Bayesian nonparametrics and the probabilistic approach to modelling

Zoubin Ghahramani

Department of Engineering
University of Cambridge, UK
zoubin@eng.cam.ac.uk
http://mlg.eng.cam.ac.uk/zoubin

Modelling is fundamental to many fields of science and engineering. A model can be thought of as a representation of possible data one could predict from a system. The probabilistic approach to modelling uses probability theory to express all aspects of uncertainty in the model. The probabilistic approach is synonymous with Bayesian modelling, which simply uses the rules of probability theory in order to make predictions, compare alternative models, and learn model parameters and structure from data. This simple and elegant framework is most powerful when coupled with flexible probabilistic models. Flexibility is achieved through the use of Bayesian nonparametrics. This article provides an overview of probabilistic modelling and an accessible survey of some of the main tools in Bayesian nonparametrics. The survey covers the use of Bayesian nonparametrics for modelling unknown functions, density estimation, clustering, time series modelling, and representing sparsity, hierarchies, and covariance structure. More specifically it gives brief non-technical overviews of Gaussian processes, Dirichlet processes, infinite hidden Markov models, Indian buffet processes, Kingman's coalescent, Dirichlet diffusion trees, and Wishart processes.

Key words: probabilistic modelling; Bayesian statistics; nonparametrics; machine learning.

1. Introduction

Modelling is central to the sciences. Models allow one to make predictions, to understand phenomena, and to quantify, compare and falsify hypotheses.

Modelling is also at the core of intelligence. Both artificial and biological systems that exhibit intelligence must be able to make predictions, anticipate outcomes of their actions, and update their ability to make predictions in light of new data. It is hard to imagine how a system could do this without building models of the environment that the system interacts with. It is thus not surprising that many theories in cognitive science are based around the idea of building internal models (Wolpert et al., 1995; Knill and Richards, 1996; Griffiths and Tenenbaum, 2006).

A model is simply a compact representation of possible data one could observe. As such it may be more interpretable than the observed data itself, providing a useful representation of data. A model must be able to make forecasts of possible future data, otherwise it seems impossible to falsify a model in light of new data.

Phil. Trans. R. Soc. A 1–27; doi: 10.1098/rspa.00000000

I will use the term *forecast* in a very general way to refer to the process of making any claims about unobserved data on the basis of observed data; I will also use *predict* interchangeably with forecast. For all nontrivial phenomena, forecasts have to include some representation of the forecasting uncertainty. Deterministic forecasts (e.g. tomorrow's high temperature *will be* 17 degrees C) are too brittle and therefore easy to falsify.

Ideally, a model should also be adaptive. By *adaptive* I mean that the forecasts of the model should change depending on the data observed so far. Such adaptation, or learning, should hopefully have the effect of making the model's forecasts be better aligned with actual data. For example, if the forecasts are in the form of probability distributions, adaptation should have the effect that the forecast probability assigned to what actually happens should increase after the model has seen more data, although this cannot be generally guaranteed.

It is clear that all forecasts need to represent uncertainty. Observable data is generally corrupted by noise in the measurement process, and this noise needs to be incorporated in the forecast uncertainty. But uncertainty lurks at many other levels in any model. The amount of noise in the measurement process may itself be unknown. The model may have a number of parameters which are unknown. The structure of the model itself may be uncertain, or there may be multiple plausible competing models for the data. The forecasting system should ideally produce forecasts that incorporate all reasonable sources of uncertainty. As we will see, the Bayesian framework provides a natural and coherent approach for representing and manipulating all forms of uncertainty in modelling.

2. The Bayesian framework

The fundamental idea in Bayesian modelling is to use the mathematics of probability theory to represent and manipulate all forms of uncertainty in the model. This is a surprisingly simple yet powerful idea.

The good news is that there are only two rules of probability theory one needs to remember, the sum rule and the product rule.[1] Consider a pair of random variables x and y taking on values in some spaces $\mathcal{X}$ and $\mathcal{Y}$ respectively. The sum rule states that if I know the joint probability of two random variables, x and y, I can obtain the marginal probability of x by summing over all possible values of y,

$$P(x) = \sum_{y \in \mathcal{Y}} P(x, y).$$

If y is continuous we simply replace the sum with an integral. If I have a model for the joint probability distribution of the high temperature in London and Cambridge I can obtain the marginal distribution for the high temperature in Cambridge summing out London's temperature.

The product rule states that the joint probability of x and y can be decomposed into the product of the marginal probability of x and the conditional

[1] With apologies to probabilists and statisticians my presentation and notation will eschew formality in favour of didactic clarity.

probability of y given x (or the other way around):

$$P(x,y) = P(x)P(y|x) = P(y)P(x|y).$$

Combining the sum and product rule and rearranging a bit we obtain Bayes rule as a corollary:

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)} = \frac{P(x|y)P(y)}{\sum_{y' \in \mathcal{Y}} P(x|y')P(y')}$$

Let us now use these equations in a canonical modelling context. We will use $\mathcal{D}$ to represent the observable data. Often we talk about the data being a *set* consisting of a number of "data points" or measurements, $\mathcal{D} = \{x_1, \ldots, x_N\}$ but this is in no way a necessary assumption. For example, the data could be a single image, an observed graph, or an ordered *sequence* of measurements rather than a set. Our model will be indexed by m and we may want to consider multiple alternative models, e.g. m, m' etc. Each model usually has a number of free parameters, which we will denote with θ which can be a vector if needed.

First we need to make sure that the model m is well defined, in the sense that it can predict or forecast data. As previously discussed we use probability theory to represent the model forecasts. For any given setting of the model parameters, the model must be able to produce a forecast of the form

$$P(\mathcal{D}|\theta, m).$$

This probability of the data as a function of the parameters is the *likelihood* of the parameters. With the likelihood, for any given setting of the parameters we are in the position of making forecasts. However, the model m is not fully defined until we specify a "range" for the parameter values, θ. A linear regression model which states that the slope can take values between -1 and $+1$ is a very different model from one which specifies that the slope can take values between -100 and $+100$. In fact, to fully specify a model, we need to do a bit more than specify the range of parameters, we need to define a distribution over this range. Only then will our model m be able to make forecasts. We see this by writing the forecast probability of the model using the sum and product rules:

$$P(\mathcal{D}|m) \overset{\text{sum}}{=} \int P(\mathcal{D}, \theta|m) d\theta \overset{\text{prod}}{=} \int P(\mathcal{D}|\theta, m) P(\theta|m) d\theta \tag{2.1}$$

The expression $P(\mathcal{D}|m)$ is variously called the *marginal likelihood*, *model evidence*, or *integrated likelihood*. The *prior* over the parameters, $P(\theta|m)$, plays the role of specifying the range as described above (for example it could be uniform on $[-1, +1]$) in the form of a distribution over the allowable values of the parameters. Without a prior our model is not well-defined: we cannot generate or forecast data until we know how to choose values for θ.[2] Once the prior and likelihood are defined, and only then, is the model m fully specified in the sense that it can *generate possible data sets*.

People often object to Bayesian methods on the basis that it forces one to define priors on the parameters. This, in my opinion, is completely misguided.

[2] Of course, our model may have a fixed value of θ, e.g. 7.213, which corresponds to a delta function prior.

All models make assumptions; without assumptions it is impossible to make any forecasts or predictions from observed data. The first stage of the Bayesian modelling framework is to explicitly state all assumptions using the language of probability theory. Specifying both the prior and likelihood is a necessary requirement so that the model is well defined. In fact, the distinction between prior and likelihood is arbitrary[3]; both are essential parts of the *model*.

People also object to the use of priors on the parameters on the grounds that they don't think of the parameters as being "random" variables. For example, if one is estimating the mass of a planet from astronomical data, the mass of the planet is not "random" in the colloquial sense of the word.[4] This is a misunderstanding of the semantics of probabilities in Bayesian modelling. Probabilities are used to represent our *uncertainty* about unknown quantities. They are just as good at modelling uncertainty for repeatable experiments such as the outcome of a roll of a die, as they are for modelling uncertainty in the mass of a planet. Incidentally, both forms of uncertainty are fundamentally subjective; one's uncertainty about a roll of a die depends on one's knowledge of the exact initial conditions of the roll, in just the same way as the uncertainty about the mass of the planet depends on knowledge of the observational data on the planet's orbit.

Finally, people trained in the sciences are uncomfortable with the very notion of subjectivity in data analysis and modelling. This again is deeply misguided: all models involve assumptions, and all conclusions drawn from data analysis are conditional on assumptions. The probabilistic framework *forces* the scientist to be completely transparent about the assumptions, by expressing all assumptions as distributions on unknown quantities. These assumptions can then be easily contested, and the same data can be reanalysed under different modelling assumptions (and priors). The fact that conclusions could change depending on the assumptions is essential to good scientific practice. Fortunately, given enough data, the effect of the prior is generally overcome by the likelihood and posterior conclusions will converge (Doob, 1949; Le Cam, 1986; Van der Vaart, 2000). This is directly analogous to the progress of science, where of many possible hypotheses only the ones consistent with the data will survive. Bayesian modelling is *subjective* but not *arbitrary*: given a full specification of the model, and the data, there is only one way to reason about any quantity of interest.

