Probabilistic and Bayesian Modelling in Machine Learning and Artificial Intelligence

Manfred Opper

Sebastian Thiel

Background reading

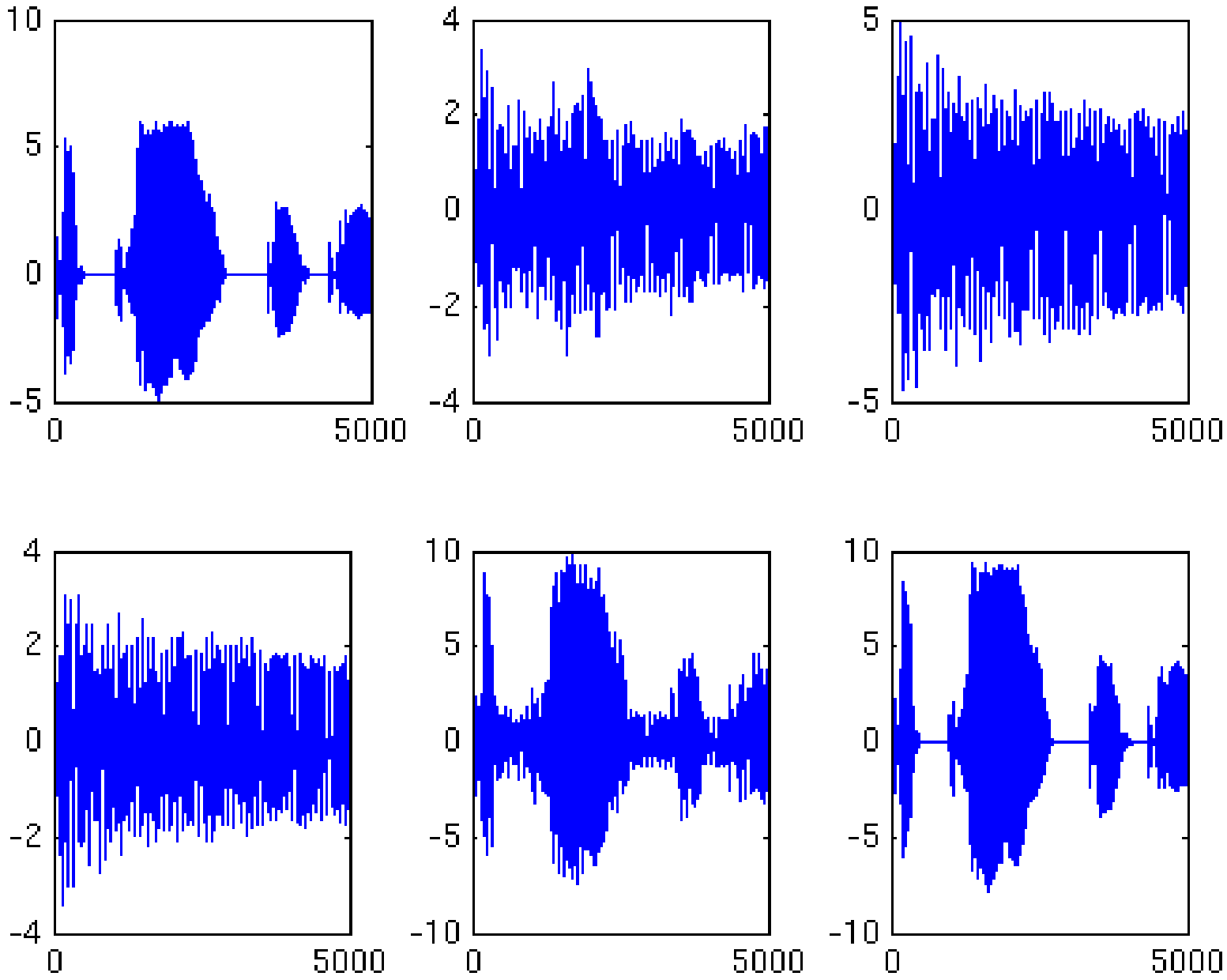
Pattern Recognition and Machine Learning, Christopher M. Bishop, Springer, 2006.

Information Theory, Inference, and Learning Algorithms, David J C MacKay, Cambridge University Press, 2003.

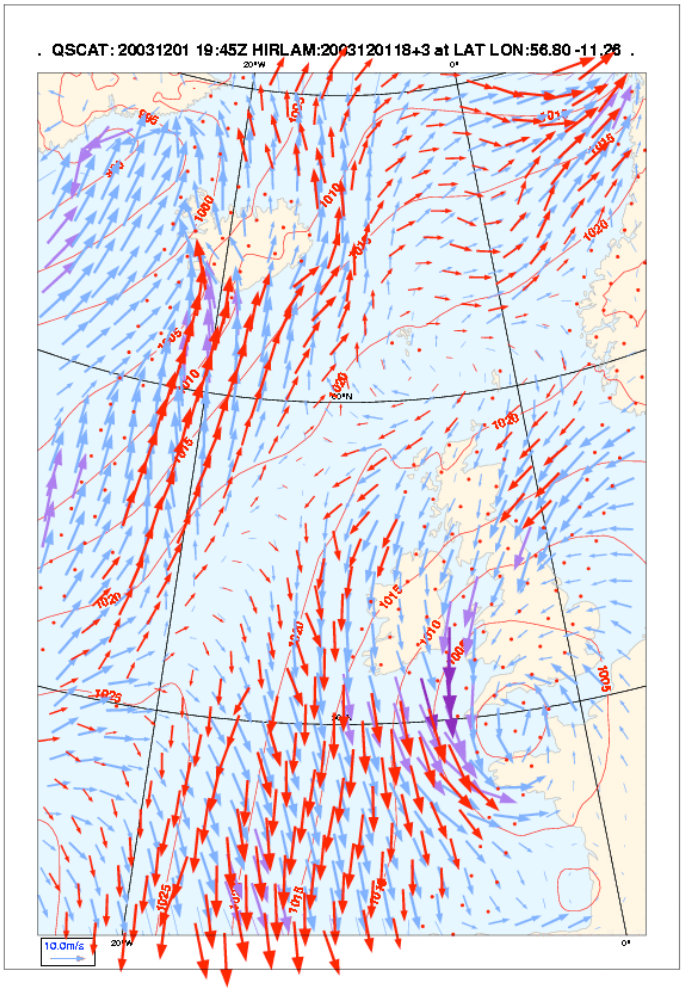
Bayesian Reasoning and Machine Learning, David Barber, Cambridge University Press, 2012.

Machine Learning - A probabilistic Perspective, Kevin P. Murphy, The MIT Press, 2012.

Advanced Mean Field Methods, M Opper and D Saad (eds.), The MIT Press, 2001.


10
5
0
-5
0
5000
4
2
0
-2
-4
0
5000
5
0
-5
0
5000
4
2
0
-2
-4
0
5000
10
5
0
-5
-10
0
5000
10
5
0
-5
-10
0
5000

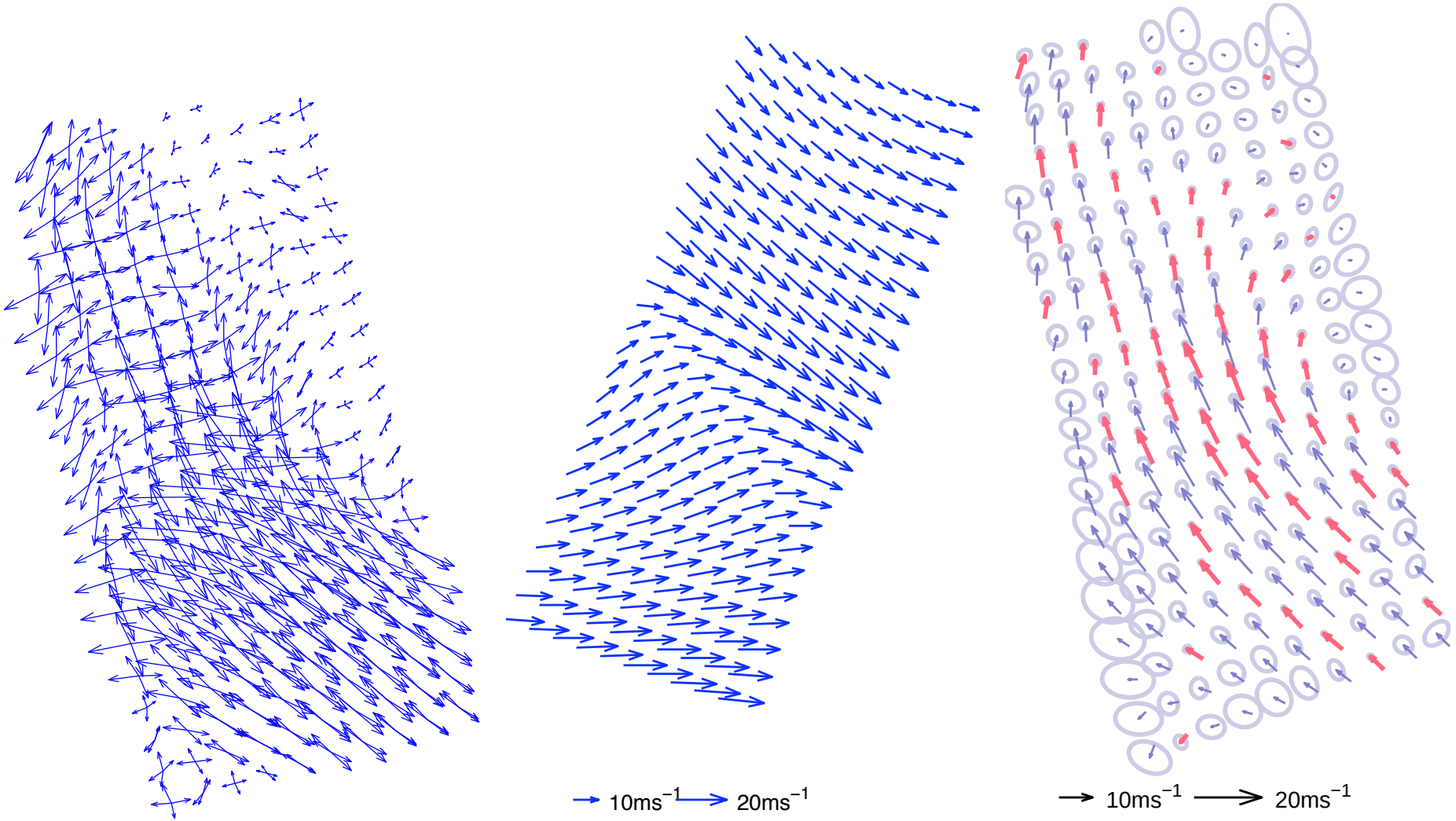
Measuring Windfields



(Ad Stoffelen/KNMI)

Scatterometry: Measuring windfields using radar backscattering on waterwaves (from satellites).

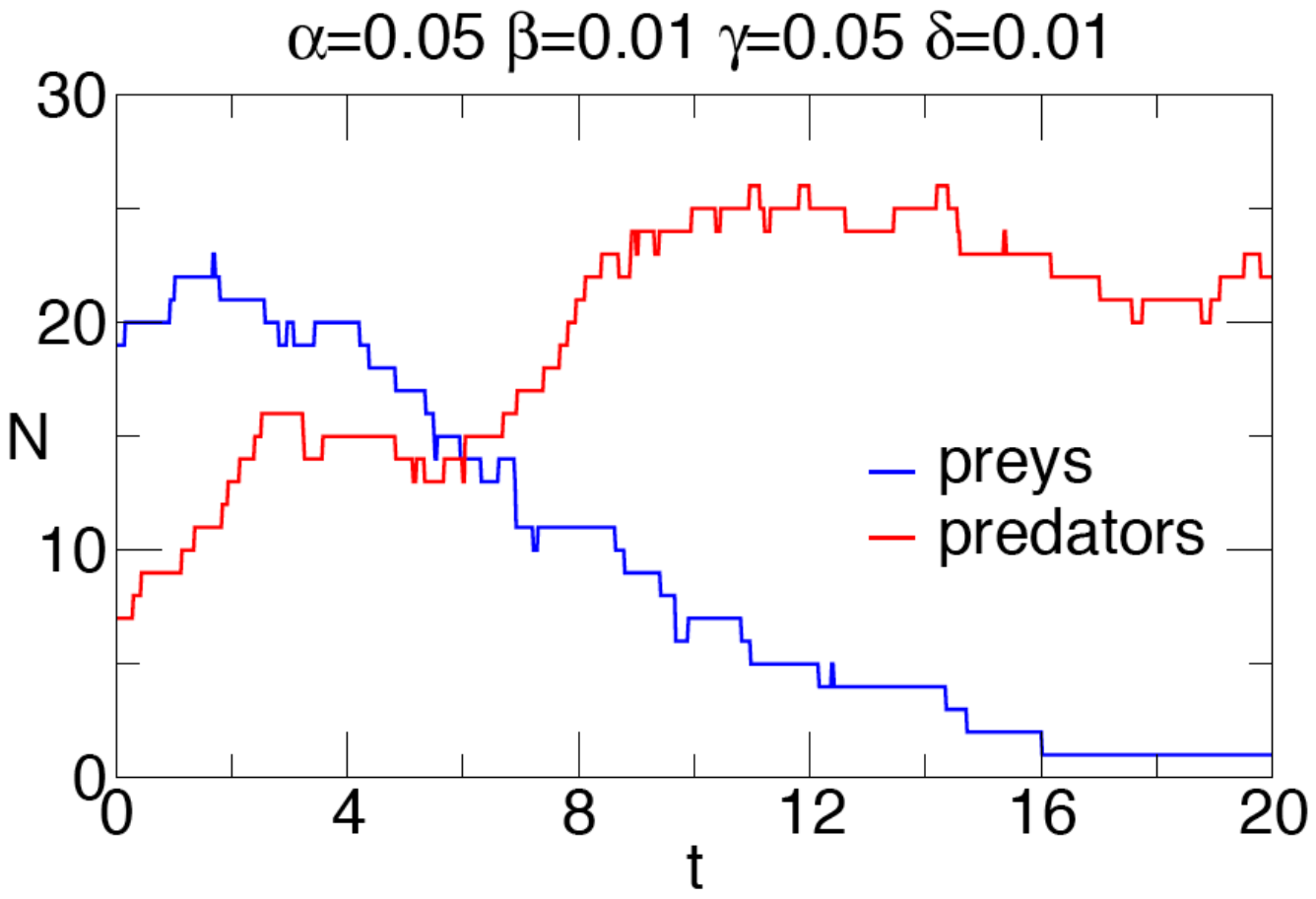
Ambiguities and prior knowledge



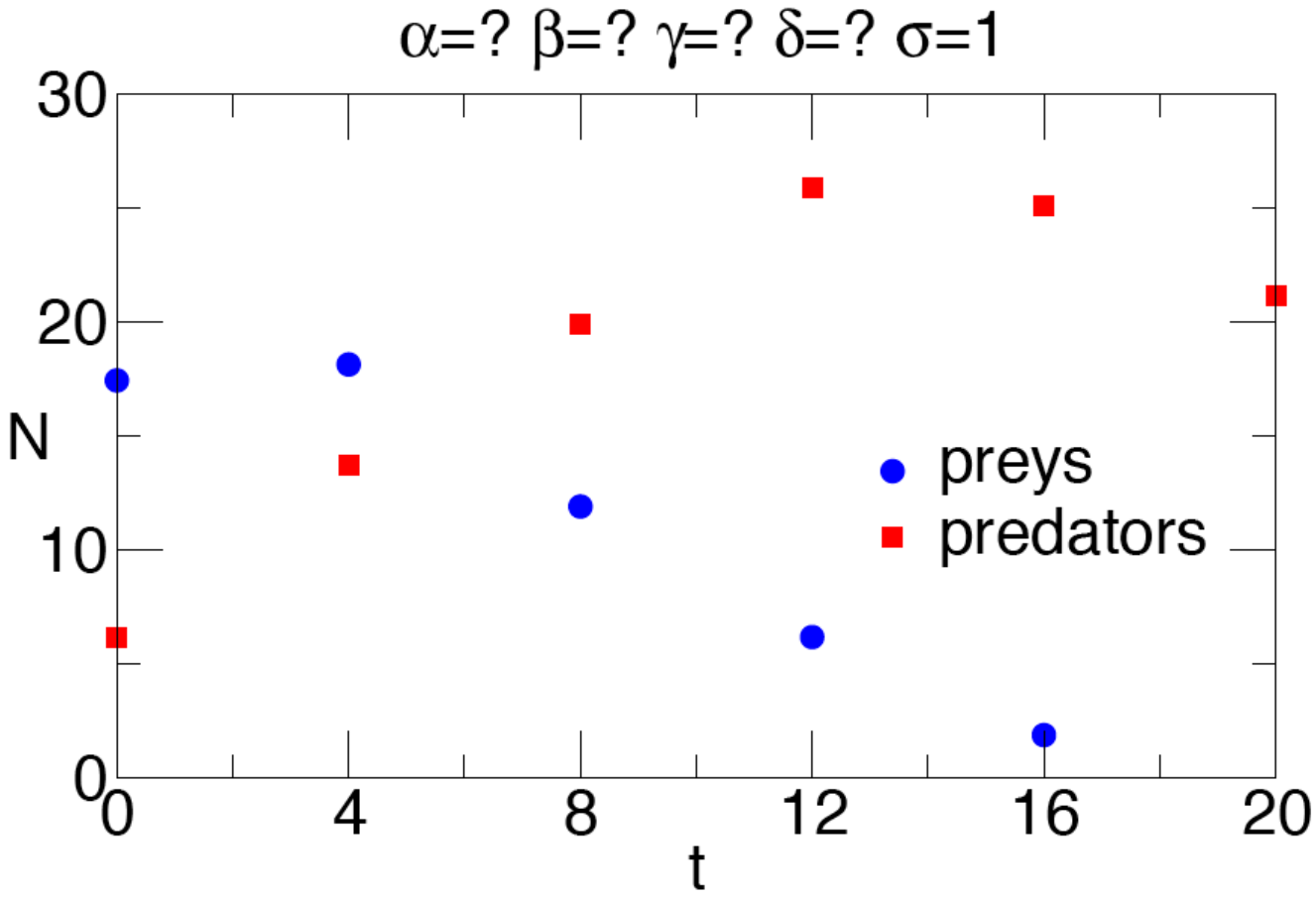
Likelihood typical a priori sample mean prediction.

Stochastic Lotka Volterra Model

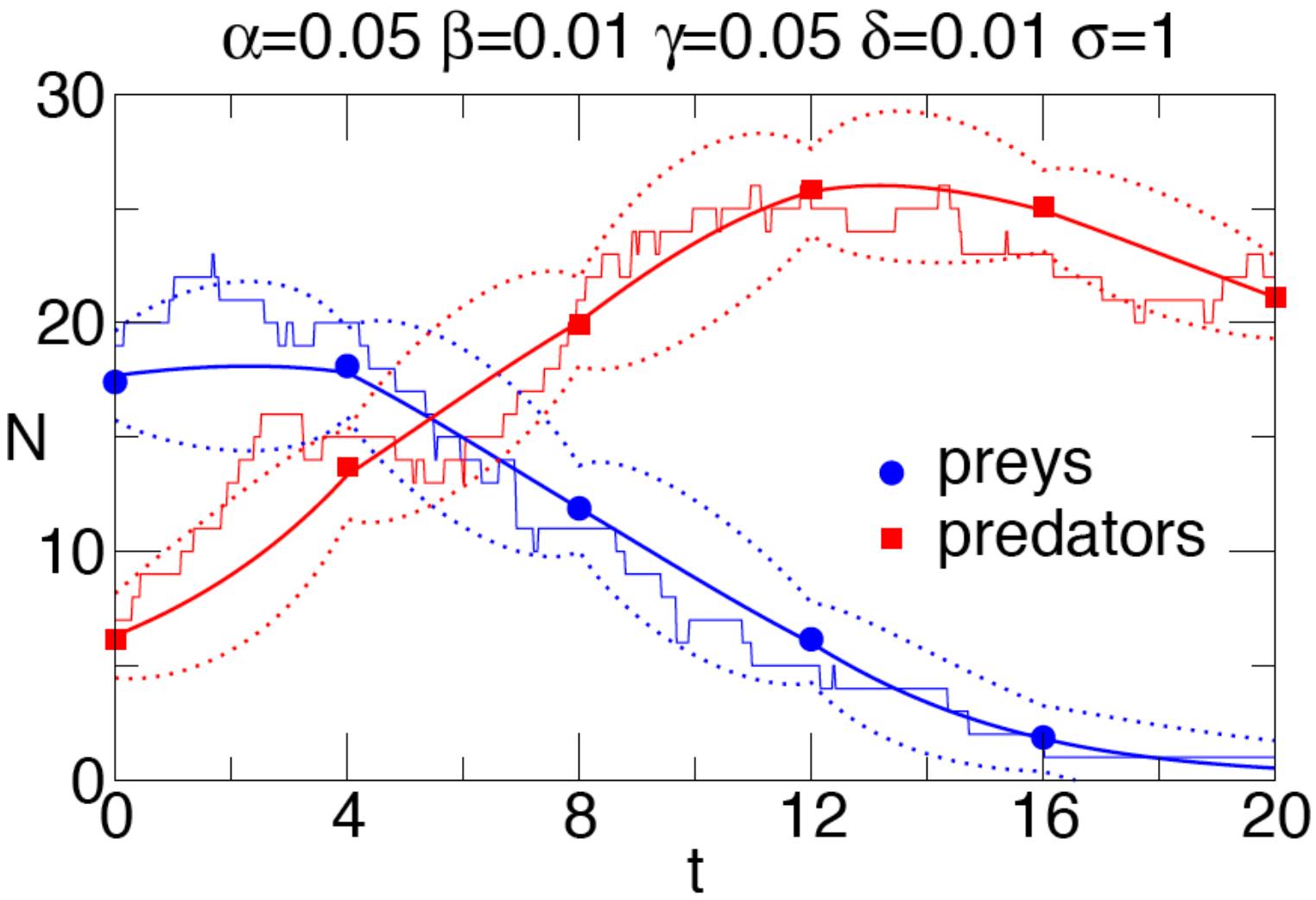
$$\begin{aligned}
\text{Prey} &\rightarrow 2\,\text{Prey} && \text{with Rate} && \alpha X_{\text{Prey}} \\
\text{Prey} &\rightarrow \emptyset && \text{with Rate} && \beta X_{\text{Prey}} X_{\text{Pred}} \\
\text{Predator} &\rightarrow 2\,\text{Predator} && \text{with Rate} && \delta X_{\text{Prey}} X_{\text{Pred}} \\
\text{Pred} &\rightarrow \emptyset && \text{with Rate} && \gamma X_{\text{Pred}}
\end{aligned}$$



The actual time series and the reaction constants



Discrete observations from a continuous time series


α=0.05 β=0.01 γ=0.05 δ=0.01 σ=1
30
20
10
0
N
0
4
8
12
16
20
t
preys
predators

Some probability essentials

Definitions

Sample Space Ω: Space of possible outcomes ω of a random experiment.

Events: (measurable) subsets of Ω.

Probabilities: Number $P(A)$ assigned to events A.

We have $0 \leq P(A) \leq 1$, $P(\emptyset) = 0$ and $P(\Omega) = 1$.

Addition Rule: If $A \cap B = \emptyset$ Then $P(A \cup B) = P(A) + P(B)$ (extends to countable sequence of disjoint events).

Random Variables are functions of outcomes $X(\omega)$.

For *discrete* rvs we define *the probability mass function* $P_X(x) = P(X = x)$. Often we speak (sloppily) about *the distribution* of X.

Joint distribution of two random variables:

$$P_{X,Y}(x,y) = P(X = x, Y = y) \ .$$

Marginal distributions: $P_X(x) = \sum_y P_{X,Y}(x,y)$ and $P_Y(y) = \sum_x P_{X,Y}(x,y)$.

For continuous random variables we define a *probability density* $p_X(x)$ by $\int_a^b p_X(x)\, dx = P(a < X < b)$.

A *joint density* can be defined for two (and more) variables:

$$\int \int_S p_{X,Y}(x,y)\, dxdy = P((X,Y) \in S)$$

for a set $S \in R^2$. *.

Marginal densities are obtained e.g. as $p(x) = \int_{-\infty}^{\infty} p(x,y)dy$

*Note: When it is clear which random variables are involved, I often write simply $p(x)$ instead of $p_X(x)$.

Transformation of random variables and their densities:

Let $y = f(x)$ be an invertible transformation and let the density of x be $p(x)$. We are interested in the density $q(y)$ of the random variable y.

Using $p(x)dx = q(y)dy$, we get

$$q(y) = p(x(y)) \left| \frac{dx}{dy} \right| = p(x(y)) \frac{1}{\left| \frac{dy}{dx} \right|}$$

Conditional Probabilities

$P(A|B) = \frac{P(A \cap B)}{P(B)}$ and similarly for conditional distributions: $P(x|y) = \frac{P(x,y)}{P(y)}$ and *conditional densities* $p(x|y) = \frac{p(x,y)}{p(y)}$.

Bayes Rule!!!

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)} = \frac{P(y|x)P(x)}{\sum_{x'} P(y|x')P(x')}.$$

Expectations

The expectation of X is defined as

$E(X) = \sum_x P(x)\ x$ (discrete case) or $E(X) = \int p(x)\ x\ dx$ (continuous case). For a function g of the rva X, we can show that

$E(g(X)) = \sum_x P(x)\ g(x)$ (discrete) or $E(g(X)) = \int p(x)\ g(x)\ dx$ (continuous).

Mean: $\mu = E[X]$

Variance: $Var(X) = E((X - \mu)^2) = E(X^2) - (E(X))^2$.

Linearity

$E(aX + bY) = aE(X) + bE(Y)$

Conditional Expectation

$E(Y|X = x)$ or $E(Y|x)$:

$E(g(Y)|X = x) = \sum_y g(y)\ P(y|x)$ (discrete case) and $E(g(Y)|X = x) = \int g(y)\ p(y|x)\ dy$ (continuous case).

Independence

(*Multiplication rule*):

A family of events $A_1, A_2, \ldots$ are called *independent* if for any subset $\{A_{i_1}, A_{i_2}, \ldots, A_{i_k}\}$ $P(A_{i_1} \cap \ldots \cap A_{i_k}) = P(A_{i_1})P(A_{i_2}) \cdots P(A_{i_k})$.

A family of random variables $X_1, X_2, \ldots$ are called *independent* if for any subset $\{X_{i_1}, X_{i_2}, \ldots, X_{i_k}\}$ $P(X_{i_1}, X_{i_2}, \ldots, X_{i_k}) = P(X_{i_1})P(X_{i_2}) \cdots P(X_{i_k}) = \prod_{j=1}^{k} P(x_{i_j})$ (with an analogous definition for densities). Hence, if X and Y independent then $P(x|y) = \frac{P(x,y)}{P(y)} = P(x)$.

Some properties of independent random variables $X_1, X_2, \ldots, X_N$:

- $E(X_1 \cdot X_2 \cdots X_N) = \prod_{i=1}^{N} E(X_i)$.
- $\mathrm{Var}\left(\sum_{i=1}^{N} X_i\right) = \sum_{i=1}^{N} \mathrm{Var}(X_i)$.

- Law of large numbers

  Let $X_1, X_2, \ldots, X_N$, i.i.d. with finite variance σ^2 and $S_N = \frac{1}{N}\sum_{i=1}^{N} X_i$, then one can show that

  $\lim_{N\to\infty} P(|S_N - E(X)| > \varepsilon) = 0.$

  Hence, when N large, with high probability we have $\frac{1}{N}\sum_{i=1}^{N} X_i \approx E(X)$.

  The proof uses addititivity of VAR and *Markov's* inequality.

Reminder of Gaussian densities

1-D Gaussian density

The density of a one dimensional Gaussian random variable $x \sim \mathcal{N}(\mu, \sigma^2)$ with *mean* $E(x) = \mu$ and variance $\sigma^2 = E(x-\mu)^2$ is given by

$$p(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-\frac{(x-\mu)^2}{2\sigma^2}}.$$

The d-dimensional Gaussian distribution

Let $\mathbf{x} = (x_1, \ldots, x_d)^T$ and $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_d)^T$

The Gaussian density for $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is given by

$$p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{d}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}}\, \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\} \qquad (1)$$

$\boldsymbol{\mu} = E[\mathbf{x}]$ is **mean** vector and $\boldsymbol{\Sigma}$ is a $d \times d$ *covariance* matrix. One can show that

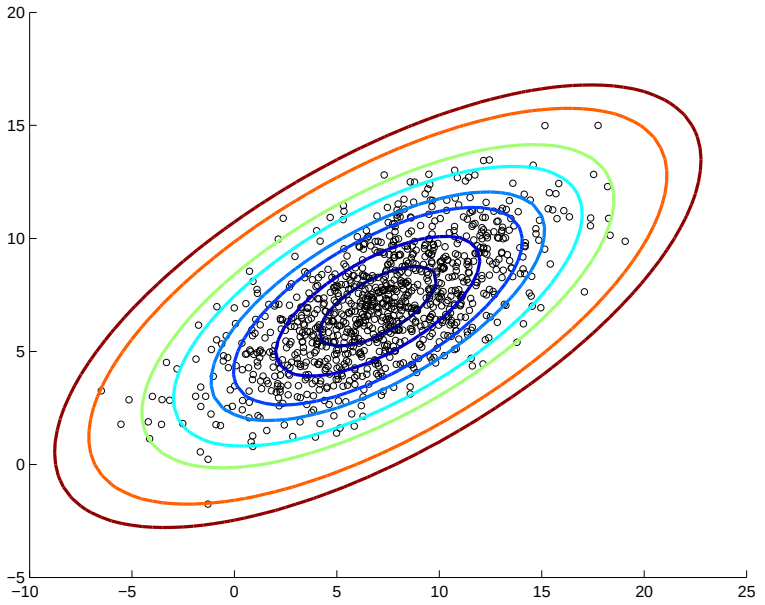
$$\Sigma_{ij} = E(x_i - \mu_i)(x_j - \mu_j)$$

or $\Sigma = E(\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^T$.

Example:

Lines of constant density and random data for a two dimensional Gaussian. The mean is $\boldsymbol{\mu} = (7, 7)^T$ and the covariance matrix is $\boldsymbol{\Sigma} =$

$$\begin{pmatrix} 16.6 & 6.8 \\ 6.8 & 6.4 \end{pmatrix}$$



Eigenvalue problem for $\mathbf{\Sigma}$

To understand the properties of this density, we need to make a little detour and consider

$$\mathbf{\Sigma}\mathbf{u}_i = \lambda_i \mathbf{u}_i \tag{2}$$

with an eigenvector $\mathbf{u}_i$ and eigenvalue λ_i, where $i = 1, \ldots, d$. $\mathbf{\Sigma}$ is a real symmetric matrix with orthonormal eigenvectors $\mathbf{u}_i \cdot \mathbf{u}_j = \mathbf{u}_i^T \mathbf{u}_j = \delta_{ij}$. With the $d \times d$ *orthogonal* matrix formed by the d column eigenvectors

$$\mathbf{U} = (\mathbf{u}_1 \mathbf{u}_2 \cdots \mathbf{u}_d). \tag{3}$$

we have $\mathbf{U}^T \mathbf{U} = \mathbf{I}$.

Using (3) and the diagonal matrix $\mathbf{\Lambda} = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & \lambda_n \end{pmatrix}$ we can rewrite the eigenvalue equations (2) as $\mathbf{\Sigma U} = \mathbf{U\Lambda}$ or

$$\mathbf{\Sigma} = \mathbf{U\Lambda U}^{-1} = \mathbf{U\Lambda U}^{T} \tag{4}$$

and

$$\mathbf{\Sigma}^{-1} = \mathbf{U\Lambda}^{-1}\mathbf{U}^{-1} = \mathbf{U\Lambda}^{-1}\mathbf{U}^{T} \tag{5}$$

$\mathbf{U}$ defines an *orthogonal* transformation by $\mathbf{y} = \mathbf{U}^T(\mathbf{x} - \boldsymbol{\mu})$, or $\mathbf{x} = \boldsymbol{\mu} + \mathbf{U}\mathbf{y}$. This transformation preserves inner products, i.e. we have for two vectors $\mathbf{y}_1$ and $\mathbf{y}_2$ that $\mathbf{y}_1^T\mathbf{y}_2 = (\mathbf{x}_1 - \boldsymbol{\mu})^T(\mathbf{x}_2 - \boldsymbol{\mu})$. It can be understood as a transformation to a new coordinate system given by a combination of a *shift* and a *rotation*. We also get

$$(\mathbf{x} - \boldsymbol{\mu})^T\mathbf{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}) = \mathbf{y}^T\mathbf{\Lambda}^{-1}\mathbf{y} = \frac{y_1^2}{\lambda_1} + \frac{y_2^2}{\lambda_2} + \ldots + \frac{y_d^2}{\lambda_d}$$

Using the new coordinate system, we see that

- surfaces of constant probability density for the Gaussian density $p(\mathbf{x})$, eq. (1) are *ellipsoids*.

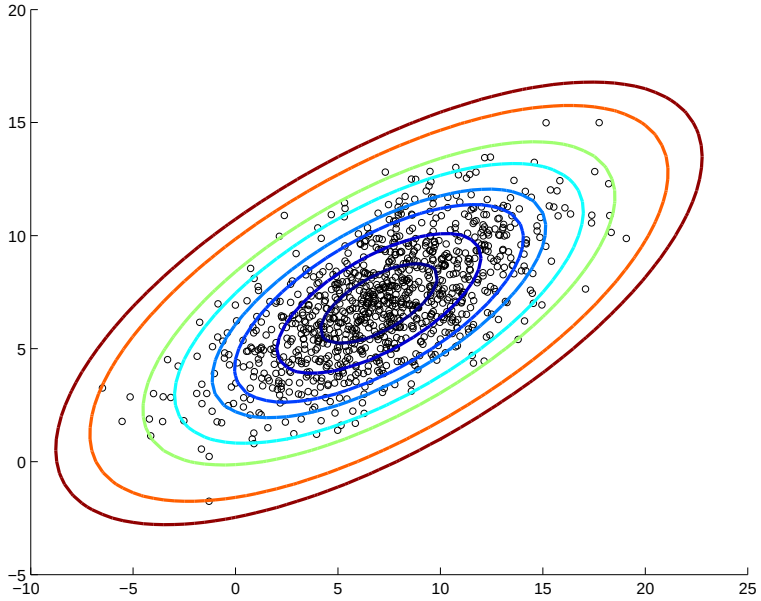
- the random variables defined by y coordinates $\mathbf{Y} = \mathbf{U}^T(\mathbf{X} - \boldsymbol{\mu})$ are *independent*, ie.

  $p(\mathbf{y}) = \prod_{i=1}^{d} \frac{1}{\sqrt{2\pi\lambda_i}} e^{-\frac{y_i^2}{2\lambda_i}}$

- We see that Σ is indeed the matrix of covariances, i.e

  $\Sigma_{ij} = E(x_i - \mu_i)(x_j - \mu_j)$, i.e. $\Sigma = E(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^T$.

Back to the example:



The covariance matrix is $\mathbf{\Sigma} = \begin{pmatrix} 16.6 & 6.8 \\ 6.8 & 6.4 \end{pmatrix}$. The eigenvalues are $\lambda_1 = 20$ and $\lambda_2 = 3$ with eigenvectors $\mathbf{u}_1 = \frac{1}{\sqrt{5}}(2,1)^T$, and $\mathbf{u}_2 = \frac{1}{\sqrt{5}}(1,-2)^T$.

- Generate Gaussian distributed random vectors $\mathbf{x}$ with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ from vectors $\mathbf{z}$ with *indepedent* normal components $E(z_i z_j) = \delta_{ij}$ by the transformation $\mathbf{x} = \mathbf{U}\boldsymbol{\Lambda}^{\frac{1}{2}}\mathbf{z} + \boldsymbol{\mu}$.

  Alternative method: Perform *Cholesky decomposition* $\boldsymbol{\Sigma} = \mathbf{A}\mathbf{A}^\top$. Then set $\mathbf{x} = \mathbf{A}\mathbf{z}$.

- Sums of jointly Gaussian random variables are Gaussian. Marginal & conditional densities of jointly Gaussian random variables are Gaussian.

- Central limit theorems: For i.i.d. x_i with finite variance, the normalised sum $z_n = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} (x_i - m)$ becomes asymptotically Gaussian distributed.

Some inequalities

Cauchy–Schwarz:

$$\{E(xy)\}^2 \leq E(x^2)E(y^2) \ .$$

Equality $=$ if and only if $P(sx = ty) = 1$ for some nonrandom s and t.

Markov:

$$P(x \geq a) \leq \frac{E(x)}{a}$$

for $x \geq 0$.

Chebychev:

$$P(|x| \geq a) \leq \frac{E(x^2)}{a^2}$$

Follows from *Markov* by substituting $x \to x^2$.

Jensen

For $f(\cdot)$ **convex** (i.e. $f''(x) \geq 0$ for all x) we have

$$E[f(X)] \geq f(E[X])$$

Proof: For fixed (non random y), Use the Taylor expansion

$$f(X) = f(y) + (X-y)f'(y) + \frac{1}{2}(X-y)^2 f''(\xi) \geq f(y) + (X-y)f'(y)$$

where $\xi \in [X, y]$. we have

$$E[f(X)] \geq f(y) + (E[X]-y)f'(y)$$

The result follows by setting $y = E[X]$. If f strictly convex: Equality $=$ if and only if $X = E(X)$ a.e.

The KL divergence

For any two distributions $p(\mathbf{x})$ and $q(\mathbf{x})$, we can show using Jensen's inequality that the **Kullback–Leibler divergence**

$$KL(p,q) = E_p\left[\ln \frac{p(\mathbf{x})}{q(\mathbf{x})}\right] \geq 0$$

where E_p denotes expectation wrt to p. One has equality $= 0$ if and only if $p = q$ almost everywhere. The KL is a asymmetric dissimilarity measure between distributions. It is invariant against transformations of the random variables.

Model Parameter Estimation by Maximum Likelihood

Example I: The biased coin

Consider a data sequence $D = (x_1, x_2, \ldots, x_n)$ of bits $x_i \in \{0, 1\}$ which we belive are generated independently at random with the same probability. Call θ the **unknown** probabiliy of 1. The probability of the sequence D under this **model** is

$$P(D|\theta) = \prod_{i=1}^{n} \theta^{x_i} \ (1-\theta)^{1-x_i}$$

If D is observed (ie fixed), we study $P(D|\theta)$ as a function of θ. We call it the **likelihood**.

To **estimate** the **true parameter** θ of the model from which the data was generated we use the method of Maximum Likelihood choosing $\widehat{\theta} = \operatorname{argmax} P(D|\theta)$. For this parameter, the observed data have the highest probability. Equivalent we maximize the log–likelihood

$$\ln P(D|\theta) = \sum_{i=1}^{n} (x_i \ln \theta + (1-x_i) \ln(1-\theta)) = n_1 \ln \theta + (n-n_1) \ln(1-\theta)$$

Differentiating gives

$$\frac{d \ln P(D|\theta)}{d\theta} = 0 \qquad \longrightarrow \qquad \widehat{\theta} = \frac{n_1}{n} \ .$$

Example II: Gaussian density

The density of a one dimensional Gaussian random variable with *mean* $E(X) = \mu$ and variance $\sigma^2 = E(X - \mu)^2$ is given by

$$p(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \, e^{-\frac{(x-\mu)^2}{2\sigma^2}} .$$

The goal is to estimate μ, σ^2 from a set of data $D = (x_1, x_2, \ldots, x_n)$. Each data is assumed to be drawn independently from $p(x|\mu, \sigma^2)$. Maximizing the Likelihood is equivalent to *minimizing*

$$- \ln p(D|\mu, \sigma^2) = \frac{1}{2} \sum_{i=1}^{N} \left\{ \frac{(x_i - \mu)^2}{\sigma^2} + \ln(2\pi\sigma^2) \right\}$$

Minimization with respect to μ and σ^2 leads to the *Maximum Likelihood Estimates*

$$\widehat{\mu} = \frac{1}{n}\sum_{i=1}^{N} x_i$$

$$\widehat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n} (x_i - \widehat{\mu})^2$$

Example III: Gaussian noise and Linear Regression

Observe a set of input–ouput data $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ with x = input, y = target values. Try to fit a linear function $y = w_0 + w_1 x$ to the data. We represent this as a probabilistic model and assume that n observations are generated as

$$y_i = w_0 + w_1 x_i + \text{ noise}_i$$

for $i = 1, \ldots, n$. For independent Gaussian noise of variance σ^2 we can write

$$p(y, x|\mathbf{w}) = p(y|x, \mathbf{w})p(x) = \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-\frac{(y-w_0-w_1x)^2}{2\sigma^2}}\, p(x)$$

The unknown parameters are $\mathbf{w} = (w_0, w_1)$ and σ^2.

Hence, the negative **log–likelihood** is

$$-\ln P(D|\mathbf{w}, \sigma^2) = \text{const} + \frac{1}{2\sigma^2} \sum_{i=1}^{N} (y_i - w_0 - w_1 x_i)^2$$

and ML estimation of w_0 and w_1 becomes equivalent to *Least Squares* fitting!

Generalised linear models

Assume data generated as $y_i = f(x_i) + \nu_i$ for $i = 1, \ldots, N$, with $f(\cdot)$ unknown, ν_i i.i.d. $\sim \mathcal{N}(0, \sigma^2)$.

Polynomial regression:

$$f_{\mathbf{w}}(x) = \sum_{j=0}^{K} w_j x^j$$

allowing for different orders K. The **likelihood** is

$$p(D|\mathbf{w}) = \frac{1}{(2\pi\sigma^2)^{N/2}} \exp\left[-\sum_{i=1}^{N} \frac{(y_i - f(x_i))^2}{2\sigma^2}\right]$$

Properties of Estimators

- Parameter estimates $\widehat{\theta}(D)$ are random variables with respect to the random drawing of the data. The *bias* of an estimator is defined as $E_D(\widehat{\theta}) - \theta$ and its *variance* as $E_D\left(\widehat{\theta} - E_D(\widehat{\theta})\right)^2$, where the expectation E_D is over datasets which are drawn at random from a distribution with *true* parameter θ.

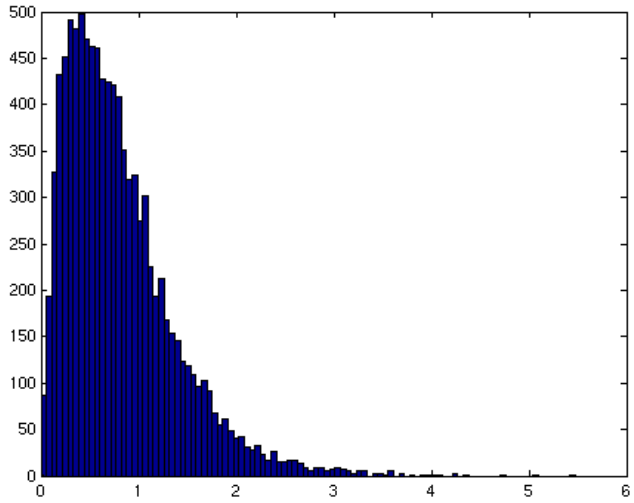
- "Good" estimators should become asymptotically *consistent*, i.e. the estimates should converge to the *true* parameters as $N \to \infty$. This means that bias and variance must go to 0 as $N \to \infty$.

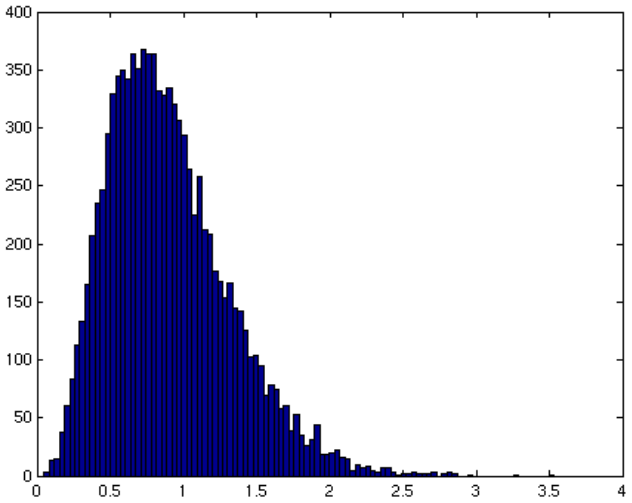
- ML estimators are consistent under rather general circumstances. Note that

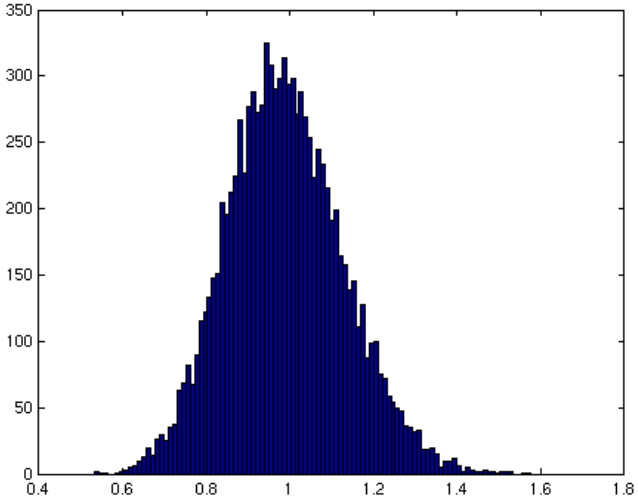
$$-\frac{1}{n} \ln P(D|\theta) = -\frac{1}{n} \sum_i \ln p(x_i|\theta) \to -E_D \ln p(x|\theta)$$

  Hence, minimizing $-\frac{1}{n} \ln P(D|\theta)$ becomes asymptotically equivalent of minimizing $KL(p_{\text{true}}, p_\theta)$!

ML estimation of the variance (10.000 repetitions) for $n = 5, 10, 100$







Exponential families

ML estimates look simple (analytically computable) for models from the so–called (*regular*†) **exponential families** which in their **canonical representation** are written as

$$p(x|\boldsymbol{\theta}) = f(x)\ \exp[\boldsymbol{\psi}(\boldsymbol{\theta}) \cdot \boldsymbol{\phi}(x) + g(\boldsymbol{\theta})]\ .$$

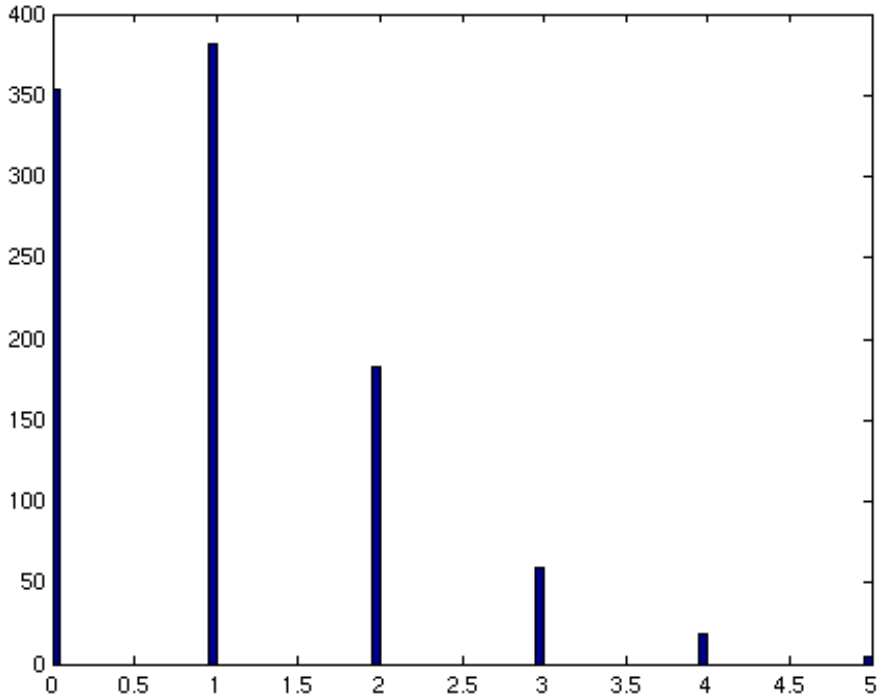
For a Gaussian, take $\boldsymbol{\psi}(\boldsymbol{\theta}) = (\mu/\sigma^2, 1/2\sigma^2)$ and $\boldsymbol{\phi}(x) = (x, -x^2)$.

(† regular means that the range of the data x is independent of the parameter θ).

Another exponential family: Poisson distributions

$$p(n|\theta) = e^{-\theta}\frac{\theta^n}{n!}$$

for $n = 0, 1, 2, \ldots$. This shows the distribution for $\theta = 1$.



Example: Multinomial family

Let $\mathbf{n} = (n_1, \ldots, n_K)$, with $n_j \in N$ and $\sum_j n_j = n$, we define the Multimomial family as

$$P(\mathbf{n}|\boldsymbol{\theta}) = \frac{n!}{\prod_{j=1}^{K} n_j!} \prod_{j=1}^{K} \theta_j^{n_j}$$

where $\sum_{j=1}^{K} \theta_j = 1$. Useful for **histogramme** data (counts, e.g. in *Bag of words* model).

Sufficiency: Let $p(x|\theta)$ be a parametric familiy. A statistics $T(\mathbf{x})$ of the sample $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ is called **sufficient** if the conditional probability

$$p(\mathbf{x}|T(\mathbf{x}) = t, \theta)$$

is independent of θ. Thus $T(\mathbf{x})$ incorporates all relevant information of the parameter θ!

