Exploring the structure of mental representations by implementing computer algorithms with people

Adam N. Sanborn
University of Warwick

Thomas L. Griffiths
University of California, Berkeley

Abstract

Computer metaphors have been a mainstay of cognitive psychology, treating minds like information processing systems that operate on memory and mix serial and parallel processes. Recent work has taken this idea beyond a metaphor, examining the consequences of treating people as components in an algorithm that is usually executed by digital computers. By implementing a popular class of randomized algorithms – Markov chain Monte Carlo algorithms – with people, we are able to generate samples from subjective probability distributions that reveal the mental representations that inform people's judgments. In this chapter, we summarize the key idea behind this approach and review a number of ways in which it has recently been extended.

The computer metaphor has had a long and productive history in cognitive psychology. One of its first uses was in the presentation of the Atkinson and Shiffrin (1968) model of human memory. The memory system was described as a computer under the control of a remote operator: structural features of memory were described as hardware or built-in programs, while memory control processes were described as the commands and programs under the control of the remote operator. The computer metaphor has yielded many fruitful results, including further insights into human memory (Raaijmakers & Shiffrin, 1981; Shiffrin & Steyvers, 1997) and the characterization of cognitive processes as serial or parallel (Schneider & Shiffrin, 1977; Shiffrin & Schneider, 1977; Sternberg, 1966; Townsend & Ashby, 1983). Recently, the connections between computer science and cognitive psychology have been made tighter, as algorithms used in computer science have been proposed as candidates for cognitive processes (Daw, Courville, & Dayan, 2008; Griffiths, Vul, & Sanborn, 2012; Sanborn & Silva, 2013; Sanborn, Griffiths, & Navarro, 2010; Shi, Griffiths, Feldman, & Sanborn, 2010). Importation of ideas has run both ways: studying computa-

ANS was supported by grant number ES/K004948/1 from the UK Economic and Social Research Council and TLG was supported by grants FA-9550-07-1-0351 and FA-9550-10-1-0232 from the Air Force Office of Scientific Research.

tional problems in psychology has also led to the development of new methods in computer science (Anderson & Matessa, 1992; Griffiths & Ghahramani, 2005).

One opportunity arising from the closer links between psychology and computer science is that it is now relatively easy to characterize the capabilities of both people and machines in a common language to allow them to work together for a common goal. In some domains this is an established and essential idea – researchers in human-computer interaction have designed machines and trained people to improve cooperative performance on specific tasks, and the catastrophes resulting from failures of cooperation have highlighted its importance (Kemeny et al., 1979). However, this is still a relatively new idea in the context of developing methods for psychological research. Here we review work in which a formal characterization of human decision making allows a person and a machine to work together as components of an algorithm used to investigate a person's beliefs.

Following the computer metaphor, beliefs can be thought of as the data upon which cognitive algorithms operate. For much of the cognitive data in which psychologists might be interested, such as the strength of category membership (Nosofsky, 1988; Rosch, 1973), the strength of a memory (Shiffrin & Steyvers, 1997), or the strength of an economic preference (Kahneman & Tversky, 1979; Savage, 1954), the default assumption is that there is a single non-negative number that represents that strength. The beliefs that experimenters are interested in are therefore the set of strengths over a set of stimuli. The problem with investigating these beliefs is that for any sort of naturalistic problem there is a huge number of stimuli, but relatively few stimuli are relevant to a particular question (and for interesting questions the relevant stimuli are not known before the experiment begins). As a result, the most straightforward methods for testing people's beliefs, such as providing ratings for every stimulus, will result in many wasted trials.

Computer scientists and statisticians often encounter exactly the same problem – evaluating a function that has large values for only a few inputs. For example, high-dimensional probability distributions typically concentrate their mass on a small part of a large space. A number of sophisticated algorithms have been developed to make it possible to work with such distributions. One class of algorithms, *Markov chain Monte Carlo* (MCMC), is used to draw samples from these distributions by constructing a Markov chain that visits higher probability regions with greater frequency. These samples can then be used to characterize the form of the distribution, sometimes as simply as constructing a histogram from the samples.

MCMC is normally used to draw samples from probability distributions represented within a computer. By dividing each non-negative strength by the sum of all the strengths, a person's beliefs can also be represented as a probability distribution over the set of possible stimuli. This probability distribution is implicitly represented inside human heads, not digital computers. However, this is not an impediment to running an MCMC algorithm: using a standard model of people's decision making process, we can specify a task that allows people to act as an element of such an algorithm. The algorithm produces samples drawn from the distribution reflecting a person's beliefs and these samples can then be used to reconstruct that probability distribution. Because samples occur more often in high probability regions, this approach reduces the number of trials wasted on irrelevant stimuli, as illustrated in Figure 2. The resulting method for exploring people's beliefs is called *Markov chain Monte Carlo with people* (MCMCP; Sanborn, Griffiths, & Shiffrin,

