
A Kernel Approach to Tractable Bayesian Nonparametrics

Ferenc Huszár
University of Cambridge

Simon Lacoste-Julien
University of Cambridge

Abstract

Inference in popular nonparametric Bayesian models typically relies on sampling or other approximations. This paper presents a general methodology for constructing novel tractable nonparametric Bayesian methods by applying the kernel trick to inference in a parametric Bayesian model. For example, Gaussian process regression can be derived this way from Bayesian linear regression. Despite the success of the Gaussian process framework, the kernel trick is rarely explicitly considered in the Bayesian literature. In this paper, we aim to fill this gap and demonstrate the potential of applying the kernel trick to tractable Bayesian parametric models in a wider context than just regression. As an example, we present an intuitive Bayesian kernel machine for density estimation that is obtained by applying the kernel trick to a Gaussian generative model in feature space.

1 Introduction

The popularity of nonparametric Bayesian methods has steadily risen in machine learning over the past decade. Bayesian inference in almost all current nonparametric models relies on approximations which typically involve Markov chain Monte Carlo, or more recently variational approximations [1]. There is an ever-growing interest in developing nonparametric Bayesian methods in which inference and prediction can be expressed in closed form and no approximations are needed. Such methods would be quite useful and desired in many applications, but are unfortunately rare at present. Perhaps the only known example is Gaussian process (GP) regression. In GP regression [2], the posterior predictive distribution is a tractable Gaus-

sian with parameters that can be computed from data exactly in polynomial time. The method is widely adopted and also has favourable frequentist asymptotic properties [3]. But what is the secret behind the remarkable algorithmic clarity of GP regression? What makes closed-form computations possible? We argue that the key is that GP regression is also a *kernel machine*: the method is arrived at by applying the kernel trick in a parametric Bayesian model, namely linear regression (see e.g. Chapter 2 in [2])

Kernel methods [4, 5] use a simple trick, widely known as the kernel trick, to overcome the limitations of a linear model: the observations $\mathbf{x}_i \in \mathcal{X}$ are first embedded in a feature space \mathcal{F} using a nonlinear mapping $\varphi: \mathcal{X} \mapsto \mathcal{F}$. A linear algorithm is then applied on the embedded representations $\phi_n = \varphi(\mathbf{x}_n)$ instead of the observations themselves. If the algorithm only makes use of scalar products $\langle \phi_n, \phi_m \rangle$, then by replacing all scalar products by tractable *kernel* evaluations $k(\mathbf{x}_n, \mathbf{x}_m) := \langle \phi_n, \phi_m \rangle$, the expressive power can be substantially increased with only a minor increase in computational costs. Notably, the kernel trick allows one to construct nonparametric machine learning methods from parametric ones.

Despite its popularity in “non-Bayesian” studies, the kernel trick is rarely considered as a construction tool in Bayesian nonparametrics. GP regression is a rare, if not the only example. GPs are therefore often called *Bayesian kernel machines*. In this paper, we consider finding new examples of Bayesian kernel machines, i.e. broadening the intersection between kernel machines and nonparametric Bayesian methods. For Bayesian nonparametrics, the kernel approach offers invaluable closed-form computations and a rigorous analysis framework associated with reproducing kernel Hilbert spaces (RKHS). For kernel machines, a Bayesian formulation offers benefits, such as novel probabilistic approaches to setting kernel hyperparameters and means of incorporating such methods in hierarchical Bayesian models [6, 7, 8].

In this paper, we present a methodology for finding novel Bayesian kernel machines, following the recipe of GP regression:

1. Start with a simple Bayesian model of observations.
2. Derive exact Bayesian inference in the model.
3. Express the posterior predictive distribution in terms of dot products and apply the kernel trick.

The crucial point is finding the basic model in step 1, in which both steps 2 and 3 are possible. Fortunately, this search is guided by intuitive orthonormal invariance considerations that will be demonstrated in this paper. We present an example Bayesian kernel machine for density estimation, based on the linear Gaussian generative model underlying principal component analysis (PCA) [9]. We show that the kernel trick can be applied in the Bayesian method by choosing prior distributions over the parameters which preserve invariance to orthonormal transformations.

The rest of the paper is organised as follows. In Sec. 2, we review Bayesian inference in a Gaussian generative model and discuss consequences of applying the kernel trick to the predictive density. We consider the infinite dimensional feature spaces case in Subsec. 2.2. In Sec. 4, we present experiments on high dimensional density estimation problems comparing our method to other Bayesian and non-Bayesian nonparametric methods.

2 A Gaussian model in feature space

Assume that we have observations \mathbf{x}_i in a d -dimensional Euclidean space and that our task is to estimate their density. In anticipation of the sequel, we embed the observations \mathbf{x}_i into a high-dimensional feature space \mathcal{F} with an injective smooth nonlinear mapping $\varphi: \mathcal{X} \mapsto \mathcal{F}$. Our density estimation method is based on a simple generative model on the ambient feature space of the embedded observations $\phi_i = \varphi(\mathbf{x}_i)$. For now, think of \mathcal{F} as a D -dimensional Euclidean space and ϕ_i as *arbitrary elements* of \mathcal{F} (i.e. they are not necessary constrained to lie on the *observation manifold* $\mathcal{O} = \{\varphi(\mathbf{x}) : \mathbf{x} \in \mathcal{X}\}$). We suppose that ϕ_i were sampled from a Gaussian distribution with unknown mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$:

$$\phi_{1:N} | \boldsymbol{\mu}, \boldsymbol{\Sigma} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \text{ i. i. d.} \quad (1)$$

The notation $\phi_{1:N}$ is used to denote the set of vectors ϕ_1 up to ϕ_N .

Now we will consider estimating parameters of this model in a Bayesian way. In Bayesian inference, one

defines prior distributions over the parameters and then uses Bayes' rule to compute the posterior over them. Importantly, now we also require that the resulting Bayesian procedure is still amenable to the kernel trick. We therefore start by discussing a necessary condition for kernelisation which can then guide our choice of prior distributions.