For exponential families, $\mathbf{T}(\mathbf{x}) = \sum_{i=1}^{n} \boldsymbol{\phi}(x_i)$ is a sufficient statistics.

Some exponential families might be too complex for Maximum Likelihood

- Ising Model for binary data: Let $x_i = \pm 1$ for $i = 1, \ldots, N$. Joint distribution of the variables is defined as (Markov random field)

$$p_{Ising}(\mathbf{x}) = \frac{1}{Z_p} \exp\left(\sum_{(i,j)} \theta_{ij} x_i x_j + \theta_i x_i\right)$$

- Used to predict effective couplings between neurons. The model also appears as a "Boltzmann machine" in AI. More general (Potts) models (x_i has more than 2 states) were used to predict interactions between amino acids in proteins.

- ML estimation of parameters θ_{ij} and θ_i by gradient descent requires computation of $E[x_i x_j]$. Computation of

$$Z_p = \sum_{\{x_i = \pm 1\}_{i=1}^N} \exp\left(\sum_{(i,j)} \theta_{ij} x_i x_j + \sum_i \theta_i x_i\right)$$

requires 2^N summations !

- Maximise logarithm of *Pseudo–likelihood:*

$$\sum_{k=1}^{M} \sum_{i=1}^{N} \ln P(x_i^k | \mathbf{x}_{-i}^k, \theta)$$

for M data $\mathbf{x}^1, \ldots \mathbf{x}^M$ instead !

- Justification: Show that

$$\sum_{i=1}^{N} E\left[\nabla_\theta \ln P(x_i | \mathbf{x}_{-i}, \theta)\right] = 0$$

Efficiency & Rao–Cramér inequality

This limits the speed at which the estimate $\hat{\boldsymbol{\theta}}$ approaches the true parameter θ on average. For a single (scalar) parameter

$$\mathrm{Var}(\hat{\theta}) \geq \frac{(\partial_\theta E(\hat{\theta}))^2}{nJ(\theta)}$$

with $J(\theta) = E_\theta \left[\frac{d \ln p(x|\theta)}{d\theta} \right]^2$.

Generalization to a k dimensional vector of parameters: For any real vector $(z_1, \ldots, z_k)$ (we specialise to **unbiased** estimators $E(\hat{\theta}) = \theta$ for simpicity)

$$E\left(\sum_i z_i (\hat{\theta}_i - \theta_i) \right)^2 \geq \frac{1}{n} \sum_{ij} z_i z_j (J^{-1}(\boldsymbol{\theta}))_{ij} \ , \tag{6}$$

with the **Fisher Information** matrix

$$J_{ij}(\theta) = \int dx \ p(x|\boldsymbol{\theta}) \partial_i \ln p(x|\boldsymbol{\theta}) \partial_j \ln p(x|\boldsymbol{\theta}) \ .$$

For $z_i \geq 0$, we can interprete the left hand side as a squared weighted average of the individual error components $\widehat{\theta}_i - \theta_i$. Estimators which fulfill these relations with an **equality**, are called **efficient**. Under weak assumptions, ML estimators are asymptotically efficient.

One can show that (under some technical conditions)

$$\widehat{\theta}_{ML} \sim \mathcal{N}\left(\theta, \frac{1}{n} J^{-1}(\theta)\right)$$

for $n \to \infty$. To use this result for the computation of error bars, we can use the approximation

$$J_{ij}(\boldsymbol{\theta}) \approx -\frac{1}{n} \partial_i \partial_j \sum_i \ln p(x_i|\widehat{\boldsymbol{\theta}}_{ML})$$

Note: A different representation of the Fisher Information is

$$J_{ij}(\boldsymbol{\theta}) = -\int dx \; p(x|\boldsymbol{\theta}) \partial_i \partial_j \ln p(x|\boldsymbol{\theta}) \; .$$

In the case, where the family $p(x|\theta)$ **does not contain the true distribution** $p(x)$ one has a similar result

$$\widehat{\theta}_{ML} \sim \mathcal{N}\left(\theta_0, \frac{1}{n} J^{-1} K J^{-1}\right)$$

for $n \to \infty$. where

$$J_{ij} = -\int dx\; p(x) \partial_i \partial_j \ln p(x|\theta_0) \;.$$

and

$$K_{ij} = \mathrm{COV}_p[\nabla \ln p(x|\theta_0)] \;.$$

with $\theta_0 = \arg\min KL(p, p(\cdot|\theta))$ gives the model closest (in relative entropy) to the true distribution p.

S. Amari has developed a differential geometric (Information geometry) approach to estimation. Here, one defines a **metric** in parameter space by

$$||d\theta||^2 \propto \sum_{ij} d\theta_i J_{ij}(\theta) d\theta_j = d\boldsymbol{\theta}^T \mathbf{J}(\theta) d\boldsymbol{\theta}. \tag{7}$$

which reflects how well neighbouring distributions can be distinguished by an estimation based on random data. Assuming that the probability distribution of efficient estimators is Gaussian (at large n) with a covariance given by (6), the probability density that a point close to the true value θ will be the estimate for θ, depends only on the distance $||d\theta||$.

Online Learning

As a learning algorithm, one can use e.g. a gradient descent algorithm and iterate

$$\boldsymbol{\theta}' = \boldsymbol{\theta} + \eta \, \nabla_\theta \sum_{k=1}^{n} \ln p(x_k|\boldsymbol{\theta})$$

until convergence. This requires storage of all previous data.

Goal of online learning: Calculate new estimate only based on the new data point x_{n+1}, the old estimate $\hat{\boldsymbol{\theta}}(n)$ (and possibly a set of other auxiliary quantities which have to be updated at each time step, but are much smaller in number than the entire set of previous training data).

Popular idea:

$$\boldsymbol{\theta}(n+1) = \boldsymbol{\theta}(n) + \eta(n) \, \nabla_\theta \ln p(x_{n+1}|\boldsymbol{\theta}(n))$$

If the algorithm should converge asymptotically, the learning rate $\eta(n)$ must be decreased during learning. A schedule $\eta \propto 1/n$ yields the fastest rate of convergence, but the prefactor must be chosen with

care, in order to avoid that the algorithm gets stuck away from the optimal parameter.

Natural gradient learning

S. Amari: Replace scalar learning rate $\eta(n)$ by a tensor. This is derived from the natural **distance** $||\Delta\theta||$ which reflects distances between probability distributions and is invariant against transformations of the parameters. A simple Euklidian distance will not satisfy this condition.

In the **natural gradient** algorithm the update is defined by a minimization of the training energy under the condition that $||\Delta\theta||^2$ is kept fixed. Solving the constrained variational problem for small $\Delta\theta$ yields

$$\boldsymbol{\theta}(n+1) = \boldsymbol{\theta}(n) + \gamma_n \mathbf{J}^{-1}(\boldsymbol{\theta}(n))\nabla_\theta \ln p(x_{t+1}|\boldsymbol{\theta}(n)).$$

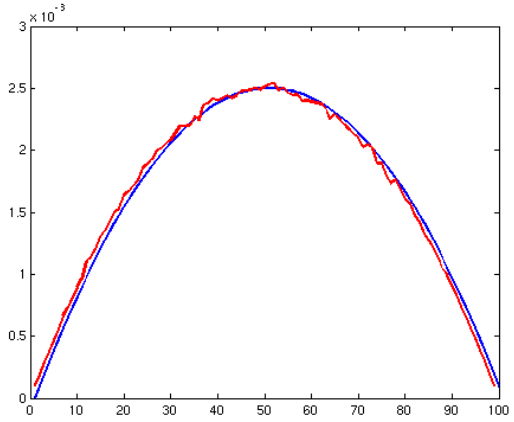
The differential operator $\mathbf{J}^{-1}(\boldsymbol{\theta}(n))\nabla_\theta$ is termed natural gradient. For the choice $\gamma_n = \frac{1}{n}$, one can show that the online algorithm yields *asymptotically efficient* estimation.

Example: Fisher Information

Bernoulli random variables

$p(x|\theta) = \theta^x(1-\theta)^{1-x}$ has $J(\theta) = \frac{1}{\theta(1-\theta)}$

$E(\widehat{\theta} - \theta)^2$ and $\frac{1}{J(\theta)n}$ as a function of θ



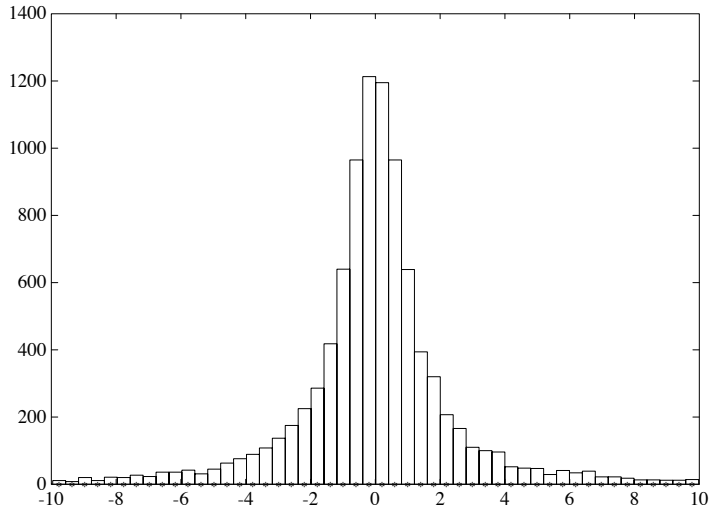
Cauchy density

$p(x|\theta) = \frac{1}{\pi(1+(x-\theta)^2)}$ has $J(\theta) = \pi/8$.

Estimating a Cauchy Density

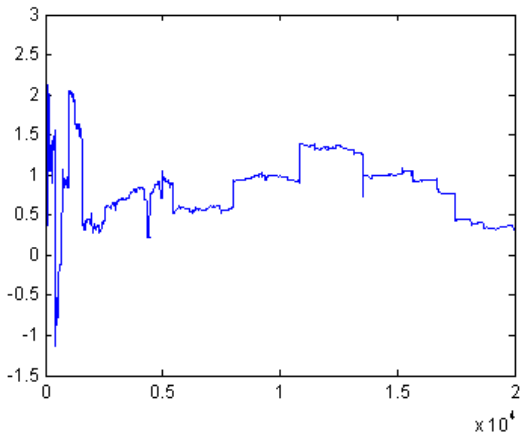
We consider the family of Cauchy densities given by

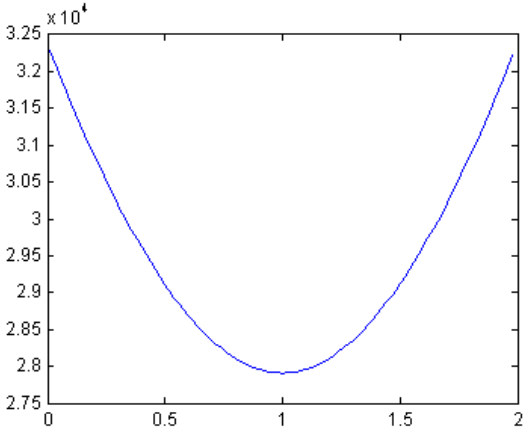
$$p(x|\theta) = \frac{1}{\pi(1 + (x - \theta)^2)} \ .$$



with location parameter θ.

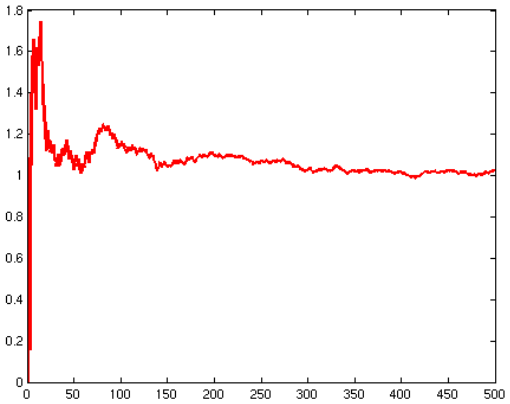
Naive estimate $\widehat{\theta} = \frac{1}{n}\sum_i x_i$ negative log–likelihood $-\ln p(D|\theta)$.
(true $\theta = 1$).

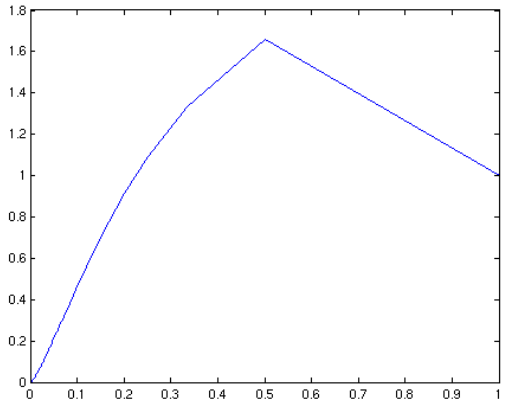




Natural gradient $\theta_{n+1} = \theta_n + \frac{4(x_{n+1}-\theta_n)}{n(1+(x_{n+1}-\theta_n)^2)}$

Prediction θ_n (single run) Average error (10.000 runs) vs $1/n$.





Independent Component Analysis (ICA): A latent variable model

- Find something "interesting" in signals:

  'cocktail party problem'

  EEG, ECG signals, FMRI data

- Feature extraction.

Generative Model

$$\mathbf{x}(t) = \mathbf{A}\mathbf{S}(t) + \textbf{ noise}$$

- $\mathbf{x} = (x_1, \ldots, x_d)$ vector of observed data (signals, images), $t =$ index
- $\mathbf{S} = (s_1, \ldots, s_m)$ vector of statistically independent latent source variables (unknown!)
- $\mathbf{A}$: $(d \times m)$ Mixing Matrix (unknown parameter !)

Goal:

Demix the signals and recover sources

$$\widehat{\mathbf{S}}(t) = \mathbf{W}\mathbf{x}(t)$$

with $\mathbf{W} = \mathbf{A}^{-1}$ for square matrices and no noise.

Ambiguities: Permutation of Sources, Scaling $s_i \to \lambda s_i$.

Some Interpretations of ICA

- $x_i(t) = \sum_j A_{ij} s_j(t)$

  $x_i(t)$ is signal at sensor i & $s_j(t)$ speaker j at time t.

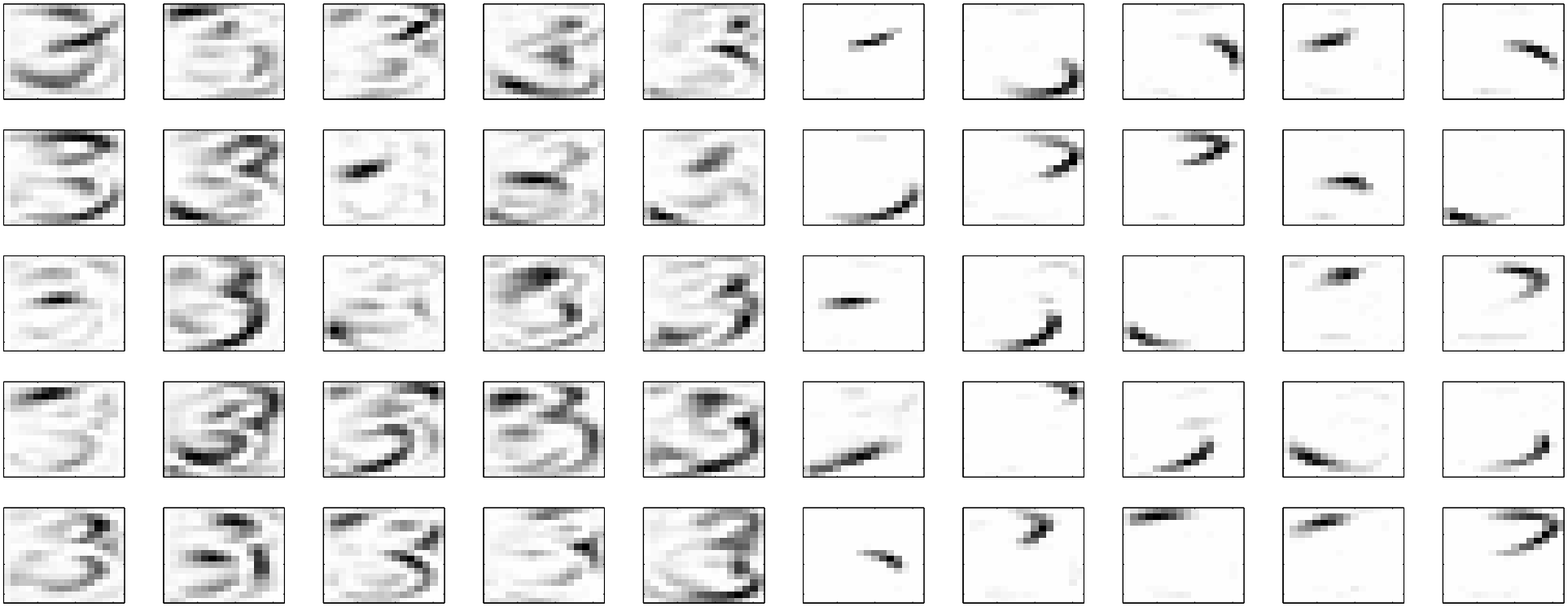
- $x_i(t) = \sum_j A_{ij} s_j(t)$

  Vector $x_i(t)$ of pixel intensities of image t is expanded into features $\mathbf{A}_{\bullet j}$ and the $s_j(t)$ are the statistically independent coefficients.

- $x_t(i) = \sum_j A_{tj} s_j(i)$

  $x_t(i)$ intensity of each pixel i at time t is a time dependent mixture of time independent activity pattern $s_j(i)$.

Feature Extraction

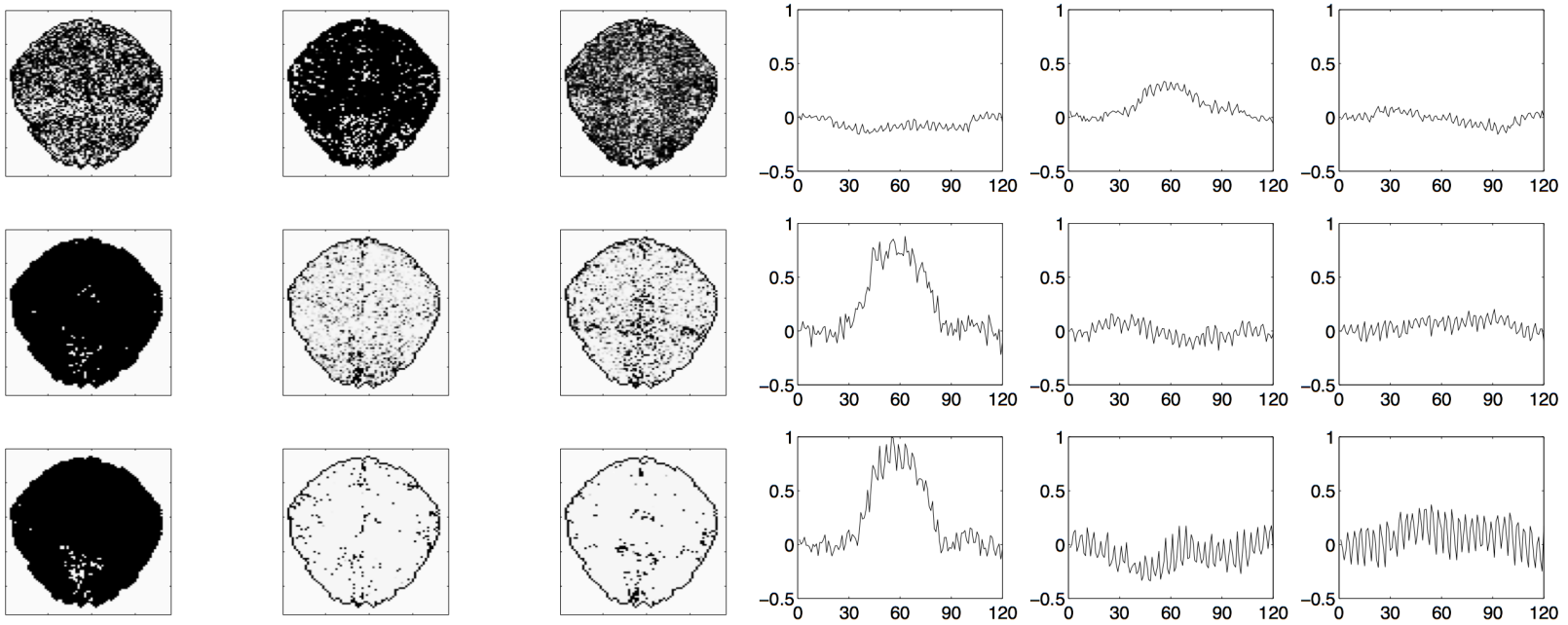


left: unconstrained **right:** constrained (positive) mixing matrix $\mathbf{A}$.

$x_i(t) =$ sequence of 500 images (handwritten '3's).$p(s) = e^{-s}$, $s \geq 0$.
Shown are the $m = 25$ columns $A_{\bullet j}$ of the matrix $\mathbf{A}$.

Functional Magnetic Resonance Imaging (fMRI) from: Højen–Sørensen, Hansen & Winther.

left: Posterior mean sources **right:** responses $A_{\bullet i}$ for $i = 1, \ldots, 9$.



Computing the Likelihood

Assume no noise and $d = m$

- Assume all n data are independent (no temporal structure):

$$p(D|\mathbf{A}) = \prod_{t=1}^{n} p(\mathbf{x}(t)|\mathbf{A})$$

- Look at a single data point: $p(\mathbf{x}|\mathbf{A}) = \int d\mathbf{S}\, p(\mathbf{x}|\mathbf{A},\mathbf{S})\, p(\mathbf{S})$

  with $p(\mathbf{S}) = \prod_{i=1}^{d} p_i(s_i)$ (ICA assumption) and

  $p(\mathbf{x}|\mathbf{A},\mathbf{S}) = \prod_{k=1}^{d} \delta\left(x_k - (\mathbf{AS})_k\right)$ Dirac - δ distributions (i.e. no noise).

The Likelihood cont'd

$$p(\mathbf{x}|\mathbf{A}) = \int d\mathbf{S}\; p(\mathbf{x}|\mathbf{A},\mathbf{S})\; p(\mathbf{S}) = \frac{1}{|\det \mathbf{A}|} \prod_{i=1}^{d} p_i((\mathbf{A}^{-1}\mathbf{x})_i)$$

With $\mathbf{W} = \mathbf{A}^{-1}$, we get for the negative log-likelihood

$$-\ln p(D|\mathbf{W}) = -n \ln |\det \mathbf{W}| - \sum_t \sum_i \ln p_i((\mathbf{W}\mathbf{x}(t))_i)$$

which must be minimized with respect to the matrix $\mathbf{W}$.

Modeling the sources

- Relation to PCA

  Let $\mathbf{U}$ matrix of eigenvectors of covariance matrix, i.e. $\mathbf{\Sigma U} = \mathbf{U\Lambda}$. If we set $\mathbf{W} = \mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{U}^T$, then the vector

  $\mathbf{Wx} \doteq \mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{U}^T\mathbf{x}$ has decorrelated components with unit variance.

  For Gaussian signals: decorrelated = independent!

  BUT any $\mathbf{QW}$ with orthogonal $\mathbf{Q}$ (i.e. $\mathbf{QQ}^T = \mathbf{I}$) will also decorrelate the signal: Estimation of "true" mixing matrix impossible for Gaussian signals/sources. Rotating a spherical Gaussian doesn't change its shape!

- Hence, *assume* non-Gaussian sources like e.g.

  the **super–Gaussian** $p_i(s) \propto \frac{1}{e^s + e^{-s}}$.

Disadvantages of Simple Model

- Noise ?
- Constraints on Mixing Matrix (positivity) ?
- Number of sources $\neq$ number of sensors ?
- How many sources are enough ?

Other approaches I: Minimize Mutual Information

Goal: Find $\mathbf{W}$ such that $\mathbf{S} \doteq \mathbf{W}\mathbf{x}$ has independent components.

Minimize Mutual information

$$I = \int d\mathbf{S}\, p(\mathbf{S}) \ln \frac{p(\mathbf{S})}{\prod_{i=1}^{m} p_i(s_i)}$$

with respect to $\mathbf{W}$. Problem: Find good estimate for I from data sample $\mathbf{x}(1), \mathbf{x}(2), \ldots, \mathbf{x}(T)$.

Practical Solutions:

- Approximate I using low order cumulants.
- Assume source model, eg $p(s) = \frac{1}{\pi \cosh(s)}$ - equivalent to Maximum Likelihood (Bell & Sejnowski, Cardoso & Laheld, MacKay)

Other approaches II: Non − Gaussianity

Mixing sources $\sim \sum$ of independent random variables $\sim$ Gaussian distribution.

Demixing Make distribution $p(S)$ of $S \doteq \mathbf{Wx} = \mathbf{WAx}$ as non Gaussian as possible!

Possible 'contrast functions' for Minimization

- Higher Cumulants such as kurt$(s) \doteq E[s^4] - 3(E[s^2])^2$ (Hyvärinen's *FastICA*)

- 'Negentropy': $H_{Gauss} - H[\mathbf{S}]$.

More approaches

- Use temporal structure
- Kernel ICA
- ...

The ICA model was an example of

Latent Variable Models

- Simple models (like exponential families) allow for simple analytic parameter estimation by Maximum Likelihood.

- More complex models explain data by hidden (unobserved) variables, the so called latent variables. Such models are very useful in practice.

- However, even Maximum Likelihood (ML) estimation can become a hard computational task.

Overview

- Latent variable models: Definition
- Examples
- ML with the EM Algorithm

Latent variable Models: Definition

$\mathbf{y} =$ observed variables.

$\boldsymbol{\theta} = (\boldsymbol{\theta}_{\mathbf{y}}, \boldsymbol{\theta}_{\mathbf{x}})$ sets of parameters.

$\mathbf{x} =$ latent, unobserved variables.

Total likelihood

$$p(\mathbf{y}|\boldsymbol{\theta}) = \sum_{\mathbf{x}} p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}_{\mathbf{y}}) p(\mathbf{x}|\boldsymbol{\theta}_{\mathbf{x}})$$

If the $\mathbf{x}$'s would be known, ML would often be easy!

Example I: Mixtures of Gaussians

Model for multimodal densities

$$p(y|\{\mu_c, \sigma_c, p(c)\}_{c=1}^K) = \sum_c p(c) \frac{1}{\sqrt{2\pi\sigma_c^2}} \exp\left[-\frac{(y-\mu_c)^2}{2\sigma_k^2}\right]$$
$$\equiv \sum_c p(c) p(y|c, \boldsymbol{\theta})$$

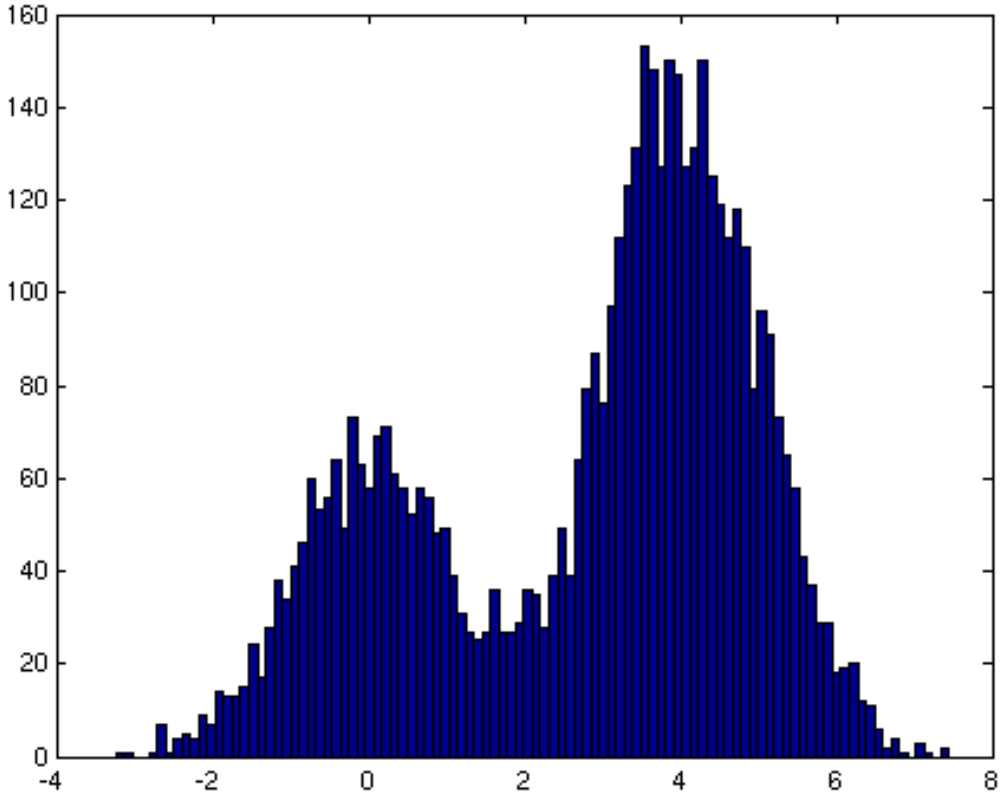
Total likelihood $p(D|\boldsymbol{\theta}) = \prod_i p(y_i|\boldsymbol{\theta})$

y_i observed, component c_i hidden,

$\boldsymbol{\theta} = \{\mu_c, \sigma_c, p(c)\}_{c=1}^K$ parameters to be estimated by ML.

Take $\nabla_{\boldsymbol{\theta}} \ln p(D|\boldsymbol{\theta}) = 0$ results in complicated set of nonlinear equations.

Data from a mixture of 2 Gaussians



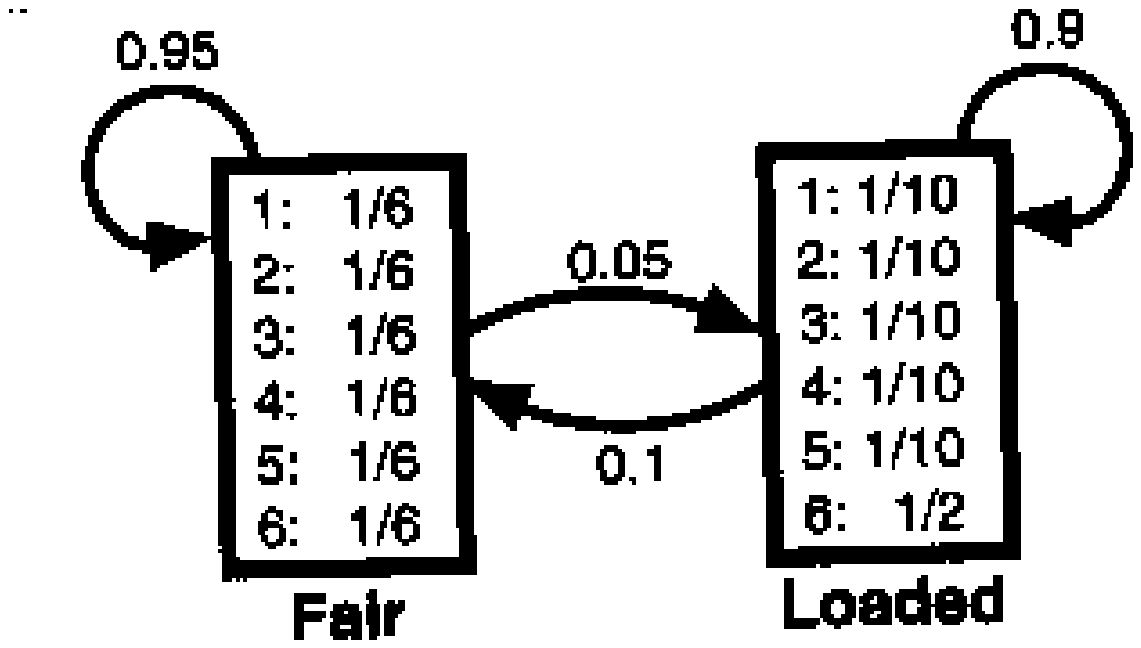
Example II: Hidden Markov Models

Modelling dependencies in one dimensional data structures, eg

- Speech recognition (Word models etc)
- Biosequences (DNA, proteins)

Example: The occasional dishonest casino (Durbin et al)

The HMM



Hidden Markov Models: Definitions

- Observations $\mathbf{y} = (y_1, y_2, \ldots, y_T)$ are *independent given* the sequence of states $\mathbf{S} = (s_1, s_2, \ldots, s_T)$. ie

$$P(\mathbf{y}|\mathbf{S}) = \prod_{i=1}^{T} P(y_i|s_i) = \prod_{i=1}^{T} b_{s_i}(y_i)$$

  with the matrix of *emission probabilities* $b_k(l) = P(y = l|s = k)$.

- States are not observed (hidden) and generated from a *Markov chain*

$$P(\mathbf{S}) = \pi_{s_1} P(s_2|s_1) P(s_3|s_2) \ldots P(s_T|s_{T-1}) \ .$$

- The total probability of the observed sequences is obtained by marginalization of the joint probability $P(\mathbf{y}, \mathbf{S}) = P(\mathbf{y}|\mathbf{S})P(\mathbf{S})$ over the states

$$P(\mathbf{y}) = \sum_{\mathbf{S}} P(\mathbf{y}|\mathbf{S}) P(\mathbf{S})$$

  For N states, there are N^T different paths in the sum!!

Probabilistic ICA (IFA –Independent Factor Analysis)

Use probabilities for everything unknown

$$\mathbf{y}(t) = \mathbf{A}\mathbf{S}(t) + \mathbf{\Gamma}(t)$$

Probability of observations (given the sources) for Gaussian noise $\mathbf{\Gamma}$

$$p(\mathbf{y}|\mathbf{S}, \mathbf{A}, \mathbf{\Sigma}) = (2\pi \det \mathbf{\Sigma})^{-d/2} \; e^{-\frac{1}{2}(\mathbf{y}-\mathbf{A}\mathbf{S})^T \mathbf{\Sigma}^{-1} (\mathbf{y}-\mathbf{A}\mathbf{S})} \; .$$

Prior density model of sources

$$p(\mathbf{S}) = \prod_{i=1}^{m} p_i(s_i)$$

Complete Data Likelihood for n datapoints $\{\mathbf{y}\}_{i=1}^n$

$$p(\{\mathbf{y}\}_{i=1}^n | \mathbf{A}, \mathbf{\Sigma}) = \prod_{i=1}^{n} \int d\mathbf{S} \; p(\mathbf{y}_i | \mathbf{S}, \mathbf{A}, \mathbf{\Sigma}) \; p(\mathbf{S})$$

The Expectation–Maximisation (EM) Algorithm

1. Start with arbitrary $\boldsymbol{\theta}_0$

   Iterate:

2. (E-Step): Compute the expectation

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\theta}_t) \equiv \sum_{\mathbf{x}} p(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta}_t) \ln p(\mathbf{y}, \mathbf{x}, \boldsymbol{\theta})$$

   with the **posterior probability** (given the observations) of the latent variables

$$p(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta}_t) = \frac{p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}_t) p(\mathbf{x}|\boldsymbol{\theta}_t)}{p(\mathbf{y}|\boldsymbol{\theta}_t)}$$

3. (M–Step) Maximise

$$\boldsymbol{\theta}_{t+1} = \operatorname*{argmax}_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\theta}_t)$$

Claim: $\ln p(\mathbf{y}|\boldsymbol{\theta}_{t+1}) \geq \ln p(\mathbf{y}|\boldsymbol{\theta}_t)$ Likelihood is not decreasing!

Analysis of EM

The proof requires the *Kullback–Leibler divergence* which fulfils

$$KL(q,p) = \sum_{\mathbf{x}} q(\mathbf{x}) \ln \frac{q(\mathbf{x})}{p(\mathbf{x}|\mathbf{y},\boldsymbol{\theta})} \geq 0 \ .$$

for any $q(\mathbf{x})$. By rearranging we get

$$-\ln p(\mathbf{y}|\boldsymbol{\theta}) \leq F(q,\theta) \equiv \sum_{\mathbf{x}} q(\mathbf{x}) \ln \frac{q(\mathbf{x})}{p(\mathbf{y},\mathbf{x}|\boldsymbol{\theta})}$$

For fixed $\boldsymbol{\theta}$, the right is minimal (equality!!!) if $q(\mathbf{x}) = p(\mathbf{x}|\mathbf{y},\boldsymbol{\theta})$.

Let $q_t(\mathbf{x}) \doteq p(\mathbf{x}|\mathbf{y},\boldsymbol{\theta}_t)$, then $\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\theta}_t) = -F(q_t,\theta) + \sum_{\mathbf{x}} q_t(\mathbf{x}) \ln q_t(\mathbf{x})$

Hence, the EM algorithm can be reformulated as:

1. E–Step: Minimise $F(q,\boldsymbol{\theta}_t)$ w.r.t $q \to q_t(\mathbf{x})$ and compute $\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\theta}_t)$.

2. M–Step Minimise $F(q_t,\boldsymbol{\theta}) = -\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\theta}_t) + \sum_{\mathbf{x}} q_t(\mathbf{x}) \ln q_t(\mathbf{x})$ w.r.t. $\boldsymbol{\theta}$.

We get

$$-\ln p(\mathbf{y}|\boldsymbol{\theta}) \leq F(q_t, \theta)$$

and

$$-\ln p(\mathbf{y}|\boldsymbol{\theta}_t) = F(q_t, \theta_t)$$

Hence,

$$\ln p(\mathbf{y}|\boldsymbol{\theta}_{t+1}) - \ln p(\mathbf{y}|\boldsymbol{\theta}_t) \geq -F(q_t, \theta_{t+1}) + F(q_t, \theta_t) \geq 0$$

Likelihood is not decreasing!

Example: Mixture of Gaussians

- (E-Step): Compute

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\theta}_t) \equiv \sum_{\mathbf{c}} p(\mathbf{c}|\mathbf{y}, \boldsymbol{\theta}_t) \ln \left\{ \prod_i p(y_i, c_i|\boldsymbol{\theta}) \right\}$$

  with

$$p(\mathbf{c}|\mathbf{y}, \boldsymbol{\theta}_t) = \prod_i p(c_i|y_i, \boldsymbol{\theta}_t) = \prod_i \frac{p(y_i|c_i, \boldsymbol{\theta}_t) p(c_i|\boldsymbol{\theta}_t)}{p(y_i|\boldsymbol{\theta}_t)}$$

  and

$$p(y_i, c_i, \boldsymbol{\theta}) = p(y_i|c_i, \boldsymbol{\theta}) p(c_i|\boldsymbol{\theta})$$

- (M–Step) Update $\boldsymbol{\theta}_{t+1} = \text{argmax}_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\theta}_t)$

Explicit Formulas

- Variation with respect to μ_c

$$\sum_i (y_i - \mu_c) p(c|y_i, \boldsymbol{\theta}_t) = 0 \ \rightarrow \mu_{c,t+1} = \frac{\sum_i y_i \, p(c|y_i, \boldsymbol{\theta}_t)}{\sum_i p(c|y_i, \boldsymbol{\theta}_t)}$$

- Variation with respect to σ_c^2

$$\sigma_{c,t+1}^2 = \frac{\sum_i (y_i - \mu_{c,t+1})^2 \, p(c|y_i, \boldsymbol{\theta}_t)}{\sum_i p(c|y_i, \boldsymbol{\theta}_t)}$$

- Variation with respect to $p_{t+1}(c) = p(c|\boldsymbol{\theta}_{t+1})$

$$p_{t+1}(c) \equiv p(c|\boldsymbol{\theta}_{t+1}) = \frac{1}{n} \sum_i p(c|y_i, \boldsymbol{\theta}_t)$$

Low dimensional representations

Observations $\mathbf{y} \in R^d$ live effectively on lower dimensional manifold (+ noise). Introduce latent variables $\mathbf{x} \in R^q \sim \mathcal{N}(0, \mathbf{I})$

Factor analysis:

$$\mathbf{y} = \mathbf{W}\mathbf{x} + \boldsymbol{\mu} + \mathbf{u}$$

with $\mathbf{W} = d \times q$, $\mathbf{x} \sim \mathcal{N}(0, \mathbf{I})$ $\mathbf{u} \sim \mathcal{N}(0, \mathbf{D})$ and $\mathbf{D}$ diagonal.

Probabilistic PCA (Tipping & Bishop)

$$\mathbf{y} = \mathbf{W}\mathbf{x} + \boldsymbol{\mu} + \mathbf{u}$$

with $\mathbf{u} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$.

We have

$$p(\mathbf{x}) \propto \exp\left[-\frac{1}{2}||\mathbf{x}||^2\right]$$

and

$$p(\mathbf{y}|\mathbf{x}) \propto \exp\left[-\frac{1}{2\sigma^2}||\mathbf{y} - (\mathbf{W}\mathbf{x} + \boldsymbol{\mu})||^2\right]$$

Posterior of latent variables

$$p(\mathbf{x}|\mathbf{y}) \propto \exp\left[-\frac{1}{2\sigma^2}||\mathbf{y}-(\mathbf{W}\mathbf{x}+\boldsymbol{\mu})||^2-\frac{1}{2}||\mathbf{x}||^2\right] \propto$$
$$\exp\left[-\frac{1}{2\sigma^2}\left(\mathbf{x}-\mathbf{M}^{-1}\mathbf{W}^T(\mathbf{y}-\boldsymbol{\mu})\right)^T\mathbf{M}\left(\mathbf{x}-\mathbf{M}^{-1}\mathbf{W}^T(\mathbf{y}-\boldsymbol{\mu})\right)\right]$$

with

$$\mathbf{M}=(\sigma^2\mathbf{I}+\mathbf{W}^T\mathbf{W})$$

Full probability of data

$$p(\mathbf{y}) \propto \exp\left[-\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu})^T\mathbf{C}^{-1}(\mathbf{y}-\boldsymbol{\mu})\right]$$

with

$$\mathbf{C}=\mathbf{W}\mathbf{W}^T+\sigma^2\mathbf{I}$$

Maximum Likelihood:

Minimise

$$-\ln p(\mathbf{Y}) = -\sum_{i=1}^{n} \ln p(\mathbf{y}_i) = \text{const} + \frac{n}{2}\left(\ln|\mathbf{C}| + \text{trace}(\mathbf{C}^{-1}\mathbf{S})\right)$$

with respect to $\mathbf{W}$, $\boldsymbol{\mu}$ and σ^2.

$\mathbf{S}$ is the empirical **data covariance**

$$\mathbf{S} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{y}_i - \boldsymbol{\mu})(\mathbf{y}_i - \boldsymbol{\mu})^T$$

One can show (not surprisingly) that

$$\boldsymbol{\mu}_{ML} = \bar{\boldsymbol{\mu}} = \frac{1}{N}\sum_{i}\mathbf{y}_i$$

One can show that optimality is achieved for

$$\mathbf{W} = \mathbf{U}_q(\mathbf{\Lambda}_q - \sigma^2\mathbf{I})^{1/2}\mathbf{R}$$

$\mathbf{U}$ contains the q PCs with eigenvalues in the diagonal $\mathbf{\Lambda}$ of the data covariance $\mathbf{\Sigma}$. $\mathbf{R}$ is an arbitrary orthogonal $(q \times q)$ matrix.

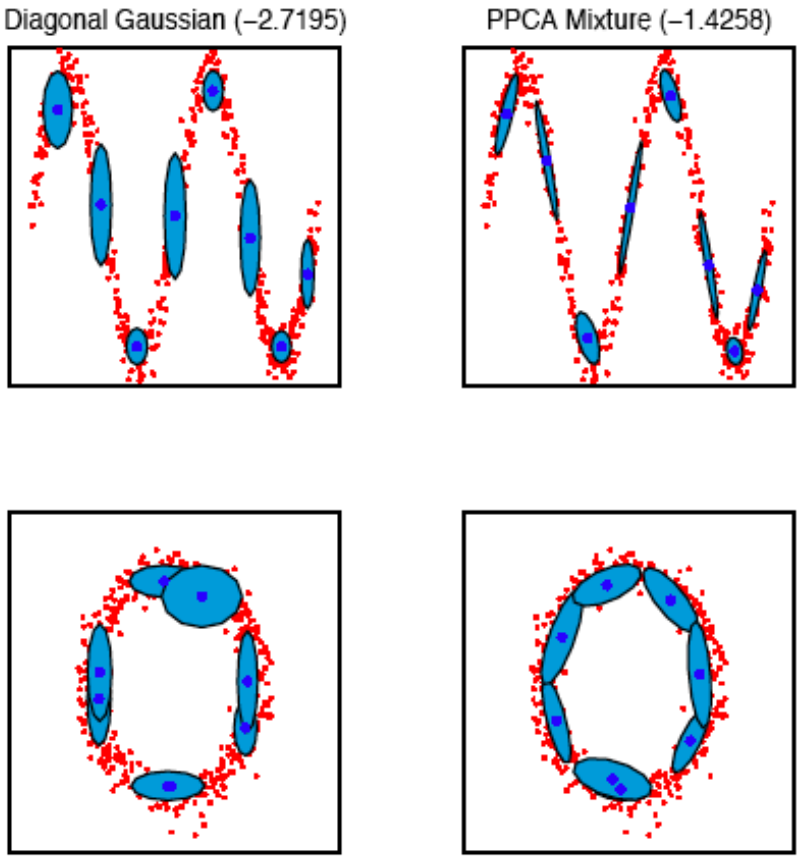
Advantages over conventional PCA

- One can use EM algorithm (in certain cases computationally more efficient)
- can treat missing values
- PPCA can be extended to mixtures of PPCA using

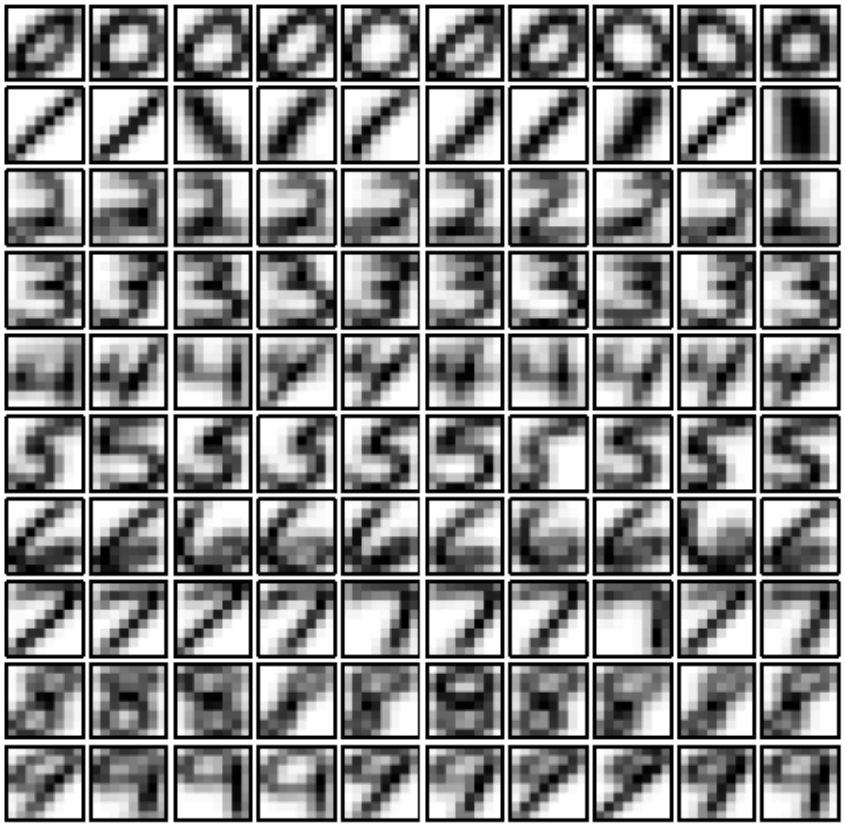
  $$p(\mathbf{y}) = \sum_k p_k p(\mathbf{y}|k)$$

  where $p(\mathbf{y}|k)$ is given by PPCA.

- Can be extended to Bayesian treatment (optimal model order)
- PPCA is a can be used for modelling class conditional densities (classification)
- Likelihood can be used for comparison with other density models



8: Comparison of an 8-component diagonal variance Gaussian mixture model with a mixture of PPCA model. The upper two plots give a view perpendicular to the major axis of the spiral, while the lower two plots show the end elevation. The covariance structure of each mixture component is shown by projection of a unit Mahalanobis distance ellipse and the log-likelihood per data-point is given in brackets above the figures.



igure 9: The mean vectors $\boldsymbol{\mu}_i$, illustrated as gray-scale digits, for each of the ten digit models. The model for a given digit is a mixture of ten PPCA models, one centred at each of the pixel vectors shown on the corresponding row. Note how different components can capture different styles of digit.

Image compression **Bishop & Tipping**

720×360 pixel image segmented into 8×8 non-overlapping blocks $\rightarrow$ dataset of 4050 64 dim vectors.

Single PCA $q = 4$ versus mixtures of PPCA (12 mixing components, $q = 4$), left half of image used for training. Compress by quantising transform variable and component label.

Figure 5: The original image (left), and detail therein (right).

