Sparse Parametric Gaussian Processes

Ed Snelson and Zoubin Ghahramani
{snelson,zoubin}@gatsby.ucl.ac.uk

Sheffield GP Workshop

9th June 2005

Problems with many sparse GP regression methods

1. Restricted to choosing **active set points** from amongst training data

2. Lack a reliable way to find **kernel hyperparameters**

1. Restriction of active set to training data

- Most sparse GP methods use some kind of **information criterion** for **selecting data points** to include into an active set
- Many methods do not *explicitly* restrict the active set to be selected from data
- However in practice there are not obvious ways in which to choose active set **points from outside the data set**
- **SPGP chooses points by gradient descent** on a suitable cost function

2. Learning kernel hyperparameters

- **Active set selection interferes with hyperparameter learning**

- Reselecting active set causes non-smooth fluctuations in the marginal likelihood and its gradients

- Cannot get smooth convergence

- **SPGP learns hyperparameters together with active set points** in one joint gradient optimization

GP Notation

N input vectors $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^N$ — dimension D

latents $\mathbf{u} = \{u_n\}_{n=1}^N$, targets $\mathbf{y} = \{y_n\}_{n=1}^N$, noise σ^2

covariance $[\mathbf{K}_N]_{nn'} = K(\mathbf{x}_n, \mathbf{x}_{n'})$, hyperparameters $\boldsymbol{\theta}$

marginal likelihood: $p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{K}_N + \sigma^2\mathbf{I})$

predictive distribution:

$$p(y|\mathbf{x}, \mathcal{D}, \boldsymbol{\theta}) = \mathcal{N}\big(y\big|\mathbf{k}_\mathbf{x}^\top(\mathbf{K}_N + \sigma^2\mathbf{I})^{-1}\mathbf{y},\ K_{\mathbf{xx}} - \mathbf{k}_\mathbf{x}^\top(\mathbf{K}_N + \sigma^2\mathbf{I})^{-1}\mathbf{k}_\mathbf{x} + \sigma^2\big)$$

where $[\mathbf{k}_\mathbf{x}]_n = K(\mathbf{x}_n, \mathbf{x})$ and $K_{\mathbf{xx}} = K(\mathbf{x}, \mathbf{x})$

Parametric Gaussian processes

- **GP predictive distribution effectively parameterised** by training data point locations
- **Consider a *parametric* model** with likelihood given by GP predictive distribution
- Parameterised by **pseudo data set** of M fake observations: pseudo inputs $\bar{\mathbf{X}} = \{\bar{\mathbf{x}}_m\}_{m=1}^M$, pseudo targets $\bar{\mathbf{u}} = \{\bar{u}_m\}_{m=1}^M$

Single data point likelihood:

$$p(y|\mathbf{x}, \bar{\mathbf{X}}, \bar{\mathbf{u}}) = \mathcal{N}\big(y\big|\mathbf{k}_\mathbf{x}^\top \mathbf{K}_M^{-1}\bar{\mathbf{u}},\ K_{\mathbf{xx}} - \mathbf{k}_\mathbf{x}^\top \mathbf{K}_M^{-1}\mathbf{k}_\mathbf{x} + \sigma^2\big)$$

where $[\mathbf{K}_M]_{mm'} = K(\bar{\mathbf{x}}_m, \bar{\mathbf{x}}_{m'})$ and $[\mathbf{k}_\mathbf{x}]_m = K(\bar{\mathbf{x}}_m, \mathbf{x})$, for $m = 1, \ldots, M$

Likelihood and prior

Target data — i.i.d. given inputs:

$$p(\mathbf{y}|\mathbf{X}, \bar{\mathbf{X}}, \bar{\mathbf{u}}) = \prod_{n=1}^{N} p(y_n|\mathbf{x}_n, \bar{\mathbf{X}}, \bar{\mathbf{u}}) = \mathcal{N}(\mathbf{y}|\mathbf{K}_{MN}^{\top}\mathbf{K}_M^{-1}\bar{\mathbf{u}},\ \mathbf{\Lambda}_N)$$

where $\mathbf{\Lambda}_N = \mathrm{diag}(\boldsymbol{\lambda})$, $\lambda_n = K_{nn} - \mathbf{k}_n^{\top}\mathbf{K}_M^{-1}\mathbf{k}_n + \sigma^2$, and $[\mathbf{K}_{MN}]_{mn} = K(\bar{\mathbf{x}}_m, \mathbf{x}_n)$.

Learning involves finding a suitable pseudo data set. However we can **integrate out the pseudo targets $\bar{\mathbf{u}}$**.

