Sparse Gaussian processes using pseudo-inputs

Ed Snelson (snelson@gatsby.ucl.ac.uk)

Gatsby Computational Neuroscience Unit, UCL

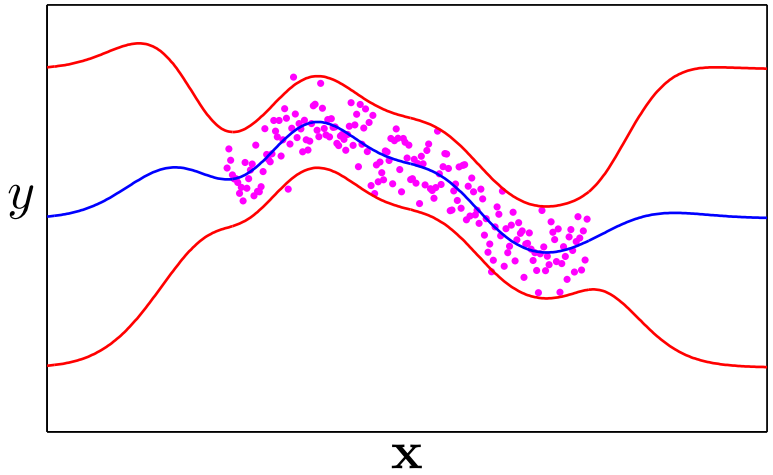
NIPS 2005, 6th December

Work done with Zoubin Ghahramani

Nonlinear regression

Consider the problem of nonlinear regression:

You want to learn a function f with error bars from data $\mathcal{D} = \{\mathbf{X}, \mathbf{y}\}$

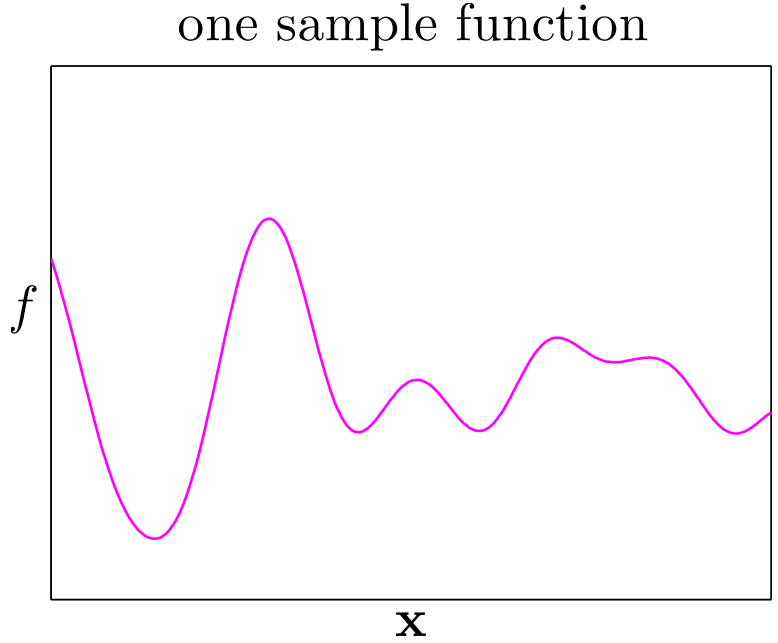


A Gaussian process is a prior over functions $p(f)$ which can be used for Bayesian regression:

$$p(f|\mathcal{D}) = \frac{p(f)p(\mathcal{D}|f)}{p(\mathcal{D})}$$

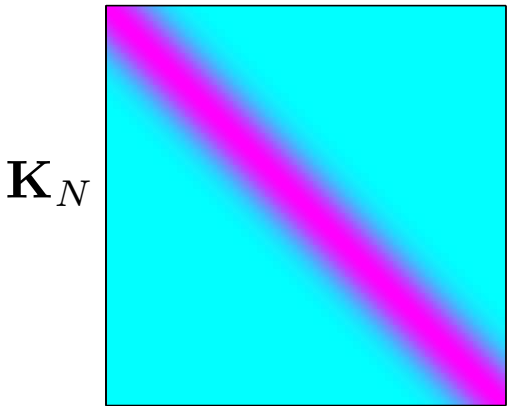
Gaussian process (GP) priors

GP: consistent Gaussian prior on any set of function values $\mathbf{f} = \{f_n\}_{n=1}^N$, given corresponding inputs $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^N$



prior

$$p(\mathbf{f}|\mathbf{X}) = \mathcal{N}(\mathbf{0}, \mathbf{K}_N)$$

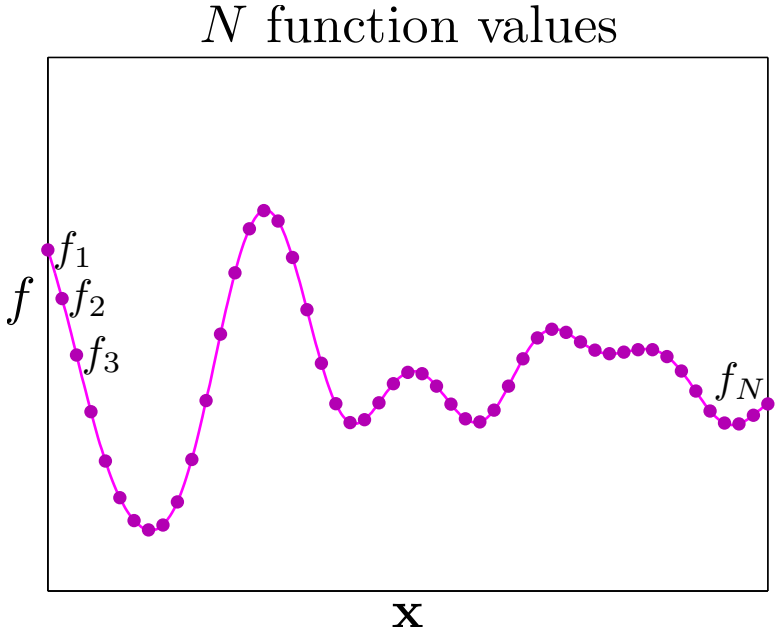


Covariance: $\mathbf{K}_{nn'} = K(\mathbf{x}_n, \mathbf{x}_{n'}; \boldsymbol{\theta})$, hyperparameters $\boldsymbol{\theta}$

$$\mathbf{K}_{nn'} = v \exp\left[-\frac{1}{2}\sum_{d=1}^{D}\left(\frac{x_n^{(d)} - x_{n'}^{(d)}}{r_d}\right)^2\right]$$

