Gaussian Processes for Regression: An Overview

Alexis Boukouvalas, based on Dan Cornford's slides

Outline of the talk

Part I

An introduction to Gaussian processes

Good reference: Gaussian Processes for Machine Learning book, free at `http://www.gaussianprocess.org/gpml/chapters/`

GP definition

A Gaussian process is a collection of random variables, over an index set $\mathcal{X}$, with a joint Gaussian distribution.

- It is defined by the mean function, $m(\mathbf{x})$, and covariance function $k(\mathbf{x}, \mathbf{x}')$.
- We will write $\boxed{f(\mathbf{x}) \sim N[m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')]}$, where $\overset{D\times 1}{\mathbf{x}}, \mathbf{x}' \in \mathcal{X}$.
- $\mathcal{X}$, is the index set, or input range, e.g. $[0, 1] \times [0, 1]$, of the GP.
- Typically we assume $m(\mathbf{x}) \equiv 0$ – but we'll revisit this later.

GPs with noise free observations

Assume we have points $\overset{D\times n}{X}$ at which we know $\overset{n\times 1}{f}$, and points $\overset{D\times m}{X^*}$ where we wish to know f^*:

By the properties of a multivariate Gaussian distribution we can write:

$$\begin{bmatrix} f \\ f^* \end{bmatrix} \sim N\left[0, \begin{bmatrix} k(X,X) & k(X,X^*) \\ k(X^*,X) & k(X^*,X^*) \end{bmatrix}\right]$$

without knowing about f anywhere else. So the conditional distribution is:

$$\overset{m\times 1}{f^*}\,|X^*, X, f \sim N[\overset{m\times n}{k(X^*,X)}\overset{n\times n}{k(X,X)^{-1}}\overset{n\times 1}{f},$$
$$\overset{m\times m}{k(X^*,X^*)} - \overset{m\times n}{k(X^*,X)}\overset{n\times n}{k(X,X)^{-1}}\overset{n\times m}{k(X,X^*)}]$$

GPs with noisy observations

We observe $y_i = f(\mathbf{x}_i) + \epsilon$, with $\epsilon \sim N(0, \sigma_n^2)$.

The joint distribution is thus:

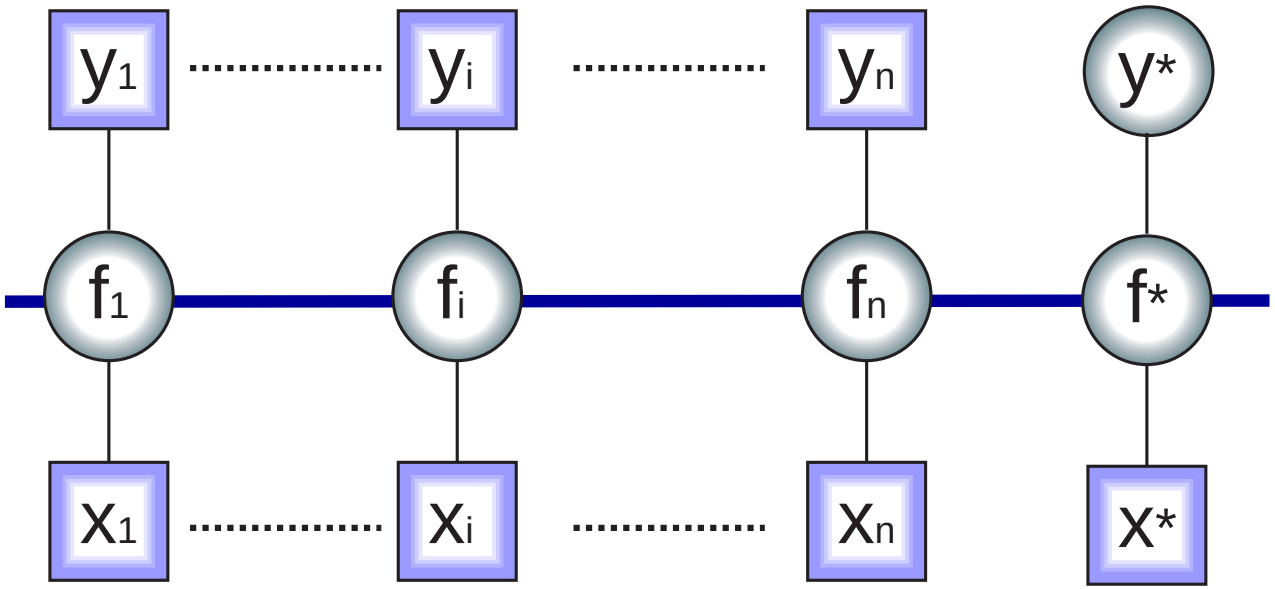
$$\begin{bmatrix} y \\ f^* \end{bmatrix} \sim N\left[\begin{matrix}0 \\ 0'\end{matrix}, \begin{bmatrix} k(X,X) + \sigma_n^2 I & k(X,X^*) \\ k(X^*,X) & k(X^*,X^*) \end{bmatrix}\right]$$

The conditional distribution is:

$$\overset{m\times 1}{f^*} | X^*, X, y \sim N[\overset{m\times n}{k(X^*,X)}(\overset{n\times n}{k(X,X) + \sigma_n^2 I})^{-1}\overset{n\times 1}{y},$$
$$\overset{m\times m}{k(X^*,X^*)} - \overset{m\times n}{k(X^*,X)}(\overset{n\times n}{k(X,X) + \sigma_n^2 I})^{-1}\overset{n\times m}{k(X,X^*)}]$$

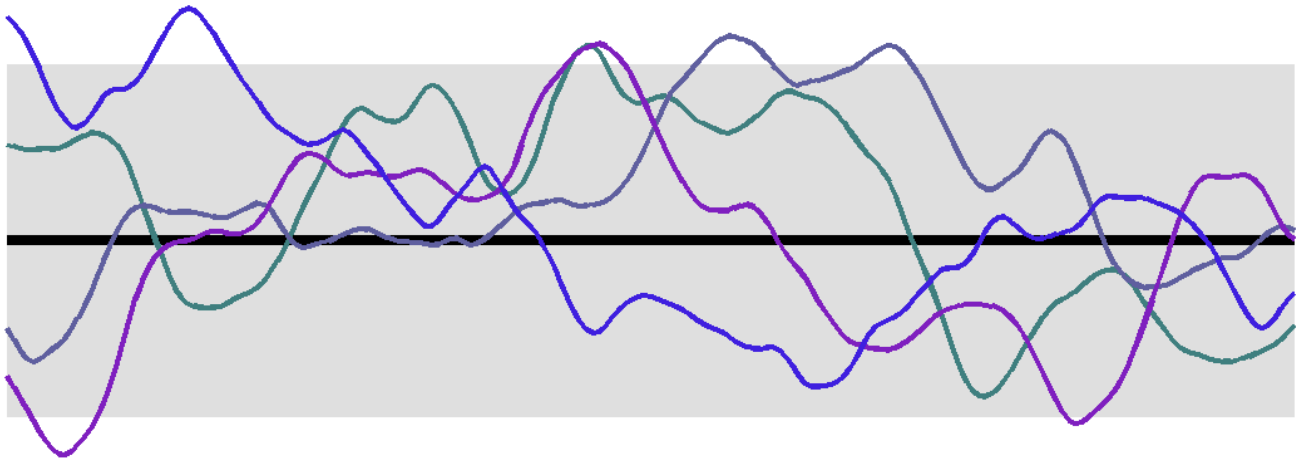
So far all we have used are the properties of multivariate Gaussians!

GPs and Bayesian inference



- Gaussian processes are generally used as prior models, placed over functions $f(\mathbf{x})$ – c.f $N[\mu, \Sigma]$ over vectors.
- The graphical model for a GP with noisy observations looks simple ...

GPs specify priors over functions



The GP is specified by $m(\mathbf{x})$, and covariance function $k(\mathbf{x}, \mathbf{x}')$.

The covariance between outputs is written in terms of functions of the inputs:

$$Cov[f(\mathbf{x}), f(\mathbf{x}')] = k(\mathbf{x}, \mathbf{x}')$$

The covariance function determines the properties of f.

GPs in weight space

- Define a linear model $\mathbf{y} = \overset{n\times 1}{g(X)} + \epsilon = \overset{n\times N}{\Phi(X)}^T \overset{N\times 1}{\mathbf{w}} + \epsilon$.
- Assuming a Gaussian prior $\mathbf{w} \sim N[0, \Sigma_p]$, we find:

$$p(\mathbf{w}|X, \mathbf{y}) \propto p(\mathbf{y}|X, \mathbf{w})p(\mathbf{w}) \sim N\left[\frac{1}{\sigma_n^2} \overset{N\times N}{A^{-1}} \overset{N\times n}{\Phi(X)} \overset{n\times 1}{\mathbf{y}}, A^{-1}\right]$$

  where $A = \sigma_n^{-2}\Phi(X)\Phi(X)^T + \Sigma_p^{-1}$.
- Now $p(f^*|X^*, X, \mathbf{y})$

$$= \int p(f^*|X^*, \mathbf{w})p(\mathbf{w}|X, \mathbf{y})\mathrm{d}\mathbf{w}$$

$$\sim N\left[\frac{1}{\sigma_n^2} \overset{m\times N}{\Phi(X^*)}^T \overset{N\times N}{A^{-1}} \overset{N\times n}{\Phi(X)} \overset{n\times 1}{\mathbf{y}}, \Phi(X^*)^T A^{-1}\Phi(X^*)\right]$$

GPs: function space / weight space

The GP can be seen as being derived in two ways:

As a prior over functions:

$$E[f(\mathbf{x})] = m(\mathbf{x}), \qquad Cov[f(\mathbf{x}), f(\mathbf{x}')] = k(\mathbf{x}, \mathbf{x}'),$$

or as a linear (feature space) model with priors on the weights:

$$p(\mathbf{w}), \qquad \phi(\mathbf{x}).$$

The relation between the views can be seen by noting:

$$\overset{1\times 1}{k(\mathbf{x}, \mathbf{x}')} = \overset{1\times N}{\phi(\mathbf{x})^T} \overset{N\times N}{\Sigma_p} \overset{N\times 1}{\phi(\mathbf{x}')}$$

Mercer's theorem tells us $\forall$ positive definite kernels $k(\mathbf{x}, \mathbf{x}')$ $\exists$ an (∞) (eigen)expansion in terms of some basis functions $\phi(\mathbf{x})$.

Part II

The covariance function and kernels:
Priors and learning from data.

Parameters of covariance functions

- The choice of the covariance function determines the function class over which the GP prior is set.
- For a variety of reasons we often prefer isotropic, stationary covariance functions $k(\mathbf{x}, \mathbf{x}') = k(r)$ where $r = |\mathbf{x} - \mathbf{x}'|$, i.e. radial basis functions.
- A general form for an isotropic, stationary covariance function is: $k(r) = \sigma_f^2 \cdot \psi(r, l) + \sigma_g^2 \delta(r)$.
- σ_f^2 is called the process variance, l is the process length scale, σ_g^2 is the nugget variance.
- Not all functions $\psi()$ are valid covariance functions: the function must be positive semi-definite to produce valid covariance matrices.

Convergence in random variables

If $\{z_n\}$ is a sequence of random variables $\{z_n\}$ is said to converge to Z (also a random variable) with

- mean square convergence:

$$z_n \xrightarrow{m.s.} Z \iff \lim_{n\to\infty} E[|z_n - Z|^2] = 0$$

- almost sure convergence:

$$z_n \xrightarrow{a.s.} Z \iff p\left(\lim_{n\to\infty} z_n = Z\right) = 1$$

Mean square convergence is to do with expected values of the random variable and almost sure convergence with samples from the process.

Convergence in stochastic processes: continuity

Let $f(\mathbf{x})$ be a stationary GP with $\mathbf{x}_0$ a fixed point, then:

$$f(\mathbf{x}) \xrightarrow{m.s.} f(\mathbf{x}_0) \iff E[f(\mathbf{x}_i)f(\mathbf{x}_j)] \to E[f(\mathbf{x}_0)^2] \qquad \mathbf{x}_i, \mathbf{x}_j \to \mathbf{x}_0 \ ,$$

defines mean square convergence of the GP.

The stationary GP $f(\mathbf{x})$ is m.s. continuous at $\mathbf{x}$ if:

$$f(\mathbf{x}+r) \xrightarrow{m.s.} f(\mathbf{x}) \iff \lim_{r \to 0} E[|f(\mathbf{x}+r) - f(\mathbf{x})|^2] = 0 \ .$$

$f(\mathbf{x})$ is m.s. continuous at $\mathbf{x}_0$ if and only if $k(\mathbf{x}, \mathbf{x}')$ is continuous at $(\mathbf{x} = \mathbf{x}_0, \mathbf{x}' = \mathbf{x}_0)$.

- For stationary processes we only need to check continuity of k(r) as $r \to 0$.

Convergence in stochastic processes: continuity II

Let $f(\mathbf{x})$ be a stationary GP with $\mathbf{x}_0$ a fixed point, then:

$$f(\mathbf{x}+r) \xrightarrow{a.s.} f(\mathbf{x}) \iff p\left(\lim_{r\to 0} f(\mathbf{x}+r) = f(\mathbf{x})\right) = 1$$

defines almost sure continuity of the GP.

