Bayesian Approaches to Distribution Regression

Ho Chung Leon Law[*1] ho.law@spc.ox.ac.uk
Dougal J. Sutherland[*2] dougal@gmail.com
Dino Sejdinovic[1] dino.sejdinovic@stats.ox.ac.uk
Seth Flaxman[3] s.flaxman@imperial.ac.uk

Abstract

Distribution regression has recently attracted much interest as a generic solution to the problem of supervised learning where labels are available at the group level, rather than at the individual level. Current approaches, however, do not propagate the uncertainty in observations due to sampling variability in the groups. This effectively assumes that small and large groups are estimated equally well, and should have equal weight in the final regression. We account for this uncertainty with a Bayesian distribution regression formalism, improving the robustness and performance of the model when group sizes vary. We frame our models in a neural network style, allowing for simple MAP inference using backpropagation to learn the parameters, as well as MCMC-based inference which can fully propagate uncertainty. We demonstrate our approach on illustrative toy datasets, as well as on a challenging problem of predicting age from images.

1 INTRODUCTION

Distribution regression is the problem of learning a regression function from samples of a distribution to a single set-level label. For example, we might attempt to infer the sentiment of texts based on word-level features, to predict the label of an image based on small patches, or even perform traditional parametric statistical inference by learning a function from sets of samples to the parameter values.

Recent years have seen wide-ranging applications of this framework, including inferring summary statistics in Approximate Bayesian Computation (Mitrovic et al., 2016), estimating Expectation Propagation messages (Jitkrittum et al., 2015), predicting the voting behaviour of demographic groups (Flaxman et al., 2015, 2016), and learning the total mass of dark matter halos from observable galaxy velocities (Ntampaka et al., 2015, 2016). Closely related distribution classification problems also include identifying the direction of causal relationships from data (Lopez-Paz et al., 2015) and classifying text based on bags of word vectors (Yoshikawa et al., 2014; Kusner et al., 2015).

One particularly appealing approach to the distribution regression problem is to represent the input set of samples by their kernel mean embedding (described in Section 2.1), where distributions are represented as single points in a reproducing kernel Hilbert space. Standard kernel methods can then be applied for distribution regression, classification, anomaly detection, and so on. This approach was perhaps first popularized by Muandet et al. (2012); Szábo et al. (2016) provided a recent learning-theoretic analysis.

In this framework, however, each distribution is simply represented by the empirical mean embedding, ignoring the fact that large sample sets are much more precisely understood than small ones. Most studies also use point estimates for their regressions, such as kernel ridge regression or support vector machines, thus ignoring uncertainty both in the distribution embeddings and in the regression model.

1.1 Our Contributions

We propose a set of Bayesian approaches to distribution regression. First, we build on a recently proposed Bayesian nonparametric model of uncertainty in kernel mean embeddings (Flaxman et al., 2016), and then use a sparse representation of the desired function in the RKHS for prediction in the regression model. This model allows for a full account of uncertainty in the mean embedding, but requires a point estimate of the regression function for conjugacy; we thus use backpropagation to obtain a MAP estimate for it as well as various hyperparameters. Alternatively, we can use point estimates of the input embeddings but account for uncertainty in the regression model with simple Bayesian linear regression. We then combine the treatment of the two sources of uncertainty into a fully Bayesian model which uses Hamiltonian Monte Carlo for efficient inference. Depending on the inferential goals, each model can be useful. We demonstrate our approaches on an illustrative toy problem as well as a challenging real-world age estimation task.

[*]These authors contributed equally. [1]Department of Statistics, University of Oxford, UK [2]Gatsby Unit, University College London, UK [3]Department of Mathematics and Data Science Institute, Imperial College London, UK.

2 BACKGROUND

2.1 Problem Overview

Distribution regression is the task of learning a classifier or a regression function that maps probability distributions to labels. The challenge of distribution regression goes beyond the standard supervised learning setting: we do not have access to exact input-output pairs since the true inputs, probability distributions, are observed only through samples from that distribution:

$$\left(\{x_j^1\}_{j=1}^{N_1}, y_1\right), \ldots, \left(\{x_j^n\}_{j=1}^{N_n}, y_n\right), \tag{1}$$

so that each bag $\{x_j^i\}_{j=1}^{N_i}$ has a label y_i along with N_i individual observations $x_j^i \in \mathcal{X}$. We assume that the observations $\{x_j^i\}_{j=1}^{N_i}$ are i.i.d. samples from some unobserved distribution P_i, and that the true label y_i depends only on P_i. We wish to avoid making any strong parametric assumptions on the P_i. For the present work, we will assume the labels y_i are real-valued; Appendix B shows an extension to binary classification. We typically take the observation space $\mathcal{X}$ to be a subset of $\mathbb{R}^p$ but it could easily be structured (e.g. text or images), since we access it only through a kernel (Gärtner, 2008).

We consider the standard approach to distribution regression, which relies on kernel mean embeddings and kernel ridge regression. For any positive definite kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, there exists a unique reproducing kernel Hilbert space (RKHS) $\mathcal{H}_k$, a possibly infinite-dimensional space of functions $f : \mathcal{X} \to \mathbb{R}$ where evaluation can be written as an inner product, and in particular $f(x) = \langle f, k(\cdot, x)\rangle_{\mathcal{H}_k}$ for all $f \in \mathcal{H}_k, x \in \mathcal{X}$.

Given a probability measure P on $\mathcal{X}$, let us define the kernel mean embedding into $\mathcal{H}_k$ as

$$\mu_{\mathsf{P}} = \int k(\cdot, x)\, \mathsf{P}(dx) \in \mathcal{H}_k. \tag{2}$$

Notice that μ_{P} serves as a high- or infinite-dimensional vector representation of P. For the kernel mean embedding of P into $\mathcal{H}_k$ to be well-defined, it suffices that $\int \sqrt{k(x,x)}\mathsf{P}(dx) < \infty$, which is trivially satisfied for all P if k is bounded. Analogously to the reproducing property of RKHS, μ_{P} represents the expectation function on $\mathcal{H}_k$: $\int h(x)\mathsf{P}(dx) = \langle h, \mu_{\mathsf{P}}\rangle_{\mathcal{H}_k}$. For so-called *characteristic* kernels (Sriperumbudur et al., 2010), every probability measure has a unique embedding, and thus μ_{P} completely determines the corresponding probability measure.

2.2 Estimating Mean Embeddings

For a set of samples $\{x_j\}_{j=1}^n$ drawn iid from P, the empirical estimator of μ_{P} is given by

$$\widehat{\mu_{\mathsf{P}}} = \mu_{\hat{\mathsf{P}}} = \int k(\cdot, x)\, \hat{\mathsf{P}}(dx) = \frac{1}{n}\sum_{j=1}^{n} k(\cdot, x_j). \tag{3}$$

This is the standard estimator used by previous distribution regression approaches, which the reproducing property of $\mathcal{H}_k$ shows us corresponds to the kernel

$$\langle \widehat{\mu_{\mathsf{P}_i}}, \widehat{\mu_{\mathsf{P}_j}}\rangle_{\mathcal{H}_k} = \frac{1}{N_i N_j}\sum_{\ell=1}^{N_i}\sum_{r=1}^{N_j} k(x_\ell^i, x_r^j). \tag{4}$$

But (3) is an empirical mean estimator in a high- or infinite-dimensional space, and is thus subject to the well-known *Stein phenomenon*, so that its performance is dominated by the James-Stein shrinkage estimators. Indeed, Muandet et al. (2014) studied shrinkage estimators for mean embeddings, which can result in substantially improved performance for some tasks (Ramdas and Wehbe, 2015). Flaxman et al. (2016) proposed a Bayesian analogue of shrinkage estimators, which we now review.

