Chapter 1

Expressing Structure through Kernels

1.1 Nonparametric Inference via Bayesian Quadrature

We extend the approximate integration method of Bayesian Quadrature to infinite structured domains, introducing a new inference method for nonparametric models.This method admits more flexible sampling schemes than Markov chains, for example active learning, and provides natural convergence diagnostics. We give conditions necessary for consistency, and show how to construct kernels between structures which take advantage of symmetries in the likelihood function. We demonstrate our inference method on both the Dirichlet process mixture model and the Indian buffet process.

1.2 Introduction

The central problem of probabilistic inference is to compute integrals over probability distributions of the form

$$Z_{f,p} = \int f(\theta)p(\theta)d\theta \tag{1.1}$$

Examples include computing marginal distributions, making predictions while marginalizing over parameters, or computing the Bayes risk in a decision problem. Machine learning has produced a rich set of methods for computing these integrals, such as Expectation Propagation[cite], Variational Bayes[cite], and many variations of Markov chain Monte

Carlo (mcmc) [cite Iain Murray].

In non-parametric models, estimating (**??**) is especially challenging, as the domain of integration in is infinite-dimensional. A variety of mcmc methods have been developed to tackle this problem. However, mcmc has known weaknesses, such as difficulty diagnosing convergence, the requirement of a burn-in period, and difficulty obtaining samples from a given subset of possible θ.

Bayesian quadrature (bq) **?**, also known as Bayesian Monte Carlo **?**, is a model-based method of approximate integration. bq infers a posterior distribution over f conditioned on a set of samples $f(\theta_s)$, and gives a posterior distribution on $Z_{f,p}$. bq remains a relatively unexplored integration method, and has so far only been used in low-dimensional, continuous spaces **?**.

Summary of contributions In this paper, we extend the bq method to infinite, structured domains, introducing a new family of inference methods for non-parametric models. We give conditions necessary for consistency. We introduce kernels for inference problems using the Indian Buffet process and Dirichlet Proces mixture model which encode the many symmetries of the likelihood functions.

We then demonstrate the advantages of model-based inference on synthetic datasets, such as uncertainty estimates, flexibility in sampling methods, and post-hoc sensitivity analysis of prior and likelihood parameters. We then discuss limitations of the framework as it stands.

1.3 Bayesian Quadrature

In contrast to mcmc, a procedure which computes (**??**) in the limit of infinite samples, Bayesian quadrature is a *model-based* integration method. This means that bq puts a prior distribution on f, then conditions on some observations $f(\theta_s)$ at some query points θ_s. The posterior distribution over f then implies a distribution over $Z_{f,p}$, which can be obtained by integrating over all possible f. See Figure **??** for an illustration of Bayesian Quadrature.

It may seem circular to introduce an integral over an uncountable number of functions in order to solve what was originally an integral over a single function. However, the gpposterior has a simple form which makes integration possible in closed form in many cases. If f is assigned a Gaussian process prior with kernel function k and mean 0, then

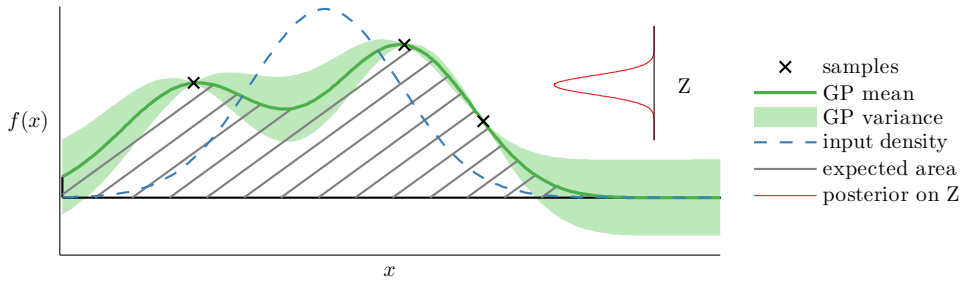

Fig. 1.1 An illustration of Bayesian Quadrature. The function $f(x)$ is sampled at a set of input locations. This induces a Gaussian process posterior distribution on f, which is integrated in closed form against the target density, $p(\mathbf{x})$. Since the amount of volume under f is uncertain, this gives rise to a (Gaussian) posterior distribution over $Z_{f,p}$.