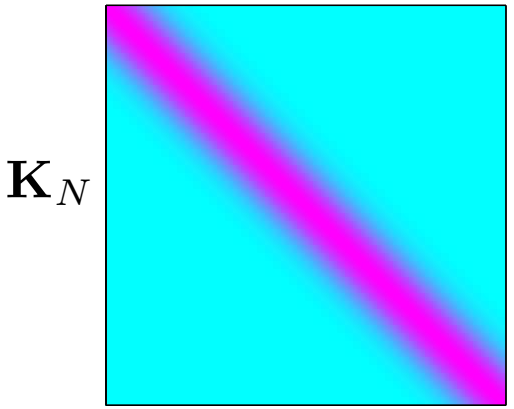
Gaussian process (GP) priors

GP: consistent Gaussian prior on any set of function values $\mathbf{f} = \{f_n\}_{n=1}^N$, given corresponding inputs $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^N$



prior

$$p(\mathbf{f}|\mathbf{X}) = \mathcal{N}(\mathbf{0}, \mathbf{K}_N)$$

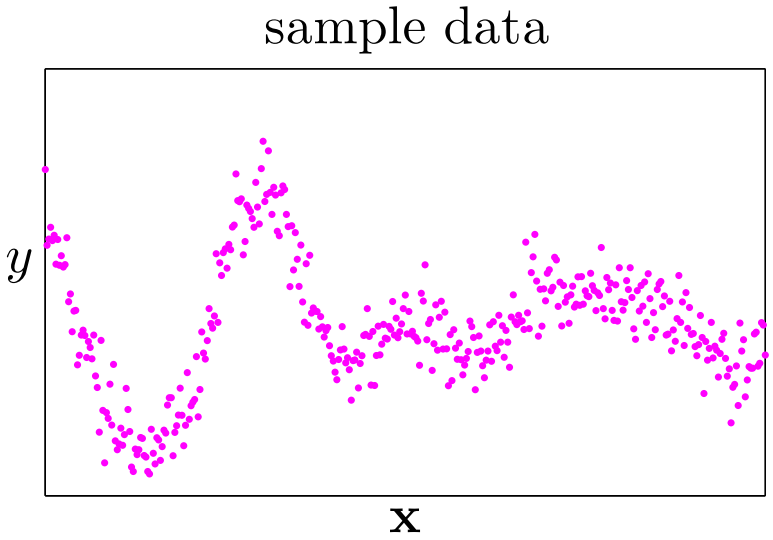


Covariance: $\mathbf{K}_{nn'} = K(\mathbf{x}_n, \mathbf{x}_{n'}; \boldsymbol{\theta})$, hyperparameters $\boldsymbol{\theta}$

$$\mathbf{K}_{nn'} = v \exp\left[-\frac{1}{2}\sum_{d=1}^{D}\left(\frac{x_n^{(d)} - x_{n'}^{(d)}}{r_d}\right)^2\right]$$

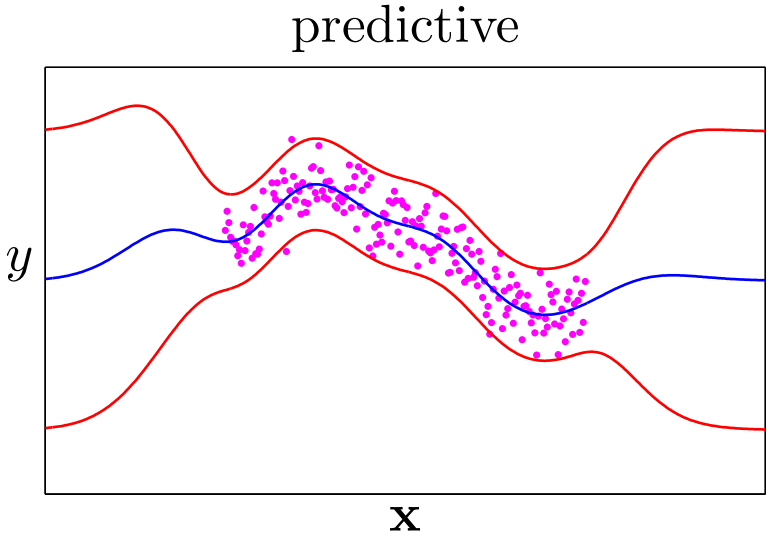
GP regression

Gaussian observation noise: $y_n = f_n + \epsilon_n$, where $\epsilon_n \sim \mathcal{N}(0, \sigma^2)$



marginal likelihood

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



predictive distribution

$$p(y_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \mathcal{N}(\mu_*, \sigma_*^2)$$

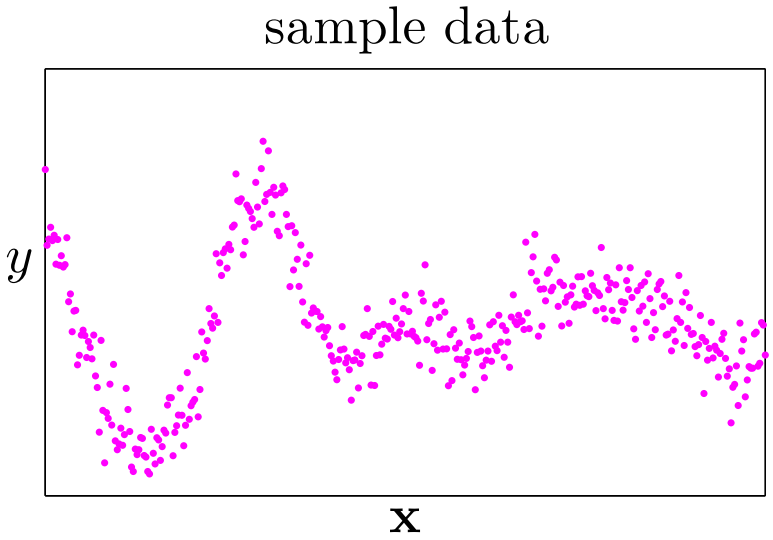
$$\mu_* = \mathbf{K}_{*N}(\mathbf{K}_N + \sigma^2\mathbf{I})^{-1}\mathbf{y}$$

$$\sigma_*^2 = K_{**} - \mathbf{K}_{*N}(\mathbf{K}_N + \sigma^2\mathbf{I})^{-1}\mathbf{K}_{N*} + \sigma^2$$

