Quasi-Bayesian Nonparametric Density Estimation via Autoregressive Predictive Updates

Sahra Ghalebikesabi\(^1\) Chris Holmes\(^2\) Edwin Fong\(^{*2}\) Brieuc Lehmann\(^{*3}\)

\(^1\)University of Oxford \(^2\)Novo Nordisk \(^3\)University College London

Abstract

Bayesian methods are a popular choice for statistical inference in small-data regimes due to the regularization effect induced by the prior. In the context of density estimation, the standard nonparametric Bayesian approach is to target the posterior predictive of the Dirichlet process mixture model. In general, direct estimation of the posterior predictive is intractable and so methods typically resort to approximating the posterior distribution as an intermediate step. The recent development of quasi-Bayesian predictive copula updates, however, has made it possible to perform tractable predictive density estimation without the need for posterior approximation. Although these estimators are computationally appealing, they struggle on non-smooth data distributions. This is due to the comparatively restrictive form of the likelihood models from which the proposed copula updates were derived. To address this shortcoming, we consider a Bayesian nonparametric model with an autoregressive likelihood decomposition and a Gaussian process prior. While the predictive update of such a model is typically intractable, we derive a quasi-Bayesian update that achieves state-of-the-art results in small-data regimes.

1 INTRODUCTION

Modelling the joint distribution of multivariate random variables with density estimators is a central topic in modern unsupervised machine learning research [Durkan et al., 2019, Papamakarios et al., 2017]. As well as providing insight into the statistical properties of the data, density estimates are used in a number of downstream applications, including image restoration [Zoran and Weiss, 2011], density-based clustering [Scaldelai et al., 2022], and simulation-based inference [Lueckmann et al., 2021]. In small-data regimes, Bayesian methods are a popular choice for a wide range of machine learning tasks, including density estimation, thanks to their attractive generalization capacities. For density estimation, the typical Bayesian approach is to target the Bayesian predictive density, \( p_n(x) = \int f(x|\theta)\pi_n(\theta)d\theta \), where \( \pi_n \) denotes the posterior density of the model parameters \( \theta \) after observing \( x_1, \ldots, x_n \), and \( f \) denotes the likelihood function.

De Finetti’s representation theorem [De Finetti, 1937, Hewitt and Savage, 1955] states that an exchangeable joint density fully characterises a Bayesian model, which then implies a sequence of predictive densities. Further, Fong et al. [2021] recently showed that a sequence of predictive densities can be sufficient for full Bayesian posterior inference. This provides theoretical motivation for an iterative approach to Bayesian predictive density estimation by updating the predictive \( p_{i-1}(x) \) to \( p_i(x) \) given observation \( x_i \) for \( i = 1, \ldots, n \). The idea of recursive Bayesian updates goes back to at least Hill [1968], but was only recently made more widely applicable through the relaxation of the assumption of exchangeability in favour of conditionally identically distributed [Berti et al., 2004] sequences.

Here, we focus on a particular class of one-step-ahead predictive updates \( p_{i-1}(x) \rightarrow p_i(x) \) based on bivariate copulas, which were first introduced by Hahn et al. [2018] for univariate data, and extended by Fong et al. [2021] to the multivariate setting and to regression analyses. This class of updates is inspired by Bayesian models and thus retains many desirable Bayesian properties, such as coherence and regularization. However, we emphasize that the copula updates do not correspond exactly, nor approximately, to a traditional Bayesian likelihood-prior model, and we thus refer to them...
We briefly recap predictive density estimation via biwork based approaches, such as masked autoregressive variate copula updates, before describing a particular inspired density estimator. In particular:

- By considering a Dirichlet Process Mixture Model (DPMM) with an AR likelihood and a Gaussian process (GP) prior, we formulate a tractable copula update based on which we propose a new Dirichlet Process Mixture Model (DPMM) inspired density estimator. In particular:

  - Contributions This motivates our main contribution, namely the formulation of a more flexible autoregressive (AR) copula update based on which we propose a new Dirichlet Process Mixture Model (DPMM) inspired density estimator.

  - By considering a DPMM with an AR likelihood and a Gaussian process (GP) prior, we formulate a tractable copula update with a novel data-dependent bandwidth based on the Euclidean metric in data space. Our method, Autoregressive Recursive Bayesian Predictives (AR-BP), outperforms traditional density estimators on tabular data with up to 63 features, and 10,000 samples.

  - We observe in practice that the Euclidean metric used in AR-BP can be inadequate for highly non-smooth data distributions. For such cases, we propose using an AR neural network [Bengio and Bengio 1999; Frey et al. 1998; Germain et al. 2015; Larochelle and Murray 2011] that maps the observations into a latent space before bandwidth estimation. This introduces additional non-linearity through the dependence of the bandwidth on the data, leading to a density estimator, ARnet-BP, that is more accurate on non-smooth densities.

