Generalized Variable Selection Algorithms for Gaussian Process Models by LASSO-like Penalty

Dipak Dey, Zhiyong Hu

Department of Statistics
University of Connecticut

UCONN
UNIVERSITY OF CONNECTICUT

Background

- With the rapid development of modern technology, massive amounts of data are generated. However, it is common to encounter the case where data has high dimensional inputs but limited number of observations.
- Out of the large amount of inputs, it is often the case that only a few features are really meaningful or active.
- Unlike the classic generalized linear models which can select the variables using LASSO ([Tibshirani, 1996], [Park and Casella, 2008]), it is challenging to identify those active variables in Gaussian process models.

Current Methods

- Recall the RBF kernel:

$$C(\mathbf{x}_i, \mathbf{x}_j) = \sigma^2 \exp\left(-\frac{1}{2}\sum_{k=1}^{K}\frac{\left(\mathbf{x}_{i,k} - \mathbf{x}_{j,k}\right)^2}{\gamma_k^2}\right), \qquad (1)$$

  where marginal variance σ^2 and length-scales $\{\gamma_k^2\}_{k=1}^K$ are the hyperparameters that control the shape of the Gaussian process.

- The inverse of length-scale γ_k decides how relevant the k_{th} feature is. As the length-scale γ_k increases, the k_{th} feature becomes less relevant. When γ_k reaches a huge value, the k_{th} feature would only have ignorable impact on the covariance kernel, thus it can be removed from the model. This is so-called Automatic Relevance Determination (ARD) (e.g. [Williams and Rasmussen, 1995], [Neal, 2012])

Current Methods

- ARD is one of the most commonly used methods for variable selection in Gaussian process models ([Rasmussen and Williams, 2006]). However, it is known that ARD is open ended, which does not have a clear rule for the threshold.
- Recently, [Dance and Paige, 2022] introduced the spike and slab variational Gaussian process (SSVGP), which demonstrated superior performance compared to benchmark algorithms such as LASSO. Despite the attractive performance of the SSVGP, it is not applicable for classification tasks.
- More generalized approaches for variable selection in Gaussian process models are needed.

Inverse RBF Kernel

- Instead of the commonly used version of RBF kernel, we re-parameterize the RBF kernel to have the form:

$$C(x_i, x_j) = \sigma^2 \exp\left(-\frac{1}{2}\sum_{d=1}^{k} \ell_d^2 \times \left(x_{i,d} - x_{j,d}\right)^2\right), \qquad (2)$$

  where the inverse length-scale ℓ_d determines how relevant an input is.
- If the inverse length-scale ℓ_k^2 has a large value, any difference in k_{th} input would cause large impact on the covariance. Conversely, the input would only have ignorable influence on the covariance kernel if the inverse length-scale has a value that is close or even equal to 0, thus we can remove it from the model effectively.
- The interpretation is similar to that of the coefficients of regular GLM models.

Regularizing Prior

- In order to achieve the goal of selecting variables, regularization is needed to shrink the inverse length-scales of unnecessary features.
- In terms of Bayesian perspective, priors with large proportion of mass concentrated near 0 is needed for the inverse length-scales.
- [Park and Casella, 2008] discussed the Bayesian LASSO and demonstrated that the LASSO regression can be interpreted as a Bayesian regression with Laplace priors, which has density function proportional to

$$\pi(\beta) \propto \exp\left(-\tau|\beta|\right), \quad \beta \in (-\infty, \infty). \tag{3}$$

Regularizing Exponential Prior

- Since the inverse length-scale ℓ^2 is non-negative, the Laplace prior is not applicable in this case, therefore we set the prior for ℓ^2 as

$$\pi(\ell^2) \propto \begin{cases} \exp(-\tau|\ell^2|) & \text{if} \quad \ell^2 \geq 0; \\ 0 & \text{if} \quad \ell^2 < 0 \end{cases} \Rightarrow \pi(\ell^2) \propto \exp(-\tau\ell^2), \tag{4}$$

  which is actually an exponential distribution with mean $\frac{1}{\tau}$.
- The joint distribution of the model can be expressed as:

$$\prod_{i=1}^{n} p(y_i|f_i) \times \frac{1}{\sqrt{(2\pi)^n|\mathbf{C}|}} e^{-\frac{1}{2}\mathbf{f}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{f}} \times \pi(\sigma^2) \times \prod_{k=1}^{K} \pi(\ell_k^2), \tag{5}$$

  where $\prod_{i=1}^{n} p(y_i|f_i)$ is the model likelihood, $\mathbf{f}$ is the realization of Gaussian process, $\mathbf{C}$ is the corresponding covariance matrix, and $\pi(\sigma^2)$ represents the hyper-prior for marginal variance σ^2.

