Bayesian Kernel Methods

Alexander J. Smola
Machine Learning Group, RSISE
The Australian National University
Canberra, ACT 0200
Alex.Smola@anu.edu.au

Slides available at http://mlg.anu.edu.au/~smola/icml2002/

THE AUSTRALIAN
NATIONAL UNIVERSITY

Overview

THE AUSTRALIAN NATIONAL UNIVERSITY

- Correlated Observations
- Inference with Gaussian Processes
- An Extended Dependence Model
- Examples and Algorithms
  - ⋆ Regression with Normal Noise (Sparse Greedy Gaussian Processes)
  - ⋆ Approximate Solution (Newton's Method)
  - ⋆ Transduction (EM-Algorithm)
- Summary of Gaussian Processes
- A Crash Course on Support Vectors
- The Big Picture

A Simple Idea ...

Pairs of Observations:

For some x_1, x_2 we observe values y_1, y_2, e.g., temperature on consecutive days.

Goal:

Given a new y_1, can we guess y_2?

Idea:

Exploit the correlation between y_1 and y_2.

Simple Assumption:

Assume that (y_1, y_2) are drawn from a normal distribution with mean μ and covariance K.

Insight:

If we know mean and covariance, we can predict y_2 from y_1.

Inference with Normal Distributions

THE AUSTRALIAN NATIONAL UNIVERSITY

Goal

After observing $\mathbf{y} := (y_1, \ldots, y_m)$ we would like to infer the distribution of y at locations $x'_1, \ldots, x'_m$, i.e., we would like to infer about $\mathbf{y}' := (y(x'_1), \ldots, y(x'_m))$.

Conditional Density

We know that $p(\mathbf{y}, \mathbf{y}') = p(\mathbf{y}'|\mathbf{y})p(\mathbf{y})$ and therefore

$$p(\mathbf{y}'|\mathbf{y}) = \frac{p(\mathbf{y}, \mathbf{y}')}{p(\mathbf{y})} \propto p(\mathbf{y}_{\text{fixed}}, \mathbf{y}').$$

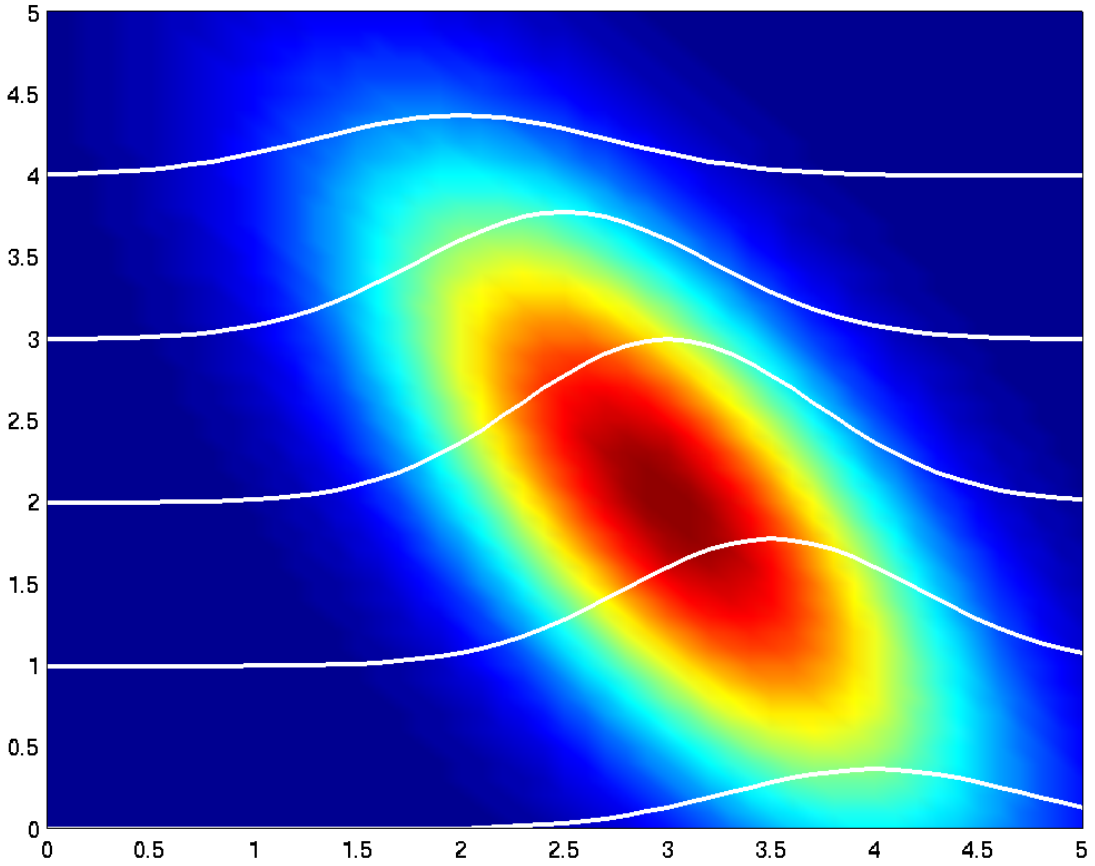
Lazy Trick

For normal distributions we only need to compute **mean** and **covariance** to determine the density completely (including normalization factors).

Recipe: collect all terms from $p(\mathbf{y}, \mathbf{y}')$ dependent on $\mathbf{y}'$ and ignore the rest.

Predicting y_2 from y_1

THE AUSTRALIAN NATIONAL UNIVERSITY



A Closer Look at the Normal Distribution

THE AUSTRALIAN NATIONAL UNIVERSITY

Density

$$p(\mathbf{y}) = \frac{1}{(2\pi)^{\frac{m}{2}}|K|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{y}-\mu)^\top K^{-1}(\mathbf{y}-\mu)\right).$$

So, if we split $\mathbf{y}$ into $\mathbf{y}$ and $\mathbf{y}'$, we have

$$p(\mathbf{y},\mathbf{y}') \propto \exp\left(-\frac{1}{2}\left(\begin{bmatrix}\mathbf{y}\\ \mathbf{y}'\end{bmatrix} - \begin{bmatrix}\mu\\ \mu'\end{bmatrix}\right)^\top \begin{bmatrix}K_{\mathbf{yy}} & K_{\mathbf{yy}'}\\ K_{\mathbf{y}'\mathbf{y}} & K_{\mathbf{y}'\mathbf{y}'}\end{bmatrix}^{-1} \left(\begin{bmatrix}\mathbf{y}\\ \mathbf{y}'\end{bmatrix} - \begin{bmatrix}\mu\\ \mu'\end{bmatrix}\right)\right)$$

Predicting $\mathbf{y}'$ from $\mathbf{y}$ (it's all just linear algebra)

Compute lower right part of inverse covariance matrix: this gives us the resulting variance.

Compute linear terms in $\mathbf{y}'$ to get the resulting mean.

Prediction: The Gory Details

THE AUSTRALIAN NATIONAL UNIVERSITY

Inverting the Covariance Matrix

$$\begin{bmatrix} K_{\mathbf{yy}} & K_{\mathbf{yy'}} \\ K_{\mathbf{yy'}}^\top & K_{\mathbf{y'y'}} \end{bmatrix}^{-1} = \begin{bmatrix} {K_{\mathbf{yy}}}^{-1} - \left({K_{\mathbf{yy}}}^{-1} K_{\mathbf{yy'}}^\top\right)^\top \chi^{-1} \left({K_{\mathbf{yy}}}^{-1} K_{\mathbf{yy'}}^\top\right) & -\left({K_{\mathbf{yy}}}^{-1} K_{\mathbf{yy'}}^\top\right) \chi^{-1} \\ -\chi^{-1} \left({K_{\mathbf{yy}}}^{-1} K_{\mathbf{yy'}}^\top\right)^\top & \chi^{-1} \end{bmatrix}$$

where $\chi = K_{\mathbf{y'y'}} - K_{\mathbf{yy'}}^\top {K_{\mathbf{yy}}}^{-1} K_{\mathbf{yy'}}$ (Schur complement).

Reduced Covariance

From the inverse of the covariance matrix we obtain that the only quadratic part in $\mathbf{y}'$ is given by χ. Thus the **variance in $\mathbf{y}'$ is reduced** from $K_{\mathbf{y'y'}}$ to $K_{\mathbf{y'y'}} - K_{\mathbf{yy'}}^\top {K_{\mathbf{yy}}}^{-1} K_{\mathbf{yy'}}$ by observing $\mathbf{y}$.

Predictive Mean

Instead of μ' the mean is shifted to $\mu' + K_{\mathbf{yy'}}^\top K_{\mathbf{yy}}^{-1}(\mathbf{y} - \mu)$.

Gaussian Process

THE AUSTRALIAN NATIONAL UNIVERSITY

Problem

What to do if do not know K and μ beforehand?

Simple Solution

We simply **assume** to know the covariance K and mean μ, based on prior knowledge. Simplifying assumption: $\mu = 0$.

Gaussian Process

A stochastic process, where any set of $y(x_1), \ldots, y(x_m)$ is normally distributed, is a Gaussian Process.

Covariance Function

We denote by $k(x, x')$ the function generating the covariance matrix, i.e., $k(x, x') = \mathrm{Cov}(y(x), y(x'))$.