2 BACKGROUND

We briefly recap predictive density estimation via biwork updates, before describing a particular such update inspired by DPMMs.

2.1 UNIVARIATE PREDICTIVE DENSITY UPDATES

To compute predictive densities quickly, Hahn et al. [2018] propose an iterative approach. For $x \in \mathbb{R}$, any sequence of Bayesian posterior predictive densities $p_i(x)$ with likelihood $f$ and posterior $\pi_i$, conditional on $x_{1:i}$, can be expressed as

$$p_i(x) = \int f(x|\theta)\pi_i(\theta)d\theta = p_{i-1}(x)h_i(x, x_i),$$  

(1)

for some bivariate function $h_i(x, x_i)$ [Hahn et al. 2018].

Rearranging for $h_i$, we have

$$h_i(x, x_i) = \frac{p_i(x)}{p_{i-1}(x)} = \frac{(a)p_{i-1}(x|x_i)}{(b)p_{i-1}(x)} = \frac{p_{i-1}(x, x_i)}{p_{i-1}(x)}$$  

(2)

where ($a$) holds by definition, and ($b$) $p_{i-1}(x|x_i) = p_{i-1}(x|x_i)p_{i-1}(x_i) = p_i(x)p_{i-1}(x_i)$ holds by Bayes’ law. Hahn et al. [2018] show that $h_i(x, x_i)$ is the transformation of a bivariate copula density. A bivariate copula is a bivariate cumulative distribution function (CDF) $C : [0, 1]^2 \rightarrow [0, 1]$ with uniform marginal distributions that is used to characterise the dependence between two random variables independent of their marginals:

**Theorem 1** (Sklar’s theorem [Sklar 1959]). For any bivariate density $f(y_1, y_2)$ with continuous marginal CDFs $F_1(y_1)$ and $F_2(y_2)$, and marginal densities $f_1(y_1)$ and $f_2(y_2)$, there exists a unique bivariate copula $C$ with density $c$ such that

$$f(y_1, y_2) = c \{ F_1(y_1), F_2(y_2) \} f_1(y_1)f_2(y_2).$$

Applying the copula factorization from Sklar’s theorem to (2) yields that there exists some bivariate copula density $c_i$ such that $p_{i-1}(x, x_i) = c_i(P_{i-1}(x), P_{i-1}(x_i))p_{i-1}(x)p_{i-1}(x_i)$, and thus $h_i(x, x_i) = c_i(P_{i-1}(x), P_{i-1}(x_i))$, where $P_{i-1}$ is the CDF corresponding to the predictive density $p_{i-1}$. Given prior $\pi$ and likelihood $f$, Equation 2 suggests that the update function can be written as

$$h_i(x, x_i) = \frac{\int f(x|\theta)f_{x|x_i}(\theta)p_{i-1}(\theta)d\theta}{\int f(x|\theta)p_{i-1}(\theta)d\theta \int f(x|\theta)p_{i-1}(\theta)d\theta}.$$
For each Bayesian model, there is thus a unique sequence of symmetric copula densities $c_i(u, v) = c_i(v, u)$. This sequence has the property that $c_n(\cdot, \cdot) \to 1$ converges to a constant function as $n \to \infty$, ensuring that the predictive density converges asymptotically with sample size $n$.

In general, the above equation is intractable due to the posterior so it is not possible to compute the iterative update in (1) for fully Bayesian models. Alternatively, we will consider sequences of $h_i$ that match the Bayesian model for $i = 1$, but not for $i > 1$. As mentioned above, this copula update no longer corresponds to a Bayesian model, nor are the resulting predictive density estimates approximations to a Bayesian model. Nevertheless, if the copula updates are conditionally identically distributed, they still exhibit desirable Bayesian characteristics such as coherence and regularization, and are hence referred to as quasi-Bayes. Please refer to [Berti et al., 2004] for details.

2.2 MULTIVARIATE PREDICTIVE DENSITY UPDATES

The above arguments cannot directly be extended to multivariate $x \in \mathbb{R}^d$ since $h_i$ cannot necessarily be written as $c_i \{P_{i-1}(x), P_{i-1}(x_i)\}$ for $d > 1$. However, (2) still holds, and recursive predictive updates with bivariate copulas as building blocks can be derived explicitly given a pre-defined likelihood model and a prior, which we now exhibit.

