Learning and Inference with Gaussian Processes

An Overview of Gaussian Processes with some state of the art applications

Neil Lawrence
Department of Computer Science
University of Sheffield

21st August 2006

Outline

1. Introduction to Gaussian Processes
    - Distributions over Functions
    - Samples from a Gaussian Distribution
    - Covariance functions
    - Different Covariance Functions
2. Prediction with Gaussian Processes
    - Interpolation with Gaussian Processes
    - Regression with Gaussian Processes
    - Parametric Models vs GPs
    - Learning Kernel Parameters
3. Examples
    - Transcription Factor Concentration Inference
    - Dimensional Reduction
4. Conclusions

Online Resources

All source code and slides are available online

- This talk available from my home page (see talks link on side).
- MATLAB examples in the 'oxford' toolbox (vrs 0.13).
  - `http://www.dcs.shef.ac.uk/~neil/oxford/`.
- And the 'gpsim' toolbox (vrs 0.1).
  - `http://www.dcs.shef.ac.uk/~neil/gpsim/`.
- MATLAB commands used for examples given in `typewriter font`.

Introduction to Gaussian Processes

Inference about functions

- Many Machine Learning problems can be reduced to inference about functions.
    - We will see some examples later.
- Gaussian processes (GPs) are probabilistic models for functions. [6, 7, 8]
- GPs allow inference about functions in the presence of uncertainty.

Defining a Distribution over Functions

Gaussian Process

- What is meant by a distribution over functions?
- Functions are infinite dimensional objects:
    - Defining a distribution over functions seems non-sensical.

Gaussian Distribution

- Start with a standard Gaussian distribution.
- Consider the distribution over a fixed number of instantiations of the function.

Gaussian Distribution

Zero mean Gaussian distribution

- A multi-variate Gaussian distribution is defined by a mean and a covariance matrix.

$$N\left(\mathbf{f}|\mu,\mathbf{K}\right)=\frac{1}{\left(2\pi\right)^{\frac{N}{2}}\left|\mathbf{K}\right|^{\frac{1}{2}}}\exp\left(-\frac{\left(\mathbf{f}-\mu\right)^{\mathrm{T}}\mathbf{K}^{-1}\left(\mathbf{f}-\mu\right)}{2}\right).$$

- We will consider the special case where the mean is zero,

$$N\left(\mathbf{f}|\mathbf{0},\mathbf{K}\right)=\frac{1}{\left(2\pi\right)^{\frac{N}{2}}\left|\mathbf{K}\right|^{\frac{1}{2}}}\exp\left(-\frac{\mathbf{f}^{\mathrm{T}}\mathbf{K}^{-1}\mathbf{f}}{2}\right).$$

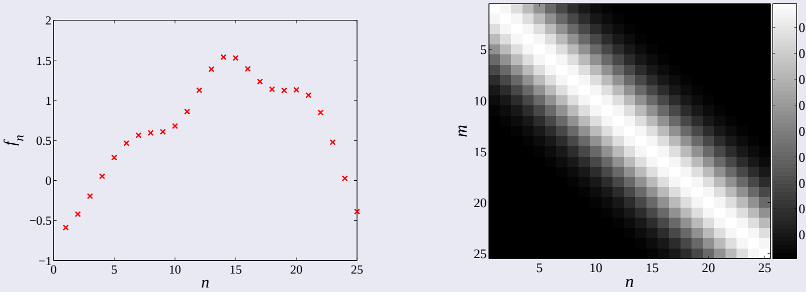
Sampling a Function

Multi-variate Gaussians

- We will consider a Gaussian with a particular structure of covariance matrix.
- Generate a single sample from this 25 dimensional Gaussian distribution, $\mathbf{f} = [f_1, f_2 \dots f_{25}]$.
- We will plot these points against their index.

Gaussian Distribution Sample

demGPSample



(a)

(b)

Figure: (a) 25 instantiations of a function, f_n, (b) greyscale covariance matrix.

Covariance Function

The covariance matrix

- Covariance matrix shows correlation between points f_m and f_n if n is near to m.
- Less correlation if n is distant from m.
- Our ordering of points means that the *function appears smooth*.
- Let's focus on the joint distribution of two points form the 25.

Covariance Function

The covariance matrix

- Covariance matrix shows correlation between points f_m and f_n if n is near to m.
- Less correlation if n is distant from m.
- Our ordering of points means that the *function appears smooth*.
- Let's focus on the joint distribution of two points form the 25.

Covariance Function

The covariance matrix

- Covariance matrix shows correlation between points f_m and f_n if n is near to m.
- Less correlation if n is distant from m.
- Our ordering of points means that the *function appears smooth*.
- Let's focus on the joint distribution of two points form the 25.

Covariance Function

The covariance matrix

- Covariance matrix shows correlation between points f_m and f_n if n is near to m.
- Less correlation if n is distant from m.
- Our ordering of points means that the *function appears smooth*.
- Let's focus on the joint distribution of two points form the 25.

Prediction of f_2 from f_1

demGPCov2D([1 2])

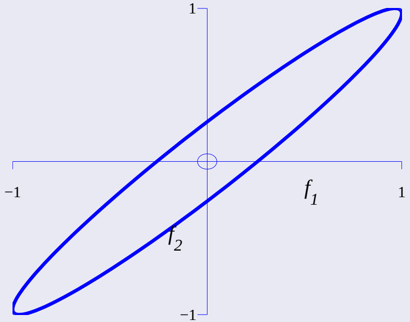


Figure: Covariance for $\begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$ is $\mathbf{K}_{12} = \begin{bmatrix} 1 & 0.966 \\ 0.966 & 1 \end{bmatrix}$.

Prediction of f_2 from f_1

`demGPCov2D([1 2])`

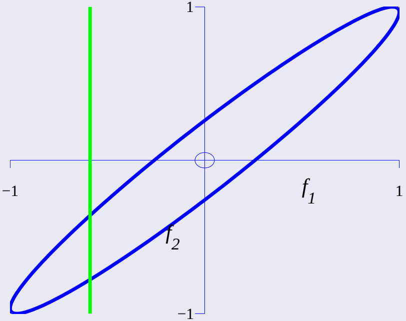


Figure: Covariance for $\begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$ is $\mathbf{K}_{12} = \begin{bmatrix} 1 & 0.966 \\ 0.966 & 1 \end{bmatrix}$.

Prediction of f_2 from f_1

`demGPCov2D([1 2])`

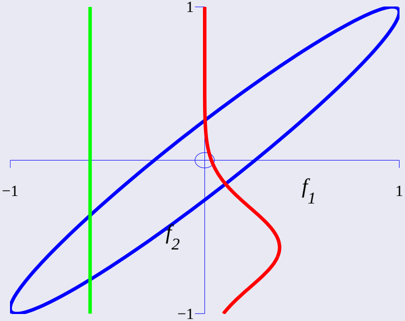


Figure: Covariance for $\begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$ is $\mathbf{K}_{12} = \begin{bmatrix} 1 & 0.966 \\ 0.966 & 1 \end{bmatrix}$.