Some Covariance Functions

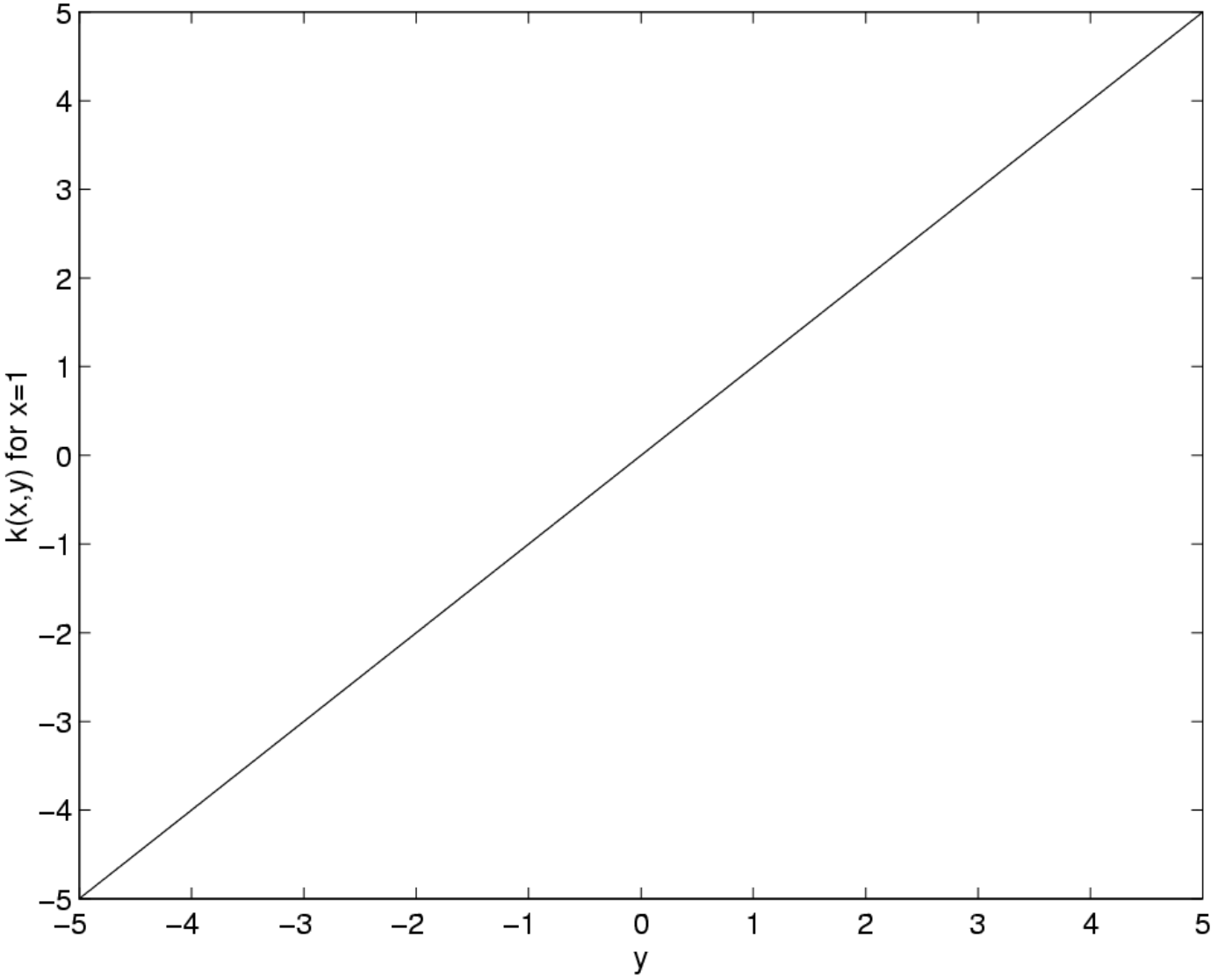
THE AUSTRALIAN NATIONAL UNIVERSITY

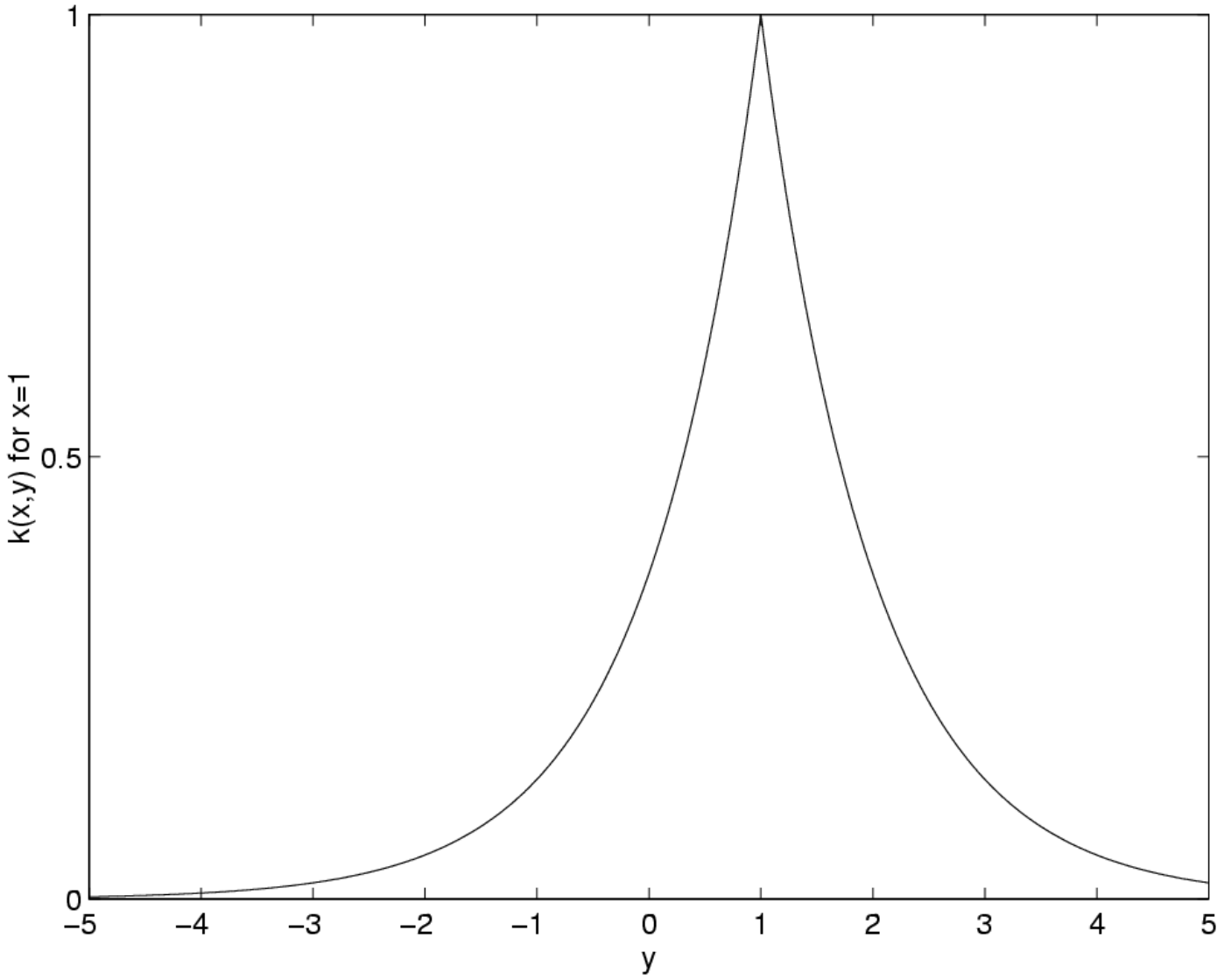
Idea

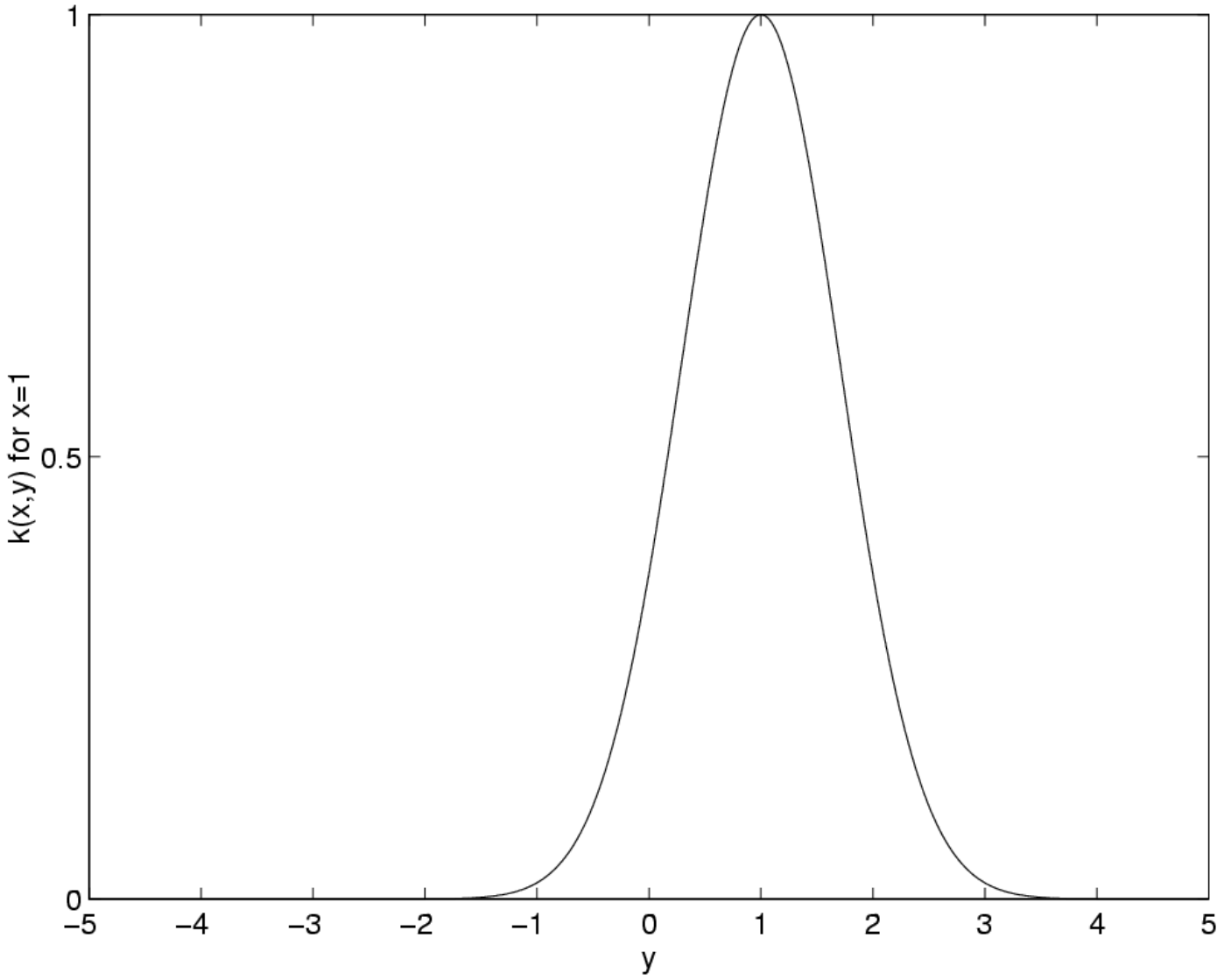
Any function k leading to a symmetric matrix with nonnegative eigenvalues is a valid covariance function.