[Hahn et al., 2018] and [Fong et al., 2021] propose to use DPMMs as a general-use nonparametric model. The DPMM [Escobar, 1988] Escobar and West [1995] can be written as

$$ f(x|G) = \int_{\Theta} K(x|\theta) dG(\theta), \text{ with } G \sim \text{DP}(c, G_0) \quad (3) $$

where $\theta \in \Theta = \mathbb{R}^d$ are parameter vectors, the prior assigned to $G$ is a Dirichlet process (DP) prior with base measure $G_0$ and concentration parameter $c > 0$ [Ferguson, 1973], and $K(x|\theta)$ is a user-specified kernel (not to be confused with the covariance function of a GP). In particular, Fong et al. [2021] consider the base measure $G_0 = \mathcal{N}(0, \tau^{-1}I_d)$ for some precision parameter $\tau \in \mathbb{R}_{>0}$, and the factorized kernel $K(x|\theta) = \mathcal{N}(x|\theta, I_d)$ where $I_d$ is the $d$-dimensional identity matrix. The likelihood is then

$$ f(x|G) = \int \prod_{j=1}^d \mathcal{N}(x^j | \theta^j, 1) dG(\theta), \quad (4) $$

where the dimensions of $x$ are conditionally independent given $\theta$. Following Hahn et al. [2018], we denote the dimension $j$ of a vector $y$ with $y^j$. We note that the strong assumption of a factorised kernel form drastically impacts the performance of the regular [DPMM] and also influences the form and modelling capacity of the corresponding copula update.

This model inspires the following recursive predictive density update $p_i(x) = h_i(x, x_i)p_{i-1}(x)$ for which the first $d' \in \{1, \ldots, d\}$ marginals take on the form

$$ \frac{p_i(x^{1:d'})}{p_{i-1}(x^{1:d'})} = 1 - \alpha_i + \alpha_i \prod_{j=1}^{d'} c \left( u_{i-1}^j(x^j), v_{i-1}^j; \rho_0 \right), \quad (5) $$

$$ u_{i-1}^j(x^j) := p_{i-1} \left( x^j | x^{1:j-1} \right), \quad v_{i-1}^j := p_{i-1} \left( x_i^j | x_i^{1:j-1} \right), $$

where $c(u, v; \rho_0)$ is the bivariate Gaussian copula density with correlation $\rho_0 = 1/(1 + \tau)$, $\rho_0$ can be any chosen prior density, and $\alpha_i = (2 - 1/\tau)^{1/\tau}$ (see Supplement A and Fong et al. [2021]). Note that the above update requires a specific ordering of the feature dimensions, and the Gaussian copula follows from the Gaussian distribution in the kernel and $G_0$ for the [DPMM]. Unlike the DPMM, there are now no underlying parameters (beyond $\rho_0$) in the copula update as we have integrated out $\theta$, so we do not carry out clustering directly. While $\rho_0$ is a scalar here, Fong et al. [2021] also consider the setting with a distinct bandwidth parameter for each dimension. We refer to these recursive Bayesian predictives as R$_d$-BP, or simply R-BP if the dimensions share a single bandwidth.

3 AR-BP: AUTOREGRESSIVE BAYESIAN PREDICTIVES

For smooth data distributions, the recursive update defined in (5) generates density estimates that are highly competitive against other popular density estimation procedures such as kernel density estimation (KDE) and [DPMM] Fong et al. [2021]. Moreover, the iterative updates provide a fast estimation alternative to fitting the full [DPMM] through Markov chain Monte Carlo (MCMC). When considering more structured data, however, performance suffers due to the choices of the factorized kernel $K(\cdot|\theta) = \mathcal{N}(\cdot|\theta, I_d)$ and simple base measure $G_0 = \mathcal{N}(0, \tau^{-1}I_d)$ in the [DPMM]. These choices induce a priori independence between the data dimensions, and are thus insufficiently flexible to capture more complex dependencies.

3.1 BAYESIAN MODEL FORMULATION

We therefore propose employing more general kernels and base measures in the [DPMM] and show that these inspire a more general tractable recursive predictive
update. In particular, we allow the kernel to take on an autoregressive structure

\[ K(x|\theta) = \prod_{j=1}^{d} \mathcal{N}(x^j | \theta^j(x^{1:j-1}), 1), \] (6)

where \( \theta^j : \mathbb{R}^j \rightarrow \mathbb{R} \) is now an unknown mean function, and not scalar, for dimension \( x^j \), which we allow to depend on the previous \( j-1 \) dimensions of \( x \). Thus, specifying our \text{DPMM} requires a base measure supported on the function space in which \( \theta^1, \ldots, \theta^d \) is valued. We specify this base measure as a product of independent \text{GP} priors on the functional parameters

\[ \theta^j \sim \text{GP}(0, \tau^{-1}k^j) \text{ for } j = 1, \ldots, d \] (7)

