# Gaussian Process Ensembles and the Bayesian Committee Machine

Joint work with Vincent Dutordoir (University of Cambridge)

Nicolas Durrande (Monumo) — LIKE23 Bern
Bern, June 2022

## Gaussian process models do scale

## Gaussian process models do scale

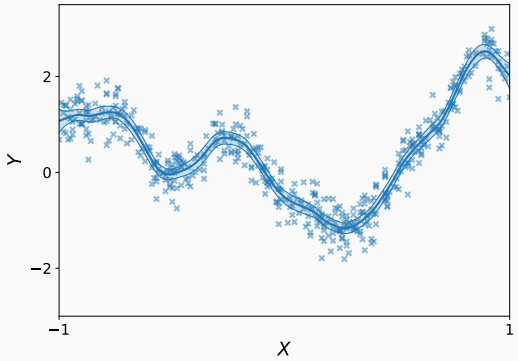
We have various tools at our belt to do so:

- Exploit structure in kernel matrices (GP with Markov property, ...)
- **Sparse GPs** (variational inference, ...)
- Solving matrix inverse approximately (conjugate gradients, ...)
- **GP ensembles**

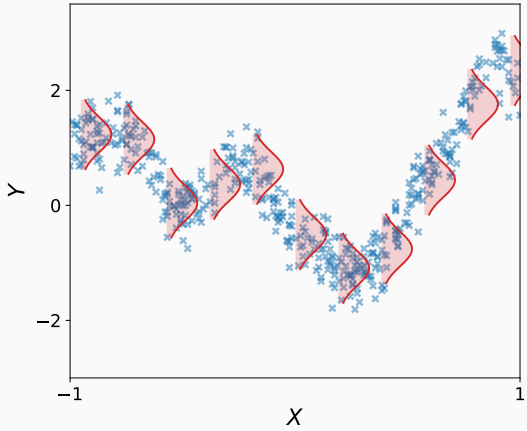
# Sparse GP models

# Sparse GPs

Sparse GPs is an approach to cope with large datasets ($10^4$ to $10^6$ points)

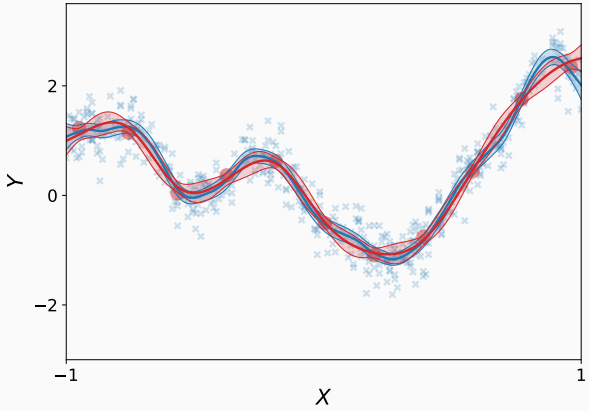


Sparse GPs replace the $n$ observations $(X, Y)$ by $m$ "pseudo-observations" $(Z, U)$ where $U \sim \mathcal{N}(\mu, \Sigma)$



The **approximate posterior** distribution is $\mathcal{GP}(m_{sparse}, c_{sparse})$ with

$$m_{sparse}(x) = k(x, Z)k(Z, Z)^{-1}\mu$$
$$c_{sparse}(x, y) = k(x, y) - k(x, Z)k(Z, Z)^{-1}k(Z, y) + k(x, Z)k(Z, Z)^{-1}\Sigma k(Z, Z)^{-1}k(Z, y)$$



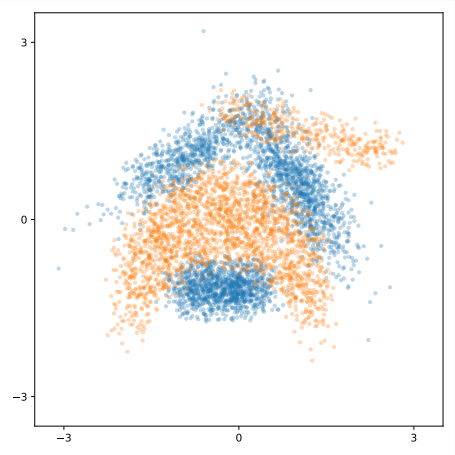
The distribution of the inducing variables $U \sim \mathcal{N}(\mu, \Sigma)$ is chosen by minimising the Kullback-Leibler divergence:

$$\min_{\mu,\Sigma} \mathcal{KL}\left( \underbrace{\int p(f(.)|f(Z) = U)dU}_{q_f} \middle| \underbrace{p(f(.)|f(X) + \varepsilon = Y)}_{p_{f|Y}} \right)$$

Computational complexity of Sparse GPs is $\mathcal{O}(nm^2 + m^3)$.

