Scalable Gaussian Process Regression and Variable Selection under Automatic Relevance Determination Kernels

Jian Cao, Joseph Guinness, Marc Genton, and Matthias Katzfuss

June 30th 2022

Gaussian Process Regression (GPR)

A Gaussian process $f(\cdot)$ defined over a d-dimensional domain $\mathcal{X}$ is defined as:

$$\forall n \in \mathbb{N}, \mathbf{y} = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)] \sim N_n(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \text{ where}$$
$$\boldsymbol{\mu} = [\mu(\mathbf{x}_1), \ldots, \mu(\mathbf{x}_n)],$$
$$\boldsymbol{\Sigma} = \begin{bmatrix} K(\mathbf{x}_1, \mathbf{x}_1) & K(\mathbf{x}_1, \mathbf{x}_2) & \cdots & K(\mathbf{x}_1, \mathbf{x}_n) \\ K(\mathbf{x}_1, \mathbf{x}_2) & K(\mathbf{x}_2, \mathbf{x}_2) & \cdots & K(\mathbf{x}_2, \mathbf{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ K(\mathbf{x}_n, \mathbf{x}_1) & K(\mathbf{x}_n, \mathbf{x}_2) & \cdots & K(\mathbf{x}_n, \mathbf{x}_n). \end{bmatrix}$$

The log-likelihood

$$\ell_{\boldsymbol{\theta}} = -\frac{n}{2} \log 2\pi - \frac{1}{2} \log \det(\boldsymbol{\Sigma}) - \frac{1}{2} (\mathbf{y} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{y} - \boldsymbol{\mu})$$

is maximized to estimate $\boldsymbol{\theta}$ that parameterizes $\mu(\cdot)$ and $K(\cdot, \cdot)$.

ARD Kernel

- We assume a flexible structure on $K(\cdot, \cdot)$, the **automatic relevance determination (ARD)**:

$$K(\mathbf{x}_i, \mathbf{x}_j) = K(q(\mathbf{x}_i, \mathbf{x}_j)),$$

$$q(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{l=1}^{d} (x_{il} - x_{jl})^2 r_l^2},$$

  where r_l is the "relevance" or inverse range of the l-th covariate and $K(\cdot)$ becomes an isotropic kernel.
- ARD kernel accommodates heterogeneity in $\mathcal{X}$ and is a way to achieve model inference and variable selection simultaneously
- The computation bottleneck of GPR is the covariance structure. For simplification, we assume $\mu(\mathbf{x}_i) = 0$.

I. Vecchia Approximation

Use $p_{\boldsymbol{\theta}}(\cdot)$ to denote the probability density function (PDF):

$$p_{\boldsymbol{\theta}}(\mathbf{y}) = p_{\boldsymbol{\theta}}(y_1)p_{\boldsymbol{\theta}}(y_2|y_1)p_{\boldsymbol{\theta}}(y_3|y_1,y_2)\cdots p_{\boldsymbol{\theta}}(y_n|y_1,\ldots,y_{n-1})$$

$$\overset{\textbf{Vecchia}}{\approx} p_{\boldsymbol{\theta}}(y_1)p_{\boldsymbol{\theta}}(y_2|\mathbf{y}_{c(2)})p_{\boldsymbol{\theta}}(y_3|\mathbf{y}_{c(3)})\cdots p_{\boldsymbol{\theta}}(y_n|\mathbf{y}_{c(n)}),$$

