Bayesian Kernel Methods

Unit 1: Bayes Rule, Approximate Inference, Hyperparameters

Unit 2: Gaussian Processes, Covariance Function, Kernel

Unit 3: GP: Regression

Unit 4: GP: Classification

Unit 5: Implementation: Laplace Approximation, Low Rank Methods

Unit 6: Implementation: Low Rank Methods, Bayes Committee Machine

Unit 7: Relevance Vector Machine: Priors on Coefficients

Unit 8: Relevance Vector Machine: Efficient Optimization and Extensions

http://mlg.anu.edu.au/∼smola/summer2002/

Overview of Unit 3: GP Regression

01: Noiselsess Case: Joint Covariance
02: Conditioning on Observations
03: Adding two Normal Distributions
04: Regression with Normal Noise
05: Examples
06: Error Bars for GP Regression
07: Examples
08: Hyperparameters
09: Automatic Relevance Determination
10: Generic Noise Models
11: More Noise Models
12: Even More Noise Models
13: Transformation into Lagrange Multipliers
14: MAP Approximation
15: Connection to Support Vectors
16: Scaling Problems

Noiselsess Case: Joint Covariance

Recall: Assumptions

Observations $\mathbf{t}$ are samples from a Gaussian process with mean μ and covariance matrix K.

Recall: Goal

After observing $\mathbf{t} := (t(x_1), \dots, t(x_m))$ we would like to infer the distribution of t at locations $x'_1, \dots, x'_n$, i.e., we would like to infer about $\mathbf{t}' := (t(x'_1), \dots, t(x'_n))$.

Lazy Trick

The solution is to study $p(\mathbf{t}'|\mathbf{t})$. For normal distributions we only need to compute **mean** and **covariance** to determine the density completely (including normalization factors). We have

$$p(\mathbf{t}, \mathbf{t}') \propto \exp\left(-\frac{1}{2}\left(\begin{bmatrix}\mathbf{t}\\ \mathbf{t}'\end{bmatrix} - \begin{bmatrix}\mu\\ \mu'\end{bmatrix}\right)^\top \begin{bmatrix}K_{\mathbf{t}\mathbf{t}} & K_{\mathbf{t}\mathbf{t}'}\\ K_{\mathbf{t}'\mathbf{t}} & K_{\mathbf{t}'\mathbf{t}'}\end{bmatrix}^{-1} \left(\begin{bmatrix}\mathbf{t}\\ \mathbf{t}'\end{bmatrix} - \begin{bmatrix}\mu\\ \mu'\end{bmatrix}\right)\right)$$

Recall: Inverting the Covariance Matrix

Inverting the Covariance Matrix

$$\begin{bmatrix} K_{\mathbf{tt}} & K_{\mathbf{tt}'} \\ K_{\mathbf{tt}'}^{\top} & K_{\mathbf{t}'\mathbf{t}'} \end{bmatrix}^{-1} = \begin{bmatrix} {K_{\mathbf{tt}}}^{-1} - \left({K_{\mathbf{tt}}}^{-1} K_{\mathbf{tt}'}^{\top}\right)^{\top} \chi^{-1} \left({K_{\mathbf{tt}}}^{-1} K_{\mathbf{tt}'}^{\top}\right) & -\left({K_{\mathbf{tt}}}^{-1} K_{\mathbf{tt}'}^{\top}\right) \chi^{-1} \\ -\chi^{-1} \left({K_{\mathbf{tt}}}^{-1} K_{\mathbf{tt}'}^{\top}\right)^{\top} & \chi^{-1} \end{bmatrix}$$

where $\chi = K_{\mathbf{t}'\mathbf{t}'} - K_{\mathbf{tt}'}^{\top} {K_{\mathbf{tt}}}^{-1} K_{\mathbf{tt}'}$ (Schur complement).

Reduced Covariance

From the inverse of the covariance matrix we obtain that the only quadratic part in $\mathbf{t}'$ is given by χ. Thus the **variance in t′ is y reduced** from $K_{\mathbf{t}'\mathbf{t}'}$ to $K_{\mathbf{t}'\mathbf{t}'} - K_{\mathbf{tt}'}^{\top} {K_{\mathbf{tt}}}^{-1} K_{\mathbf{tt}'}$ by observing $\mathbf{t}$.

Predictive Mean

Instead of μ' the mean is shifted to $\mu' + K_{\mathbf{tt}'}^{\top} K_{\mathbf{tt}}^{-1} (\mathbf{t} - \mu)$.

Adding two Normal Distributions

Goal

Regression with Gaussian Processes with additive normal noise: here we need to compute the distribution obtained from the sum of two normal distributions.

Theorem (for simplicity only in $\mathbb{R}$)

Denote by ξ, ξ' random variables with $\xi \sim \mathcal{N}(\mu, \sigma^2)$ and $\xi' \sim \mathcal{N}(\mu', \sigma'^2)$. Then $\xi + \xi' \sim \mathcal{N}(\mu + \mu', \sigma^2 + \sigma'^2)$.

Proof

The density arising from the sum of two random variables is given by the convolution of the densities, i.e. $p(\xi + \xi') = (p \circ p')(\xi + \xi')$. The means are clearly given by $\mu + \mu'$. For the rest assume zero mean:

$$p \circ p' = \mathcal{F}^{-1}[\mathcal{F}[p] \cdot \mathcal{F}[p']] \propto \mathcal{F}^{-1}\left[e^{-\frac{\sigma^2}{2}\omega^2} e^{-\frac{\sigma'^2}{2}\omega^2}\right] = \mathcal{F}^{-1}\left[e^{-\frac{\sigma^2+\sigma'^2}{2}\omega^2}\right]$$

Here we see that the covariances add up, hence we obtain $\mathcal{N}(\mu + \mu', \sigma^2 + \sigma'^2)$. The general case can be reduced to $\mathbb{R}$ by simultaneous diagonalization.