The stationary GP $f(\mathbf{x})$ is a.s. continuous, i.e. has continuous sample paths if:

$$k(r) \leq \frac{\alpha}{|\log(r)|^{1+\beta}}, \qquad \forall r < 1, \text{for some finite } \alpha > 0, \beta > 0$$

This is not a tight bound. Abrahamsen claims that all stationary GPs with continuous covariance functions possess continuous sample paths with probability one, which implies m.s. continuity.

Convergence in stochastic processes: differentiability

A stationary GP, $f(\mathbf{x})$, is m.s. differentiable to the v'th order if the $2v$'th partial derivative of $k(r)$ exists and is finite at $r = 0$:

$$\frac{\partial^v f(\mathbf{x}_i)}{\partial x_i^v}$$

exists in the m.s. sense iff:

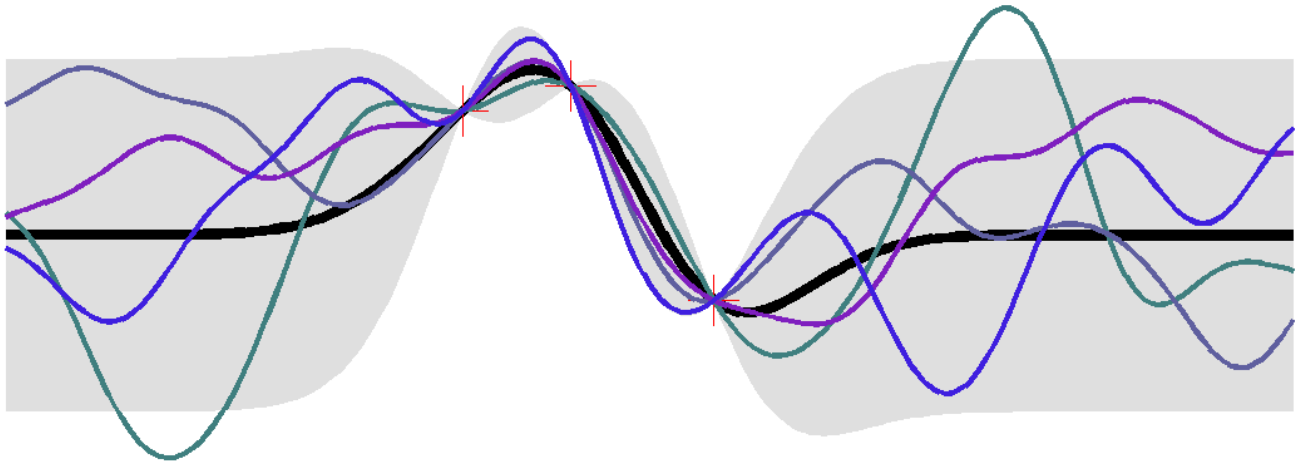
$$\left.\frac{\partial^{2v} k(r)}{\partial r^{2v}}\right|_{r=0} < \infty \ .$$

The sample paths are a.s. differentiable if the corresponding doubly differentiated covariances are a.s. continuous.

What matters about covariance functions?

- For stationary GPs almost all covariance functions produce continuous sample paths, and means.
- The functional form of the covariance function controls the roughness (differentiability) of the sample paths and mean.
- If you don't know this apriori it might be very difficult to infer from data!
- Where possible understanding the real processes that generated the observed field can help a lot.

The squared exponential covariance function

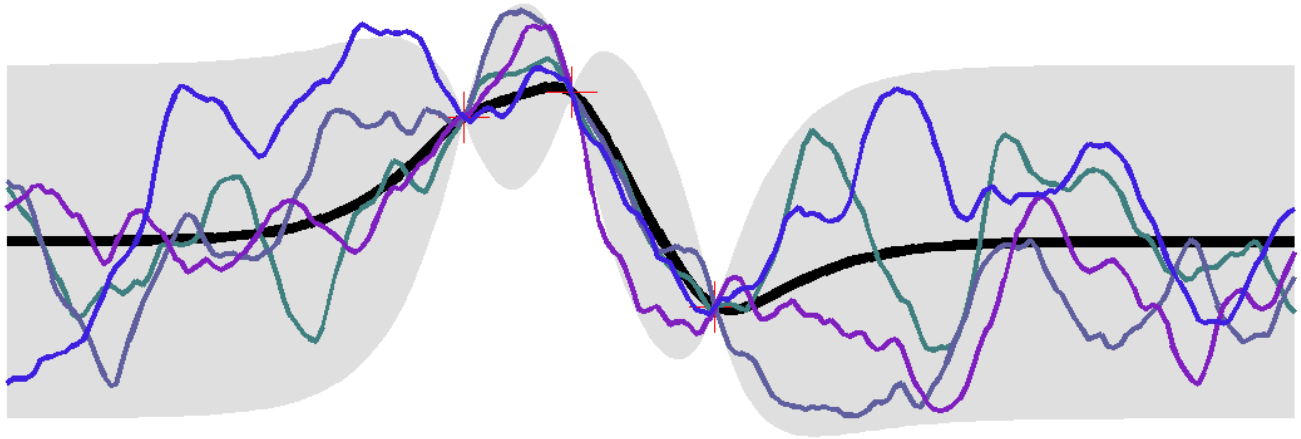


This covariance function is ubiquitous in machine learning, but has some major issues! It is defined by:

$$k(r) = \sigma_f^2 \cdot \exp\left(-\frac{r^2}{2l^2}\right) + \sigma_g^2 \delta(r)$$

It produces realisations that are analytic, and often gives badly conditioned covariance matrices.

The Matern covariance function

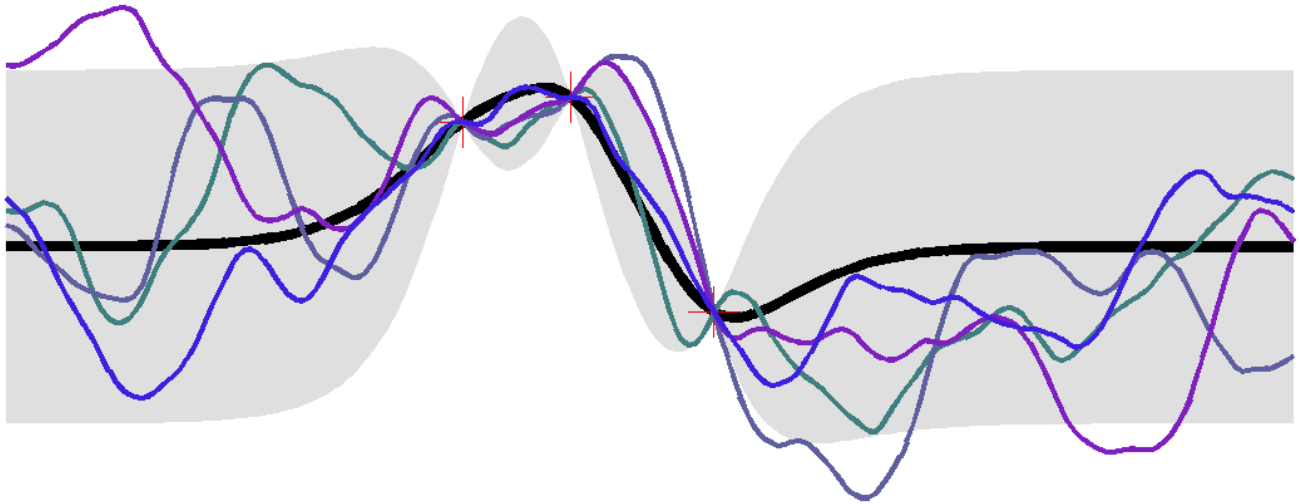


This covariance function has an extra parameter, ν, that controls the roughness of the process:

$$k(r) = \sigma_f^2 \cdot \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu} \cdot r}{l} \right)^{\nu} K_\nu \left(\frac{\sqrt{2\nu} \cdot r}{l} \right) + \sigma_g^2 \delta(r)$$

It is very flexible, but computationally demanding.

The Matern covariance function

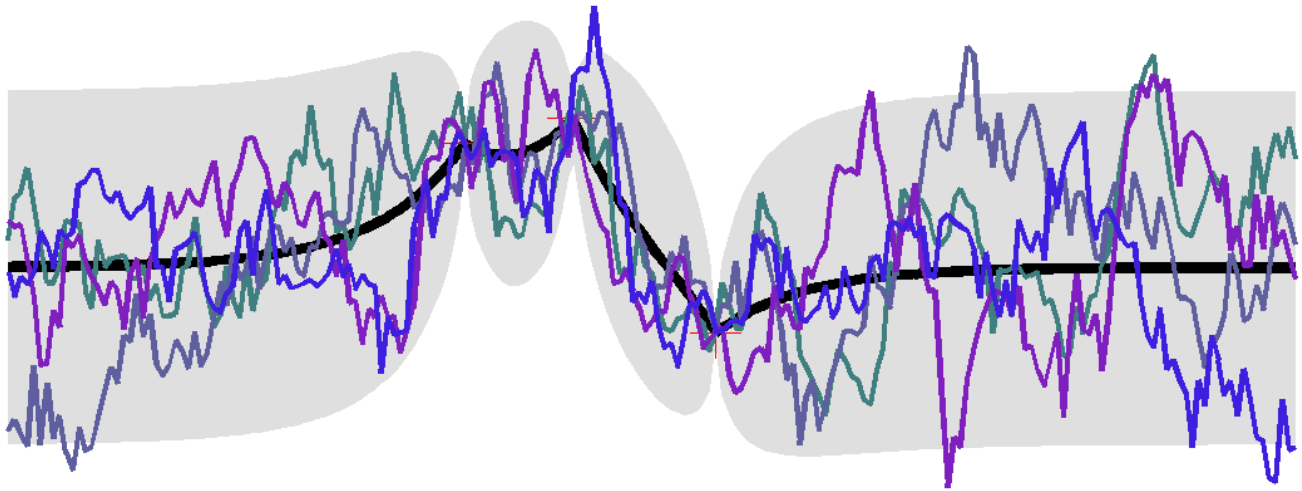


This covariance function has an extra parameter, ν, that controls the roughness of the process:

$$k(r) = \sigma_f^2 \cdot \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu} \cdot r}{l} \right)^{\nu} K_\nu \left(\frac{\sqrt{2\nu} \cdot r}{l} \right) + \sigma_g^2 \delta(r)$$

It is very flexible, but computationally demanding.

The exponential covariance function

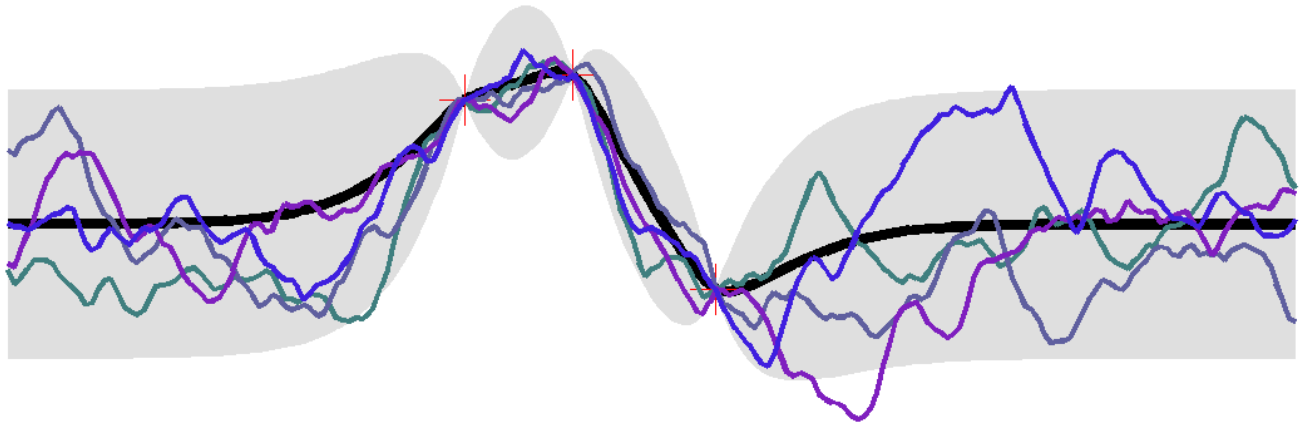


This covariance represents process that are quite rough:

$$k(r) = \sigma_f^2 \cdot \exp\left(-\frac{r}{l}\right) + \sigma_g^2 \delta(r)$$

It can be derived from the Ornstein-Uhlenbeck process, which is defined by the following SDE: $dX_t = -0.5X_t dt + dW_t$.

Other covariance functions



- We can create new covariance functions since the sum or product of any two covariance functions is also a valid covariance function.
- In many applications it makes sense to believe that we are observing the sum / product of several processes acting at potentially different scales.

Inference in GPs

- Recall:

$$p(\mathbf{w}|X,\mathbf{y}) = \frac{p(\mathbf{y}|X,\mathbf{w})p(\mathbf{w})}{\int p(\mathbf{y}|X,\mathbf{w})p(\mathbf{w})d\mathbf{w}} \ .$$

The quantity $p(\mathbf{y}|X) = \int p(\mathbf{y}|X,\mathbf{w})p(\mathbf{w})d\mathbf{w}$ is called the evidence (or marginal likelihood). In the function space view we would normally write this as:

$$\overset{\text{evidence}}{p(\mathbf{y}|X,\boldsymbol{\theta})} = \int \overset{\text{likelihood}}{p(\mathbf{y}|X,f)} \overset{\text{prior}}{p(f|X,\boldsymbol{\theta})} df \ ,$$

where typically $\boldsymbol{\theta} = \{\sigma_f^2, l, \sigma_g^2\}$ – the hyper-parameters.

