Model Comparison in Psychology

Jay I. Myung# and Mark A. Pitt
Department of Psychology
Ohio State University
Columbus, OH 43210
{myung.1, pitt.2}@osu.edu
#Corresponding author

July 5, 2016

Word count: 12,569

# Contents

1 Abstract 3  
2 Introduction 3  
3 Foundations of Model Comparison 4  
   3.1 Model Evaluation Criteria 4  
   3.2 Follies of a Good Fit 5  
   3.3 Generalizability: The Yardstick of Model Comparison 6  
   3.4 The Importance of Model Complexity 6  
4 The Practice of Model Comparison 9  
   4.1 Model Falsifiability, Identifiability, and equivalence 9  
   4.2 Model Estimation 13  
   4.3 Methods of Model Comparison 14  
      4.3.1 Penalized-likelihood Methods 14  
         AIC, AICc and BIC: 15  
         Fisher Information Approximation (FIA): 16  
         Normalized Maximum Likelihood (NML): 17  
      4.3.2 Bayesian Methods 18  
         Bayesian Model Selection (BMS): 18  
         Deviance Information Criterion (DIC): 20  
      4.3.3 Direct Estimation Methods 21  
         Cross Validation (CV): 21  
         Generalization Criterion (GC): 22  
         Accumulative Prediction Error (APE): 23  
   4.4 Illustrated Example 24  
5 Conclusion 26  
6 Acknowledgement 27  
7 Appendix A: Matlab Code for Illustrated Example 27  
8 Appendix B: R2JAGS Code for Illustrated Example 33
1 Abstract

How does one decide between competing explanations of the phenomenon under study from limited and noisy data? This is the problem of model comparison, which is at the very foundation of scientific progress in all disciplines. This chapter introduces quantitative methods of model comparison in the context of cognitive modeling. The chapter begins with a conceptual discussion of some key elements of model evaluation. This is followed by a comprehensive and comparative review of major methods of model comparison developed to date, especially for modeling data in cognitive science. The final section illustrates the use of selected methods in choosing among four mathematical models of memory retention.

*Key Terms:* model evaluation and comparison, model fitting, model complexity, Akaike Information Criterion, cross validation, minimum description length, Bayes factor.

2 Introduction

Models in cognitive science are formal descriptions of a cognitive process (e.g., memory, decision making, learning). They are an attractive and powerful tool for studying cognition because their specification is so explicit and their performance so precise (Fum et al., 2007). These qualities afford thorough evaluation of the model, from how it instantiates theoretical assumptions, to its ability to mimic human data, and to its performance relative to other models. Another chapter in this volume by Stephan Lewandowsky covers the first two of these; the present chapter addresses the third.

Model comparison is inseparable from model building. Whether one is comparing cognitive models, or even purely statistical models, qualitative and quantitative methods are needed to guide their evaluation and justify choosing one model over its competitors. The goal in model comparison is to determine which of a set of models under consideration provides the “best” approximation in some defined sense to the cognitive process given the data observed. It is important to approach this enterprise from a position of considerable humility. All contending models are wrong the moment they are proposed (Box, 1976, p. 792). They have to be wrong given the contrast between how little we know about the properties and operation of what is undoubtedly a highly complex cognitive process, and how primitive and uninformed the models are themselves. Even when there is a vast literature on which to draw while building a model, it rarely can provide the detail necessary to justify critical decisions such as parameterization and choice of functional form. This state of affairs makes it clear that the task of identifying the true model is overly ambitious and somewhat misguided. Rather, a more productive question to ask is which of the models under consideration provides the most reasonable quantitative account of and explanation for the data, recognizing that all models are in essence deliberate simplifications of a vastly complex system (McClelland, 2009; Shiffrin, 2010).

In this chapter, we review quantitative methods of model comparison within the context of mathematical models of cognition, which includes neurocomputational models. By a mathematical model, we mean a model for which the likelihood function is given explicitly in analytic form as a function of parameters. In statistical terms, it is defined as a parametric family of probability distributions that are generated from varying a model’s parameters across their ranges. The narrow focus of this chapter is a reflection of the field itself. Most model comparison methods
in statistics have been developed for comparing this class of models. There are rich traditions in other styles of modeling in cognitive science, such as simulation-based models (e.g., Shiffrin and Steyvers, 1997), connectionist models (e.g., Plaut et al., 1996), and cognitive architectures (e.g. Anderson and Lebiere, 1998). Their formulation precludes use of many, but not all, of the methods we review here. We will point out which methods are sufficiently versatile to be used to compare them. Readers who are interested in model comparison methods for a broad class of psychometric linear and nonlinear models, such as generalized linear mixed-effects models (GLMM; e.g., Gries, 2015), structural equation models (e.g., Bollen, 1989), and item response models (e.g., Hambleton et al., 1991), should also find this chapter of interest.

We begin the chapter by describing the criteria used to evaluate models and then elaborate on those that have been quantified. This is followed by a discussion of some of the most widely used model comparison methods and an application example comparing a subset of them. The chapter ends with some guidelines on their use. Additional readings on model comparison that the reader might be interested in reading include Myung and Pitt (2002), Shiffrin et al. (2008), Vandekerckhove et al. (2015), and Myung et al. (2016). Note that Myung and Pitt (2002) appeared in the third edition of the same handbook series as the current one. The present chapter is written as a revised and updated version of this earlier chapter, focusing solely on model comparison.

3 Foundations of Model Comparison

3.1 Model Evaluation Criteria

The problem of model comparison is that of choosing one model, among a set of candidate models, that is “best” in some defined sense. However, before we discuss quantitative methods for identifying such a model, it is important that any model be evaluated for some minimum level of adequacy, as there would be no point in considering further models that fail to meet this standard. One can think of a number of criteria under which the adequacy of a model can be evaluated. What follows is a list of some of these along with short definitions. Further discussion of many can be found in Jacobs and Grainger (1994) and Pitt et al. (2002).

**Plausibility:** A model is said to be plausible if its assumptions, whether behavioral or physiological, are not contrary to the established findings in the literature.

**Explanatory adequacy:** A model satisfies this if the model provides a principled account of the phenomenon of interest that is consistent with what is known and accepted in the field.

**Interpretability:** This criterion refers to the extent to which the parameters of a model are linked to known processes so that the value of each parameter reflects the strength or activity of the presumed underlying process.

**Faithfulness:** A model is faithful if a model’s success in accounting for the phenomenon under study derives largely from the theoretical principles it embodies and not from the non-theoretical choices made in its computational implementation (Myung et al., 1999). Model faithfulness is closely related to what Lewandowsky (1993) refers to as the *irrelevant specification problem* (see also, Fum et al., 2007).

**Confirmability:** A model is confirmable if there exists a unique data structure that could only be accounted for by the model, but not by other models under consideration, as succinctly stated in the following quote, “[I]t must be possible to verify a new prediction that only this theory makes” (Smolin, 2006, pp. xiii).
Goodness of fit: Goodness of fit (GOF) is a descriptive adequacy criterion of model evaluation, as opposed to an explanatory criterion, as described above. Simply put, a model satisfies the GOF criterion if it fits well observed data. Examples of GOF measures include the coefficient of determination (i.e., r-squared, $r^2$), the root mean square error (RMSE), and the maximum likelihood (ML). The first two of these measure the discrepancy between model predictions and actual observations and are often used to summarize model fit in a regression analysis. The ML is obtained by maximizing the probability of the observed data under the model of interest, and as such represents a measure of how closely, in the sense of probability theory, the model can capture the data (Myung, 2003).

Generalizability: Generalizability, or predictive accuracy, refers to how well a model predicts new and future observations from the same process that generated the currently observed data, and is the gold standard by which to judge the viability of a model, or a theory, for that matter (Taagepera, 2007).

Model complexity/simplicity concerns whether a model captures the phenomenon in the simplest possible manner. To the extent this is achieved, a model would satisfy this criterion. The conventional wisdom is that the more parameters a model has, the more complex it is. Although intuitive, we will show later in this chapter that this view of model complexity based solely on the number of model parameters does not fully capture all aspects of complexity.

Whereas each of the above eight criteria is important to consider in model comparison, the last three (goodness of fit, generalizability, and complexity) are particularly pertinent to choosing among mathematical models, and quantitative methods have been developed with this purpose in mind. In the following sections, we begin by defining these three criteria in more detail and then demonstrate their inter-relationship in an illustrated example.

3.2 Follies of a Good Fit

Goodness of fit is a necessary component of model comparison. Because data are our only link to the cognitive process under investigation, if a model is to be considered seriously, then it must be able to describe well the output from this process. Failure to do so invalidates the model. Goodness of fit, however, is not a sufficient condition for model comparison. This is because model comparison based solely on goodness of fit may result in the choice of a model that over-fits the data. Why? Because the model will capture variability present in the particular data set that comes from sources other than the underlying process of interest.

Statistically speaking, the observed data are a sample generated from a population, and therefore contain at least three types of variation: (1) variation due to sampling error because the sample is only an estimate of the population; (2) variation due to individual differences, and (3) variation due to the cognitive process of interest. Most of the time it is only the third of these that we are interested in modeling, yet goodness-of-fit measures do not distinguish between any of them. Measures such as $r^2$ and maximum likelihood treat all variation identically. They are blind to its source, and try to absorb as much of it as possible as demonstrated below. What is needed is a means of filtering out or mitigating these unwanted sources of variation, essentially random noise or errors. Generalizability achieves this.
3.3 Generalizability: The Yardstick of Model Comparison

Generalizability (GN), which is often used interchangeably with the term “predictive accuracy,” refers to a model’s ability to fit not only the observed data in hand but also future, unseen data sets generated from the same underlying process. To illustrate, suppose that the model is fitted to the initial set of data and its best-fitting parameter values are obtained. If the model, with these parameter values held constant, also provides a good fit to additional data samples collected from replications of that same experiment (i.e., the same underlying probability distribution or regularity), then the model is said to generalize well. Only under such circumstances can we be sure that a model is accurately capturing the underlying process, and not the idiosyncracies (i.e., noise) of a particular sample.

![Figure 1: Illustration of the trade-off between goodness of fit and generalizability. The three fictitious models (curves) were fitted to the same data set (solid circles), and new observations are shown by the x symbols.](image)

The superiority of this criterion over GOF becomes readily apparent in the following illustration. In Figure 1, the solid circles represent observed data points and the curves represent best-fits by three hypothetical models. Model A, a linear model, clearly does a poor job in accounting for the curve-linear trend of the downward shift, and thus can be eliminated from further consideration. Model B not only captures the general trend in the current data but also does a good job in capturing new observations (x symbols). Model C, on the other hand, provides a much better fit to the observed data than model B, but apparently it does so by fitting the random fluctuations of each data point as well as the general trend, and consequently, suffers in fit when new observations are introduced into the sample, thereby representing an instance of overfitting. As the example shows, generalizability is a reliable way to overcome the problem of noise and extract the regularity present in the data. In short, among the three models considered, Model B is the “best generalizing” model. Further examples below will demonstrate why generalizability should be adopted as the primary quantitative criterion on which the adequacy of a model is evaluated and compared.