Gaussian prior:

$$p(\bar{\mathbf{u}}|\bar{\mathbf{X}}) = \mathcal{N}(\bar{\mathbf{u}}|\mathbf{0}, \mathbf{K}_M)$$

Posterior and predictive distributions

Consider pseudo inputs known for now. Bayes rule gives the posterior:

$$p(\bar{\mathbf{u}}|\mathcal{D},\bar{\mathbf{X}}) = \mathcal{N}\big(\bar{\mathbf{u}}|\mathbf{K}_M\mathbf{Q}_M^{-1}\mathbf{K}_{MN}\mathbf{\Lambda}_N^{-1}\mathbf{y},\ \mathbf{K}_M\mathbf{Q}_M^{-1}\mathbf{K}_M\big)$$

where $\mathbf{Q}_M = \mathbf{K}_M + \mathbf{K}_{MN}\mathbf{\Lambda}_N^{-1}\mathbf{K}_{MN}^\top$.

New input $\mathbf{x}_*$ — predictive distribution:

$$p(y_*|\mathbf{x}_*,\mathcal{D},\bar{\mathbf{X}}) = \int \mathrm{d}\bar{\mathbf{u}}\, p(y_*|\mathbf{x}_*,\bar{\mathbf{X}},\bar{\mathbf{u}})\, p(\bar{\mathbf{u}}|\mathcal{D},\bar{\mathbf{X}}) = \mathcal{N}(y_*|\mu_*,\sigma_*^2)$$

where

$$\mu_* = \mathbf{k}_*^\top\mathbf{Q}_M^{-1}\mathbf{K}_{MN}\mathbf{\Lambda}_N^{-1}\mathbf{y}$$

$$\sigma_*^2 = K_{**} - \mathbf{k}_*^\top(\mathbf{K}_M^{-1} - \mathbf{Q}_M^{-1})\mathbf{k}_* + \sigma^2$$

After precomputations, $\mathcal{O}(M)$ for mean, $\mathcal{O}(M^2)$ for variance per test case

Marginal likelihood

How to find pseudo input locations $\bar{\mathbf{X}}$ and hyperparameters $\boldsymbol{\Theta} = \{\boldsymbol{\theta}, \sigma^2\}$?

Maximize marginal likelihood by gradient ascent:

$$p(\mathbf{y}|\mathbf{X}, \bar{\mathbf{X}}, \boldsymbol{\Theta}) = \int \mathrm{d}\bar{\mathbf{u}}\, p(\mathbf{y}|\mathbf{X}, \bar{\mathbf{X}}, \bar{\mathbf{u}})\, p(\bar{\mathbf{u}}|\bar{\mathbf{X}})$$
$$= \mathcal{N}(\mathbf{y}|\mathbf{0},\ \mathbf{K}_{MN}^\top \mathbf{K}_M^{-1} \mathbf{K}_{MN} + \boldsymbol{\Lambda}_N)$$

Gradient calculations long and tedious! Closely follow Seeger et al. (2003)

Overfitting?

- $MD + |\mathbf{\Theta}|$ parameters instead of $|\mathbf{\Theta}|$
- Sensible nature of **noise model prevents overfitting**
- **Consider** $M = N$. A marginal likelihood maximum occurs when $\bar{\mathbf{X}} = \mathbf{X}$.
  - Here $\mathbf{K}_{MN} = \mathbf{K}_M = \mathbf{K}_N$, $\mathbf{\Lambda}_N = \sigma^2\mathbf{I}$, and SPGP and full GP marginal likelihoods and predictive distributions coincide
  - Gives confidence in solution for $M < N$

Relations to other methods

- Closely related to Csató and Opper (2002), also Seeger et al. (2003): *projected latent variables* (PLV) method

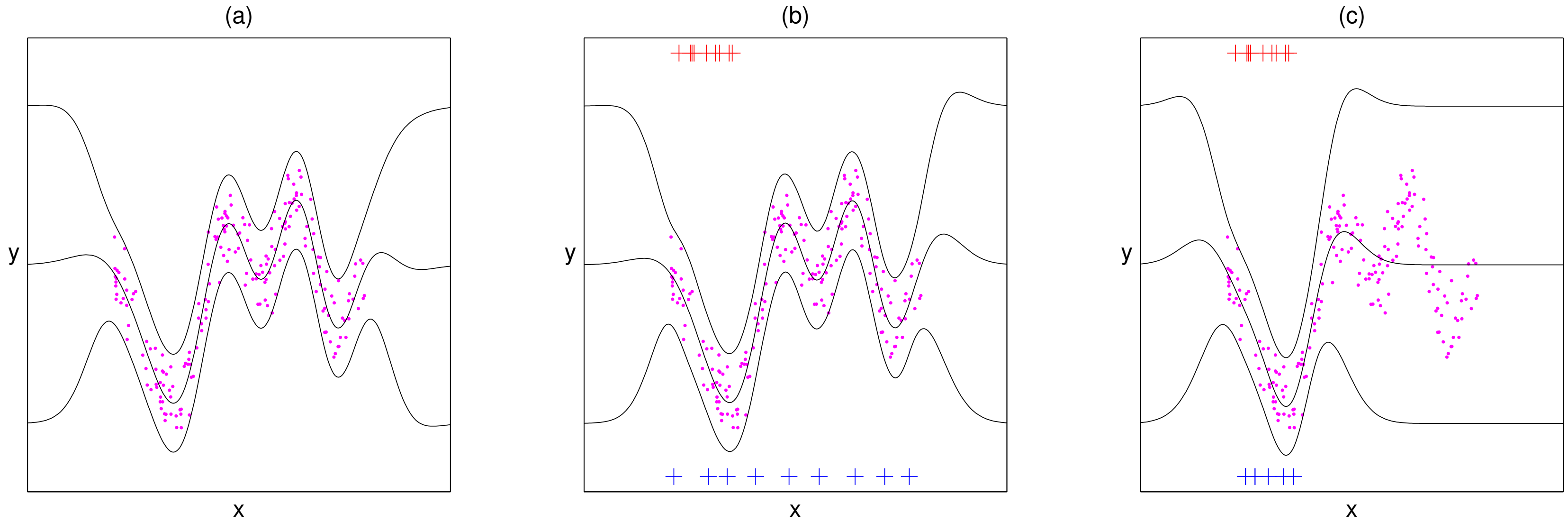
- Replace $\mathbf{\Lambda}_N$ with $\sigma^2\mathbf{I}$ and we get exactly their expressions for predictive distribution and marginal likelihood

