Running head: TECHNICAL INTRODUCTION

Technical Introduction:

A Primer on Probabilistic Inference

Thomas L. Griffiths

Department of Cognitive and Linguistic Sciences

Brown University

Alan Yuille

Department of Statistics

University of California, Los Angeles

Abstract

Research in computer science, engineering, mathematics, and statistics has produced a variety of tools that are useful in developing probabilistic models of human cognition. We provide an introduction to the principles of probabilistic inference that are used in the papers appearing in this special issue. We lay out the basic principles that underlie probabilistic models in detail, and then briefly survey some of the tools that can be used in applying these models to human cognition.

Technical Introduction:

A Primer on Probabilistic Inference

Introduction

Probabilistic models aim to explain human cognition by appealing to the principles of probability theory and statistics, which dictate how an agent should act in situations that involve uncertainty. While probability theory was originally developed as a means of analyzing games of chance, it was quickly realized that probabilities could be used as a guide to rational action [1, 2]. Probabilistic models are used in many disciplines, and are the method of choice for an enormous range of applications, including artificial systems for medical inference (e.g., [3]), bioinformatics (e.g., [4]), and computer vision (e.g., [5]). Applying probabilistic models to human cognition thus provides the opportunity to draw upon work in computer science, engineering, mathematics, and statistics, often revealing surprising connections.

Developing probabilistic models of cognition involves two challenges. The first challenge is specifying a suitable model. This requires considering the computational problem faced by an agent and the knowledge available to that agent [6, 7]. The second challenge is evaluating model predictions. Probabilistic models can capture the structure of extremely complex problems, but as the structure of the model becomes richer, probabilistic inference becomes harder. Being able to compute the relevant probabilities is a practical issue that arises when using probabilistic models, and also raises the question of how people might be able to make similar computations.

In this paper, we will introduce some of the tools that can be used to address these challenges. The plan of the paper is as follows. First, we introduce Bayesian inference,

which is at the heart of many probabilistic models. We then consider how to define structured probability distributions, introducing some of the key ideas behind *graphical models*, which can be used to represent the dependencies among a set of variables. Finally, we discuss two algorithms that are used to evaluate the predictions of probabilistic models: the *Expectation-Maximization (EM)* algorithm, and *Markov chain Monte Carlo (MCMC)*. Several books provide a more detailed discussion of these topics in the context of statistics [8, 9, 10], machine learning [11, 12, 13], and artificial intelligence [14, 15, 16].

Fundamentals of Bayesian inference

Probabilistic models of cognition are often referred to as Bayesian models, reflecting the central role that Bayesian inference plays in reasoning under uncertainty. In this section, we will introduce the basic ideas behind Bayesian inference, and discuss how it can be used in different contexts.

Basic Bayes

Bayesian inference is based upon a simple formula known as *Bayes' rule* [1]. Assume that we have an agent who is attempting to infer the process that was responsible for generating some data, d. Let h be a hypothesis about this process, and $P(h)$ indicate the probability that the agent would have ascribed to h being the true generating process, prior to seeing d (known as a *prior probability*). How should that agent go about changing his beliefs in the light of the evidence provided by d? To answer this question, we need a procedure for computing the *posterior probability*, $P(h|d)$.

Bayes' rule provides just such a procedure, defining the posterior probability to be

$$P(h|d) = \frac{P(d|h)P(h)}{P(d)}, \tag{1}$$

where the probability of the data given the hypothesis, $P(d|h)$, is known as the *likelihood*. The denominator is obtained by summing over hypotheses, a procedure known as

marginalization, to obtain

$$P(d) = \sum_{h' \in \mathcal{H}} P(d|h')P(h'), \tag{2}$$

where $\mathcal{H}$ is the set of all hypotheses considered by the agent, sometimes referred to as the *hypothesis space*. This formulation of Bayes' rule makes it apparent that the posterior probability of h is directly proportional to the product of the prior probability and the likelihood. The sum in the denominator simply ensures that the resulting probabilities are normalized to sum to one. Box 1 discusses how the resulting posterior distribution can be used as a guide to rational action.

Comparing two simple hypotheses

The setting in which Bayes' rule is usually introduced is the comparison of two simple hypotheses. For example, imagine that you are told that a box contains two coins: one that produces heads 50% of the time, and one that produces heads 90% of the time. You choose a coin, and then flip it ten times, producing the sequence `HHHHHHHHHH`. Which coin did you pick? What would you think if you had flipped `HHTHTHTTHT` instead?

To translate this problem into a Bayesian inference, we need to identify the hypothesis space, $\mathcal{H}$, the prior distribution, $P(h)$, and the likelihood, $P(d|h)$. We have two coins, and thus two hypotheses. If we use θ to denote the probability that a coin produces heads, then h_0 is the hypothesis that $\theta = 0.5$, and h_1 is the hypothesis that $\theta = 0.9$. Since we have no reason to believe that we would be more likely to pick one coin than the other, the prior probabilities are $P(h_0) = P(h_1) = 0.5$. The probability of a particular sequence of coinflips containing N_H heads and N_T tails being generated by a coin which produces heads with probability θ is

$$P(d|\theta) = \theta^{N_H}(1-\theta)^{N_T}. \tag{3}$$

The likelihoods associated with h_0 and h_1 can thus be obtained by substituting the appropriate value of θ into Equation 3.