3.4 The Importance of Model Complexity

Intuitively, model complexity refers to the flexibility inherent in a model that enables it to fit diverse patterns of data (e.g., Myung and Pitt, 1997; Myung, 2000). For the moment, think of it as a continuum, with simple models at one end and complex models at the other. A simple model
Table 1: Goodness of fit and generalizability of four models differing in complexity.

<table>
<thead>
<tr>
<th>Model</th>
<th>LIN</th>
<th>EXP (true)</th>
<th>POW</th>
<th>EXPOWS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of params</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>Goodness of fit (Sample 1)</td>
<td>0.790 (0%)</td>
<td>0.884 (0%)</td>
<td>0.710 (0%)</td>
<td>0.905 (100.0%)</td>
</tr>
<tr>
<td>Generalizability (Sample 2)</td>
<td>0.761 (2.1%)</td>
<td>0.860 (81.9%)</td>
<td>0.684 (0.9%)</td>
<td>0.835 (15.1%)</td>
</tr>
</tbody>
</table>

Note: Shown above are the mean $r^2$ value of the fit of each model and the percentage of samples (out of 1000) in which the model provided the best fit to the data (in parenthesis). The four models that predict proportion correct are defined as follows: LIN: $p = at + b$; EXP: $p = ae^{-bt}$; POW: $p = a(t+1)^{-b}$; EXPOWS: $p = ae^{-bt}+c(t+1)^{-d}sin(et)+f$.

A thousand pairs of binomial samples were generated from model EXP with $a = 0.95$ and $b = 0.10$ under the binomial likelihood, $Bin(N = 20, p)$, for a set of 21 time intervals, $t = (0.5, 1.7, 2.9, ..., 23.3, 24.5)$, spaced in an increment of 1.2. Goodness of fit (GOF) was assessed by fitting each model to the first sample (Sample 1) of each pair and finding the maximum likelihood estimates (Myung, 2003) of the model parameters. Generalizability (GN) was assessed by the model’s fit to the second sample (Sample 2) of the same pair without further parameter tuning.

assumes that a relatively narrow range of more of less similar patterns will be present in the data. When the data exhibit one of these few patterns, the model fits the data very well; otherwise, its fit will be rather poor. All other things being equal, simple models are attractive because they are sufficiently constrained to make them easily falsifiable, requiring a small number of data points to disprove the model. In contrast, a complex model is usually one with many parameters that are combined in a highly nonlinear fashion, do not assume a single structure in the data. Rather, like a chameleon, the model is capable of assuming multiple structures by finely adjusting its parameter values. This enables the model to fit a wide range of data patterns. This extra complexity does not necessarily make it suspect. Rather the extra complexity must be justified to choose the more complex model over the simpler one.

There seems to be at least two dimensions of model complexity: (1) the number of free parameters and (2) the functional form. The latter refers to the way in which the parameters are combined in the model equation. For example, consider the following two models with normal errors: $y = ax + b + e$ and $y = ax^b + e$, where $e \sim N(0, \sigma^2)$. They both have the same number of parameters (three, i.e., $\theta = (a, b, \sigma)$) but differ in functional form. A concrete example that demonstrates the influence of functional form complexity in the context of structural equation modeling is discussed in Preacher (2006). Further, Vanpaemel (2009) and Veksler et al. (2015) each developed a stand-alone quantitative measure of model flexibility that is sensitive to both the number of parameters and the functional form and that can be quite useful for assessing a model’s intrinsic flexibility to fit a wide spectrum of data patterns. In any case, the two dimensions of model complexity, and their interplay, can improve a model’s fit to the data without necessarily improving its generalizability. This is illustrated next with simple models of retention.

In Table 1, four models were compared on their ability to fit two data samples generated by the two-parameter exponential model denoted by EXP, which, by definition, is the true model. Goodness of fit (GOF) was assessed by finding parameter values for each model that gave the best fit to the first sample. With these parameters fixed, generalizability (GN) was then assessed by fitting the models to the second sample. In the first row of the table are each model’s mean fit, measured by $r^2$, to the data drawn from EXP. As can be seen, EXP fitted the data better than LIN or POW, which are incorrect models. What is more interesting are the results for EXPOWS. This model has four more parameters than the first three models and contains the true model as a special
case. Note that EXPOWS provided a better fit to the data than any of the other three models including EXP. Given that the data were generated by EXP, one would have expected EXP to fit its own data best at least some of the time, but this never happened. Instead, EXPOWS always fitted better than the other three models including the true one. The improvement in fit of EXPOWS over EXP represents the degree to which the data were overfitted. The four extra parameters in the former model enabled it to absorb non-systematic variation (i.e., random error) in the data, thus improving fit beyond what is needed to capture the underlying regularity. Interestingly also, note the difference in fit between LIN and POW ($r^2 = 0.790$ vs. $r^2 = 0.710$). This difference in fit must be due to functional form because these two models differ only in how the parameters and data are combined in the model equation.

The results in the second row of Table 1 demonstrate that overfitting the data in Sample 1 results in a loss of generalizability to Sample 2. The $r^2$ values are now worse (i.e., smaller) for EXPOWS than for EXP (0.835 vs. 0.860), the true model, and also, the overly complex model yielded the best fit much less often than EXP (15.1% vs. 81.9%).

To summarize, the above example demonstrates that the best-fitting model does not necessarily generalize the best, and that model complexity can significantly affect generalizability and goodness of fit. A complex model, because of its extra flexibility, can fit a single data set better than a simple model. The cost of the superior fit shows up in a loss of generalizability when fitted to new data sets, precisely because it overfitted the first data set by absorbing random error. It is for this reason that quantitative methods for measuring how well a model fits a data set ($r^2$, percent variance accounted for, maximum likelihood) are inadequate as model comparison criteria. Goodness of fit is a necessary dimension that a comparison criterion must capture, but it is insufficient because model complexity is ignored.

Figure 2 illustrates the relationship among goodness of fit (GOF), generalizability (GN), and model complexity. Fit index such as $r^2$ is represented along the vertical axis and model complexity along the horizontal axis. GOF keeps increasing as complexity increases. GN also increases positively with complexity but only up to the point where the model is sufficiently complex to capture the regularities underlying in the data. Additional complexity beyond this point will cause a drop in generalizability as the model begins to capture random noise, thereby overfitting the data. The three graphs in the bottom of the figure represent fits of three fictitious models—the same as those in Figure 1. The linear model on the left panel is not complex enough to match the complexity of the data (solid circles). The curve-linear model on the center panel is well matched to the complexity of the data, achieving the peak of the generalizability function. On the other hand, the cyclic model on the right panel is an overly complex one that captures idiosyncratic variations in the data and thus generalizes poorly to new observations (x symbols).

In conclusion, a model must not be chosen based solely on its goodness of fit. To do so risks selecting an overly complex model that generalizes poorly to other data generated from the same underlying process, thus resulting in a “very bad good fit” (Lewandowsky and Farrell, 2011, p. 198). If the goal is to develop a model that most closely approximates the underlying process, the model must be able to fit not only the current but also all future data well. Only generalizability can measure this property of the model, and thus should be used in model comparison.
Figure 2: A schematic illustration among goodness of fit (GOF), generalizability (GN), and model complexity. Shown at the bottom are concrete examples of three models that differ in model complexity.

4 The Practice of Model Comparison

It is necessary to ensure that the models of interest satisfy a few prerequisites prior to applying model comparison methods. We describe them and then review three classes of model comparison methods.

4.1 Model Falsifiability, Identifiability, and equivalence

Before one can contemplate the evaluation and comparison of a set of models, as a minimally necessary condition for the exercise, one should ensure that each model be both falsifiable and identifiable. Otherwise, the comparison is likely to be of little value because the models themselves are uninterpretable or cannot be taken seriously. In addition to these two concepts, also discussed in this section is model equivalence, which the reader should find particularly useful in his/her enterprise of cognitive modeling.
Model falsifiability: Falsifiability (Popper, 1959), also called testability, refers to whether there exist potential observations that are inconsistent with the model (i.e., data that it does not predict).\footnote{Falsifiability is related to confirmability introduced earlier, in that the latter is a stronger criterion than the former: A confirmable model is always falsifiable but a falsifiable model might not be confirmable.} This is a necessary precondition for testing a model; unless a model is falsifiable, there is no point in testing the model. Put another way, an unfalsifiable model is one that can describe all possible patterns of data that can arise in a given experiment. Figure 3 shows an example of an unfalsifiable model. The one-parameter model, defined as $y = (\sin(at)+1)/2$, is unfalsifiable because the model’s oscillation frequency parameter ($a$) can be changed to an infinite number of positive values and the function will still pass through all of the data points.

Figure 3: Example of an unfalsifiable model. The solid circles denote data points, and the solid curve represents the model equation defined as $y = (\sin(at)+1)/2$, $(0 < t < 25)$ with $a = 8$. This one-parameter model becomes unfalsifiable for $0 < a < \infty$.

A rule of thumb, often used with linear models, is to judge that a model is falsifiable if the number of its parameters is less than the number of data points, or equivalently, if the degrees of freedom is positive. This “Counting Rule” turns out to be imperfect and even misleading, especially for nonlinear models. The case in point is Luce’s choice model (Luce, 1956). The model assumes that the probability of choosing choice alternative $i$ over alternative $j$ is determined by their respective utility values in the following form:

$$P_{i \succ j} = \frac{u_i}{u_i + u_j}, \quad (i,j = 1,\ldots,s)$$

where $u_i (> 0)$ is the utility parameter for choice alternative $i$ to be estimated from the data. Note that the number of parameters in the model is equal to the number of choice alternatives ($s$) whereas the number of independent observations is equal to $s(s - 1)/2$. Hence, for $s = 3$, both
the number of parameters and the number of observations are equal, yet it is easy to show that
the model is falsifiable in this case. In another, more dramatic example, Bamber and van Santen
(1985, p. 453) showed that the number of parameters (7) in a model exceeded the number of data
observations (6), yet the model was still falsifiable! Jones and Dzhafarov (2014) discusses a more
recent example of unfalsifiability for a class of stochastic process models of choice reaction time.