Prediction of f_5 from f_1

`demGPCov2D([1 5])`

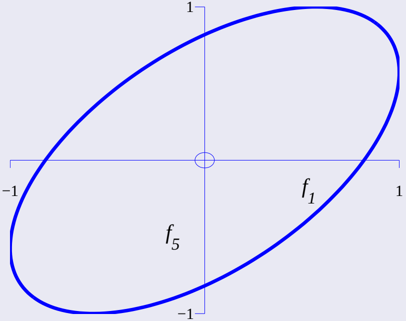


Figure: Covariance for $\begin{bmatrix} f_1 \\ f_5 \end{bmatrix}$ is $\mathbf{K}_{15} = \begin{bmatrix} 1 & 0.574 \\ 0.574 & 1 \end{bmatrix}$.

Prediction of f_5 from f_1

`demGPCov2D([1 5])`

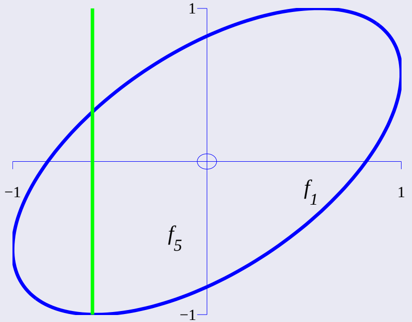


Figure: Covariance for $\begin{bmatrix} f_1 \\ f_5 \end{bmatrix}$ is $\mathbf{K}_{15} = \begin{bmatrix} 1 & 0.574 \\ 0.574 & 1 \end{bmatrix}$.

Prediction of f_5 from f_1

`demGPCov2D([1 5])`

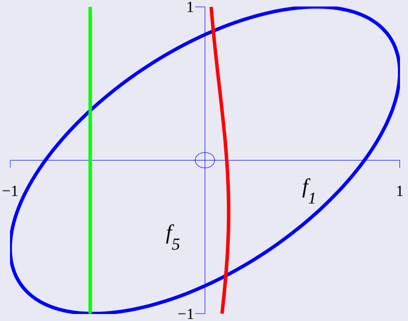


Figure: Covariance for $\begin{bmatrix} f_1 \\ f_5 \end{bmatrix}$ is $\mathbf{K}_{15} = \begin{bmatrix} 1 & 0.574 \\ 0.574 & 1 \end{bmatrix}$.

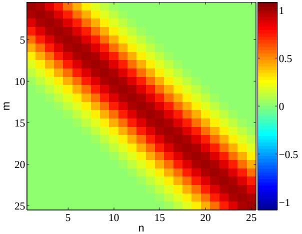
Covariance Functions

Where did this covariance matrix come from?

RBF Kernel Function

$$k\left(\mathbf{x}_m, \mathbf{x}_n\right) = \alpha \exp\left(-\frac{\left|\left|\mathbf{x}_m - \mathbf{x}_n\right|\right|^2}{2l^2}\right)$$

- Covariance matrix is built using the *inputs* to the function $\mathbf{x}_n$.
- For the example above it was based on Euclidean distance.
- The covariance function is also know as a kernel.

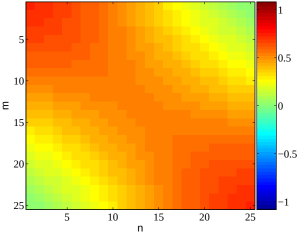


Different Covariance Functions

MLP Kernel Function

$$k\left(\mathbf{x}_m, \mathbf{x}_n\right) = \alpha \sin^{-1}\left(\frac{w\mathbf{x}_m^{\mathrm{T}}\mathbf{x}_n + b}{\sqrt{w\mathbf{x}_m^{\mathrm{T}}\mathbf{x}_m + b + 1}\sqrt{w\mathbf{x}_n^{\mathrm{T}}\mathbf{x}_n + b + 1}}\right)$$

- A non-stationary covariance matrix [10].
- Derived from a multi-layer perceptron (MLP).

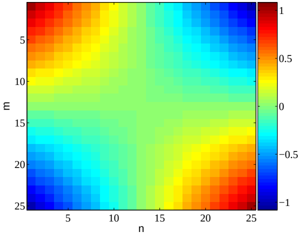


Different Covariance Functions

Linear Kernel Function

$$k\left(\mathbf{x}_m, \mathbf{x}_n\right) = \alpha \mathbf{x}_m^{\mathrm{T}} \mathbf{x}_n$$

- Allows for a linear trend.
- Derived from a neural network.
- Note the anti-correlations in the matrix.

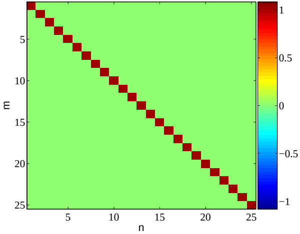


Different Covariance Functions

White noise

$$k(\mathbf{x}_m, \mathbf{x}_n) = \alpha\delta_{mn}$$

- Where δ_{mn} is the Kronecker delta.
- Simply represents uncorrelated independent noise.