Regression with Normal Noise

Idea

If we have $y_i = t_i + \xi_i$ where $\mathbf{t} \sim \mathcal{N}(0, K)$ and $\xi_i \sim \mathcal{N}(0, \sigma^2)$, we know that $\mathbf{y}$, being the sum of two normal random variables, satisfies $\mathbf{y} \sim \mathcal{N}(0, K + \sigma^2 \mathbf{1})$.

Posterior Density

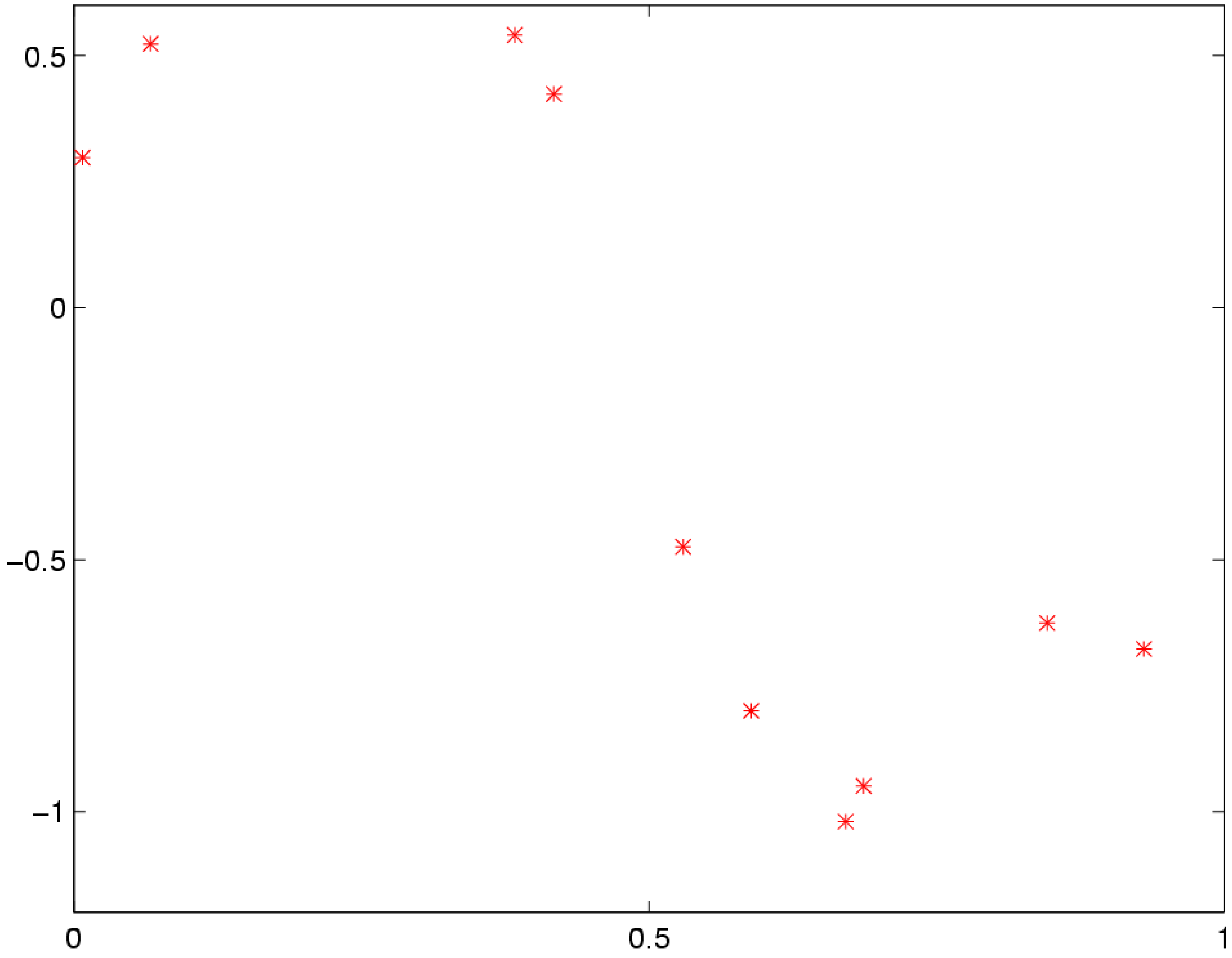
$$p(\mathbf{y}|X) = (2\pi)^{-\frac{n}{2}} (\det(K + \sigma^2 \mathbf{1}))^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\mathbf{y}^\top (K + \sigma^2)^{-1} \mathbf{y}\right)$$

