

# Quasi-Bayes meets Vines

David Huk, Yuanhe Zhang, Mark Steel and Rito Dutta

Algorithms and Computationally Intensive Inference Seminar

28/06/2024

Our work proposes a method for nonparametric density estimation with sampling that is suited for **high dimensions**, a central issue in probabilistic modelling.

Our work proposes a method for nonparametric density estimation with sampling that is suited for **high dimensions**, a central issue in probabilistic modelling.

Two categories of such models exist:

## Analytical expression

- Kernel Density Estimation [Chen, 2017]
- Dirichlet Process Mixture Models [Hjort et al., 2010]
- Normalising Flows [Rezende and Mohamed, 2015]

## Non-analytical expression

- Variational Auto-Encoders [Kingma and Welling, 2013]
- Generative Adversarial Networks [Goodfellow et al., 2014]
- Diffusion Models [Song et al., 2021]

# A Bayesian nonparametric model: DPMM

The Dirichlet Process Mixture Model (DPMM) can be interpreted as an infinite Gaussian mixture model.

The Dirichlet Process Mixture Model (DPMM) can be interpreted as an infinite Gaussian mixture model.

$$f(x, g) = \int \mathcal{N}(x | \theta) g(\theta) d\theta,$$

where  $g$  is a Dirichlet process prior  $\text{DP}(c, G_0)$  equipped with a base measure  $G_0$  as  $\mathcal{N}(0, \tau^{-1})$  and precision parameter  $c > 0$ .

The Dirichlet Process Mixture Model (DPMM) can be interpreted as an infinite Gaussian mixture model.

$$f(x, g) = \int \mathcal{N}(x | \theta) g(\theta) d\theta,$$

where  $g$  is a Dirichlet process prior  $\text{DP}(c, G_0)$  equipped with a base measure  $G_0$  as  $\mathcal{N}(0, \tau^{-1})$  and precision parameter  $c > 0$ .

- × Inference is reliant on MCMC, which is especially costly for updates to predictive densities  $p_n$  to  $p_{n+1}$ :

$$p_n(x|x_{1:n}) = \frac{\int f(x|g) \cdot f(x_n|g) \cdot \pi_{n-1}(g|x_{1:n-1}) dg}{p_{n-1}(x_n|x_{1:n-1})}.$$

# A Bayesian nonparametric model: DPMM

The Dirichlet Process Mixture Model (DPMM) can be interpreted as an infinite Gaussian mixture model.

$$f(x, g) = \int \mathcal{N}(x | \theta) g(\theta) d\theta,$$

where  $g$  is a Dirichlet process prior  $\text{DP}(c, G_0)$  equipped with a base measure  $G_0$  as  $\mathcal{N}(0, \tau^{-1})$  and precision parameter  $c > 0$ .

- × Inference is reliant on MCMC, which is especially costly for updates to predictive densities  $p_n$  to  $p_{n+1}$ :

$$p_n(x|x_{1:n}) = \frac{\int f(x|g) \cdot f(x_n|g) \cdot \pi_{n-1}(g|x_{1:n-1}) dg}{p_{n-1}(x_n|x_{1:n-1})}.$$

→ Need a **faster** update!

Newton's method [Newton et al., 1998] provides a solution to modelling DPMM's predictive densities with a recursive approach.

For a mixture density  $f(x, g) = \int k(x | \theta)g(\theta)d\theta$ , the Predictive Recursion (PR) estimates the mixing density  $g$  by starting with an initial guess  $g_0$  and recursively updating it as:



Newton's method [Newton et al., 1998] provides a solution to modelling DPMM's predictive densities with a recursive approach.

For a mixture density  $f(x, g) = \int k(x | \theta)g(\theta)d\theta$ , the Predictive Recursion (PR) estimates the mixing density  $g$  by starting with an initial guess  $g_0$  and recursively updating it as:

$$g_i(\theta) = (1 - \alpha_i) \cdot g_{i-1}(\theta) + \alpha_i \cdot \frac{k(x_i | \theta) g_{i-1}(\theta)}{\int_{\Theta} k(x_i | z) g_{i-1}(z) \mu(dz)}$$

where

- $x_1, x_2, \dots, x_i$  are a sequence of observed data.

Newton's method [Newton et al., 1998] provides a solution to modelling DPMM's predictive densities with a recursive approach.

For a mixture density  $f(x, g) = \int k(x | \theta)g(\theta)d\theta$ , the Predictive Recursion (PR) estimates the mixing density  $g$  by starting with an initial guess  $g_0$  and recursively updating it as:

$$g_i(\theta) = (1 - \alpha_i) \cdot g_{i-1}(\theta) + \alpha_i \cdot \frac{k(x_i | \theta) g_{i-1}(\theta)}{\int_{\Theta} k(x_i | z) g_{i-1}(z) \mu(dz)}$$

where

- $x_1, x_2, \dots, x_i$  are a sequence of observed data.
- $k$  is the mixing kernel.

Newton's method [Newton et al., 1998] provides a solution to modelling DPMM's predictive densities with a recursive approach.

For a mixture density  $f(x, g) = \int k(x | \theta)g(\theta)d\theta$ , the Predictive Recursion (PR) estimates the mixing density  $g$  by starting with an initial guess  $g_0$  and recursively updating it as:

$$g_i(\theta) = (1 - \alpha_i) \cdot g_{i-1}(\theta) + \alpha_i \cdot \frac{k(x_i | \theta) g_{i-1}(\theta)}{\int_{\Theta} k(x_i | z) g_{i-1}(z) \mu(dz)}$$

where

- $x_1, x_2, \dots, x_i$  are a sequence of observed data.
- $k$  is the mixing kernel.
- $\alpha_i \in [0, 1]$  are deterministic weights such that  $\sum_{i=1}^{\infty} \alpha_i = \infty$ , and  $\sum_{i=1}^{\infty} \alpha_i^2 < \infty$ .

Newton's method [Newton et al., 1998] provides a solution to modelling DPMM's predictive densities with a recursive approach.

For a mixture density  $f(x, g) = \int k(x | \theta)g(\theta)d\theta$ , the Predictive Recursion (PR) estimates the mixing density  $g$  by starting with an initial guess  $g_0$  and recursively updating it as:

$$g_i(\theta) = (1 - \alpha_i) \cdot g_{i-1}(\theta) + \alpha_i \cdot \frac{k(x_i | \theta) g_{i-1}(\theta)}{\int_{\Theta} k(x_i | z) g_{i-1}(z) \mu(dz)}$$

where

- $x_1, x_2, \dots, x_i$  are a sequence of observed data.
- $k$  is the mixing kernel.
- $\alpha_i \in [0, 1]$  are deterministic weights such that  $\sum_{i=1}^{\infty} \alpha_i = \infty$ , and  $\sum_{i=1}^{\infty} \alpha_i^2 < \infty$ .

Then, recover the DPMM density as  $f_i(x, g) = \int k(x | \theta)g_i(\theta)d\theta$ .

The PR density estimator is **Quasi-Bayesian**, as it no longer respects Bayes updates, and instead targets the DPMM predictive mean.