For a formal definition of falsifiability, along with more rigorous rules for determining
whether a model is falsifiable or not, especially for nonlinear models, the reader is directed to

Model identifiability: Model identifiability refers to whether the parameters of a model are
unique given observed data. That is, if two or more different sets of the parameter values can
yield an equally good fit, the model is not identifiable (i.e., unidentified). When this happens, the
parameter values of the model become uninterpretable.

To illustrate, consider a three-parameter model of \( y = a + bx + cx^2 \) and suppose that two data
points are obtained, say \((x_1, y_1) = (1, 1)\) and \((x_2, y_2) = (2, 5)\). The model is then unidentifiable
given these data. This is because there exist multiple sets of the models parameter values \( \theta =
(a, b, c) \) that fit the data equally well and in fact, perfectly, for example, \((a, b, c) = (-1, 1, 1)\) and
\((a, b, c) = (-5, 7, -1)\). There are indeed an infinite number of such parameter values of the model
that can provide an equally good description of the data. In order for this model to be identifiable,
three or more data points are needed.

A rule of thumb often used to assess model identifiability is to see if the number of ob-
servations exceeds the number of free parameters, i.e., a positive value of the degrees of freedom
(df). Again, as is in the case with model falsifiability discussed above, Bamber and van Santen
(1985) showed that this heuristic “counting rule” is imperfect, especially for nonlinear models, and
provided a proper definition as well as testing scheme of identifiability.

As alluded above, falsifiability and identifiability are related to each other but they are not
the same. A case in point is the Fuzzy Logical Model of Perception (FLMP; Oden and Massaro,
1978). To demonstrate this situation, consider a letter recognition experiment in which participants
have to classify the stimulus as belonging to one of two categories, A and B. Assume that the
probability of classifying a stimulus as a member of category A is a function of the extent to which
the two feature dimensions of the stimulus \((i\ and\ j)\) support the category response (Massaro and
Friedman, 1990). Specifically, FLMP assumes that the response probability is a function of two
parameters, \( c_i \) and \( o_j \), each of which represents the degree of support for a category A response
given the specific \( i \) and \( j \) feature dimensions of an input stimulus:

\[
P_{ij}(c_i, o_j) = \frac{c_i o_j}{c_i o_j + (1 - c_i)(1 - o_j)}
\]  

where \( 0 < c_i, o_j < 1, 1 \leq i \leq s \) and \( 1 \leq j \leq v \). In the equation, \( s \) and \( v \) represent the number
of stimulus levels on the two feature dimensions, \( i \) and \( j \), respectively, and together constitute the
design of the experiment.

FLMP is falsifiable, which can be shown using the falsifiability test mentioned earlier (Bam-
ber and van Santen, 1985). For example, one can easily come up with a set of response probabilities
that do not fit into the model equation, such as \( P_{ij}(a_i, b_j) = (a_i + b_j)/2 \) for \( 0 < a_i, b_j < 1 \).

Regarding the identifiability of FLMP, for the \( s \times v \) factorial design, the number of inde-
pendent observations is \( sv \), and the number of parameters of FLMP is \( (s + v) \). For example, for
s = v = 8, the number of observations is 64, which far exceeds the number of parameters in the model (16). Surprisingly, however, Crowther et al. (1995) showed that FLMP is still unidentifiable. According to their analysis, for any given set of parameter values \((c_i, o_j)\) that satisfy the model equation in Eq. (2), another set of parameter values \((c'_i, o'_j)\) that also produce the same prediction can always be obtained through the following transformation:

\[
c'_i = \frac{c_i}{1 + z(1 - c_i)}; \quad o'_j = \frac{o_j(1 + z)}{1 + zo_j}
\]  

for a constant \(z > -1\). Given that there are an infinite number of possible \(z\) values, there will be an equally infinite number of parameter sets, each of which provides exactly the same fit to the observed data. Figure 4 shows four selected sets of parameters obtained by applying the above equation. For example, given a parameter set, say \(c = (0.1, 0.3, 0.5, 0.7, 0.9)\), one can obtain another set \(c' = (0.36, 0.68, 0.83, 0.92, 0.98)\) for \(z = -0.8\), or \(c' = (0.05, 0.18, 0.33, 0.54, 0.82)\) for \(z = 1\). Note that interestingly, the parameter values do change under the transformation in Eq. (3), but their ordinal relationships are preserved, in fact for all choices of the constant \(z (> -1)\). In short, given the unidentifiability of FLMP, one cannot meaningfully interpret the magnitudes of its parameters, except their ordinal structure.

Can FLMP be made identifiable? The answer is “yes.” For instance, one of its parameters can be fixed to a preset constant (e.g., \(c_k = 0.25\), for some \(k\)). Alternatively, the model equation can be modified to accommodate four response categories instead of two. For further details, the reader is referred to Crowther et al. (1995).

**Model equivalence:** This is a scarcely mentioned but important concept every cognitive modeler should be familiar with. For a given model equation, one can rewrite it in an equivalent form through a reparameterization of its parameters. As a concrete example of what is known as the
reparametrization technique in statistics, consider a one-parameter exponential model defined as $y = e^{-ax}$ where $0 < a < \infty$. This model can be rewritten as $y = b^x$, where $0 < b < 1$, using the following reparametrization of the original parameter $a$: $b = e^{-a}$.

To provide another and more substantive example of model equivalence, let us revisit the model \( FLMP \) defined in Eq. (2). For this model, there exist at least two equivalent forms (e.g., Crowther et al., 1995, pp. 404-405), and they are

\[
P_{ij}(\alpha_i, \beta_j) = \frac{1}{1 + e^{-(\alpha_i + \beta_j)}}, \quad (\alpha_i = \ln \frac{c_i}{1 - c_i}; \beta_j = \ln \frac{o_j}{1 - o_j})
\]

\[
P_{ij}(u_i, v_j) = \frac{1}{1 + u_i v_j}, \quad (u_i = \frac{1 - c_i}{c_i}; v_j = \frac{1 - o_j}{o_j})
\]

where $-\infty < \alpha_i, \beta_j < \infty$ and $0 < u_i, v_j < \infty$. Note that $P_{ij}(\alpha_i, \beta_j) = P_{ij}(u_i, v_j) = P_{ij}(c_i, o_j)$ for all pairs of $(i, j)$.

It is important to note that different model forms created by reparametrization are equivalent to one another for all statistical and practical purposes. That is, the equivalent models would all fit any given data exactly the same, albeit with different values of their parameters, and would lead to exactly the same interpretation of, and conclusion from the data.

### 4.2 Model Estimation

Once data have been collected and the model is shown to be falsifiable and identifiable, one is in a position to assess the model’s goodness of fit to the experimental data. Recall that a model is defined as a parametric family of probability distributions indexed by model parameters. Formally, \( model M = \{ f(y|\theta) \mid \theta \in \Theta \} \), where \( y = (y_1, ..., y_n) \), \( \theta = (\theta_1, ..., \theta_k) \), and \( \Theta \) is the parameter space.

As such, the model contains many (theoretically infinite) probability distributions, each associated with a distinct set of parameter values. The main objective of model estimation is to find a set of parameter values that best fits the observed data in some defined sense—the procedure called parameter estimation in statistics.

There are two generally accepted methods of parameter estimation (Myung, 2003): least square estimation (LSE) and maximum likelihood estimation (MLE). In LSE, the parameter values that minimize the sum of squared errors (SSE) between observations and predictions are sought:

\[
SSE(\theta) = \sum_{i=1}^{n} (y_i - y_{prd,i}(\theta))^2
\]

where $y_{prd,i}(\theta)$ is the model’s prediction for observation $y_i$. On the other hand, in MLE the parameter values that maximize the likelihood of the data, $f(y|\theta)$, or equivalently, the logarithm of the likelihood are sought:

\[
\ln f(y|\theta) = \sum_{i=1}^{n} \ln f(y_i|\theta)
\]

under the assumption of independent observations. The parameter values that maximize MLE or minimizes MSE are usually sought numerically using optimization algorithms implemented on computer. The LSE solution tends to differ from the MLE solution unless all $y_i$’s are normally distributed with the same variance. MLE is generally a preferred method of parameter estimation.
particularly in model comparison. From this point on, unless otherwise stated, we will assume that a model’s goodness of fit is assessed by MLE, and the best-fitting parameter vector obtained in MLE is denoted by \( \hat{\theta} \).

MLE is solely a method of model evaluation, not a method of model comparison. The latter requires additional conceptualization and formalization, to which we now turn our attention.

4.3 Methods of Model Comparison

The trade-off between goodness-of-fit and complexity illustrated earlier is what makes model comparison so difficult. The model must be complex enough to describe the variation in any data sample that is due to the underlying process, yet not overfit the data by absorbing noise and thus losing generalizability. Conversely, the model must not be too simple to capture the underlying process and thereby underfit the data, which will also lower generalizability. The goal of model comparison methods is to estimate a model’s generalizability by weighting fit against complexity.

In this section we provide a comprehensive overview of major methods of model comparison developed to date, especially for modeling data in the behavioral and social sciences. We divide them into three classes based on the approach. The first class of methods we discuss are penalized-likelihood methods, so called because they include a measure of fit along with one or more additional terms that penalize the model for its complexity. The third class of methods are direct estimation methods. As the name suggests, they are prescriptions for directly estimating generalizability, thus obviating the need to quantify fit and complexity independently. The second class of methods are Bayesian comparison methods. We view them as a hybrids of the others. While some are Bayesian equivalents of penalized-likelihood measures, fit and complexity are not partialled out, making them functionally more akin to direct estimation methods.

For additional and complementary treatments of the topic, the interested reader should consult three special journal issues on model comparison (Myung et al., 2000; Wagenmakers and Waldorp, 2006; Gluck et al., 2008), and recent review articles (Shiffrin et al., 2008; Vandekerckhove et al., 2015).

4.3.1 Penalized-likelihood Methods

Among the many methods of model comparison, one class of methods that are those that make an appropriate adjustment to a model’s goodness of fit by separately quantifying complexity and combining it with a measure of fit.