The kernel trick requires that the algorithm be expressed solely in terms of scalar products $\langle \phi_i, \phi_j \rangle$. Scalar products are invariant under orthonormal transformations of the space, i.e. if \mathcal{A} is an orthonormal transformation then $\langle \mathbf{u}, \mathbf{v} \rangle = \langle \mathcal{A}\mathbf{u}, \mathcal{A}\mathbf{v} \rangle$ for all \mathbf{u} , and \mathbf{v} in the space. Thus, if one wants to express an algorithm in terms of scalar products, it has to be – at least – invariant under orthonormal transformations, such as rotations, reflections and permutations. It is well known that PCA has this property, but, for example, factor analysis (FA) does not [9], thus one cannot expect a kernel version of FA without any restrictions.

Another desired property of the method is analytical convenience and tractability, which can be ensured by using conjugate priors. The conjugate prior of the Gaussian likelihood is the *Normal-inverse-Wishart*, which in our case has to be restricted to meet the orthonormal invariance condition:

$$\begin{aligned} \boldsymbol{\Sigma}; \sigma_0^2, \alpha &\sim \mathcal{W}^{-1}(\sigma_0^2 \mathbf{I}, \alpha) \\ \boldsymbol{\mu} | \boldsymbol{\Sigma}, \beta &\sim \mathcal{N}\left(\mathbf{0}, \frac{1}{\beta} \boldsymbol{\Sigma}\right), \end{aligned} \quad (2)$$

where \mathcal{W}^{-1} and \mathcal{N} denote the inverse-Wishart and Gaussian distributions with the usual parametrisation. The general Normal-inverse-Wishart family had to be restricted in two ways: firstly, the mean of $\boldsymbol{\mu}$ was set to zero; secondly, the scale matrix of the inverse-Wishart was set to be spherical. These sensible restrictions ensure that the marginal distribution of ϕ is centered at the origin and is spherically symmetric and therefore orthonormal invariance holds, which is required for kernelisation.

Having defined the hierarchical generative model in eqns. (1)–(2), our task is to estimate the density of ϕ 's given the previous observations $\phi_{1:N}$, which in a Bayesian framework is done by calculating the following posterior predictive distribution:

$$p(\phi | \phi_{1:N}; \sigma_0^2, \alpha, \beta) = \int p(\phi | \boldsymbol{\mu}, \boldsymbol{\Sigma}) p(\boldsymbol{\mu}, \boldsymbol{\Sigma} | \phi_{1:N}; \sigma_0^2, \alpha, \beta) d\boldsymbol{\mu} d\boldsymbol{\Sigma}$$

By straightforward computation, it can be shown that the posterior predictive distribution is a D -dimensional student- \mathcal{T} distribution of the form (with

the dependence on hyper-parameters made implicit):

$$p(\phi|\phi_{1:N}) \propto \quad (3)$$

$$\left(\gamma + \tilde{\phi}^\top \left(\sigma_0^2 \mathbf{I} + \tilde{\Phi} \left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^\top}{N + \beta} \right) \tilde{\Phi}^\top \right)^{-1} \tilde{\phi} \right)^{-\frac{1+N+\alpha}{2}}$$

where $\gamma = \frac{1+\beta+N}{\beta+N}$, $\tilde{\phi} = \phi - \frac{N\bar{\phi}}{N+\beta}$, $\tilde{\Phi} = [\phi_1 - \bar{\phi}, \dots, \phi_N - \bar{\phi}]$, $\bar{\phi} = \frac{1}{N} \sum_{n=1}^N \phi_n$ is the empirical mean in feature space and $\mathbf{1}$ is a $N \times 1$ vector of ones.

In order to obtain an expression which only contains scalar products, we invoke Woodbury's matrix inversion formula [10]:

$$p(\phi|\phi_{1:N}) \propto \left(\gamma + \frac{\tilde{\phi}^\top \tilde{\phi}}{\sigma_0^2} - \quad (4)$$

$$\tilde{\Phi}^\top \tilde{\Phi} \left(\sigma_0^4 \left(\mathbf{I} + \frac{\mathbf{1}\mathbf{1}^\top}{\beta} \right) + \sigma_0^2 \tilde{\Phi}^\top \tilde{\Phi} \right)^{-1} \tilde{\Phi}^\top \tilde{\phi} \right)^{-\frac{1+N+\alpha}{2}}.$$

2.1 Kernel trick

Until now, we assumed that ϕ was an arbitrary point in D -dimensional space. In reality, however, we only want to assign probabilities (4) to points on the so-called observation manifold, $\mathcal{O} = \{\varphi(\mathbf{x}) : \mathbf{x} \in \mathcal{X}\}$, i. e. to points that can be realised by mapping an observation $\mathbf{x} \in \mathcal{X}$ to feature space $\phi = \varphi(\mathbf{x})$. Restricted to \mathcal{O} , we can actually make use of the kernel trick, assuming as well that $\phi_i = \varphi(\mathbf{x}_i)$ for the previous observations. Indeed, Eq. (4) then only depends on $\mathbf{x}_{1:N}$ and \mathbf{x} through $\tilde{\phi}^\top \tilde{\phi}$, $\tilde{\phi}^\top \tilde{\Phi}$ and $\tilde{\Phi}^\top \tilde{\Phi}$, which can all be expressed in terms of pairwise scalar products $\phi_n^\top \phi_m = \langle \varphi(\mathbf{x}_n), \varphi(\mathbf{x}_m) \rangle = k(\mathbf{x}_n, \mathbf{x}_m)$ as follows:

$$\tilde{\phi}^\top \tilde{\phi} = k(\mathbf{x}, \mathbf{x}) - 2 \frac{\sum_i k(\mathbf{x}, \mathbf{x}_i)}{N + \beta} + \frac{\sum_{i,j} k(\mathbf{x}_i, \mathbf{x}_j)}{(N + \beta)^2} \quad (5)$$

$$\left[\tilde{\phi}^\top \tilde{\Phi} \right]_n = k(\mathbf{x}, \mathbf{x}_n) - \frac{\sum_i k(\mathbf{x}, \mathbf{x}_i)}{N} + \frac{\sum_{i,j} k(\mathbf{x}_i, \mathbf{x}_j) - N \sum_i k(\mathbf{x}_n, \mathbf{x}_i)}{N(N + \beta)}$$

$$\left[\tilde{\Phi}^\top \tilde{\Phi} \right]_{n,m} = k(\mathbf{x}_n, \mathbf{x}_m) - \frac{\sum_i k(\mathbf{x}_n, \mathbf{x}_i) + \sum_i k(\mathbf{x}_m, \mathbf{x}_i)}{N} + \frac{\sum_{i,j} k(\mathbf{x}_i, \mathbf{x}_j)}{N^2}$$