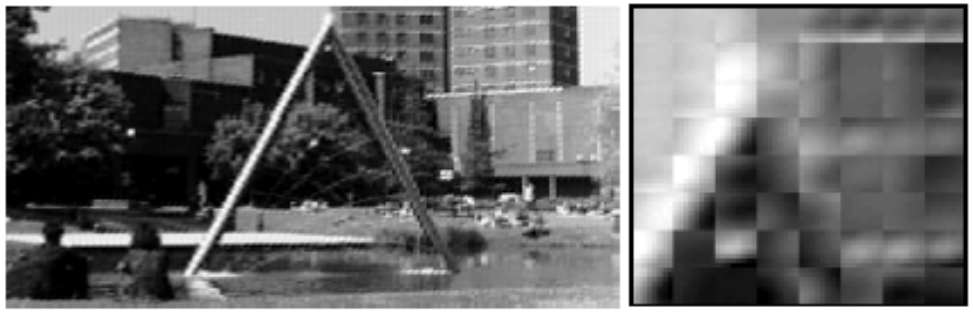


Figure 6: The PCA reconstructed image, at 0.5 bits-per-pixel.

Figure 7: The mixture of PPCA reconstructed image, using the same bit-rate as Figure 6.

Latent variable models for data visualisation

Visualise high (d) dim. data $\mathbf{y}$ in low (~ 2) dim data space $\mathcal{H}$ using latent variables $\mathbf{u}$.

Generative Topographic Mapping (GTM) (Bishop, Svenson & Williams)

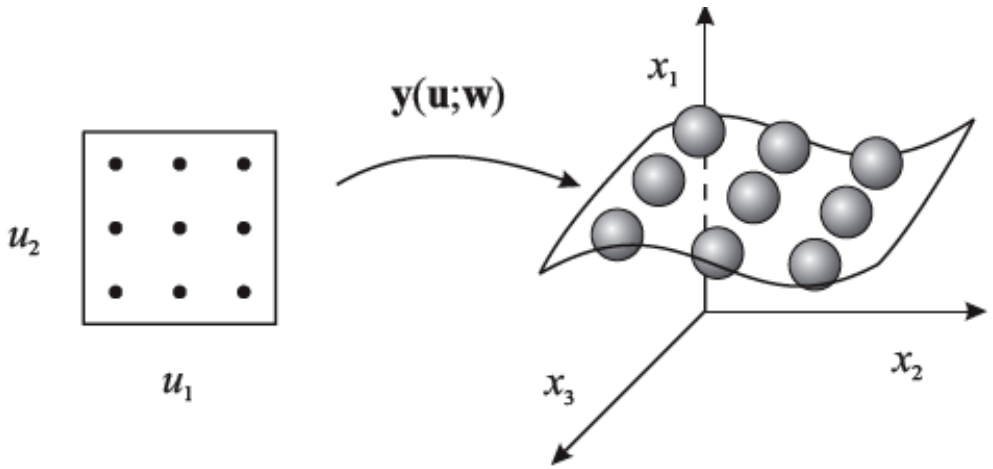
$$p(\mathbf{y}|\mathbf{u},\mathbf{W}) = \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\left\{\frac{-||\mathbf{f}(\mathbf{u},\mathbf{W}) - \mathbf{y}||^2}{2\sigma^2}\right\}$$

Latent variables $\mathbf{x}$ are assumed to be on a **discrete grid** with $p(\mathbf{u}) = \frac{1}{K}\sum_k \delta(\mathbf{u} - \mathbf{u}_k)$. $\mathbf{f}$ is a smooth mapping, e.g. $\mathbf{f}(\mathbf{u},\mathbf{W}) = \mathbf{W}\boldsymbol{\phi}(\mathbf{u})$ with fixed nonlinear (e.g. radial) basis functions $\boldsymbol{\phi}$ and a $d \times M$ matrix $\mathbf{W}$ to be optimised.

The total probability is

$$p(\mathbf{y}|\mathbf{W}) = \frac{1}{K}\sum_k p(\mathbf{y}|\mathbf{u}_k,\mathbf{W})$$

Projection of data points: Use mean or mode of posterior $p(\mathbf{u}|\mathbf{y})$.


y(u;w)
u_2
u_1
x_1
x_2
x_3

Latent trait models

(Kabán, Girolami)

Replace Gaussians by more general exponential families. Helps e.g. to visualise discrete data.

$$p(\mathbf{y}|\mathbf{u},\mathbf{W}) = p_0(\mathbf{y}) \exp\left\{\mathbf{y}\cdot\mathbf{f}_{\mathbf{W}}(\mathbf{u}) - g(\mathbf{f}_{\mathbf{W}}(\mathbf{u}))\right\}$$

with a nonlinear mapping $\mathbf{f}_{\mathbf{W}}$ from latent space to data.

Example 1: Bernoulli distribution for binary data

Let $\mathbf{y} = (y_1, \ldots, y_d) \in \{0,1\}^d$. Then we define $m_k = \text{sigmo}((\mathbf{W}\boldsymbol{\phi}(\mathbf{u}))_k)$ with $\text{sigmo}(z) = \frac{e^z}{1+e^z}$ Finally:

$$p(\mathbf{y}|\mathbf{u},\mathbf{W}) = \prod_{k=1}^{d} m_k^{y_k} \left(1-m_k\right)^{1-y_k}$$

Example II: Multinomial Model

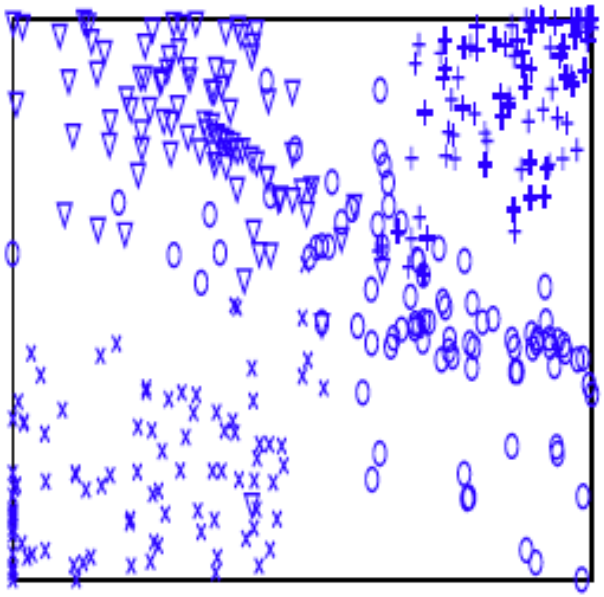
Here, $y_k \in N_0$. We set $m_k = \frac{\exp[(\mathbf{W}\boldsymbol{\phi}(\mathbf{u}))_k)]}{\sum_{k'=1}^{d} \exp[(\mathbf{W}\boldsymbol{\phi}(\mathbf{u}))_{k'})]}$

$$p(\mathbf{y}|\mathbf{u}, \mathbf{W}) = \prod_{k=1}^{d} m_k^{y_k}$$

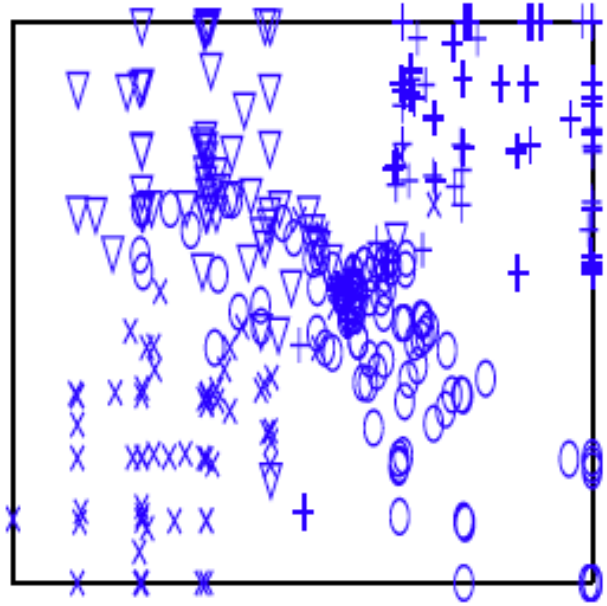
Application:

E.g. in text mining, where for the Bernoulli case $y_k = 1$ indicates that term k is present in a document. The multinomial case is represented by a histogramme of word occurrences. The conditional model represents independent samples from a 'bag of words'. The order of words is irrelevant.

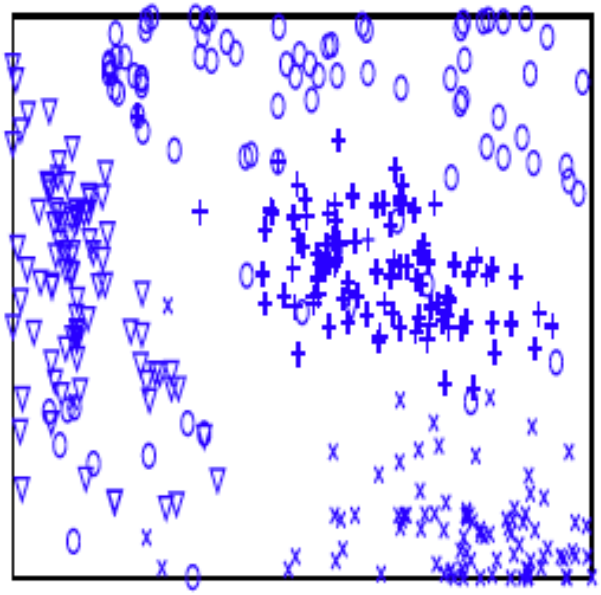
Bernoulli



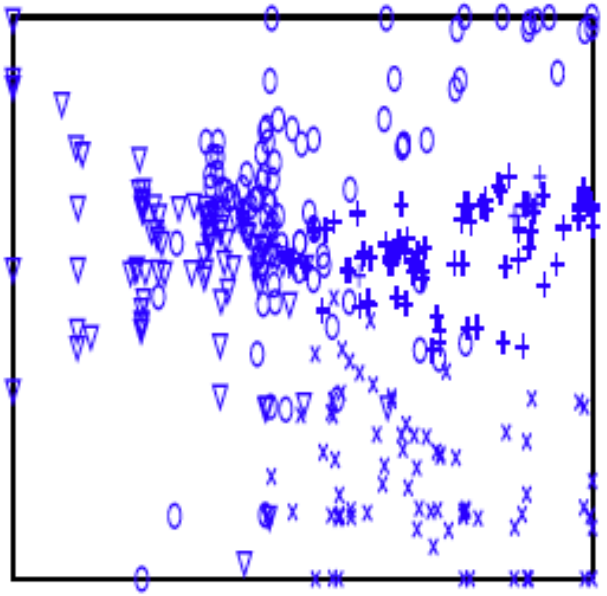
Gaussian on binary data

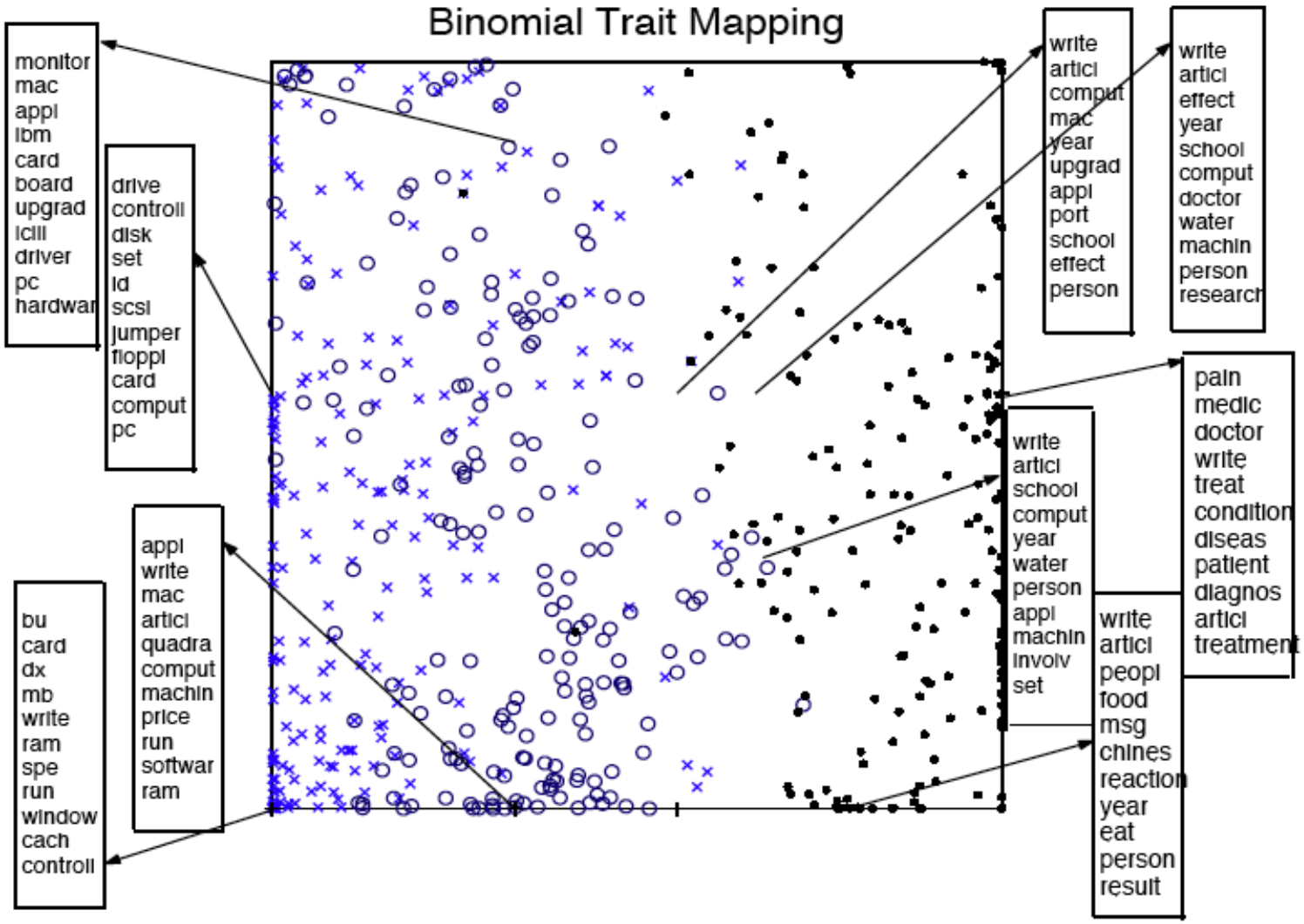


Multinomial

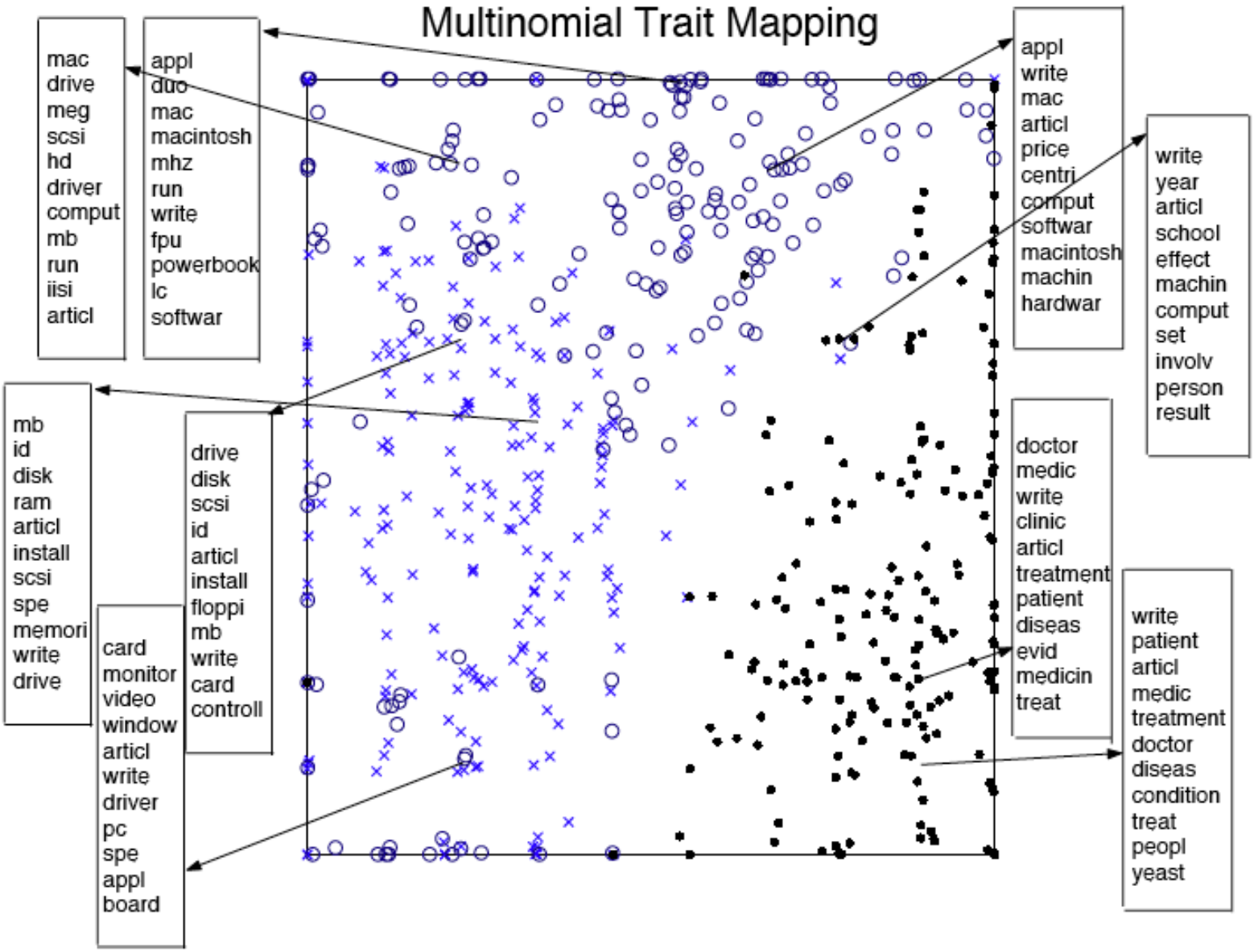


Gaussian on normalized freq. data





$\times$ = `comp.sys.ibm.pc.hardware`, 0 = `comp.sys.mac.hardware`, $\cdot$ = `sci.med`



$\times$ = `comp.sys.ibm.pc.hardware`, 0 = `comp.sys.mac.hardware`, $\cdot$ = `sci.med`

The Bayesian approach to statistics: Basics

For Bayesians, all prior knowledge (or lack of) about unknown parameters should be described by a probability density.

Back to the biased coin

The Bayesian statistician may assume that his **lack of knowledge** (or **prior belief**) about θ **before** she/he has seen the data, should be represented by a prior distribution. Take eg

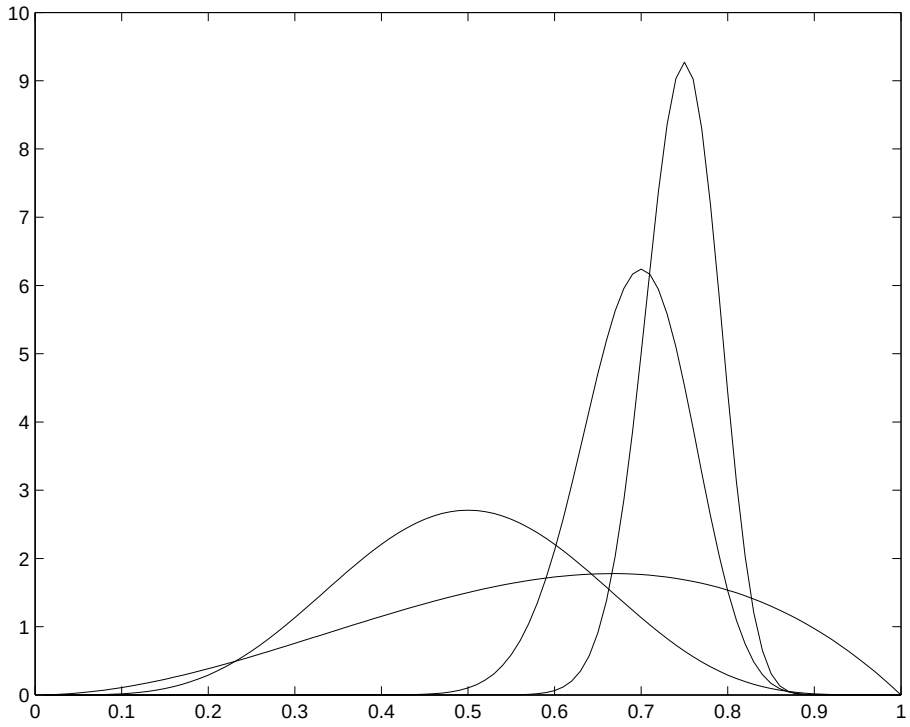
$$p(\theta) = 1 \qquad \text{for } 0 \leq \theta \leq 1 \ .$$

The information **from the data** is described by the likelihood $P(D|\theta)$. Using **Bayes rule**, we compute the **posterior distribution** which gives our belief about θ **after** seeing the data

$$p(\theta|D) = \frac{P(D|\theta)p(\theta)}{P(D)}$$

with the **evidence**

$$P(D) = \int_0^1 P(D|\theta)\ p(\theta)\ d\theta \ .$$



Posterior density of θ for the biased coin for $n = 3, 10, 50, 100$. The true value under which the data were generated was $\theta = 0.7$.

Estimators:

A reasonable estimate for the unknown parameter could be the **MAP value** for θ, ie the value which has the **Ma**ximum **Posterior** probability (density). For our choice of prior, this coincides with the ML value.

Another estimator is the the **posterior** mean of θ which is given by

$$\widehat{\theta}_{pm} = \int_0^1 \theta \; p(\theta|D) \; d\theta = \frac{n_1 + 1}{n + 2}$$

$\widehat{\theta}_{pm}$ minimises the **loss function**

$$L_2(\widehat{\theta}) = \int \left(\widehat{\theta} - \theta\right)^2 p(\theta|D) \; d\theta$$

For large n, we see that the posterior mean $\widehat{\theta}_{pm} \to \widehat{\theta}_{ML}$ and the **posterior variance** $\to 0$.

In general, the **Bayes optimal prediction** for the unknown distribution is the **predictive distribution**

$$p(x|D) = \int_{-\infty}^{\infty} p(x|\theta)p(\theta|D)d\theta$$

Properties of Bayes procedures

- Implements prior knowledge
- Regularises problem if small amount of data
- Simple approach to model selection, error bars
- Conceptually simple but often computationally hard
- Could be sensitive to wrong priors, but we can learn priors too!

Bayes for Gaussian densities: 1-D

We assume that σ^2 is known but μ is unknown. Use a (conjugate) prior

$$p(\mu) = \frac{1}{\sqrt{2\pi\sigma_0^2}}\, e^{-\frac{(\mu-\mu_0)^2}{2\sigma_0^2}}$$

This yields the posterior density

$$p(\mu|D) = \frac{p(D|\mu)p(\mu)}{p(D)} = \frac{p(\mu)}{p(D)} \prod_i \left\{ \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-\frac{(x_i-\mu)^2}{2\sigma^2}} \right\} = \frac{1}{\sqrt{2\pi\sigma_n^2}}\, e^{-\frac{(\mu-\mu_n)^2}{2\sigma_n^2}}$$

with

$$\begin{aligned} \mu_n &= \frac{n\sigma_0^2}{n\sigma_0^2+\sigma^2}\overline{x} + \frac{\sigma^2}{n\sigma_0^2+\sigma^2}\mu_0, \\ \frac{1}{\sigma_n^2} &= \frac{n}{\sigma^2} + \frac{1}{\sigma_0^2}, \end{aligned}$$

where $\overline{x}$ is the sample mean $\sum_i x_i/n$.

Conjugate priors

For exponential families, conjugate priors allow for simple computations:

$$p(\boldsymbol{\theta}|\boldsymbol{\tau}, n_0) \propto \exp\left[\boldsymbol{\psi}(\boldsymbol{\theta}) \cdot \boldsymbol{\tau} + n_0 g(\boldsymbol{\theta})\right]$$

In this case, the posterior will be of the same form:

$$p(\boldsymbol{\theta}|D\boldsymbol{\tau}, n_0) \propto \exp\left[\boldsymbol{\psi}(\boldsymbol{\theta}) \cdot (\sum_{i=1}^{n} \boldsymbol{\phi}(x_i) + \boldsymbol{\tau}) + (n + n_0) g(\boldsymbol{\theta})\right]$$

We simply replace $n_0 \to n_0 + n$ and $\boldsymbol{\tau} \to \sum_{i=1}^{n} \boldsymbol{\phi}(x_i) + \boldsymbol{\tau}$

Bayes Model selection

If we have a variety of models $\mathcal{M}_1$, $\mathcal{M}_2$, ... with different priors on parameters $p(\theta_1|\mathcal{M}_1)$, $p(\theta_2|\mathcal{M}_2)$, etc, the optimal thing would be a prior over models $P(\mathcal{M})$ and mix them all together. One may then calculate the posterior probability of a model

$$P(\mathcal{M}|D) = \frac{P(D|\mathcal{M})P(\mathcal{M})}{P(D)} = \frac{P(\mathcal{M}) \int P(D|\theta,\mathcal{M})p(\theta|\mathcal{M})d\theta}{P(D)}$$

and vote for the most likely one. For equal priors $P(\mathcal{M})$ we choose the model with the largest **evidence** $\int P(D|\theta,\mathcal{M})p(\theta|\mathcal{M})d\theta$.

Example: Bayesian polynomial regression

Assume data generated as $y_i = f(x_i) + \nu_i$ for $i = 1, \ldots, N$, with $f(\cdot)$ unknown, ν_i i.i.d. $\sim \mathcal{N}(0, \sigma^2)$.

Class of models: polynomials

$$f_{\mathbf{w}}(x) = \sum_{j=0}^{K} w_j x^j$$

allowing for different orders K. The **likelihood** is

$$p(D|\mathbf{w}) = \frac{1}{(2\pi\sigma^2)^{N/2}} \exp\left[-\sum_{i=1}^{N} \frac{(y_i - f_{\mathbf{w}}(x_i))^2}{2\sigma^2}\right]$$

Prior distribution on weights $p(\mathbf{w}) = \frac{1}{(2\pi\sigma_0^2)^{(K+1)/2}} \exp\left[-\frac{\sum_{j=0}^{K} w_j^2}{2\sigma_0^2}\right]$

Posterior density of the parameters $\mathbf{w}$ is given by

$$p(\mathbf{w}|D) = \frac{p(D|\mathbf{w})p(\mathbf{w})}{p(D)}$$

which is a multivariate Gaussian. The *evidence* of the data:

$$p(D) = \int p(D|\mathbf{w})\, p(\mathbf{w}) d\mathbf{w}$$

The posterior density is a multivariate Gaussian density with mean

$$E[\mathbf{w}|D] = \left(\frac{\sigma^2}{\sigma_0^2}\mathbf{I}_{K+1} + \mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y} \tag{8}$$

where the matrix elements of $\mathbf{X}$ are given by $X_{lk} = x_l^k$.

We can show that the evidence of the data is given by:

$$\ln p(D) = -\frac{N}{2}\ln 2\pi - \frac{1}{2}\ln|\mathbf{\Sigma}| - \frac{1}{2}\mathbf{y}^T\mathbf{\Sigma}^{-1}\mathbf{y} \ , \tag{9}$$

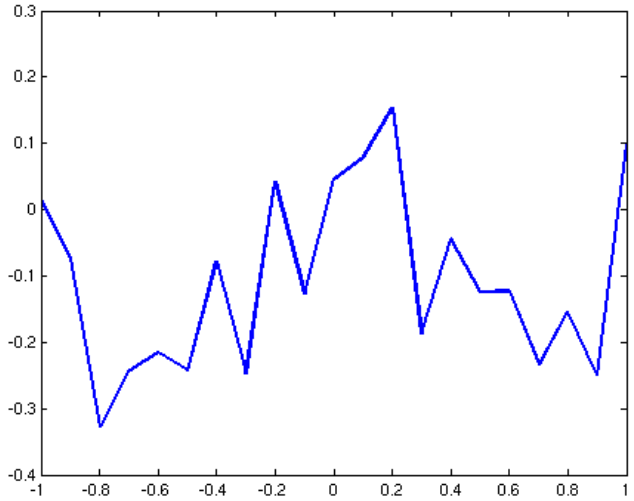
where

$$\mathbf{\Sigma} = \sigma_0^2\mathbf{X}\mathbf{X}^\top + \sigma^2\mathbf{I}_N \tag{10}$$

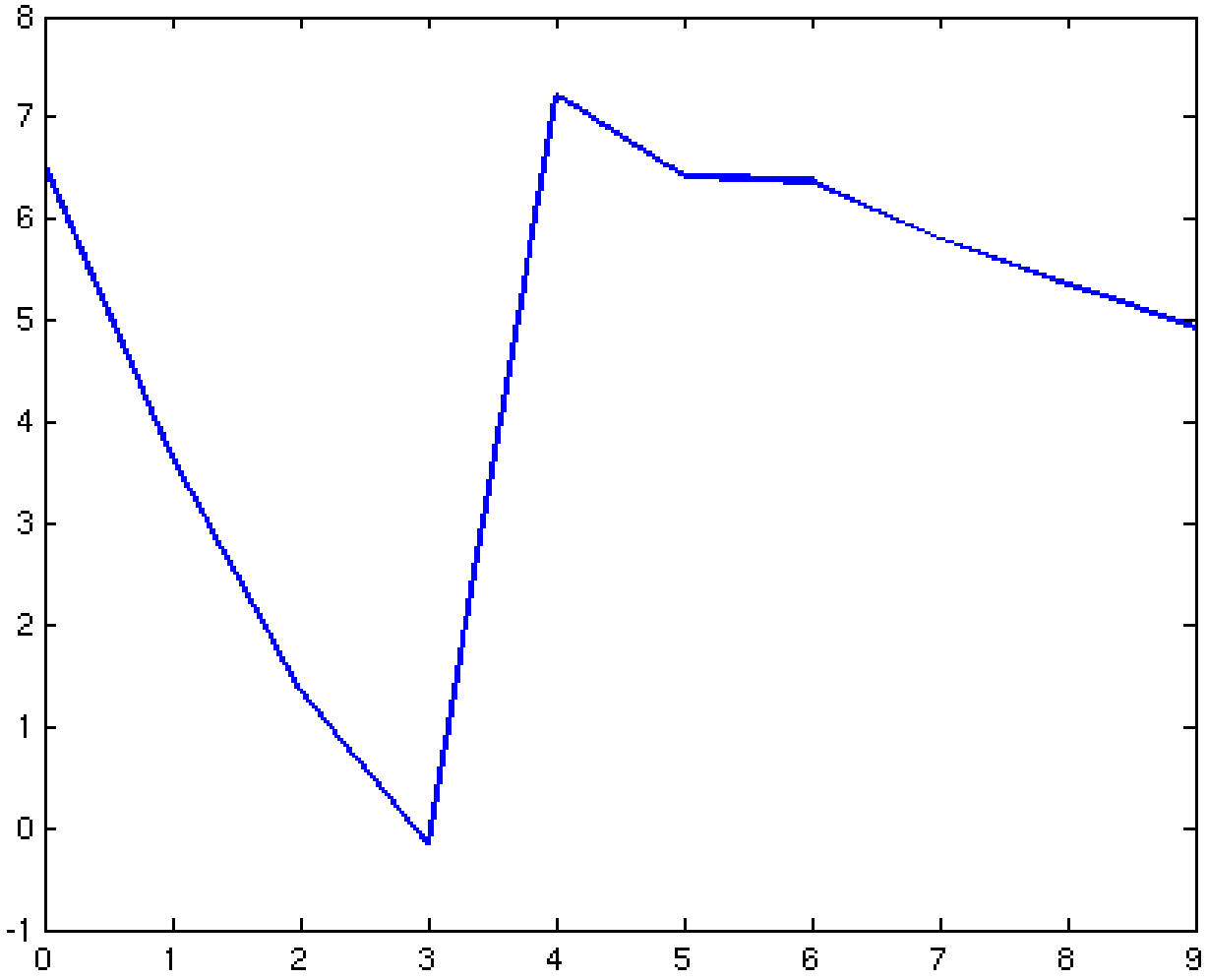
Experiment: $N = 21$ data-points y_i, equally spaced inputs x_i, with true $f(x) = x^4 - x^2$ and $\sigma^2 = 0.01$ in the interval $[-1, 1]$.

prior distribution with variance $\sigma_0^2 = 1$.

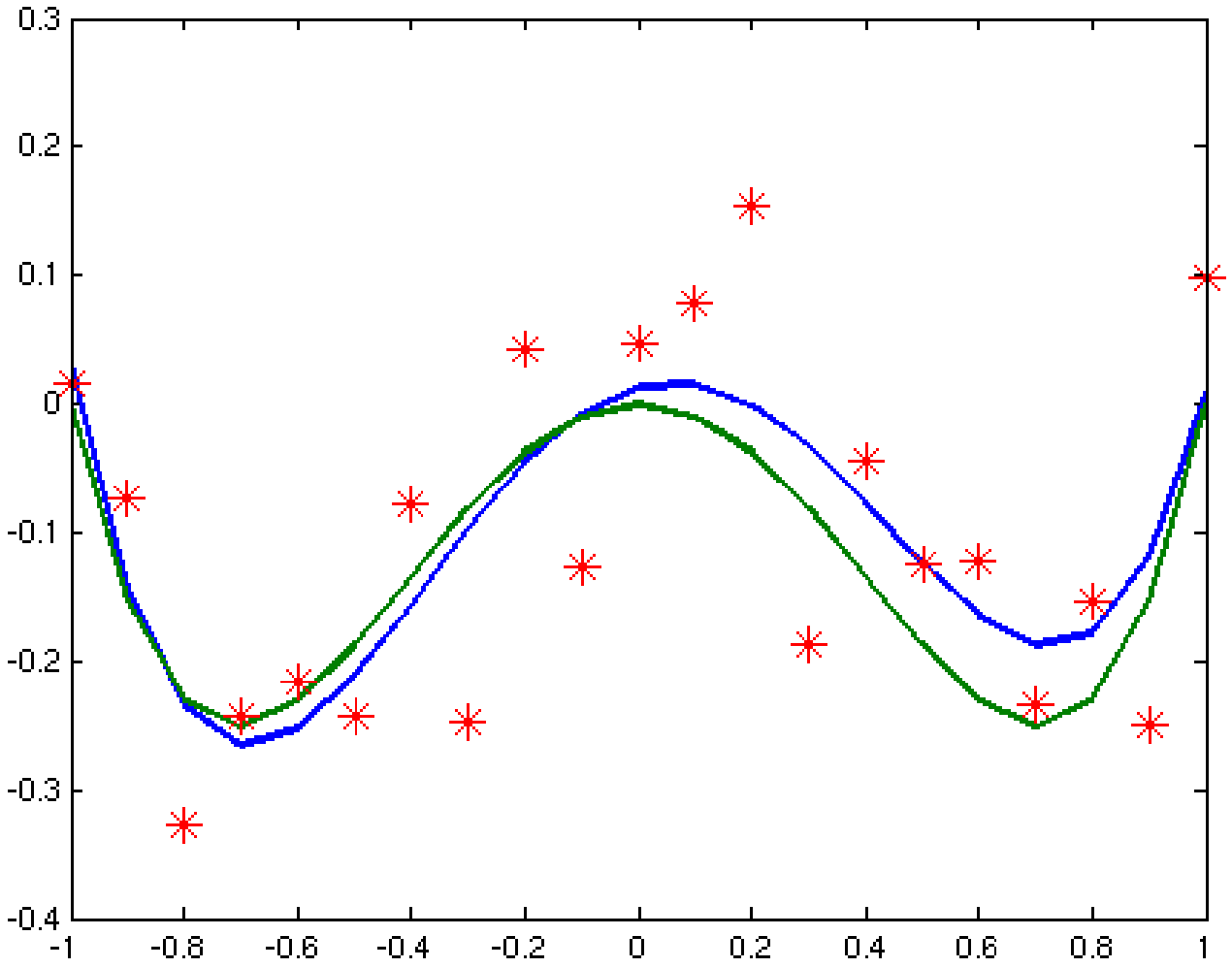
Typical observations



Log-evidence as function of K

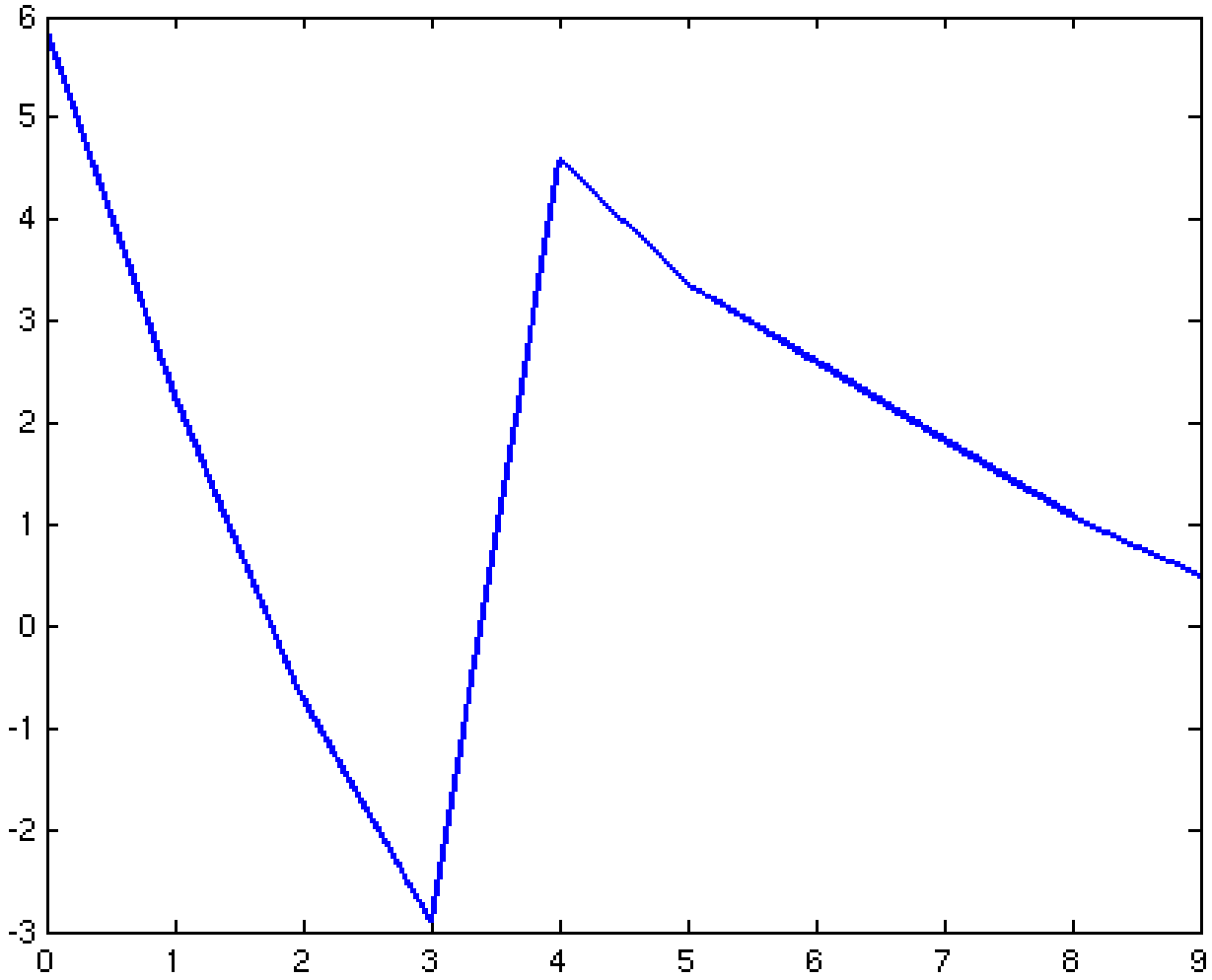


Reconstruction using posterior mean $E[\mathbf{w}|D] = \int d\mathbf{w}\ p(\mathbf{w}|D)\ f_{\mathbf{w}}(x)$

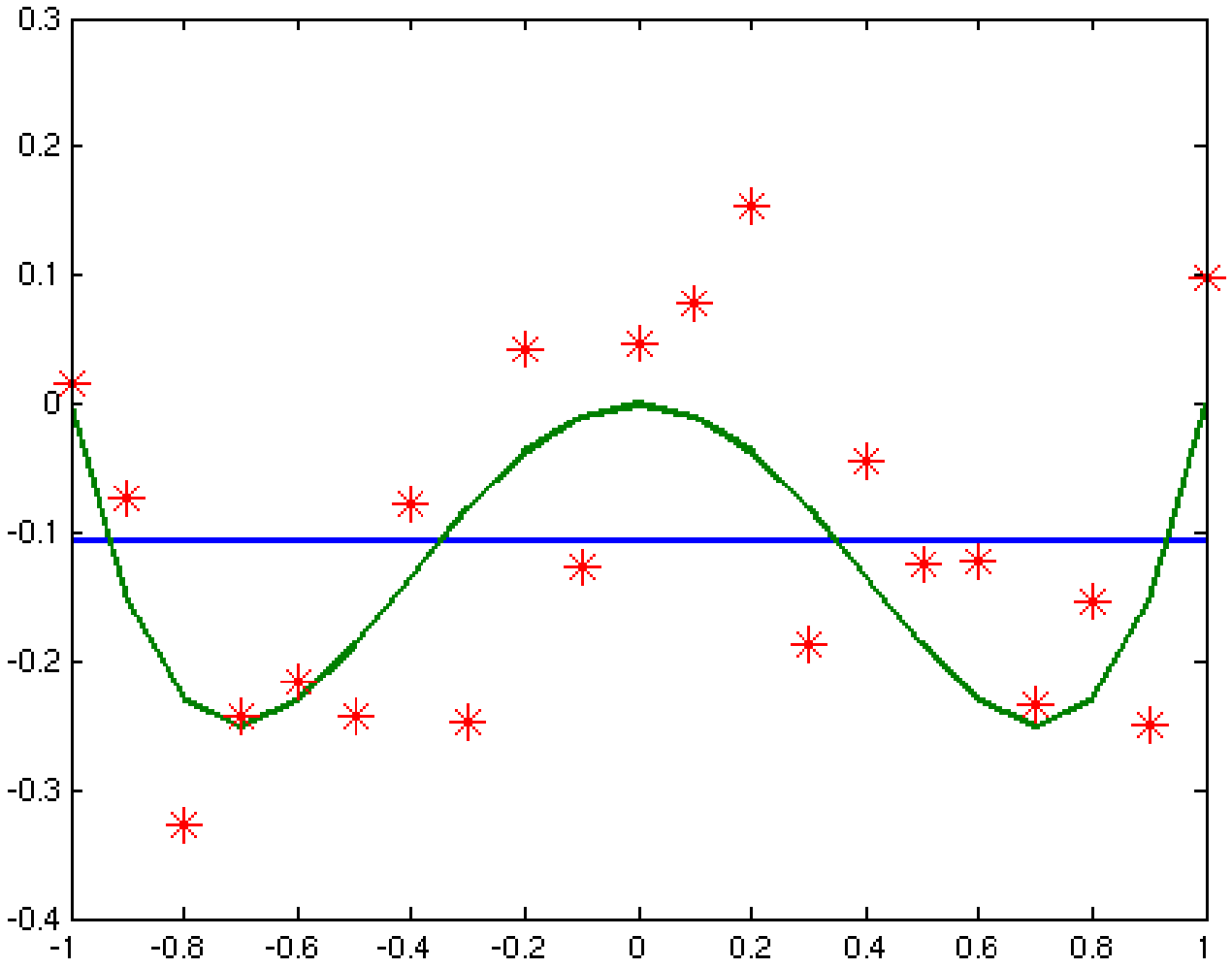


The same, but now with a different prior $\sigma_0 = 2$

Log-evidence as function of K

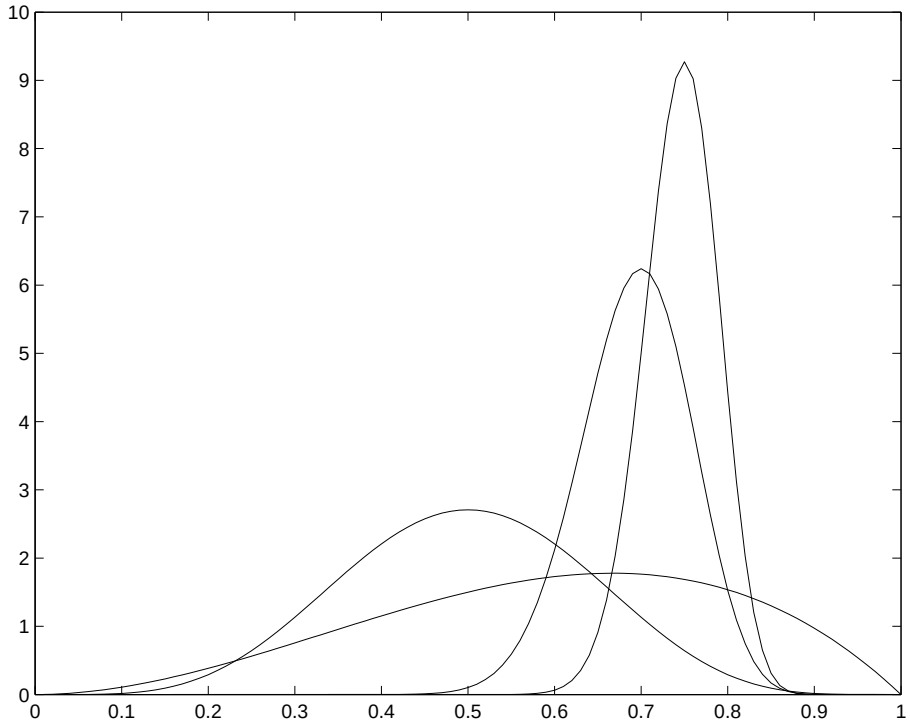


Reconstruction using posterior mean $E[\mathbf{w}|D] = \int d\mathbf{w}\ p(\mathbf{w}|D)\ f_{\mathbf{w}}(x)$



Computational tools I: Laplace approximation

Idea: For large n, the posterior will be concentrated around the MAP $\sim$ ML value $\widehat{\theta}$ and (for continuous θ) can be approximated by a Gaussian. This stems from the behaviour of the likelihood for large n.



Posterior density of θ for the biased coin for $n = 3, 10, 50, 100$. The true value under which the data were generated was $\theta = 0.7$.

$$\ln p(D|\theta) = \sum_{i=1}^{n} \ln p(x_i|\theta) = \sum_{i=1}^{n} \ln p(x_i|\widehat{\theta}) + \frac{c_2}{2} n \left(\theta - \widehat{\theta}\right)^2 + \frac{c_3}{3!} n \left(\theta - \widehat{\theta}\right)^3 + \ldots$$

with

$$c_k = \frac{1}{n} \sum_{i=1}^{n} \partial_\theta^k \ln p(x_i|\theta)_{|\widehat{\theta}} \approx E_x[\partial_\theta^k \ln p(x|\theta)_{|\widehat{\theta}}] = O(1)$$

Hence, in the posterior, the dominating term

$$p(\theta|D) \propto \exp\left[-\frac{-|c_2|}{2} n \left(\theta - \widehat{\theta}\right)^2\right] \left(1 + \frac{c_3}{3!} n \left(\theta - \widehat{\theta}\right)^3 + \ldots\right)$$

is a Gaussian and the corrections are small: With high posterior probability, we have $|\theta - \widehat{\theta}| \sim \frac{1}{\sqrt{n}}$ and $n \left|\theta - \widehat{\theta}\right|^3 \sim \frac{1}{\sqrt{n}}$.

Bayes asymptotics

For finite dimensional parametric models with continuous priors we have

$$p(\theta|D) \approx \mathcal{N}\left(\hat{\boldsymbol{\theta}}, \mathbf{I}^{-1}(\hat{\boldsymbol{\theta}})\right)$$

for $n \to \infty$, where $\hat{\boldsymbol{\theta}}$ is the ML estimator and $\mathbf{I}_{ij}(\boldsymbol{\theta}) = -\partial_i \partial_j \sum_{k=1}^{n} \ln p(x_k|\boldsymbol{\theta})$. This should be compared to the asymptotic errors of ML estimation !

Laplace approximation

Compute integrals by Taylor expansion to 2nd order at maximum $\hat{\mathbf{z}}$.

$$\int e^{-h(\mathbf{z})}\, d\boldsymbol{\theta} \approx e^{-h(\hat{\mathbf{z}})} \int \exp\left[-\frac{1}{2}(\mathbf{z}-\hat{\mathbf{z}})^T \mathbf{A}(\mathbf{z}-\hat{\mathbf{z}})\right]\, d\mathbf{z}$$

$$= e^{-h(\hat{\mathbf{z}})} \frac{(2\pi)^{K/2}}{|\mathbf{A}|^{1/2}}$$

with $\mathbf{A} = \nabla^2 h(\hat{\mathbf{z}})$.

Approximating the evidence

$$-\ln p(D) = -\ln \int p(D|\boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta} \approx -\ln p(D|\hat{\boldsymbol{\theta}}) - \ln p(\hat{\boldsymbol{\theta}}) - \frac{K}{2}\ln(2\pi) + \frac{1}{2}\ln|\mathbf{A}|$$

with $\mathbf{A} = -\nabla^2 \ln p(\hat{\boldsymbol{\theta}}|D)$ and $\hat{\boldsymbol{\theta}}$ is the MAP estimator.