Five representative methods that are currently in use are the Akaike Information Criterion (AIC; Akaike, 1973; Bozdogan, 2000), the second-order AIC (AICc; Hurvich and Tsai, 1989), the Bayesian Information Criterion (BIC; Schwarz, 1978), the Fisher Information Approximation (FIA; Rissanen, 1996; Grünwald, 2000), and the Normalized Maximum Likelihood (NML; Rissanen, 2001; Myung et al., 2006).
They are defined as follows:

\[
AIC = -2 \ln f(y|\hat{\theta}) + 2k \quad \quad AIC_c = -2 \ln f(y|\hat{\theta}) + 2k + \frac{2k(k+1)}{n-k-1} \\
BIC = -2 \ln f(y|\hat{\theta}) + k \ln(n) \\
FIA = -\ln f(y|\hat{\theta}) + \frac{k}{2} \ln \frac{n}{2\pi} + \ln \int \sqrt{\det(I(\theta))} \, d\theta \\
NML = -\ln f(y|\hat{\theta}) + \ln \int f(z|\hat{\theta}(z)) \, dz.
\]  

In the above equation, \( \ln \) is the natural logarithm of base \( e \), \( y = (y_1, ..., y_n) \) is a vector of observed data, \( f(y|\hat{\theta}) \) is the maximum likelihood of the data, \( z \) is a vector variable of potential data, \( k \) and \( n \) are the number of free parameters and the sample size, respectively, and finally, \( I(\theta) \) is the Fisher information matrix of sample size 1 (e.g., Schervish, 1995, p. 111). Note that the model with normally distributed errors with a constant variance, the first term of AIC and BIC, \( -2 \ln f(y|\hat{\theta}) \), can be replaced by \( (n \ln(SSE) - n \ln(n)) \) where SSE defined in Eq. (5) is the minimized sum of squared errors obtained by the method of least squares estimation (Burnham and Anderson, 2010, p. 63). Each of these five methods of model comparison prescribes that the model that minimizes a given criterion should be preferred.

Each of the criteria in the above equation consists of two factors. The first factor involving \( -\ln f(y|\hat{\theta}) \) in the first term represents the lack of fit. The second factor that constitutes the remaining terms is naturally interpreted as model complexity. Model comparison in each criterion is carried out by trading lack-of-fit for complexity. A complex model with many parameters, having a large value in the complexity factor, will not be chosen unless its fit justifies the extra complexity. It is in this sense that the model comparison criteria formalize the Principle of Occam’s Razor, which states, "Entities must not be multiplied beyond necessity." (William of Ockham, c. 1287-1347).

AIC, AICc and BIC: AIC is probably the most commonly used method of model comparison in practice and is developed based on frequentist statistics. Specifically, AIC is derived as an asymptotic (i.e., large sample) approximation of the Kullback-Leibler information divergence or distance between the true data-generating model and the fitted model. As such, the model with the smallest value of AIC is the one that represents the closest approximation to the truth. AICc, which is a variation of AIC, includes a small-sample bias adjustment term, and its use is recommended over AIC when the sample size \( n \) is relatively small with respect to the number of parameters \( k \), specifically, when \( n/k < 40 \) (Burnham and Anderson, 2010, p. 66). BIC originates from Bayesian statistics, is derived as a large sample approximation of Bayesian Model Selection (BMS) described later in this chapter, and as such, the model with the smallest value of this criterion is the one that is considered most likely to have generated observed data.

The number of parameters is the only dimension of complexity that is considered by these three methods. As discussed earlier, functional form can also significantly affect model fit and

\[I_{ij}(\theta) = -\frac{1}{n} E \left( \frac{\partial^2 \ln f(y|\theta)}{\partial \theta_i \partial \theta_j} \right) \quad (i, j = 1, ..., k),\]

where \( E \) denotes the statistical expectation with respect to the probability density function \( f(y|\theta) \). For an example calculation of \( I(\theta) \) for retention models, the reader is directed to Pitt et al. (2002, pp. 488-490).
therefore needs to be taken into account in model comparison. The comparison methods introduced next are sensitive to functional form as well as the number of parameters.

It is worth noting that AIC and AICc, as well as BIC for that matter, are on an interval scale of measurement and thus should be interpreted accordingly. All that matters is the difference in AIC values, not the absolute values. In particular, the AIC differences can be transformed into what are known as the Akaike weights that have probabilistic interpretations (Burnham and Anderson, 2010; Wagenmakers and Farrell, 2004). The Akaike weight for model \( M_i \) among a set of \( m \) models being compared is defined as

\[
w_i(AIC) = \frac{\exp\left(-\frac{1}{2}(AIC_i - AIC_{min})\right)}{\sum_{r=1}^{m} \exp\left(-\frac{1}{2}(AIC_r - AIC_{min})\right)}, \quad (i = 1, 2, ..., m) \tag{8}
\]

where \( AIC_{min} \) is the minimum AIC value among the \( m \) models. The weight \( w_i(AIC) \), as the weight of evidence in favor of model \( M_i \), is interpreted as the probability that \( M_i \) is the one, among the set of the \( m \) candidate models, that minimizes the Kullback-Leibler information distance to the true data-generating model (Burnham and Anderson, 2010, p. 75). For AICc and BIC, one can also calculate the corresponding weights as was done for the Akaike weights in Eq. (8), and the resulting weights are interpreted in the same probabilistic manner (Vandekerckhove et al., 2015).

Can AIC (or AICc) disagree with BIC? This is entirely possible, though not too frequent, given that the two criteria are motivated from different schools of thought in statistics that differ not only in mathematical foundations but also in philosophical foundations. Accordingly, when a disagreement in model selection arises between AIC and BIC, it would be pointless trying to resolve it. Rather, one should either choose one over the other based on his or her statistical standpoint (frequentist or Bayesian).

**Fisher Information Approximation (FIA):** FIA was derived from the principle of Minimum Description Length (MDL; Grünwald, 2007) in algorithmic coding theory in computer science. According to this principle, the goal of model comparison is to choose the model that permits the greatest compression of data in its description. The basic idea behind this approach is that a model, by definition, implies the presence of certain regularities, or equivalently redundancy, in data. That is, the more the data can be compressed using the redundancy extracted with the help of the model, the more we learn about the underlying regularities, and in turn, the better the model generalizes as the extracted regularities will help the model make more accurate predictions for future observations.

As in the three methods of model comparison discussed above, the first term of FIA is a lack of fit measure. The second and third terms together constitute the intrinsic complexity of the model. Importantly, and unique to FIA, the functional form dimension of model complexity is reflected through the Fisher information matrix \( I(\theta) \) in the third term. That is, the Fisher information \( I(\theta = (a, b, \sigma)) \) for the linear model defined as \( y = ax + b + N(0, \sigma^2) \) would be different from that for the power model defined as \( y = ax^b + N(0, \sigma^2) \). Additional examples of functional form complexity can be found in Pitt et al. (2002) in which the influence of functional form on model comparison was demonstrated in three areas of cognitive modeling, namely, psychophysics, information integration, and category learning.

Finding the value of the third, functional-form complexity term can be challenging, though not impossibly difficult. To do so, one would need first to obtain the Fisher information matrix, which is defined as the expectation of the second derivatives of the log-likelihood function with
respect to the parameter vector, and then to integrate the determinant of the resulting matrix over the parameter space. Whereas the Fisher information can usually be obtained in analytic form, the integral must almost always be solved numerically using Monte Carlo methods (e.g., Robert and Casella, 2004). Concrete examples of the calculation of FIA that include the third term for selected models of cognition can be found in these sources (Su et al., 2005; Wu et al., 2010a; Klauer and Kellen, 2011; Kellen and Klauer, 2011; Singmann and Kellen, 2013; Klauer and Kellen, 2015).

Again, regarding the third term of FIA, it is worth noting that this term does not depend upon the sample size $n$, and therefore, as the sample size increases, its relative contribution to model complexity becomes negligible in comparison to that of the second term, which is a logarithmic function of $n$. Consequently, for sufficiently large $n$, FIA is reduced essentially to BIC, i.e., approximately one half of it.

**Normalized Maximum Likelihood (NML):** Like FIA, NML is also motivated from the same MDL principle. The two methods are related to each other such that FIA is derived as an asymptotic approximation to NML (e.g., Barron et al., 1998; Myung et al., 2006), and importantly, NML represents a full solution to a minimax problem of inductive inference (Rissanen, 1996, 2001), as we discuss in what follows.

Specifically, given the model $f(y|\theta)$ under consideration, the minimax problem is defined as finding one probability distribution $g^*(y)$ that minimizes its maximum distance to the best-fitting member, i.e., $f(y|\hat{\theta}(y))$, of the parametric family of the model where the data $y$ are generated from another probability distribution $h(y)$:

$$g^*(y) = \arg \min_g \max_h E_h \left[ \ln \frac{f(y|\hat{\theta}(y))}{g(y)} \right]. \quad (9)$$

In the above equation, $\hat{\theta}(y)$ is the maximum likelihood estimate obtained by MLE, and $g$ and $h$ range over the set of virtually all probability distributions and are not required to belong to the model family under consideration. Note that the distance $E_h[\cdot]$ is measured by the Kullback-Leibler information divergence between two distributions, $f(y|\hat{\theta}(y))$ and $g(y)$ where the expectation is taken with respect to the data generating distribution $h(y)$, but not $f(y|\hat{\theta}(y))$ as it would be done normally.

The solution to the minimax problem (Rissanen, 2001) is obtained as

$$g^*(y) = \frac{f(y|\hat{\theta}(y))}{\int f(z|\hat{\theta}(z)) \, dz}. \quad (10)$$

This optimal distribution assigns a probability number to each data vector $y$ that is proportional to the maximized likelihood value $f(y|\hat{\theta}(y))$ and divided by the normalizing constant, $\int f(z|\hat{\theta}(z)) \, dz$, so that it becomes a proper probability density function satisfying $\int g^*(y) \, dy = 1$. As such, $g^*(y)$ is called the *normalized maximum likelihood* (NML) distribution. Note that the normalizing constant is the sum of maximum likelihood values of all possible data vectors that could potentially be observed in a given experimental setting.

The NML criterion in Eq. (7) is then obtained from Eq. (10) by taking the minus logarithm such that $NML := -\ln g^*(y)$. Accordingly, the lower the NML criterion value, the higher probability the NML distribution assigns to the observed data.
How should we interpret the NML distribution and likewise the NML criterion? First of all, the idea behind the minimax problem in Eq. (9) which both are derived from is that we wish to identify and adopt one probability distribution as a representative of the entire model family. From the way the minimax problem is set up in Eq. (9), it follows that the representative distribution as a solution to the minimax problem is sought as the one that most closely mimics the model’s data-fitting behavior under virtually all practical situations—that is, for all possible data generated by all kinds of models—even including the case in which the data may not come from the model under consideration, i.e., under model misspecification. Further, the solution \( g^*(y) \) is not even required to be a member of the model family. As such, the minimax problem encapsulates a minimalist and pragmatic approach to model comparison. Accordingly and deservingly, we believe that NML, as the solution to the minimax problem, is one of the most complete and robust methods of model comparison the field has to offer.