after conditioning on function evaluations $\mathbf{y} = f(\theta_1) \dots f(\theta_N)$, we obtain:

$$p(\mathbf{f}(\theta_\star)|\mathbf{y}) = \mathcal{N}\Big(\mathbf{f}(\theta_\star)|\bar{\mathbf{f}}(\theta_\star), \mathrm{cov}(\theta_\star, \theta'_\star)\Big) \tag{1.2}$$

where

$$\bar{\mathbf{f}}(\mathbf{x}_\star) = k(\theta_\star, \theta_s)K^{-1}\mathbf{y} \tag{1.3}$$

$$\mathrm{cov}(\mathbf{x}_\star, \mathbf{x}'_\star) = k(\theta_\star, \theta_\star) - k(\theta_\star, \theta_s)\mathbf{K}^{-1}k(\theta_s, \theta_\star) \tag{1.4}$$

and $\mathbf{K}_{mn} = k(\theta_m, \theta_n)$. Conveniently, the gp posterior implies a closed form for the expectation and variance of (**??**):

$$\mathbb{E}\left[Z_{f,p}|\mathbf{y}\right] = \mathbb{E}_{\mathrm{gp}(f|\mathbf{y})}\left[\int f(\theta)p(\theta)d\theta\right] = \left[\int k(\theta, \theta_s)p(\theta)d\theta\right]\mathbf{K}^{-1}\mathbf{y} = \mathbf{z}^\mathsf{T}\mathbf{K}^{-1}\mathbf{y} \tag{1.5}$$

$$\mathbb{V}\left[Z_{f,p}|\mathbf{y}\right] = \mathbb{V}_{\mathrm{prior}}\left[Z_{f,p}\right] - \mathbf{z}^\mathsf{T}\mathbf{K}^{-1}\mathbf{z} \tag{1.6}$$

where

$$z_n = \int k(\theta, \theta_n)p(\theta)d\theta \tag{1.7}$$

$$\mathbb{V}_{\mathrm{prior}}\left[Z_{f,p}\right] = \iint k(\theta, \theta')p(\theta)p(\theta')d\theta d\theta'. \tag{1.8}$$

For longer derivations, see the supplementary material. This brings us to the one of the main technical constraints of this method: Choosing a form for the kernel function $k(\theta, \theta')$ such that we can compute (**??**) and (**??**) efficiently. We must also set or integrate out any parameters of k.

1.3.1 BQ as an inference method

If we assume that in (**??**), the form of $p(\theta)$ is such that we can compute z_n, then applying bq is straighforward. For example, in **?**, bq was used to solve problems where $p(\theta)$ was a Gaussian prior over parameters, and $f(\theta) = p(x|\theta)$ was a likelihood function, making Z the *model evidence*, a useful quantity when comparing models.

Typically, however, we are also interested in other quantities besides the model evidence, such as marginal distributions of latent parameter. In that case, we must solve a more difficult integral, where $p(\theta) = p^*(\theta|x)$ is a possibly unnormalized distribution of unknown form.

$$\mathbb{E}_{p(\theta|\mathbf{x})}\left[f(\theta)\right] = \int f(\theta)p(\theta|x)d\theta \quad = \frac{1}{Z}\int f(\theta)p(x|\theta)p(\theta)d\theta \quad \text{where} \quad Z = \int p(x|\theta)p(\theta)d\theta \tag{1.9}$$

Integrals of this form can also be solved using Bayesian quadrature. For a thorough treatment of this problem, see [Mike's BQR paper].

This will show that bq can be applied in several ways: For instance, if we wish to perform inference about an unknown parameter τ, we can include it as a variable to be integrated over by the gp. However, if for some technical reason this is difficult, we can also simply vary our extra parameter over some range, recomputing $\mathbf{y}(\tau)$ and obtaining a marginal distribution over $Z_{f,p}$ for each value of τ. [Reference an experiment?] This approach is useful for post-hoc sensitivity analysis.

