

Running head: RATIONAL APPROXIMATIONS TO CATEGORY LEARNING

Rational approximations to rational models:

Alternative algorithms for category learning

Adam N. Sanborn

Gatsby Computational Neuroscience Unit, University College London

Thomas L. Griffiths

University of California, Berkeley

Daniel J. Navarro

University of Adelaide

Send Correspondence To:

Adam Sanborn

Gatsby Computational Neuroscience Unit

17 Queen Square

London WC1N 3AR

United Kingdom

+44 07942 551970

asanborn@gatsby.ucl.ac.uk

Abstract

Rational models of cognition typically consider the abstract computational problems posed by the environment, assuming that people are capable of optimally solving those problems. This differs from more traditional formal models of cognition, which focus on the psychological processes responsible for behavior. A basic challenge for rational models is thus explaining how optimal solutions can be approximated by psychological processes. We outline a general strategy for answering this question, namely to explore the psychological plausibility of approximation algorithms developed in computer science and statistics. In particular, we argue that Monte Carlo methods provide a source of “rational process models” that connect optimal solutions to psychological processes. We support this argument through a detailed example, applying this approach to Anderson’s (1990, 1991) Rational Model of Categorization (RMC), which involves a particularly challenging computational problem. Drawing on a connection between the RMC and ideas from nonparametric Bayesian statistics, we propose two alternative algorithms for approximate inference in this model. The algorithms we consider include Gibbs sampling, a procedure appropriate when all stimuli are presented simultaneously, and particle filters, which sequentially approximate the posterior distribution with a small number of samples that are updated as new data become available. Applying these algorithms to several existing datasets shows that a particle filter with a single particle provides a good description of human inferences.

**Rational approximations to rational models:
Alternative algorithms for category learning**

Rational models of cognition aim to explain human thought and behavior as an optimal solution to the computational problems that are posed by our environment (Anderson, 1990; Chater & Oaksford, 1999; Marr, 1982; Oaksford & Chater, 1998). This approach has been used to model several aspects of cognition, including memory (Anderson, 1990; Shiffrin & Steyvers, 1997), reasoning (Oaksford & Chater, 1994), generalization (Shepard, 1987; Tenenbaum & Griffiths, 2001a), and causal induction (Anderson, 1990; Griffiths & Tenenbaum, 2005). However, executing optimal solutions to these problems can be extremely computationally expensive, a point that is commonly raised as an argument against the validity of rational models (e.g., Gigerenzer & Todd, 1999; Tversky & Kahneman, 1974). This establishes a basic challenge for advocates of rational models of cognition: identifying psychologically plausible mechanisms that would allow the human mind to approximate optimal performance.

The question of how rational models of cognition can be approximated by psychologically plausible mechanisms addresses a fundamental issue in cognitive science: bridging levels of analysis. Rational models provide answers to questions posed at Marr's (1982) computational level – questions about the abstract computational problems involved in cognition. This is a different kind of explanation to those provided by other modeling approaches, which tend to operate at the level of algorithms, considering the concrete processes that are assumed to operate in the human mind. Theories developed at these different levels of analysis provide different kinds of explanations for human behavior, with the computational level explaining *why* we do the things we do, and the algorithmic level explaining *how* these things are done. Both levels of analysis contribute to the development of a complete account of human cognition, just as our understanding

of bird flight is informed by knowing both how the shape of wings results from aerodynamics and how those wings are articulated by muscle and bone.

Despite the importance of both the computational and algorithmic level to understanding human cognition, there has been relatively little consideration of how the two levels might be connected. In differentiating these levels of analysis, Marr (1982) clearly stated that they were not independent, with the expectation that results yielded at one level would provide constraints on theories at another. However, accounts of human cognition are typically offered at just one of these levels, offering theories of either the abstract computational problem or the psychological processes involved. Cases where rational and process models can be explicitly connected are rare and noteworthy, such as the equivalence of exemplar and prototype models of categorization to different forms of density estimation (Ashby & Alfonso-Reese, 1995), although recent work has begun to explore how rational models might be converted into process models (e.g., Kruschke, 2006b).

Considering the processes by which human minds might approximate optimal solutions to computational problems thus provides us with an opportunity not just to address a challenge for rational models of cognition, but to consider how one might develop a general strategy for bridging levels of analysis. In this paper, we outline a strategy that is applicable to rational analyses of probabilistic inference tasks. In such tasks, the learner needs to repeatedly update a probability distribution over hypotheses as more information about those hypotheses becomes available. Due to the prevalence of such tasks, our strategy provides tools that can be used to derive rational approximations to a variety of rational models of cognition.

The key idea behind our approach is that efficient implementation of probabilistic inference is not just a problem in cognitive science – it is an issue that arises in computer science and statistics, resulting in a number of general purpose algorithms (for an

introduction and examples, see Bishop, 2006; Hastie, Tibshirani, & Friedman, 2001; Mackay, 2003). These algorithms often provide asymptotic guarantees on the quality of the approximation they provide, meaning that with sufficient resources they can approximate the optimal inference to any desired level of precision. The existence of these algorithms suggests a strategy for bridging levels of analysis: starting with rational models, and then considering efficient approximations to those models as candidates for psychological process models. The models inspired by these algorithms will not be rational models, but instead will be process models that are closely tied to rational models, and typically come with guarantees of good performance as approximations – a kind of “rational process model”.

Our emphasis in this paper will be on one class of approximation algorithms: Monte Carlo algorithms, which approximate a probability distribution with a set of samples from that distribution. Sophisticated Monte Carlo schemes provide methods for sampling from complex probability distributions (Gilks, Richardson, & Spiegelhalter, 1996), and for recursively updating a set of samples from a distribution as more data are obtained (Doucet, Freitas, & Gordon, 2001). These algorithms provide an answer to the question of how learners with finite memory resources might be able to maintain a distribution over a large hypothesis space. We introduce these algorithms in the general case, and then provide a detailed illustration of how these algorithms can be applied to one of the first rational models of cognition: Anderson’s (1990, 1991) rational model of categorization.

Our analysis of Anderson’s rational model of categorization draws on a surprising connection between this model and work on density estimation in nonparametric Bayesian statistics. This connection allows us to identify two new algorithms that can be used in evaluating the predictions of the model. These two algorithms both asymptotically approximate ideal Bayesian inference, and help to separate the predictions that arise from the underlying statistical model from those that are due to the inference algorithm. We

evaluate these algorithms by comparing the results to the full posterior distribution and to human data. The new algorithms better approximate the posterior distribution and fit human data at least as well as the original algorithm proposed by Anderson. In addition, we show that these new algorithms have greater psychological plausibility and provide better fits to data that have proved challenging to Bayesian models. These results illustrate the use of rational process models to explain how people perform probabilistic inference, provide a tool for exploring the relationship between rational models and human performance, and begin to bridge the gap between computational and algorithmic levels of analysis.

The plan of the paper is as follows. In the first part of the paper we describe the general approach. We begin with a discussion of the challenges associated with performing probabilistic inference, followed by a description of various Monte Carlo methods that can be used to address these challenges, leading finally to the development of the rational approximation framework. In the second part of the paper, we apply the rational approximation idea to categorization problems, using Anderson's rational model. We first describe this model, and use its connection to nonparametric statistics to motivate new approximate inference algorithms. We then evaluate the psychological plausibility of these algorithms at both a descriptive level and with comparisons to human performance in several categorization experiments.

The challenges of probabilistic inference

The computational problems that people need to solve are often *inductive problems*, requiring an inference from limited data to underdetermined hypotheses. For example, when learning about a new category of objects, people need to infer the structure of the category from examples of its members. This inference is inherently inductive, since the category structure is not completely specified by the limited set of examples given to the

learner; and because of this, it is not possible to know exactly which structure is correct. That is, the optimal solution to problems of this kind requires the learner to make *probabilistic* inferences, evaluating the plausibility of different hypotheses in light of the information provided by the observed data. In the remainder of this section, we discuss two challenges that a learner attempting to implement the ideal solution faces: reasoning about hypotheses that are composed of large numbers of variables, and repeatedly updating beliefs about a set of hypotheses as more information becomes available over time.

Reasoning about large numbers of variables

One of the most fundamental challenges in performing probabilistic inference concerns the situation when the number of hypotheses is very large. This is typically encountered when each hypothesis corresponds to a statement about a number of different variables. The number of hypotheses then suffers from a combinatoric explosion. For example, many theories of category learning assume that people assign objects to clusters. If so, then each hypothesis is composed of many assignment variables, one per object. Likewise, in causal learning, hypotheses about causal structure can often be expressed in terms of all of the individual causal relationships that make up a given structure, thus requiring multiple variables. Reasoning about hypotheses comprised of large numbers of variables poses a particular challenge, because of the combinatorial nature of the hypothesis space: the number of hypotheses to be considered can increase exponentially in the number of relevant variables. The number of possible clusterings of n objects, for example, is given by the n th Bell number, with the first ten values being 1, 2, 5, 15, 52, 203, 877, 4140, 21147, and 115975. In such cases, brute force enumeration of all hypotheses will be extremely computationally expensive, and scale badly with the number of variables under consideration.

Updating beliefs over time

When making probabilistic inferences, we rarely have all the information we need to definitively evaluate a hypothesis. As a result, when a learner observes a piece of data and uses this to form beliefs, he or she generally remains somewhat uncertain about which hypothesis is really the correct one. When a new piece of information arrives, this distribution needs to be updated to the new beliefs. The consequence is that an ideal learner needs to constantly update a probability distribution over hypotheses as more data are observed.

Updating beliefs over time is computationally challenging because it requires the learner to draw inferences every time new information becomes available. Unless the learner uses methods that allow the efficient updating of his or her beliefs, he or she would be required to perform the entire inference from scratch every time new information arrives. The cost of probabilistic inference is thus multiplied by the number of observations that have to be processed. As one would expect, this becomes particularly expensive with large hypothesis spaces, such as the combinatorial spaces that result from having hypotheses expressed over large numbers of random variables. Making probabilistic inference computationally tractable thus requires developing strategies for efficiently updating a probability distribution over hypotheses as new data are observed.

Algorithms to address the challenges

Some of the challenges of probabilistic inference can be addressed by approximating optimal solutions using algorithms based on the Monte Carlo principle. This principle is one of the most basic ideas in statistical computing: rather than performing computations using a probability distribution, we perform those computations using a set of samples from that distribution. The resulting approximation becomes increasingly accurate as the number of samples grows, and the relative costs of computing time and errors in

approximation can be used to determine how many samples should be generated. This principle forms the foundation of an entire class of approximation algorithms (Motwani & Raghavan, 1996). Monte Carlo methods provide a way to efficiently approximate probabilistic inference. However, generating samples from posterior distributions is typically not straightforward: generating samples from a distribution requires knowing the form that distribution takes, which is a large part of the challenge of probabilistic inference in the first place. Consequently, sophisticated algorithms need to be used in order to generate samples. Here we introduce two such algorithms at an intuitive level: Gibbs sampling and particle filters. A parallel mathematical development of the general algorithms is given in the Appendix and toy examples of these algorithms applied to categorization and a discussion of their psychological plausibility are given later.

Gibbs sampling

Gibbs sampling (Geman & Geman, 1984) is a very commonly used Monte Carlo method for sampling from probability distributions. This algorithm is initialized with a particular set of values for each variable, often with random values. Gibbs sampling works on the principle of sampling a single random variable at each step. One random variable is selected, and the value of this variable is sampled, conditioned on the values of all of the other random variables and the data. The process is repeated for each variable; each is sampled conditioned on the values of all of the other variables and the data. Intuitively, Gibbs sampling corresponds to the process of inspecting one's beliefs about each random variable conditioned on one's beliefs about all of the other random variables, and the data. Reflecting on each variable in turn provides the opportunity for changes to propagate through the set of random variables. A complete run through sampling all of the random variables is an iteration and the algorithm is usually engaged for many iterations.

Though the algorithm will eventually sample from the desired distribution, it starts

at a particular, often random, set of values. The early iterations show the algorithm converging to the desired distribution, but are not yet samples from this distribution. These iterations are known as the *burn-in* and are thrown away. An additional difficulty is that iterations following the burn-in iterations often show strong dependency from one iteration to the next. These iterations are then thinned, which means keeping every n th iteration and discarding the rest. The remaining iterations after burn-in and thinning are used as samples from the desired distribution. This process provides a way to generate samples from probability distributions defined over large numbers of variables without ever having to enumerate the entire hypothesis space, providing a tractable way to perform probabilistic inference in these cases.

Particle filters

A second class of Monte Carlo algorithms, particle filters, are specifically designed to deal with sequential data. Particle filters are underpinned by a simpler algorithm known as importance sampling, which is used in cases in which it is hard to sample from the target distribution, but easy to sample from a related distribution (known as the *proposal* distribution). The basic idea of importance sampling is that we generate samples from the proposal distribution, and then assign those samples weights that correct for the difference from the target distribution. Samples that are more likely under the proposal than the target distribution are assigned lower weights, since they should be over-represented in a set of draws from the proposal distribution, and samples that are more likely under the target than the proposal are assigned higher weights, increasing their influence.

Particle filters extend importance sampling to a sequence of probability distributions, typically making use of the relationship between successive distributions to use samples from one distribution to generate samples from the next (for more details, see Doucet et al., 2001). The particle filter was originally developed for making inferences

about variables in a dynamic environment – the problem of “filtering” is to infer the current state of the world given a sequence of observations. However, it also provides a natural solution to the general problem of updating a probability distribution over time. Each particle is a sample from the posterior distribution on the previous trial, and these samples are updated when new data become available.

Rational approximations to rational models

Monte Carlo algorithms provide efficient schemes for approximating probabilistic inference, and come with the asymptotic guarantee that they can produce an arbitrarily good approximation if sufficient computational resources are available. These algorithms thus seem like good candidates for explaining how human minds could be capable of performing probabilistic inference, bridging the gap between the computational-level analyses typically associated with rational models of cognition and the algorithmic level at which psychological process models are defined. In particular, Gibbs sampling and particle filters provide solutions to the challenges posed by probabilistic inference with large numbers of variables and updating probability distributions over time.

Part of the attraction of the Monte Carlo principle as the basis for developing rational process models is that it reduces probabilistic computations to one operation: generating samples from a probability distribution. The notion that people might be capable of generating samples from internalized probability distributions has previously appeared in psychological process models of decision making (Stewart, Chater, & Brown, 2006), estimation (Fiedler & Juslin, 2006), and prediction (Mozer, Pashler, & Homaei, 2008). Indeed, the foundational premise of the highly successful “sequential sampling” framework (Ratcliff, 1978; P. L. Smith & Ratcliff, 2004; Vickers, 1979) is that choice behavior is fundamentally reliant on people drawing and evaluating samples from probability distributions that in some cases derive from internally stored stimulus

representations (Lee & Cummins, 2004; Ratcliff, 1978; Vandekerckhove, Verheyen, & Tuerlinckx, 2010). Taken together, these models provide support for the idea that the basic ingredients required for Monte Carlo simulation are already part of the psychological toolbox.