- For a GP:

$$2\log(p(f|X,\boldsymbol{\theta})) = -f^T K(\boldsymbol{\theta})^{-1} f - \log(|K(\boldsymbol{\theta})|) - n\log(2\pi) \ .$$

Hyper-parameter determination in GPs

- So we will seek to maximise $p(\mathbf{y}|X,\boldsymbol{\theta})$ to find the most probable parameters, given our assumptions.
- Now $-2\log(p(\mathbf{y}|X,\boldsymbol{\theta})) =$

$$\overset{\text{misfit}}{\mathbf{y}^T(K(\boldsymbol{\theta})+\sigma_n^2 I)^{-1}\mathbf{y}} + \overset{\text{complexity}}{\log(|K(\boldsymbol{\theta})+\sigma_n^2 I|)} + \overset{\text{const}}{n\log(2\pi)}\,.$$

- In general minimising this expression requires non-linear optimisation; thus derivatives wrt $\boldsymbol{\theta}$ are required.
- In fully Bayesian settings we would ideally sample from $p(\boldsymbol{\theta}|X,\mathbf{y})$ having defined some appropriate priors over $\boldsymbol{\theta}$.
- Plotting profile marginal likelihoods can be instructive to see how well identified a parameter is.

Part III

Specific topics for Gaussian processes

Validation of Stochastic model emulators

Defining properness

A scoring rule is proper if $S(Q, Q) \geq S(P, Q)$ for all P and Q. It is strictly proper if $S(Q, Q) = S(P, Q)$ only if $Q = P$.

Intuition

Using the true generative distribution will get the maximum score. However a useless prediction can achieve a score close to maximum. Properness is a good starting point but is not enough by itself.

Classification of scoring rules

Local

The score only depends on the value of the predictive density at the true target value. e.g. logarithmic score.

Non-Local

Other characteristics of the predictive distribution are taken into account. Distance-sensitive rules are a special case which favour to have the bulk of the probability mass near the true target value. e.g. CRPS.

Validation: How good is my model?

Standardised Mean Squared Error (MSE)

With regards to the mean only. We utilise a standardised form (divided by the sample variance of the observations):

$$\mathrm{SMSE} = \frac{1}{N\mathrm{Var}[y]} \sum_{i=1}^{N} \left(E[t_*^i] - y^i\right)^2,$$

where $E[t_*^i]$ the GP predictive mean for test point $i \in \{1, \ldots, N\}$, y^i the observation at that point and $\mathrm{Var}[y]$ the sample variance of the observations.
Best error $= 0$. Anything over 1 means trivial model (predicting mean of training data).

Validation: Looking at variance (univariate) 1/2

Non-negative Likelihood Predictive Distribution (NLPD)

Weighs the errors on the mean by the predictive variance, therefore penalising incorrect variance estimates:

$$\mathrm{NLPD} = -\log\, p(y_*|D, x_*) = \frac{1}{2N_v}\sum_{i=1}^{N_v}\left[\log(2\pi\sigma_i^2) + \frac{(y_i - t_i)^2}{\sigma_i^2}\right].$$

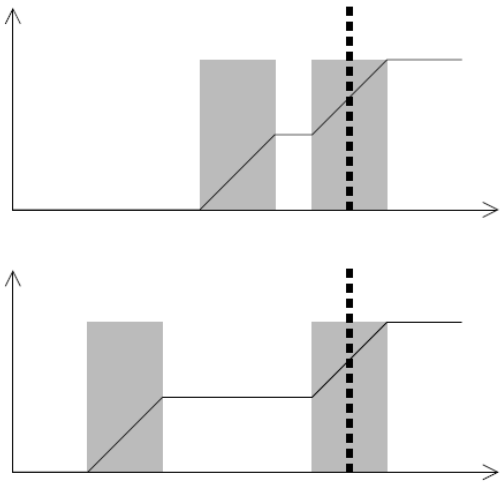
The NLPD is also known as the logarithmic score. The standardised log likelihood (SLL) is defined as the NLPD minus the NLPD of the trivial model which predicts the mean and variance of the training data. The SLL will be approximately zero for simple methods and negative for better methods.

Validation: Looking at variance (univariate) 2/2

Continuous Ranked Probability Score (CRPS) for a Gaussian predictive distribution

$$\mathrm{CRPS} = \sigma \left[\frac{1}{\sqrt{\pi}} - 2\phi\left(\frac{x-\mu}{\sigma}\right) - \frac{x-\mu}{\sigma}\left(2\Phi\left(\frac{x-\mu}{\sigma}\right) - 1\right)\right],$$

Distance based metric. ϕ and Φ the pdf and cdf of the standard normal distribution respectively.



Distance: NLPD same, CRPS much lower for bottom, from Kohonen and Suomela (2005).

Multivariate Metrics

CRPS $\rightarrow$ Energy

$$\text{Energy}(\mathbf{z}, \mathbf{x}) = \frac{1}{2(M-1)} \sum_{i=1}^{M-1} ||\mathbf{z}_i - \mathbf{z}_{i+1}|| - \frac{1}{M} \sum_{i=1}^{M} ||\mathbf{z}_i - \mathbf{y}||,$$

where $||.||$ the Euclidean norm and $\mathbf{z}_i$, $i \in \{1, \ldots, M\}$ samples from the predictive distribution. Distance based metric.

NLPD $\rightarrow$ Dawid score

$$\text{Dawid} = \log |\Sigma| + (\mathbf{y} - \mathbf{t})^T \Sigma^{-1} (\mathbf{y} - \mathbf{t})$$

$|\ldots|$ denotes the determinant and Σ the covariance matrix of the joint predictive distribution at the set of test points. Proper scoring rule. Difference of two models related to Bayes factor.

Validation: Mahalanobis distance

Not proper score rule but useful diagnostic.

$$D_{MD} = (\mathbf{y} - E[\mathbf{t}_*])^T \mathrm{Cov}[\mathbf{t}_*, \mathbf{t}_*]^{-1} (\mathbf{y} - E[\mathbf{t}_*]),$$

- Theoretical sampling distribution is known for Gaussian and Student-t processes. For GP : χ^2 distribution with n degrees of freedom where n is the size of the test set.
- Theoretical mean is the number of test points.
- The difference of the empirical Mahalanobis distance to its theoretical mean value can be used as a validation metric.
- Lower values than the theoretical mean can signify an underconfident GP where the predictive variance it too high.
- Higher values on the other hand typically occur when the GP predictions are overconfident.

Comparing validation metrics

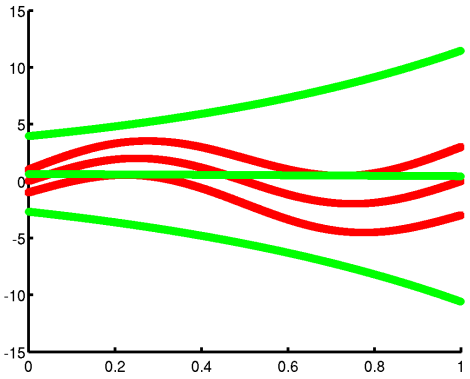


Figure: Best Mahalanobis Variance too big, mean wrong

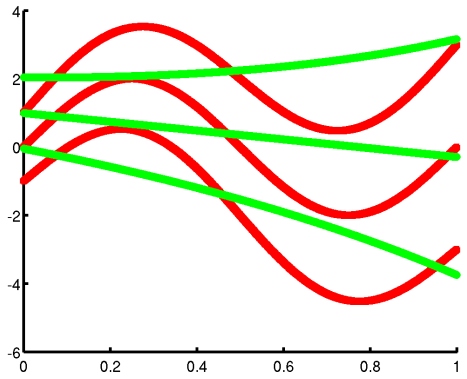


Figure: Best Dawid Variance spot on, mean wrong

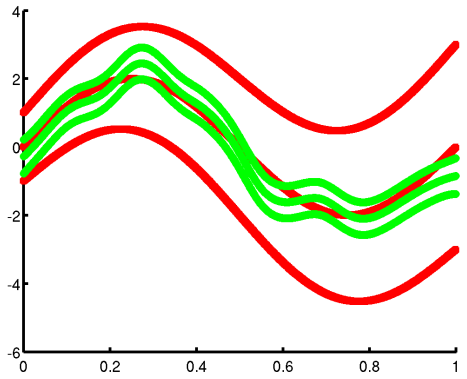


Figure: Best CRPS Variance too small, mean spot on

Which scoring rule to use?

Logarithmic score

Kohonen (2005): In a pure inferential setting where we care about maximising our information about the targets.

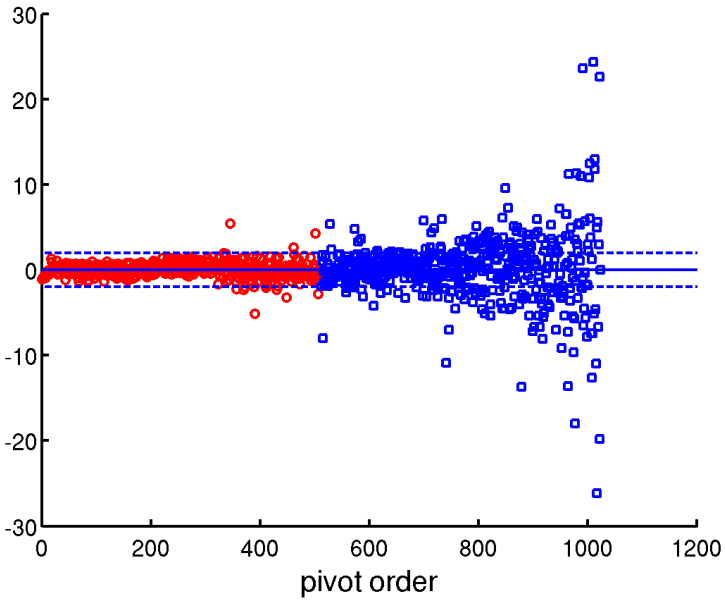
Distance-based

However in many practical situations, a distance-based or other non-local score may make more sense. Epstein (1969) gives an example in meteorology - temperature forecast A (from low to high ranges) $(0.1, 0.3, 0.5, 0.1)$, forecast B $(0.5, 0.3, 0.1, 0.1)$, the fourth class corresponding to the observed temperature. Same local score but forecast B would lead to preparation for much colder weather $\rightarrow$ low utility since it incurs high-cost for policy maker.

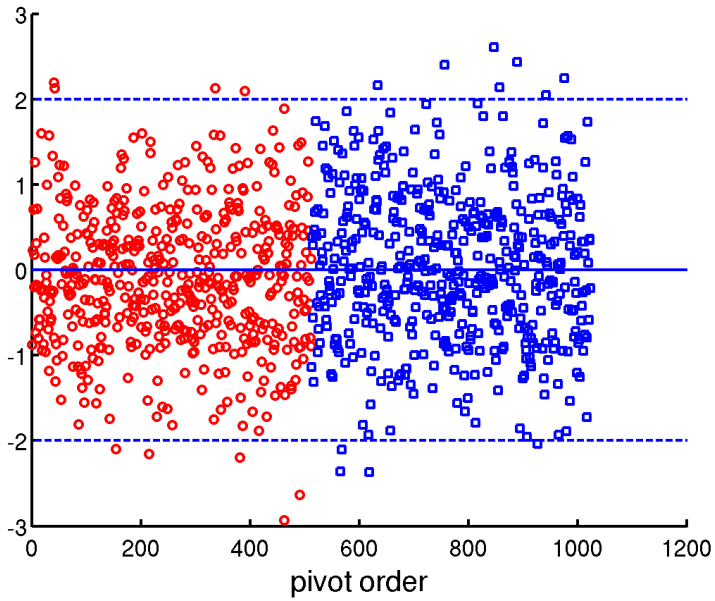
Validation: Mahalanobis Decomposition

- Can also be decomposed to trace source of error using the Cholesky decomposition to $D_{MD} = \mathbf{v}^T\mathbf{v}$.
- $\mathbf{v}$ are uncorrelated errors with theoretical distribution $N(0, 1)$.
- Can use Pivoted Choleksy Decomposition (PCD) for meaningful order of x axis.
- In PCD the data is permuted such that the first element is the one with the largest variance, the second element is the one with the largest predictive variance conditioned on the first element and so on.
- Errors early in sequence are typically on test points far from training data where the predictive variance is high and possible causes include non-stationarity of the function output and misidentification of the process-variance/nugget terms.
- Errors at the end of the sequence are typically from test points close to training points or test points close to other test points and point to a problem in the identification of the correlation structure.

Validation: Mahalanobis Decomposition Example



(a) Model 1



(b) Model 2

Figure: PCD(x) vs Uncorrelated MD error (y). Theoretical distribution is $N(0, 1)$ with two standard deviations plotted.

What do you think the problem is with Model 1?

Validation conclusions

Local vs Non-local

- Local rules place equal emphasis on all information regarding the target value and ignore other features of the forecast.
- So for many practical problems non-local scoring rules may better reflect the end utility of making decisions.

Recommendations

- Mahalanobis useful to check validity of emulator and understand source of error.
- Ranking emulators - a problem of model choice.
- Multiple proper scoring rules exist - which one to use is a decision problem.

Validation: References

Properness and list various proper scoring rules

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. Journal of the American Statistical Association, 2004.

Chapter 6 extensively discusses proper scoring rules in the context of comparing different models.