1.4 Guidelines for Constructing a Kernel

As pointed out above, one of the most important design decisions in bq is the choice of kernel function $k(\theta, \theta')$, which specifies the prior covariance between the values of the likelihood function $p(\mathbf{x}|\theta)$ and $p(\mathbf{x}|\theta')$. This function is somewhat analogous to the proposal distribution required to construct a Metropolis-Hastings sampler [cite]. In this section, we give guidance on how to construct an appropriate kernel.

The kernel typically should encode as much prior knowledge about the function being modeled as possible. In regression problems, this usually amounts to specifying the smoothness properties of the function being modeled. When doing inference, however, we typically know the likelihood function in closed form. The more properties of the likelihood function we can encode in the kernel, the fewer samples we will need in order to learn about the value of its integral. In particular, we should encode any known

symmetries:

Symmetry Encoding: The prior correlation $\frac{k(\theta,\theta')}{\sqrt{k(\theta,\theta)}\sqrt{k(\theta',\theta')}}$ should equal 1 when $f(\theta) = f(\theta')$. That is to say, if two parameter settings are indistinguishable under the likelihood, our model can converge more quickly if it enforces that those likelihood values are identical. This is another way of saying that the covariance function should encode all known symmetries.

The ability to encode symmetries in the kernel is one of the major advantages of bq over Monte Carlo methods. In unidentifiable models such as mixture models, often it is known that there exist many symmeteric modes in the posterior, which represents a major difficulty when computing model evidence. By encoding these symmetries in the prior over likelihood functions, bq neatly solves the problem of identifiability when estimating $Z_{f,p}$.

1.4.1 Convergence

In the existing bq literature [cite only 4 papers], the integration problems considered have been low-dimensional, and the kernel function used has always been, to the best of the authors' knowledge, the squared-exponential kernel. In that case, existing results on the consistency of gp regression [cite consistency] imply that, under some conditions, the bq estimator (a linear transform of the gp posterior) is also consistent.

For infinite-dimnensional spaces with complex kernels, it is more difficult to prove consistency, although the known structure of the likelihood functions may help. In this section, we give conditions necessary for convergence.

First, the kernel must be positive-definite **?**. In addition, in order to ensure convergence, we must have that the following condition holds:

Identifiability: The prior correlation $\frac{k(\theta,\theta')}{\sqrt{k(\theta,\theta)}\sqrt{k(\theta',\theta')}}$ must be less than 1 if it is possible that $f(\theta) \neq f(\theta')$. That is to say, if two values of the likelihood function can be different, our model must not enforce that those two function values are identical.

Proposition 1. *The above condition is necessary to guarantee convergence of the BQ posterior to the true value of Z.*

Proof sketch. Consider a prior $p(\theta) = \frac{1}{2}\delta_{\theta_1}(\theta) + \frac{1}{2}\delta_{\theta_2}(\theta)$ where $\theta_1 \neq \theta_2$. If the prior correlation between $f(\theta_1)$ and $f(\theta_2)$ is one, then after observing one of those two values, say $f(\theta_1)$ we have that $\mathbb{E}_{\text{gp}}[Z] = f(\theta_1)$ and $\mathbb{V}_{\text{gp}}[Z] = 0$. However if $f(\theta_1) \neq f(\theta_2)$, then

$Z_0 = \frac{1}{2}f(\theta_1) + \frac{1}{2}f(\theta_2) \neq f(\theta_1)$. Thus the estimator will have converged to the wrong hypothesis. □

For a more detailed proof, see the supplementary materical. The statements in this section also hold for the problem of gp regression in general, but are specially releveant for the problem of inferring likelihood functions over latent structures. This is for two reasons: First, likelihood functions over structures can typically by shown to have many symmetries. Secondly, in the quadrature setting, we must learn about the function everywhere under the prior, not just in a small region or manifold, as is typical for regression problems. Exploiting the symmetries of the likelihood function is both possible, and necessary for fast convergence.