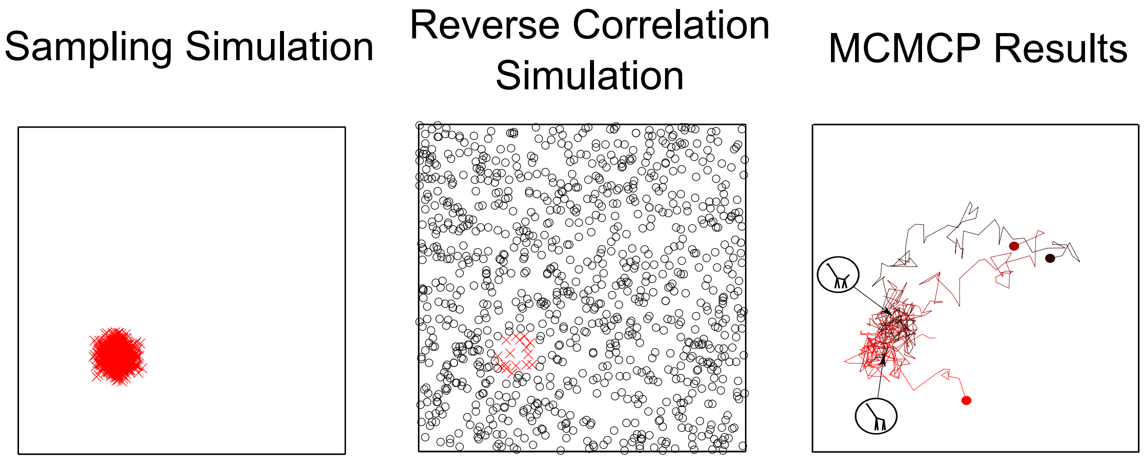

Figure 1. Comparison of different methods for learning about participant beliefs. The sampling simulation is 1,000 samples from the belief distribution, which accurately reflect its structure. The reverse correlation simulation shows 1,000 samples drawn uniformly and labelled as part of the belief distribution if they exceed a fixed probability threshold (1/30th of the distribution peak). The MCMCP plot shows empirical results from Experiment 3 of Sanborn et al. (2010): three chains (in different colors) from a participant judging whether stick figures were giraffes. This plot is a two-dimensional projection of the nine-dimensional parameter space. Adapted from Sanborn et al. (2010), with permission from Elsevier.

2010).

In this chapter, we summarize the ideas behind MCMCP, and describe a number of recent extensions to the method. We begin with a more detailed introduction to MCMC algorithms and MCMCP. Next, we review evidence of the effectiveness of MCMCP and compare it to another well-known method for eliciting beliefs about complex naturalistic stimuli, reverse correlation. We then discuss three ways in which MCMCP has been extended: taking advantage of a manifold structure, working with discrete stimuli, and focusing trials in areas of interest to the experimenter.

Markov Chain Monte Carlo with People

Markov chain Monte Carlo algorithms generate samples from complex and high-dimensional probability distributions by constructing a Markov chain that has the target distribution as it stationary distribution (for a general introduction, see Gilks, Richardson, & Spiegelhalter, 1996). A Markov chain is a sequence of random variables where each variable is independent of all other preceding variables when conditioned on the variable that immediately precedes it in the sequence. The values that each variable can take on are known as the *state space* of the Markov chain (with the value of a single variable being its *state*), and the probability each variable takes on a particular value given the value of the preceding variable is called the *transition probability.*

A good example of a Markov chain is shuffling a deck of cards. The state space is all possible orderings of the cards, and the procedure used for shuffling defines the transition probabilities. If you know the order the cards were in prior to the most recent shuffle, knowing what order they were in before that provides no further information. If this were not the case, you would have to worry about the entire history of games played with a deck of cards every time you played a game.

The reason we shuffle cards is to put them in a random order. More precisely, a good shuffling procedure is one that, regardless of what order the cards begin in, results in a uniform distribution over all possible orderings after some number of shuffles. The fact that it is possible to do this is a result of the convergence of Markov chains to a stationary distribution: provided the transition probabilities of a Markov chain satisfy a set of properties that make the chain *ergodic* (see Norris, 1997), the probability that the last variable takes on a particular value converges to a fixed probability as the length of the chain increases, regardless of the value of the first variable. These fixed probabilities constitute the stationary distribution of the chain.

The stationary distribution of a good shuffling scheme is uniform over all orders of cards, but we can construct Markov chains with different stationary distributions. For example, a shuffling scheme with a slight tendency to put red cards on top of the deck would result in a stationary distribution that favors orders where red cards are higher in the deck. MCMC algorithms exploit this principle, setting the transition probabilities of a Markov chain in such a way that a particular distribution – the distribution we want to sample from – is the stationary distribution. Once the Markov chain converges to its stationary distribution, its values can be treated like samples from that distribution.

The most widely-used MCMC method is the Metropolis algorithm (Metropolis, Rosenbluth, Rosenbluth, Teller, & Teller, 1953; Neal, 1993). In this algorithm, the transition probabilities of the Markov chain are decomposed into two multiplicative parts. The first part is the proposal probability, which is the probability of proposing a possible new state for the Markov chain. The second part is the acceptance probability, which is the probability of accepting the proposal and transitioning to the proposed state. This decomposition is useful because the proposal distribution requires no knowledge of the target distribution; only the acceptance function uses information about the target distribution.

A sufficient condition for specifying transition probabilities that result in a particular stationary distribution is known as *detailed balance*,

$$p(x)q(x^*|x)a(x^*;x) = p(x^*)q(x|x^*)a(x;x^*), \tag{1}$$

where $p(x)$ is the target distribution (which we want to be the stationary distribution), $q(x^*|x)$ is the probability of proposing a new state x^* given the current state x, and $a(x^*;x)$ is the probability of accepting that proposal. A Markov chain that satisfies detailed balance has equal probability of moving in either direction between two states once it reaches its stationary distribution.

If the proposal distribution is set to be symmetric, meaning that the probability of proposing x^* given the current state x is the same as proposing the current state x given x^*, then we can cancel $q(x|x^*)$ and $q(x^*|x)$ in Equation 1. Consequently, we can satisfy detailed balance using an acceptance function that depends just on the probabilities of the current and the proposed state. One acceptance function that satisfies detailed balance is the Barker or Boltzmann acceptance function (Barker, 1965; Flinn & McManus, 1961)

$$a(x^*;x) \quad = \quad \frac{p(x^*)}{p(x^*)+p(x)}, \tag{2}$$

where the acceptance probability is proportional to the probability of the proposed and the current state under the target distribution. A nice feature of the method is that the acceptance function only requires a knowledge of the ratios of the probabilities, meaning that any values proportional to the probabilities can be used in place of the probabilities.