We can place the priors and likelihoods defined above directly into Equation 1 to compute the posterior probability of each of our hypotheses. However, with just two hypotheses it is often easier to work with the *posterior odds*, the ratio of the posterior probabilities. If we use Bayes' rule to find the posterior probability of h_0 and h_1, it follows that the posterior odds in favor of h_1 are

$$\frac{P(h_1|d)}{P(h_0|d)} = \frac{P(d|h_1)}{P(d|h_0)} \frac{P(h_1)}{P(h_0)} \tag{4}$$

where we have used the fact that the denominator of Equation 1 is constant. The first and second terms on the right hand side are called the *likelihood ratio* and the *prior odds* respectively.

Equation 4 helps to clarify how prior knowledge and new data are combined in Bayesian inference. The two terms on the right hand side each express the influence of one of these factors: the prior odds are determined entirely by the prior beliefs of the agent, while the likelihood ratio expresses how these odds should be modified in light of the data d. Returning to our example, we can use Equation 4 to compute the posterior odds of our two hypotheses for any observed sequence of heads and tails. Using the priors and likelihood ratios from the previous paragraph gives odds of approximately 357:1 in favor of h_1 for the sequence `HHHHHHHHHH` and 165:1 in favor of h_0 for the sequence `HHTHTHTTHT`.

Comparing infinitely many hypotheses

The analysis outlined above for two simple hypotheses generalizes naturally to other finite (or countably infinite) sets. However, Bayesian inference can also be applied in contexts where there are (uncountably) infinitely many hypotheses to evaluate – a situation that arises surprisingly often. For example, imagine that rather than choosing between two alternatives for the probability that a coin produces heads, θ, we were willing to consider any value of θ between 0 and 1. What should we infer about the value of θ from a sequence such as `HHHHHHHHHH`?

In Bayesian statistics, inferring θ is treated just like any other Bayesian inference. If we assume that θ is a random variable, then we can apply Bayes' rule to obtain

$$p(\theta|d) = \frac{P(d|\theta)p(\theta)}{P(d)} \tag{5}$$

where

$$P(d) = \int_0^1 P(d|\theta)p(\theta)\, d\theta. \tag{6}$$

The key difference from Bayesian inference with a finite or countably infinite set of hypotheses is that the posterior distribution is now characterized by a *probability density*, which we indicate by using $p(\cdot)$ instead of $P(\cdot)$, and the sum over hypotheses becomes an integral.

The posterior distribution over θ contains more information than a single point estimate: it indicates not just which values of θ are probable, but also how much uncertainty there is about those values. Collapsing this distribution down to a single number discards information, so Bayesians prefer to maintain distributions wherever possible (this attitude is similar to Marr's "principle of least commitment" [6]). However, there are two methods that are commonly used to obtain a point estimate from a posterior distribution. The first method is *maximum a posteriori (MAP)* estimation: choosing the value of θ that maximizes the posterior probability, as given by Equation 5. The second method is computing the *posterior mean* of the quantity in question. For example, we could compute the posterior mean value of θ, which would be

$$\bar{\theta} = \int_0^1 \theta\, p(\theta|d)\, d\theta. \tag{7}$$

For the case of coinflipping, the posterior mean also corresponds to the *posterior predictive distribution*: the probability with which one should predict the next toss of the coin will produce heads.

Different choices of the prior, $p(\theta)$, will lead to different guesses at the value of θ. A first step might be to assume a *uniform* prior over θ, with $p(\theta)$ being equal for all values of

θ between 0 and 1. Using a little calculus, it is possible to show that the posterior distribution over θ produced by a sequence d with N_H heads and N_T tails is

$$p(\theta|d) = \frac{(N_H + N_T + 1)!}{N_H!\ N_T!}\ \theta^{N_H}(1-\theta)^{N_T}. \tag{8}$$

This is actually a distribution of a well known form, being a beta distribution with parameters $N_H + 1$ and $N_T + 1$, denoted Beta$(N_H + 1, N_T + 1)$ [17]. Using this prior, the MAP estimate for θ is $\frac{N_H}{N_H+N_T}$ (which is also the *maximum-likelihood* estimate, being the value of θ that maximizes $P(d|\theta)$), but the posterior mean is $\frac{N_H+1}{N_H+N_T+2}$. Thus, the posterior mean is sensitive to the fact that we might not want to put as much weight in a single head as a sequence of ten heads in a row: on seeing a single head, we should predict that the next toss will produce a head with probability $\frac{2}{3}$, while a sequence of ten heads should lead us to predict that the next toss will produce a head with probability $\frac{11}{12}$.

We can also use priors that encode stronger beliefs about the value of θ. For example, we can take a Beta$(V_H + 1, V_T + 1)$ distribution for $p(\theta)$, where V_H and V_T are greater than -1. This distribution has a mean at $\frac{V_H+1}{V_H+V_T+2}$, and gradually becomes more concentrated around that mean as $V_H + V_T$ becomes large. For instance, taking $V_H = V_T = 1000$ would give a distribution that strongly favors values of θ close to 0.5. Using such a prior, we obtain the posterior distribution

$$p(\theta|d) = \frac{(N_H + N_T + V_H + V_T + 1)!}{(N_H + V_H)!\ (N_T + V_T)!}\ \theta^{N_H+V_H}(1-\theta)^{N_T+V_T}, \tag{9}$$

which is Beta$(N_H + V_H + 1, N_T + V_T + 1)$. Under this posterior distribution, the MAP estimate of θ is $\frac{N_H+V_H}{N_H+N_T+V_H+V_T}$, and the posterior mean is $\frac{N_H+V_H+1}{N_H+N_T+V_H+V_T+2}$. Thus, if $V_H = V_T = 1000$, seeing a sequence of ten heads in a row would induce a posterior distribution over θ with a mean of $\frac{1011}{2012} \approx 0.5025$.