1.5 Inference in the Indian Buffet Process

In this section, we construct a kernel and demonstrate the use of bq for inference in an infinite latent model, the Indian buffet process. The Indian buffet process (ibp) **?** is a distribution over binary matrices, usually used as a prior over latent features of a set of items. The ibp is nonparametric in the sense that the number of latent features is unbounded. For example, in **?**, a model of images is constructed where the entries of a binary matrix specify which objects appear in which images. Although the number of objects is unknown beforehand, given a set of images, the flexible ibp prior allows inference on both the number of objects, and their presence over the dataset.

Under an ibp prior, the probability of seeing a matrix $\mathbf{Z}$ with K columns is

$$P(\mathbf{Z}|\alpha) = \prod_{k=1}^{K} \frac{\frac{\alpha}{K}\Gamma(m_k + \frac{\alpha}{K})\Gamma(N - m_k + 1)}{\Gamma(N + 1 + \frac{\alpha}{K})} \tag{1.10}$$

where m_k is the number of ones in column k, and α is the concentration parameter. There is an unfortunate clash of notation here, where $Z_{f,p}$ denotes the model evidence, and $\mathbf{Z}$ is used to denote a binary matrix.

To fully specify a model, we must also specify the likelihood of a set of observations $\mathbf{X}$ given the latent structure, $p(\mathbf{X}|\mathbf{Z})$. For simplicity, we will use as a simple example the linear-Gaussian model from **?**. This model assumes the data is generated as $\mathbf{X} = \mathbf{A}\mathbf{Z} + \epsilon$, with $A_{ij} \sim \mathcal{N}(0, \sigma_A)$ and $\epsilon_{ij} \sim \mathcal{N}(0, \sigma_X)$. The features in $\mathbf{A}$ are turned on and off for each row of $\mathbf{X}$ by the entries of $\mathbf{Z}$. Conveniently, we can integrate out the matrix $\mathbf{A}$ to

obtain a collapsed likelihood:

$$p(\mathbf{X}|\mathbf{Z},\sigma_X,\sigma_A) = \frac{\exp\left\{-\frac{1}{2\sigma_X^2}\operatorname{tr}(\mathbf{X}^T(\mathbf{I}-\mathbf{Z}(\mathbf{Z}^T\mathbf{Z}+\frac{\sigma_X^2}{\sigma_A^2}\mathbf{E}_)^{-1}\mathbf{Z}^T)\mathbf{X})\right\}}{(2\pi)^{ND/2}\sigma_X^{(N-K)D}\sigma_A^{KD}|\mathbf{Z}^T\mathbf{Z}+\frac{\sigma_X^2}{\sigma_A^2}\mathbf{I}|^{\frac{D}{2}}} \tag{1.11}$$

The goal of inference in the ibp is usually do discover statistics about the matrices $\mathbf{Z}$ and $\mathbf{A}$, or to produce a predictive distribution over new rows of $\mathbf{X}$. In this paper, we will show how to [...]

1.5.1 A kernel between binary matrices

Here we give a kernel which satisfies the guidelines given in section **??**. Since the likelihood in (**??**) does not depend on the order of columns of $\mathbf{Z}$, we will construct a kernel function $k(\mathbf{Z},\mathbf{Z}')$ which is also invariant to permutations over columns of both $\mathbf{Z}$ and $\mathbf{Z}'$:

$$k(\mathbf{Z},\mathbf{Z}') = \sum_{k=1}^{K}\sum_{k'=1}^{K'}\sum_{n=1}^{N}\mathbf{Z}_{n,k}\mathbf{Z}'_{n,k'} \tag{1.12}$$

This kernel has the property that, for a given number of ones in $\mathbf{Z}$ and $\mathbf{Z}'$, it attains its maximum value when every element of some permutation of columns of $\mathbf{Z}$ is equal to $\mathbf{Z}'$ (i.e., they are in the same equivalence class). [TODO: show that it achieves identifiability condition]

1.5.2 Computing z_n and the prior variance

Now that we have defined our prior and kernel, we can compute the "mini-normalization constants", $z_1,\ldots,z_n$, given by (**??**). These quantities represent the expected covariance of the likelihood of a latent structure θ', with the likelihood of another structure drawn from the prior.

