Gaussian Processes

Daniel McDuff

MIT Media Lab

December 2, 2010

Quote of the Day

"Only in Britain could it be thought a defect to be too clever by half. The probability is that too many people are too stupid by three-quarters."

Former Prime Minister John Major assessing the priors on intelligence in the UK.

Outline

Introduction

Books and Resources

- Gaussian Processes for Machine Learning - C. Rasmussen and C. Williams. MATLAB code to accompany.
- Information Theory, Inference, and Learning Algorithms - D. Mackay.
- Video tutorials, slides, software: www.gaussianprocess.org

Notation

$\mathcal{D}$ = Training dataset, $\{ (\mathbf{x}_i, y) \mid i = 1,...n \}$
$\mathbf{x}$ = Training feature vector
X = Training features
y = Training labels
$\mathbf{x}^*$ = Test feature vector
y^* = Test labels
$\mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x'}))$ = Gaussian process with mean function, $m(\mathbf{x})$, and covariance function, $k(\mathbf{x}, \mathbf{x'})$.

Parametric Models vs. Non-parametric Models

Non-parametric models differ from parametric models. The model structure is not specified *a priori* but is determined from the data.

Parametric models:

- Support Vector Machines
- Neural Networks

Non-parametric models:

- KNN
- Gaussian processes

Supervised Learning

Our Problem: To map from finite training examples to a function that predicts for all possible inputs.

- Restrict the solution functions that we consider. **Example:** Linear regression.
  - -ve: If wrong form of function is chosen then predictions will be poor.
- Apply prior probabilities to functions that we consider more likely.
  - -ve: Infinite possibilities of functions to consider.

Theory

Definition: A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution.

- A Gaussian process is a generalization of the Gaussian probability distribution.
- It is a distribution over functions rather a distribution over vectors.
- It is a non-parametric method of modeling data.
- A Gaussian Process is of infinite dimensions. However, we only work with a finite subset.
- Fortunately, this yields the same inference as if we were to consider all the infinite possibilities.

Theory

Gaussian distributions
$\mathcal{N}(\mu, \Sigma)$

Distribution over vectors.
Fully specified by a mean and covariance.
The position of the random variable in the vector plays the role of the index.

Gaussian processes
$\mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x'}))$

Distribution over functions.
Fully specified by a mean function and covariance function.
The argument of the random function plays the role of the index.

Theory

- Covariance function defines the properties in the function space.
- Data points "anchor" the function as specific locations.

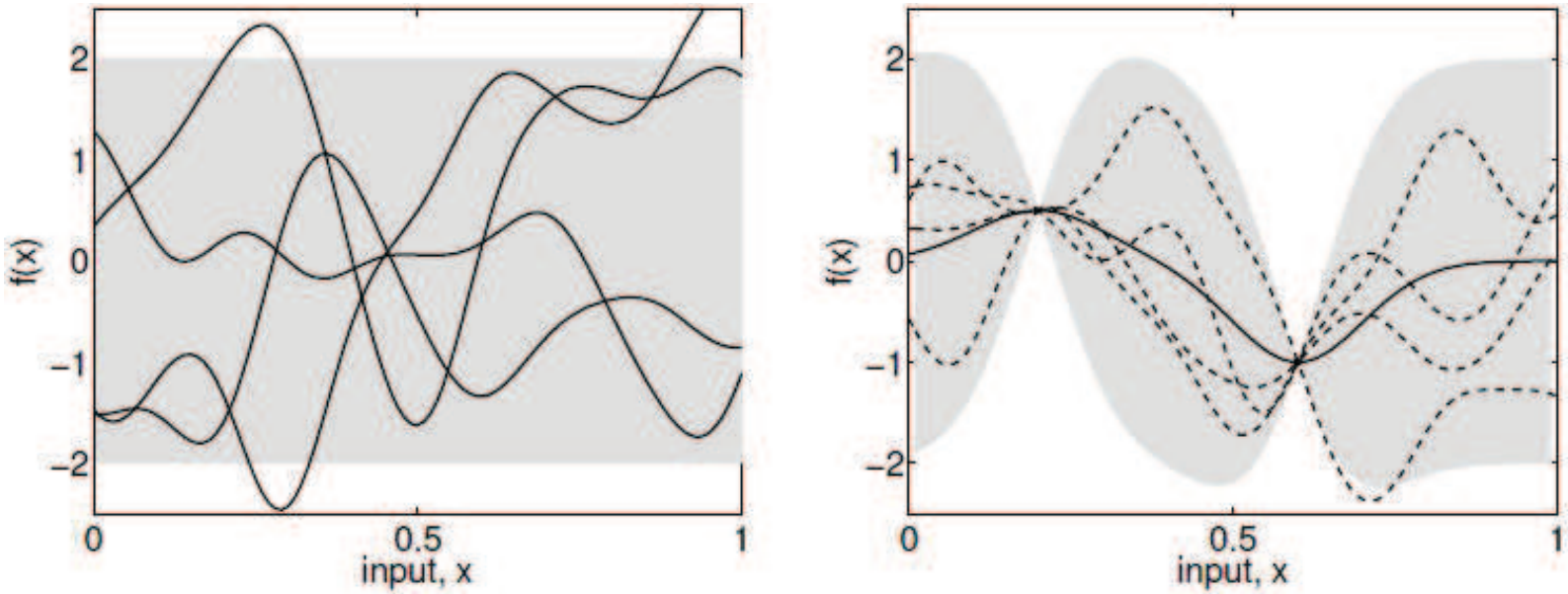

Figure: **Left:** Samples drawn from the prior distribution of functions with no observed datapoints. **Right:** Samples drawn from the prior distribution of functions with two observed datapoints. Solid black line shows the mean and shaded region twice the standard deviation. Credit: Rasmussen & Williams.

Theory

What do we need to define?

- **Mean function**, m(**x**) - Usually defined to be zero. Justified by manipulating the data.
- **Covariance function**,$k(\mathbf{x}, \mathbf{x'})$ - This defines the prior properties of the functions considered for inference.
  - Stationarity
  - Smoothness
  - Length-scales

Gaussian processes provide a well defined approach for learning model and hyperparameters from the data.

Regression

Prediction of continuous quantity, y, from input, $\mathbf{x}^*$.

