Sparse Multiscale Gaussian Process Regression

Christian Walder, Kwang In Kim, Bernhard Schölkopf

Max Planck Institute for Biological Cybernetics

July 7, 2008, Helsinki.

Outline

Basis functions of the form $k(x, \cdot)$

A Brief History of Sparse GP Algorithms

- The posterior mean of the GP takes the form

$$\mu(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i k(\boldsymbol{x}_i, \boldsymbol{x})$$

- The simplest sparse approximation enforces

$$\mu_{\text{sparse}}(\boldsymbol{x}) = \sum_{i \in \mathcal{S}} \beta_i k(\boldsymbol{x}_i, \boldsymbol{x}), \quad \mathcal{S} \subset \{1, 2, \ldots n\}$$

- Alternatively one may sacrifice training time for testing time:

$$\mu_{\text{sparser}}(\boldsymbol{x}) = \sum_{i=1}^{m} \gamma_i k(\boldsymbol{z}_i, \boldsymbol{x})$$

Basis functions NOT of the form $k(x, \cdot)$

- The next logical step to obtain greater sparsity enforces

$$\mu_{\text{even more sparse}}(\boldsymbol{x}) = \sum_{i=1}^{m} c_i u_i(\boldsymbol{x})$$

  where the u_i belong to some prescribed set of basis functions
- This generalizes the previous approximations which set, *e.g.*

$$u_i(\boldsymbol{x}) = k(\boldsymbol{z}_i, \boldsymbol{x})$$

- This was done by Walder *et al.* [1] using compactly supported basis functions (translated and dilated B_3-splines)
- Gehler and Franz [2] used $u_i(\boldsymbol{x}) = (\boldsymbol{x}^\top \boldsymbol{x}_i)^p$
- Presently we consider dilated/translated Gaussian basis functions for u_i

Prior probability of arbitrary Gaussian mixtures

Via an infinite limit

- Let $\mathcal{G}(k)$ be the zero mean GP with covariance k
- Let u be drawn from $\mathcal{G}(k)$ and define the random variable

$$\boldsymbol{u}_{\mathcal{X}} = \begin{pmatrix} u(\boldsymbol{x}_1) & u(\boldsymbol{x}_2) & \cdots & u(\boldsymbol{x}_n) \end{pmatrix}^\top$$

- Let $K_{xx} = (k(\boldsymbol{x}_i, \boldsymbol{x}_j))_{ij}$

By the definition of a GP we have

$$p_{\boldsymbol{u}_{\mathcal{X}}}\left(\sum_{i=1}^m c_i \boldsymbol{u}_i\right) = \left|2\pi K_{xx}^{-1}\right|^{-\frac{1}{2}} \exp\left(-\tfrac{1}{2}\sum_{i,j=1}^m c_i c_j \boldsymbol{u}_i^\top K_{xx}^{-1} \boldsymbol{u}_j\right).$$

We take $n \to \infty$ with uniformly distributed $\boldsymbol{x}_i$ so that

- $\boldsymbol{u}_i$ becomes a function u_i
- $\boldsymbol{u}_i^\top K_{xx}^{-1} \boldsymbol{u}_j$ becomes

$$\int\int k^{-1}(\boldsymbol{x}, \boldsymbol{y}) u_i(\boldsymbol{x}) u_j(\boldsymbol{y}) \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{y}.$$

Probability of arbitrary Gaussian mixtures (2)

But what is $k^{-1}(\cdot,\cdot)$?