Further approximation: Bayes Information Criterion(BIC) :

Use $|A| = O(N^K)$ and $\hat{\boldsymbol{\theta}} \approx \boldsymbol{\theta}_{ML}$

$$-\ln p(D) \approx -\ln p(D|\boldsymbol{\theta}_{ML}) + \frac{K}{2}\ln n$$

Posterior expectations

Approximate

$$\langle g(\boldsymbol{\theta})\rangle \doteq E[g(\boldsymbol{\theta})|D] = \frac{\int e^{-h^*(\boldsymbol{\theta})}\, d\boldsymbol{\theta}}{\int e^{-h(\boldsymbol{\theta})}\, d\boldsymbol{\theta}}$$

with

$$-h^*(\boldsymbol{\theta}) = \ln p(\boldsymbol{\theta}) + \ln p(D|\boldsymbol{\theta}) + \ln g(\boldsymbol{\theta})$$
$$-h(\boldsymbol{\theta}) = \ln p(\boldsymbol{\theta}) + \ln p(D|\boldsymbol{\theta})$$

and let $\hat{\boldsymbol{\theta}}^*$, $\hat{\boldsymbol{\theta}}$ the maximisers of h^* and h. Then

$$\langle g(\boldsymbol{\theta})\rangle \approx \sqrt{\frac{\left|\nabla^2 h(\hat{\boldsymbol{\theta}})\right|}{\left|\nabla^2 h^*(\hat{\boldsymbol{\theta}}^*)\right|}}\, \exp\left[-h^*(\hat{\boldsymbol{\theta}}^*) + h(\hat{\boldsymbol{\theta}})\right]$$

Application: Bayesian Neural Networks

Consider neural network input-ouput

$$f_{\mathbf{w}}(\mathbf{x}) = \sum_j W_j \sigma(\mathbf{w}_j^T \mathbf{x})$$

where e.g. $\sigma(z) = \tanh(z)$.

Probabilistic model:

$$p(y|\mathbf{x}, \mathbf{w}) \propto \exp\left(-\frac{\beta}{2}(y - f_{\mathbf{w}}(\mathbf{x}))^2\right) \qquad \text{Regression}$$

$$p(y|\mathbf{x}, \mathbf{w}) = \left(\frac{1}{1 + e^{-f_{\mathbf{w}}(\mathbf{x})}}\right)^y \left(\frac{1}{1 + e^{f_{\mathbf{w}}(\mathbf{x})}}\right)^{1-y} \qquad \text{Classification}$$

Priors:

$$p(\mathbf{w}) \propto \exp\left(-\frac{1}{2}\sum_k \alpha_k ||\mathbf{w}_k||^2\right)$$

Approximate posterior (Regression)

Introduce

$$E_D = \sum_i (y_i - f_{\mathbf{w}}(\mathbf{x}_i))^2$$

$$E_W = ||\mathbf{w}||^2$$

and the minimiser as $\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} (\beta E_D + \alpha E_W)$, we get the posterior approximation

$$p(\mathbf{w}|D) \propto e^{-(\beta E_D + \alpha E_W)} \approx \exp\left[-\frac{1}{2}\Delta\mathbf{w}^T \mathbf{A} \Delta\mathbf{w}\right]$$

where $\Delta\mathbf{w} = \mathbf{w} - \hat{\mathbf{w}}$ and $\mathbf{A} = \beta\nabla^2 E_D^{MP} + \alpha\mathbf{I}$

Approximate Predictive distribution

Linearise $f_{\mathbf{w}}(\mathbf{x}) \approx f_{\hat{\mathbf{w}}}(\mathbf{x}) + \mathbf{g}^T \Delta\mathbf{w}$

$$p(y|x, D) \approx C \int p(y|x, \mathbf{w}) \exp\left[-\frac{1}{2}\Delta\mathbf{w}^T \mathbf{A} \Delta\mathbf{w}\right] \approx \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y - y_{MP})^2}{2\sigma^2}\right)$$

with $y_{MP} = f_{\hat{\mathbf{w}}}(\mathbf{x})$ and $\sigma^2 = \frac{1}{\beta} + \mathbf{g}^T \mathbf{A}^{-1} \mathbf{g}$.

Evidence approximation

$$-\ln p(D|\alpha,\beta) = \beta E_D^{MP} + \alpha E_W^{MP} + \frac{1}{2}\ln|\mathbf{A}| - \frac{W}{2}\ln\alpha - \frac{n}{2}\ln\beta + \frac{n}{2}\ln(2\pi)$$

Estimate hyperparameters:

Compute $\gamma = \sum_{k=1}^{W} \frac{\lambda_k}{\lambda_k+\alpha}$, where the λ_k are eigenvalues of $\beta\nabla^2 E_D^{MP}$. Start with some values of α and β, optimise $\hat{\mathbf{w}}$ and re-estimate

$$\alpha^{new} = \frac{\gamma}{2E_W}$$

$$\beta^{new} = \frac{n-\gamma}{2E_D}$$

optimise $\hat{\mathbf{w}}$ and repeat until convergence.

ARD: The method can be extended to separate α_ks for each input neuron. Large α_k leads to a 'shut off' for the corresponding weights.

Example

Artificial data set: *Friedman data* generated as

$$y(\mathbf{x}) = 0.1e^{4x_1} + \frac{4}{1 + e^{-20(x_2 - \frac{1}{2})}} + 3x_3 + 2x_4 + x_5 + 0 \cdot \sum_{i=6}^{10} x_i + \nu$$

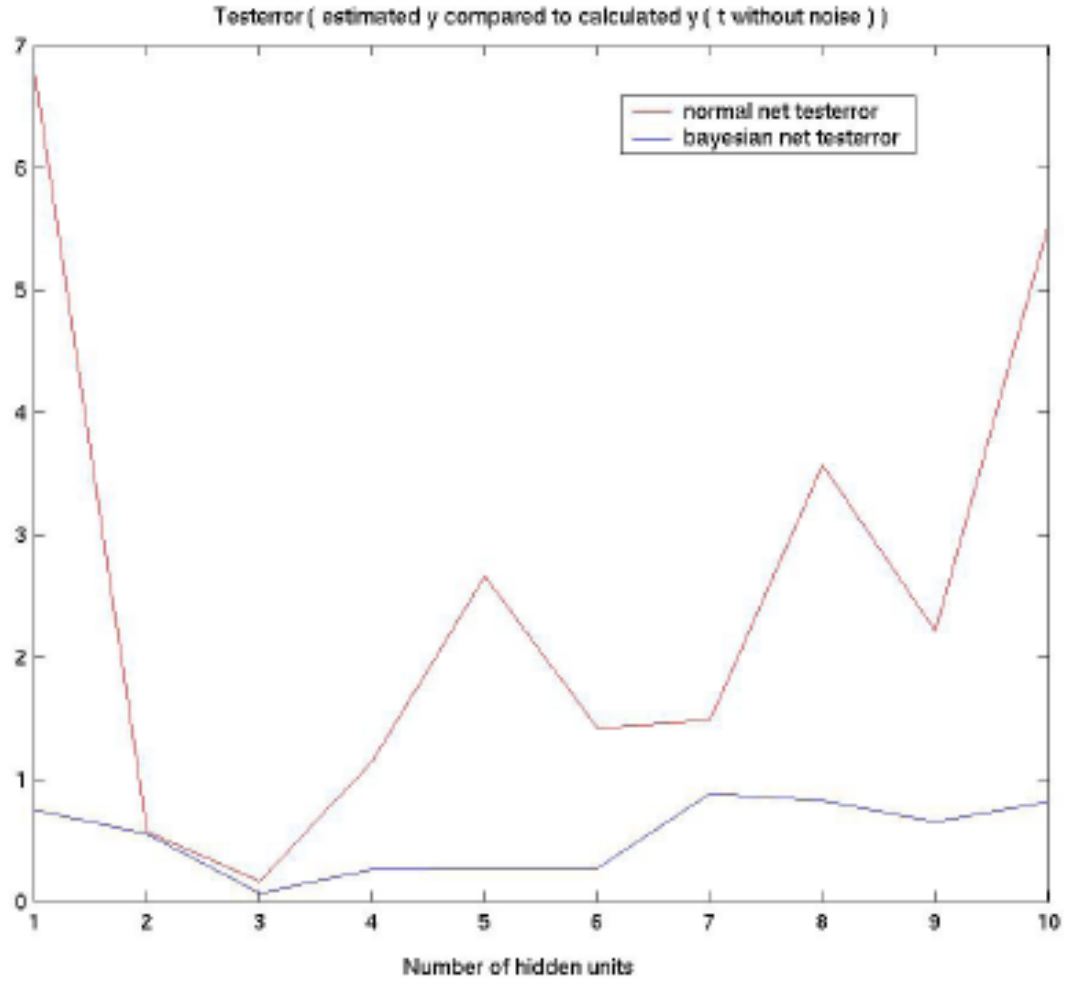


Figure 6: 200 training samples, 30 training loops, 30 evidence-iterations

ARD: α_i for network inputs x_i:

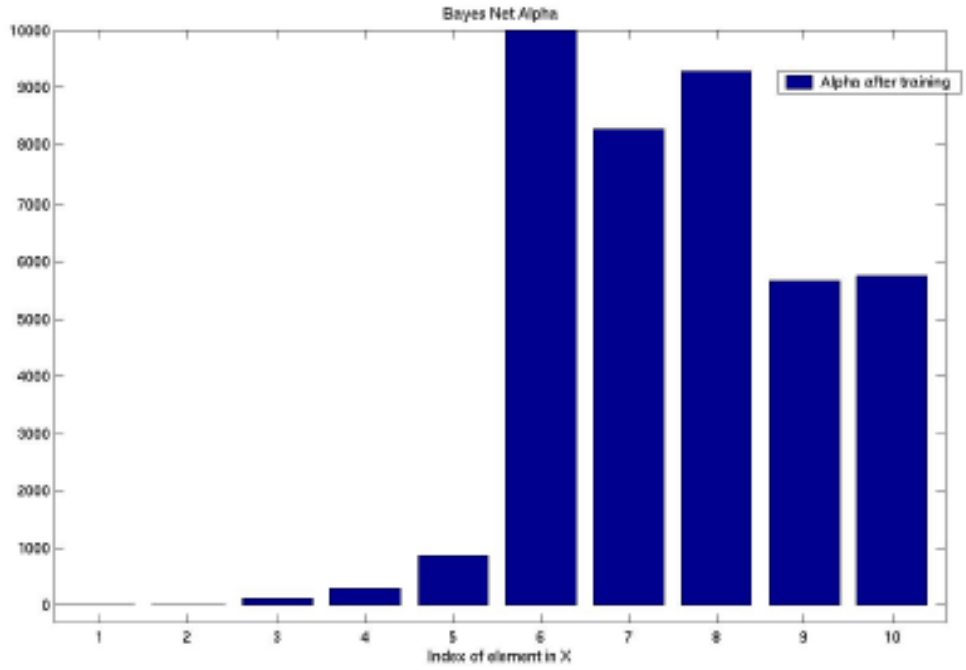


Figure 5: 5 hidden units, 200 training samples, 100 training loops, 50 evidence-iterations, zoomed into diagram

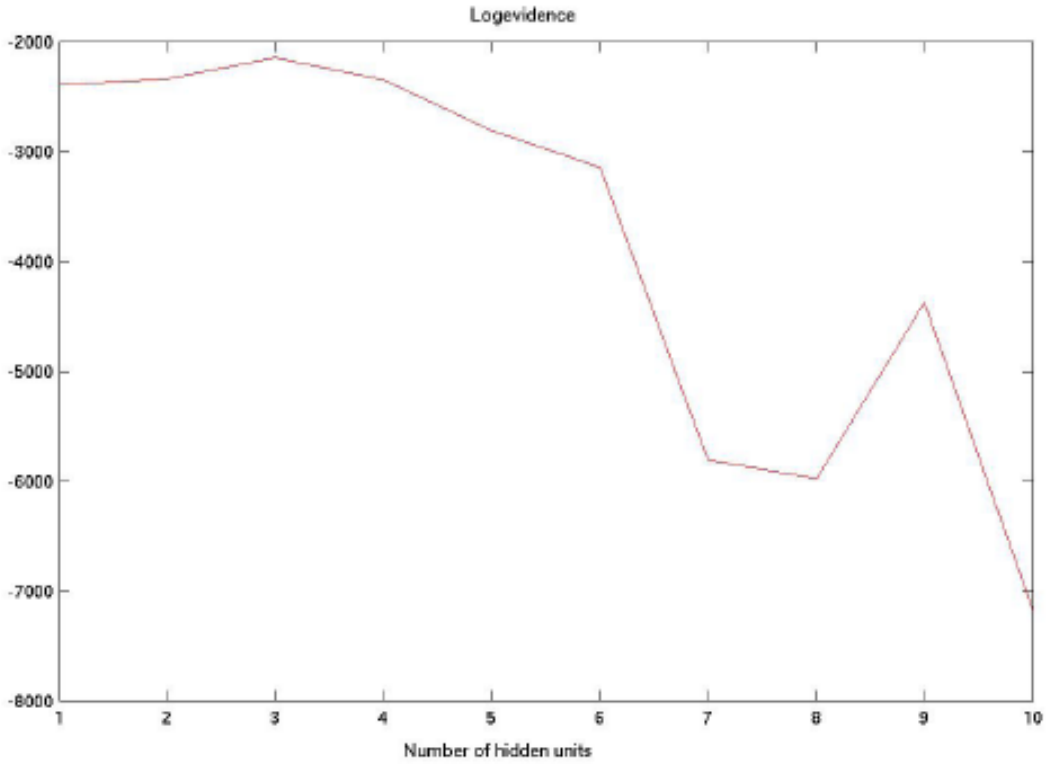


Figure 7: 200 training samples, 30 training loops, 30 evidence-iterations

Summary: Laplace approximation

- Approximates posterior (log posterior) by a Gaussian (2nd order Taylor expansion around MAP value).

- Becomes exact for large number of data for finite dimensional models with continuous parameters (under technical conditions).

- Advantages: Integration is replaced by optimisation, i.e. by finding the MAP. The Hessian which is required for the covariance can also be used for a Newton Raphson algorithm.

- Disadvantages: local approximation, takes into account only MAP and curvature. Ignores other posterior modes. Can't be used for discrete variables.

Optimal compression and Fisher information

- Sequential compression of sequence $x_1, \ldots x_n$.

- Minimal code length $H[X]$ if probability $P(x|\theta)$ is known.

- Statistician does not no θ, but uses Bayes method with prior $p(\theta)$ in sequential coding

- Bayes estimate

$$P(x|x_{i-1}, \ldots, x_1) = \frac{P(x, x_{i-1}, \ldots, x_1)}{P(x_{i-1}, \ldots, x_1)}$$

  with

$$P(x_i, \ldots, x_1) = \int \prod_{j=1}^{i} P(x_i|\theta)\ p(\theta) d\theta$$

- Total expected extra code length (risk) after n steps

$$R_n(\theta) = \sum_{x_1,\ldots x_n} \prod_{i=1}^{n} P(x_i|\theta) \{-\ln P(x_n,\ldots,x_1)\} - nH[X]$$

- One can show that for large n

$$R_n(\theta) = \frac{D}{2} \ln \frac{n}{2\pi} - \frac{D}{2} - \ln p(\theta) + \frac{1}{2} \ln \det J(\theta) + o(1)$$

where the **Fisher Information** is given by

$$J_{ij}(\theta) = \sum_x P(x|\theta) \frac{\partial \ln P(x|\theta)}{\partial \theta_i} \frac{\partial \ln P(x|\theta)}{\partial \theta_j}$$

- If we use **Jeffrey's prior**

$$p(\theta) \propto \sqrt{\det J(\theta)}$$

for Bayes compression, then the Bayes risk $R_n(\theta)$ is (asymptotically) independent of θ.

- Bernoulli model $P(x|\theta) = \theta^x (1-\theta)^{1-x}$ and

$$J(\theta) = \frac{1}{\theta(1-\theta)}$$

and Jeffrey's prior is

$$p_{\text{Jeff}}(\theta) \propto \frac{1}{\sqrt{\theta(1-\theta)}}$$

- Jeffrey's prior is also invariant against reparametrisations of the parameter.

Computational tools II: Monte Carlo (MC) methods

Goal: Represent probability distributions by random samples.

Hence, we have to be able to generate (usually dependent!) samples from a given distribution $p(x)$. In the application to Bayesian models case x is set of parameters and p the posterior.

Basic method: Transformation method and rejection method with proposal density

- Problem: Need random variables with density $p(x)$ (target density), have random variables with density $q(x)$ (proposal density).

- **Transformation method:**

  Find a transformation $x = f(y)$ such that the distribution of x is $p(x)$.

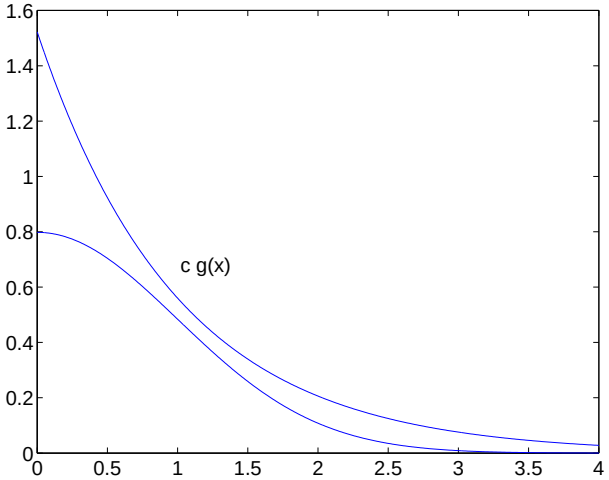
  Let $F(z) = P(x \leq z)$ with density $p(x) = F'(x)$. Let $y \sim U(0,1)$ a random variable with uniform density. Then the transformed $x = F^{-1}(y)$ has density $p(x)$.

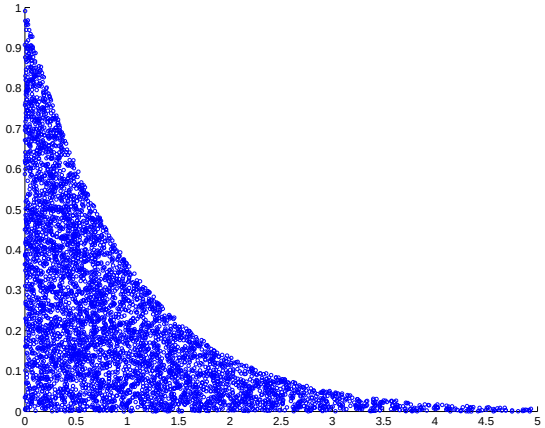
- **Rejection method:**

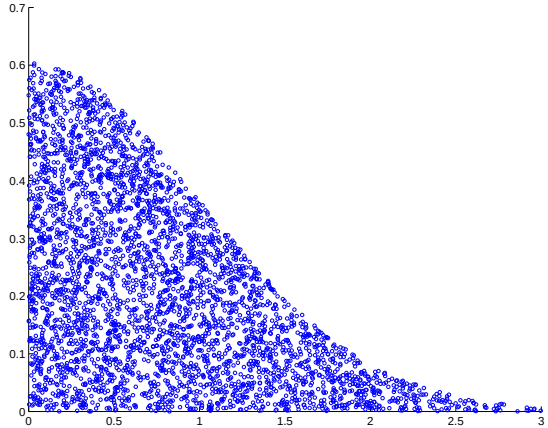
  Assume $\frac{p(x)}{q(x)} \leq c$. Generate two independent random variables $x \sim q(x)$ and $u \sim U(0,1)$. If $u \leq \frac{p(x)}{cq(x)}$ accept x. Otherwise start again.

Example: Exponential $\rightarrow$ Normal

- We can get *positive normal (Gaussian)* random variables with density $p(x) = \frac{2}{\sqrt{2\pi}} \exp(-\frac{1}{2}x^2)$ for $0 \leq x < \infty$ by the *rejection method* using exponentially distributed. A good candidate is $c = \sqrt{2e/\pi}$ and $\frac{p(x)}{cq(x)} = \exp(-(x-1)^2/2)$.







Note: The rejection method can also be applied to the case where we know the desired distribution only up to a normalisation constant, i.e. $p(\mathbf{x}) = \frac{\tilde{p}(\mathbf{x})}{Z}$ with unknown Z.

Markov Chain Monte Carlo

- It easy to sample from simple low dimensional distributions by the transformation or the rejection methods. But this doesn't work well for higher dimensions.

- General Strategy: Construct a Markov chain with a transition probability $T(y|x)$ that has $p(x)$ as its stationary distribution.

- Let us assume that there is only a single stationary distribution and that any initial distribution converges to it. Then, asymptotically (that is if we wait long enough), the distribution of samples X_t drawn from the Markov chain is very close to $p(x)$.

Stationary distributions

Let $p_t(x)$ denote the marginal distribution of X_t. The update of the marginal distribution given by

$$p_{t+1}(x) = \int T(x|y)p_t(y)\, dy$$

The *stationary distribution* must fulfil stationarity

$$p(x) = \int T(x|y)p(y)\, dy$$

Hence, we should find transition probabilities which leave our target distribution invariant.

Detailed balance

Consider a Markov chain with transition probability $T(x|y)$ for going from y to x.

The update of the marginal distribution given by

$$p_{t+1}(x) = \int T(x|y)p_t(y)\ dy$$

This can be written as

$$p_{t+1}(x) - p_t(x) = \int T(x|y)p_t(y)\ dy - p_t(x)\int T(y|x)\ dy$$

Hence, the *stationary distribution* must fulfil

$$0 = \int T(x|y)p(y)\ dy - \int p(x)T(y|x)\ dy$$

If the transition probability $T(y|x)$ is constructed in such a way that we have

$$T(x|y)p(y) = p(x)T(y|x)$$

we say that the Markov chain fulfills **detailed balance**. The chain is also known as a reversible Markov chain.

The Metropolis - Hastings method

- Define a **proposal distribution** $q(x'|x)$.

- Given a state $x = x_t$ at *step* t generate a new state x' with probability distribution $q(x'|x)$.

- Define **acceptance ratio**

$$A(x'; x) = \min\left(1, \frac{p(x')q(x|x')}{p(x)q(x'|x)}\right)$$

- **Accept** new state $x_{t=1} = x'$ with probability $A(x'; x)$

  Reject new state, ie **keep old** state $x_{t+1} = x$ with probability $1 - A(x'; x)$

Analysis

- We see that this defines a Markov chain with transition probability (assume x' was accepted)

$$T(x'|x) = A(x';x)q(x'|x) + (1 - \alpha(x))\delta(x - y) \ .$$

  where $\alpha(x) = \int A(y;x)q(y|x)\ dy$.

- It fulfills detailed balance: For $x' \neq x$, we have

$$p(x)T(x'|x) = p(x)q(x'|x)\,\min\left(1, \frac{p(x')q(x|x')}{p(x)q(x'|x)}\right) =$$

  $\min\Big(q(x'|x)p(x), q(x|x')p(x')\Big) = p(x')q(x|x')A(x;x') = p(x')T(x|x')$
  with the stationary distribution $p(x)$.

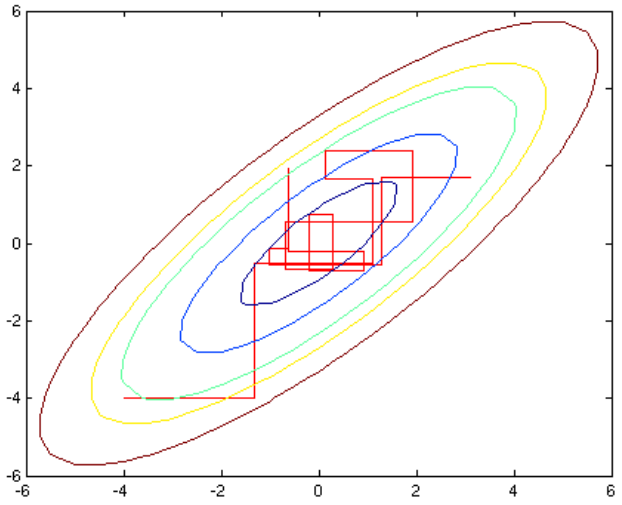
- Note, that only ratios of probabilities $\frac{p(x')}{p(x)}$ are required. Hence, normalization constants of probabilities are not needed.

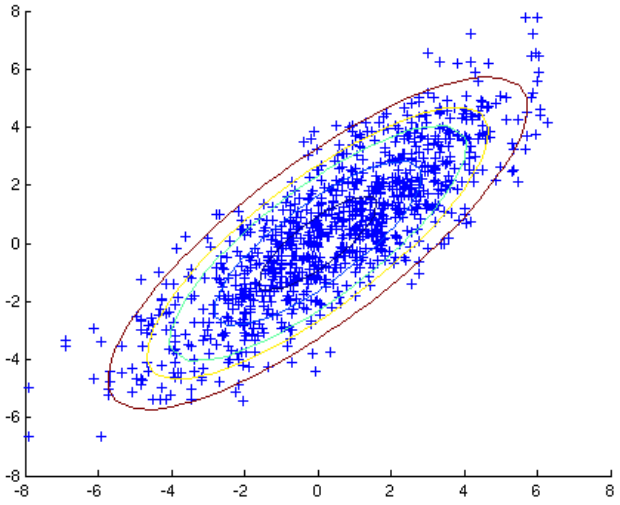
- This general method depends on clever choices of proposals q.

Gibbs sampling

is easily applied when one can sample from the conditional probabilities $p(x_i|\mathbf{x}_{-i})$ where $\mathbf{x}_{-i} = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_N)$. At step $\tau + 1$, one cycles through the components of $\mathbf{x}$ and samples

$$
\begin{aligned}
x_1^{\tau+1} &\sim p(x_1|x_2^\tau, x_3^\tau, \ldots, x_N^\tau) \\
x_2^{\tau+1} &\sim p(x_2|x_1^{\tau+1}, x_3^\tau, \ldots, x_N^\tau) \\
&\ldots \quad \ldots \quad \ldots \\
x_j^{\tau+1} &\sim p(x_j|x_1^{\tau+1}, \ldots, x_{j-1}^{\tau+1}, x_{j+1}^\tau, \ldots, x_N^\tau) \\
&\ldots \quad \ldots \quad \ldots \\
x_N^{\tau+1} &\sim p(x_N|x_1^{\tau+1}, \ldots, x_{N-1}^{\tau+1})
\end{aligned}
$$





Gibbs sampler from Metropolis Hastings

Consider the Gibbs proposal $q(\mathbf{x}'|\mathbf{x}) = p(x'_i|\mathbf{x}_{-i})$ with $\mathbf{x}_{-i} = \mathbf{x}'_{-i}$ Then

$$A(\mathbf{x}';\mathbf{x}) = \frac{p(\mathbf{x}')p(x_i|\mathbf{x}_{-i})}{p(\mathbf{x})p(x'_i|\mathbf{x}_{-i})} = \frac{p(x'_i|\mathbf{x}_{-i})p(\mathbf{x}_{-i})p(x_i|\mathbf{x}_{-i})}{p(x_i|\mathbf{x}_{-i})p(\mathbf{x}_{-i})p(x'_i|\mathbf{x}_{-i})} = 1$$

The proposal is always accepted !

Application: Change point model

Disasters can occur at years $i \in \{1, 2, \ldots, n\}$. Number of disasters are distributed as a Poisson variable, ie $p(x|\lambda) = e^{-\lambda}\frac{\lambda^x}{x!}$. But the rate of disasters change from λ_1 to λ_2 at unknown **change point** $K \in \{1, 2, \ldots, n\}$.

To estimate K we assume the following hierarchical Bayesian model

- K has a discrete prior distribution $p(K)$.
- Given K and $\lambda_{1,2}$, the data are independent $x_i \sim e^{-\lambda}\frac{\lambda^x}{x!}$.
- The rates $\lambda_{1,2}$ are independent with the *conjugate prior* $\lambda_{1,2} \sim \text{Gamma}(a_{1,2}, \eta_{1,2})$ density. $\eta_{1,2}$ are *unknown* hyperparameters and $a_{1,2}$ are known.

- $\eta_{1,2}$ are independent hyperparameters with prior distribution $\eta_{1,2} \sim$ Gamma$(b_{1,2}, c_{1,2})$ with known $b_{1,2}$ and $c_{1,2}$.

Note that the Gamma density is given by

$$p(x|\alpha, \beta) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}$$

with $E[X] = \frac{\alpha}{\beta}$ and $Var[X] = \frac{\alpha}{\beta^2}$.

Problem: Given a set of observations $D = (x_1, \ldots, x_n)$ over n years, draw samples from the **posterior distribution** $p(K, \eta, \lambda|\mathbf{x})$.

- Joint distribution

$$p(\mathbf{x}, \lambda_{1,2}, \eta_{1,2}, K) = p(D|\lambda_{1,2}, K)p(\lambda_{1,2}|\eta_{1,2})p(\eta_{1,2})p(K) =$$

$$\prod_{i=1}^{K} e^{-\lambda_1}\frac{\lambda_1^{x_i}}{x_i!} \times \prod_{K+1}^{n} e^{-\lambda_2}\frac{\lambda_2^{x_i}}{x_i!} \times$$

$$\times\frac{\eta_1^{a_1}}{\Gamma(a_1)}\lambda_1^{a_1-1}e^{-\eta_1\lambda_1} \times \frac{\eta_2^{a_2}}{\Gamma(a_2)}\lambda_2^{a_2-1}e^{-\eta_2\lambda_2} \times$$

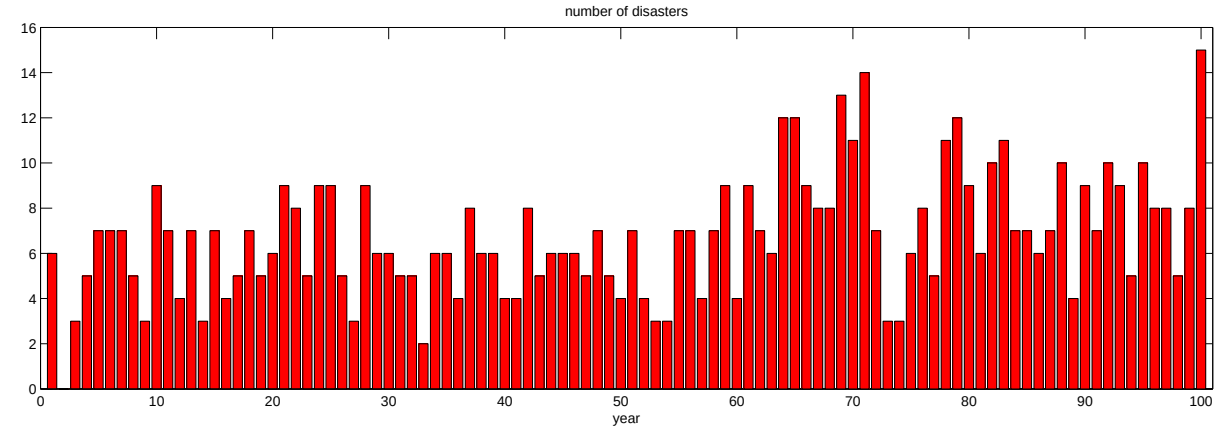
$$\times\frac{c_1^{b_1}}{\Gamma(b_1)}\eta_1^{b_1-1}e^{-c_1\eta_1} \times \frac{c_2^{b_2}}{\Gamma(b_2)}\eta_2^{b_2-1}e^{-c_2\eta_2} \times$$

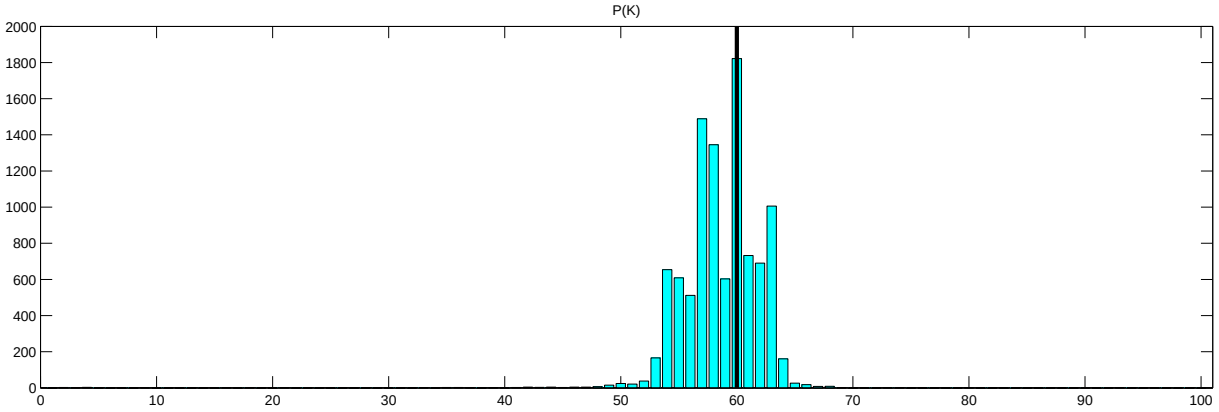
$$\times p(K)$$

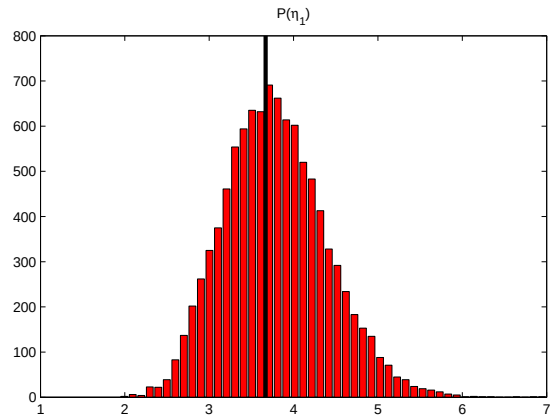
- Conditional distributions for Gibbs sampler

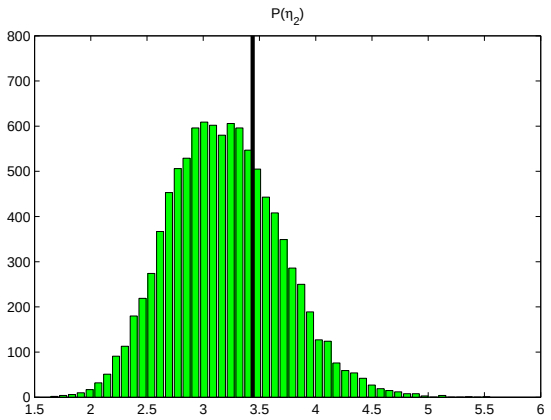
$$\begin{aligned}
\lambda_2|\lambda_1, \eta_{1,2}, K, \mathbf{x} &\sim \text{Gamma}(a_2 + \sum_{K+1}^{n} x_i, n - K + \eta_2) \\
\eta_1|\lambda_{1,2}, \eta_2, K, \mathbf{x} &\sim \text{Gamma}(a_1 + b_1, \lambda_1 + c_1) \\
K|\lambda_{1,2}, \eta_{1,2}, \mathbf{x} &\sim \text{const} \times p(K)e^{-K(\lambda_1-\lambda_2)}(\lambda_1/\lambda_2)^{\sum_{i=1}^{K} x_i}
\end{aligned}$$

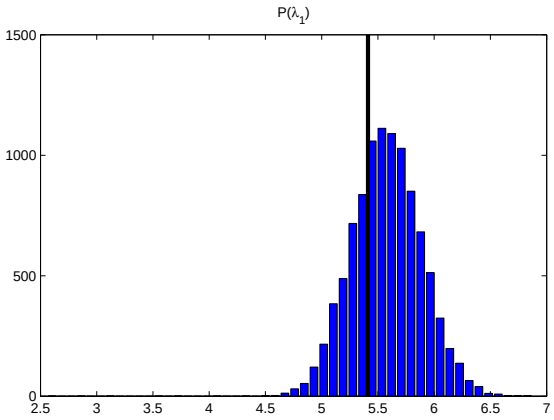
Simulations

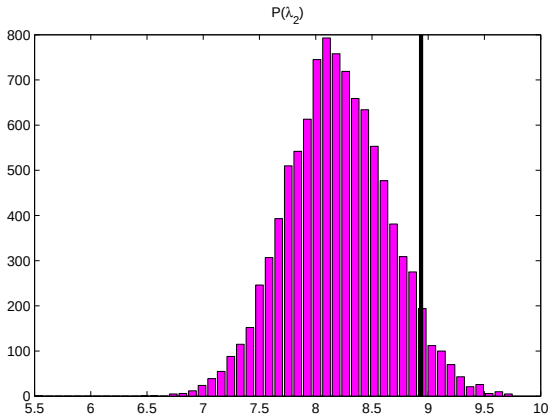




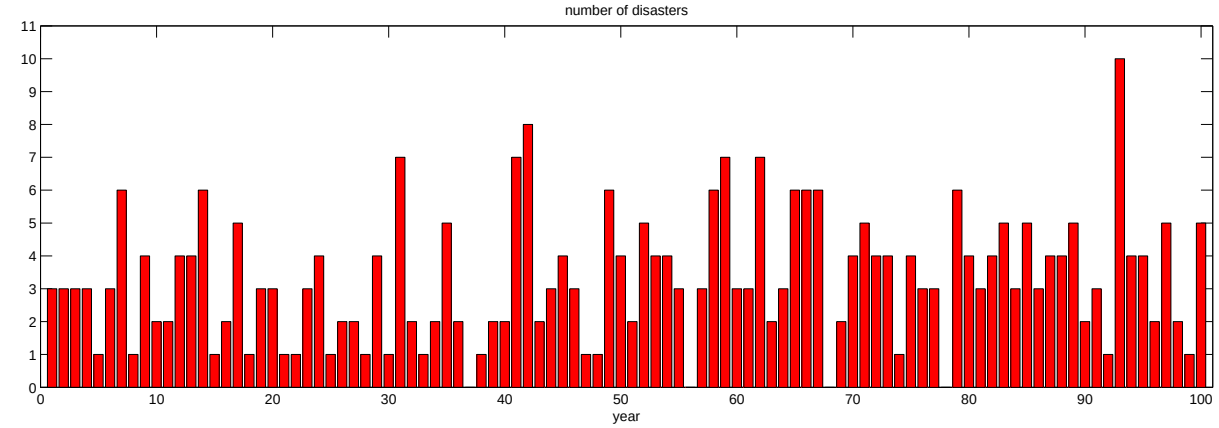

P(η1)
800
700
600
500
400
300
200
100
0
1
2
3
4
5
6
7

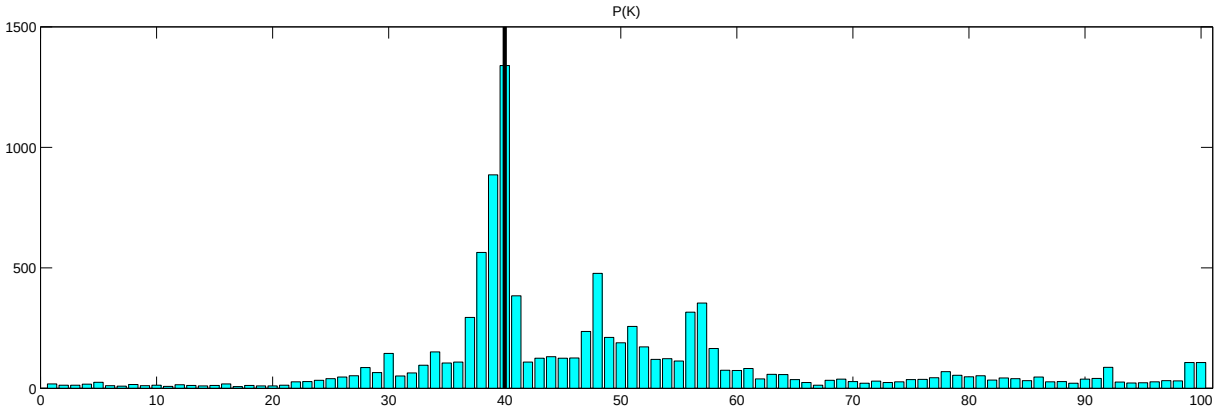

P(η2)
800
700
600
500
400
300
200
100
0
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6

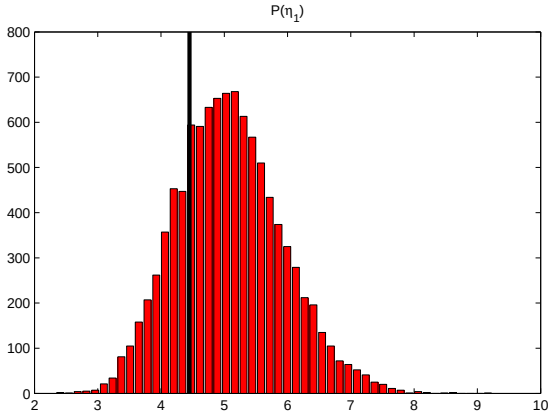

P(λ1)
1500
1000
500
0
2.5
3
3.5
4
4.5
5
5.5
6
6.5
7

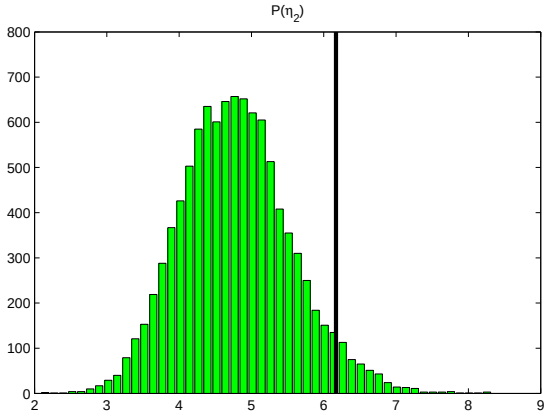

P(λ2)
800
700
600
500
400
300
200
100
0
5.5
6
6.5
7
7.5
8
8.5
9
9.5
10

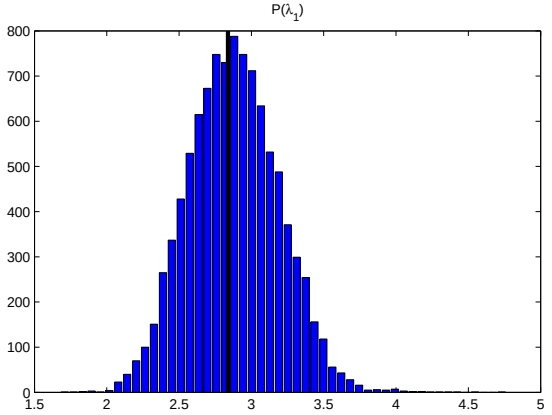
with somewhat more similar λ_{12}

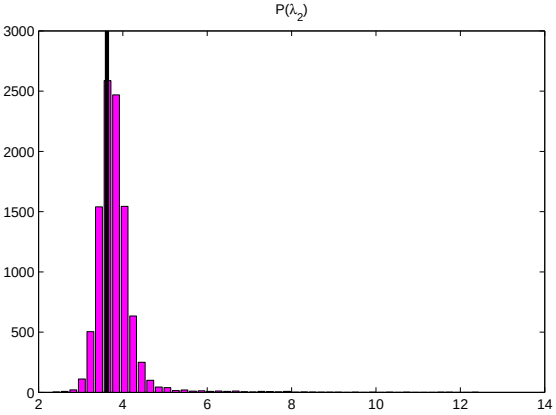





P(η1)
800
700
600
500
400
300
200
100
0
2
3
4
5
6
7
8
9
10


P(η2)
800
700
600
500
400
300
200
100
0
2
3
4
5
6
7
8
9


P(λ1)
800
700
600
500
400
300
200
100
0
1.5
2
2.5
3
3.5
4
4.5
5


P(λ2)
3000
2500
2000
1500
1000
500
0
2
4
6
8
10
12
14

Factor analysis

Observed data $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ are explained by a set of latent variables $\mathbf{F} = (\mathbf{f}_1, \ldots, \mathbf{f}_n)$. The model is given by notation

$$\mathbf{x} = \mathbf{\Lambda f} + \mathbf{e}$$

- $d =$ dimensionality of data. $n =$ number of observations.
- The *factor loadings* matrix $\mathbf{\Lambda}$ is $d \times q$.
- The noise covariance is $E[\mathbf{e}_i \mathbf{e}_i^\top] = \text{diag}\,(\psi_1^2, \ldots, \psi_d^2) = \mathbf{\Psi}$.
- $p(\mathbf{x}|\mathbf{x}, \mathbf{\Lambda}, \mathbf{\Psi}) = \mathcal{N}(\mathbf{x}|\mathbf{\Lambda f}, \mathbf{\Psi})$
- $p(\mathbf{f}_i) = \mathcal{N}(0, \mathbf{I})$.
- Integrating out $\mathbf{f}$, we get $p(\mathbf{x}|\mathbf{\Lambda}, \mathbf{\Psi}) = \mathcal{N}(\mathbf{x}|\mathbf{\Lambda\Lambda}^\top + \mathbf{\Psi})$

- Non–identifiability: Let $\mathbf{f}' \doteq \mathbf{Q}^\top \mathbf{f}$ and $\mathbf{\Lambda}' \doteq \mathbf{\Lambda}\mathbf{Q}$ then $\mathbf{\Lambda}'\mathbf{f}' = \mathbf{\Lambda}\mathbf{f}$ if $\mathbf{Q}^\top\mathbf{Q} = \mathbf{I}$. Also $\mathbf{f}'$ and $\mathbf{f}$ have the same distribution.

Non - Bayesian Inference

- One can use the EM algorithm to estimate Maximum Likelihood estimators of $\mathbf{\Lambda}$ and $\mathbf{\Psi}$.

- *Sparsity* of factor loadings: Use nonidentifiability and apply *rotations* with orthogonal $\mathbf{Q}$ to trained loading matrix $\mathbf{\Lambda}$: $\mathbf{\Lambda}_{rot} = \mathbf{\Lambda Q}$ to create sparse $\mathbf{\Lambda}_{rot}$.

  Use sparsity penalty. e.g.

$$\sum_{k=1}^{q} \sum_{l=1}^{d} \tanh\left(\alpha \lambda_{lk}^2\right)$$

  or *procrustes* rotation with penalty

$$\sum_{k=1}^{q} \sum_{l=1}^{d} (\lambda_{lk} - \tau_{lk})^2$$

  where τ_{lk} is a *target* matrix.

Bayesian inference (E. Fokoue)

- *Bayesian* approach: Introduce sparsity prior, e.g. by products of student densities

$$p(\lambda_{lk}|\alpha,\beta) \propto \frac{1}{\left(\beta + \frac{1}{2}\lambda_{lk}^2\right)^{\alpha+\frac{1}{2}}}$$

  which has high probability densities at the coordinate axes:

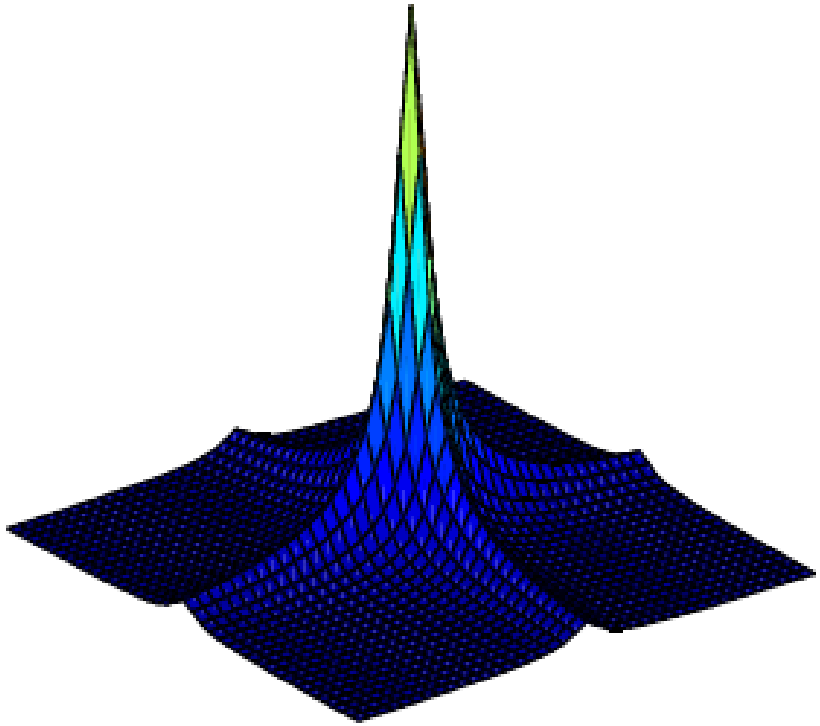


Figure 1: The 2-dimensional marginal prior for a row Λ_i

Sampling from $p(\mathbf{\Lambda}|\mathbf{X})$ is not feasible:

Posterior has complicated dependency on $\mathbf{\Lambda}$

$$p(\Lambda|\mathbf{X}) \propto p(\mathbf{X}|\mathbf{\Lambda})p(\mathbf{\Lambda}) = \prod_i \mathcal{N}(\mathbf{x}_i|\mathbf{m}, \mathbf{\Lambda}\mathbf{\Lambda}^\top + \mathbf{\Psi})p(\mathbf{\Lambda}) \propto$$

$$|\mathbf{\Lambda}\mathbf{\Lambda}^\top + \mathbf{\Psi}|^{-n/2} \exp\left[-\frac{1}{2}\sum_i \mathbf{x}_i^\top(\mathbf{\Lambda}\mathbf{\Lambda}^\top + \mathbf{\Psi})^{-1}\mathbf{x}_i)\right] p(\mathbf{\Lambda})$$

Data Augmentation

Introducing the auxiliary variables δ_{lk} with

$$\begin{aligned} p(\lambda_{lk}|\delta_{lk}) &= \mathcal{N}(0, 1/\delta_{lk}) \\ p(\delta_{lk}|\alpha, \beta) &= \frac{\delta_{lk}^{\alpha-1}\beta^{\alpha}}{\Gamma(\alpha)} e^{-\beta\delta_{lk}} \end{aligned}$$

The marginal distribution is just

$$p(\lambda_{lk}|\alpha, \beta) \propto \frac{1}{\left(\beta + \frac{1}{2}\lambda_{lk}^2\right)^{\alpha+\frac{1}{2}}}$$

- Try to sample from $p(\boldsymbol{\Delta}, \boldsymbol{\theta}, \mathbf{F}|\mathbf{X})$ instead.