Modelling thus becomes a very simple procedure:

- write down your assumptions (possible models, parameters, noise processes, etc), representing all forms of uncertainty using the language of probability theory

[3] Consider for example a model which defines the probability of the data (i.e. the likelihood) using a student t-distribution. For a single data point, this can be equivalently expressed as a model with a Gaussian likelihood and a Gamma prior on the precision of that Gaussian.

[4] I don't like the term 'random variable' because it suggest that there is some true source of randomness driving some process. To give another example, people would naturally and understandably flinch at the idea of calling something like the millionth digit of π or the number of people alive today a random variable. The notion that Bayesians consider true parameters to be "random variables" makes the framework less intuitive that it actually is. It is much more natural to use the term *uncertain*. Clearly I can be uncertain about the million'th digit of π even though its value is not random at all. My uncertainty about this digit will change after I look it up on the web, but presumably will not go to zero as there is always the chance of someone having made a mistake somewhere.

- given the data, use probability theory to make inferences about any unknown quantities in your model, or to make predictions from the model.

This process lends itself very naturally to a sequential processing of data. Your posterior after observing some data $\mathcal{D}_{\text{old}}$, $P(\theta|\mathcal{D}_{\text{old}}, m)$ is your prior before observing new data, $\mathcal{D}_{\text{new}}$:

$$P(\theta|\mathcal{D}_{\text{new}}, \mathcal{D}_{\text{old}}, m) \propto P(\mathcal{D}_{\text{new}}|\mathcal{D}_{\text{old}}, \theta, m)P(\theta|\mathcal{D}_{\text{old}}, m).$$

The sum and product rule also tell us how to make predictions from a model. Consider predicting some unknown quantity x (e.g. the next data point) given observed data $\mathcal{D}$ and model m:

$$P(x|\mathcal{D}, m) = \int P(x|\theta, \mathcal{D}, m)P(\theta|\mathcal{D}, m)d\theta. \tag{2.2}$$

This is a very intuitively satisfying expression which tells us that our predictions or forecasts are a weighted average of the forecasts from different parameter values, weighted by the posterior probability of each parameter value given the data observed so far. For parametric models, this simplifies since *given* the parameters forecasts are independent of the observed data: $P(x|\theta, \mathcal{D}, m) = P(x|\theta, m)$. We will revisit this point as we discuss parametric vs nonparametric models in Section 3. If we are considering a number of models m, m', etc, then by the sum and product rules, our forecasts are an average over models weighted by their posteriors.

The probabilistic modelling framework also provides intuitive answers to problems in model comparison. Assuming a set of competing probabilistic models $\mathcal{M}$, given some observed data we can evaluate the posterior probability of a particular model m,

$$P(m|\mathcal{D}) = \frac{P(\mathcal{D}|m)P(m)}{P(\mathcal{D})} = \frac{P(\mathcal{D}|m)P(m)}{\sum_{m' \in \mathcal{M}} P(\mathcal{D}|m')P(m')}.$$

Note the prominent role played by the marginal likelihood, $P(\mathcal{D}|m)$.

Importantly, this marginal likelihood captures a preference for simpler models known as Bayesian Occam's Razor (Jefferys and Berger, 1992; MacKay, 2003; Rasmussen and Ghahramani, 2001). Consider a set of nested models, for example different order polynomials (e.g. constant, linear, quadratic, cubic, etc.) used to fit some regression relationship (Figure 1). Clearly a higher order model such as the cubic polynomial is strictly more complex than a lower order model such as the linear polynomial. Model fitting procedures based on optimisation (such as maximum likelihood methods or penalised likelihood methods) need to take great care not to *overfit* the data by fitting the parameters of an overly complex model to a relatively small data set. Overfitting is not a problem for fully Bayesian methods, as there is no "fitting" of the model to the data. We only have the sum rule and the product rule to work with, there is no "optimise" rule in probability theory. A more complex model, say one that has more parameters, simply spreads its predictive probability mass over more possible data sets than a simpler model. If all models are specified as probability distributions over data sets, since probability distributions have to sum to one, all models have the same amount of probability mass to spread over possible data (Figure 2). Given a particular data set, it therefore becomes possible to reject both models that are too simple or too complex simply by using the rules of probability theory.

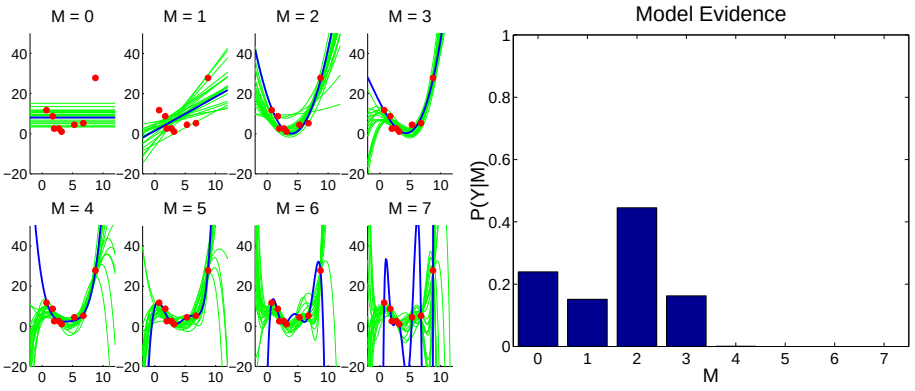

Figure 1. Marginal likelihoods, Occam's razor, and overfitting: Consider modelling a function $y = f(x) + \epsilon$ describing the relationship between some input variable x, and some output or response variable y. (left) The red dots in the plots on the left side are a data set of eight (x, y) pairs of points. There are many possible f that could model this given data. Let's consider polynomials of different order, ranging from constant ($M = 0$), linear ($M = 1$), quadratic ($M = 2$), etc to seventh order ($M = 7$). The blue curves depict maximum liklihood polynomials fit to the data under Gaussian noise assumptions (i.e. least squares fits). Clearly the $M = 7$ polynomial can fit the data perfectly, but it seems to be overfitting wildly, predicting that the function will shoot off up or down between neighbouring observed data points. In contrast, the constant polynomial may be underfitting, in the sense that it might not pick up some of the structure in the data. The green curves indicate 20 random samples from the Bayesian posterior of polynomials of different order given this data. A Gaussian prior was used for the coefficients, and an inverse gamma prior on the noise variance (these conjugate choices mean that the posterior can be analytically integrated). The samples show that there is considerable posterior uncertainty given the data, and also that the maximum likelihood estimate can be very different from the typical sample from the posterior. (right) The normalised model evidence or marginal likelihood for this model is plotted as a function of the model order, $P(Y|M)$, where the data set Y are the eight observed output y values. Note that given the data, model orders ranging from $M = 0$ to $M = 3$ have considerably higher marginal likelihood that other model orders, which seems plausible given the data. Higher order models, $M > 3$, have relatively much smaller marginal likelihood which is not visible on this scale. The decrease in marginal likelihood as a function of model order is a reflection of the automatic Occam's razor which results from Bayesian marginalisation.

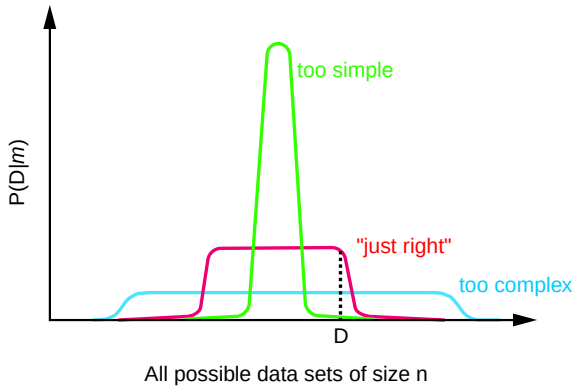

Figure 2. An illustration of Occam's Razor. Consider all possible data sets of some fixed size n. Competing probabilistic models correspond to alternative distributions over the data sets. Here we have illustrated three possible models which spread their probability mass in different ways over these possible data sets. A *complex* model (shown in blue) spreads its mass over many more possible data sets, while a *simple* model (shown in green) concentrates its mass on a smaller fraction of possible data . Since probabilities have to sum to one, the complex model spreads its mass at the cost of not being able to model simple data sets as well as a simple model—this normalisation is what results in an automatic Occam's razor. Given any particular data set, here indicated by the dotted line, we can use the marginal likehood to reject both overly simple models, and overly complex models. This figure is inspired by a figure from MacKay (1991), and an actual realisation of this figure on a toy classification problem is discussed in Murray and Ghahramani (2005).