Recent research has also identified correspondences between the kind of sophisticated Monte Carlo methods discussed above and psychological process models. Shi, Feldman, and Griffiths (2008) showed that the basic computations involved in importance sampling are identical to those used in exemplar models (also see Shi, Griffiths, Feldman, & Sanborn, in press). Exemplar models assume that people store stimuli in memory, activating them based on their similarity to new stimuli (e.g., Medin & Schaffer, 1978; Nosofsky, 1986). An importance sampler can be implemented by storing hypotheses in memory, and activating them in proportion to the probability of observed data under that hypothesis. Moreover, this interpretation of exemplars as stored hypotheses links exemplar-based learning nicely to previous rational analyses of exemplar-based decisions as a form of sequential analysis (see Navarro, 2007; Nosofsky & Palmeri, 1997). That is, the importance sampling method allows people to efficiently learn and store a posterior distribution, and the sequential analysis method allows efficient decisions to be made on the basis of this stored representation. This thus constitutes a natural, psychologically plausible scheme for approximating some probabilistic computations.

Several recent papers have also examined the possibility that particle filters might be relevant to understanding how people can update probability distributions over time. This idea was first raised by Sanborn, Griffiths, and Navarro (2006), and particle filters have subsequently been used to explain behavioral patterns observed in several tasks. Daw and Courville (2008) argued that a particle filter with a small number of particles could explain rapid transitions seen in associative learning tasks with animals. Brown and Steyvers (2009) used particle filters to explain individual differences in a change-point

detection task, where variation of the number of particles being considered captured one dimension along which participants varied. Finally, Levy, Reali, and Griffiths (2009) showed that garden path effects in sentence processing could be accounted for by using a particle filter for parsing, where the frequency with which the parser produced no valid particles was predictive of the difficulty that people had interpreting the sentence.

Evaluating Monte Carlo algorithms as candidates for rational process models requires exploring how the predictions of rational models of cognition vary under these different approximation schemes, and examining how well these predictions correspond to human behavior. In the remainder of the paper, we provide a detailed investigation of the performance of different approximation algorithms for Anderson's (1990; 1991) rational model of categorization. This model is a good candidate for such an investigation, since it involves an extremely challenging computational problem: evaluating a posterior distribution over all possible partitions of a set of objects into clusters. This problem is so challenging that Anderson's original presentation of the model resorted to a heuristic solution. We use a connection between this rational model and a model that is widely used in Bayesian statistics to specify a Gibbs sampler and particle filter for this model, which we evaluate against a range of empirical data.

The Rational Model of Categorization

The problem of category learning is to infer the structure of categories from a set of stimuli labeled as belonging to those categories. The knowledge acquired through this process can ultimately be used to make decisions about how to categorize new stimuli. Several rational analyses of category learning have been proposed (Anderson, 1990; Ashby & Alfonso-Reese, 1995; Nosofsky, 1998). These analyses essentially agree on the nature of the computational problem involved, casting category learning as a problem of *density estimation*: determining the probability distributions associated with different category

labels. Viewing category learning in this way helps to clarify the assumptions behind the two main classes of psychological models: exemplar models and prototype models.

Exemplar models assume that a category is represented by a set of stored exemplars, and categorizing new stimuli involves comparing these stimuli to the set of exemplars in each category (e.g., Medin & Schaffer, 1978; Nosofsky, 1986). Prototype models assume that a category is associated with a single prototype and categorization involves comparing new stimuli to these prototypes (e.g., Reed, 1972). These approaches to category learning correspond to different strategies for density estimation used in statistics, being nonparametric and parametric density estimation respectively (Ashby & Alfonso-Reese, 1995).

Anderson's (1990, 1991) rational analysis of categorization takes a third approach, modeling category learning as Bayesian density estimation. This approach encompasses both prototype and exemplar representations, automatically selecting the number of clusters to be used in representing a set of objects. Unfortunately, the inference for this model is extremely complex, requiring an evaluation of every possible way of partitioning exemplars into clusters, with the number of possible partitions growing exponentially with the number of exemplars. Anderson proposed an approximation algorithm in which stimuli are sequentially assigned to clusters, and assignments of stimuli are fixed once they are made. However, this algorithm does not provide any asymptotic guarantees for the quality of the resulting assignments, and is extremely sensitive to the order in which stimuli are observed, a property which is not intrinsic to the underlying statistical model. As a result, evaluations of the model are tied to the particular approximation algorithm that was used.

Before we consider alternative approximation algorithms for Anderson's model, we need to provide a detailed specification of the model and the original algorithm. In this section, we first outline the Bayesian view of categorization, showing how exemplar and prototype models are special cases of the approach, and then describe the specific

approach taken by Anderson.

Bayesian categorization models

Rational models of categorization must solve the density estimation problem outlined above and use this estimate to identify the category label or some other unobserved property of an object using its observed properties (Anderson, 1990; Ashby & Alfonso-Reese, 1995; Rosseel, 2002). This prediction problem has a natural interpretation as a form of Bayesian inference, which we now outline. Suppose that the learner has previously been shown a set of two stimuli and their labels, where the two stimuli are the first two stimuli in Figure 1. We let y_i refer to the category label given to the i th object in this list (often a nonsense syllable such as “DAX”), and the mental representation of the object is assumed to be characterized by a collection of features, denoted x_i . So for instance if the stimulus is the first stimulus, it could be simply described in terms of features such as “is circular”, “is black”, and “is large”. Thus, if the learner is told “the first stimulus is a DAX”, we would describe the trial by the pair (x_i, y_i) . Across the set of two labelled objects, the information available to the learner can be thought of as a collection of statements (e.g., “the first stimulus is a DAX” and “the second stimulus is a ZUG”) that can be formally characterized by the collection of stimulus representations $\mathbf{x}_2 = (x_1, x_2)$, along with the labels given to each of these objects $\mathbf{y}_2 = (y_1, y_2)$. More generally we will refer to these already known stimuli as the first $N - 1$ stimuli with representations $\mathbf{x}_{N-1} = (x_1, x_2, \dots, x_{N-1})$, and labels $\mathbf{y}_{N-1} = (y_1, y_2, \dots, y_{N-1})$

With that in mind, the problem facing the learner can be written in the following way: on the N th trial in the experiment, he or she is shown a new stimulus x_N (e.g., the third stimulus in Figure 1), and asked what label it should be given. If there are J possible labels involved in the task, the problem is to determine if the N th object should be given the j th label (i.e., infer that $y_N = j$), on the basis of the information available,

$(x_N, \mathbf{x}_{N-1}, \mathbf{y}_{N-1})$. If we apply Bayes' rule to this problem, we are able to see that

$$P(y_N = j | x_N, \mathbf{x}_{N-1}, \mathbf{y}_{N-1}) = \frac{P(x_N | y_N = j, \mathbf{x}_{N-1}, \mathbf{y}_{N-1}) P(y_N = j | \mathbf{y}_{N-1})}{\sum_{y=1}^J P(x_N | y_N = y, \mathbf{x}_{N-1}, \mathbf{y}_{N-1}) P(y_N = y | \mathbf{y}_{N-1})}. \quad (1)$$

In this expression, $P(x_N | y_N = j, \mathbf{x}_{N-1}, \mathbf{y}_{N-1})$ denotes the estimated probability that an element of the j th category would possess the collection of features x_N observed in the novel object, and $P(y_N = j | \mathbf{y}_{N-1})$ is an estimate of the prior probability that a new object would belong to the j th category. Additionally, we have assumed that the prior probability of an object coming from a particular category is independent of the features of the previous objects. Thus, this expression makes clear that the probability that an object with features x_N should be given the label $y_N = j$ is related both the probability of sampling an object with features x_N from that category, and the prior probability of choosing that category label. Category learning, then, becomes a matter of determining these probabilities – the problem known as density estimation.

One advantage to describing categorization in terms of the density estimation problem is that both exemplar models and prototype models can be described as different methods for determining the probabilities described by Equation 1. Specifically, Ashby and Alfonso-Reese (1995) observed that if the learner uses a simple form of nonparametric density estimation known as kernel density estimation (e.g., Silverman, 1986) in order to compute the probability $P(x_N | y_N = j, \mathbf{x}_{N-1}, \mathbf{y}_{N-1})$, then an exemplar model of categorization is the result. On the other hand, they note that the learner could use a form of parametric density estimation (e.g., Rice, 1995), in which the category distribution is assumed to have some known form, and the learner's goal is to estimate the unknown parameters of that distribution. If the learner uses this approach, then the result is a prototype model, with the centroid being an appropriate estimate for distributions whose parameters characterize their mean. To illustrate the point, Figure 2 shows a prototype model on the left, in which the category distribution is assumed to be normal

distribution centered over the prototype, and an exemplar model on the right, in which a separate normal distribution (the “kernel”) is placed over each exemplar, and the resulting category distribution is a mixture model.

Having cast the problem in these terms, it is clear that exemplar and prototype models are two extremes along a continuum of possible approaches to category representation. As illustrated in the middle panel of Figure 2, the learner might choose to break the category up into several clusters of stimuli, denoted \mathbf{z}_{N-1} , where $z_i = k$ if the i th stimulus is assigned to the k th cluster. Each such cluster is then associated with a simple parametric distribution, and the category distribution as a whole then becomes a mixture model (e.g. Rosseel, 2002; Vanpaemel & Storms, 2008). Expressed in these terms, prototype models map naturally onto the idea of a one-cluster representation, and exemplar models arise when there is a separate cluster for each object. In between lies a whole class of intermediate category representations, such as the one shown in the middle of Figure 2. In this case, the learner has divided the five objects into two clusters, and the resulting category distribution is a mixture of two normal distributions.

The appeal of this more general class of category representations is that it allows people to use prototype-like models when called for, and to move to the more flexible exemplar-like models when needed. However, by proposing category representations of this form, we introduce a new problem: for a set of N objects how many clusters K are appropriate to represent the categories, and how should the cluster assignments \mathbf{z}_N be made in light of the available data $(\mathbf{x}_N, \mathbf{y}_N)$? It is to this topic that we now turn.

Statistical model

A partial solution to this problem was given by Anderson (1990), in the form of the Rational Model of Categorization (RMC). The RMC is somewhat different to the various mixture models described in the previous section insofar as it treats the category labels as

being equivalent to unobserved features. As a consequence, the RMC specifies a joint distribution on features and category labels, rather than assuming that the distribution over category labels is estimated separately and then combined with a distribution on features for each category. This distribution is a mixture, with

$$P(\mathbf{x}_N, \mathbf{y}_N) = \sum_{\mathbf{z}_N} P(\mathbf{x}_N, \mathbf{y}_N | \mathbf{z}_N) P(\mathbf{z}_N) \quad (2)$$

where $P(\mathbf{z}_N)$ is a distribution over possible partitions of the N objects into clusters. Importantly, the number of clusters K in the partition \mathbf{z}_N is not assumed to be fixed in advance, but is rather something that the learner infers from the data. The RMC provides an explicit form for this prior distribution, namely

$$P(\mathbf{z}_N) = \frac{(1-c)^K c^{N-K}}{\prod_{i=0}^{N-1} [(1-c) + ci]} \prod_{k=1}^K (M_k - 1)! \quad (3)$$

where c is a parameter called the *coupling probability*, M_k is the number of objects assigned to cluster k , and K is the total number of clusters in \mathbf{z}_N . Although this distribution appears unwieldy, it is in fact the distribution that results from sequentially assigning objects to clusters with probability

$$P(z_i = k | \mathbf{z}_{i-1}) = \begin{cases} \frac{cM_k}{(1-c) + c(i-1)} & \text{if } M_k > 0 \text{ (i.e., } k \text{ is old)} \\ \frac{(1-c)}{(1-c) + c(i-1)} & \text{if } M_k = 0 \text{ (i.e., } k \text{ is new)} \end{cases} \quad (4)$$

where the counts M_k are accumulated over \mathbf{z}_{i-1} . Thus, each object can be assigned to an existing cluster with probability proportional to the number of objects already assigned to that cluster, or to a new cluster with probability determined by c . Since the prior distribution is set up in a way that allows K to grow as more objects are encountered, the RMC allows the learner to infer the number of clusters via the usual process of Bayesian updating.

Intuitively, we can look at the prior as a rich-get-richer scheme: if a cluster already contains many objects, then it has a higher prior probability for new objects. The

coupling probability is the parameter that determines the severity of this scheme. For high values of the coupling parameter, then larger clusters will be favored in the prior, while for low values of the coupling parameter, smaller clusters will be favored. The cluster sizes that actually result depend on the likelihoods as well as the prior.

The local MAP algorithm

When considering richer representations than prototypes and exemplars it is necessary to have a method for learning the appropriate representation from data. Using Equation 2 to make predictions about category labels and features requires summing over all possible partitions \mathbf{z}_N . This sum rapidly becomes intractable for large N , since the number of partitions grows rapidly with the number of stimuli according to the Bell number introduced earlier. Consequently, an approximate inference algorithm is needed and Anderson (1990, 1991) developed a simple inference algorithm to solve this problem. We will refer to this algorithm as the *local MAP* algorithm, as it involves assigning each stimulus to the cluster that has the highest posterior probability given the previous assignments (i.e., the maximum a posteriori or MAP cluster). The algorithm is a local implementation of the MAP because it makes an assignment for each new stimulus as it arrives, which does not necessarily result in the global MAP.

The local MAP algorithm approximates the sum in Equation 2 with just a single clustering of the N objects, \mathbf{z}_N . This clustering is selected by assigning each object to a cluster as it is observed. At this point, the features and labels of all stimuli, along with the cluster assignments \mathbf{z}_{i-1} for the previous $i - 1$ stimuli are given. Thus, the posterior probability that stimulus i was generated from cluster k is

$$P(z_i = k | \mathbf{z}_{i-1}, x_i, \mathbf{x}_{i-1}, y_i, \mathbf{y}_{i-1}) \propto \tag{5}$$

$$P(x_i | z_i = k, \mathbf{z}_{i-1}, \mathbf{x}_{i-1}) P(y_i | z_i = k, \mathbf{z}_{i-1}, \mathbf{y}_{i-1}) P(z_i = k | \mathbf{z}_{i-1})$$

where $P(z_i = k | \mathbf{z}_{i-1})$ is given by Equation 4. Under the local MAP algorithm, x_i is

assigned to the cluster k that maximizes Equation 5. Iterating this process results in a single partition of a set of N objects.

To illustrate the local MAP algorithm, we show in Figure 3 how it would be applied it to the simple example of sequentially presented stimuli in Figure 1. Each stimulus is parameterized by three binary features and the likelihood

$P(x_i|z_i = k, \mathbf{z}_{i-1}, \mathbf{x}_{i-1})P(y_i|z_i = k, \mathbf{z}_{i-1}, \mathbf{y}_{i-1})$ is calculated using binomial distributions that are independent for each feature. These binomial likelihoods are parameterized by the probability of the outcome, and need a prior distribution over this probability. The standard prior for binomial likelihoods is the Beta distribution (see the Appendix for details). For the toy example, we used a symmetric Beta prior for the binomial likelihood, with $\beta = 1$. The symmetric Beta distribution with $\beta = 1$ is a simple choice, because it is equivalent to the uniform distribution.

The local MAP algorithm initially assigns the first observed stimulus to its own cluster. When the second stimulus is observed, the algorithm generates each possible partition: either it is assigned to the same cluster as the first stimulus or to a new cluster. The posterior probability of each of these partitions is calculated and the partition with the highest posterior probability is always chosen as the representation. After the third stimulus is observed, the algorithm produces all possible partitions involving the third stimulus, assuming that the clustering for the first two stimuli remains the same. Note that not all possible partitions of the three stimuli are considered, because the algorithm makes an irrevocable choice for the partition of the first two stimuli and the possible partitions on later trials have to be consistent with this choice. The local MAP algorithm will always produce the same final partition for a given sequential order of the stimuli, assuming there are no ties in the posterior probability.