# GP ensembles

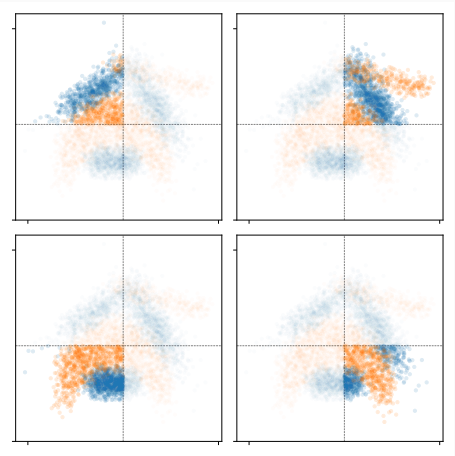
# GP ensembles in a nutshell




Three basic steps:

1. Split data into subsets
2. Train one GP model per subset
3. At prediction time, aggregate submodels posteriors

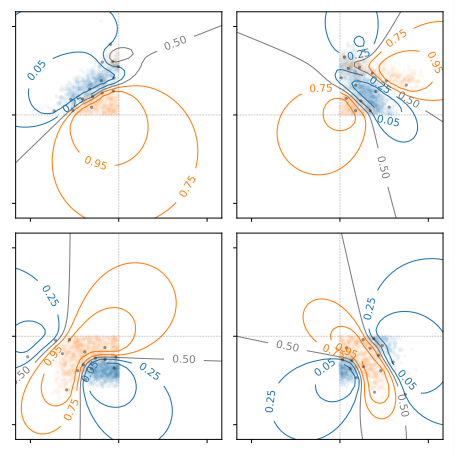
# GP ensembles in a nutshell



Three basic steps:

1. **Split data into subsets**
2. Train one GP model per subset
3. At prediction time, aggregate submodels posteriors

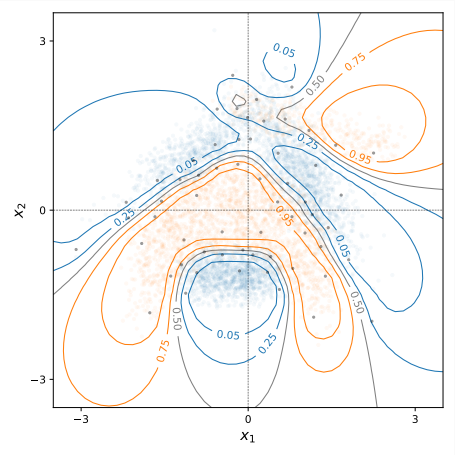
# GP ensembles in a nutshell




Three basic steps:

1. Split data into subsets
2. **Train one GP model per subset**
3. At prediction time, aggregate submodels posteriors

# GP ensembles in a nutshell




Three basic steps:

1. Split data into subsets
2. Train one GP model per subset
3. **At prediction time, aggregate submodels posteriors**

# Aggregation methods

**Historical**

- Bayesian Committee Machine [Tresp 2000]
- Product of Experts [Hinton 2002]

**Improvements**

- Generalised Product of Experts [Cao 2014]
- Robust Bayesian Committee Machine [Deisenroth 2015]
- Generalized Robust Bayesian Committee Machine [Liu 2018]*

**Others**

- Nested GPs [Rullière 2018]
- Barycentre GPs [Cohen 2020]
- Modular GPs [Moreno-Muñoz 2021]*

* Not included in our benchmarks.