As said previously, we are only interested in points in the (curved) observation manifold. The restriction of the predictive distribution (4) to the observation manifold induces the same density function

$q(\varphi(\mathbf{x})|\mathbf{x}_{1:N}) = p(\varphi(\mathbf{x})|\phi_{1:N})$ (which is now unnormalised), but with respect to a d -dimensional Lebesgue base measure in the geodesic coordinate system of the manifold. Finally, we map the density $q(\varphi(\mathbf{x})|\mathbf{x}_{1:N})$ back to the original input space by implicitly inverting the mapping φ , yielding the (unnormalised) predictive density $q_k(\mathbf{x}|\mathbf{x}_{1:N})$ defined on the input space. By doing so, we need to include a multiplicative Jacobian correction term which relates volume elements in the tangent space of the manifold to volume elements in the input space \mathcal{X} :

$$q_k(\mathbf{x}|\mathbf{x}_{1:N}) = q(\varphi(\mathbf{x})|\mathbf{x}_{1:N}) \cdot \det \left(\left\langle \frac{\partial \varphi(\mathbf{x})}{\partial x^i}, \frac{\partial \varphi(\mathbf{x})}{\partial x^j} \right\rangle \right)_{i,j=1,\dots,d}^{\frac{1}{2}}. \quad (6)$$

This formula can be derived from the standard change-of-variable formula for densities, as we show in the Appendix A.

So what exactly did we gain by applying the kernel trick? Despite the fact that the simple density model in the whole feature space is unimodal, in the local coordinate system of the non-linear observation manifold, the predictive density may appear multimodal and hence possess interesting nonlinear features. This is illustrated in Fig. 1. Assume that we observe draws from a complicated distribution, which cannot be conveniently modelled with a linear Gaussian model (Fig. 1.A). We map observations to a two dimensional feature space using the mapping $\varphi^1(x) = x, \varphi^2(x) = (x)^2$ hoping that the embedded observations are better fitted by a Gaussian, and carry out inference there (Fig. 1.B). Note how all possible observations in the original space get mapped to the observation manifold \mathcal{O} , which is now the parabola $\phi^2 = (\phi^1)^2$. The outcome of inference is a posterior predictive distribution in feature space, which takes a student- \mathcal{T} form. To obtain the predictive distribution in the observation space we have to look at the magnitude of the predictive distribution along the observation manifold. This is illustrated by the contour lines in panel B. The restricted predictive distribution is then ‘‘pulled back’’ to observation space and has to be multiplied by the Jacobian term (Fig. 1.C) to yield the final estimation of the density of the observed samples (Fig. 1.D). Remarkably, our method computes the unnormalised density estimate (before the Jacobian term is added, Fig. 1.C) directly from the data (Fig. 1.A). All intermediate steps, including the inversion of the mapping, are *implicit* in the problem formulation, and we never have to work with the feature space directly. As the method essentially estimates the density by a student- \mathcal{T} in feature space, we shall call it *kernel student- \mathcal{T} density estimation*.

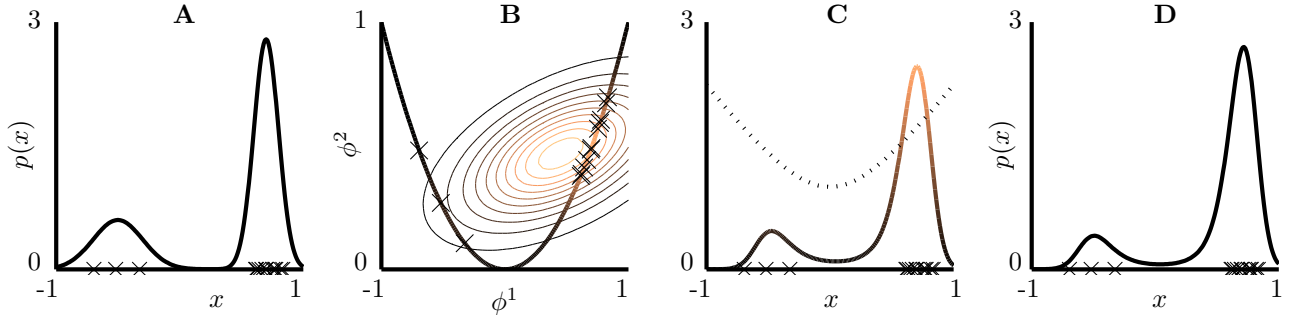


Figure 1: Illustration of kernel student- \mathcal{T} density estimation on a toy problem. **A**, Fourteen data points $x_{1:14}$ (crosses) drawn from a mixture distribution (solid curve). **B**, Embedded observations $\phi_{1:14}$ (crosses) in the two dimensional feature space and contours of the predictive density $q(\phi_{15}|x_{1:14})$. The observation manifold is a parabola (solid curve), and we are interested in the magnitude of the predictive distribution to this manifold (colouring). **C**, The restricted posterior predictive distribution pulled back to the real line (solid curve) gets multiplied by the Jacobian term $\sqrt{1+4x^2}$ (dotted curve) to give **D**, the predictive distribution $q_k(x_{15}|x_{1:14})$ (solid curve) in observation space. All distributions are scaled so that they normalise to one.

2.2 Infinite dimensional feature spaces

Of course, we constructed the previous toy example so that the simple normal model in this second order polynomial feature space is able to model the bimodal data distribution. Should the data distribution be more complicated, e.g. would have three or more modes, the second order features would hardly be enough to pick up all relevant statistical properties. Intuitively, adding more features results in higher potential expressive power, while the Bayesian integration safeguards us from overfitting. In practice, therefore, we will use the method in conjunction with rich, perhaps infinite dimensional feature spaces, which will allow us to relax these parametric restrictions and to apply our method to model a wide class of probability distributions.