This approach consists of (1) a Gaussian Process prior $\mu_{\mathsf{P}} \sim \mathcal{GP}(m_0, r(\cdot,\cdot))$ on $\mathcal{H}_k$, where r is selected to ensure that $\mu_{\mathsf{P}} \in \mathcal{H}_k$ almost surely and (2) a normal likelihood $\widehat{\mu_{\mathsf{P}}}(\mathbf{x}) \mid \mu_{\mathsf{P}}(\mathbf{x}) \sim \mathcal{N}(\mu_{\mathsf{P}}(\mathbf{x}), \Sigma)$. Here, conjugacy of the prior and the likelihood leads to a Gaussian process posterior on the true embedding μ_{P}, given that we have observed $\widehat{\mu_{\mathsf{P}}}$ at some set of locations $\mathbf{x}$. The posterior mean is then essentially identical to a particular shrinkage estimator of Muandet et al. (2014), but the method described here has the extra advantage of a closed form uncertainty estimate, which we utilise in our distributional approach. For the choice of r, we use a Gaussian RBF kernel k, and choose either $r = k$ or, following Flaxman et al. (2016), $r(x, x') = \int k(x, z)\, k(z, x')\, \nu(dz)$ where ν is proportional to a Gaussian measure. For details of our choices, and why they are sufficient for our purposes, see Appendix A.

This model accounts for the uncertainty based on the number of samples N_i, shrinking the embeddings for small sample sizes more. As we will see, this is essential in the context of distribution regression, particularly when bag sizes are imbalanced.

2.3 Standard Approaches to Distribution Regression

Following Szábo et al. (2016), assume that the probability distributions P_i are each drawn randomly from some unknown meta-distribution over probability distributions, and take a two-stage approach, illustrated as in Figure 1. Denoting the feature map $k(\cdot, x) \in \mathcal{H}_k$ by $\phi(x)$, one uses the empirical kernel mean estimator (3) to separately estimate the mean of each group:

$$\widehat{\mu_1} = \frac{1}{N_1}\sum_{j=1}^{N_1}\phi(x_j^1), \quad \ldots, \quad \widehat{\mu_n} = \frac{1}{N_n}\sum_{i=1}^{N_n}\phi(x_j^n). \tag{5}$$

Next, one uses kernel ridge regression (Saunders et al., 1998) to learn a function $f : \mathcal{H}_k \to \mathbb{R}$, by minimizing

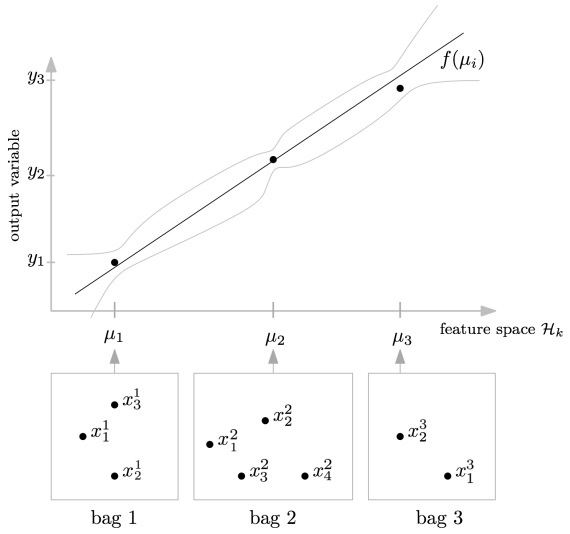

Figure 1: Each bag i is summarised by a kernel mean embedding $\mu_i \in \mathcal{H}_k$, and a regression function $f : \mathcal{H}_k \to \mathbb{R}$ is learnt to predict labels $y_i \in \mathbb{R}$. We propose a Bayesian approach so that we can propagate uncertainty due to the number of samples in each bag and ultimately obtain posterior credible intervals, illustrated in gray.

the squared loss with an RKHS complexity penalty:

$$\hat{f} = \operatorname*{argmin}_{f \in \mathcal{H}_K} \sum_i (y_i - f(\widehat{\mu_i}))^2 + \lambda \|f\|^2_{\mathcal{H}_K}.$$

Here $K : \mathcal{H}_k \times \mathcal{H}_k \to \mathbb{R}$ is a "second-level" kernel on mean embeddings. Szábo et al. (2016) consider a variety of kernels K corresponding to the Hilbert space $\mathcal{H}_k$. If K is a linear kernel on the RKHS $\mathcal{H}_k$, then the resulting method can be interpreted as a linear (ridge) regression on mean embeddings, which are themselves nonlinear transformations of the inputs. In some cases, adding a second-level nonlinear kernel on the RKHS can improve performance (Muandet et al., 2012).

Naively implementing distribution regression using the kernel trick is not scalable for even modestly sized datasets, as the relevant kernel matrix over distributions has $\mathcal{O}(n^2)$ entries, but the computation of entry (i, j) requires time $\mathcal{O}(N_i N_j)$. Thus many applications have relied on versions of random Fourier features (Rahimi and Recht, 2007). In this paper we take a simpler approach and expand in terms of landmark points drawn randomly from the observations, yielding radial basis networks (Broomhead and Lowe, 1988) with a mean pooling operation to construct the mean embedding.

3 RELATED WORK

As previously mentioned, Szábo et al. (2016) provides a thorough learning-theoretic analysis of the regression model discussed in Section 2.3. This formalism considering a kernel method on distributions using their embedding representations, or various scalable approximations to it, has been widely applied (e.g. Muandet et al., 2012; Yoshikawa et al., 2014; Flaxman et al., 2015; Jitkrittum et al., 2015; Lopez-Paz et al., 2015; Mitrovic et al., 2016). There are also several other notions of similarities on distributions in use (not necessarily falling within the framework of kernel methods and RKHSs), as well as local smoothing approaches, mostly based on estimates of various probability metrics (Moreno et al., 2003; Jebara et al., 2004; Póczos et al., 2011; Oliva et al., 2013; Poczos et al., 2013; Kusner et al., 2015). For a partial overview, see the recent thesis of Sutherland (2016).

Other related problems of learning on instances with group-level labels include learning with label proportions (Quadrianto et al., 2009; Patrini et al., 2014), ecological inference (King, 1997; Gelman et al., 2001), pointillistic pattern search (Ma et al., 2015), multiple instance learning (Dietterich et al., 1997; Kück and de Freitas, 2005; Zhou et al., 2009; Krummenacher et al., 2013) and learning with sets (Zaheer et al., 2017).[1]

There have also been some Bayesian approaches in related contexts, though most do not follow our setting where the label is a function of the underlying distribution rather than the set of observed instances. Kück and de Freitas (2005) consider an MCMC method with group-level labels but focus on individual-level classifiers, while Jackson et al. (2006) use hierarchical Bayesian models on a combination of individual-level and aggregate data for ecological inference. Flaxman et al. (2015) and Jitkrittum et al. (2015) quantify the uncertainty of the distribution regression model by interpreting the kernel ridge regression on embeddings as Gaussian Process regression. However, Jitkrittum et al. (2015) consider embeddings of a parametric family of distributions, so there is no uncertainty in embedding representations, while Flaxman et al. (2015) treat empirical embeddings as fixed inputs to the learning problem.

4 OUR MODELS

We propose three different Bayesian models, with each model encoding different types of uncertainty. We begin with a non-Bayesian RBF network formulation of the standard approach to distribution regression as a baseline, before refining this approach to better propagate uncertainty in bag size, as well as model parameters.

4.1 Base Model

Our RBF network formulation is based on a variation on the approach of Broomhead and Lowe (1988), Law et al.

[1]For more, also see `giorgiopatrini.org/nips15workshop`.