# Aggregation methods

**Historical**

- **Bayesian Committee Machine [Tresp 2000]**
- Product of Experts [Hinton 2002]

**Improvements**

- Generalised Product of Experts [Cao 2014]
- Robust Bayesian Committee Machine [Deisenroth 2015]
- Generalized Robust Bayesian Committee Machine [Liu 2018]*

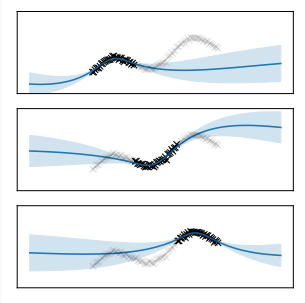
**Others**

- Nested GPs [Rullière 2018]
- Barycentre GPs [Cohen 2020]
- Modular GPs [Moreno-Muñoz 2021]*

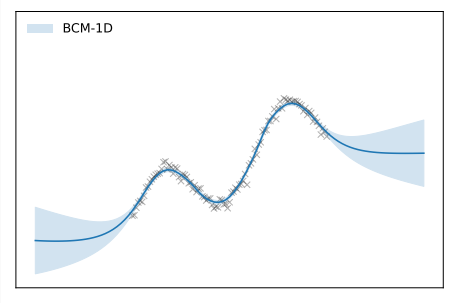
* Not included in our benchmarks.

# Bayesian Committee Machine

Given two data subsets $\mathcal{D}_i \neq \mathcal{D}_j$ and a prediction point $x^* \in X^*$, BCM makes the approximation that $\mathcal{D}_i \perp\!\!\!\perp \mathcal{D}_j | f(x^*)$.

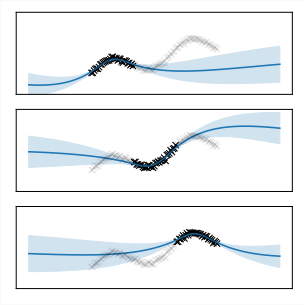


$\Rightarrow$

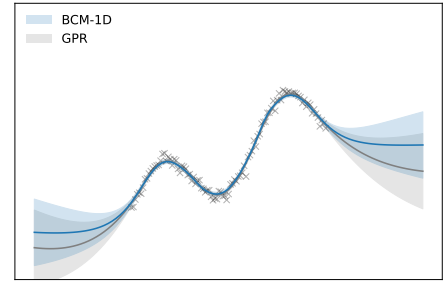


# Bayesian Committee Machine

Given two data subsets $\mathcal{D}_i \neq \mathcal{D}_j$ and a prediction point $x^* \in X^*$, BCM makes the approximation that $\mathcal{D}_i \perp\!\!\!\perp \mathcal{D}_j | f(x^*)$.



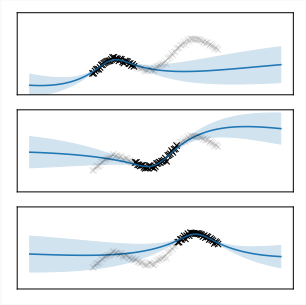
$\Rightarrow$



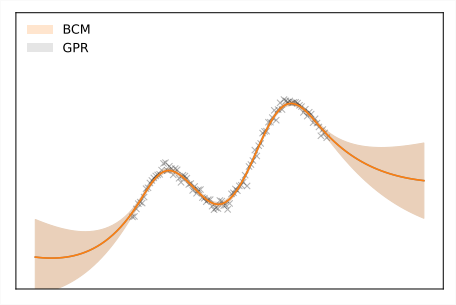

Performance is poor when predictions are made independently for each $x^*$... but this is not the original prescription!

# Bayesian Committee Machine

In Tresp [2000], test points are processed jointly so the approximation is $\mathcal{D}_i \perp\!\!\!\perp \mathcal{D}_j | f(X^*)$ which is a much weaker assumption:



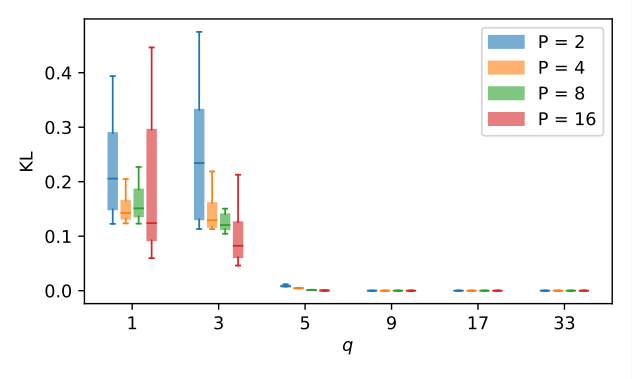
$\Rightarrow$




Prediction cost is $\mathcal{O}(n_{test}^3)$, but the ensemble predictions cannot be distinguished from GPR!