Problem: N^3 computation

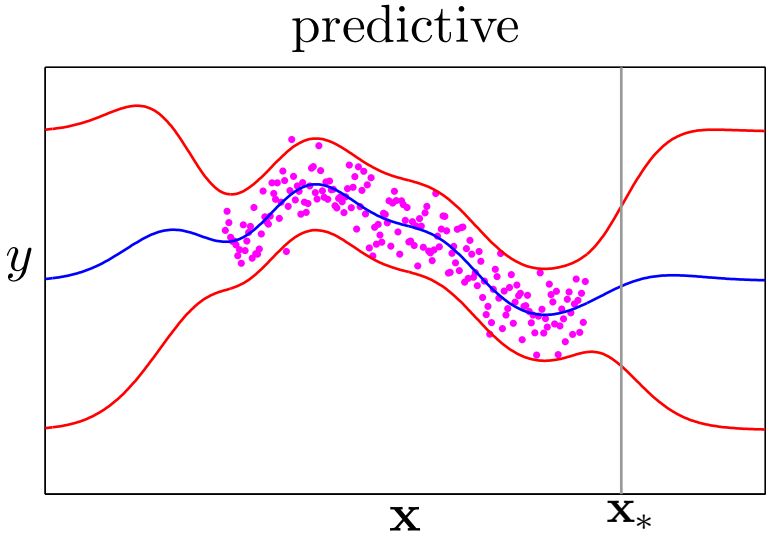
GP regression

Gaussian observation noise: $y_n = f_n + \epsilon_n$, where $\epsilon_n \sim \mathcal{N}(0, \sigma^2)$



marginal likelihood

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



predictive distribution

$$p(y_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \mathcal{N}(\mu_*, \sigma_*^2)$$

$$\mu_* = \mathbf{K}_{*N}(\mathbf{K}_N + \sigma^2\mathbf{I})^{-1}\mathbf{y}$$

$$\sigma_*^2 = K_{**} - \mathbf{K}_{*N}(\mathbf{K}_N + \sigma^2\mathbf{I})^{-1}\mathbf{K}_{N*} + \sigma^2$$

Problem: N^3 computation

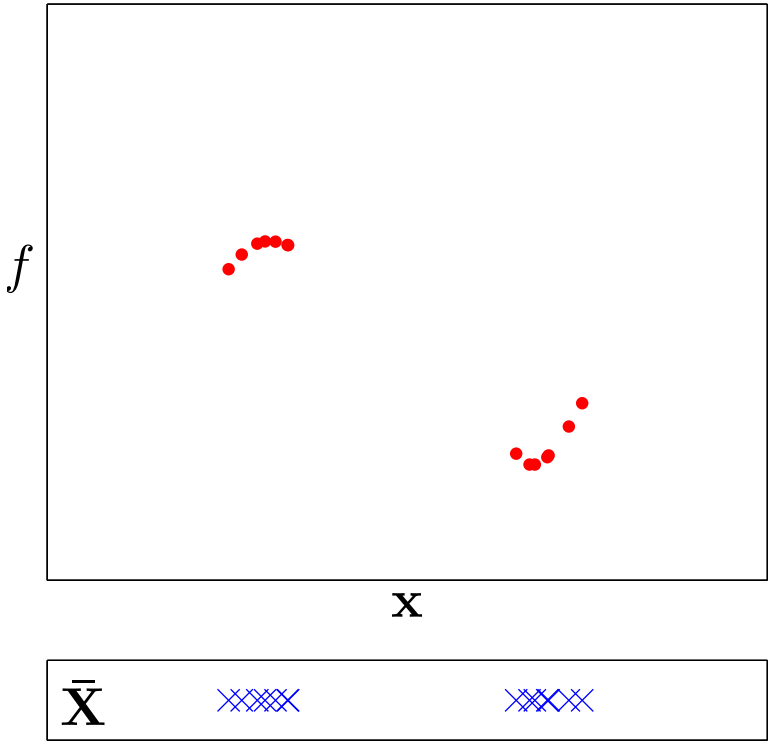
Overview

This talk contains 2 key ideas:

1. A new sparse Gaussian process approximation based on a small set of M 'pseudo-inputs' ($M \ll N$). This reduces computational complexity to $\mathcal{O}(M^2 N)$

2. A gradient based learning procedure for finding the pseudo-inputs and hyperparameters of the Gaussian process, in one joint optimization

Two stage generative model



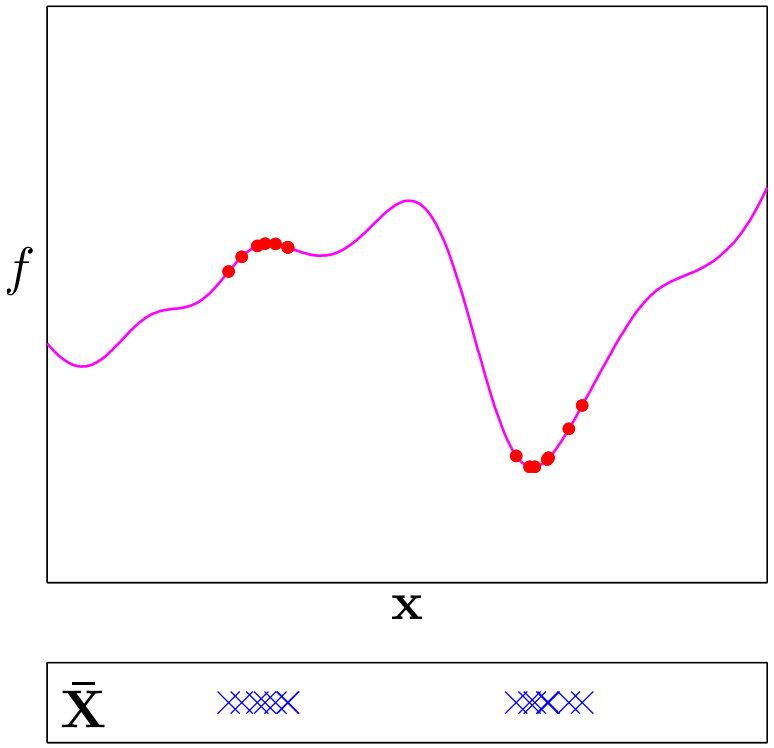
pseudo-input prior

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

1. Choose any set of M (pseudo-) inputs $\bar{\mathbf{X}}$

2. Draw corresponding function values $\bar{\mathbf{f}}$ from prior

Two stage generative model

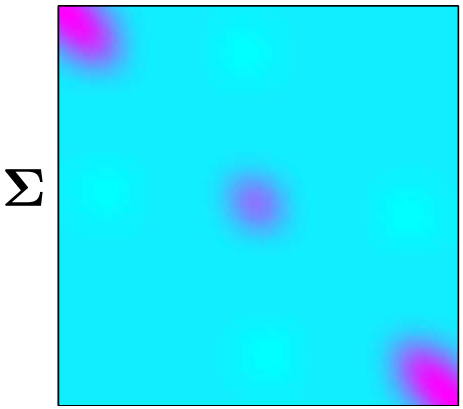


conditional

$$p(\mathbf{f}|\bar{\mathbf{f}}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