The local MAP algorithm approximates the complete joint distribution using only

this partition. In effect, it assumes that

$$P(\mathbf{x}_N, \mathbf{y}_N) \approx P(\mathbf{x}_N, \mathbf{y}_N | \mathbf{z}_N) \quad (6)$$

where \mathbf{z}_N is produced via the procedure outlined above. The probability that a particular object receives a particular category label would likewise be computed using a single partition.

Summary

The RMC specifies a rational model of categorization, capturing many of the ideas embodied in other models and allowing the representation to be inferred from the data. However, the model is still significantly limited, because the approximate algorithm used for assigning objects to clusters in the RMC can be a poor approximation to the posterior. In particular, this makes it hard to discriminate the predictions that result from the underlying statistical model from those that are a consequence of the algorithm being used. In order to explore alternative approximation algorithms, we now discuss the connections between the RMC and nonparametric Bayesian statistics.

Dirichlet process mixture models

One of the most interesting properties of the RMC is that it has a direct connection to a model used in nonparametric Bayesian statistics (Neal, 1998). The rationale for using nonparametric methods is that real data are not generally sampled from some neat, finite-dimensional family of distributions, so it is best to avoid this assumption at the outset. From a Bayesian perspective, the nonparametric approach requires us to use priors that include as broad a range of densities of possible, thereby allowing us to infer very complex densities if they are warranted by data. The most commonly used method for placing broad priors over probability distributions is the *Dirichlet process* (DP; Ferguson, 1973). The distributions indexed by the Dirichlet process can be expressed as countably

infinite mixtures of point masses (Sethuraman, 1994), making them ideally suited to act as priors in infinite mixture models (Escobar & West, 1995; Rasmussen, 2000). When used in this fashion, the resulting model is referred to as a *Dirichlet process mixture model* (DPMM; Antoniak, 1974; Ferguson, 1983; Neal, 1998).

Although a complete description of the Dirichlet process is beyond the scope of this paper (for more details, see Navarro, Griffiths, Steyvers, & Lee, 2006), what matters for our purposes is that the Dirichlet process implies a distribution over partitions: any two observations in the sample that were generated from the same mixture component may be treated as members of the same cluster, allowing us to specify priors over an unbounded number of clusters. In the case where N observations have been made, the prior probability that a Dirichlet process will partition those observations into the clusters \mathbf{z}_N is

$$P(\mathbf{z}_N) = \frac{\alpha^K}{\prod_{i=0}^{N-1} [\alpha + i]} \prod_{k=1}^K (M_k - 1)! \quad (7)$$

where α is the dispersion parameter of the Dirichlet process. This distribution over partitions can be produced by a simple sequential stochastic process (Blackwell & MacQueen, 1973). If observations are assigned to clusters one after another and the probability that observation $i + 1$ is assigned to cluster k is

$$P(z_i = k | \mathbf{z}_{i-1}) = \begin{cases} \frac{M_k}{i-1+\alpha} & \text{if } M_k > 0 \text{ (i.e., } k \text{ is old)} \\ \frac{\alpha}{i-1+\alpha} & \text{if } M_k = 0 \text{ (i.e., } k \text{ is new)} \end{cases} \quad (8)$$

we obtain Equation 7 for the probability of the resulting partition. This distribution has a number of nice properties, with one of the most important being *exchangeability*: the prior probability of a partition is unaffected by the order in which the observations are received (Aldous, 1985). Intuitively, exchangeability is similar to independence, but slightly weaker.

To make some of these ideas more concrete, Figure 4 presents a visual depiction of the relationship between the partitioning implied by the DP, the distribution over parameters that is sampled from the DP, and the resulting mixture distribution over

stimuli that results in the DPMM. The partitioning implied by the DPMM shows that items are divided into discrete clusters. Each of these clusters is given a parameter drawn from the prior distribution over parameters. A large number of parameter draws are shown in Figure 4b. Each spike is a new parameter value and the height of the bars depends on the number of clusters that use that parameter. Finally, combining the parameter values with a continuous likelihood function, such as a Gaussian distribution, gives the mixture distribution shown in Figure 4c.

It should be apparent from our description of the prior distribution used in the DPMM that it is similar in spirit to the prior distribution underlying the RMC. In fact, the two are directly equivalent, a point that was first made in the statistics literature by Neal (1998). If we let $\alpha = (1 - c)/c$, Equations 3 and 7 are equivalent, as are Equations 4 and 8. Thus the prior over cluster assignments used in the RMC is exactly the same as that used in the DPMM. Anderson (1990, 1991) thus independently discovered one of the most celebrated models in nonparametric Bayesian statistics, deriving this distribution from first principles.

Alternative approximate inference algorithms

The connection between the RMC and the DPMM suggests a solution to the shortcomings of the local MAP algorithm. In this section, we draw on the extensive literature on approximate inference for DPMMs to offer two alternative algorithms for the RMC: Gibbs sampling and particle filtering. These algorithms are less sensitive to order and are asymptotically guaranteed to produce accurate predictions.

As discussed above, both Gibbs sampling and particle filters are Monte Carlo methods. This means that they provide ways of approximating the intractable sum over partitions numerically using a collection of samples. Specifically, to compute the probability that a particular object receives a particular category label, a Monte Carlo

approximation gives

$$\begin{aligned} P(y_N = j | \mathbf{x}_N, \mathbf{y}_{N-1}) &= \sum_{\mathbf{z}_N} P(y_N = j | \mathbf{x}_N, \mathbf{y}_{N-1}, \mathbf{z}_N) P(\mathbf{z}_N | \mathbf{x}_N, \mathbf{y}_{N-1}) \\ &\approx \frac{1}{M} \sum_{\ell=1}^M P(y_N = j | \mathbf{x}_N, \mathbf{y}_{N-1}, \mathbf{z}_N^{(\ell)}) \end{aligned} \quad (9)$$

where $\mathbf{z}_N^{(1)}, \dots, \mathbf{z}_N^{(M)}$ are M samples from $P(\mathbf{z}_N | \mathbf{x}_N, \mathbf{y}_{N-1})$, and the approximation becomes exact as $M \rightarrow \infty$. The two algorithms differ only in how these samples are generated.

Gibbs sampling

Gibbs sampling is the approximate inference algorithm most commonly used with the DPMM (e.g., Escobar & West, 1995; Neal, 1998). It provides a way to construct a Markov chain that converges to the posterior distribution over partitions. The state space of the Markov chain is the set of partitions, and transitions between states are produced by sampling the cluster assignment of each stimulus from its conditional distribution, given the current assignments of all other stimuli. The clustering evolves by sequentially sampling each z_i from the distribution

$$\begin{aligned} P(z_i = k | \mathbf{z}_{-i}, x_i, \mathbf{x}_{-i}, y_i, \mathbf{y}_{-i}) &\propto \\ P(x_i | z_i = k, \mathbf{z}_{-i}, \mathbf{x}_{-i}) &P(y_i | z_i = k, \mathbf{z}_{-i}, \mathbf{y}_{-i}) P(z_i = k | \mathbf{z}_{-i}) \end{aligned} \quad (10)$$

where \mathbf{z}_{-i} refers to all cluster assignments except for the i th.

Equation 10 is extremely similar to Equation 5, although it gives the probability of a cluster based on the all of the trials in the entire experiment except for the current trial, instead of just the previous trials. The statistical property of exchangeability, briefly noted above, means that these probabilities are actually computed in exactly the same way: the order of the observations can be rearranged so that any particular observation is considered the last observation. Hence, we can use Equation 8 to compute $P(z_i | \mathbf{z}_{-i})$, with old clusters receiving probability in proportion to their popularity, and a new cluster

being chosen with probability determined by α (or, equivalently, c). The other terms reflect the probability of the features and category label of stimulus i under the partition that results from this choice of z_i , and depends on the nature of the features.

The Gibbs sampling algorithm for the DPMM is straightforward (Neal, 1998), and is illustrated for the simple example in Figure 5. First, an initial assignment of stimuli to clusters is chosen, with a convenient choice being all stimuli assigned to a single cluster. Unlike the local MAP algorithm, Gibbs sampling is not a sequential algorithm; all stimuli must be observed before it can be run. Next, we choose a single stimulus and consider all possible reassignments of that stimulus to clusters, including not making a change in assignments or assigning the stimulus to a new cluster. Equation 10 gives the probability of each partition and one of the partitions is sampled based on its posterior probability, making this algorithm stochastic, unlike the local MAP. The stochastic nature of the algorithm is evident in the example in Figure 5, because the first circled assignment has lower probability than the alternatives. The example shows two iterations of Gibbs sampling, in which each stimulus is cycled through and reassigned. In an actual application the algorithm would go through many iterations, with the output of one iteration providing the input to the next. Since the probability of obtaining a particular partition after each iteration depends only on the partition produced on the previous iteration, this is a Markov chain.

After enough iterations for the Markov chain to converge, we begin to save the partitions it produces. The partition produced on one iteration is not independent of the next, so the results of some iterations are discarded to approximate independence. The partitions generated by the Gibbs sampler can be used in the same way as samples $\mathbf{z}_N^{(\ell)}$ in Equation 9. As with standard Monte Carlo approximations, the quality of the approximation increases as the number of partitions in that collection increases. The Gibbs sampler provides an effective means of constructing the approximation in

Equation 9, and thus of making accurate predictions about the unobserved features of stimuli.

Particle filtering

There are several ways to construct a particle filter for the DPMM. The method we will use is most closely related to the one discussed by Fearnhead (2004). The key idea is to treat each new observation as a new “time step”, with each particle being a partition $\mathbf{z}_i^{(\ell)}$ of the stimuli from the first i trials. Unlike the local MAP algorithm, in which the posterior distribution is approximated with a single partition, the particle filter uses M partitions. Summing over these particles gives us an approximation to the posterior distribution over partitions

$$P(\mathbf{z}_i|\mathbf{x}_i, \mathbf{y}_i) \approx \sum_{\ell=1}^M \frac{1}{M} \delta(\mathbf{z}_i, \mathbf{z}_i^{(\ell)}) \quad (11)$$

where $\delta(\mathbf{z}, \mathbf{z}')$ is 1 when $\mathbf{z} = \mathbf{z}'$, and 0 otherwise. If Equation 11 is used as an approximation to the posterior distribution over partitions \mathbf{z}_i after the first i trials, then we can approximate the distribution of \mathbf{z}_{i+1} given the observations $\mathbf{x}_i, \mathbf{y}_i$ in the following manner:

$$\begin{aligned} P(\mathbf{z}_{i+1}|\mathbf{x}_i, \mathbf{y}_i) &= \sum_{\mathbf{z}_i} P(\mathbf{z}_{i+1}|\mathbf{z}_i) P(\mathbf{z}_i|\mathbf{x}_i, \mathbf{y}_i) \\ &\approx \sum_{\mathbf{z}_i} P(\mathbf{z}_{i+1}|\mathbf{z}_i) \sum_{\ell=1}^m \frac{1}{m} \delta(\mathbf{z}_i, \mathbf{z}_i^{(\ell)}) \\ &= \frac{1}{m} \sum_{\ell=1}^m P(\mathbf{z}_{i+1}|\mathbf{z}_i^{(\ell)}) \end{aligned} \quad (12)$$

where $P(\mathbf{z}_{i+1}|\mathbf{z}_i)$ is given by Equation 8. We can then incorporate the information conveyed by the features and label of stimulus $i + 1$, arriving at the approximate posterior probability

$$\begin{aligned} P(\mathbf{z}_{i+1}|\mathbf{x}_{i+1}, \mathbf{y}_{i+1}) &\propto P(x_{i+1}|\mathbf{z}_{i+1}, \mathbf{x}_i) P(y_{i+1}|\mathbf{z}_{i+1}, \mathbf{y}_i) P(\mathbf{z}_{i+1}|\mathbf{x}_i, \mathbf{y}_i) \\ &\approx \frac{1}{m} \sum_{\ell=1}^m P(x_{i+1}|\mathbf{z}_{i+1}, \mathbf{x}_i) P(y_{i+1}|\mathbf{z}_{i+1}, \mathbf{y}_i) P(\mathbf{z}_{i+1}|\mathbf{z}_i^{(\ell)}) \end{aligned} \quad (13)$$

The result is a discrete distribution over all the previous particle assignments and all possible assignments for the current stimulus. Drawing M samples from this distribution provides us with our new set of particles.

The particle filter for the simple example is illustrated in Figure 6. The particle filter for the DPMM is initialized with the first stimulus assigned to the first cluster for all M particles, in this case $M = 2$. On observing each new stimulus, the distribution in Equation 13 is calculated, based on the particles sampled in the last trial. Like the local MAP, the particle filter updates the partition as each new stimulus is observed, and like the local MAP, only new partitions that are consistent with the previous choices made by the algorithm are considered. This consistency can be seen in the potential partitions when the third stimulus is observed in Figure 6: each descendant is consistent with the partition choices made by its ancestor. Intuitively, the psychological processes involved in this approximation are very similar to those involved in the local MAP algorithm. People update their beliefs incrementally, keeping the assignments of old items fixed, and making the assignments of new items conditional on these fixed beliefs. There are two key differences between the local MAP and particle filter algorithms. The first is that the choice of new partitions is stochastic instead of deterministic. The particle filter algorithm samples new partitions based on their posterior probabilities instead of always selecting the partition with the maximum probability. A particle filter with $M = 1$ particles is equivalent to the local MAP algorithm, except that the new partition is sampled instead of deterministically selected. The second difference is that multiple particles means that multiple partitions can be used instead of the single partition passed forward by the local MAP. The M partitions are selected without regard for ancestry, allowing a partition that was selected for the early observations to die out as the descendants of other partitions replace it.

Approximation quality

The differences in quality of the various approximations can be explored by a toy example using sequential observations of the stimuli in Figure 1. We compared the local MAP, a particle filter with $M = 100$ particles, a particle filter with $M = 1$ particle, and Gibbs sampling to the exact posterior. For each algorithm a symmetric Beta prior in which $\beta = 1$ was used for the likelihood (see Appendix for details). The local MAP was run a single time, because its outcome is deterministic on a fixed stimulus order. The particle filters were each replicated 10,000 times, and the Gibbs sampler was run for 101,000 iterations. For the Gibbs sampler, the first 1,000 iterations were discarded and every 10th iteration was taken as a sample, yielding 10,000 samples. The results of this comparison are shown in Figure 7. The local MAP algorithm has selected a single partition as an approximation to the exact posterior. In this example, the partition selected by the local MAP is also the MAP of the exact posterior distribution, but the two will not always be equivalent. By taking the MAP partition as each stimulus arrives, the local MAP can be misled to choose a partition that is not the global MAP, if the initial trials are not representative of the whole run of trials. An example of this can be seen in the experiment by Anderson and Matessa in a later section.

The particle filter with $M = 100$ particles and the Gibbs sampler both produce a posterior distribution that is nearly indistinguishable from the exact posterior. The single-particle particle filter is an interesting intermediate case. Each run of the single-particle particle filter produces a single partition, not the distribution produced by a particle filter with $M > 1$ particles. However, averaging over runs of the single-particle particle filter gives an approximation that is much closer to the exact posterior than the local MAP. Unlike the asymptotic performance of the Gibbs sampler and particle filter with infinite particles, the approximation of the single-particle particle filter is slightly biased, as can be seen in the figure. The bias is much less than the local MAP because the

algorithm is stochastic, but is still present because each run of the $M = 1$ particle filter cannot correct its previous assignments by resampling.