Now let us examine carefully the normalizing constant of the NML distribution in Eq. (10), which is defined as the sum of all best fits the model can provide collectively for all possible data patterns. The logarithm of this constant corresponds to the complexity penalty term of the NML criterion in Eq. (7). Therefore, a complex model is the one that fits well a wide range of data patterns, regardless of whether they are empirically observable or not. It is in this sense that the normalizing constant captures our intuition about model complexity, that is, “the flexibility inherent in a model that enables it to fit diverse patterns of data” (Myung and Pitt, 1997, p. 80). In short, from the NML standpoint, a model to be favored is the one that provides an excellent fit to the observed data but does poorly otherwise, in accordance with the notion of a “good and persuasive” fit (Roberts and Pashler, 2000, Fig. 1).

Finally, as is the case with FIA, the calculation of the complexity term of the NML criterion can be challenging given that it involves an integration over the data space. Concrete examples of calculating this term can be found in recent articles (e.g., Su et al., 2005; Wu et al., 2010b; Klauer and Kellen, 2011; Kellen and Klauer, 2011; Klauer and Kellen, 2015).

### 4.3.2 Bayesian Methods

Bayesian methods of model comparison were developed as a Bayesian alternative to the frequentist-oriented methods such AIC. The attractions of Bayesian statistics in general are many and include: (1) subjectivity of uncertainty quantification (degree-of-personal-belief interpretation of probability); (2) directness of inference (direct estimation of the probability of an unknown quantity); (3) cumulative nature of inference (combining a prior belief and data using Bayes rule to form a new updated belief, which in turn serves as a prior in the next cycle); and (4) ease of computation (Markov chain Monte Carlo makes it possible to simulate effortlessly any arbitrary posterior distribution). It is then no surprise that we have recently witnessed a dramatic increase in the interest and practice of Bayesian modeling in social and behavioral sciences (Lancaster, 2004; Lynch, 2007; Gill, 2008; Kaplan, 2014; Kruschke, 2014; Lee and Wagenmakers, 2014). This shift in turn naturally create the issue of comparing among Bayesian models of the phenomenon under investigation. In this section, we review two commonly used methods, the Bayes factor (BF) and the deviance information criterion (DIC).

**Bayesian Model Selection (BMS):** BMS is the principal method of model comparison in Bayesian inference. The goal of BMS is to select the one model, among the set of candidate models, that is most likely to have generated observed data. This is achieved by minimizing the
criterion value defined as:

$$BMS = -\ln \int f(y|\theta) p(\theta) d\theta,$$

(11)

where \(f(y|\theta)\) is the likelihood function and \(p(\theta)\) is the (parameter) prior. The integral on the right-hand side of the equation is called the marginal likelihood, denoted by \(p(y) = \int f(y|\theta) p(\theta) d\theta\). That is, BMS is equal to the minus logarithm of the marginal likelihood. The method prescribes that the model with the smallest BMS should be preferred.

The difference in BMS between two models, \(M_1\) and \(M_2\), is related directly to the Bayes factor (BF; Kass and Raftery, 1995). The Bayes factor is defined as the ratio of the marginal likelihood under one model to that under the other model, i.e., \(BF_{12} = p(y|M_1)/p(y|M_2)\). Such as, the following equation shows the relationship between BMS and BF:

$$BMS_2 - BMS_1 = \ln BF_{12}$$

(12)

The last equality above is from Bayes rule, \(p(M_1|y) p(M_2|y) = p(M_1) p(M_2) \times BF_{12}\), under the assumption of equal model priors, i.e., \(p(M_1) = p(M_2)\). It is then straightforward to express a model’s posterior probability in terms of its BMS value as:

$$p(M_i|y) = \frac{e^{-BMS_i}}{\sum_{j=1}^{m} e^{-BMS_j}}, \quad (i = 1, \ldots, m)$$

(13)

for a set of \(m\) models being compared. In short, the smaller the BMS value of a model, the greater the model’s posterior probability. It is in this sense that minimization of BMS amounts to maximization of the posterior model probability.

Now, we make several important observations about BMS.

First, note that the marginal likelihood, \(p(y) = \int f(y|\theta) p(\theta) d\theta\), from which BMS is derived is simply the weighted mean of the likelihood \(f(y|\theta)\) across the parameter space with the prior \(p(\theta)\) as the weight. It is this mean likelihood that allows BMS to avoid overfitting, unlike the maximum likelihood that is a GOF measure and thus is necessarily susceptible to the problem. In other words, BMS is equipped with a built-in complexity penalization to safeguard against overfitting, thereby ensuring good generalizability.

Second and interestingly, the exact form of FIA in Eq. (7) is obtained from an asymptotic expansion of BMS under the Jeffreys prior (Balasubramanian, 1997; 2005; Myung et al., 2006). This “surprising” connection between the two seemingly disparate theoretical frameworks (i.e., algorithmic coding theory of data compression versus Bayesian theory of statistical inference) points to a future and potentially fruitful area of research. Related and as noted earlier, BMS is reduced to one half of BIC for large sample size \(n\) (Raftery, 1993).

Third, as is the case for FIA and NML, BMS can be nontrivial to compute due to its integral expression. The integral is generally not amenable to an analytic solution, and therefore, often must be solved numerically using Monte Carlo techniques.

Finally and importantly, it turns out that the calculation of BMS and so BF is considerably simplified for comparison with nested models. A model is said to be nested with another model if the former is obtained from the latter by fixing the values of one or more parameters of the latter. For example, a model defined \(y = at\) is nested within another model defined as \(y = at + bt^2 + c\) since
the former model is obtained by fixing $b = c = 0$ in the latter model. Specifically and formally, let us consider two nested models, $M_1$ and $M_2$, in which model $M_1$ has a parameter vector $\theta$ and model $M_2$ has an extra parameter vector $\phi$ such that $M_1$ corresponds to $M_2$ with $\phi = \phi_0$ for some fixed constant $\phi_0$. To illustrate, in the example just discussed, the notation translates to $M_1 : y = at$, $M_2 : y = at + bt^2 + c$, $\theta = (a)$, $\phi = (b, c)$, and $\phi_0 = (0, 0)$. In any case, the BF for $M_1$ versus $M_2$ simply becomes the ratio of the posterior to prior density values under $M_2$ at $\phi = \phi_0$:

$$BF_{12} = \frac{p(y|M_1)}{p(y|M_2)} = \frac{p(\phi = \phi_0|y, M_2)}{p(\phi = \phi_0|M_2)}.$$ (14)

The above ratio is known as the Savage-Dickey density ratio (Dickey, 1971; O’Hagan and Forster, 2004, pp. 174-177). This Savage-Dickey method is especially useful in Bayesian hypothesis testing with equality and inequality constraints (e.g., $H_0: \mu = 0$ vs $H_1: \mu \neq 0$; $H_0: \mu = 0$ vs $H_1: \mu > 0$). For in-depth treatments of the topic with concrete example applications in cognitive modeling, the reader is advised to read these excellent sources (Wagenmakers et al., 2010; Wetzels et al., 2010; Lee and Wagenmakers, 2014).

**Deviance Information Criterion (DIC):** DIC (Spiegelhalter et al., 2002; Gelman et al., 2013) is a Bayesian analog of AIC defined as

$$DIC = -2 \ln f(y|\overline{\theta}) + 2p_D.$$ (15)

In the above equation, $f(y|\overline{\theta})$ is the likelihood evaluated at the posterior mean $\overline{\theta}$ (i.e., mean of the posterior distribution, $p(\theta|y)$), and $p_D$ is a model complexity measure called the effective number of parameters:

$$p_D = 2 \ln f(y|\overline{\theta}) - 2 E_{\theta|y} [\ln f(y|\theta)],$$ (16)

where the expectation $E[\cdot]$ in the second term is taken with respect to $p(\theta|y)$.\(^3\)

There are a few things worth mentioning about DIC. First, note the similarity between DIC and AIC in Eq. (7): the former is obtained from the latter by first substituting the maximum likelihood estimate $\hat{\theta}$ for the posterior mean $\overline{\theta}$ and then substituting the number of parameters $k$ for the effective number of parameters $p_D$. Second, DIC is a predictive accuracy measure, the goal of which is to identify a model that achieves best predictions for future observations. Third, the effective number of parameters, $p_D$, takes on a continuous positive value and is sensitive to the number of parameters and also importantly, the functional form. Lastly, the calculation of DIC is routine and straightforward; all that is required are samples drawn from the posterior distribution, which can be done using Markov chain Monte Carlo (MCMC: e.g., Brooks et al., 2011). This is unlike BMS and BF, for which an easy-to-implement and general-purpose computational algorithm has yet to be developed. The latter two properties of DIC, in particular, make the criterion well-suited for its usage in hierarchical Bayesian modeling that has recently become increasingly popular in cognitive modeling (e.g., Rouder and Lu, 2005; Lee, 2011; Lee and Wagenmakers, 2014). Many software packages including BUGS (Spiegelhalter et al., 2003) and JAGS (http://mcmc-jags.sourceforge.net) provide DIC values for hierarchical as well as non-hierarchical models.

Before closing, we should mention two other Bayesian criteria, each of which represents an improvement on DIC. They are the Bayesian Predictive Information Criterion (BPIC; Ando, 2007)\(^3\) DIC can be expressed in another equivalent form as $DIC = D + p_D$, where $D = -2 E_{\theta|y} [\ln f(y|\theta)]$.\(^3\)
and the Watanabe-Akaike Information Criterion (WAIC; Watanabe, 2010). BPIC improves upon DIC in that it does not require, as does DIC, the assumption that the model is correctly specified (i.e., containing the true, data-generating process). On the other hand, WAIC is a Bayesian analog of leave-one-out-cross-validation (LOOCV, discussed below) that is reparametrization-invariant (DIC is not) (Spiegelhalter et al., 2002, p. 612). The Bayesian package Stan (http://mc-stan.org) provides WAIC values.

4.3.3 Direct Estimation Methods

In this third section we introduce methods in which we obtain directly a sample-based estimate of a model’s generalizability, without relying upon an explicit measure of complexity. This direct estimation of generalizability is achieved by simulating procedurally the two steps of data collection and model prediction, separately one at a time. The exact details of how this is done depend upon the specific method chosen.

Cross Validation (CV): This is probably the most popular method of model comparison within the class of direct estimation methods (e.g., Stone, 1974; Browne, 2000). In CV, we first randomly divide the observed data sample into two sub-samples of equal size, calibration ($y_{cal}$) and validation ($y_{val}$). We then use the calibration sample to obtain the best-fitting parameter values of the model by maximum likelihood estimation. These values, denoted by $\hat{\theta}(y_{cal})$ are then applied directly without further parameter tuning to fit the validation sample to obtain the model’s prediction accuracy, which is taken as an estimate of the model’s generalizability.