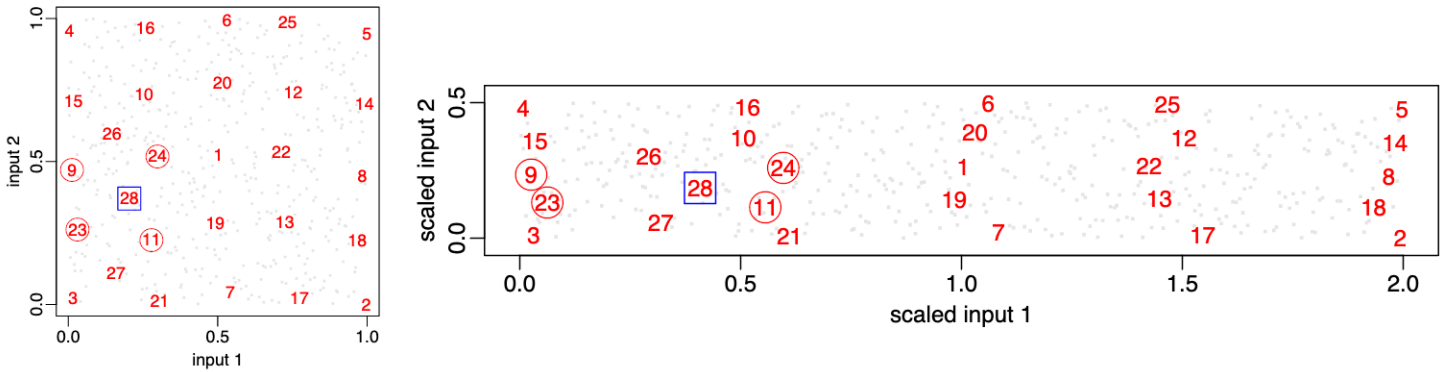
where $c(i)$ is a subset of $\{1,\ldots,i-1\}$ and $c(1)=\phi$. Notice that this is not limited to GPs. Obviously, ordering and choosing $c(i)$ matters:

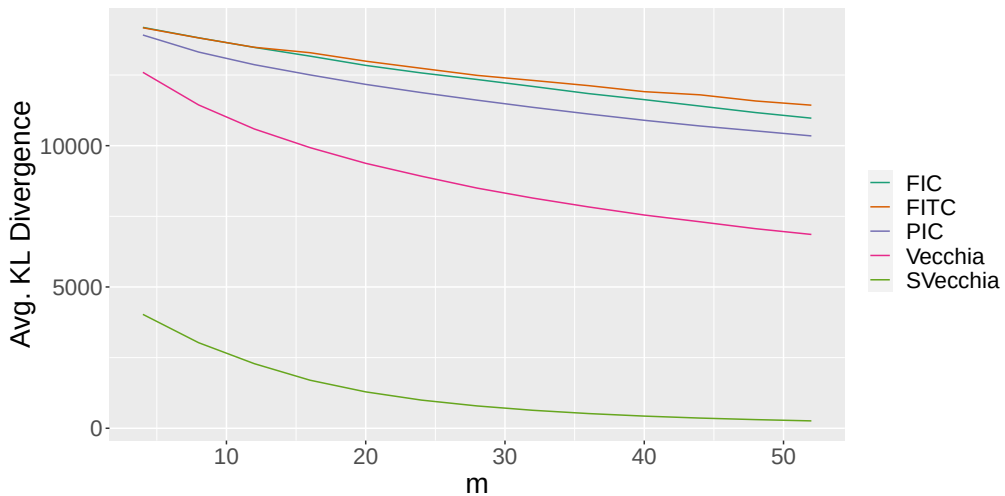
- **Maximin ordering** that maximizes the distance towards previously ordered locations is usually used
- $c(i)$ is typically chosen to be the m **nearest neighbors** among previous observations

Complexity $O(n^3) \to O(nm^3)$.

I. Vecchia Approximation

Scaled Vecchia (SVecchia) has been shown to improve the Vecchia approximation [2]. It is closely related to ARD kernels as it scales the i-th dimension of the domain by the i-th relevance parameter.





The top figure is taken from [2].

I. Vecchia Approximation

Using the definition of the KL distance, we can prove that:

Theorem

y *is from a stochastic process parameterized by* $\boldsymbol{\theta}_0$. $\hat{\ell}_{\boldsymbol{\theta}}$ *is the Vecchia approximation of the log-likelihood of* $\ell_{\boldsymbol{\theta}}$. *Then* $\boldsymbol{\theta}_0$ *is a global maximum of* $E_{\mathbf{y}}[\hat{\ell}_{\boldsymbol{\theta}}]$.

This theorem indicates:

- $\nabla\hat{\ell}_{\boldsymbol{\theta}}$ are unbiased estimating equations
- Using the Vecchia approximation **does not lead to bias**

Hopefully, one can also show:

$$\mathbf{Var}[\frac{1}{n}\hat{\ell}_{\boldsymbol{\theta}}] \xrightarrow{p} 0 \text{ uniformly for } \boldsymbol{\theta},$$

as in the maximum likelihood estimation.