This approach to Bayesian model comparison can be used to solve a vast range of problems in learning the structure of complex models. For example, it has been used to learn the number of clusters in a mixture model (Roberts et al., 1998; Ghahramani and Beal, 2000), finding relevant variables or features in a prediction problem (MacKay, 1994; Neal, 1998), discovering the number of states in a hidden Markov model (MacKay, 1997), and learning the dependency structure between variables in a probabilistic graphical model (Friedman and Koller, 2003).

The above approach to model comparison relies on the ability to enumerate a set of models $\mathcal{M}$ to be compared. This is often reasonable in scientific settings where there are a number of competing hypotheses to explain some phenomenon. Bayesian Occam's razor ensures that overly complex models are adequately penalised when doing model comparison. However, the complexity of real-world phenomena often requires us to consider complex models that are flexible enough to capture the structure in real data. Flexible models are not only more realistic, but will also generally result in more reasonable forecasts than simpler models. Bayesian nonparametrics provides a natural framework for defining flexible models.

3. Nonparametric models

The best way to understand nonparametric models is by first reminding ourselves of what parametric models are. A parametric model has some finite set of

parameters θ. Given these parameters, future predictions, x, are independent of the observed data:

$$p(x|\theta, \mathcal{D}) = p(x|\theta).$$

The parameters therefore capture everything there is to know about the data that is relevant for predicting future data.

We can think of all models as *information channels* from past data $\mathcal{D}$ to future predictions x. The parameters in a parametric model constitute a bottleneck in this information channel. The complexity of the model, and the capacity of the channel, is bounded, even if the amount of observed data becomes unbounded. Parametric models are therefore not generally very flexible.

In contrast a *nonparametric* model assumes that the data distribution cannot be defined in terms of such a finite set of parameters. However, often we can think of nonparametric models as being defined in terms of an infinite dimensional θ. More formally, the infinite dimensional θ is often represented as a function. The term *nonparametric* is therefore a bit of a misnomer; it's not that nonparametric models don't have parameters, in fact they have infinitely many parameters. Because of this, nonparametric models cannot be explicitly represented in terms of their parameters.

From the information channel viewpoint, we have removed the bottleneck. The amount of information that θ can capture about the data $\mathcal{D}$ grows as the amount of data grows. This makes nonparametric models more flexible than parametric models. [5]

There is another way to view the difference between parametric and nonparametric models. Predictions from a parametric model are explicitly and compactly summarised through the parameters θ, $P(x|\theta)$. Nonparametric models, in contrast, cannot be summarised in this way. Because of this predictions from a nonparametric are necessarily *memory-based*, $P(x|\mathcal{D})$; to make predictions we need to store or remember a growing amount of information about the training data, $\mathcal{D}$.

Nonparametric models are inextricably tied to the notion of *exchangeability*. A sequence is exchangeable if its joint distribution is invariant under arbitrary permutation of the indices. Consider modelling a data set $\{x_1, \ldots, x_N\}$ under the assumption that the ordering of the elements is uninformative. This data set may be a collection of documents, images, or any other sort of object collected in a manner such that either the ordering is irrelevant or completely unknown.[6] De Finetti's theorem states that a sequence is exchangeable if and only if there exists some θ such that the elements x_n are independently and identically distributed (iid) from some unknown distribution indexed by θ (Kallenberg, 2005). Importantly, θ may need to be infinite dimensional as it needs to index the space of probability measures (non-negative functions that normalise to one).

---

[5] To the best of my knowledge, this information channel view of nonparametric modelling is not explicitly articulated in the literature, although to be sure, there are a number of strong links between Bayesian inference and information theory (Zellner, 1988; MacKay, 2003). These ideas could certainly be further formalised by explicitly attempting to measure or estimate the channel capacity of different models. This would help provide a further definition of the *complexity* of a model, measured in bits per data point, which in many ways is more satisfactory than definitions that rely on counting parameters.

[6] There are generalisations of exchangeability to handle time series and arrays.

The consequence of de Finetti's theorem is that if we want to model exchangeable data in full generality, we need to consider putting distributions on unknown probability measures.

Distributions on measures, functions, and other infinite dimensional objects, are thus central to Bayesian nonparametric modelling. Many of these distributions are infinite dimensional versions of their finite dimensional counterparts, and in fact a good strategy for deriving Bayesian nonparametric models is to start from a parametric model and "take the infinite limit" (Neal, 2000). Distributions on infinite dimensional objects are the main subject of study in stochastic process theory, and therefore much of the terminology used in Bayesian nonparametrics is borrowed from this field.[7]

Two of the classical building blocks for Bayesian nonparametric models are the Gaussian process and the Dirichlet process. I will give an overview of these models in sections 4 and 5, with an emphasis on their applications to general problems in machine learning and statistics, including regression, classification, clustering, and density estimation (these problems will also be described in those sections). I will also cover newer building blocks that can represent nonparametric distributions over sparse matrices, hierarchies, covariances, and time series (sections 6 to 9). In order to give a broad overview I will necessarily have to avoid going in much depth into any of the specific topics, providing instead pointers to the relevant literature. Admittedly, my overview is based on my personal view of the field, and is therefore biased towards areas of Bayesian nonparametrics to which my colleagues and I have contributed, and misses other important areas.

4. Modelling functions, classification and regression: Gaussian processes

Gaussian processes (GPs) are a distribution over functions which can be used in numerous contexts where one's model requires one to represent an unknown function (Rasmussen and Williams, 2006). One dimensional GPs indexed by time are familiar to many fields: Brownian motion, Wiener processes, Ornstein-Uhlenbeck processes, linear Gaussian state-space models, and many random walks, are all examples of GPs. For historical reasons, GPs are also sometimes associated with spatial statistics, for example modelling temperature as a function of spatial location. Within machine learning, GPs are often used for nonlinear regression and classification.

Consider the following simple model for nonlinear regression

$$y_n = f(x_n) + \epsilon_n.$$

Here, we wish to model the relationship between an input (or covariate) x and an output (or response variable) y. The subscript n indexes data points in our data set $\mathcal{D}$, ϵ_n is some additive noise, and crucially, f is the unknown regression function we wish to learn about from data.

---

[7] The word "process" usually evokes the idea of something evolving temporally, but it's important to note that stochastic processes can be indexed by time, space, or any other index space. Many of the uses of stochastic processes in Bayesian nonparametrics do not correspond to indexing by time.

Assume we have a distribution over functions f and we evaluate the marginal distribution it assigns to the vector $\mathbf{f} = (f(x_1), \dots f(x_N))$. If, for any choice of input points, $(x_1, \dots, x_N)$, the marginal distribution over $\mathbf{f}$ is multivariate Gaussian, then the distribution over the function f is said to be a *Gaussian process*. In other words, a GP is an infinite dimensional generalisation of the multivariate Gaussian. Analogously to the Gaussian, a GP is parameterised by a mean function, and a covariance function.

The application of GPs to regression is straightforward. Starting with a prior on functions $p(f)$ we condition on the data to obtain a posterior $p(f|\mathcal{D})$. When the noise ϵ is assumed to be Gaussian, all computations reduce to operations on N dimensional Gaussians. The infinitely many other dimensions of f can be marginalised out analytically.

Classification problems correspond to predicting categorical response or output variables, e.g. $y \in \{\text{cat}, \text{dog}\}$. GP regression can be modified to do classification simply by introducing a link function that maps the real values $f(x)$ into probabilities over the classes. Computing $p(f|\mathcal{D})$ exactly becomes intractable but many good methods exist for approximating the required integrals (section 10).

5. Density estimation and clustering: Dirichlet processes and Chinese restaurant processes

We now consider two distinct problems—density estimation and clustering—and describe a close link between the two when approached from a Bayesian nonparametric modelling approach.

Density estimation refers to the problem of inferring an unknown density p from data $\mathcal{D} = \{x_1, \dots, x_N\}$. Let us first consider a very simple situation where the data points belong to a discrete and finite space with K possible values, $x \in \mathcal{X} = \{1, \dots, K\}$. Any distribution on $\mathcal{X}$ can be represented by a K-dimensional non-negative vector $\mathbf{p}$ that sums to one. To infer $\mathbf{p}$ from $\mathcal{D}$ we need a reasonable prior on finite distributions. The Dirichlet distribution is a natural choice which takes the form:

$$P(\mathbf{p}|\alpha, \mathbf{h}) = \frac{1}{Z} \prod_{k=1}^{K} p_k^{\alpha h_k - 1}.$$

Here Z is a normalising constant, $\mathbf{h}$ is the mean of $\mathbf{p}$ and $\alpha > 0$ controls the dispersion around the mean. A very attractive property of the Dirichlet distribution is that it is *conjugate* in the sense that the posterior $P(\mathbf{p}|\mathcal{D}, \alpha, \mathbf{h})$ is also Dirichlet. Another nice property is that for all but pathological choices of the parameters it has *good coverage* in the sense that it puts some nonzero probability mass near all possible values of $\mathbf{p}$.