$$\boldsymbol{\mu} = \mathbf{K}_{NM}\mathbf{K}_M^{-1}\bar{\mathbf{f}}$$

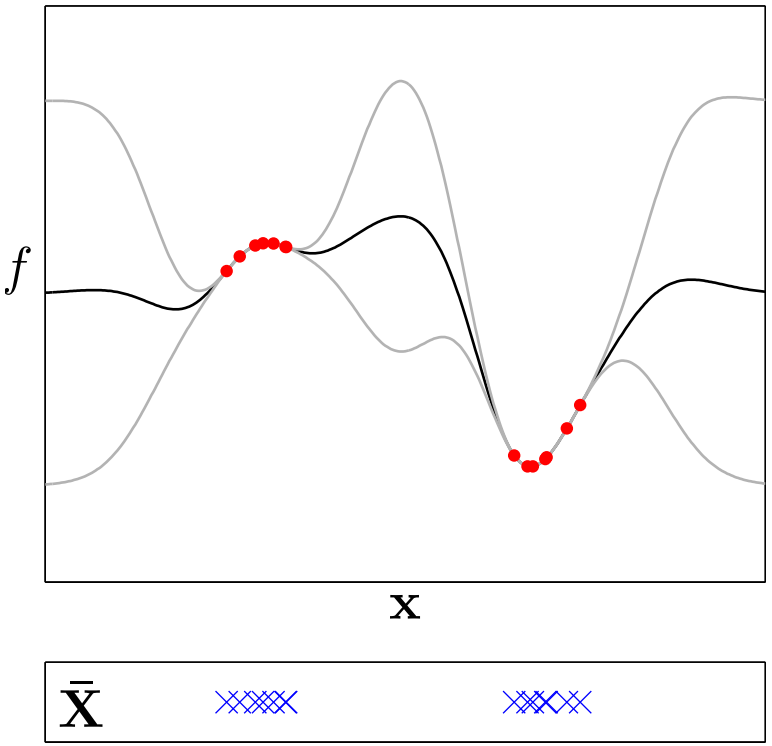
$$\boldsymbol{\Sigma} = \mathbf{K}_N - \mathbf{K}_{NM}\mathbf{K}_M^{-1}\mathbf{K}_{MN}$$



3. Draw $\mathbf{f}$ conditioned on $\bar{\mathbf{f}}$

- This two stage procedure defines exactly the same GP prior
- We have not gained anything yet, but it inspires a sparse approximation ...

Factorized approximation

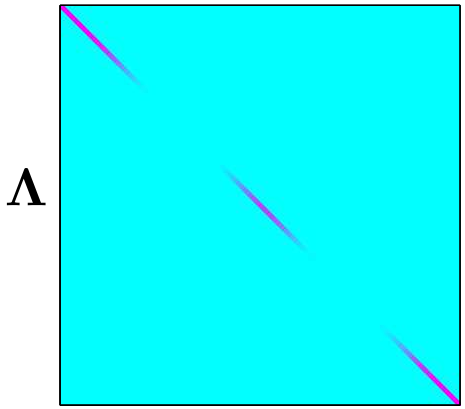


single point conditional

$$p(f_n|\bar{\mathbf{f}}) = \mathcal{N}(\mu_n, \lambda_n)$$

$$\mu_n = \mathbf{K}_{nM}\mathbf{K}_M^{-1}\bar{\mathbf{f}}$$

$$\lambda_n = K_{nn} - \mathbf{K}_{nM}\mathbf{K}_M^{-1}\mathbf{K}_{Mn}$$



Approximate: $p(\mathbf{f}|\bar{\mathbf{f}}) \approx \prod_n p(f_n|\bar{\mathbf{f}}) = \mathcal{N}(\boldsymbol{\mu}, \mathbf{\Lambda})\ , \qquad \mathbf{\Lambda} = \mathrm{diag}(\boldsymbol{\lambda})$

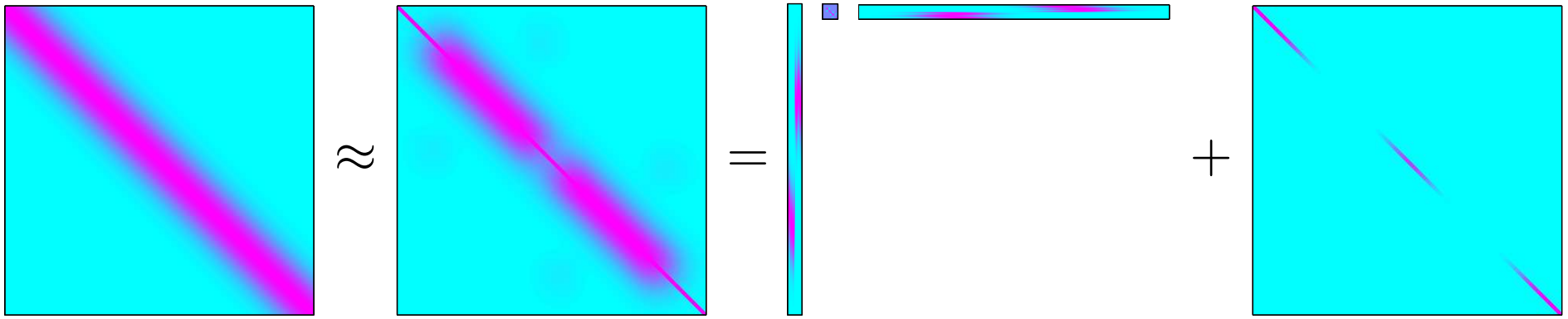
Minimum KL: $\min_{q_n} \mathrm{KL}\left[p(\mathbf{f}|\bar{\mathbf{f}}) \parallel \prod_n q_n(f_n)\right]$

Sparse pseudo-input Gaussian processes (SPGP)

Integrate out $\bar{\mathbf{f}}$ to obtain SPGP prior: $p(\mathbf{f}) = \int \mathrm{d}\bar{\mathbf{f}} \prod_n p(f_n|\bar{\mathbf{f}})\, p(\bar{\mathbf{f}})$

GP prior $\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad$ SPGP prior

$$\mathcal{N}(\mathbf{0}, \mathbf{K}_N) \approx p(\mathbf{f}) = \mathcal{N}(\mathbf{0}, \mathbf{K}_{NM}\mathbf{K}_M^{-1}\mathbf{K}_{MN} + \mathbf{\Lambda})$$



- SPGP covariance inverted in $\mathcal{O}(M^2N) \Rightarrow$ sparse
- SPGP $=$ GP with non-stationary covariance parameterized by $\bar{\mathbf{X}}$
- Given data $\{\mathbf{X}, \mathbf{y}\}$ with noise σ^2, predictive mean and variance can be computed in $\mathcal{O}(M)$ and $\mathcal{O}(M^2)$ per test case respectively

How to find pseudo-inputs?