However, moving to infinite dimensional feature spaces introduces additional technical difficulties. As for one, Lebesgue measures do not exist in these spaces, therefore our derivations based on densities with respect to Euclidean base measure should be reconsidered. For a fully rigorous description of the method in infinite spaces, one may consider using Gaussian base measures instead. In this paper, we address two specific issues that arise when the feature space is large or infinite dimensional.

Firstly, because of the Jacobian correction term, the predictive density q_k in input space is still not fully kernelised in (6). We note though that this term is the determinant of a small $d \times d$ matrix of inner products, which can actually be computed efficiently for some kernels. For example, the all-subsets kernel [11] has a feature space with exponential dimension $D = 2^d$, but

each inner product of derivatives can be computed in $\mathcal{O}(d)$. But what about nonparametric kernels with infinite D ? Interestingly, we found that shift invariant kernels – for which $k(\mathbf{x}, \mathbf{y}) = k(\mathbf{x} + \mathbf{z}, \mathbf{y} + \mathbf{z})$ for all \mathbf{z} – actually yield a *constant Jacobian term* (see proof in the Appendix B) so it can be safely ignored to obtain a fully kernelised – although unnormalised – predictive density $q_k(\mathbf{x}|\mathbf{x}_{1:N}) = q(\varphi(\mathbf{x})|\mathbf{x}_{1:N})$. This result is important, as the most commonly used nonparametric kernels, such as the squared exponential, or the Laplacian fall into this category.

The second issue is normalisation: Eq. (6) only expresses the predictive distribution up to a multiplicative constant. Furthermore, the derivations implicitly assumed that $\alpha > D - 1$, where D is the dimensionality of the feature space, otherwise the probability densities involved become improper. This assumption clearly cannot be satisfied when D is infinite. However, one can argue that even if the densities in the feature space are improper, the predictive distributions that we use may still be normalisable when restricted to the observation manifold for $\alpha > d - 1$. Indeed, the two-dimensional student- \mathcal{T} is normalisable when restricted to the parabola as in Fig. 1 for all $\alpha > 0$ rather than $\alpha > 1$ (to show this, consider $N = 0$ so that (3) takes the simple form $q(\phi) = (\text{cnst.} + \phi^T \phi)^{-\frac{1+\alpha}{2}}$). So although distributions in feature space may be improper, the predictive densities may remain well-defined even when nonparametric kernels are used.

Given these observations, we thus decide to *formally* apply equation (6) for our predictive density estimation algorithm, ignoring the normalisation constant and also the Jacobian term when shift-invariant nonparametric kernels are applied.

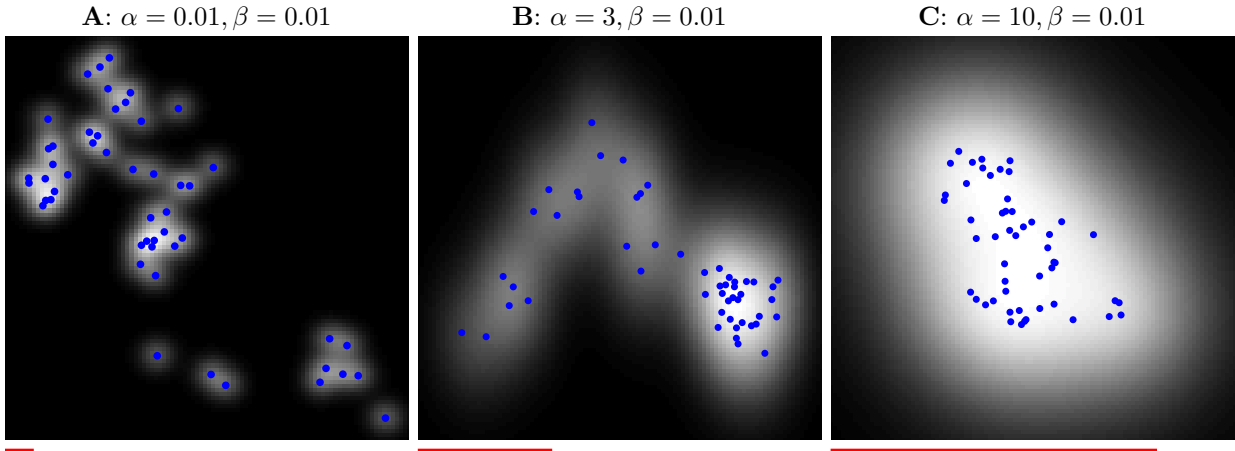


Figure 2: Fantasy data drawn from a two dimensional kernel student- \mathcal{T} model with SE kernel. The length-scale $\ell = 1$ (bars below panels) and $\beta = 0.01$ are fixed, α varies between 0.01 and 10 (A to C). Panels show 50 points (blue dots) drawn from the generative model and the predictive density (gray level) of the 51st draw.

Finally, we want to emphasize that the method does not imply nor does it require that arbitrary distributions, when embedded in rich Hilbert spaces, will look exactly like Gaussian or student- \mathcal{T} measures. Instead, the method uses the potentially infinitely flexible family of Gaussian measures to approximately model distributions in the feature space, and uses Bayesian integration to address model-misspecification via explicit representation of uncertainty. As demonstrated in Fig. 1, the richness and flexibility of the density model stems from projecting the distribution onto a nonlinear observation manifold, allowing us to overcome limitations of Gaussian measures, such as unimodality.

3 Related work

Our method is closely related to kernel principal component analysis [12, kPCA] as both of them essentially use the same underlying generative model (Eq. 1). While our method is based on Bayesian inference, kPCA performs constrained maximum likelihood estimation [13] of parameters μ and Σ . kPCA is a very effective and popular method for nonlinear feature extraction. Its pitfall from a density estimation perspective is that it does not generally induce a sensible probability distribution in observation space. To understand this, consider performing kPCA of order one (i.e. recover just a single nonlinear component) on the toy data in Fig. 1.A. The solution defines a degenerate Gaussian distribution in feature space which is concentrated along a line, roughly the long axes of the equiprobability ellipses in Fig. 1.B. As such a distribution can intersect the observation manifold at most twice, the predictive distribution in observation space will be a mixture of two delta distributions. This

degeneracy can be ruled out by recovering at least as many nonlinear components as the dimensionality of feature space, but this is clearly not a viable option in infinite or large dimensional feature spaces. The Bayesian approach sidesteps the degeneracy problem by integrating out the mean and covariance, thereby averaging many, possibly degenerate distributions to obtain a non-degenerate, smooth student- \mathcal{T} distribution in feature space, which results in a smooth density estimate in observation space.