Covariance Samples

`demCovFuncSample`

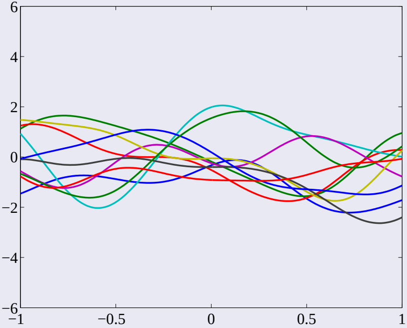


Figure: RBF kernel with $\gamma = 10$, $\alpha = 1$

Covariance Samples

`demCovFuncSample`

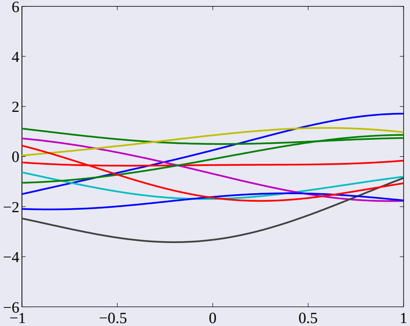


Figure: RBF kernel with $l = 1$, $\alpha = 1$

Covariance Samples

`demCovFuncSample`

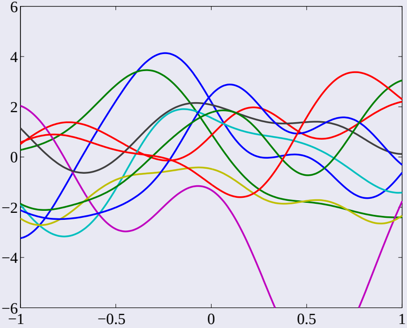


Figure: RBF kernel with $l = 0.3$, $\alpha = 4$

Covariance Samples

`demCovFuncSample`

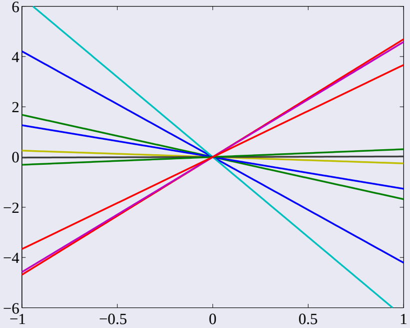


Figure: linear kernel with $\alpha = 16$

Covariance Samples

`demCovFuncSample`

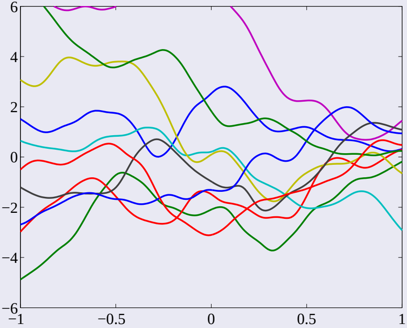


Figure: MLP kernel with $\alpha = 8$, $w = 100$ and $b = 100$

Covariance Samples

`demCovFuncSample`

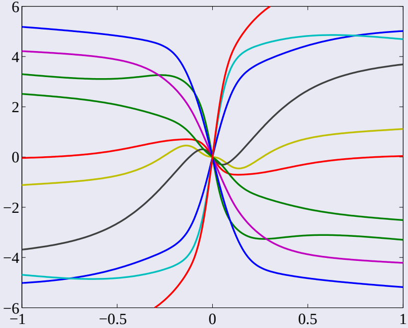


Figure: MLP kernel with $\alpha = 8$, $b = 0$ and $w = 100$

Covariance Samples

`demCovFuncSample`

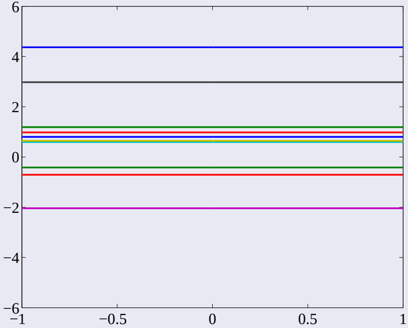


Figure: bias kernel with $\alpha = 1$ and

Covariance Samples

`demCovFuncSample`

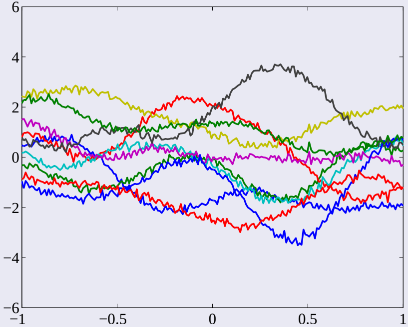


Figure: summed combination of: RBF kernel, $\alpha = 1$, $l = 0.3$; bias kernel, α =1; and white noise kernel, $\beta = 100$

Joint Distribution

Making Predictions

- Covariance function provides the joint distribution over the instantiations.
- Conditional distribution provides predictions.
- Denoting the training set as $\mathbf{f}$ and test set as $\mathbf{f}_*$.
  - Predict using $p\left(\mathbf{f}_*|\mathbf{f}\right)$.
  - This conditional distribution is also Gaussian.

Gaussian Process Interpolation

`demInterpolation`

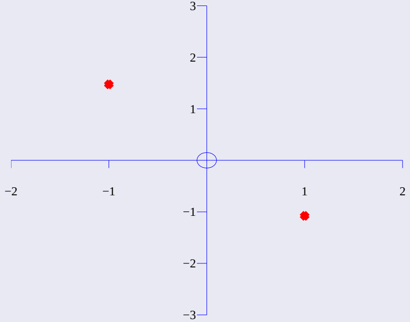


Figure: Real example: BACCO (see *e.g.* [5]). Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

Gaussian Process Interpolation

`demInterpolation`

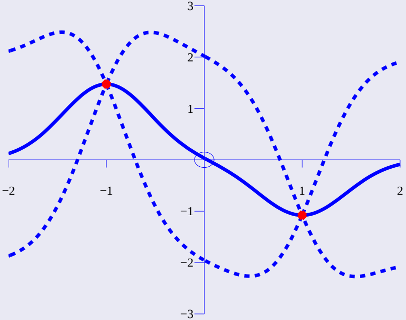


Figure: Real example: BACCO (see *e.g.* [5]). Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

Gaussian Process Interpolation

`demInterpolation`

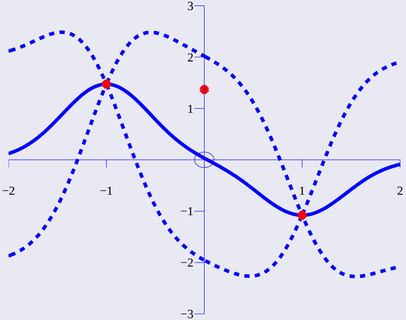


Figure: Real example: BACCO (see *e.g.* [5]). Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

Gaussian Process Interpolation

`demInterpolation`

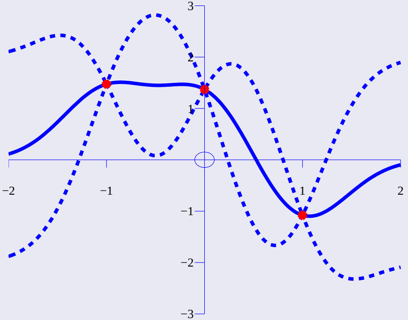


Figure: Real example: BACCO (see *e.g.* [5]). Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

Gaussian Process Interpolation

`demInterpolation`

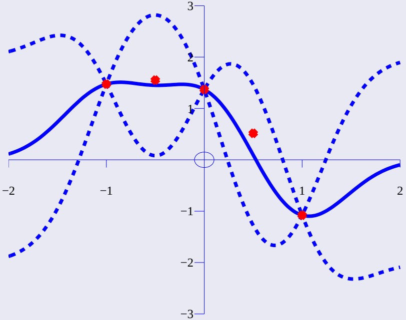


Figure: Real example: BACCO (see *e.g.* [5]). Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

Gaussian Process Interpolation

`demInterpolation`

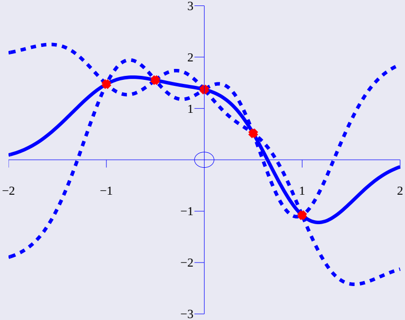


Figure: Real example: BACCO (see *e.g.* [5]). Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

Gaussian Process Interpolation

`demInterpolation`

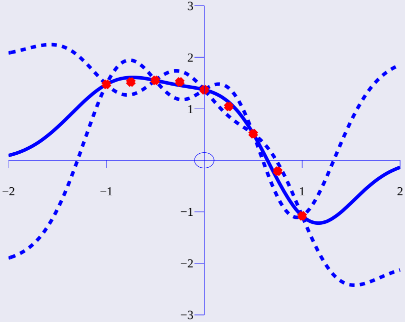


Figure: Real example: BACCO (see *e.g.* [5]). Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

Gaussian Process Interpolation

`demInterpolation`

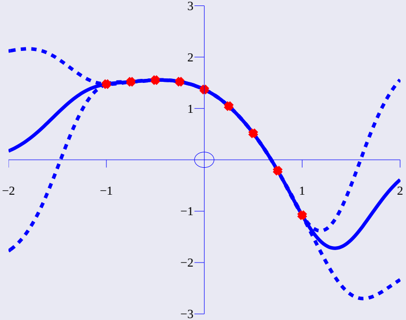


Figure: Real example: BACCO (see *e.g.* [5]). Interpolation through outputs from slow computer simulations (*e.g.* atmospheric carbon levels).