L. S. Bastos. Validating Gaussian Process Models in Computer Experiments. PhD thesis, University of Sheffield, 2010.

Differences between local and global scoring rules:

J. Kohonen and J. Suomela. Lessons learned in the challenge : Making predictions and scoring them. Machine learning challenges : Evaluating Predictive Uncertainty, PASCAL Machine Learning Challenges Workshop, 2006.

Heteroscedastic Gaussian Process

- Two GPs coupled to predict mean and input-dependent variance.
- Tricky to do inference since not tractable.
- Can do Monte Carlo or some approximation.
- Most Likely method described next.

Heteroscedastic Coupled Gaussian Process model

Notation: Two sets of observations r=replica observations, s single model evaluations.

1. We estimate a standard homoscedastic GP: $\mathbf{G_H}$ by maximum likelihood on the two sets of observations $t = \{t_r, t_s\}$. Set $\mathbf{G}_\mu$ =$\mathbf{G_H}$.
2. We train a GP on the log(variance) $\mathbf{G}_\sigma$.
   - For set r correct bias due to log transformation.
   - For set s we sample from $\mathbf{G}_\mu$ to estimate the variance at that point.
3. Estimate the heteroscedastic GP $\mathbf{G}_\mu$ to jointly predict the mean and variance.

$$\mu_* = K^*(K + RP^{-1})^{-1}t$$

$$\Sigma_* = K^{**} + R^* - K^{*^T}(K + RP^{-1})^{-1}K^*$$

   where $R = \mathrm{diag}[r(x_1) \dots r(x_N)]$ the most likely variance estimate from $\mathbf{G}_\sigma$. $P = \mathrm{diag}(n_1 \dots n_N)$ the number of replicates at each training point.
4. If s non empty, repeat from step 3.

Heteroscedastic Coupled Gaussian Process model

Notation: Two sets of observations r=replica observations, s single model evaluations.

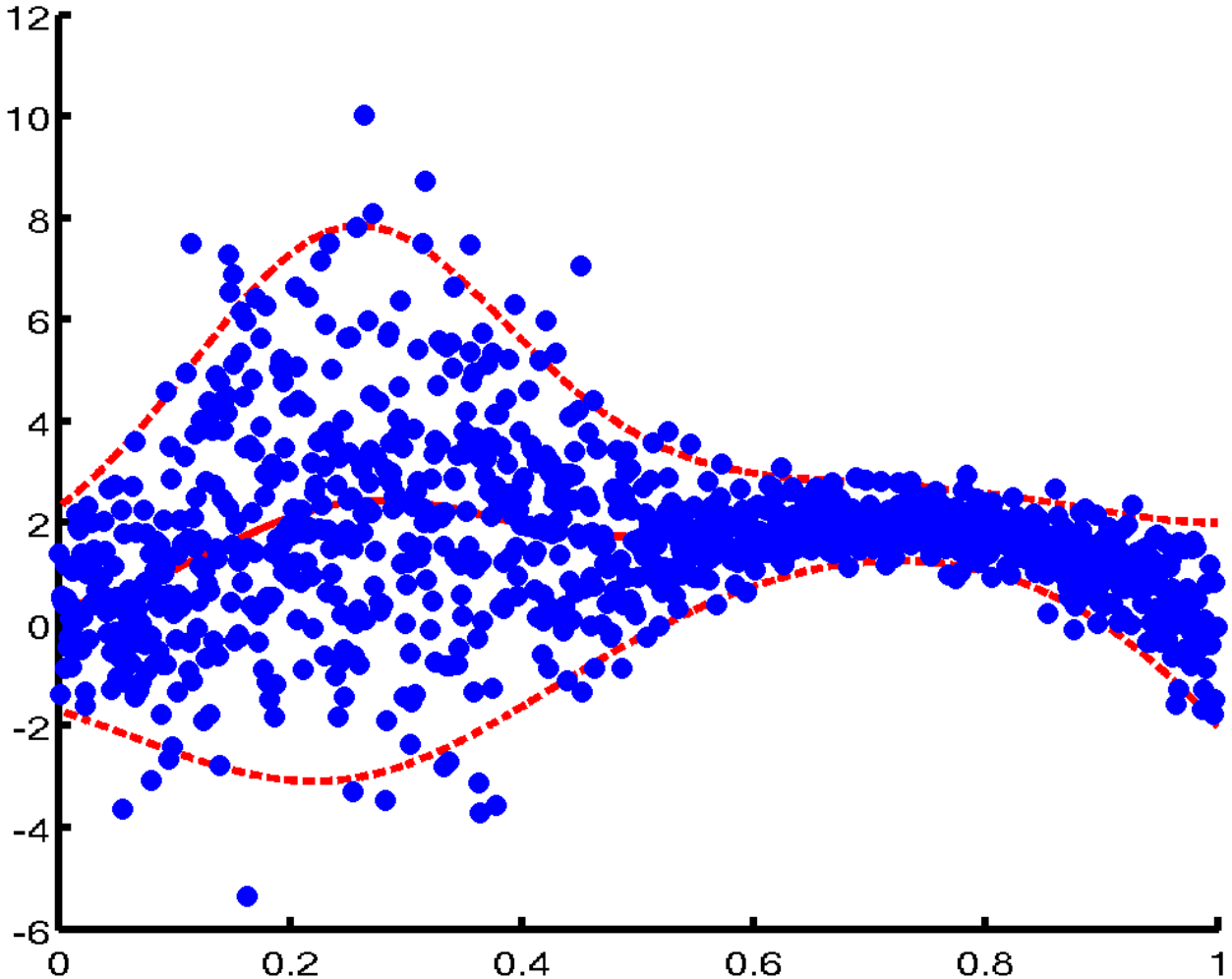
1. We estimate a standard homoscedastic GP: $\mathbf{G_H}$ by maximum likelihood on the two sets of observations $t = \{t_r, t_s\}$. Set $\mathbf{G}_\mu$ =$\mathbf{G_H}$.
2. We train a GP on the log(variance) $\mathbf{G}_\sigma$.
   - For set r correct bias due to log transformation.
   - For set s we sample from $\mathbf{G}_\mu$ to estimate the variance at that point.
3. Estimate the heteroscedastic GP $\mathbf{G}_\mu$ to jointly predict the mean and variance.

$$\mu_* = K^*(K + RP^{-1})^{-1}t$$
$$\Sigma_* = K^{**} + R^* - K^{*^T}(K + RP^{-1})^{-1}K^*$$

   where $R = \mathrm{diag}[r(x_1) \ldots r(x_N)]$ the most likely variance estimate from $\mathbf{G}_\sigma$. $P = \mathrm{diag}(n_1 \ldots n_N)$ the number of replicates at each training point.
4. If s non empty, repeat from step 3.

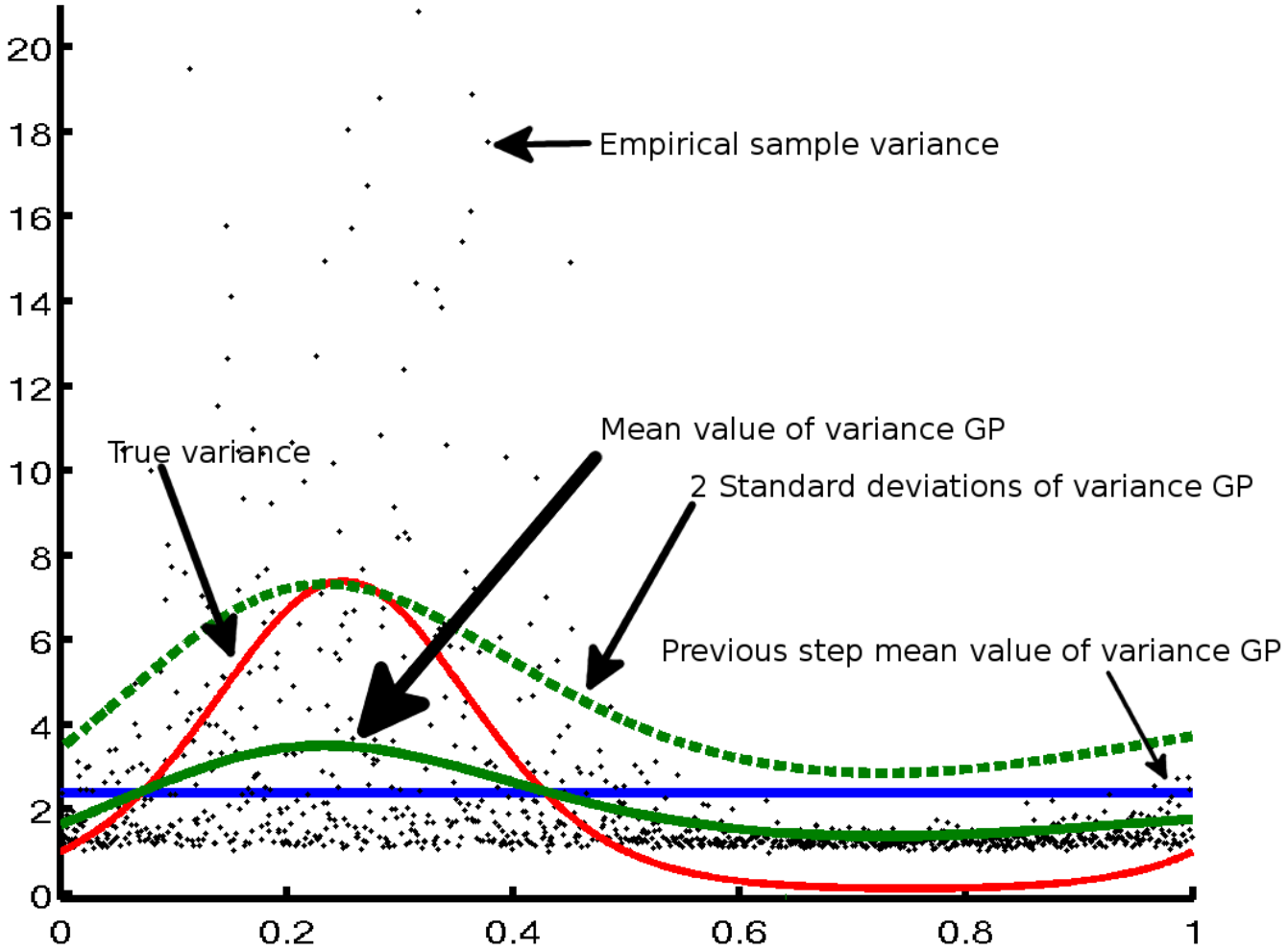
Example

Figure: Yuhba function with 1080 observations.