Putting People into the Algorithm

The algorithm presented above defines a Markov chain that will converge to the distribution $p(x)$, which is usually a probability distribution represented by a computer. But we can use the same algorithm to explore people's beliefs if we can define a task where those beliefs affect people's behavior in exactly the way that $p(x)$ affects the probability of accepting a proposed state. In the standard task used in MCMCP, a participant is given two options and instructed to select the one that better answers a particular question (Sanborn, Griffiths, & Shiffrin, 2010). For example, participants might be asked to pick which of two stick figures is a better example of a giraffe.

A default description of people's decisions in a two-alternative forced choice task is the ratio rule (Luce, 1959; Shepard, 1957). This rule relates the goodness of each option, described as $f(x)$ and $f(x^*)$ respectively, to the probability of choosing one option over the other

$$a(x^*; x) = \frac{f(x^*)}{f(x^*) + f(x)}. \tag{3}$$

This equation is formally identical to the Barker-Boltzmann acceptance function given in Equation 2, meaning that participants in this task act equivalently to the acceptance function in the MCMC algorithm, converging to a distribution $p(x)$ proportional to $f(x)$. Crucially, the experimenter does not need to know $p(x)$ or $f(x)$ in order for this method to work – these quantities only govern the probability of a participant's choice. This scheme matches the stochasticity required by the MCMC algorithm to the stochasticity in human choice behavior, so that human choices are, at least to the extent that they are well-described by the ratio rule, a perfect implementation of the algorithm.

A more accurate characterization of human behavior in this task is that people often respond more deterministically than predicted by Equation 3 (Ashby & Gott, 1988; Vulkan, 2000). As a result a more general exponentiated form of the ratio rule has been used (Ashby & Maddox, 1993)

$$a(x^*; x) = \frac{f(x^*)^\gamma}{f(x^*)^\gamma + f(x)^\gamma}, \tag{4}$$

where γ is a constant that makes it possible to capture more deterministic behavior than the probability matching of Equation 3: a value of $\gamma = 1$ is equivalent to Equation 3, but values of $\gamma > 1$ increase the determinism of the choice. Equation 4 is still useful as an acceptance function, but it converges to a distribution $p(x)$ proportional to $f(x)^\gamma$ rather than $f(x)$. An unknown γ introduces a degree of freedom into the results, so the order of the probabilities of different options will be known, but cannot be directly evaluated. Consequently, the distribution has the same peaks as when γ is known, and the ordering of the variances for the dimensions is unchanged, but the exact values of the variances will depend on the parameter γ.

Using this correspondence between human behavior and the acceptance function, a long sequence of trials can be used to produce samples from a distribution that reveals people's beliefs. On each trial one of the alternatives is the state of the Markov chain and the other is the proposal, where the proposal is chosen probabilistically by the computer. Participants choose between the two alternatives and the chosen alternative is used as the state on the next trial. This procedure continues through a "burn-in" period, to remove the influence of the starting state, and states of the chain following the burn-in period are then treated as (dependent) samples from the target distribution. These samples can then be used to reconstruct the probability distribution proportional to the beliefs about the question that was asked to participants.

An example for illustration is shown in Figure 2, showing the results of a single participant who was asked which stick figure was a better example of a giraffe. Stick figures were represented as points in a nine-dimensional stimulus space, using dimensions such as head length and neck angle (Olman & Kersten, 2004). Three interleaved chains were started at experimenter-chosen points in the space, and proposals were chosen mainly from nearby points in the space, with the constraint that the proposals had to be within bounds chosen to prevent impossible-to-display or very strange stick figures. While the stimulus space was not carefully chosen to align with psychological or perceptual variability, the participant's choices quickly caused the chains to converge on the same small area of the nine-dimensional parameter space. After removal of the initial burn-in trials, the samples show what this participant believes a giraffe looks like.

Testing the Method

Markov chain Monte Carlo with people has been tested for accuracy by investigating whether samples from the algorithm matched category distributions upon which people were trained (Sanborn, Griffiths, & Shiffrin, 2010). These samples were also compared to people's category discrimination judgments to show that the output of the algorithm could predict more standard category discrimination judgments. Below we review these findings and demonstrate the generality and efficiency of MCMCP by evaluating it against an extant method for investigating people's mental representations in a high-dimensional space: reverse correlation.

Accuracy

In order to test the accuracy of MCMCP, in initial experiments participants were trained on one-dimensional categories, in which the dimension described the height of fish presented on a computer screen (Sanborn, Griffiths, & Shiffrin, 2010). In two experiments, participants were trained on a uniform distribution (characterizing the size of fish who came from the open ocean) and a Gaussian distribution with a particular mean and standard deviation (characterizing the size of fish who came from a fish farm). The training trials were discriminative trials in which participants decided whether a single example belonged to the uniform or the Gaussian distribution.

The first experiment using MCMCP investigated a set of questions to determine which would be able to produce samples that reflected the trained distribution. Participants were asked one of three questions aimed to elicit judgments that matched the desired decision rule: "Which fish came from the fish farm?", "Which fish is a better example of a fish that came from the fish farm?", and "Which fish was more likely to have come from the fish farm?". Each of the questions led to participants producing samples that were a close match to both the mean and standard deviation of the training distribution.

The second experiment investigated whether training participants on different distributions would result in sets of samples that were distinguishable from one another. Four groups of participants were trained on a factorial combination of low and high means and low and high standard deviations, and the means and standard deviations recovered showed reliable differences between different training distributions. The difference in the average recovered distributions for participants trained with the same means and different standard deviations are shown in Figure 2 for the fish stimuli. These two experiments demonstrated that people's trained beliefs could be accurately estimated from the samples generated by MCMCP.

