Flexible and efficient Gaussian process models

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

Gatsby Computational Neuroscience Unit, UCL

GPiP workshop, 12th June 2006

Work done with Zoubin Ghahramani

Overview

Several techniques to improve efficiency and/or flexibility of GPs:

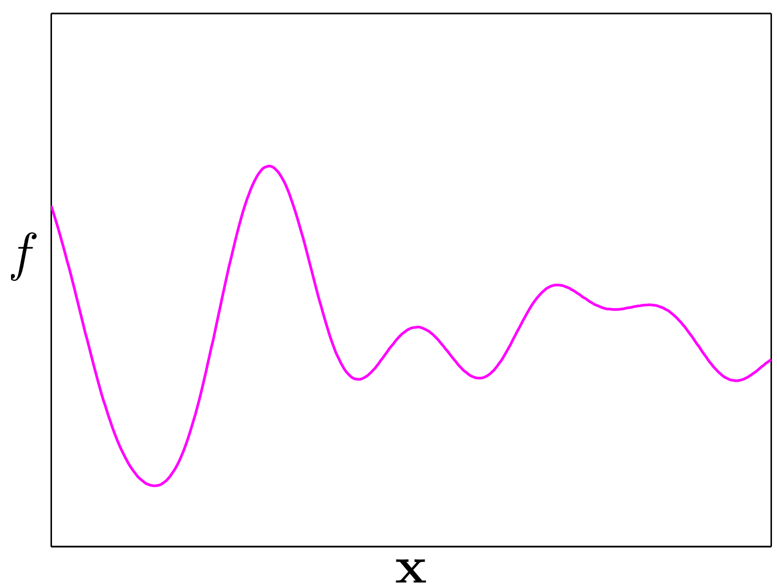
1. A sparse Gaussian process approximation (SPGP/FITC) based on a small set of M 'pseudo-inputs' ($M \ll N$). This reduces computational complexity from $\mathcal{O}(N^3)$ to $\mathcal{O}(M^2 N)$
2. A gradient based learning procedure for finding the pseudo-inputs and hyperparameters of the GP, in one joint optimization
3. Supervised dimensionality reduction for problems with large numbers of input features[1]
4. Modeling input dependent noise[1]

---

[1] to appear, UAI 2006

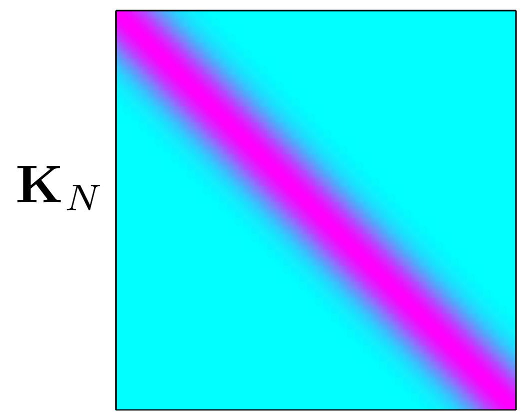
Gaussian process (GP) priors

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

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

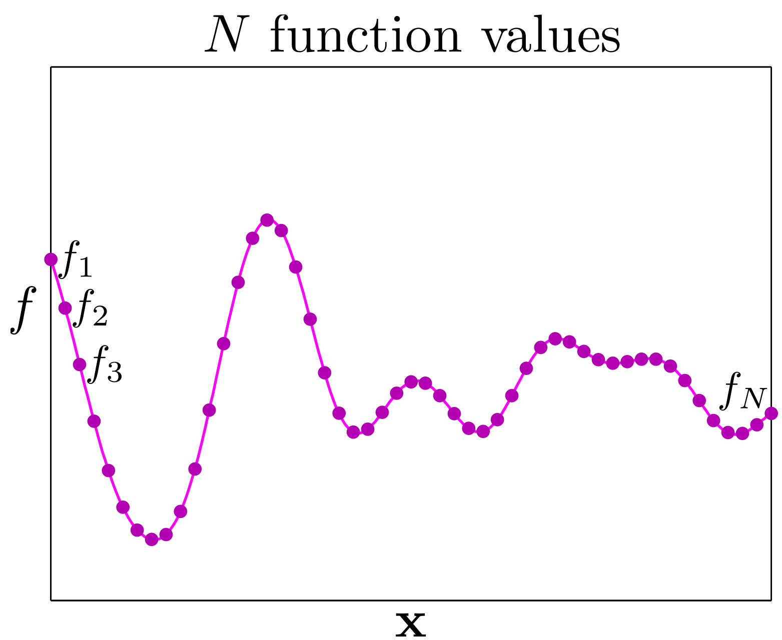


Covariance: $\mathbf{K}_{nn'} = K(\mathbf{x}_n, \mathbf{x}_{n'}; \boldsymbol{\theta})$, hyperparameters $\boldsymbol{\theta}$