where \( k^j : \mathbb{R}^{j-1} \times \mathbb{R}^{j-1} \rightarrow \mathbb{R} \) and \( k^j \) can be any given covariance function that takes as input a pair of \( x^{1:j-1} \) values. In practice, we use the same functional form of \( k \) for each \( j \), so we will drop the superscript \( j \). For later convenience, we have also written the scaling term \( \tau^{-1} \) explicitly. We highlight that for \( j = 1 \), \( \theta^1 \sim \mathcal{N}(0, \tau^{-1}) \). Under this choice, the mean of the normal kernels in the \text{DPMM} for each dimension \( j \) is thus a flexible function of the first \( j-1 \) dimensions \( x^{1:j-1} \), on which we elicit independent \text{GP} priors. The conjugacy of the \text{GP} with the Gaussian \text{DPMM} kernel in (6) is crucial for deriving a tractable density update.

Remark. The proposed \text{DPMM} kernel in (6) is in fact more flexible than a general multivariate kernel, \( K(x | \theta) = \mathcal{N}(x | \theta, \Sigma) \). This is because the multivariate kernel also implies an \text{AR} form like (6) but where the parameters \( \theta^j \) are restricted to be linear in \( x^{1:j-1} \); see Wade et al. [2014] for details.

3.2 ITERATIVE PREDICTIVE DENSITY UPDATES

Computing the Bayesian posterior predictive density induced by the \text{DPMM} with kernel given by (6) and base measure given by (7) through posterior estimation is intractable and requires \text{MCMC}. However, as before, we can utilize the model to derive tractable iterative copula updates. In Supplement A.1, we derive the corresponding recursive predictive density update \( p_i(x) = h_i(x, x_i)p_{i-1}(x) \) for the first \( d' \) marginals and show that it takes on the form

\[ p_{i-1}(x^{1:d'}) / p_{i-1}(x^{1:d'}) = 1 - \alpha_i + \alpha_{i-1}, \] (8)

\[ d' \prod_{j=1}^{d'} c \left( u^j_{i-1}(x^j), v^j_{i-1}; \rho^j(x^{1:j-1}, x^{1:j-1}) \right), \]

with \( u^j_{i-1}(x^j), v^j_{i-1} \) defined as in (5), \( \alpha_i = (2 - \frac{1}{t}) \frac{1}{1 + t} \), and the bandwidth given by

\[ \rho^j(x^{1:j-1}, x^{1:j-1}) = \rho_0 k \left( x^{1:j-1}, x^{1:j-1} \right), \] (9)

for \( \rho_0 = 1/(1 + \tau) \), and \( \rho^j_0 = \rho_0 \). Where appropriate, we henceforth drop the argument \( x \) for brevity. The conditional CDF \( u^j_{i-1} \) can also be computed through an iterative closed form expression similarly to (5) (Supplement B.3). Please see Figure 2 for a simplified overview of the density estimation pipeline.

Note that the estimation is identical to the update given in (5) induced by the factorized \text{DPMM} kernel, except for the main difference that the bandwidth \( \rho \) is no longer a constant, but is now data-dependent. More precisely, the bandwidth for dimension \( j \) is a transformation of the \text{GP} covariance function \( k \) on the first \( j-1 \) dimensions. The additional flexibility afforded by the inclusion of \( k \) enables us to capture more complex dependency structures, as we do not enforce a-priori independence between the dimensions of the parameter \( \theta \). Similarly to the extension of R-BP to R2-BP, we can also define AR2-BP by introducing dimension dependence in \( \rho_0 \). Finally, we highlight that extending R-BP to mixed data is possible as given in Appendix E.1.3 of Fong et al. [2021], which also extends naturally to AR-BP.

Remark. The data-dependent bandwidth also appears when starting from other Bayesian nonparametric models, such as dependent \text{DPMs} and \text{GPs} (see Supplement
Our approach can be viewed as a Bayesian version of an online KDE procedure. To see this, note that a KDE trained on \( i - 1 \) observations – yielding the density estimate \( q_{i-1}(x) \) – can be updated after observing the \( i \)th observation \( x_i \) via \( q_i(x) = (1 - \alpha_i)q_{i-1}(x) + \alpha_id(x, x_i) \), where \( \alpha_i = 1/i \) and \( d(\cdot, \cdot) \) denotes the kernel of the KDE. Rather than adding a weighted kernel term directly, AR-BP instead adds an adaptive kernel that depends on a notion of distance between \( x \) and \( x_i \) based on the predictive CDF conditional on \( x_{1:i-1} \).