If matrices $\mathbf{Z}$ and $\mathbf{Z}'$ have K and K' columns respectively, then combining (**??**) and (**??**), we have:

$$z_n(\mathbf{Z}) = \sum_{\mathbf{Z}'}k(\mathbf{Z},\mathbf{Z}')p(\mathbf{Z}'|\alpha) = \sum_{\mathbf{Z}'}\left[\sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{k'=1}^{K'}\mathbf{Z}_{n,k}\mathbf{Z}'_{n,k'}\right]\left[\prod_{k^*=1}^{K}P(\mathbf{Z}'_{:,k^*}|\alpha)\right] \tag{1.13}$$

$$= \sum_{k=1}^{K}\sum_{n=1}^{N}\mathbf{Z}_{n,k}\frac{\alpha}{1+\frac{\alpha}{K'}} = \alpha\sum_{k=1}^{K}\sum_{n=1}^{N}\mathbf{Z}_{n,k} \qquad \text{as } K'\to\infty \tag{1.14}$$

See the supplementary material for longer derivations. We can interpret (**??**) as saying that the prior covariance of the likelihood $p(\mathbf{X}|\mathbf{Z})$ with a randomly drawn likelihood $p(\mathbf{X}|\mathbf{Z}')$ is proportional to the number of ones in $\mathbf{Z}$.

Next we compute the prior variance (**??**), which follows a similar derivation. This is the expected variance of Z before we have seen any data.

$$V_{\text{prior}} = \sum_{\mathbf{Z}'} \sum_{\mathbf{Z}} p(\mathbf{Z}|\alpha) k(\mathbf{Z}, \mathbf{Z}') p(\mathbf{Z}'|\alpha) = \frac{\alpha^2 N}{1 + \frac{\alpha}{K}} = \alpha^2 N \qquad \text{as } K \to \infty \tag{1.15}$$

This form is intuitive: it scales with the expected number of ones per row, α. As α or N approach zero, the likelihood surface can be expected to become less varied, since there will be fewer ways that individual samples of $\mathbf{Z}$ can differ.

1.6 Infinite Mixture Models

In this section we develop a more general example, placing a kernel between mixture distributions. This will allow us to perform model-based inference in Dirichlet process mixture models.

A finite mixture model with k components gives a distribution over the observations x as follows: $p(x|\pi, \theta) = \sum_{i=1}^{k} \pi_i p(x|\theta_i)$ where θ_i represents a single set of mixture parameters. By specifying the form of θ, we can perform inference in a wide variety of infinite mixture models, such as an latent Dirichlet allocation-type model, infinite mixture of regressors, or a mixture of Gaussians.

We will divide our derivation into two parts. First, we will define a kernel between multinomial distributions, and derive (**??**) and (**??**) for this kernel, leaving unspecified the kernel between individual mixture elements. Then, we will continue the derivation for an infinite mixture of Gaussians.

1.6.1 A Kernel Between Multinomial Distributions

Here we define a kernel between multinomial distributions, possibly of different dimension. Here, π represents weights which sum to one, and θ the atoms of the multinomial distribution.

$$k(\pi, \theta, \pi', \theta') = \sum_{i}^{n_\theta} \sum_{j}^{n_{\theta'}} \pi_i \pi'_j k_a(\theta_i, \theta'_j) \tag{1.16}$$

where $k_a(\theta, \theta')$ specifies the covariance between individual atoms of the distribution. Changing the order of mixture components, or splitting a given mixture component among two identical atoms, will not affect the value of this covariance function. If $k_a(\theta_i, \theta'_j)$ is a Mercer kernel, then so is (**??**).