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

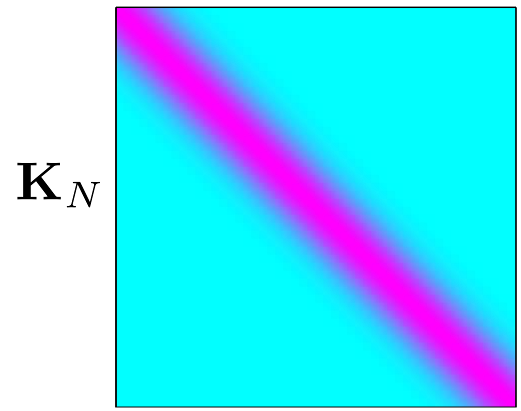
Gaussian process (GP) priors

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

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

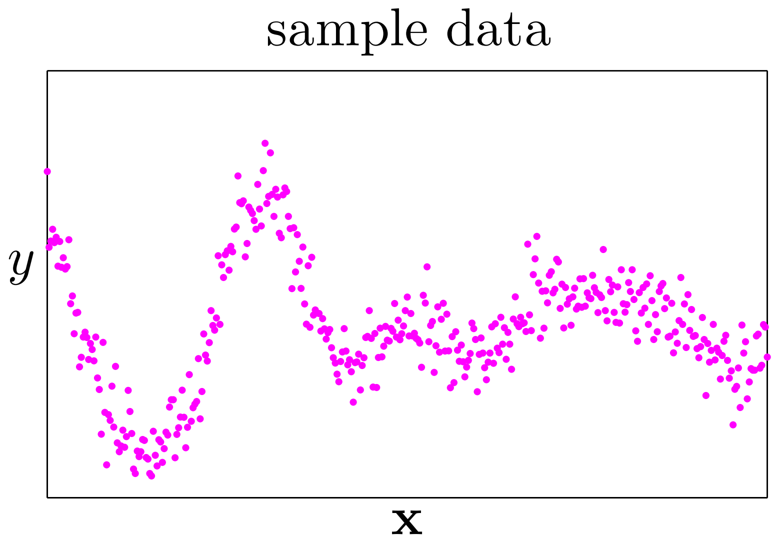


Covariance: $\mathbf{K}_{nn'} = K(\mathbf{x}_n, \mathbf{x}_{n'}; \boldsymbol{\theta})$, hyperparameters $\boldsymbol{\theta}$

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

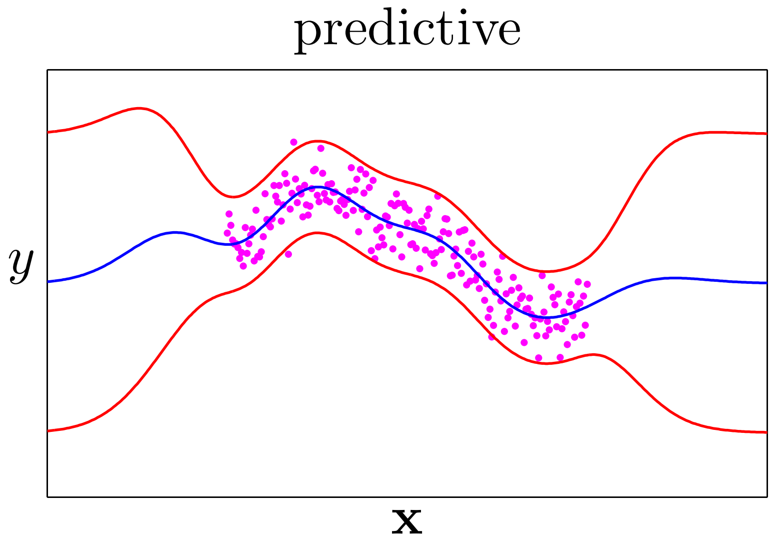
GP regression

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

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

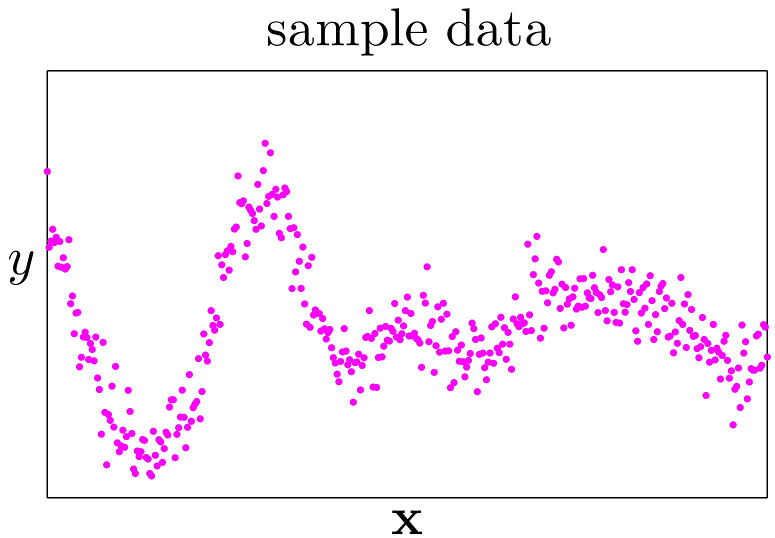
$$\mu_* = \mathbf{K}_{*N}(\mathbf{K}_N + \sigma^2\mathbf{I})^{-1}\mathbf{y}$$