II. Default Optimization for Vecchia

For Vecchia approximation of Gaussian processes, the Fisher scoring algorithm:

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} - \left\{ E_{\mathbf{y}|\boldsymbol{\theta}_0=\boldsymbol{\theta}^{(k)}}[\nabla^2 \hat{\ell}_{\boldsymbol{\theta}^{(k)}}] \right\}^{-1} \nabla \hat{\ell}_{\boldsymbol{\theta}^{(k)}}$$

is the state-of-the-art method for parameter estimation because the gradient and the Fisher information matrix (FIM) are:

- analytically available
- computed within $O(nm^3)$
- computed in parallel

Notice that the FIM can be very different from the negative Hessian matrix, especially when $\boldsymbol{\theta}^{(k)}$ is far away from $\boldsymbol{\theta}_0$.

II. Default Optimization for Vecchia

Fisher scoring may not perform well when the number of covariates d is large:

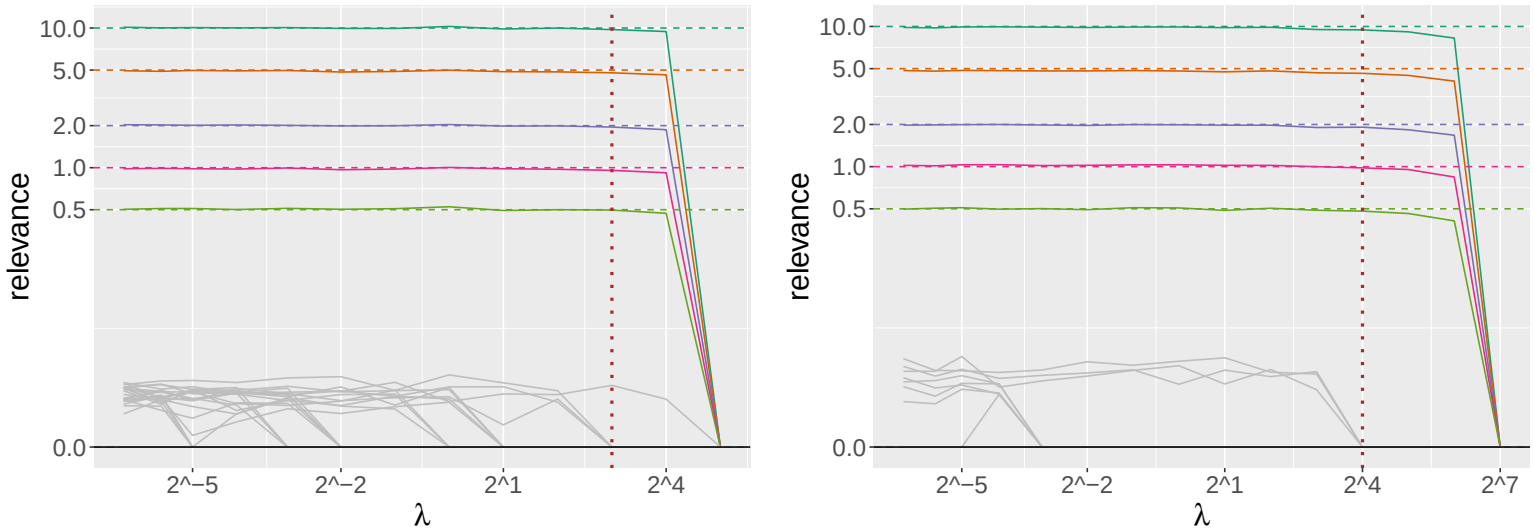
- local optima are more difficult to avoid when the number of optimization parameters becomes bigger
- computing FIM has a complexity of $\mathcal{O}(d^2)$
- cannot really deselect covariates in the middle of optimization

We aim to **gradually increase** the number of covariates involved in optimization through traversing the regularization path from strong to weak penalization:

- sequentially adding candidate covariates based on the **gradient of the log-likelihood**
- deselecting irrelevant covariates via a new **quadratic constrained coordinate descent** (QCCD) algorithm

III. Vecchia GPR

We name our method Vecchia Gaussian process regression, with the acronym of VGPR. A numerical illustration of VGPR is as follows:



(a) Independent covariates (b) Dependent covariates

Figure: Regularization path computed for the **adaptive bridge penalty** assuming covariates are either independent or dependent where λ is the penalty strength multiplier. Five true covariates are in colors whereas 995 fake covariates are in grey.