Some reflection upon the results in the previous paragraph yields two observations: first, that the prior and posterior are from the same family of distributions (both being beta distributions), and second, that the parameters of the prior, V_H and V_T, act as

"virtual examples" of heads and tails, which are simply combined with the real examples tallied in N_H and N_T to produce the posterior. These two properties are not accidental: they are characteristic of a class of priors called *conjugate priors* [9]. The likelihood determines whether a conjugate prior exists for a given problem, and the form that the prior will take. The results we have given in this section exploit the fact that the beta distribution is the conjugate prior for the Bernoulli or binomial likelihood (Equation 3) – the uniform distribution on $[0, 1]$ is also a beta distribution, being Beta$(1, 1)$. Conjugate priors exist for many distributions, including Gaussian, Poisson, and multinomial distributions, and greatly simplify many Bayesian calculations.

Comparing hypotheses that differ in complexity

Whether there were a finite number or not, the hypotheses that we have considered so far were relatively homogeneous, each offering a single value for the parameter θ characterizing our coin. However, many problems require comparing hypotheses that differ in complexity. For example, the problem of inferring whether a coin is fair or biased based upon an observed sequence of heads and tails requires comparing a hypothesis that gives a single value for θ – if the coin is fair, then $\theta = 0.5$ – with a hypothesis that allows θ to take on any value between 0 and 1.

Choosing between models that differ in their complexity is often called the problem of *model selection* [18, 19]. The Bayesian approach to model selection is a seamless application of the methods discussed so far. Hypotheses that differ in their complexity can be compared directly using Bayes' rule, once they are reduced to probability distributions over the observable data [20].

To illustrate this principle, assume that we have two hypotheses: h_0 is the hypothesis that $\theta = 0.5$, and h_1 is the hypothesis that θ takes a value drawn from a uniform distribution on $[0, 1]$. If we have no a priori reason to favor one hypothesis over

the other, we can take $P(h_0) = P(h_1) = 0.5$. The likelihood of the data under h_0 is straightforward to compute, using Equation 3, giving $P(d|h_0) = 0.5^{N_H+N_T}$. But how should we compute the likelihood of the data under h_1, which does not make a commitment to a single value of θ?

The solution to this problem is to compute the marginal probability of the data under h_1. Applying the principle of marginalization mentioned above, we can define the joint distribution over d and θ given h_1, and then integrate over θ to obtain

$$P(d|h_1) = \int_0^1 P(d|\theta, h_1)p(\theta|h_1)\, d\theta \tag{10}$$

where $p(\theta|h_1)$ is the distribution over θ assumed under h_1 – in this case, a uniform distribution over $[0, 1]$. This does not require any new concepts – it is exactly the same kind of computation as we needed to perform to compute the normalizing constant for the posterior distribution over θ (Equation 6). Performing this computation, we obtain $P(d|h_1) = \frac{N_H!\ N_T!}{(N_H+N_T+1)!}$, where again the fact that we have a conjugate prior provides us with a neat analytic result. Having computed this likelihood, we can apply Bayes' rule just as we did for two simple hypotheses. Figure 1 (a) shows how the log posterior odds in favor of h_1 change as N_H and N_T vary for sequences of length 10.

The ease with which hypotheses differing in complexity can be compared using Bayes' rule conceals the fact that this is a difficult problem. Complex hypotheses have more degrees of freedom that can be adapted to the data, and can thus always be made to fit better than simple hypotheses. For example, for any sequence of heads and tails, we can always find a value of θ that would give higher probability to that sequence than the hypothesis that $\theta = 0.5$. It seems like a complex hypothesis would thus have a big advantage over a simple hypothesis. The Bayesian solution to the problem of comparing hypotheses that differ in their complexity takes this into account. More degrees of freedom provide the opportunity to find a better fit to the data, but this greater flexibility also

makes a worse fit possible. For example, for d consisting of the sequence HHTHTTHHHT, $P(d|\theta, h_1)$ is greater than $P(d|h_0)$ for $\theta \in (0.5, 0.694]$, but is less than $P(d|h_0)$ outside that range. Marginalizing over θ averages these gains and losses: a more complex hypothesis will be favored only if its greater complexity consistently provides a better account of the data. This penalization of complex models is known as the "Bayesian Occam's razor" [21, 13], and is illustrated in Figure 1 (b).

Representing structured probability distributions

Probabilistic models go beyond "hypotheses" and "data". More generally, a probabilistic model defines the joint distribution for a set of random variables. For example, imagine that a friend of yours claims to possess psychic powers – in particular, the power of psychokinesis. He proposes to demonstrate these powers by flipping a coin, and influencing the outcome to produce heads. You suggest that a better test might be to see if he can levitate a pencil, since the coin producing heads could also be explained by having substituted a two-headed coin. We can express all possible outcomes of the proposed tests, as well as their causes, using the binary random variables X_1, X_2, X_3, and X_4 to represent (respectively) the truth of the coin being flipped and producing heads, the pencil levitating, your friend having psychic powers, and the use of a two-headed coin. Any set of of beliefs about these outcomes can be encoded in a joint probability distribution, $P(x_1, x_2, x_3, x_4)$. For example, the probability that the coin comes up heads ($x_1 = 1$) should be higher if your friend actually does have psychic powers ($x_3 = 1$).

Once we have defined a joint distribution on X_1, X_2, X_3, and X_4, we can reason about the implications of events involving these variables. For example, if flipping the coin produces heads ($x_1 = 1$), then the probability distribution over the remaining variables is

$$P(x_2, x_3, x_4|x_1 = 1) = \frac{P(x_1 = 1, x_2, x_3, x_4)}{P(x_1 = 1)}. \tag{11}$$

This equation can be interpreted as an application of Bayes' rule, with X_1 being the data,

and X_2, X_3, X_4 being the hypotheses. However, in this setting, as with most probabilistic models, any variable can act as data or hypothesis. In the general case, we use probabilistic inference to compute the probability distribution over a set of *unobserved* variables (here, X_2, X_3, X_4) conditioned on a set of *observed* variables (here, X_1).