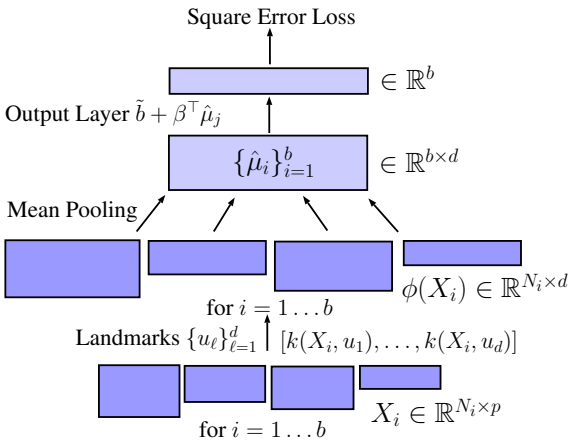

Figure 2: Our baseline model, a RBF network for distribution regression. X_i represents the matrix of samples for bag i, while $k(X_i, u_\ell)$ represents the element wise operation on each row of X_i, with b representing the batch size for stochastic gradient descent.

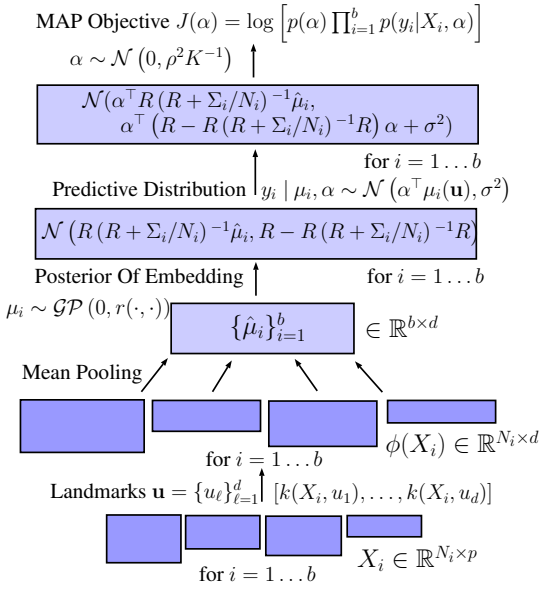

Figure 3: Our mean shrinkage pooling model, for this diagram, we have take $m_0 = \mathbf{0}$, $\eta = 1$ and $\mathbf{u} = \mathbf{z}$, so that $R = R_{\mathbf{z}} = R_{\mathbf{zz}}$, and $K_{\mathbf{z}} = K$.

(2017), and Zaheer et al. (2017). As shown in Figure 2, the initial input is a minibatch consisting of several bags X_i, each containing N_i points. Each point is then converted to an explicit featurisation, taking the role of ϕ in (5), by a radial basis layer: each point $x_j^i \in \mathbb{R}^p$ is mapped to

$$\phi(x_j^i) = [k(x_j^i, u_1), \dots, k(x_j^i, u_d)]^\top \in \mathbb{R}^d$$

where $\mathbf{u} = \{u_\ell\}_{\ell=1}^d$ are landmark points. A mean pooling layer yields the estimated mean embedding $\hat{\mu}_i$ corresponding to each of the bags j represented in the minibatch, where $\hat{\mu}_i = \frac{1}{N_i}\sum_{j=1}^{N_i}\phi(x_j^i)$.[2] Finally, a fully connected output layer gives real-valued labels $\hat{y}_i = \beta^\mathsf{T}\hat{\mu}_i + b$. As a loss function we use the mean square error $\frac{1}{n}\sum_i(\hat{y}_i - y_i)^2$. For learning, we use backpropagation with the Adam optimizer (Kingma and Ba, 2015). To regularise the network, we use early stopping on a validation set, as well as an L_2 penalty corresponding to a normal prior on β.

4.2 Mean Shrinkage Pooling Model

A shortcoming of the base model, and of the standard approach in Szábo et al. (2016), is that it ignores uncertainty in the first level of estimation due to varying number of samples in each bag. Ideally we would estimate not just the mean embedding per bag, but also a measure of the sample variance, in order to propagate this information regarding uncertainty from the bag size through the model. Bayesian tools provide a natural framework for this problem.

We can use the Bayesian nonparametric prior over kernel mean embeddings (Flaxman et al., 2016) described in Section 2.2, and observe the empirical embeddings at the landmark points $\mathbf{u_i}$. For $\mathbf{u_i}$, we take a fixed set of landmarks, which we can choose via k-means clustering (in a similar spirit to Zhang and Kwok, 2010) or sample without replacement. Using the conjugacy of the model, and the Gaussian process prior $\mu_i \sim \mathcal{GP}(m_0, \eta r(.,.))$, we obtain a closed form posterior Gaussian process whose evaluation at points $\mathbf{h} = \{h_s\}_{s=1}^{n_h}$ is:

$$\mu_i(\mathbf{h}) \mid \mathbf{x_i} \sim \mathcal{N}\Big(R_{\mathbf{h}}\left(R+\Sigma_i/N_i\right)^{-1}(\hat{\mu}_i - m_0) + m_0,$$
$$R_{\mathbf{hh}} - R_{\mathbf{h}}\left(R+\Sigma_i/N_i\right)^{-1}R_{\mathbf{h}}^\top\Big)$$

where $R_{st} = \eta r(u_s, u_t), (R_{\mathbf{hh}})_{st} = \eta r(h_s, h_t), (R_{\mathbf{h}})_{st} = \eta r(h_s, u_t)$, and $\mathbf{x_i}$ denotes the set $\{x_j^i\}_{j=1}^{N_i}$. We take the prior mean m_0 to be the average of the $\hat{\mu}_i$; under a linear kernel K, this means we shrink predictions towards the mean prediction. Note η essentially controls the strength of the shrinkage: a smaller η means we shrink more strongly towards m_0. We take Σ_i to be the average of the empirical covariance of $\{\varphi(x_j^i)\}_{j=1}^{N_i}$ across all bags, to avoid poor estimation of Σ_i for smaller bags. Some more intuition about the behaviour of this estimator can be found in Appendix C.

Now, supposing we have normal observation error σ^2, and use a linear kernel as our second level kernel K, we have:

$$y_i \mid \mu_i, f \sim \mathcal{N}\left(\langle f, \mu_i\rangle_{\mathcal{H}_k}, \sigma^2\right) \tag{6}$$

where $f \in \mathcal{H}_k$. Clearly, this is difficult to work with; hence we parameterise f as $f = \sum_{\ell=1}^s \alpha_\ell k(\cdot, z_\ell)$, where $\mathbf{z} = \{z_\ell\}_{\ell=1}^s$ is a set of landmark points for f, which we

[2] For implementation, we stack all of the bags X_i into a single matrix of size $\sum_j N_j \times d$ for the first layer, then implement pooling via sparse matrix multiplication.

can learn or fix. (Appendix D gives a motivation for this approximation using the representer theorem.) Using the reproducing property, our likelihood model becomes:

$$y_i \mid \mu_i, \alpha \sim \mathcal{N}\left(\alpha^\mathsf{T}\mu_i(\mathbf{z}), \sigma^2\right) \tag{7}$$

where $\mu_i(\mathbf{z}) = [\mu_i(z_1), \ldots, \mu_i(z_s)]^\top$. For fixed α and $\mathbf{z}$ we can analytically integrate out the dependence on μ_i, and the predictive distribution of a bag label becomes

$$\begin{aligned} y_i \mid \mathbf{x}_i, \alpha &\sim \mathcal{N}(\xi_i^\alpha, \nu_i^\alpha) \\ \xi_i^\alpha &= \alpha^\top R_\mathbf{z}\left(R + \frac{\Sigma_i}{N_i}\right)^{-1}(\hat{\mu}_i - m_0) + \alpha^\mathsf{T} m_0 \\ \nu_i^\alpha &= \alpha^\mathsf{T}\left(R_{\mathbf{zz}} - R_\mathbf{z}\left(R + \frac{\Sigma_i}{N_i}\right)^{-1} R_\mathbf{z}^\mathsf{T}\right)\alpha + \sigma^2. \end{aligned}$$

Adding the prior $\alpha \sim \mathcal{N}(0, \rho^2 K_\mathbf{z}^{-1})$, where $K_\mathbf{z}$ is the kernel matrix for k on $\mathbf{z}$, gives us the standard regularisation on f of $\|f\|^2_{\mathcal{H}_k}$. Our MAP objective is:

$$\frac{1}{2}\sum_{i=1}^{n}\left\{\log \nu_i^\alpha + \frac{(y_i - \xi_i^\alpha)^2}{\xi_i^\alpha}\right\} + \frac{\alpha^\mathsf{T} K_\mathbf{z}\alpha}{2\rho^2}.$$

We can use backpropagation to learn the parameters α, σ, and if we wish η, $\mathbf{z}$, and any kernel parameters. The full model is illustrated in Figure 3. This Bayesian approach allows us to directly encode uncertainty based on bag size in the objective function, and provides predictions with uncertainty intervals.