Pseudo-inputs are like extra hyperparameters: we jointly maximize marginal likelihood w.r.t. $(\bar{\mathbf{X}}, \boldsymbol{\theta}, \sigma^2)$

$$p(\mathbf{y}|\mathbf{X}, \bar{\mathbf{X}}, \boldsymbol{\theta}, \sigma^2) = \mathcal{N}\left(\mathbf{0},\ \mathbf{K}_{NM}\mathbf{K}_M^{-1}\mathbf{K}_{MN} + \boldsymbol{\Lambda} + \sigma^2\mathbf{I}\right)$$

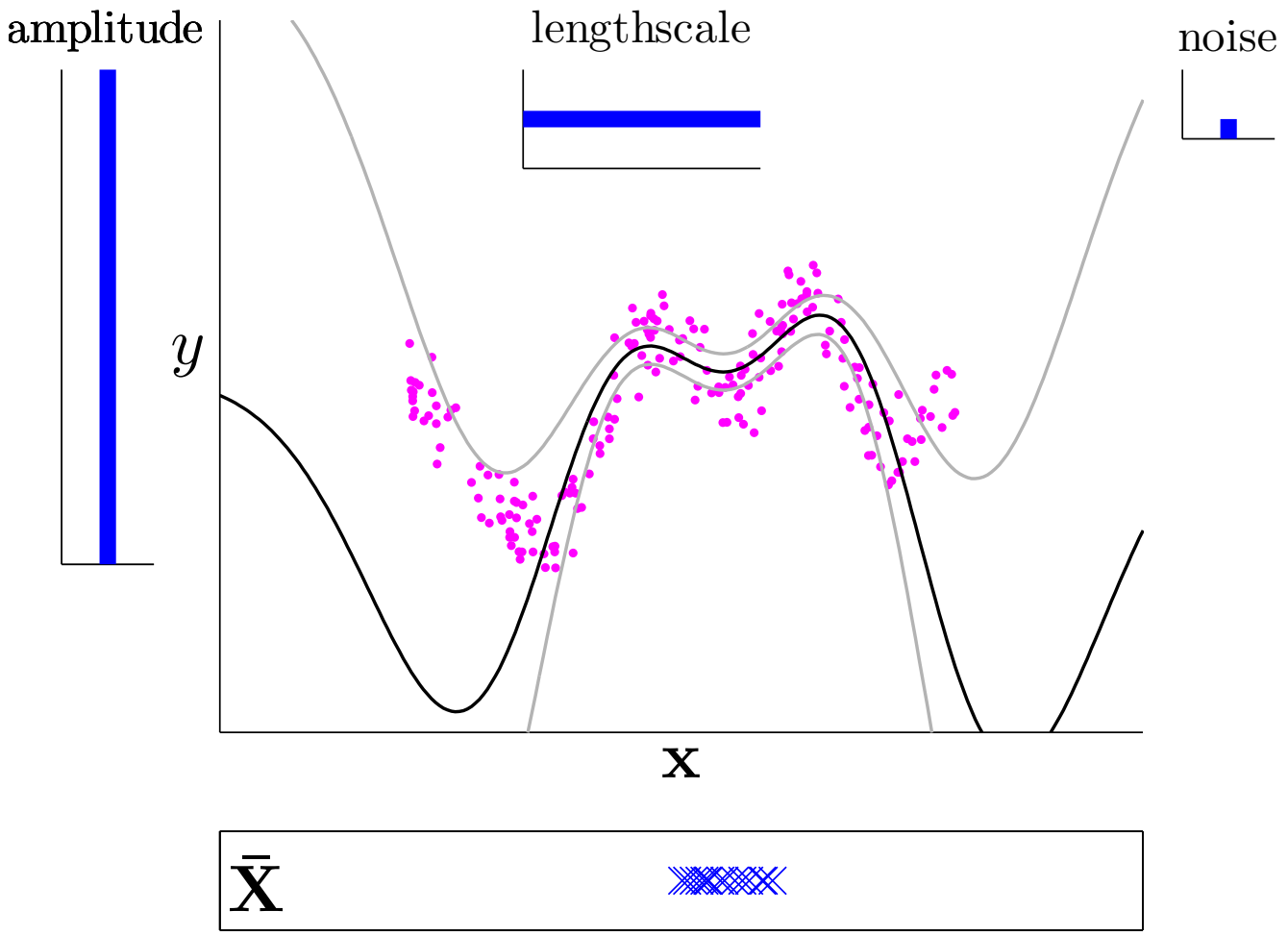
Key advantages over many related sparse methods [1]:

1. Pseudo-inputs not constrained to subset of data ('active set') = **improved accuracy and flexibility**

2. Joint optimization **avoids discontinuities** that arise when active set selection is interleaved with hyperparameter learning

[1] Tresp (2000), Smola & Bartlett (2001), Csató & Opper (2002), Seeger et al. (2003)