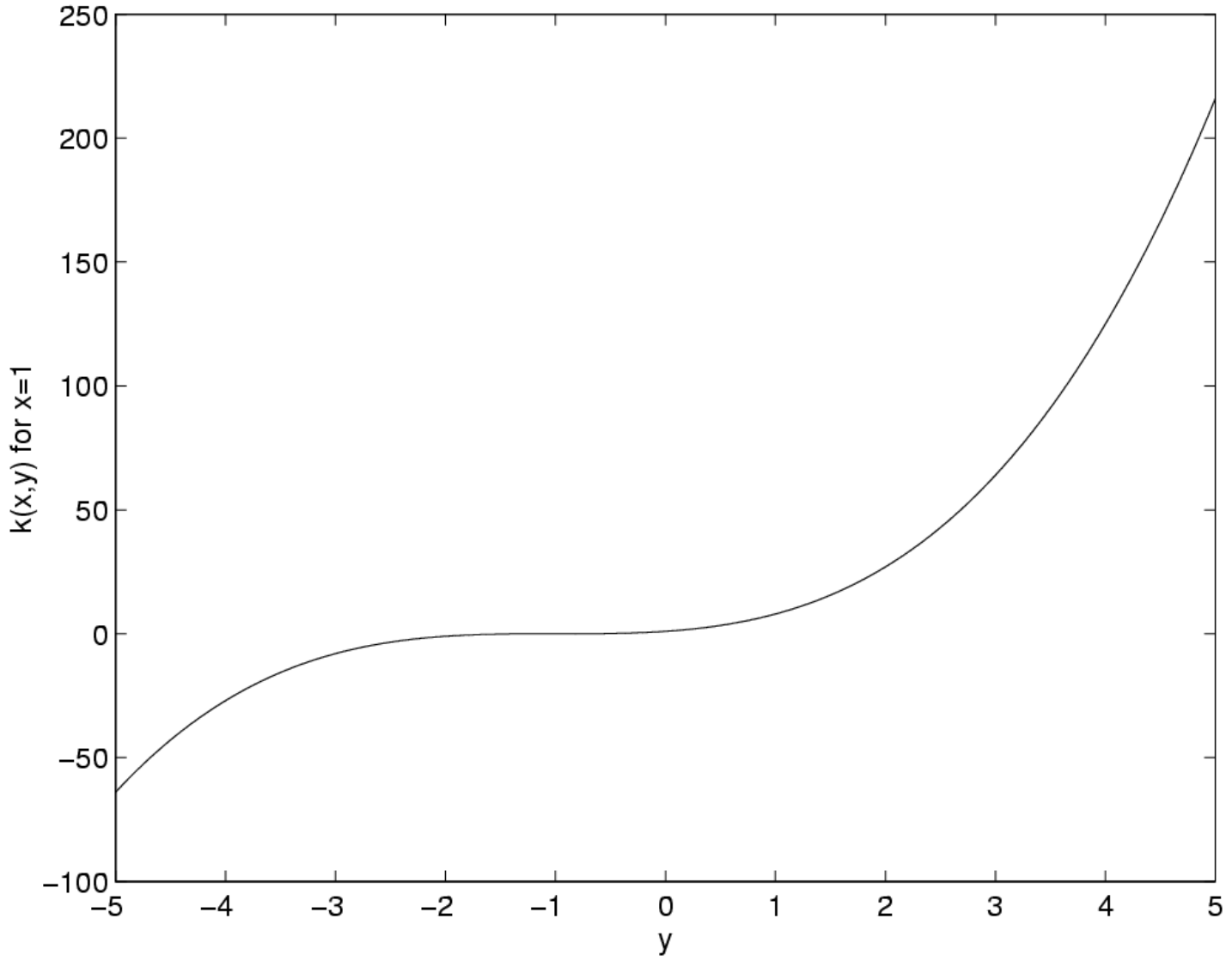
Examples

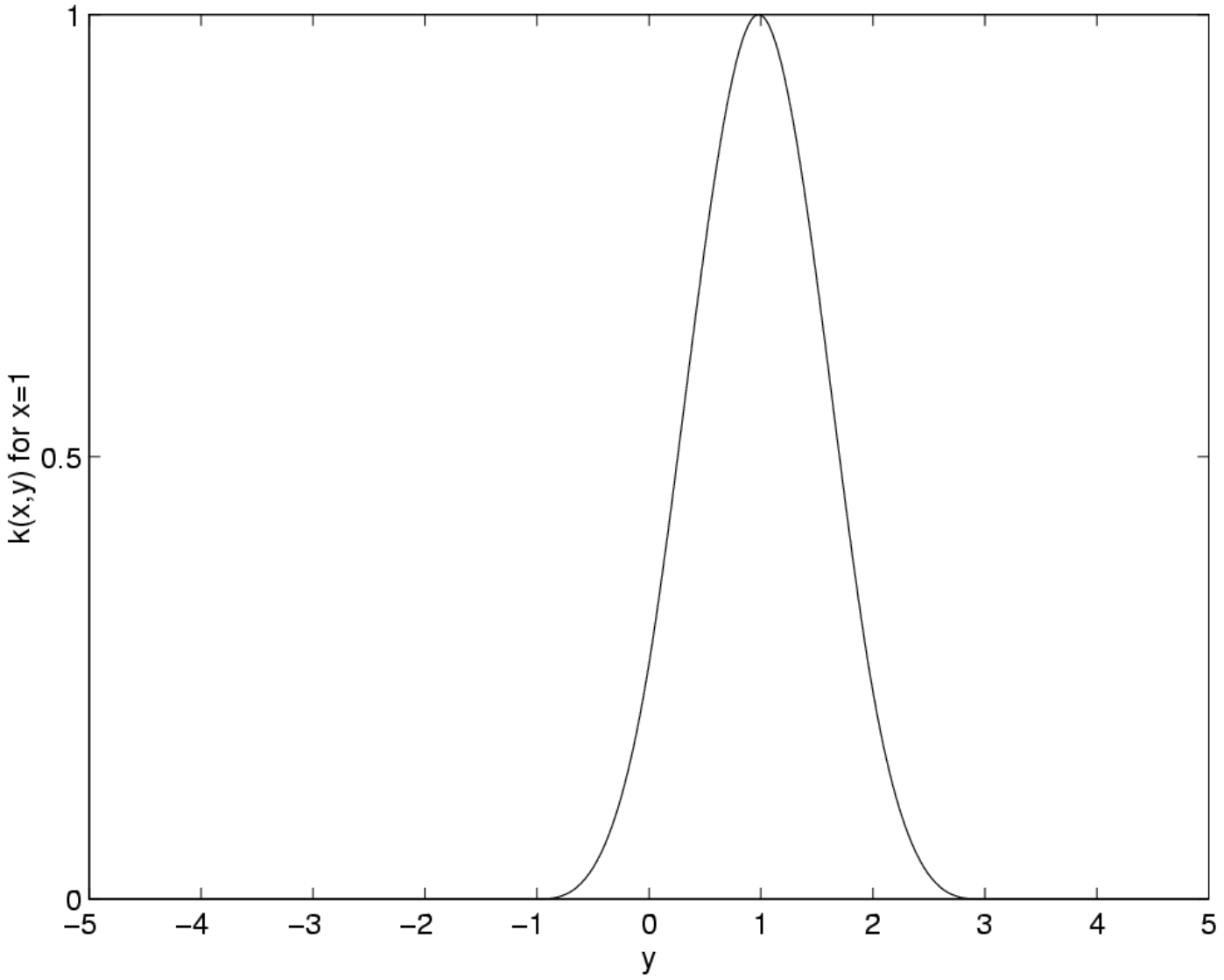
$$
\begin{aligned}
k(x, x') &= \langle x, x' \rangle \text{ Linear Kernel} \\
k(x, x') &= \exp\left(-\frac{1}{2\sigma}\|x - x'\|\right) \text{ Laplacian Kernel} \\
k(x, x') &= \exp\left(-\frac{1}{2\sigma^2}\|x - x'\|^2\right) \text{ Gaussian RBF Kernel} \\
k(x, x') &= \left(\langle x, x' \rangle + c\right)^d \text{ with } c \geq 0,\ d \in \mathbb{N} \text{ Polynomial Kernel} \\
k(x, x') &= B_{2n+1}(x - x') \text{ Spline kernel} \\
k(x, x') &= \mathbf{E}_c[p(x|c)p(x'|c)] \text{ Conditional Expectation Kernel}
\end{aligned}
$$


k(x,y) for x=1
y
5
4
3
2
1
0
−1
−2
−3
−4
−5
−5
−4
−3
−2
−1
0
1
2
3
4
5


1
0.5
0
k(x,y) for x=1
−5
−4
−3
−2
−1
0
1
2
3
4
5
y


1
0.5
0
k(x,y) for x=1
−5
−4
−3
−2
−1
0
1
2
3
4
5
y


250
200
150
100
50
0
−50
−100
k(x,y) for x=1
−5
−4
−3
−2
−1
0
1
2
3
4
5
y


1
0.5
0
k(x,y) for x=1
−5
−4
−3
−2
−1
0
1
2
3
4
5
y

A Simple Example: Linear Kernel

THE AUSTRALIAN NATIONAL UNIVERSITY

Covariance Function

Assume that $\mathrm{Cov}(y(x), y(x')) = \langle x, x' \rangle$ with $x \in \mathbb{R}^n$, i.e., that we have an n-dimensional Normal distribution, where the covariance between observations is a bilinear function of x and x' and furthermore that $\mu = 0$.