Another particularly interesting related topic is that of *characteristic kernels* [14]. A positive definite kernel $k(x, x') = \langle \varphi(x), \varphi(x') \rangle$ is called characteristic if the mean element $\mu_\pi = \mathbb{E}_{\mathbb{X} \sim \pi}[\varphi(\mathbb{X})]$ uniquely characterises any probability measure π . From a density estimation perspective, this suggests that estimating a distribution in observation space boils down to estimating the mean in a characteristic kernel space. The kernel moment matching framework [15] tries to exploit characteristic kernels in a very direct way: parameters of an approximating distribution are chosen by minimising maximum mean discrepancy in feature space. Our method can be interpreted as performing Bayesian estimation of first and second moments in feature space, therefore one may hope that work on characteristic kernels will help us further study and understand properties of the algorithm.

The Gaussian process has been used as a nonparametric prior over functions for unsupervised learning tasks in several studies, such as in Gaussian process latent variable models [16, GPLVMs] or the Gaussian process density sampler [8, GPDS]. What sets the present approach apart from these is that while they incorporated Gaussian processes as a building block in an un-

METHOD	NOVELTY (AuC[%])	RECONS. (conf[%])
<i>kst</i>	96.82 \pm 0.8	2.15 \pm 1.0
<i>kde</i>	94.56 \pm 0.6	9.24 \pm 1.9
<i>dpm</i>	73.56 \pm 4.3	23.85 \pm 8.2

Table 1: Performance of *kst*, *kde* and *dpm* in novelty detection (*novelty*, *left*) and label reconstruction (*recons.*, *right*) on the USPS data. The table shows average AuC and confusion values \pm one standard deviation measured over 10 randomised experiments.

supervised model, we re-used the construction of GP regression and explicitly applied the kernel trick in an parametric unsupervised method. The GPDS method relies on Markov chain Monte Carlo (MCMC) methods for so called doubly-intractable distributions [17]. These methods can be used to sample from the posterior over hyper-parameters of probabilistic methods where normalisation is intractable. In particular, they could be used to integrate out the hyper-parameters α, β and σ_0 , of our method. We emphasise that while MCMC is a crucial component of the GPDS, inference in our model is essentially closed-form.

4 Experiments

A common way to examine properties of unsupervised learning models is to draw *fantasy datasets* from the generative model. The kernel student- \mathcal{T} model can be simulated as follows: first, we fix $\mathbf{x}_1 = \mathbf{0}$ for convenience. Our generative model with shift-invariant kernels define an improper prior over the whole space when there is no observation (as it should since only difference between points is meaningful for shift invariant kernels). Then we draw each subsequent \mathbf{x}_n from the kernel student- \mathcal{T} predictive distribution $q_k(\mathbf{x}_n | \mathbf{x}_{1:n-1}; \sigma_o, \alpha, \beta)$ conditioned on previous draws. Sampling from the predictive distribution is possible via hybrid Monte Carlo [18].

Fig. 2 shows example fantasy datasets for different settings of parameters. We observe that the generative process has an approximate clustering property, in which it behaves similarly to the Chinese restaurant process. We also note that the ‘‘checkerboard-like’’ eigenfunctions of the squared exponential kernel induce a tree like refinement of the space, in which the method is similar to Dirichlet diffusion trees.

The performance of density models where the normalisation constants are not available – similar examples in the past included Markov random fields [19] and deep belief nets – is hard to evaluate directly. These density models are typically used in unsupervised tasks, such as reconstruction of missing data, novelty detec-

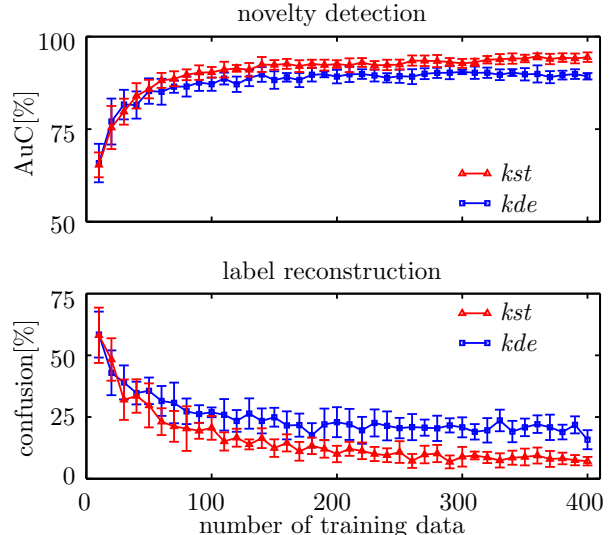


Figure 3: Performance of *kst* and *kde* as a function of the number of training examples on USPS data. Errorbars show \pm one standard deviation.

tion and image denoising, where the important quantity is the shape of the distribution, rather than the absolute density values. In the following, we consider three such unsupervised tasks that are indicative of the model’s performance: novelty detection, assessment of reconstruction performance and the recently introduced task of relative novelty detection [20]. The purpose of these experiments is to demonstrate, as proof of concept, that the method is able to model relevant statistical properties in high dimensional data.

4.1 Novelty detection

Novelty detection, recognition of patterns in test data that are unlikely under the distribution of the training data, is a common unsupervised task for which density estimation is used. Here, we considered the problem of detecting mislabelled images of handwritten digits given a training set of correctly labelled images from the USPS dataset. The dataset consists of 7291 training and 2007 test images, each of which is a labelled, 16×16 gray-scale image of a handwritten digit (0-9).