Regularizing Exponential Prior

- Inserting the exponential priors for $\{\ell_k^2\}_{k=1}^K$ and taking logarithm of equation (5), the log joint density of the model is proportional to

$$\sum_{i=1}^{n} \log p(y_i|f_i) - \frac{1}{2}\log|\mathbf{C}| - \frac{1}{2}\mathbf{f}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{f} + \log\pi(\sigma^2) - \tau\sum_{k=1}^{K}\ell_k^2. \quad (6)$$

- it is obvious to see the last element $\tau\sum_{k=1}^{K}\ell_k^2$ plays a role of regularization. That is, as the ℓ^2 gets larger, it introduces more penalty to the log joint density.
- Also, by increasing the rate parameter τ for the exponential priors, the degree of penalty grows up.

Existing Methods

- We take advantage of the reference distribution variable selection (RDVS) method proposed by [Linkletter et al., 2006], which augments the design matrix X by including an inert variable and compares it with the existing experimental variables.
- However, the RDVS is also designed for normal response and only considers the case in which Markov chain Monte Carlo (MCMC) method is used.
- Next, we will provide two general variable selection algorithms for Gaussian process models.

Variable Selection Algorithms

- Consider the design matrix $\mathrm{X} = (x_1, x_2, \ldots, x_K)$, where x_k is the column vector of k_{th} feature, a nuisance vector x_0 that is irrelevant to the data is binded to it, then the augmented design matrix is $\mathrm{X}^* = (x_0, \mathrm{X})$.
- The variable x_0 is known to be irrelevant, therefore it is expected to have no impact on the prediction if it is included in the model.
- In other words, the corresponding inverse length-scale ℓ_0^2 for x_0 should be very close to 0, then the researchers are confident to remove any inputs from the model if their inverse length-scales are even smaller than the ℓ_0^2.

Variable Selection Algorithms

- Ideally, the added nuisance column should be absolutely irrelevant to the data in order to eliminate any effect on the model. However, this is often not achievable when the sample size is small but the dimension is relatively high.
- To address this issue, the data augmentation should be repeated several times, say $\mathcal{M}$ times, and record the parameter estimates for the inverse length-scales of x_0 and all columns in X at each time.
- By repeating the procedure, it averages over the effect of the inert columns.
- As for the parameter estimates, it is natural to consider the maximum a posteriori probability (MAP) estimate, where variational inference is used for the computation.

Variable Selection Algorithms

- Denote $\hat{\ell}_j^2 = (\hat{\ell}_{1,j}^2, \ldots, \hat{\ell}_{m,j}^2, \ldots, \hat{\ell}_{\mathcal{M},j}^2)^{\mathrm{T}}$, where $\hat{\ell}_{m,j}^2$ is the MAP estimate for the inverse length-scale of x_j in the m_{th} iteration of the algorithm, $j = 0, 1, \ldots, K$.
- For each $j \in \{1, \ldots, K\}$, we compare the distribution of $\hat{\ell}_j^2$ with the distribution of $\hat{\ell}_0^2$ to evaluate whether the j_{th} feature is active or not.
- More specifically, the q_{th} percentile of $\hat{\ell}_0^2$, denoted as α_q, is used as the threshold for selecting active inputs. That is, if the median of $\hat{\ell}_j^2$ is smaller than α_q, then the j_{th} feature x_j is considered inactive, thus it can be dropped from the model effectively.

Algorithm 1

Algorithm 1 summarizes the steps for variable selection with random nuisance columns.