Further research with natural categories has demonstrated that MCMCP can produce sensible looking results in a variety of domains including stick figure animals (Sanborn, Griffiths, & Shiffrin, 2010), intuitive physics (Cohen & Ross, 2009), and facial expressions (Martin, Griffiths, & Sanborn, 2012; McDuff, 2010). In addition, the data gathered in these experiments were also able to predict choices in the discrimination trials. Prediction of discrimination trials was above chance in both fish experiments, and this was also the case in the separate experiments using stick figure animals (Sanborn, Griffiths, & Shiffrin, 2010) and cartoon facial expressions (McDuff, 2010). Individual differences were reflected in the samples, as Huszár, Noppeney, and Lengyel (2010) showed that prediction of discrimination results was better than an ideal observer trained on the fish category distributions shown to participants.

Generality and Efficiency

While MCMCP was developed to deal with the problem of determining people's beliefs about high-dimensional stimuli, it is not the only method that does so. A popular existing method for determining people's beliefs about high-dimensional stimuli is known as *classification images* or *reverse correlation* analysis. In its most basic form, this method involves collecting a set of stimuli associated with each category and determining the difference in the means between the stimuli from the two categories. (Ahumada & Lovell, 1971; Beard & Ahumada, 1998; Olman & Kersten, 2004). This provides an estimate of a linear decision boundary between the categories. This approach has been used extensively in the literature on visual perception, being applied to problems such as visualizing the illusory contours people see (Gold, Murray, Bennett, & Sekuler, 2000) and visualizing facial affect categories (Kontsevich & Tyler, 2004).

MCMCP is more general than reverse correlation, as the linear decision bound between two categories provides much less information than is in the full distribution for each category recovered by the MCMCP method (Sanborn, Griffiths, & Shiffrin, 2010; Victor, 2005).[1] A simple example in which MCMCP would outperform reverse correlation is recovering two categories with the same mean but different variances, as in the fish experiment described above. MCMCP recovers the mean and variance of each category as was demonstrated, but reverse correlation would recover the same mean for each category.

[1] Although it is worth noting that the reverse correlation method can be extended to capture other quantities such as second order decision boundaries (Neri, 2004) or features (Gold, Cohen, & Shiffrin, 2006).

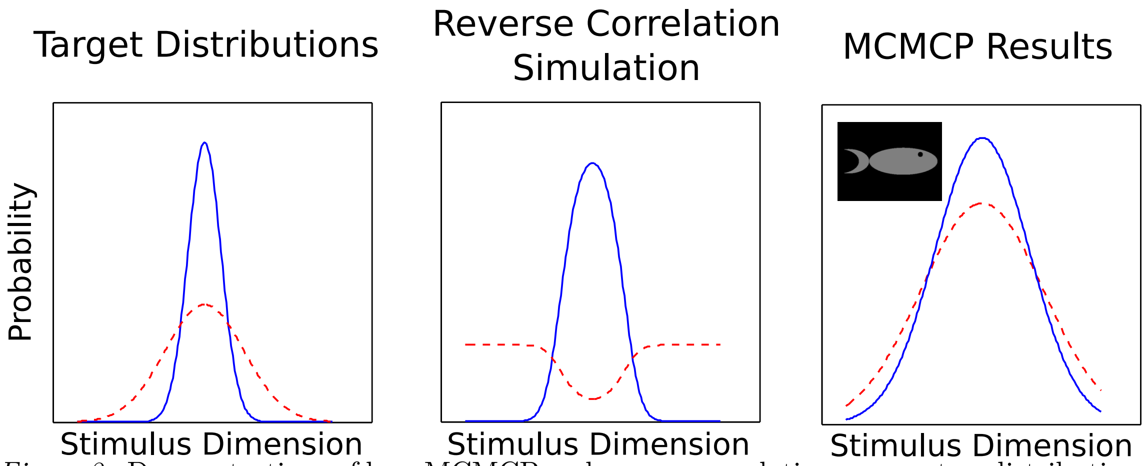

Figure 2. Demonstrations of how MCMCP and reverse correlation recover two distributions over a single parameter that have the same mean, but different standard deviations. The two distributions are shown in the target distributions plot. The reverse correlation simulation takes a uniform sample of stimuli from the space and labels them using the ratio decision rule. The MCMCP plot shows empirical results from Sanborn et al (2010): the average distributions of participants in the two different standard deviation training conditions of Experiment 2, with an example of the stimuli inset. The recovered distributions have higher variance than the target distribution, perhaps due to perceptual noise or $\gamma < 1$.

Even looking at the full collection of trials assigned each label would be misleading – the collection of trials associated with the more variable category would look bimodal with a hole in the center, when it is actually unimodal with a peak in the center, as shown in Figure 2. The discrimination trials show a hole because even though the high-variance category has its highest probability at the common mean, the low-variance category has even higher probability and shows its greatest relative advantage at the common mean.

Aside from the generality of the method, there is also the question of efficiency: how does MCMCP compare to reverse correlation for categories for which reverse correlation is expected to perform well? As there are many fewer degrees of freedom in determining a decision boundary than in determining an entire distribution, in principle reverse correlation should be more efficient in some situations. However, MCMCP has a second advantage over reverse correlation due to the focused sampling of the MCMC algorithm. Unlike reverse correlation, in which the stimuli that people make judgments about must be selected before the experiment begins, the stimuli used in MCMCP are automatically concentrated on the interesting portions of the space.