Density for y

$$p(\mathbf{y}) = (2\pi)^{-\frac{n}{2}} \left(\det X^\top X\right)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\mathbf{y}^\top (XX^\top)^* \mathbf{y}\right)$$

where $X = (\mathbf{x}_1, \ldots, \mathbf{x}_m)$ and $(XX^\top)^*$ is the pseudoinverse of $XX^\top$.

Prediction

If we predict $\mathbf{y}'$ on X' we know that the variance is given by

$$k_{\mathbf{y}'\mathbf{y}'} - K_{\mathbf{y}\mathbf{y}'}^\top {K_{\mathbf{y}\mathbf{y}}}^{-1} K_{\mathbf{y}\mathbf{y}'} = X'^\top X' - X'^\top X (X^\top X)^{-1} X^\top X' = X'^\top (\mathbf{1} - P_X) X'.$$

And we can predict the mean of $\mathbf{y}'$ via $K_{\mathbf{y}\mathbf{y}'}^\top K_{\mathbf{y}\mathbf{y}}^{-1} \mathbf{y} = X'^\top X (X^\top X)^{-1} \mathbf{y} = X'\alpha$.

Hence μ' is a **linear function of** X'.

Consequences

More Observations than Dimensions

Here the predictive variance is zero, since it is given by $\mu' = X'^\top(\mathbf{1} - P_X)X'$, X spans $\mathbb{R}^n$, and hence we have $\mathbf{1} - P_X = 0$.

The mean can be found as $X'^\top X(X^\top X)^{-1}\mathbf{y}$. This means that $\mathbf{y}$ lives in an n-dimensional subspace.

Strange Result

After observing n data pairs we can predict with certainty.

Problem

We cannot cope with $\mathbf{y}$ that do not live in an n-dimensional subspace, spanned by $XX^\top$.

Idea

What if we did not observe $\mathbf{y}$ directly but rather $y_i = t_i + \xi$, where ξ is some additional random variable.

A More Sophisticated Model

Indirect Observations

$$X \longrightarrow \mathbf{t} \longrightarrow \mathbf{y}$$

- $\mathbf{t}$ is drawn from a normal distribution with covariance K
- $\mathbf{y}$ is conditionally independent of X, that is $p(\mathbf{y}|X, \mathbf{t}) = p(\mathbf{y}|\mathbf{t})$.

Effective Density: Integrating out t

$$p(\mathbf{y}|X) = \int p(\mathbf{y}|\mathbf{t})p(\mathbf{t}|X)d\mathbf{t}$$

Three Practical Solutions

- $p(\mathbf{y}|X)$ can be computed explicitly and it is "nice". Then we can use $p(\mathbf{y}|X)$ directly for prediction. Example: normal distribution.
- We cannot compute the integral, so we could maximize $p(\mathbf{y}, \mathbf{t}|X) = p(\mathbf{y}|\mathbf{t})p(\mathbf{t}|X)$ over $\mathbf{t}_{\text{train}}, \mathbf{t}_{\text{test}}$ and $\mathbf{y}_{\text{test}}$.
- We can approximate the integral by Markov-Chain Monte-Carlo.

Regression with Normal Noise

THE AUSTRALIAN NATIONAL UNIVERSITY

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 $\mathbf{y} \sim \mathcal{N}(0, K + \sigma^2 \mathbf{1})$.

Explicit Solution: Posterior Density

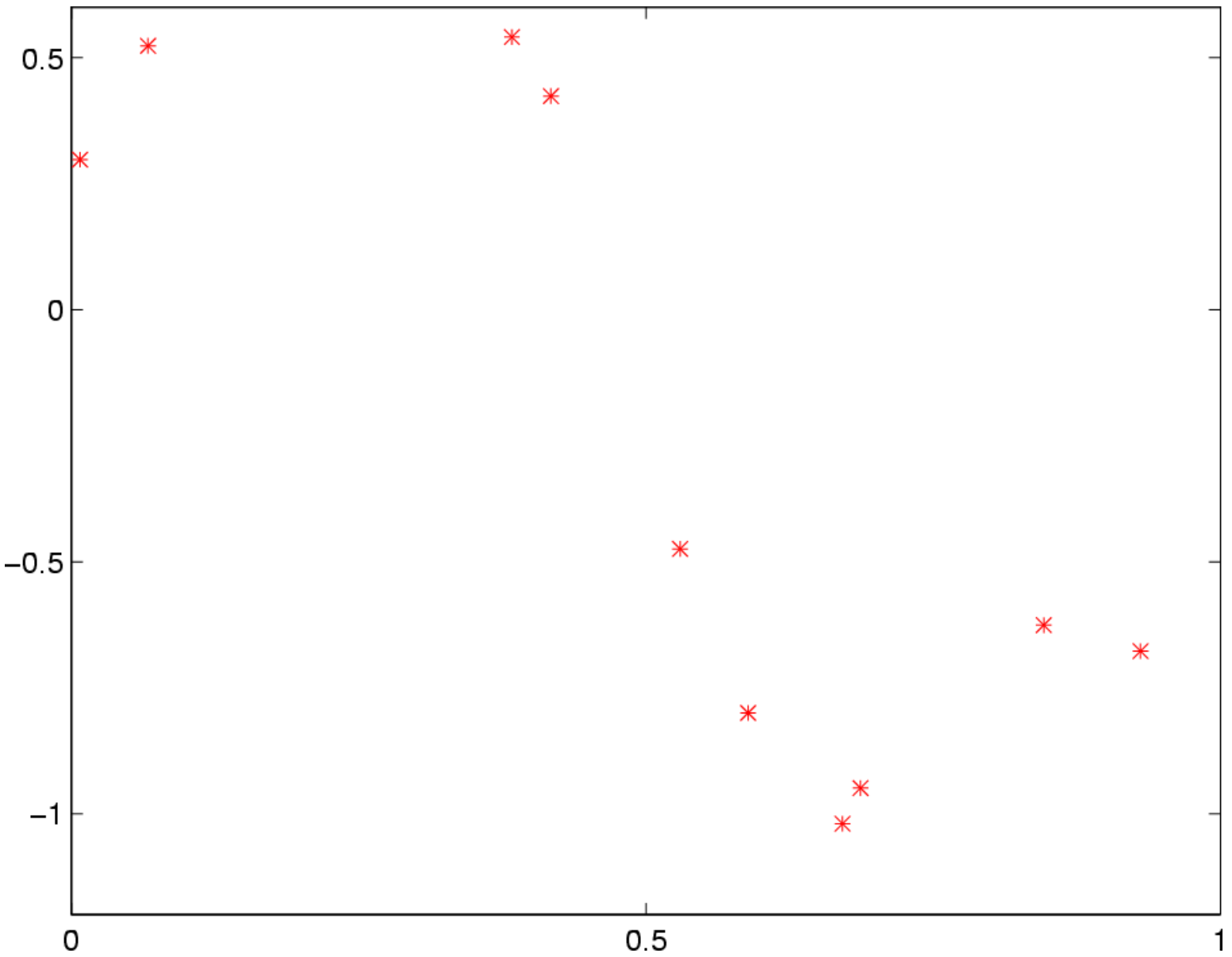
$$p(\mathbf{y}|X) = (2\pi)^{-\frac{n}{2}} (\det(K + \sigma^2 \mathbf{1}))^{-\frac{1}{2}} \exp\left(-\frac{1}{2} \mathbf{y}^\top (K + \sigma^2 \mathbf{1})^{-1} \mathbf{y}\right)$$

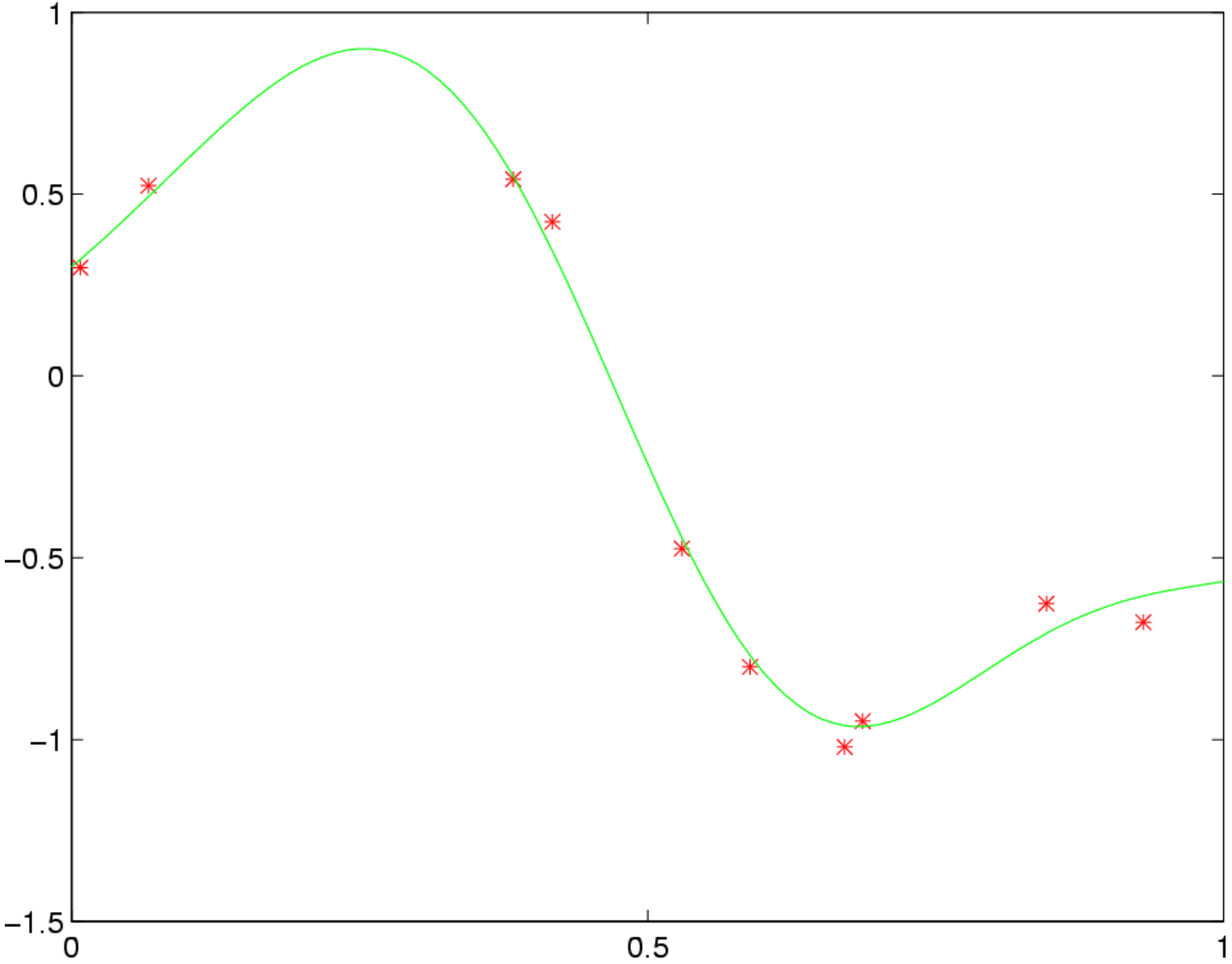
Note that the problem of non-invertibility of the covariance matrix disappeared (similar to regularization to improve the condition of a matrix).