Psychological plausibility of the algorithms

Before turning to a quantitative comparison of the algorithms with human data it is worth considering their psychological plausibility at a descriptive level, to see whether they are appropriate for human cognition. We take as a starting point Anderson's (1990, 1991) two desiderata for an approximate inference algorithm: that it be incremental, and that people see objects as arising from a single cause. These desiderata were based on beliefs about the nature of human category learning. In tasks in which people see objects presented sequentially and must judge which category they arise from "people need to be able to make predictions all the time not just at particular junctures after seeing many objects and much deliberation" (Anderson, 1991, p. 412), and "people tend to perceive objects as coming from specific categories" (Anderson, 1991, p. 411).

In addition to these two desiderata, we are concerned with how these algorithms might introduce new order effects into a model. Often statistical models, such as the DPMM, are invariant to the order in which observations arrive. However, the approximations used in practical applications of these models tend to introduce order effects as a side effect of limited computation. People show effects of the order of presentation of stimuli (Anderson, 1990; Medin & Bettger, 1994; Murdock, 1962), and to have a psychologically plausible algorithm, the cumulative order effects of the model and those introduced by the approximation should match the order effects displayed by people. In the remainder of this section we summarize the psychological plausibility of the local MAP, Gibbs sampling, and particle filters. We relate these algorithms to the properties of incrementalism, a single interpretation of how the data arise, and the order effects introduced by the algorithms, which are summarized in Table 1.

Local MAP

Anderson (1990, 1991) introduced the local MAP algorithm to satisfy his two desiderata for psychological plausibility. The first desideratum is satisfied because the local MAP is updated incrementally. In addition, the second desideratum is satisfied because only a single partition of the stimuli into clusters is available to the algorithm in order to make judgments about new stimuli. However, as a result of the single interpretation and its maximization operation, the local MAP algorithm is extremely sensitive to the order in which stimuli are observed. For example, Anderson and Matessa (reported in Anderson, 1990) showed that the predictions of the local MAP algorithm depended strongly on the order the stimuli were introduced in their clustering experiment. For one type of order, the local MAP always predicted one partition of the stimuli, but for the other order it always predicted a second partition. We will explore how the local MAP can be led down garden paths when we compare the algorithms quantitatively.

Gibbs sampling

Gibbs sampling draws samples from one random variable conditioned on all of the rest and all of the data, thus requiring all of the data be present before inference begins. New data cannot be incrementally added to the sampling scheme, so in order to sample from a posterior distribution when a new piece of data arrives, Gibbs sampling must start from scratch. This property makes the algorithm computationally wasteful if sequential judgments are required. For other tasks, however, Gibbs sampling is more psychologically plausible. In tasks in which all of the data arrive simultaneously, such as when a researcher gives participants a set of objects to sort into groups, participants do not need to make judgments until all of the stimuli are present. Here Gibbs sampling seems psychologically plausible.

Standard implementations of Gibbs sampling do not provide a single interpretation

of the data. The algorithm gathers a set of samples from a probability distribution and all of these samples are used to infer other properties about the data, such as category labels. However, we should note that it would be possible to implement a modified version of the Gibbs sampling algorithm that would provide a fixed interpretation of the data. Instead of keeping all of the iterations, we could create a very forgetful Gibbs sampler that would only recall the current values of the variables when making inferences. Likewise, referring to our third property, Gibbs sampling is asymptotically unbiased, meaning that generating a huge number samples would not introduce any order effects not already present in the statistical model. Again though, the iterations of Gibbs sampling are dependent on one another, so in the forgetful Gibbs sampler we would have iteration to iteration dependence. This iteration to iteration dependence would not be an effect of the order in which the stimuli were presented, but instead an autocorrelation of judgments made by this model.

Particle filters

Particle filters are designed as sequential algorithms that explicitly use incremental updating, which clearly satisfies the first property and makes this algorithm appropriate for modeling sequential judgments. For the second property, the answer depends on the number of particles. Each particle is a sample from the posterior distribution, so a single-particle particle filter will provide a single interpretation of the data. With a multi-particle particle filter, the interpretation becomes probabilistic. The order effects introduced depend on the number of particles, analogous to how the Gibbs sampler's order effects depend on the number of samples. With an infinite number of particles, the particle filter is a very faithful representation of the posterior distribution and thus does not introduce any order effects not present in the statistical model. However, small numbers of particles will introduce order effects and we explore this property in detail later.

Comparing the algorithms

Using the local MAP algorithm, the Rational Model of Categorization (RMC; Anderson, 1991) has successfully predicted human choices in a wide range of experimental paradigms. We introduced two new algorithms for the RMC in the above sections: the Gibbs sampler and the particle filter. We have demonstrated that both of these algorithms provide a closer approximation to the underlying model than the local MAP algorithm and both share some aspects of its psychological plausibility. In this section, we compare the local MAP algorithm, a sequential updating algorithm, against the sequential algorithm we have introduced: the particle filter. Most empirical investigations of human categorization use a sequential trial structure, so we have focused on this comparison. We compare the fits of the multi-particle particle filter, the single-particle particle filter, and the local MAP algorithm to show that the particle filter provides comparable fits to the human data and for some paradigms, the particle filter algorithm actually allows the RMC to better predict human choices.

There are a large number of categorization paradigms on which we could compare the algorithms – we chose to compare the algorithms on several data sets for which the local MAP algorithm performs well, including several cases from Anderson’s (1990; 1991) original evaluation of the model. Testing our algorithm against data on which the local MAP is known to perform well provides a strong test of the particle filter algorithm. We examine the effect of specific instances with binary (Medin & Schaffer, 1978) and continuous parameters (Nosofsky, 1988), and show the algorithms predict a similar correspondence with human data. Next we explore paradigms that have been chosen to highlight differences between the local MAP algorithm and the particle filter. The effects of trial order (Anderson, 1990), how linearly separable and non-separable categories are learned (J. D. Smith & Minda, 1998), and the wider class of learning problems in the Shepard, Hovland, and Jenkins (1961) task (Nosofsky, Gluck, Palmeri, McKinley, &

Glauthier, 1994) are employed to illustrate the advantages of using the particle filter to approximate the RMC.

Effect of specific instances

In a classic paper, Medin and Schaffer (1978) tested whether categorization judgments were influenced by the central tendency of a category alone. In their Experiment 1, the stimuli were designed so as to test whether the nearness of stimuli had an effect above contributing to the category center. The stimuli consisted of six training items, each with five binary features (including the category label, listed last): 11111, 10101, 01011, 00000, 01000, and 10110. In the transfer session, the training items and additional items were rated. The transfer stimuli are presented in Table 2, ordered by human category ratings. These transfer stimuli were structured so that some were closer to specific instances than others, while the distance to the category centers was constant. In this experiment, an effect of specific instances was found in the ratings.

Anderson (1991) ran the local MAP algorithm for several different values of the coupling parameter, but with a fixed prior of $\beta = 1$. The order of the training items was randomized on each block. Low values of the coupling parameter, such as $c = 0.3$, produced high correlations to human ratings ($r = 0.87$). At such values of the coupling parameter, the representation tends to be more exemplar-like than prototype-like, which is consistent with an effect of specific instances. We ran the particle filter algorithm on this experimental design with $M = 100$ and $M = 1$ particles. The particle filter with $M = 1$ particle was replicated 1,000 times and the $M = 100$ particle particle filter was replicated 10 times. The results are shown in Figure 8. Using the same coupling parameter, $c = 0.3$, we found good correlations for the multi-particle particle filter ($r = 0.78$) and for the single-particle particle filter ($r = 0.77$). We also examined lower values of the coupling parameter. For $c = 0.1$ the local MAP algorithm produced nearly the same correlation,

$r = 0.88$, but the single-particle improved somewhat, to $r = 0.84$, as did the particle filter with $M = 100$ particles ($r = 0.84$).

Prediction performance and the range of predicted probabilities both increase if the model is trained with the same number of blocks human subjects were trained (ten) instead of just a single block. Across coupling parameters, the best correlation with human ratings were high for the local MAP ($r = 0.95$), the particle filter with $M = 1$ particles ($r = 0.90$), and the particle filter with $M = 100$ particles ($r = 0.93$). Overall, the results in Figure 8 look accurate for all of the models, except for a serious disagreement between the human data and model predictions for 1110, the seventh stimulus from the left. Human ratings for 1110 diverged from the ratings of 0111 and 1101, the fourth and fifth stimuli from the left. However these three stimuli are the same distances from the training stimuli, so the models tended to give these three stimuli the same probability of Category 1 as a result.

Specific instances with continuous features

The effect of specific instances has been studied with continuous features in Nosofsky (1988). In this study, subjects were trained on 12 stimuli that varied in brightness and saturation. As in Medin and Schaffer (1978), the category structure could not be learned using only one feature. However, in this experiment, the frequency of specific examples was manipulated. Over the course of two experiments, subjects showed a sensitivity to the presentation frequency of specific colors. Anderson (1991) fit the RMC to these data using a likelihood function (following Gelman, Carlin, Stern, & Rubin, 2004) appropriate for continuous data. The continuous likelihood used was a Gaussian distribution for each cluster and the chosen parameters are described in the Appendix. In this simulation, the values of the continuous dimension prior parameters were $\lambda_0 = 1$ and $a_0 = 1$. The label prior parameter was set to $\beta = 1$. Using these parameters, the local MAP algorithm had

an overall correlation between the two experiments of $r = 0.98$ with the human data.

Both the single-particle particle filter and the particle filter with $M = 100$ particles were run with these same parameters. There were 1,000 replications of the single-particle particle filter and 10 repetitions of the $M = 100$ particle particle filter. On each replication, the stimuli were presented in a new random order. The overall correlation between the human data in the two experiments and the average output of the model was $r = 0.97$ for the single-particle particle filter and $r = 0.98$ for $M = 100$ particles. Here again, both types of particle filters perform as well as the local MAP algorithm.

Order effects

Order effects provide a strong challenge to stationary Bayesian models, such as the statistical model underlying the RMC (Kruschke, 2006a, 2006b). A DPMM by nature does not produce order effects, because the observations are exchangeable under the model. However, order effects are easily found in investigations of human cognition, most saliently in the primacy and recency effects found in free recall of a list of words (Murdock, 1962). In categorization research, order effects are well established (Medin & Bettger, 1994). We examine the order effects found including order sensitivity data collected by Anderson and Matessa (reported in Anderson, 1990) to support the approximation used in the RMC.

The rational model is not able to predict these order effects, but approximations to the rational model can. Approximations only assign mass to a small portion of the posterior space over partitions, in effect embodying only a small number of hypotheses about how the stimuli should be clustered. When a new trial is added to the representation, the possible new representations are extensions of the previous representations. So, if a particular partition of the existing stimuli is not present among the particles, then it will never appear when the representation has been updated. In this way, the approximation to the DPMM can be led down a garden path by presenting many

early trials that point toward a particular type of representation¹. If the likelihood of this type of representation is large enough, then the particles will all tend to show that particular representation. Later trials that point toward a different partition of the early trials will not be able to change the partition of the early trials. As a result, early examples can have a greater influence than later trials.

In Anderson and Matessa’s experiment, subjects were presented with a set of 16 stimuli in one of two orders, shown in Table 3. These stimuli were designed to either emphasize the first two features (“front-anchored stimuli”) or the last two features (“end-anchored stimuli”) in the first eight trials. Subjects were trained in one of the two orders. Following the training phase, subjects were shown the full set of stimuli on a sheet of paper and asked to divide the stimuli into two categories of eight stimuli each. Eleven of twenty subjects presented with the front-anchored order split the stimuli into groups along one of the two features emphasized by the front-anchored ordering. Fourteen of twenty subjects presented with the end-anchored order split the stimuli along the features that were emphasized by that ordering. Overall, there was a significant result as twenty-five of forty subjects (62.5%) produced the anticipated order effect.

We compared order effects produced by the range of approximation algorithms to the human data. For all algorithms, $c = 0.5$ and $\beta = 1$, the values used for the local MAP by Anderson and Matessa (Anderson, 1990). The Adjusted Rand Index (Hubert & Arabie, 1985), a standard measure of distance between partitions, was used to find the similarity of the output of the local MAP and particle filter to each of the four partitions that split the stimuli along a single feature. The single-feature-based partition that had the highest Adjusted Rand Index was selected as the partition for that sample. If there was a tie, one of the best was selected with equal probability.

In this experiment the local MAP algorithm predicts that participants will always produce the anticipated ordering effect. We ran the single-particle particle filter for 1,000

repetitions and the $M = 100$ particle particle filter for 10 repetitions in this experimental design to compare it with the local MAP. The single-particle particle filter produces the anticipated order effect on 63% of trials, while the particle filter with $M = 100$ particles produces the order effect only 52% of the time. In this experiment, the particle filter with a single particle is closer to the human results than either the local MAP algorithm or the particle filter with a large number of particles.

Linear separability

The property of linear separability, in which two categories can be perfectly discriminated using a line as a decision bound, has been used in experimental designs to test different types of category representations (Medin & Schwanenflugel, 1981; Nosofsky & Zaki, 2002; J. D. Smith & Minda, 1998). Many models, such as prototype models, inherently predict that linearly separable categories are easier to learn than non-linearly separable categories. In contrast, models such as the RMC do not necessarily predict that linearly separable categories are easier to learn (Anderson, 1991).

An interesting aspect to the study of non-linearly separable categories is exploring how category outliers are learned. The standard design is to select two category centers, with most training stimuli clustered near to the center. A small number of outliers, however, are actually very close to the center of the other category. Examples of these types of structures can be seen in Table 4. In both these designs, Category A consists of binary features mainly set to zero, and Category B consists of binary features mainly set to one. One stimulus in each category is an outlier and is a better match to the stimuli in the other category than to the stimuli in its own category.

Prototype models predicts that these outlier stimuli will always be classified in the incorrect category, while exemplar models can predict that they will be classified fairly accurately. J. D. Smith and Minda (1998) ran a series of experiments that examined the

time course of learning central and outlier members of categories. Initially outlier items were classified as belonging to the incorrect category, but performance improved over blocks of training trials. Figure 9 displays these average results as well as the results of individual subjects. The data of the individual subjects were noisy, so the training blocks are grouped into three bins which are summarized in bar graphs. The outlier stimuli could either both be classified incorrectly (labelled “opposite categories”), both classified in one category or another, or both classified correctly. The decrease in the number of individuals who classify both outliers incorrectly and increase in the number who classify both outliers correctly over blocks mirrors the average results.

J. D. Smith and Minda (1998) proposed that the crossover of the outliers from misclassified to classified correctly seen in the human data was the result of a shift from prototype-like to exemplar-like processing. These results were fit with a mixture of prototypes and exemplars. Later work with a variant of the DPMM showed the crossover could be due to an initial prior for simple representations that is eventually overwhelmed by the data (Griffiths, Canini, Sanborn, & Navarro, 2007). An alternative explanation was proposed by Nosofsky and Zaki (2002), who explained the crossover as a transition from focused attention to a single dimension to more equal weights across all dimensions. In Experiment 2, Nosofsky and Zaki (2002) demonstrated that the exemplar model constrained to attend to a few dimensions did not fit the transfer data after a few blocks significantly worse than the full exemplar model. These additional data provide an interesting counterpoint to the representational change explanation, but we are unable to address them because of the computational complexity in fitting the RMC with any approximation algorithm if all the weights can vary independently. Here we focus on what algorithms allow the RMC to predict a human-like crossover effect.