- For finite n if we let $\boldsymbol{u} = K_{xx}\boldsymbol{\alpha}$ then $\boldsymbol{\alpha} = K_{xx}^{-1}\boldsymbol{u}$.
- Following this finite analogy, if $u = \int \alpha(\boldsymbol{x})k(\boldsymbol{x},\cdot)\,\mathrm{d}\boldsymbol{x}$, then k^{-1} should satisfy

$$\int u(\boldsymbol{x})k^{-1}(\boldsymbol{x},\cdot)\,\mathrm{d}\boldsymbol{x} = \alpha(\cdot).$$

Hence if we define

$$M_k : \alpha \mapsto M_k\alpha = \int \alpha(\boldsymbol{x})k(\boldsymbol{x},\cdot)\,\mathrm{d}\boldsymbol{x},$$

then k^{-1} is by definition the Green's function of M_k, as it satisfies

$$\int \big(M_k\alpha\big)(\boldsymbol{x})k^{-1}(\boldsymbol{x},\cdot)\,\mathrm{d}\boldsymbol{x} = \alpha(\cdot).$$

Prior probability of arbitrary Gaussian mixtures (3)

Now define g to be the normalised Gaussian on $\mathbb{R}^d \times \mathbb{R}^d$,

$$g(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\sigma}) \equiv |2\pi \mathrm{diag}\,(\boldsymbol{\sigma})|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\sum_{i=1}^{d} \frac{([\boldsymbol{x}-\boldsymbol{y}]_i)^2}{[\boldsymbol{\sigma}]_i}\right).$$

If we choose

- $k(\boldsymbol{x}, \boldsymbol{y}) = c g(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\sigma})$, where $c > 0$, $\boldsymbol{\sigma} > \mathbf{0} \in \mathbb{R}^d$
- $u_i(\boldsymbol{x}) = g(\boldsymbol{x}, \boldsymbol{v}_i, \boldsymbol{\sigma}_i)$

then everything is Gaussian and we obtain in closed form

$$\int\int k^{-1}(\boldsymbol{x}, \boldsymbol{y}) u_i(\boldsymbol{x}) u_j(\boldsymbol{y})\, \mathrm{d}\boldsymbol{x}\, \mathrm{d}\boldsymbol{y} = \frac{1}{c}\, g(\boldsymbol{v}_i, \boldsymbol{v}_j, \boldsymbol{\sigma}_i + \boldsymbol{\sigma}_j - \boldsymbol{\sigma}).$$

Prior probability of arbitrary Gaussian mixtures (4)

Summarising Expressions

GP	$$p_{\mathcal{G}(cg(\cdot,\cdot,\boldsymbol{\sigma}))}\left(\sum_{i=1}^{m} c_i g(\cdot, \boldsymbol{v}_i, \boldsymbol{\sigma}_i)\right)$$ $$\propto \exp\left(-\frac{1}{2}\sum_{i,j=1}^{m}\frac{1}{c}c_i c_j g(\boldsymbol{v}_i, \boldsymbol{v}_j, \boldsymbol{\sigma}_i + \boldsymbol{\sigma}_j - \boldsymbol{\sigma})\right).$$
RKHS	Let $\mathcal{H}$ be the RKHS with kernel $g(\cdot,\cdot,\boldsymbol{\sigma})$. Then $$\langle g(\cdot, \boldsymbol{v}_i, \boldsymbol{\sigma}_i), g(\cdot, \boldsymbol{v}_j, \boldsymbol{\sigma}_j)\rangle_{\mathcal{H}} = g(\boldsymbol{v}_i, \boldsymbol{v}_j, \boldsymbol{\sigma}_i + \boldsymbol{\sigma}_j - \boldsymbol{\sigma}).$$

Prior probability of arbitrary Gaussian mixtures (5)