III. Vecchia GPR - covariate selection

The gradient of the log-likelihood w.r.t. **squared relevances** indicates the significance order of the covariates:

Theorem

Suppose there are d covariates, among which the first d_0 are true. When the gradient of the log-likelihood is evaluated at $[r_{10}, \ldots, r_{d_1 0}, 0, \ldots, 0]$, with $0 < d_1 < d_0$ and r_{l0} being the true value for r_l, we have

$$E[\frac{\partial \ell(\boldsymbol{\theta})}{\partial r_{l_1}^2}] \geq E[\frac{\partial \ell(\boldsymbol{\theta})}{\partial r_{l_2}^2}],$$

where $d_1 < l_1 \leq d_0 < l_2 \leq d$.

Note that the theorem is only proved for the squared exponential kernel. We conjecture that it is also valid for other kernels.

III. Vecchia GPR - covariate selection

Numerical support for covariate selection using the gradient of the log-likelihood:

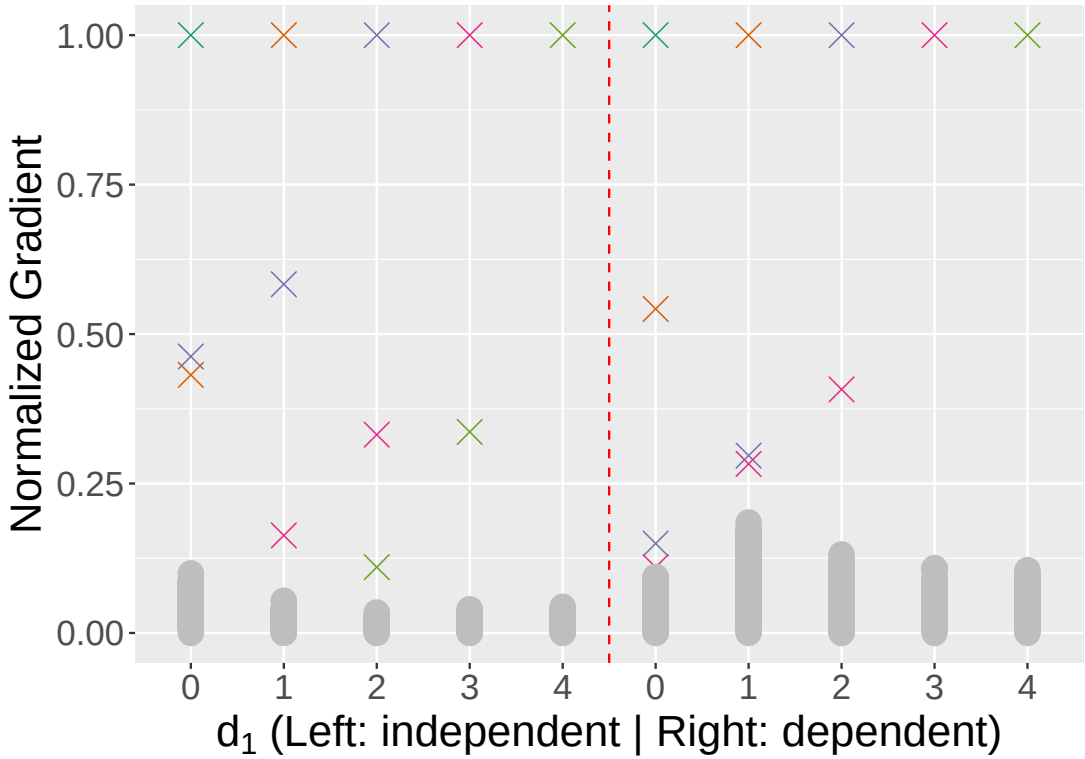


Figure: Scatter plots of the coefficients in the gradient of the log-likelihood evaluated at different d_1 values. Colored crosses correspond to true covariates and grey dots correspond to fake covariates. The true relevance parameters are $\mathbf{r}_0 = [10, 5, 2, 1, 0.5, 0, \ldots]$. Covariates are assumed either dependent of independent.

III. Vecchia GPR - covariate deselection

We introduce the **Quadratic Constrained Coordinate Descent (QCCD)** algorithm:

- $\boldsymbol{\theta} = \boldsymbol{\theta}^{(k)}$
- $i = 1, 2, \ldots$
  - $l = \text{mod}(i, d) + 1$
  - $\theta_l = \arg\max_{\theta_l} \ell_Q(\boldsymbol{\theta})$ subject to $\theta_l \geq 0$. Notice that $\boldsymbol{\theta}$ is considered fixed except for its l-th coefficient
- $\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}$

ℓ_Q is the quadratic approximation of the (Vecchia) log-likelihood at $\boldsymbol{\theta} = \boldsymbol{\theta}^{(k)}$.