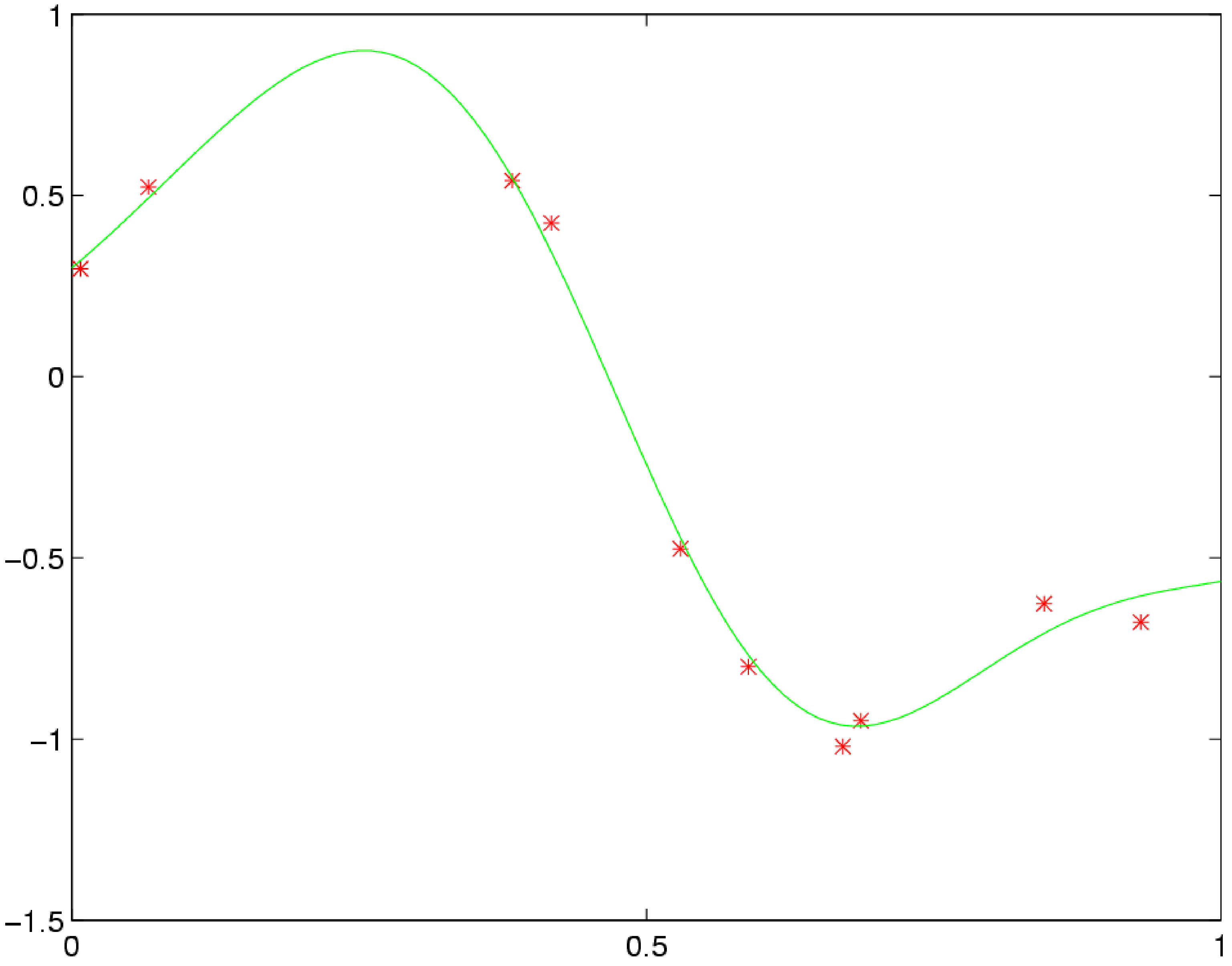
Note that the problem of non-invertibility of the covariance matrix disappeared (similar to regularization to improve the condition of a matrix).