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

Problem: N^3 computation

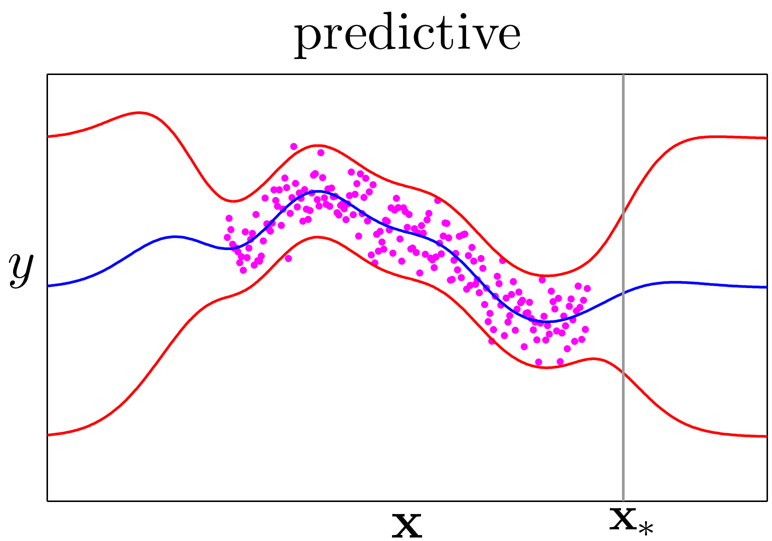
GP regression

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

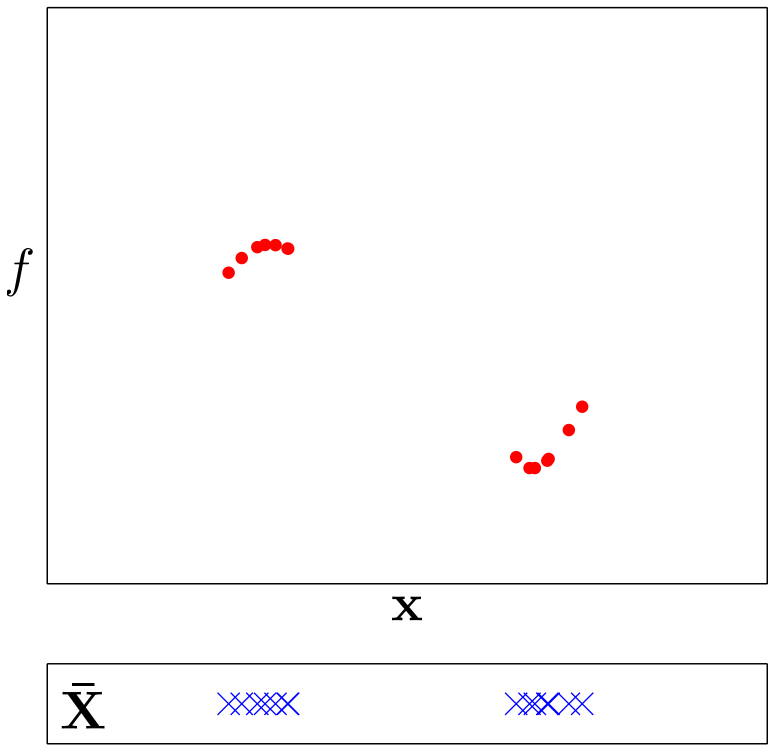
$$p(y_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \mathcal{N}(\mu_*, \sigma_*^2)$$

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

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

Problem: N^3 computation

Two stage generative model

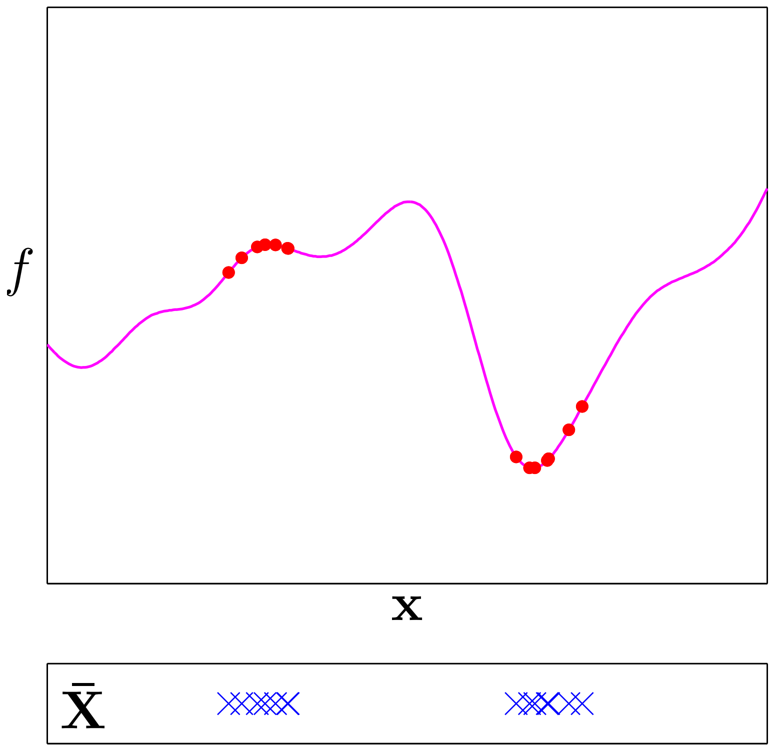


pseudo-input prior

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

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

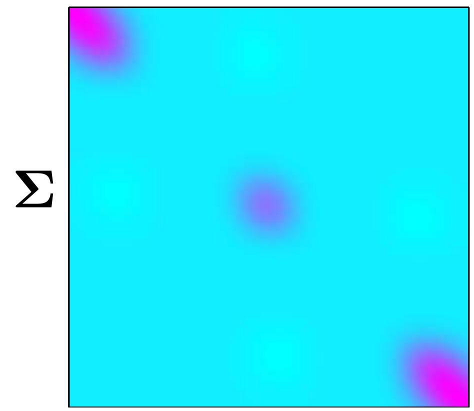


conditional

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

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

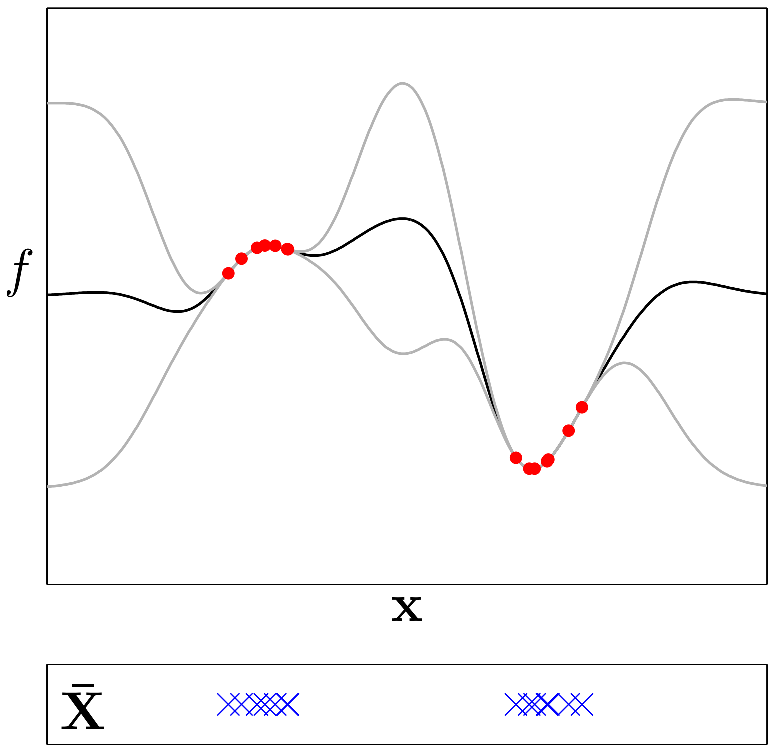
$$\boldsymbol{\Sigma} = \mathbf{K}_N - \mathbf{K}_{NM}\mathbf{K}_M^{-1}\mathbf{K}_{MN}$$