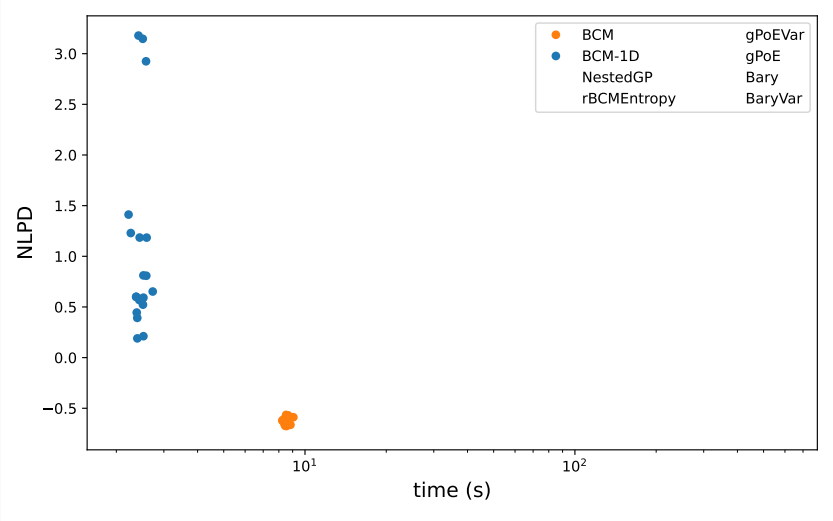
# Experimental results 1/2

In practice, increasing the size $q$ of the test set makes a big difference...



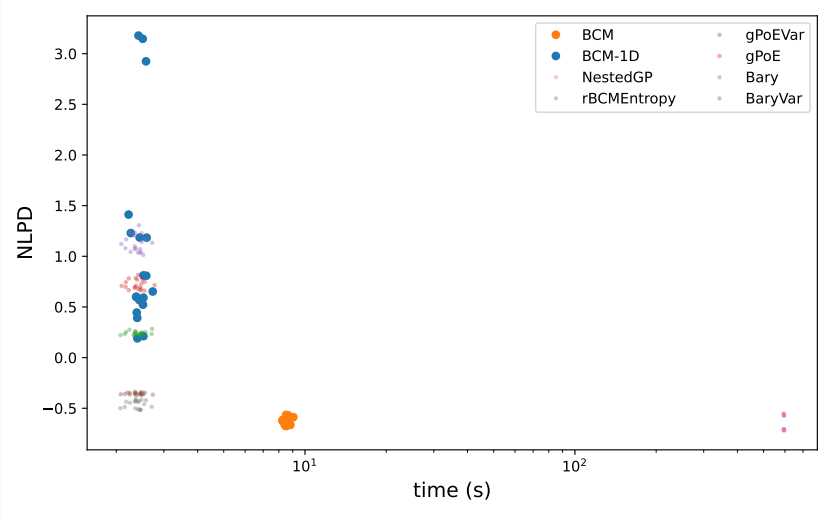
# Experimental results 2/2

20 test functions given by Matérn 5/2 GP samples: $D = [0, 1]^5$, $n_{train} = 20k$, $n_{test} = 1k$, $p = 32$, $\sigma^2 = 1$, $\theta = 0.5$, $\tau^2 = 0.01$



# Experimental results 2/2

20 test functions given by Matérn 5/2 GP samples: $D = [0, 1]^5$, $n_{train} = 20k$, $n_{test} = 1k$, $p = 32$, $\sigma^2 = 1$, $\theta = 0.5$, $\tau^2 = 0.01$



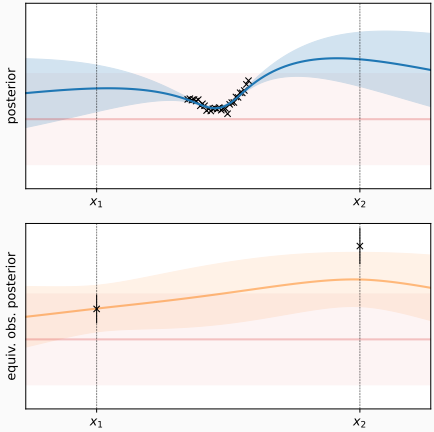
An alternative way to derive the BCM predictor is to introduce "pseudo-observations" that encapsulate the information required by the submodels to recover their prediction at $X^* \in D^q$.