To better understand the importance of the data-dependent bandwidth, we compare the conditional predictive mean of R-BP and AR-BP in the bivariate setting \( X \times Y \). Under the simplifying assumption of Gaussian predictive densities, we show in Supplement A.3 that the conditional mean of \( Y \mid X \) is given by

\[
\mu_i(x) = \frac{\alpha_i \alpha_i c(P_{i-1}(x), P_{i-1}(x); \rho)}{1 - \alpha_i + \alpha_i c(P_{i-1}(x), P_{i-1}(x); \rho)}.
\]

Note that \( \rho(x, x_i) = \rho_0 \) for R-BP. Intuitively, the updated mean is the previous mean plus a residual term at \( y_i \) scaled by some notion of distance between \( x \) and \( x_i \). For R-BP, this distance between \( x \) and \( x_i \) depends only on their predictive CDF values through \( \alpha_i(x, x_i) \), as a function of \( x \), is not centred at \( x_i \). Counterintuitively, there is thus an \( x > x_i \) where \( \mu_i(x) \) is updated more than at the actual observed \( x = x_i \). This follows from the lack of focus on conditional density estimates for R-BP, which is alleviated by AR-BP. In the AR case, \( \rho(x, x_i) \) takes into account the Euclidean distance between \( x \) and \( x_i \) in the data space. We see in the lower plot in Figure 3(a) that the peak of \( \alpha_i(x, x_i) \) is closer to \( x_i \). Figure 3(b) further demonstrates this difference on another toy example - we see that R-BP struggles to fit a linear conditional mean function for \( n = 4 \), focussing density in data sparse regions, while AR-BP succeeds to assign significant density only to points on the data manifold.

### Training the update parameters

In order to compute the predictive density \( p_\ell(x^*) \), we require the vector of conditional CDFs \( [v_{1:1}^i, \ldots, v_{n-1}^i] \) where \( v_i^j = P_i(x_{j+1}^i \mid x_{i+1}^{j-1} \)\]. Given a bandwidth parameterization, obtaining this vector thus amounts to model-fitting, and each \( v_i^j \) requires \( i - 1 \) iterations (Supplement B.3), for \( i \in \{1, \ldots, n\} \). We note that the order of samples and dimensions influences the prediction performance in AR density estimators [Vinyals et al., 2015]. In practice, averaging over different permutations of these improves performance (Supplement B.3). Full implementation details can be found in Supplement B.

#### Computational complexity

The above procedure results in a computational complexity of \( O(Mn^2) \) at the training stage where \( M \) is the number of permutations. At test time, we have already obtained the necessary conditional prequential CDFs \( v_n^j \) in computing the prequential log-likelihood above. As a result, we have a computational complexity \( O(Mn) \) for each test observation. Note that the introduction of a data-dependent bandwidth does not increase the computational complexity at train or test time relative to R-BP and only adds a negligible factor to the computational time for the calculation of the bandwidth.

### 3.3 BANDWIDTH PARAMETRISATION

The choice of covariance function in [7] provides substantial modelling flexibility in our AR-BP framework. Moreover, the additional parameters associated with the covariance function allow us to tune the implied covariance structure according to the observed data. This formulation enables us to draw upon the rich literature on the choice of covariance functions for Gaussian processes [Williams and Rasmussen, 2006]. For simplicity we only consider the most popular such choice here, but study the more flexible rational-quadratic covariance in Supplement C.2. The radial basis function (RBF) covariance function is defined as \( k_i(x_1:j, x_1:j) = \exp[-\sum_{k=1}^{j-1}((x^k - x')/(\ell^k))^2] \), where \( \ell \in \mathbb{R}_{>0}^d \) is the length scale.

#### Neural parameterisation

As we saw in the motivating example of the density estimation of a chessboard distribution in Figure 1, the RBF kernel can restrict the capacity of the predictive density update to capture intricate nonlinearities if the training data size is not sufficient. While the parameterization of the bandwidth in [7] was initially derived via the first predictive update for a DPMM all we require is that the bandwidth function \( \rho^j : \mathbb{R}^{j-1} \times \mathbb{R}^{j-1} \rightarrow \mathbb{R} \) lies in \((0,1)\). We would also like \( \rho^j(x_1:j-1, x_1:j-1) \) to take larger values when \( x_1:j-1 \) and \( x_1:j-1 \) are ‘close’ in some sense. Motivated by this observation, we now consider more expressive bandwidth functions that can lead to increased predictive performance. In particular, we formulate an AR neural network \( f_w : \mathbb{R}^d \rightarrow \mathbb{R}^d \) for \( d' \in \mathbb{N} \) with the property that the \( j \)th row of the output depends only on the first \( j - 1 \) dimensions of the input. Let \( Z = f_w(x) \) and denoting \( z^j \) to be the \( j \)th row of the matrix \( Z \), the covariance function is then computed as \( \rho^j(x_1:j-1, x_1:j-1) = \rho_0 \exp(-\sum_{k=1}^{j-1}\|z^k - z'\|_2^2) \).