- Gibbs sampler: Alternate sampling between $p(\mathbf{F}|\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\Delta})$, $p(\boldsymbol{\theta}|\boldsymbol{\Delta}, \mathbf{F}, \mathbf{X})$ and $p(\boldsymbol{\Delta}|\boldsymbol{\theta}, \mathbf{F}, \mathbf{X})$

- Conditional of factors is a Gaussian

$$\mathbf{f}_i|\mathbf{x}_i, \mathbf{\Lambda}, \mathbf{\Psi} \sim \mathcal{N}\left(\mathbf{f}_i|(\mathbf{I}_q + \mathbf{\Lambda}^\top \mathbf{\Psi}^{-1}\mathbf{\Lambda})^{-1}\mathbf{\Lambda}^\top \mathbf{\Psi}^{-1}\mathbf{x}_i, (\mathbf{I}_q + \mathbf{\Lambda}^\top \mathbf{\Psi}^{-1}\mathbf{\Lambda})^{-1}\right)$$

- Conditional of $\mathbf{\Lambda}$

$$p(\mathbf{\Lambda}|\mathbf{X},\mathbf{F},\mathbf{\Delta}) \propto p(\mathbf{X}|\mathbf{F},\mathbf{\Lambda},\mathbf{\Delta})p(\mathbf{\Lambda}|\mathbf{\Delta}) =$$
$$\mathcal{N}(\mathbf{X}|\mathbf{\Lambda F},\mathbf{\Psi})p(\mathbf{\Lambda}|\mathbf{\Delta}) \propto$$
$$\exp\left[-\frac{1}{2}(\mathbf{X}-\mathbf{\Lambda F})^{\top}\mathbf{\Psi}^{-1}(\mathbf{X}-\mathbf{\Lambda F}))\right] p(\mathbf{\Lambda}|\mathbf{\Delta})$$

  is also Gaussian !

- Finally

$$p(\mathbf{\Delta}|\mathbf{\Lambda},\mathbf{X},\mathbf{F},\mathbf{M})$$

  is a product of *Gamma* densities.

Independence sampler

Let $q(x'|x) = q(x')$ independent of x in the Metropolis method.

Then the acceptance probability is

$$A(\mathbf{x}'; \mathbf{x}) = \min\left\{\frac{p(x')q(x)}{p(x)q(x')}, 1\right\}$$

This is similar to a rejection method, but samples are dependent. Again, q should be similar to p to achieve good acceptance rates.

Random walk sampler

This method can be easily applied to continuos states. As the proposal, one often chooses a move

$$\mathbf{x}' = \mathbf{x} + \sqrt{\rho}\mathbf{z}$$

where $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$. This is a **symmetric proposal** with $q(x'|x) = q(x|x')$

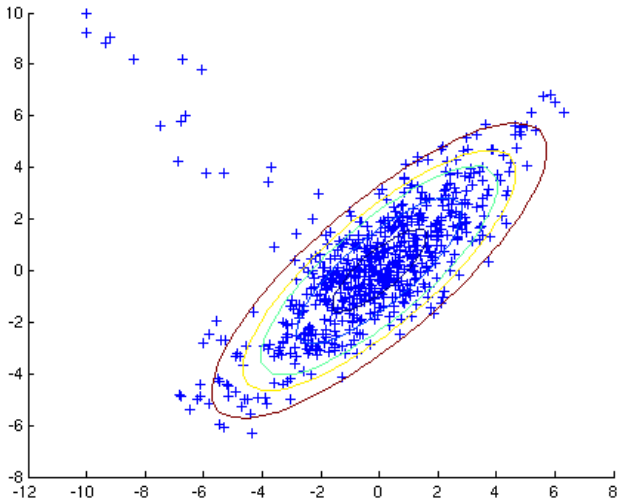
The acceptance probability is then

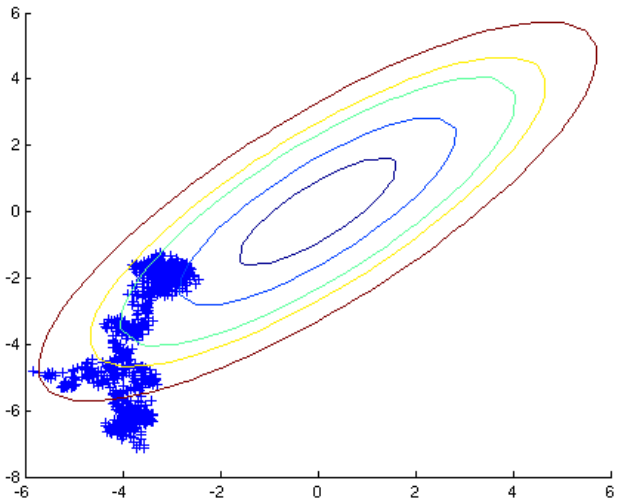
$$A(\mathbf{x}'; \mathbf{x}) = \min\left\{\frac{p(x')}{p(x)}, 1\right\}$$

With this form of A, one speaks of a **Metropolis sampler**.

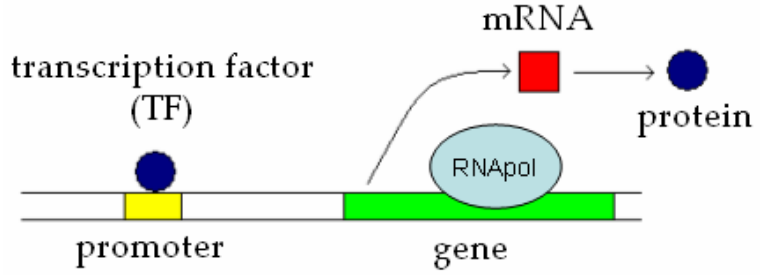
The choice of ρ is important. For large ρ acceptance will be unlikely. Small ρ will lead to high acceptance rates but too a very slow **diffusion**.

Example: Two dimensional Gaussian with $\rho = 1$ and $\rho = 0.1$ (1000 samples).


10
8
6
4
2
0
-2
-4
-6
-8
-12
-10
-8
-6
-4
-2
0
2
4
6
8


6
4
2
0
-2
-4
-6
-8
-6
-4
-2
0
2
4
6

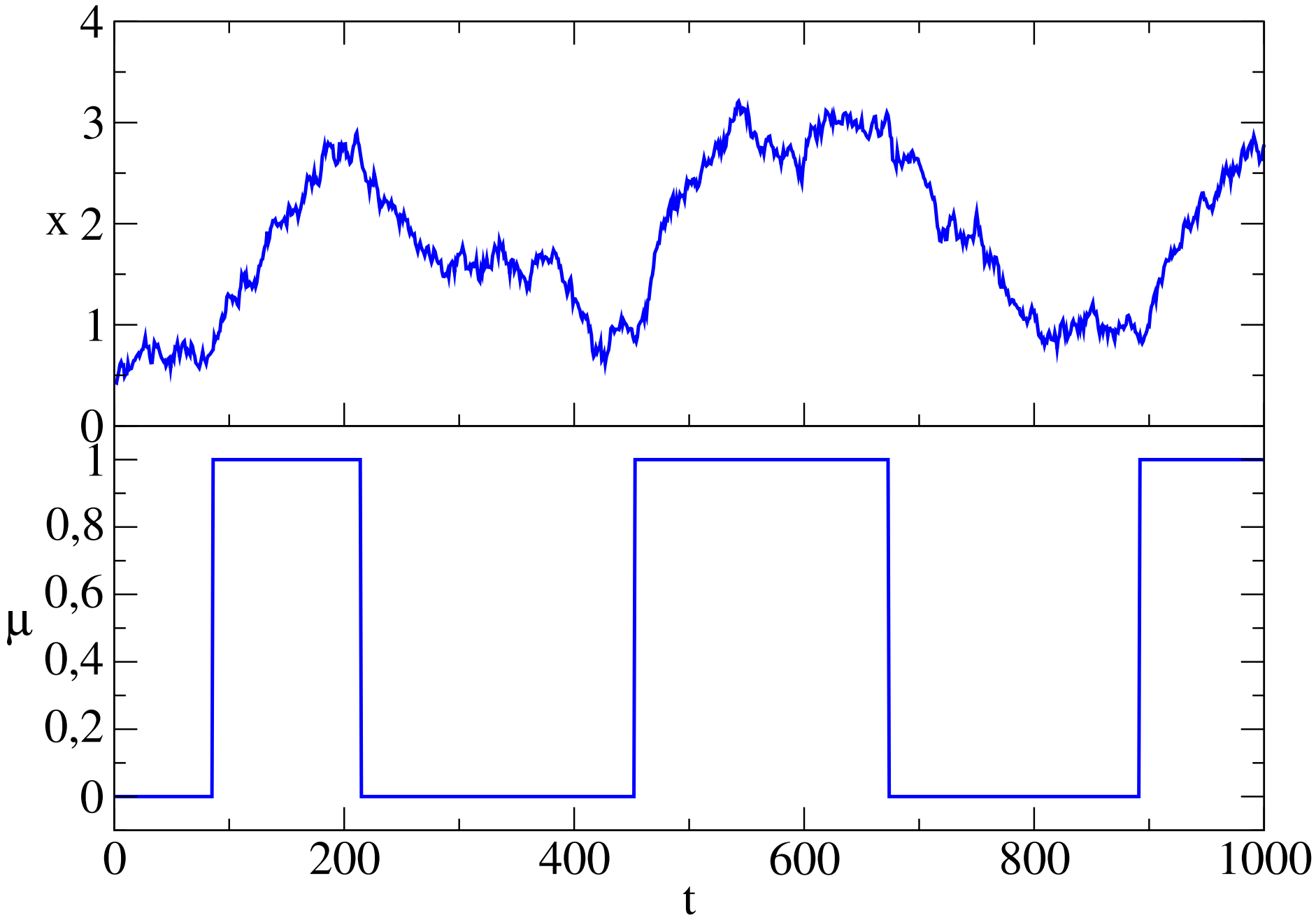
Stochastic processes as a prior for unobserved dynamics : Model for transcriptional regulation:



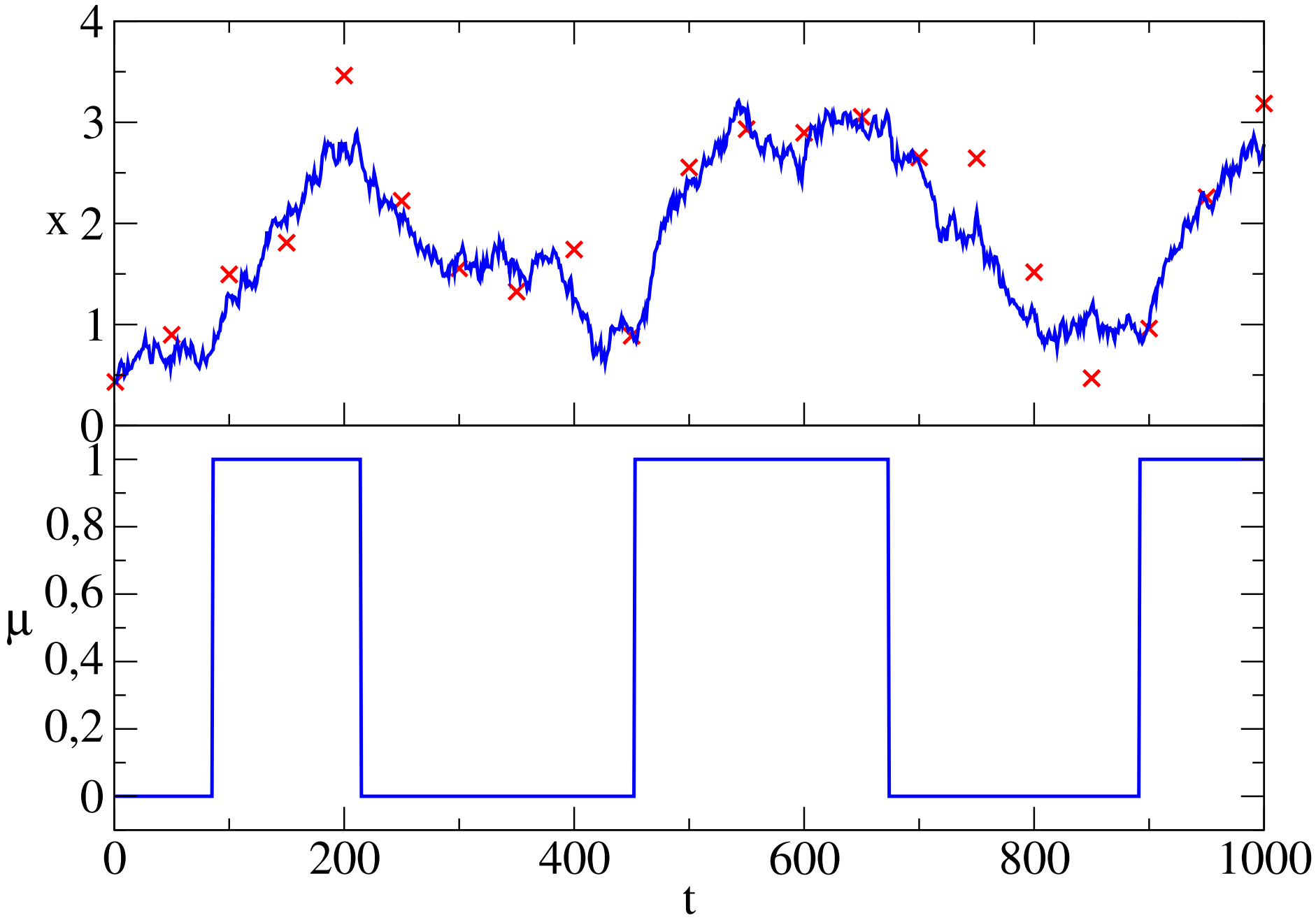
- $X(t) =$ mRNA concentration of target gene. modelled by an Ornstein - Uhlenbeck process

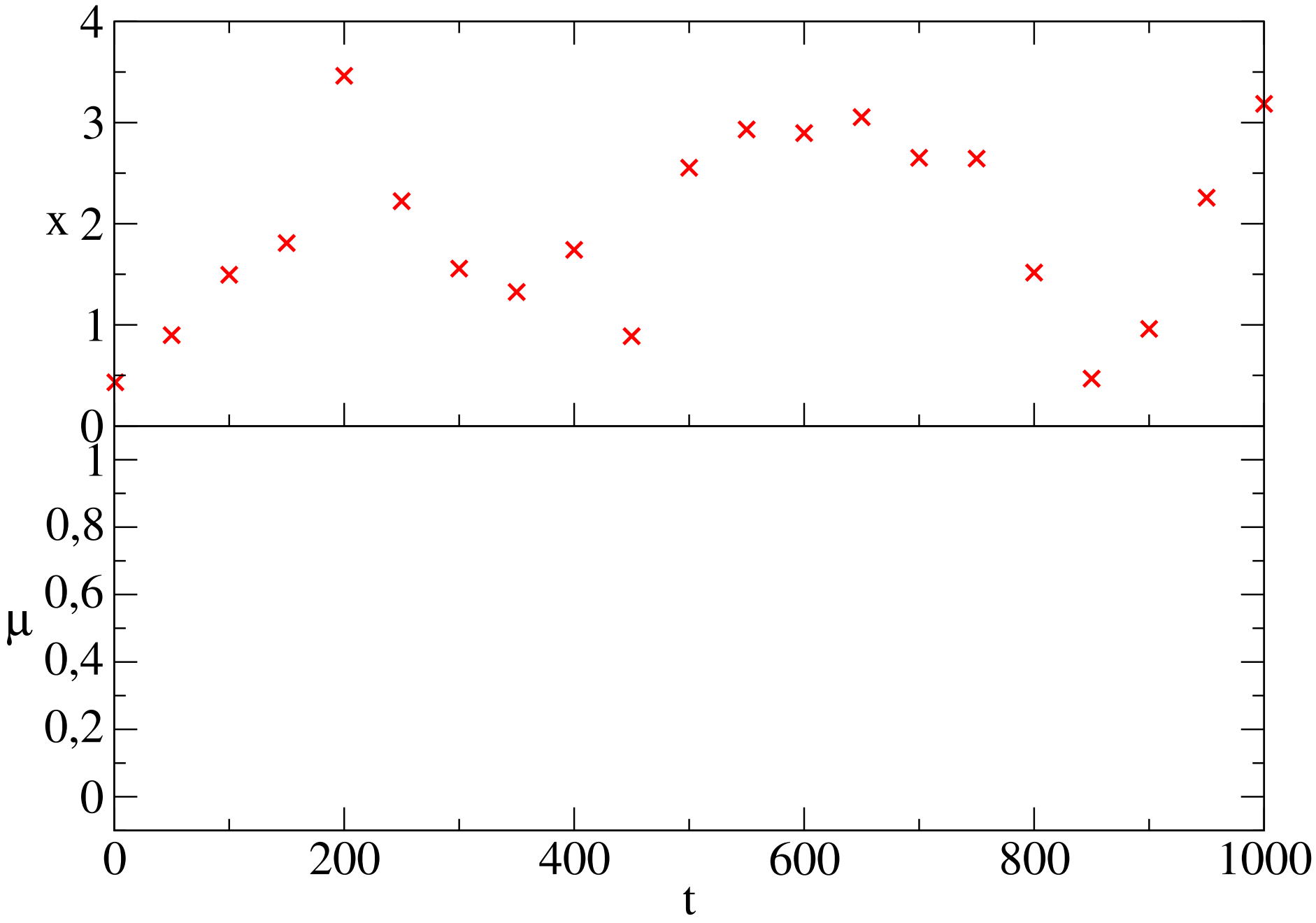
$$dX(t) = (A(t) + b - \lambda X(t))dt + \sigma dW(t)$$

- $A(t) =$ fast switching transcription factor activity (unobserved) modelled by $A(t) \sim \mathcal{TP}(f_{\pm})$ a **random telegraph process**.

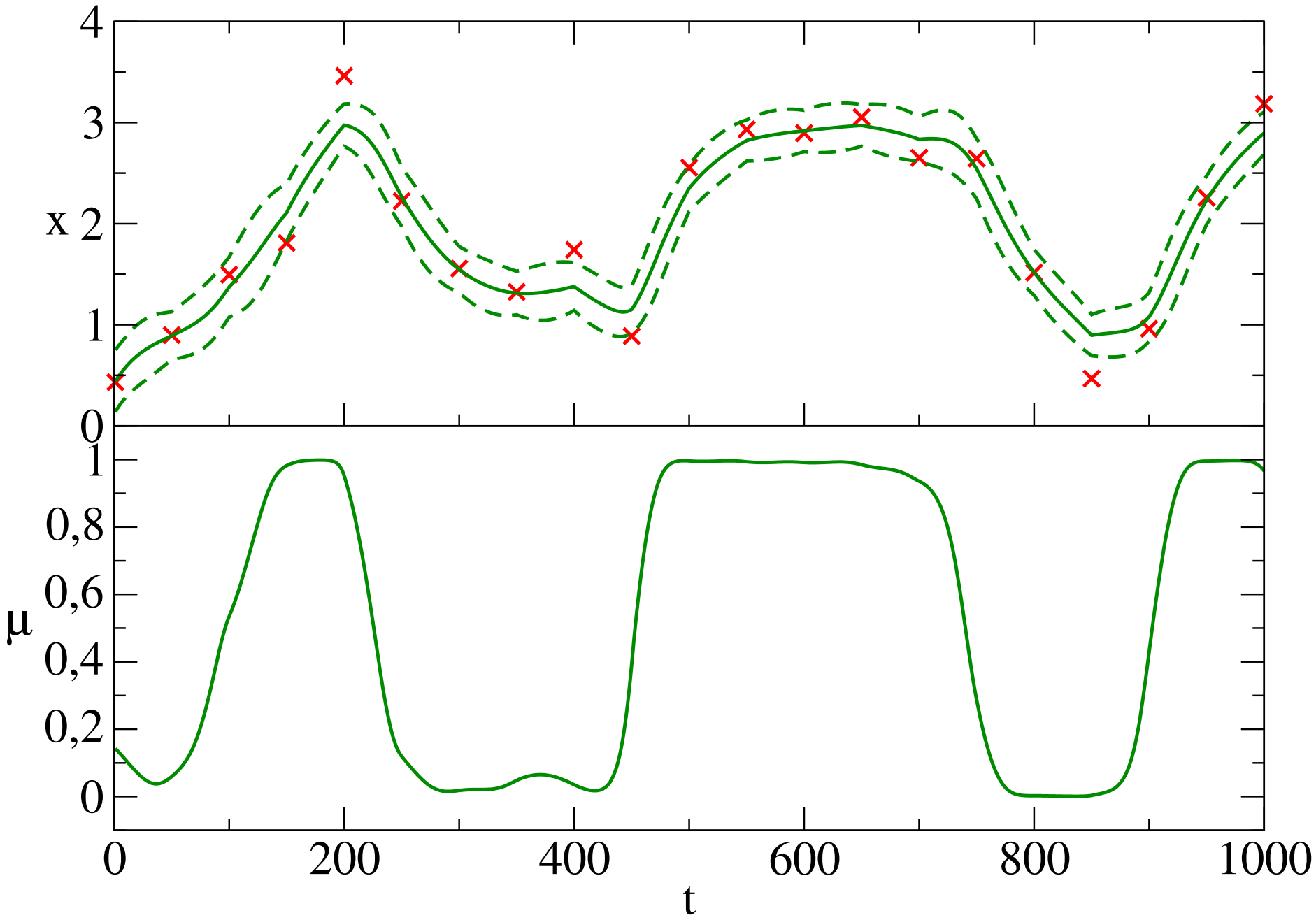


Simulation of processes


4
3
x 2
1
0
1
0,8
0,6
μ
0,4
0,2
0
0
200
400
600
800
1000
t



(Noisy) observations at discrete times.



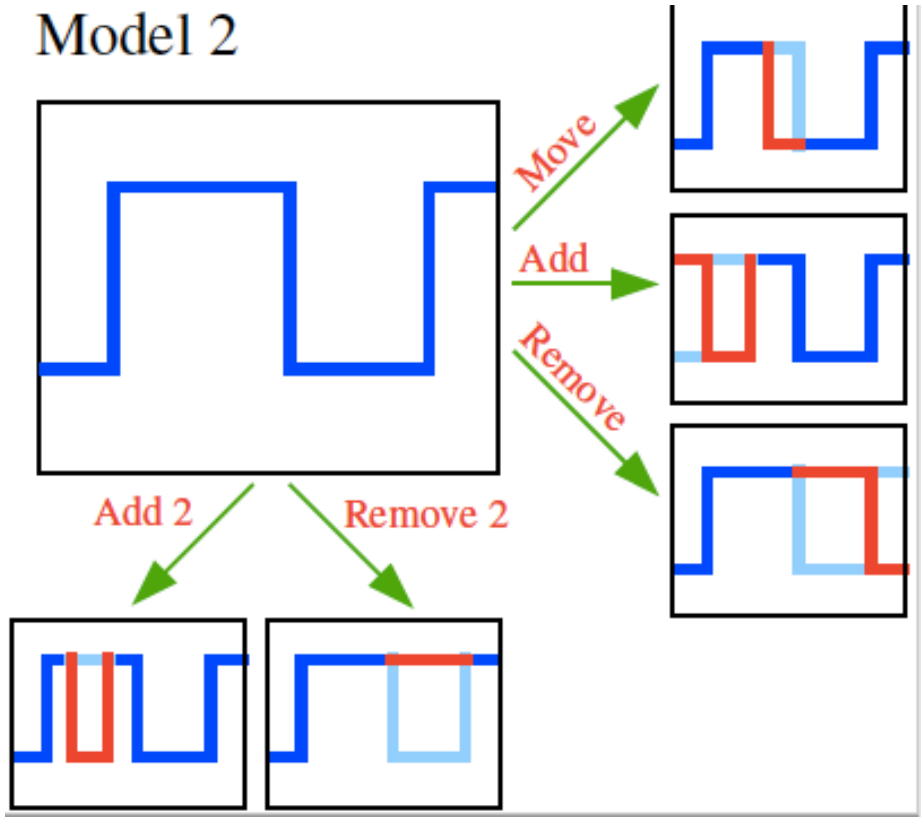
Inference of state variable.

MCMC sampler

- Joint process $X(t), A(t)$ Markov, but hard to sample from $p(X(0:T), A(0:T)|\mathbf{Y})$.

- Integrate out simple process $X(0:T)$ analytically given observations $\mathbf{Y}$ and $A(0:T)$.

- Sampling from $p(A(0:T)|\mathbf{Y})$ efficient if number of jumps small: Use **Metropolis–Hastings sampler:** Generate proposal changes of $A(0:T)$ (piecewise constant) and accept/reject with appropriate probabilities.


Model 2
Move
Add
Remove
Add 2
Remove 2

From Generalized Linear Models to GPs

Consider

$$f(\mathbf{x}) = \sum_{k=1}^{K} w_k \phi_k(\mathbf{x})$$

with *nonlinear functions* $\phi_k(\mathbf{x})$ of $\mathbf{x}$, but which is is *linear in the parameters* w_k!

A zero mean Gaussian prior $p(\mathbf{w}) = \prod_{k=1}^{d} \left(\frac{1}{\sqrt{2\pi\lambda_k}} e^{-\frac{w_k^2}{2\lambda_k}} \right)$ induces a Gaussian prior distribution over **the space of functions** $f(\mathbf{x})$ making f a zero mean **Gaussian process** with covariance **kernel**

$$K(\mathbf{x}, \mathbf{x}') = E[f(\mathbf{x}) f(\mathbf{x}')] = \sum_{k=1}^{K} \lambda_K \phi_k(\mathbf{x}) \phi_k(\mathbf{x}') = \sum_{k=1}^{k} \Psi_k(\mathbf{x}) \Psi_k(\mathbf{x}') \quad (11)$$

with $\Psi_k(\mathbf{x}) = \sqrt{\lambda_k}\phi_k(\mathbf{x})$. For proper choices of ϕ_k and λ_k, we can study models with $K = \infty$!

Gaussian Processes

Family (possibly infinite) of random variables $f(x)$, such that for any finite collection $\mathbf{f} = (f(x_1), f(x_2), \ldots, f(x_n))$ their joint distribution is Gaussian.

- For zero mean, this joint density reads:

$$p(\mathbf{f}) = \frac{1}{(2\pi)^{\frac{n}{2}}|\mathbf{K}|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}\mathbf{f}^T\mathbf{K}^{-1}\mathbf{f}\right\}$$

  with the covariance matrix $K_{ij} = K(x_i, x_j)$.

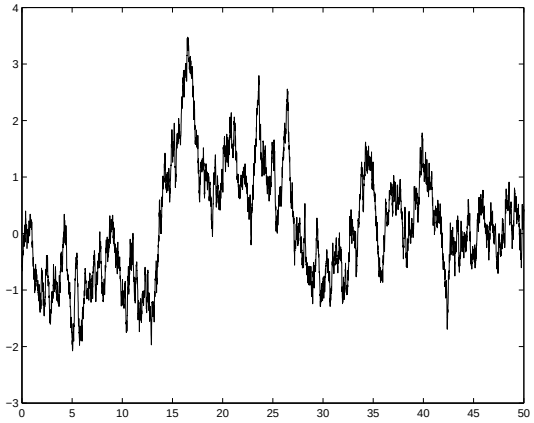
- Prior distribution depends only on the kernel $K(x, x')$, NOT on the individual functions ϕ_k. Hence, as a simple alternative start by defining a sensible covariance kernel K for GPs. This must be a positive semidefinite kernel which means that for any vector $\mathbf{a} = (a_1, \ldots, a_n)$, and any n we have $\mathbf{a}^\top\mathbf{K}\mathbf{a} \geq 0$. *Mercer's theorem* then guarantees that the kernel $K(x, x')$ will always have a representation (11), for some **often infinite dimensional set** ϕ_k and we can construct a Gaussian process from this. But there is no real need to find these explicitly, because all computations can be done using the kernel directly !

- Example: Stationary kernels $K(x - x')$ constructed from $K(x) = \int_{-\infty}^{\infty} e^{i\omega x}\widehat{K}(\omega)\, d\omega$ with non-negative $\widehat{K}(\omega) > 0$.

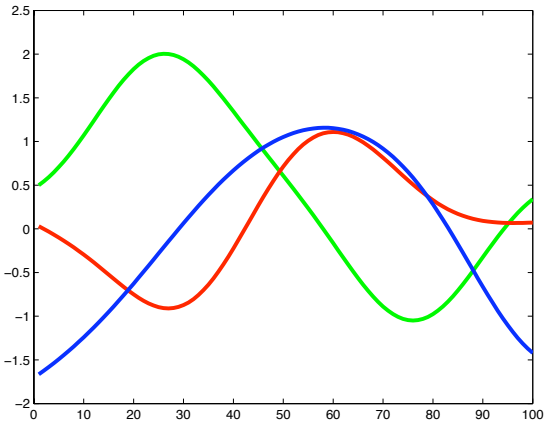
Samples from the GP prior

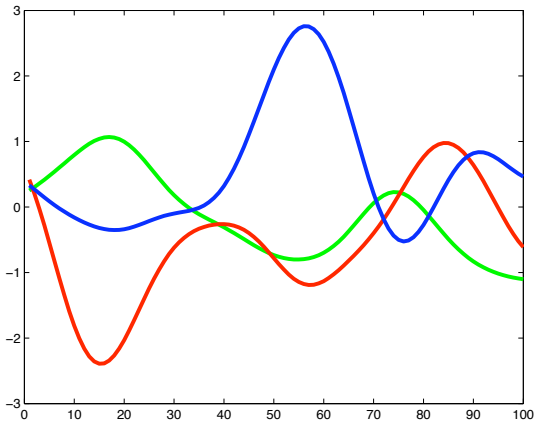
By choosing specific kernels we can express our prior belief or knowledge about the typical shape (smoothness) of the functions $f(x)$.

Samples from a GP with $K(x, x') = e^{-|x-x'|}$



3 random samples from GPs with $K(x, x') = e^{-3(x-x')^2}$ and $K(x, x') = e^{-10(x-x')^2}$





Of course, kernels can be constructed for d dimensional inputs $\mathbf{x} = (x(1), x(2), x(3), \ldots, x(d))$ where $x(i)$ is the i-the coordinate of $\mathbf{x}$. A popular choice is the *RBF–kernel*

$$K(\mathbf{x}, \mathbf{x}') = \prod_{k=1}^{d} e^{-\lambda_k (x(k) - x'(k))^2}$$

allowing for different *hyperparameters* (lengthscales) λ_k.

Gaussian Process Regression

Assume a Gaussian noise model with a likelihood

$$P(D|f(x)) \propto \exp\left[-\sum_i \frac{1}{\sigma^2}(y_i - f(x_i))^2\right]$$

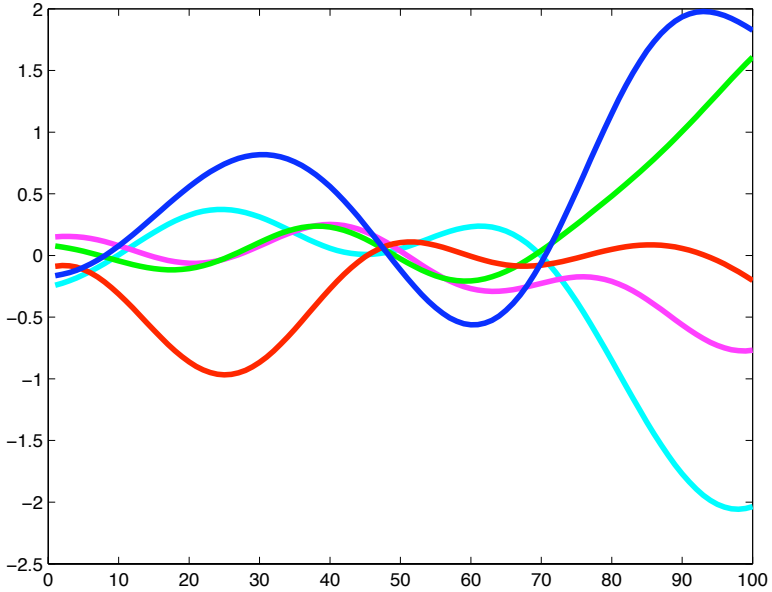
Given the training set $D = \{y(x_1), y(x_2), \ldots, y(x_n)\}$ and **test point** x, we are interested in the *posterior density* of the unknown function values $f(x_i)$ which we denote by the augmented vector $\mathbf{f}_+ = (\mathbf{f}, f(x))^T$. Defining $\mathbf{k} = (K(x, x_1), K(x, x_2), \ldots, K(x, x_n))^T$ and the covariance matrix $\mathbf{K}_+ = \begin{pmatrix} \mathbf{K} & \mathbf{k}^\top \\ \mathbf{k} & K(x,x) \end{pmatrix}$, we get

$$p(\mathbf{f}_+|D) \propto \exp\left[-\frac{1}{2}\mathbf{f}_+^T\mathbf{K}_+^{-1}\mathbf{f}_+ - \frac{1}{\sigma^2}\sum_{i=1}^{n}(y_i - f(x_i))^2\right] \qquad (12)$$

Samples from the GP posterior

After we observe data $D = (y(x_1), \ldots, y(x_n))$, the uncertainty changes. The form of eq. (12) remains essentially the same for an arbitrary number of augmented test points $\mathbf{x}$. By sampling from the *joint* posterior of function values $\{f(x_{j,\text{test}})\}_{j=1}^{M}$ for a large number of (test) points, we can display the typical shape of random functions from the posterior.

5 Samples from a GP posterior with $K(x, x') = e^{-3(x-x')^2}$ and 3 datapoints:
$y(0.1) = y(0.5) = y(0.7) = 0$ and noise $\sigma^2 = 0.01$ obtained at $M = 100$ equidistant input points.


2
1.5
1
0.5
0
−0.5
−1
−1.5
−2
−2.5
0
10
20
30
40
50
60
70
80
90
100

Marginalisation & Conditioning

Let

$$p(x,y) \propto \exp\left[-\frac{1}{2}(x\ y)^\top \Omega\ (x\ y) + (x\ y)^\top \xi\right]$$

with the information matrix $\Omega = \begin{pmatrix} \Omega_{xx} & \Omega_{xy} \\ \Omega_{yx} & \Omega_{yy} \end{pmatrix}$ and $\xi = (\xi_x\ \xi_y)^\top$.

Then

$$\begin{aligned} \Sigma &= \Omega^{-1} \\ \mu &= \Sigma\xi \end{aligned}$$

The marginal of x is

$$p(x) \propto \exp\left[-\frac{1}{2}x^\top \bar{\Omega}_{xx}\ x + x^\top \bar{\xi}_x\right]$$

where

$$\begin{aligned} \bar{\Omega}_{xx} &= \Omega_{xx} - \Omega_{xy}\Omega_{yy}^{-1}\Omega_{yx} \\ \bar{\xi}_x &= \xi_x - \Omega_{xy}\Omega_{yy}^{-1}\xi_y \end{aligned}$$

and the conditional density

$$p(x|y) \propto \exp\left[-\frac{1}{2}x^\top \Omega_{xx}\ x + x^\top(\xi_x - \Omega_{xy}y)\right]$$

Inverse of partitioned matrix

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix}^{-1} = \begin{pmatrix} M & -MBD^{-1} \\ -D^{-1}CM & D^{-1} + D^{-1}CMBD^{-1} \end{pmatrix}$$

with

$$M = \left(A - BD^{-1}C\right)^{-1}$$

Predictions & Uncertainty

The *posterior mean* prediction at x (which equals the MAP for this model) is

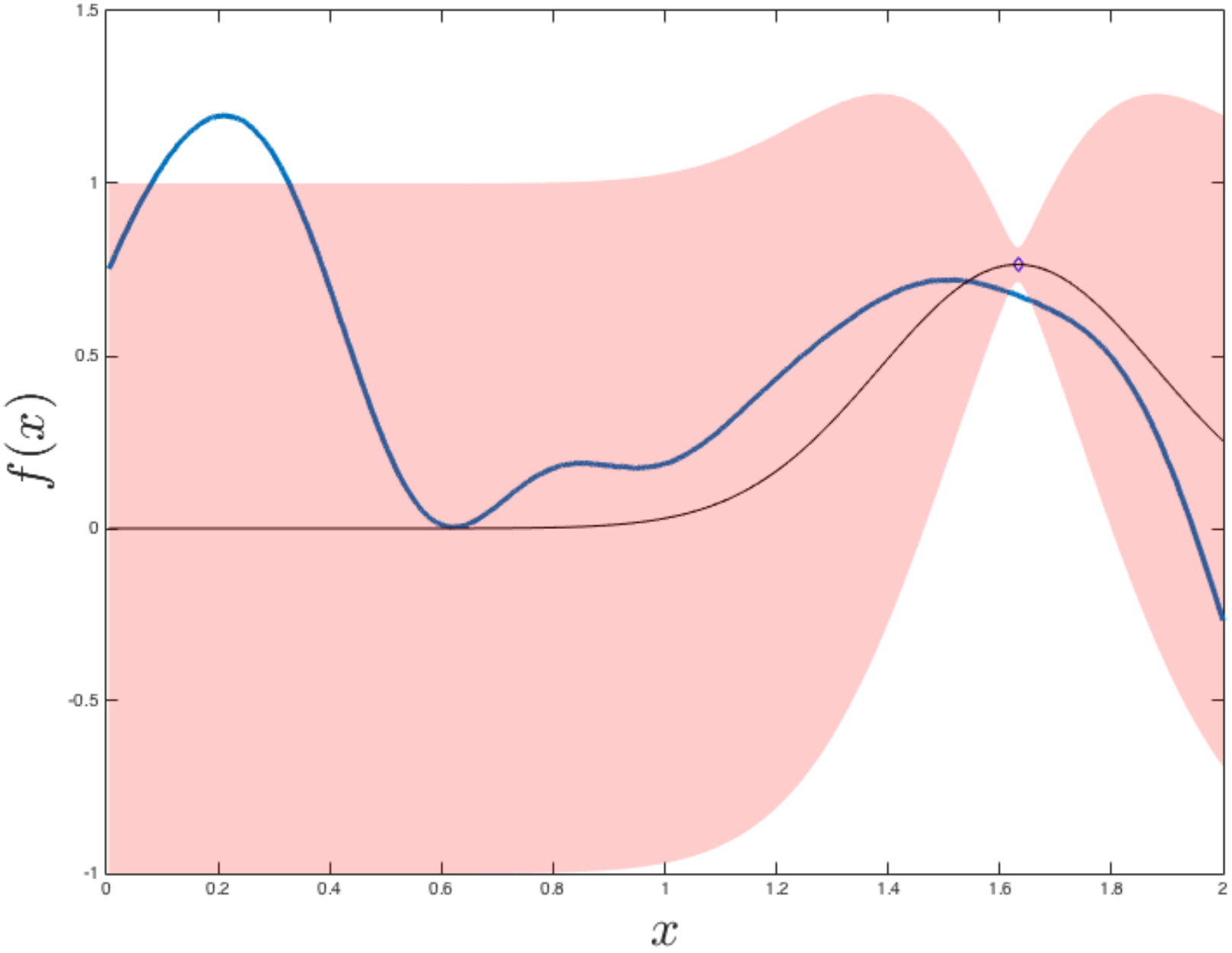
$$\widehat{f}(x) = \int d\mathbf{f}_+ \, p(\mathbf{f}_+) f(x) = \mathbf{k}^T \left(\mathbf{K} + \sigma^2 \mathbf{I}\right)^{-1} \mathbf{y}$$

with $\mathbf{y} = (y_1, y_2, \ldots, y_n)^T$. The posterior variance (Bayesian error bar) is

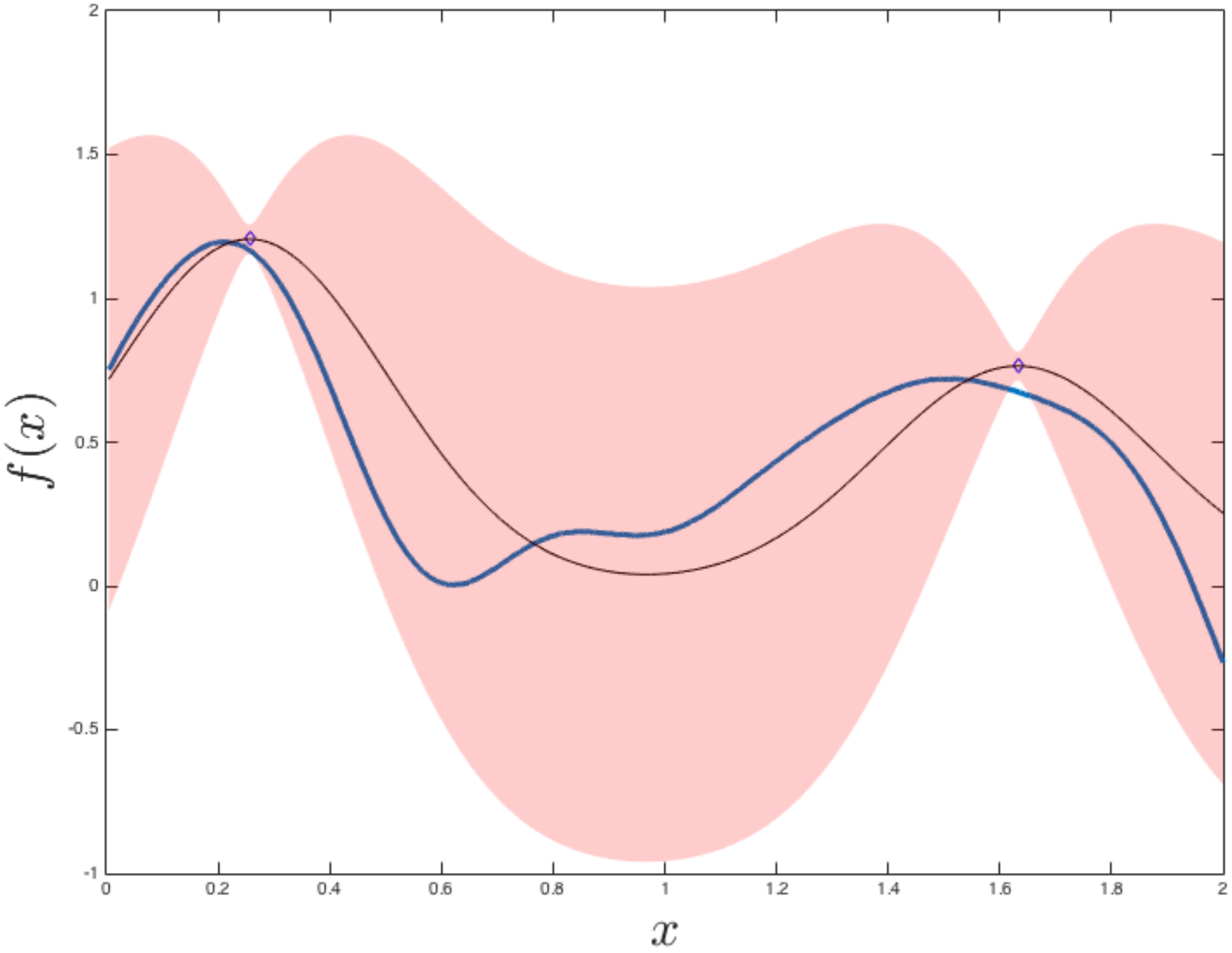
$$\sigma_n^2(x) = K(x,x) - \mathbf{k}^T \left(\mathbf{K} + \sigma^2 \mathbf{I}\right)^{-1} \mathbf{k}$$

This gives a measure for the uncertainty of the prediction at point x.