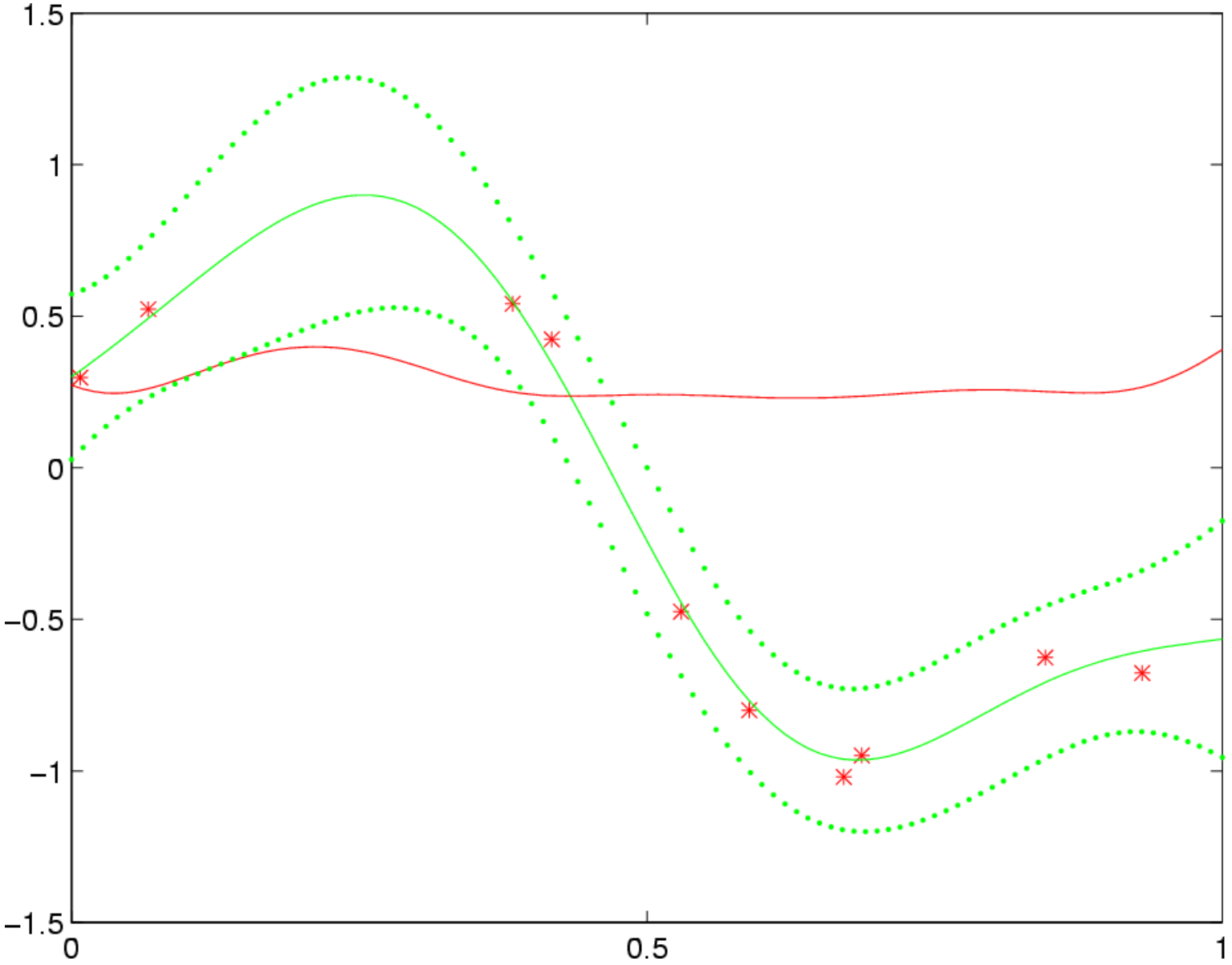
Inference

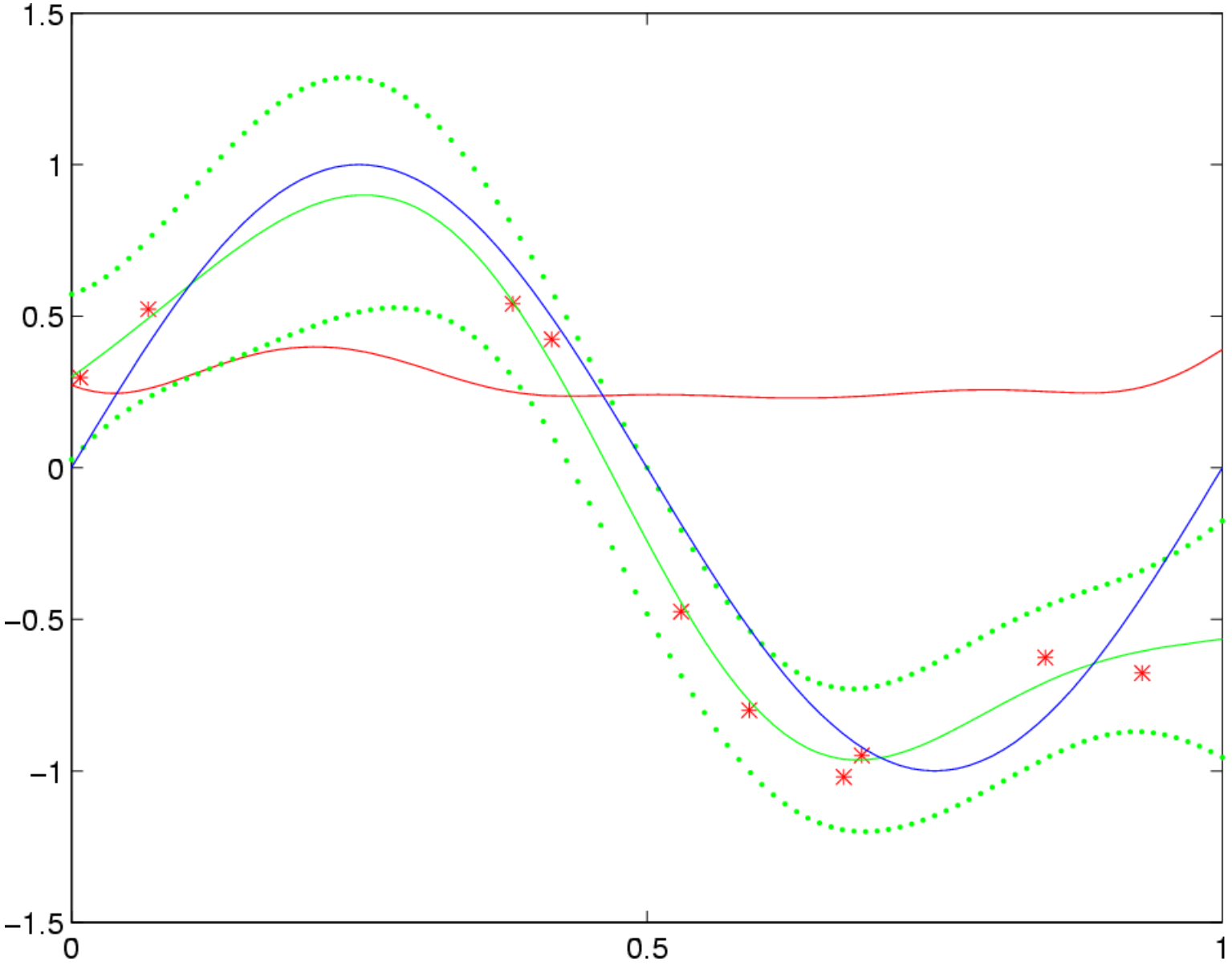
We can simply re-use the results from inference without noise and obtain (for inferring $\mathbf{y}'$ after observing $\mathbf{y}, X, X'$): $\mathbf{y}' \sim \mathcal{N}(\mu_y, \Sigma_y)$where