Algorithm 1: Variable Selection with Random Nuisance Columns

Data: X, $\mathbf{y}$
Input: $\mathcal{M}$, q
standardize X;
for $m \in \{1, \ldots, \mathcal{M}\}$ **do**
  augment X by a random vector x_0 from the design space, s.t. $\mathrm{X}^* = (x_0, \mathrm{X})$;
  obtain and record MAP estimates for all inverse length-scales;
end
store $\mathcal{L} = (\hat{\ell}_0^2, \hat{\ell}_1^2, \ldots, \hat{\ell}_K^2)$ and calculate α_q: the q_{th} percentile of $\hat{\ell}_0^2$;
for $k \in \{1, \ldots, K\}$ **do**
  if $median(\hat{\ell}_k^2) \geq \alpha_q$ **then**
    x_k is active;
  else
    x_k is inactive.
  end
end
Output: Index of active features

Remarks

- A larger q indicates the researcher prefers lower false discovery of inactive inputs, however, it increases the chance that some weakly active inputs are determined to be inactive. From the empirical results, an intuitive explanation of the threshold is to consider $(100 - q)\%$ as the maximum false discovery rate.
- As for the number of iterations $\mathcal{M}$, a larger value is always preferred since more iterations of augmentation can average down the effect of the nuisance columns.
- However, $\mathcal{M}$ is restricted by the computational power and computing time. By our empirical experiments, it turns out that in general $\mathcal{M} = 20$ iterations are enough for identifying active features accurately.

Algorithm 2

- We also propose to conduct principal component analysis (PCA) on design matrix X and use the last few PCA transformed columns as the nuisance columns for variable selection.
- The first few principle components explain most of the variance in the data, while the last few PCA transformed columns are often considered nuisance.
- PCA is an orthogonal linear transformation such that transformed data are uncorrelated, thus we can ensure that the added column in each iteration is uncorrelated to others.

Algorithm 2

Algorithm 2 summarizes the steps for variable selection with PCA transformed Columns.

Algorithm 2: Variable Selection with PCA Transformed Columns

Data: X, **y**
Input: $\mathcal{M}$, q
standardize X;
run PCA transformation on X to get X_{PCA};
standardize X_{PCA};
for $m \in \{1, \ldots, \mathcal{M}\}$ **do**
    augment X by m_{th} from the last column of X_{PCA};
    obtain and record MAP estimates for all inverse length-scales;
end
store $\mathcal{L} = (\hat{\ell}_0^2, \hat{\ell}_1^2, \ldots, \hat{\ell}_K^2)$ and calculate α_q: the q_{th} percentile of $\hat{\ell}_0^2$;
for $k \in \{1, \ldots, K\}$ **do**
    if $median(\hat{\ell}_k^2) \geq \alpha_q$ **then**
        x_k is active;
    else
        x_k is inactive.
    end
end
Output: Index of active features

Remarks

- When the sample size is small but the dimension is relatively high, the Algorithm 2 often outperforms the Algorithm 1.
- However, when the dimension of data is relatively low, the Algorithm 2 is not recommended since the last few PCA transformed columns are likely to be correlated to the original X and the value of $\mathcal{M}$ is limited.

Simulation: Binary Response Variable

- The design matrix X of this example is generated by drawing a 500×56 matrix (i.e., $n = 500, K = 56$) of standard normal variables.
- The latent probability of the dependent variables is determined by

$$E(y_i|\mathbf{x}_i) = \mu_i = g^{-1}(\mathbf{x}_{i,1} - \mathbf{x}_{i,2} - \tfrac{1}{2}\mathbf{x}_{i,3} + \tfrac{1}{4}\mathbf{x}_{i,4} + \tfrac{1}{8}\mathbf{x}_{i,5} + \tfrac{1}{16}\mathbf{x}_{i,6}), \quad i = 1, \ldots, 500, \tag{7}$$

- The format of the linear function is designed to test the sensitivity of the algorithms. The elements with smaller magnitude of coefficient tend to be less active.

Power of Priors

- The simulation is repeated 50 times with different exponential priors on the inverse length-scales to check the power of the regularization of the exponential prior using Algorithm 1.
- We also run the algorithms without regularization (i.e., no prior on inverse length-scales) to show the importance of the proposed exponential priors.

Power of Priors

- The variable selection algorithm is not able to detect the active features when no regularization is assigned. Instead, with the proposed exponential prior, the algorithm can work very well.
- By increasing the rate parameter τ for the exponential prior, the degree of penalty increases.
- The false discovery rate decreases as the rate parameter for the exponential prior increases. However, the weakly active features are less likely to be detected if the penalty is too large.