- **PLV marginal likelihood**:
  $p(\mathbf{y}|\mathbf{X}, \bar{\mathbf{X}}, \mathbf{\Theta}) = \mathcal{N}(\mathbf{y}|\mathbf{0},\ \mathbf{K}_{MN}^\top \mathbf{K}_M^{-1} \mathbf{K}_{MN} + \sigma^2\mathbf{I})$

- **Major difference** – we select pseudo inputs by gradient ascent

- What happens if we try to use PLV likelihood instead for learning pseudo input locations by gradients?

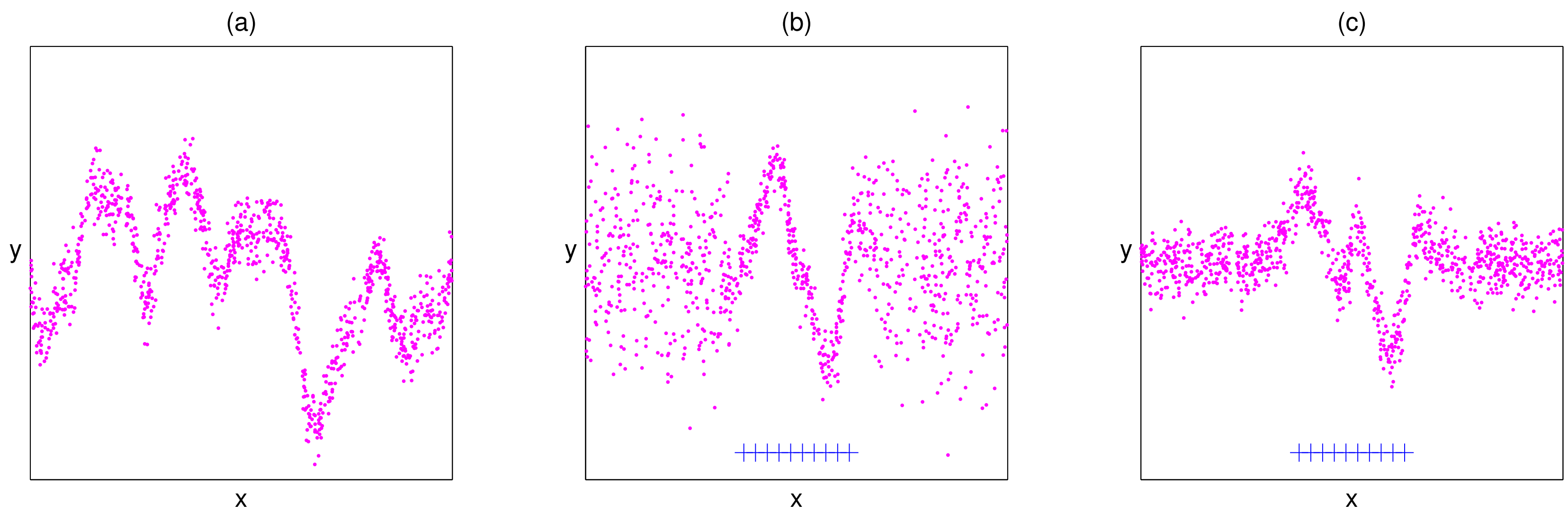
1D (adversarial!) demo



Predictive distributions for: (a) full GP, (b) gradient ascent on SPGP likelihood, (c) gradient ascent on PLV likelihood.

Initial pseudo point positions — red crosses
Final pseudo point positions — blue crosses

Samples from marginal likelihoods



Sample data drawn from the marginal likelihood of: (a) a full GP, (b) SPGP, (c) PLV.