Numerous AR neural network models have been extensively used for density estimation [Dinh et al., 2014; Huang et al., 2018; Kingma et al., 2016]. In our experiments, we use a relatively simple model with parameter
sharing inspired by NADE, an AR neural network designed for density estimation [Larochelle and Murray, 2011]. More advanced properties like the permutation invariance of MADE [Papamakarios et al., 2017] create an additional overhead that cannot be used in the copula formulation as the predictive update is not permutation-invariant. We refer to Bayesian predictive densities estimated using AR neural networks as ARnet Bayesian predicatives (ARnet-BP).

Tuning the bandwidth function Recall that the bandwidths ρ_i(·, ·) are parameterised by ρ_0 and the parameters of the chosen covariance functions or neural embedders. For AR-BP, these are the length scales ℓ of the RBF covariance function, while for ARnet-BP, these are the parameters w of the AR neural network. We fit these tunable parameters in a data-driven approach by maximising the prequential [Dawid, 1997] log-likelihood \( \sum_{i=1}^{n} \log p_i(x_i) \) which is analogous to the Bayesian marginal likelihood – the tractable predictive density allows us to compute this exactly, and this approach is analogous to empirical Bayes. Specifically, we use gradient descent optimisation with Adam, sampling a different random permutation of the training data at each optimisation step (Supplement B.3).

4 RELATED WORK

Our work falls into the broad area of multivariate density estimation [Scott, 2015]. While AR networks have been previously used directly for the task of density estimation [Bengio and Bengio, 1999, Germain et al., 2015, Larochelle and Murray, 2011], we use them to elicit a data-dependent bandwidth in the predictive update to mitigate the smoothing effect observed in AR-BP. Neural network based approaches, however, often underperform in small-data regimes. Deep learning approaches that do target few-shot density estimation require complex meta-learning and pre-training pipelines [Gu et al., 2020, Reed et al., 2017].

Our work directly extends the contributions of Hahn et al. [2018] and Fong et al. [2021] through an alternative specification of the nonparametric Bayesian model in the recursive predictive update scheme. R-BP has recently been used for nonparametric solvency risk prediction [Hong and Martin, 2019], and survival analysis [Fong and Lehmann, 2022, Berti et al., 2021a,b, 2004] also focus on univariate predictive updates in the Bayesian nonparametric paradigm, specifically exploring the use of the conditionally identically distributed condition as a relaxation of the standard exchangeability assumption. Other studies have investigated quasi-Bayesian updates in the special case of the mixing distribution in nonparametric mixture models [Dixit and Martin, 2022, Fortini and Petrone, 2020, Martin, 2018, Tokdar et al., 2009], though these typically focus on univariate or low-dimensional spaces. See also Martin [2021] for a survey.

Finally, copulas are a well-studied tool for modelling the correlations in multivariate data (see e.g. Kauermann et al. [2013], Ling et al. [2020], Nelsen [2007]). Copula density estimation aims to construct density estimates whose univariate marginals are uniform [Gijbels and Mielniczuk, 1990], and often focus on modelling strong tail dependencies [Wiese et al., 2019]. In contrast, we employ bivariate copulas for generic multivariate density estimation as a tool to model the correlations between subsequent subjective predictive densities, rather than across the data dimensions directly.

5 EXPERIMENTS

We demonstrate the benefits of AR-BP, ARd-BP and ARnet-BP for density estimation and prediction tasks in an experimental study with five baseline approaches.
Table 1: Average NLL with standard error over five runs on data sets analysed by Fong et al. [2021].