Threshold	Prior	Discovery Rate						
		Inactive Features	x_1	x_2	x_3	x_4	x_5	x_6
α_{80}	No Prior	0.17	0.14	0.26	0.12	0.12	0.18	0.18
	$Exp(1)$	0.10	1	1	0.94	0.62	0.20	0.04
	$Exp(2)$	0.06	1	1	0.96	0.72	0.22	0.04
	$Exp(4)$	0.05	1	1	0.98	0.68	0.24	0.04
	$Exp(10)$	0.03	1	1	0.96	0.58	0.18	0.02

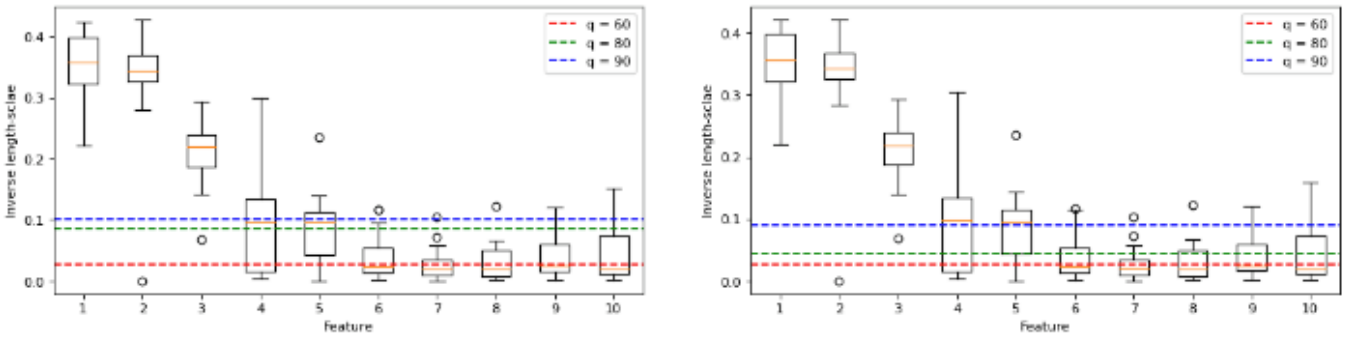
Simulation: Binary Response Variable

- The following table shows the discovery rate of the actual active or weakly active features $x_1, \ldots, x_6$ and all other 50 inactive features using the $Exp(4)$ prior for inverse length-scales under different thresholds.
- Both algorithms can easily identify the first 3 strongly active features. Algorithm 2 performs slightly better on identifying the 4_{th} feature. Although the 5_{th} feature that has small coefficient is hard to be detected, it is still identified more often than the absolutely inactive features. As for the 6_{th} feature, since its coefficient is too small the algorithm considers it as inactive.

Threshold	Algorithm	Discovery Rate						
		Inactive Features	x_1	x_2	x_3	x_4	x_5	x_6
α_{60}	Random	0.35	1	1	1	0.86	0.58	0.22
	PCA	0.38	1	1	1	0.94	0.62	0.16
α_{80}	Random	0.05	1	1	0.98	0.68	0.24	0.04
	PCA	0.08	1	1	0.96	0.72	0.28	0.06
α_{90}	Random	0.01	1	1	0.92	0.48	0.12	0.02
	PCA	0.01	1	1	0.90	0.50	0.14	0.02

Simulation: Binary Response Variable

The following figure shows the Box-plots of the first 10 features of $\mathcal{L}$ in one repetition of simulations, while other 46 features are ignored due to the limited space. The first 2 features are most active, thus their boxes locate above all others obviously. As the magnitude of the 3_{rd} to 5_{th} features' coefficients decreases, the corresponding boxes locate lower and lower. As for the 6_{th} feature, it is similar to those of inactive features as its coefficient is too small.



(A) Random nuisance columns (B) PCA transformed columns

FIGURE 4.3.2: Box-plots of the first 10 features of $\mathcal{L}$ obtained from (a) Algorithm 1 and (b) Algorithm 2 in example 2. The red, green and blue horizontal dashed lines represent α_{60}, α_{80} and α_{90} respectively.

Correlated Predictors

- Following the design of correlated features in [Barber and Candès, 2015], each row of the design matrix is drawn from a $\mathcal{N}(\mathbf{0}, \Theta)$ distribution, where $\Theta_{i,i} = 1$ for all i and $\Theta_{i,j} = 0.3$ for all $i \neq j$.
- The following table shows the discovery rate of the features using the $Exp(2)$ prior for inverse length-scales when all features are correlated, under different thresholds of Algorithm 2. The performance of the proposed algorithm is only degraded on the weakly active features.