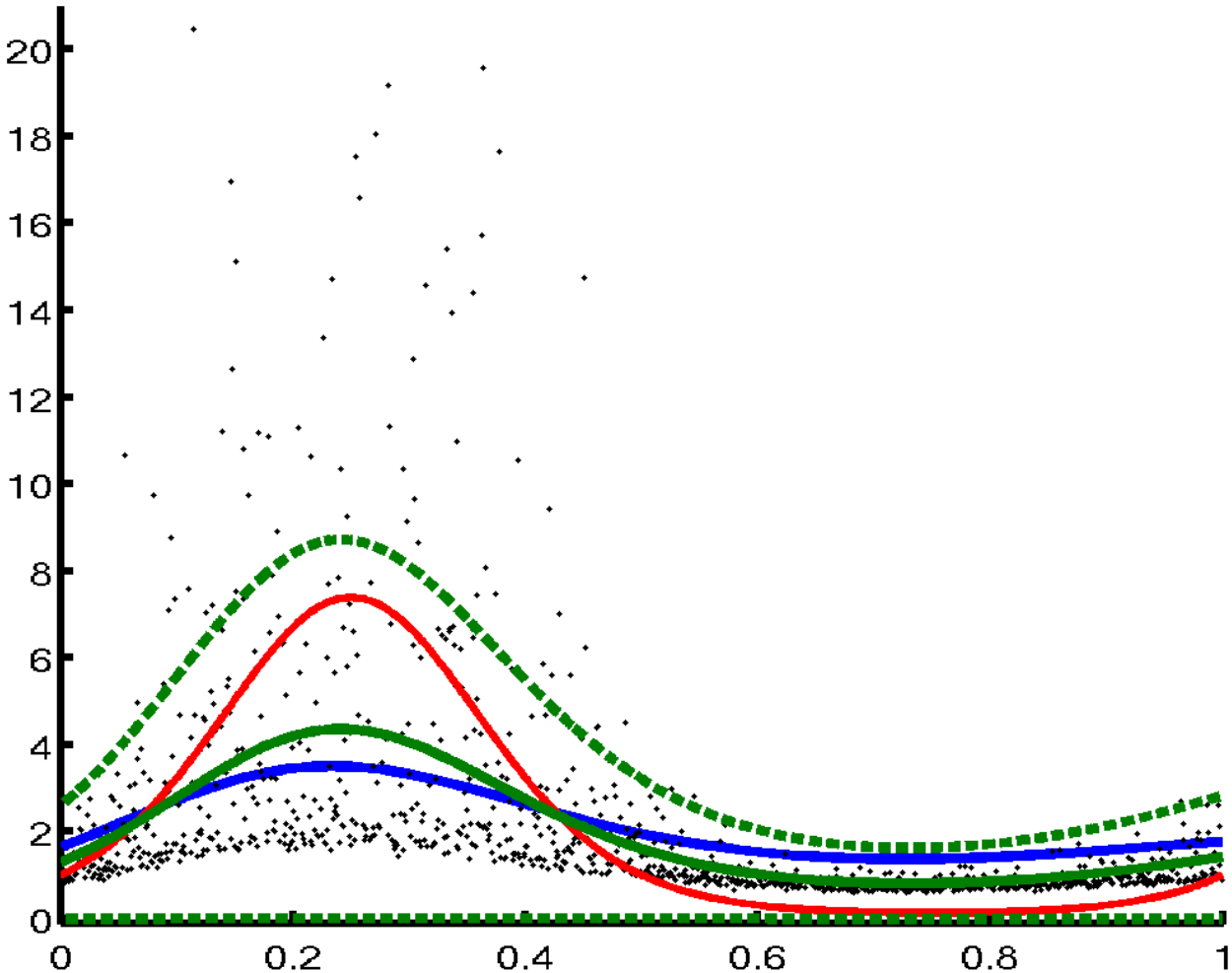
Example

Figure: Step 1



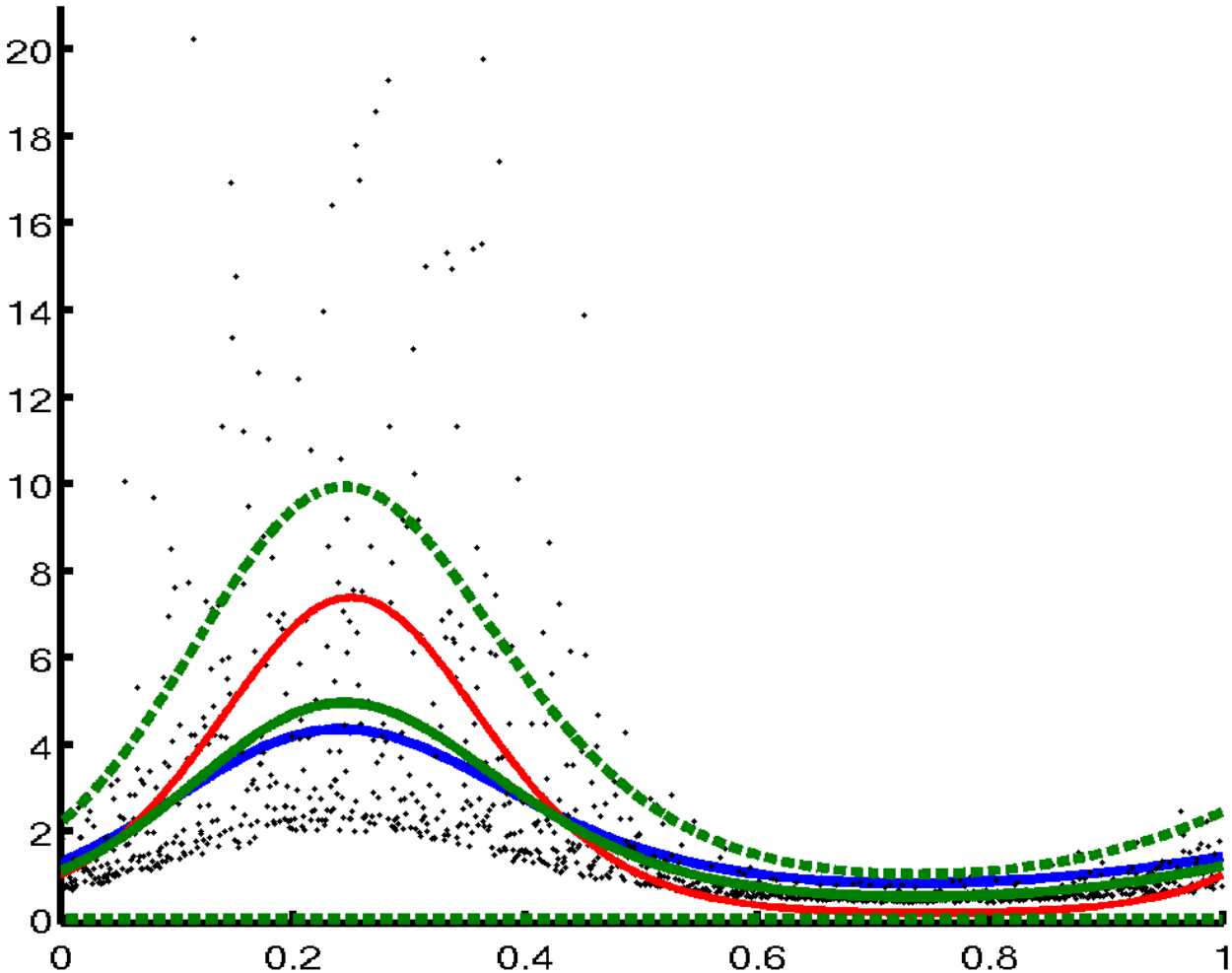
Example

Figure: Step 2



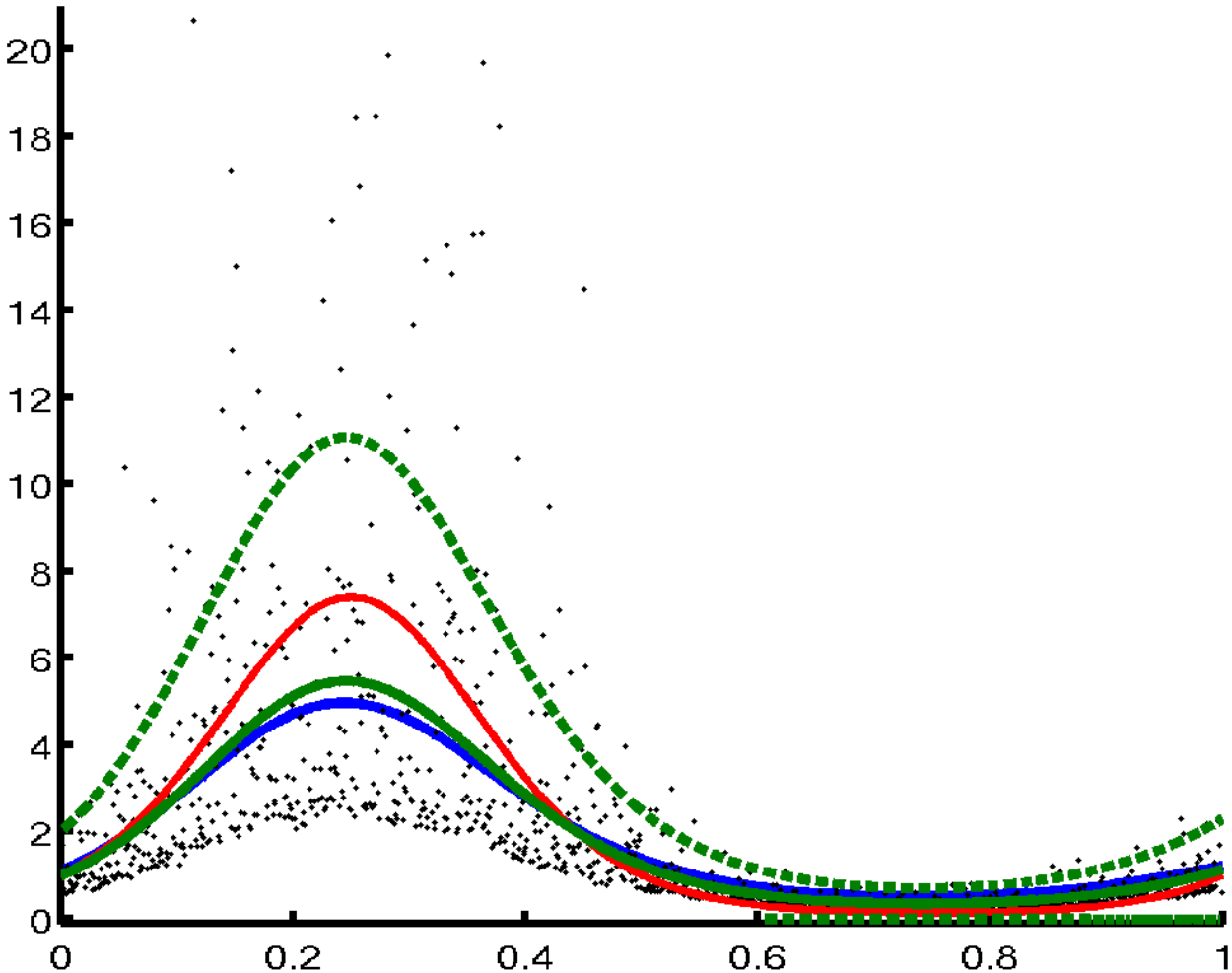
Example

Figure: Step 3



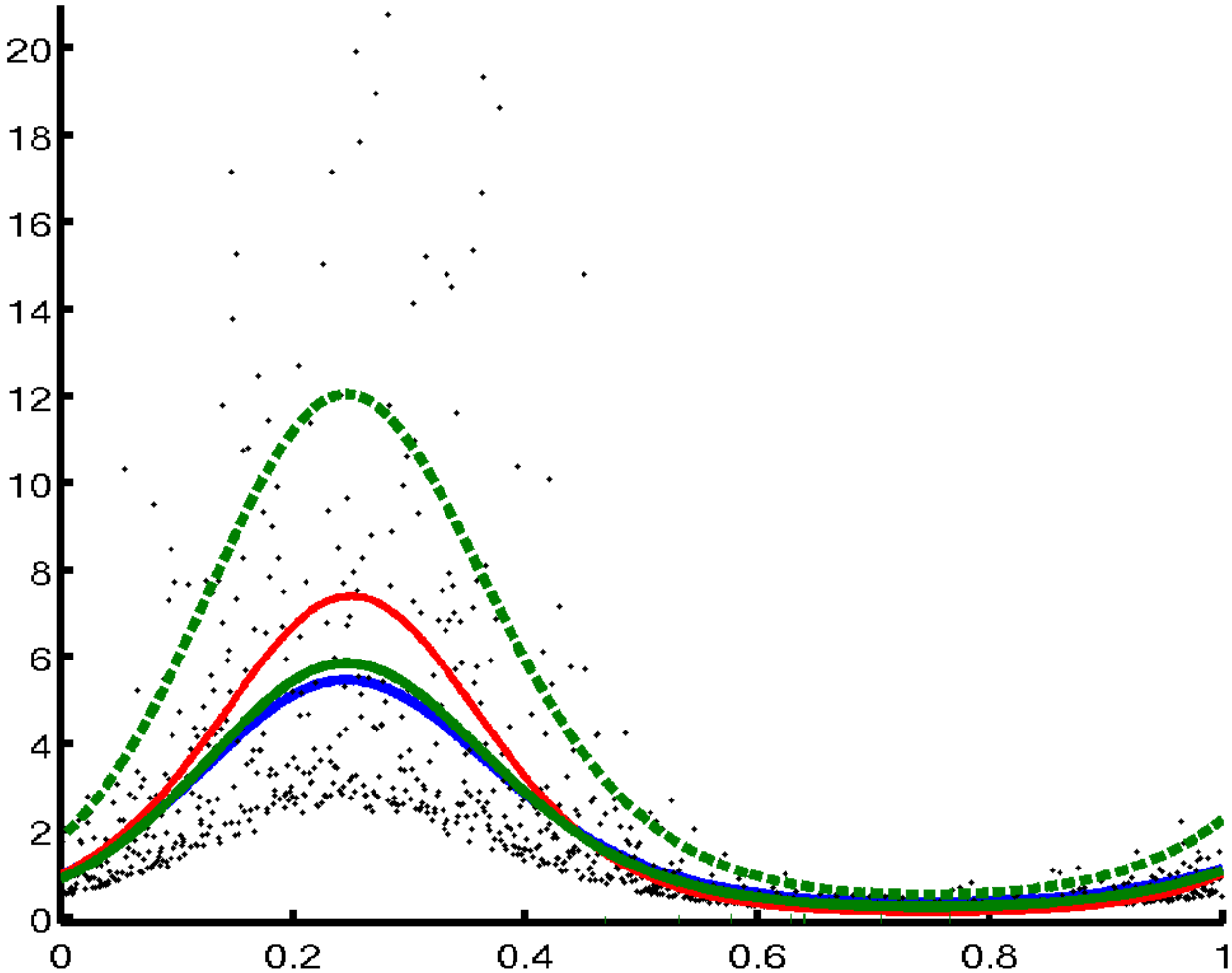
Example

Figure: Step 4



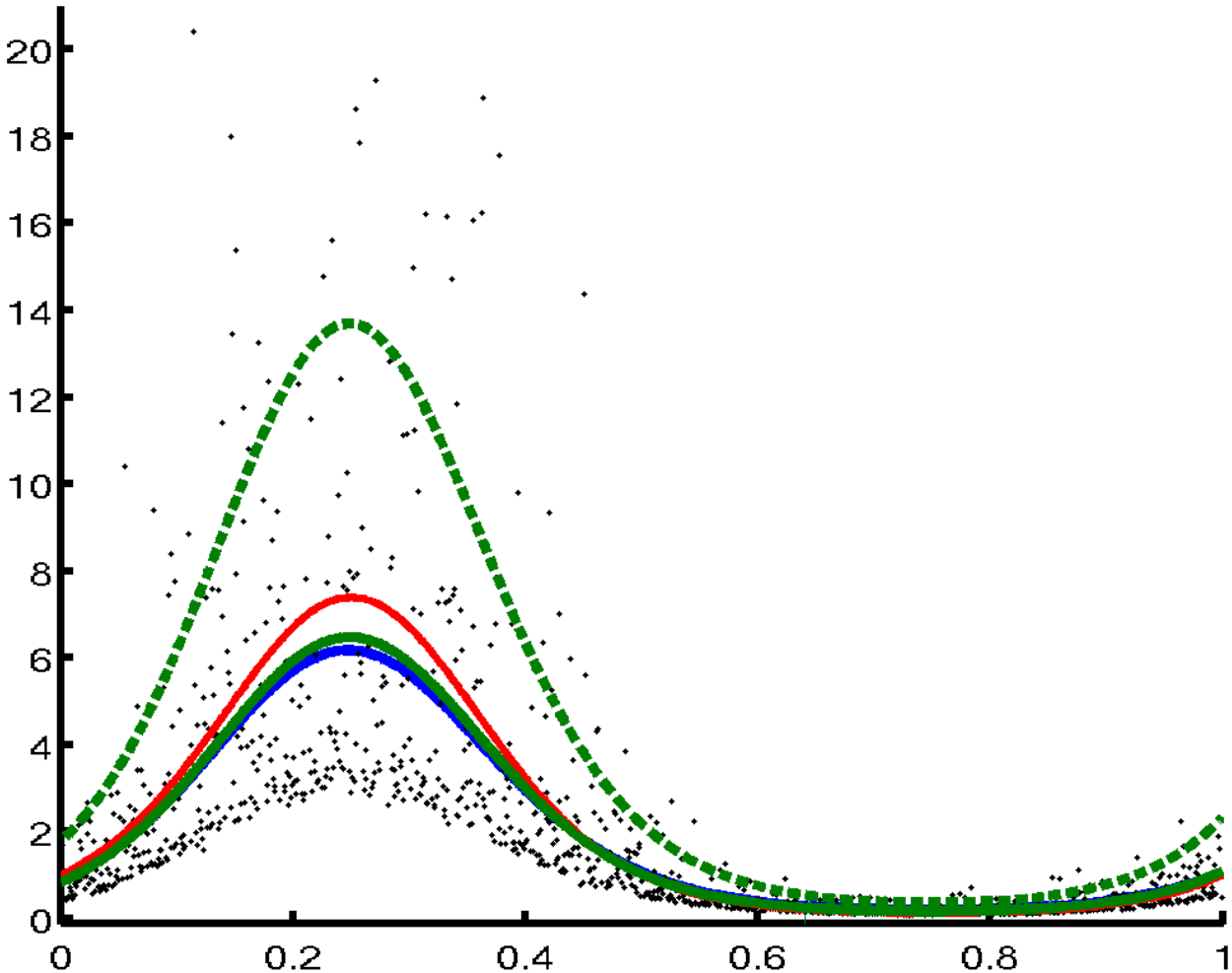
Example

Figure: Step 5



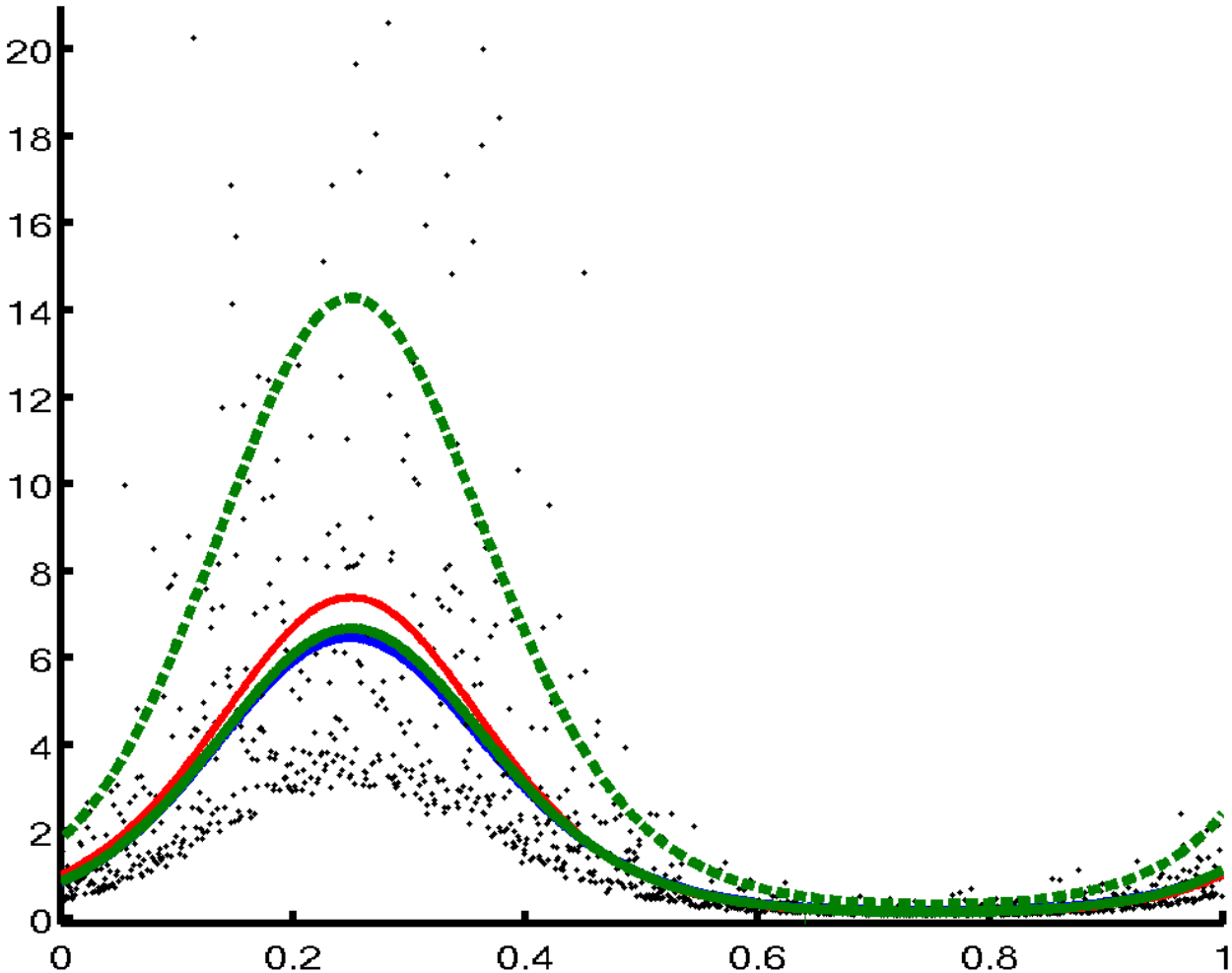
Example

Figure: Step 7



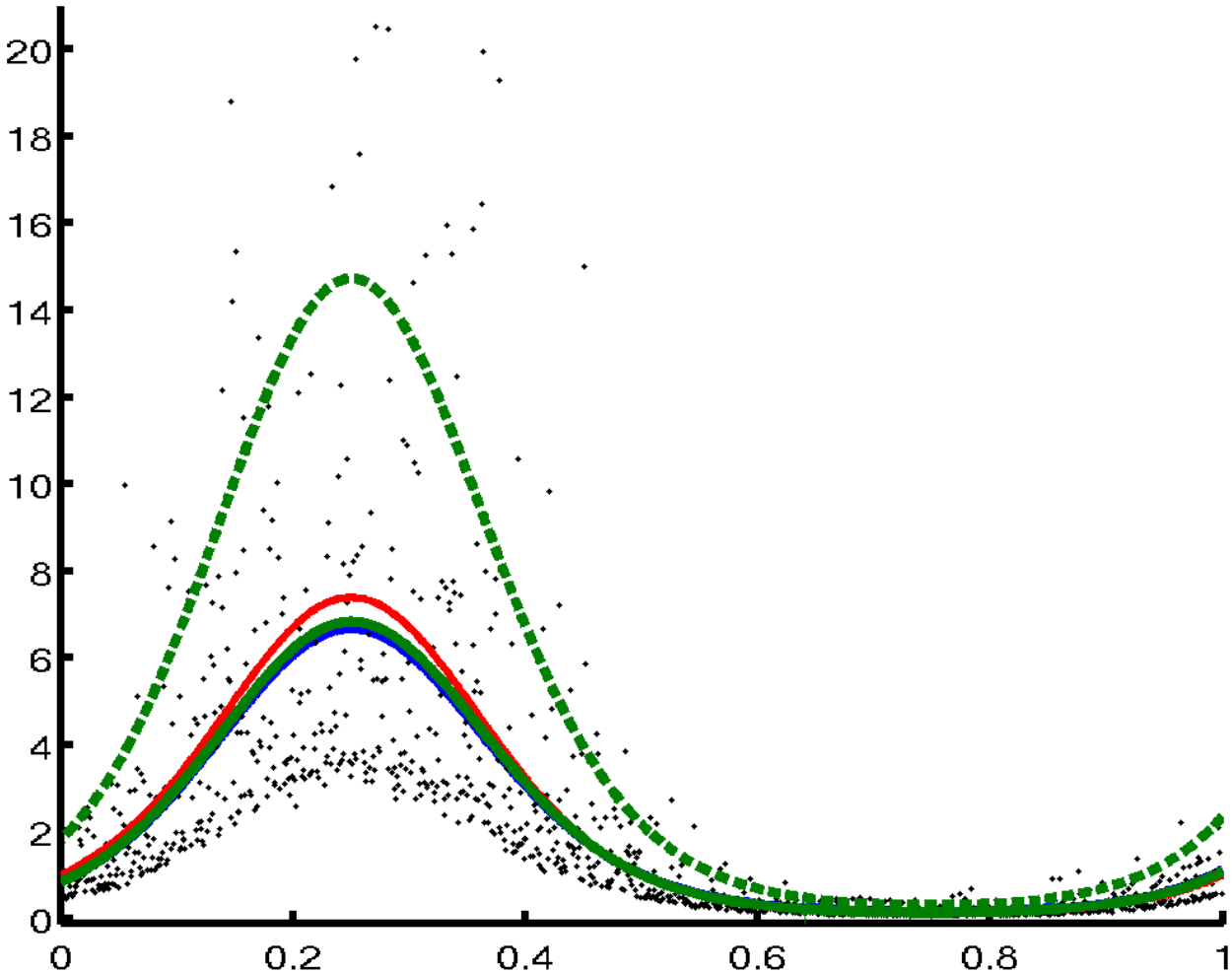