While the rules of probability can, in principle, be used to define and reason about probabilistic models involving any number of variables, two factors can make large probabilistic models difficult to use. First, it is hard to simply write down a sensible joint distribution over a large set of variables. Second, the resources required to represent and reason about probability distributions increases exponentially in the number of variables involved. A probability distribution over four binary random variables requires $2^4 - 1 = 15$ numbers to specify, which might seem quite reasonable. If we double the number of random variables to eight, we would need to provide $2^8 - 1 = 255$ numbers to fully specify the joint distribution over those variables. Fortunately, research in statistics and computer science has produced a formal language for describing probability distributions that simplifies both representation and reasoning. This is the language of *graphical models*, in which the statistical dependencies that exist among a set of variables are represented graphically.

Directed graphical models

The most common kind of graphical models are *directed graphical models*, also known as *Bayesian networks* or *Bayes nets*, which consist of a set of nodes, representing random variables, together with a set of directed edges from one node to another [16]. Typically, nodes are drawn as circles, and the existence of a directed edge from one node to another is indicated with an arrow between the corresponding nodes. If an edge exists from node A to node B, then A is referred to as the "parent" of B, and B is the "child" of A. This genealogical relation is often extended to identify the "ancestors" and

"descendants" of a node. Bayes nets use acyclic graphs, meaning that no node can be its own ancestor.

The directed graph used in a Bayes net has one node for each random variable in the associated probability distribution. The edges express the statistical dependencies between the variables in a fashion consistent with the *Markov condition*: conditioned on its parents, each variable is independent of all other variables except its descendants [16, 22]. This has an important implication: a Bayes net specifies a canonical factorization of a probability distribution into the product of the conditional distribution for each variable conditioned on its parents. Thus, for a set of variables $X_1, X_2, \ldots, X_m$, we can write $P(x_1, x_2, \ldots, x_m) = \prod_i P(x_i|\mathrm{Pa}(X_i))$ where $\mathrm{Pa}(X_i)$ is the set of parents of X_i.

Figure 2 shows a Bayes net for the example of the friend who claims to have psychic powers. This Bayes net identifies a number of assumptions about the relationship between the variables involved in this situation. For example, X_1 and X_2 are assumed to be independent given X_3, indicating that once it was known whether or not your friend was psychic, the outcomes of the coin flip and the levitation experiments would be completely unrelated. By the Markov condition, we can write
$P(x_1, x_2, x_3, x_4) = P(x_1|x_3, x_4)P(x_2|x_3)P(x_3)P(x_4)$. This factorization allows us to use fewer numbers in specifying the distribution over these four variables: we only need one number for each variable, conditioned on each set of values taken on by its parents. In this case, this adds up to 8 numbers rather than 15. Furthermore, recognizing the structure in this probability distribution simplifies some of the computations we might want to perform. There are a number of specialized algorithms for efficient probabilistic inference in Bayes nets, which make use of the dependencies among variables [16, 14].

Undirected graphical models

Undirected graphical models, also known as Markov Random Fields (MRFs), consist of a set of nodes, representing random variables, and a set of undirected edges, defining a neighborhood structure on the graph which indicates the probabilistic dependencies of the variables at the nodes [16]. Each set of fully-connected neighbors is associated with a *potential* function, which varies as the associated random variables take on different values. When multiplied together, these potential functions give the probability distribution over all the variables. Unlike directed graphical models, there need be no simple relationship between these potentials and the local conditional probability distributions. These models are widely used in computer vision [5], and some kinds of artificial neural networks inspired by ideas from statistical physics can be interpreted as undirected graphical models [23].

Uses of graphical models

Different aspects of graphical models are emphasized in their use in the artificial intelligence and statistics communities. In the artificial intelligence community [15, 16, 14], the emphasis is on Bayes nets as a form of knowledge representation and an engine for probabilistic reasoning. Recently, research has begun to explore the use of graphical models for the representation of causal relationships (see Box 2). In statistics, graphical models tend to be used to clarify the dependencies among a set of variables, and to identify the *generative model* assumed by a particular analysis. A generative model is a step-by-step procedure by which a set of variables are assumed to take their values, defining a probability distribution over those variables. Any Bayes net specifies such a procedure: each variable without parents is sampled, then each successive variable is sampled conditioned on the values of its parents. By considering the process by which observable data are generated, it becomes possible to postulate that the structure

contained in those data is the result of underlying unobserved variables. The use of such *latent variables* is extremely common in probabilistic models (see Box 3), and is at the heart of probabilistic models of language (see Box 4).

Algorithms for inference

The presence of latent variables in a model poses two problems: inferring the values of the latent variables conditioned on observed data, and learning the probability distribution characterizing both the observable and latent variables. In the probabilistic framework, both these forms of inference reduce to inferring the values of unknown variables, conditioned on known variables. This is conceptually straightforward, but the computations involved are difficult and can require complex algorithms.