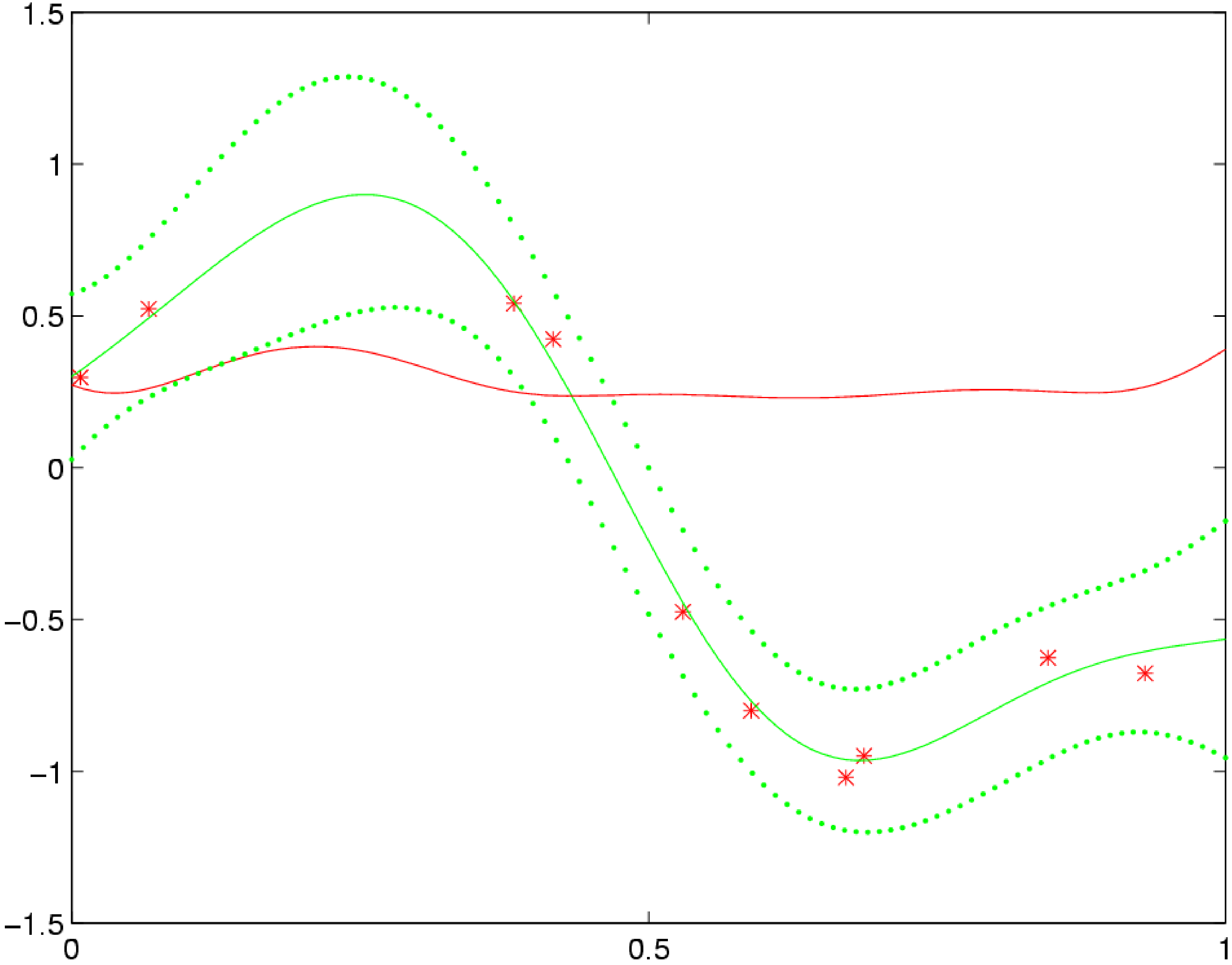
Inference

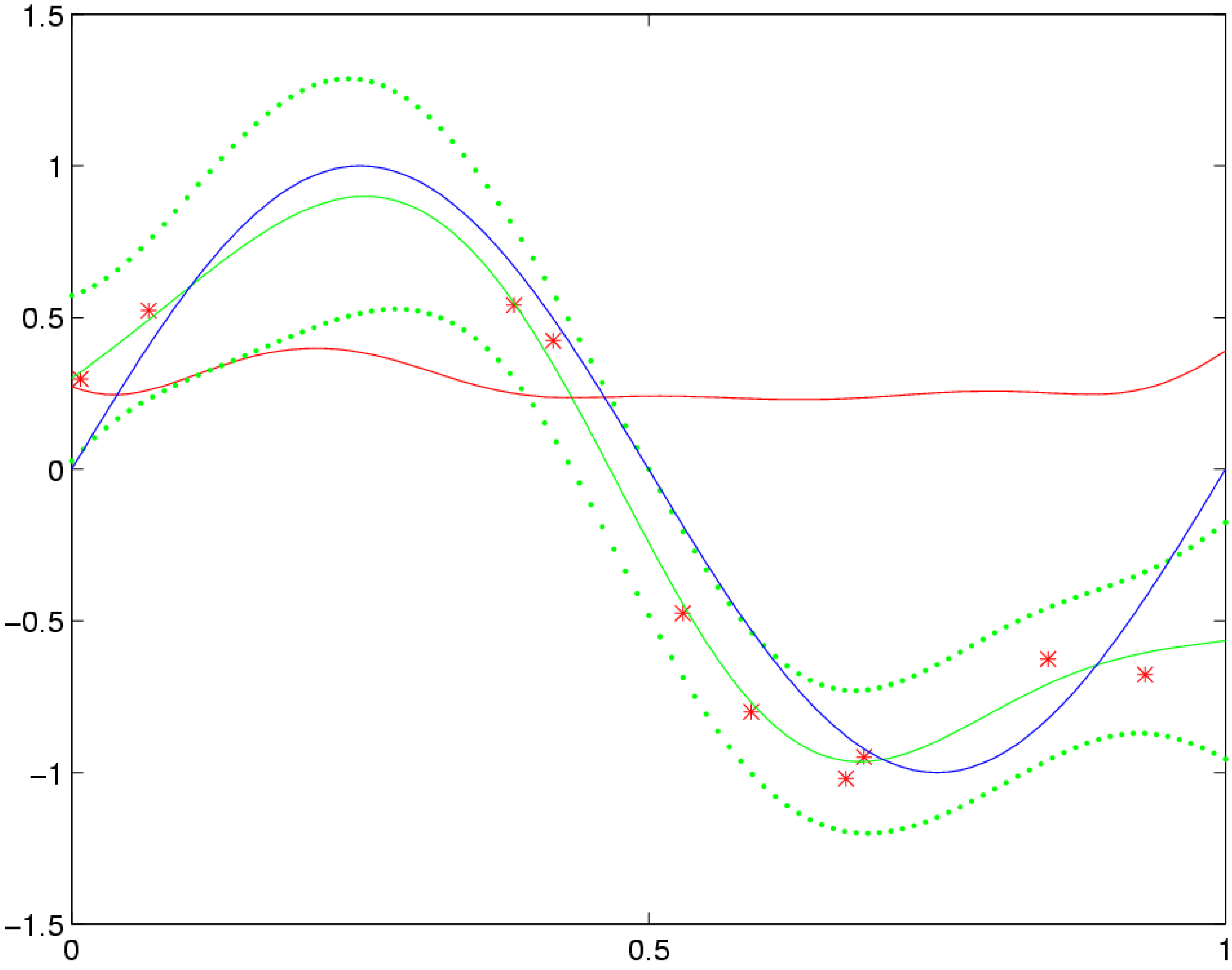
We can simply re-use the results from inference without noise and obtain (for inferring $\mathbf{y}'$ after observing $\mathbf{y}, X, X'$): $\mathbf{y}' \sim \mathcal{N}(\mu_y, \Sigma_y)$where