If our mixture θ has n_θ components, then

$$z(\pi, \theta) = \iint k(\pi, \theta, \pi', \theta') p(\theta') p(\pi') d\theta' d\pi' = \iint \left[\sum_{i=1}^{n_\theta} \sum_{j=1}^{n_{\theta'}} \pi_i \pi'_j k(\theta_i, \theta'_j) \right] \left[\prod_a p(\theta'_a) \right] p(\pi') d\theta' d\pi'$$

$$= \int \sum_{i=1}^{n_\theta} \sum_{j=1}^{n_{\theta'}} \pi_i \pi'_j \underbrace{\int k(\theta_i, \theta'_j) p(\theta'_j) d\theta'_j}_{z_a(\theta_i)} p(\pi') d\pi' = \sum_{i=1}^{n_\theta} \pi_i z_a(\theta_i) \tag{1.17}$$

$$\mathbb{V}_{\text{prior}}[Z_{f,p}] = \iint z(\pi, \theta) p(\theta) p(\pi) d\theta d\pi = \underbrace{\int z(\theta_i) p(\theta_i) d\theta_i}_{V_a} \underbrace{\int p(\pi) \sum_{i=1}^{n_\theta} \pi_i d\pi}_{\text{sums to one}} = V_a \tag{1.18}$$

where $V_a = \iint p(\theta) k_a(\theta, \theta') p(\theta') d\theta' d\theta$ is the prior variance of k_a, which will be defined below.

Perhaps surprisingly, neither z_n nor V_{prior} depend on the number of proposed mixture components. This means that we are free to take the infinite limit $n_\theta \to \infty$. Perhaps this makes sense: The likelihood does not change if we divide up our clusters to give equivalent mixtures but having more components. These quantities also do not depend on the form of the prior over mixture components $p(\pi)$, nor the prior over individual components $p(\theta)$, as long as it factorizes over components.

1.6.2 Infinite Mixture of Gaussians

The above kernel between mixtures can be used for inference in a wide variety of infinite mixture models. For simplicity, in this paper we will use as an example the infinite mixture of Gaussians **?**:

$$p(x|\pi, \theta) = \sum_{i=1}^{k} \pi_i p(x|\theta_i) = \sum_{i=1}^{k} \pi_i \mathcal{N}(y|\mu_i, \Sigma_i) \tag{1.19}$$

where $\theta_i = \{\mu_i, \Sigma_i\}$ represent the parameters of a single Gaussian. To complete our example, all that remains is to specify a kernel k_a between individual mixture components, and to compute z_k and V_a. For simplicity, we will specify a Gaussian kernel between

densities, and assume that the variance of each mixture component is identical:

$$k(\mu, \Sigma, \mu', \Sigma') = \mathcal{N}(\mu|\mu', \Sigma_k) \tag{1.20}$$

where the entries of Σ_k are kernel parameters. Next, we give (**??**) and (**??**) for this kernel:

$$z_k(\mu_i, \Sigma_i) = \iint k(\mu_i, \Sigma_i, \mu'_j, \Sigma'_j) p(\mu'_j, \Sigma'_j) d\mu'_j d\Sigma'_j = \mathcal{N}(\mu_i|\lambda, \Sigma_k + \Sigma_p) \tag{1.21}$$

$$V_a = \int z_k(\mu_i, \Sigma_i) p(\mu_i, \Sigma_i) d\mu_i d\Sigma_i = \mathcal{N}(0|0, \Sigma_k + 2\Sigma_p) \tag{1.22}$$

Note that the prior variance V_k depends only on the shape of the prior Σ_p, and not its location.

1.7 Related Work

String kernels **?**. Graph kernels. Tree-structured kernels. Sequence kernels. A review of kernels in structured domains can be found in **?**.

1.8 Experiments

Integrating Kernel Hyperparameters One complicating issue of inference using bq is how to set kernel hyperparameters. In **?**, these were set by maximum likelihood. In our experiments, hyperparameters were integrated out numerically, except for the *output variance* (a scale factor in front of the kernel) which is possible to integrate out in closed form, giving a final posterior variance of $\sigma_2 = \frac{1}{N} [\mathbf{y}^t \mathbf{K}^{-1} \mathbf{y}] [V_k - \mathbf{z}_t \mathbf{K}^{-1} \mathbf{z}]$. For a derivation, see the supplementary material.