We will assume the prediction to have two parts:

- A systematic variation: Accurately predicted by an underlying process $f(\mathbf{x})$.
- A random variation: Unpredictability of the system.

The result:

$$y_i = f(\mathbf{x}_i) + \epsilon_i \tag{1}$$

Bayesian Linear Regression

A simple example of Gaussian Processes:

$$y = \mathbf{x}^T \mathbf{w} + \epsilon \tag{2}$$

- Estimate a distribution over weights.
- Marginalise over weights to derive prediction.

Predictive distribution:

$$f_* | \mathbf{x}_*, X, \mathbf{y} = \mathcal{N}(\mathbf{x}_*^T A^{-1} X \, \mathbf{y}, \mathbf{x}_*^T A^{-1} \mathbf{x}_*) \tag{3}$$

Projections Into Feature Space

In linear regression we constrain the relationship between the observations and the output to be linear. An alternative is to use a kernel:

$$\Phi = \begin{bmatrix} \phi(||\mathbf{x}_1 - \mathbf{x}_1||) & \phi(||\mathbf{x}_1 - \mathbf{x}_2||) & \dots & \phi(||\mathbf{x}_1 - \mathbf{x}_n||) \\ \phi(||\mathbf{x}_2 - \mathbf{x}_1||) & \phi(||\mathbf{x}_2 - \mathbf{x}_2||) & \dots & \phi(||\mathbf{x}_2 - \mathbf{x}_n||) \\ \dots & \dots & \dots & \dots \\ \phi(||\mathbf{x}_n - \mathbf{x}_1||) & \phi(||\mathbf{x}_n - \mathbf{x}_2||) & \dots & \phi(||\mathbf{x}_1 - \mathbf{x}_n||) \end{bmatrix}$$

Measures similarity between feature points.

Projections Into Feature Space

Change the model to:

$$y = \phi(\mathbf{x})^T \mathbf{w} + \epsilon \tag{4}$$

Where $\phi()$ projects $\mathbf{x}$ into a new space.

The predictive distribution becomes:

$$f_* | \mathbf{x}_*, X, \mathbf{y} = \mathcal{N}(\frac{1}{\sigma_n} \phi(\mathbf{x}_*)^T A^{-1} \phi \ \mathbf{y}, \phi(\mathbf{x}_*)^T A^{-1} \phi(\mathbf{x}_*)) \tag{5}$$

Where: ϕ is an aggregation of all $\phi(\mathbf{x})$'s in the training set.

Function Space View

To generate functions we generate random Gaussian vectors with a covariance matrix defined by your input points:

$$\mathbf{f}_* \sim \mathcal{N}(\mathbf{0}, K(X_*, X_*)) \tag{6}$$

Function Space View

Since we are interested in making predictions more than just generating functions we want to incorporate the knowledge of training data. For a GP with zero mean and covariance function, $K(X, X) + \sigma_n{}^2 I$ the join distribution of training outputs and test outputs is:

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \mathrm{N}(\mathbf{0}, \begin{bmatrix} K(X,X) + \sigma_n^2 I & K(X, X_*) \\ K(X_*, X) & K(X_*, X_*) \end{bmatrix})$$

The predictive equations for GP regression:

$$\mathbf{f}_* | X, \mathbf{y}, X_* \sim \mathcal{N}(m(\mathbf{x}), covf) \tag{7}$$

Where:

$$m(\mathbf{x}) = K(X_*, X)[K(X,X) + \sigma_n^2 I]^{-1}\mathbf{y}$$

$$covf = K(X_*, X_*) - K(X_*, X)[K(X,X) + \sigma_n^2 I]^{-1} K(X, X_*)$$

MATLAB Example

MATLAB Example. Code adapted from GPML demo - gaussianprocess.org

Covariance Functions

Represents the covariance between pairs of random variables.

$$K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) \tag{8}$$

$$\mathrm{K}_{xx} = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \dots & k(x_1, x_n) \\ k(x_2, x_1) & k(x_1, x_1) & \dots & k(x_2, x_n) \\ \dots & \dots & \dots & \dots \\ k(x_n, x_1) & k(x_n, x_2) & \dots & k(x_n, x_n) \end{bmatrix}$$

Examples:

- Squared exponential
- Matern
- γ -exponential
- Rational quadratic
- Noise (i.e. White noise - flat spectrum).

Covariance Functions

The covariance function must be:

- Positive semi-definite
- Symmetric

Covariance functions can be split broadly into two groups:

- **Stationary**: A function of $\mathbf{x}_i = \mathbf{x}_j$. Invariant to translations in the input space.
- **Non-stationary**: Functions vary with translation.

Most important sub-set of these covariance functions are dot-product functions:

- Linear - $\mathrm{k}(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$
- Polynomial - $\mathrm{k}(\mathbf{x}_i, \mathbf{x}_j) = (\langle \mathbf{x}_i, \mathbf{x}_j \rangle + \mathrm{d})^d$

Squared Exponential

Commonly used covariance function:

$$K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) = \alpha exp\{-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma_1^2}\} \tag{9}$$

Hyperparameters:

- σ_1 = Characteristic lengthscale
- α = Signal variance

Large for inputs that are close to one another. Decreases are distances in the input space increases.

Squared Exponential

We normally assume some prediction noise:

$$K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) = \alpha exp\{-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma_1^2}\} + \sigma_n^2 \delta_{ij} \tag{10}$$

Hyperparameters:

- σ_1 = Characteristic lengthscale
- α = Signal variance

Noise

The impact of different values of noise variance:

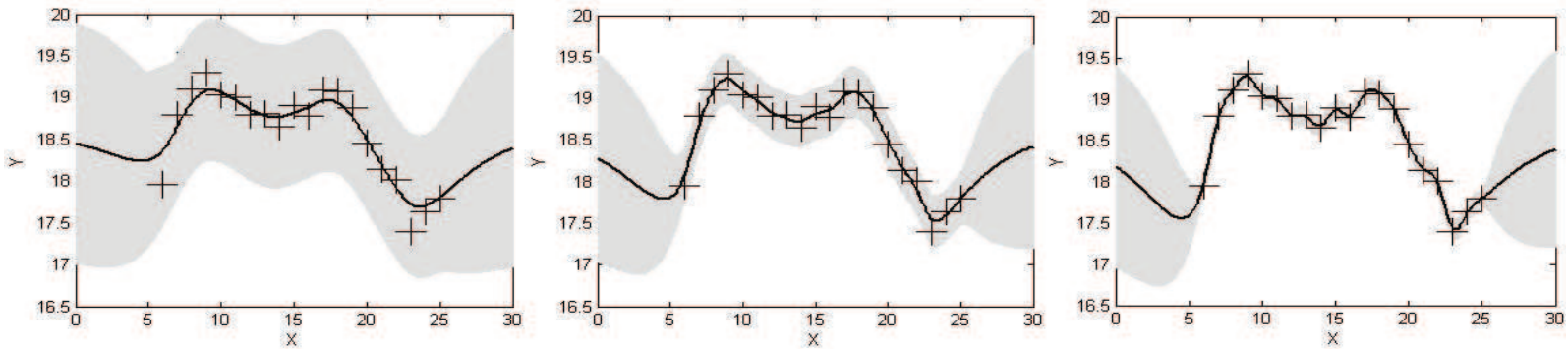

Figure: Comparison of Gaussian processes with different values of noise variance. Left: Too great?, Middle: Just right?, Right: Too small?

Squared Exponential

We normally assume some prediction noise:

$$K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) = \alpha exp\{-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma_1^2}\} + \sigma_n^2 \delta_{ij} \tag{11}$$

Hyperparameters:

- σ_1 = Characteristic lengthscale
- α = Signal variance

Lengthscales

- Lengthscales

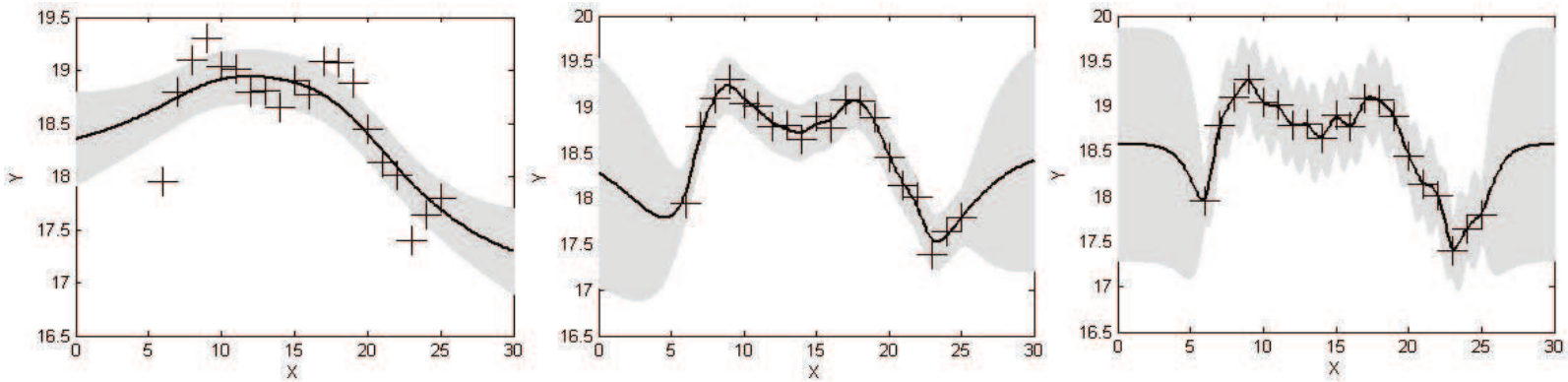

Figure: Comparison of Gaussian processes with different lengthscales.

Learning in Gaussian Processes

Bayesian evidence is the probability of the data given the model.

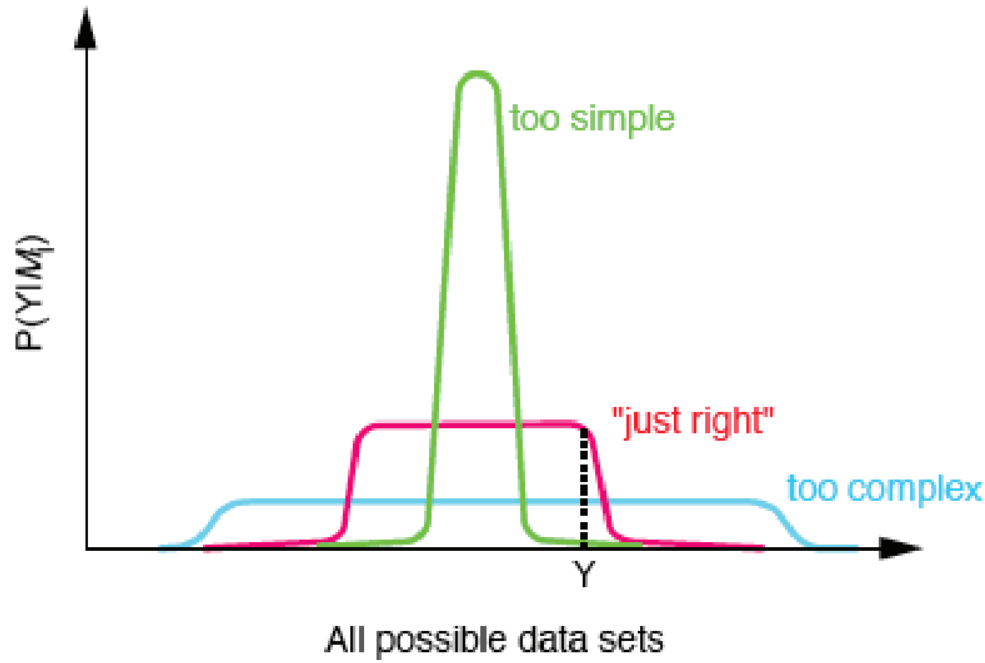

Figure: Complex models that account for many datasets only achieve modest evidence. If the model is too simple evidence may be high but only for few datasets (Credit: Carl Rasmussen).

Remember Occam's Razor.