Example

Figure: Step 8



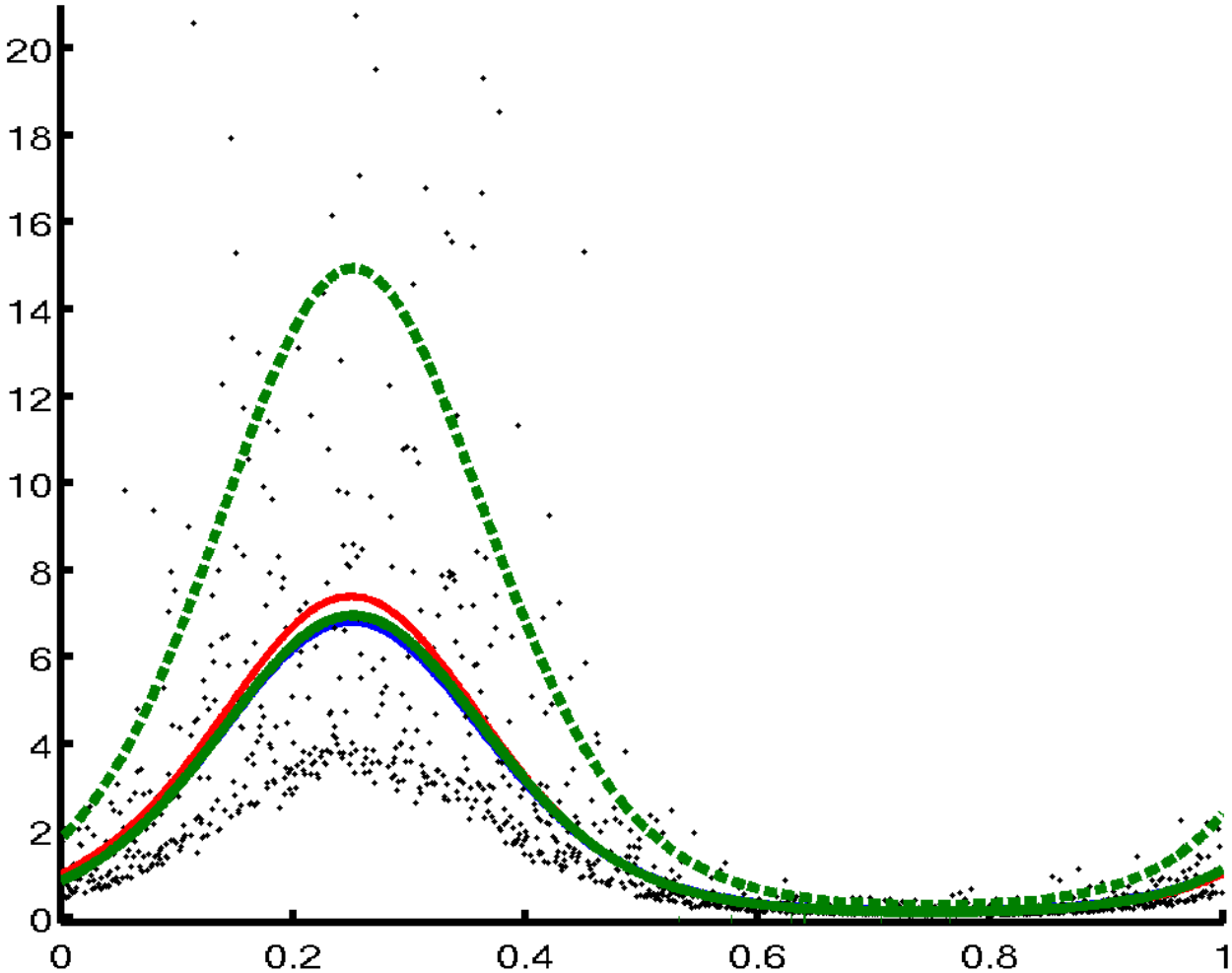
Example

Figure: Step 9



Example

Figure: Step 10



Experimental Design

Joint Likelihood Model

- Coupled model too complex for design calculations.
- Use parametric deterministic variance model.
- Optimisation of the mean and variance model parameters proceeds jointly $\rightarrow$ tractable optimal design calculations.
- Efficient inference with replicated observations.