1.8.1 IBP Experiments

We used collapsed IBP sampling code from **?** to obtain samples and likelihood values. We used data from **?**, where the true value of $\sigma_x = 0.5$. We set the number of datapoints to be small (N = 25), since this is a regime where VB is known to perform poorly **?**.

Figure **??** demonstrates the use of bq, showing that a set of samples gathered under one parameter setting can be used to make inferences the likelihood of other parameter settings. This allows us to use incorrect samplers, use samples from the burn-in period,

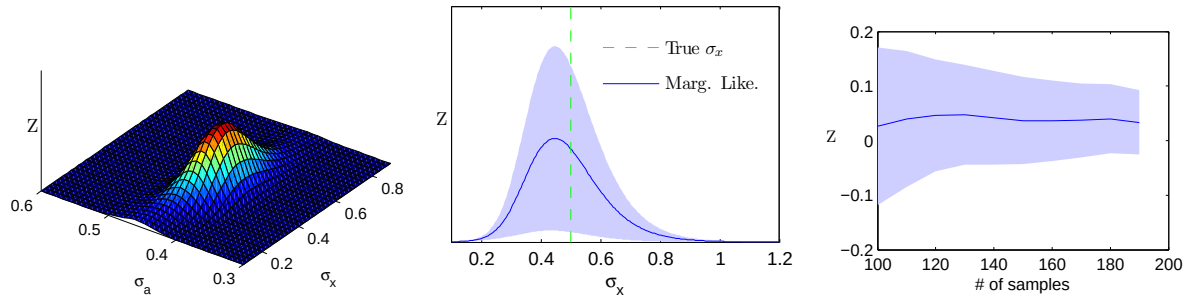

Fig. 1.2 Computing marginal likelihoods with BQ. Left: The marginal likelihood as a function of two nuisance parameters. Center: A slice of the 2-D marginal likelihood function, at $\sigma_A = 0.4$. The true value of σ_X lies roughly in the center of the likelihood function. Right: The marginal likelihood as a function of the number of evaluations of the likelihood function. The shaded error represents uncertainty about the value of the marginal likelihood.

etc. For example, the likelihood function (**??**) has two "nuisance parameters", σ_X and σ_A. In [cite Finale], these parameters are simply set in an ad-hoc way. This may be acceptable, but using MCMC, it is not clear how to tell whether the answer is sensitive to these nuisance parameters without re-running the chain under several different settings. Figure **??** shows that bq allows one to run a post-hoc sensitivity analysis to check whether extra parameters are important to the analysis. In addition, we recover an estimate of the certainty of our analysis, indicating whether or not the existing samples are sufficient to draw strong conclusions.

1.8.2 DP Mixture Experiments

Figure **??** demonstrates the use of bq to examine sensitivity to prior parameters.

Figures **??** and **??** show not only the marginal likelihood of different parameter settings, but also the marginal variance of the likelihood functions. We must distinguish between two types of uncertainty represented here. First, the shape of the likelihood function indicates our posterior uncertainty over its parameter, given the data. Second, the gp posterior uncertainty in the likelihood function(represented by the shaded areas) represents our uncertainty about this likelihood function. Our uncertainty about the likelihood function can be made arbitrarily small by continuing to sample the likelihood function. However, we may still remain uncertain about the value of the latent parameter.

Code to produce all experiments will be made available at the authors' website.