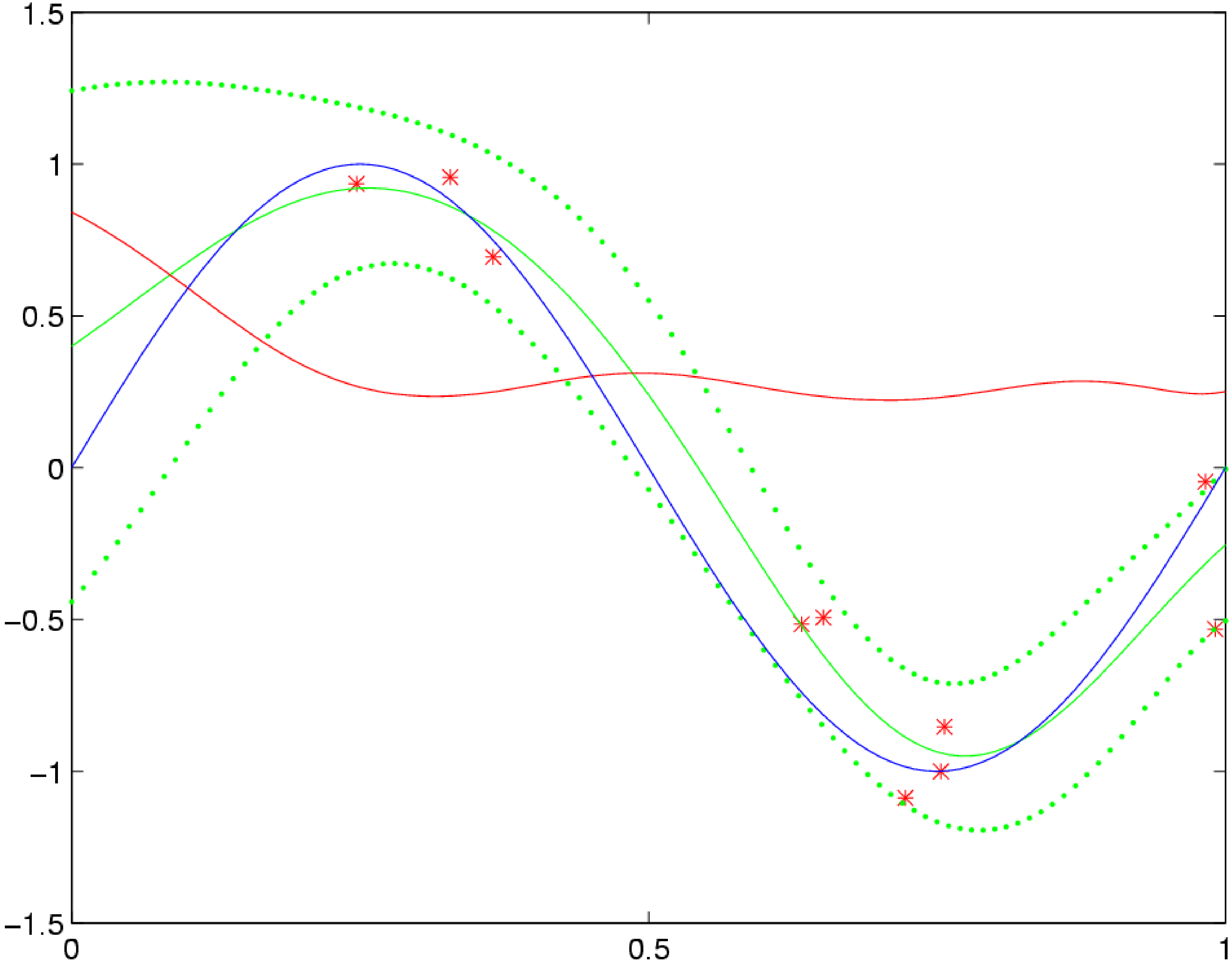
$$\mu_y = K_{\mathbf{tt}'}^\top (K_{\mathbf{tt}} + \sigma^2 \mathbf{1})^{-1} \mathbf{y} \text{ and } \Sigma_y = K_{\mathbf{t}'\mathbf{t}'} + \sigma^2 \mathbf{1} - K_{\mathbf{tt}'}^\top (K_{\mathbf{tt}} + \sigma^2 \mathbf{1})^{-1} K_{\mathbf{tt}'}$$


0.5
0
−0.5
−1
0
0.5
1


1
0.5
0
−0.5
−1
−1.5
0
0.5
1


1.5
1
0.5
0
−0.5
−1
−1.5
0
0.5
1


1.5
1
0.5
0
−0.5
−1
−1.5
0
0.5
1


1.5
1
0.5
0
−0.5
−1
−1.5
0
0.5
1

Hyperparameters

Problem

We do not know the exact values of σ, the correlation width ω of the kernel (for Gaussian RBF), etc., so we have to avoid making too specific guesses.

Solution

Treat σ, ω as hyperparameters and put a prior on the distribution of them. For simplicity, we only study σ:

$$p(f|X, Y) = \int p(f|X, Y, \sigma)p(\sigma)d\sigma$$

MAP2 approximation leads to $\mathrm{argmax}_{f,\sigma}\, p(Y|f, X, \sigma)p(f)p(\sigma)$.

Regression with Normal Noise

We can take advantage of the fact that $\mathbf{y}$ is taken from a normal distribution. So the problem of finding an appropriate value of σ reduces to

$$\operatorname*{argmax}_{\sigma} \frac{1}{2} \log\det(K + \sigma^2\mathbf{1}) + f^\top(K + \sigma^2\mathbf{1})^{-1} f$$

Matrix Magic

THE AUSTRALIAN NATIONAL UNIVERSITY

Derivatives of the Inverse

We need to compute $\partial_{\sigma^2} f^\top (K + \sigma^2 \mathbf{1})^{-1} f$.

$$0 = \partial_t (A^{-1} A) = \partial_t A^{-1} A + A^{-1} \partial_t A \text{ hence } \partial_t A^{-1} = A^{-1} (\partial_t A) A^{-1}$$

This leads to

$$\partial_{\sigma^2} f^\top (K + \sigma^2 \mathbf{1})^{-1} f = \|(K + \sigma^{-2} \mathbf{1})^{-1}\|^2$$

Derivatives of the Log-Determinant

To compute $\partial_{\sigma^2} \log \det (K + \sigma^2 \mathbf{1})$ note that $\frac{d}{dA} \log \det A = A^*$. The latter can be seen as follows:

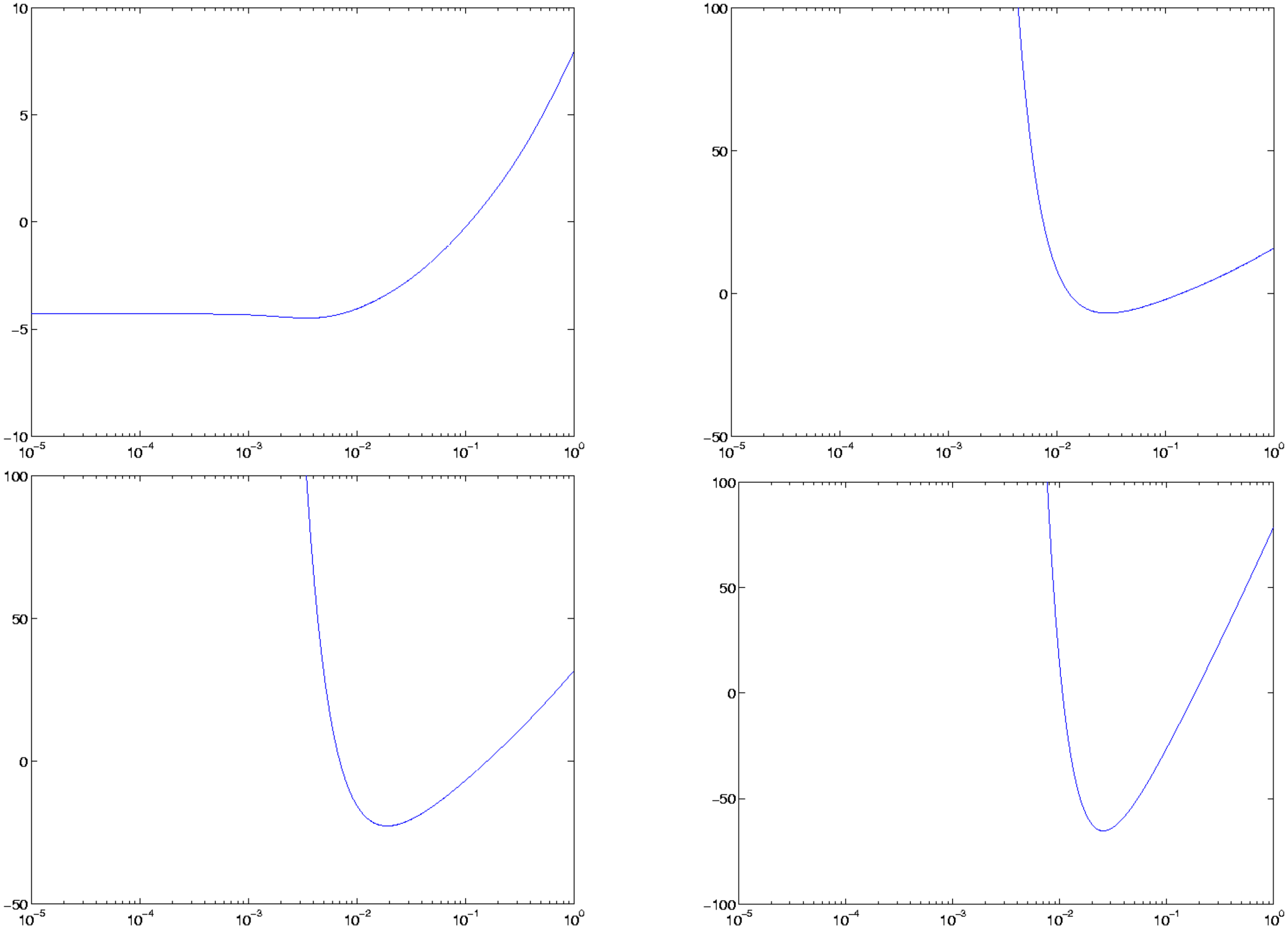
$$\partial_{A_{ij}} \log \det A = \frac{1}{\det A} \partial_{A_{ij}} \det A = \frac{1}{\det A} \partial_{A_{ij}} \det \bar{A}_{ij}$$

where $\bar{A}$ is the matrix of cofactors of A. This yields

$$\partial_{\sigma^2} \log \det (K + \sigma^2 \mathbf{1}) = \operatorname{tr} \left((K + \sigma^2 \mathbf{1})^{-1} \partial_{\sigma^2} (K + \sigma^2 \mathbf{1}) \right) = \operatorname{tr} (K + \sigma^2 \mathbf{1})^{-1}.$$

This allows us to compute the gradient wrt. σ^2 and optimize.

Example: Adjusting σ $(m = 5, 10, 20, 50)$



Automatic Relevance Determination

Problem

Which is the proper scale of the data (some inputs more important than others)? Which inputs are relevant?