Ghosh, Tokdar and Martin publish a suite of papers on the PR analysing the convergence of the stochastic approximation of the mixing density to the true mixture under various settings: [Ghosh and Tokdar, 2006, Martin and Ghosh, 2008, Martin and Tokdar, 2009, Tokdar et al., 2009, Martin, 2012, Ghosal and Van der Vaart, 2017, Martin, 2021].

The PR remains limited in practice to **at most 3 dimensional**  $\Theta$  due to the normalising constant at every step  $\int_{\Theta} k(x_i | z) g_{i-1}(z) \mu(dz)$  having no elegant solution.

In [Dixit and Martin, 2023], the **PRticle Filter** is proposed as a solution to extend the PR to multiple dimensions. The recursion is adapted to support a sequential Importance Sampling (IS) approach, reweighting a batch of samples to approximate the normalising constant. But this is still not well-equipped for high dimensions due to inherent IS drawbacks...

# Predictive Recursion: the PRticle Filter

In [Dixit and Martin, 2023], the **PRticle Filter** is proposed as a solution to extend the PR to multiple dimensions. The recursion is adapted to support a sequential Importance Sampling (IS) approach, reweighting a batch of samples to approximate the normalising constant. But this is still not well-equipped for high dimensions due to inherent IS drawbacks...

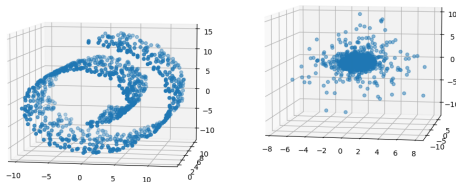


Figure: Example samples of the PRticle Filter in practice with 6 parameters.

→ Currently, the PR approach is not equipped to handle high dimensions.

Go back to the Bayesian predictive density for  $x \in \mathbb{R}$ :

$$p^{(n)}(x|x^{1:n}) = \frac{\int f(x|\theta) \cdot f(x^n|\theta) \cdot \pi^{(n-1)}(\theta|x^{1:n-1}) d\theta}{p^{(n-1)}(x^n|x^{1:n-1})}.$$



Go back to the Bayesian predictive density for  $x \in \mathbb{R}$ :

$$p^{(n)}(x|x^{1:n}) = \frac{\int f(x|\theta) \cdot f(x^n|\theta) \cdot \pi^{(n-1)}(\theta|x^{1:n-1}) d\theta}{p^{(n-1)}(x^n|x^{1:n-1})}.$$

and multiply both sides of the fraction by the predictive from the previous step  $p^{(n-1)}$ :

$$p^{(n)}(x|x^{1:n}) = p^{(n-1)}(x|x^{1:n-1}) \cdot \frac{\int f(x|\theta) \cdot f(x^n|\theta) \cdot \pi^{(n-1)}(\theta|x^{1:n-1}) d\theta}{p^{(n-1)}(x^n|x^{1:n-1}) \cdot p^{(n-1)}(x|x^{1:n-1})}$$

In [Hahn et al., 2018], it is revealed that a 1D Bayesian predictive corresponds to a sequence of copula updates:

$$p^{(n)}(x|x^{1:n}) = p^{(n-1)}(x|x^{1:n-1}) \cdot \frac{\overbrace{\int f(x|\theta) \cdot f(x^n|\theta) \cdot \pi^{(n-1)}(\theta|x^{1:n-1}) d\theta}^{\text{Joint density for } x, x^n}}{\underbrace{p^{(n-1)}(x^n|x^{1:n-1})}_{\text{Marginal for } x^n} \cdot \underbrace{p^{(n-1)}(x|x^{1:n-1})}_{\text{Marginal for } x}}$$

$$p^{(n)}(x|x^{1:n}) = p^{(n-1)}(x|x^{1:n-1}) \cdot c^{(n)}(P_{n-1}(x), P_{n-1}(x^n))$$

where  $c^{(n)}$  is the copula for step  $n$ , with  $c^{(n)} \rightarrow 1$  as  $n \rightarrow \infty$ .

$$p^{(n)}(x|x^{1:n}) = p^{(n-1)}(x|x^{1:n-1}) \cdot c^{(n)}(P_{n-1}(x), P_{n-1}(x^n))$$

### Advantages:

- ✓ Bayesian predictive updates without any MCMC!

$$p^{(n)}(x|x^{1:n}) = p^{(n-1)}(x|x^{1:n-1}) \cdot c^{(n)}(P_{n-1}(x), P_{n-1}(x^n))$$

### Advantages:

- ✓ Bayesian predictive updates without any MCMC!

### Disadvantages:

- ✗ Copula form is not known in general.

$$p^{(n)}(x|x^{1:n}) = p^{(n-1)}(x|x^{1:n-1}) \cdot c^{(n)}(P_{n-1}(x), P_{n-1}(x^n))$$

### Advantages:

- ✓ Bayesian predictive updates without any MCMC!

### Disadvantages:

- ✗ Copula form is not known in general.
- ✗ Copula interpretation only holds in 1D.

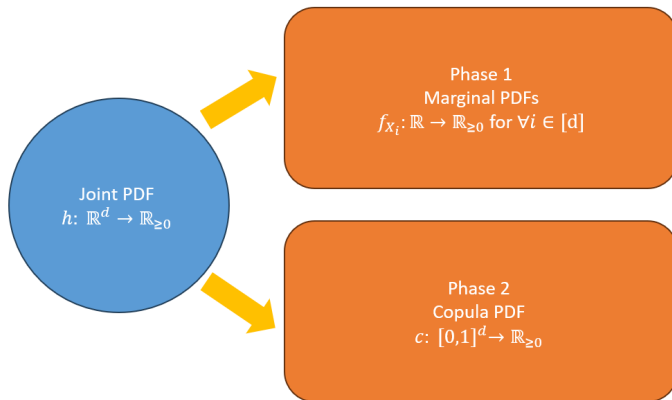
→ Need an *interperable* extensions to *high dimensions*.

**Can we extend the Recursive Bayesian Predictive to high-dimensions?**

$$p^{(n)}(\mathbf{x}|\mathbf{x}^{1:n}) = p^{(n-1)}(\mathbf{x}|\mathbf{x}^{1:n-1}) \cdot \gamma^{(n)}(\mathbf{x}|\mathbf{x}^{1:n})$$

# Quasi-Bayesian Vine: A two-stage estimation

By **Sklar's theorem**, we can divide a single task into multiple sub-tasks.



Our objective:  $\hat{f}(x^1, \dots, x^d) = \prod_{i=1}^d \hat{f}_{X_i}(x^i) \cdot \hat{c}(F_{X_1}(x^1), \dots, F_{X_d}(x^d))$

In the case of independent data  $x_1, x_2$  we have that their joint distribution factorises as:

$$f(x_1, x_2) = f_1(x_1) \cdot f_2(x_2)$$



In the case of independent data  $x_1, x_2$  we have that their joint distribution factorises as:

$$f(x_1, x_2) = f_1(x_1) \cdot f_2(x_2)$$

Now assume that  $x_1, x_2$  are not independent. We then have:

$$f(x_1, x_2) = f_1(x_1) \cdot f_2(x_2) \cdot$$

In the case of independent data  $x_1, x_2$  we have that their joint distribution factorises as:

$$f(x_1, x_2) = f_1(x_1) \cdot f_2(x_2)$$

Now assume that  $x_1, x_2$  are not independent. We then have:

$$f(x_1, x_2) = f_1(x_1) \cdot f_2(x_2) \cdot c(x_1, x_2)$$

meaning

$$c(x_1, x_2) = \frac{f(x_1, x_2)}{f_1(x_1) \cdot f_2(x_2)}$$

The function  $c$  is precisely a copula. It provides a notion of dependence, adding what is missing in the independent case.