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

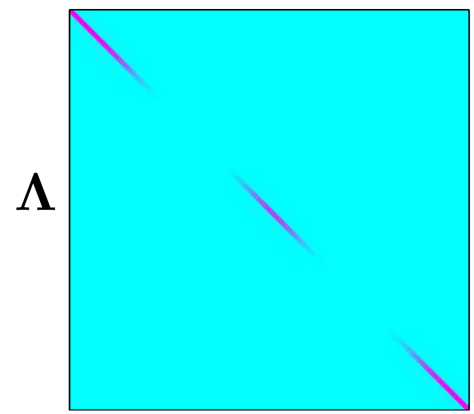


single point conditional

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

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

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



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

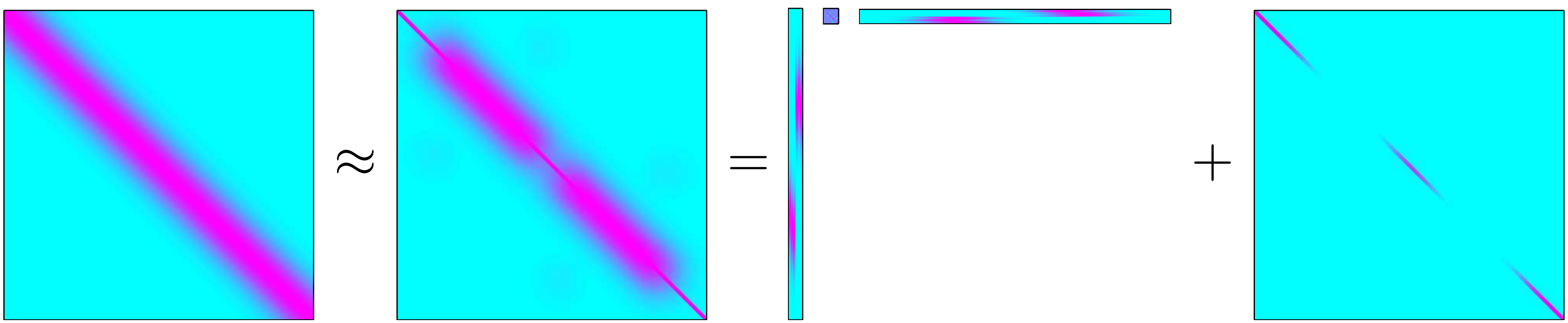
Minimum KL: $\min_{q_n} \mathrm{KL}\left[p(\mathbf{f}|\bar{\mathbf{f}}) \parallel \prod_n q_n(f_n)\right]$

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 SPGP/FITC prior

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



- SPGP/FITC 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. $(\bar{\mathbf{X}}, \boldsymbol{\theta}, \sigma^2)$

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

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)

Local maxima and overfitting?

- Many local maxima, but can initialize pseudo-inputs on random subset of data. Hyperparameter initialization more tricky

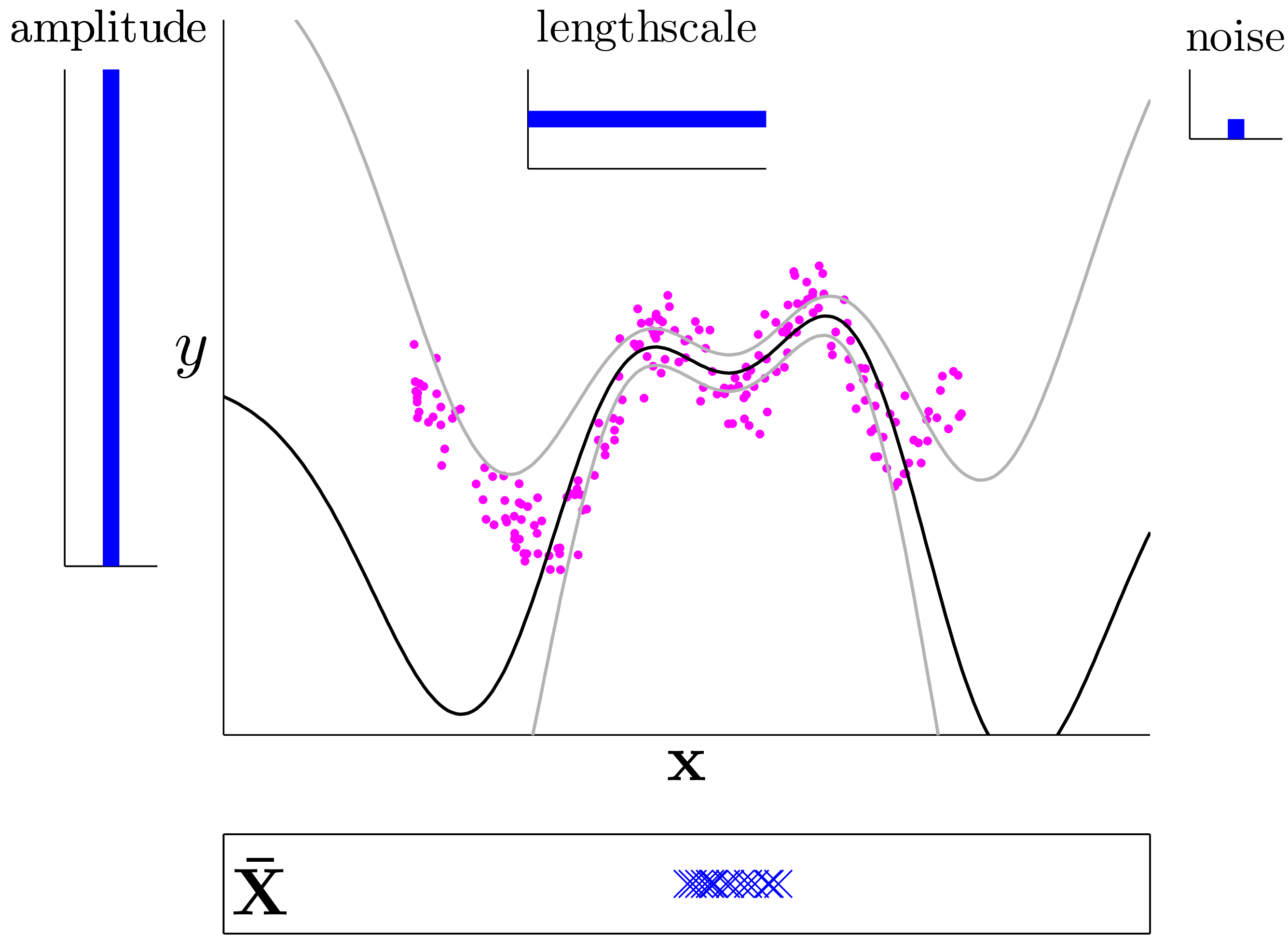
- Many parameters: $MD + |\boldsymbol{\theta}| + 1$ instead of $|\boldsymbol{\theta}| + 1$. Overfitting? ($D =$ input space dimension, $M =$ no. of pseudo-inputs)