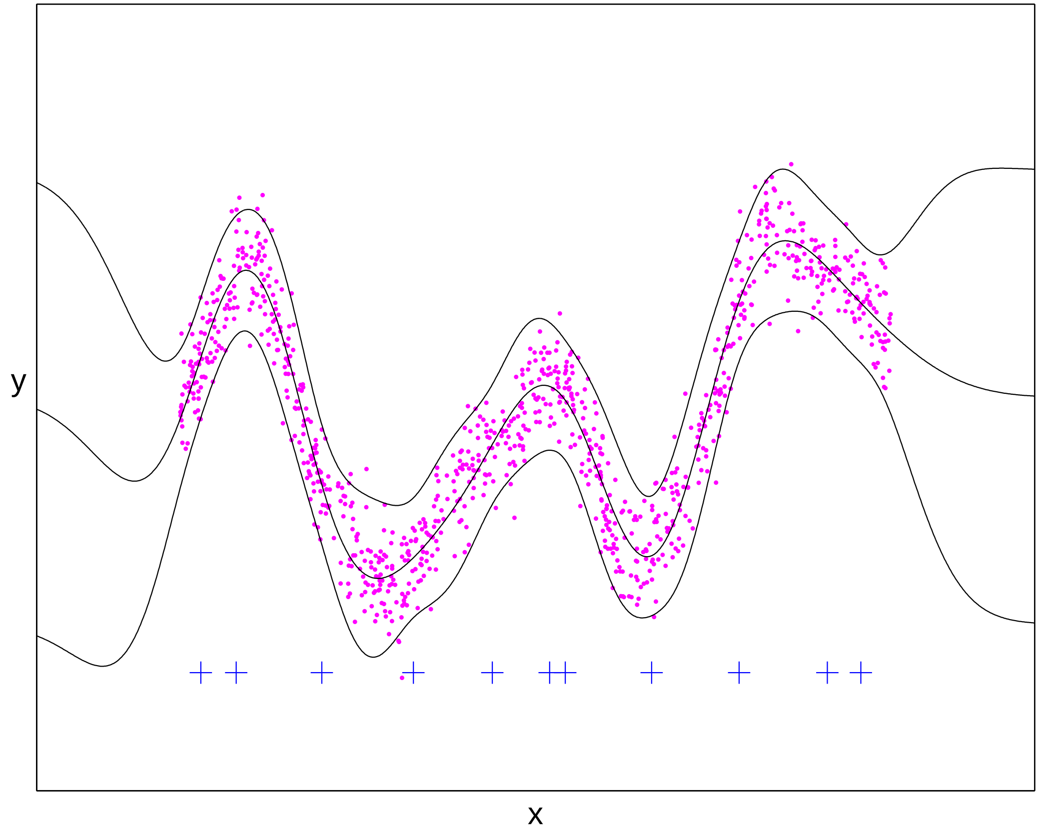
10 pseudo input points — blue crosses

Away from pseudo data points, PLV noise $= \sigma^2$, SPGP noise $\rightarrow K_{nn} + \sigma^2$

Which likelihood?

- The ***global*** optimum of the PLV likelihood may well be a good solution, but it is going to be difficult to find with ***gradients***
- The SPGP likelihood also suffers from local optima, but not so seriously
- The two likelihoods are very similar if the pseudo points are in **'good' locations**
- They differ significantly when the pseudo points are in **'poor' locations**
- Which is better for hyperparameter selection?

Successful determination of hyperparameters in 1D



Experiments

Two data sets, as tested in Seeger et al. (2003):