The most extensive comparison of MCMCP and reverse correlation has used categories of human faces. Kontsevich and Tyler (2004) showed that for facial affect categories, the mean of the stimuli associated with categories of happy and sad faces produced images that clearly distinguished the two categories. Martin et al. (2012) tested MCMCP against reverse correlation in this problem, using a 175-dimensional representation of faces. The dimensions were the first 175 eigenfaces from the California Facial Expression database (Dailey, Cottrell, & Reilly, 2001). Participants in the MCMCP condition made choices between pairs of faces to answer the question of which was “happier” or “sadder” while participants in the reverse correlation condition labelled a randomly generated set of faces as “happy”, “sad”, or “neither”. To evaluate the goodness of the resulting estimates of the means of the categories, another set of participants rated the mean faces for each category produced by each method after various numbers of choices had been made in each experiment to determine which method converged more quickly. Comparing the mean faces from the two methods after the same number of choices had been made, participants judged the mean happy faces from MCMCP to be statistically significantly happier than the mean happy faces from reverse correlation, with a similar result holding for sad faces.

Extensions

Markov chain Monte Carlo with people can provide more information about the structure of categories, and do so faster, than competing methods. With enough trials, MCMCP will be able to estimate any probability distribution, no matter how complex, as long as the Markov chain is ergodic. However, there are problems for which the number of MCMCP trials required would be impracticably large. For example, there could be multiple well-separated modes in the target distribution, and if the proposal distribution is relatively narrow the expected number of trials required to transition between modes would be large. Another potential problem is an experimenter who is interested in the lower probability regions of a target distribution – relatively few samples will be collected in these regions, increasing the total number of trials required.

Recent work with MCMCP has explored other ideas from computer science and statistics that can be used to address these problems. Below we review three such extensions: taking advantage of a manifold structure, working with discrete stimuli, and focusing trials in areas of interest to the experimenter.

Taking Advantage of a Manifold Structure

Many of the parameterized stimulus spaces that experimenters construct are not a good match to mental representations. As an example, reverse correlation usually adds independent luminance noise to each pixel (cf. Olman & Kersten, 2004), even though adding in this type of noise to an image of an object is extremely unlikely to result in an image of another object. The images of objects lie on a manifold within the pixel space, a lower-dimensional representation that is not known to the experimenter.

For MCMCP, a mismatch between the stimulus space and the mental representation reduces the efficiency of the algorithm. A general problem in sampling is producing good proposal distributions, as illustrated in Figure 3 which shows a toy example of attempting to sample golden rectangles (rectangles that have a fixed ratio of height to width of 1.618, a ratio that played an important role in ancient art) from the space of all possible rectangles. For this distribution only a particular kind of change to both the height and width results in another golden rectangle. MCMCP proposes without knowledge of the structure of the space and as a result will produce many poor proposals for this problem.

Many different types of algorithms have been developed to make MCMC more efficient for this type of problem, such as Hybrid Monte Carlo (Duane, Kennedy, Pendleton, & Roweth, 1987). Very roughly, this method uses the idea of momentum, so that the proposals

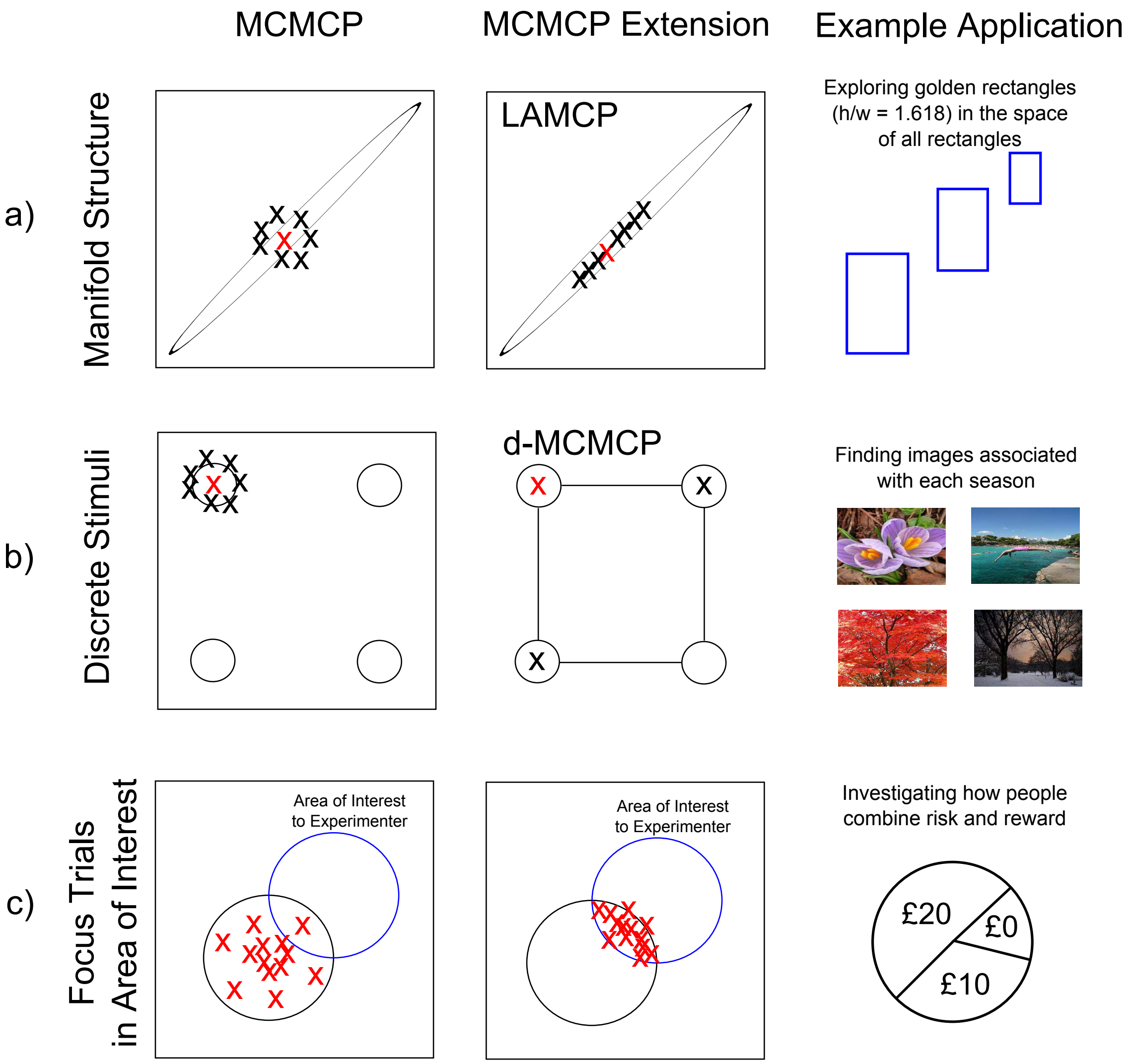
MCMCP
MCMCP Extension
Example Application
a)
Manifold Structure
LAMCP
Exploring golden rectangles (h/w = 1.618) in the space of all rectangles
b)
Discrete Stimuli
d-MCMCP
Finding images associated with each season
c)
Focus Trials in Area of Interest
Area of Interest to Experimenter
Area of Interest to Experimenter
Investigating how people combine risk and reward
£20
£0
£10