More precisely, we define the equivalent observation at $X^*$ as the tuple $\left(Y^*,\ \varepsilon^*\right)$, such that

$$f(X^*)\big|\left\{f(X^*) + \varepsilon^* = Y^*\right\} \overset{dist}{=} f(X^*)\big|\{f(X) + \varepsilon = Y\},$$

In this expression, the free variables that are tuned to reach equality are $Y^*$ and the covariance matrix of $\varepsilon^*$ (say $T$).

Same explanation with a picture...




With the notation

$$f(X^*) \sim \mathcal{N}(\mu_0, \Sigma_0)$$
$$f(X^*) \big| \{f(X) + \varepsilon = Y\} \sim \mathcal{N}(\mu_1, \Sigma_1)$$

the equivalent observation is given by:

$$Y^* = \mu_0 + T\Sigma_1^{-1}(\mu_1 - \mu_0)$$
$$T = \left(\Sigma_1^{-1} - \Sigma_0^{-1}\right)^{-1}.$$

In order to use equivalent observations in an aggregation procedure, we can:

1. associate to each submodel an equivalent observation $(Y_i^*, \varepsilon_i^*)$ located at $X^*$
2. compute the values of $Y_i^*$ and $T_i$ according to previous slide
3. generate predictions at $X^*$ by conditioning the prior on all equivalent observations:

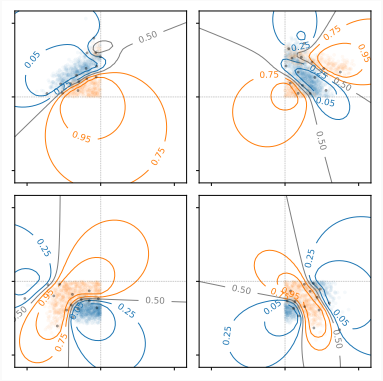
$$f(X^*) | \{f(X^*) + \varepsilon_i^* = Y_i^*\}_{i=1}^p \ .$$

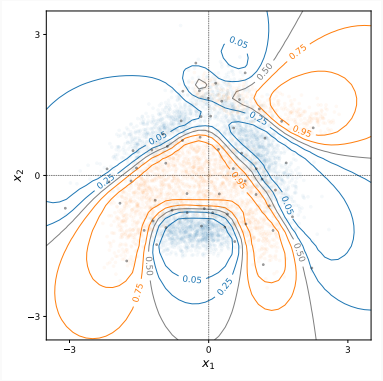
The resulting posterior is normally distributed with mean and variance

$$\mu^* = \Sigma^* \sum_{i=0}^p {T_i}^{-1} Y_i^*$$
$$\Sigma^* = \left( \sum_{i=0}^p {T_i}^{-1} \right)^{-1} .$$

# Combining BCM and Sparse GPs

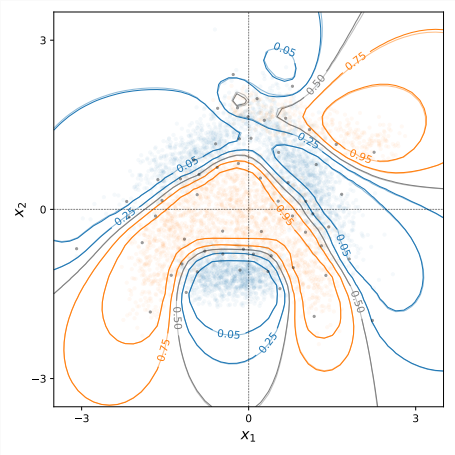
Ensembles can be used to merge the **variational distributions** of SVGPs submodels by setting $X^* = \bigcup_i Z_i$ :






With BCM, this results in a specific structure: *variational precision = prior precision + block diagonal*

In this example, training the models independently and aggregating the variational distributions drastically reduces the number of parameters to be trained (540 instead of 1890!) but yields a very good accuracy nonetheless.