kin-40k: 10000 training, 30000 test, 9 attributes

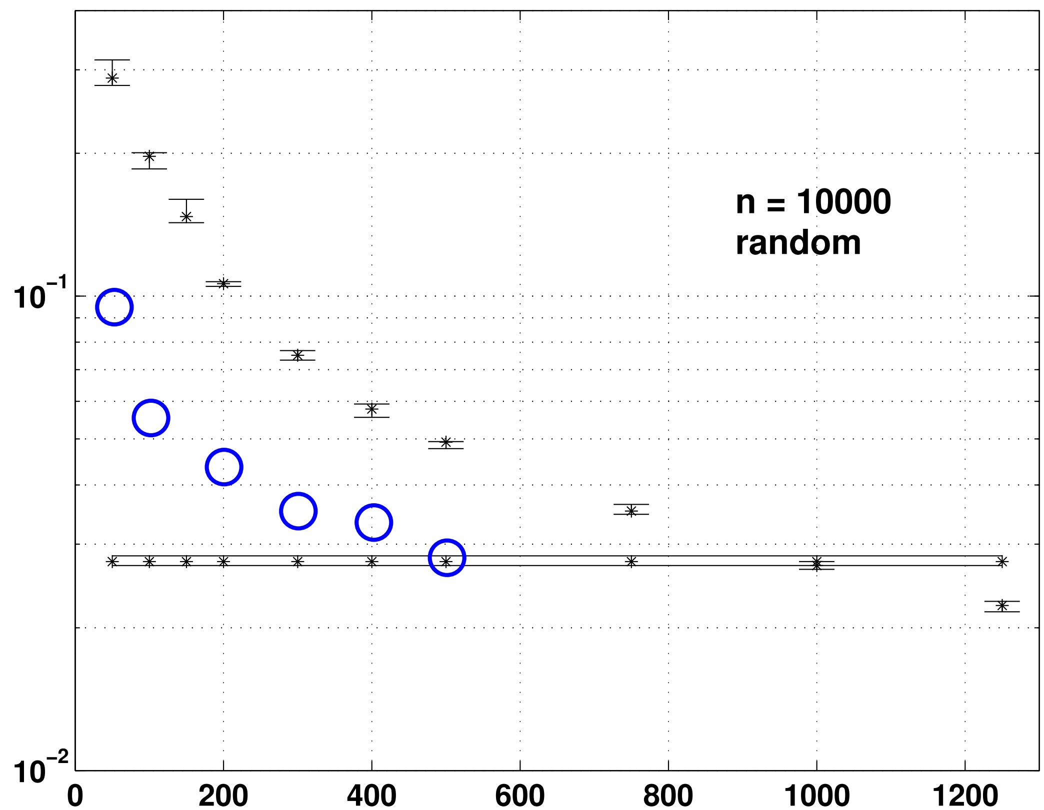
pumadyn-32nm: 7168 training, 1024 test, 33 attributes

Plot test mean squared error as function of active/pseudo set size M

Compare to 3 sparse methods: random active set selection, Seeger's greedy selection, and Smola and Bartlett's greedy selection

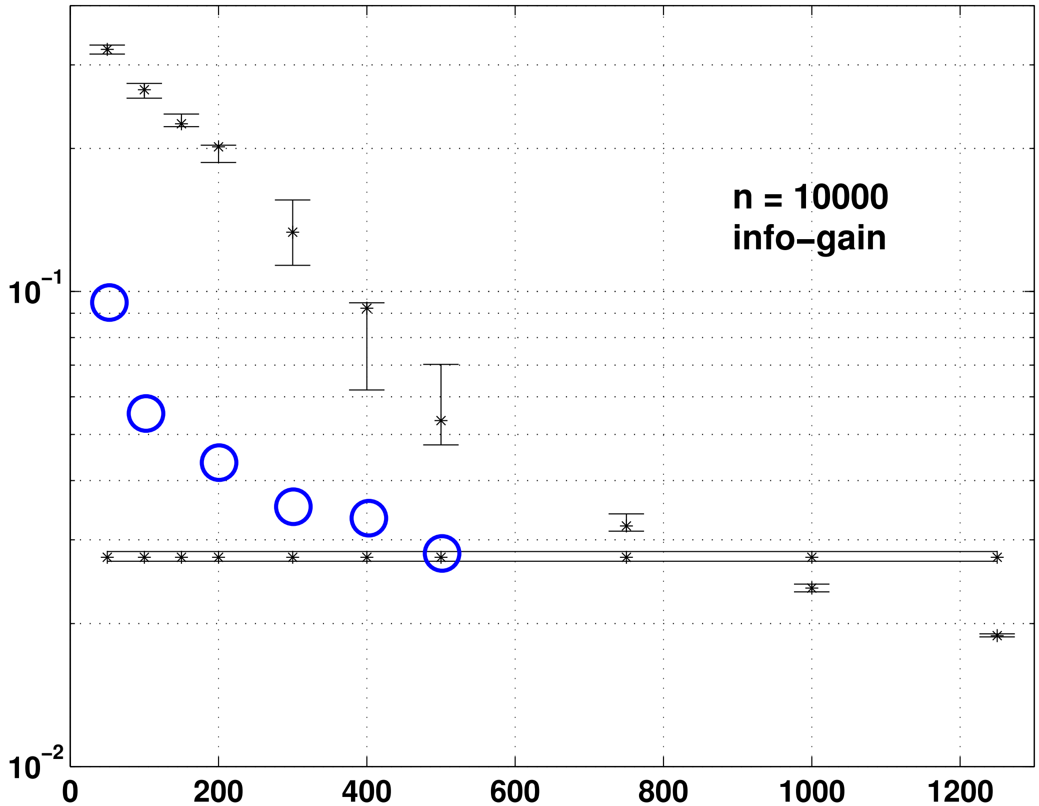
Also full GP trained on large subset of data