A standard approach to solving the problem of estimating probability distributions involving latent variables is the Expectation-Maximization (EM) algorithm [24]. Imagine we have a model for data $\mathbf{x}$ that has parameters θ, and latent variables $\mathbf{z}$. A mixture model is one example of such a model (see Box 3). The likelihood for this model is $P(\mathbf{x}|\theta) = \sum_{\mathbf{z}} P(\mathbf{x}, \mathbf{z}|\theta)$ where the latent variables $\mathbf{z}$ are unknown. The EM algorithm is a procedure for obtaining a maximum-likelihood (or MAP) estimate for θ, without resorting to generic methods such as differentiating $\log P(\mathbf{x}|\theta)$. The key idea is that if we knew the values of the latent variables $\mathbf{z}$, then we could find θ by using the standard methods for estimation discussed above. Even though we might not have perfect knowledge of $\mathbf{z}$, we can still assign probabilities to $\mathbf{z}$ based on $\mathbf{x}$ and our current guess of θ, $P(\mathbf{z}|\mathbf{x}, \theta)$. The EM algorithm for maximum-likelihood estimation proceeds by repeatedly alternating between two steps: evaluating the expectation of the "complete log-likelihood" $\log P(\mathbf{x}, \mathbf{z}|\theta)$ with respect to $P(\mathbf{z}|\mathbf{x}, \theta)$ (the E-step), and maximizing the resulting quantity with respect to θ (the M-step). This algorithm is guaranteed to converge to a *local* maximum of $P(\mathbf{x}|\theta)$ [24], and both steps can be interpreted as performing hillclimbing on a single "free energy"

function [25]. An illustration of EM for a mixture of Gaussians appears in Figure 3 (a).

Another class of algorithms, Markov chain Monte Carlo (MCMC) methods, provide a means of obtaining samples from complex distributions, which can be used both for inferring the values of latent variables and for identifying the parameters of models. These algorithms were originally developed to solve problems in statistical physics [26], and are now widely used both in physics [27] and in statistics [28, 13, 29]. As the name suggests, Markov chain Monte Carlo is based upon the theory of Markov chains – sequences of random variables in which each variable is independent of all of its predecessors given the variable that immediately precedes it [30]. One well known property of Markov chains is their tendency to converge to a *stationary distribution*: as the length of a Markov chain increases, the probability that a variable in that chain takes on a particular value converges to a fixed quantity. In MCMC, a Markov chain is constructed such that its stationary distribution is the distribution from which we want to generate samples. For example, we could construct a Markov chain that would converge to the posterior distribution over the parameters of a mixture of Gaussians given the data. The states of the Markov chain would be the parameter values, and could be used like samples from the posterior distribution once the Markov chain converged. There are a variety of standard MCMC algorithms, including *Gibbs sampling* [5], in which the Markov chain moves between states by drawing each of a set of variables from its distribution when conditioned on the values of all other variables, and the more general *Metropolis-Hastings algorithm* [26, 31]. The results of applying Gibbs sampling to a mixture of Gaussians are shown in Figure 3 (b).

Conclusion

Probabilistic models provide a unique opportunity to develop a rational account of human cognition that combines statistical learning with structured representations.

However, specifying and using these models can be challenging. The widespread use of probabilistic models in computer science and statistics has led to the development of valuable tools that address some of these challenges. Graphical models provide a simple and intuitive way of expressing a probability distribution, making clear the assumptions about how a set of variables are related, and can greatly speed up probabilistic inference. The EM algorithm and Markov chain Monte Carlo can be used to estimate the parameters of models that incorporate latent variables, and to work with complicated probability distributions of the kind that often arise in Bayesian inference. By using these tools and continuing to draw on advances in the many disciplines where probabilistic models are used, it may ultimately become possible to define models that can capture some of the complexity of human cognition.

References

[1] T. Bayes. Studies in the history of probability and statistics: IX. Thomas Bayes's Essay towards solving a problem in the doctrine of chances. *Biometrika*, 45:296–315, 1763/1958.

[2] P. S. Laplace. *A philosophical essay on probabilities*. Dover, New York, 1795/1951.

[3] M. Shwe, B. Middleton, D. Heckerman, M. Henrion, E. Horvitz, H. Lehmann, and G. Cooper. Probabilistic diagnosis using a reformulation of the INTERNIST-1/QMR knowledge base I. The probabilistic model and inference algorithms. *Methods of Information in Medicine*, 30:241–255, 1991.

[4] J. K. Pritchard, M. Stephens, and P. Donnelly. Inference of population structure using multilocus genotype data. *Genetics*, 155:945–955, 2000.

[5] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[6] D. Marr. *Vision*. W. H. Freeman, San Francisco, CA, 1982.

[7] J. R. Anderson. *The adaptive character of thought*. Erlbaum, Hillsdale, NJ, 1990.

[8] J. O. Berger. *Statistical decision theory and Bayesian analysis*. Springer, New York, 1993.

[9] J. M. Bernardo and A. F. M. Smith. *Bayesian theory*. Wiley, New York, 1994.

[10] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian data analysis*. Chapman & Hall, New York, 1995.

[11] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification*. Wiley, New York, 2000.

[12] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: Data mining, inference, and prediction*. Springer, New York, 2001.

[13] D. J. C. Mackay. *Information theory, inference, and learning algorithms*. Cambridge University Press, Cambridge, 2003.

[14] S. J. Russell and P. Norvig. *Artificial intelligence: A modern approach*. Prentice Hall, Englewood Cliffs, NJ, 2nd edition, 2002.

[15] K. Korb and A. Nicholson. *Bayesian artificial intelligence*. Chapman and Hall/CRC, Boca Raton, FL, 2003.

[16] J. Pearl. *Probabilistic reasoning in intelligent systems*. Morgan Kaufmann, San Francisco, CA, 1988.

[17] J. Pitman. *Probability*. Springer-Verlag, New York, 1993.

[18] I. J. Myung and Mark A. Pitt. Applying Occam's razor in modeling cognition: A Bayesian approach. *Psychonomic Bulletin and Review*, 4:79–95, 1997.