## Theorem (Sklar for predictive densities)

Let  $\mathbf{P}^{(n)}$  be an  $d$ -dimensional predictive distribution function with continuous marginal distributions  $P_1^{(n)}, P_2^{(n)}, \dots, P_d^{(n)}$ . Then there exists a copula distribution  $\mathbf{C}^{(n)}$  such that for all  $\mathbf{x} = (x_1, x_2, \dots, x_d) \in \mathbb{R}^d$ :

$$\mathbf{P}^{(n)}(x_1, \dots, x_d) = \mathbf{C}^{(n)}(P_1^{(n)}(x_1), \dots, P_d^{(n)}(x_d)) \quad (1)$$

And if a probability density function (pdf) is available:

$$\mathbf{p}^{(n)}(x_1, \dots, x_d) = p_1^{(n)}(x_1) \cdot \dots \cdot p_d^{(n)}(x_d) \cdot \mathbf{c}^{(n)}(P_1^{(n)}(x_1), \dots, P_d^{(n)}(x_d)) \quad (2)$$

where  $p_1^{(n)}(x_1), \dots, p_d^{(n)}(x_d)$  are the marginal predictive probability density functions (pdf), and  $\mathbf{c}^{(n)} : [0, 1]^d \rightarrow \mathbb{R}$  is the copula pdf.

We use Sklar on the joint predictive density:

$$\mathbf{p}^{(n)}(x_1, \dots, x_d) = \prod_{i=1}^d \left\{ p_i^{(n)}(x_i) \right\} \cdot \mathbf{c}^{(n)}(P_1^{(n)}(x_1), \dots, P_d^{(n)}(x_d)).$$

We use Sklar on the joint predictive density:

$$\mathbf{p}^{(n)}(x_1, \dots, x_d) = \prod_{i=1}^d \left\{ p_i^{(n)}(x_i) \right\} \cdot \mathbf{c}^{(n)}(P_1^{(n)}(x_1), \dots, P_d^{(n)}(x_d)).$$

Doing this for  $\mathbf{p}^{(n)}$  and  $\mathbf{p}^{(n-1)}$ , we get a recursive relationship:

## Strategy: Side-stepping the high dimensional recursion

We use Sklar on the joint predictive density:

$$\mathbf{p}^{(n)}(x_1, \dots, x_d) = \prod_{i=1}^d \{p_i^{(n)}(x_i)\} \cdot \mathbf{c}^{(n)}(P_1^{(n)}(x_1), \dots, P_d^{(n)}(x_d)).$$

Doing this for  $\mathbf{p}^{(n)}$  and  $\mathbf{p}^{(n-1)}$ , we get a recursive relationship:

$$\frac{\mathbf{p}^{(n)}}{\mathbf{p}^{(n-1)}} = \underbrace{\prod_{i=1}^d \left\{ \frac{p_i^{(n)}}{p_i^{(n-1)}} \right\}}_{\text{Independent recursions}} \cdot \underbrace{\frac{\mathbf{c}^{(n)}(P_1^{(n)}(x_1), \dots, P_d^{(n)}(x_d))}{\mathbf{c}^{(n-1)}(P_1^{(n-1)}(x_1), \dots, P_d^{(n-1)}(x_d))}}_{\text{Implicit recursion on copulas}}. \quad (3)$$

## Strategy: Side-stepping the high dimensional recursion

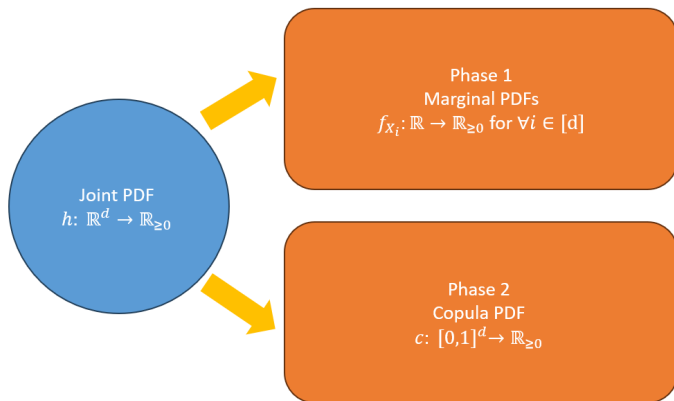
We use Sklar on the joint predictive density:

$$\mathbf{p}^{(n)}(x_1, \dots, x_d) = \prod_{i=1}^d \left\{ p_i^{(n)}(x_i) \right\} \cdot \mathbf{c}^{(n)}(P_1^{(n)}(x_1), \dots, P_d^{(n)}(x_d)).$$

Doing this for  $\mathbf{p}^{(n)}$  and  $\mathbf{p}^{(n-1)}$ , we get a recursive relationship:

$$\frac{\mathbf{p}^{(n)}}{\mathbf{p}^{(n-1)}} = \underbrace{\prod_{i=1}^d \left\{ \frac{p_i^{(n)}}{p_i^{(n-1)}} \right\}}_{\text{Independent recursions}} \cdot \underbrace{\frac{\mathbf{c}^{(n)}(P_1^{(n)}(x_1), \dots, P_d^{(n)}(x_d))}{\mathbf{c}^{(n-1)}(P_1^{(n-1)}(x_1), \dots, P_d^{(n-1)}(x_d))}}_{\text{Implicit recursion on copulas}}. \quad (3)$$

→ If we are only interested in the final step  $\mathbf{p}^{(n)}$ , we **only need to recurse on marginals**, leaving the recursion on copulas implicit, and **fit a single copula at the last step  $n$** .





Consider an approximation of the DPMM called **Recursive Bayesian Predictive (R-BP)** proposed in [Hahn et al., 2018]:

Consider an approximation of the DPMM called **Recursive Bayesian Predictive (R-BP)** proposed in [Hahn et al., 2018]:

$$p_{n+1}(\cdot) = (1 - \alpha_n) \cdot p_n(\cdot) + \alpha_n \cdot c_\rho(\mathbb{P}_n(\cdot), \mathbb{P}_n(x_n)) \cdot p_n(\cdot)$$

$$\mathbb{P}_{n+1}(\cdot) = (1 - \alpha_n) \cdot \mathbb{P}_n(\cdot) + \alpha_n \cdot H_\rho(\mathbb{P}_n(\cdot) | \mathbb{P}_n(x_n)) \cdot$$

where

- $p_{n+1}(X) = p(X|x_{1:n})$  be  $X$ 's  $n + 1^{\text{th}}$  predictive probability density function (pdf)
- $\mathbb{P}_{n+1}$  be the corresponding predictive distribution function (cdf)
- $c_\rho$  be the bivariate Gaussian copula pdf
- $H_\rho(\cdot | \cdot)$  be the associated conditional cdf of  $c_\rho$
- $(\alpha_n)_{n \geq 1}$  be a sequence of weights decreasing in  $n$
- $\rho$  is the correlation parameter for Gaussian copula and the **only** free (hyper)parameter

$$p_{n+1}(\cdot) = (1 - \alpha_n) \cdot p_n(\cdot) + \alpha_n \cdot c_\rho(\mathbb{P}_n(\cdot), \mathbb{P}_n(x_n)) \cdot p_n(\cdot)$$

$$\mathbb{P}_{n+1}(\cdot) = (1 - \alpha_n) \cdot \mathbb{P}_n(\cdot) + \alpha_n \cdot H_\rho(\mathbb{P}_n(\cdot) | \mathbb{P}_n(x_n)).$$