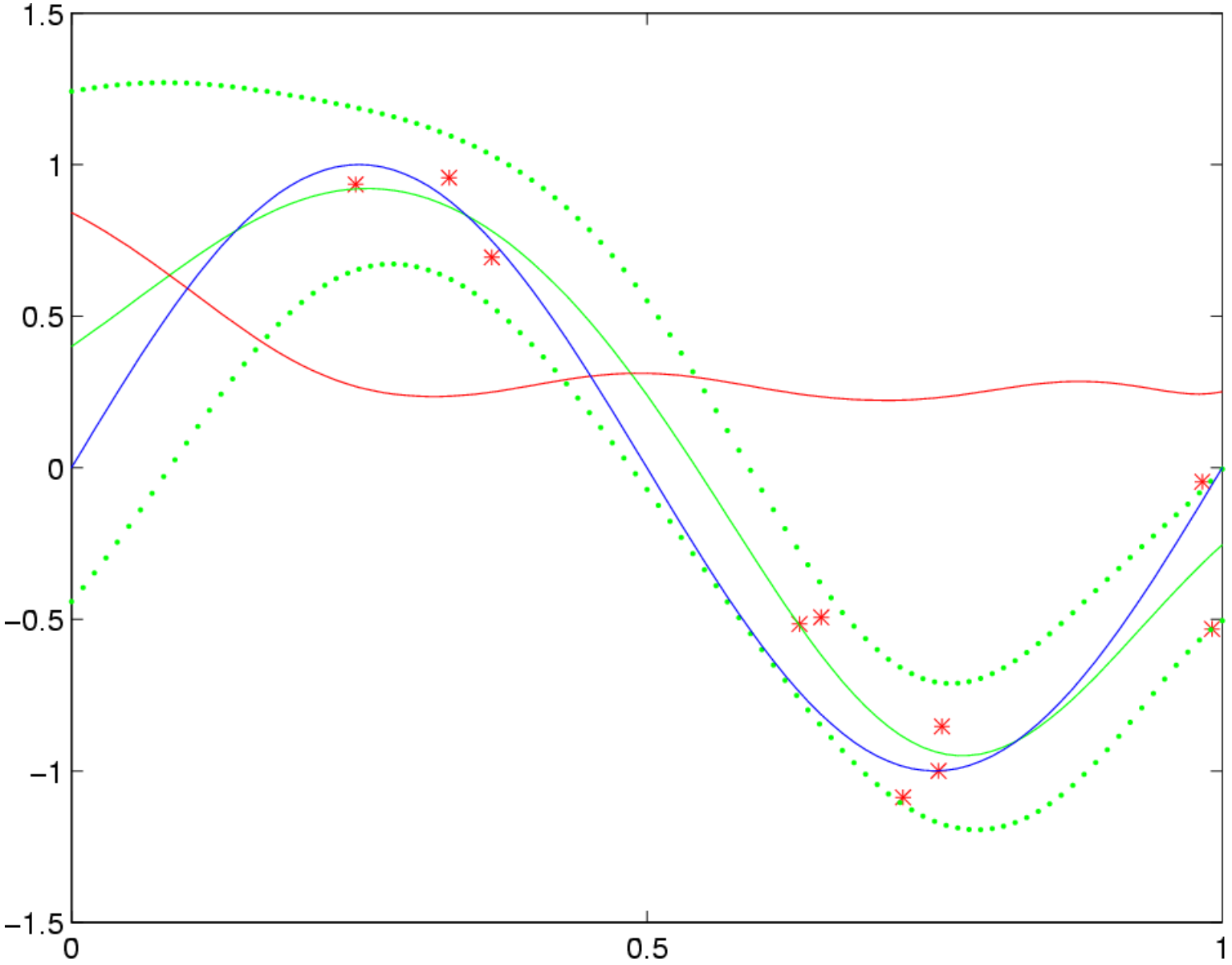
$$\mu_y = K_{\mathbf{tt}'}^\top (K_{\mathbf{tt}} + \sigma^2 \mathbf{1})^{-1} \mathbf{y} \text{ and } \Sigma_y = K_{\mathbf{t}'\mathbf{t}'} + \sigma^2 \mathbf{1} - K_{\mathbf{tt}'}^\top (K_{\mathbf{tt}} + \sigma^2 \mathbf{1})^{-1} K_{\mathbf{tt}'}$$


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

Maximization of $p(\mathbf{y}|X)$

THE AUSTRALIAN NATIONAL UNIVERSITY

General Idea

Rather than maximizing $p(\mathbf{y}|X) = \int p(\mathbf{y}|\mathbf{t})p(\mathbf{t}|X)d\mathbf{t}$ with respect to $\mathbf{y}$, we take the mode of the integrand and maximize $p(\mathbf{y}|\mathbf{t})p(\mathbf{t}|X)$ directly with respect to $\mathbf{y}, \mathbf{t}$.

This is a Maximum A Posteriori (MAP) approximation.

Strategy

Since we know $\mathbf{y}_{\text{train}}$, we only need to maximize over $\mathbf{t}_{\text{train}}$, $\mathbf{t}_{\text{test}}$ and $\mathbf{y}_{\text{test}}$. This means that we maximize

$$p(\mathbf{y}_{\text{train}}|\mathbf{t}_{\text{train}})p(\mathbf{y}_{\text{test}}|\mathbf{t}_{\text{test}})p(\mathbf{t}_{\text{train}}, \mathbf{t}_{\text{test}}|X)$$

Special Case

For fixed $\mathbf{t}$, the optimal $\mathbf{y}$ is given by $\mathbf{y} = \mathbf{t}$, hence we only need to maximize $p(\mathbf{y}_{\text{train}}|\mathbf{t}_{\text{train}})p(\mathbf{t}_{\text{train}}, \mathbf{t}_{\text{test}}|X)$.

Since $p(\mathbf{t}_{\text{train}}, \mathbf{t}_{\text{test}}|X)$ is a normal distribution, we can always predict $\mathbf{t}_{\text{test}} = \mu_{\text{test}}$, which makes it independent of the $\mathbf{t}_{\text{test}}$.

Maximization of $p(\mathbf{y}|X)$, Part II

THE AUSTRALIAN NATIONAL UNIVERSITY

Optimization Problem

In the special case that $p(\mathbf{t}|\mathbf{t})$ is the maximizer of $p(\mathbf{y}|\mathbf{t})$ we can perform inference as follows:

1. Maximize $p(\mathbf{y}_{\text{train}}|\mathbf{t}_{\text{train}})p(\mathbf{t}_{\text{train}}|X)$ with respect to $\mathbf{t}_{\text{train}}$. This is equivalent to maximizing
$$\sum_{i=1}^{m} \log p(y_i|t_i) - \frac{1}{2}\mathbf{t}^\top K_{\mathbf{tt}}^{-1}\mathbf{t}$$
2. Find mode of $p(\mathbf{t}_{\text{test}}|\mathbf{t}_{\text{train}}, X)$. For $\mu = 0$ this can be found at
$$K_{\mathbf{tt}'}^\top K_{\mathbf{tt}}^{-1}\mathbf{t}_{\text{train}} = K_{\mathbf{tt}'}^\top \alpha$$
3. Predict $\mathbf{y}_{\text{test}} = \mathbf{t}_{\text{test}}$.

Direct Application

Regression, such as, $p(y|t) \propto \exp(-\lambda|y-t|)$ (Laplacian Noise)

One more Approximation (mainly classification)

Even if $p(\mathbf{t}|\mathbf{t})$ is not the maximizer of $p(\mathbf{y}|\mathbf{t})$, we can use the method above ...

Maximization of $p(\mathbf{y}|X)$, Part III

THE AUSTRALIAN NATIONAL UNIVERSITY

Practical Solution

The step to compute $\mathbf{t}_{\text{train}}$ is the most expensive bit, since we have to maximize

$$\sum_{i=1}^{m} \log p(y_i|t_i) - \frac{1}{2}\mathbf{t}^\top K_{\mathbf{tt}}^{-1}\mathbf{t}$$

For convenience we reparametrize $\mathbf{t} = K\alpha$ and minimize

$$R(\alpha) := \sum_{i=1}^{m} -\log p(y_i|[K\alpha]_i) + \frac{1}{2}\alpha^\top K\alpha$$

Special Case

The likelihood terms $-\log(y_i|t_i)$ are convex in t_i. This means that

- There exists one global minimum wrt. $\mathbf{t}$.
- We can use convex optimization, e.g., the Newton method.

$$\alpha \to \alpha - \left(\partial_\alpha^2 R(\alpha)\right)^{-1} \partial_\alpha R(\alpha)$$

Alternatives are Conjugate Gradient Descent or sparse greedy methods.