Threshold	Discovery Rate						
	Inactive Features	x_1	x_2	x_3	x_4	x_5	x_6
α_{60}	0.37	1	1	1	0.94	0.58	0.26
α_{80}	0.07	1	1	1	0.64	0.12	0.08
α_{90}	0.02	1	1	0.98	0.42	0.06	0.02

Simulation: Normal Response Variable

- The design matrix X of this example is generated by drawing a 500×54 matrix (i.e., $n = 500, K = 54$) of standard normal variables.
- The mean of the dependent variables is determined by

$$E(y_i|\mathbf{x}_i) = \mu_i = \sin \mathbf{x}_{i,1} + \tfrac{3}{2} \cos \mathbf{x}_{i,2} + 2 \sin \mathbf{x}_{i,3} + \tfrac{5}{2} \cos \mathbf{x}_{i,4}, \quad i = 1, \ldots, 500, \tag{8}$$

  where the dependent variable y_i is sampled from $\mathcal{N}(\mu_i, 1)$.
- In this example, a more complex non-linear latent probability function is considered.

Simulation: Normal Response Variable

Again, the simulation is repeated 50 times. the following table shows the discovery rate of the active features $x_1, \ldots, x_4$, and the inactive features from SSVGP and Algorithm 2 with $Exp(2)$ prior for inverse length-scales. From the results, our proposed method can achieve even better performance than SSVGP by selecting an optimal threshold.

Prior	Algorithm	Discovery Rate				
		Inactive Features	x_1	x_2	x_3	x_4
	SSVGP	0.09	1	0.98	1	0.98
$Exp(2)$	Algorithm 2 with α_{80}	0.16	1	1	1	1
	Algorithm 2 with α_{90}	0.04	1	1	1	1

Application to Electroencephalography Data

- An application to multi-subject electroencephalography (EEG) data that studies alcoholic levels of experimental subjects is conducted.
- Clinically, EEG refers to the recording of the brain's spontaneous electrical activity over a period of time, as recorded from multiple electrodes placed on the scalp.
- The EEG data is usually a 3-dimensional tensor that has dimension of $n \times K \times t$, where n, K and t are the number of experimental subjects, the number of electrodes (locations) and the number of time points respectively.
- The data used in our work is from an experiment on studying the EEG correlates of genetic predisposition to alcoholism (see [Hu and Allen, 2015] and [Mohammed et al., 2019]) with dimension $122 \times 57 \times 256$ (i.e., $n = 122, K = 57, t = 256$).

Electroencephalography Data

- The experimental subjects are divided into two groups with 77 subjects in the alcoholic group and the other 45 subjects in the control group.
- During the experiment, stimulus was applied to each subject and the electrical activity is recorded.
- The target is to predict the alcoholic status (binary) of the subject given the EEG records.

Electroencephalography Data

- Since the number of features 57×256 is extremely high, it is not suitable to use the general GLM in this case.
- We adopt the local aggregate modeling approach proposed by [Mohammed et al., 2019] that fits a local model at each time point separately.
- The complexity of brain activities is well-known, and it surpasses what linear latent functions can represent. Thus, Gaussian process models are considered more suitable for capturing this level of complexity.

Variable Selection

- At each time point, given the 122×57 design matrix, a Gaussian process model is fitted as the local model.
- However, at a particular time point it is clear that not all the regions of brain are activated. Learning which regions of brain are correlated with the stimulus is also a crucial topic.
- In addition to the local Gaussian process model (LGP), variable selection is also conducted to find the active locations at each time point using the proposed algorithms.
- For notational simplicity, the LGP incorporated with Algorithm 1 is denoted as LGP.1, while the LGP incorporated with Algorithm 2 is denoted as LGP.2.

Setups

- Both LGP.1 and LGP.2 use $Exp(2)$ prior for the inverse length-scales.
- We predict the alcoholic status using each local model and record the responses sequentially through all time points. Thus, for each individual, we will have a binary prediction vector with length 256. As suggested in [Mohammed et al., 2019], we predict the subject level responses as the class indicator with the longest length of run.
- [Mohammed et al., 2019] proposed a local Bayesian model (LBM) with independent spike-and-slab prior for this problem. [Mohammed et al., 2020] then update the LBM using structured spike-and-slab prior that utilizes spatial information. The results from these two methods are also included for comparison.