Scaling of Data

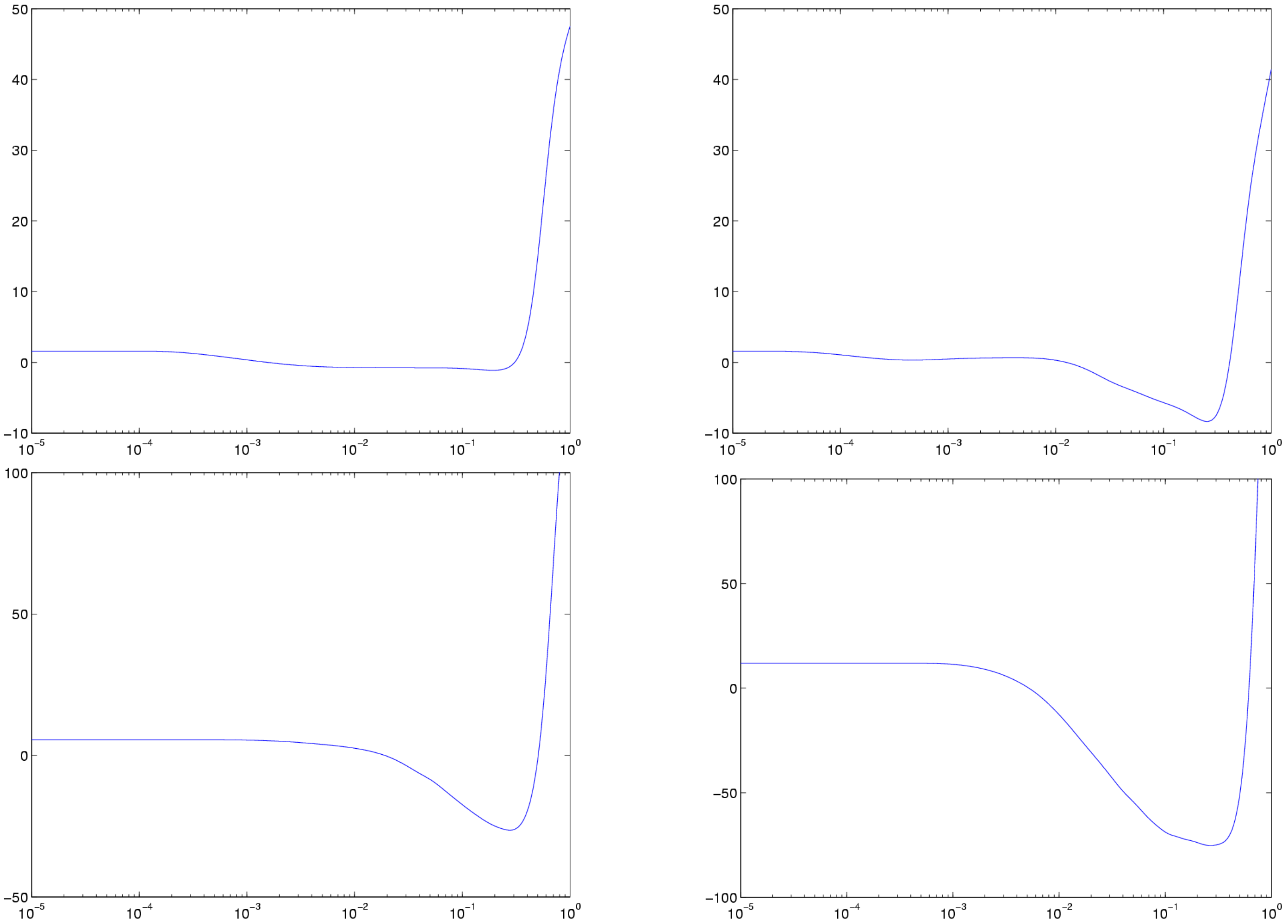
Rescale inputs $\mathbf{x}$ by scaling matrix Ω, i.e. $x \to \Omega x$ (typically we use a diagonal matrix, as it has fewer parameters). Assume hyperprior on Ω and repeat MAP2 procedure. This leads to

$$p(f|X,Y) \propto \int p(Y|\Omega X, f)p(f)p(\Omega)d\Omega$$

Improper Prior

Often it is convenient to use a function $p(\omega)$ (not only for ARD, though), that does not correspond to a finite measure, often called an improper prior (since there $\int p(f|\sigma)p(\sigma)d\sigma$ is not defined). **Note: the MAP2 procedure works regardless.**

Example: Adjusting ω $(m = 5, 10, 20, 50)$



Additive Noise Models

Additive Noise: Often, we have an underlying effect, say $f(x)$, which is corrupted by additive noise ξ such that we observe $y = f(x) + \xi$.

Simplifying Assumptions

Typically we assume that the random variables ξ are **uncorrelated and have zero mean**, i.e. $\mathbf{E}\xi = 0$ and $\mathbf{E}\xi\xi' = 0$ for all ξ, ξ'.

Furthermore we typically assume that ξ **is independent of** x (no heteroscedasticity). This means that there exists one density $p(\xi)$ governing the whole noise process. Under the iid assumption the posterior can now be written as

$$p(f|X, Y) \propto p(f) \prod_{i=1}^{m} p(y_i - f(x_i))$$

Note

There are many cases where the noise depends on the size of f itself, such as measurements which provide only relative precision. We are treating only a **very special** case (which works very well in practice, though).

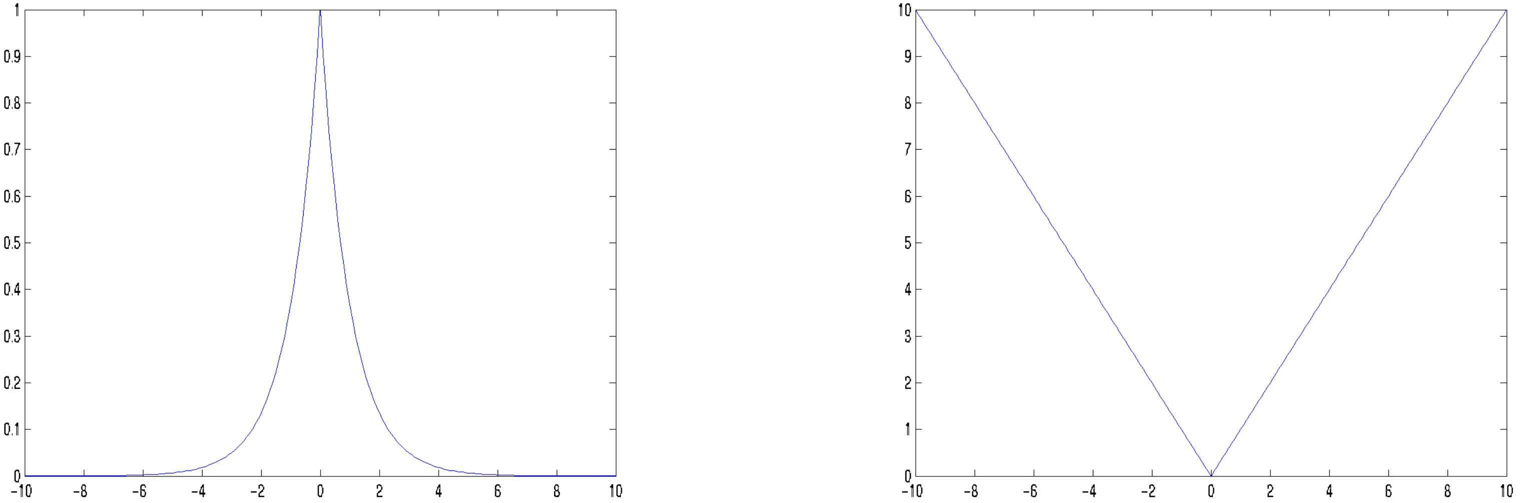
Laplacian Noise

Noise Model

$$p(\xi) = \frac{\sigma}{2} \exp(-\sigma|\xi|)$$

This is a very long-tailed distribution. It occurs, e.g., in the decay of atoms: at any time, the probability that a given fraction of atoms will decay is constant. Result: even after 1000s of years there's still some C^{14} left.