[19] I. J. Myung, M. R. Forster, and M. W. Browne. Model selection [special issue]. *Journal of Mathematical Psychology*, 44, 2000.

[20] R. E. Kass and A. E. Rafferty. Bayes factors. *Journal of the American Statistical Association*, 90:773–795, 1995.

[21] W. H. Jeffreys and J. O. Berger. Ockham's razor and Bayesian analysis. *American Scientist*, 80(1):64–72, 1992.

[22] P. Spirtes, C. Glymour, and R. Schienes. *Causation prediction and search*. Springer-Verlag, New York, 1993.

[23] G. E. Hinton and T. J. Sejnowski. Learning and relearning in Boltzmann machines. In D. E. Rumelhart and J. L. McClelland, editors, *Parallel distributed processing: Explorations in the microstructure of cognition*, volume 1. MIT Press, Cambridge, MA, 1986.

[24] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, B*, 39, 1977.

[25] R. M. Neal and G. E. Hinton. A view EM algorithm that justifies incremental, sparse, and other variants. In M. I. Jordan, editor, *Learning in graphical models*. MIT Press, Cambridge, MA, 1998.

[26] A. W. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.

[27] M. E. J. Newman and G. T. Barkema. *Monte Carlo methods in statistical physics*. Clarendon Press, Oxford, 1999.

[28] W.R. Gilks, S. Richardson, and D. J. Spiegelhalter, editors. *Markov Chain Monte Carlo in Practice*. Chapman and Hall, Suffolk, 1996.

[29] R. M. Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, University of Toronto, 1993.

[30] J. R. Norris. *Markov Chains*. Cambridge University Press, Cambridge, UK, 1997.

[31] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109, 1970.

[32] D. P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, Nashua, NH, 2000.

[33] J. Pearl. *Causality: Models, reasoning and inference*. Cambridge University Press, Cambridge, UK, 2000.

[34] S. Sloman. *Causal models: How people think about the world and its alternatives.* Oxford University Press, Oxford, 2005.

[35] C. Glymour. *The mind's arrows: Bayes nets and graphical causal models in psychology*. MIT Press, Cambridge, MA, 2001.

[36] T. L. Griffiths and J. B. Tenenbaum. Theory-based causal induction, in prep.

[37] D. Heckerman. A tutorial on learning with Bayesian networks. In Michael I. Jordan, editor, *Learning in Graphical Models*, pages 301–354. MIT Press, Cambridge, MA, 1998.

[38] C. Glymour and G. Cooper. *Computation, Causation, and Discovery*. MIT Press, Cambridge, MA, 1999.

[39] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77:257–286, 1989.

[40] C. Manning and H. Schütze. *Foundations of statistical natural language processing*. MIT Press, Cambridge, MA, 1999.

Box 1: Decision theory and control theory

Bayesian decision theory introduces a loss function $L(h, \alpha(d))$ for the cost of making a decision $\alpha(d)$ when the input is d and the true hypothesis is h. It proposes selecting the decision rule $\alpha^*(.)$ that minimizes the risk, or expected loss, function:

$$R(\alpha) = \sum_{h,d} L(h, \alpha(d)) P(h, d). \tag{12}$$

This is the basis for rational decision making [8].

Usually the loss function is choosen so that the same penalty is paid for all wrong decisions: $L(h, \alpha(d)) = 1$ if $\alpha(d) \neq h$ and $L(h, \alpha(d)) = 0$ if $\alpha(d) = h$. Then the best decision rule is the maximum a posteriori (MAP) estimator $\alpha^*(d) = \arg\max P(h|d)$. Alternatively, if the loss function is the square of the error $L(h, \alpha(d)) = \{h - \alpha(d)\}^2$ then the best decision rule is the posterior mean $\sum_h hP(h|d)$.

In many situations, we will not know the distribution $P(h, d)$ exactly but will instead have a set of labelled samples $\{(h_i, d_i) : i = 1, ..., N\}$. The risk (Equation 12) can be approximated by the empirical risk $R_{emp}(\alpha) = (1/N) \sum_{i=1}^{N} L(h_i, \alpha(d_i))$. Some methods used in machine learning, such as neural networks and support vector machines, attempt to learn the decision rule directly by minimizing $R_{emp}(\alpha)$ instead of trying to model $P(h, d)$ [11, 12].

The Bayes risk (Equation 12) can be extended to dynamical systems where decisions need to be made over time. This leads to optimal control theory [32] where the goal is to minimize a cost functional to obtain a control law (analogous to the Bayes risk and the decision rule respectively). In this case, the notation is changed to use a control variable u to replace the decision variable α. Optimal control theory lays the groundwork for theories of animal learning and motor control.

Box 2: Causal graphical models

Causal graphical models augment standard directed graphical models with a stronger assumption about the relationship indicated by an edge between two nodes: rather than indicating statistical dependency, such an edge is assumed to indicate a direct causal relationship [33, 22]. This assumption allows causal graphical models to represent not just the probabilities of events that one might observe, but also the probabilities of events that one can produce through intervening on a system. The implications of an event can differ strongly, depending on whether it was the result of observation or intervention. For example, *observing* that nothing happened when your friend attempted to levitate a pencil would provide evidence against his claim of having psychic powers; *intervening* to hold the pencil down, and thus guaranteeing that it did not move during his attempted act of levitation, would remove any opportunity for this event to provide such evidence.