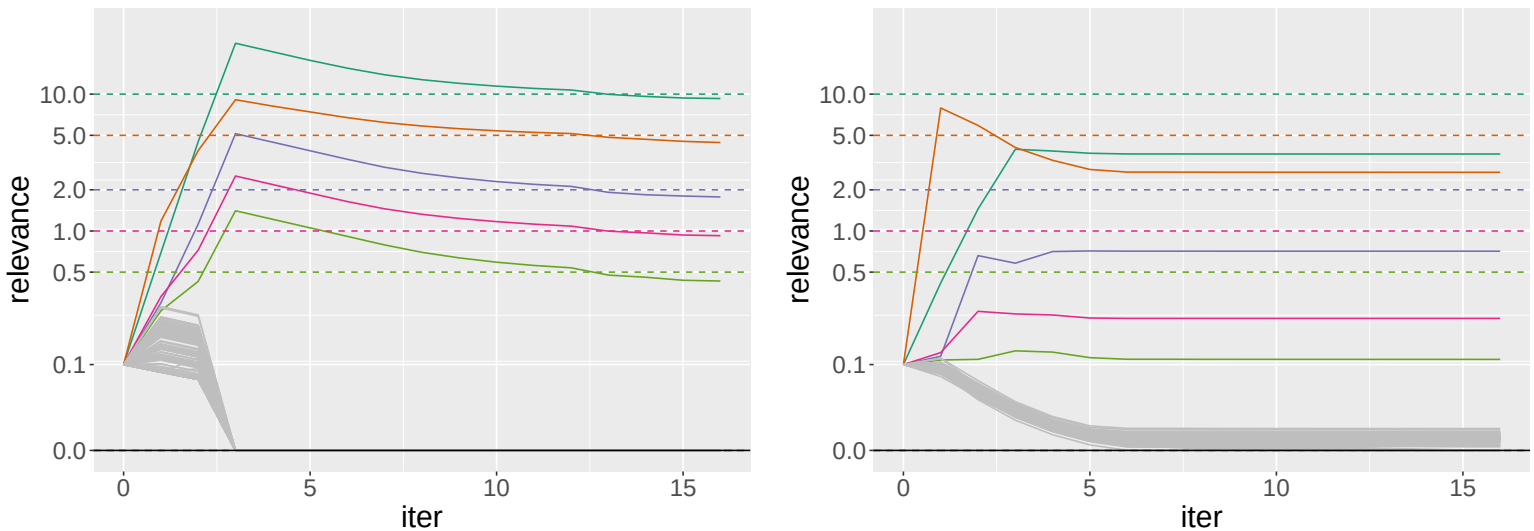
In practice, we also use a line search from $\boldsymbol{\theta}^{(k)}$ to $\boldsymbol{\theta}^{(k+1)}$ to enhance the algorithm stability.

III. Vecchia GPR - covariate deselection

QCCD improves Fisher scoring in:

- Ability to reach boundary values, e.g., $r_l = 0$, in optimization
- Better empirical convergence, w.r.t. both accuracy and speed

Comparison using $d = 100$ covariates, i.e., the Gaussian process is defined over a 100-dimensional space.



(a) QCCD $d = 100$ (b) Fisher scoring $d = 100$

Notice that the y-axis is on the log scale.

III. Vecchia GPR - covariate deselection

To enhance the deselection capacity, we introduce the **adaptive bridge penalty**. Bridge penalty:

$$w_\lambda(\boldsymbol{\theta}) = \lambda \sum_{l=1}^{d} r_l^{1/2}, \tag{1}$$

Adaptive bridge penalty:

$$w_\lambda(\boldsymbol{\theta}) = \lambda \sum_{l=1}^{d} (c_{\iota,l}^{\kappa} + r_l^2)^{1/4}, \tag{2}$$

where ι is the iteration number during optimization and $c_{\iota,l}^{\kappa}$ is the sum of the parameter r_l^2 over the previous κ iterations.
It is **sharper than Lasso** and **more stable than the bridge**. The choice of κ will be discussed in mini-batch subsampling.