Figure 3. Extensions to MCMCP for three different methodological challenges. The plots show how MCMCP and extensions address these challenges, and an example of each challenge is shown as well. In each plot, the red x's correspond to the current state or past states of the Markov chain and the black x's correspond to proposals based on the current state. The black circles and ellipses correspond to the target distributions of participants and the blue circle corresponds to the target distribution of the experimenter. A different extension is used for each challenge, and these are discussed in the text. a) The challenge is that the strength distribution lies on a one-dimensional manifold in a two-dimensional parameter space. b) The challenge is that the stimuli are discrete and the space is poorly parameterized such that high probability regions are far from one another. c) The challenge is that the experimenter also has a preference distribution for which stimuli should be tested.

will tend to be made along the line in which the chain has been traveling, taking advantage of a smooth manifold structure. We could not find a way to implement this algorithm directly with people, so instead we implemented an analogue of momentum in order to take advantage of the manifold structure in the stimulus space, a method called *look-ahead Monte Carlo with people* (LAMCP; Blundell, Sanborn, & Griffiths, 2012).

In LAMCP, participants alternate between two types of trials. One is a regular MCMCP trial. The other is a trial in which participants picked between various proposal distributions for the next trial. The proposal distributions are lower-dimensional slices of the full stimulus space, and participants can choose between animations that illustrated what sorts of proposals each type of distribution would generate. The proposed directions do not necessarily have to be linear slices, but like state proposals the direction proposals must be symmetric: the probability of proposing direction d^* given the previous direction was d is the same as the converse probability. Crucially, slices that were previously chosen would appear again as a choice on the next trial, allowing the participants to continue along a direction if they so wished.

Blundell et al. (2012) tested this method on the golden rectangle problem. The full stimulus space included all possible rectangles, and participants were asked to select the golden rectangle on each trial. The animations shown to participants were smooth linear trajectories through the space of rectangles determined by height and width. As a result of using this form of momentum, participants in the LAMCP condition were able to converge to their target distribution at a much faster rate than participants in the MCMCP condition.

Working with Discrete Stimuli

While LAMCP is useful for investigating questions with categories that form a manifold within a parameterized stimulus space, there are other types of stimuli, such as images and text from large online databases, for which the parameterization may be too poor to use. Here, to increase efficiency, it can be better to use the examples themselves rather than construct a parameterized space.

To use MCMCP with discrete items, some way of making local proposals must be found. Hsu, Martin, Sanborn, and Griffiths (2012) did this by constructing a graph of the discrete items: the nodes of the graph were the items themselves and the edges of the graph linked items that were most similar to one another, using a rough and ready measure of similarity. The proposal distribution used in the MCMC algorithm then follows the structure of the graph, proposing a move to states that are neighbors of the current state. Figure 3 illustrates how making proposals in a graph is superior to making proposals in a poorly-parameterized space. In the case of images, the experimenter can avoid asking participants to make judgments about uninterpretable images, instead restricting their choices to images that are interpretable and possibly relevant to the question.

Constructing the graph is not trivial. The assumption that the proposals are symmetric means that every node in the graph must have equal degree (i.e., an equal number of links to other nodes). If the k nearest neighbors in terms of similarity are selected for each item, then it is not necessarily the case that if item j is one of the k nearest neighbors of item i that the converse is true. In order to make symmetric links, we might either make edges if either of the items are in the k nearest neighbors of the other or make edges only if both items are in the k nearest neighbors of the other. Either of these combination rules lead to nodes having different numbers of links and thus not producing a useable graph for the purposes of MCMCP.

Again the solution to this problem can be found in the computer science and statistics literature. Formally, we want to find the graph that maximizes the similarity along edges while keeping the degree of each node constant

$$\arg\max_{G} \sum_{ij} G_{ij} S_{ij} \text{ such that } \sum_{i} G_{ij} = b; G_{ii} = 0; G_{ij} = G_{ji} \tag{5}$$

where G is the adjacency matrix of the graph, with $G_{ij} = 1$ if there is an edge from i to j and $G_{ij} = 0$ otherwise, and S is the symmetric similarity matrix which gives a rough measure of similarity between pairs of items. S could be constructed, for example, using the similarity between the color histograms of images. In computer science, this is the *maximum weight b-matching* problem (Papadimitriou & Steiglitz, 1998). There exist exact algorithms for solving this problem, such as the blossom algorithm (Edmonds, 1965), but these algorithms are impractical for large-scale applications. Consequently, we approximate the b-matching using a common approximation from computer science and statistics (Jebara & Shchogolev, 2006). The use of the b-matching for proposal distribution was termed discrete MCMCP (d-MCMCP).

d-MCMCP was tested using the same facial affect categories as used in Martin et al. (2012). However, instead of using the points in the eigenspace of faces, the actual faces themselves were used. Like the comparison between reverse correlation and MCMCP, d-MCMCP was compared to MCMCP after various numbers of choices to determine which method converged faster. The mean happy faces from d-MCMCP were a significant improvement over the mean happy faces from MCMCP after the same number of choices, demonstrating the effectiveness of the method. In addition d-MCMCP was applied to images of seasons downloaded from search engines. After constructing a graph based on image similarity, the probability distribution over images associated with each season was found, with the results showing plausible distributions for each season.