We modelled the joint distribution of the image (a 256 dimensional vector) and the label (a ten dimensional sparse vector, where the l^{th} element is 1 and the rest are 0’s if the label is l) by a kernel student- \mathcal{T} density (*kst*). We used a squared exponential kernel with length-scale chosen to be the median distance between data-points along digit dimensions and length-scale 1 along the label dimensions. We trained the algorithm on a randomly selected subset of available

training data and calculated predictive probabilities on a test dataset composed of 100 correctly labelled and 100 mislabelled test points. Digit-label pairs with predictive probability under a threshold were considered *mislabelled*. We compared the performance of our method to two baseline methods, kernel density estimation (*kde*, also called Parzen window estimate) and Dirichlet process mixtures of Gaussians (*dpm*), based on the area under the receiver operating characteristic curve (AuC) metric. Hyper-parameters of all three methods were chosen by grid search so that the average AuC was maximised on a separate validation set. Table 1 summarises the results of this comparison based on ten randomised iteration of the experiment with 2000 training and 200 test samples. We found that *kst* outperformed both competing methods significantly ($p < 0.01$, two sample T-test). The performance of *dpm* was substantially worse than that of the other two, which demonstrates that general mixture models are very inefficient in modelling high dimensional complex densities. To investigate the difference between the performance of *kde* and *kst*, we carried out a second sequence of experiments where the size of the training set ranged from 10 to 400. Fig. 3 shows average AuC values as a function of training set size. We observed that the two algorithms perform similarly for small amounts of training data, the difference in performance becomes significant ($p < 0.05$) for 60 training points or more.

4.2 Label reconstruction

In these experiments, we considered learning the classification of handwritten digits. In order to assess the density modelling capabilities of our method, we posed this problem as an image reconstruction problem¹. Again, we augmented the gray-scale image with 10 additional ‘pixels’ representing a sparse coding of the labels 0-9. We first trained our density model on the augmented images on randomly selected training subsets of various sizes. Then, on a separate subset of test images we computed the probability of each label 0-9 conditioned on the image by computing the joint probability of the image with that label and renormalising. For each image we assigned the label for which the conditional predictive probability was the highest (maximum *a posteriori* reconstruction). Results of comparison to *kde* and *DPM* with 2000 training and 400 test points are shown in Fig. 3. As measure of performance, we used average confusion, i.e. the fraction of misclassified test images. We again found that *kst* outperformed both other methods, and now

¹We could have considered more general image reconstruction tasks, but reconstructing labels equips us with intuitive measures of loss, such as confusion and AuC.

the advantage of our method compared to *kde* is more substantial. Fig. 3 shows confusion of *kst* and *kde* as a function of the number of training examples. The difference between the two methods becomes significant ($p < 0.01$) for 60 training examples or more.

4.3 Relative novelty detection

Finally, we considered the problem of relative novelty detection [20], for which we re-used the experimental setup in [20]. The data consists of two 200×200 color satellite images, the *background* and the *target* (Fig. 4.A–B). The task is to detect novel objects on the target image relative to the background image. Treating RGB values in each non-overlapping 2×2 patches as data points, we estimated the density over such patches in both the background ($\hat{q}(\mathbf{x})$) and in the target ($\hat{p}(\mathbf{x})$) images. A patch \mathbf{x} was identified as relatively novel if the ratio $\hat{p}(\mathbf{x})/\hat{q}(\mathbf{x})$ was above a certain threshold. The results produced by our algorithm are similar to that obtained by using state of the art novelty detection algorithms (c.f. Fig. 2 in [20]), even though here we only used a 3% random subsample of available image patches to estimate target and background densities.

5 Discussion

Summary In this paper, we considered applying the kernel trick as a tool for constructing nonparametric Bayesian methods, and suggested a general recipe for obtaining novel analytically tractable *Bayesian kernel machines*. The starting point is a suitably simple parametric Bayesian model in which inference is tractable. We have demonstrated how arguments based on orthonormal invariance can be used to guide our choice of models that are amenable to kernel trick. Having defined the basic model, one needs to express posterior predictive distributions in terms of kernel evaluations, then apply the kernel trick in those equations. By using kernels with infinite dimensional feature spaces, novel tractable nonparametric Bayesian methods may be obtained in this way.

To illustrate our general approach, we studied the problem of density estimation and presented kernel student- \mathcal{T} density estimation, a novel Bayesian kernel machine for density modelling. We followed our general methodology to arrive at a closed form expression that is exact up to a multiplicative constant. We also carried out numerical experiments to investigate the performance of the method. The results demonstrate that the method is capable of estimating the distribution of high dimensional real-world data, and can be used in various applications such as novelty detection and reconstruction. The most apparent limitation of



Figure 4: **A**, Background and **B**, target images for relative novelty detection. **C**, Top 5% novel patches in the target image are highlighted in *red*. The results are qualitatively similar to those presented in [20, Fig. 2].

the kernel student- \mathcal{T} density model is the intractability of its normalisation constant. Nevertheless, the unnormalised density can still be used in a variety of unsupervised learning tasks, such as image reconstruction or novelty detection. As other kernel methods, our algorithm has cubic complexity in the number of training points, though sparse approximations [21] can be used to construct faster algorithms for large-scale applications. Another advantage of the kernel approach is that the expressions can be formally extended to define non-trivial discrete distributions over e. g. graphs, permutations or text.

Conclusions and future work We believe that the present paper opens the door for developing a novel class of Bayesian kernel machines. The methodology presented in this paper can be applied in further problems to obtain nonparametric methods based on parametric Bayesian models. Investigating merits and limitations of this general approach in various statistical estimation and decision problems is certainly an interesting theme for future research. One particularly exciting direction that we plan to pursue is a Bayesian two-sample hypothesis test analogous to the frequentist test based on characteristic kernels [14]. For hypothesis testing, one needs to apply the kernel trick to a *ratio* of marginal likelihoods, rather than to a predictive density, which enables cancellations which simplify the issues with normalisation or Jacobians. Other potentially fruitful directions include semi-supervised regression and Bayesian numerical analysis [22].

Acknowledgements We would like to thank Zoubin Ghahramani, Carl Rasmussen, Arthur Gretton, Peter Orbanz and Le Song for helpful discussions. In particular, the proof in Appendix B is mainly due

to Le Song. Part of this work was supported under the EPSRC grant EP/F026641/1. Ferenc Huszár is supported by Trinity College Cambridge.