Prediction Accuracy

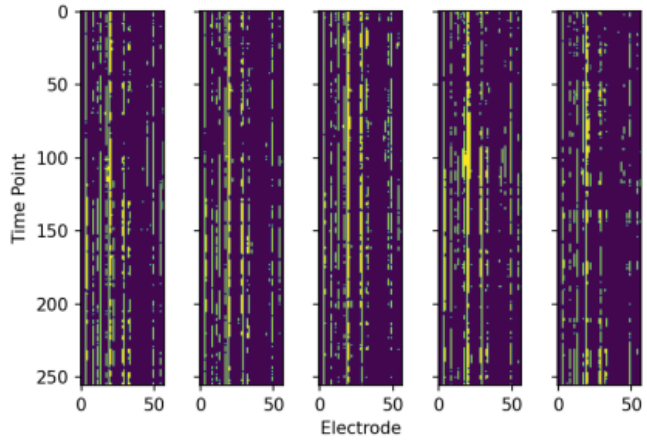
- The following table shows the average prediction accuracy and standard error across multiple 5-fold cross-validations for the EEG data using different methods.
- It is easy to see that the LGP methods are more accurate than the LBM methods.
- The standard errors of LGP methods are also lower than LBM, which suggests our proposed models are more stable.
- As for the comparison among LGP methods, the LGP.2 shows higher accuracy in prediction with lower standard errors.

Method	Accuracy	Std. Err.
LBM (Independent)	0.701	0.040
LBM (Structured)	0.717	0.029
LGP.1 with α_{50}	0.734	0.021
LGP.1 with α_{55}	0.727	0.024
LGP.2 with α_{50}	0.746	0.018
LGP.2 with α_{55}	0.729	0.012

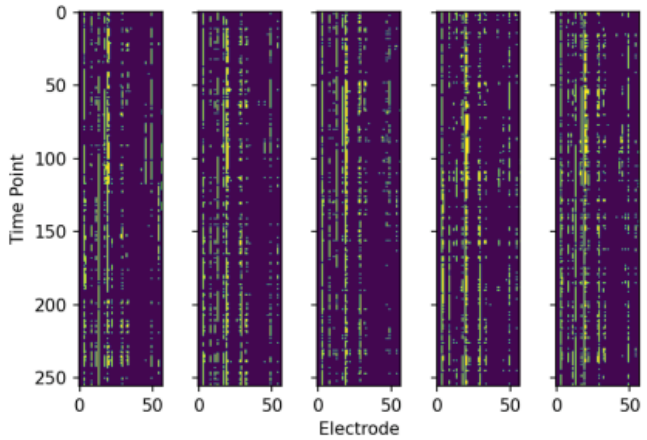
Remarks

- Unlike the simulation experiments that have known highly active features, a threshold larger than α_{60} would over-sparsify the models for the EEG data, resulting poor estimations.
- Though the prediction accuracy using threshold α_{55} is slightly lower compared to α_{50}, it provides more reasonably sparse models. Therefore, we discuss the LGP models with α_{55} in the remaining part.
- Besides the prediction accuracy, identifying locations of brain that are correlated with the stimulus is also of interest.

EEG Activated Regions



(A) LGP.1 with α_{55}



(B) LGP.2 with α_{55}

FIGURE 4.4.1: EEG activated locations obtained from (a) LGP.1 with α_{55} and (b) LGP.2 with α_{55} in a 5-fold cross-validation. The black area indicates that the location of electrode at corresponding time point is not considered active by the algorithm. The plot from left to right in a sub-figure is obtained from cross-validation fold 1 to 5 respectively.

Findings

- The activated locations tend to stay active for a long time, which suggests the particular regions of brain are reacting to the stimulus consistently.
- The identified active regions of both methods are consistent across different folds of subjects.
- Although the detected active location sets of Algorithm 2 are more sparse, the major part of the selected locations is quite similar, showing that both algorithms can identify the active regions consistently.

Summary

- This work introduces two variable selection algorithms for Gaussian process models, which use artificial nuisance columns as baseline for identifying the active features.
- We also propose to use inverse-RBF kernel and regularizing exponential prior on inverse length-scale parameters.
- The simulation experiments and application to EEG data demonstrate the performance of the proposed algorithms. In particular, the Algorithm 2 shows promising capability of identifying sparse active features while keeping the important information from the EEG data.