The *Dirichlet process* (DP) extends the Dirichlet distribution to countable or uncountably infinite spaces $\mathcal{X}$. It has the property that for any *finite* partition of $\mathcal{X} = (\mathcal{X}_1, \dots \mathcal{X}_K)$ it marginalises to a K-dimensional Dirichlet distribution on the measure (i.e. mass) assigned to each element of the partition (Ferguson, 1973). We write that the probability measure G is drawn from a Dirichlet process prior as $G \sim \text{DP}(\alpha, H)$ analogously to our notation for the Dirichlet distribution. An excellent introduction to the Dirichlet process is provided by Teh (2010).

Conjugacy and good coverage suggest that the DP could be a very good general purpose nonparametric density estimator. Unfortunately, the distributions drawn from a DP prior are, with probability one, discrete so they don't have a density. In fact, a draw from a DP prior can be represented in the following way:

$$G = \sum_{k=1}^{\infty} \pi_k \delta_{x_k} \tag{5.1}$$

where the sum is an infinite sum, the π_k are masses that sum to one, δ is the Dirac-delta function, and the x_k are locations for those probability masses drawn iid from the base measure H which controls the mean of G. To alleviate the problem of discreteness, the DP is often used as a prior on the parameters of a mixture model, with the whole model now called a Dirichlet process mixture (DPM) (Antoniak, 1974; Ferguson, 1983; Lo, 1984; Neal, 2000).

The particular example of a Dirichlet process mixture of Gaussians, also known as an infinite Gaussian mixture model (Rasmussen, 2000) is widely used for both density estimation and clustering. *Clustering* refers to the problem of finding groupings of objects or data points such that similar objects belong to the same group and dissimilar objects belong to different groups. An example application of clustering is finding groups of similar celestial objects in a large astronomical survey. Posed abstractly in this way, clustering is not a well-defined problem (how many groups should we find? what does "similar" mean?). However, if we restrict ourselves to assuming that each cluster can be captured by some parameterised probability distribution over data points, such as a Gaussian, then clustering becomes a well-defined probabilistic inference problem. Bayesian nonparametric clustering using Dirichlet process mixtures is a natural extension of classical clustering algorithms such as the EM algorithm for finite mixture models or k-means (Duda and Hart, 1973). In a finite mixture model, the data is assumed to come from a distribution composed of K components:

$$p(x|\boldsymbol{\pi}, \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k p(x|\theta_k) \tag{5.2}$$

with mixing weights $\boldsymbol{\pi}$ that sum to one, and parameters θ_k for each component. An infinite mixture model considers the limit of $K \to \infty$, and has the property that it allows the number of observed clusters to grow with the number of observed data points. A DPM can be obtained as an infinite limit of a finite mixture model in many ways, but let us consider the following construction:

$$\begin{aligned} x_n &\sim p(x|\theta_n) \\ \theta_n &\sim G \\ G &\sim \mathrm{DP}(\alpha, H). \end{aligned}$$

Because G is discrete, the values of θ_n will repeat, which results in a clustering of the data points. By equation (5.1), the π_k correspond to the mixing weights of the infinitely many clusters, which can be compared to the finite counterpart (equation 5.2). The distribution over partitions of the data points induced by the DPM is known as a *Chinese Restaurant Process* (CRP; Aldous (1985)).

The DPM, apart from being mathematically elegant, has some practical advantages over traditional clustering algorithms. There can be computational advantages to running a single instance of a DPM inference algorithm which automatically infers the number of clusters, rather than having to run multiple instances of an algorithm to compare different hypotheses on the number of clusters. Moreover, at test time, the DPM always allows for the possibility that a new test point (e.g. a new astronomical object) belongs to a cluster that was not observed in the training data. This makes DPM predictions somewhat more robust to outliers.

6. Discrete state time series: infinite hidden Markov models

Many processes of interest produce sequential data for which models that assume exchangeability are inadequate. For example, natural language text, protein amino acid sequences, financial time series, and natural sounds all have sequential structure which is important to model.

One of the most widely used models for times series is the *hidden Markov model* (HMM). An HMM assumes that a series of observations $(x_1, \ldots, x_T)$ was generated by a process which can be in one of K different discrete states at each point in time, $s_t \in \{1, \ldots, K\}$. Moreover, in an HMM, s_t fully captures the state of the system at time t in the sense that given s_t the future evolution of the system does not depend on the state at times previous to t; this is the *Markov property*: $P(s_{t+1}|s_1, \ldots, s_t) = P(s_{t+1}|s_t)$. Finally, the observations are assumed to be drawn independently given the hidden states, through an emission process $P(x_t|s_t)$.

An example of an interesting application of HMMs in the physical and biological sciences is the modelling of single ion channel kinetics (Chung et al., 1990). Here, the measured time series of currents from an ion channel are modelled by assuming that at each point in time the channel macromolecule can be in one of many different conformational states, and that there is a transition matrix defining the probability of transitioning between each of these states.

Learning an HMM involves inferring both parameters of the transition process $P(s_{t+1}|s_t)$, which is in general a $K \times K$ transition matrix, and parameters of the emission process, $P(x_t|s_t)$ (Rabiner and Juang, 1986). The problem of learning the structure of an HMM corresponds to inferring the number of hidden states, K, from data. Rather than doing model selection over varying K, we would like to develop a nonparametric approach to HMMs where the model has countably infinitely many hidden states at its disposal. This can be useful both when we don't believe that any finite HMM can capture the data generating process well, and in situations where we believe that the data was actually generated from a finite-state process, but we simply don't know how many states this process should have.[8]

---

[8] There is a subtle but important distinction between assuming an infinite model, and assuming a finite but potentially unbounded model. The difference comes to light when we consider what we expect to happen in the limit of infinitely large data sets. In the former case, the posterior will have visited infinitely many distinct states, while in the latter case, the posterior should converge on some finite number of states.

The key insight that allows one to develop a nonparametric HMM is that the finite HMM is a time series generalisation of finite mixture models. At each time step, the HMM assumes that the observation x_t was generated by one of K mixture components, where s_t indicated the component (or cluster) used. The only difference between HMMs and mixture models is that the mixture indicators in an HMM depend on the ones at the previous time step.

Using this insight, Beal et al. (2002) developed the infinite HMM model (iHMM). The basic idea was to consider a Bayesian HMM with countably infinitely many states. The main difficulty was to define a sensible prior on the parameters of the $\infty \times \infty$ transition matrix. In the finite K dimensional case, one would typically use independent symmetric Dirichlet prior distributions for each row of the transition matrix (where a k^{th} row corresponds to the vector of all outgoing transition probabilities from state k). In the infinite limit, the independent Dirichlet prior doesn't result in a sensible model, as under this prior, with probability one, the HMM will keep transitioning to new states rather than revisiting previous states. The solution developed in Beal et al (Beal et al., 2002) was to couple the rows by using a hierarchical Dirichlet process, a solution analogous to a reinforced urn process in probability theory (Fortini and Petrone, 2012). This work was followed up in the elegant paper by Teh et al (Teh et al., 2006) which further developed the hierarchical Dirichlet process and proposed an improved MCMC sampler for the iHMM.

Since the original paper, there have been a number of conceptual and algorithmic developments of the infinite HMM. The beam sampler provides an efficient way of sampling the iHMM by using dynamic programming forward-backward style message passing (Van Gael et al., 2008a).[9] Parallel and distributed implementations of the iHMM allow larger scale deployments (Bratieres et al., 2010). The block diagonal iHMM is an extension which groups the hidden states into clusters of states, effectively hierarchically partitioning the state-space (Stepleton et al., 2009). The infinite HMM can be extended to have a power-law structure on the hidden states by using the Pitman-Yor process (Van Gael and Ghahramani, 2011; Blunsom and Cohn, 2011) and has been successfully applied to diverse problems such as language modelling (Blunsom and Cohn, 2011), and speaker diarization (Fox et al., 2008).

7. Sparse matrices and overlapping clusters: Indian buffet processes

One limitation of Dirichlet process mixtures, the infinite HMM, and clustering models in general, is that each data point is modelled as belonging to one of a set of mutually exclusive clusters. Although the nonparametric variants are flexible, in that they allow countably infinitely many clusters, they don't allow a data point to belong to multiple clusters at the same time – they fundamentally define distributions over *partitions* of the data.