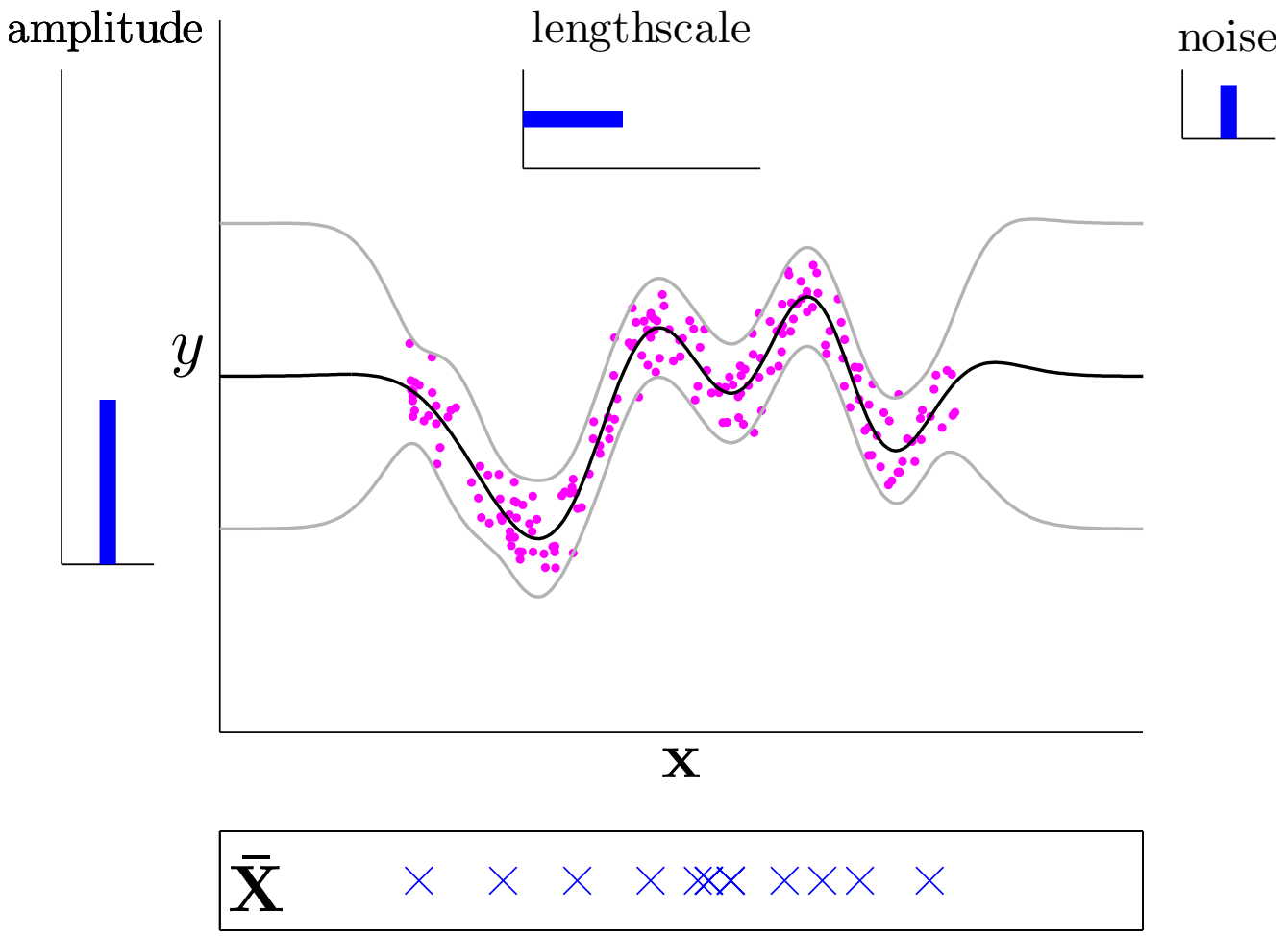
1D demo



Initialize adversarially:

amplitude and lengthscale too big
noise too small
pseudo-inputs bunched up

1D demo

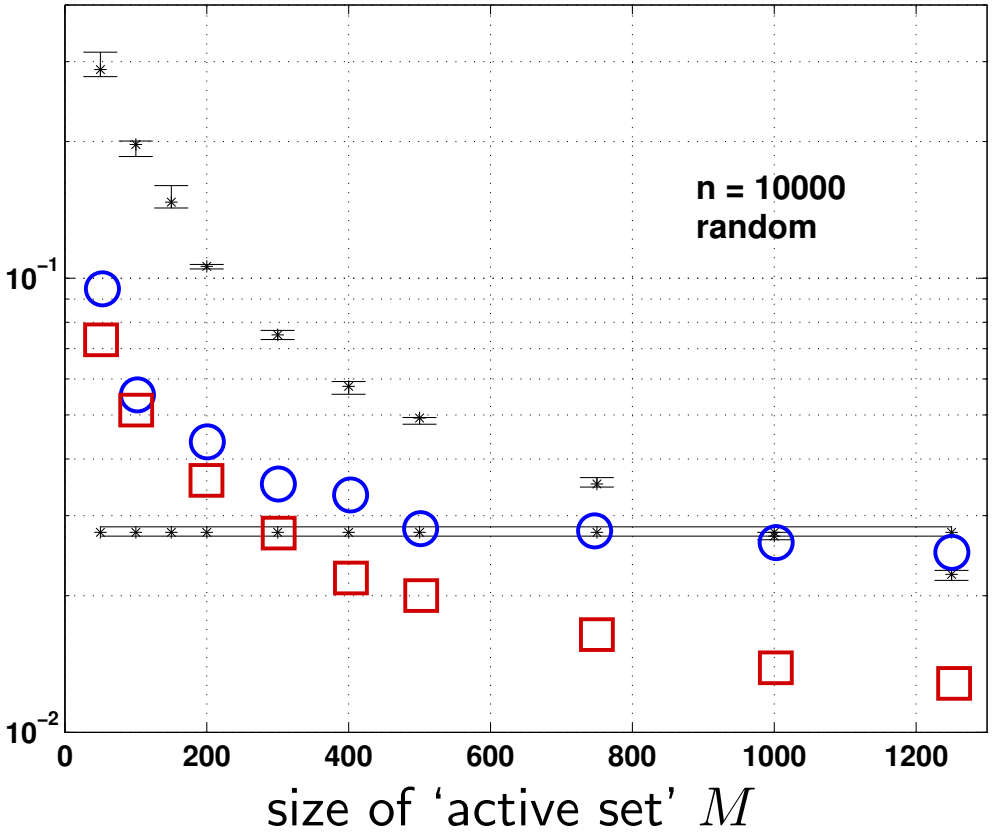


Pseudo-inputs and hyperparameters optimized

Selected Results:

(more at poster)

kin40k[1] — SPGP vs random



10000 train
30000 test
9 attributes

horizontal line – full GP on subset size 2000 ($*$)

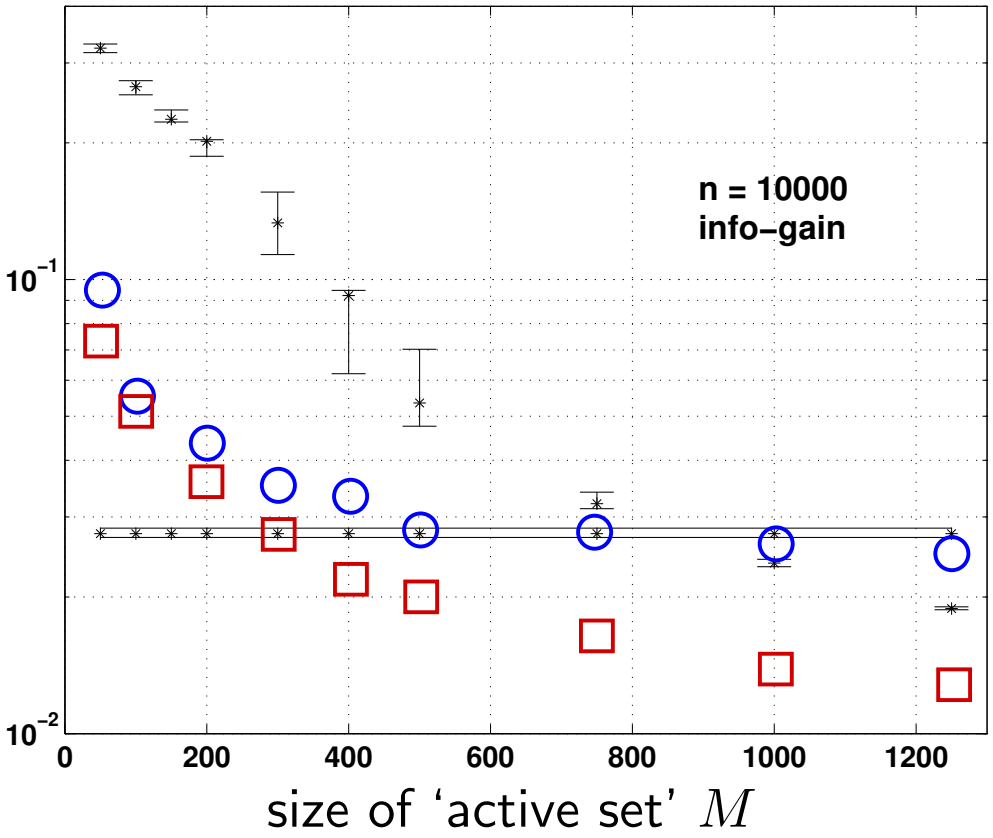
black – random subset – hyperparameters obtained from $*$

red squares – SPGP – pseudo-inputs optimized, hyperparameters obtained from $*$

blue circles – SPGP – pseudo-inputs and hyperparameters optimized

[1] as tested in Seeger et al. (2003)

kin40k — SPGP vs info-gain



test m.sq. error

10000 train
30000 test
9 attributes

horizontal line – full GP on subset size 2000 ($*$)