Focusing Trials in Areas of Interest

MCMCP and the above extensions all work toward the goal of sampling from the target distribution, whether the stimulus parameterization is good or not. However, there are also situations in which an experimenter might find the stimuli that have moderate strength to be more theoretically interesting.

As an example, consider a gamble in which the probabilities of three outcomes, £0, £10, and £20, are manipulated. The participant's highest preference will almost certainly be for a 100% chance of £20, but the experimenter will be more interested in how the participant evaluates gambles away from the mode of the subjective distribution. One particular motivation, especially for insurance companies or casino owners, is to instead determine where the participant overvalues the gambles with respect to the expected value.

Noguchi, Sanborn, and Stewart (2013) dealt with this problem by introducing another agent into the algorithm. After a proposal was made by the computer, first the computer agent decided whether to accept the proposal with probability $\frac{g(x)}{g(x)+g(x^*)}$. Only if the computer agent accepted the proposal, did the participant have a chance to judge between the proposal and the state. If the participant also accepted the proposal, then the proposal became the new state on the next trial. This scheme fulfills detailed balance, and the target distribution becomes $f(x)g(x)$. This $f(x)g(x)$ distribution is essentially the agreement distribution between the participant and the agent and as a result, trials are concentrated on the items that have high agreement, as shown in Figure 3.

For the problem of determining where participants overvalue gambles with respect to expected value, the experimenter can set the computer agent's utility function to be equal to the inverse of the expected value. As a result, the distribution converges to gambles on which the participant's utility exceeds that of the expected value. Participants were run in this experiment and the results showed that participants overvalued gambles in which the probability of receiving £0 was near zero, showing an aversion to the worst outcome.

Conclusions and Future Work

The computer metaphor has provided much inspiration for psychology. The recent close links between computer science and cognitive psychology have led to even more productive developments, and here we examined the consequences of treating people as components in an algorithm usually executed on digital computers. Using people as components in a sampling algorithm has resulted in methods for efficiently exploring the representations that people have of naturalistic stimuli. The power of this approach is further demonstrated in extensions of the method, which are able to use other closely related algorithms from computer science to improve the performance of the method in for particular applications. In the future we hope to apply this method to even more interesting problems, such as eliciting information from experts and finding areas of agreement between multiple participants.

Markov chain Monte Carlo with people is just one instance of a deeper idea: we can use human beings as elements of algorithms that are more commonly executed by digital computers, and that we can learn something about human cognition as a result. The principle of treating human decisions as a sample from a particular probability distribution is one that can be generalized to make it possible to implement other randomized algorithms with people. We view this as an exciting possibility: there is a rich literature on randomized algorithms that might contain other important insights about new methods by which we can address the questions of cognitive psychology.

References

Ahumada, A. J., & Lovell, J. (1971). Stimulus features in signal detection. *Journal of the Acoustical Society of America*, *49*, 1751-1756.

Anderson, J. R., & Matessa, M. (1992). Explorations of an incremental Bayesian algorithm for categorization. *Machine Learning*, *9*, 275-308.

Ashby, F. G., & Gott, R. E. (1988). Decision rules in the perception and categorization of multidimensional stimuli. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *14*, 33-53.

Ashby, F. G., & Maddox, W. T. (1993). Relations between prototype, exemplar, and decision bound models of categorization. *Journal of Mathematical Psychology*, *37*, 372-400.

Atkinson, R. C., & Shiffrin, R. M. (1968). Human memory: A proposed system and its control processes. *The psychology of learning and motivation*, *2*, 89–195.

Barker, A. A. (1965). Monte Carlo calculations of the radial distribution functions for a proton-electron plasma. *Australian Journal of Physics*, *18*, 119-133.

Beard, B. L., & Ahumada, A. J. (1998). A technique to extract relevant image features for visual tasks. In *Proceedings of spie* (Vol. 3299, pp. 79–85).

Blundell, C., Sanborn, A. N., & Griffiths, T. L. (2012). Look-ahead Monte Carlo with people. In *Proceedings of the 34th annual conference of the cognitive science society* (p. 1356-1361).

Cohen, A., & Ross, M. (2009). Exploring mass perception with Markov chain Monte Carlo. *Journal of Experimental Psychology: Human Perception and Performance*, *35*, 1833-1844.

Dailey, M., Cottrell, G., & Reilly, J. (2001). California facial expressions (cafe). *Unpublished digital images, University of California, San Diego.*

Daw, N. D., Courville, A. C., & Dayan, P. (2008). Semi-rational models of conditioning: the case of trial order. In N. Chater & M. Oaksford (Eds.), *The probabilistic mind* (p. 431-452). Oxford, UK: Oxford University Press.

Duane, S., Kennedy, A. D., Pendleton, B. J., & Roweth, D. (1987). Hybrid monte carlo. *Physics letters B*, *195*(2), 216–222.

Edmonds, J. (1965). Paths, trees, and flowers. *Canadian Journal of mathematics*, *17*(3), 449–467.

Flinn, P., & McManus, G. (1961). Monte Carlo calculation of the order-disorder transformation in the body-centered cubic lattice. *Physical Review*, *124*(1), 54.

Gilks, W. R., Richardson, S., & Spiegelhalter, D. J. (Eds.). (1996). *Markov chain Monte Carlo in practice.* Suffolk: Chapman and Hall.