The specific CV criterion can be expressed using an appropriate fit measure such as the minus log-likelihood or the root mean squared error. In terms of the former, CV is defined as follows:

$$CV_{split-half} = -\ln f(y_{val}|\hat{\theta}(y_{cal})), \quad (17)$$

which is actually an index of the lack of generalizability. Accordingly, the model with the smallest CV value should be chosen as the best generalizing model. This particular method of cross validation is known as the split-half CV. One issue is that the resulting CV value would depend on how the calibration and validation samples are selected. This sampling dependency can be minimized by repeatedly performing half-split CV for a large number of splits, each randomly chosen, and then calculating the average CV value as a model’s generalizability measure.

There is another method of cross-validation called as the leave-one-out-cross-validation (LOOCV) that by construction avoids the sampling dependency problem in split-half CV. Specifically, in LOOCV, each of the $n$ observations in a data set serves as the validation sample, with the remaining $(n-1)$ observations serving as the calibration sample. The standard calibration-validation step is repeated for all observations, exactly $n$ times. The model’s generalizability is then estimated as the average of $n$ minus log-likelihoods as

$$LOOCV = -\frac{1}{n} \sum_{i=1}^{n} \ln f(y_i|\hat{\theta}(y_{\neq i})). \quad (18)$$

In the above equation, $y_{\neq i}$ denotes the calibration sample consisting of $(n-1)$ observations excluding $y_i$, which itself is treated as the validation sample. A schematic diagram of how LOOCV works
is illustrated in the left panel of Figure 5. It is worth noting that LOOCV is related to AIC such that model choice under both criteria is asymptotically equivalent provided that certain regularity conditions are met (Stone, 1977).

Cross validation somehow takes into account the effects of both dimensions of model complexity (the number of parameters and functional form), though how this is accomplished is not clear. It is therefore not possible to get an explicit measure of model complexity. The method is equally applicable to comparing formal statistical models and non-formal models without likelihoods, such as connectionist models and simulation-based models. In short, its ease of implementation and versatility make cross validation a highly attractive and recommendable method of model comparison.

**Generalization Criterion (GC):** This criterion, due to Busemeyer and Wang (2000), has been proposed as a formal implementation of the *strong inference test* (Platt, 1964) and is similar to cross validation, at least in spirit if not in substance. The basic idea of GC is to compare and choose among alternative explanations (models) of the phenomenon of interest based on their “*a priori* predictions (made before observing data) rather than *post hoc* fits (made after observing the data)” (Busemeyer and Wang, 2000, p. 172). Specifically, in GC, the complete experimental design is partitioned into two subdesigns, a calibration design and a generalization design. The first partition is used to estimate a model’s best-fitting parameter values, which in turn without further parameter tuning are used to compute the new predictions for the second, generalization-design partition. The model, among a set of candidate models, that makes the most accurate predictions under an appropriate fit measure is preferred. Ahn et al. (2008) provides an example application of GC in cognitive modeling.
Despite the apparent similarity between the two, the generalization criterion differs in a
important way from cross validation: In GC, models are evaluated in their ability to generalize
to new and importantly, different experimental design or tasks.\footnote{For instance, a decision scientist might be interested in how well the parameter values of a risky choice model estimated based on current observations from a decision-from-description (DFD) task can account for new, future observations from another and different decision-from-experience (DFE) task. In the DFD task, the participant is asked to choose between two fictitious gambles with probabilities of rewards described in an explicit numerical form. On the other hand, in the DFE task, the participant is not given the probability information and instead, must learn the probabilities in an experiential manner, i.e., by observing the outcomes of the chosen gambles over choice trials (e.g., Hertwig and Erev, 2009).} Note that in CV or LOOCV as well as in other methods of model comparison for that matter, the goal is to estimate a model’s
generalizability from one sample data to another sample data, both of which are drawn from the
same experimental task or design setting.

In summary, GC is conceptually intuitive and easy to understand and use. In our view, GC
represents a major step towards extending the current theory of model comparison to a more general
and scientifically relevant goal. Unlike the other methods of model comparison we reviewed in the
present chapter, however, the theoretical foundation of GC is not well understood and established.
For example, it is not entirely clear what the criterion is designed to achieve theoretically. How does
it take into account model complexity so as to avoid overfitting? In what sense is it an “optimal”
method of model comparison? How does it behave asymptotically?

**Accumulative Prediction Error (APE):** This is another direct estimation method in which
a model’s generalizability is estimated in an accumulative fashion under the premise that the data
arrive in a sequentially ordered stream (Dawid, 1984; Wagenmakers et al., 2006).

Specifically, for a given model with $k$ parameters and a data set of $n$ observations, we fit
the model to the first $(k + 1)$ observations as a calibration sample, obtain the maximum likelihood
estimate, and then treat the $(k + 2)$ – $t$h observation as a validation sample of size 1 to estimate
the model’s generalizability measured by its prediction accuracy for the validation sample. In this
first round of calibration-validation split, we have used just the first $(k + 2)$ observations out of the
total of $n$ observations in the data. In the next round, the calibration sample increases in size by
one by taking in the $(k + 2)$ – $t$h observation, and the $(k + 3)$ – $t$h observation now becomes the
validation sample. The accumulative process is repeated until we arrive at the $n$ – $t$h observation
as the validation sample, as illustrated in the right panel of Figure 5.

Formally, the APE criterion is defined as the average of a total of $(n - k - 1)$ individual
generalizability estimates:

$$APE = - \frac{1}{(n - k - 1)} \sum_{i=k+2}^{n} \ln f(y_i|\hat{\theta}(y_1, y_2, ..., y_{i-1})).$$  \hspace{1cm} (19)

The method prescribes that the model with the lowest APE value should be preferred as the best
generalizing model.

Like CV and LOOCV discussed earlier, APE is easy to implement and takes into account,
though implicitly, the effects of both the number of parameters and functional form dimensions
of model complexity, and therefore is highly recommended for all model comparison situations.
Further, APE and BIC are related to each other such that they are asymptotically equivalent
under certain conditions (Dawid, 1984, p. 288).
4.4 Illustrated Example

In this section we illustrate the use of six comparison methods in assessing the same four models of memory retention as in Table 1. They are defined as

\begin{align*}
\text{LIN} & : \quad p(\theta = (a, b), t) = at + b \\
\text{EXP} & : \quad p(\theta = (a, b), t) = ae^{-bt} \\
\text{POW} & : \quad p(\theta = (a, b), t) = a(t + 1)^{-b} \\
\text{EXPOWS} & : \quad p(\theta = (a, b, c, d, e, f), t) = ae^{-bt} + c(t + 1)^{-d} \sin(et) + f
\end{align*}

(20)

The log likelihood to be maximized is given by

\begin{equation}
\ln f(y|\theta) = \sum_{i=1}^{n} \left[ \ln N! - \ln(N - y_i)! - \ln y_i! + y_i \ln p_i(\theta, t_i) + (N - y_i) \ln(1 - p_i(\theta, t_i)) \right]. \tag{21}
\end{equation}

Each model was fitted by maximum likelihood estimation (MLE) to an artificial data set of 21 binomial counts \((n = 21)\) of successes out of 50 Bernoulli trials \((N = 50)\) for the same 21 retention intervals as those used in Table 1.\(^5\) Figure 6 shows the best-fits of the four models.

The model comparison results are summarized in Table 2. As we saw in Table 1, the most complex model (EXPOWS) is favored by goodness-of-fit measures, in this case \(r^2\) and the logarithm of the maximum likelihood (\(\text{LogLik}\)). In the next three rows, the three penalized likelihood methods of model comparison were used to assess the models, all of which take into account the number of parameters (and sample size in the case of BIC). As can be seen, EXPOWS now loses out to EXP because of its four additional parameters, whose added complexity is not justified given the simplicity of the data pattern. The last three rows contain the performance of one Bayesian method (DIC) and two direct estimation methods (LOOCV and APE), all of which should consider both the functional form of the model and the number of parameters. DIC favors EXP, but the other two favor EXPOWS over EXP, suggesting that the additional complexity of this six-parameter model might be warranted given the improved fit that it provides over the data relative to EXP.

The sole purpose of this illustrated example is to demonstrate the use of various methods of model comparison introduced in this chapter. As such, the reader is cautioned not to over-generalize the results in Table 2. The particular performance pattern among the six methods should not be taken as representative of how they will generally behave in other settings.

Given the variety of comparison methods, it is reasonable to wonder when it is appropriate to use each. We end this section by offering some informal guidelines. In the ideal situation in which the models being compared are all nested within one another, one of them is correctly specified, and the sample size is sufficiently large (e.g., 200), all methods of model comparison are likely to perform similarly to one another. On the other hand, when models being compared are non-nested, functional form has the potential to play a significant role in model performance, as illustrated in Tables 1 and 2. In this situation, the methods that consider only the number of parameters (AIC, AICc, BIC) will in general perform worse than the methods that are sensitive to this dimension of model complexity (FIA, NML, BMS, DIC, CV, APE). As a rule of thumb, the latter six may be the safest to use, though there is no guarantee they will always perform the best. In any case, it is important to note that relative performance of any methods of model comparison

\(^5\)The data vector of 21 counts was \(y = (48, 42, 29, 34, 23, 26, 23, 19, 14, 19, 14, 14, 10, 15, 9, 8, 6, 10, 3, 8, 1)\).
can vary considerably depending upon the specific set of models being compared, such as nested vs. non-nested, correctly specified vs. misspecified, sample size, the level of random noise, and the characteristics of the data.

Computational considerations will also influence the choice of method. NML and BMS are likely to be the most general and powerful methods, performing most accurately across a range of conditions. Unfortunately, they are nontrivial to implement and require substantial mathematical sophistication to use. The other methods are easier to implement and are likely to perform satisfactorily under restricted conditions. For example, when models have the same number of parameters but differ in functional form, DIC, CV, and APE are recommended because unlike AIC, AICc or BIC, they are sensitive to the functional form dimension of complexity. If models differ only in number of parameters and the sample size relatively large, then AIC, AICc and BIC should do a good job.

5 Conclusion

In this chapter we have reviewed many model comparison methods. Some, such as AIC and CV, are in wide use across disciplines, whereas others, such as NML, are newer and their adoption is likely to be stymied by the challenges of implementation. That so many different methods exist for
Table 2: Model comparison results for the four retention models in Eq. (20) with artificial data.