4.3 Bayesian Linear Regression Model

An alternative approach to encode uncertainty in the model is to encode uncertainty over regression parameters β only, with the following model:

$$\beta \sim \mathcal{N}(0, \rho^2) \qquad y_i \mid \mathbf{x}_i, \beta \sim \mathcal{N}(\beta^\mathsf{T}\hat{\mu}_i, \sigma^2)$$

which is essentially Bayesian linear regression on the empirical mean embeddings. Here, we are working directly with the finite-dimensional $\hat{\mu}_i$, unlike the infinite-dimensional μ_i before. Due to the conjugacy of the model, we can easily obtain the predictive distribution $y_i \mid \mathbf{x_i}$, integrating out the uncertainty over β. This again provides us uncertainty intervals for the predictions y_i. For model tuning, we can maximise the model evidence, i.e. the marginal log-likelihood (see Bishop (2006) for details), and use backpropagation through the network to learn σ and ρ and any kernel parameters of interest.[3]

4.4 Bayesian Distribution Regression

From a modelling perspective, it is natural to combine the two Bayesian approaches above, fully propagating uncertainty in estimation of the mean embedding and of the regression coefficients α. Unfortunately, conjugate Bayesian inference is no longer available. Thus, we consider a Markov Chain Monte Carlo (MCMC) sampling based approach, using Hamiltonian Monte Carlo (HMC) for efficient inference. Whereas inference above used gradient descent to maximise the marginal likelihood, with the gradient calculated using automatic differentiation, here we use automatic differentiation to calculate the gradient of the joint log-likelihood and follow this gradient as we perform sampling over the parameters we wish to infer.

We can still exploit the conjugacy of the mean shrinkage layer, obtaining closed form expressions for the posterior over the mean embeddings. Conditional on the mean embeddings, we have a Bayesian linear regression model with parameters α which we sample with HMC, specifically NUTS (Hoffman and Gelman, 2014; Stan Development Team, 2014). An implementation of our Bayesian Distribution Regression (BDR) model in Stan is provided in Appendix E.

5 EXPERIMENTS

We will now demonstrate our various Bayesian approaches: the mean-shrinkage pooling method with $r = k$ (*shrinkage*) and with $r(x, x') = \int k(x, z)k(z, x')\nu(\mathrm{d}z)$ for ν proportional to a Gaussian measure (*shrinkageC*), Bayesian linear regression (*BLR*), and the full Bayesian distribution regression model with $r = k$ (*BDR*).

We first demonstrate the characteristics of our models on a synthetic dataset, and then evaluate them on a real life age prediction problem. Throughout, for simplicity, we take $\mathbf{u} = \mathbf{z}$, i.e. $R = R_\mathbf{z} = R_{\mathbf{zz}}$, and $K_\mathbf{z} = K$ — although $\mathbf{u}$ and $\mathbf{z}$ could be different, with $\mathbf{z}$ learnt. Here k is taken to be the standard RBF kernel. For all our experiments, in the shrinkage, shrinkageC, radial and BLR network, we tune the learning rate, number of landmarks, bandwidth of the kernel and regularisation parameters on a validation set, and learn any other parameters. Similarly, for BDR, we place weakly informative normal priors (truncated at zero for non-negative parameters) as detailed in Appendix E, and tune via a validation set the number of landmarks, bandwidth of the kernel, and prior parameters.

5.1 Gamma Synthetic Data

We create a synthetic dataset by repeatedly sampling from the following hierarchical model:

$$\begin{aligned} y_i &\sim \text{Uniform}(4, 8) \\ \left[x_j^i\right]_\ell \mid y_i &\overset{\text{iid}}{\sim} \frac{1}{y_i}\left[\Gamma\left(\frac{y_i}{2}, \frac{1}{2}\right)\right] + \varepsilon \text{ for } j \in [N_i], \ell \in [5]. \end{aligned}$$

Here y_i is the label for the ith bag, and each $x_j^i \in \mathbb{R}^5$ has entries i.i.d. according to the given gamma distribution,

[3] Note that unlike the models above, here we cannot do mini-batch stochastic gradient descent, as the marginal log-likelihood does not decompose for each individual data point.

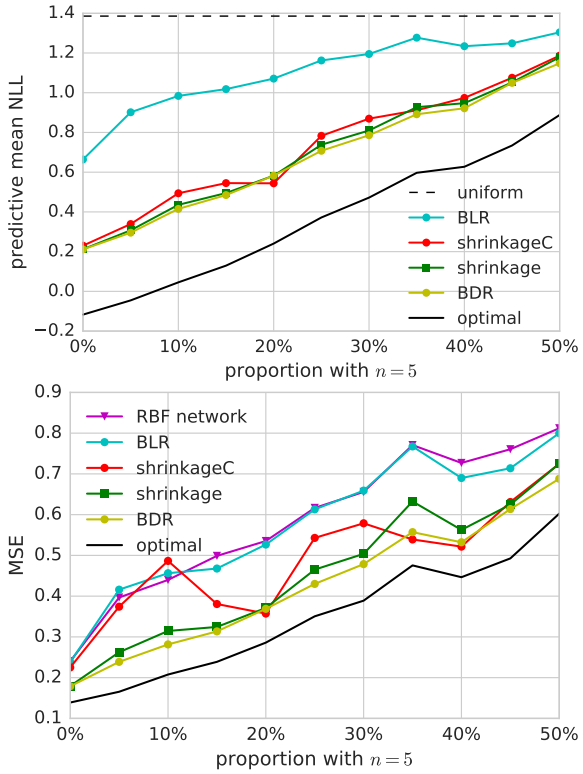

Figure 4: Top: negative log-likelihood. Bottom: mean-squared error. For context, performance of the Bayes-optimal predictor is also shown, and for NLL 'uniform' shows the performance of a uniform prediction on the possible labels. For MSE, the constant overall mean label predictor achieves about 1.3.

scaled down by y_i so as to have unit mean. ε is an added noise term which differs for the two experiments below.

In these experiments, we generate $1\,000$ bags for training, 500 bags for a validation set for parameter tuning, 500 bags to use for early-stopping of the models, and $1\,000$ bags for testing. Here, the landmark points $\mathbf{u}$ are chosen via k-means (fixed across all models). For context, we also show results of the Bayes-optimal model, which gives true posteriors according to the correct data-generating process; this is the best performance any model could hope to achieve.

Varying bag size: Uncertainty in the inputs. In order to study the behaviour of our models with varying bag size, we fix four sizes $N_i \in \{5, 20, 100, 1\,000\}$. For each generated dataset, 25% of the bags have $N_i = 20$, and 25% have $N_i = 100$. Among the other half of the data, we vary the ratio of $N_i = 5$ and $N_i = 1\,000$ bags to demonstrate our methods' efficacy at dealing with varied bag sizes: we let s_5 be the overall percentage of bags with $N_i = 5$, ranging from $s_5 = 0$ (in which case no bags have size $N_i = 5$) to $s_5 = 50$ (in which case 50% of the overall bags have size $N_i = 5$). Here we do not add additional noise: $\varepsilon = 0$.