In causal graphical models, the consequences of intervening on a particular variable are be assessed by removing all incoming edges to the variable that was intervened on, and performing probabilistic inference in the resulting "mutilated" model [33]. This procedure produces results that align with our intuitions in the psychic powers example: intervening on X_2 breaks its connection with X_3, rendering the two variables independent. As a consequence, X_2 cannot provide evidence as to the value of X_3. Good summaries exist for both the psychological [34, 35, 36] and technical uses of graphical models [37, 38].

Box 3: Latent variables and mixture models

In many unsupervised learning problems, the observable data are believed to reflect some kind of underlying latent structure. For example, in a clustering problem, we might only see the location of each point, but believe that each point was generated from one of a small number of clusters. Associating the observed data with random variables X_i and the latent variables with random variables Z_i, we might want to define a probabilistic model for X_i that explicitly takes into account the latent structure Z_i. Such a model can ultimately be used to make inferences about the latent structure associated with new datapoints, as well as providing a more accurate model of the distribution of X_i.

A common example of a latent variable model is a *mixture model*, in which the distribution of X_i is assumed to be a mixture of several other distributions. For example, in the case of clustering, we might believe that our data were generated from two clusters, each associated with a different Gaussian (i.e. normal) distribution. If we let z_i denote the cluster from which the datapoint x_i was generated, and assume that there are K such clusters, then the probability distribution over x_i is

$$P(x_i) = \sum_{k=1}^{K} P(x_i|z_i = k)P(z_i = k) \tag{13}$$

where $P(x_i|z_i = k)$ is the distribution associated with cluster k, and $P(z_i = k)$ is the probability that a point would be generated from that cluster. If we can estimate the parameters that characterize these distributions, we can infer the probable cluster membership (z_i) for any datapoint (x_i). An example of a Gaussian mixture model (also known as a mixture of Gaussians) appears in Figure 3.

Box 4: Latent variable models for language

Latent variables are particularly useful in models of language, where they can be used to capture some of the unobserved structure that governs the words appearing in sentences. Hidden Markov models (HMMs) are one class of latent variable models that have been used for problems such as speech and language processing [39]. In an HMM, each word W_i is assumed to be generated from a latent class Z_i, where Z_i is chosen from a distribution that depends on the class of the previous word, Z_{i-1}. The classes thus form a Markov chain, which is "hidden" because only the words are observable. The graphical model for an HMM is shown in Figure 4.

A probabilistic context free grammar (PCFG) is a context-free grammar that associates a probability with each of its production rules [40]. The joint probability of a sentence and its parse tree is the product of the probabilities of the production rules used to generate the sentence. For example, we can define a PCFG as follows (see Figure 4). The non-terminal nodes $S, NP, VP, AT, NNS, VBD, PP, IN, DT, NN$ where S is a sentence, VP is a verb phrase, VBD is a verb,NP is a noun phrase, NN is a noun, and so on [40]. The terminal nodes are words from a dictionary (e.g. "the", "cat", "sat", "on", "mat".) We then define production rules which are applied to non-terminal nodes to generate child nodes (e.g. $S \mapsto NP, VP$ or $NN \mapsto$ "cat"), and specify probabilities for these production rules. The production rules enable us to generate a sentence starting from the root node S. We sample a production rule starting with S and apply it to generate child nodes. We repeat this process on the child nodes and stop when all the nodes are terminal (i.e. words). To parse a sentence, we compute the most probable way the sentence could have been generated by the production rules. We can learn the production rules, and their probabilities, directly from sentences. This can be done in a supervised way, where the correct parses are known, or in an unsupervised way, as described in the article by Manning and Chater in this issue.

Figure Captions

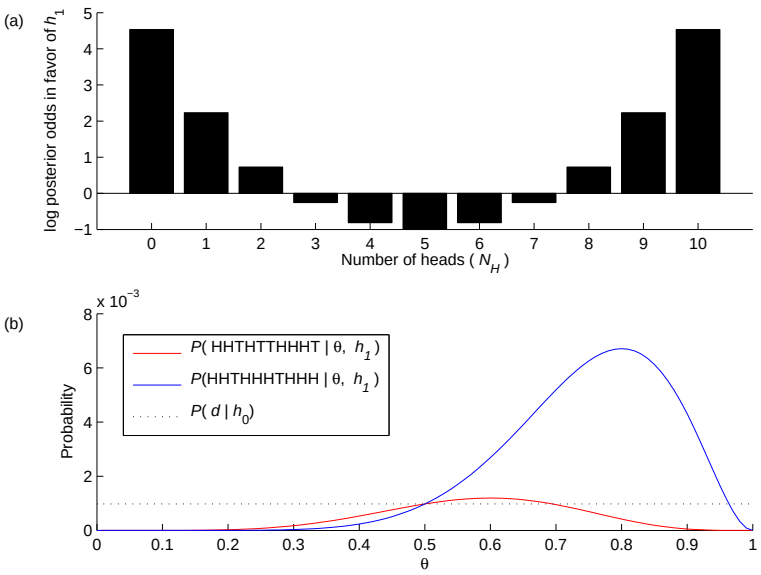
Figure 1. Comparing hypotheses about the weight of a coin. (a) The vertical axis shows log posterior odds in favor of h_1, the hypothesis that the probability of heads (θ) is drawn from a uniform distribution on $[0, 1]$, over h_0, the hypothesis that the probability of heads is 0.5. The horizontal axis shows the number of heads, N_H, in a sequence of 10 flips. As N_H deviates from 5, the posterior odds in favor of h_1 increase. (b) The posterior odds shown in (a) are computed by averaging over the values of θ with respect to the prior, $p(\theta)$. This averaging takes into account the fact that hypotheses with greater flexibility can produce both better and worse predictions. The red line shows the probability of the sequence `HHTHTTHHHT` for different values of θ, while the dotted line is the probabiltiy of any sequence of length 10 under h_0. On average, the greater flexibility of h_1 results in lower probabilities, so h_0 is favored over h_1. In contrast, a wide range of values of θ result in higher probability for for the sequence `HHTHHHTHHH`, as shown by the blue line, and h_1 is favored over h_0.

