–1002.

Evermann, J. and Tate, M. (2011) “Fitting Covariance Models for Theory Generation,” *Journal of the AIS,* (12)9, pp. 632-661.

French, B. F. and Finch, W. H. (2006). “Confirmatory factor analytic procedures for the determination of measurement invariance,” *Structural Equation Modeling*, (13)3, pp. 378–402.

Gefen, D., Straub, D.W. and Rigdon, E.E. (2011) “An Update and Extension to SEM Guidelines for Administrative and Social Science Research,” *MIS Quarterly*, (35)2, pp. iii-xiv.

Gelman, A., Carlin, J.B., Stern, H.S. and Rubin, D.B. (2004)*. Bayesian Data Analysis*, 2nd ed. London: CRC Press.

Gelman, A., Meng, X.-L., and Stern, H. (1996) “Posterior Predictive Assessment of Model Fitness via Realized Discrepancies,” *Statistica Sinica,* 6, 733-807.

Geweke, J. (1992). “Evaluating the accuracy of sampling-based approaches to calculating posterior moments. In J.M. Bernado, J.O. Berger, A.P. Dawid and A.F.M. Smith *Bayesian Statistics 4*th ed. Clarendon Press, Oxford, UK.

Heidelberger, P. and Welch, P.D. (1983). “Simulation run length control in the presence of an initial transient.” *Operations Research* (31), pp. 1109-1144.

Joseph, D., Ng. K-Y., Koh, C. and Ang, S. (2007). “Turnover of Information Technology Professionals: A Narrative Review, Meta-Analytic Structural Equation Modeling, and Model Development,” *MIS Quarterly*, (31)3, pp. 547-577.

King, W.R. and He, J. (2005). “Understanding the Role and Methods of Meta-Analysis in IS Research,” *Communications of the AIS,* (16), pp. 665-686.

Kruschke, J.K., Aguinis, H. and Joo, H. (2012). “The Time has Come: Bayesian Methods for Data Analysis in the Organizational Sciences,” *Organizational Research Methods*. (15)4, pp. 722-752.

Lee, S.Y. (2007). Structural Equation Modeling – A Bayesian Approach. John Wiley & Sons, Chichester, England.

Lee, S.Y., Song, X.Y. and Cai, J.H. (2010). “A Bayesian approach for nonlinear structural equation models with dichotomous variables using Logit and Probit links,” *Structural Equation Modeling,* (17)2, pp. 280-302.

Lunn, D., Jackson, C., Best, N, Thomas, A. and Spiegelhalter, D. (2013). *The BUGS Book – A practical introduction to Bayesian analysis,* Boca Raton, FL: CRC Press.

Ma, Q. and Liu, L. (2004). “The Technology Acceptance Model: A Meta-Analysis of Empirical Findings,” *Journal of Organizational and End User Computing*, (16)1, pp. 59-72.

Muthén, B. and Asparouhov, T. (2012) “Bayesian Structural Equation Modeling: A More Flexible Representation of Substantive Theory,” *Psychological Methods,* (17)3, 313-335.

Plummer, M., Best, N., Cowles, K., and Vines, K. (2006). “CODA: Convergence diagnosis and output analysis for MCMC”, *R News,* (6)1, 7-11.

Raftery, A.E. and Lewis, S.M. (1992) “One long run with diagnostics: Implementation strategies for Markov chain Monte Carlo,” *Statistical Science*, (7), pp. 493-497.

Raftery, A.E. and Lewis, S.M. (1995) “The number of iterations, convergence diagnostics and generic Metropolis algorithms,” In: W.R. Gilks, D.J. Spiegelhalter, and S.Richardson (eds.) *Practical Markov Chain Monte Carlo*, London, U.K.: Chapman and Hall.

Rupp, A.A., Dey, D.K. and Zumbo, B.D. (2009) “To Bayes or not to Bayes, From whether to when: Applications of Bayesian Methodology to Modeling,” *Structural Equation Modeling,* 11(3), 424-451.

Scheines, R., Hoijtink, H., and Boomsma, A. (1999) “Bayesian Estimation and Testing of Structural Equation Models,” *Psychometrika,* 64(1), 37-52.

Song, X.Y., Lee, S.Y. and Hsu, H.T. (2001) “Model Selection in Structural Equation Models with Continuous and Polytomous Data,” *Structural Equation Modeling,* 8(3), 378-396.

Song, X.Y. and Lee, S.Y. (2008). “A Bayesian approach for analyzing hierarchical data with missing outcomes through structural equation models,” *Structural Equation Modeling,* (15)2, pp. 272-300.

Song, X.Y. and Lee, S.Y. (2012). *Basic and Advanced Bayesian Structural Equation Modeling.* John Wiley and Sons, Chichester, UK.

Spiegelhalter, D.J., Best, N.G., Carlin, BP. And van der Linde, A. (2002). “Bayesian measure of model complexity and fit,” *Journal of the Royal Statistical Society, Series B* (64)4, pp. 583-639.

Yuan, Y. and MacKinnon, D.P. (2009) “Bayesian Mediation Analysis,” *Psychological Methods,* (14)4, pp. 301-322.

Zheng, Z. and Pavlou, P. (2010) “Toward a Causal Interpretation from Observational Data: A New Bayesian Networks Method for Structural Models with Latent Variables,” *Information Systems Research.* (21)2, pp. 365-391.

Zyphur, M.J. and Oswald, F.L. (2013) “Bayesian Estimation and Inference: A User’s Guide,” *Journal of Management*. OnlineFirst on August 11, 2013.

ABOUT THE AUTHORS

Joerg Evermann is an associate professor of Information Systems at Memorial University of Newfoundland. Joerg received his PhD from the Sauder School of Business at The University of British Columbia. His research interests include quantitative research methods and his work has been published in journals such as *Journal of the AIS, Organizational Research Methods* and *Structural Equation Modeling* and he has published numerous methodology papers at the *International Conference on Information Systems*. His other interests are in conceptual modeling, data integration, and business process management. His work on these topics has appeared in journals such as *IEEE Transactions on Software Engineering, IEEE Transactions on Knowledge and Data Engineering, Information Systems,* and *Information Systems Journal.*

Mary Tate is a senior lecturer at Victoria University of Wellington, New Zealand. She is the author of more than 50 peer-reviewed publications, and her work has been presented in journals such as *Journal of the Association for Information Systems, Communications of the Association for Information Systems*, and *Behaviour and Information Technology*; and leading international conferences. She has a strong interest in research methods, measurement theory, and the measurement of IS value and impact using multi-methods. Her other interests include service quality, service delivery, and channel management including new media channels.