Learning in Gaussian Processes

One approach this is to maximise the marginal likelihood:

$$p(\mathbf{x}|X,\theta) = \mathcal{N}(\mathbf{y};\mathbf{0}, K_\theta(X,X) + \sigma_n^2 I) \tag{12}$$

In log form:

$$log(p(\mathbf{x}|X,\theta)) = -\frac{1}{2}\mathbf{y}^T\Sigma_y^{-1}\mathbf{y} - \frac{1}{2}log(|\Sigma_y|) - \frac{n}{2}log(2\pi) \tag{13}$$

Where:

$$\Sigma_y = K_\theta(X,X) + \sigma_n^2 I$$

- Data fit: -$\frac{1}{2}\mathbf{y}^T{\Sigma_y}^{-1}\mathbf{y}$
- Complexity penalty: -$\frac{1}{2}$log($|\Sigma_y|$)

Learning in Gaussian Processes

Optimising the Hyper-parameters:

Partial differentials wrt to each θ_j (see Appendix for matrix identities):

$$\frac{\partial}{\partial \theta_j} log(p(\mathbf{x}|X,\theta)) = \frac{1}{2}\mathbf{y}^T K^{-1}\frac{\partial K}{\partial \theta_j}K^{-1}\mathbf{y} - \frac{1}{2}tr(K^{-1}\frac{\partial K}{\partial \theta_j}) \quad (14)$$

$$= \frac{1}{2}tr((K^{-1}\mathbf{y}\mathbf{y}^T(K^T)^{-1} - K^{-1})\frac{\partial K}{\partial \theta_j}) \quad (15)$$

We cannot necessarily guarantee we will find a global optimum here and different solutions may lead to different interpretations of the data.

Impact of Covariance Functions

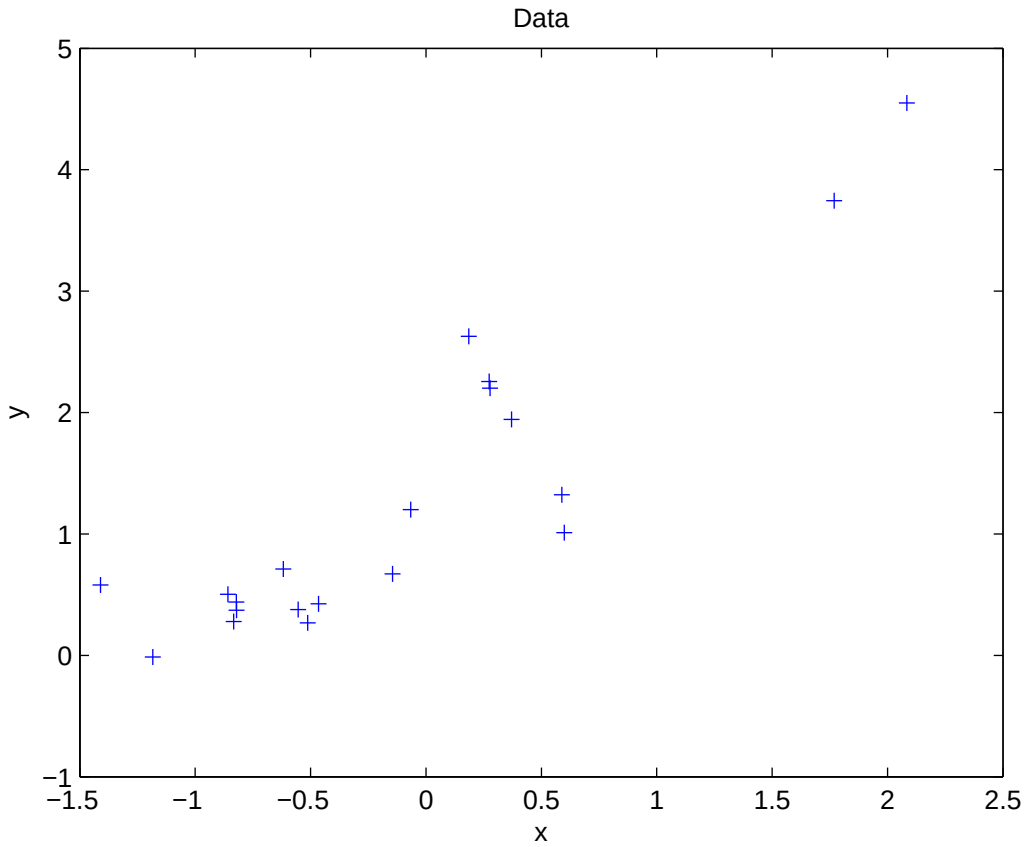

Figure: Data generated from a GP with affine mean function and Matern covariance function with Gaussian noise.