- 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

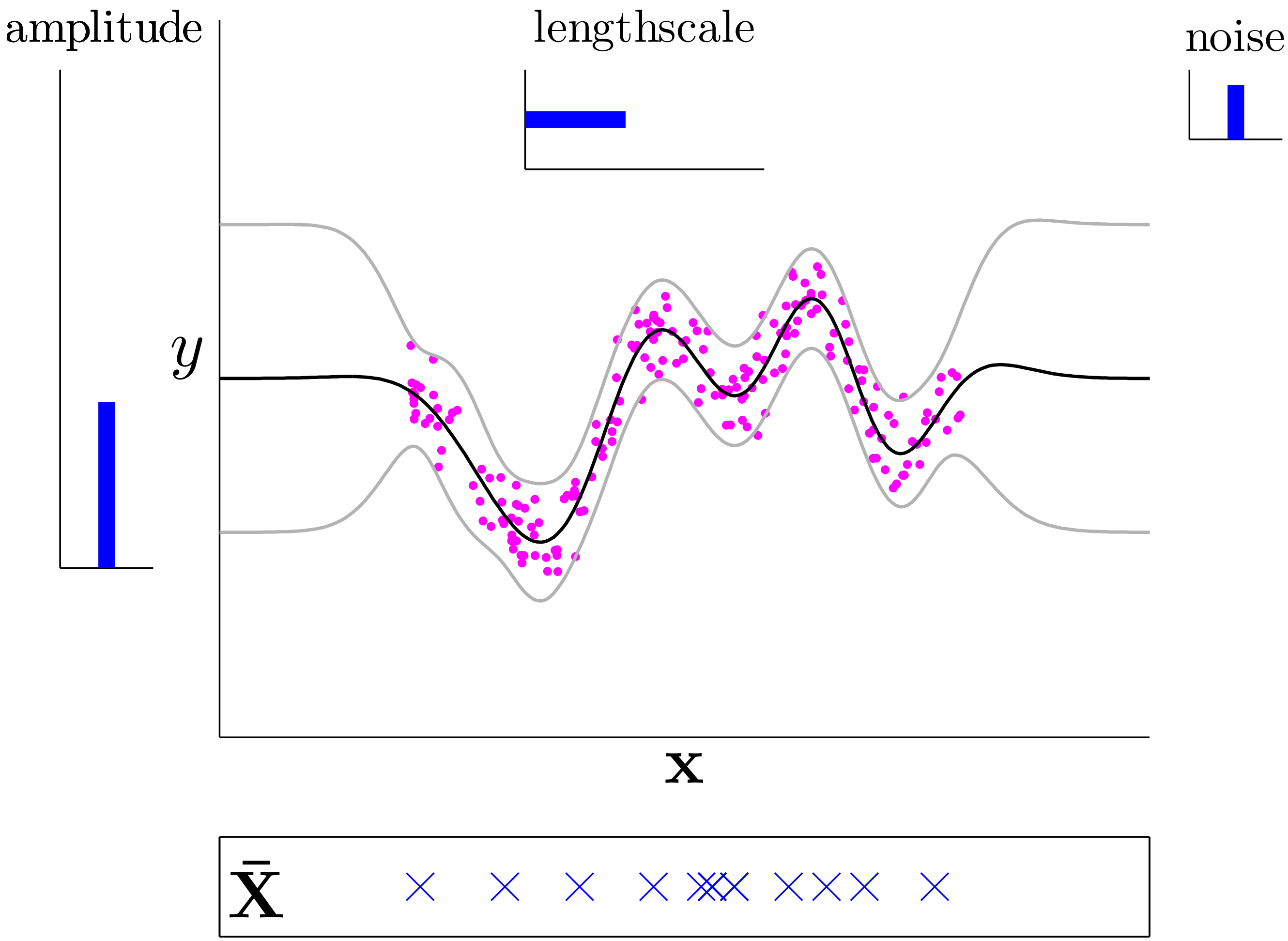
1D demo



Initialize adversarially:

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

1D demo



Pseudo-inputs and hyperparameters optimized

Dimensionality reduction

- Optimizing pseudo-inputs becomes unfeasible for high dimensional input spaces – $MD + |\boldsymbol{\theta}| + 1$ sized optimization space
  ($D =$ input space dimension, $M =$ no. of pseudo-inputs)

- M is a user contolled parameter that can be used to trade off accuracy and computation – D is not