The RMC using the local MAP and single-particle particle filter algorithms were fit to both the non-linearly separable and linearly separable conditions in the first three

experiments of J. D. Smith and Minda (1998). To fit the models, a grid search was performed over model parameters, using values of 0.01, 0.1, 0.5, and 1 for the β prior parameters. Independent β prior parameters were used for the physical dimensions, β_p , and for the label, β_l . The coupling parameter was varied using the values 0.1, 0.3, 0.5, 0.7, and 0.9. Each simulation was repeated 1,000 times with the stimuli re-randomized within block on each simulation, which was the same randomization scheme used for the human participants.

For all eighty settings of the parameters, the combined likelihoods over all conditions and experiments was compared. The single-particle particle filter produced a higher likelihood than the local MAP algorithm did for each of the eighty settings. To better understand how well the two approximation algorithms fit the outlier stimuli, we re-calculated the likelihoods for each parameter setting using only the outlier stimuli. For these stimuli, the single-particle particle filter produced a better fit to the data on seventy-six of the eighty parameter settings. The best-fitting parameters for the local MAP were $\beta_p = 0.1$ for the physical dimensions, $\beta_l = 1$ for the label dimension, and $c = 0.7$ for the coupling parameter. For the $M = 1$ particle filter, the best fitting parameters were $\beta_p = 1$, $\beta_l = 0.5$, and $c = 0.5$. The maximum likelihood fits for the local MAP and single-particle particle filter are shown in Figure 9. The local MAP algorithm produces a cross-over of the average of the outliers: going from both mis-classified to both classified correctly over blocks, at least for Experiments 1 and 2. However, the results of the individual runs show that the local MAP does not produce crossovers on individual runs of the algorithm. Instead, examination of the bar plots of individual runs show that the local MAP crossover is an artifact of averaging. Unlike the local MAP, the single-particle particle filter produces both average crossovers and individual crossovers, as seen in the changing bar plots of individual runs.

The intuitive reason the local MAP algorithm does not produce human-like

crossovers for individual runs is because it becomes stuck in a pattern based on the initial ordering of the stimuli. To illustrate this idea, we will make the simplifying assumption that each of the central items of Category A are assigned to one cluster and all of the central items of Category B are assigned to a second cluster. The logical possibilities for an outlier is that it is assigned to the correct cluster, assigned to the incorrect cluster, or assigned to its own cluster. Whatever cluster it is initially assigned to, which depends on the parameter settings and the order of the stimuli, it will likely be assigned to the same cluster in later blocks. The repetition occurs because the cluster the outlier was assigned to initially had the highest probability of generating that stimulus, and on subsequent blocks this cluster will contain a copy of the outlier, which increases the likelihood of assignment to this cluster. The local MAP algorithm always assigned stimuli to the maximum likelihood cluster, so that the initial assignment of the outlier is almost perfectly predictive of its later assignment. In fact, examining samples of 100 runs of the local MAP using the best parameters on each experiment's non-linearly separable condition, the initial assignment was perfectly predictive of all later assignments.

In contrast, the stochastic assignment of the single-particle particle filter allows for individual runs of the RMC to display crossovers. Unlike the local MAP, the particle filter allows an outlier to be assigned to a less-probable cluster, depending on the relative probability of the new cluster. One way in which sampling can cause crossing-over is if an outlier is initially assigned to a cluster containing the central stimuli from the other category. This outlier will initially be categorized incorrectly. But in later blocks, the outlier has the possibility of being assigned to a new cluster that contains only that outlier. Once the outlier is assigned to a new cluster, the outlier in later blocks tends to be assigned to the same cluster, because the new cluster contains only the outlier and thus is a very good likelihood match. Prediction of the outlier's category label will become more accurate, because the cluster containing only the outlier will have a stronger influence over

blocks and it predicts the correct category label. As the assignments are stochastic, the block on which the crossover occurs will vary over runs of the algorithm. The prediction of individual crossovers at variable blocks in training matches the human data.

The prediction of the single-particle particle filter stands in contrast with the prediction of a particle filter with a very large number of particles. Each block contains a random ordering of all of the training stimuli, so as the number of particles becomes very large, the distribution over partitions on each run of the model after each block will be the same. Unlike the single-particle particle filter, a particle filter with many particles will not be able to predict between-subject variability with the same parameters, which is an interesting consequence of the single-particle particle filter. The number of particles needed to produce the same outcome on each block is actually quite large, as simulations with $M = 1,000$ particles still showed between-run variability, so this may only be a problem for the ideal statistical model.

Learning types of category structures

A wide range of learning problems were examined in the classic experimental design of Shepard et al. (1961). Binary stimuli with three dimensions were divided into all categories of equal size, and six interesting categorization problems emerged. These problems, shown in Figure 10, were numbered by their difficulty, with Type I the easiest and Type VI the hardest. In a later replication and extension of this design, Nosofsky et al. (1994) collected data on the time course of learning for these six problems, shown in Figure 11.

In addition to running the experiment, Nosofsky et al. (1994) fit the RMC using the local MAP algorithm to the data. The best fitting parameters were $\beta_p = 0.488$, $\beta_l = 0.046$, $c = 0.318$, and a response mapping parameter (used as an exponent to scale the responses) of 0.93. This algorithm predicted a sum squared deviation across learning

problems (SSD) of 0.182. Attempting to replicate this result with the local MAP revealed some surprising subtleties of the local MAP algorithm. First, sometimes there are ties between clusters for the cluster with the maximum probability, for which the local MAP algorithm must be adjusted. A straightforward solution is to assign the new stimulus with equal probability to any cluster that shares the maximum probability.

A more troubling discovery is that there are clusters of the stimuli that have only slightly less probability than the cluster with the maximum posterior probability. Using the best parameters of Nosofsky et al. (1994), we found that the maximum ratio of the second-best posterior probability to the maximum posterior probability could be as high as 0.9997. The behavior of the local MAP algorithm should be very different in the case of tied probabilities and not-quite-tied probabilities, but the difference between the two cases can be very subtle and depend on the precision of the numbers used in the simulation. We found this to be the case when using these best-fitting parameters: using the double precision numbers of Matlab (64 bits) and assigning ties equally to the best clusters, the SSD of the local MAP at these parameters rises to 0.32, and the ordering of problem difficulty on the final block is changed.

A grid search of parameters for both the local MAP and particle filter algorithms was done using the same grid as in the linear separability section with 1000 repetitions per algorithm. A new random order for the stimuli was set for each replication, and the randomization scheme was the same within-block randomization scheme as used in Nosofsky et al. (1994). Over the set of all parameters, the single-particle particle filter algorithm fit better than the local MAP algorithm on 58% of parameter settings. In addition, the best fit of the local MAP was a total SSD of 0.31, while the best SSD for the single-particle particle filter was 0.24. The best fitting parameters were $\beta_p = 0.5$, $\beta_l = 0.01$, and $c = 0.3$ for the local MAP and $\beta_p = 0.1$, $\beta_l = 0.1$, and $c = 0.3$ for the particle filter with $M = 1$ particles. These results, shown in Figure 11, demonstrate that

the single-particle particle filter exceeds the performance of the local MAP for the parameters we tested. However, the brittleness of local MAP algorithm in this paradigm means that there are probably very specific parameter sets that may provide a much better match to the human data.

Summary of simulations

We began with experimental paradigms on which the local MAP algorithm performs well (Anderson, 1991), and the simulations we have performed demonstrate that the particle filter algorithm, especially the single-particle particle filter performs as well or better in these categorization paradigms. For the effects of specific instances with binary (Medin & Schaffer, 1978) and continuous data (Nosofsky, 1988), the single-particle particle filter and the multi-particle particle filter performed about as well as the local MAP algorithm. However, in the later simulations the local MAP algorithm was outperformed by the particle filter, especially by the single-particle particle filter. For the order effects of stimuli presentation, the local MAP algorithm predicts order effects that are stronger than those displayed by human subjects. A particle filter with $M = 100$ particles predicted almost no order effects, but for the single-particle particle filter the size of the order effect was similar to the empirical average.

Further advantages of the particle filter were found for newer experiments with categories that differed in linear separability (J. D. Smith & Minda, 1998). The statistical model underlying the RMC predicts that outlier stimuli will initially be categorized incorrectly, but over blocks will eventually be categorized correctly. The local MAP algorithm did not predict the crossover in individual runs with its best-fitting parameters, and imitates it in the average data by averaging over different trial orders. In contrast, the single-particle particle filter predicts both the crossover in average data, as well as individual variability in how quickly the outlier is learned to be classified correctly.

Finally, the local MAP is extremely sensitive to small changes in probability, as demonstrated with the data and model fits of Nosofsky et al. (1994). The absolute fit and even the order of errors of the six problems depended on the precision of the representation and how ties were dealt with. In the particle filter, the clustering of a new example is sampled, providing a much more plausible implementation that is not sensitive to small changes in relative probability.

Discussion

Bridging the gap between *why* human cognition might operate the way it does (as described by rational analysis) and *how* the mind performs the operations required to do so (as per process models) is a fundamental question in cognitive science. This bridge can be built in a number of different ways: by establishing isomorphisms between models framed at these two levels (e.g., Ashby & Alfonso-Reese, 1995; Griffiths et al., 2007; Shi et al., 2008, in press), by describing the rational foundations of process theories (e.g., Gigerenzer & Brighton, 2009; Perfors & Navarro, 2009; Tenenbaum & Griffiths, 2001b) or by building models that are able to interpolate between heuristic processes and rational accounts (e.g., Brown & Steyvers, 2009; Daw & Courville, 2008; Lee & Cummins, 2004; Sanborn et al., 2006). In this paper we have pursued the third option, arguing that the Monte Carlo principle can provide a foundation for an entire class of “rational process models” that are equivalent to rational models when given unlimited processing resources, but give rise to fast, simple heuristics when computational resources are scarce.

Our analysis of the Rational Model of Categorization provides a good example of how this idea can be put to good use. The RMC is an example of a successful Bayesian model of cognition. It provides a reasonable explanation of how objects should be grouped into clusters and the result of this clustering can be used to explain many categorization experiments. As a purely rational analysis, however, the RMC runs into difficulties

because the complexity of the computational problems involved makes inference difficult, and the fact that the underlying statistical model cannot produce order effects.

Approximation algorithms address both issues, by simplifying inferences and inducing order effects. However, the original “local MAP” approximation produces some order effects that could be considered too strong, and unlike people it learns by deterministic assignments rather than probabilistic ones. Using the Monte Carlo principle, however, we are able to derive a particle filtering algorithm that retains the strengths of the local MAP algorithm but fixes its weaknesses. A single-particle particle filter retains the desiderata of Anderson (1991): online updating of the representation plus a single partition of all of the stimuli into clusters. The only difference of the algorithm is that it uses sampling instead of a maximization operation in order to select new partitions.

The change to using sampling produces some important differences. Averaging many runs of the same order with the local MAP approximation produces the same result every time. However, averaging many runs with the same order using sampling produces a much better approximation to the true posterior. Though each run of a single-particle particle filter produces a potentially extreme result, the aggregate of these results resembles the optimal solution. This effect echoes the wisdom of the crowds: the accuracy of the average over individuals can exceed the accuracy of the individuals (Surowiecki, 2004). This effect has also been found for averaging the judgments of a single individual (Vul & Pashler, 2008). In addition, for a task that requires learning categories that are not linearly separable, sampling allows for the model to occasionally assign a repeated item to a new cluster, allowing it reproduce the finding that people initially categorize an outlier stimulus incorrectly, but slowly learn the correct response. The single-particle particle filter shows a real advantage on this task: not only can it produce the same results as many particles at a lower computational cost, it produces realistic-looking individual differences over runs of the model.

Sampling also avoids the necessity of precise representations. It is implausible that people would make deterministic choices based on values that are almost exactly equal, but this is what the local MAP algorithm assumes. Nearly indiscriminable choice probabilities arise in fitting the local MAP algorithm to learning data under a plausible set of parameters. In contrast, the particle filter algorithm samples, so that choices between representations that have nearly equal probability are chosen nearly equally often. This algorithm, or one that interpolates between pure sampling and pure maximization makes for a more psychologically plausible alternative to the local MAP. These results, combined with recent work that has successfully applied particle filters to a range of problems (Brown & Steyvers, 2009; Daw & Courville, 2008; Levy et al., 2009; Yi, Steyvers, & Lee, in press), lead us to believe that particle filters have the potential to be a powerful tool for producing rational process models.

The introduction of these new algorithms also inspires the development of intermediate cases. It seems necessary to limit the precision of the local MAP algorithm in some way to create a psychologically plausible algorithm. One possible way to do this is by casting the local MAP as a sampling algorithm. As each new stimulus is presented, the local MAP algorithm computes the posterior probability, $f(x)$, that the new stimulus belongs to each of the existing clusters and to a new cluster. The local MAP algorithm selects the maximum of $f(x)$, which we can represent by sampling. If we construct a new distribution, $g(x) \propto f(x)^\gamma$, and set $\gamma = \infty$, then sampling from $g(x)$ will be equivalent to taking the maximum value of $f(x)$. We refer to the γ parameter as the distributional scaling parameter². The usefulness of this representation is that we can use values of γ that are less than ∞ . Using smaller values of γ produces a soft-max rule, which greatly changes the behavior of the algorithm when the best two clusters for a new stimulus have nearly the same, but not exactly the same probability. Now, instead of always selecting the highest probability cluster, the adjusted algorithm will select the top two clusters with

nearly equal probability, which is more psychologically plausible. At the other end of the range of the γ parameter, when $\gamma = 1$, this representation is equivalent to a particle filter with $M = 1$ particles, which selects clusters according to their posterior probability.

We should note that our simulations are not particularly constraining on the number of particles that might best be used to fit human participants. The second desideratum for psychological plausibility stated that there should be a single interpretation of which cluster generated an object. This desideratum is debatable, because it may be that people can hold multiple hypotheses of how objects are generated. In simulations we did not present, we looked at a range of approximations that varied both the number of particles and distributional scaling parameter. Our simulations were not particularly constraining for these parameters. For example, a 100 particles with $\gamma = 2$ produces order effects in the Anderson and Matessa experiment that were approximately equal to that produced by the single-particle particle filter. We elected to test the local MAP to the single-particle particle filter in most of the simulations because it provided a clean comparison between maximization and sampling. However, we do not draw the conclusion that a single-particle particle filter is necessarily the way forward. Other work has successfully fit individual subject data by varying the number of particles (Brown & Steyvers, 2009).

More generally, the Monte Carlo principle can be used to motivate other interesting psychological processes. As noted earlier, exemplar-based category learning can be interpreted as a kind of importance sampling (Shi et al., 2008, in press), and the field of decision-making already has many sampling-based theories (Lee & Cummins, 2004; Ratcliff, 1978; Stewart et al., 2006; Vickers, 1979), but other possibilities exist. In the area of problem solving – which, to a large extent is defined in terms of a focus on difficult learning problems – several avenues of work seem promising. For instance, to the extent that incubation effects in problem solving (S. M. Smith & Blankenship, 1989; Wallas, 1926) relate to a loss of fixation of mental set, they could be interpreted as a form of

particle rejuvenation. Similarly, while trial-and-error learning can be quite complex (Anzai & Simon, 1979), it is nevertheless a natural candidate for Markov Chain Monte Carlo explanations. More speculatively, the fact that human problem solving is not invariant to changes in surface form (Kotovsky, Hayes, & Simon, 1985) makes sense given the Monte Carlo principle, insofar as reparameterization of the hypothesis space can make an inference problem harder or easier.