## Advantages:

- ✓ Nonparametric
- ✓ Quasi-Bayesian
- ✓ Very fast density evaluation and sampling
- ✓ Easily parallelisable across dimensions  $d$ .

$$p_{n+1}(\cdot) = (1 - \alpha_n) \cdot p_n(\cdot) + \alpha_n \cdot c_\rho(\mathbb{P}_n(\cdot), \mathbb{P}_n(x_n)) \cdot p_n(\cdot)$$

$$\mathbb{P}_{n+1}(\cdot) = (1 - \alpha_n) \cdot \mathbb{P}_n(\cdot) + \alpha_n \cdot H_\rho(\mathbb{P}_n(\cdot) | \mathbb{P}_n(x_n)).$$

## Advantages:

- ✓ Nonparametric
- ✓ Quasi-Bayesian
- ✓ Very fast density evaluation and sampling
- ✓ Easily parallelisable across dimensions  $d$ .

## Disadvantages:

- ✗ Selecting hyperparameter  $\rho$  is not obvious

The robust estimation of simulation-based models has been studied in [Pacchiardi and Dutta, 2021, Dellaporta et al., 2022] and for copulas in [Alquier et al., 2022, Huk et al., 2023].

The robust estimation of simulation-based models has been studied in [Pacchiardi and Dutta, 2021, Dellaporta et al., 2022] and for copulas in [Alquier et al., 2022, Huk et al., 2023].

We estimate  $\rho$  via minimizing the Energy score (a proper divergence)

$$S_E^\beta(\mathbb{P}_\rho, y) = 2 \cdot \mathbb{E}_{X \sim \mathbb{P}_\rho} \|X - y\|_2^\beta - \mathbb{E}_{X, X' \sim \mathbb{P}_\rho} \|X - X'\|_2^\beta.$$

With

$$S_E^\beta(\mathbb{P}_\rho, y) = 0 \iff \rho = \rho^*.$$

Due to the analytical form of  $\mathbb{P}_n$ , we can employ inverse probability sampling.

The robust estimation of simulation-based models has been studied in [Pacchiardi and Dutta, 2021, Dellaporta et al., 2022] and for copulas in [Alquier et al., 2022, Huk et al., 2023].

We estimate  $\rho$  via minimizing the Energy score (a proper divergence)

$$S_E^\beta(\mathbb{P}_\rho, y) = 2 \cdot \mathbb{E}_{X \sim \mathbb{P}_\rho} \|X - y\|_2^\beta - \mathbb{E}_{X, X' \sim \mathbb{P}_\rho} \|X - X'\|_2^\beta.$$

With

$$S_E^\beta(\mathbb{P}_\rho, y) = 0 \iff \rho = \rho^*.$$

Due to the analytical form of  $\mathbb{P}_n$ , we can employ inverse probability sampling.

→ Due to IPS, we only have to recurse on distributions, **halving the computational time** compared to a likelihood optimisation on densities and distributions.

## Theorem (Almost sure convergence [Hahn et al., 2018])

Let  $p_n$  be the Bayesian predictive density via the R-BP algorithm for  $X_n$  given observations  $x_1, \dots, x_n$ , with weight sequence  $(w_n)_{n \geq 1}$  satisfies

$$\sum_{i=1}^{\infty} w_i = \infty, \quad \text{and} \quad \sum_{i=1}^{\infty} w_i^2 < \infty,$$

with correlation parameter  $\rho \in (0, 1)$ . If the true density  $p^*$  of the data generating process is continuous and the corresponding support can be covered by  $\mathbb{P}_0$ , then

$$\text{KL}(p^*, p_n) \xrightarrow{\mathbb{P}^* \text{-a.s.}} 0$$



## Theorem (Almost sure convergence [Hahn et al., 2018])

Let  $p_n$  be the Bayesian predictive density via the R-BP algorithm for  $X_n$  given observations  $x_1, \dots, x_n$ , with weight sequence  $(w_n)_{n \geq 1}$  satisfies

$$\sum_{i=1}^{\infty} w_i = \infty, \quad \text{and} \quad \sum_{i=1}^{\infty} w_i^2 < \infty,$$

with correlation parameter  $\rho \in (0, 1)$ . If the true density  $p^*$  of the data generating process is continuous and the corresponding support can be covered by  $\mathbb{P}_0$ , then

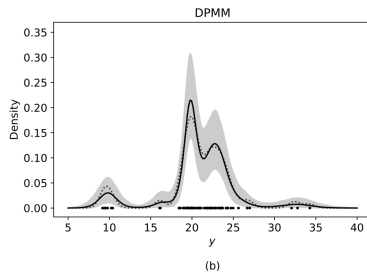
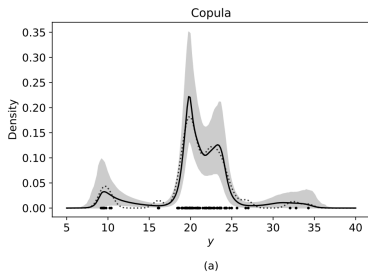
$$\text{KL}(p^*, p_n) \xrightarrow{\mathbb{P}^* \text{-a.s.}} 0$$

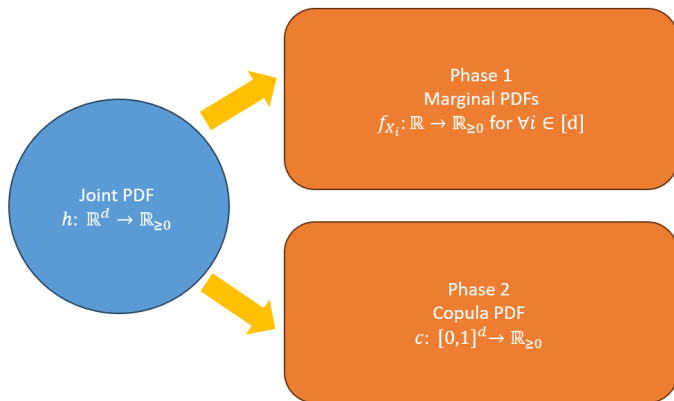
## Lemma (Stochastic Boundedness of R-BP)

For all  $x \in \mathbb{R}$ , the R-BP distribution function  $P^{(n)}(x)$  is stochastically bounded with

$$\left| P^{(\infty)}(x) - P^{(n)}(x) \right| = \mathcal{O}_p \left( n^{-1/2} \right).$$