- We can extend the SPGP by learning a low dimensional projection of the input space

- We learn a linear projection of the data points $\mathbf{x}_n^{\text{new}} = P\mathbf{x}_n$ in a supervised manner – contrast: PCA

Dimensionality reduction

Again this involves a modification to the covariance function[1]:

$$K(\mathbf{x}_n, \mathbf{x}_{n'}) = c \exp\left[-\tfrac{1}{2}\big(P(\mathbf{x}_n - \mathbf{x}_{n'})\big)^\top P(\mathbf{x}_n - \mathbf{x}_{n'})\right]$$

When combined with the SPGP, the pseudo-inputs now live in the reduced dimensional (G) space:

$$K(\mathbf{x}_n, \bar{\mathbf{x}}_m) = c \exp\left[-\tfrac{1}{2}(P\mathbf{x}_n - \bar{\mathbf{x}}_m)^\top (P\mathbf{x}_n - \bar{\mathbf{x}}_m)\right]$$
$$K(\bar{\mathbf{x}}_m, \bar{\mathbf{x}}_{m'}) = c \exp\left[-\tfrac{1}{2}(\bar{\mathbf{x}}_m - \bar{\mathbf{x}}_{m'})^\top (\bar{\mathbf{x}}_m - \bar{\mathbf{x}}_{m'})\right]$$

Training: we maximize marginal likelihood w.r.t. pseudo-inputs $\bar{\mathbf{X}}$, the projection matrix P, the size c, and the noise σ^2

[1] Vivarelli & Williams, 1999

Dimensionality reduction – selected results

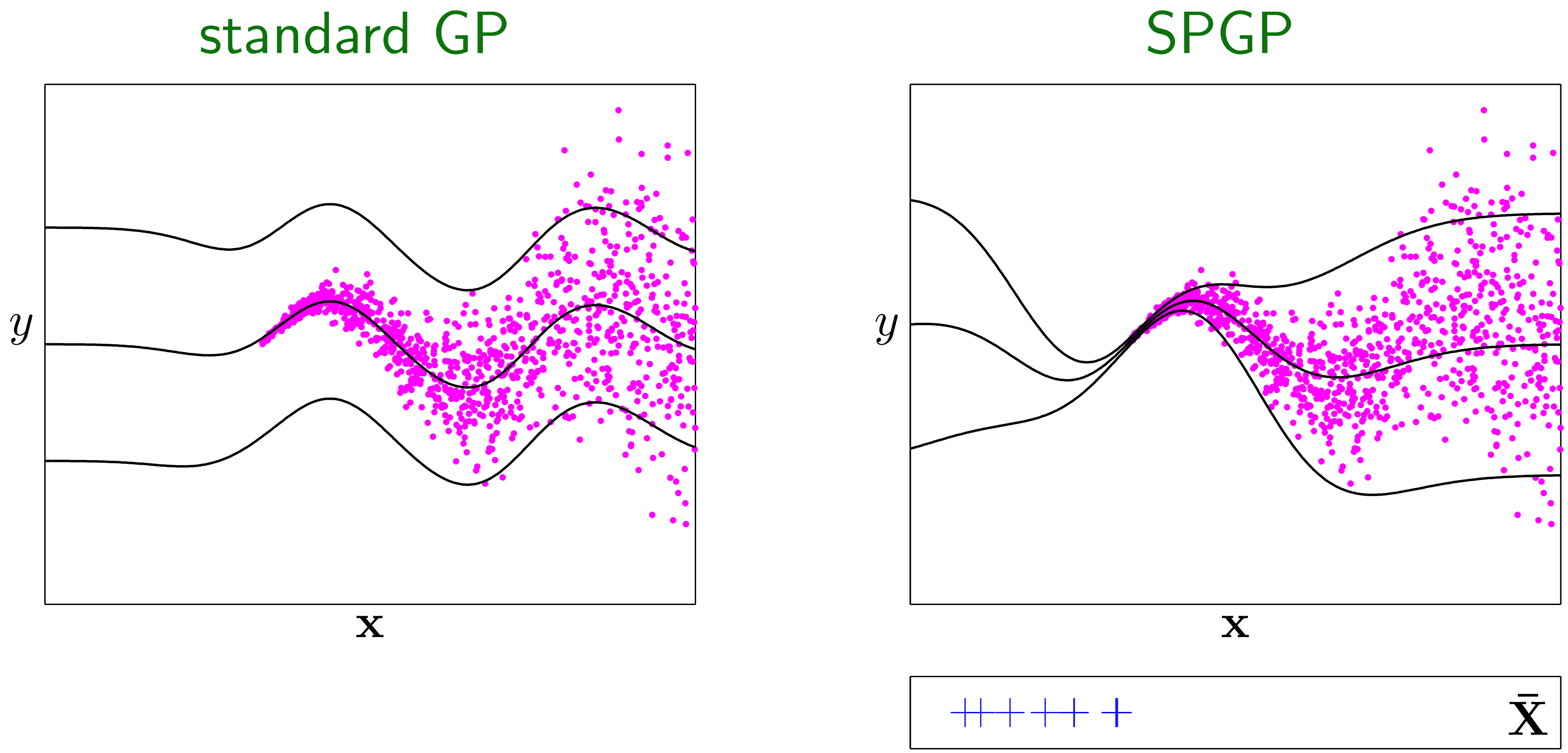
Predictive Uncertainty in Environmental Modeling Competition[1]

`Temp` data set: $D = 106$, $N_{\text{train}} = 7117$, $N_{\text{valid}} = 3558$, $N_{\text{test}} = 3560$

	Validation		Time /s	
Method	NLPD	MSE	Train	Test
SPGP	0.063	0.0714	4420	0.567
+DR 2	0.106(2)	0.0754(5)	180(10)	0.043(1)
+DR 5	0.071(8)	0.0711(7)	340(10)	0.061(1)
+DR 10	0.112(10)	0.0739(12)	610(20)	0.091(1)
+DR 20	0.181(5)	0.0805(7)	1190(50)	0.148(1)
+DR 30	0.191(6)	0.0818(7)	1740(50)	0.206(3)
+PCA 5	0.283(1)	0.1093(1)	200(10)	0.047(2)