<table>
<thead>
<tr>
<th>Model</th>
<th>LIN</th>
<th>EXP</th>
<th>POW</th>
<th>EXPOWS</th>
</tr>
</thead>
<tbody>
<tr>
<td>k</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>(r^2)</td>
<td>0.846</td>
<td>0.937</td>
<td>0.791</td>
<td>0.950</td>
</tr>
<tr>
<td>LogLik</td>
<td>-565.50</td>
<td>-549.83</td>
<td>-575.92</td>
<td>-547.01</td>
</tr>
<tr>
<td>AIC (w)</td>
<td>1,134.99 (0.000)</td>
<td>1,103.66 (0.766)</td>
<td>1,106.03 (0.234)</td>
<td>1,107.98 (0.116)</td>
</tr>
<tr>
<td>AICc (w)</td>
<td>1,135.25 (0.000)</td>
<td>1,103.91 (0.884)</td>
<td>1,107.98 (0.116)</td>
<td>1,117.50 (0.007)</td>
</tr>
<tr>
<td>BIC (w)</td>
<td>1,138.82 (0.000)</td>
<td>1,107.48 (0.993)</td>
<td>1,159.66 (0.000)</td>
<td>1,117.50 (0.007)</td>
</tr>
<tr>
<td>DIC</td>
<td>141.0</td>
<td>109.4</td>
<td>164.0</td>
<td>111.6</td>
</tr>
<tr>
<td>LOOCV</td>
<td>27.29</td>
<td>26.39</td>
<td>27.57</td>
<td>26.05</td>
</tr>
<tr>
<td>APE</td>
<td>29.10</td>
<td>27.00</td>
<td>28.40</td>
<td>25.64</td>
</tr>
</tbody>
</table>

Note: LogLik stands for the log maximum likelihood without the constant, parameter-independent term, i.e., \((\ln N! - \ln(N - y_i)! - \ln y_i!))\). For AIC, AICc, and BIC, their weights of evidence as defined in Eq. (8) are included in parenthesis. The Matlab and R2JAGS programs that generated the above table are included in the appendices.

Comparing models speaks to the ubiquity and importance of the enterprise. Data are our only link to the cognitive processes we investigate. They are thus priceless in advancing our understanding, which includes choosing among alternative explanations (models) of those data. This is particularly true when evaluating a model’s ability to predict the data from a new experiment, one that is a fresh test of model behavior rather than data generated in past experiments (Smolin, 2006).

The focus of this chapter on quantitative and statistical methods of model comparison should not be taken to imply that they are the most important criteria in determining model adequacy or superiority. They are but one type of information that the researcher should use. Qualitative and non-statistical criteria of model evaluation can be just as or even more important, especially in the early stages of model development and during model revision. For example, plausibility (sound assumptions), explanatory adequacy (principled account of the process) and model faithfulness (model behavior truly stems from the theoretical principles it embodies) are foundational criteria that must be satisfied to take a model seriously. Otherwise one possesses a statistical model, or at best a nonsensical cognitive model.

Heavy or exclusive reliance on comparison techniques, however sophisticated, can be ill-advised when one is splitting hairs in choosing among models. When models mimic each other, accounting of the same list of behavioral or neurophysiological data similarly well, efforts should focus on designing experiments that can differentiate the models more clearly, or concede that the models are functionally isomorphic and thus indistinguishable. Of course, it is not easy to design a clever experiment that can decisively differentiate one model from another, but we believe it is ultimately the more productive path to follow. The results from a discriminating experimental design will usually be more persuasive than a large Bayes Factor, for example. As we have noted elsewhere (Navarro et al., 2004), model comparison methods are limited by the informativeness of the data collected in experiments, so anything that can be done to improve data quality should benefit the research enterprise. Readers interested in this topic should consult writings on optimal experimental design (Myung and Pitt, 2009; Myung et al., 2013).

In closing, model comparison methods are but one tool that can be used to guide model selection. They seek to maximize generalizability under the belief that it is the best known way to capture the regularities of a noisy system. Although they vary widely in theoretical orientation,
ease of implementation, and comprehensiveness, they are functionally similar in that they evaluate
the match between the complexity of the data and the corresponding complexity of the models.
The model for which this match is optimal should be preferred.

6 Acknowledgement
The authors wish to thank Hairong Gu for his helpful comments on earlier drafts of the chapter.

7 Appendix A: Matlab Code for Illustrated Example

This appendix includes the Matlab code that generated the simulation results for AIC, AICc, BIC,
LOOCV and APE in Table 2.

```matlab
%%++++++++++++++++++++++++++++++++++++++++++++++++++++++
%% modelcomparison.m (Spring 2016)
%% MATLAB Code for Model Comparison Simulation
%% Author: Jay Myung (Ohio State University), myung.1@osu.edu
%% Distribution: Public & Unlimited
%%++++++++++++++++++++++++++++++++++++++++++++++++++++++

%%%--- Initialization and data -----------------------------
clear;
global m n t x;
opts=optimset('DerivativeCheck','off','Display','off','TolX',1e-7,'TolFun',
                 1e-7,'Diagnostics','off','MaxIter', 500, 'LargeScale','on');

m=21;% number of data points
n=50;% binomial sample size
t=(.5:1.2:24.5)';
x=[48 42 29 34 23 26 23 19 14 19 14 14 10 15 9 8 6 10 3 8 1]';
    % binomial count data
y=x/n;% proportion correct
datasum=zeros(4,7);% model-fit summary
parmsum=zeros(4,6);% parameters summary

%%%--- MLE & GOF -------------------------------------------
nrep=10;% Try multiple initial values to avoid local minima problem
templik=zeros(nrep,3);
tempparm=zeros(nrep,6);
for rep=1:nrep
    [am1,loglik1]=fmincon('linear',rand(2,1),[],[],[],[],[-1 0]',[0 1]',[],opts,t,x);
    [am2,loglik2]=fmincon('expo',rand(2,1),[],[],[],[],zeros(2,1),[1 1]',[],opts,t,x);
    [am3,loglik3]=fmincon('pow',rand(2,1),[],[],[],[],zeros(2,1),[1 1]',[],opts,t,x);
```
templik(rep,:)=[loglik1 loglik2 loglik3];
temparm(rep,:)=[am1' am2' am3'];
end;
[pp,qq]=min(templik);
loglik1=pp(1,1);loglik2=pp(1,2);loglik3=pp(1,3);
am1=temparm(qq(1,1),1:2)';
am2=temparm(qq(1,2),3:4)';
am3=temparm(qq(1,3),5:6)';

templik=zeros(nrep,1);
temparm=zeros(nrep,6);
for rep=1:nrep;
    [am4,loglik4]=fmincon('expsine',rand(6,1),[],[],[],[],[0 0 0 0 0],[1 1 1 1 1]',[],opts,t,x);
    templik(rep,1)=loglik4;
temparm(rep,:)=am4';
end;
[pp,qq]=min(templik);
loglik4=pp;
am4=temparm(qq(1,1),:)';

logml=[-loglik1 -loglik2 -loglik3 -loglik4]';% maximized loglikelihoods

y1=am1(1,1)*t+am1(2,1);
y1=(y1 < ones(m,1)).*y1+(y1 >= ones(m,1))*999999;
y1=(y1 > zeros(m,1)).*y1 + (y1 < zeros(m,1)).*.000001;
y2=am2(1,1)*exp(-am2(2,1)*t);
y2=(y2 < ones(m,1)).*y2+(y2 >= ones(m,1))*999999;
y2=(y2 > zeros(m,1)).*y2 + (y2 < zeros(m,1)).*.000001;
y3=am3(1,1)*(t+1).^(-am3(2,1));
y3=(y3 < ones(m,1)).*y3+(y3 >= ones(m,1))*999999;
y3=(y3 > zeros(m,1)).*y3 + (y3 < zeros(m,1)).*.000001;
y4=am4(1,1)*exp(-am4(2,1)*t)+am4(3,1)*(t+1).^(-am4(4,1)).*sin(am4(5,1)*t)+am4(6,1);
y4=(y4 < ones(m,1)).*y4+(y4 >= ones(m,1))*999999;
y4=(y4 > zeros(m,1)).*y4 + (y4 < zeros(m,1)).*.000001;

sst=sum((y-mean(y)).^2);
r2=[1-sum((y1-y).^2)/sst 1-sum((y2-y).^2)/sst 1-sum((y3-y).^2)/sst 1-sum((y4-y).^2)/sst]; % r^2 values

% AIC, AICc & BIC -------------------------------------
aic=[2*loglik1+2*2 2*loglik2+2*2 2*loglik3+2*2 2*loglik4+2*6]';
aicc=[2*loglik1+2*2+2*2*(2+1)/(n-2-1) 2*loglik2+2*2+2*2*(2+1)/(n-2-1) ... 2*loglik3+2*2+2*2*(2+1)/(n-2-1) 2*loglik4+2*6+2*6*(6+1)/(n-6-1)]';
bic=[2*loglik1+2*log(n) 2*loglik2+2*log(n) 2*loglik3+2*log(n) 2*loglik4+6*log(n)]';
datasum(:,1:5)=[r2 logml aic aicc bic];
parmsum(1:3,1:2)=[am1'; am2'; am3'];
parmsum(4,:)=[am4'];

%-- plots

tt=(0.1:.1:25)';
ylin=am1(1,1)*tt+am1(2,1);
yexp=am2(1,1)*exp(-am2(2,1)*tt);
ypow=am3(1,1)*tt+am3(1,1)*tt+am3(2,1);
yexpows=am4(1,1)*exp(-am4(2,1)*tt)+am4(3,1)*(tt+1).*(-am4(4,1)).*sin(am4(5,1)*tt)...+am4(6,1);

clf;
plot(tt,ylin,'k-',tt,yexp,'b--',tt,ypow,'r:',tt,yexpows,'g-.','LineWidth',3);
hold on;
xlim([0 25]);ylim([0 1]);xlabel('Time t', 'FontSize', 20);
ylabel('Probability p','FontSize', 24);
legend('LIN', 'EXP','POW','EXPOWS','Location','NorthEast');
plot(t,y,'ko','MarkerFaceColor','k','MarkerSize',11);

%%--- LOOCV -----------------------------------------------
bm1=am1;bm2=am2;bm3=am3;bm4=am4;
tcv=zeros(20,1);xcv=zeros(20,1);
loocv=zeros(21,4);
for jj=1:21
if jj==1; tcv=t(2:21,:);xcv=x(2:21,:);
elseif jj==21;tcv=t(1:20,:);xcv=x(1:20,:);
else tcv=[t(1:jj-1,:);t(jj+1:21,:)];xcv=[x(1:jj-1,:);x(jj+1:21,:)];
end;