Joint Likelihood Model

Crucial simplification: consideration of only deterministic variance models. The heteroscedastic GP prior is thus:

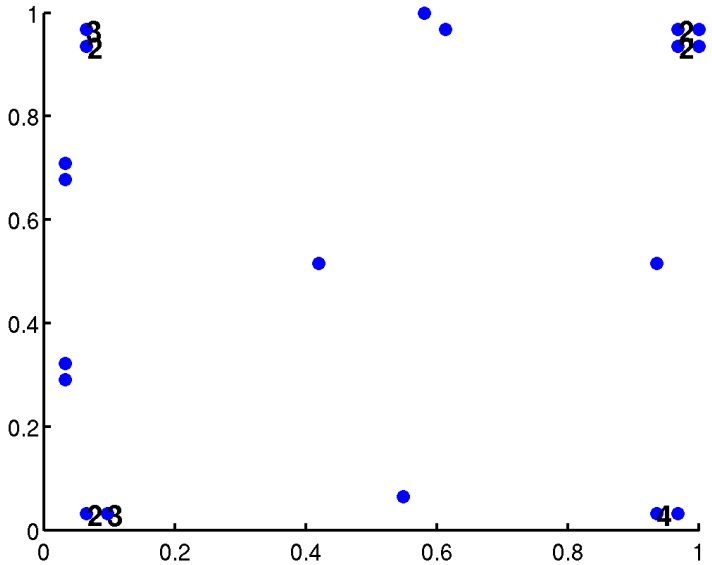
$$p(\mu|\theta,\beta) = N\left(0, K_\theta + \mathrm{diag}(\exp(f_{\sigma^2}(x,\beta))P^{-1}\right),$$

where $f_{\sigma^2}(x,\beta)$ is the deterministic variance model.
The joint log likelihood of the sample mean $\hat{\mu}$ and variance s^2 for N observations:

$$\log p(\hat{\mu}, s^2|\mathbf{X},\theta,\beta) = \left(\sum_{i=1}^{N} \log p(s_i^2|\beta, x_i, n_i)\right) + \log N\left(\hat{\mu}|0, K_\theta + RP^{-1}\right),$$

where K_θ the GP covariance function with parameters θ, R the diagonal matrix with elements $\exp\left(f_{\sigma^2}(x_i,\beta)\right)$.

Optimal Design for Heteroscedastic Gaussian Process Regression with replicated observations



- Design to minimise parameter uncertainty $\rightarrow$ D-optimality
- Minimise Fisher information of design ξ:

$$\mathcal{F}(\xi) = E\left[\frac{\partial^2}{\partial\theta^2}\ln L(X|\theta,\xi)\right]$$

- Analytic solution derived for GP with parametric variance model.

Joint Likelihood: Fisher Information

The FIM for a design ξ is defined as:

$$\mathcal{F}(\xi) = \int \left(\frac{\partial^2}{\partial \theta^2} \ln \left[L(X|\theta, \xi) \right] \right) L(X|\theta, \xi) \, \partial X,$$

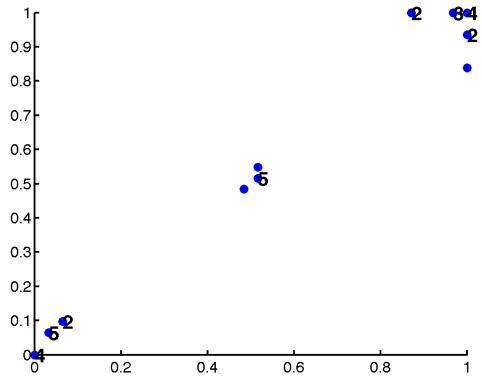
where $L(X|\theta, \xi)$ is the likelihood function.
For Joint Likelihood model FIM can be calculated analytically:

$$\boxed{\mathcal{F}_{ij} = \sum_{m=1}^{M} F_{ij}^{s} + F_{ij}^{N},} \tag{1}$$

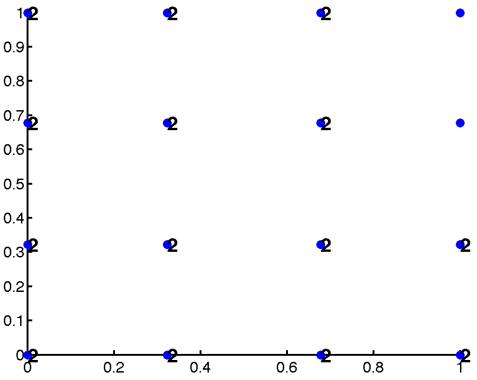
where

- M the number of design points.
- $F_{ij}^{s} = \frac{n_i - 1}{2} \frac{\partial f}{\partial \theta_i} \frac{\partial f}{\partial \theta_j}$ where n_i the number of replicate observations at design point i and $\frac{\partial f}{\partial \theta_j}$ the derivative of the variance model $f(\theta)$ with respect to parameter θ_j.
- $F_{ij}^{N} = \frac{1}{2} \mathrm{tr}(\Sigma^{-1} \frac{\partial \Sigma}{\partial \theta_i} \Sigma^{-1} \frac{\partial \Sigma}{\partial \theta_j})$.

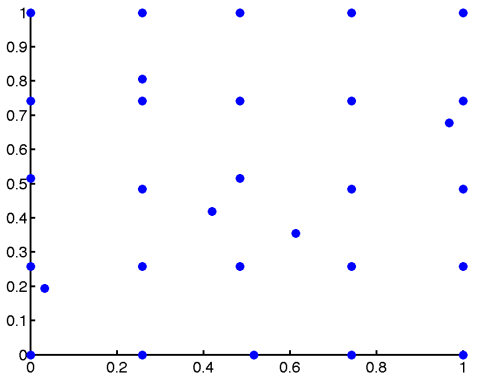
Optimal Design Example 1/2



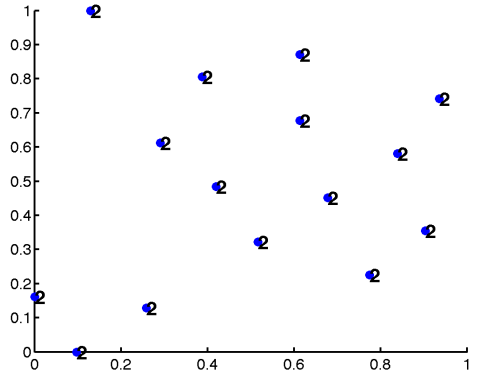
(a) Greedy



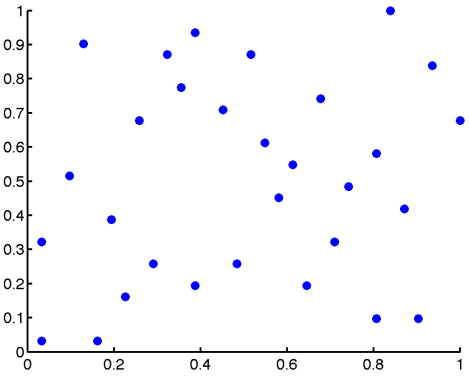
(b) Replicate Grid



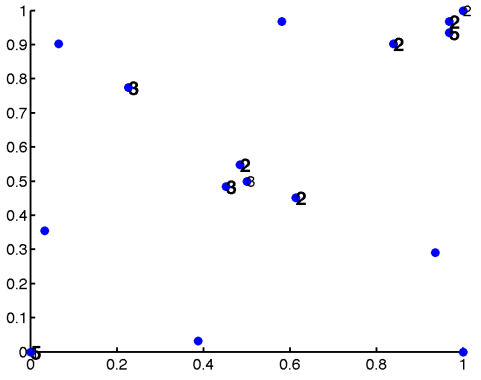
(c) Grid



(d) Latin Hypercube Rep



(e) Latin Hypercube



(f) Sim Annealing

Optimal Design Example 2/2

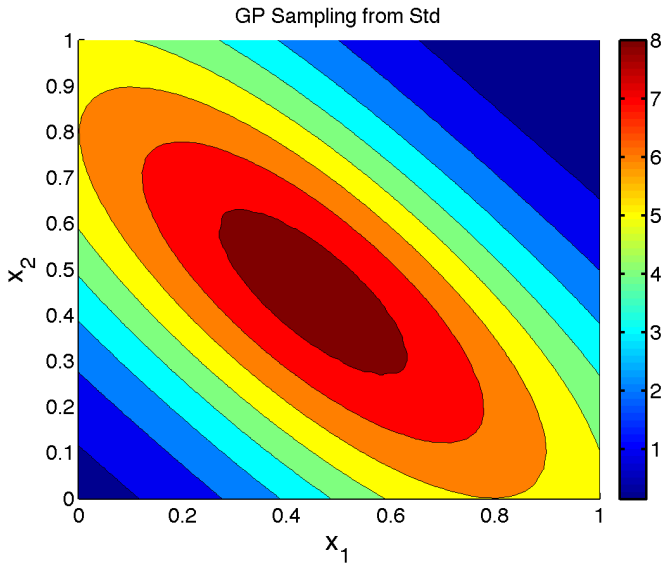


Figure: Variance surface.

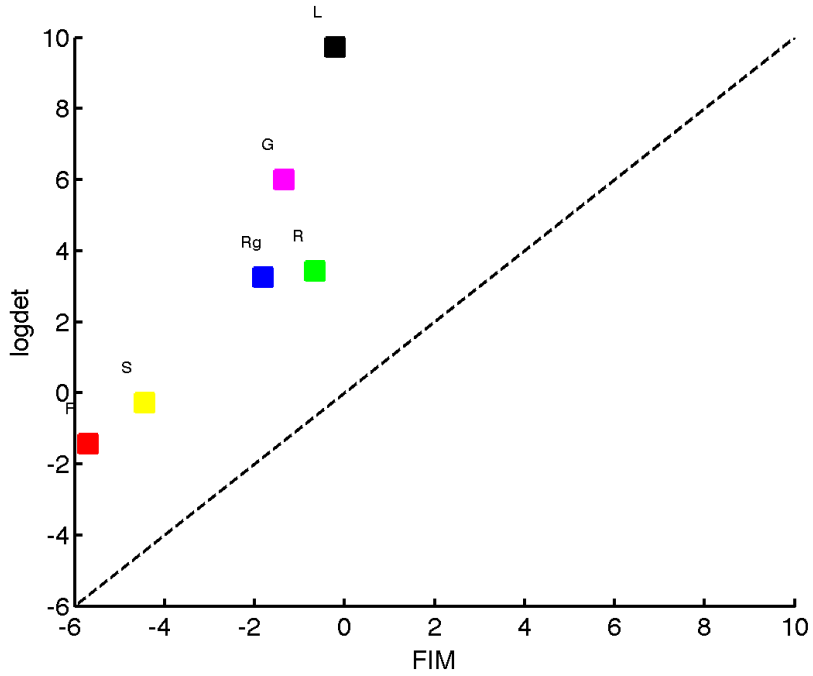


Figure: FIM (x axis) and LDM (y axis).

Table: Variance model parameter errors

Greedy	Replicate Grid	Grid	Latin Rep	Latin	Sim Ann
0.22	0.46	0.66	0.49	0.82	0.25

Design Conclusions

Benefits

- Fisher Designs minimise kernel parameter estimation variance.
- Utilising Replicated observations beneficial.

Challenges

- Difficult optimisation problems $\rightarrow$ Computationally expensive.
- Bayesian design criterion require numerical (Monte Carlo) or approximate integration (e.g. quadrature).

Screening: Automatic Relevance Determination (ARD)

- The correlation length scales δ_i in a covariance function can be used to determine the input relevance.
- ARD is typically applied using a zero mean GP emulator
- The inputs must standardised, i.e. are on the same scale.
- With a linear mean, correlation length scales indicate non-linear and interaction effects.
- A range of covariance functions can be used. For example the Rational Quadratic :

$$v(x_p, x_q) = \sigma^2[1 + (x_p - x_q)^\tau P^{-1}(x_p - x_q)/(2\alpha)]^{-\alpha},$$

where σ is the scale parameter and $P = \mathrm{diag}(\delta_i)^2$ a diagonal matrix of correlation length scale parameters.

- Maximum Likelihood inference for length-scales used.

ARD example on synthetic data

$f(x_1, x_2) = \sin(x_1/10) + 0 \times x_2$, i.e. a two variable function which ignores the second input altogether.