[1] `http://theoval.sys.uea.ac.uk/competition/`

Modeling input dependent noise



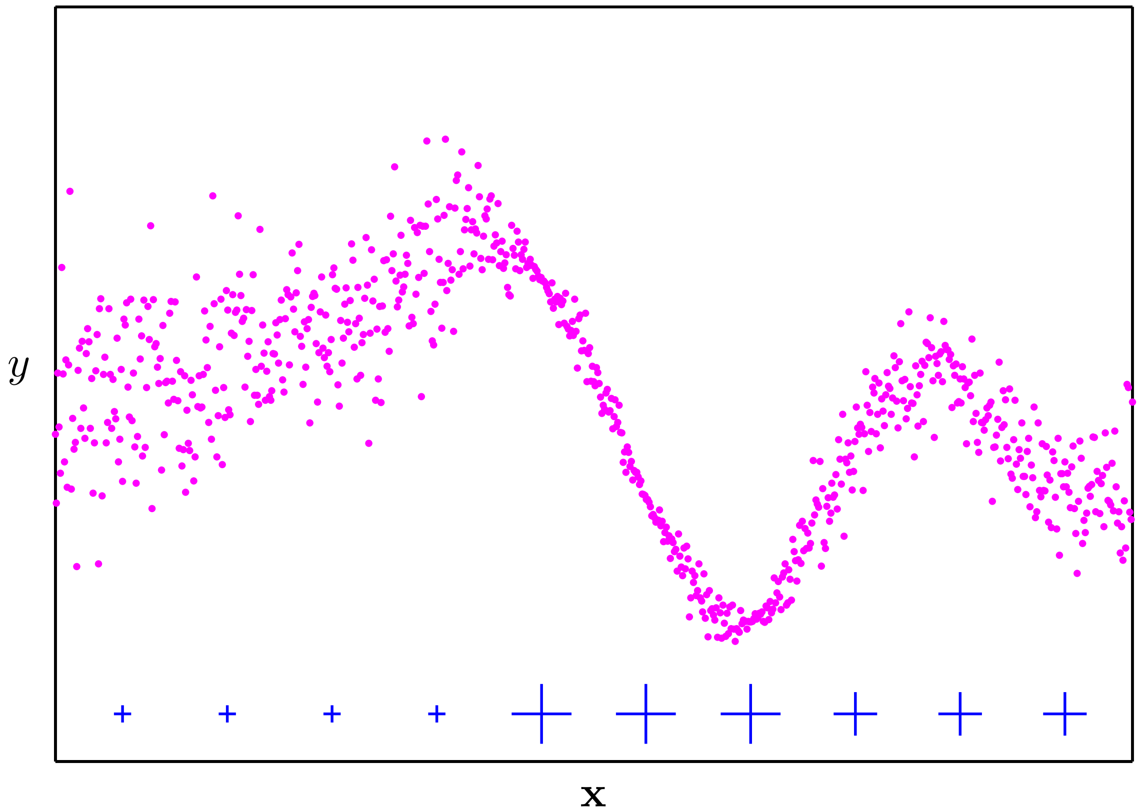
Extra flexibility of SPGP allows some non-stationary effects to be modeled, but in a somewhat limited way

A better solution

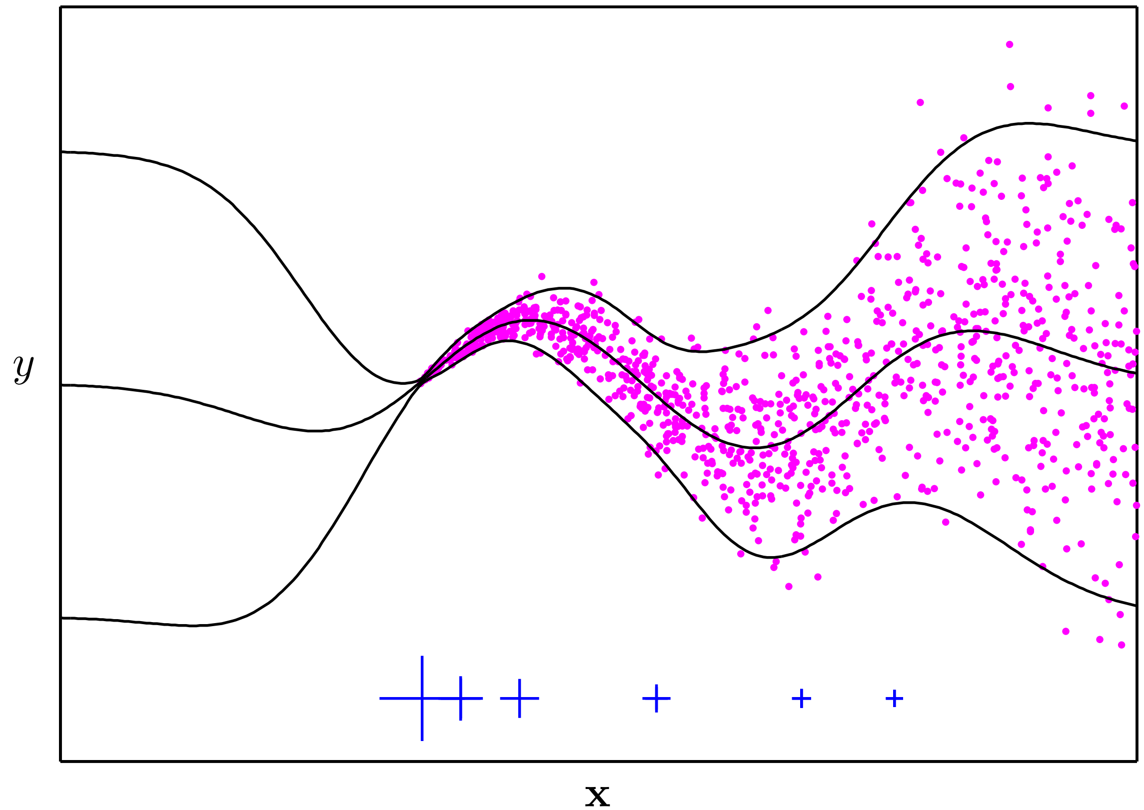
We make a modification to the covariance of the pseudo-inputs:

$$\mathbf{K}_M \to \mathbf{K}_M + \mathrm{diag}(\mathbf{h})$$

$\mathbf{h}$ is a (+ve) vector of uncertainties to 'switch off' pseudo-inputs



Modeling input dependent noise ... revisited



Uncertainties $\mathbf{h}$ are extra parameters to be learned by ML

They adjust to the local noise level, and the pseudo-inputs are not forced left as before

`Temp` data set ... revisited

	Validation		Time /s	
Method	NLPD	MSE	Train	Test
SPGP	0.063	0.0714	4420	0.567
+DR 5	0.071(8)	0.0711(7)	340(10)	0.061(1)
+HS,DR 5	0.077(5)	0.0728(3)	360(10)	0.062(3)

- It was suggested that the `Temp` data set is heteroscedastic
- However SPGP+HS did no better than SPGP
- We took a subset of the data (size 1000), and found an SPGP on the subset significantly outperformed a full GP on the subset
- Indicates SPGP modeling the variable noise well

Limitations and possible extensions

- We have introduced a great deal of flexibility into the GP covariance function

- Care needs to be taken to avoid overfitting these extra parameters

- 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

- All the methods presented can be viewed as GPs with complex parameterized covariance functions
- These developments allow GP methods to be applied to a wide range of data sets
- We can handle a large number of data points, high dimensional input spaces, with variable noise
- The desirable properties of the standard GP are retained – sensible predictive error bars, and a principled determination of hyperparameters
- Performance increases over other methods have been shown on real data sets, including a winning competition entry

Relation of SPGP/FITC to PLV/DTC[1]

SPGP/FITC

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/DTC

Approximate conditional:

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

uncertainty not taken into account – deterministic approximation

Marginal likelihood:

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

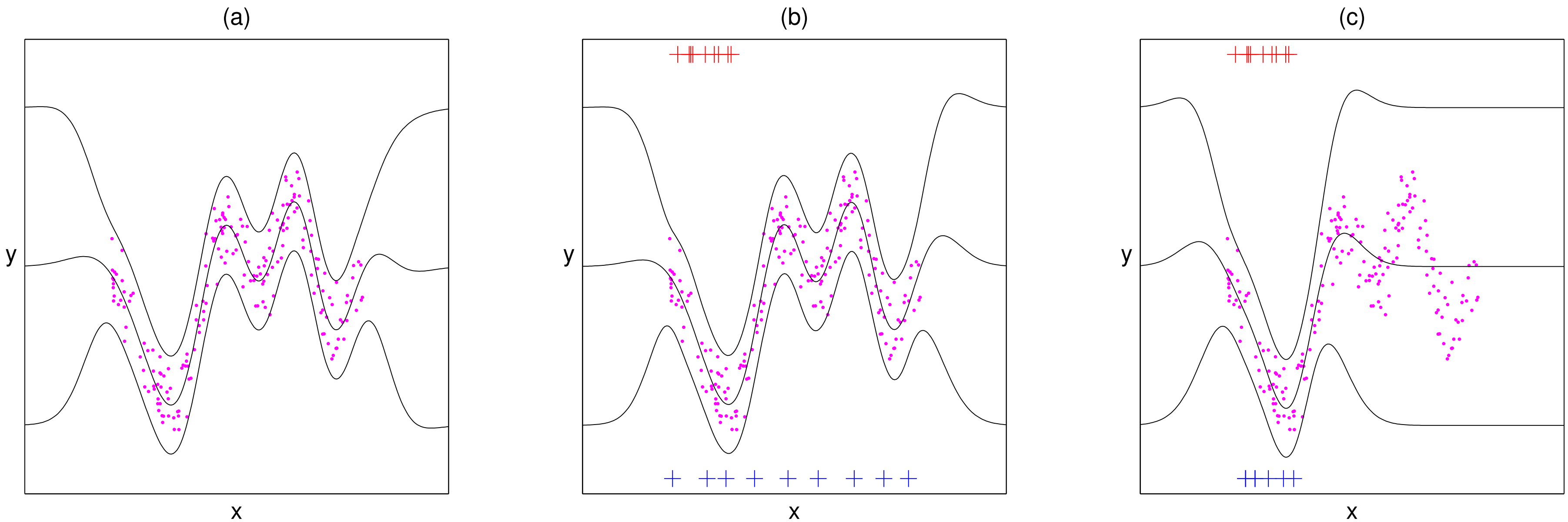
marginal variances decay to σ^2 away from 'active set' points

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/DTC 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

Comparison to RBF networks

The idea of basis functions with movable centres (pseudo-inputs) dates back to RBF networks:

$$f(\mathbf{x}_*) = \sum_m K(\mathbf{x}_*, \bar{\mathbf{x}}_m)\alpha_m$$

The SPGP *mean* predictor could be regarded as an RBF predictor with a certain set of weights $\boldsymbol{\alpha}$:

$$\mu_* = \mathbf{K}_{*M}\mathbf{Q}^{-1}\mathbf{K}_{MN}(\boldsymbol{\Lambda} + \sigma^2\mathbf{I})^{-1}\mathbf{y}$$

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

where $\mathbf{Q} = \mathbf{K}_M + \mathbf{K}_{MN}(\boldsymbol{\Lambda} + \sigma^2\mathbf{I})^{-1}\mathbf{K}_{NM}$

However the SPGP has sensible predictive variances, and a principled ML method for choosing the pseudo-inputs and hyperparameters