kin40k — SPGP and random



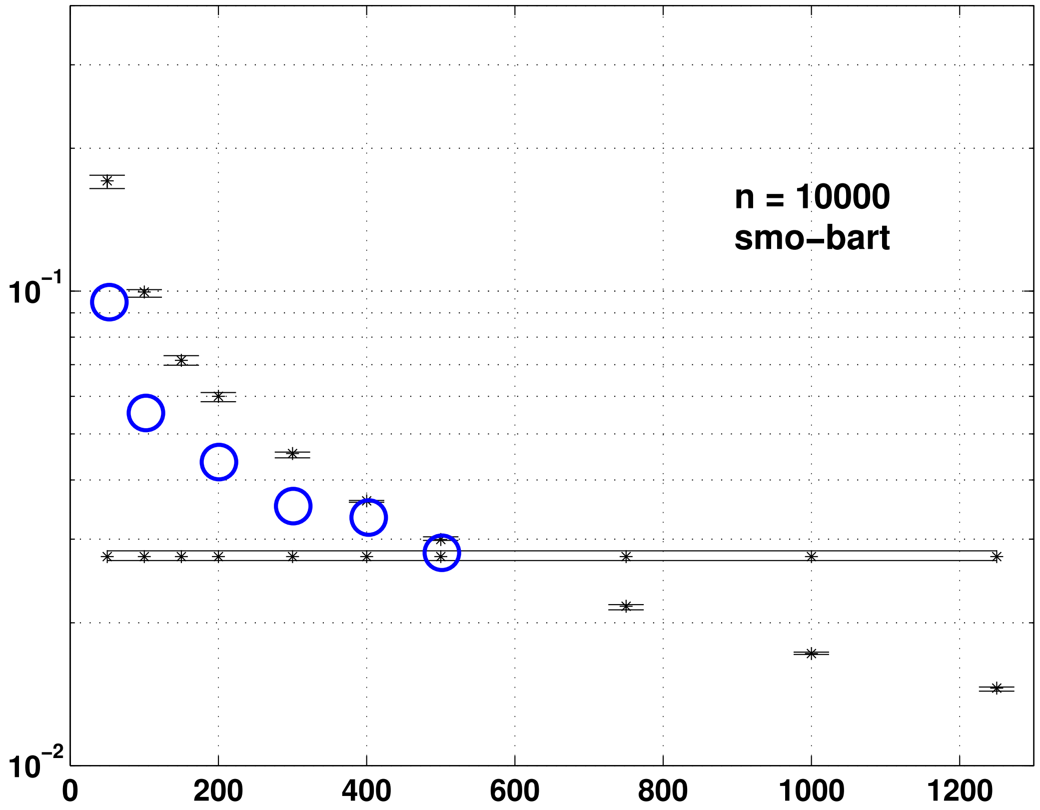
blue circles – SPGP, black – random
horizontal line – full GP on subset

kin40k — SPGP and info-gain



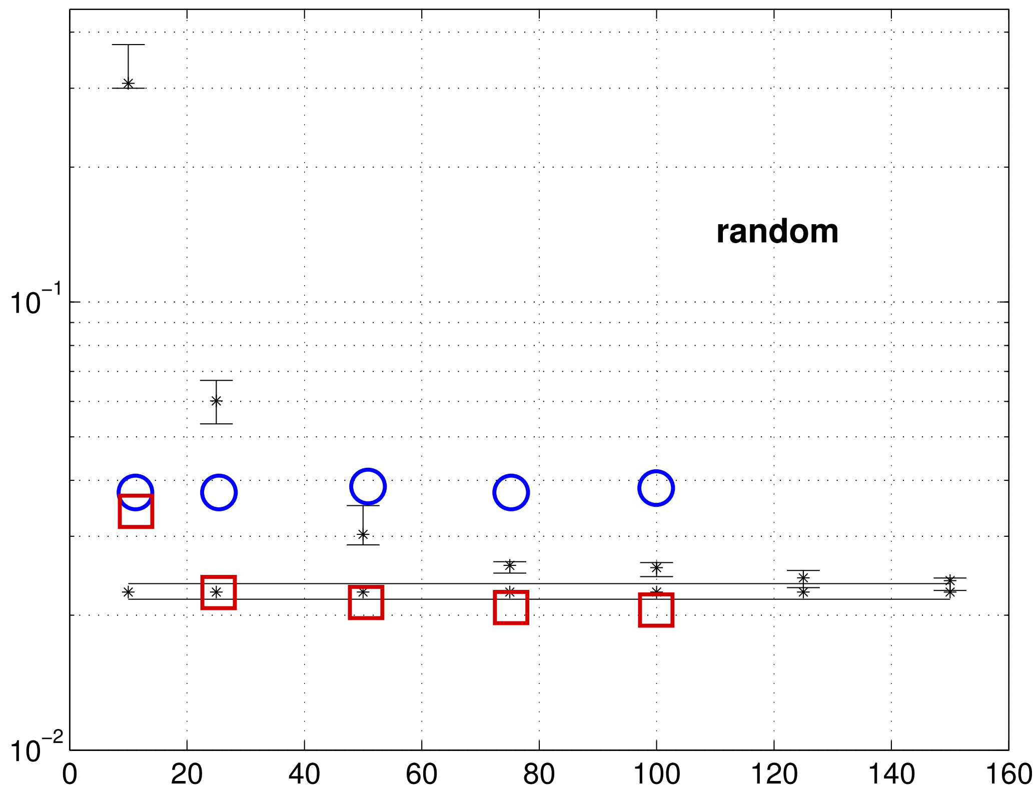
blue circles – SPGP, black – info-gain
horizontal line – full GP on subset