Impact of Covariance Functions

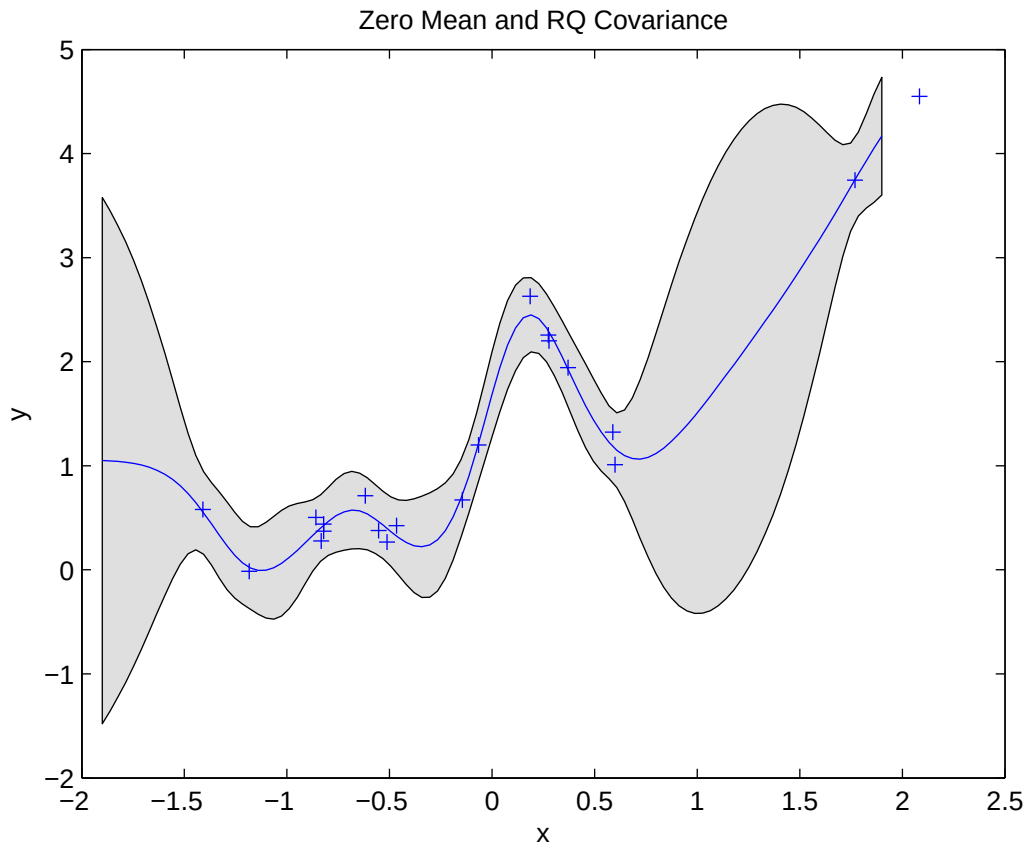

Figure: Negative log marginal likelihood: 12.54.

Impact of Covariance Functions

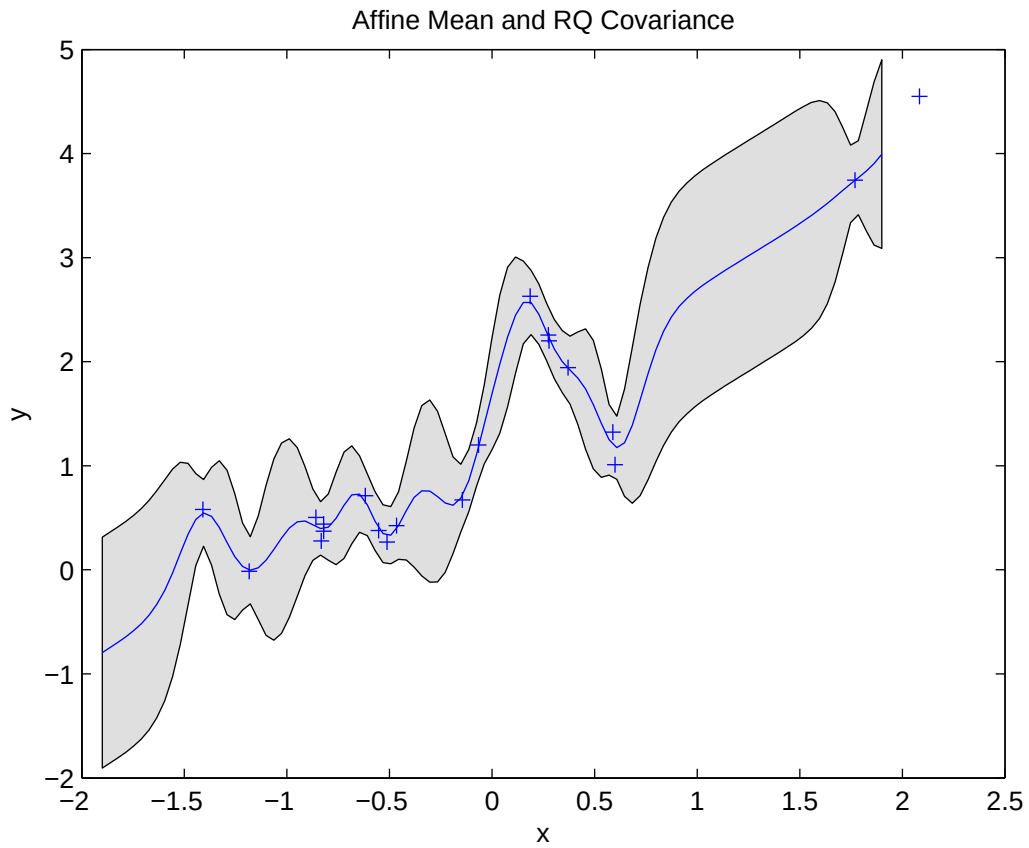

Figure: Negative log marginal likelihood: 6.90.

Impact of Covariance Functions

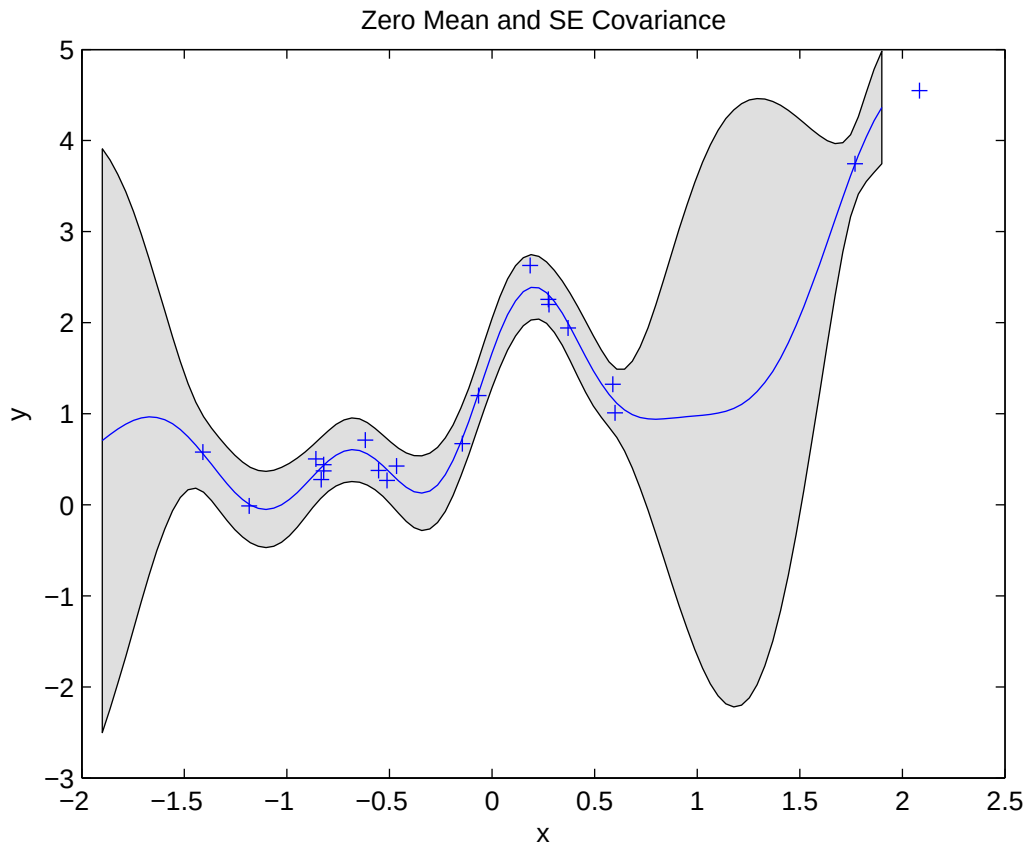

Figure: Negative log marginal likelihood: 14.13.