References

- [1] Blei, D.M. Variational methods for the Dirichlet process. in *ICML 21* , (2004).
- [2] Rasmussen, C.E. & Williams, C.K.I. *Gaussian Processes for Machine Learning* (The MIT Press, 2006).
- [3] van der Vaart, A.W. & van Zanten, J.H. Rates of contraction of posterior distributions based on Gaussian process priors. *Annals of Statistics* **36**, 1435–1463 (2008).
- [4] Hofman, T., Schölkopf, B. & Smola, A.J. Kernel methods in machine learning. *Annals of Statistics* **36**, 1171–1220 (2008).
- [5] Schölkopf, B. & Smola, A.J. *Learning with Kernels* (2002).
- [6] Rasmussen, C.E. & Ghahramani, Z. Infinite mixtures of Gaussian process experts. in *NIPS 14* , 881–888. MIT Press, (2002).
- [7] Chu, W., Sindhvani, V., Ghahramani, Z. & Keerthi, S.S. Relational learning with Gaussian processes. in *NIPS 19* (2007).
- [8] Adams, R., Murray, I. & Mackay, D. The Gaussian process density sampler. in *NIPS 21* 9–16 (2009).
- [9] Roweis, S.T. & Ghahramani, Z. A unifying review of linear Gaussian models. *Neural Computation* **11**, 305–345 (1999).
- [10] Hager, W.W. Updating the inverse of a matrix. *SIAM Rev.* **31**, 221–239 (1989).

- [11] Shawe-Taylor, J. & Cristianini, N. *Kernel Methods for Pattern Analysis* (Cambridge University Press, 2004).
- [12] Schölkopf, B., Smola, A. & Müller, K.R. Non-linear component analysis as a kernel eigenvalue problem. *Neural Comput.* **10**, 1299–1319 (1998).
- [13] Rosipal, R. & Girolami, M. An expectation-maximization approach to nonlinear component analysis. *Neural Computation* **13**, 505–510 (2001).
- [14] Sriperumbudur, B.K., Gretton, A., Fukumizu, K., Lanckriet, G.R.G. & Sholkopf, B. Injective Hilbert space embeddings of probability measures. in *COLT 21* , (2008).
- [15] Song, L., Zhang, X., Smola, A., Gretton, A. & Schölkopf, B. Tailoring density estimation via reproducing kernel moment matching. in *ICML 25* , 992–999, (2008).
- [16] Lawrence, N.D. Gaussian process latent variable models for visualisation of high dimensional data. in *NIPS 16* (2004).
- [17] Murray, I., Ghahramani, Z. & MacKay, D.J.C. MCMC for doubly-intractable distributions. in *UAI* 359–366 (2006).
- [18] Neal, R.M. MCMC using Hamiltonian dynamics. in *Handbook of Markov Chain Monte Carlo* (eds. S. Brooks, A. Gelman, G.J. & Meng, X.L.) (Chapman & Hall / CRC Press, 2010).
- [19] Derin, H. & Elliott, H. Modeling and segmentation of noisy and textured images using Gibbs random fields. *TPAMI* **9**, 39–55 (1987).
- [20] Smola, A., Song, L. & Teo, C. Relative novelty detection. in *AISTATS 12* Vol. 5 , Vol. 5, 536–543, (2009).
- [21] Fine, S., Scheinberg, K., Cristianini, N., Shawe-taylor, J. & Williamson, B. Efficient SVM training using low-rank kernel representations. *JMLR* **2**, 243–264 (2001).
- [22] O’Hagan, A. Some bayesian numerical algebra. in *Bayesian Statistics* Vol. 8 (Oxford University Press, 1992).

A Change-of-variable formula

Theorem Let \mathcal{X} and \mathcal{F} be d and D dimensional Euclidean spaces respectively ($D > d$), $\varphi : \mathcal{X} \mapsto \mathcal{F}$ be a smooth injective mapping and let $q(\phi)$ define a density function with respect to the Lebesgue measure on \mathcal{F} . Then the restriction of distribution q to the observation manifold $\varphi(\mathcal{X})$ (i. e. conditioning on the event of being in the observation manifold) induces the following density in the input space \mathcal{X} :

$$p(\mathbf{x}) \propto q(\varphi(\mathbf{x})) \cdot \det \left(\left\langle \frac{\partial \varphi(\mathbf{x})}{\partial x^i}, \frac{\partial \varphi(\mathbf{x})}{\partial x^j} \right\rangle \right)_{i,j=1,\dots,d}^{\frac{1}{2}}. \quad (7)$$

We suspect this result may have appeared in several forms in the literature, but we didn't succeed to find a citation, and so we provide a proof here.

Proof Note that we cannot apply the standard change-of-variable formula directly as the two spaces have different dimensions. So we augment the space \mathcal{X} with an Euclidean space Ξ of dimension $D - d$. We then define the following mapping $F : \mathcal{X} \times \Xi \mapsto \mathcal{F}$:

$$F(\mathbf{x}, \boldsymbol{\xi}) = \varphi(\mathbf{x}) + V_{\mathbf{x}} \boldsymbol{\xi},$$

where $V_{\mathbf{x}}$ is a $D \times (D - d)$ matrix of mutually orthonormal vectors to the tangent space of the manifold at \mathbf{x} (i. e. a basis for the orthogonal space to the tangent space at \mathbf{x}). For small enough $\boldsymbol{\xi}$'s (which could depend on \mathbf{x}), F will map small neighborhoods of $(\mathbf{x}, \mathbf{0})$ to neighborhoods of $\varphi(\mathbf{x})$ in an injective manner. We thus apply the standard change of variable formula to map the density $q(\phi)$ to a density on an open set around $\mathcal{X} \times \{\mathbf{0}\}$:

$$p(\mathbf{x}, \boldsymbol{\xi}) = q(F(\mathbf{x}, \boldsymbol{\xi})) |\det J_F(\mathbf{x}, \boldsymbol{\xi})|,$$

where $J_F(\mathbf{x}, \boldsymbol{\xi})$ is the Jacobian matrix of the F mapping. To condition on being in the observation manifold, we simply need to condition on $\boldsymbol{\xi} = \mathbf{0}$, and so the density we want to compute is:

$$p(\mathbf{x}) \propto p(\mathbf{x}, \mathbf{0}) = q(F(\mathbf{x}, \mathbf{0})) |\det J_F(\mathbf{x}, \mathbf{0})|.$$