kin40k — SPGP and Smo-Bart



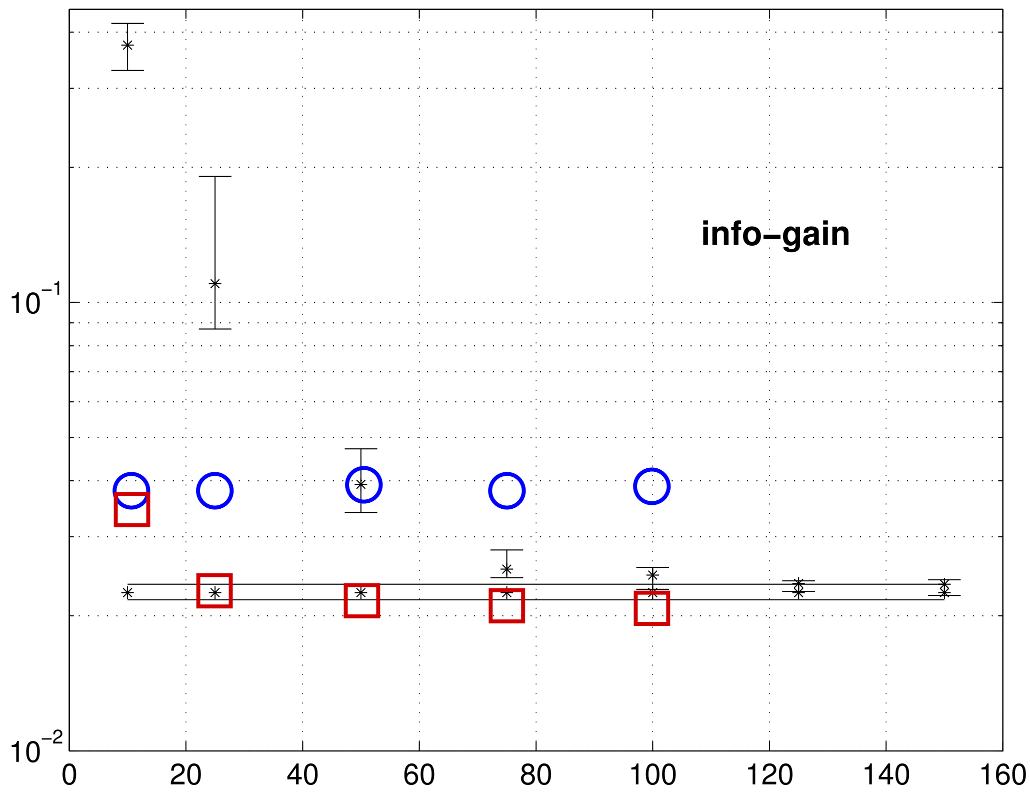
blue circles – SPGP, black – Smo-Bart
horizontal line – full GP on subset

pumadyn-32nm — SPGP and random



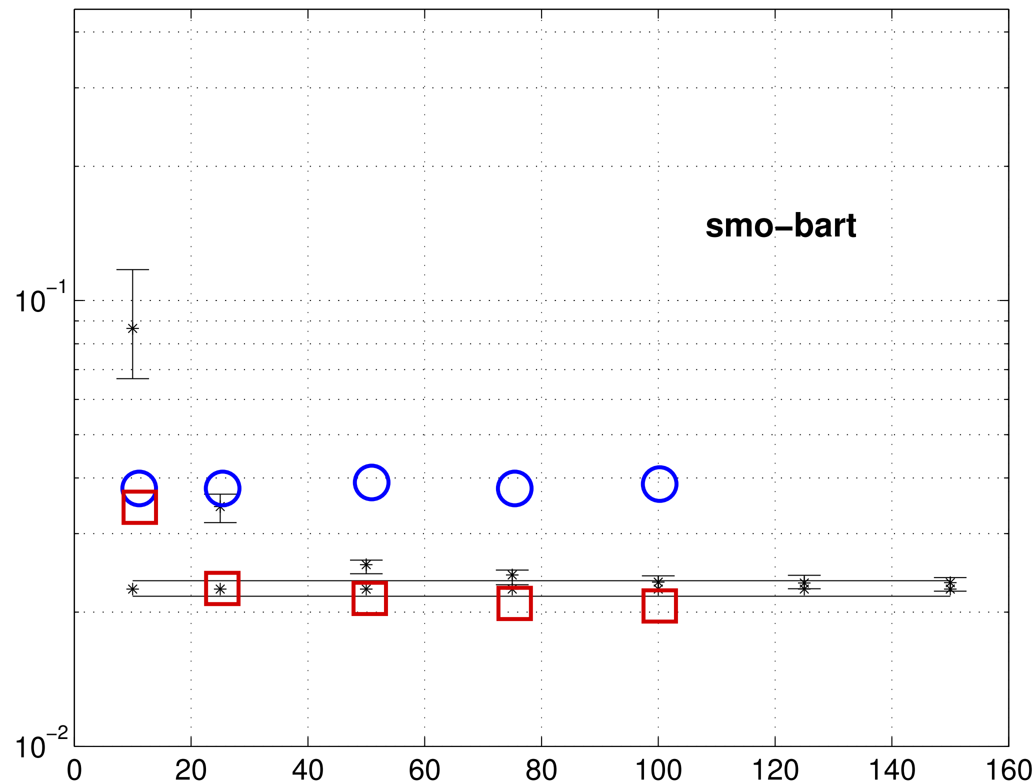
blue circles – SPGP random hyperparameter initialisation
red squares – SPGP hyperparameters initialised from full GP