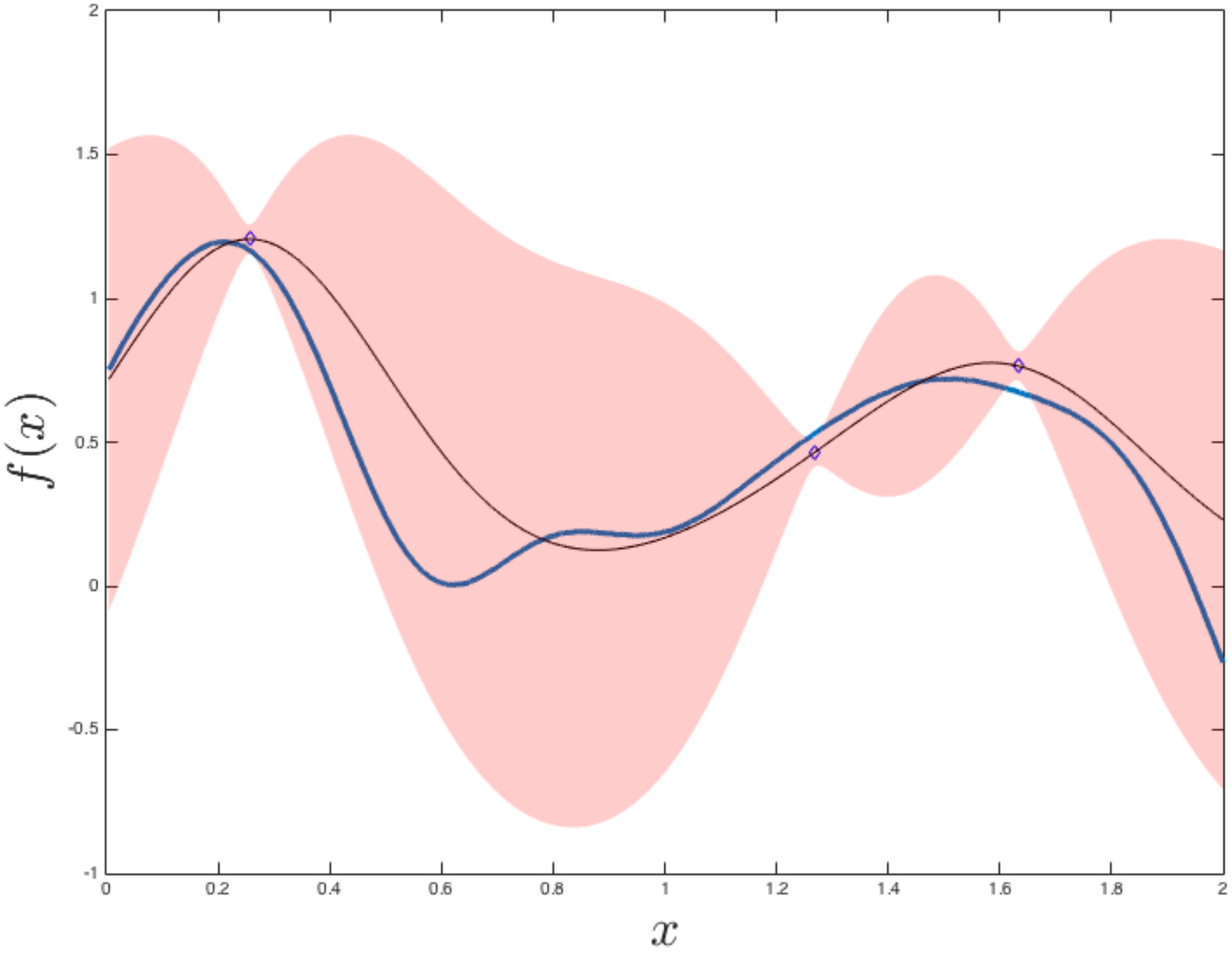
1 observation



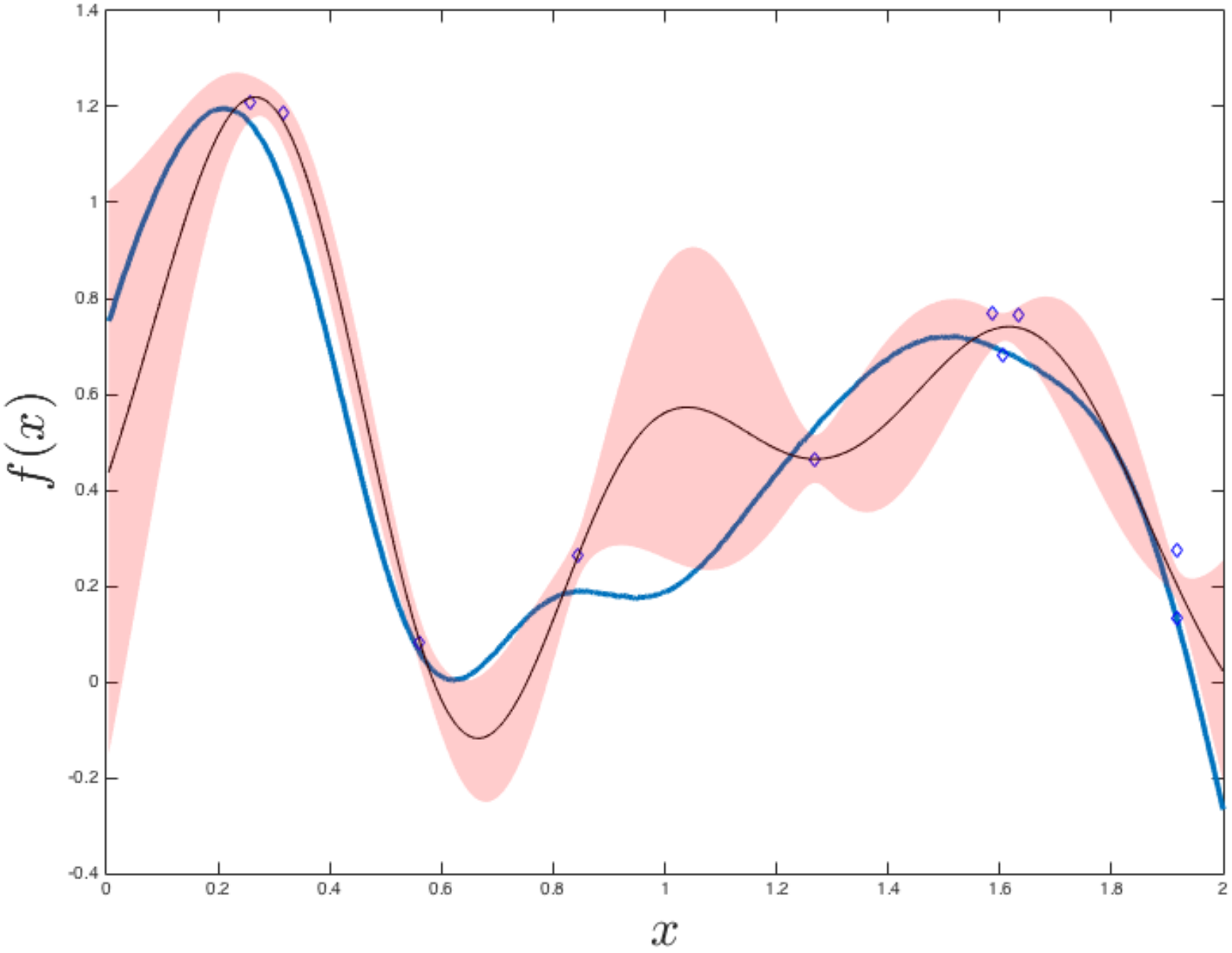
2 observations



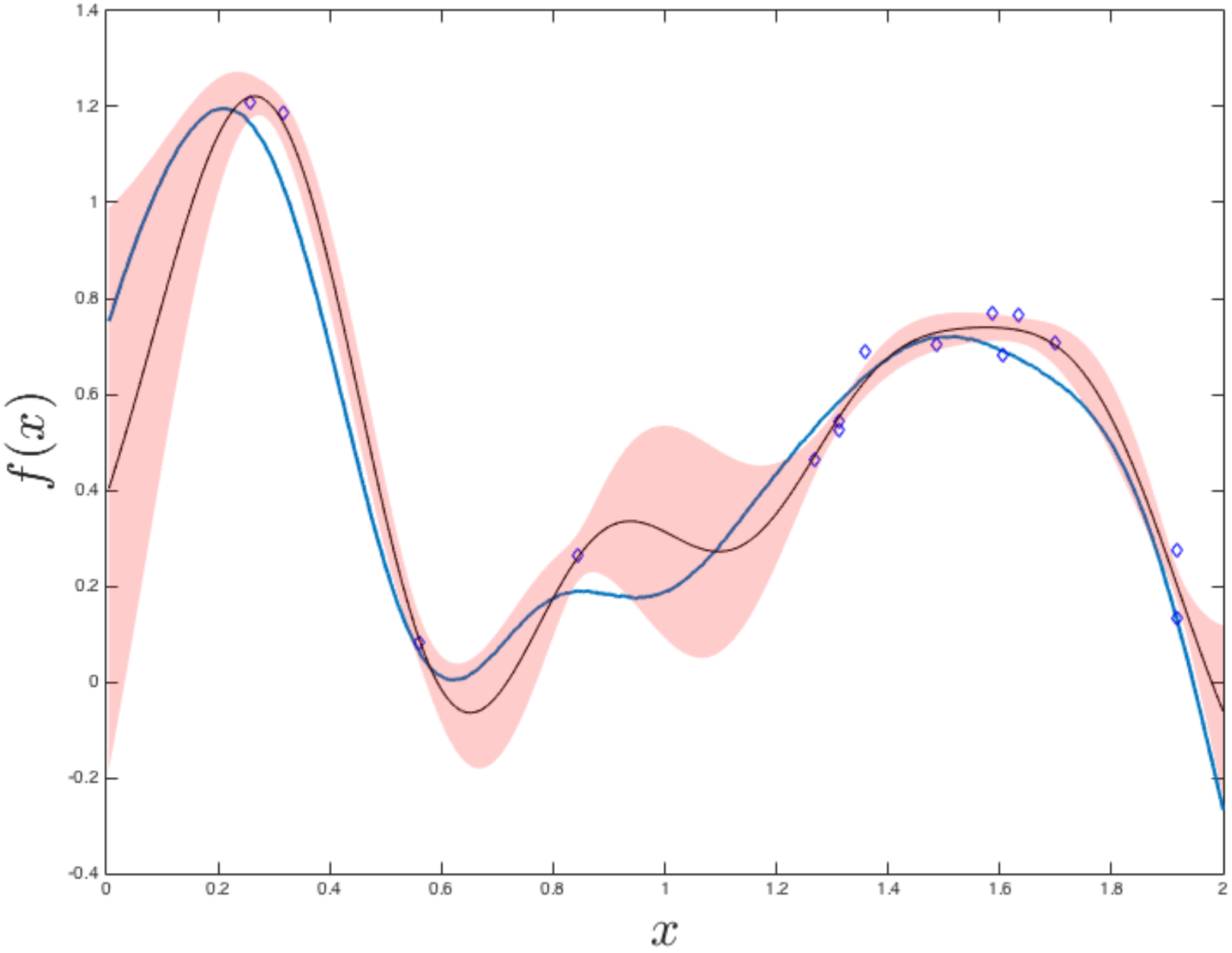
3 observations



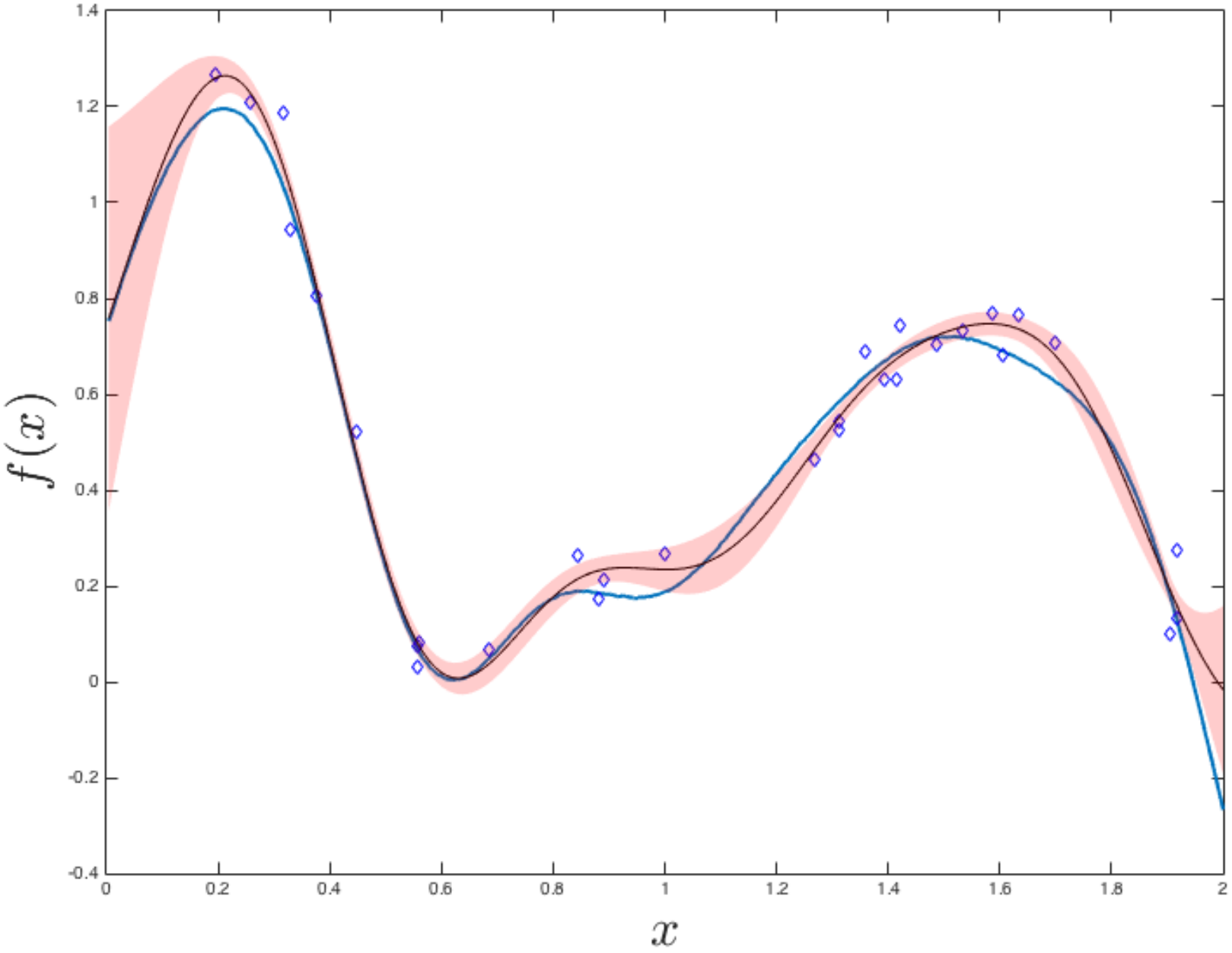
10 observations



15 observations



30 observations

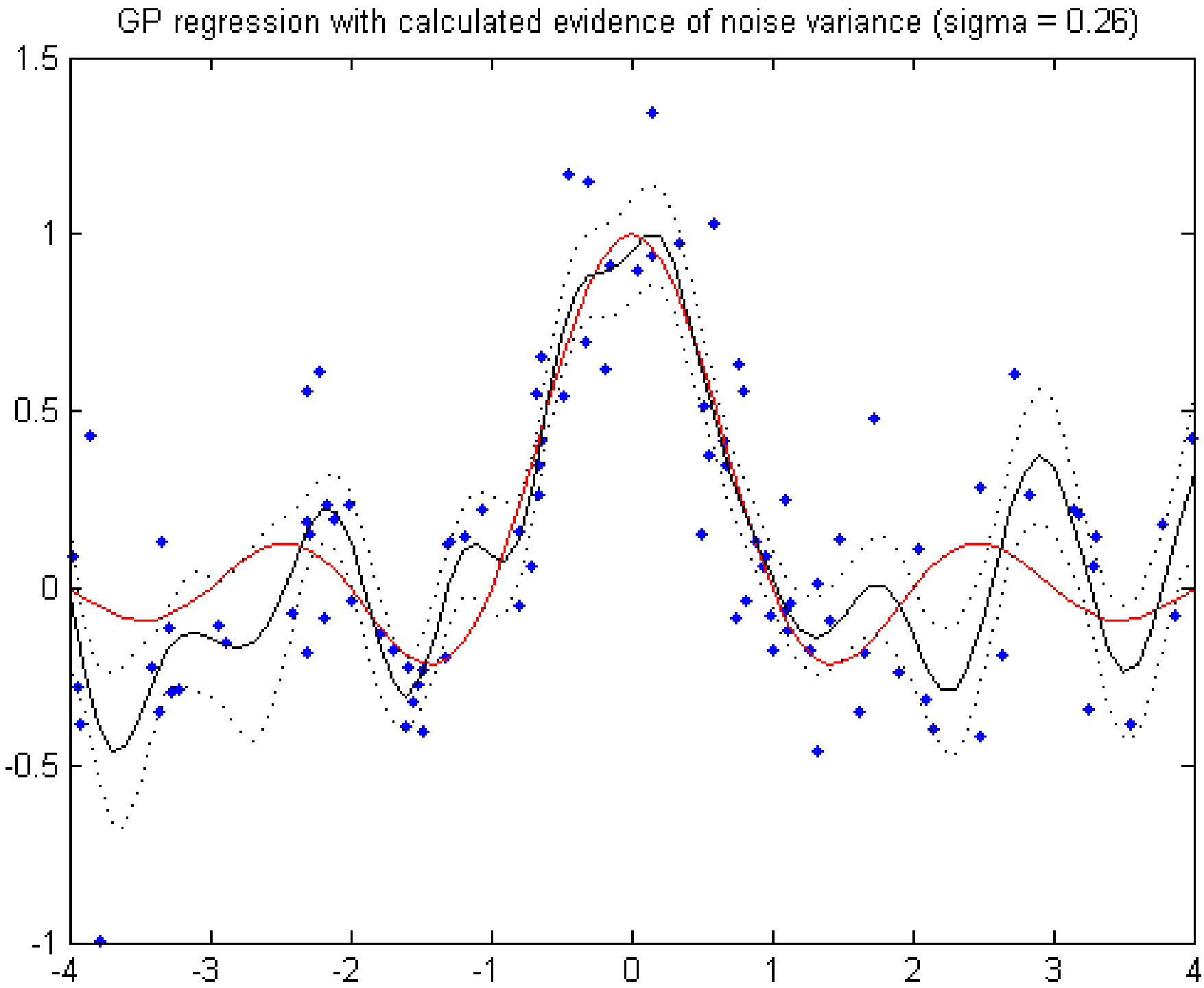


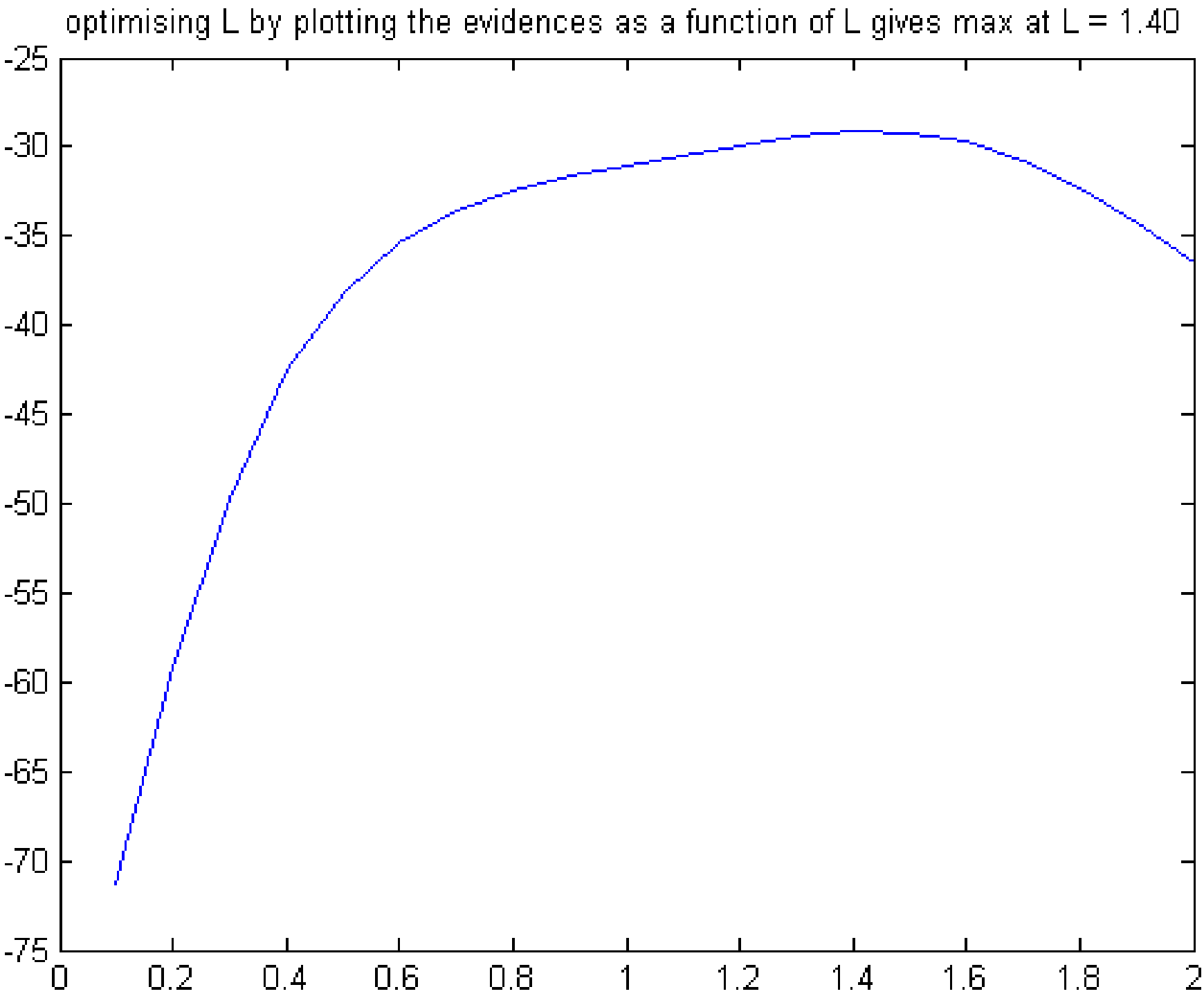
Model selection using the evidence

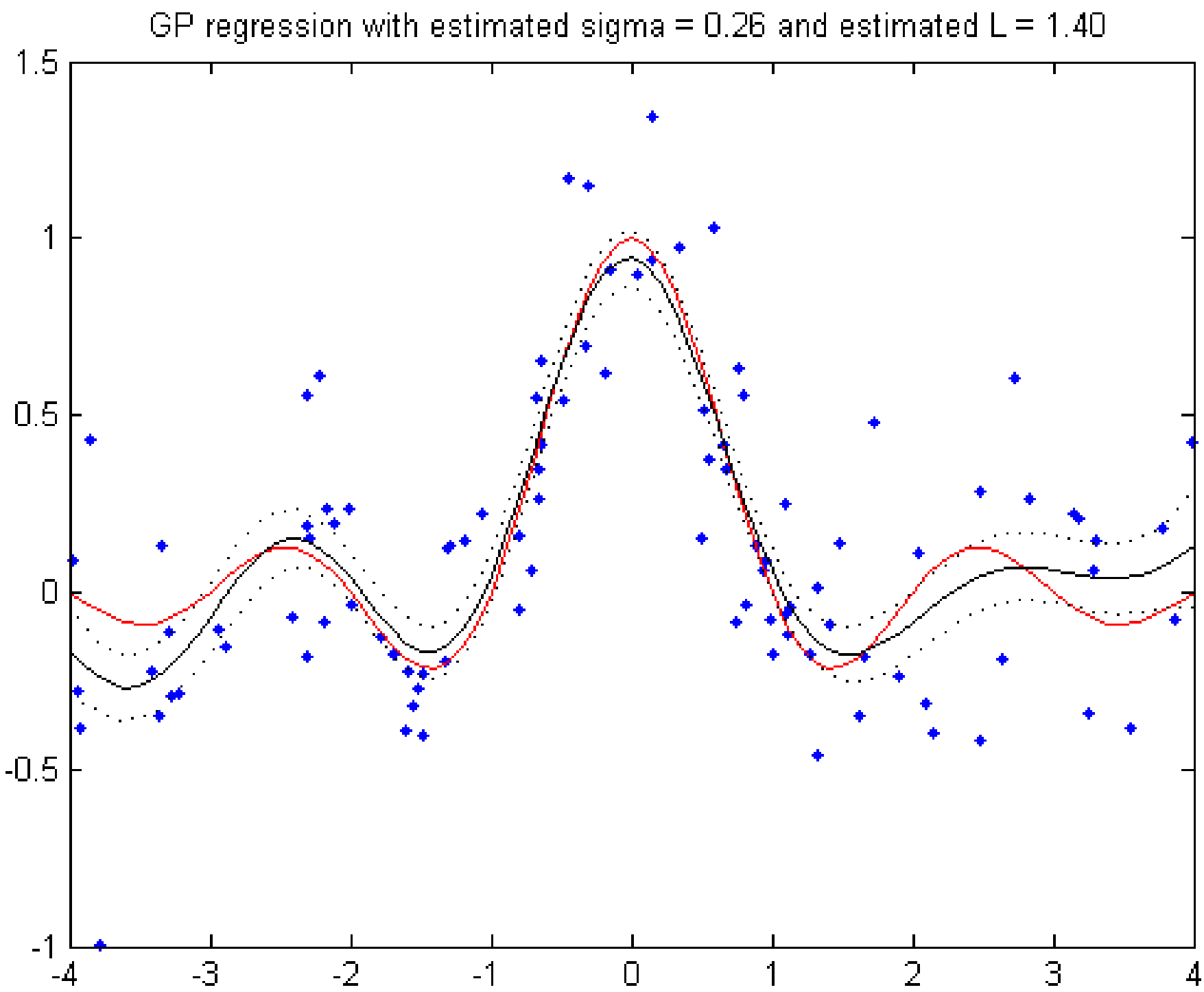
Sensible values for **kernel hyperparameters** and noise σ^2 can be obtained by numerically maximising the evidence
(Maximum Likelihood II)

$$\begin{aligned} p(D) &= \\ &= \int d\mathbf{f}\; p(\mathbf{f})\; p(D|\mathbf{f}) \\ &= \frac{1}{(2\pi)^{n/2}|\det(\mathbf{K}+\sigma^2\mathbf{I})|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}\mathbf{y}^T(\mathbf{K}+\sigma^2\mathbf{I})^{-1}\mathbf{y}\right] \end{aligned}$$

Equivalent, we minimize $-\ln(p(D))$.


GP regression with calculated evidence of noise variance (sigma = 0.26)
1.5
1
0.5
0
-0.5
-1
-4
-3
-2
-1
0
1
2
3
4


optimising L by plotting the evidences as a function of L gives max at L = 1.40
-25
-30
-35
-40
-45
-50
-55
-60
-65
-70
-75
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2


GP regression with estimated sigma = 0.26 and estimated L = 1.40
1.5
1
0.5
0
-0.5
-1
-4
-3
-2
-1
0
1
2
3
4

Further properties

- The **predictor** is of the form $\widehat{f}(x) = \sum_i \alpha_i K(x, x_i)$ as for non–Bayesian kernel machines.

- **Automatic Relevance Determination (ARD)** : Consider

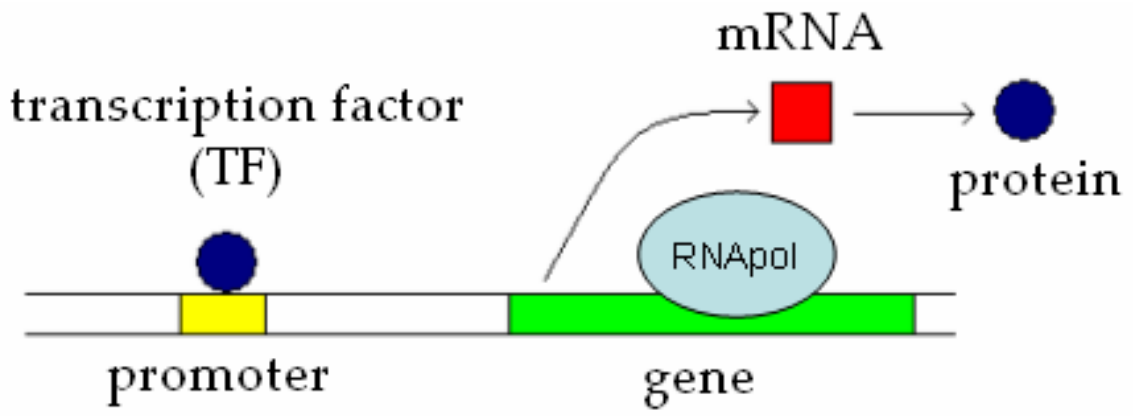
$$K(\mathbf{x}, \mathbf{x}') = \prod_{k=1}^{d} e^{-\lambda_k (x(k) - x'(k))^2}$$

  If evidence maximisation leads to $\lambda_i \to 0$ for some input features i, the corresponding input has **no influence** on the prediction.

- Easy to include *derivatives* of $f(x)$ or other linear functionals of f as (noisy) observations.

Inference of transcriptional regulation using Gaussian processes

(Lawrence, Sanguinetti & Rattray)



- Transcription factors regulate genes by binding to specific sites.
- Hard to measure transcription factor activity directly. Inference must be based on measurement of mRNA concentration of target genes at discrete times $y_{ik} = x_i(t_k) +$ noise.
- Model equation (ordinary differential equation)

$$\frac{dx_i}{dt} = B_i - D_i x_i(t) + S_i f(t)$$

where $f(t)$ is the transcription factor activity.

- Model $f(t)$ by a Gaussian process. Since the differential equation is linear, $x(t)$ and $f(t)$ are jointly Gaussian processes.

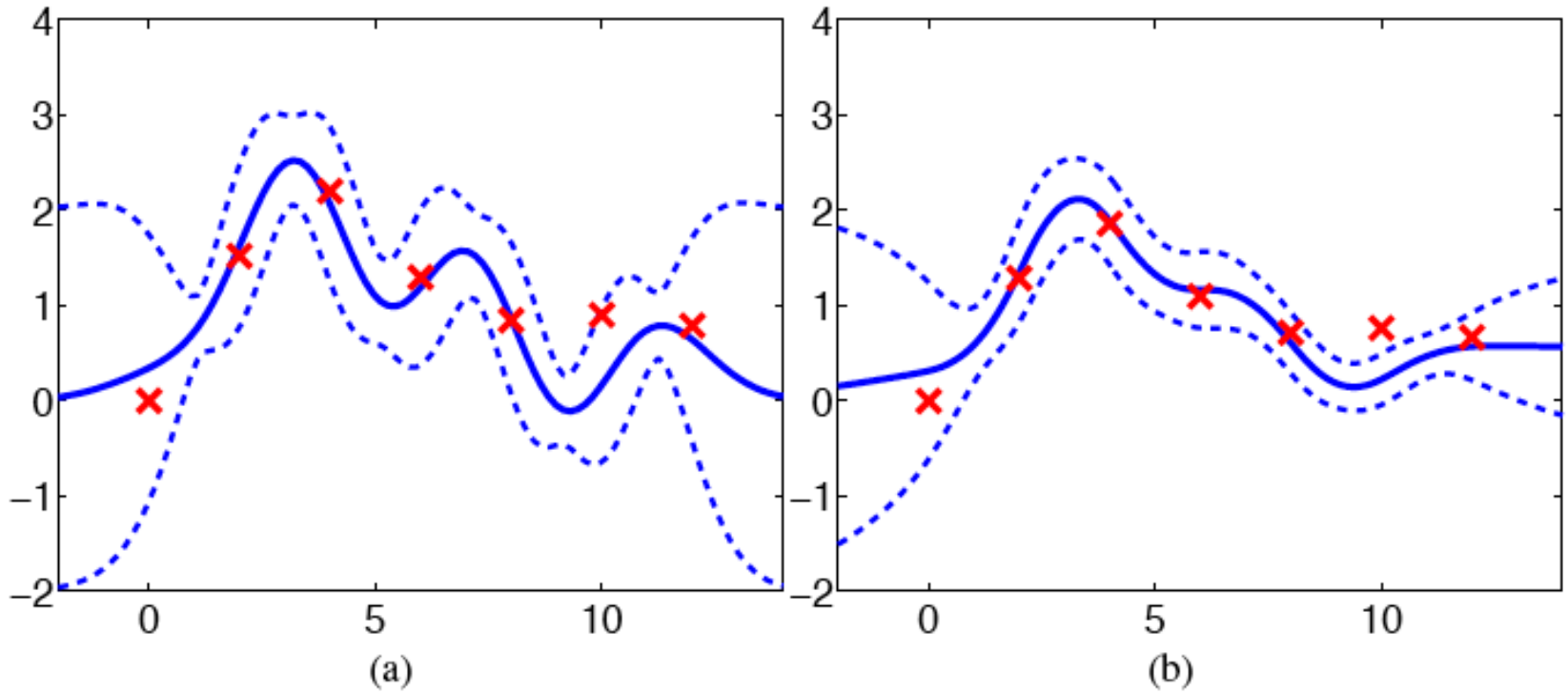


Figure 1: Predicted protein concentration for p53 using a linear response model: (a) squared exponential prior on f; (b) MLP prior on f. Solid line is mean prediction, dashed lines are 95% credibility intervals. The prediction of Barenco *et al.* was pointwise and is shown as crosses.

Linear ordinary differential equations

- Linear operations on GPs leads to GPs !

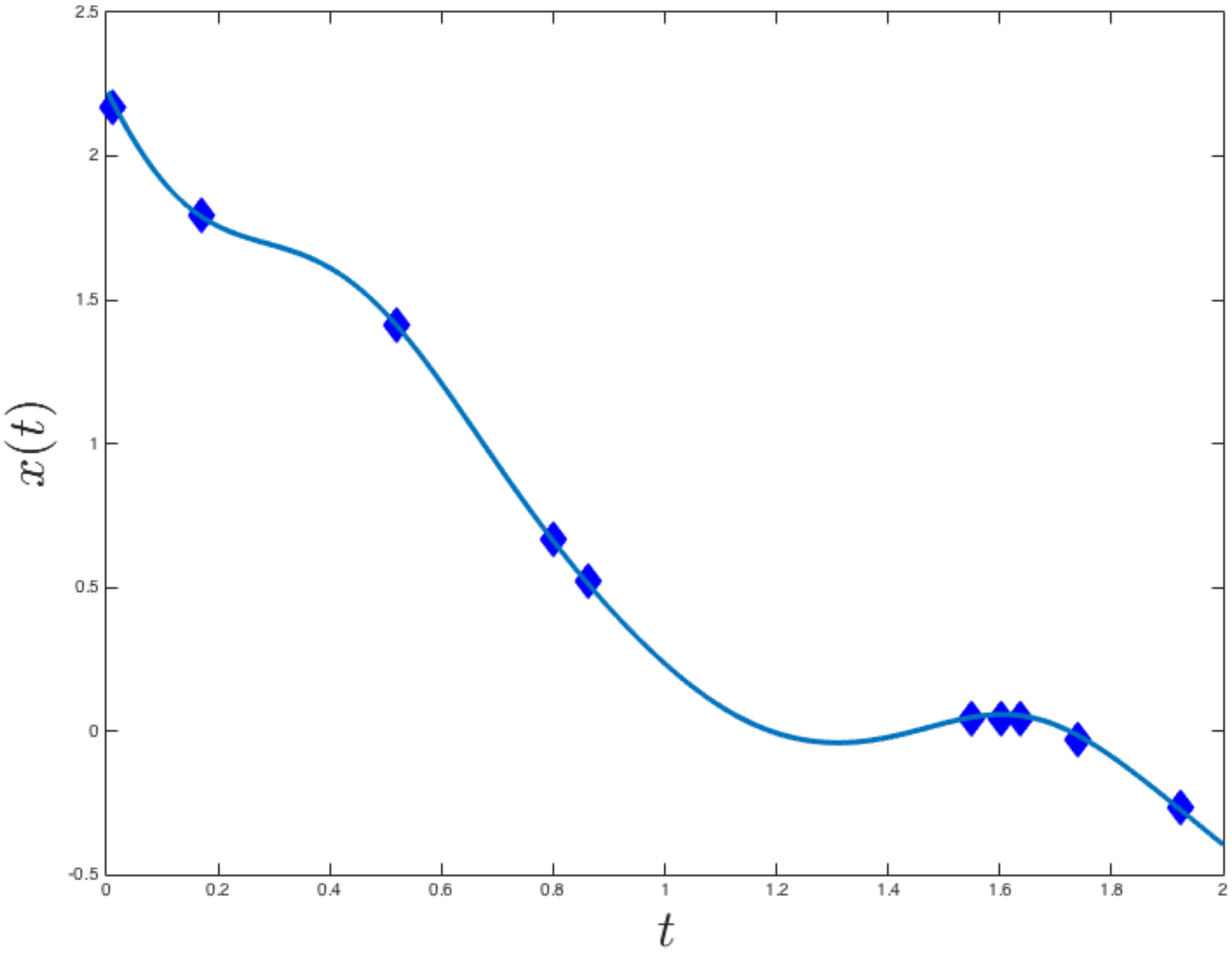
- A dynamical model

$$\begin{aligned} \frac{dx(t)}{dt} &= -\lambda x(t) + f(t) \\ y_i &= x(t_i) + \nu_i, \qquad i = 1, \ldots, n \end{aligned}$$

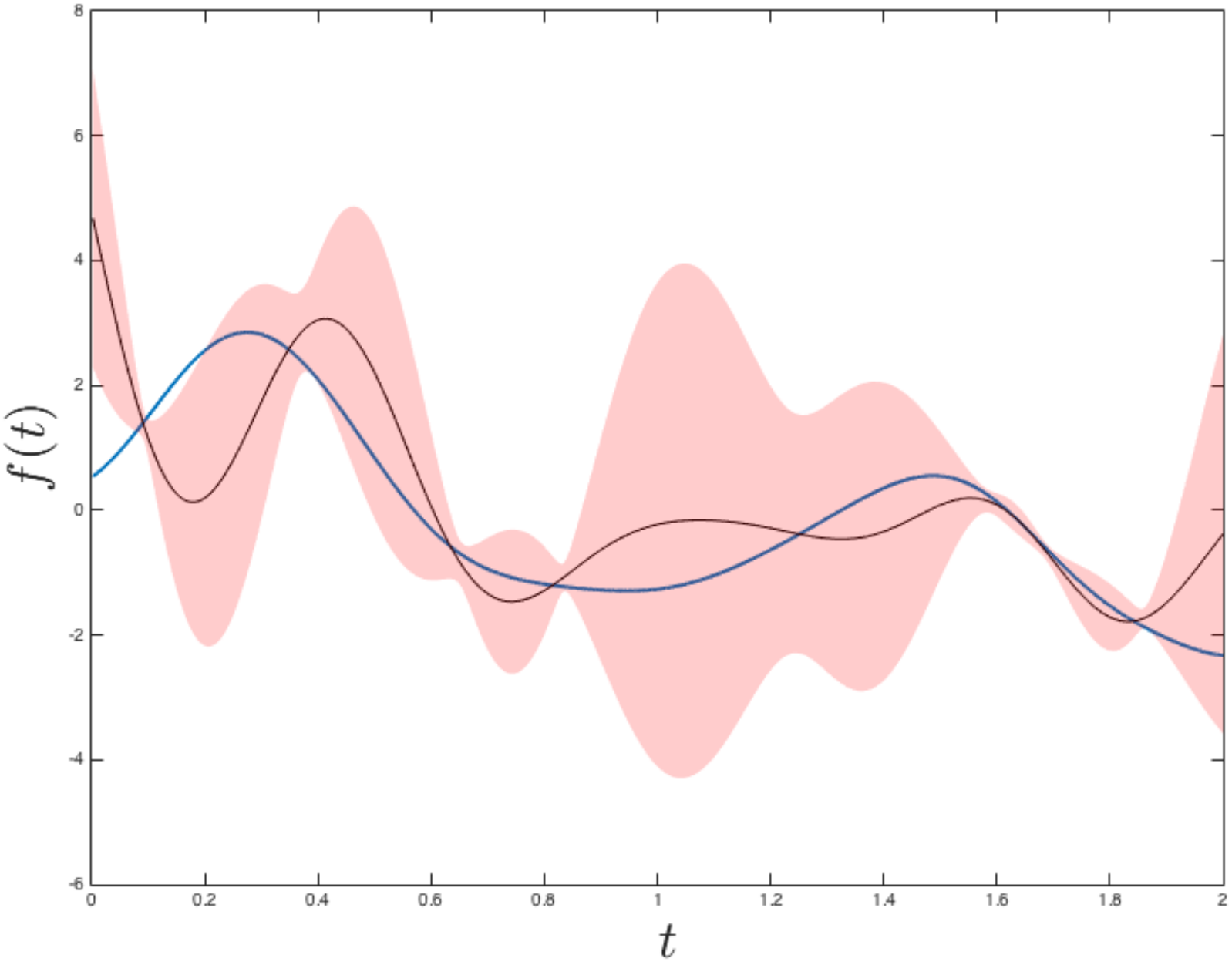
- $f(t)$ is an unknown function to be estimated.

- Use a GP prior over $f(\cdot)$.

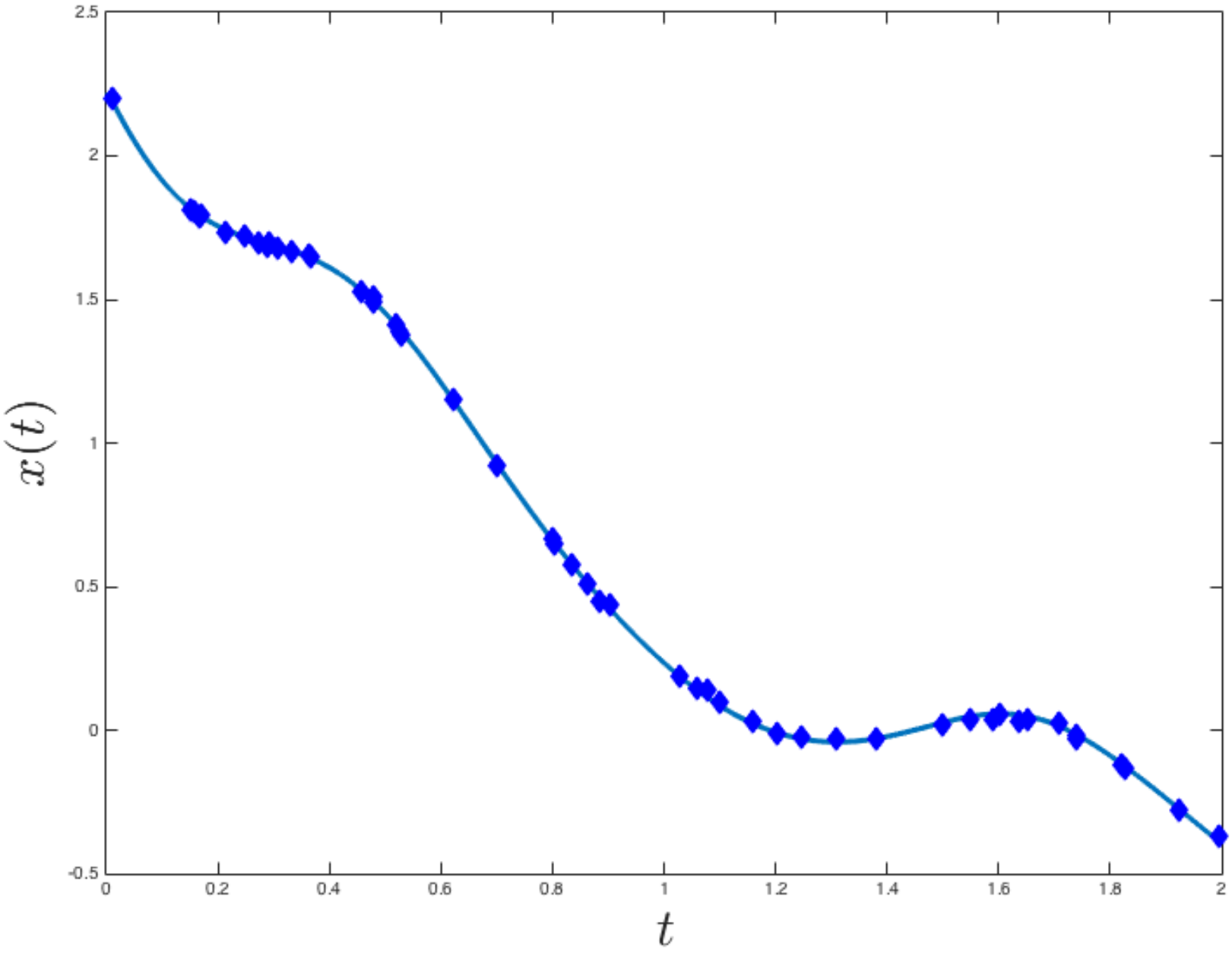
With 10 observations ...



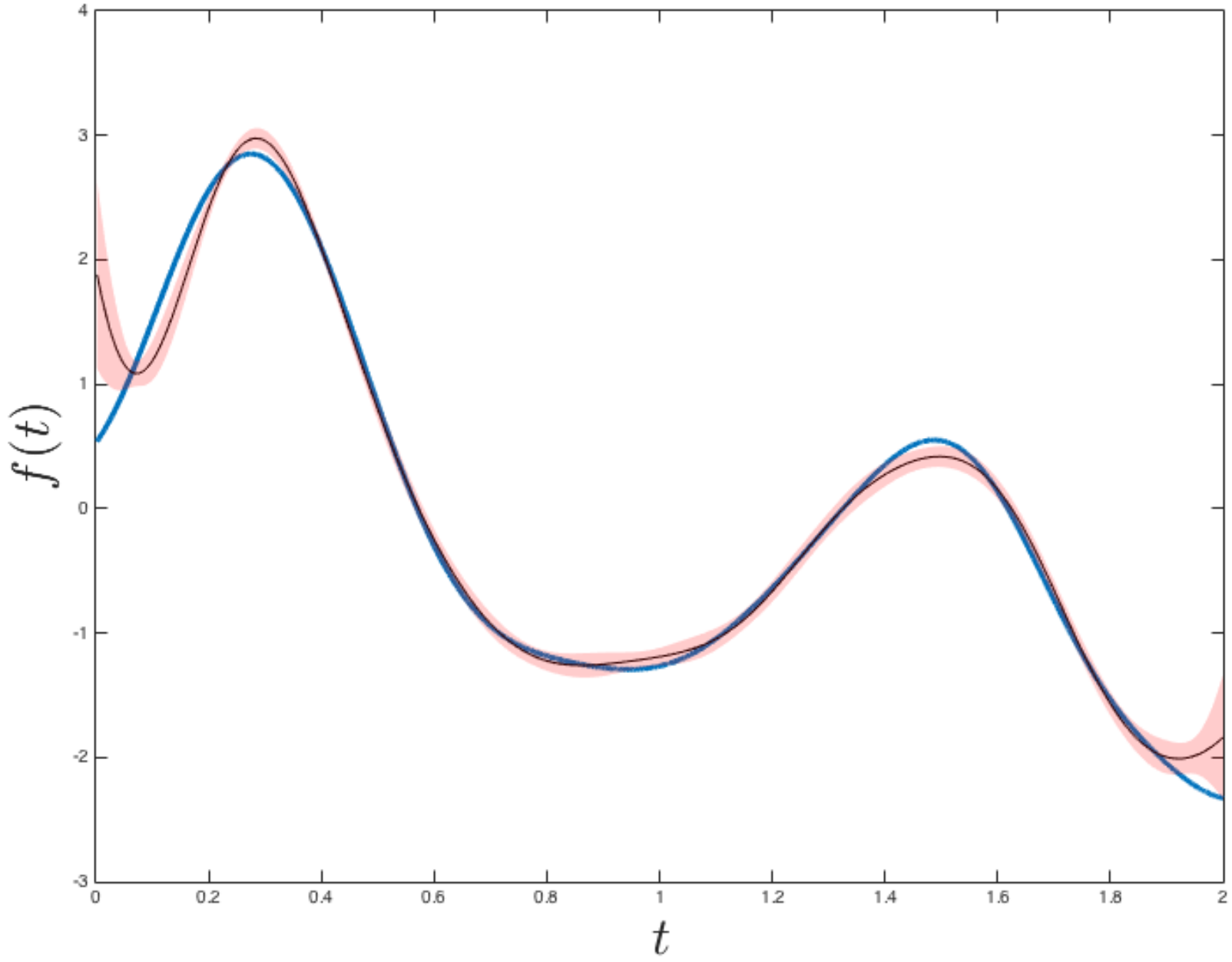
With 10 observations ...



With 50 observations ...



With 50 observations ...



GP Emulators

O'Hagan & Kennedy (see e.g.
`http://www.tonyohagan.co.uk/academic/GEM/index.html`

and the MUCM (MANAGING UNCERTAINTY IN COMPLEX MODELS) page
`http://www.mucm.ac.uk`

Emulate complex simulation software packages. These evaluate functions $y = f(x)$ using very lengthy computations.

Learn a Gaussian process approximation $y = m(x) + \mathcal{GP}(0, K)$ from a small set of data.

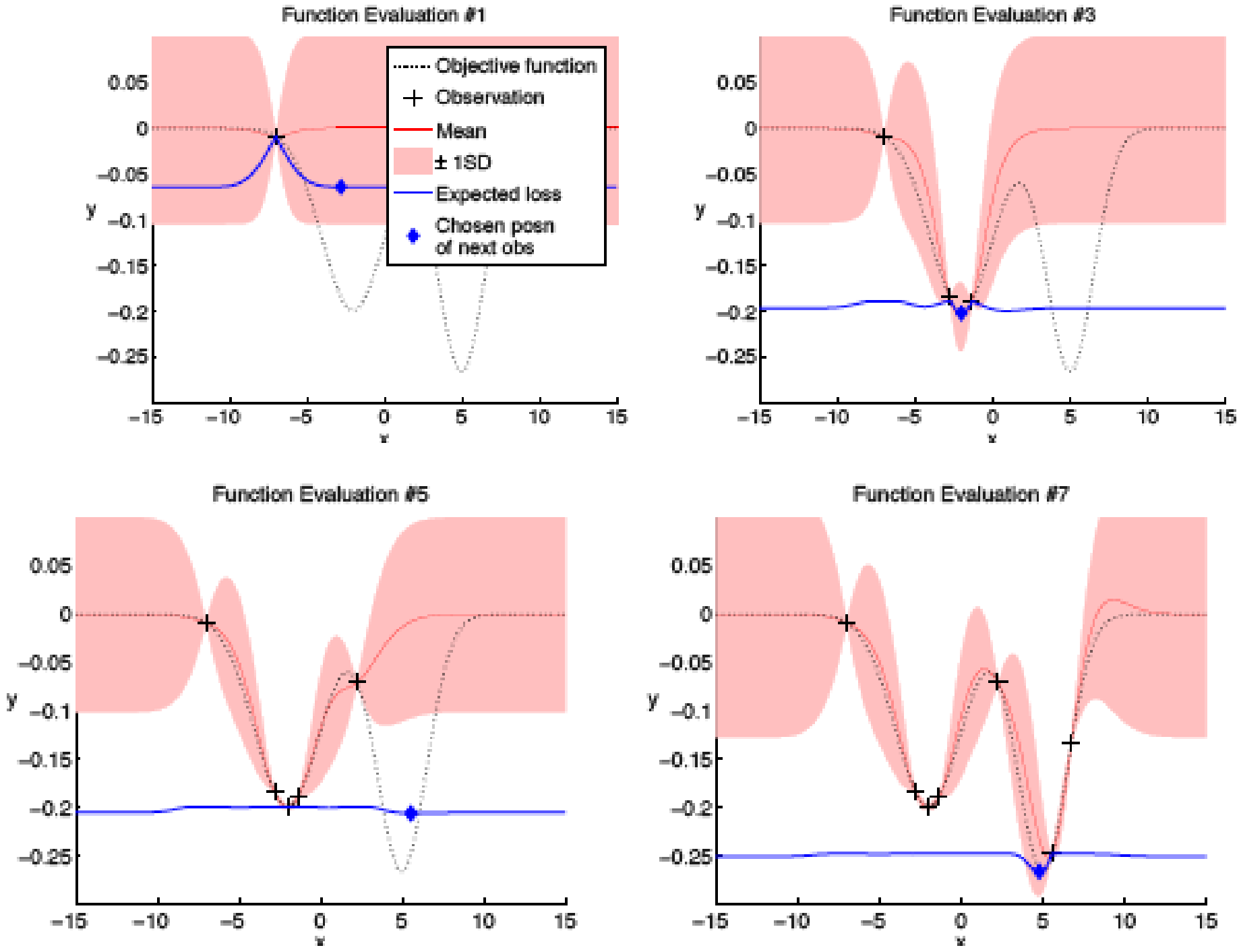
Sensitivity analysis: Changes of outputs under small input changes.

Uncertainty analysis: Uncertainty of outputs based on uncertainty in inputs modelled by distribution $p(x)$.