Examples of $p(y|t)$: Laplacian Noise

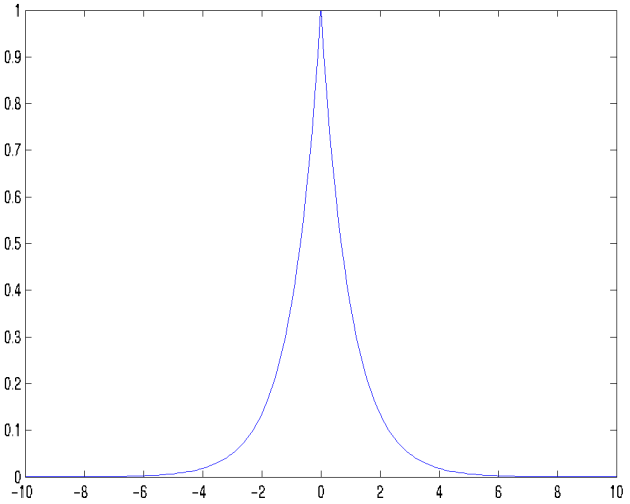
THE AUSTRALIAN NATIONAL UNIVERSITY

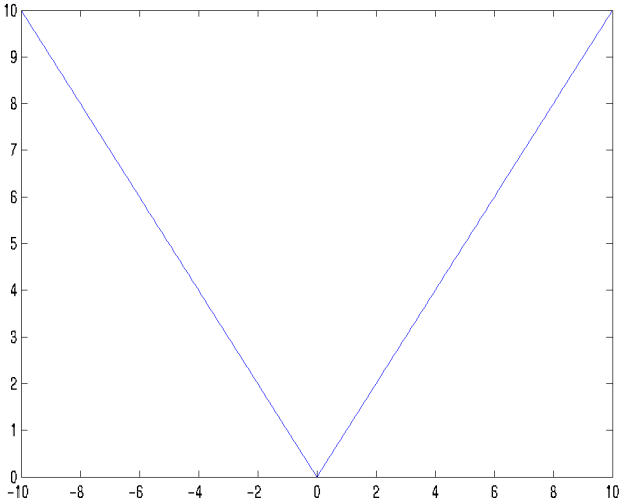
Noise Model

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

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





Examples of $p(y|t)$: Huber's Density

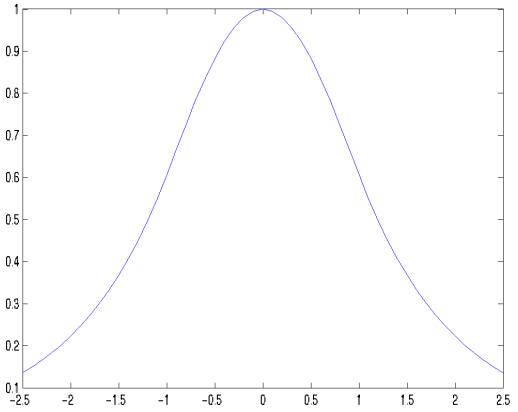
THE AUSTRALIAN NATIONAL UNIVERSITY

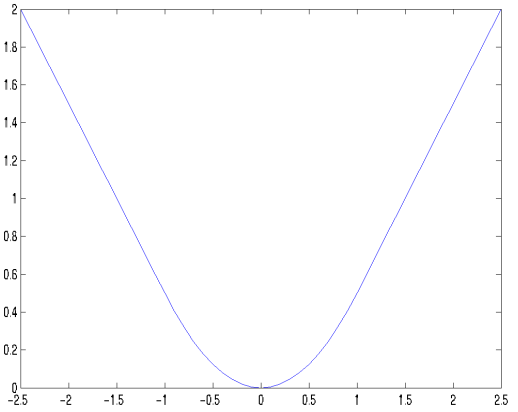
Problem: Sometimes we may not know what the additive density model of the likelihood is, in particular, how long-tailed the distribution may be.

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

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

Density and Log-Likelihood





Logistic Regression

THE AUSTRALIAN NATIONAL UNIVERSITY

Basic Idea

For classification purposes we are mainly interested in the ratio between $p(y = 1|t)$ and $p(y = -1|t)$, since this tells us the Bayes optimal classifier (i.e., the classifier with minimal error).

Making the Problem Symmetric

Estimating $\frac{p(y=1|t)}{p(y=-1|t)}$ would help us find a classifier, but it isn't symmetric with respect to y. So we attempt to find t with

$$t(x) = \log \frac{p(y = 1|t)}{p(y = -1|t)} \Rightarrow p(y = 1|t) = \frac{1}{1 + \exp(-t)}.$$

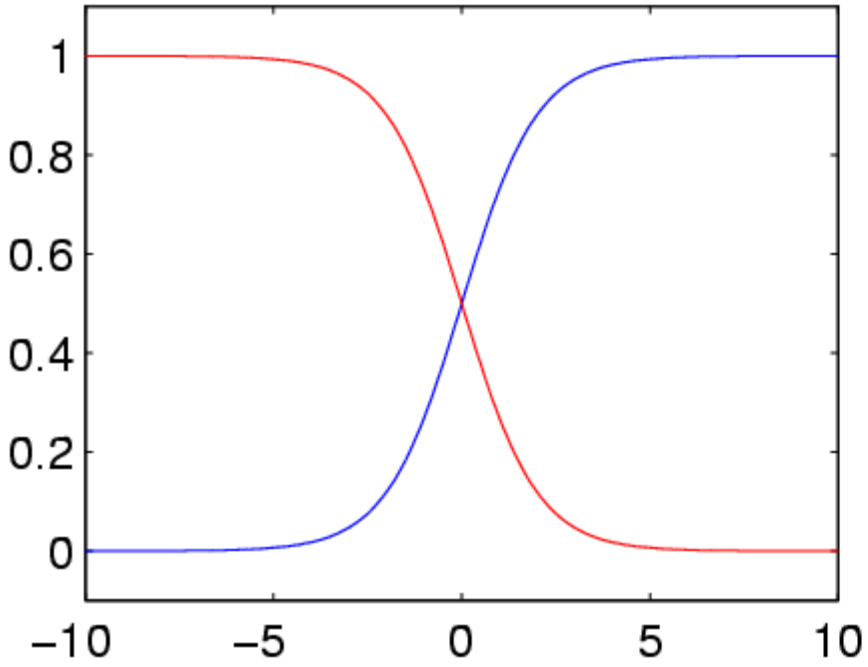
This is equivalent to assuming that $p(y = 1) \propto \exp(t)$ (after normalization).

Likelihood

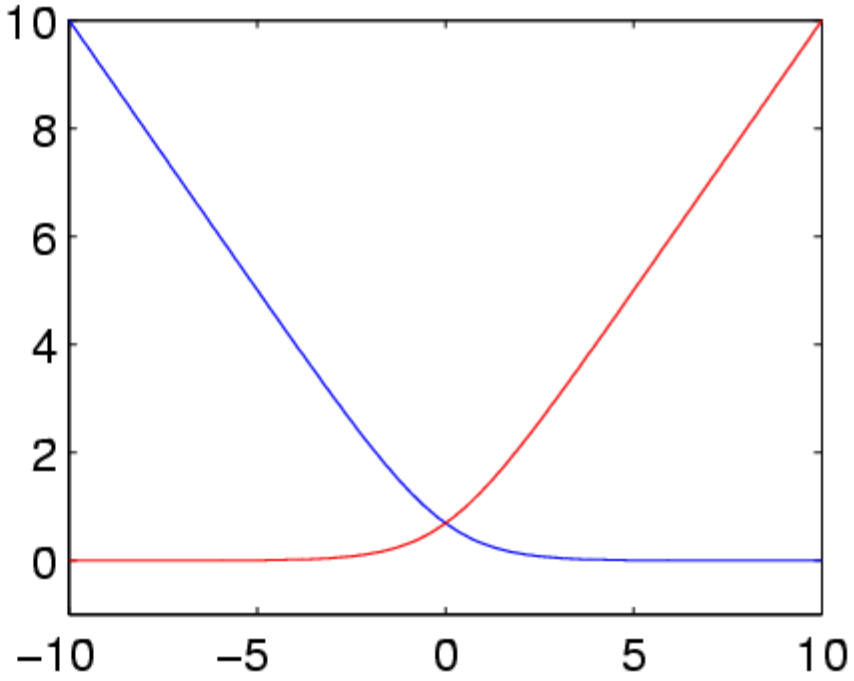
For the likelihood we obtain

$$p(\mathbf{y}|\mathbf{t}) = \prod_{i=1}^{m} \frac{1}{1 + \exp(-y_i t_i)} \Rightarrow -\log p(Y|X, f) = \sum_{i=1}^{m} \log(1 + \exp(-y_i t_i)).$$

Logistic Regression



Probability $p(y|t)$



Negative Log-Likelihood $-\log p(y|t)$

Multiclass Logistic Regression

THE AUSTRALIAN NATIONAL UNIVERSITY

Observation

We may write $p(y|x,t)$ as follows

$$p(y=1|t) = \frac{\exp(\frac{1}{2}t)}{\exp(\frac{1}{2}t)+\exp(-\frac{1}{2}t)}$$

$$p(y=-1|t) = \frac{\exp(-\frac{1}{2}t)}{\exp(\frac{1}{2}t)+\exp(-\frac{1}{2}t)}$$

Idea

For more than two classes, estimate one t_j per class and compute probabilities $p(y|t)$ via

$$p(y|t) = \frac{\exp(t_j)}{\sum_{i=1}^{N}\exp(t_i)}$$

Putting it Together

$$p(\mathbf{y},\mathbf{t}|X) = \prod_{i=1}^{m}\frac{\exp(t_{i,y_i})}{\sum_{j=1}^{N}\exp(t_{i,j})}p(\mathbf{t}|X)$$

Label Noise

Basic Idea

We want to perform classification in the presence of random label noise (in addition to the noise model $p_0(y|t)$ discussed previously).

Here, a label is randomly *assigned* to observations with probability 2η (note that this is the same as randomly *flipping* with probability η). We then write

$$p(y|t) = \eta + (1 - 2\eta)p_0(y|t).$$

Consequence

The influence of $p_0(y|t)$ on the posterior is descreased, hence η has a "regularizing" effect on the estimate.

Discriminant Analysis

Basic Idea

Assume that the classes to be separated (we assume $N = 2$ for simplicity) correspond to **Normal distributions** in some space, and that $f(x)$ are **projections** from this space onto a line.

Result

Projections on a real line yield normal distributions. Hence we can model the probability $p(y|t)$ by

$$p(y|t) \propto \exp\left(-\frac{1}{2}(y-t)^2\right).$$

Algorithmic Result

This is essentially **regression on the labels**, which can be done very cheaply.

Problem: often the assumption of a normal distribution is not so well satisfied.

What we really wanted to do ...

Indirect Observations

$$X \longrightarrow \mathbf{t} \longrightarrow \mathbf{y}$$

- $\mathbf{t}$ is drawn from a normal distribution with covariance K
- $\mathbf{y}$ is conditionally independent of X, that is $p(\mathbf{y}|X, \mathbf{t}) = p(\mathbf{y}|\mathbf{t})$.

Effective Density: Integrating out t

$$p(\mathbf{y}|X) = \int p(\mathbf{y}|\mathbf{t})p(\mathbf{t}|X)d\mathbf{t}$$

A Practical Idea

We cannot compute the integral, so we could maximize $p(\mathbf{y}, \mathbf{t}|X) = p(\mathbf{y}|\mathbf{t})p(\mathbf{t}|X)$ over $\mathbf{t}_{\text{train}}, \mathbf{t}_{\text{test}}$ and $\mathbf{y}_{\text{test}}$.

But: we do not know $\mathbf{y}_{\text{test}}$ but want to maximize $p(\mathbf{y}, \mathbf{t}|X)$

EM-Algorithm

For a good guess of $\mathbf{y}_{\text{test}}$ compute $\mathbf{t}$ and vice versa.