# Phase 1: Comparison between BPR and DPMM





**R-Vine copula** decomposes high dimensional estimation into 2D copula building blocks. Use Sklar's theorem on conditional densities

$$p_{a|b}(x_a|x_b) = c_{a,b}(P_a(x_a), P_b(x_b)) \cdot p_a(x_a)$$

**R-Vine copula** decomposes high dimensional estimation into 2D copula building blocks. Use Sklar's theorem on conditional densities

$$p_{a|b}(x_a|x_b) = c_{a,b}(P_a(x_a), P_b(x_b)) \cdot p_a(x_a)$$

to decompose all conditional densities of

$$p(x_1, x_2, x_3) = p_1(x_1) \cdot p_{2|1}(x_2|x_1) \cdot p_{3|2,1}(x_3|x_2, x_1)$$

**R-Vine copula** decomposes high dimensional estimation into 2D copula building blocks. Use Sklar's theorem on conditional densities

$$p_{a|b}(x_a|x_b) = c_{a,b}(P_a(x_a), P_b(x_b)) \cdot p_a(x_a)$$

to decompose all conditional densities of

$$\begin{aligned} p(x_1, x_2, x_3) &= p_1(x_1) \cdot p_{2|1}(x_2|x_1) \cdot p_{3|2,1}(x_3|x_2, x_1) \\ &= \prod_{i=1}^d \{p_i(x_i)\} \prod_{j=1}^{d(d-1)/2} c_{S_j}(u_{S_j}, v_{S_j}), \end{aligned}$$

to end up with  $d \cdot (d - 1)/2$  copulas. Now, we only have to estimate bivariate copulas; much simpler.

→ a divide-and-conquer approach to copulas.

**R-Vine copula** decomposes high dimensional estimation into 2D copula building blocks.

**Example:**

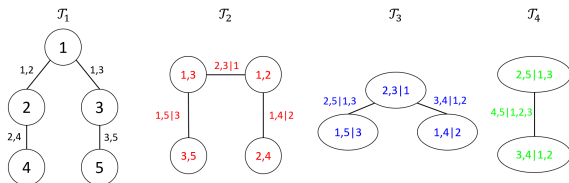


Figure: R-Vine example for 5D data.

$$\begin{aligned}
 c_{R\text{-vine}}(u_1, u_2, u_3, u_4, u_5) &= c(u_1, u_2) \cdot c(u_1, u_3) \cdot c(u_2, u_4) \cdot c(u_3, u_5) \\
 &\quad \cdot c_{1,5|3}(u_{1|3}, u_{5|3}) \cdot c_{2,3|1}(u_{2|1}, u_{3|1}) \cdot c_{1,4|2}(u_{1|2}, u_{4|2}) \\
 &\quad \cdot c_{2,5|1,3}(u_{2|1,3}, u_{5|1,3}) \cdot c_{3,4|1,2}(u_{3|1,2}, u_{4|1,2}) \\
 &\quad \cdot c_{4,5|1,2,3}(u_{4|1,2,3}, u_{5|1,2,3}) \cdot
 \end{aligned}$$

**R-Vine copula** decomposes high dimensional estimation into 2D copula building blocks.

**Example:**

$$\begin{aligned} C_{R-vine}(u_1, u_2, u_3, u_4, u_5) = & c(u_1, u_2) \cdot c(u_1, u_3) \cdot c(u_2, u_4) \cdot c(u_3, u_5) \\ & \cdot c_{1,5|3}(u_{1|3}, u_{5|3}) \cdot c_{2,3|1}(u_{2|1}, u_{3|1}) \cdot c_{1,4|2}(u_{1|2}, u_{4|2}) \\ & \cdot c_{2,5|1,3}(u_{2|1,3}, u_{5|1,3}) \cdot c_{3,4|1,2}(u_{3|1,2}, u_{4|1,2}) \\ & \cdot c_{4,5|1,2,3}(u_{4|1,2,3}, u_{5|1,2,3}) \cdot \end{aligned}$$

**Advantages:**

- ✓ Nonparametric with KDE bivariate copulas
- ✓ Convergence rate **independent of dimensions** (under assumptions)
- ✓ Very fast sampling



**R-Vine copula** decomposes high dimensional estimation into 2D copula building blocks.

**Example:**

$$\begin{aligned}c_{R-vine}(u_1, u_2, u_3, u_4, u_5) &= c(u_1, u_2) \cdot c(u_1, u_3) \cdot c(u_2, u_4) \cdot c(u_3, u_5) \\ &\quad \cdot c_{1,5|3}(u_{1|3}, u_{5|3}) \cdot c_{2,3|1}(u_{2|1}, u_{3|1}) \cdot c_{1,4|2}(u_{1|2}, u_{4|2}) \\ &\quad \cdot c_{2,5|1,3}(u_{2|1,3}, u_{5|1,3}) \cdot c_{3,4|1,2}(u_{3|1,2}, u_{4|1,2}) \\ &\quad \cdot c_{4,5|1,2,3}(u_{4|1,2,3}, u_{5|1,2,3}).\end{aligned}$$

**Advantages:**

- ✓ Nonparametric with KDE bivariate copulas
- ✓ Convergence rate **independent of dimensions** (under assumptions)
- ✓ Very fast sampling

**Disadvantages:**

- ✗ Selecting hyperparameter for bandwidth of KDE  
→ We use a similar Energy score sampling-based approach.

### Theorem (Convergence of Quasi-Bayesian Vine)

Assuming a correctly identified simplified vine structure for  $\mathbf{c}^{(\infty)}(\mathbf{u})$ , and using univariate R-BP marginal distributions with a simplified vine copula, the copula estimator error is stochastically bounded  $\forall \mathbf{x} \in \mathbb{R}^d$  with

$$|\mathbf{c}^{(\infty)}(\mathbf{x}) - \mathbf{c}^{(n)}(\mathbf{x})| = \mathcal{O}_p(n^{-r})$$

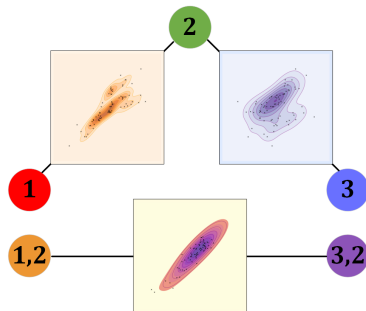
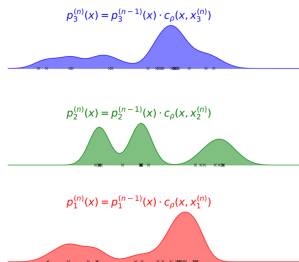
where  $n^{-r}$  is the convergence rate of the KDE pair-copula.

Then, our final model is:

$$\hat{f}(x^1, \dots, x^d) = \underbrace{\prod_{i=1}^d p_{n+1}^i(x^i)}_{\text{Phase 1}} \cdot \underbrace{c_K^r(\mathbb{P}^1(x^1), \dots, \mathbb{P}^n(x^n))}_{\text{Phase 2}},$$

where each  $p_{n+1}^i(\cdot)$  is the R-BP density estimator from Phase 1 and  $c_K^r(\cdot)$  is the robust R-vine KDE copula estimator from Phase 2.