We would like building blocks for our models that can allow overlapping cluster membership. For example, to understand a person's network of friendships we really need models which can capture the idea that a person can belong

[9] The beam sampler is available in the software package http://mloss.org/software/view/205/

simultaneously to many possible social groupings based on workplace, family, housing location, high school, hobbies, etc. This type of hidden structure in data is sometimes called *factorial* structure (Hinton and Zemel, 1994; Saund, 1994; Ghahramani, 1995).

The *Indian buffet process* (IBP; (Griffiths and Ghahramani, 2006, 2011)) is a probabilistic object which can be used to represent nonparametric factorial structure. There are a number of ways of understanding and deriving the IBP. Let's start with a sparse binary matrix view of IBPs.

Consider an $N \times K$ binary matrix, Z, where we can think of the rows of the matrix as corresponding to objects or data points, and the columns as features or clusters. An element of this matrix $z_{nk} = 1$ may denote that object n possesses hidden feature k, (or belongs to cluster k), and features (clusters) are not mutually exclusive in that an object can have multiple features. We wish to define a very simple distribution that is exchangeable over the objects (rows). A very simple choice is to assume that the columns are independent. We can therefore model each column through a single unknown parameter, θ_k, representing the frequency of feature k, $p(z_{nk} = 1|\theta_k) = \theta_k$. Full specification for this model requires some choice for the prior distribution of θ_k. A natural choice is the beta distribution (the special case of a Dirichlet when there are only two outcomes) which happens to be conjugate to the Bernoulli likelihood, allowing θ_k to be integrated out.

Like in the previous cases, we wish to consider models with infinitely many possible features or clusters, so we therefore have to examine the limit $K \to \infty$. The beta distribution has two parameters α, β and the mean of the beta distribution is $\alpha/(\alpha + \beta)$. For fixed α,β, and N in the limit $K \to \infty$, the matrix Z will have infinitely many ones in it, which makes it computationally and statistically of limited interest. However, consider scaling the first parameter of the beta distribution and setting the second parameter to 1, i.e. using a beta(α/K,1) prior for each θ_k.[10] In this case, the limit on the distribution of Z has a number of nice properties (a) the number of ones in each row is distributed as Poisson(α); (b) the total expected number of ones is αN; (c) the number of nonzero columns grown as $O(\alpha \log N)$; and (d) the rows are exchangeable. This distribution is the Indian Buffet Process.

Note that the distribution described above has infinitely many columns of zeros. Since the columns are all iid, once we sample a matrix, we can reorder its columns into a more usable representation such as the one shown in Figure 3. In many applications we are specifically interested in sparse binary matrices; for example to represent which factors are non-zero in sparse latent factor models (Knowles and Ghahramani, 2007), for binary matrix factorisation (Meeds et al., 2007), or to represent the adjacency matrix in a graph (Adams et al., 2010). However, it sometimes useful to view the IBP as a more abstract probabilistic object. Whereas the Chinese Restaurant Process is an infinitely exchangeable distribution over *partitions* of a set of objects, the IBP is an infinitely exchangeable distribution over *subsets* of a set of objects. Each object belongs to Poisson(α) subsets (or clusters), and as N increases, both the number of subsets grows (logarithmically) and the size of each subset grows (linearly) with N.

Since its introduction, many interesting properties and extensions of the IBP have been uncovered. Since the IBP defines an exchangeable distribution, it has

[10] It is not hard to reintroduce the second parameter to get the two-parameter IBP.

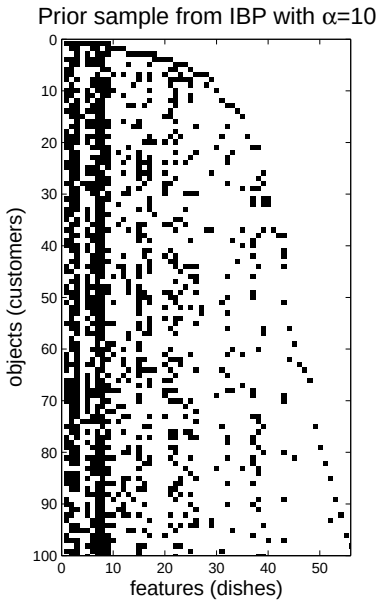

Figure 3. A sample from an IBP matrix, with columns reordered. Each row has on average 10 ones. Note the logarithmic growth of nonzero columns with rows. For the "restaurant" analogy where customers enter a buffet with infinitely many dishes you can refer to the original IBP papers.

a de Finetti mixing distribution; for the IBP Thibaux and Jordan (Thibaux and Jordan, 2007) showed that this is the beta process (Hjort, 1990). An important extension of the IBP is the three-parameter model which exhibits a power law growth in the number of clusters (Teh and Görür, 2009).

Nonparametric models which use the IBP to define sparse latent variables have been applied to a number of different problems, as reviewed in Griffiths and Ghahramani (2011). A very interesting application of the IBP is to the problem of *network modelling:* modelling the connections between objects or entities in social, biological and physical networks (Newman, 2010). While many models of networks are based on the idea of discovering communities or clusters of the nodes, the IBP allows each node to belong to multiple overlapping communities or clusters, a property which can be exploited to obtain improved predictive performance in tasks such as link prediction (Miller et al., 2009; Mørup et al., 2011; Palla et al., 2012).

Figure 4 shows a useful visualisation of the relationship between a number of models. Here we can see that the Dirichlet process mixture (section 5), the infinite HMM (section 6) and the IBP (section 7) can all be related to each other. Combining the key features of all models results in the infinite factorial HMM, a Bayesian nonparametric time series model with factorial hidden structure (Van Gael et al., 2008b).

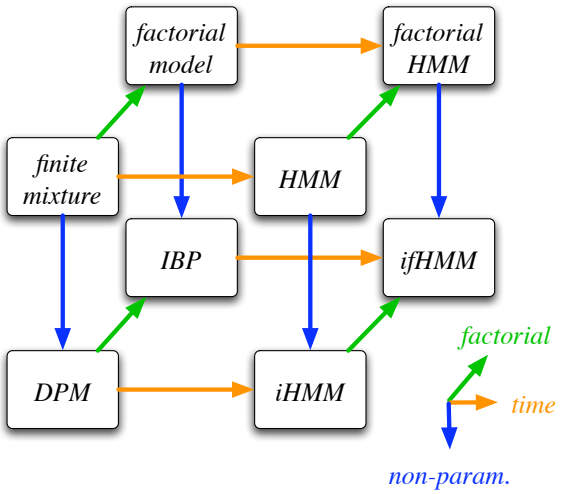

Figure 4. A diagram representing how some models relate to each other. We start from finite mixture models and consider three different ways of extending them. Orange arrows correspond to time series versions of static (iid) models. Blue arrows correspond to Bayesian nonparametric versions of finite parameteric models. Green arrows correspond to factorial (overlapping subset) versions of clustering (non-overlapping) models.

8. Hierarchies: Kingman's coalescent, and Dirichlet and Pitman-Yor Diffusion Trees

It is useful to distinguish between *flat* clustering models and *hierarchical* clustering models. In the former, data are partitioned into clusters, while in the latter, these clusters in turn are also partitioned into superclusters in a hierarchical manner. This results in a *tree* (or dendrogram) where at the top level we have a single root node corresponding to one coarse cluster with all the data points, while at the bottom we have leaves corresponding to fine clusters with a single data point in each cluster (Duda and Hart, 1973).

Such hierarchical clustering models are useful for many reasons. Firstly, many natural phenomena are well-modelled through hierarchies. For example, the evolution of biological organisms results in phylogenetic trees which are well-modelled by a hierarchy, and real-world objects can often be decomposed into parts, subparts, etc. Second, hierarchies allow one to tie together parameters of complex models so as to improve generalisation in learning. For example, if one is building statistical models of patient outcome across a country, it might be natural to group together parameters of regions, cities, hospitals, and individual doctors, corresponding to multiple levels of a hierarchy. Third, hierarchies provide good abstractions for interpretability of complex models. For example, rather than trying to understand what the hundreds of states in an HMM each do, it would be useful to have an HMM which partitions the states in a coarse to fine hierarchy so that one can start out by interpreting the coarse states and gradually move down to the fine states.

Note that while hierarchical clustering models are often described in terms of hierarchies over data points, in the above examples we have seen that hierarchies can be useful more generally over internal components of models, such as hidden states of an HMM, or parameters of a model.

We have seen that the Dirichlet process mixture model (and the associated Chinese Restaurant Process) can be used to define Bayesian nonparametric models for flat clustering. Are there equivalent Bayesian nonparametric models which result in hierarchical clusterings?

The answer is yes. Here we very briefly touch upon two frameworks for generating hierarchies which can be used in nonparametric models. For the nonparametric setting, the key requirement is that the models define an infinitely exchangeable distribution over the data points. For this to hold, the models must be projective in the sense that marginalising over the $(N+1)^{\text{th}}$ point should result in a coherent model over N points (Orbanz, 2009).