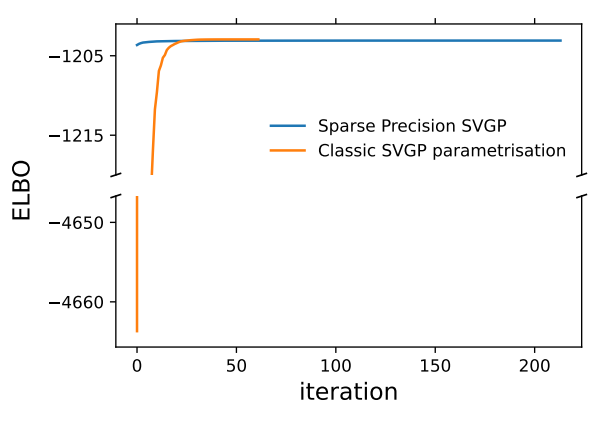
One can show that the aggregated model is equivalent to a sparse GP model with inducing variable

$$U \sim \mathcal{N}\big(K_0(K_0+T)^{-1}Y^*,\, K_0 - K_0(K_0+T^*)^{-1}K_0\big)\,.$$

$$\text{where } Y^* = \begin{pmatrix} Y^*_{Z_1} \\ \vdots \\ Y^*_{Z_p} \end{pmatrix} \qquad T^* = \begin{pmatrix} T_{Z_1} & & 0 \\ & \ddots & \\ 0 & & T_{Z_p} \end{pmatrix}.$$

Can the model be improved by retraining the ELBO of the aggregated ensemble?

Unfortunately the answer is not really!




Underlying problem: we hit the issue identified in E. Khan [2013] where parametrising SVGP in precision space results in non-convex optimisation problems...

# Conclusion

# Conclusion

## Summary

- GP ensembles are good alternatives for large datasets
- Bayesian Committee Machine works better than most people think!
- Interesting connections between ensemble methods and sparse models

For more details, see:

`www.github.com/NicolasDurrande/guepard`

# References


Cao, Y. and Fleet, D. J. (2014). Generalized product of experts for automatic and principled fusion of Gaussian process predictions. *arXiv:1410.7827*.

Cohen, S., Mbuvha, R., Marwala, T., and Deisenroth, M. P. (2020). Healing products of Gaussian process experts. In *International Conference on Machine Learning*, pages 2068–2077.

Deisenroth, M. P. and Ng, J. W. (2015). Distributed Gaussian processes. In *International Conference on Machine Learning*, pages 1481–1490.

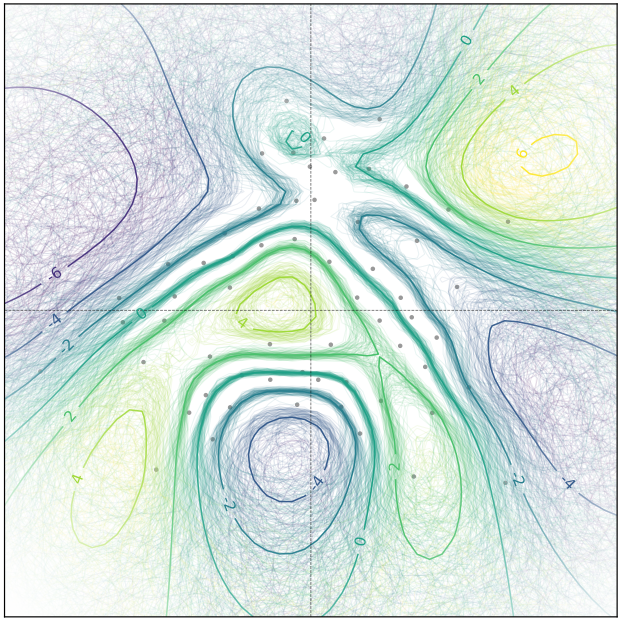
Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural computation*, pages 1771–1800.

Khan, M. E., Aravkin, A., Friedlander, M., and Seeger, M. (2013). Fast dual variational inference for non-conjugate latent Gaussian models. In *International Conference on Machine Learning*, pages 951–959.

Liu, H., Cai, J., Wang, Y., and Ong, Y. S. (2018). Generalized robust Bayesian committee machine for large-scale Gaussian process regression. In *International Conference on Machine Learning*, pages 3131–3140.

Moreno-Muñoz, P., Artés, A., and Álvarez, M. (2021). Modular Gaussian processes for transfer learning. *Advances in Neural Information Processing Systems*.

Rullière, D., Durrande, N., Bachoc, F., and Chevalier, C. (2018). Nested kriging predictions for datasets with a large number of observations. *Statistics and Computing*, pages 849–867.


0
2
4
6
-6
-4
-2
0
2
4
-4
-2
0
2
4
-4
-2
0
2
4
6