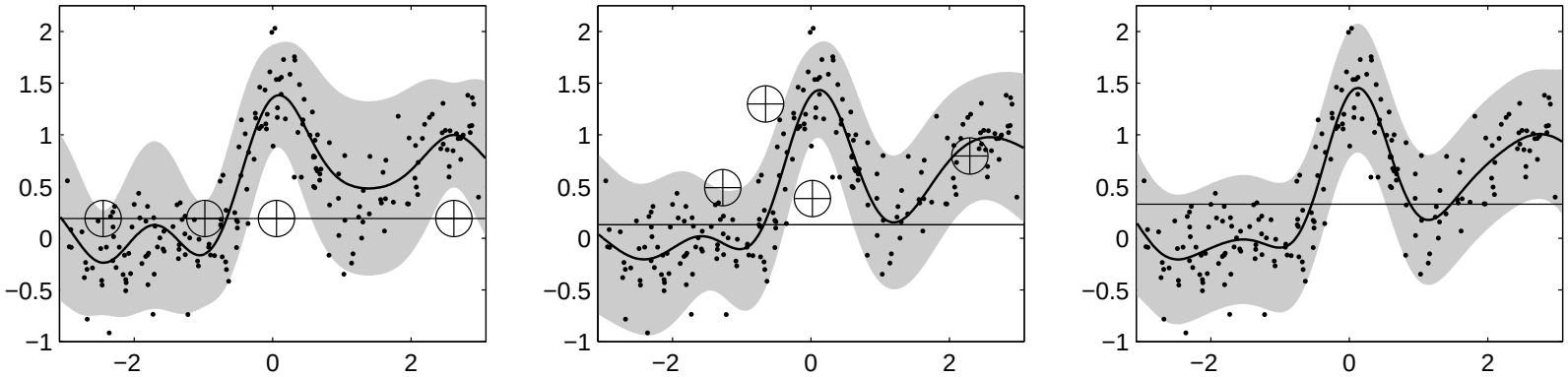
Interpretation

- $\boldsymbol{\sigma}_1 = \boldsymbol{\sigma}_2 = \cdots = \boldsymbol{\sigma}_n = \boldsymbol{\sigma}$ recovers the "normal" methods.
- The most likely single Gaussian function u_1 has $\boldsymbol{\sigma}_1 = \boldsymbol{\sigma}$.
- As the dimension increases, the probability density in $\boldsymbol{\sigma}_1$ centres around $\boldsymbol{\sigma}$ (so smaller dimensions yield greater gains)
- As noted by Bach and Jordan [3], for all $j = 1, 2, \ldots, d$

$$\lim_{[\boldsymbol{\sigma}_1]_j \to \left(\frac{1}{2}[\boldsymbol{\sigma}]_j\right)^+} p_{\mathcal{G}(g(\cdot,\cdot,\boldsymbol{\sigma}))}\big(u_1(\cdot)\big) = 0.$$

Kernel machines such as the GP cannot recover "any" function — in fact not even a Gaussian function that is too narrow!

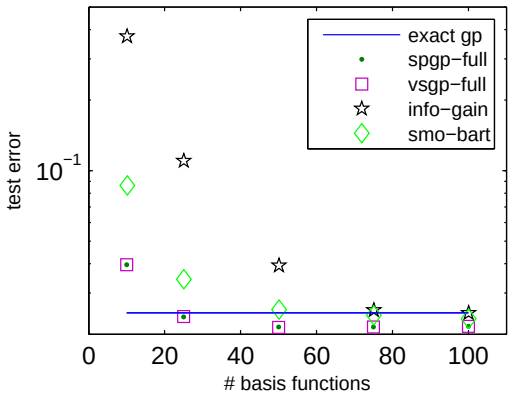
One Dimensional Toy Example



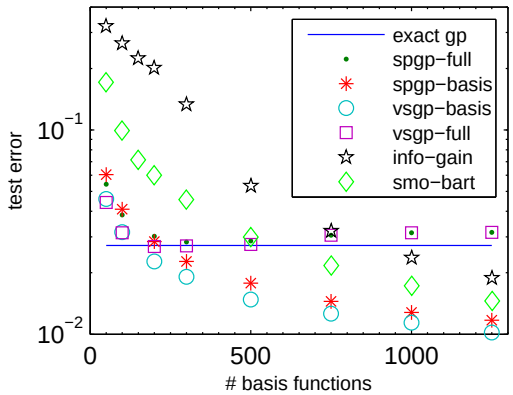
(a) Basis $\boldsymbol{\sigma}_i$'s fixed to $\boldsymbol{\sigma}$. (b) Basis $\boldsymbol{\sigma}_i$'s variable. (c) Exact g.p.