Noise Models

Graph of a GP

- Relates input variables, $\mathbf{X}$, to vector, $\mathbf{y}$, through $\mathbf{f}$ given kernel parameters θ.
- Plate notation indicates independence of $y_n|f_n$.
- Noise model, $p\left(y_n|f_n\right)$ can take several forms.
- Simplest is Gaussian noise.

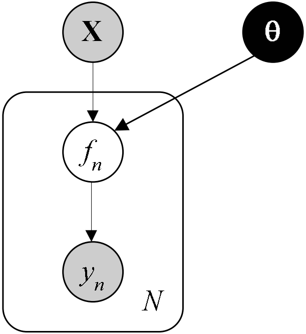


Figure: The Gaussian process depicted graphically.

Gaussian Process Regression

`demRegression`

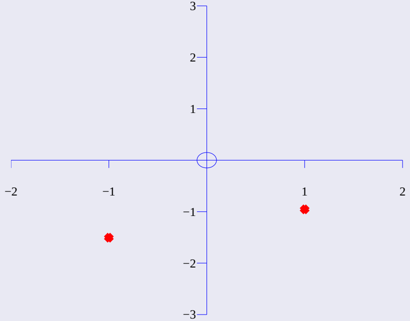


Figure: Examples include WiFi localization, C14 callibration curve.

Gaussian Process Regression

demRegression

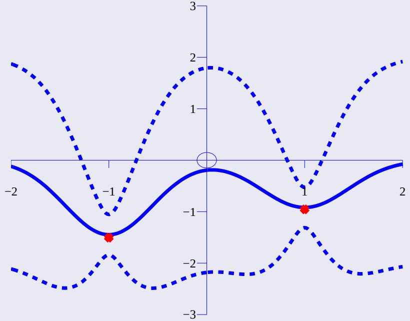


Figure: Examples include WiFi localization, C14 callibration curve.

Gaussian Process Regression

`demRegression`

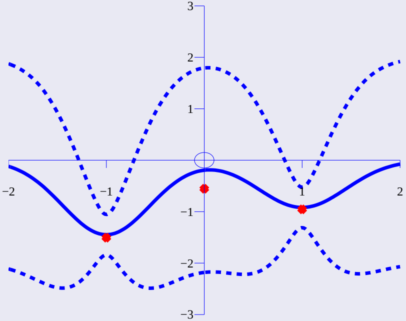


Figure: Examples include WiFi localization, C14 callibration curve.

Gaussian Process Regression

demRegression

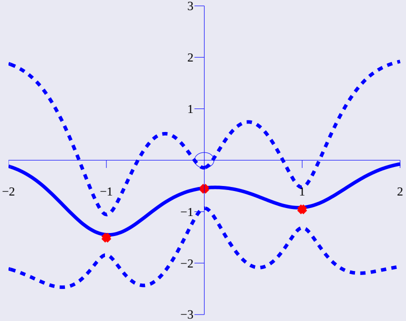


Figure: Examples include WiFi localization, C14 callibration curve.

Gaussian Process Regression

`demRegression`

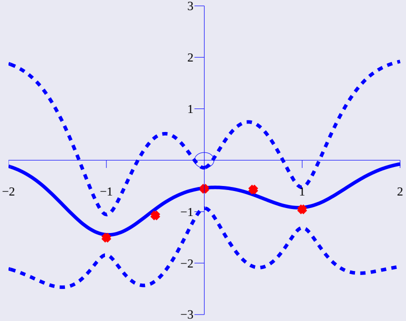


Figure: Examples include WiFi localization, C14 callibration curve.

Gaussian Process Regression

demRegression

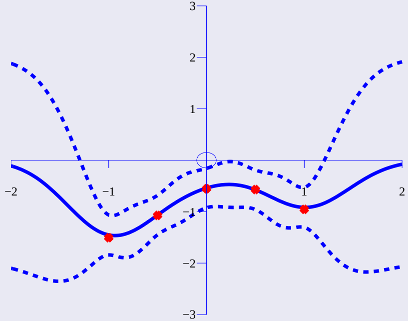


Figure: Examples include WiFi localization, C14 callibration curve.

Gaussian Process Regression

demRegression

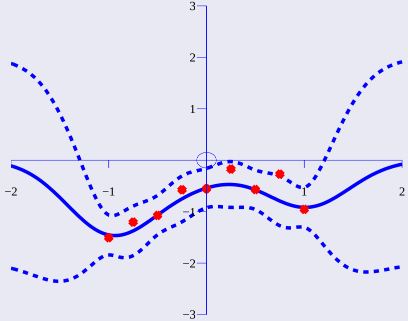


Figure: Examples include WiFi localization, C14 callibration curve.

Gaussian Process Regression

demRegression

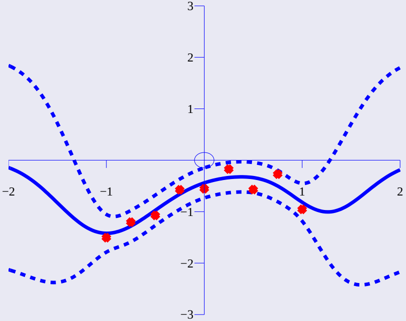


Figure: Examples include WiFi localization, C14 callibration curve.

A Paradigm Shift from i.i.d.

Parametric Model

$$p\left(y_n|\mathbf{x}_n, \mathbf{w}\right) = N\left(y_n|\mathbf{w}^{\mathrm{T}}\mathbf{x}_n, \sigma^2\right)$$

$$p\left(\mathbf{y}|\mathbf{X}, \mathbf{w}\right) = \prod_{n=1}^{N} p\left(y_n|\mathbf{x}_n, \mathbf{w}\right)$$

Parameteric models normally assume independence given parameters.

A Paradigm Shift from i.i.d.

Gaussian process

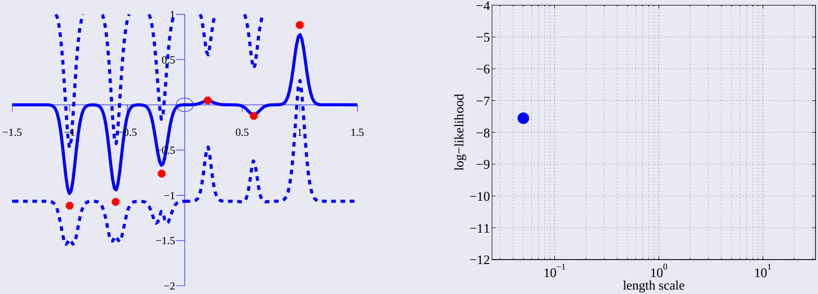
$$p\left(\mathbf{y}|\mathbf{X}\right) = N\left(\mathbf{y}|\mathbf{0},\mathbf{K}\right)$$

In GPs no i.i.d. assumption is made the kernel expresses correlations.