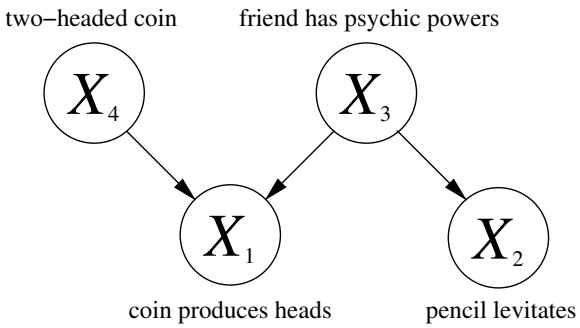
Figure 2. Directed graphical model (Bayes net) showing the dependencies among variables in the "psychic friend" example discussed in the text. X_1, X_2, X_3, and X_4 are random variables characterizing possible events that could occur, and the arrows between them reflect statistical dependencies.

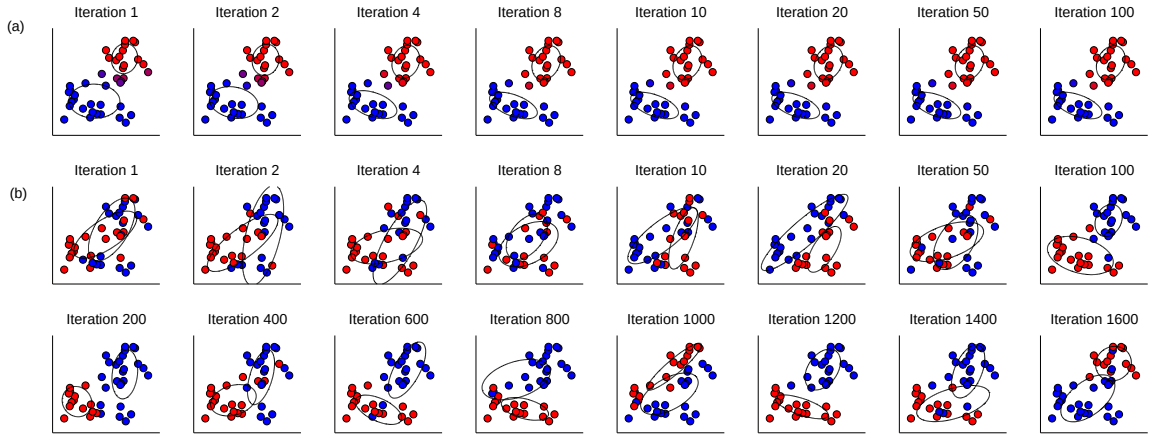
Figure 3. Expectation-Maximization (EM) and Markov chain Monte Carlo (MCMC) algorithms applied to a Gaussian mixture model with two clusters. Colors indicate the assignment of points to clusters (red and blue), with intermediate purples representing probabilistic assignments. The ellipses are a single probability contour for the Gaussian distributions reflecting the current parameter values for the two clusters. (a) The EM algorithm assigns datapoints probabilistically to the two clusters, and converges to a single solution that is guaranteed to be a local maximum of the log-likelihood. (b) In contrast,

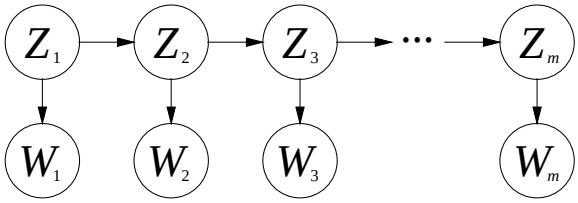
an MCMC algorithm samples cluster assignments, so each datapoint is assigned to a cluster at each iteration, and samples parameter values conditioned on those assignments. This process converges to the posterior distribution over cluster assignments and parameters: more iterations simply result in more samples from this posterior distribution.

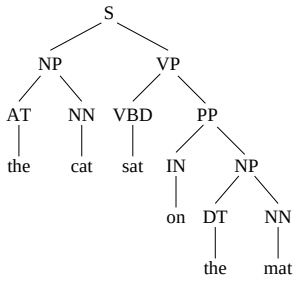
Figure 4. The left panel shows the graphical model for a hidden Markov model, the right is a parse tree from a probabilistic context-free grammar.


(a)
log posterior odds in favor of h_1
5
4
3
2
1
0
−1
0
1
2
3
4
5
6
7
8
9
10
Number of heads (N_H)
(b)
x 10^{-3}
Probability
8
6
4
2
0
P(HHTHTTHHHT | θ, h_1)
P(HHTHHHTHHH | θ, h_1)
P(d | h_0)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
θ


two−headed coin
friend has psychic powers
X_4
X_3
X_1
X_2
coin produces heads
pencil levitates


(a)
Iteration 1
Iteration 2
Iteration 4
Iteration 8
Iteration 10
Iteration 20
Iteration 50
Iteration 100
(b)
Iteration 1
Iteration 2
Iteration 4
Iteration 8
Iteration 10
Iteration 20
Iteration 50
Iteration 100
Iteration 200
Iteration 400
Iteration 600
Iteration 800
Iteration 1000
Iteration 1200
Iteration 1400
Iteration 1600


Z_1
Z_2
Z_3
Z_m
W_1
W_2
W_3
W_m


S
NP
VP
AT
NN
VBD
PP
the
cat
sat
IN
NP
on
DT
NN
the
mat