The first solution is given by *Kingman's coalescent* (Kingman, 1982). This model has been widely used for modelling population genetics (Donnelly and Tavare, 1995) and more recently in the machine learning community for hierarchical clustering (Teh et al., 2008). The coalescent defines distributions over trees by starting with every data point in its own cluster and considering a process that merges clusters *backwards* in time until only one cluster remains. The key property of the coalescent is that for each pair of clusters the event that they merge is independent of the event that any other pair merges, and the time to this event is drawn from an exponential distribution with constant rate (which we can set to 1 without loss of generality). This process is well-defined in the limit of $N \to \infty$ data points, and defines an infinitely exchangeable distribution over points.

A second solution is given by the Dirichlet Diffusion Tree (DDT; Neal (2003)). Like Kingman's coalescent, the DDT also defines a tree by considering a Markovian process evolving in time; however, the DDT starts at time $t=0$ and evolves *forward* for 1 unit of time.[11] We denote the evolution of the n^{th} point in time via $x_n(t)$. The first data point starts at a location $x_1(0)=0$ and follows a Brownian motion process, ending up at some point drawn marginally from a unit variance Gaussian, $x_1(1) \sim N(0,1)$. The second data point exactly follows the path of the first data point until a divergence event occurs, after which point its path is independent of the path of the first point. The time to divergence is parameterised through a *hazard function*, an object commonly used in survival analysis. Subsequent data points follow the paths of previous data points, diverging according to a scaled form of the hazard function, and when reaching a branch point choosing a branch with probability proportional to the number of points that chose that branch before. This process defines an exchangeable distribution over data points, parameterised by the unknown tree. Using the DDT prior, the problem of hierarchical clustering becomes one of inferring the unknown tree given some data.

---

[11] Note that both in the coalescent and in the DDT "time" is an indexing variable used to generate a tree structure, and need not correspond to a real notion of time.

Both Kingman's coalescent and the Dirichlet diffusion tree generate binary trees with probability one.[12] A generalisation of the DDT that allows arbitrary branching of the trees is given by the Pitman-Yor diffusion tree (PYDT; (Knowles and Ghahramani, 2011). The process is generalised to allow, at each branch point, for the new data point either to follow the path of one of the previous points, or to create a new branch. Like the Pitman Yor process (Pitman and Yor, 1997) the PYDT has two parameters controlling its branching structure. Certain settings of these parameters result in the DDT, while other settings recover the distribution over trees induced by Kingman's coalescent. The tree distribution induced by the PYDT is the multifurcating Gibbs fragmentation tree (McCullagh et al., 2008), the most general Markovian exchangeable distribution over trees. General distributions over hierarchies and other clustering structures can also be derived through the elegant theory of fragmentation-coagulation processes (Teh et al., 2011).

9. Covariance matrices: Generalised Wishart processes

Often we are interested in modelling the relationship between a number of variables. One can express relationships between variables in many different ways, but perhaps the simplest representation of dependence is the *covariance matrix*, Σ, a symmetric positive (semi)definite matrix representing second order statistics of the variables. The covariance matrix plays a key role in parameterising many distributions, most notably the multivariate Gaussian but also extensions such as the multivariate t-distribution or more generally elliptical distributions (Muirhead, 1982).

In certain applications, we wish to model how such a covariance matrix might depend on some other variables. For example, in econometrics and finance, one is often interested in modelling a time-varying covariance matrix $\Sigma(t)$—this is the key object of interest in the field of multivariate stochastic volatility (Asai et al., 2006). More generally, we would like to place distributions on covariance matrices that can depend on arbitrary variables, $\Sigma(x)$, not just scalar time. Viewed as a *function* of x, we want to be able to define distributions on matrix-valued functions restricted to the space of symmetric-positive-definite (s.p.d.) matrix values. Is there a convenient and simple way to define such a stochastic process?

Indeed there is, and the key insight comes from the observation that one can generate s.p.d. matrices by taking the outer products of random vectors. Consider for following construction, where we draw D-dimensional vectors independently from a multivariate Gaussian distribution $\mathbf{u}_i \sim \mathcal{N}(0, V)$ with covariance matrix V, and we define

$$\Sigma = \sum_{i=1}^{\nu} \mathbf{u}_i \mathbf{u}_i^\top .$$

Such a matrix Σ is Wishart distributed with ν degrees of freedom and is s.p.d. with probability one as long as $\nu \geq D$. The mean of Σ is proportional to V.

---

[12] In the case of the DDT, binary trees result in all but pathological cases where the hazard function diverges before $t = 1$.

We can generalise the Wishart distribution to a stochastic process indexed by x in any space $\mathcal{X}$ by replacing the elements of each $\mathbf{u}_i$ with draws from a Gaussian process: $\mathbf{u}_i(x) = (u_{i1}(x), u_{i2}(x), \ldots, u_{iD}(x))$ where

$$u_{id} \sim \mathrm{GP}(0, K),$$

where K is the covariance function or kernel of the GP Rasmussen and Williams (2006). The desired stochastic process is obtained by the same construction.

$$\Sigma(x) = \sum_{i=1}^{\nu} \mathbf{u}_i(x)\mathbf{u}_i(x)^\top.$$

A special case of such a construction where the Gaussian processes are assumed to be Brownian has been studied in probability theory and is known as a *Wishart process* (Bru, 1991); these processes have recently been applied to problems in econometrics (Philipov and Glickman, 2006; Gouriéroux et al., 2009). The more general case for arbitrary Gaussian processes is developed in (Wilson and Ghahramani, 2011) where applications to multivariate stochastic volatility are also explored. In this general framework, the model parameters controlling V, K and ν can be learned from data.

10. Inference

So far we have focused on describing a general probabilistic approach to modelling and some nonparametric distributions which can be used in models of complex data sets. The three key ingredients in any approach to modelling are the data, the model, and an *algorithm for doing probabilistic inference* in the model. The problem of probabilistic inference corresponds to computing the conditional probability of some variables of interest given some observed variables, whilst marginalising out all other variables. Thus, both the computation of a model's marginal likelihood (equation 2.1) and prediction (equation 2.2) are inference problems. Filling in missing data given observed data, computing the parameter posterior, or evaluating expectations of various quantities can all also be phrased as instances of the inference problem.

Generally, the inference problem boils down to a problem of numerical integration or summation over large state-spaces. For most models of interest, especially nonparametric models, exact inference is computationally *intractable*, in the sense that all known algorithms for computing the exact probabilities of interest scale exponentially with some aspect of the problem, such as the number of data points or variables. In many problems, even approximating these exact probabilities to within some small error tolerance is intractable in the worst case.[13]

A wide variety of approximation methods have been developed to solve Bayesian inference problems. These can be roughly divided into stochastic approximations (which make extensive use of random numbers) and deterministic approximations. Some examples of widely used stochastic approximate inference methods include Markov chain Monte Carlo methods (for an excellent review

[13] Most theoretical results on intractability focus on the worst case of a problem instance. The real-world inference problem may in fact be much easier to approximate.

see Neal (1993)), exact sampling methods (Propp and Wilson, 1996) and particle filtering methods (Doucet et al., 2000). Some examples of deterministic algorithms include the Laplace approximation, variational methods (Jordan et al., 1999), and expectation propagation (Minka, 2001). Both deterministic and stochastic algorithms for inference can often exploit the conditional independence relationships that exist between the variables in a model to perform the relevant computations efficiently using local messages passed between nodes of a graphical model (Winn and Bishop, 2005; Minka, 2005; Koller and Friedman, 2009).

A complete review of approximate inference methods is beyond the scope of this paper, but a couple of points are worth making. All approximate inference methods can be characterised in terms of a speed-accuracy tradeoff. Some methods are fast but often inaccurate, while other methods are slow or, like MCMC, can be run for increasing amounts of time to give increasingly accurate results. There is no general rule of thumb for which approximate inference method is best—different models and problems tend to favour different methods. However, for a particular problem, the difference between a good choice of inference algorithm and a poor choice can be orders of magnitude of computation. Thus, it is well worth being familiar with a number of inference algorithms and being willing to try several methods on a particular problem.

The field of Bayesian statistics has thrived in recent years, both due to the availability of better inference algorithms and the dramatic growth in computing power. Ironically, the most widely used inference method in Bayesian statistics is MCMC, which itself is a thoroughly frequentist (non-Bayesian) method for approximating integrals. From a Bayesian perspective, numerical computation problems are also inference problems. In numerical integration, one is trying to infer the value of an integral by computing the integrand at a limited number of locations. The values of the integrand are the *data*, which combined with a prior on the integrand, can result in a posterior on the value of the integral. The choice of where to evaluate the integrand can be made using Bayesian decision theory, clearly the random evaluations of MCMC are not an optimally efficient method for evaluating integrals. The Bayesian approach to numerical integration is known variously as Bayesian quadrature or Bayesian Monte Carlo, and although it is not as widely used as MCMC it can be dramatically more efficient in situations where evaluating the integrand is computationally costly (Diaconis, 1988; O'Hagan, 1991; Rasmussen and Ghahramani, 2003; Osborne et al., 2012).