Learning Kernel Parameters

Can we determine length scales and noise levels from the data?

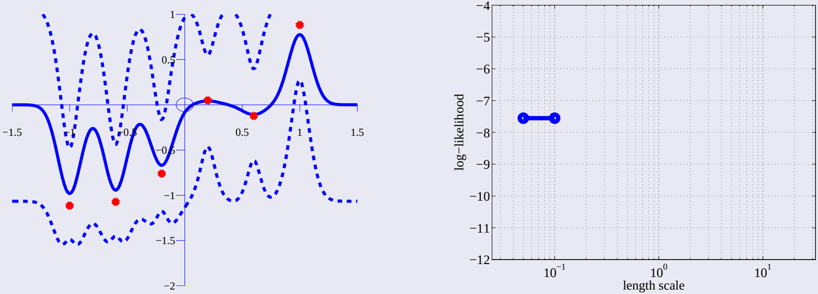
`demOptimiseKern`



Learning Kernel Parameters

Can we determine length scales and noise levels from the data?

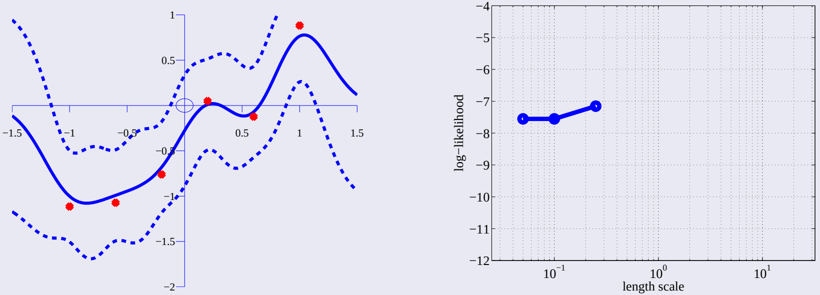
`demOptimiseKern`



Learning Kernel Parameters

Can we determine length scales and noise levels from the data?

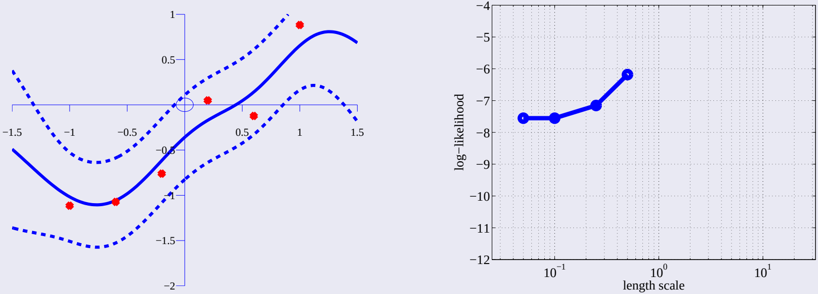
demOptimiseKern



Learning Kernel Parameters

Can we determine length scales and noise levels from the data?

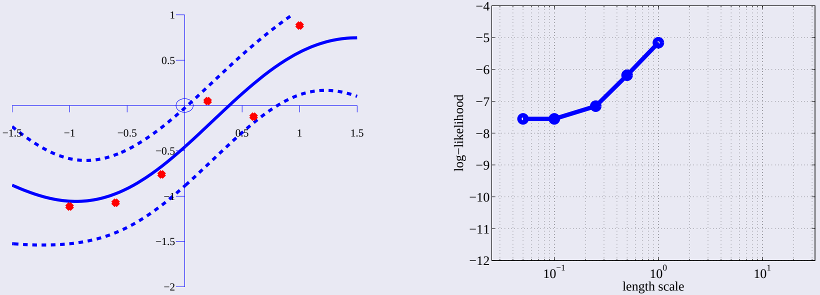
`demOptimiseKern`



Learning Kernel Parameters

Can we determine length scales and noise levels from the data?

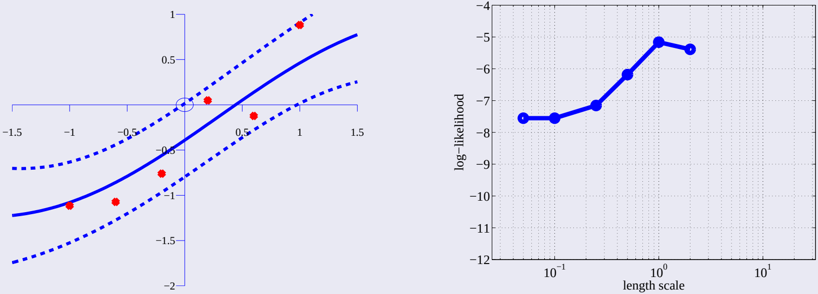
`demOptimiseKern`



Learning Kernel Parameters

Can we determine length scales and noise levels from the data?

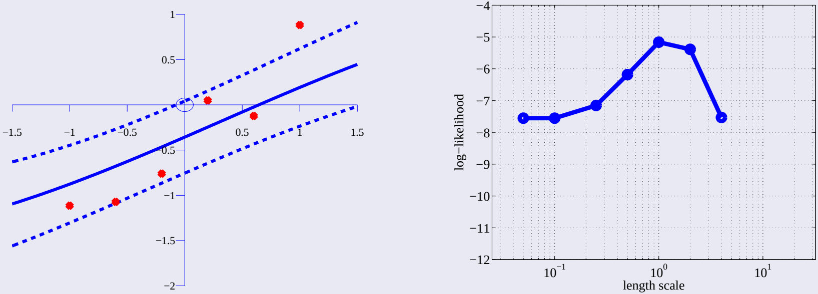
demOptimiseKern



Learning Kernel Parameters

Can we determine length scales and noise levels from the data?

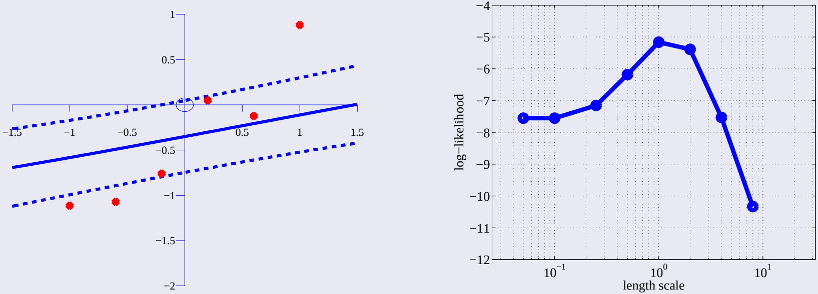
`demOptimiseKern`



Learning Kernel Parameters

Can we determine length scales and noise levels from the data?