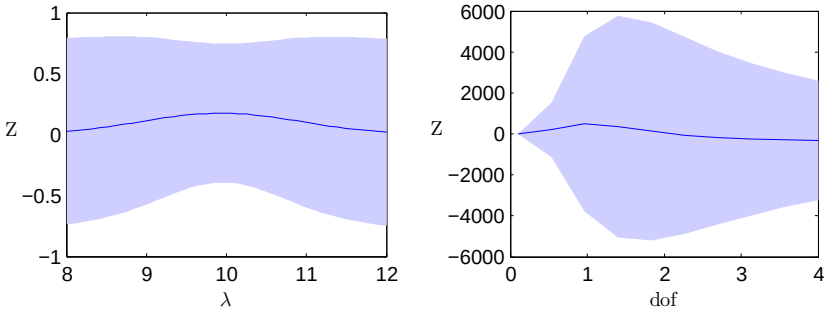

Fig. 1.3 A demonstration of computing marginal likelihoods. Left: The marginal likelihood versus the prior mean. Right: Marginal likelihood versus the degrees of freedom of the mixture components. dof $= 1$ is Cauchy, dof $= 2$ is Student's t, dof $= \infty$ is Gaussian. The shaded error represents uncertainty about the value of the marginal likelihood.

1.9 Discussion

Nuisance parameters such as σ_X could also be dealt with in the bq by adding them to the kernel and the domain of integration if desired.

1.9.1 Appropriateness of the GP prior

Placing a gp prior on the likelihood function allows us to take advantage of our knowledge of the smoothness of this function. However, there is ample reason to be believe that the gp prior is inappropriate for modeling likelihood functions. In **?** it is suggested to place the gp on the log-likelihood function, which would presumably make the additive form of the kernels given in the paper much more appropriate. This was done by [cite BQR paper], and resulted in better performance, at the cost of a much more complicated inference procedure.

As is generally true for Bayesian methods, there exist many cases where bq will underestimate its uncertainty. Our response to this objection is that any uncertainty estimate is better than none at all. In our experiments, we observed cases in which the model's posterior uncertainty is significant, alerting us to the fact that we have not yet observed enough about the likelihood function to be certain about its shape. In the case of MCMC, this sort of uncertainty estimate is typically unavailable. That is to say, our model cannot account for 'unknown unknowns', but it can at least account for 'known unknowns', which is a step up from the point estimates provided by MCMC.

1.10 Conclusions

We have extended Bayesian quadrature to infinite, structured domains, and demonstrated that this method can be used for inference in nonparametric models. We have given necessary conditions for convergence and examples of how to construct kernels which take advantage of the many symmetries of typical likelihood functions. We demonstrated some properties of this method, which include uncertainty estimates, flexibility in sampling methods, and the ability to re-use samples from on setting to perform post-hoc sensitivity analysis of nuisance parameters.

References

F. Doshi-Velez and Z. Ghahramani. Accelerated sampling for the indian buffet process. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 273–280. ACM, 2009. (page 11)

T. Gärtner. A survey of kernels for structured data. *ACM SIGKDD Explorations Newsletter*, 5(1):49–58, 2003. (page 10)

T. Griffiths and Z. Ghahramani. Infinite latent feature models and the indian buffet process. 2005. (page 6)

H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins. Text classification using string kernels. *The Journal of Machine Learning Research*, 2:419–444, 2002. (page 10)

K.T. Miller, J. Van Gael, and Y.W. Teh. Variational inference for the indian buffet process. In *Proceedings of the Intl. Conf. on Artificial Intelligence and Statistics*. Citeseer, 2009. (page 11)

A. O'Hagan. Bayes-Hermite quadrature. *Journal of Statistical Planning and Inference*, 29:245–260, 1991. (page 2)

C. E. Rasmussen and Z. Ghahramani. Bayesian monte carlo. In S. Becker and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15. MIT Press, Cambridge, MA, 2003. (pages 2, 4, 10, and 12)

C.E. Rasmussen. The infinite Gaussian mixture model. *Advances in Neural Information Processing Systems*, 12(5.2):2, 2000. (page 10)

C.E. Rasmussen and CKI Williams. Gaussian Processes for Machine Learning. *The MIT Press, Cambridge, MA, USA*, 2006. (page 5)

F. Wood and T.L. Griffiths. Particle filtering for nonparametric bayesian matrix factorization. *Advances in Neural Information Processing Systems*, 19:1513, 2007. (page 11)