III. Vecchia GPR - putting together

Putting pieces together, we can traverse the regularization path from strong to weak penalization, sequentially adding candidate covariates to a candidate set ζ based on the gradient of the log-likelihood and deselecting incorrectly selected covariates via the QCCD algorithm:

1. Initialize $\zeta = \phi$ and λ as a big value
2. Compute the gradient of the log-likelihood and select covariate(s) based on it. Add the new covariates to ζ
3. Run QCCD to estimate $\{r_l, l \in \zeta\}$, deselect covariate(s) with zero relevance
4. Go back to Step 2 until convergence
5. Reduce λ and go back to Step 2 until $\lambda = 0$

IV. Mini-batch subsampling for large n

Vecchia approximation has transformed the overall complexity to $O(n)$ with good parallel properties. For even larger n (e.g., 10^6) we design a mini-batch subsampling under the Vecchia approximation:

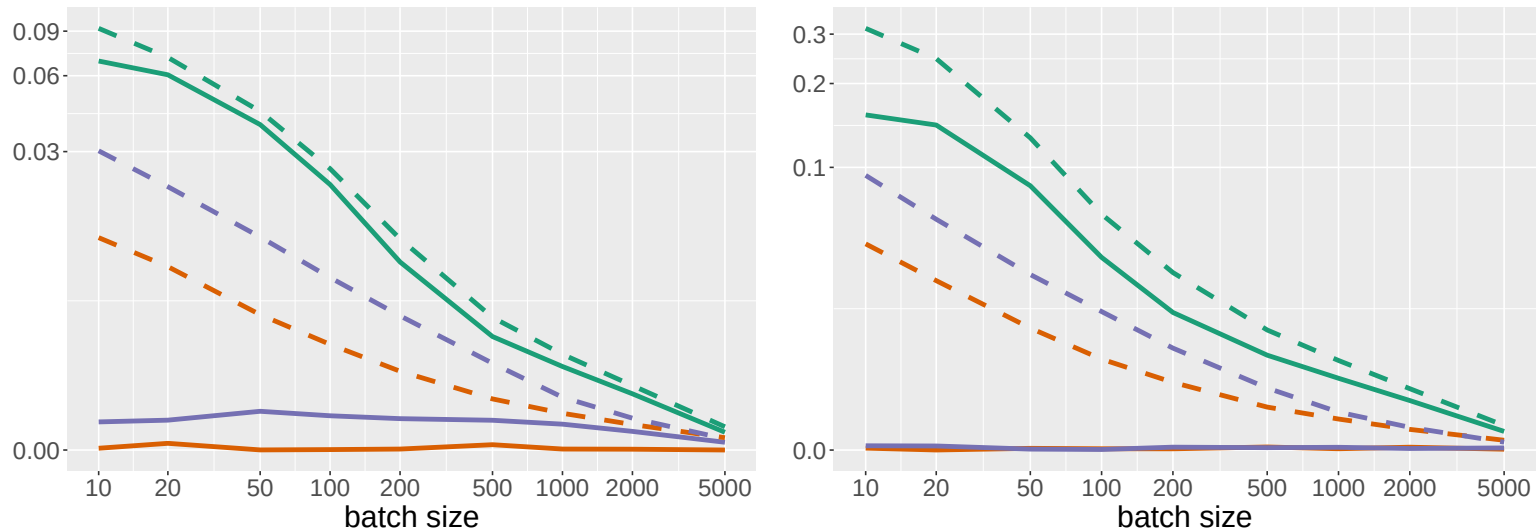
In each iteration, sample $\mathcal{S} \subset \{1, \ldots, n\}$ and take one step for optimizing $\sum_{i \in \mathcal{S}} \log p_{\boldsymbol{\theta}}(y_i | \mathbf{y}_{c(i)})$.

We compare it with two other mini-batch subsampling techniques:

1. Sample $\mathcal{S} \subset \{1, \ldots, n\}$ and take one step for optimizing $\log p_{\boldsymbol{\theta}}(y_{\mathcal{S}})$
2. Sample $\mathcal{S} \subset \{1, \ldots, n\}$ with probabilities $\{1^{-\frac{1}{d}}, \ldots, n^{-\frac{1}{d}}\}$ and take one step for optimizing $\sum_{i \in \mathcal{S}} \log p_{\boldsymbol{\theta}}(y_i | \mathbf{y}_{c(i)})$

IV. Mini-batch subsampling for large n

Comparison of the gradient estimators of the three mini-batch subsampling methods in terms of **absolute bias** and **RMSE**:

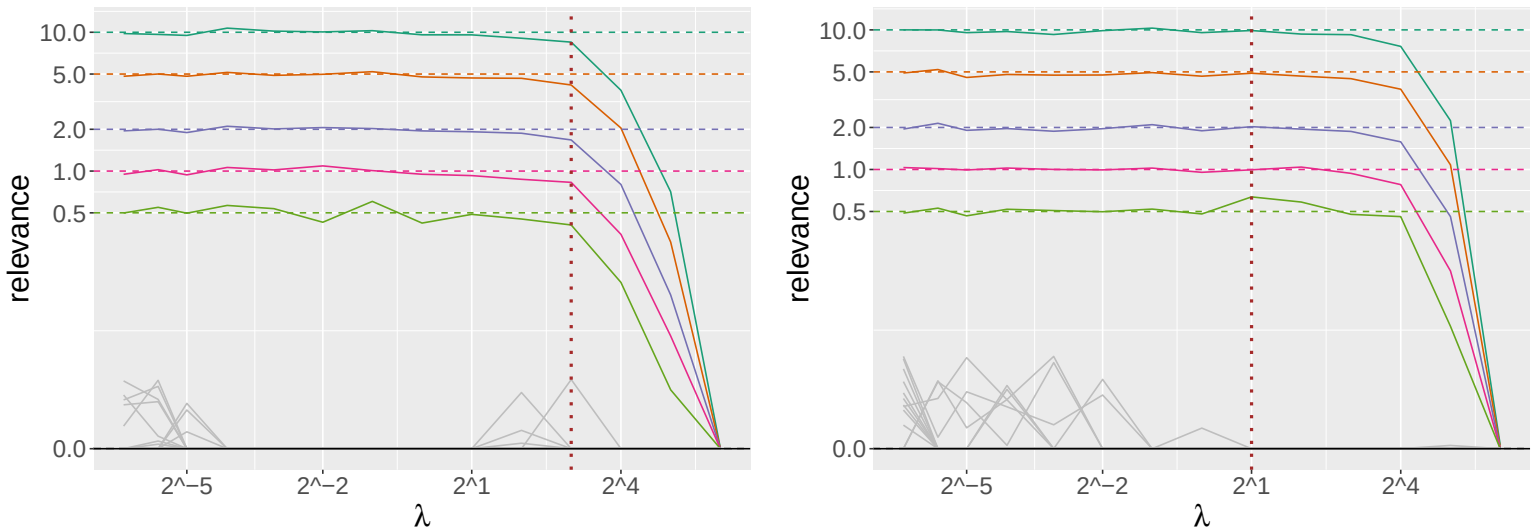


(a) Bias and RMSE w.r.t. $\frac{\partial \ell(\boldsymbol{\theta})}{\partial \mathbf{r}_1^2}$ (b) Bias and RMSE w.r.t. $\frac{\partial \ell(\boldsymbol{\theta})}{\partial \mathbf{r}_2^2}$

Figure: Absolute bias (solid) and RMSE (dashed) of three sampling methods. Green and blue correspond to the first and second comparison methods. Our proposed method can be shown theoretically to be unbiased.

IV. Mini-batching for large n

The **variance of the gradient estimator** from mini-batch subsampling is our motivation to propose the adaptive bridge penalty so that zero is no longer a stationary point for r_l. The figure below shows the regularization path with mini-batching with batch size of 128:



(a) Independent covariates (b) Dependent covariates

Figure: Regularization path computed using the proposed **mini-batch subsampling** assuming covariates are either independent or dependent where λ is the penalty strength multiplier. Five true covariates are in colors whereas 995 fake covariates are in grey.