Rational models of cognition provide a way to understand how human behavior can be explained in terms of optimal solutions to problems posed by the environment. The promise of rational process models is that they can link the Platonic world of ideal forms and ideal learners to the less lofty reality of inexact representations and limited resources. By linking these two levels of analysis more closely, we can build models that more completely characterize both the why and the how of human cognition.

References

- Aldous, D. (1985). Exchangeability and related topics. In *École d'été de probabilités de Saint-Flour, XIII—1983* (pp. 1–198). Berlin: Springer.
- Anderson, J. R. (1990). *The adaptive character of thought*. Hillsdale, NJ: Erlbaum.
- Anderson, J. R. (1991). The adaptive nature of human categorization. *Psychological Review*, *98*(3), 409–429.
- Antoniak, C. (1974). Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics*, *2*, 1152–1174.
- Anzai, Y., & Simon, H. A. (1979). The theory of learning by doing. *Psychological Review*, *86*, 124–140.
- Ashby, F. G., & Alfonso-Reese, L. A. (1995). Categorization as probability density estimation. *Journal of Mathematical Psychology*, *39*, 216–233.
- Bishop, C. M. (2006). *Pattern recognition and machine learning*. New York, NY: Springer.
- Blackwell, D., & MacQueen, J. (1973). Ferguson distributions via Polya urn schemes. *The Annals of Statistics*, *1*, 353–355.
- Brown, S. D., & Steyvers, M. (2009). Detecting and predicting changes. *Cognitive Psychology*, *58*, 49–67.
- Chater, N., & Oaksford, M. (1999). Ten years of the rational analysis of cognition. *Trends in Cognitive Science*, *3*, 57–65.
- Daw, N. D., & Courville, A. C. (2008). The pigeon as particle filter. In J. Platt, D. Koller, Y. Singer, & S. Roweis (Eds.), *Advances in neural information processing systems 20* (pp. 369–376). Cambridge, MA: MIT Press.
- Doucet, A., Freitas, N. de, & Gordon, N. (2001). *Sequential Monte Carlo methods in practice*. New York: Springer.
- Escobar, M. D., & West, M. (1995). Bayesian density estimation and inference using

- mixtures. *Journal of the American Statistical Association*, *90*, 577-588.
- Fearnhead, P. (2004). Particle filters for mixture models with an unknown number of components. *Statistics and Computing*, *14*, 11-21.
- Ferguson, T. S. (1973). A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, *1*, 209-230.
- Ferguson, T. S. (1983). Bayesian density estimation by mixtures of normal distributions. In M. Rizvi, J. Rustagi, & D. Siegmund (Eds.), *Recent advances in statistics* (p. 287-302). New York: Academic Press.
- Fiedler, K., & Juslin, P. (Eds.). (2006). *Information sampling and adaptive cognition*. Cambridge: Cambridge University Press.
- Gelman, A., Carlin, J. B., Stern, H. S., & Rubin, D. B. (2004). *Bayesian data analysis*. Boca Raton: Chapman & Hall/CRC.
- Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *6*, 721-741.
- Gigerenzer, G., & Brighton, H. (2009). Homo heuristicus: Why biased minds make better inferences. *Topics in Cognitive Science*, *1*, 107-143.
- Gigerenzer, G., & Todd, P. M. (1999). *Simple heuristics that make us smart*. Oxford: Oxford University Press.
- Gilks, W. R., Richardson, S., & Spiegelhalter, D. J. (Eds.). (1996). *Markov chain Monte Carlo in practice*. Suffolk: Chapman and Hall.
- Gordon, N. J., Salmond, D. J., & Smith, A. F. M. (1993). Novel approach to non-linear/non-Gaussian Bayesian state estimation. *IEE Proceedings-F*, *140*, 107-113.
- Griffiths, T. L., Canini, K. R., Sanborn, A. N., & Navarro, D. (2007). Unifying rational models of categorization via the hierarchical Dirichlet process. In *Proceedings of the*

- 23rd annual meeting of the cognitive science society*. Hillsdale, NJ: Erlbaum.
- Griffiths, T. L., & Tenenbaum, J. B. (2005). Structure and strength in causal induction. *Cognitive Psychology*, *51*, 354-384.
- Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning: Data mining, inference, and prediction*. New York: Springer.
- Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification*, *2*, 193-218.
- Kotovsky, K., Hayes, J. R., & Simon, H. A. (1985). Why are some problems hard? evidence from the Tower of Hanoi. *Cognitive Psychology*, *17*, 248-294.
- Kruschke, J. K. (2006a). Locally Bayesian learning. In *Proceedings of the 28th annual meeting of the cognitive science society*. Hillsdale, NJ: Erlbaum.
- Kruschke, J. K. (2006b). Locally Bayesian learning with applications to retrospective reevaluation and highlighting. *Psychological Review*, *113*, 677-699.
- Lee, M. D., & Cummins, T. D. R. (2004). Evidence accumulation in decision making: Unifying the 'take the best' and 'rational' models. *Psychonomic Bulletin and Review*, *11*, 343-352.
- Levy, R., Reali, F., & Griffiths, T. L. (2009). Modeling the effects of memory on human online sentence processing with particle filters. In D. Koller, D. Schuurmans, Y. Bengio, & L. Bottou (Eds.), *Advances in Neural Information Processing Systems 21* (pp. 937-944).
- Mackay, D. J. C. (2003). *Information theory, inference, and learning algorithms*. Cambridge: Cambridge University Press.
- Marr, D. (1982). *Vision*. San Francisco, CA: W. H. Freeman.
- Medin, D. L., & Bettger, J. G. (1994). Presentation order and recognition of categorically related examples. *Psychonomic Bulletin & Review*, *1*, 250-254.
- Medin, D. L., & Schaffer, M. M. (1978). Context theory of classification learning.

- Psychological Review*, 85, 207-238.
- Medin, D. L., & Schwanenflugel, P. J. (1981). Linear separability in classification learning. *Journal of Experimental Psychology: Human Learning and Memory*, 7, 355-368.
- Motwani, R., & Raghavan, P. (1996). Randomized algorithms. *ACM Computing Surveys*, 28, 33-37.
- Mozer, M., Pashler, H., & Homaei, H. (2008). Optimal predictions in everyday cognition: The wisdom of individuals or crowds? *Cognitive Science*, 32, 1133-1147.
- Murdock, B. B. (1962). The serial position effect of free recall. *Journal of Experimental Psychology*, 64, 482-488.
- Navarro, D. J. (2007). On the interaction between exemplar-based concepts and a response scaling process. *Journal of Mathematical Psychology*, 51, 85-98.
- Navarro, D. J., Griffiths, T. L., Steyvers, M., & Lee, M. D. (2006). Modeling individual differences using Dirichlet processes. *Journal of Mathematical Psychology*, 50, 101-122.
- Neal, R. M. (1993). *Probabilistic inference using Markov Chain Monte Carlo methods* (Tech. Rep. No. CRG-TR-93-1). Department of Computer Science, University of Toronto.
- Neal, R. M. (1998). *Markov chain sampling methods for Dirichlet process mixture models* (Tech. Rep. No. 9815). Department of Statistics, University of Toronto.
- Nosofsky, R. M. (1986). Attention, similarity, and the identification-categorization relationship. *Journal of Experimental Psychology: General*, 115, 39-57.
- Nosofsky, R. M. (1988). Similarity, frequency, and category representations. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 14, 54-65.
- Nosofsky, R. M. (1998). Optimal performance and exemplar models of classification. In M. Oaksford & N. Chater (Eds.), *Rational models of cognition* (p. 218-247). Oxford: Oxford University Press.

- Nosofsky, R. M., Gluck, M., Palmeri, T. J., McKinley, S. C., & Glauthier, P. (1994). Comparing models of rule-based classification learning: A replication and extension of Shepard, Hovland, and Jenkins (1961). *Memory and Cognition*, *22*, 352-369.
- Nosofsky, R. M., & Palmeri, T. J. (1997). An exemplar-based random walk model of speeded classification. *Psychological Review*, *104*, 266-300.
- Nosofsky, R. M., & Zaki, S. R. (2002). Exemplar and prototype models revisited: Response strategies, selective attention, and stimulus generalization. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *28*, 924-940.
- Oaksford, M., & Chater, N. (1994). A rational analysis of the selection task as optimal data selection. *Psychological Review*, *101*, 608-631.
- Oaksford, M., & Chater, N. (Eds.). (1998). *Rational models of cognition*. Oxford: Oxford University Press.
- Perfors, A. F., & Navarro, D. J. (2009). Confirmation bias is rational when hypotheses are sparse. In N. Taatgen, H. van Rijn, L. Schomaker, & J. Nerbonne (Eds.), (p. 2471-2476). Austin, TX: Cognitive Science Society.
- Rasmussen, C. (2000). The infinite Gaussian mixture model. In *Advances in Neural Information Processing Systems 12*. Cambridge, MA: MIT Press.
- Ratcliff, R. (1978). A theory of memory retrieval. *Psychological Review*, *85*, 59-108.
- Reed, S. K. (1972). Pattern recognition and categorization. *Cognitive Psychology*, *3*, 393-407.
- Rice, J. A. (1995). *Mathematical statistics and data analysis* (2nd ed.). Belmont, CA: Duxbury.
- Rosseel, Y. (2002). Mixture models of categorization. *Journal of Mathematical Psychology*, *46*, 178-210.
- Sanborn, A. N., Griffiths, T. L., & Navarro, D. J. (2006). A more rational model of categorization. In *Proceedings of the 28th Annual Conference of the Cognitive*

- Science Society*. Mahwah, NJ: Erlbaum.
- Sethuraman, J. (1994). A constructive definition of Dirichlet priors. *Statistica Sinica*, 4, 639-650.
- Shepard, R. N. (1987). Toward a universal law of generalization for psychological science. *Science*, 237, 1317-1323.
- Shepard, R. N., Hovland, C. I., & Jenkins, H. M. (1961). Learning and memorization of classifications. *Psychological Monographs*, 75. (13, Whole No. 517)
- Shi, L., Feldman, N., & Griffiths, T. L. (2008). Performing Bayesian inference with exemplar models. In *Proceedings of the Thirtieth Annual Conference of the Cognitive Science Society*. Austin, TX: Cognitive Science Society.
- Shi, L., Griffiths, T. L., Feldman, N. H., & Sanborn, A. N. (in press). Exemplar models as a mechanism for performing Bayesian inference. *Psychological Bulletin and Review*.
- Shiffrin, R. M., & Steyvers, M. (1997). A model for recognition memory: REM: Retrieving Effectively from Memory. *Psychonomic Bulletin & Review*, 4, 145-166.
- Silverman, B. W. (1986). *Density estimation*. London: Chapman and Hall.
- Smith, J. D., & Minda, J. P. (1998). Prototypes in the mist: The early epochs of category learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 24, 1411-1436.
- Smith, P. L., & Ratcliff, R. (2004). Psychology and neurobiology of simple decisions. *Trends in Neuroscience*, 27, 161-168.
- Smith, S. M., & Blankenship, S. E. (1989). Incubation effects. *Bulletin of the Psychonomic Society*, 27, 311-314.
- Stewart, N., Chater, N., & Brown, G. D. A. (2006). Decision by sampling. *Cognitive Psychology*, 53, 1-26.
- Surowiecki, J. (2004). *The wisdom of crowds: Why the many are smarter than the few and how collective wisdom shapes business, economics, societies, and nations*.

Doubleday.

- Tenenbaum, J. B., & Griffiths, T. (2001b). The rational basis of representativeness. In J. Moore & K. Stenning (Eds.), *Proceedings of the 23rd annual meeting of the cognitive science society*. Hillsdale, NJ: Erlbaum.
- Tenenbaum, J. B., & Griffiths, T. L. (2001a). Generalization, similarity, and Bayesian inference. *Behavioral and Brain Sciences*, *24*, 629-641.
- Tversky, A., & Kahneman, D. (1974). Judgment under uncertainty: heuristics and biases. *Science*, *185*, 1124-1131.
- Vandekerckhove, J., Verheyen, S., & Tuerlinckx, F. (2010). A crossed random effects diffusion model for speeded semantic categorization decisions. *Acta Psychologica*, *133*, 269-282.
- Vanpaemel, W., & Storms, G. (2008). In search of abstraction: the varying abstraction model of categorization. *Psychonomic Bulletin & Review*, *15*, 732-749.
- Vickers, D. (1979). *Decision processes in visual perception*. London: Academic Press.
- Vul, E., & Pashler, H. (2008). Measuring the crowd within: probabilistic representations within individuals. *Psychological Science*, *19*, 645-647.
- Wallas, G. (1926). *The art of thought*. New York: Harcourt, Brace.
- Yi, S., Steyvers, M., & Lee, M. (in press). Modeling human performance in restless bandits using particle filters. *Journal of Problem Solving*.

Appendix

The Appendix has six parts. The first three sections provide a mathematical introduction to Gibbs sampling, importance sampling, and particle filters. The fourth section develops the Rational Model of Categorization from the standard equations for a mixture model. The fifth and sixth sections respectively present the likelihood equations used in this paper for binary and continuous features.

Gibbs sampling

Gibbs sampling is a part of a broader class of algorithms that exploit the properties of Markov chains. A Markov chain is a sequence of random variables where each variable depends only on that which precedes it in the sequence. For example, a learner might entertain a series of hypotheses, generating the next hypothesis by probabilistically manipulating the previous one in some way. Markov chains have the property that, provided certain simple conditions are satisfied, they converge to a *stationary distribution*: the probability that a variable takes on a particular value approaches this distribution as the length of the sequence increases. Markov chain Monte Carlo algorithms are procedures for constructing Markov chains that have a particular target distribution as their stationary distribution. Simulating the resulting Markov chain then provides a way of generating samples from the target distribution (for details, see Gilks et al., 1996).

Gibbs sampling (Geman & Geman, 1984) is a Markov chain Monte Carlo algorithm for sampling from a distribution that is applicable when hypotheses consist of large numbers of distinct variables. Assume that each hypothesis h consists of assignments of values to a set of N random variables. We can write out the assignments of these values in a vector $\mathbf{z}_N = (z_1, \dots, z_N)$, and express our posterior distribution $P(h|d)$ as a distribution over these vectors $P(\mathbf{z}|d)$. The Gibbs sampler for this distribution is the Markov chain defined by drawing each z_j from the conditional distribution

$P(z_j|z_1, \dots, z_{j-1}, z_{j+1}, \dots, z_N, d)$, either cycling through the z_j in turn or choosing one at random to sample at each iteration. After many cycles, this process will produce a sample from the distribution $P(\mathbf{z}|d)$.

Importance sampling

Importance sampling generates samples from a proposal distribution, and then assign those samples weights that correct for the difference from the target distribution. For example, let us assume that $f(x)$ is a probability density that gives the probability of sample x under the target distribution. If $q(x)$ is the probability of x under the proposal distribution from which x was in fact generated, then the weight is $w(x) \propto f(x)/q(x)$ (for details, see Neal, 1993).