# Strategy: in a diagram



$$\mathbf{p}^{(n)}(x_1, x_2, x_3) = p_1^{(n)}(x_1) \cdot p_2^{(n)}(x_2) \cdot p_3^{(n-1)}(x_3) \times c_{1,2}(u_1, u_2) \cdot c_{2,3}(u_2, u_3) \cdot c_{1,3|2}(u_{1|2}, u_{3|2})$$

Figure: Quasi-Bayesian Vine

# Experiments: UCI density estimation datasets

n/d	WINE 89/12	BREAST 97/14	PARKIN 97/16	IONO 175/30	BOSTON 506/13
KDE	13.69	10.45	12.83	32.06	8.34
PRticle Filter	37.04	41.95	50.32	150.96	46.68
DPMM (Diag)	17.46	16.26	22.28	35.30	7.64
DPMM (Full)	32.88	26.67	39.95	86.18	9.45
MAF	39.60	10.13	11.76	140.09	56.01
RQ-NSF	38.34	26.41	31.26	54.49	-2.20
R-BP	13.57	7.45	9.15	21.15	4.56
$R_d$ -BP	13.32	6.12	7.52	19.82	-13.50
AR-BP	13.45	6.18	8.29	17.16	-0.45
$AR_d$ -BP	<b>13.22</b>	6.11	7.21	16.48	-14.75
ARnet-BP	14.41	6.87	8.29	15.32	-5.71
QB-Vine	13.76	<b>4.67</b>	<b>4.93</b>	<b>-16.08</b>	<b>-31.04</b>

By rewriting the conditional density, we can simplify the marginals to obtain:

$$p(y|\mathbf{x}) = \frac{p(y, \mathbf{x})}{p(\mathbf{x})} = \frac{p_y(y) \cdot \prod_{i=1}^d \{p_i(x_i)\} \cdot \mathbf{c}(y, x_1, \dots, x_d)}{\prod_{i=1}^d \{p_i(x_i)\} \cdot \mathbf{c}(x_1, \dots, x_d)} = \frac{\mathbf{c}(y, x_1, \dots, x_d) \cdot p_y(y)}{\mathbf{c}(x_1, \dots, x_d)}.$$

By rewriting the conditional density, we can simplify the marginals to obtain:

$$p(y|\mathbf{x}) = \frac{p(y, \mathbf{x})}{p(\mathbf{x})} = \frac{p_y(y) \cdot \prod_{i=1}^d \{p_i(x_i)\} \cdot \mathbf{c}(y, x_1, \dots, x_d)}{\prod_{i=1}^d \{p_i(x_i)\} \cdot \mathbf{c}(x_1, \dots, x_d)} = \frac{\mathbf{c}(y, x_1, \dots, x_d) \cdot p_y(y)}{\mathbf{c}(x_1, \dots, x_d)}.$$

→ Estimate 2 Vines and  $d + 1$  marginals for the complete conditional model.

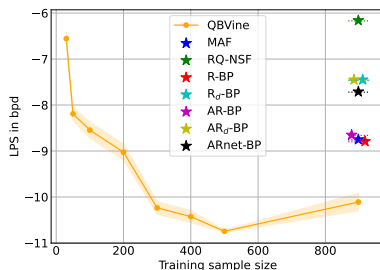
For discrete data, we use an approximation by adding small gaussian noise to the class values, making them continuous. This is needed for the copula to be unique.

# Experiments: UCI regression and classification datasets

<i>n/d</i>	Regression			Classification	
	BOSTON 506/13	CONCR 1,030/8	DIAB 442/10	IONO 351/33	PARKIN 195/22
Linear	0.87 $\pm$ 0.03	0.99 $\pm$ 0.01	1.07 $\pm$ 0.01	0.33 $\pm$ 0.01	0.38 $\pm$ 0.01
GP	0.42 $\pm$ 0.08	0.36 $\pm$ 0.02	1.06 $\pm$ 0.02	0.30 $\pm$ 0.02	0.42 $\pm$ 0.02
MLP	1.42 $\pm$ 1.01	2.01 $\pm$ 0.98	3.32 $\pm$ 4.05	0.26 $\pm$ 0.05	0.31 $\pm$ 0.02
R-BP	0.76 $\pm$ 0.09	0.87 $\pm$ 0.03	1.05 $\pm$ 0.03	0.26 $\pm$ 0.01	0.37 $\pm$ 0.01
R <sub>d</sub> -BP	0.40 $\pm$ 0.03	0.42 $\pm$ 0.00	1.00 $\pm$ 0.02	0.34 $\pm$ 0.02	0.27 $\pm$ 0.03
AR-BP	0.52 $\pm$ 0.13	0.42 $\pm$ 0.01	1.06 $\pm$ 0.02	0.21 $\pm$ 0.02	0.29 $\pm$ 0.02
AR <sub>d</sub> -BP	0.37 $\pm$ 0.10	0.39 $\pm$ 0.01	0.99 $\pm$ 0.02	0.20 $\pm$ 0.02	0.28 $\pm$ 0.03
ARnet-BP	0.45 $\pm$ 0.11	<b>-0.03</b> $\pm$ 0.00	1.41 $\pm$ 0.07	0.24 $\pm$ 0.04	0.26 $\pm$ 0.04
QB-Vine	<b>-0.81</b> $\pm$ 1.26	0.54 $\pm$ 0.34	<b>0.87</b> $\pm$ 0.20	<b>-1.85</b> $\pm$ 1.16	<b>-0.76</b> $\pm$ 0.28



# Experiments: Digits dataset $n = 1797$ , $d = 64$



**Figure:** Density estimation on the Digits data ( $n = 1797$ ,  $d = 64$ ) with reduced training sizes for the QB-Vine against other models fitted on the full training set. The QB-Vine achieves competitive performance for training sizes as little as  $n = 50$  and outperforms all competitors once  $n > 200$ .

- We sample from a mixture of 4 Gaussians with non-trivial covariances.

$$p(\mathbf{y}) = \sum_{k=1}^4 \pi_k \cdot \phi(\mathbf{y}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k),$$

where  $(\pi_1, \pi_2, \pi_3, \pi_4) = (0.2, 0.3, 0.1, 0.4)$  and

$$\boldsymbol{\mu}_k \stackrel{i.i.d.}{\sim} \mathcal{U}[-50, 50]^d, \quad \boldsymbol{\Sigma}_k \stackrel{i.i.d.}{\sim} \text{Wishart}(d, \mathbf{I}_d).$$

- We sample from a mixture of 4 Gaussians with non-trivial covariances.

$$p(\mathbf{y}) = \sum_{k=1}^4 \pi_k \cdot \phi(\mathbf{y}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k),$$

where  $(\pi_1, \pi_2, \pi_3, \pi_4) = (0.2, 0.3, 0.1, 0.4)$  and

$$\boldsymbol{\mu}_k \stackrel{i.i.d.}{\sim} \mathcal{U}[-50, 50]^d, \quad \boldsymbol{\Sigma}_k \stackrel{i.i.d.}{\sim} \text{Wishart}(d, \mathbf{I}_d).$$

- The Gaussians have a varying dimension  $d$  and we sample various amounts of samples  $n$ .