- Predictive distributions
  (mean curve with $\pm$ two standard deviations shaded)
- For the sparse algorithms, we plot the crossed circles at the $(\boldsymbol{v}_i, \boldsymbol{\sigma}_i) \in \mathbb{R} \times \mathbb{R}$
- The horizontal lines denote the resulting $\boldsymbol{\sigma} \in \mathbb{R}$ of the covariance function $cg(\cdot, \cdot, \boldsymbol{\sigma})$

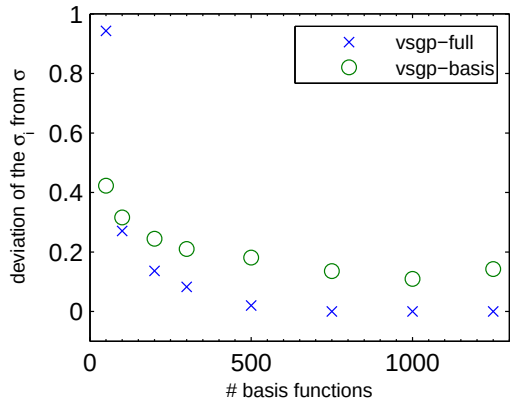
Real World Examples



(d) pumadyn-32nm error



(e) kin-40K error



(f) kin-40K $\boldsymbol{\sigma}_i$ deviation

- *kin-40k*: 10000 training, 30000 test, 9 attributes, see `www.igi.tugraz.at/aschwaig/data.html`
- *pumadyn-32nm*: 7168 training, 1024 test, 33 attributes, see `www.cs.toronto/delve`
- On the right we plot $\frac{1}{md}\sum_{i=1}^{m}\sum_{j=1}^{d}([\boldsymbol{\sigma}_i - \boldsymbol{\sigma}]_j)^2$

This Can be Applied to any Kernel Machine

- Let $\mathcal{H}$ be the RKHS with kernel $g(\cdot, \cdot, \boldsymbol{\sigma})$.
- We can always "multi-scale sparsify" the solution to