When performing Bayesian inference, the target distribution is almost always the posterior $P(h|d)$, and a standard method is to use the prior distribution $P(h)$ as the proposal. If we take samples of hypotheses h from the prior, then the weight assigned to each sample is proportional to the ratio of the posterior to the prior

$$w(h) \propto \frac{P(h|d)}{P(h)} = \frac{P(d|h)}{\sum_{h \in \mathcal{H}} P(d|h)P(h)} \propto P(d|h) \quad (\text{A-1})$$

where the normalizing constant is simply the sum of the likelihood $P(d|h)$ across all of the samples. As a result, this method is known as *likelihood weighting*, because samples from the prior are simply weighted by the likelihood.

Particle filters

The *particle filter* is a sequential version of importance sampling. It was originally developed for making inferences about variables in a dynamic environment, but it also provides a natural solution to the general problem of updating a probability distribution over time, giving a way to efficiently generate samples from the distribution over hypotheses h_t given data d_1, \dots, d_t .

The basic idea behind the particle filter is that at each point in time, we approximate the posterior $P(h_t|d_1, \dots, d_t)$ using importance sampling. In particular, we can perform importance sampling where we use the “prior” on h_t , $P(h_t|d_1, \dots, d_{t-1})$ as our proposal. If we can generate samples from $P(h_t|d_1, \dots, d_{t-1})$, then we can construct an importance sampler by giving each sample weight proportional to $P(d_t|h_t)$. To decompose this equation further, we can write

$$P(h_t|d_1, \dots, d_{t-1}) = \sum_{h_{t-1}} P(h_t|h_{t-1})P(h_{t-1}|d_1, \dots, d_{t-1}).$$

Thus, samples from $P(h_t|d_1, \dots, d_{t-1})$ can be generated by taking samples from $P(h_{t-1}|d_1, \dots, d_{t-1})$ and then drawing a sample of h_t from $P(h_t|h_{t-1})$ for each sampled value of h_{t-1} . We can write this as

$$P(h_t|d_1, \dots, d_{t-1}) \approx \sum_{\ell} P(h_t|h_{t-1}^{(\ell)})w_{\ell} \quad (\text{A-2})$$

where $h_{t-1}^{(\ell)}$ is the ℓ th sample of h_{t-1} and w_{ℓ} is its associated weight, if it in turn was generated by importance sampling.

The procedure outlined in the previous paragraph identifies an interesting recursion – it gives us a way to sample from $P(h_t|d_1, \dots, d_t)$ as long as we can sample from $P(h_{t-1}|d_1, \dots, d_{t-1})$. This sets us up to introduce the sequential Monte Carlo scheme that underlies the particle filter. First, we generate samples from $P(h_1|d_1)$. Then, for each sample $h_1^{(\ell)}$ we generate $h_2^{(\ell)}$ from $P(h_2|h_1^{(\ell)})$, and assign each resulting sample a weight proportional to $P(d_2|h_2^{(\ell)})$. We repeat this procedure for all t , multiplying each sample $h_t^{(\ell)}$ (now called a “particle”) by $P(d_t|h_t^{(\ell)})$. There is no need to normalize the particle weights at each step – this can be done at the end of the process.

This simple recursive scheme is known as *sequential importance sampling*. However, it has one big problem: over time, the weights of the particles can diverge hugely, since some of the particles are likely to have ended up with a sequence of h_t values that are very unlikely. In some ways, this is a waste of computation, since those particles with very

small weights will make little contribution to later probabilistic calculations. To address this problem, we can use an alternative approach known as *sequential importance resampling*. Under this approach, we regularly sample a new set of particles from a probability distribution corresponding to the normalized weights. This increases the number of particles that correspond to good hypotheses.

While the standard particle filtering algorithm follows this schema, this only scratches the surface of possible sequential Monte Carlo techniques. Even within particle filtering, there are many options that can be used to improve performance on certain problems. For example, in some cases it is possible to enumerate all possible values of h_t given the values of h_{t-1} represented by the current particles, and combine this with the likelihood $P(d_t|h_t)$ to obtain a more accurate proposal distribution. Since the quality of the approximation depends on the match between the proposal and the target, as with other importance sampling methods, improving the proposal distribution directly improves the performance of the particle filter.

The Rational Model of Categorization as a mixture model

The RMC as described in Equation 2 is defined in terms of the joint distribution of \mathbf{x}_N and \mathbf{y}_N , rather than by directly specifying the category distributions as would be the case for the simpler mixture models such as the exemplar model. So it may not be immediately clear how to map it onto the framework of density estimation. However, it is possible to rewrite the RMC in terms of the “standard” Bayesian categorization model (Equation 1), and thereby make the link explicit. To do so, we note that the first term of the numerator of Equation 1 can be rewritten to include assignments of the stimuli to clusters,

$$P(x_N|y_N = j, \mathbf{x}_{N-1}, \mathbf{y}_{N-1}) = \sum_{k=1}^K \sum_{\mathbf{z}_{N-1}} P(x_N|z_N = k, \mathbf{z}_{N-1}, \mathbf{x}_{N-1})P(z_N = k, \mathbf{z}_{N-1}|y_N = j, \mathbf{x}_{N-1}, \mathbf{y}_{N-1}) \quad (\text{A-3})$$

where K is the total number of clusters, \mathbf{z}_{N-1} is the partition of the first $N - 1$ objects into clusters, $P(x_N|z_N = k, \mathbf{z}_{N-1}, \mathbf{x}_{N-1})$ is the probability of x_N under cluster k , and $P(z_N = k, \mathbf{z}_{N-1}|y_N = j, \mathbf{x}_{N-1}, \mathbf{y}_{N-1})$ is the joint probability of generating a new object from cluster k and the partition of the previous $N - 1$ objects. Then this joint probability is given by,

$$P(z_N = k, \mathbf{z}_{N-1}|y_N = j, \mathbf{x}_{N-1}, \mathbf{y}_{N-1}) \propto P(y_N = j|z_N = k, \mathbf{z}_{N-1}, \mathbf{y}_{N-1})P(z_N = k|\mathbf{z}_{N-1})P(\mathbf{z}_{N-1}|\mathbf{x}_{N-1}, \mathbf{y}_{N-1}) \quad (\text{A-4})$$

where we take into account the fact that this new observation belongs to category y_N . The second term on the right hand side is given by Equation 4. This defines a distribution over the same K clusters regardless of j , but the value of K depends on the number of clusters in \mathbf{z}_{N-1} . Substituting this expression into Equation A-3 provides the relevant mixture model for the RMC. In general, the probabilities in the second term on the bottom line of Equation A-3 will never be precisely zero for any combination of cluster k and category j , so all clusters contribute to all categories. The RMC can therefore be viewed as a form of the mixture model in which all clusters are shared between categories but the number of clusters is inferred from the data. The dependency of the RMC between both features and category labels means that the prior over y_N depends on \mathbf{x}_{N-1} as well as \mathbf{y}_{N-1} , violating the (arguably sensible) independence assumption made by the other models and embodied in Equation 1.

In the equations for the RMC above, all clusters are used in each category, but a generalization of the RMC allows for other assumptions about the structure of categories. This generalization casts the exemplar, prototype, and RMC as restricted versions of Hierarchical Dirichlet Processes (Griffiths et al., 2007). Other models in this framework allow for probabilistic sharing of clusters between categories, or even completely independent sets of clusters for different categories.

Likelihood for discrete features

Given the cluster, the value on each feature is assumed to have a Bernoulli distribution. Integrating out the parameter of this distribution with respect to a symmetric Beta(β, β) prior, we obtain

$$P(x_{N,d} = v | z_N = k, \mathbf{x}_{N-1}, \mathbf{z}_{N-1}) = \frac{B_v + \beta}{B. + 2\beta} \quad (\text{A-5})$$

where B_v is the number of stimuli with value v on the d th feature that \mathbf{z}_N identifies as belonging to the same cluster as x_N . $B.$ denotes the number of other stimuli in the same cluster.

Likelihood for continuous features

Each feature within a cluster is assumed to follow a Gaussian distribution, with unknown mean and variance. The variance has an inverse χ^2 prior and the mean given the variance has a Gaussian prior

$$\sigma^2 \sim \text{Inv-}\chi^2(a_0, \sigma_0^2) \quad (\text{A-6})$$

$$\mu | \sigma \sim \text{N}\left(\mu_0, \frac{\sigma^2}{\lambda_0}\right) \quad (\text{A-7})$$

$$(\text{A-8})$$

where σ_0^2 is the prior variance, a_0 is the confidence in the prior variance, μ_0 is the prior mean, and λ_0 is the confidence in the prior variance. The parameters were set as they were in Anderson (1991): σ_0^2 is the square of one quarter of the dimension's range, μ_0 is the mean of the dimension. The parameters a_0 and λ_0 are described in the text.

Using these conjugate priors, the posterior predictive distribution is Student's t

$$P(x_{N,d} = v | z_N = k, \mathbf{x}_{N-1}, \mathbf{z}_{N-1}) \sim t_{a_i}\left(\mu_i, \sigma_i \left(1 + \frac{1}{\lambda_i}\right)\right) \quad (\text{A-9})$$

where

$$\lambda_i = \lambda_0 + n \quad (\text{A-10})$$

$$a_i = a_0 + n \quad (\text{A-11})$$

$$\mu_i = \frac{\lambda_0 \mu_0 + n \bar{x}}{\lambda_0 + n} \quad (\text{A-12})$$

$$\sigma_i^2 = \frac{a_0 \sigma_0^2 + (n-1)s^2 + \frac{\lambda_0 n}{\lambda_0 + n} (\mu_0 - \bar{x})^2}{a_0 + n} \quad (\text{A-13})$$

and \bar{x} is the mean and s^2 is the variance of the values for dimension d in cluster k , and n is the number of objects in cluster k .

Author Note

The authors would like to thank Jonathan Nelson, Josh Tenenbaum, Keith Rayner, John Anderson, Robert Nosofsky, John Kruschke, and Mike Oaksford for helpful comments and Matthew Loper for running preliminary simulations using particle filters in the RMC. ANS was supported by an NSF Graduate Research Fellowship and a Royal Society USA Research Fellowship. TLG was supported by grant number FA9550-07-1-0351 from the Air Force Office of Scientific Research and grant number IIS-0845410 from the National Science Foundation. DJN was supported by an Australian Research Fellowship (ARC grant DP0773794).

Footnotes

¹We should note that users of particle filter algorithms in computer science and statistics fight garden path effects through the use of particle rejuvenation or particle jittering (Fearnhead, 2004; Gordon, Salmond, & Smith, 1993). In rejuvenation, variance is added to the particles particles, possibly using different sampling algorithms such as Markov chain Monte Carlo. The additional sampling over partitions of stimuli would allow for the particle filter to explore possible clusterings of earlier stimuli that are irretrievable under our scheme. However, particle rejuvenation comes at the cost of additional computation so we have not implemented it in order to keep the simplest possible algorithms.

²In physics and statistics, $1/\gamma$ is often known as the temperature of the distribution.

Table 1

Approximation algorithms and their properties

Algorithms	Properties		
	Incremental	Single interpretation	Order effects introduced
Local MAP	Yes	Yes	Strong
Standard Gibbs sampling	No	No	Asymptotically none
Modified Gibbs sampling	No	Yes	In judgments
Multi-particle particle filter	Yes	No	Asymptotically none
Single-particle particle filter	Yes	Yes	Weak

Table 2

Transfer stimuli ordered by category 1 subject ratings from Medin and Schaffer (1978)

1111 0101 1010 1101 0111 0001 1110 1000 0010 1011 0100 0000

Table 3

Presentation order of Anderson and Matessa training stimuli (from Anderson, 1990)

Order Type	
Front-Anchored	End-Anchored
1111	0100
1101	0000
0010	1111
0000	1011
0011	0011
0001	0111
1110	1000
1100	1100
0111	1010
1010	0001
1000	0101
0101	1110
0110	1001
1011	0010
1001	0110
0100	1101

Table 4

Non-linearly separable category structures used in Experiments 1-3 of Smith and Minda (1998)

Experiment	Category A	Category B
	000000	111111
	100000	011111
	010000	101111
Exps 1 & 2	001000	110111
	000010	111011
	000001	111110
	111101	000100
	0000	1111
Exp 3	0100	1010
	0001	0111
	1011	1000

Figure Captions

Figure 1. Three example stimuli with their binary feature descriptions and order of presentation. The first feature codes for circle or square, the second feature codes for solid or empty, and the third feature codes for large or small.

Figure 2. Three different approaches to estimating the category distribution $p(x_N|y_N, \mathbf{x}_{N-1}, \mathbf{y}_{N-1})$. In all three cases, the learner knows that five objects (corresponding to the marked locations x_1 through x_5) all belong to a category, and the solid line plots the probability (density) with which a new object sampled from that category would be expected to fall in each location. The left panel shows a prototype model, in which all objects are clustered together, and are used to estimate the mean of this distribution (dashed line). On the right is an exemplar model, in which each object corresponds to a unique cluster, leading to a peak located over the top of each object. The intermediate case in the middle clusters objects 1-3 together and objects 4-5 together (i.e., $\mathbf{z} = [11122]$), with the result that there are now two peaks in the category distribution.

Figure 3. Illustration of the local MAP approximation algorithm, applied to stimuli shown in Figure 1. The local MAP begins on the left side with the initial stimulus. Every possible assignment of the new stimulus (marked by the arrow) that is consistent with the parent partition is enumerated and the posterior probability of each is written below each partition. Not all possible paths are followed. The local MAP algorithm chooses the partition with the highest posterior probability as its representation. The final output of the algorithm, a partition of the stimuli into clusters, is circled in red.

Figure 4. The relationship between (a) the clustering implied by the DP, (b) the distribution over parameters that is sampled from the DP, and (c) the mixture distribution over stimuli that results in the DPMM. The clustering assignments in (a)

were produced by drawing sequentially from the stochastic process defined in Equation 8, and each cluster is associated with a parameter value θ . The x stimuli are a set of undefined stimuli in which the features influence the clusters they belong to, but we are focusing on exploring the prior in this figure. After an arbitrarily large number of cluster assignments have been made, we can estimate the probability of each cluster, and hence of the corresponding parameter value. The resulting probability distribution is shown in (b). If each value of θ is treated as the mean of a simple normal distribution (with fixed variance) over the value of some continuous stimulus dimension, then the resulting mixture distribution drawn from the DPMM is the one illustrated in (c). While the applications considered in this paper also use stimuli that have discrete features, the notion of a mixture distribution is more intuitive in the continuous setting.

Figure 5. Illustration of the Gibbs sampling approximation algorithm, applied to stimuli shown in Figure 1. The Gibbs sampler begins on the left side with an initial partition of all of the stimuli and moves to the right side. Each box is a partition that contains one or more stimuli and the presence of a separating vertical line indicates that the stimuli belong to different clusters. The partitions in each column are the partitions under consideration, given the partitions in the previous column. These children partitions are all the possible reassignments of the stimulus marked by the arrow. Numbers underneath each partition show the posterior probability of that partition. After an iteration through each stimulus, the end state is retained and is also used as the initial partition for the next iteration. The final outputs of the algorithm, two samples of partitions of the stimuli into clusters, are circled in red.