Gaussian Processes for Global Optimisation

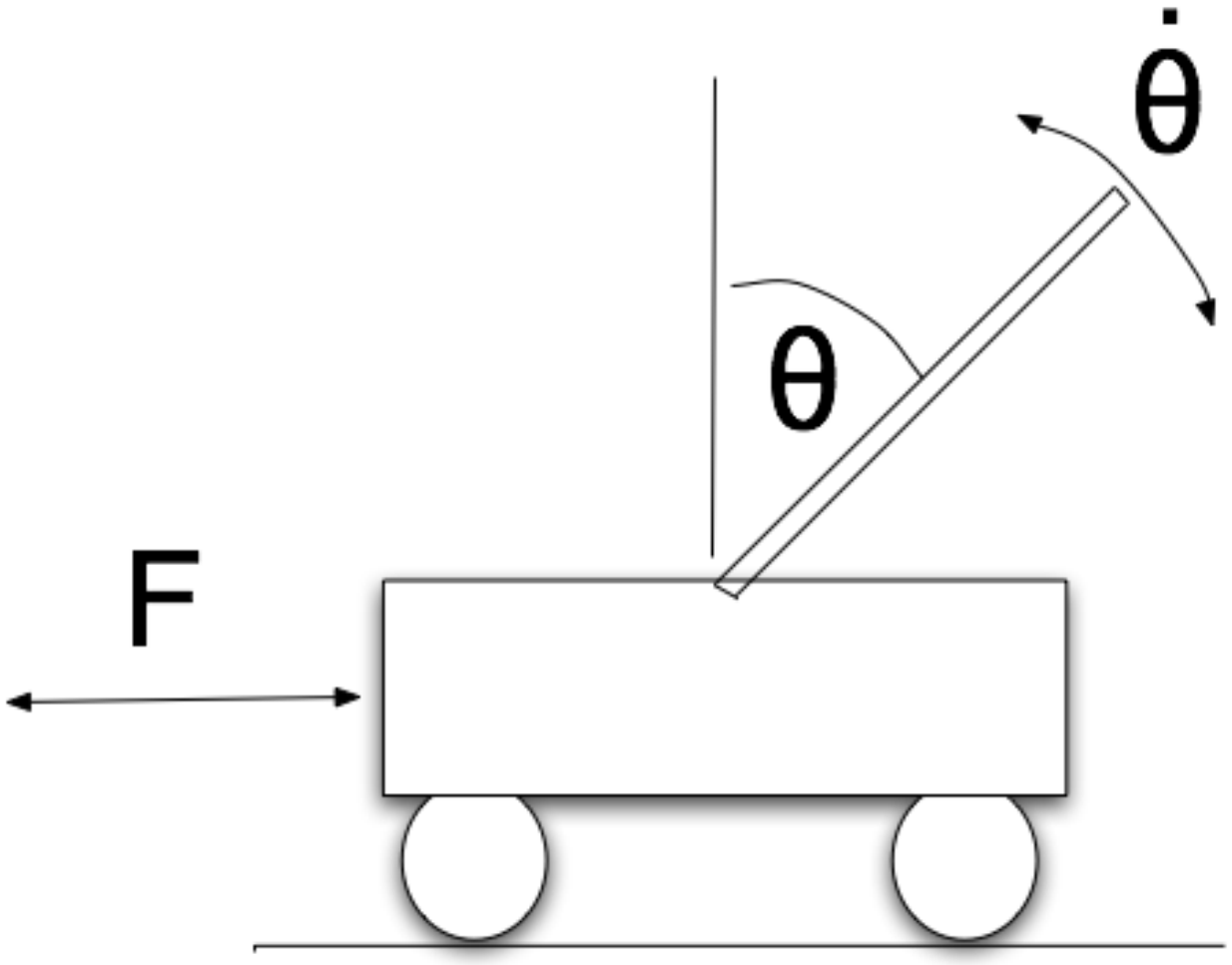
Gaussian process (GP) models: Flexible Bayesian machine learning approach. Allows for estimating functions from data. Also provides confidence intervals.

- Problem: Find global optimum when function evaluations are costly.
- (Osborne et al:) Use function evaluations to approximate unknown function $f(x)$ by a GP $y(x)$.
- Find new candidate point x_{n+1} for minimiser by minimising posterior expectation of *risk* $= \min\{y(x), f(x_n)\}$ with respect to x. This will take both mean and uncertainty of $y(x)$ into account.


Function Evaluation #1
Objective function
Observation
Mean
± 1SD
Expected loss
Chosen posn of next obs
Function Evaluation #3
Function Evaluation #5
Function Evaluation #7
y
x
0.05
0
−0.05
−0.1
−0.15
−0.2
−0.25
−15
−10
−5
0
5
10
15

Solving control problems

(Learning to balance a pole, by Marc Deisenroth et al)



'Standard' approach

Let $\mathbf{x}_k = (\theta, \dot{\theta})$

- Use the exact ODE to get $\mathbf{x}_{k+1} = f\left(\mathbf{x}_k, \text{Force}(\mathbf{x}_k)\right)$

- Use dynamic programming to find Force($\mathbf{x}_k$) which minimises expected costs.

  Bellman equation:

$$V_k^*(\mathbf{x}_k) = \min_u E\left[g(\mathbf{x}_k, \mathbf{u}_k) + V_{k+1}^*(f(\mathbf{x}_k, \mathbf{u}_k))\right]$$

- **Problem:** Needs exact knowledge of dynamics (ODEs). Exact solution of Bellman equation computationally hard. It requires (continuous state space).

Gaussian process approach

1. Create **example** time series with random force.

2. **Emulator:** Train a Gaussian process (GP) regression model on examples to learn the dynamics $\mathbf{x}_{k+1} = f\left(\mathbf{x}_k, \text{Force}(\mathbf{x}_k)\right)$ (with uncertainty).

3. Use GP to interpolate/extrapolate $Q(\mathbf{x}, \mathbf{u}) \doteq g(\mathbf{x}, \mathbf{u}) + V^*_{k+1}(f(\mathbf{x}, \mathbf{u}))$ from discrete set of $\mathbf{x}$ and $\mathbf{u}$. The uncertainty in the GP can be used to compute the expectation. in the Bellman equation.

Modeling and interpolation of the ambient magnetic field

A. Solin et al `arXiv:1509.04634v1`

- Goal: Use maps of magnetic fields in buildings for localisation.
- Magnetic fields are vectors $\mathbf{H}(\mathbf{x})$ which fulfil $\nabla \times \mathbf{H}(\mathbf{x})$. Thus we have the scalar field representation $\mathbf{H}(\mathbf{x}) = -\nabla\phi(\mathbf{x})$.
- Observation model

$$\begin{aligned} \phi(\mathbf{x}) &\sim \mathcal{GP}(0, K) \\ \mathbf{y}_i &= -\nabla\phi(\mathbf{x}_i) + \boldsymbol{\epsilon}_i \end{aligned}$$

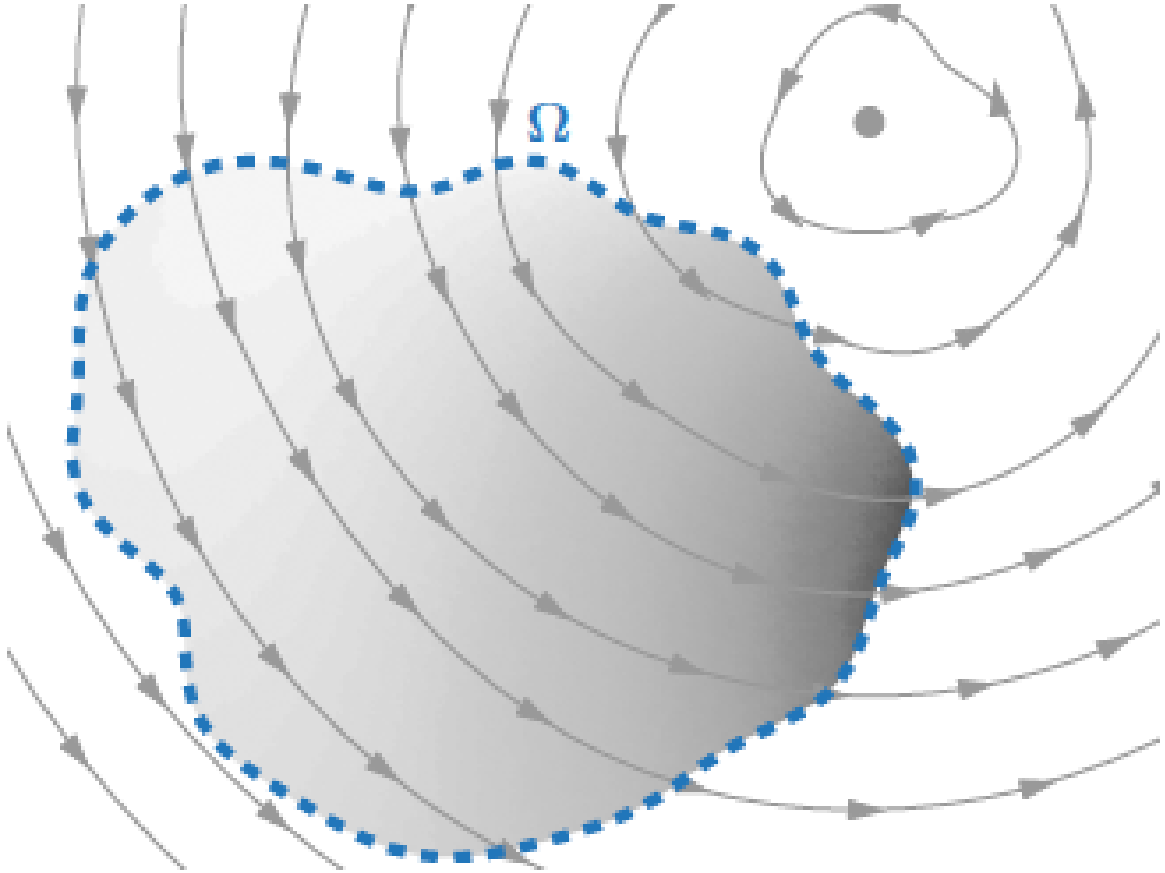
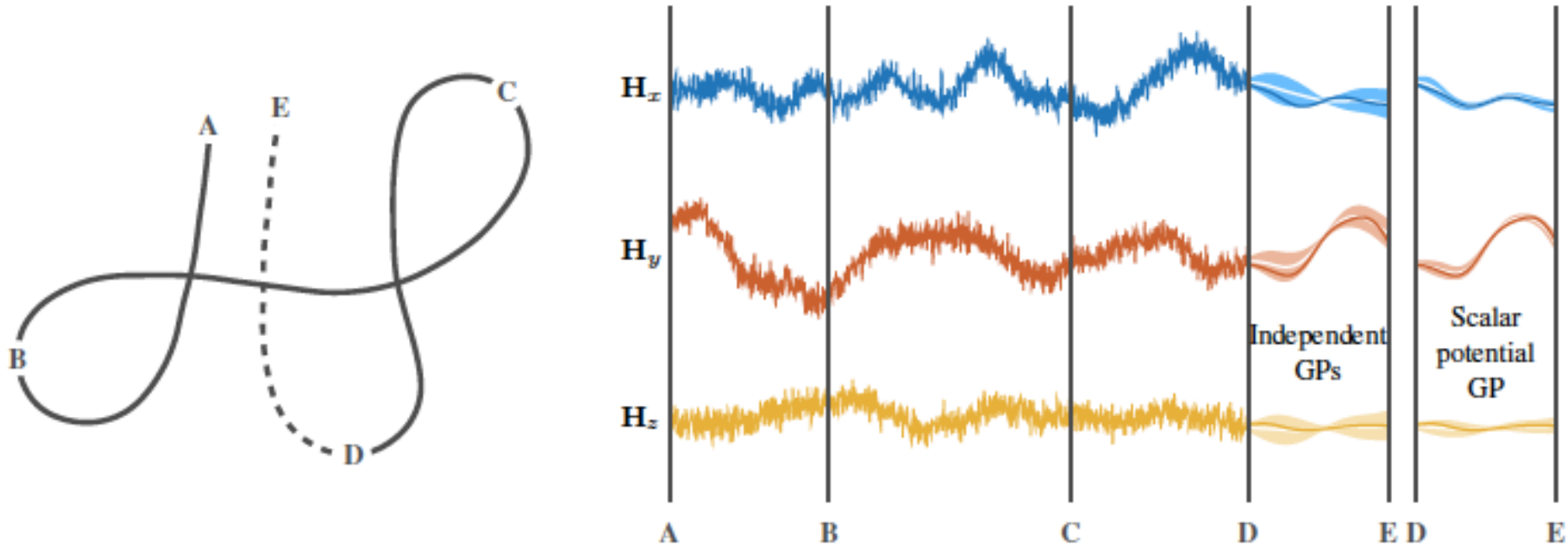


Figure 2. Illustration of a vector field with non-zero curl. The vortex point makes it non-curl-free as the vector field curls around it. However, the subset Ω excludes the vortex point and the vector field is curl-free in this region. To this region a scalar potential φ can be associated, here illustrated with shading.



(a) The route

(b) The data and the GP prediction for D–E

Figure 3. A simulated example of the interpolation problem. **(a)** Training data has been collected along the route A–D, but the magnetic field between D–E is unknown. **(b)** The noisy observations of the magnetic field between A–D, and GP predictions with 95% credibility intervals. Both the independent GP modeling approach (with shared hyperparameters) and the scalar potential based curl-free GP approach are visualized. The simulated ground truth is shown by the solid lines.

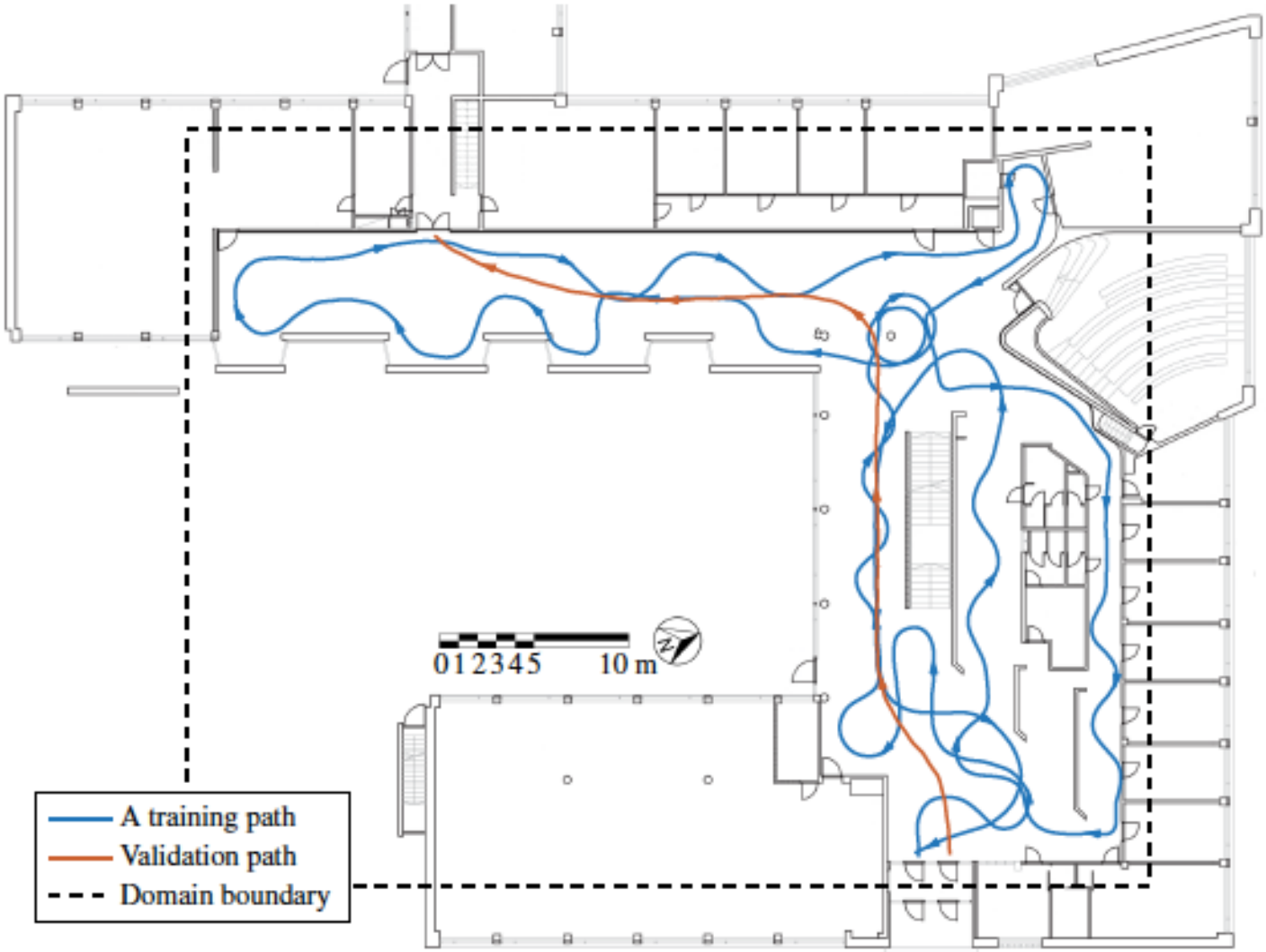
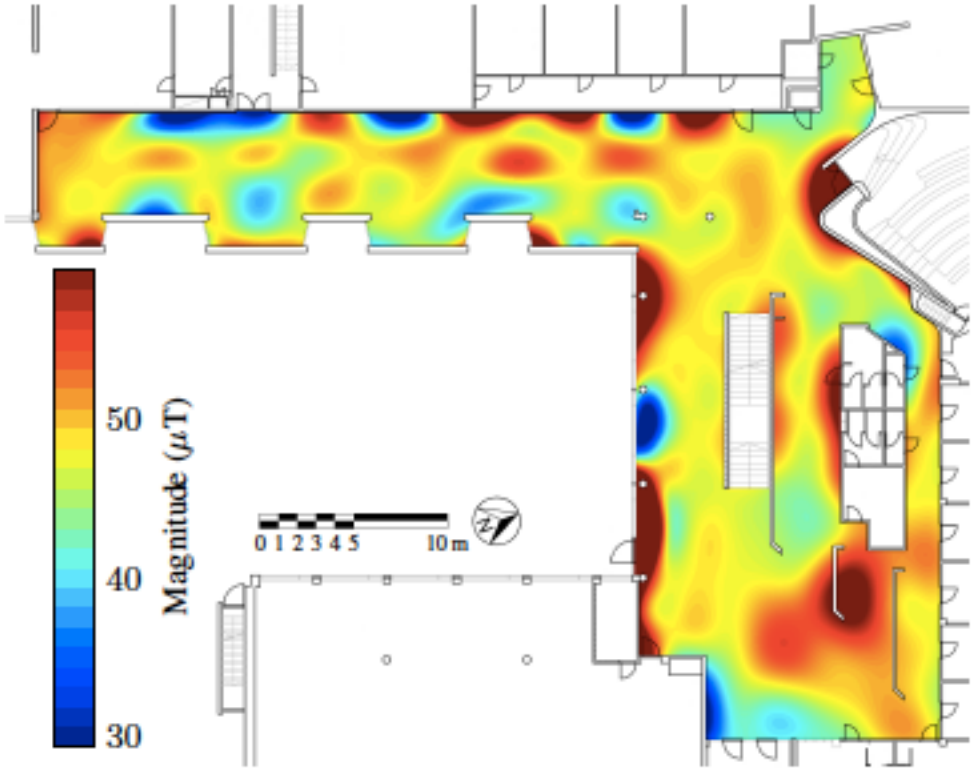
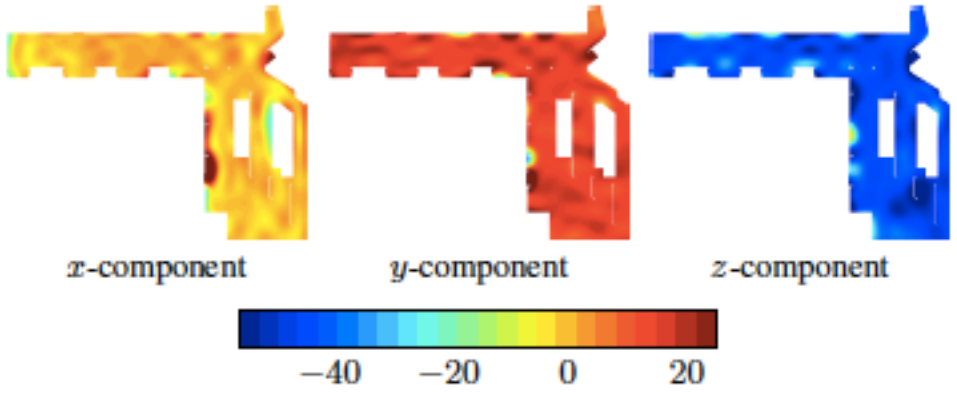


Figure 8. A training (red) and validation (blue) free-walking path that was used in the experiment. Trajectories were collected by a mobile phone, and the magnetometer data was corrected for gravitation direction and heading using the inertial sensors in the device. Walking direction markers are shown every 10 meters. The domain boundaries for the reduced-rank method are shown by the dashed line.



(a) Interpolated magnetic field strength



(b) Vector field components

Figure 9. **(a)** The magnetic field strength ($\|\mathbf{f}\|$) interpolated by the scalar potential GP model. **(b)** The separate field components of the estimate.

Computational tools III: Variational approximations

For a joint distribution $p(\mathbf{x}, \mathbf{y})$ of hidden variables $\mathbf{x}$ and observed data $\mathbf{y}$ the posterior

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

describes our knowledge about $\mathbf{x}$ when we observe $\mathbf{y}$.

- The computation of the marginal probability of the data $p(\mathbf{y}) = \int d\mathbf{x}\, p(\mathbf{x}, \mathbf{y})$ requires high dimensional sums or integrals and is often intractable.

- For the same reasons we often can't compute marginals $p_i(x_i|\mathbf{y})$, or expectations using these densities which are e.g. required in the EM algorithm.

The Variational Approximation

Approximate $p(\mathbf{x}|\mathbf{y})$ by $q(\mathbf{x}) \in \mathcal{F}$ where $\mathcal{F}$ tractable family of distributions such that the Kullback-Leibler divergence

$$KL(q,p) = \int d\mathbf{x} q(\mathbf{x}) \ln \frac{q(\mathbf{x})}{p(\mathbf{x}|\mathbf{y})} \geq 0$$

is minimized.

- From $p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{x},\mathbf{y})}{p(\mathbf{y})}$, we get an **upper bound** for any q

$$-\ln p(\mathbf{y}) \leq F(q) \doteq \int d\mathbf{x}\, q(\mathbf{x}) \ln q(\mathbf{x}) - E_q[\ln p(\mathbf{x},\mathbf{y})]$$

  with the **variational free energy** $F(q)$

- Dependency on parameters for optimal q:

$$\frac{dF(q^*(\theta),\theta)}{d\theta} = \frac{\partial F(q^*,\theta)}{\partial \theta}$$

The Mean Field Method

An important case is given by the family of factorising densities

$$q(\mathbf{x}) = \prod_{i=1}^{M} q_i(x_i)$$

In this case, we speak of a **mean field approximation**. Optimise q_i such that the free energy

$$F(q) = \int d\mathbf{x}\, q(\mathbf{x}) \ln q(\mathbf{x}) - E_q[\ln p(\mathbf{x}, \mathbf{y})]$$

is minimial. The solution is: $q_i^*(x) = \frac{1}{Z_i} \exp\left\{E_{\setminus i}[\ln p(\mathbf{x}, \mathbf{y})]\right\}$ with $E_{\setminus i}[\ldots]$ the average over all variables except x_i.

Proof: For any q_i, we have

$F(q) = -\int dx\, q_i(x) E_{\setminus i}[\ln p(\mathbf{x}, \mathbf{y})] + \sum_j \int dx\, q_j(x)\ \ln q_j(x)$

$= KL(q_i, q_i^*) - \ln Z_i^* + \sum_{j, j\neq i} \sum_x q_j(x)\ \ln q_j(x)$. Minimal for $q_i = q_i^*$. Requires *selfconsistent solution* (e.g. sequential update).

MF Example

$$p(\mathbf{x}) = \frac{1}{Z}\prod_i \psi_i(x_i)\ \exp\left[\frac{1}{2}\mathbf{x}^T\mathbf{J}\mathbf{x}\right]$$

with $J_{ii} = 0$. For this case, we have

$$q_i(x) = \frac{1}{Z_i}\psi_i(x)\exp\left[x\sum_j J_{ij}\langle x_j\rangle_{q_j}\right]$$

where the brackets $\langle\ldots\rangle_{q_j}$ denote expectation wrt to q_i. Introduce

$$Z_i(\gamma) = \int dx\ \psi_i(x)\exp\left[x\gamma\right]$$

$$m_i(\gamma) = \frac{d\ln Z_i}{d\gamma}$$

we get the relation (exact for Gaussian models)

$$E_q[x_i] = m_i\left(\sum_j J_{ij}E_q[x_j]\right)$$

Variational EM Algorithm

Optimise model parameters by Maximum Likelihood using free energy bound

$$-\ln p(\mathbf{y}|\boldsymbol{\theta}) \leq \int q(\mathbf{x}) \ln \frac{q(\mathbf{x})}{p(\mathbf{y}, \mathbf{x}|\boldsymbol{\theta})} \equiv F(q, \boldsymbol{\theta})$$

Iterate:

1. Mimimise $F(q, \boldsymbol{\theta}_t)$ with respect to the distribution $q \in \mathcal{F} \rightarrow q_t$. Note, that the unconstrained variation gives $q_t(\mathbf{x}) = p(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta})$ (exact EM algorithm)!

2. Minimise $F(q_t, \theta)$ with respect to $\boldsymbol{\theta}$.

   This iterations will not increase (and possibly decrease) **an upper bound** on $-\ln p(\mathbf{y}|\boldsymbol{\theta})$!

Variational Bayes algorithm

This aims at performing an approximation to a full Bayesian posterior i.e. $p(\mathbf{x}, \boldsymbol{\theta}|\mathbf{y})$. We use the bound

$$-\ln p(\mathbf{y}|m) \leq F(q) = \int d\mathbf{x}\, d\boldsymbol{\theta}\; q(\mathbf{x}, \boldsymbol{\theta}) \ln \frac{q(\mathbf{x}, \boldsymbol{\theta})}{p(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta}|m)}$$

Look for minima in the space of factorising distributions $q(\mathbf{x}, \boldsymbol{\theta}) = q(\mathbf{x})q(\boldsymbol{\theta})$.

Alternate between

1. **VB - E Step**: Minimise $F(q(\mathbf{x}), q_t(\boldsymbol{\theta}))$ w.r.t. $q(\mathbf{x})$

$$q_{l+1}(\mathbf{x}) \propto \exp\left[\int q_l(\boldsymbol{\theta})) \ln p(\mathbf{y}, \mathbf{x}|\boldsymbol{\theta}, m)\; d\boldsymbol{\theta}\right]$$

2. **VB - M Step**: Minimise $F(q_{l+1}(\mathbf{x}), q(\boldsymbol{\theta}))$ w.r.t. $q(\boldsymbol{\theta})$

$$q_{l+1}(\boldsymbol{\theta}) \propto p(\boldsymbol{\theta}) \exp\left[\int q_l(\mathbf{x}) \ln p(\mathbf{y}, \mathbf{x}|\boldsymbol{\theta}, m)\; d\mathbf{x}\right]$$

A model for collaborative filtering

(U Paquet, B Thomson, O Winther; *A hierarchical model for ordinal matrix factorization*, Statistics and Computing)

- r_{mn} = Rating of customer n on item (e.g. movie) m. We have $r_{mn} \in 1, \ldots, R$

- Introduce ideal latent variable f with $p(r|f) = 1$ if $b_r \leq f \leq b_{r+1}$, where $-\infty = b_1 < b_2 < \ldots < b_{R+1} = \infty$ and $p(r|f) = 0$, else.

- The latent variable f becomes noisy using $p(f|h) = \mathcal{N}(f; h, 1)$. This leads to

$$p(r_{mn}|h_{mn}) = \prod_r \Big[\Phi(h_{mn} - b_r) - \Phi(h_{mn} - b_{r+1})\Big]^{1_{r_{mn}=r}}$$

  and the total likelihood is

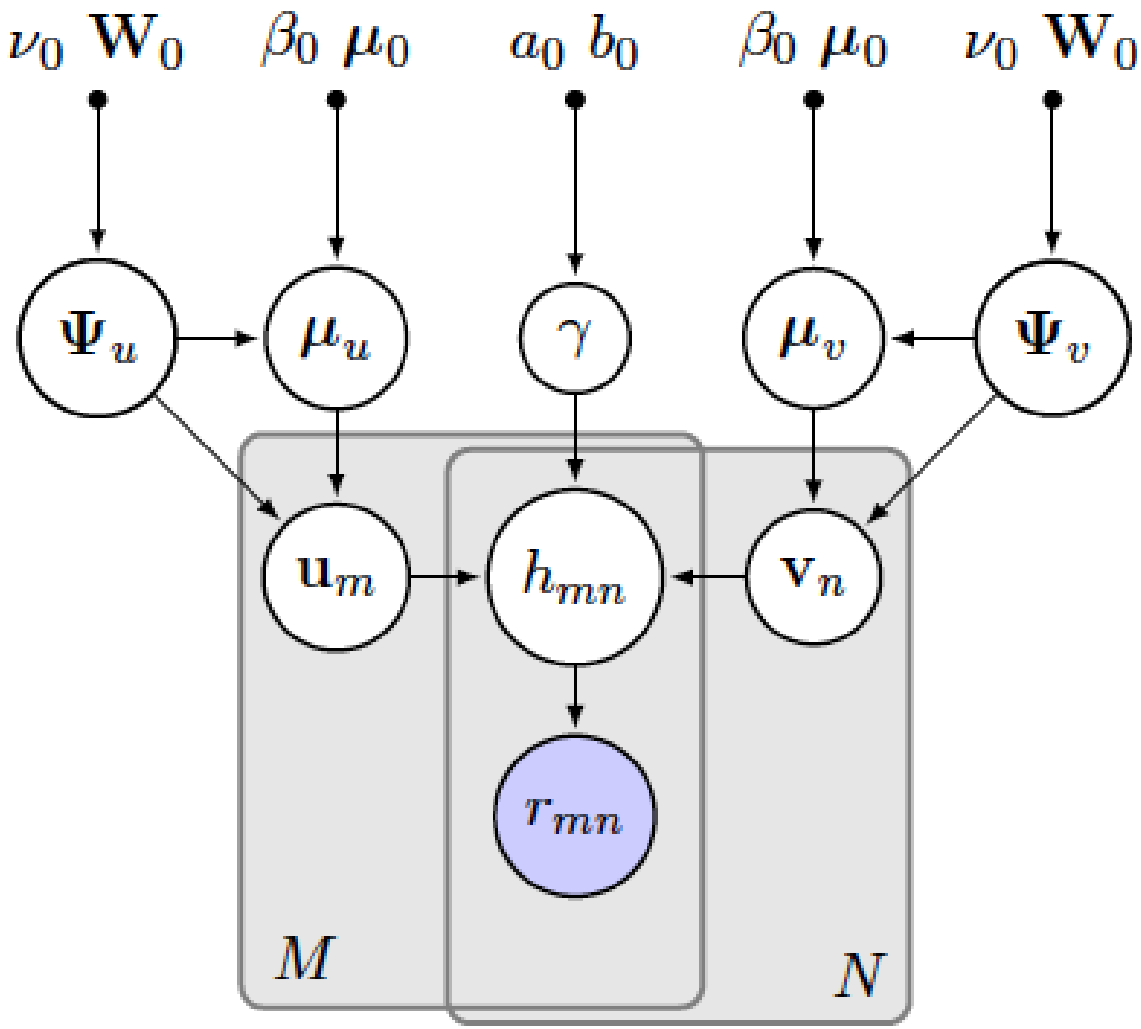
$$p(D|H) = \prod_{m,n} p(r_{mn}|h_{mn})$$

- **Low rank matrix factorization:** $h_{mn} = \mathbf{u}_m^\top \mathbf{v}_n + \epsilon_{mn}$ with $\epsilon_{mn} \sim \mathcal{N}(0, \gamma^{-1})$ i.i.d. Gaussian noise.

- $\mathbf{u}_m$ and $\mathbf{v}_n$ are factors of length K (small) corresponding to item m and customer n.

- Priors $p(\mathbf{u}_m|\boldsymbol{\mu}_u, \boldsymbol{\Psi}_u) = \mathcal{N}(\mathbf{u}_m; \boldsymbol{\mu}_u, \boldsymbol{\Psi}_u^{-1})$ and $p(\mathbf{v}_n|\boldsymbol{\mu}_v, \boldsymbol{\Psi}_v) = \mathcal{N}(\mathbf{v}_n; \boldsymbol{\mu}_v, \boldsymbol{\Psi}_v^{-1})$

- $p(\boldsymbol{\mu}_{u,v}, \boldsymbol{\Psi}_{u,v}) =$ Normal–Wishart priors. $p(\gamma)$ is a Gamma prior.


ν_0 $\mathbf{W}_0$
β_0 $\boldsymbol{\mu}_0$
a_0 b_0
β_0 $\boldsymbol{\mu}_0$
ν_0 $\mathbf{W}_0$
$\boldsymbol{\Psi}_u$
$\boldsymbol{\mu}_u$
γ
$\boldsymbol{\mu}_v$
$\boldsymbol{\Psi}_v$
$\mathbf{u}_m$
h_{mn}
$\mathbf{v}_n$
r_{mn}
M
N

- Hence, the model is of the form

$$P(\text{Data}|\boldsymbol{\theta})p(\boldsymbol{\theta}) = \prod_{mn} p(r_{mn}|h_{mn})p(h_{mn}|\mathbf{u}_m\mathbf{v}_n,\gamma)\prod_m p(\mathbf{u}_m|\boldsymbol{\mu}_u,\boldsymbol{\Psi}_u)\prod_n p(\mathbf{v}_n|\boldsymbol{\mu}_v,\boldsymbol{\Psi}_v)\times p(\boldsymbol{\mu}_u,\boldsymbol{\Psi}_u)p(\boldsymbol{\mu}_v,\boldsymbol{\Psi}_v)$$

- Variational approximation

$$q(\theta) = \prod_{mn} q(r_{mn})\prod_m q(\mathbf{u}_m)\prod_n q(\mathbf{v}_n)q(\boldsymbol{\mu}_u,\boldsymbol{\Psi}_u)q(\boldsymbol{\mu}_v,\boldsymbol{\Psi}_v)$$

- Example: $q(\mathbf{u}_m) = \mathcal{N}(\langle\mathbf{u}_m\rangle, \boldsymbol{\Sigma}_m)$

  with

$$\boldsymbol{\Sigma}_m = \left(\langle\boldsymbol{\Psi}_u\rangle + \langle\gamma\rangle \sum_{n\in\Omega(m)} \langle\mathbf{v}_n\mathbf{v}_n^\top\rangle\right)^{-1}$$

  and

$$\langle\mathbf{u}_m\rangle = \boldsymbol{\Sigma}_m\left(\langle\boldsymbol{\Psi}_u\boldsymbol{\mu}_u\rangle + \langle\gamma\rangle \sum_{n\in\Omega(m)} \langle h_{mn}\rangle\langle\mathbf{v}_n\rangle\right)$$

Dynamical Bayes Models with hidden factors

(M. J. Beal, F. Falciani, Z. Ghahramani, C. Rangel, D. L. Wild)

- Hidden causes or unmeasured genes may simplify network structure & lead to better interpretability.

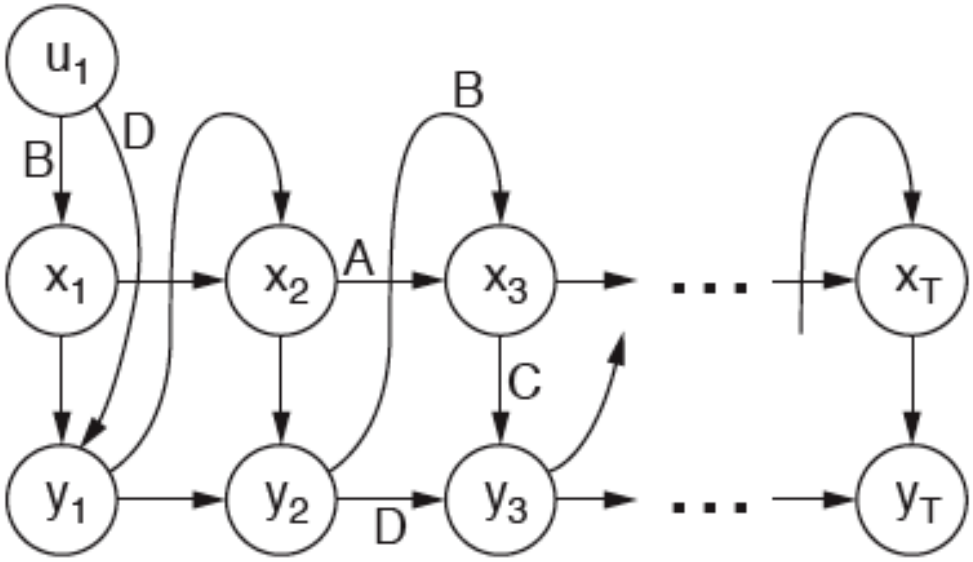


Fig. 1. The feedback graphical model with outputs feeding into inputs. Gene expression levels at time t are represented by y_t, whilst the hidden factors are represented by x_t.

- Gaussian state space models

$$\begin{aligned} \mathbf{x}_t &= \mathbf{A}\mathbf{x}_{t-1} + \mathbf{B}\mathbf{y}_{t-1} + \mathbf{w}_t \qquad \mathbf{w}_t \sim N(0, \mathbf{I}) \\ \mathbf{y}_t &= \mathbf{C}\mathbf{x}_t + \mathbf{D}\mathbf{y}_{t-1} + \mathbf{v}_t \qquad \mathbf{w}_t \sim N(0, \mathbf{R}) \end{aligned}$$

- Bayesian approach: Use (conjugate) Gaussian prior distributions over matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ and a Gamma prior over the elements of the diagonal matrix $\mathbf{R}$.

- Goal: Fit the model by maximising $p(\mathbf{y}|m)$ where m denotes the model, i.e. the dimensionality of the hidden states.

- Make predictions about *interactions* using the posterior distribution $p(\theta|\mathbf{y}, m)$ where $\theta = \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$.

  Problem: This is intractable! Approximate inference is necessary.

We have

$$p(\mathbf{y}, \mathbf{x}|\boldsymbol{\theta}, m) = \prod_t p(\mathbf{y}_t|\mathbf{x}_t, \mathbf{y}_{t-1}) \times \prod_t p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_{t-1})$$

with

$$p(\mathbf{y}_t|\mathbf{x}_t, \mathbf{y}_{t-1}) \propto \exp\left[-\frac{1}{2}(\mathbf{y}_t - \mathbf{C}\mathbf{x}_t - \mathbf{D}\mathbf{y}_{t-1})^\top \mathbf{R}^{-1}(\mathbf{y}_t - \mathbf{C}\mathbf{x}_t - \mathbf{D}\mathbf{y}_{t-1})\right]$$

and

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}_{t-1}) \propto \exp\left[-\frac{1}{2}(\mathbf{x}_t - \mathbf{A}\mathbf{x}_{t-1} - \mathbf{B}\mathbf{y}_{t-1})^\top (\mathbf{x}_t - \mathbf{A}\mathbf{x}_{t-1} - \mathbf{B}\mathbf{y}_{t-1})\right]$$

- Hidden variables possibly represent "combination of complex molecular events linking two genes"

- This leads to effective interactions (activation or inhibition) between measured genes is given by $I_{ij} = (\mathbf{CB} + \mathbf{D})_{ij}$.

- Significant evidence of interactions if $|I_{ij}|$ far away from 0 relative to standard deviation.

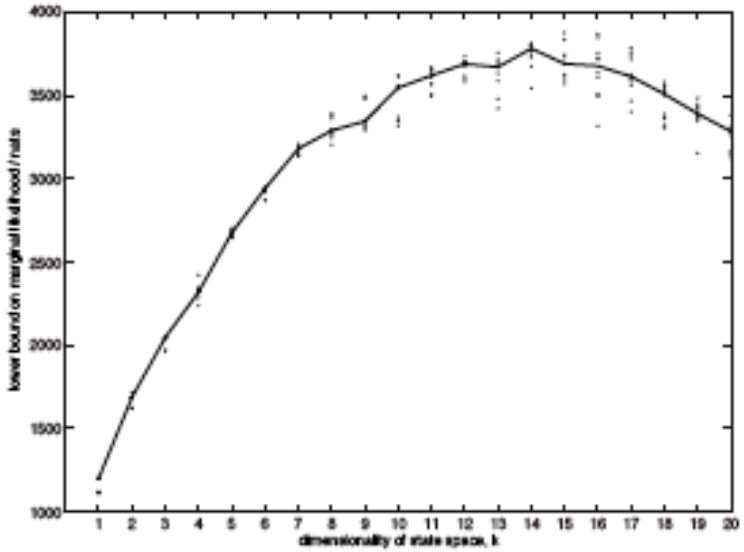


Fig. 2. Variation of $\mathcal{F}$ with hidden state dimension k for 10 random initializations of VBEM. The line represents the median $\mathcal{F}$ value.

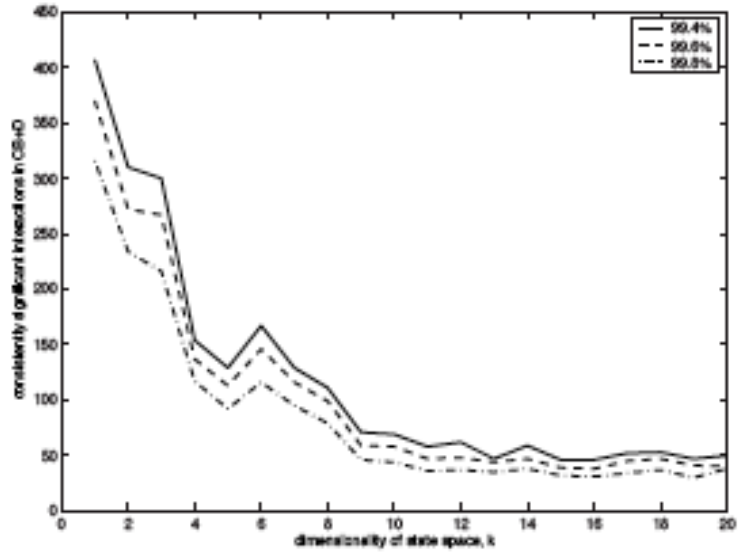


Fig. 3. The number of significant interactions that are repeated in all 10 runs of VB-EM at each value of k. There are 3 plots, each corresponding to a different significance level.

Gauss-Variational method (C. Archambeau & M. Opper)

Let $\mathbf{y}$ be observations and $\mathbf{x}$ latent parameters. Approximate posterior

$$p(\mathbf{x}|\mathbf{y},\boldsymbol{\theta}) = \frac{p(\mathbf{y},\mathbf{x},\boldsymbol{\theta})}{p(\mathbf{y},\boldsymbol{\theta})},$$

by a **tractable density** $q(\mathbf{x})$ minimising the **variational free energy**

$$F(q,\boldsymbol{\theta}) = -H[q] - E_q[\log p(\mathbf{x},\mathbf{y}|\boldsymbol{\theta})]$$

Gaussian variational densities

$$q(\mathbf{x}) = (2\pi)^{-N/2}|\boldsymbol{\Sigma}|^{-1/2}\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right).$$

The variational free energy becomes

$$F(q,\boldsymbol{\theta}) = -\frac{N}{2}\log 2\pi - \frac{1}{2}\log|\boldsymbol{\Sigma}| - \frac{N}{2} - E_q[\log p(\mathbf{x},\mathbf{y}|\boldsymbol{\theta})]$$

Setting $\nabla\mathcal{F}(q,\boldsymbol{\theta}) = 0$, we obtain

$$\begin{aligned} 0 &= \nabla_{\boldsymbol{\mu}} E_q[\log p(\mathbf{x},\mathbf{y}|\boldsymbol{\theta})] = E_q\left[\frac{\partial \log p(\mathbf{x},\mathbf{y}|\boldsymbol{\theta})}{\partial \mathbf{x}}\right] \\ \boldsymbol{\Sigma}^{-1} &= -2\nabla_{\boldsymbol{\Sigma}} E_q[\log p(\mathbf{x},\mathbf{y}|\boldsymbol{\theta})] = -E_q\left[\frac{\partial^2 \log p(\mathbf{x},\mathbf{y}|\boldsymbol{\theta})}{\partial \mathbf{x}^T \partial \mathbf{x}}\right] \end{aligned}$$

Useful Results for Gaussian expectations

To compute the minimum, we need

$$\frac{\partial \ln |\mathbf{\Sigma}|}{\partial \mathbf{\Sigma}} = -2 \frac{\partial \ln \int d\mathbf{x} \exp\left[-\frac{1}{2}\mathbf{x}^T \mathbf{\Sigma} \mathbf{x}\right]}{\partial \mathbf{\Sigma}} = \langle \mathbf{x}\mathbf{x}^T \rangle = \mathbf{\Sigma}^{-1}$$

Introducing the characteristic function

$$G(\mathbf{k}) = E_q\left[e^{i\mathbf{k}^T\mathbf{x}}\right] = \exp\left[-\frac{1}{2}\mathbf{k}^T\mathbf{\Sigma}\mathbf{k} + i\mathbf{k}^T\mathbf{m}\right]$$

of the measure q

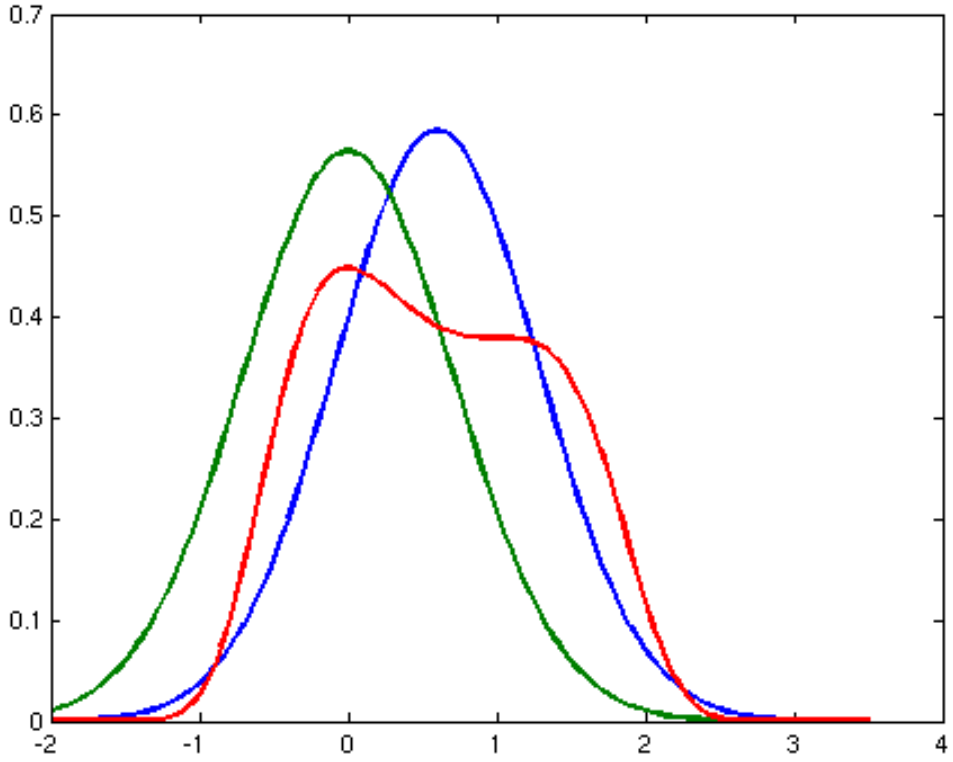
$$\int d\mathbf{x}\, q(\mathbf{x})\, F(\mathbf{x}) = \int d\mathbf{y}\; E_q\left[\delta(\mathbf{x}-\mathbf{y})\right] F(\mathbf{y}) = \frac{1}{(2\pi)^n} \int d\mathbf{y}\; d\mathbf{k}\; G(\mathbf{k}) e^{-i\mathbf{k}^T\mathbf{y}} F(\mathbf{y})$$
$$= \frac{1}{(2\pi)^n} \int d\mathbf{y}\; d\mathbf{k}\; \exp\left[-\frac{1}{2}\mathbf{k}^T\mathbf{\Sigma}\mathbf{k} + i\mathbf{k}^T(\mathbf{m}-\mathbf{y})\right] F(\mathbf{y})$$

Thus

$$\frac{\partial E_q[F(\mathbf{x})]}{\partial \mathbf{m}} = E_q\left[\frac{\partial F(\mathbf{x})}{\partial \mathbf{x}}\right]$$

and

$$\begin{aligned}\frac{\partial E_q[F(\mathbf{x})]}{\partial \boldsymbol{\Sigma}} &= -\frac{1}{2}\int d\mathbf{y}\; d\mathbf{k}\; \exp\left[-\frac{1}{2}\mathbf{k}^T\boldsymbol{\Sigma}\mathbf{k} + i\mathbf{k}^T(\mathbf{m}-\mathbf{y})\right]\mathbf{k}\mathbf{k}^T F(\mathbf{y}) \\ &= \frac{1}{2}E_q\left[\frac{\partial^2 F(\mathbf{x})}{\partial \mathbf{x}^T \partial \mathbf{x}}\right] = \frac{1}{2}\frac{\partial^2 E_q\left[F(\mathbf{x})\right]}{\partial \mathbf{m}^T \partial \mathbf{m}}\end{aligned}$$


0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-2
-1
0
1
2
3
4

GPs with factorising likelihood

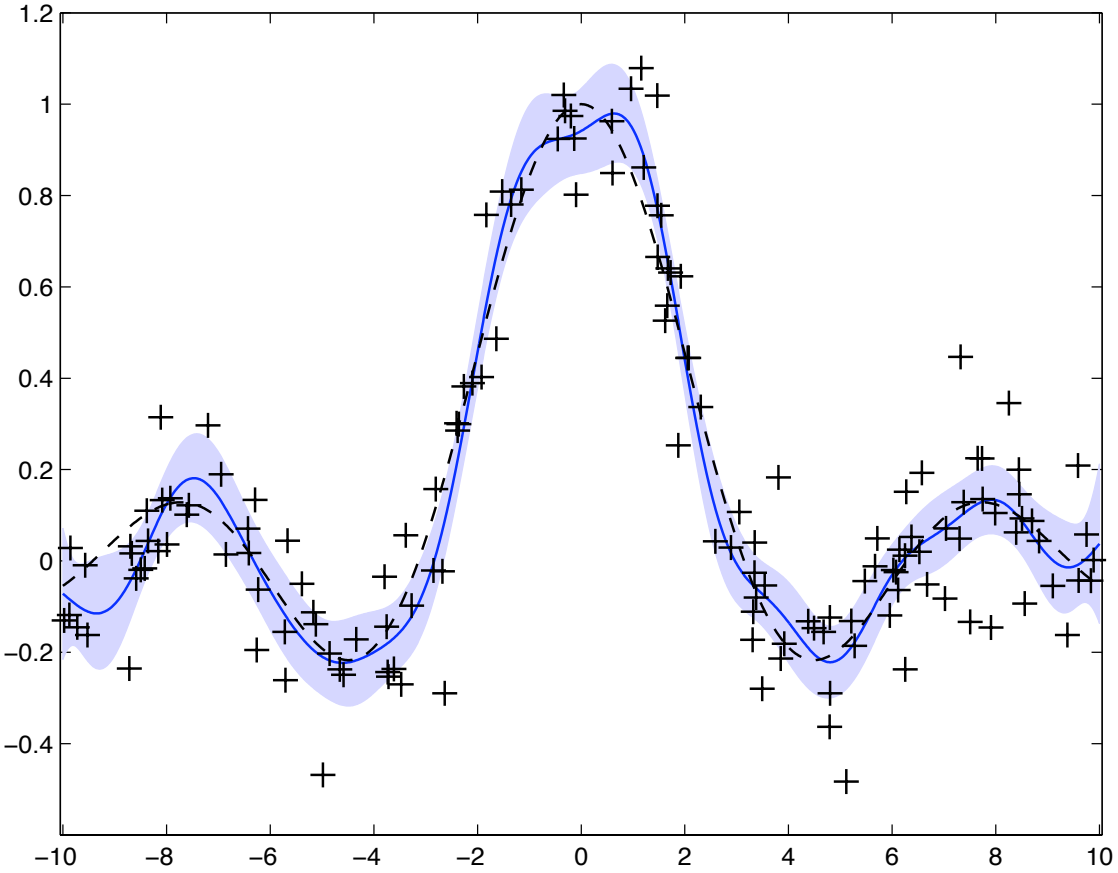
$$p(\mathbf{x}, \mathbf{y}) = \frac{1}{Z_0} \exp\left(-\sum_n V_n(y_n, x_n) - \frac{1}{2}\mathbf{x}^T \mathbf{K}^{-1} \mathbf{x} \right),$$

Covariance

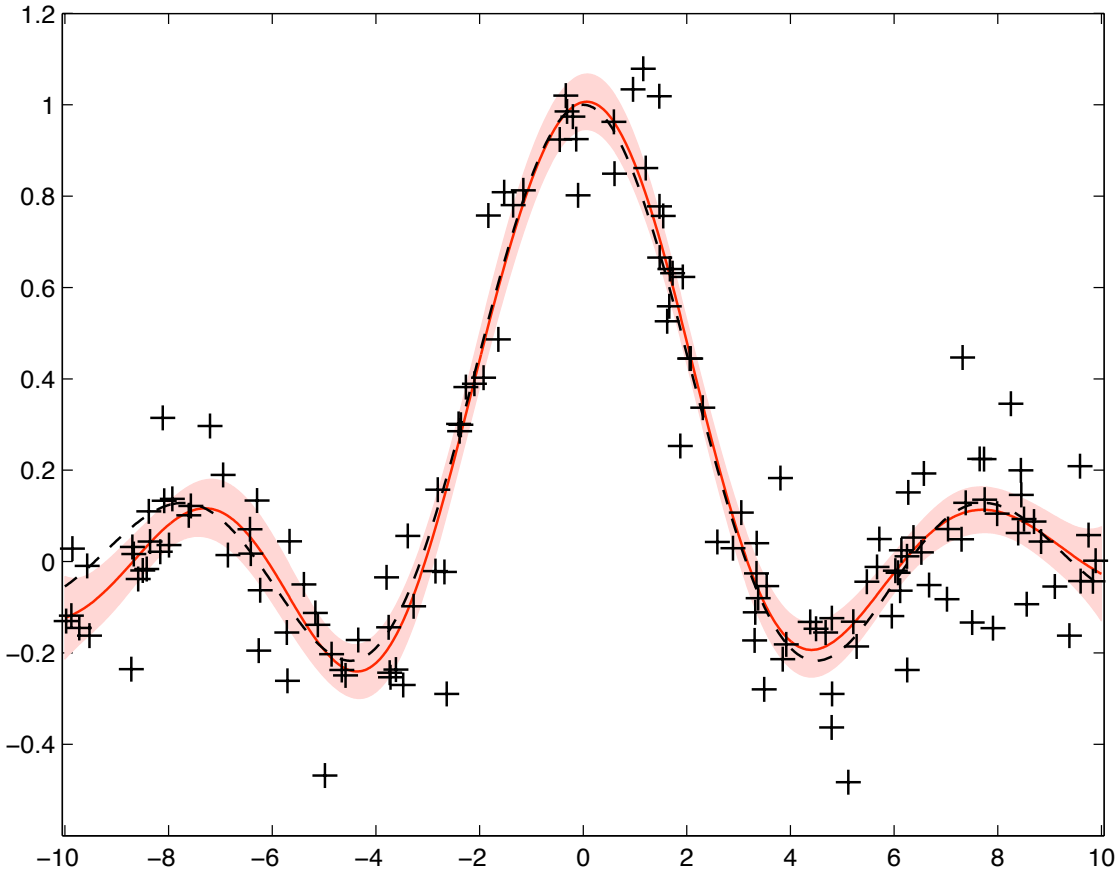
$$\boldsymbol{\Sigma}^{-1} = \mathbf{K}^{-1} + \mathrm{diag} E_q \left[\frac{\partial^2 V_n}{\partial x_n^2} \right]$$

is parametrised by N elements!

sinc function with Cauchy noise (GP with Gaussian likelihood)



sinc function with Cauchy noise (Var - GP with Cauchy likelihood)



Minimising the other KL

If we could, we would rather minimize the other KL

$$KL(p,q) = \int d\mathbf{x}\; p(\mathbf{x}) \ln \frac{p(\mathbf{x})}{q(\mathbf{x})} = \text{const} - \int d\mathbf{x}\; p(\mathbf{x}) \ln q(\mathbf{x})$$

If $q(\mathbf{x}) = \prod_i q_i(x_i)$, we have to minimize

$$-\sum_i \int dx\; p_i(x) \ln q_i(x)$$

which is minimized by the true marginal $q_i = p_i$.