<table>
<thead>
<tr>
<th>n/d</th>
<th>WINE</th>
<th>BREAST</th>
<th>PARKIN</th>
<th>IONO</th>
<th>BOSTON</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>89/12</td>
<td>97/14</td>
<td>97/16</td>
<td>175/30</td>
<td>506/13</td>
</tr>
<tr>
<td>KDE</td>
<td>13.69±0.00</td>
<td>10.45±0.24</td>
<td>12.83±0.27</td>
<td>32.09±0.00</td>
<td>8.34±0.00</td>
</tr>
<tr>
<td>DPPM (Diag)</td>
<td>17.46±0.6</td>
<td>16.26±0.71</td>
<td>22.28±0.66</td>
<td>35.30±1.28</td>
<td>7.64±0.99</td>
</tr>
<tr>
<td>DPPM (Full)</td>
<td>32.88±0.82</td>
<td>26.67±1.32</td>
<td>39.95±1.56</td>
<td>86.18±10.22</td>
<td>9.45±0.43</td>
</tr>
<tr>
<td>MAF</td>
<td>39.60±1.41</td>
<td>10.13±0.40</td>
<td>11.76±0.45</td>
<td>140.09±4.03</td>
<td>56.01±27.74</td>
</tr>
<tr>
<td>RQ-NSF</td>
<td>38.34±6.63</td>
<td>26.41±5.57</td>
<td>31.26±0.31</td>
<td>54.49±2.65</td>
<td>2.20±1.11</td>
</tr>
<tr>
<td>R-BP</td>
<td>13.57±0.04</td>
<td>7.49±0.02</td>
<td>9.15±0.04</td>
<td>21.15±0.24</td>
<td>4.56±0.04</td>
</tr>
<tr>
<td>R2-BP</td>
<td>13.32±0.01</td>
<td>6.12±0.05</td>
<td>7.52±0.05</td>
<td>19.82±0.08</td>
<td>13.50±0.59</td>
</tr>
<tr>
<td>AR-BP</td>
<td>13.22±0.04</td>
<td>6.11±0.04</td>
<td>7.21±0.12</td>
<td>16.48±0.26</td>
<td>−14.75±0.89</td>
</tr>
<tr>
<td>ARnet-BP</td>
<td>14.41±0.11</td>
<td>6.87±0.23</td>
<td>8.29±0.17</td>
<td>15.32±0.35</td>
<td>−5.71±0.62</td>
</tr>
</tbody>
</table>

Figure 4: Average NLL and standard errors over 10 runs for training sets of different size. Our models outperform neural methods for data sets up to 10,000 samples.

and 13 different data sets. The code and data is available at [https://github.com/shaebikesabi/autoregressive-bayesian-predictives](https://github.com/shaebikesabi/autoregressive-bayesian-predictives) See Supplement C for additional experimental details and results, including a sensitivity study, an ablation study, further illustrative examples, a preliminary investigation into image examples, and an empirical study of the computational complexity of the proposed methods.

5.1 DENSITY ESTIMATION

We compared our models against KDEs [Parzen, 1962, DPMs [Rasmussen, 1999], MAFs [Papamakarios et al., 2017] and RQ-NSFs [Durkan et al., 2019]. The hyperparameters of the baselines were tuned with cross-validation. Unless otherwise specified, we use respectively 10 permutations over samples and features to average the quasi-Bayesian estimates. We did not see substantial improvements with more permutations. We use the same few hyperparameters (initialisation of $\rho_0, l_1, \ldots, l_d$, number of permutations, neural network architecture, and learning rate) on all data sets as our method is robust to their choice. See Supplement C.1 for further information.

Data sets analysed by [Fong et al., 2021] See Table 1 for the negative log-likelihood (NLL) estimated on five UCI data sets [Asuncion and Newman, 2007] of small size with up to 506 samples, as investigated by [Fong et al., 2021]. Our proposed methods display highly competitive performance: AR2-BP achieved the best test NLL on four of the data sets, while ARnet-BP prevailed on ionosphere.

Data sets analysed by [Papamakarios et al., 2017] A number of UCI data sets have become the standard evaluation benchmark for deep AR models [Durkan et al., 2019, Huang et al., 2018, Papamakarios et al., 2017]. These include low-dimensional data sets with up to 63 features, but at least 29,000 with up to $10^6$ samples. In many circumstances, data sets of such a data size are not available. To investigate performance as a function of sample size, we trained the models on subsets of the full data set. We do not report results for the KDEs and the DPPM estimators here as these estimators performed significantly worse than the other approaches. Similarly, we do not report deep learning results for sample sizes smaller than $10^2$. See Supplement C.2 for complete results.

In the small-data regime, we observe that the R-BP methods significantly outperform the neural density estimators (Figure 4). As the sample size increases, the gap in performance decreases until eventually the neural density estimators outcompete the R-BP methods. The performance between the R-BP methods and our proposed AR extensions is largely similar, though we note that the AR-BP methods were generally more effective on the GAS dataset.

5.2 SUPERVISED LEARNING

R-BP methods, including AR-BP, can be used for prediction tasks such as regression and classification.
We expand upon the tractable recursive copula updates. Although Bayesian methods generally perform well in AR with little computational overhead compared to R-BP. This introduces a data-dependent bandwidth, thus increasing the flexibility of this class of models, with little computational overhead compared to R-BP. More generally, it would be of interest to integrate other machine learning methods with recursive copula updates. Furthermore, other Bayesian nonparametric models may inspire other recursive copula updates—see Appendix A.2 for an example based on GPs.