Figure 6. Illustration of the particle filter approximation algorithm, applied to stimuli shown in Figure 1. The particle filter starts on the left side with the initial stimulus as the partition represented by each of the $M = 2$ particles. Every particle produces all possible

assignments of the new stimulus, marked by the arrow, that are consistent with the previous partition. $M = 2$ partitions are sampled based on their posterior probabilities represented by numbers underneath the partitions, without regard for ancestry. After the final stimulus, the sampled partitions are used as samples from the posterior distribution. The final outputs of the algorithm, two samples of partitions of the stimuli into clusters, are circled in red.

Figure 7. Results of the approximation algorithms compared to the exact posterior. The five bar groupings correspond to the five possible partitions of the three stimuli in Figure 1. The bars within each grouping correspond to the approximation algorithms outlined in the text. Standard error bars are provided for the Gibbs sampling, multi-particle particle filter, and single-particle particle filter algorithms.




Figure 8. Probability of choosing category 1 for the stimuli from the first experiment of Medin & Schaffer (1978). The transfer stimuli (listed in order of human preference) are along the horizontal axis. In the first row only the first six trials are presented, while in the second row ten blocks of six trials each are presented. The two lines in each panel correspond to two different coupling parameters: for the triangles, $c = 0.1$ and for the circles, $c = 0.3$. Pearson correlations between the human data and the simulation data are displayed on each plot for each value of the coupling parameter.

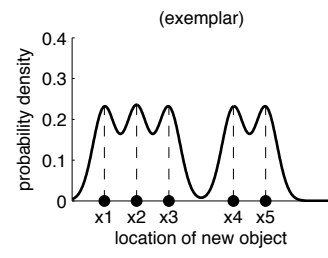
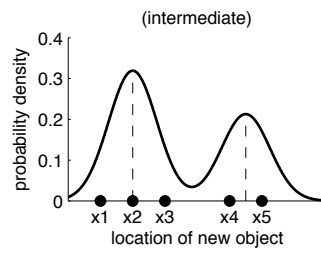
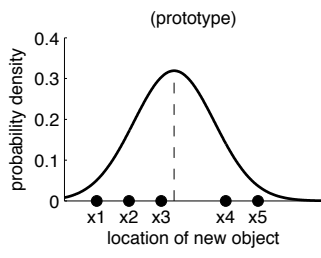
Figure 9. Data and model fits to data from Experiments 1-3 of Smith and Minda (1998). The line plots show the proportion of trials on which category A was chosen for each stimulus. Stimuli belonging to category A are marked with an x, while stimuli belonging to category B are marked with a circle. The red and blue lines highlight the stimuli that are unusual for in each category. The bar plots show how the unusual stimuli were classified in early, middle, and late blocks. When the two stimuli were both more often

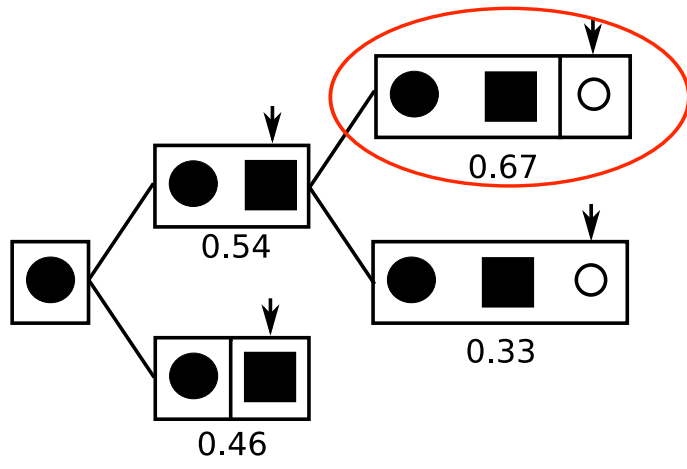
given the incorrect response, they were classified as opposite. Both A and Both B mean that both stimuli were more often classified in one of the two categories than the other, and Correct means that both unusual stimuli were classified correctly on average.

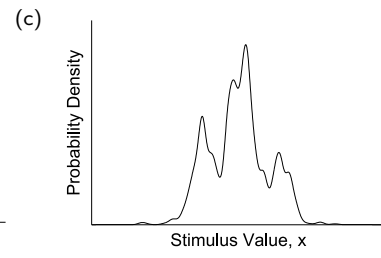
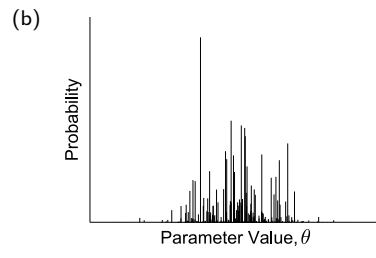
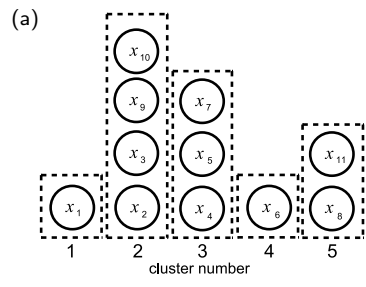
Figure 10. Problem types from Shepard, Hovland, and Jenkins (1961). The three dimensions of the cube represent the three binary dimensions of the stimuli. Each vertex of a cube is labeled as part of Category A or Category B for each of the six problems.

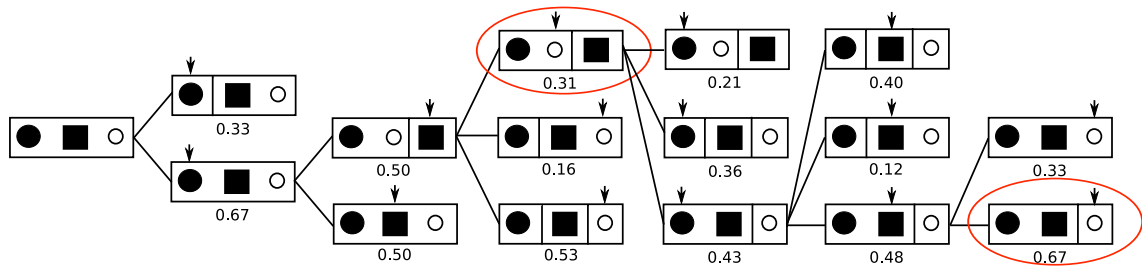
Figure 11. Human data from Nosofsky, Gluck, Palmeri, McKinley, and Glauthier (1994), along with the best fitting local MAP and single-particle particle filter algorithms to these data. Each line is a separate problem type.

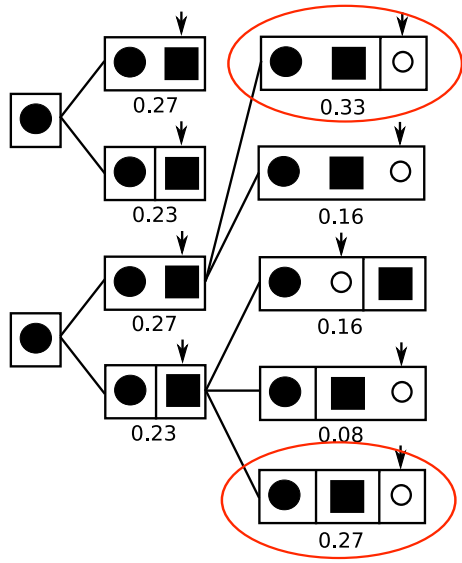
Trial	I	II	III
Stimulus			
Features	111	011	100

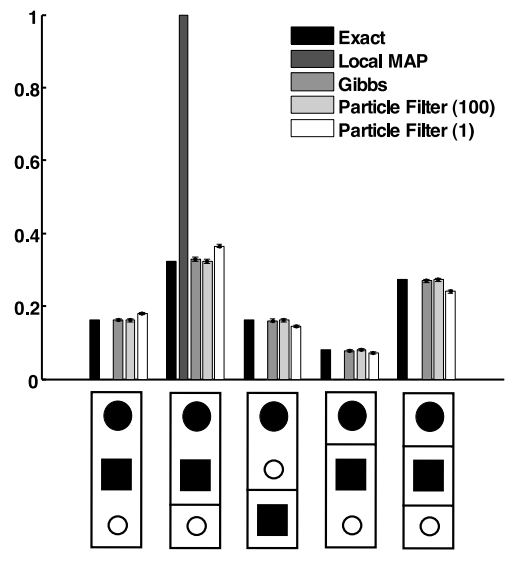


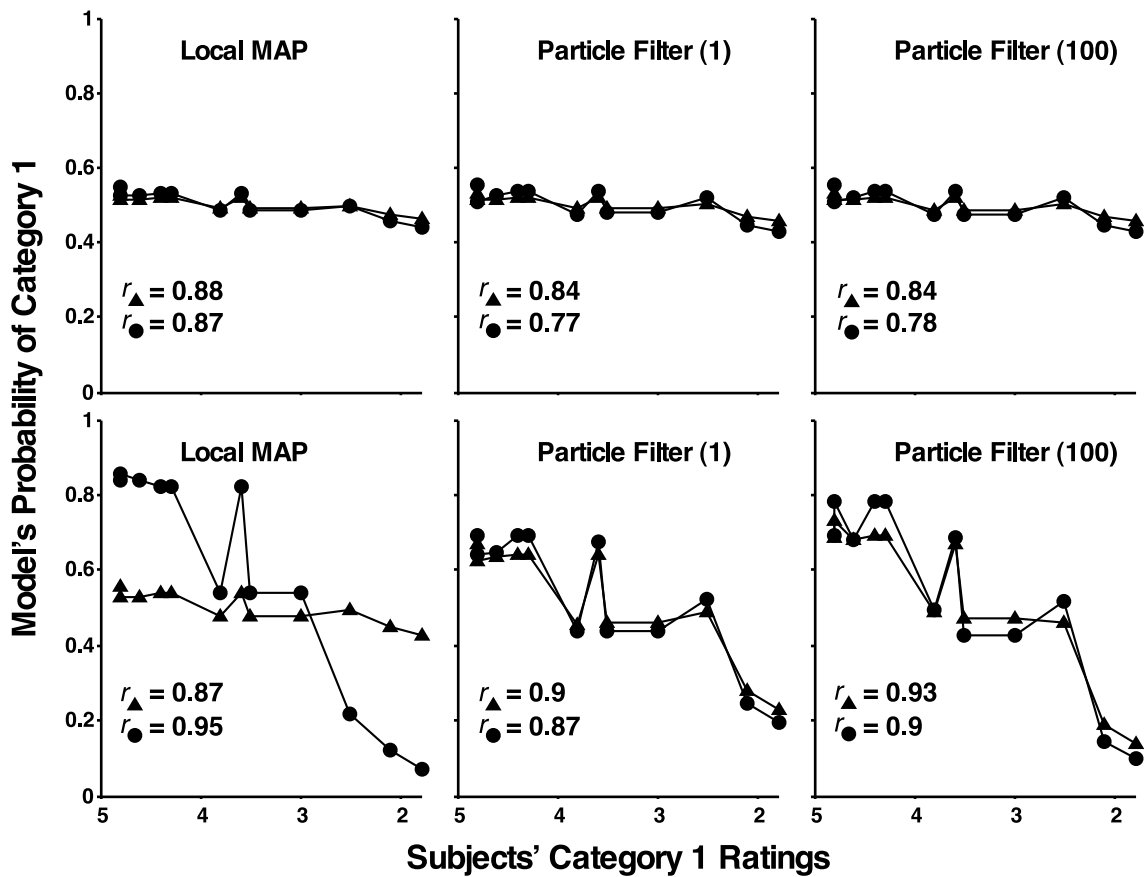




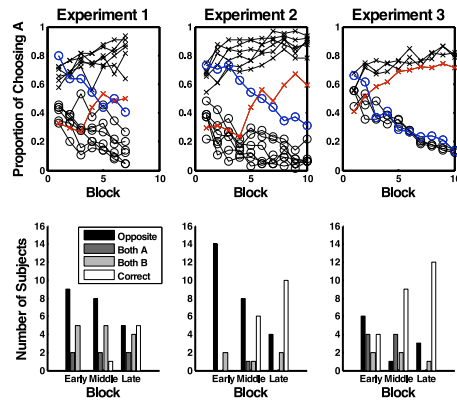




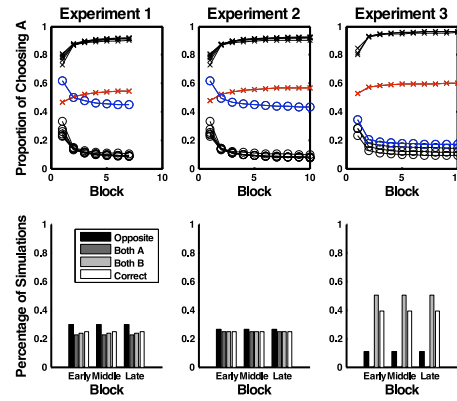




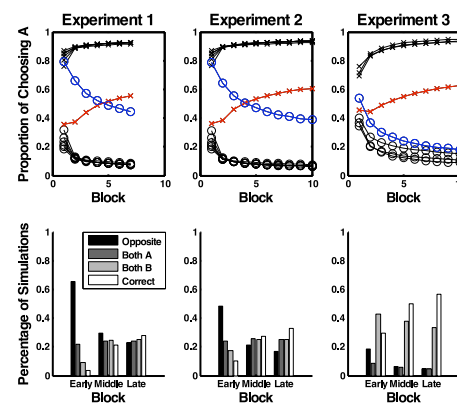
Human Data



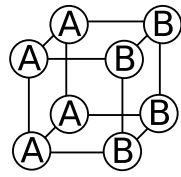
Local MAP



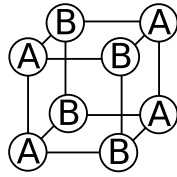
Single-Particle Particle Filter



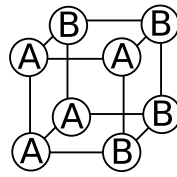
Type I



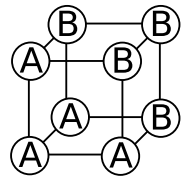
Type II



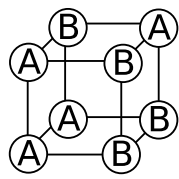
Type III



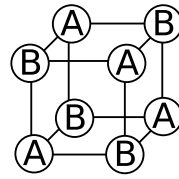
Type IV

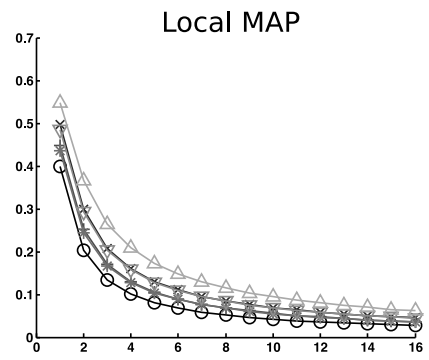
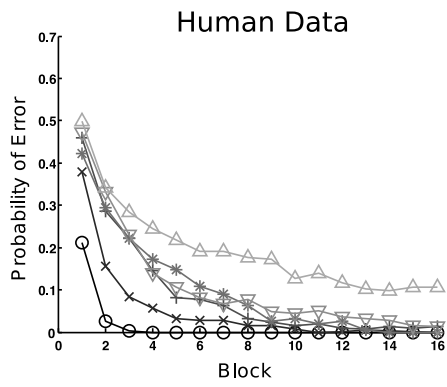


Type V



Type VI





Problem Types

- I
- × II
- ⊕ III
- * IV
- ▽ V
- △ VI