$$\arg\min_{f \in \mathcal{H}} \|f\|_{\mathcal{H}}^2 + \text{risk}$$

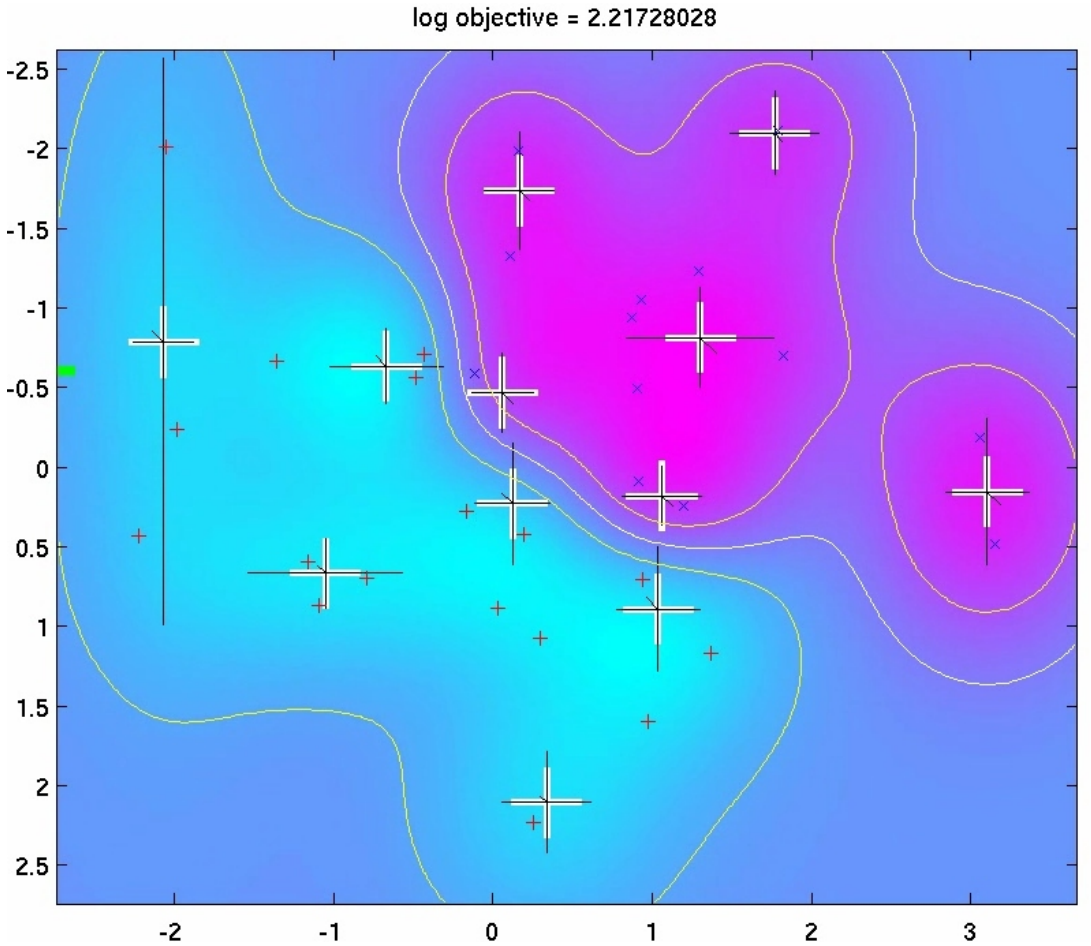
By enforcing the solution to take the form $\sum_{i=1}^{m} c_i g(\boldsymbol{v}_i, \cdot, \boldsymbol{\sigma}_i)$. The objective becomes

$$\arg \min_{c_i \in \mathbb{R}, \boldsymbol{v}_i \in \mathbb{R}^d, \boldsymbol{\sigma}_i \in \mathbb{R}^d} \boldsymbol{c}^\top U_\Psi \boldsymbol{c} + \text{risk}$$

where $U_\Psi = \left(g(\boldsymbol{v}_i, \boldsymbol{v}_j, \boldsymbol{\sigma}_i + \boldsymbol{\sigma}_j - \boldsymbol{\sigma})\right)_{i,j}$ as before.

A Video of the Optimisation Process

SVM Classifier in Two Dimensions



Summary and Outlook

- We have seen how to approximate Gaussian kernel machines using a multi scale Gaussian function basis.
- The result is a generalization of previous sparsification methods, and hence is at least as good as them.
- The method is suitable for obtaining very short *test* times, desireable in real-time applications, for example.
- The lower the input dimensionality, the more benefit the multi-scale basis can provide.
- A similar analysis could be done for other combinations of basis functions and kernels.

References


Christian Walder, Bernhard Schölkopf, and Olivier Chapelle.
Implicit surfaces with globally regularised and compactly supported basis functions.
In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 273–280, Cambridge, MA, 2007. MIT Press.


Peter Gehler and Matthias Franz.
Implicit wiener series, part ii: Regularised estimation.
Technical Report 148, Max Planck Institute for Biological Cybernetics, November 2006.


Francis R. Bach and Michael I. Jordan.
Kernel independent component analysis.
Technical Report UCB/CSD-01-1166, EECS Department, University of California, Berkeley, Nov 2001.


Edward Snelson and Zoubin Ghahramani.
Sparse gaussian processes using pseudo-inputs.
In *NIPS 18*, pages 1257–1264. MIT Press, Cambridge, MA, 2006.


J. Quiñonero-Candela and C. E. Rasmussen.
A unifying view of sparse approximate gaussian process regression.
Journal of Machine Learning Research, 6:1935–1959, 12 2005.


C. E. Rasmussen and C. K.I. Williams.
Gaussian Processes for Machine Learning.
The MIT Press, Cambridge, Massachusetts, 01 2006.