On the other hand for exponential families

$$q(x|\boldsymbol{\theta}) = f(x)\; \exp[\boldsymbol{\psi}(\boldsymbol{\theta}) \cdot \boldsymbol{\phi}(x) + g(\boldsymbol{\theta})]\;.$$

we see that the optimal $\boldsymbol{\psi}$ is such that general moments match $\langle \boldsymbol{\phi}(\mathbf{x}) \rangle_q = \langle \boldsymbol{\phi}(\mathbf{x}) \rangle_p$.

We next try to do this procedure approximately in an on–line algorithm.

Bayes Online (Assumed Density Filtering)

Exact update of the posterior, when new data y_{t+1} arrives

$$p(\mathbf{x}|D_{t+1}) = \frac{p(y_{t+1}|\mathbf{x})p(\mathbf{x}|D_t)}{\int d\mathbf{x} p(y_{t+1}|\mathbf{x})p(\mathbf{x}|D_t)}.$$

Replace $p(\mathbf{x}|D_t)$ by parametric approximation $q(\mathbf{x}|par(t))$ using the following steps:

- Update:

$$q(\mathbf{x}|y_{t+1}, par(t)) = \frac{p(y_{t+1}|\mathbf{x})p(\mathbf{x}|par(t))}{\int d\mathbf{x} p(y_{t+1}|\mathbf{x})q(\mathbf{x}|par(t))}.$$

- Project: Minimize

$$KL\left(q(\cdot|y_{t+1}, par(t))||q(\cdot|par)\right)$$

  For exponential families $p(\mathbf{x}|par) \propto \exp[\boldsymbol{\psi} \cdot \boldsymbol{\phi}(\mathbf{x})]$, we have $par = \boldsymbol{\psi}$. The projection leads to moment matching $\langle \boldsymbol{\phi}(\mathbf{x}) \rangle$ for the distributions $q(\mathbf{x}|par)$ and $q(\mathbf{x}|y_{t+1}, par(t))$.

Gaussian Approximation

for $q(\mathbf{x}|par)$. Set $par = (mean, covariance) = (\hat{\mathbf{x}}, \mathbf{C})$. Matching of moments results in the explicit update:

$$\begin{aligned}
\hat{\mathbf{x}}(t+1) &= \hat{\mathbf{x}}(t) + \sum_j C_{ij}(t) \times \\
&\quad \times \partial_j \ln E_u[p(y_{t+1}|\hat{\mathbf{x}}(t)+u)] \\
&\text{and} \\
C_{ij}(t+1) &= C_{ij}(t) + \sum_{kl} C_{ik}(t) C_{lj}(t) \times \\
&\quad \times \partial_k \partial_l \ln E_u[p(y_{t+1}|\hat{\mathbf{x}}(t)+u)].
\end{aligned}$$

with $\partial_j \doteq \frac{\partial}{\partial \hat{x}_j}$.
$\int d\mathbf{x} p(y_{t+1}|\mathbf{x}) q(\mathbf{x}|par(t))$ was written as $E_u[p(y_{t+1}|\hat{\mathbf{x}}(t)+u)]$ where u is a zero mean Gaussian random vector with covariance $\mathbf{C}(t)$.

Asymptotic Error

Assume data are generated from **true density** $p^*(y)$

$$E_D[\epsilon_i(t)\epsilon_j(t)] = \frac{1}{t}\ (\mathbf{A}^{-1}\mathbf{B}\mathbf{A}^{-1})_{ij}, \qquad t \to \infty.$$

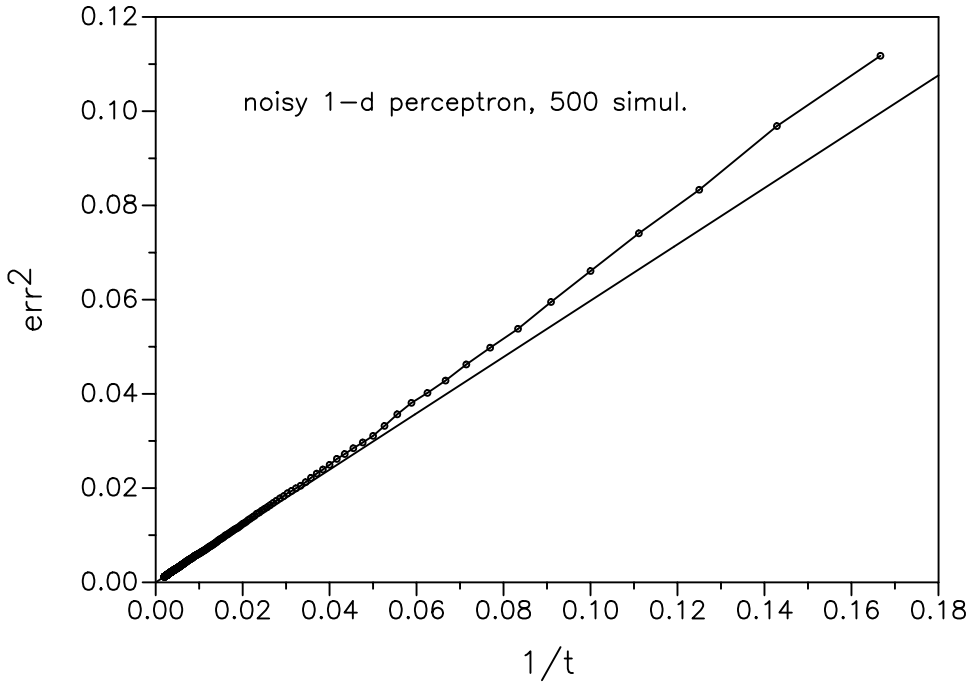
with

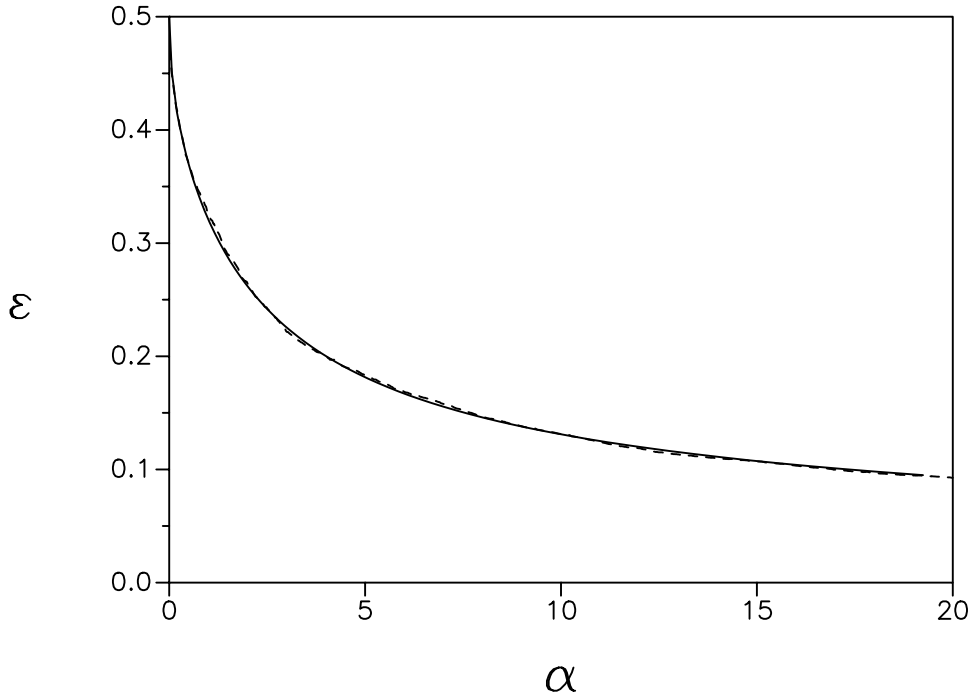
$$\begin{aligned} B_{ij} &= \int dy\ p^*(y)\partial_i \ln p(y|\mathbf{x}^*)\partial_j \ln p(y|\mathbf{x}^*) \\ A_{ij} &= -\int dy\ p^*(y)\partial_i\partial_j \ln p(y|\mathbf{x}^*). \end{aligned}$$

The same rate as for batch algorithms (Max. Likelihood or Bayes) : Asymptotic Efficiency!

Toy applications: Perceptrons

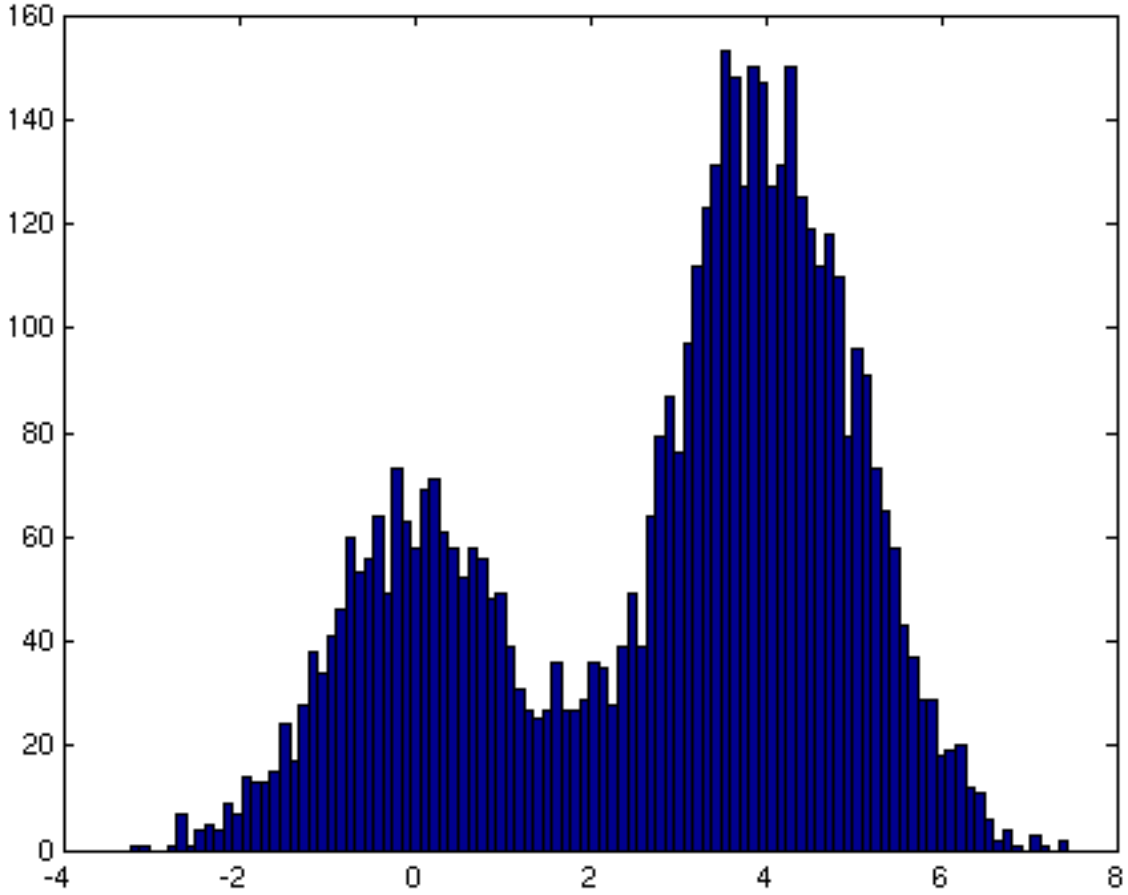
Results for $d = 1$ and $d = 50$ (probit model, spherical Gaussian inputs, realizable target, $\alpha \doteq \frac{\#\text{data}}{d}$). Right–dashed line: Bayes optimal (batch).





Method Kernelizes (Csato & Opper) also basis of *Informative Vector Machine* (Lawrence et al).

Mixture models



$$p(x) = \sum_{l=1}^{K} \pi_l p(x|\theta_l)$$

Bayesian generative model for mixtures

- Generate K parameters $\theta_1, \ldots, \theta_K$ from prior $h(\theta)$.
- Generate random weights $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)$ from a prior (Dirichlet–) distribution.
- For each observation i generate indicator variable z_i with $P(z_i = k) = \pi_k$.
- For each i generate observations x_i from $p(x_i|\theta_{z_i})$

How large should K be ? **Can we deal with $K \to \infty$?**

Bayesian Mixture models with non–fixed number of components

Generative model

- Draw random (discrete) probability measure $G \sim \mathrm{DP}(\alpha, H)$ over parameter space Θ from Dirichlet process.

- Draw n (= number of observations) parameters from G. Some θ's will be the same.

- For each $i = 1, \ldots, n$ draw data points $x_i \sim p(\cdot|\theta_i)$.

Dirichlet distribution

- **Multinomial model** model: Let $z_i \in \{1, \ldots, K\}$ and $\pi_j = \mathsf{Pr}(z_i = j)$ $n_j =$ number of items j in set of data $(z_1, \ldots, z_n)$. With $\mathbf{n} = (n_1, \ldots, n_K)$, with $n_j \in N$ and $\sum_j n_j = n$, we have

$$\mathsf{Pr}(\mathbf{n}|\boldsymbol{\pi}) = \frac{n!}{\prod_{j=1}^{K} n_j!} \prod_{j=1}^{K} \pi_j^{n_j}$$

- Conjugate prior is the prior probability distributions over discrete probabilities π_j (Dirichlet distribution) with hyperparameter $\alpha = (\alpha_1, \ldots, \alpha_K)$

$$p(\pi|\alpha) = \frac{1}{B(\alpha)} \prod_{j=1}^{K} \pi_j^{\alpha_j - 1}$$

  with the constraint $\sum_j \pi_j = 1$!

- Posterior probabilities obtained by $\alpha_j \to \alpha_j + n_j$

- Moments (set $a = \sum_j \alpha_j$)

$$E[\pi_i|\alpha] = \frac{\alpha_i}{\sum_j \alpha_j}$$

$$\text{VAR}(\pi_i|\alpha) = \frac{\alpha_i(a - \alpha_i)}{a^2(a+1)}$$

Dirichlet processes

- Consider some set Θ, $A \subseteq \Theta$. The Dirichlet process generates random probability measures $G(\cdot)$ where $G(A) = \Pr(\theta \in A|G)$.

- A parameter of the process is the **base measure** H: $H(A) = \Pr(\theta \in A|H)$. If we have a density, $H(A) = \int_A h(\theta)d\theta$.

- Define Dirichlet process $\mathrm{DP}(\alpha, H)$ through finite dimensional marginals: For any **finite partition** $A_1, \ldots, A_r$ of the Θ space, the r dimensional vector of the probabilities

$$\pi_i \doteq G(A_i)$$

is distributed as

$$(\pi_1, \ldots, \pi_r) \sim \mathrm{Dir}(\alpha_1, \ldots, \alpha_r)$$

$$p(\pi_1, \ldots, \pi_r|\alpha_1, \ldots, \alpha_r) \propto \prod_{j=1}^{r} \pi_j^{\alpha_j - 1} \delta(\sum_{j=1}^{r} \pi_j - 1)$$

where $\alpha_i = \alpha H(A_i)$. α is called **concentration parameter**.

- Since

$$E[\pi_i|\alpha] = \frac{\alpha_i}{\sum_j \alpha_j}$$

$$\mathsf{VAR}(\pi_i|\alpha) = \frac{\alpha_i(a-\alpha_i)}{a^2(a+1)}$$

we get

$$\begin{aligned} E[G(A)] &= H(A) \\ \mathsf{Var}[G(A)] &= \frac{H(A)(1-H(A))}{\alpha+1} \end{aligned}$$

Generating data samples from a DP

- Generate G from a DP. Generate 'data' θ_i i.i.d. from G

$$G \sim \mathsf{DP}(\alpha, H)$$
$$\theta_1, \ldots, \theta_n \sim G$$

- We are now interested in drawing samples $\theta_1, \ldots, \theta_n$. If we are not interested in G, we can marginalize G out.

- Try sequential approach: Suppose we have a sample $\theta_1, \ldots, \theta_n$. We want to find the probability of a new θ_{n+1} given the previous ones. To get this, consider first the DP process conditioned on $\theta_1, \ldots, \theta_n$.

- Set n_k = Number of points $\theta_j \in A_k$. Construct conditional distribution of $(\pi_1, \ldots, \pi_r) \doteq (G(A_1), \ldots, G(A_r))$ by

$$p(\pi_1, \ldots, \pi_r | \theta_1, \ldots, \theta_n) =$$

$$\text{const} \times P(n_1, \ldots, m_r | \pi_1, \ldots, \pi_r) \prod_{j=1}^{r} \pi_j^{\alpha_j - 1} \delta(\sum_{j=1}^{r} \pi_j - 1) =$$

$$\frac{n!}{\prod_{j=1}^{r} n_j!} \prod_{j=1}^{K} \pi_j^{n_j} \prod_{j=1}^{r} \pi_j^{\alpha_j - 1} \delta(\sum_{j=1}^{r} \pi_j - 1) =$$

$$\propto \prod_{j=1}^{r} \pi_j^{\alpha_j + n_j - 1} \delta(\sum_{j=1}^{r} \pi_j - 1)$$

- Since all marginals are Dirichlet, we conclude that the posterior **process** itself is a DP

$$G | \theta_1, \ldots, \theta_n \sim \text{DP}\,(\beta_n, H_n)$$

where $\beta_n = \alpha + n$ and H_n is a measure with density

$$h_n(\theta) = \frac{\alpha}{\alpha + n} h(\theta) + \frac{n}{\alpha + n} \frac{\sum_k \delta(\theta - \theta_k)}{n}$$

Chinese restaurant process

- Next, integrate out G

$$\Pr(\theta_{n+1} \in A | \theta_1, \ldots, \theta_n) = E\left[G(A) | \theta_1, \ldots, \theta_n\right] = H_n(A)$$

- This can be written as

$$p(\theta_{n+1} | \theta_1, \ldots, \theta_n) = \frac{1}{\alpha + n}\left(\alpha h(\theta) + \sum_{j=1}^{m} n_j \delta(\theta - \theta_j^*)\right)$$

where θ_j^* denote the distinct items among the θ_k and n_j = number of times θ_j^* appears in the sample.

- Hence, with probability $\frac{\alpha}{\alpha+n}$ draw $\theta_{n+1} \sim h$ from base measure and get a new value.

- With probability $\frac{n_j}{\alpha+n}$ choose the 'old' value $\theta_{n+1} = \theta_j$.

- Expected number of distinct values θ (clusters)

$$E[m|n] = \sum_{i=1}^{n} \frac{\alpha}{\alpha + i - 1} \simeq \alpha \ln (1 + n/\alpha)$$

Dirichlet distribution

- Reminder: Random coins $x_i \in \{, 0, 1\}$ with

$$P(x_1, \ldots, x_n) = \theta^{n_1}(1-\theta)^{n-n_1}$$

.

- Conjugate prior $p(\theta) \propto \theta^{\alpha-1}(1-\theta)^{\beta-1}$ (beta density).

- General: Probability distributions over discrete probabilities π_j which give e.g. the weights needed for mixture models

$$p(\boldsymbol{\pi}|\boldsymbol{\alpha}) = \frac{1}{B(\boldsymbol{\alpha})} \prod_{j=1}^{K} \pi_j^{\alpha_j - 1}$$

with the constraint $\sum_j \pi_j = 1$!

- Normalisation

$$B(\boldsymbol{\alpha}) = \frac{\prod_j \Gamma(\alpha_j)}{\Gamma\left(\sum_j \alpha_j\right)}$$

- Marginals $p(\pi_i|\boldsymbol{\alpha}) \propto \pi_i^{\alpha_i - 1} \, (1 - \pi_i)^{\sum_{j=1}^{K} \alpha_j - \alpha_i - 1}$

- Moments (set $a = \sum_j \alpha_j$)

$$E[\pi|\boldsymbol{\alpha}] = \frac{\alpha_i}{\sum_j \alpha_j}$$

$$\text{VAR}(\pi|\boldsymbol{\alpha}) = \frac{\alpha_i(a - \alpha_i)}{a^2(a + 1)}$$

- Sampling

$$\begin{aligned} y_j &\sim p(\cdot|\alpha_j) \propto y^{\alpha_j - 1} e^{-y_j} \; i.i.d. \qquad \text{Gamma density} \\ \pi_j &= \frac{y_j}{\sum_j y_j} \end{aligned}$$

Example: Latent Dirichlet Allocation (Blei, Ng & Jordan)

LDA models each document as a mixture over latent topics. A topic is a distribution over words.

Generative Model:

- Prior assumption 1: Sample topic probabilities $\boldsymbol{\theta}$ from $Dir(\alpha)$:

$$p(\boldsymbol{\theta}|\alpha) = \prod_{j=1}^{D} \left\{ \frac{\Gamma(K\alpha)}{\Gamma^K(\alpha)} \prod_{k=1}^{K} \theta_{jk}^{\alpha-1} \right\}$$

- Prior assumption 2: Sample word probabilities coefficients $\boldsymbol{\phi}$ from $Dir(\beta)$:

$$p(\boldsymbol{\phi}|\alpha) = \prod_{k=1}^{K} \left\{ \frac{\Gamma(W\beta)}{\Gamma^W(\beta)} \prod_{w=1}^{W} \phi_{kw}^{\beta-1} \right\}$$

- For each word i position in document j choose topic with $P[z_{ij} = k] = \theta_{jk}$

- For each word position i in document j and topic k choose word with $P[x_{ij} = w] = \phi_{kw}$.

- Estimated $E[\theta_{jk}]$ can be used as features for classification of documents.

β and α are hyperparameters that can be optimised (learned) from a training corpus of data.

"Arts"	"Budgets"	"Children"	"Education"
NEW	MILLION	CHILDREN	SCHOOL
FILM	TAX	WOMEN	STUDENTS
SHOW	PROGRAM	PEOPLE	SCHOOLS
MUSIC	BUDGET	CHILD	EDUCATION
MOVIE	BILLION	YEARS	TEACHERS
PLAY	FEDERAL	FAMILIES	HIGH
MUSICAL	YEAR	WORK	PUBLIC
BEST	SPENDING	PARENTS	TEACHER
ACTOR	NEW	SAYS	BENNETT
FIRST	STATE	FAMILY	MANIGAT
YORK	PLAN	WELFARE	NAMPHY
OPERA	MONEY	MEN	STATE
THEATER	PROGRAMS	PERCENT	PRESIDENT
ACTRESS	GOVERNMENT	CARE	ELEMENTARY
LOVE	CONGRESS	LIFE	HAITI

The William Randolph Hearst Foundation will give $1.25 million to Lincoln Center, Metropolitan Opera Co., New York Philharmonic and Juilliard School. "Our board felt that we had a real opportunity to make a mark on the future of the performing arts with these grants an act every bit as important as our traditional areas of support in health, medical research, education and the social services," Hearst Foundation President Randolph A. Hearst said Monday in announcing the grants. Lincoln Center's share will be $200,000 for its new building, which will house young artists and provide new public facilities. The Metropolitan Opera Co. and New York Philharmonic will receive $400,000 each. The Juilliard School, where music and the performing arts are taught, will get $250,000. The Hearst Foundation, a leading supporter of the Lincoln Center Consolidated Corporate Fund, will make its usual annual $100,000 donation, too.

Variational Approximation

Teh, Newman & Welling (NIPS06) show:

- Simple factorisation:

$$q(\mathbf{z}, \boldsymbol{\theta}, \boldsymbol{\phi}) = \prod_{ij} q_{ij}(z_{ij}) \prod_{j} q_j(\boldsymbol{\theta}_j) \prod_{k} q_k(\boldsymbol{\phi}_k)$$

  does not work so well!

- Integrate out $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ to get

$$p(\mathbf{z}, \mathbf{x}) \propto \prod_{j=1}^{D} \left\{ \frac{\prod_k \Gamma(\alpha + \sum_w n_{jkw})}{\Gamma(K\alpha + \sum_{kw} n_{jkw})} \right\} \prod_{k=1}^{K} \left\{ \frac{\prod_w \Gamma(\beta + \sum_j n_{jkw})}{\Gamma(W\beta + \sum_{wj} n_{jkw})} \right\}$$

- Approximate $p(\mathbf{z}|\mathbf{x})$ by

$$q(\mathbf{z}) = \prod_{ij} q_{ij}(z_{ij})$$

  yields better approximation.

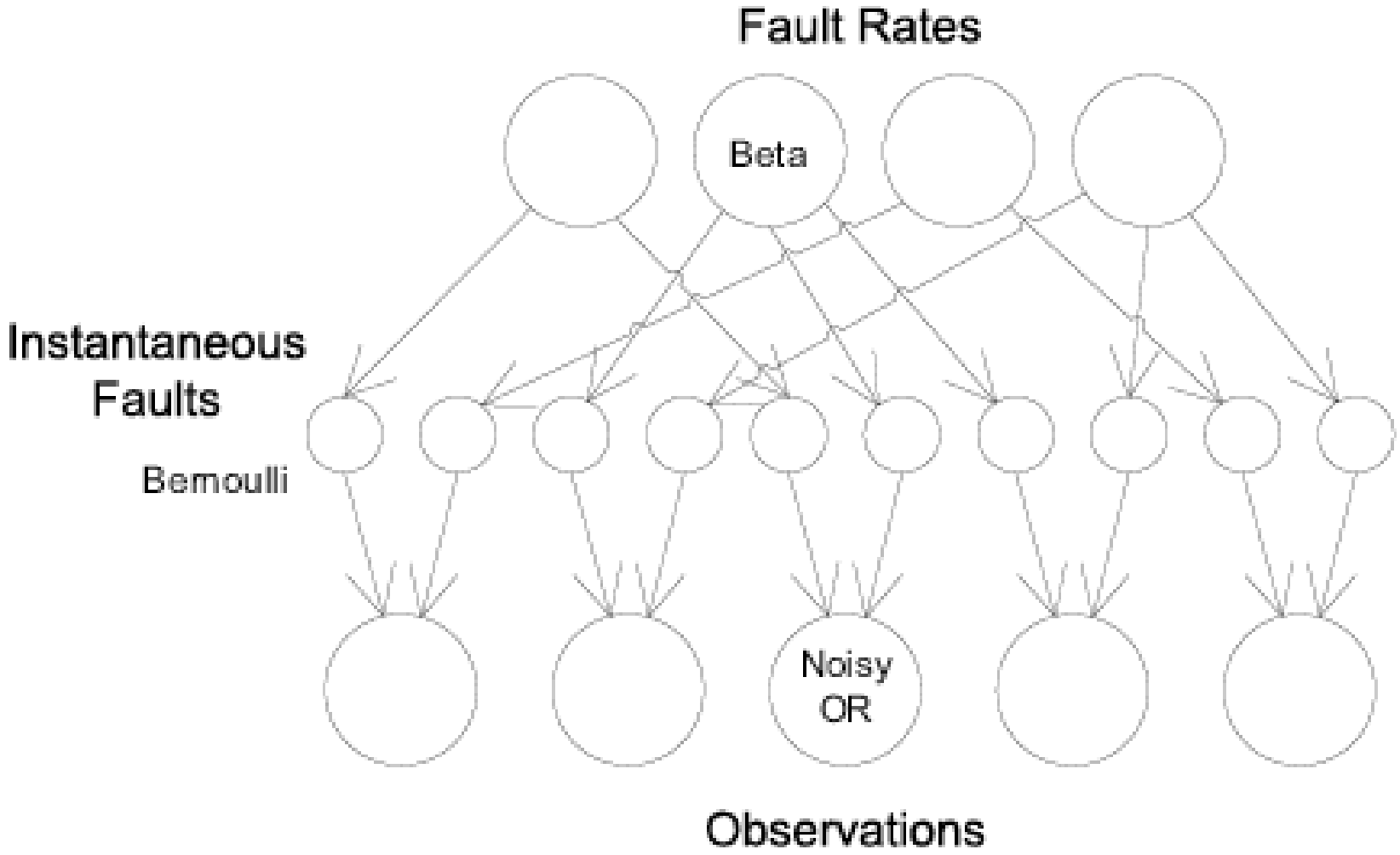


Figure 1: The full graphical model for the diagnosis of Internet faults

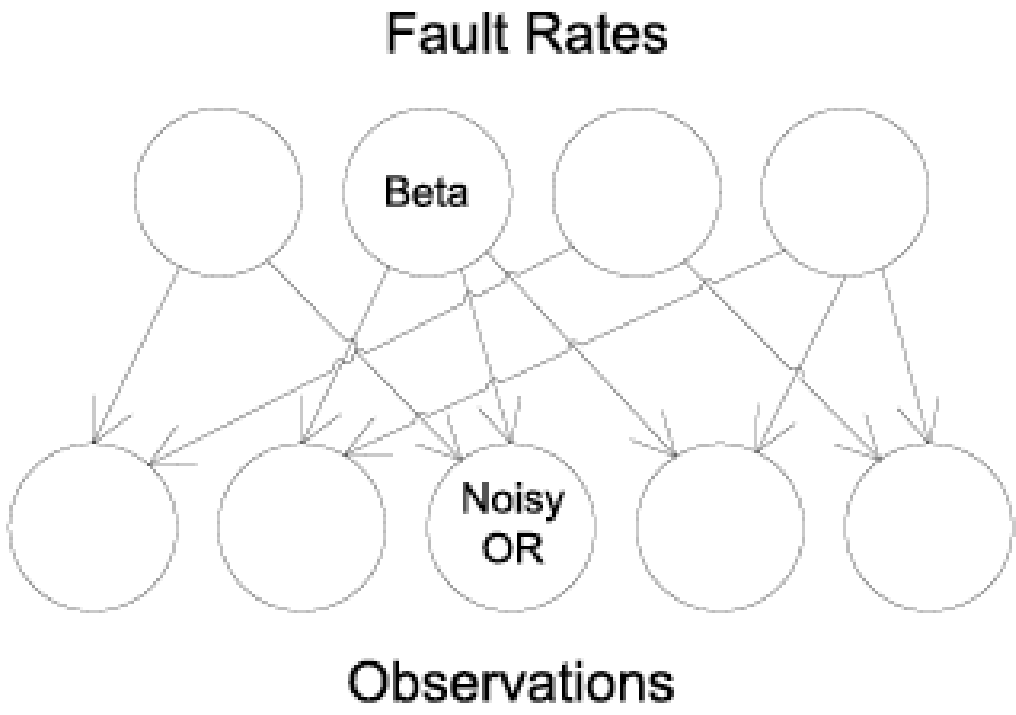


Figure 2: Graphical model after integrating out instantaneous faults: a bipartite noisy-OR network with Beta distributions as hidden variables

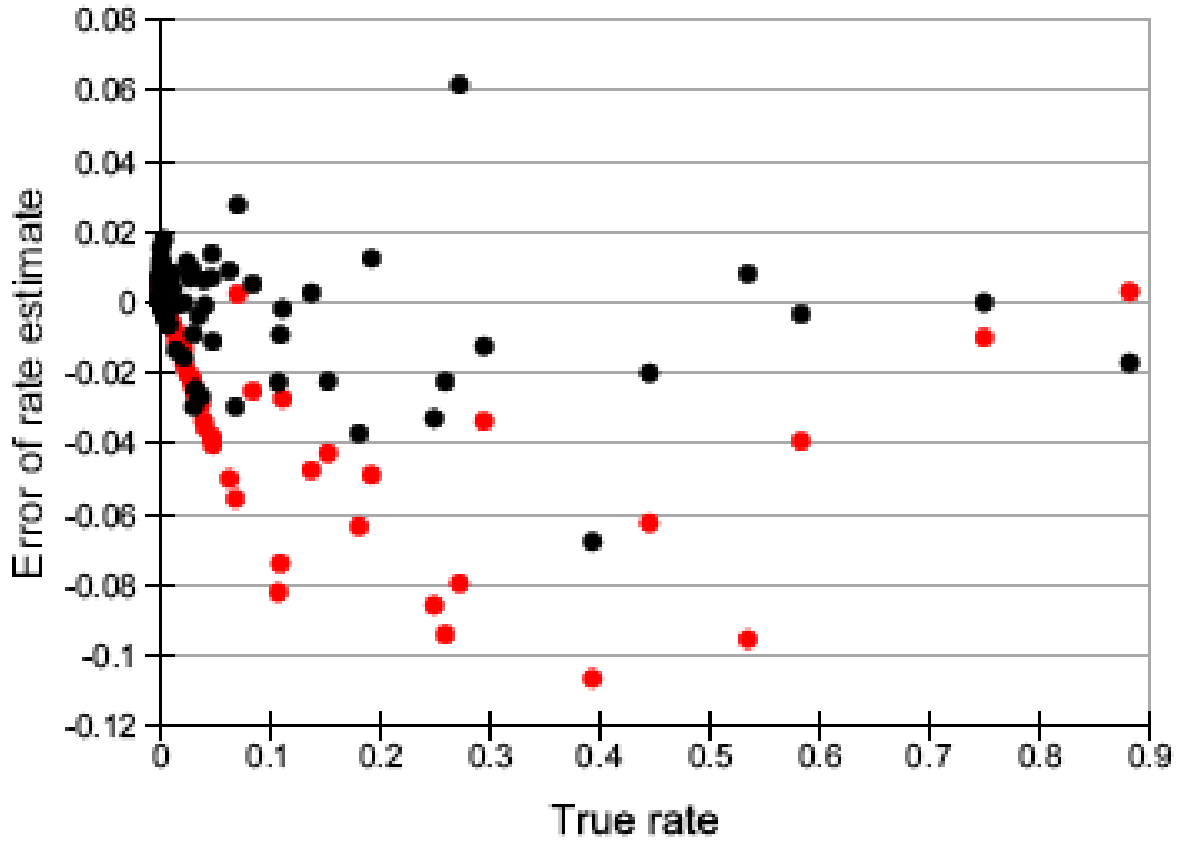


Figure 3: The error in estimate of rate versus true underlying rate. Black dots are L-BFGS, Red dots are Stochastic Gradient Descent with 20 epochs.

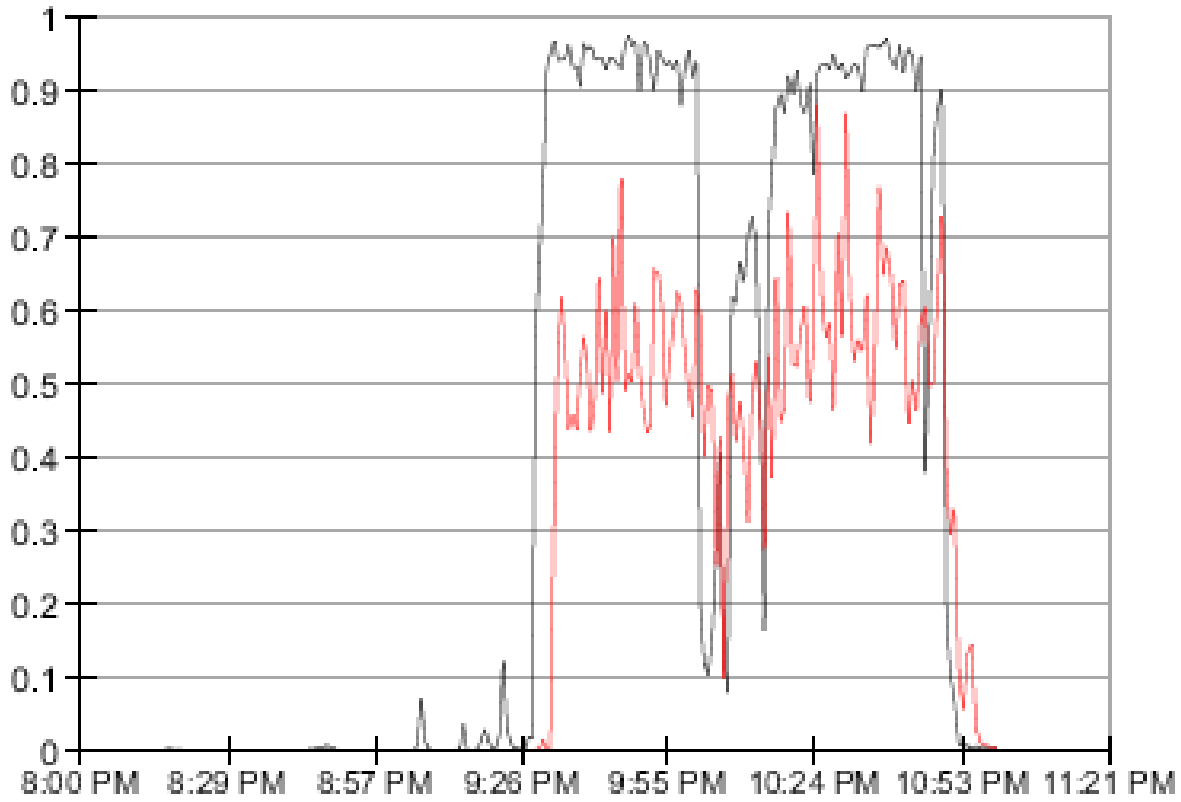


Figure 4: The inferred fault rate for two Autonomous Systems, as a function of time. These are the only two faults with high rate.

Inference for mixture models

Gibbs sampler

$$p(z_i = k|\mathbf{z}_{-i}, \mathbf{x}, \boldsymbol{\theta}) \propto p(z_i = k|\mathbf{z}_{-i}) p(x_i|\mathbf{x}_{-i}, z_i = k, \mathbf{z}_{-i}, \boldsymbol{\theta})$$

Gibbs sampler for mixture models

- Mixture model with K components and parameters $\theta_1, \ldots, \theta_K$.

$$p(y|\boldsymbol{\theta}, \boldsymbol{\pi}) = \sum_{k=1}^{K} \pi_k p(y|\theta_k)$$

- Consider set of data $D = (y_1, \ldots, y_n)$. Introduce latent variables $\vec{c} = (c_1, \ldots, c_n)$ with $c_i =$ component of datapoint i.

  Likelihood: (assuming we know the c_i)

$$p(D, \vec{c}|\boldsymbol{\theta}, \boldsymbol{\pi}) = \prod_{i=1}^{n} p(y_i|\theta_{c_i}) \prod_{j=1}^{K} \pi_j^{n_j}$$

  with $n_j =$ number of data with $c_i = j$.

- Priors: $p(\boldsymbol{\theta}) = \prod_{j=1}^{K} p(\theta_j)$

Prior for mixture weights: Dirichlet distribution

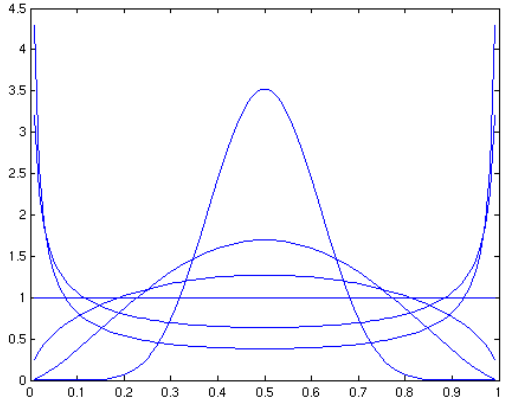
These give prior distributions over discrete probabilities π_j which give e.g. the weights needed for mixture models

$$p(\boldsymbol{\pi}|\boldsymbol{\beta}) = \frac{1}{B(\boldsymbol{\beta})} \prod_{j=1}^{K} \pi_j^{\beta_j - 1} \delta\left(\sum_j \pi_j - 1\right)$$

with normalisation

$$B(\boldsymbol{\beta}) = \frac{\prod_j \Gamma(\beta_j)}{\Gamma\left(\sum_j \beta_j\right)}$$

Marginals $\quad p(\pi_i) \propto \pi_i^{\beta_i - 1} (1 - \pi_i)^{\sum_{j=1}^{K} \beta_j - \beta_i - 1}$



- Joint probabilities of all variables

$$p(D, \vec{c}, \boldsymbol{\theta}, \boldsymbol{\pi}) \propto \prod_{i=1}^{n} p(y_i|\theta_{c_i}) \prod_{j=1}^{K} \pi_j^{n_j} \prod_{j=1}^{K} \pi_j^{\beta_j - 1} \prod_{j=1}^{K} p(\theta_j)$$

- Integrate out $\boldsymbol{\pi}$

$$\int \prod_{j=1}^{K} \pi_j^{n_j + \beta_j - 1} d\boldsymbol{\pi} \propto \prod_{j=1}^{K} \Gamma(n_j + \beta_j)$$

Hence

$$p(D, \vec{c}, \boldsymbol{\theta}) \propto \prod_{i=1}^{n} p(y_i|\theta_{c_i}) \prod_{j=1}^{K} \Gamma(n_j + \beta_j) \prod_{j=1}^{K} p(\theta_j)$$

Gibbs sampling

Goal: Draw samples from posterior $p(\boldsymbol{\theta}, \vec{c}|D)$

- Sample the components: For each datapoint i note that $n_c = n_{-i,c} + 1$ and

$$\frac{\Gamma(n_{-i,c} + 1 + \beta_c)}{\Gamma(n_{-i,c} + \beta_c)} = n_{-i,c} + \beta_c$$

  Hence

$$p(c_i = c|\vec{c}_{-i}, \boldsymbol{\theta}, D) \propto p(y_i|\theta_c)(n_{-i,c} + \beta_c)$$

- Sample the parameters: For each component j

$$p(\theta_j|\vec{c}, \boldsymbol{\theta}_{-j}, D) \propto p(\theta_j) \prod_{i=1, c_i=j}^{n} p(y_i|\theta_j)$$

Example: Mixture of 2 Gaussians

We use fixed variances $\sigma_{1,2}^2 = 1$. The variables are $\boldsymbol{\theta} = \mu_1, \mu_2$ and $c_i \in \{1, 2\}$. We assume a flat (improper prior) on $\boldsymbol{\mu}$ and $\beta_k = 1$ for the (flat) Dirichlet density for simplicity.

The **update for the** c_i is

$$c_i = 1 \quad \text{with probability } b(n_{-i,1} + 1)e^{-\frac{1}{2}(y_i - \mu_1)^2}$$
$$c_i = 2 \quad \text{with probability } b(n_{-i,2} + 1)e^{-\frac{1}{2}(y_i - \mu_2)^2}$$

where $1/b = (n_{-i,1} + 1)e^{-\frac{1}{2}(y_i - \mu_1)^2} + (n_{-i,2} + 1)e^{-\frac{1}{2}(y_i - \mu_2)^2}$.

The **means are updated** as $\mu_{1,2} \sim \mathcal{N}(\widehat{\mu}_{1,2}, 1/n_{1,2})$ where

$$\widehat{\mu}_{1,2} = \frac{1}{n_{1,2}} \sum_{i:c_i=1,2} x_i$$

ML estimates were $\pi_1 = 0.35$, $\mu_1 \approx -0.2413$, $\mu_2 \approx 3.9424$.