Deriving the approximate inference equations for each new model can be tedious and error-prone, thereby inhibiting the researcher's ability to explore many variations on any given model. The field of *probabilistic programming* offers an exciting alternative approach to building and evaluating models. Three notable examples of probabilistic programming frameworks are *BUGS*, *Church*, and *Infer.NET* (Lunn et al., 2000; Goodman et al., 2008; Minka et al., 2010). The basic idea is to write down a computer program defining the generative model in a programming language that is augmented to have random variables. This computer program that defines the generative model can then be *automatically* transformed or manipulated into a program that performs approximate posterior inference in the model. Thus the process of deriving the approximate inference equations is automated, reducing the risk of human error. One potential disadvantage of this approach is that the automatically derived inference code

may not be as efficient as an expertly designed inference method for a particular model. However, the rapid model development cycle which can result from the use of probabilistic programming languages offers tremendous advantages for the future of probabilistic modelling.

11. Conclusions

Modelling is fundamentally a process of quantifying uncertainty about possible predictions given available information. Probability theory provides an elegant framework for representing all sources of uncertainty in a model. Probability theory also provides the simple rules with which to manipulate models so as to obtain predictions, to compare models, and to learn models from data. Bayesian statistics is simply the application of the rules of probability theory to modelling from data.

The probabilistic approach to modelling is most effective when the models are flexible enough to capture relevant aspects of the problem domain. Flexibility can be achieved by allowing models either to have many parameters, or in the limiting case to have infinitely many parameters, in other words to be *nonparametric*. Learning models with infinitely many parameters may seem statistically and computationally daunting. However, statistically we have seen in section 2 that the process of Bayesian averaging or marginalisation avoids overfitting and therefore allows one to use models with infinitely many parameters (section 3). Computationally we have seen that there are a wide range of approximation methods that can be used to perform the relevant marginalisations required for inference (section 10).

This paper has attempted to give an overview of the basics of Bayesian modelling, the motivation for Bayesian nonparametrics, and some of the more widely used nonparametric models. We have touched upon some computational issues, and briefly alluded to some successful applications of Bayesian nonparametrics. Of course there is a great deal of current work in this rapidly advancing field that was not covered in this paper.

There are four areas of future work worth highlighting: theory, new models, scalable algorithms, and new applications. Regarding *theory*, developing frequentist consistency and convergence rate results for Bayesian nonparametric models is challenging but important. In particular, the effect of the prior on an infinite dimensional space can result in inconsistent models in certain cases, so careful study of consistency is required.

While we have many building blocks for models, *new models* are an exciting area for work as they can lead to novel applications. General results on the construction of Bayesian nonparametric models, such as the recent work of Orbanz (2009), are particularly useful for developing new models.

Many modern applications of statistical modelling and machine learning require the analysis of very large datasets (this is sometimes called "Big Data"). While a number of very good approximate inference methods have been developed in the last few decades, making these highly *scalable* is a challenge that requires extensive tools from computer science including efficient data structures, and parallel and distributed computing paradigms.

Finally, novel *applications* of Bayesian nonparametric modelling will advance the field in many ways. Firstly, many new models are derived in response to the challenges arising from new applications. Secondly, applications can motivate general scalable inference solutions. Finally, successful applications will help in widening the adoption of a Bayesian nonparametric approach to modelling.

References

Adams, R. P., Wallach, H., and Ghahramani, Z. (2010). Learning the structure of deep sparse graphical models. In Teh, Y. W. and Titterington, M., editors, *13th International Conference on Artificial Intelligence and Statistics*, pages 1–8, Chia Laguna, Sardinia, Italy.

Aldous, D. (1985). Exchangeability and related topics. In *École d'été de probabilités de Saint-Flour XIII–1983*, pages 1–198. Springer, Berlin.

Antoniak, C. E. (1974). Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *Annals of Statistics*, 2(6):1152–1174.

Asai, M., McAleer, M., and Yu, J. (2006). Multivariate stochastic volatility: a review. *Econometric Reviews*, 25(2):145–175.

Beal, M. J., Ghahramani, Z., and Rasmussen, C. E. (2002). The infinite hidden Markov model. *Advances in Neural Information Processing Systems*, 14:577 – 584.

Blunsom, P. and Cohn, T. (2011). A hierarchical Pitman-Yor process HMM for unsupervised part of speech induction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 865–874. Association for Computational Linguistics.

Bratieres, S., Van Gael, J., Vlachos, A., and Ghahramani, Z. (2010). Scaling the iHMM: Parallelization versus Hadoop. In *Proceedings of the International Workshop on Scalable Machine Learning and Applications*, Bradford, UK. IEEE International Conference on Computing and Information Technology.

Bru, M. (1991). Wishart processes. *Journal of Theoretical Probability*, 4(4):725–751.

Chung, S. H., Moore, J. B., Xia, L., Premkumar, L. S., and Gage, P. W. (1990). Characterization of single channel currents using digital signal processing techniques based on hidden markov models. *Phil. Trans. R. Soc. Lond. B*, 329(1254):265–285.

Diaconis, P. (1988). Bayesian numerical analysis. In *Statistical Decision Theory and Related Topics IV*, volume 1, pages 163âĂŞ–175. Springer-Verlag.

Donnelly, P. and Tavare, S. (1995). Coalescents and genealogical structure under neutrality. *Annual Review of Genetics*, 29(1):401–421.

Doob, J. L. (1949). Application of the theory of martingales. *Colloques Internationaux du Centre National de la Recherche Scientifique*, 13:23–27.

Doucet, A., de Freitas, J. F. G., and Gordon, N. J. (2000). *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, New York.

Duda, R. O. and Hart, P. E. (1973). *Pattern classification and scene analysis*. Wiley.

Ferguson, T. S. (1973). A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1(2):209–230.

Ferguson, T. S. (1983). Bayesian density estimation by mixtures of normal distributions. *Recent advances in statistics*, pages 287–303.

Fortini, S. and Petrone, S. (2012). Hierarchical reinforced urn processes. *Preprint*.

Fox, E. B., Sudderth, E. B., Jordan, M. I., and Willsky, A. S. (2008). An HDP-HMM for systems with state persistence. In *Proceedings of the 25th International Conference on Machine Learning*, volume 25, Helsinki.

Friedman, N. and Koller, D. (2003). Being Bayesian about Network Structure: A Bayesian Approach to Structure Discovery in Bayesian Networks. *Machine Learning*, 50:95–126.

Ghahramani, Z. (1995). Factorial learning and the EM algorithm. In Tesauro, G., Touretzky, D. S., and Leen, T. K., editors, *Advances in Neural Information Processing Systems 7*, pages 617–624, Cambridge, MA. MIT Press.

Ghahramani, Z. and Beal, M. J. (2000). Variational inference for Bayesian mixtures of factor analysers. *Advances in neural information processing systems*, 12:449–455.

Goodman, N., Mansinghka, V., Roy, D., Bonawitz, K., and Tenenbaum, J. (2008). Church: a language for generative models. In *Uncertainty in Artificial Intelligence*, volume 22, page 23.

Gouriéroux, C., Jasiak, J., and Sufana, R. (2009). The Wishart autoregressive process of multivariate stochastic volatility. *Journal of Econometrics*, 150(2):167–181.

Griffiths, T. L. and Ghahramani, Z. (2006). Infinite latent feature models and the Indian Buffet Process. In Weiss, Y., Schölkopf, B., and Platt, J., editors, *Advanced in Neural Information Processing Systems 18*, pages 475–482, Cambridge, MA, USA. MIT Press.

Griffiths, T. L. and Ghahramani, Z. (2011). The Indian buffet process: An introduction and review. *Journal of Machine Learning Research*, 12:1185–1224.

Griffiths, T. L. and Tenenbaum, J. B. (2006). Optimal predictions in everyday cognition. *Psychological Science*, 17(9):767–773.

Hinton, G. E. and Zemel, R. S. (1994). Autoencoders, minimum description length, and Helmholtz free energy. In Cowan, J., Tesauro, G., and Alspector, J., editors, *Advances in Neural Information Processing Systems 6*. Morgan Kaufmann Publishers, San Francisco, CA.

Hjort, N. L. (1990). Nonparametric Bayes estimators based on Beta processes in models for life history data. *Annals of Statistics*, 18:1259–1294.