Expectation-Maximization Algorithm ...

THE AUSTRALIAN NATIONAL UNIVERSITY

Expectation Step

For $\mathbf{t}_{\text{train}}, \mathbf{t}_{\text{test}}$ compute $p(\mathbf{y}_{\text{test}}|\mathbf{t}_{\text{test}})$. For the logistic regression, e.g., we have

$$p(y=1|t) = \frac{1}{1+\exp(-t)} \text{ and } p(y=-1|t) = \frac{1}{1+\exp(t)}.$$

Next compute the expected log-likelihood (denote $p(y_i = 1|t_i^{\text{old}}) = \pi_i$)

$$\begin{aligned} Q(\mathbf{t}) &:= \mathbf{E}_{\mathbf{y}_{\text{test}}}\left[-\log p(\mathbf{t}, \mathbf{y}_{\text{train}}, \mathbf{y}_{\text{test}}|X)\right] \\ &= \sum_{\text{test}} \pi_i \log p(y_i = 1|t_i) + (1-\pi_i)\log p(y_i = -1|t_i) \\ &\quad + \sum_{\text{train}} \log p(y_i|t_i) - \frac{1}{2}\mathbf{t}^\top K \mathbf{t}. \end{aligned}$$

Maximization Step

Maximize $Q(\mathbf{t})$ with respect to $\mathbf{t}$ (e.g., via Newton's Method).

Iterate until converged

Expectation-Maximization, Part II

THE AUSTRALIAN NATIONAL UNIVERSITY

Initialization

We start with the approximation obtained from the *labelled* data alone (i.e., we maximize $p(\mathbf{y}_{\text{train}}, \mathbf{t}_{\text{train}}|X)$ directly). This gives us a first guess for $\mathbf{y}_{\text{test}}$.

Caveat

The EM algorithm will only converge to a **local minimum**. Random initializations can help.

Also note that exact integration would be better (of course).

Side Effect

We can use lots of additional unlabelled data to improve our estimate (in SVM this is called **transduction**).

Gaussian Processes: Summing Up

THE AUSTRALIAN NATIONAL UNIVERSITY

Observations

We observe $\mathbf{y}$ which depends on $\mathbf{t}$ via $p(\mathbf{y}|\mathbf{t})$ (regression, classification). Furthermore $\mathbf{t}$ is distributed according to a Gaussian Process with covariance K (and zero mean). This yields

$$p(\mathbf{y}|X) = \int p(\mathbf{y}|\mathbf{t})p(\mathbf{t})d\mathbf{t}$$

Often $p(\mathbf{t})$ is called a **prior** on $\mathbf{t}$ (since we don't know $\mathbf{t}$).

Kernels

The covariance function for $\mathbf{t}$ is $k(\mathbf{x}, \mathbf{x}')$, that is $\mathrm{Cov}[t(\mathbf{x}), t(\mathbf{x}')] = k(\mathbf{x}, \mathbf{x}')$.

Approximations

If we can compute $p(\mathbf{y}|X)$ everything is fine.

Otherwise maximize $p(\mathbf{y}, \mathbf{t}|X) = p(\mathbf{y}|\mathbf{t})p(\mathbf{t}|X)$, either directly via the EM-algorithm, or (in yet another approximation) maximize $p(\mathbf{y}_{\text{train}}, \mathbf{t}_{\text{train}}|X)$, subsequently $p(\mathbf{t}_{\text{train}}, \mathbf{t}_{\text{test}}|X)$, and finally $p(\mathbf{y}_{\text{test}}, \mathbf{t}_{\text{test}}|X)$.

Gaussian Processes: Optimization

THE AUSTRALIAN NATIONAL UNIVERSITY

Newton Method and Conjugate Gradient Descent

Standard method for convex minimization problems

Nystrom Method, Sparse Greedy Approximation

We pick a subset of entries in the kernel matrix and express $\mathbf{t}$ by such a linear combination (choice at random, randomized optimal, or by diagonal pivoting)

Bayes Committee Machine

Split data into small subsets, solve on them, and combine the posterior distributions

Markov-Chain Monte Carlo

Not quite an optimization method but needed in the case of not-explicitly solvable integrals

The Bigger Picture

THE AUSTRALIAN NATIONAL UNIVERSITY

	Gaussian Processes	Support Vector Machines
"True" Goal		
	find $p(\mathbf{y}_{\text{test}}\|X, \mathbf{y}_{\text{train}})$	minimize expected risk $\mathbf{E}\left[c(x, y, f(x))\right]$.
Algorithmic Goal		
	find the mode of the posterior probability	minimize regularized risk
	mimize $-\log p(\mathbf{y}\|\mathbf{t}) - \log p(\mathbf{t}\|X)$	$\frac{1}{m}\sum_{i=1}^{m} c(x_i, y_i, f(x_i)) + \frac{\lambda}{2}\|f\|^2_{\mathcal{H}}$
Optimization Methods		
	Newton Method	Quadratic Programming
	Conjugate Gradient Descent	Linear Programming

The Bigger Picture

THE AUSTRALIAN NATIONAL UNIVERSITY

Data Dependent Term

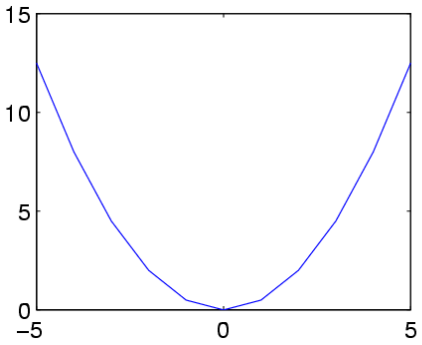
Negative Log-Likelihood

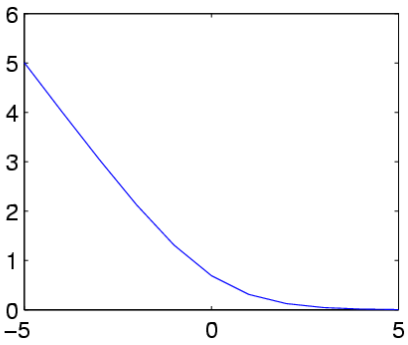
$$\sum_{i=1}^{m} -\log p(y_i|t_i)$$

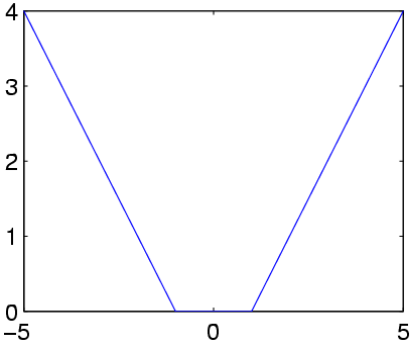
Empirical Risk

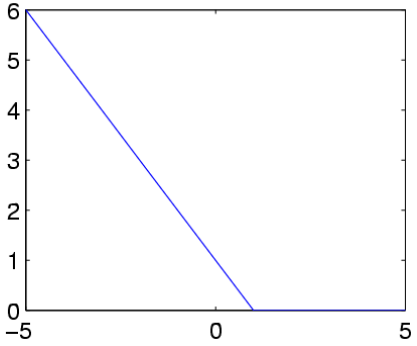
$$\frac{1}{m}\sum_{i=1}^{m} c(x_i, y_i, f(x_i))$$

Typical Loss/Likelihood Functions









$$-\log p(y|t) = \frac{1}{2}(y-t)^2 + c \qquad c(x, y, f(x)) = \max(0, |y - f(x)| - \epsilon)$$

$$-\log p(y|t) = -\log(1 + \exp(yt)) \qquad c(x, y, f(x)) = \max(0, 1 - yf(x))$$

The Bigger Picture

THE AUSTRALIAN NATIONAL UNIVERSITY

Data Independent Term

Prior Probability

$$-\log p(\mathbf{t}|X) = \frac{1}{2}\mathbf{t}^\top K^{-1}\mathbf{t}$$

Regularizer

$$\|f\|^2 = \alpha^\top K\alpha = \mathbf{t}^\top K^{-1}\mathbf{t} \text{ where } \mathbf{t} = K\alpha.$$

Prediction

$$\mu_{\text{test}} = K_{\text{test,train}}K^{-1}\mathbf{t}_{\text{train}}$$

we can compute $K^{-1}\mathbf{t}_{\text{train}}$ beforehand.

$$f(x) = \sum_{i=1}^{m} \alpha_i k(x_i, x)$$

Quality Measure

Mode of the log-posterior

$$\frac{1}{2}\log|K| + \frac{1}{2}\mathbf{t}^\top K^{-1}\mathbf{t} - \log p(\mathbf{y}|\mathbf{t})$$

Regularized Risk Functional

$$\sum_{i=1}^{m} c(x_i, y_i, [K\alpha]_i) + \frac{\lambda}{2}\alpha^\top K\alpha$$

The Bigger Picture

THE AUSTRALIAN NATIONAL UNIVERSITY

Parameter Tuning

Automatic Relevance Determination

maximize $p(\mathbf{y}, \mathbf{t}|X, \omega)p(\omega)$

- Margin Maximization

  $$\alpha^\top K \alpha$$

- Kernel Target Alignment

  $$\mathbf{y}^\top K \mathbf{y}$$

- Bound Minimization

  $$\Pr\{R[f] - R_{\text{emp}} > \epsilon\} \leq \delta$$

Summary

Normality Assumption for Latent Variables

We observe $\mathbf{y}$, which depends on a normally distributed $\mathbf{t}$

Approximations

When exact solution is not possible, maximize the joint distribution $p(\mathbf{y}, \mathbf{t}|X)$. This can be done approximately or via the EM algorithm.

Covariance Functions

Kernels $k(\mathbf{x}, \mathbf{x}')$ determine the shape of the covariance matrix (this encodes prior knowledge).

Connection to SVM

Negative Log Likelihood = Loss Function, Negative Log-Prior = Regularizer

For more information see

http://www.kernel-machines.org

http://www.learning-with-kernels.org