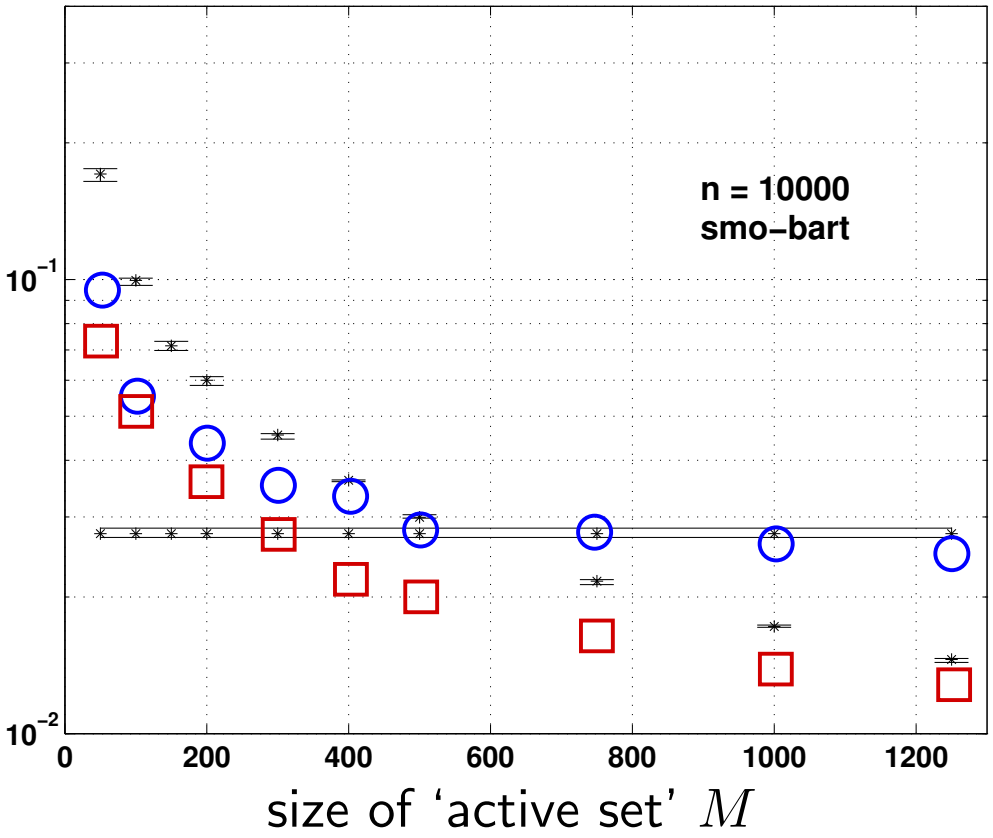
black – info-gain[1] – hyperparameters obtained from $*$

red squares – SPGP – pseudo-inputs optimized, hyperparameters obtained from $*$

blue circles – SPGP – pseudo-inputs and hyperparameters optimized

[1] Seeger et al. (2003)

kin40k — SPGP vs Smo-Bart



test m.sq. error

10000 train
30000 test
9 attributes

horizontal line – full GP on subset size 2000 ($*$)

black – Smo-Bart[1] – hyperparameters obtained from $*$

red squares – SPGP – pseudo-inputs optimized, hyperparameters obtained from $*$

blue circles – SPGP – pseudo-inputs and hyperparameters optimized

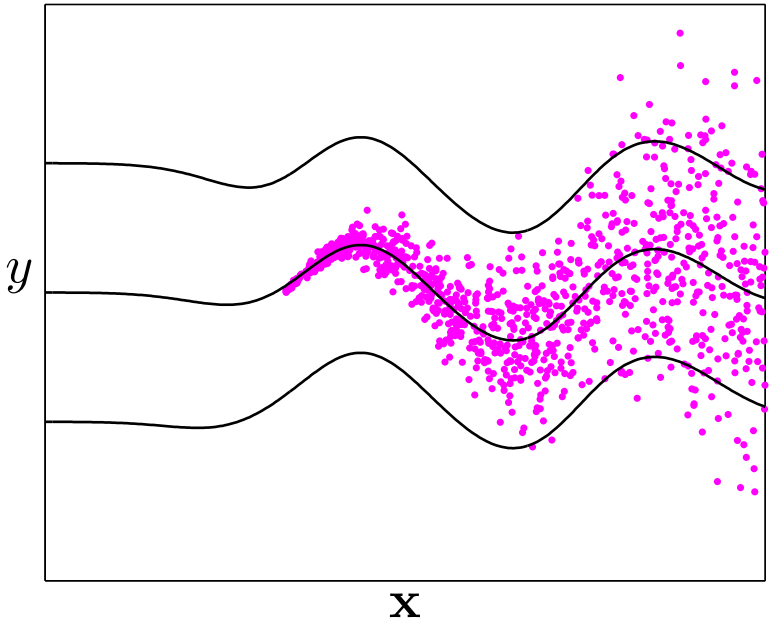
[1] Smola & Bartlett (2001)

Local maxima and overfitting?