Results are shown in Figure 4. For context, we also show the results of the Bayes-optimal model based on the true generating distribution. Note that our learning models, which treat the inputs as five-dimensional, fully nonparametric distributions, are at a substantial disadvantage even in the way they view the data compared to this true model.

BDR and shrinkage methods, which take into account bag size uncertainty, perform well here compared to the other methods. The full BDR model slightly outperforms the shrinkage model in both likelihood and in mean-squared error. We also see that the choice of r affects the results; in this case, taking $r = k$ performs somewhat better.

Figure 5 demonstrates in more detail the difference between these models. It shows test set predictions of each model on the bags of different sizes. Here, we can see explicitly that the shrinkage and BDR models are able to take into account the bag size, with decreasing variance for larger bag sizes, while the BLR model just outputs the same variance for all bag size predictions. Furthermore, the shrinkange and BDR models can shrink their predictions towards the mean in the smaller bag sizes without doing so for the larger bags: this improves performance on the small bags while still allowing for good predictions on large bags, contrary to the BLR model.

Fixed bag size: Uncertainty in the regression model. The previous experiment showed the efficacy of the shrinkage estimator in our models, but demonstrated little gain from posterior inference for regression weights β over their MAP estimates, i.e. there is no discernible improvement of BLR over RBF network. To isolate the effect of quantifying uncertainty in the regression model, we now consider the case where there is no variation in bag size at all and normal noise is added onto the observations. In particular we take $N_i = 1000$ and $\varepsilon \sim \mathcal{N}(0, 1)$ and use the same experimental setup as before, sampling landmarks randomly from the training set.

Results are shown in Table 1 (over 10 simulation of the dataset). Here, BLR or BDR outperform all other methods on all runs, highlighting that uncertainty in the regression model is also important for predictive performance. We note that the BDR method performs well in this regime as well as in the previous one.

5.2 IMDb-WIKI: Age Estimation

We now demonstrate our methods on a celebrity age estimation problem, using the IMDb-WIKI database (Rothe et al., 2016) which consists of $397\,949$ images of $19\,545$ celebrities[4], with corresponding age labels. This database was constructed by crawling IMDb for images of its most

[4]We used only the IMDb images, and removed some implausible images, including one of a cat and several of people with supposedly negative age, or ages of several hundred years.

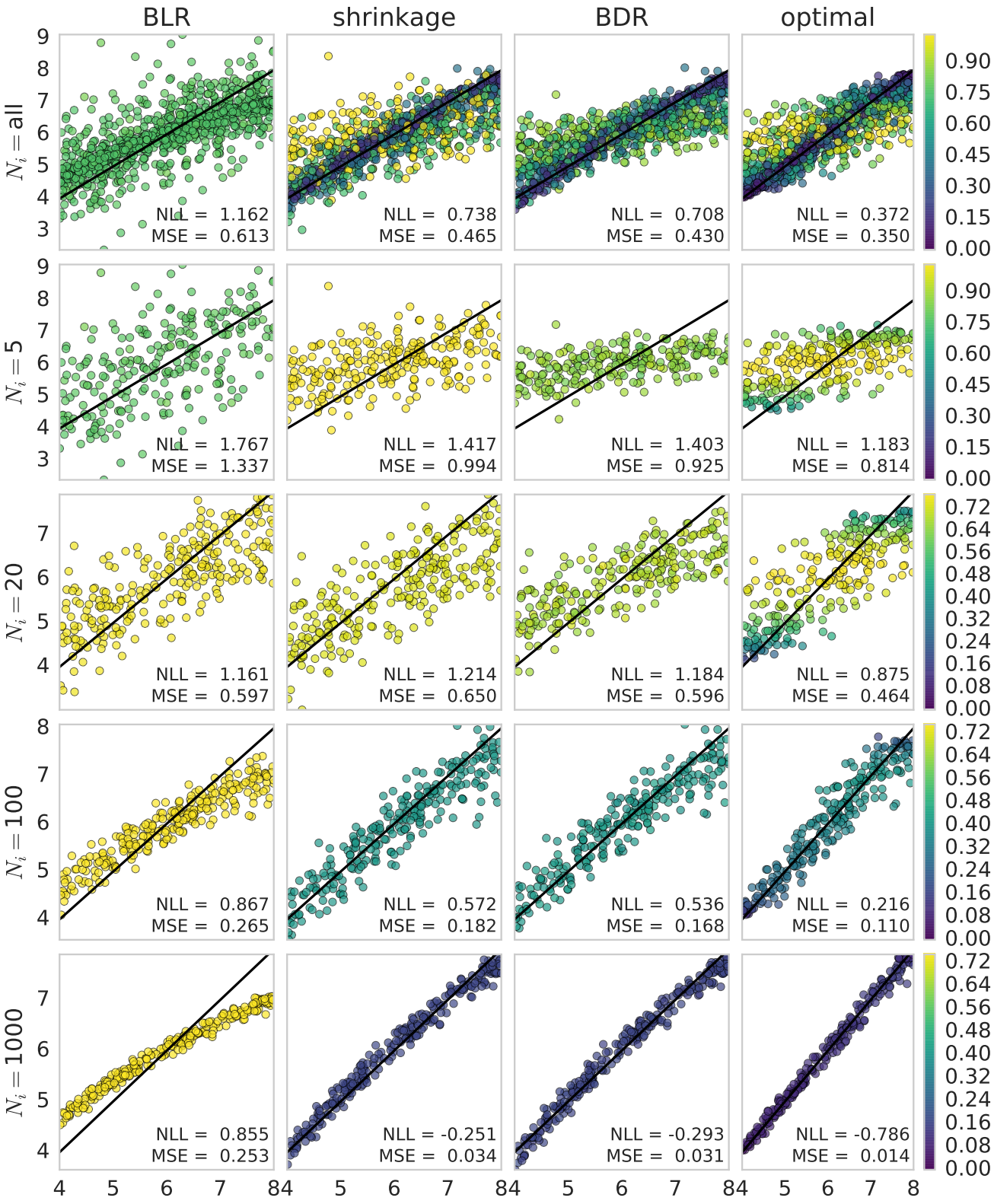

Figure 5: Predictions for the varying bag size experiment of Section 5.1. Each column corresponds to a single prediction method. Each point in an image represents a single bag, with its horizontal position the true label y_i, and its vertical position the predicted label. The black lines show theoretical perfect predictions. The rows represent different subsets of the data: the first row shows all bags, the second only bags with $N_i = 5$, and so on. Colors represent the predictive standard deviation of each point. Note that vertical axis limits and color scales are shared across each row.

Table 1: Results on the synthetic dataset, over 10 runs (standard deviations in parentheses). BLR/BDR performs best on all runs in both metrics.

METHOD	MSE	NLL
Optimal	0.170 (0.009)	0.401 (0.018)
RBF network	0.235 (0.014)	N/A
shrinkage	0.237 (0.014)	0.703 (0.027)
shrinkageC	0.236 (0.013)	0.700 (0.029)
BLR	**0.228 (0.012)**	**0.681 (0.025)**
BDR	**0.227 (0.012)**	**0.683 (0.025)**

Table 2: Results on the grouped IMDb-WIKI dataset over ten runs (standard deviations in parentheses). Here **shrinkage** performs the best across all 10 runs in both metrics.

METHOD	RMSE	NLL
CNN	10.25 (0.22)	3.80 (0.034)
RBF network	9.51 (0.20)	N/A
shrinkage	**9.28 (0.20)**	**3.54 (0.021)**
BLR	9.55 (0.19)	3.68 (0.021)

popular actors and directors, with potentially many images for each celebrity over time. Rothe et al. (2016) use a convolutional neural network (CNN) with a VGG-16 architecture to perform 101-way classification, with one class corresponding to each ages in $\{0, \ldots, 100\}$.