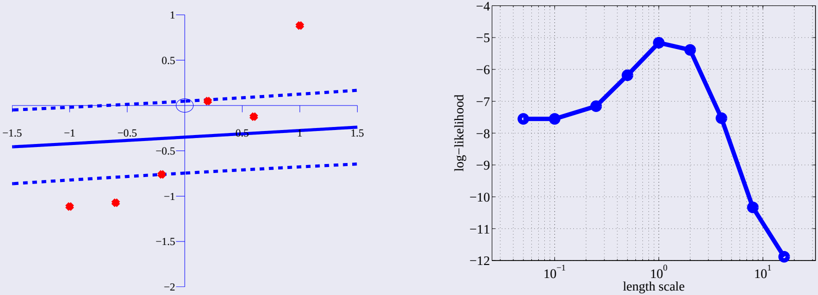
`demOptimiseKern`



Learning Kernel Parameters

Can we determine length scales and noise levels from the data?

`demOptimiseKern`



Biological Problem

Inference of p53 Concentration

- Gene expression levels are controlled by *transcription factors*.
- Transcription factor concentration is difficult to measure
- Gene expression can be measured with microarray technology.

Differential Equation model

- Simple linear model differential equation model recently used by Barenco *et al* [1].
- They inferred transcription factor concentrations using Markov Chain Monte Carlo (10^7 iterations).
- We repeat their experiments with Gaussian processes.

Simple Linear Model

Linear model of regulation

$$\frac{dy_i\left(t\right)}{dt} = B_i + S_i f\left(t\right) - D_i y_i\left(t\right)$$

where:

$y_i\left(t\right)$ — expression of the ith gene at time t.
$f\left(t\right)$ — concentration of the transcription factor at time t.
D_i — gene's decay rate.
B_i — basal transcription rate.
S_i — sensitivity to the transcription factor.

Equation Solution

Solve via Laplace Transforms

- Solution to the equation:

$$y_i\left(t\right) = \frac{B_i}{D_i} + S_i \exp\left(-D_i t\right) \int_0^t f\left(u\right) \exp\left(D_i u\right) du.$$

If $f\left(t\right)$ is a zero mean Gaussian process then $y_i\left(t\right)$ is also a Gaussian process with mean $\frac{B_i}{D_i}$.

Two Properties of GPs

Integral of Gaussian Process

The integral of a GP is also a GP,

$$f(t) \sim N(\mathbf{0}, \mathbf{K}_{ff})$$

and

$$g(t) = \int_0^t f(u)\,du$$

then

$$g(t) \sim N(\mathbf{0}, \mathbf{K}_{gg}),$$

where

$$k_{gg}(t, t') = \int_0^t \int_0^{t'} k_{ff}(u, u')\,du du'$$

Two Properties of GPs

Product with deterministic function

The integral of a GP is also a GP,

$$f\left(t\right) \sim N\left(\mathbf{0}, \mathbf{K}_{ff}\right),$$

and

$$g\left(t\right) = f\left(t\right) h\left(t\right)$$

where $h\left(t\right)$ is a deterministic function then,

$$g\left(t\right) \sim N\left(\mathbf{0}, \mathbf{K}_{gg}\right),$$

where

$$k_{gg}\left(t, t'\right) = h\left(t\right) k_{ff}\left(t, t'\right) h\left(t'\right)$$

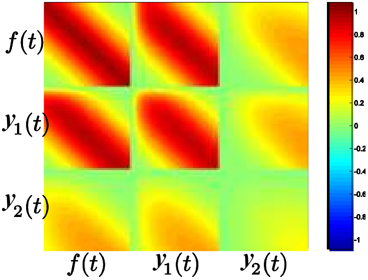
Covariance for Transcription Model

RBF Kernel function for $f(t)$

$$y_i(t) = \frac{B_i}{D_i} + S_i \exp(-D_i t) \int_0^t f(u) \exp(D_i u)\, du.$$

- Joint distribution for $y_1(t)$, $y_2(t)$ and $f(t)$.
- Here:

D_1	S_1	D_2	S_2
5	5	0.5	0.5



Joint Sampling of $y(t)$ and $f(t)$ from Covariance

gpsimTest

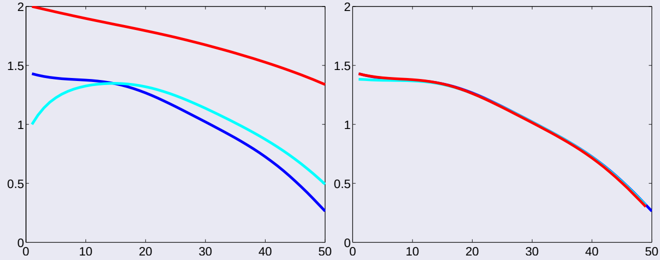


Figure: *Left*: joint samples from the transcription covariance, *blue*: $f(t)$, *cyan*: $y_1(t)$ and *red*: $y_2(t)$. *Right*: numerical solution for $f(t)$ of the differential equation from $y_1(t)$ and $y_2(t)$ (blue and cyan). True $f(t)$ included for comparison.

Joint Sampling of $y(t)$ and $f(t)$ from Covariance

gpsimTest

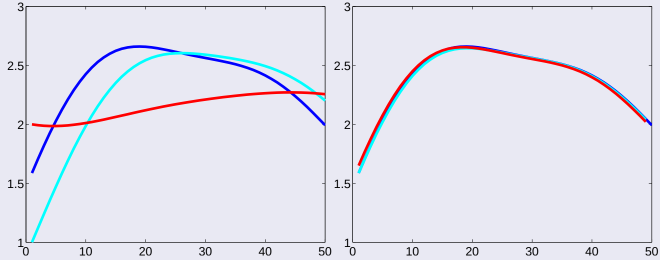


Figure: *Left*: joint samples from the transcription covariance, *blue*: $f(t)$, *cyan*: $y_1(t)$ and *red*: $y_2(t)$. *Right*: numerical solution for $f(t)$ of the differential equation from $y_1(t)$ and $y_2(t)$ (blue and cyan). True $f(t)$ included for comparison.

Joint Sampling of $y(t)$ and $f(t)$ from Covariance

gpsimTest

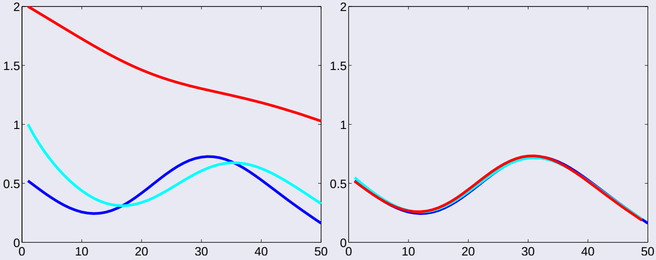


Figure: *Left*: joint samples from the transcription covariance, *blue*: $f(t)$, *cyan*: $y_1(t)$ and *red*: $y_2(t)$. *Right*: numerical solution for $f(t)$ of the differential equation from $y_1(t)$ and $y_2(t)$ (blue and cyan). True $f(t)$ included for comparison.