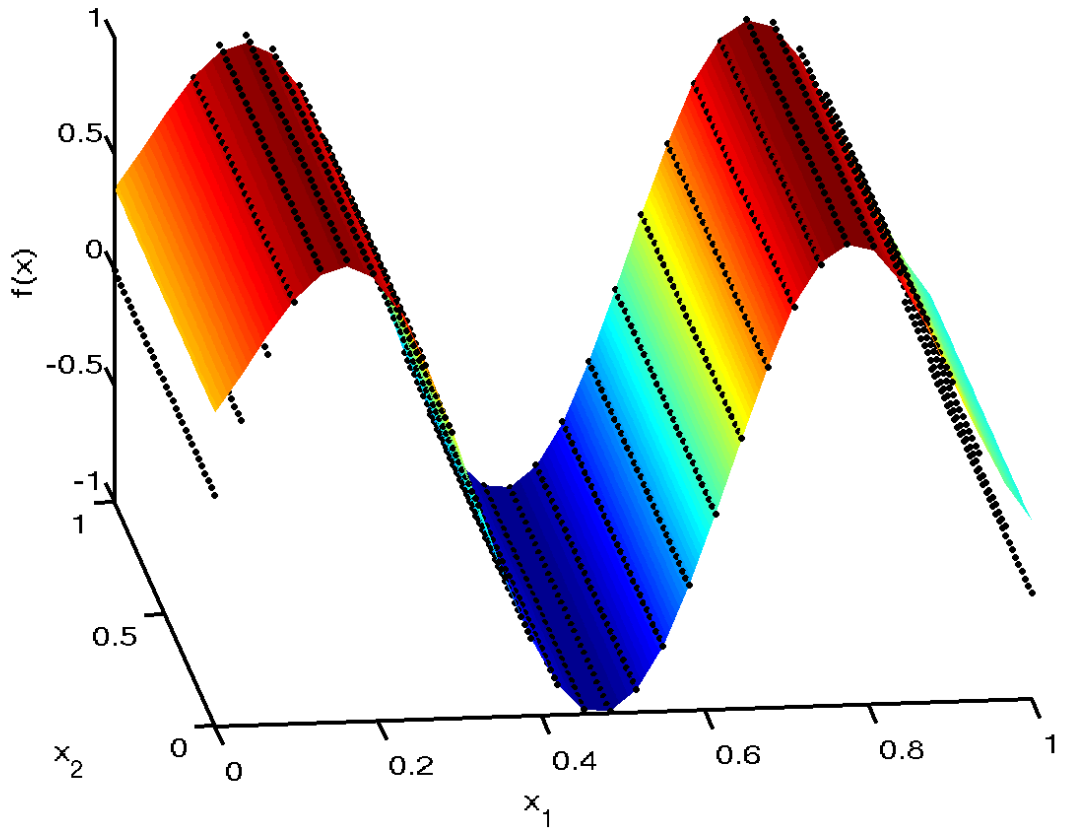


Figure: Validation of ARD Emulator. The simulator values are plotted in black dots and the emulation mean prediction is the smooth coloured surface.

Restricted Maximum Likelihood

- When specifying a GP $N(m(x)\beta, \sigma C(x, x'))$.
- we can specify a proper or improper prior on β and integrate out that uncertainty.
- In the case of improper prior, this is equivalent to the Restricted Maximum Likelihood method.
- The result is a Gaussian Process with inflated variance.

Read more:
`http://leg.ufpr.br/geoR/geoRdoc/bayeskrige.pdf` Section 4.
The Advanced Theory of Statistics, Vol. 2B: Bayesian Inference by A. O'Hagan, J. Forster.

Student-t process

- When specifying a GP $N(m(x)\beta, \sigma C(x, x'))$.
- we can specify a proper or improper prior on σ and integrate out that uncertainty.
- The result is a Student-t process.
- This does not exclude integrating out the β we saw in the previous slide.

Read more:
`http://leg.ufpr.br/geoR/geoRdoc/bayeskrige.pdf` Section 4.

Multivariate Output

- When considering multiple outputs, to ensure kernel is positive definite not easy.
- Separable kernel using kronecker product.
- Linear Model of Coregionalisation.

Read more: `http://mucm.aston.ac.uk/MUCM/MUCMToolkit/index.php?page=ThreadVariantMultipleOutputs.html`.

Part IV

Dealing with large data sets:
the sparse, sequential GP framework.

Problems applying GPs to large data sets

- GPs are a very flexible and rich model class, but computationally scale as $O(n^3)$, due to the matrix inversion.
- This might not seem too big a problem since for prediction the inverse need only be computed once (and we can use a variety of linear algebra tricks).
- However to estimate hyper-parameters in the covariance function will require many inversions.
- In really big data sets just storing the covariance matrix becomes problematic.
- In this section I will describe a method to address these problems, developed by Manfred Opper and Lehel Csato at the NCRG.

A parametrisation for a GP

We can parametrise the posterior GP given some observations X as:

$$f(\mathbf{x}) \sim N\left[\overset{1\times n}{\alpha^T}\overset{n\times 1}{k(X,\mathbf{x})}\,,\quad \overset{1\times 1}{k(\mathbf{x},\mathbf{x}')} + \overset{1\times n}{k(\mathbf{x},X)}\overset{n\times n}{C}\overset{n\times 1}{k(X,\mathbf{x}')}\right]$$

The α_i's and C_{ij}'s do not depend on the new points $\mathbf{x}$, $\mathbf{x}'$ – this is the essence of the representer theorem.

- This moment based parametrisation, as written, is exact; it has not yet bought us anything!
- However we can select the active set, X, to be an arbitrary set of points (but it is typically chosen to be a subset of the observations).

Inference in a sequential framework

The posterior GP given some observations X is:

$$p(f|X,\mathbf{y}) = \frac{p(\mathbf{y}|X,f)p(f|X)}{p(\mathbf{y}|X)},$$

Now let us imagine we can build up a sequential approximation to $p(f|X_t,\mathbf{y}_t)$, the posterior after seeing the first t observations.

- This could be very handy, since if we include another observation, then computing $p(f|X_{t+1},\mathbf{y}_{t+1})$ by updating $p(f|X_t,\mathbf{y}_t)$, will require only a low dimensional integral.
- In the GP prior and Gaussian likelihood case, this turns out to be equivalent to a well know method for iteratively computing inverses of matrices.

The Kullback-Leibler divergence

The KL divergence is a distance measure between pdf's:

$$\mathrm{KL}[p(\theta)\|q(\theta)] = \int \ln\left(\frac{p(\theta)}{q(\theta)}\right) p(\theta) d\theta \ .$$

The KL distance is very widely used in variational treatments of machine learning problems.

- Note the order of $p(\theta)$ and $q(\theta)$ is important.
- The approach is to minimise:

$$\mathrm{KL}[p(f|X_t, \mathbf{y}_t)\|\hat{p}(f|X_t, \mathbf{y}_t)] = \int p(f|X_t, \mathbf{y}_t) \ln\left(p(f|X_t, \mathbf{y}_t)\right) df - \int p(f|X_t, \mathbf{y}_t) \ln\left(\hat{p}(f|X_t, \mathbf{y}_t)\right) df \ ,$$

with respect to the parameters in the parametrisation shown previously, sequentially.

Sequential inference in GPs

- Assume the likelihood factorises, so that $p(\mathbf{y}|X, f) = \prod_i p(y_i|X_i, f)$.
- We now update the posterior sequentially:

$$p(f|X_t, \mathbf{y}_t) = \frac{p(y_t|\mathbf{x}_t, f)p(f|X_{t-1}, \mathbf{y}_{t-1})}{p(\mathbf{y}_t|X_t)} \ .$$

- $p(f|X_t, \mathbf{y}_t)$ is generally no longer a GP, so the best approximating GP, $\hat{p}(f|X_t, \mathbf{y}_t)$, is found minimising $\mathrm{KL}[p(f|X_k, \mathbf{y}_k)\|\hat{p}(f|X_k, \mathbf{y}_k)]$.
- The maths (and notation) gets a little tricky here.
- At each step we include a new observation, which updates our GP posterior, computed as the projection onto the optimal GP posterior using the KL divergence.

Sequential inference in GPs

- We can write the updates for the parameters $\boldsymbol{\alpha}_t$ and C_t.

$$
\begin{aligned}
\overset{t\times 1}{\mathbf{s}_t} &= \overset{t-1\times t-1}{C_{t-1}}\ \overset{t-1\times 1}{k(X_{t-1}, \mathbf{x}_t)} + \overset{t\times 1}{\mathbf{e}_t}\ , \\
\overset{t\times 1}{\boldsymbol{\alpha}_t} &= \overset{t-1\times 1}{\boldsymbol{\alpha}_{t-1}} + \overset{1\times 1}{q_t}\ \overset{t\times 1}{\mathbf{s}_t}\ , \\
\overset{t\times t}{C_t} &= \overset{t-1\times t-1}{C_{t-1}} + \overset{1\times 1}{r_t}\ \overset{t\times 1}{\mathbf{s}_t}\ \overset{1\times t}{\mathbf{s}_t^T}\ .
\end{aligned}
$$

- q_t and r_t are defined as the first and second derivatives of the logarithm of the average likelihood: $\int p(y_t|\mathbf{x}_t, f)p(f|X_{t-1}, \mathbf{y}_{t-1})df$ wrt the expectation of the marginal posterior at $t-1$ at the new point $\mathbf{x}_t$.
- Note the integral is only at the new observation, and averaging over the prior process at $t-1$ means that we can deal with rather nasty likelihoods, such as step functions.

Sequential inference in GPs

- This method still scales as $O(n^3)$.
- However it does provide a bound on the evidence for hyper-parameter optimisation.
- In practice the scalars q_t and r_t can be computed analytically for many likelihoods, but if not the 1D integrals required can be performed using a range of numerical methods.
- Next we seek an $O(nm^2)$ scaling by retaining m points in an active set which is a sparse parametrisation of the posterior process.

Sparse inference in GPs

- To address the growth in complexity of the algorithm several approaches have been suggested, including:
  - choosing a subset of the observations – e.g. Informative Vector Machine;
  - reduced rank approximations of $\tilde{K}(\cdot,\cdot) = K_{nm}K_{mm}^{-1}K_{mn}$ – e.g. Nystrom methods;
  - partitioning the input space – e.g. Bayesian Committee Machine;
  - projecting the GP to a simpler representation – e.g. sparse, sequential GP (ssGP) method.
- The essence of the ssGP method is to project the GP at each time an observation is included, to a best GP without increasing the size of the active set.
- Best is defined wrt to the KL divergence again (the other way around!) and the effect of the observation is included in updates to the parameters of the active set.

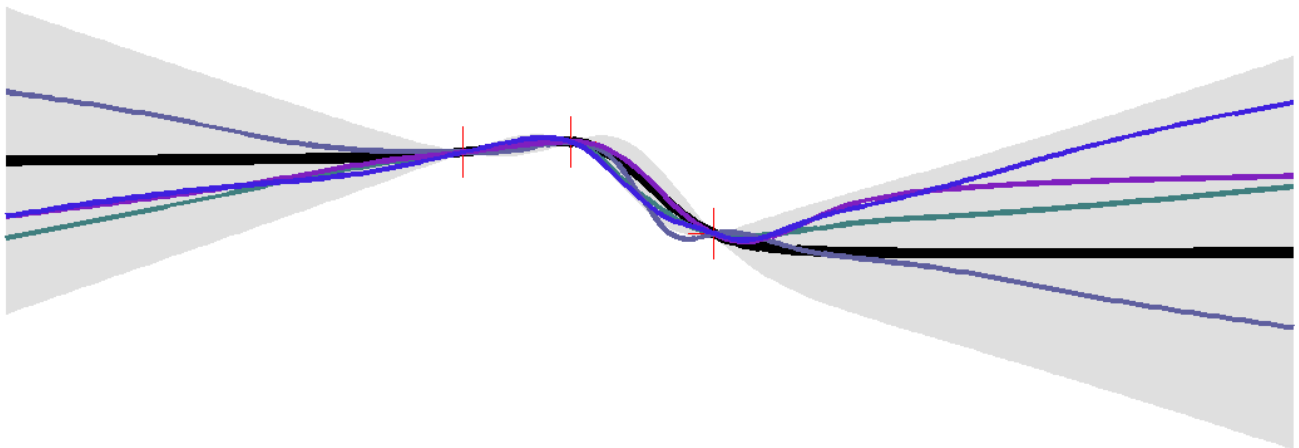
Sparse, sequential GPs

- As currently explained the ssGP algorithm has a weakness; the algorithm permits only one pass through the observations.
- For non-Gaussian likelihoods this is likely to be highly suboptimal.
- The solution is the Expectation Propagation (EP) within the ssGP framework.
- EP stores an effective likelihood for each observation added.
- The observations can then be re-used by removing the effective likelihood, then re-using the observations.
- The algorithm can be shown to converge to a fixed point, and again provides a bound on the model evidence.

GPs and ssGPs

- GP's provide a natural probabilistic model for regression problems.
- They are related to kernel methods, but impose a probabilistic model in feature space.
- Computational issues limit the applicability of GPs in large data sets.
- The ssGP framework developed by Manfred and Lehel at NCRG is a very elegant and effective solution to many of these problems.
- The framework allows us to perform GP inference on large data sets and also estimate hyper-parameters.
- There are a number of unresolved issues in the application of ssGPs: really big data sets; practical implementation / convergence; input uncertainty; fully Bayesian inference; robustness.

GP related things I haven't mentioned



- Non-stationary covariance functions (more realistic for many problems?), and mean functions.
- Covariance separability and space-time GPs.
- GPs for classification; lots of approximations!
- Hierarchical modelling and GPs.
- Relation of GPs to Gauss-Markov Random Fields.