We take a different approach, and assume that we are given several images of a single individual (i.e. samples from the distribution of celebrity images), and are asked to predict their mean age based on several pictures. For example, we have 757 images of Brad Pitt from age 27 up to 51, while we have only 13 images of Chelsea Peretti at ages 35 and 37. Note that 22.5% of the bags have only a single image. We obtain 19 545 bags, with each bag containing between 1 and 796 images of a particular celebrity, with the corresponding bag label calculated from the average of the age labels of the images inside each bag.

In particular, we use the representation $\varphi(x)$ learnt by the CNN in Rothe et al. (2016), where $\varphi(x) : \mathbb{R}^{256\times256} \to \mathbb{R}^{4096}$, from the pixel space of images to the representation before the output of the CNN. With these new representations, we can now treat them as inputs to our radial basis network, shrinkage (taking $r = k$ here) and BLR models. Although we can also use the full BDR model here, due to the computational time and memory required to perform proper parameter tuning, we relegate this to a later study.

Here, we use 9 820 bags for training, 2 948 bags for early stopping, 2 946 for validation and 3 928 for testing. We tune number of landmarks, bandwidth, regularisation and learning rate via the validation set, and learn the other parameters. Landmarks are sampled without replacement from the training set.

We repeat the experiment on 10 different splits of the data, and report the results in Table 2. The baseline *CNN* results give performance by averaging the predictive distribution from the model of Rothe et al. (2016) for each image of a bag; note that this model was trained on all of the images used here. From Table 2, we can see that shrinkage has the best performance here; in fact, it outperforms all other methods in all 10 splits of the dataset, in both metrics. This demonstrates that modeling bag size uncertainty is vital.

6 CONCLUSION

Supervised learning on groups of observations using kernel mean embeddings typically disregards sampling variability within groups. To handle this problem, we construct Bayesian approaches to modelling kernel mean embeddings within a regression model, and investigate advantages of uncertainty propagation within different components of the resulting distribution regression. The ability to take into account the uncertainty in mean embedding estimates is demonstrated to be key for constructing models with good predictive performance when group sizes are highly imbalanced. We also demonstrate that the results of a complex neural network model for age estimation can be improved by the shrinkage model.

Our models employ a neural network formulation, in order to provide more expressive feature representations and learn discriminative embeddings. Doing so makes our model easy to extend to more complicated featurisations than the simple RBF network used here. By training with backpropagation, or via approximate Bayesian methods such as variational inference, we can easily 'learn the kernel' within our framework, for example learning weights in the deep network-based kernel of Section 5.2 rather than using a pre-trained model. We can also apply our networks to structured settings, learning regression functions on sets of images, audio, or text. Such models naturally fit into the empirical Bayes framework.

On the other hand, we might extend our model to more Bayesian feature learning by placing priors over the kernel hyperparameters, building on classic work on variational approaches (Barber and Schottky, 1998) and fully Bayesian inference (Andrieu et al., 2001) in RBF networks. Such approaches are also possible using other featurisations, e.g. random Fourier features (Oliva et al., 2015).

Future distribution regression approaches will need to account for uncertainty in observation of the distribution. Our methods provide a strong, generic building block to do so.

References

Christophe Andrieu, Nando De Freitas, and Arnaud Doucet. Robust full bayesian learning for radial basis networks. *Neural Computation*, 13(10):2359–2407, 2001.

David Barber and Bernhard Schottky. Radial basis functions: a bayesian treatment. *NIPS*, pages 402–408, 1998.

C.M. Bishop. *Pattern recognition and machine learning*. Springer New York, 2006.

David S Broomhead and David Lowe. Radial basis functions, multi-variable functional interpolation and adaptive networks. Technical report, DTIC Document, 1988.

Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, 89 (1):31–71, 1997.

Seth Flaxman, Yu-Xiang Wang, and Alexander J Smola. Who supported Obama in 2012?: Ecological inference through distribution regression. In *KDD*, pages 289–298. ACM, 2015.

Seth Flaxman, Dino Sejdinovic, John P. Cunningham, and Sarah Filippi. Bayesian learning of kernel embeddings. In *UAI*, 2016.

Seth Flaxman, Dougal J. Sutherland, Yu-Xiang Wang, and Yee-Whye Teh. Understanding the 2016 US presidential election using ecological inference and distribution regression with census microdata. 2016. arXiv:1611.03787.

Thomas Gärtner. *Kernels for Structured Data*, volume 72. World Scientific, Series in Machine Perception and Artificial Intelligence, 2008.

Andrew Gelman, David K Park, Stephen Ansolabehere, Phillip N Price, and Lorraine C Minnite. Models, assumptions and model checking in ecological regressions. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 164(1):101–118, 2001.

Matthew D. Hoffman and Andrew Gelman. The no-U-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *JMLR*, pages 1593–1623, 2014.

Christopher Jackson, Nicky Best, and Sylvia Richardson. Improving ecological inference using individual-level data. *Statistics in medicine*, 25(12):2136–2159, 2006.

Tony Jebara, Risi Imre Kondor, and Andrew Howard. Probability product kernels. *JMLR*, 5:819–844, 2004.

Wittawat Jitkrittum, Arthur Gretton, Nicolas Heess, S. M. Ali Eslami, Balaji Lakshminarayanan, Dino Sejdinovic, and Zoltán Szabó. Kernel-Based Just-In-Time Learning for Passing Expectation Propagation Messages. In *UAI*, 2015.

Gary King. *A Solution to the Ecological Inference Problem*. Princeton University Press, 1997. ISBN 0691012407.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. arXiv:1412.6980.

Gabriel Krummenacher, Cheng Soon Ong, and Joachim M Buhmann. Ellipsoidal multiple instance learning. In *ICML (2)*, pages 73–81, 2013.

Hendrik Kück and Nando de Freitas. Learning about individuals from group statistics. In *UAI*, pages 332–339, 2005.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. From word embeddings to document distances. In *ICML*, pages 957–966, 2015.

H. C. L. Law, C. Yau, and D. Sejdinovic. Testing and Learning on Distributions with Symmetric Noise Invariance. *Advances in Neural Information Processing Systems (NIPS)*, 2017. arXiv:1703.07596.

David Lopez-Paz, Krikamol Muandet, Bernhard Schölkopf, and Ilya Tolstikhin. Towards a learning theory of cause-effect inference. In *ICML*, 2015.

Milan Lukić and Jay Beder. Stochastic processes with sample paths in reproducing kernel hilbert spaces. *Transactions of the American Mathematical Society*, 353(10): 3945–3969, 2001.

Yifei Ma, Dougal J. Sutherland, Roman Garnett, and Jeff Schneider. Active pointillistic pattern search. In *AISTATS*, 2015.

J. Mitrovic, D. Sejdinovic, and Y.W. Teh. DR-ABC: Approximate Bayesian Computation with Kernel-Based Distribution Regression. In *ICML*, pages 1482–1491, 2016.

Pedro J Moreno, Purdy P Ho, and Nuno Vasconcelos. A Kullback-Leibler divergence based kernel for SVM classification in multimedia applications. In *NIPS*, 2003.

Krikamol Muandet, Kenji Fukumizu, Francesco Dinuzzo, and Bernhard Schölkopf. Learning from distributions via support measure machines. In *NIPS*, 2012. arXiv:1202.6504.

Krikamol Muandet, Kenji Fukumizu, Bharath Sriperumbudur, Arthur Gretton, and Bernhard Schoelkopf. Kernel mean estimation and stein effect. In *ICML*, 2014.

Michelle Ntampaka, Hy Trac, Dougal J. Sutherland, Nicholas Battaglia, Barnabás Póczos, and Jeff Schneider. A machine learning approach for dynamical mass measurements of galaxy clusters. *The Astrophysical Journal*, 803(2):50, 2015. ISSN 1538-4357. arXiv:1410.0686.