%-- calibration
nrep=10;
templik=zeros(nrep,3);
temparm=zeros(nrep,6);
for rep=1:nrep
[am1,loglik1]=fmincon('linear',rand(2,1),[],[],[],[],[-1 0]',[0 1]',[],...opts,tcv,xcv);
[am2,loglik2]=fmincon('expo',rand(2,1),[],[],[],[],zeros(2,1),[1 1]',[],...opts,tcv,xcv);
[am3,loglik3]=fmincon('pow',rand(2,1),[],[],[],[],zeros(2,1),[1 1]',[],...opts,tcv,xcv);
templik(rep,:)=[loglik1 loglik2 loglik3];

29
temparm(rep,:)=[am1' am2' am3'];
end;
[pp,qq]=min(templik);
loglik1=pp(1,1);loglik2=pp(1,2);loglik3=pp(1,3);
am1=temparm(qq(1,1),1:2)';
am2=temparm(qq(1,2),3:4)';
am3=temparm(qq(1,3),5:6)';

templik=zeros(nrep,1);
temparm=zeros(nrep,6);
for rep=1:nrep;
[am4,loglik4]=fmincon('expsine',rand(6,1),[],[],[],[],[0 0 0 0 0]',

[1 1 1 1 1]',[],opts,t,x);
templik(rep,1)=loglik4;
temparm(rep,:)=am4';
end;
[pp,qq]=min(templik);
loglik4=pp;
am4=temparm(qq(1,1),:)';

%-- validation

loglikcv1=linear(am1,t(jj,1),x(jj,1));
loglikcv2=expo(am2,t(jj,1),x(jj,1));
loglikcv3=pow(am3,t(jj,1),x(jj,1));
loglikcv4=expsine(am4,t(jj,1),x(jj,1));
loocv(jj,:)=[-loglikcv1 -loglikcv2 -loglikcv3 -loglikcv4];
end;% jj

datasum(:,6)=mean(loocv)';

%%--- APE -------------------------------------------------

bm1=am1;bm2=am2;bm3=am3;bm4=am4;
apelin=zeros(18,1);apeexp=zeros(18,1);apepow=zeros(18,1);
for jj=1:18;
tape=t(1:2+jj,:);xape=x(1:2+jj,:);
nrep=10;
templik=zeros(nrep,3);
temparm=zeros(nrep,6);
for rep=1:nrep
[am1,loglik1]=fmincon('linear',rand(2,1),[],[],[],[],[-1 0]',

[0 1]',[],... 
opts,tape,xape);
[am2,loglik2]=fmincon('expo',rand(2,1),[],[],[],[],zeros(2,1),[1 1]',[],...
opts, tape, xape);
[am3, loglik3] = fmincon('pow', rand(2, 1), [], [], [], [], zeros(2, 1), [1 1]', [],...  
  opts, tape, xape);

templik(rep,:) = [loglik1 loglik2 loglik3];
temparm(rep,:) = [am1' am2' am3'];
end;

[pp, qq] = min(templik);
loglik1 = pp(1,1); loglik2 = pp(1,2); loglik3 = pp(1,3);
am1 = temparm(qq(1,1), 1:2)';
am2 = temparm(qq(1,2), 3:4)';
am3 = temparm(qq(1,3), 5:6)';

apelin(jj,1) = -linear(am1, t(jj+3,1), x(jj+3,1));
apexp(jj,1) = -expo(am2, t(jj+3,1), x(jj+3,1));
apexp(jj,1) = -pow(am3, t(jj+3,1), x(jj+3,1));

end;  jj

apeexpsine = zeros(14, 1);
for jj = 1:14;
tape = t(1:6+jj,:); xape = x(1:6+jj,:);

templik = zeros(nrep, 1);
temparm = zeros(nrep, 6);
for rep = 1:nrep;
  [am4, loglik4] = fmincon('expsine', rand(6, 1), [], [], [], [],...
    [0 0 0 0 0 1], [1 1 1 1 1 1]', [], opts, tape, xape);
  templik(rep, 1) = loglik4;
temparm(rep, :) = am4';
end;
[pp, qq] = min(templik);
loglik4 = pp;
am4 = temparm(qq(1,1), :)';

apeexpsine(jj, 1) = -expsine(am4, t(jj+7,1), x(jj+7,1));

end; jj

datasum(:,7) = [mean(apelin) mean(apexp) mean(apexp) mean(apeexpsine)]';

disp('--- Display model comparison results ---------------------');
disp(num2str([am1', ', ' % 10.4f'));
disp(num2str([am2', ', ' % 10.4f'));

datasum;
disp(num2str([am3'], '%10.4f'));
disp(num2str([am4'], '%10.4f'));
disp('--r2 LogML AIC AICc BIC LOOCV APE--');
disp(num2str(datasum, '%10.3f'));

function loglik = linear(a,t,x)
global n
[mc,mr]=size(x);
p=a(1,1)*t+a(2,1);
p=(p < ones(mc,1)).*p+(p >= ones(mc,1))*9999999;
p=(p > zeros(mc,1)).*p+(p < zeros(mc,1))*0.000001;
loglik=(-1)*(x.*log(p)+(n-x).*log(1-p));
loglik=sum(loglik);

function loglik = expo(a,t,x)
global n
[mc,mr]=size(x);
p=a(1,1)*exp(-a(2,1)*t);
p=(p < ones(mc,1)).*p+(p >= ones(mc,1))*9999999;
p=(p > zeros(mc,1)).*p+(p < zeros(mc,1))*0.000001;
loglik=(-1)*(x.*log(p)+(n-x).*log(1-p));
loglik=sum(loglik);

function loglik = pow(a,t,x)
global n
[mc,mr]=size(x);
p=a(1,1)*t+1).^(-a(2,1));
p=(p < ones(mc,1)).*p+(p >= ones(mc,1))*9999999;
p=(p > zeros(mc,1)).*p+(p < zeros(mc,1))*0.000001;
loglik=(-1)*(x.*log(p)+(n-x).*log(1-p));
loglik=sum(loglik);

function loglik = expsine(a,t,x)
global n
[mc,mr]=size(x);
p=a(1,1)*exp(-a(2,1)*t)+a(3,1)*t+1).^(-a(4,1)).*sin(a(5,1)*t+a(6,1));
p=(p < ones(mc,1)).*p+(p >= ones(mc,1))*9999999;
p=(p > zeros(mc,1)).*p+(p < zeros(mc,1))*0.000001;
loglik=(-1)*(x.*log(p)+(n-x).*log(1-p));
loglik=sum(loglik);
Appendix B: R2JAGS Code for Illustrated Example

This appendix includes the R2JAGS code that generated the simulation results for DIC in Table 2.

```r
library(R2jags)

m <- 21 # number of data points
n <- 50 # binomial sample size
x <- c(48,42,29,34,23,19,14,19,14,10,15,9,8,6,10,3,8,1)
  # observed data as binomial counts
# t <- c(.5,.7,2.9,1.5,3,6.5,7.7,8.9,10.1,11.3,12.5,13.7,14.9,16.1,17.3,
# 18.5,19.7,20.9,22.1,23.3,24.5) # retention intervals

data <- list("m","n","x","t")
myinits <- list(list(a=-0.005, b=0.5)) # LIN model
# myinits <- list(list(a=.5, b=0.5)) # EXP & POW models
# myinits <- list(list(a=0.8,b=0.3,c=0.9,d=0.8,e=0.2,f=0.2)) # EXPOWS model
parameters <- c("a","b") # similarly here and below for EXPOWS models

samples <- jags(data, inits=myinits, parameters, model.file="lin.txt",
  n.chains=1, n.iter=30000, n.burnin=10000, n.thin=1, DIC=T)

ia <- samples$BUGSoutput$sims.array[,,"a"]
ib <- samples$BUGSoutput$sims.array[,,"b"]

print(samples$model)
print(samples$BUGSoutput)
# print(density(ia))
print(mean(ia))
print(median(ia))
# print(hist(ia))
# print(density(ib))
print(mean(ib))
print(median(ib))
# print(hist(ib))
# traceplot(samples)
```

--- Initialization and data ------------------------------

```r
library(R2jags)

m <- 21 # number of data points
n <- 50 # binomial sample size
x <- c(48,42,29,34,23,19,14,19,14,10,15,9,8,6,10,3,8,1)
  # observed data as binomial counts
# t <- c(.5,.7,2.9,1.5,3,6.5,7.7,8.9,10.1,11.3,12.5,13.7,14.9,16.1,17.3,
# 18.5,19.7,20.9,22.1,23.3,24.5) # retention intervals

data <- list("m","n","x","t")
myinits <- list(list(a=-0.005, b=0.5)) # LIN model
# myinits <- list(list(a=.5, b=0.5)) # EXP & POW models
# myinits <- list(list(a=0.8,b=0.3,c=0.9,d=0.8,e=0.2,f=0.2)) # EXPOWS model
parameters <- c("a","b") # similarly here and below for EXPOWS models

samples <- jags(data, inits=myinits, parameters, model.file="lin.txt",
  n.chains=1, n.iter=30000, n.burnin=10000, n.thin=1, DIC=T)

ia <- samples$BUGSoutput$sims.array[,,"a"]
ib <- samples$BUGSoutput$sims.array[,,"b"]

print(samples$model)
print(samples$BUGSoutput)
# print(density(ia))
print(mean(ia))
print(median(ia))
# print(hist(ia))
# print(density(ib))
print(mean(ib))
print(median(ib))
# print(hist(ib))
# traceplot(samples)
```
# lin.txt
# JAGS code for LIN model
#
model{
  for (i in 1:m){
    x[i] ~ dbin(theta[i],n)
    theta[i] <- min(1, max(0,a*t[i]+b))
  }
  a ~ dunif(-0.1,0)
  b ~ dbeta(1,1)
}

# expo.txt
# JAGS code for EXPO model
#
model{
  for (i in 1:m){
    x[i] ~ dbin(theta[i],n)
    theta[i] <- a*exp(-b*t[i])
  }
  a ~ dbeta(1,1)
  b ~ dbeta(1,1)
}

# pow.txt
# JAGS code for POW model
#
model{
  for (i in 1:m){
    x[i] ~ dbin(theta[i],n)
    theta[i] <- a*(t[i]+1)^(-b)
  }
  a ~ dbeta(1,1)
  b ~ dbeta(1,1)
}

# expows.txt
# JAGS code for EXPOWS model
#
model{
  for (i in 1:m){
    x[i] ~ dbin(theta[i],n)
    theta[i] <- min(1, max(0, a*exp(-b*t[i])+c*((t[i]+1)^(-d))*sin(e*t[i])+f))
  }
}
a ~ dbeta(1,1)
b ~ dbeta(1,1)
c ~ dbeta(1,1)
d ~ dbeta(1,1)
e ~ dbeta(1,1)
f ~ dbeta(1,1)

References


36