Results — Transcription Rates

Estimation of Equation Parameters `demBarenco1`

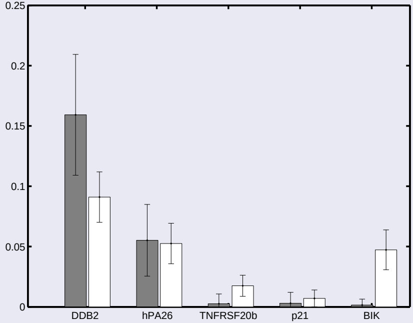


Figure: Basal transcription rates. Our results (black) compared with [1] (white).

Results — Transcription Rates

Estimation of Equation Parameters `demBarenco1`

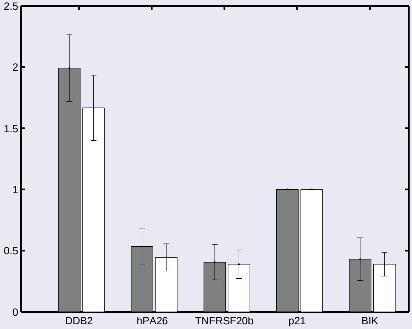


Figure: Sensitivities. Our results (black) compared with [1] (white).

Results — Transcription Rates

Estimation of Equation Parameters `demBarenco1`

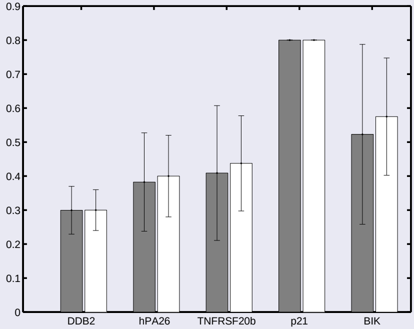
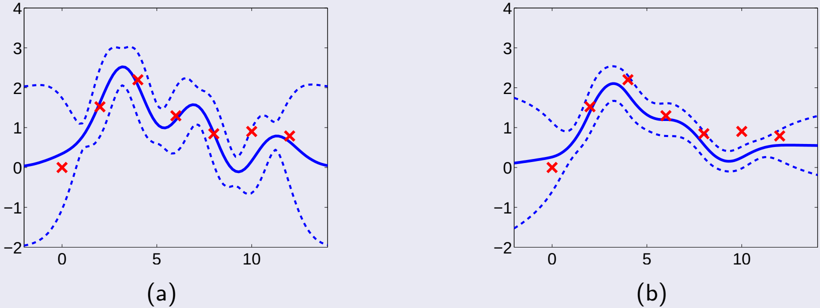


Figure: Decays. Our results (black) compared with [1] (white).

Results — Protein Concentration

Prediction with error bars of protein concentration: $p\left(\mathbf{f}\middle|\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \mathbf{y}_4, \mathbf{y}_5\right)$



(a)

(b)

Figure: (a) RBF covariance function (b) MLP covariance function. Also included are results from [1] as crosses.

Results — Positive Constrained

GP predictions in log space.

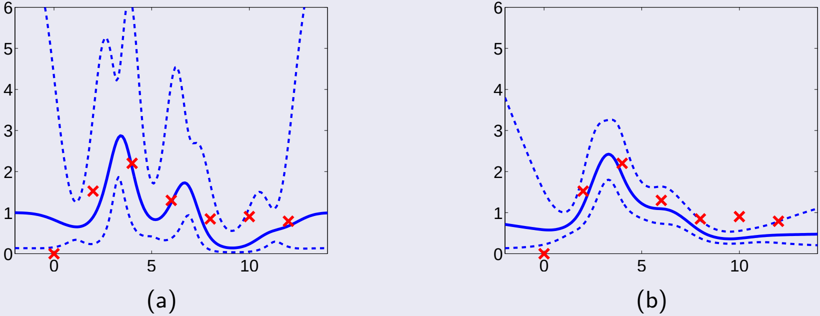


Figure: (a) RBF covariance function (b) MLP covariance function. Also included are results from Barenco *et al* [1] as crosses.

Transcription Model Summary

Progress so far and Future work

- Elegant solution of a problem with indirect observations.
- Already extended to non-linear response equations (using Laplace approximation).
- Expect to extend it to systems with *multiple transcription factors*.
- Gives results in 13 minutes vs 10^7 Monte-Carlo iterations.

Dimensional Reduction

Low Dimensional Manifolds for High Dimensional Data

- Recently proposed approach to probababilistic modelling.
- Involves mapping from low dimensional *latent* space to high dimensional data space.
- Mappings are formed from Gaussian processes.
- Several important applications including tracking [9] and graphics [3].
- Approach is a probabilistic non-linear generalisation of PCA [4].

A Latent Variable Model

How can a model designed primarily for regression be used as a technique for dimensional reduction?

Graph of GP-LVM

- **Now optimise over $\mathbf{X}$ as well as θ.**
- Relates input variables, $\mathbf{X}$, to vector, $\mathbf{y}$, through $\mathbf{f}$ given kernel parameters θ.
- Plate notation indicates independence of $\mathbf{y}_n|\mathbf{f}_n$.

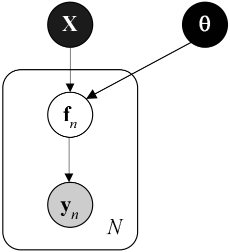


Figure: The GP-LVM depicted graphically.

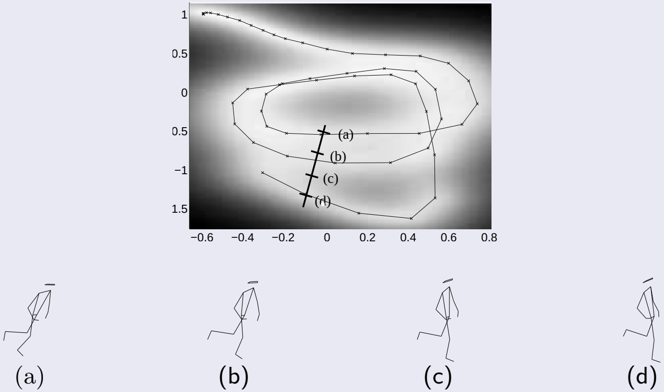
Probabilistic Model in High Dimensions

Generalization with less Data than Dimensions

- Powerful uncertainly handling of GPs leads to suprising properties.
- Non-linear models can be used where there are fewer data points than dimensions *without overfitting*.
- Example: Modelling a stick man in 102 dimensions with 55 data points!

Stick Man Results

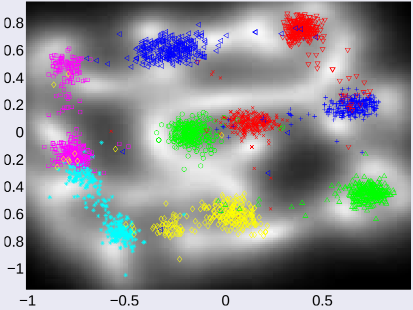
demStickResults



Projection into data space from four points in the latent space. The inclination of the runner changes becoming more upright.