Now, $J_F(\mathbf{x}, \mathbf{0}) = (J_{\varphi}(\mathbf{x}), V_{\mathbf{x}})$ where $J_{\varphi}(\mathbf{x}) = (\frac{\partial \varphi(\mathbf{x})}{\partial x^1}, \dots, \frac{\partial \varphi(\mathbf{x})}{\partial x^d})$ is the Jacobian matrix of φ . Finally, we use the fact that $|\det(J_F)| = (\det(J_F^{\top} J_F))^{1/2}$ for any squared matrix J_F to get that:

$$|\det(J_F(\mathbf{x}, \mathbf{0}))| = \det \left(\begin{array}{cc} J_{\varphi}(\mathbf{x})^{\top} J_{\varphi}(\mathbf{x}) & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{array} \right)^{\frac{1}{2}} = \det(J_{\varphi}(\mathbf{x})^{\top} J_{\varphi}(\mathbf{x}))^{\frac{1}{2}},$$

where we used that $J_{\varphi}(\mathbf{x})^{\top} V_{\mathbf{x}} = \mathbf{0}$ and $V_{\mathbf{x}}^{\top} V_{\mathbf{x}} = \mathbf{I}$ by orthonormality of the $V_{\mathbf{x}}$ with the tangent space spanned by the columns of $J_{\varphi}(\mathbf{x})$. Finally, we use the fact that:

$$J_{\varphi}(\mathbf{x})^{\top} J_{\varphi}(\mathbf{x}) = \left(\left\langle \frac{\partial \varphi(\mathbf{x})}{\partial x^i}, \frac{\partial \varphi(\mathbf{x})}{\partial x^j} \right\rangle \right)_{i,j=1,\dots,d}$$

to obtain (7). \square

B Jacobian term for translation invariant kernel

We provide the sketch of a proof that $\left\langle \frac{\partial \varphi(\mathbf{x})}{\partial x^i}, \frac{\partial \varphi(\mathbf{x})}{\partial x^j} \right\rangle$ doesn't depend on \mathbf{x} for a translation invariant kernel under suitable regularity conditions.

By Bochner's theorem (e.g. theorem 4.1 in [2]), a translation invariant kernel can be expressed as the inverse Fourier transform of a positive finite measure. If moreover we assume that the Fourier transform of the kernel exists, this measure will have a density $p(\mathbf{s})$ with respect to the Lebesgue measure, and so we can write:

$$\begin{aligned} k(\mathbf{x} - \mathbf{y}) &= \int_{\mathbb{R}^d} p(\mathbf{s}) e^{2\pi i \mathbf{s}^\top (\mathbf{x} - \mathbf{y})} d\mathbf{s} \\ &= \int p(\mathbf{s}) (\cos(2\pi \mathbf{s}^\top (\mathbf{x} - \mathbf{y})) + i \sin(2\pi \mathbf{s}^\top (\mathbf{x} - \mathbf{y}))) d\mathbf{s} \\ &= \int p(\mathbf{s}) (\cos(2\pi \mathbf{s}^\top (\mathbf{x} - \mathbf{y}))) d\mathbf{s} \end{aligned}$$

as $\sin(\mathbf{s}^\top \boldsymbol{\delta})$ is an odd function of \mathbf{s} for any $\boldsymbol{\delta}$ and $p(\mathbf{s})$ is a symmetric function given that it is the inverse Fourier transform of the symmetric function k , so the Cauchy principal value of the second term vanishes. Now using a trigonometric identity, we get:

$$\begin{aligned} k(\mathbf{x} - \mathbf{y}) &= \int p(\mathbf{s}) ((\cos(2\pi \mathbf{s}^\top \mathbf{x}) \cos(2\pi \mathbf{s}^\top \mathbf{y})) + (\sin(2\pi \mathbf{s}^\top \mathbf{x}) \sin(2\pi \mathbf{s}^\top \mathbf{y}))) d\mathbf{s} \\ &= \int \left\langle \sqrt{p(\mathbf{s})} \begin{pmatrix} \cos(2\pi \mathbf{s}^\top \mathbf{x}) \\ \sin(2\pi \mathbf{s}^\top \mathbf{x}) \end{pmatrix}, \sqrt{p(\mathbf{s})} \begin{pmatrix} \sin(2\pi \mathbf{s}^\top \mathbf{y}) \\ \sin(2\pi \mathbf{s}^\top \mathbf{y}) \end{pmatrix} \right\rangle d\mathbf{s} \end{aligned}$$

So we can obtain an explicit feature mapping by defining:

$$\varphi(\mathbf{x}) = \left(\sqrt{p(\mathbf{s})} \begin{pmatrix} \cos(2\pi \mathbf{s}^\top \mathbf{x}) \\ \sin(2\pi \mathbf{s}^\top \mathbf{x}) \end{pmatrix} \right)_{\mathbf{s} \in \mathbb{R}^d}.$$

Assuming that $p(\mathbf{s})$ goes to zero sufficiently quickly, we have that:

$$\frac{\partial \varphi(\mathbf{x})}{\partial x_k} = \left(2\pi s_k \sqrt{p(\mathbf{s})} \begin{pmatrix} -\sin(2\pi \mathbf{s}^\top \mathbf{x}) \\ \cos(2\pi \mathbf{s}^\top \mathbf{x}) \end{pmatrix} \right)_{\mathbf{s} \in \mathbb{R}^d}$$

and using a similar derivation as we used above, we get that:

$$\begin{aligned} \left\langle \frac{\partial \varphi(\mathbf{x})}{\partial x_k}, \frac{\partial \varphi(\mathbf{x})}{\partial x_l} \right\rangle &= 4\pi^2 \int s_k s_l p(\mathbf{s}) e^{2\pi i \mathbf{s}^\top (\mathbf{x} - \mathbf{x})} d\mathbf{s} \\ &= 4\pi^2 \int s_k s_l p(\mathbf{s}) e^0 d\mathbf{s} \end{aligned}$$

which doesn't depend on \mathbf{x} as we wanted to show. \square