Impact of Covariance Functions

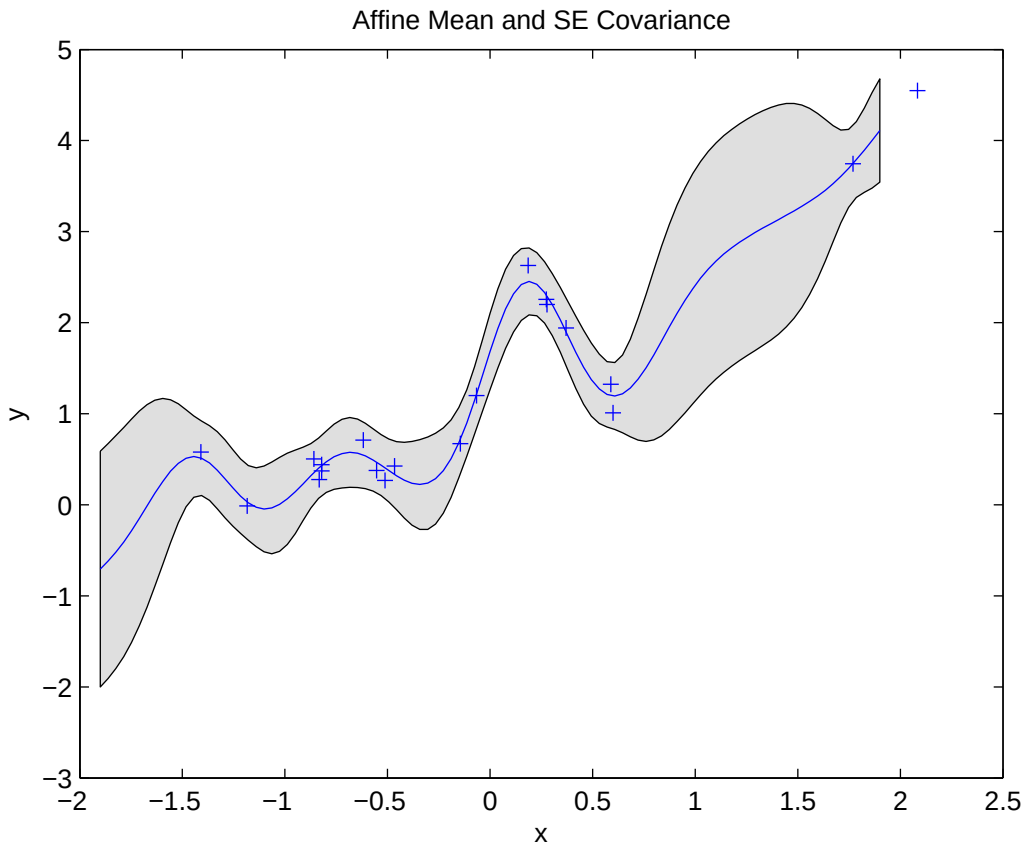

Figure: Negative log marginal likelihood: 7.07.

Classification

Prediction of discrete classes, $y \in \{$ -1, 1 $\}$, from input, $\mathbf{x}^*$. .

Classification with a Regression Model:

- Map output of GP regression from range $\{-\infty, +\infty\}$ to probabilities in the range $\{0, 1\}$.

One possible method - logistic regression:

$$P(y = +1|\mathbf{x}, \hat{\mathbf{w}}) = \frac{1}{1 + exp(-\hat{\mathbf{w}}^T \mathbf{x})} \tag{16}$$

$$P(y = -1|\mathbf{x}, \hat{\mathbf{w}}) = 1 - P(y = 1|\mathbf{x}, \hat{\mathbf{w}})$$

Classification

There is a more general form of classification. We will not go into the details of this here. However, the concept is similar to the regression case. It once again involves marginalising over the model parameter distribution.

Unfortunately unlike the regression case there is not a simple analytical solution.

Classification

In classification, the posterior $P(f|D, \Theta)$ is intractable because it involves an integral that is the product of a Gaussian and a product of sigmoids.

Approximation methods:

- Laplace approximation [Barber & Williams]
- Variational methods [Gibbs & MacKay]
- Expectation-Propagation [Minka & Ghahramani]
- MCMC sampling [Neal]

Complexity

- The complexity of Gaussian processes is of the order N^3, where N is the number of data points, due to the inversion of the N x N matrix.

$$\sigma_n^{-2}(\sigma_n^{-2}XX^T + \Sigma_p^{-1})^{-1} \tag{17}$$

Applications

Predicting financial markets:

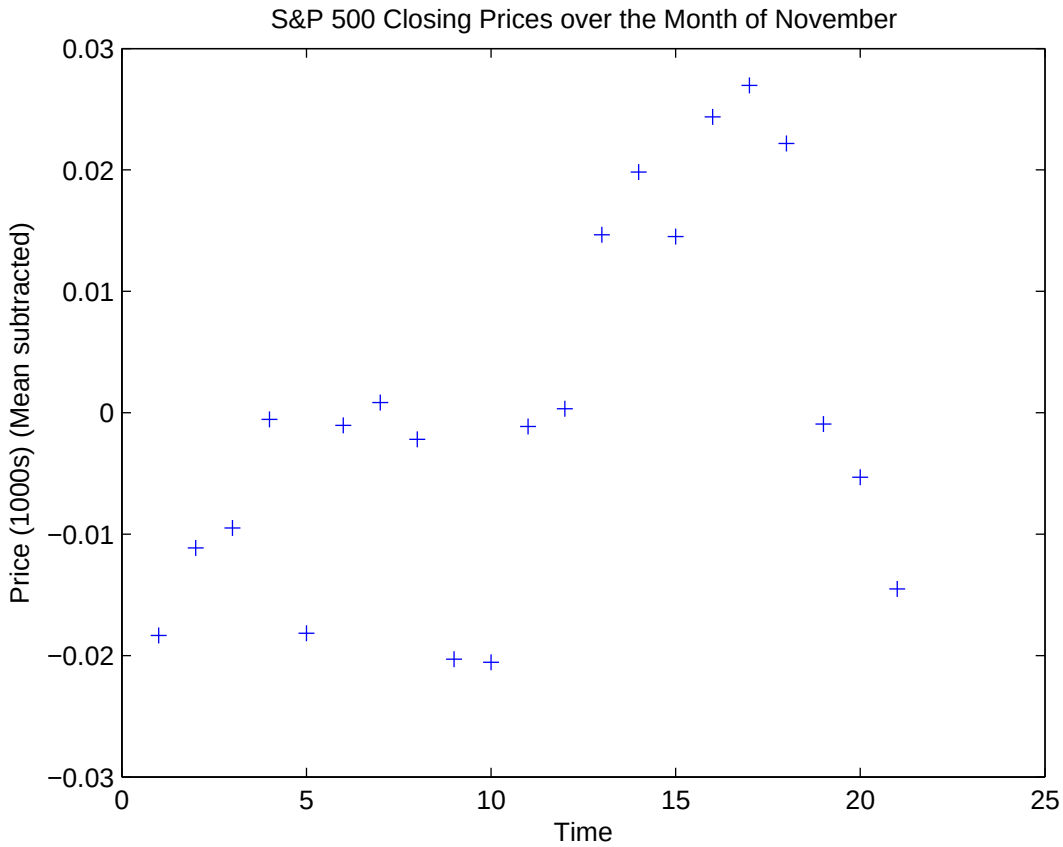

Figure: S&P closing prices from November 2010.

Applications

Predicting financial markets:

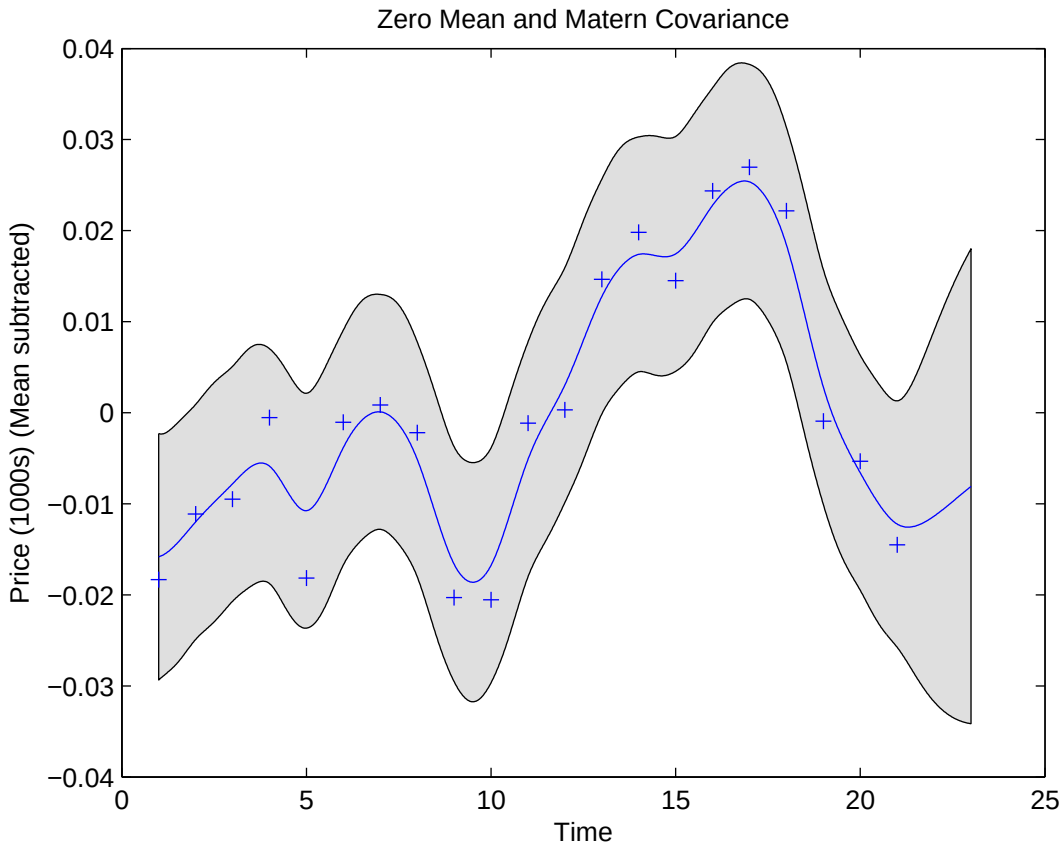

Figure: S&P closing prices from November 2010.

Applications

Predicting financial markets:

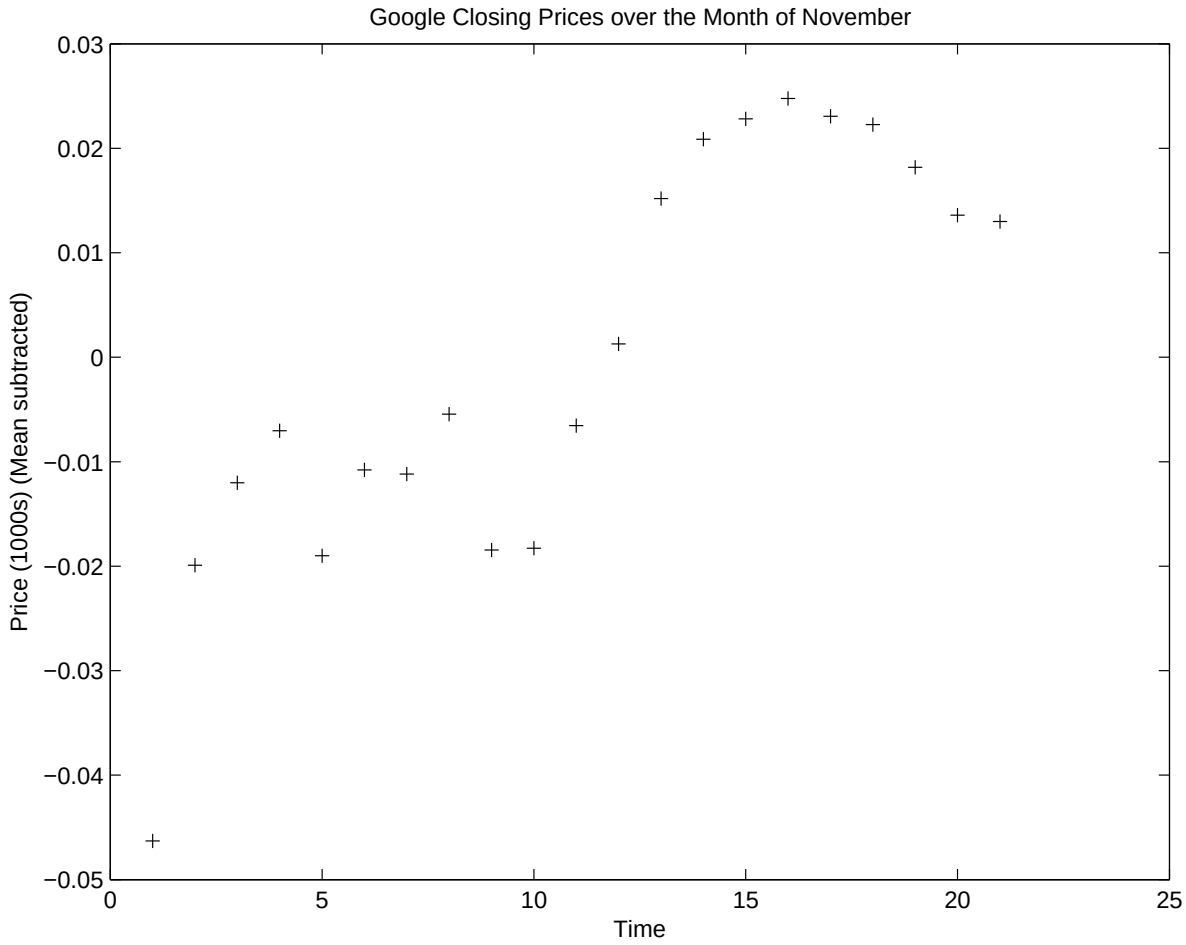

Figure: Google closing prices from November 2010.

Applications

Predicting financial markets:

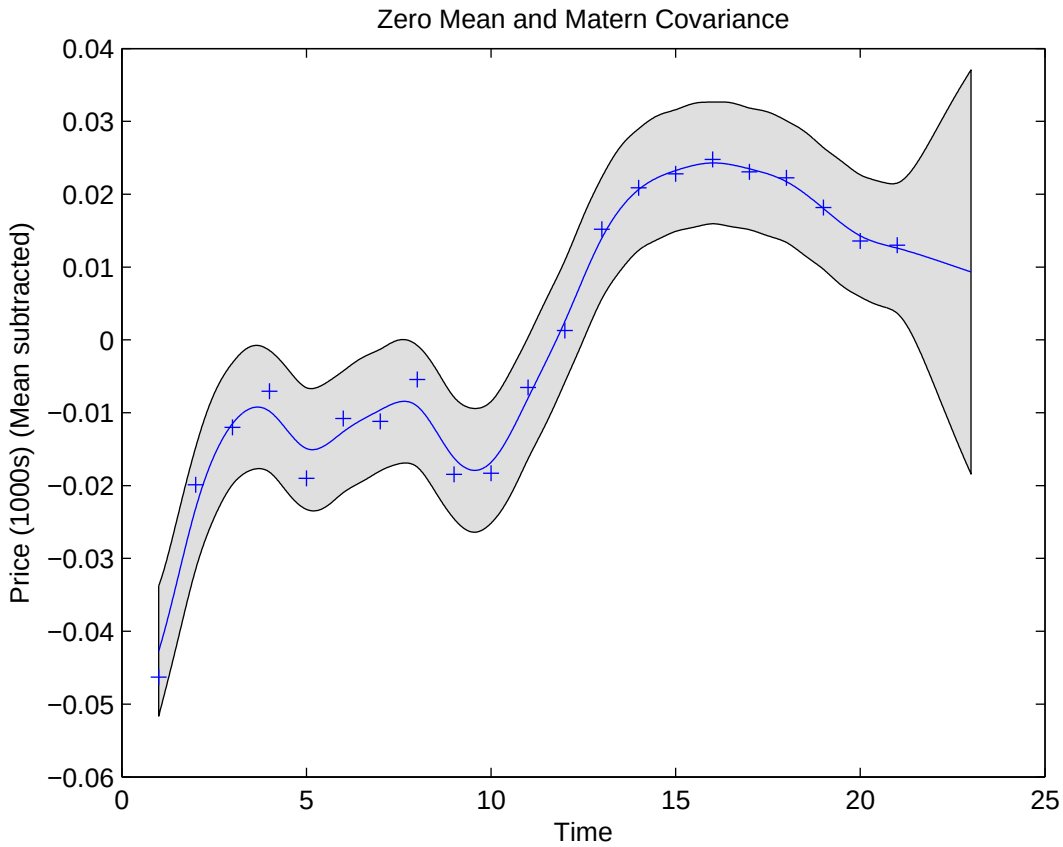

Figure: Google closing prices from November 2010.

Key Points

- Gaussian processes are non-parametric.
- They provide a structured method of model and parameter selection.
- A Gaussian process is defined by a mean and covariance function.
- Learning takes the form of setting the hyper-parameters. Occam's Razor is implicit.
- GP's can be used for regression or classification.

Questions

Appendix

Matrix identities for optimisation of hyper-parameters:

$$\frac{\partial}{\partial \theta} K^{-1} = -K^{-1} \frac{\partial K}{\partial \theta} K^{-1} \tag{18}$$

$$\frac{\partial}{\partial \theta} log(|K|) = tr(K^{-1} \frac{\partial K}{\partial \theta}) \tag{19}$$