[Barber and Candès, 2015] Barber, R. F. and Candès, E. J. (2015).
Controlling the false discovery rate via knockoffs.
The Annals of Statistics, 43(5):2055–2085.

[Dance and Paige, 2022] Dance, H. and Paige, B. (2022).
Fast and scalable spike and slab variable selection in high-dimensional gaussian processes.
In *International Conference on Artificial Intelligence and Statistics*, pages 7976–8002. PMLR.

[Hu and Allen, 2015] Hu, Y. and Allen, G. I. (2015).
Local-aggregate modeling for big data via distributed optimization: Applications to neuroimaging.
Biometrics, 71(4):905–917.

[Linkletter et al., 2006] Linkletter, C., Bingham, D., Hengartner, N., Higdon, D., and Ye, K. Q. (2006).
Variable selection for gaussian process models in computer experiments.
Technometrics, 48(4):478–490.

[Mohammed et al., 2019] Mohammed, S., Dey, D. K., and Zhang, Y. (2019).
Bayesian variable selection using spike-and-slab priors with application to high dimensional electroencephalography data by local modelling.
Journal of the Royal Statistical Society: Series C (Applied Statistics), 68(5):1305–1326.

[Mohammed et al., 2020] Mohammed, S., Dey, D. K., and Zhang, Y. (2020).
Classification of high-dimensional electroencephalography data with location selection using structured spike-and-slab prior.
Statistical Analysis and Data Mining: The ASA Data Science Journal, 13(5):465–481.

[Neal, 2012] Neal, R. M. (2012).
Bayesian learning for neural networks, volume 118.
Springer Science & Business Media.

[Park and Casella, 2008] Park, T. and Casella, G. (2008).
The bayesian lasso.
Journal of the American Statistical Association, 103(482):681–686.

[Rasmussen and Williams, 2006] Rasmussen, C. E. and Williams, C. K. I. (2006).
Gaussian processes for machine learning.
Adaptive computation and machine learning. MIT Press.

[Tibshirani, 1996] Tibshirani, R. (1996).
Regression shrinkage and selection via the lasso.
Journal of the Royal Statistical Society: Series B (Methodological), 58(1):267–288.

[Williams and Rasmussen, 1995] Williams, C. K. and Rasmussen, C. E. (1995).
Gaussian processes for regression.
In *Proceedings of the 8th International Conference on Neural Information Processing Systems*, pages 514–520.

Simulation 2: Binary Response Variable

- The design matrix X of this example is generated by drawing a 500×54 matrix (i.e., $n = 500, K = 54$) of standard normal variables.
- The latent probability of the dependent variables is determined by

$$E(y_i|\mathbf{x}_i) = \mu_i = g^{-1}(\sin \mathbf{x}_{i,1} + \tfrac{3}{2} \cos \mathbf{x}_{i,2} + 2 \sin \mathbf{x}_{i,3} + \tfrac{5}{2} \cos \mathbf{x}_{i,4}), \quad i = 1, \ldots, 500, \tag{9}$$

- In this example, a more complex non-linear latent probability function is considered.

Simulation 2: Binary Response Variable

Both algorithms can successfully detect the active features at most of the times. The $Exp(2)$ prior works better than the other one, suggesting that a better choice of prior on inverse length-scales can lead to better variable selection.

Prior	Threshold	Algorithm	Discovery Rate				
			Inactive Features	x_1	x_2	x_3	x_4
$Exp(2)$	α_{60}	Random	0.33	0.88	0.98	1	1
		PCA	0.37	0.90	1	1	1
	α_{80}	Random	0.06	0.82	0.94	1	1
		PCA	0.07	0.80	0.94	1	1
	α_{90}	Random	0.01	0.76	0.90	1	1
		PCA	0.01	0.70	0.94	1	1
$Exp(4)$	α_{60}	Random	0.32	0.86	0.92	1	1
		PCA	0.41	0.86	0.92	1	1
	α_{80}	Random	0.05	0.80	0.76	1	1
		PCA	0.07	0.78	0.78	1	1
	α_{90}	Random	0.01	0.74	0.68	1	1
		PCA	0.01	0.74	0.68	1	1

Thanks!