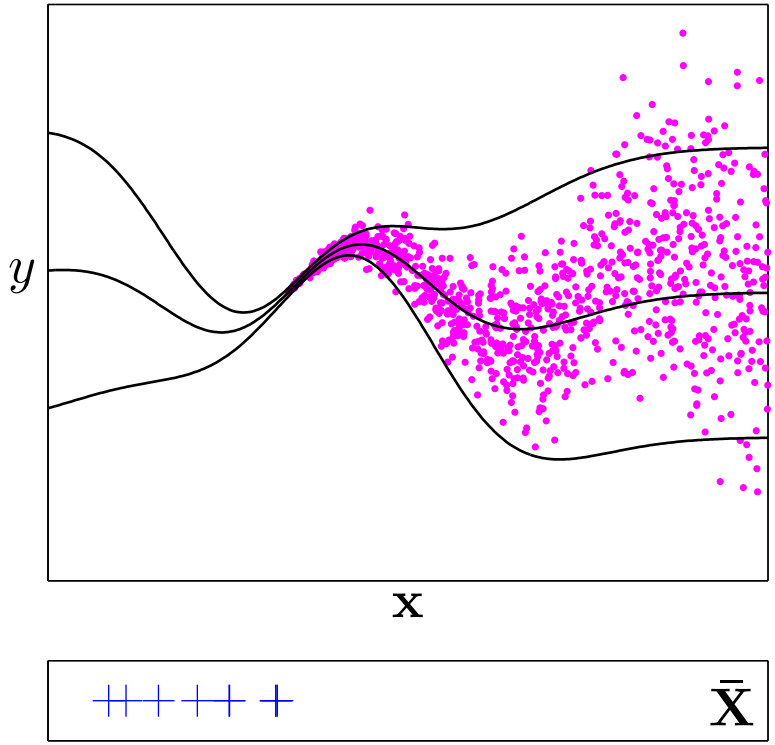
- Many local maxima, but can initialize pseudo-inputs on random subset of data. Hyperparameter initialization more tricky
- Many parameters: $MD + |\boldsymbol{\theta}| + 1$ instead of $|\boldsymbol{\theta}| + 1$. Overfitting?
  ($D =$ input space dimension, $M =$ no. of pseudo-inputs)
- Consider $M = N$ and $\bar{\mathbf{X}} = \mathbf{X}$
  - Here $\mathbf{K}_{MN} = \mathbf{K}_M = \mathbf{K}_N$, $\boldsymbol{\Lambda} = \sigma^2\mathbf{I}$
    $\Rightarrow$ SPGP collapses to full GP
- However interaction with hyperparameter learning can lead to overfitting behaviour
- For full Bayesian treatment: sample pseudo-inputs and hyperparameters from $p(\bar{\mathbf{X}}, \boldsymbol{\theta}, \sigma^2|\mathbf{X}, \mathbf{y})$ instead of optimizing

Modeling non-stationarity

standard GP



SPGP



Extra flexibility of SPGP allows some non-stationary effects to be modeled

Limitations and possible extensions

- Large pseudo set size M and/or high dimensional input space D means optimization can become impractically big
  - possible solution: learning a projection of input space into lower dimensional space [work in progress]
  - may also prevent overfitting
- We used CG or L-BFGS but many optimization schemes available:
  - Optimize subsets of variables iteratively (chunking)
  - Stochastic gradient descent
  - hybrid — pick some points randomly, optimize others
  - EM algorithm
- Extension to classification and other likelihood functions

Conclusions

- New method for sparse GP regression
- Significant decrease in test error, especially for very sparse solutions
- Added flexibility of moving pseudo-inputs which are not constrained to lie on the data leads to better solutions
- Hyperparameters jointly learned with pseudo-inputs in a single smooth optimization
- Matlab code, draft paper, and this talk available (`www.gatsby.ucl.ac.uk/~snelson`)

Acknowledgements

Sheffield GP Round-table

Carl Rasmussen
Joaquin Quiñonero-Candela
Peter Sollich
Matthias Seeger
Neil Lawrence
Chris Williams
Tom Minka
Sam Roweis

PASCAL European Network of Excellence

Relation of SPGP to PLV[1]

SPGP

Approximate conditional:

$$p(\mathbf{f}|\bar{\mathbf{f}}) \approx \prod_n p(f_n|\bar{\mathbf{f}}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Lambda})$$

minimum KL fully factorized approximation

Marginal likelihood:

$$\mathcal{N}(\mathbf{0}, \mathbf{K}_{NM}\mathbf{K}_M^{-1}\mathbf{K}_{MN} + \boldsymbol{\Lambda} + \sigma^2\mathbf{I})$$

marginal variances match full GP everywhere

Pseudo-inputs:

not constrained to data – optimized by gradient ascent on marginal likelihood, together with hyperparameters

PLV

Approximate conditional:

$$p(\mathbf{f}|\bar{\mathbf{f}}) \approx \mathcal{N}(\boldsymbol{\mu}, \mathbf{0})$$

uncertainty not taken into account – deterministic approximation

Marginal likelihood:

$$\mathcal{N}(\mathbf{0}, \mathbf{K}_{NM}\mathbf{K}_M^{-1}\mathbf{K}_{MN} + \sigma^2\mathbf{I})$$

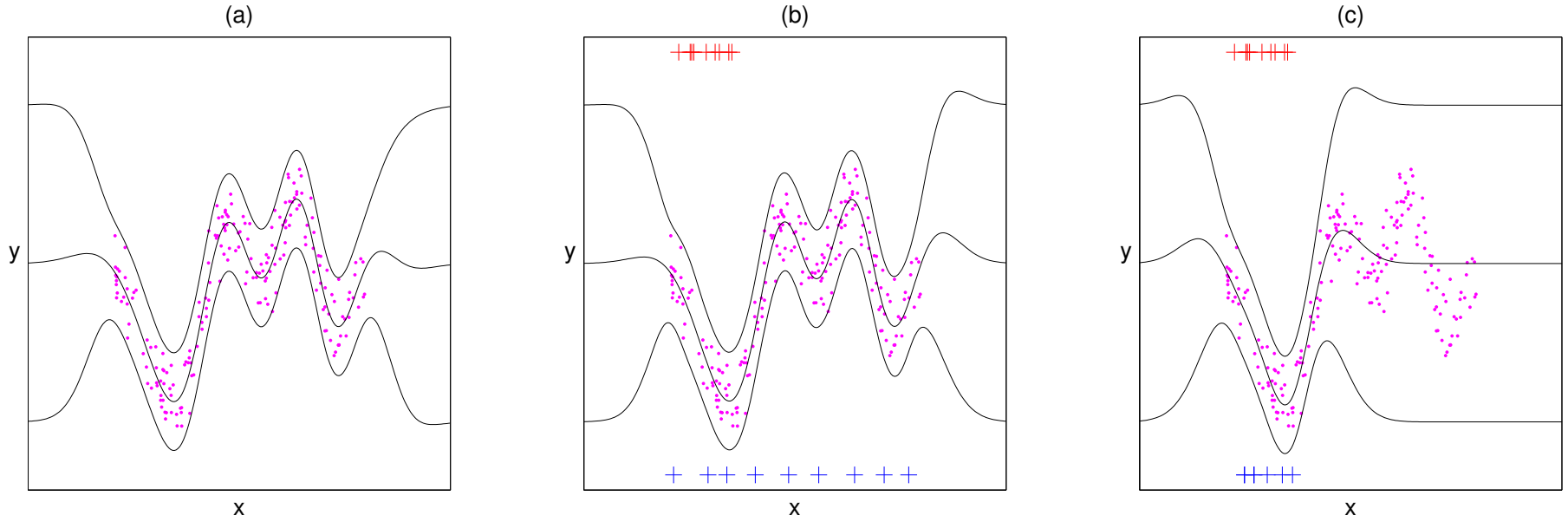
marginal variances decay to σ^2 away from 'active set' points

Active set:

chosen as subset of data using greedy info-gain criteria; active set selection and hyperparameter learning interleaved

[1] Seeger et al. (2003)

PLV with pseudo-inputs



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