- We sample from a mixture of 4 Gaussians with non-trivial covariances.

$$p(\mathbf{y}) = \sum_{k=1}^4 \pi_k \cdot \phi(\mathbf{y}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k),$$

where  $(\pi_1, \pi_2, \pi_3, \pi_4) = (0.2, 0.3, 0.1, 0.4)$  and

$$\boldsymbol{\mu}_k \stackrel{i.i.d.}{\sim} \mathcal{U}[-50, 50]^d, \quad \boldsymbol{\Sigma}_k \stackrel{i.i.d.}{\sim} \text{Wishart}(d, \mathbf{I}_d).$$

- The Gaussians have a varying dimension  $d$  and we sample various amounts of samples  $n$ .
- We compare against a normalising flow, the RQ-NSF, as a benchmark off-the-shelf estimator with analytical form and sampling.

# Experiments: Isotropic Gaussian Mixtures

**Table:** Comparison of LPS for QB-Vine (our method) and RQ-NSF on GMM with 4 clusters for changing  $n$  and  $d$ . Results for our QB-Vine method are shown as the top numbers of each row, and RQ-NSF values as the bottom numbers of each row.

$d \setminus n$	50	100	300	500	$10^3$
10	$3.98 \pm 0.23$ $36.47 \pm 4.87$	$1.73 \pm 0.29$ $17.14 \pm 1.51$	$2.15 \pm 0.06$ $12.82 \pm 0.36$	$0.94 \pm 0.31$ $7.10 \pm 0.26$	$2.43 \pm 0.17$ $7.91 \pm 0.11$
30	-	$17.94 \pm 1.06$ $91.09 \pm 7.54$	$11.04 \pm 0.35$ $50.51 \pm 2.20$	$12.87 \pm 0.17$ $48.50 \pm 0.73$	$9.85 \pm 0.40$ $34.98 \pm 0.31$
50	-	-	$38.59 \pm 4.31$ $115.64 \pm 3.06$	$25.82 \pm 0.06$ $112.16 \pm 2.05$	$26.14 \pm 0.01$ $71.43 \pm 1.65$
100	-	-	-	-	$78.20 \pm 0.23$ $268.88 \pm 1.37$

## Conclusions:

## Conclusions:

- The QB-Vine is a fast recursive density estimator with analytical form.

## Conclusions:

- The QB-Vine is a fast recursive density estimator with analytical form.
- Evades the Curse of Dimensionality for dependency modelling; is very data-efficient.



## Conclusions:

- The QB-Vine is a fast recursive density estimator with analytical form.
- Evades the Curse of Dimensionality for dependency modelling; is very data-efficient.
- Highly parallelisable for marginal density estimation.

## Conclusions:

- The QB-Vine is a fast recursive density estimator with analytical form.
- Evades the Curse of Dimensionality for dependency modelling; is very data-efficient.
- Highly parallelisable for marginal density estimation.
- Statistically well-founded model that outperforms network-based methods.

## Conclusions:

- The QB-Vine is a fast recursive density estimator with analytical form.
- Evades the Curse of Dimensionality for dependency modelling; is very data-efficient.
- Highly parallelisable for marginal density estimation.
- Statistically well-founded model that outperforms network-based methods.

## Next step:

- Apply to ultra-high dimensional data, e.g. images and compare to implicit density estimators.



Alquier, P., Chérief-Abdellatif, B.-E., Derumigny, A., and Fermanian, J.-D. (2022).

Estimation of copulas via maximum mean discrepancy.

*Journal of the American Statistical Association*, pages 1–16.



Chen, Y.-C. (2017).

A tutorial on kernel density estimation and recent advances.

*Biostatistics & Epidemiology*, 1(1):161–187.



Dellaporta, C., Knoblauch, J., Damoulas, T., and Briol, F.-X. (2022).

Robust bayesian inference for simulator-based models via the mmd posterior bootstrap.

In *International Conference on Artificial Intelligence and Statistics*, pages 943–970. PMLR.



Dixit, V. and Martin, R. (2023).

A PRticle filter algorithm for nonparametric estimation of multivariate mixing distributions.






*Statistics and Computing*, 33(4):1–14.














Fong, E., Holmes, C., and Walker, S. G. (2021).

Martingale posterior distributions.

*arXiv preprint arXiv:2103.15671*.

-  Ghosal, S. and Van der Vaart, A. (2017).  
*Fundamentals of nonparametric Bayesian inference*, volume 44.  
Cambridge University Press.
-  Ghosh, J. K. and Tokdar, S. T. (2006).  
Convergence and consistency of Newton's algorithm for estimating mixing distribution.  
In *Frontiers in statistics*, pages 429–443. World Scientific.
-  Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014).  
Generative adversarial nets.  
*Advances in neural information processing systems*, 27.
-  Hahn, P. R., Martin, R., and Walker, S. G. (2018).  
On recursive bayesian predictive distributions.  
*Journal of the American Statistical Association*, 113(523):1085–1093.
-  Hjort, N. L., Holmes, C., Müller, P., and Walker, S. G. (2010).  
*Bayesian nonparametrics*, volume 28.  
Cambridge University Press.

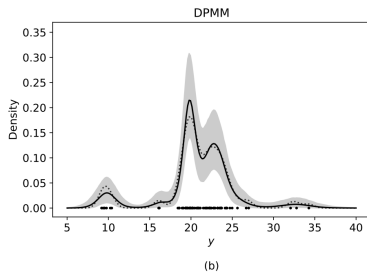
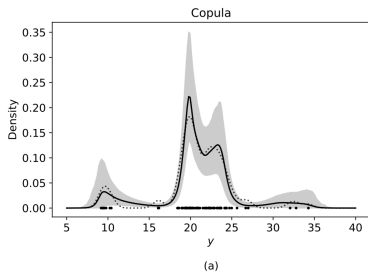
-  Huk, D., Adewoyin, R. A., and Dutta, R. (2023). Probabilistic rainfall downscaling: Joint generalized neural models with censored spatial gaussian copula. *arXiv preprint arXiv:2308.09827*.
-  Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.
-  Martin, R. (2012). Convergence rate for predictive recursion estimation of finite mixtures. *Statistics & Probability Letters*, 82(2):378–384.
-  Martin, R. (2021). A survey of nonparametric mixing density estimation via the predictive recursion algorithm. *Sankhya B*, 83:97–121.
-  Martin, R. and Ghosh, J. K. (2008). Stochastic approximation and Newton's estimate of a mixing distribution. *Statistical Science*, pages 365–382.
-  Martin, R. and Tokdar, S. T. (2009). Asymptotic properties of predictive recursion: robustness and rate of convergence.

-  Newton, M. A., Quintana, F. A., and Zhang, Y. (1998). Nonparametric bayes methods using predictive updating. In *Practical nonparametric and semiparametric Bayesian statistics*, pages 45–61. Springer.
-  Pacchiardi, L. and Dutta, R. (2021). Generalized bayesian likelihood-free inference using scoring rules estimators. *arXiv preprint arXiv:2104.03889*.
-  Rezende, D. and Mohamed, S. (2015). Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR.
-  Song, Y., Durkan, C., Murray, I., and Ermon, S. (2021). Maximum likelihood training of score-based diffusion models. *Advances in neural information processing systems*, 34:1415–1428.
-  Tokdar, S. T., Martin, R., and Ghosh, J. K. (2009). Consistency of a recursive estimate of mixing distributions. *The Annals of Statistics*, pages 2502–2522.