Jefferys, W. H. and Berger, J. O. (1992). Ockham's Razor and Bayesian Analysis. *American Scientist*, 80:64–72.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233.

Kallenberg, O. (2005). *Probabilistic Symmetries and Invariance Principles*. Springer.

Kingman, J. F. C. (1982). The coalescent. *Stochastic Processes and their Applications*, 13:235–âĂŞ248.

Knill, D. and Richards, W. (1996). *Perception as Bayesian inference*. Cambridge University Press.

Knowles, D. A. and Ghahramani, Z. (2007). Infinite sparse factor analysis and infinite independent components analysis. In *7th International Conference on Independent Component Analysis and Signal Separation*, pages 381–388, London, UK. Springer.

Knowles, D. A. and Ghahramani, Z. (2011). Pitman-Yor diffusion trees. In *Uncertainty in Artificial Intelligence (UAI 2011)*, pages 410–418.

Koller, D. and Friedman, N. (2009). *Probabilistic graphical models: principles and techniques*. The MIT Press.

Le Cam, L. (1986). *Asymptotic Methods in Statistical Decision Theory*. Springer.

Lo, A. (1984). On a class of Bayesian nonparametric estimates: I. density estimates. *The Annals of Statistics*, 12(1):351–357.

Lunn, D. J., Thomas, A., Best, N., and Spiegelhalter, D. (2000). WinBUGS–a Bayesian modelling framework: concepts, structure, and extensibility. *Statistics and computing*, 10(4):325–337.

MacKay, D. J. C. (1991). *Bayesian Methods for Adaptive Models*. PhD thesis, California Institute of Technology.

MacKay, D. J. C. (1994). Bayesian nonlinear modeling for the prediction competition. *ASHRAE transactions*, 100(2):1053–1062.

MacKay, D. J. C. (1997). Ensemble learning for hidden Markov models. Technical report, Cavendish Laboratory, University of Cambridge.

MacKay, D. J. C. (2003). *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press.

McCullagh, P., Pitman, J., and Winkel, M. (2008). Gibbs fragmentation trees. *Bernoulli*, 14(4):988–1002.

Meeds, E., Ghahramani, Z., Neal, R., and Roweis, S. (2007). Modelling dyadic data with binary latent factors. In Schölkopf, B., Platt, J., and Hofmann, T., editors, *Advances in Neural Information Processing 19*, pages 977–984, Cambridge, MA, USA. MIT Press.

Miller, K., Griffiths, T., and Jordan, M. I. (2009). Nonparametric latent feature models for link prediction. In *Advances in Neural Information Processing Systems*, volume 22, pages 1276–1284.

Minka, T. P. (2001). Expectation propagation for approximate Bayesian inference. In *Uncertainty in Artificial Intelligence*, volume 17, pages 362–369.

Minka, T. P. (2005). Divergence measures and message passing. *Microsoft Research Tech. Report MSR-TR-2005-173*.

Minka, T. P., Winn, J. M., Guiver, J. P., and Knowles, D. A. (2010). Infer.NET 2.4. Microsoft Research Cambridge. http://research.microsoft.com/infernet.

Mørup, M., Schmidt, M., and Hansen, L. (2011). Infinite multiple membership relational modeling for complex networks. In *Machine Learning for Signal Processing (MLSP), 2011 IEEE International Workshop on*, pages 1–6. IEEE.

Muirhead, R. (1982). *Aspects of multivariate statistical theory*, volume 42. Wiley Online Library.

Murray, I. A. and Ghahramani, Z. (2005). A note on the evidence and Bayesian Occam's razor. Technical Report GCNU-TR 2005-003, Gatsby Unit, University College London.

Neal, R. M. (1993). Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, Department of Computer Science, University of Toronto.

Neal, R. M. (1998). Assessing relevance determination methods using DELVE. In Bishop, C. M., editor, *Neural Networks and Machine Learning*, pages 97–129. Springer-Verlag.

Neal, R. M. (2000). Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9:249–265.

Neal, R. M. (2003). Density modeling and clustering using Dirichlet diffusion trees. In Bernardo, J. M., editor, *Bayesian Statistics 7*, pages 619–629.

Newman, M. (2010). *Networks: An Introduction*. Oxford University Press.

O'Hagan, A. (1991). Bayes-Hermite quadrature. *Journal of Statistical Planning and Inference*, 29:245–260.

Orbanz, P. (2009). Construction of nonparametric Bayesian models from parametric Bayes equations. In *Advances in Neural Information Processing Systems*, volume 22, pages 1392–1400.

Osborne, M. A., Garnett, R., Roberts, S., Hart, C., Aigrain, S., and Gibson, N. (2012). Bayesian quadrature for ratios. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2012)*.

Palla, K., Knowles, D. A., and Ghahramani, Z. (2012). An infinite latent attribute model for network data. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*.

Philipov, A. and Glickman, M. (2006). Multivariate stochastic volatility via Wishart processes. *Journal of Business and Economic Statistics*, 24(3):313–328.

Pitman, J. and Yor, M. (1997). The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *Annals of Probability*, 25:855–900.

Propp, J. G. and Wilson, D. B. (1996). Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures and Algorithms*, 1&2(9):223–252.

Rabiner, L. R. and Juang, B. H. (1986). An introduction to hidden Markov models. *IEEE Acoustics, Speech & Signal Processing Magazine*, 3:4–16.

Rasmussen, C. E. (2000). The Infinite Gaussian Mixture Model. In *Adv. Neur. Inf. Proc. Sys. 12*, pages 554–560.

Rasmussen, C. E. and Ghahramani, Z. (2001). Occam's Razor. In *Advances in Neural Information Processing Systems13*, pages 294–300, Cambridge, MA. MIT Press.

Rasmussen, C. E. and Ghahramani, Z. (2003). Bayesian Monte Carlo. In *Advances in Neural Information Processing Systems 15*, pages 489–496, Cambridge, MA, USA. MIT Press.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. MIT Press, Cambridge.

Roberts, S. J., Husmeier, D., Rezek, I., and Penny, W. (1998). Bayesian approaches to Gaussian mixture modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1133–1142.

Saund, E. (1994). Unsupervised learning of mixtures of multiple causes in binary data. In Cowan, J., Tesauro, G., and Alspector, J., editors, *Advances in Neural Information Processing Systems 6*. Morgan Kaufmann, San Francisco, CA.

Stepleton, T., Ghahramani, Z., Gordon, G., and Lee, T.-S. (2009). The block diagonal infinite hidden Markov model. In van Dyk, D. and Welling, M., editors, *12th International Conference on Artificial Intelligence and Statistics*, volume 5, pages 552–559. Microtome Publishing.

Teh, Y. W. (2010). Dirichlet processes. In *Encyclopedia of Machine Learning*. Springer.

Teh, Y. W., Blundell, C., and Elliott, L. T. (2011). Modelling genetic variations with fragmentation-coagulation processes. In *Advances in Neural Information Processing Systems 23*.

Teh, Y. W., Daumé, H., and Roy, D. M. (2008). Bayesian agglomerative clustering with coalescents. In *Advances in Neural Information Processing Systems*, volume 20.

Teh, Y. W. and Görür, D. (2009). Indian buffet processes with power-law behavior. In *Advances in Neural Information Processing Systems 22*, pages 1838–1846.

Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. (2006). Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581.

Thibaux, R. and Jordan, M. I. (2007). Hierarchical beta processes and the Indian buffet process. In *Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS 2007)*.

Van der Vaart, A. (2000). *Asymptotic statistics*. Cambridge University Press.

Van Gael, J. and Ghahramani, Z. (2011). Nonparametric hidden Markov models. In Barber, D., Cemgil, A., and Chiappa, S., editors, *Bayesian Time Series Models*, pages 317–340. Cambridge University Press.

Van Gael, J., Saatci, Y., Teh, Y. W., and Ghahramani, Z. (2008a). Beam sampling for the infinite hidden Markov model. In *Proceedings of the 25th International Conference on Machine Learning*, pages 1088–1095.

Van Gael, J., Teh, Y. W., and Ghahramani, Z. (2008b). The infinite factorial hidden Markov model. In *Advances in Neural Information Processing Systems*, volume 21, pages 1697–1704.

Wilson, A. G. and Ghahramani, Z. (2011). Generalised Wishart Processes. In *Uncertainty in Artificial Intelligence*, pages 736–744. AUAI Press.

Winn, J. and Bishop, C. M. (2005). Variational Message Passing. *The Journal of Machine Learning Research*, 6(1):661–694.

Wolpert, D. M., Ghahramani, Z., and Jordan, M. I. (1995). An internal model for sensorimotor integration. *Science*, 269:1880–1882.

Zellner, A. (1988). Optimal information processing and Bayes's Theorem. *The American Statistician*, 42(4):278–280.