Gold, J., Cohen, A. L., & Shiffrin, R. M. (2006). Visual noise reveals category representations. *Psychonomic Bulletin and Review*, *13*, 649-655.

Gold, J., Murray, R., Bennett, P., & Sekuler, A. (2000). Deriving behavioural receptive fields for visually completed contours. *Current Biology*, *10*(11), 663-666.

Griffiths, T. L., & Ghahramani, Z. (2005). *Infinite latent feature models and the Indian buffet process* (Tech. Rep. No. 2005-001). Gatsby Computational Neuroscience Unit.

Griffiths, T. L., Vul, E., & Sanborn, A. N. (2012). Bridging levels of analysis for probabilistic models of cognition. *Current Directions in Psychological Science*, *21*, 263-268.

Hsu, A. S., Martin, J. B., Sanborn, A. N., & Griffiths, T. L. (2012). Identifying representations of categories of discrete items using Markov chain Monte Carlo with people. In *Proceedings of the 34th annual conference of the cognitive science society* (p. 485-490).

Huszár, F., Noppeney, U., & Lengyel, M. (2010). Mind reading by machine learning: A doubly Bayesian method for inferring mental representations. In *Proceedings of the 32nd annual conference of the cognitive science society* (p. 2810-2815).

Jebara, T., & Shchogolev, V. (2006). B-matching for spectral clustering. In *Machine learning: Ecml 2006* (pp. 679–686). Springer.

Kahneman, D., & Tversky, A. (1979). Prospect theory: An analysis of decision under risk. *Econometrica: Journal of the Econometric Society*, 263–291.

Kemeny, J. G., Babbitt, B., Haggerty, P. E., Lewis, C., Marks, P. A., Marrett, C. B., ... Trunk, A. D. (1979). *The need for change, the legacy of TMI: report of the President's commission on the accident at Three Mile Island.* US Government Printing Office.

Kontsevich, L. L., & Tyler, C. W. (2004). What makes mona lisa smile? *Vision research*, *44*(13), 1493–1498.

Luce, R. D. (1959). *Individual choice behavior: A theoretical analysis.* New York: Wiley.

Martin, J. B., Griffiths, T. L., & Sanborn, A. N. (2012). Testing the efficiency of markov chain monte carlo with people using facial affect categories. *Cognitive science*, *36*(1), 150–162.

McDuff, D. (2010). A human-Markov chain Monte Carlo method for investigating facial expression categorization. In *Proceedings of the 10th international conference on cognitive modeling* (pp. 151–156).

Metropolis, A. W., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, *21*, 1087-1092.

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.

Neri, P. (2004). Estimation of nonlinear psychophysical kernels. *Journal of Vision*, *4*(2).

Noguchi, T., Sanborn, A. N., & Stewart, N. (2013). Non-parametric estimation of the individual's utility map. In *Proceedings of the 35th annual conference of the cognitive science society.*

Norris, J. R. (1997). *Markov chains.* Cambridge, UK: Cambridge University Press.

Nosofsky, R. M. (1988). Exemplar-based accounts of relations between classifcation, recognition, and typicality. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *14*, 700-708.

Olman, C., & Kersten, D. (2004). Classification objects, ideal observers, and generative models. *Cognitive Science*, *28*, 227-239.

Papadimitriou, C., & Steiglitz, K. (1998). *Combinatorial optimization: Algorithms and complexity.* New York: Dover.

Raaijmakers, J. G. W., & Shiffrin, R. M. (1981). Search of associative memory. *Psychological Review*, *88*, 93-134.

Rosch, E. (1973). On the internal structure of perceptual and semantic categories. In T. E. Moore (Ed.), *Cognitive development and acquisition of language.* New York: Academic Press.

Sanborn, A. N., Griffiths, T. L., & Navarro, D. J. (2010). Rational approximations to the rational model of categorization. *Psychological Review*, *117*, 1144-1167.

Sanborn, A. N., Griffiths, T. L., & Shiffrin, R. M. (2010). Uncovering mental representations with Markov chain Monte Carlo. *Cognitive Psychology*, *60*, 63-106.

Sanborn, A. N., & Silva, R. (2013). Constraining bridges between levels of analysis: A computational justification for locally Bayesian learning. *Journal of Mathematical Psychology.*

Savage, L. J. (1954). *Foundations of statistics.* New York: John Wiley & Sons.

Schneider, W., & Shiffrin, R. M. (1977). Controlled and automatic human information processing: I. detection, search, and attention. *Psychological review*, *84*(1), 1.

Shepard, R. N. (1957). Stimulus and response generalization: A stochastic model relating generalization to distance in psychological space. *Psychometrika*, *22*, 325-345.

Shi, L., Griffiths, T. L., Feldman, N. H., & Sanborn, A. N. (2010). Exemplar models as a mechanism for performing Bayesian inference. *Psychological Bulletin and Review*, *17*, 443-464.

Shiffrin, R. M., & Schneider, W. (1977). Controlled and automatic human information processing: Ii. perceptual learning, automatic attending and a general theory. *Psychological review*, *84*(2), 127.

Shiffrin, R. M., & Steyvers, M. (1997). A model for recognition memory: REM: Retrieving Effectively from Memory. *Psychonomic Bulletin & Review*, *4*, 145-166.

Sternberg, S. (1966). High-speed scanning in human memory. *Science*, *153*(3736), 652–654.

Townsend, J. T., & Ashby, F. G. (1983). *The stochastic modeling of elementary psychological processes*. CUP Archive.

Victor, J. (2005). Analyzing receptive fields, classification images and functional images: challenges with opportunities for symmetry. *Nature Neuroscience*, *8*, 1651-1656.

Vulkan, N. (2000). An economist's perspective on probability matching. *Journal of Economic Surveys*, *14*, 101-118.