# Appendix



# Phase 1: Comparison between BPR and DPMM



In order to reduce the random effect of order in the observed data, we average the marginal predictive density over 10 permutations, i.e., for  $i \in 1, \dots, d$ ,

$$\hat{p}_n^i(X_i) = \frac{1}{10} \sum_{j=1}^{10} p^i(X_i | \Pi_j(x_{1:n}^i)),$$

where  $\Pi_j(\cdot)$  is a random permutation among observations. By consequence,

$$\hat{\mathbb{P}}_n^i(X_i) = \frac{1}{10} \sum_{j=1}^{10} \hat{\mathbb{P}}^i(X_i | \Pi_j(x_{1:n}^i)).$$

**Sampling procedure** (wlog here we assume for  $j^{\text{th}}$  marginal):

- 1 Get the support for the training data, define  $e$  as a distance of extrapolation,

$$\mathcal{I} = [\min - e, \max + e]$$

- 2 Take a grid of  $T$  size points in the support  $\mathcal{I}$ , i.e.  $\{\eta_t\}_{t=1}^T$ .
- 3 Evaluate  $\{\eta_t\}_{t=1}^T$  via  $\hat{\mathbb{P}}_n^j$  to get the context set, i.e.  $\{(\hat{\mathbb{P}}_n^j(\eta_t), \eta_t)\}_{t=1}^T$
- 4 Encode the context set into linear interpolator  $\psi$ , i.e.

$$\psi(\cdot; \{(\hat{\mathbb{P}}_n^j(\eta_t), \eta_t)\}_{t=1}^T)$$

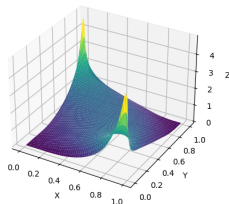
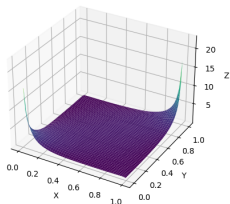
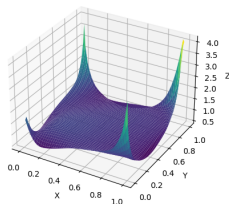
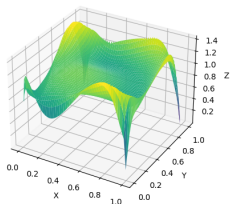
- 5 Sample noises  $\epsilon$  from uniform distribution  $\pi$  and apply  $\psi$  to get sample from  $\hat{\mathbb{P}}_n^j$ , i.e.,

$$\psi(\epsilon; \{(\hat{\mathbb{P}}_n^j(\eta_t), \eta_t)\}_{t=1}^T) \sim \hat{\mathbb{P}}_n^j$$

## Phase 2: Kernel Transformation Bivariate Copula

Suppose  $\{(U_i, V_i)\}_{i=1}^n \sim C$ , the kernel transformation copula density estimator of  $c$  with bandwidth  $h_n$  is

$$\hat{c}_n^K(u, v) = \frac{\sum_{i=1}^n K_{h_n}(\Phi^{-1}(u) - \Phi^{-1}(U_i))K_{h_n}(\Phi^{-1}(v) - \Phi^{-1}(V_i))}{n\phi(\Phi^{-1}(u))\phi(\Phi^{-1}(v))}.$$



## Introduction: Gaussian copula

A simple yet quite effective class of copulas are Gaussian copulas.

Consider  $(x_1, x_2) \sim \phi_2(\mathbf{0}, \Sigma)$  where  $\Sigma = \begin{bmatrix} 1 & 0.7 \\ 0.7 & 1 \end{bmatrix}$ . Then:

$$c(x_1, x_2) = \frac{\phi_2(x_1, x_2; \Sigma)}{\phi(x_1) \cdot \phi(x_2)}$$

The Gaussian copula is formed by a Gaussian joint and Gaussian marginals.

# Introduction: Gaussian copula

A simple yet quite effective class of copulas are Gaussian copulas.

Consider  $(x_1, x_2) \sim \phi_2(\mathbf{0}, \Sigma)$  where  $\Sigma = \begin{bmatrix} 1 & 0.7 \\ 0.7 & 1 \end{bmatrix}$ . Then:

$$c(x_1, x_2) = \frac{\phi_2(x_1, x_2; \Sigma)}{\phi(x_1) \cdot \phi(x_2)}$$

The Gaussian copula is formed by a Gaussian joint and Gaussian marginals.

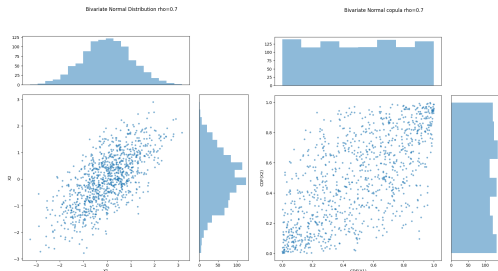


Figure: Joint plot on observation (left) and CDF (right) scales.

# Introduction: Gaussian copula

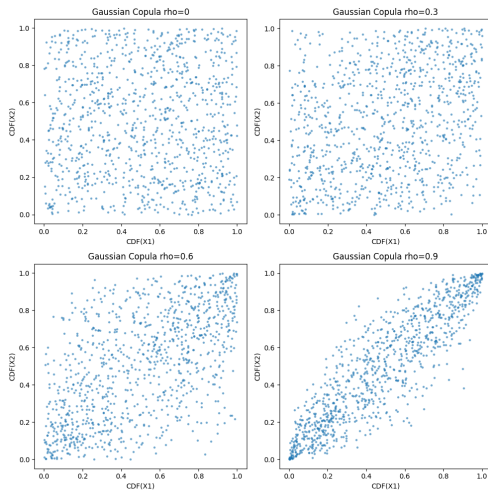


Figure: Gaussian copulas with different correlations.

## Proof.

From [Fong et al., 2021], suppose  $M > N$ , for  $\forall \epsilon > 0, \forall j \in [d]$ , we have that

$$\operatorname{argmin}_{x \in \mathbb{R}} \mathbb{P}(|\mathbb{P}_M^j(x) - \mathbb{P}_N^j(x)| \geq \epsilon) \leq 2 \exp\left(-\frac{\epsilon^2}{\frac{2\epsilon w_{N+1}}{3} + \frac{1}{2} \sum_{i=N+1}^M w_i^2}\right).$$

Then, we set

$$\delta = 2 \exp\left(-\frac{\epsilon^2}{\frac{2\epsilon w_{N+1}}{3} + \frac{1}{2} \sum_{i=N+1}^M w_i^2}\right) \simeq \mathcal{O}(e^{-N}),$$

as  $M \rightarrow \infty$ . Next, re-arrange to solve the quadratic equation with  $M \rightarrow \infty$  and we obtain

$$\begin{aligned} \epsilon &= \frac{-\log\left(\frac{\delta}{2}\right) \frac{2w_{N+1}}{3} + \sqrt{\left[\log\left(\frac{\delta}{2}\right) \frac{2w_{N+1}}{3}\right]^2 - 2 \log\left(\frac{\delta}{2}\right) \sum_{i=N+1}^M w_i^2}}{2} \\ &\simeq \mathcal{O}(N^{-0.5}). \end{aligned}$$

The last step follows that  $\sum_{i=N+1}^{\infty} w_i = \mathcal{O}(N^{-1})$  from our choice of  $\{w_i\}_{i \geq 1}$ . □