Michelle Ntampaka, Hy Trac, Dougal J. Sutherland, S. Fromenteau, B. Poczos, and Jeff Schneider. Dynamical mass measurements of contaminated galaxy clusters using machine learning. *The Astrophysical Journal*, 831 (2):135, 2016. arXiv:1509.05409.

Junier B Oliva, Barnabás Póczos, and Jeff Schneider. Distribution to distribution regression. In *ICML*, 2013.

Junier B Oliva, Avinava Dubey, Barnabás Póczos, Jeff Schneider, and Eric P Xing. Bayesian nonparametric kernel-learning. Technical report, 2015. arXiv:1506.08776.

Giorgio Patrini, Richard Nock, Tiberio Caetano, and Paul Rivera. (Almost) no label no cry. In *NIPS*. 2014.

Natesh S Pillai, Qiang Wu, Feng Liang, Sayan Mukherjee, and Robert L Wolpert. Characterizing the function space for bayesian kernel models. *Journal of Machine Learning Research*, 8(Aug):1769–1797, 2007.

Barnabás Póczos, Liang Xiong, and Jeff Schneider. Nonparametric divergence estimation with applications to machine learning on distributions. In *UAI*, 2011.

Barnabas Poczos, Aarti Singh, Alessandro Rinaldo, and Larry Wasserman. Distribution-free distribution regression. In *AISTATS*, pages 507–515, 2013. arXiv:1302.0082.

Novi Quadrianto, Alex J Smola, Tiberio S Caetano, and Quoc V Le. Estimating labels from label proportions. *JMLR*, 10:2349–2374, 2009.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *NIPS*, pages 1177–1184, 2007.

Aaditya Ramdas and Leila Wehbe. Nonparametric independence testing for small sample sizes. In *IJCAI*, 2015. arXiv:1406.1922.

Rasmus Rothe, Radu Timofte, and Luc Van Gool. Deep expectation of real and apparent age from a single image without facial landmarks. *International Journal of Computer Vision (IJCV)*, July 2016.

Craig Saunders, Alexander Gammerman, and Volodya Vovk. Ridge regression learning algorithm in dual variables. In *ICML*, 1998.

Bernhard Schölkopf, Ralf Herbrich, and Alex J. Smola. A generalized representer theorem. In *COLT*, 2001.

Bharath K Sriperumbudur, Arthur Gretton, Kenji Fukumizu, Bernhard Schölkopf, and Gert RG Lanckriet. Hilbert space embeddings and metrics on probability measures. *JMLR*, 99:1517–1561, 2010.

Stan Development Team. Stan: A c++ library for probability and sampling, version 2.5.0, 2014. URL `http://mc-stan.org/`.

Ingo Steinwart. Convergence types and rates in generic Karhunen-Loéve expansions with applications to sample path properties. *arXiv preprint arXiv:1403.1040v3*, March 2017.

Dougal J. Sutherland. *Scalable, Flexible, and Active Learning on Distributions*. PhD thesis, Carnegie Mellon University, 2016.

Zoltán Szábo, Bharath K. Sriperumbudur, Barnabás Póczos, and Arthur Gretton. Leraning theory for distribution regression. *JMLR*, 17(152):1–40, 2016. arXiv:1411.2066.

Grace Wahba. *Spline models for observational data*, volume 59. Siam, 1990.

Yuya Yoshikawa, Tomoharu Iwata, and Hiroshi Sawada. Latent support measure machines for bag-of-words data classification. In *NIPS*, pages 1961–1969, 2014.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan Salakhutdinov, and Alexander Smola. Deep sets. *Advances in Neural Information Processing Systems (NIPS)*, 2017.

Kai Zhang and James T. Kwok. Clustered nystsröm method for large scale manifold learning and dimension reduction. *IEEE Transactions on Neural Networks*, pages 1576–1587, 2010.

Zhi-Hua Zhou, Yu-Yin Sun, and Yu-Feng Li. Multi-instance learning by treating instances as non-iid samples. In *ICML*, 2009.

A Choice of $r(\cdot,\cdot)$ to ensure $\mu_{\mathsf{P}} \in \mathcal{H}_k$

We need to choose an appropriate covariance function r, such that $\mu_{\mathsf{P}} \in \mathcal{H}_k$, where $\mu_{\mathsf{P}} \sim \mathcal{GP}(0, r(\cdot,\cdot))$. In particular, it is for infinite-dimensional RKHSs not sufficient to define $r(\cdot,\cdot) = k(\cdot,\cdot)$, as draws from this particular prior are no longer in $\mathcal{H}_k$ (Wahba, 1990) (but see below). However, we can construct

$$r(x,y) = \int k(x,z)k(z,y)\nu(dz) \tag{8}$$

where ν is any finite measure on $\mathcal{X}$. This then ensures $\mu_{\mathsf{P}} \in \mathcal{H}_k$ with probability 1 by the nuclear dominance (Lukić and Beder, 2001; Pillai et al., 2007) for any stationary kernel k. In particular, Flaxman et al. (2016) provides details when k is a squared exponential kernel defined by

$$k(x,y) = \exp(-\frac{1}{2}(x-y)^\top \Sigma_k^{-1}(x-y)) \qquad x, y \in \mathbb{R}^p$$

and $\nu(dz) = \exp\left(-\frac{||z||_2^2}{2\ell^2}\right)dz$, i.e. it is proportional to a Gaussian measure on $\mathbb{R}^d$, which provides $r(\cdot,\cdot)$ with a non-stationary component. In this paper, we take $\Sigma_k = \sigma^2 I_p$, where σ^2 and ℓ are tuning parameters, or parameters that we learn.

Here, the above holds for a general set of stationary kernels, but note that by taking a convolution of a kernel with itself, it might make the space of functions that we consider overly smooth (i.e. concentrated on a small part of $\mathcal{H}_k$). In this work, however, we consider only the Gaussian RBF kernel k. In fact, recent work (Steinwart, 2017, Theorem 4.2) actually shows that in this case, the sample paths almost surely belong to (interpolation) spaces which are infinitesimally larger than the RKHS of the Gaussian RBF kernel. This suggests that we can choose r to be an RBF kernel with a length scale that is infinitesimally bigger than that of k; thus, in practice, taking $r = k$ would suffice and we do observe that it actually performs better (Fig. 4).

B Framework for Binary Classification

Suppose that our labels $y_i \in \{0,1\}$, i.e. we are in a binary classification framework. Then a simple approach to accounting for uncertainty in the regression parameters is to use bayesian logistic regression, putting priors on β, i.e.

$$\begin{aligned} \beta &\sim \mathcal{N}(0,\rho^2) \\ y_i &\sim Ber(\pi_i), \text{ where } \log\left(\frac{\pi_i}{1-\pi_i}\right) = \beta^\top \hat{\mu}_i \end{aligned}$$

however for the mean shrinkage pooling model, if we use the above $y_i \mid \mu_i, \alpha$, we would not be able to obtain an analytical solution for $p(y_i|\mathbf{x_i}, \alpha)$. Instead we use the probit link function, as given by:

$$Pr(y_i = 1|\mu_i, \alpha) = \Phi\left(\alpha^\top \mu_i(\mathbf{z})\right)$$

where Φ denotes the Cumulative Distribution Function (CDF) of a standard normal distribution, with $\mu_i(\mathbf{z}) = [\mu_i(z_1), \ldots, \mu_i(z_s)]^\top$. Then as before we have

$$\mu_i(\mathbf{z}) \mid \mathbf{x_i} \sim \mathcal{N}(M_i, C_i)$$

with M_i and C_i as defined in section 4.2. Hence, as before

$$\begin{aligned} Pr(y_i = 1|\mathbf{x}_i, \alpha) &= \int Pr(y_i = 1|\mu_i, \alpha)p(\mu_i(\mathbf{z})|\mathbf{x}_i)d\mu_i(\mathbf{z}) \\ &= c\int \Phi(\alpha^\top \mu_i(\mathbf{z}))\exp\{-\frac{1}{2}(\mu_i(\mathbf{z}) - M_i)^\top C_i^{-1}(\mu_i(\mathbf{z}) - M_i)\}d\mu_i(\mathbf{z}) \\ (\text{with } l_i = \mu_i(\mathbf{z}) - M_i) &= c\int \Phi(\alpha^\top (l_i + M_i))\exp\{-\frac{1}{2}(l_i)^\top C_i^{-1}(l_i)\}dl_i \\ &= Pr(Y \leq \alpha^\top (l_i + M_i)) \end{aligned}$$