V. Application studies

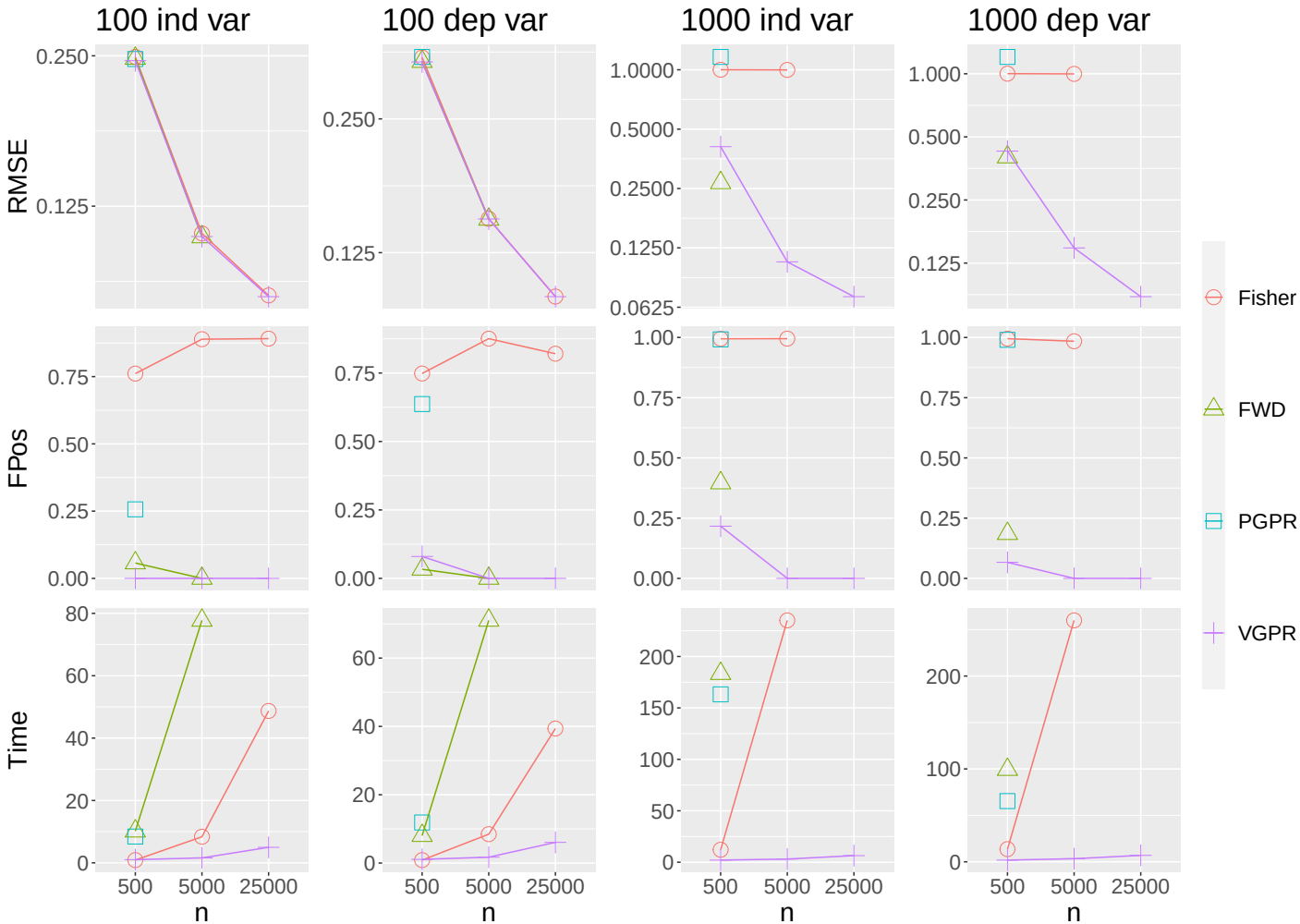


Figure: Comparison of four methods, namely enalized GP regression (PGPR), Fisher scoring (Fisher), forward selection (FWD), and our proposed VGPR in terms of prediction RMSE, false positive rates (FPos) of variable selection, and computation time in minutes

V. Application studies

Compare three variable selections methods, namely lasso linear regression, regression trees, and VGPR using three datasets. Fake covariates are generated to increase the number of covariates to 1,000.

Dataset	Method	RMSE	nSelect	FPos
Piston ($n = 10^6$)	Lasso	0.17	7	0%
	Regression Tree	0.92	6	17%
	VGPR	0.00	7	0%
Slice (UCI)	Lasso	0.44	792	59%
	Regression Tree	0.51	294	31%
	VGPR	0.32	49	18%
CASP (UCI)	Lasso	0.85	177	96%
	Regression Tree	0.92	6	17%
	VGPR	0.78	3	0%

Table: Comparison of three variable selection methods in terms of posterior prediction, model sparsity, and ratio of incorrectly selected covariates

Bibliography

[1] Jian Cao, Joseph Guinness, Marc G Genton, and Matthias Katzfuss. Scalable gaussian-process regression and variable selection using vecchia approximations. *arXiv preprint arXiv:2202.12981*, 2022.

[2] Matthias Katzfuss, Joseph Guinness, and Earl Lawrence. Scaled vecchia approximation for fast computer-model emulation. *arXiv preprint arXiv:2005.00386*, 2020.