Table 2: Average NLL over five runs reported with standard error for supervised tasks

<table>
<thead>
<tr>
<th>Method</th>
<th>NLL (± std)</th>
<th>NLL (± std)</th>
<th>NLL (± std)</th>
<th>NLL (± std)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>0.37 ± 0.02</td>
<td>0.37 ± 0.02</td>
<td>0.37 ± 0.02</td>
<td>0.37 ± 0.02</td>
</tr>
<tr>
<td>GP</td>
<td>0.42 ± 0.02</td>
<td>0.42 ± 0.02</td>
<td>0.42 ± 0.02</td>
<td>0.42 ± 0.02</td>
</tr>
<tr>
<td>MLP</td>
<td>0.37 ± 0.02</td>
<td>0.37 ± 0.02</td>
<td>0.37 ± 0.02</td>
<td>0.37 ± 0.02</td>
</tr>
<tr>
<td>R-BP</td>
<td>0.42 ± 0.02</td>
<td>0.42 ± 0.02</td>
<td>0.42 ± 0.02</td>
<td>0.42 ± 0.02</td>
</tr>
<tr>
<td>ARnet-BP</td>
<td>0.45 ± 0.11</td>
<td>0.45 ± 0.11</td>
<td>0.45 ± 0.11</td>
<td>0.45 ± 0.11</td>
</tr>
</tbody>
</table>

An appealing feature of AR-BP is that it requires no manual hyperparameter tuning. Further, on small data sets, AR-BP shows state-of-the-art generalization and is faster than competing deep learning models. It significantly increases the modelling capacity of the baseline R-BP via a data-dependent bandwidth. Additionally, ARnet-BP provides a useful illustration of how powerful neural network models can be incorporated into R-BP methods to improve density estimation. Future work can investigate alternative architectures for structured data. Our approach adds to the rich body of density estimators and thus we do not anticipate any additional negative societal impact arising from our proposal.

This strong performance of AR-BP (and other copula methods) in the small data regime is likely due to its Bayesian-like regularization towards an initial density $p_0$, as shown in the weighted sum in (8). Its weaker performance in the large data regime may be due to the importance of the sequence $\alpha_l$ which governs how regularization decays, but further theoretical work is needed to understand AR-BP’s asymptotic behaviour.

A limitation of R-BP methods, including AR-BP, is that it requires no manual hyperparameter tuning. Further, on small data sets, AR-BP shows state-of-the-art generalization and is faster than competing deep learning models. It significantly increases the modelling capacity of the baseline R-BP via a data-dependent bandwidth. Additionally, ARnet-BP provides a useful illustration of how powerful neural network models can be incorporated into R-BP methods to improve density estimation. Future work can investigate alternative architectures for structured data. Our approach adds to the rich body of density estimators and thus we do not anticipate any additional negative societal impact arising from our proposal.

This strong performance of AR-BP (and other copula methods) in the small data regime is likely due to its Bayesian-like regularization towards an initial density $p_0$, as shown in the weighted sum in (8). Its weaker performance in the large data regime may be due to the importance of the sequence $\alpha_l$ which governs how regularization decays, but further theoretical work is needed to understand AR-BP’s asymptotic behaviour.

A limitation of R-BP methods, including AR-BP, is that it requires no manual hyperparameter tuning. Further, on small data sets, AR-BP shows state-of-the-art generalization and is faster than competing deep learning models. It significantly increases the modelling capacity of the baseline R-BP via a data-dependent bandwidth. Additionally, ARnet-BP provides a useful illustration of how powerful neural network models can be incorporated into R-BP methods to improve density estimation. Future work can investigate alternative architectures for structured data. Our approach adds to the rich body of density estimators and thus we do not anticipate any additional negative societal impact arising from our proposal.

This strong performance of AR-BP (and other copula methods) in the small data regime is likely due to its Bayesian-like regularization towards an initial density $p_0$, as shown in the weighted sum in (8). Its weaker performance in the large data regime may be due to the importance of the sequence $\alpha_l$ which governs how regularization decays, but further theoretical work is needed to understand AR-BP’s asymptotic behaviour.
lutional kernels [Van der Wilk et al., 2017] would be particularly suited for computer vision tasks.

Acknowledgements

SG is a student of the EPSRC CDT in Modern Statistics and Statistical Machine Learning (EP/S023151/1) and receives funding from the Oxford Radcliffe Scholarship and Novartis. CH is supported by The Alan Turing Institute, Health Data Research UK, the Medical Research Council UK, the EPSRC through the Bayes4Health programme Grant EP/R018561/1, and AI for Science and Government UK Research and Innovation (UKRI). EF is supported by Novo Nordisk. BL is supported by the UK Engineering and Physical Sciences Research Council through the Bayes4Health programme (grant number EP/R018561/1) and gratefully acknowledges funding from Jesus College, Oxford.

REFERENCES