Note here $Y \sim \mathcal{N}(0,1)$ and $l_i \sim \mathcal{N}(0, \Sigma_i)$ Then expanding and rearranging

$$Pr(y_i = 1|\mathbf{x}_i, \alpha) \quad = \quad Pr(Y - \alpha^\top l_i \leq \alpha^\top M_i)$$

Note that since Y and l_i independent normal r.v., $Y - \alpha^\top l_i \sim \mathcal{N}(0, 1 + \alpha^\top C_i \alpha^\top)$. Let T be standard normal, then we have:

$$\begin{aligned} Pr(y_i = 1|\mathbf{x}_i, \alpha) &= Pr(\sqrt{1 + \alpha^\top C_i \alpha}\, T \leq \alpha^\top M_i) \\ &= Pr(T \leq \frac{\alpha^\top M_i}{\sqrt{1 + \alpha^\top C_i \alpha}}) \\ &= \Phi\left(\frac{\alpha^\top M_i}{\sqrt{1 + \alpha^\top C_i \alpha}}\right) \end{aligned}$$

Hence, we also have:

$$Pr(y_i = 0|\mathbf{x}_i, \alpha) \quad = \quad 1 - \Phi\left(\frac{\alpha^\top M_i}{\sqrt{1 + \alpha^\top C_i \alpha}}\right)$$

Now placing the prior $\alpha \sim \mathcal{N}(0, \rho^2 K_{\mathbf{z}}^{-1})$, we have the following MAP objective:

$$\begin{aligned} J(\alpha) &= \log\left[p(\alpha)\prod_{i=1}^{n} p(y_i|\mathbf{x}_i, \alpha)\right] \\ &= \sum_{i=1}^{n}(1 - y_i)\log(1 - \Phi\left(\frac{\alpha^\top M_i}{\sqrt{1 + \alpha^\top C_i \alpha}}\right)) \\ &\quad + y_i \log(\Phi\left(\frac{\alpha^\top M_i}{\sqrt{1 + \alpha^\top C_i \alpha}}\right)) + \frac{1}{\rho^2}\alpha^\top K_{\mathbf{z}} \alpha \end{aligned}$$

Since we have an analytical solution for $Pr(y_i = 0|\mathbf{x}_i, \alpha)$, we can also use this in HMC for BDR.

C Some more intuition on the shrinkage estimator

In this section, we provide some intuition behind the shrinkage estimator in section 4.2. Here, for simplicity, we choose $\Sigma_i = \tau^2 I$ for all bag i, and $m_0 = 0$, and consider the case where $\mathbf{z} = \mathbf{u}$, i.e. $R = R_{\mathbf{z}} = R_{\mathbf{zz}}$. We can then see that if R has eigendecomposition $U\Lambda U^T$, with $\Lambda = \mathrm{diag}(\lambda_k)$, the posterior mean is

$$U\,\mathrm{diag}\left(\frac{\lambda_k}{\lambda_k + \tau^2/N_i}\right)U^T(\hat{\mu}_i),$$

so that large eigenvalues, $\lambda_k \gg \tau^2/N_i$, are essentially unchanged, while small eigenvalues, $\lambda_k \ll \tau^2/N_i$, are shrunk towards 0. Likewise, the posterior variance is

$$U\,\mathrm{diag}\left(\lambda_k - \frac{\lambda_k^2}{\lambda_k + \frac{\tau^2}{N_i}}\right)U^T = U\,\mathrm{diag}\left(\frac{1}{\frac{N_i}{\tau^2} + \frac{1}{\lambda_k}}\right)U^T;$$

its eigenvalues also decrease as N_i/τ^2 increases.

D Alternative Motivation for choice of f

Here we provide an alternative motivation for the choice of $f = \sum_{s=1}^{k} \alpha_s k(\cdot, z_s)$. First, consider the following Bayesian model with a linear kernel K on μ_i, where $f : \mathcal{H}_k \to \mathbb{R}$:

$$y_i \mid \mu_i, f \quad \sim \quad \mathcal{N}\left(f(\mu_i), \sigma^2\right).$$

Now considering the log-likelihood of $\{\mu, Y\} = \{\mu_i, y_i\}_{i=1}^n$ (supposing we have these exact embeddings), we obtain:

$$\log p(Y|\mu, f) = \sum_{i=1}^{n} -\frac{1}{2\sigma^2}(y_i - f(\mu_i))^2$$

To avoid over-fitting, we place a Gaussian prior on f, i.e. $-\log p(f) = \lambda||f||_{\mathcal{H}_k} + c$. Minimizing the negative log-likelihood over $f \in \mathcal{H}_k$, we have:

$$f^* = \text{argmin}_{f\in\mathcal{H}_k} \sum_{i=1}^{n} \frac{1}{2\sigma^2}(y_i - f(\mu_i))^2 + \lambda||f||_{\mathcal{H}_k}$$

Now this is in the form of an empirical risk minimisation problem. Hence using the representer theorem (Schölkopf et al., 2001), we have that:

$$f = \sum_{j=1}^{n} \gamma_j K(., \mu_j)$$

i.e. we have a finite-dimensional problem to solve. Thus since K is a linear kernel:

$$y_i \mid \mu_i, \{\mu_j\}_{j=1}^n, \gamma \quad \sim \quad \mathcal{N}\left(\sum_{j=1}^{n} \gamma_j \langle \mu_i, \mu_j \rangle_{\mathcal{H}_k}, \sigma^2\right).$$

where $\langle \mu_i, \mu_j \rangle_{\mathcal{H}_k}$ can be thought of as the similarity between distributions.

Now we have the same $\mathcal{GP}$ posterior as in Section 4.2, and we would like to compute $p(y_i|\mathbf{x_i}, \gamma)$. This suggests we need to integrate out $\mu_1, \ldots \mu_n$. But it is unclear how to perform this integration, since the μ_i follow Gaussian process distributions. Hence we can take an approximation to f, i.e. $f = \sum_{s=1}^{k} \alpha_s k(\cdot, z_s)$, which would essentially give us a dual method with a sparse approximation to f.

E Stan source code for Bayesian Distribution Regression model

```
data {
  int d; // dimensionality of the observed data
  int p; // number of bags
  int ntrain; // 1 ... ntrain are for training and ntrain+1 ... p are for testing

  matrix[p,d] mu;
  matrix[d,d] Sigma[p];
  vector[ntrain] y; // labels
  vector[p] ytrue; // labels (train+test)
}
parameters {
  vector[d] beta;

  real<lower=0> sigma;
  real alpha;
  real<lower=0> kappa;
}
transformed parameters {
  vector[p] mus;
  vector[p] sds;

  for(j in 1:p) {
    mus[j] = alpha + mu[j] * beta;
    sds[j] = sqrt(quad_form(Sigma[j],beta) + sigma);
```

```
  }
}
model {
  for(j in 1:ntrain)
      y[j] ~ normal(mus[j],sds[j]);

  alpha ~ normal(0,2);
  beta ~ normal(0,kappa);
  kappa ~ normal(0,2);
  sigma ~ normal(0,2);
}
generated quantities {
  vector[p] yhat;
  vector[p] lp;
  for(j in 1:p) {
    yhat[j] = normal_rng(mus[j],sds[j]);
    lp[j] = normal_lpdf(ytrue[j] | mus[j],sds[j]);
  }
}
```