Vowel Data

Vocal Joystick System [2] (`demVowels3` in fgplvm toolbox)



The different vowels are shown as follows: */a/* cross */ae/* circle */ao/* plus */e/* asterix */i/* square */ibar/* diamond */o/* down triangle */schwa/* up triangle and */u/* left triangle.

Summary

- Gaussian Processes are a powerful flexible way to make inference about functions.
- Can be combined with differential equations:
    - Facilitates parameter learning, *much* quicker than Monte Carlo approaches.
    - Expected to be vital for larger systems (*e.g.* several transcription factors).
- GPs can be adapted for probabilistic dimensional reduction.
    - Non-linear models *even* when less data points than data dimensions.
    - Applications in graphics, vision, speech, robotics ...
- And finally ...

Acknowledgements

Collaborators:

- p53 system (BBSRC funded collaboration)
  - Guido Sanguinetti, Magnus Rattray.
- Vocal Joystick
  - Jeff Bilmes, John Malkin
- Other ongoing work with
  - Phil Torr, Carl Henrik Ek
  - Raquel Urtasun
  - Brian Ferris and Dieter Fox.

References

M. Barenco, D. Tomescu, D. Brewer, R. Callard, J. Stark, and M. Hubank.
Ranked prediction of p53 targets using hidden variable dynamic modeling.
Genome Biology, 7(3):R25, 2006.

J. Bilmes, J. Malkin, X. Li, S. Harada, K. Kilanski, K. Kirchhoff, R. Wright, A. Subramanya, J. Landay, P. Dowden, and H. Chizeck.
The vocal joystick.
In *Proceedings of the IEEE Conference on Acoustics, Speech and Signal Processing*. IEEE, May 2006. To appear.

K. Grochow, S. L. Martin, A. Hertzmann, and Z. Popovic.
Style-based inverse kinematics.
In *ACM Transactions on Graphics (SIGGRAPH 2004)*, 2004.

N. D. Lawrence.
Probabilistic non-linear principal component analysis with Gaussian process latent variable models.
Journal of Machine Learning Research, 6:1783–1816, Nov 2005.

J. Oakley and A. O'Hagan.
Bayesian inference for the uncertainty distribution of computer model outputs.
Biometrika, 89(4):769–784, 2002.

A. O'Hagan.
Curve fitting and optimal design for prediction.
Journal of the Royal Statistical Society, B, 40:1–42, 1978.

A. O'Hagan.
Some Bayesian numerical analysis.
In J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith, editors, *Bayesian Statistics 4*, pages

Consistency

Consistency of a Gaussian Process

- Predictions remain the same regardless of the number and location of the test points.

$$p\left(\mathbf{f}_*|\mathbf{f}\right) = \int p\left(\mathbf{f}_*, \mathbf{f}_+|\mathbf{f}\right) d\mathbf{f}_+,$$

- For the system to be consistent this conditional probability must be independent of the length of $\mathbf{f}_+$.
- In other words.

$$p\left(\mathbf{f}_*|\mathbf{f}\right) = \int p\left(\mathbf{f}_*, \mathbf{f}_+|\mathbf{f}\right) d\mathbf{f}_+ = \int p\left(\mathbf{f}_*, \hat{\mathbf{f}}_+|\mathbf{f}\right) d\hat{\mathbf{f}}_+$$

Joint Distribution

Joint Distribution

- The covariance function provides the joint distribution over the instantiations.
- Write down the conditional distribution provides predictions.
- Denote the training set as $\mathbf{f}$ and test set as $\mathbf{f}_*$.
    - Predict using $p\left(\mathbf{f}_*|\mathbf{f}\right)$.

The Conditional Distribution

Partioned Inverse

- Use partitioned inverse to find conditional.

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{\mathbf{f},\mathbf{f}} & \mathbf{K}_{\mathbf{f},*} \\ \mathbf{K}_{*,\mathbf{f}} & \mathbf{K}_{*,*} \end{bmatrix}$$

- Partitioned inverse is then

$$\mathbf{K}^{-1} = \left[\begin{array}{cc} \mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1} + \mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1}\mathbf{K}_{\mathbf{f},*}\Sigma^{-1}\mathbf{K}_{*,\mathbf{f}}\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1} & -\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1}\mathbf{K}_{\mathbf{f},*}\Sigma^{-1} \\ -\Sigma^{-1}\mathbf{K}_{*,\mathbf{f}}\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1} & \mathbf{-1} \end{array} \right]$$

where

$$\Sigma = \mathbf{K}_{*,*} - \mathbf{K}_{*,\mathbf{f}}\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1}\mathbf{K}_{\mathbf{f},*}.$$

Joint Distribution

Take Log of the Joint

- Logarithm of the joint distribution:

$$\begin{aligned}\log p\left(\mathbf{f}, \mathbf{f}_*\right) &= -\frac{1}{2}\mathbf{f}^{\mathrm{T}}\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1}\mathbf{f} - \frac{1}{2}\mathbf{f}^{\mathrm{T}}\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1}\mathbf{K}_{\mathbf{f},*}\Sigma^{-1}\mathbf{K}_{*,\mathbf{f}}\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1}\mathbf{f} \\ &\quad +\mathbf{f}\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1}\mathbf{K}_{\mathbf{f},*}\Sigma^{-1}\mathbf{f}_* - \frac{1}{2}\mathbf{f}_*^{\mathrm{T}}\Sigma^{-1}\mathbf{f}_* + \mathrm{const}_1\end{aligned}$$

- Conditional is found by dividing joint by the prior, $p\left(\mathbf{f}\right) = N\left(\mathbf{f}|\mathbf{0}, \mathbf{K}_{\mathbf{f},\mathbf{f}}\right)$.

Conditional Distribution

Deriving the Conditional

- In log space this is equivalent to subtraction of

$$\log p\left(\mathbf{f}\right) = -\frac{1}{2}\mathbf{f}^{\mathrm{T}}\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1}\mathbf{f} + \mathrm{const}_2$$

giving

$$\log p\left(\mathbf{f}_*|\mathbf{f}\right) = \log p\left(\mathbf{f}_*, \mathbf{f}\right) - \log p\left(\mathbf{f}\right) = \log N\left(\mathbf{f}_*|\bar{\mathbf{f}}_*, \Sigma\right).$$

where $\bar{\mathbf{f}} = \mathbf{K}_{*,\mathbf{f}}\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1}\mathbf{f}$ and $\Sigma = \mathbf{K}_{*,*} - \mathbf{K}_{*,\mathbf{f}}\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1}\mathbf{K}_{\mathbf{f},*}$.

Making Predictions

- If we observe points from the function, $\mathbf{f}$.
- We can predict the locations of functions at as yet unseen locations.
- The prediction is also a Gaussian process, with mean $\bar{\mathbf{f}}$ and covariance Σ.
- Often observe corrupted version of function.