Density and Log-Likelihood



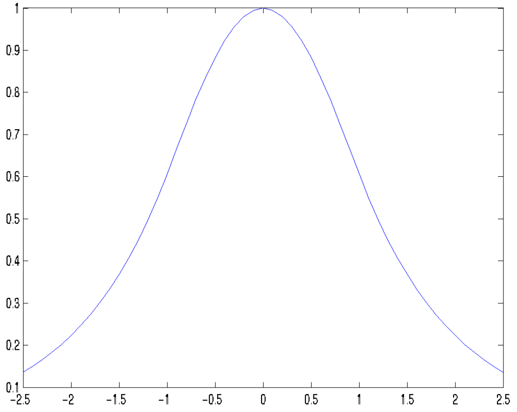
Huber's Robust Density

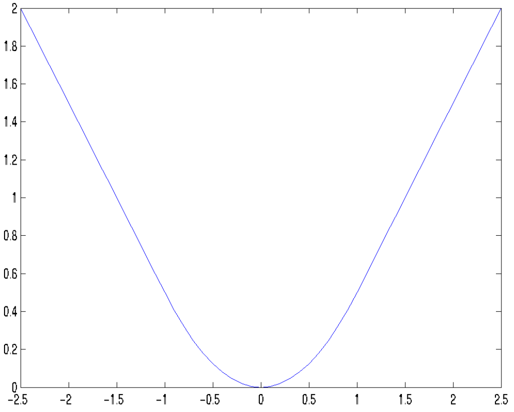
Problem: Sometimes we may not know what the additive density model of the likelihood is, in particular, how long-tailed the distribution may be.

Idea: Use the "worst" distribution as a reference. For distributions composed of a known (in our case Gaussian) part plus up to ε of an unknown part, we have the robust noise model

$$-\log p(\xi) = \begin{cases} \frac{1}{2\sigma}\xi^2 & \text{if } |\xi| \leq \sigma \\ |\xi| - \frac{\sigma}{2} & \text{otherwise} \end{cases}$$

Density and Log-Likelihood





Coordinate Transformation

Problem

Minimization in terms of $\mathbf{t}$, the latent variables, is expensive, since it involves dealing with $\log p(\mathbf{t}) \propto \mathbf{t}^\top K^{-1}\mathbf{t}$, for which every calculation costs a matrix inversion.

Idea

Variable substitution from $\mathbf{t}$ to $\mathbf{t} = K\alpha$, which leads to $\alpha^\top K\alpha$.

Posterior for α

For the likelihood term we need $y_i = \xi_i + [K\alpha]_i$, hence

$$p(\alpha|X,Y) \propto \left[\prod_{i=1}^{m} p\left(y_i - [K\alpha]_i\right)\right] |K|^{\frac{1}{2}} \exp\left(-\frac{1}{2}\alpha^\top K\alpha\right)$$

Now the posterior looks similar to one for a generalized linear model, where the functions $k(x_i,\cdot)$ are the terms into which we expand the estimate.

MAP Approximation

Well Known Problem

Integrals are expensive, so we need an approximation.

Well Known Solution

Compute the maximum of the posterior and assume a known parametric distribution around the maximum (typically we choose a normal distribution).

Result

$$\text{minimize } -\log p(\alpha|X,Y) = \sum_{i=1}^{m} -\log p(y_i - [K\alpha]_i) + \frac{1}{2}\alpha^\top K\alpha + \text{const.}.$$

Optimality Condition

$$K(c'(y_1 - [K\alpha]_1), \ldots, c'(y_m - [K\alpha]_m)) + K\alpha = 0$$

where $c(\xi) := -\log p(\xi)$. This looks very much like a loss function (see Bernhard's talk).

Connection to Support Vectors

Regularized Risk Functional

Here we minimize the loss on the training set, i.e.,

$$R_{\text{emp}}[f, X, Y] := \sum_{i=1}^{m} c(x_i, y_i, f(x_i))$$

plus a regularization term $\lambda\Omega[f]$, which typically is chosen to be $\Omega[f] = \frac{1}{2}\|f\|_{\mathcal{H}}^2$. In summary, we minimize

$$R_{\text{reg}}[f, X, Y] = R_{\text{emp}}[f, X, Y] + \lambda\Omega[f] = \sum_{i=1}^{m} c(x_i, y_i, f(x_i)) + \frac{\lambda}{2}\|f\|_{\mathcal{H}}^2$$

Empirical Risk — Log-Likelihood

Match up $-\log p(y_i|x_i, t_i)$ and $c(x_i, y_i, f(x_i))$, e.g., squared loss $\frac{1}{2}(y_i - t_i)^2$.

Regularization — Prior

Match up $-\log p(\alpha) = \frac{1}{2}\alpha^\top K\alpha + \text{cont.}$ and $\Omega[f] = \frac{1}{2}\|f\|_{\mathcal{H}} = \frac{1}{2}\alpha^\top K\alpha$.

Scaling Problems

Storage

We have to store the covariance matrix $K \in \mathbb{R}^{m \times m}$. On workstations this becomes a problem for $m > 10^4$.

Prediction

We have to sum up up to m kernel functions $k(x_i, x)$ to predict at x (covariances between training data and new test point). This becomes a problem for $m > 10^5$.

Training

Typically training involves at least one factorization of a matrix of size K. This is usually of order $O(m^3)$. On workstations we get problems if $m > 10^4$.

Solution

Approximate K by an object of lower rank. More on this later.