pumadyn-32nm — SPGP and info-gain



blue circles – SPGP random hyperparameter initialisation
red squares – SPGP hyperparameters initialised from full GP

pumadyn-32nm — SPGP and Smo-Bart



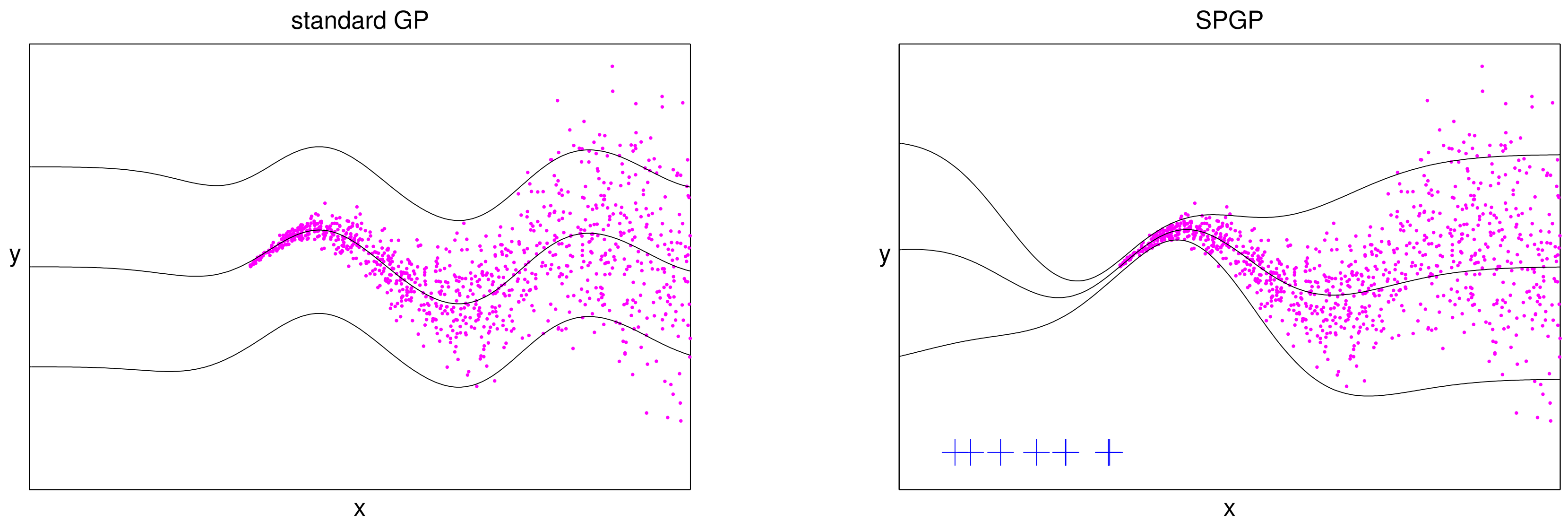
blue circles – SPGP random hyperparameter initialisation

red squares – SPGP hyperparameters initialised from full GP

Problems and possible improvements

- Large pseudo set size and/or high dimensional input space means **optimization becomes impractically big**
- So far we have simply plugged into CG minimizer
- Optimize subsets of variables iteratively (chunking)?
- Stochastic gradient descent?
- **hybrid** — pick some points randomly, optimize others?
- **feature selection** by projecting input space into lower dimensional space?

Non-stationary processes



Although not designed for this purpose, the extra flexibility of the SPGP allows **some non-stationary effects** to be modelled

Conclusions

- New method for sparse GP-like regression
- Significant decrease in test error, especially for **very sparse solutions**
- Added flexibility of moving pseudo input points which are **not constrained to lie on the true data points** leads to better solutions
- Hyperparameters can be jointly learned with pseudo input point locations in a **smooth optimization**
- Much more testing needs to be done to find the best combination of methods!