

# Probabilistic and Bayesian Modelling in Machine Learning and Artificial Intelligence

Manfred Opper & Théo Galy-Fajou

July 10, 2018



## Background reading

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

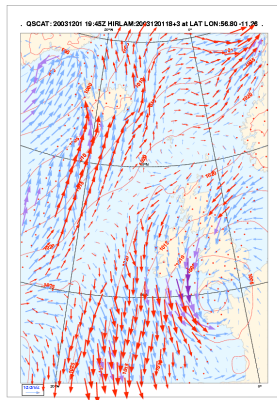
Information Theory, Inference, and Learning Algorithms, David J C MacKay, Cambridge University Press, 2003.

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

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

Computer Age Statistical Inference, Bradley Efron and Trevor Hastie, Cambridge University Press, 2016.

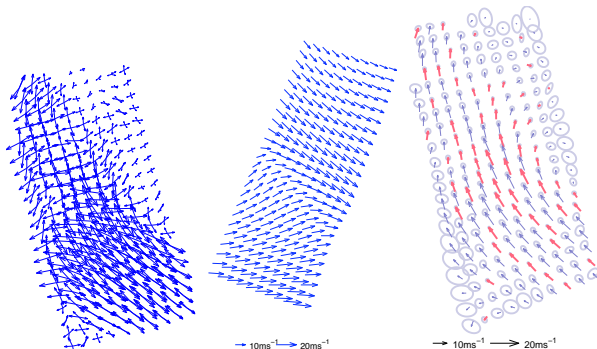
# Measuring Windfields



(Ad Stoffelen/KNMI)

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

# Ambiguities and prior knowledge



Likelihood  
mean prediction.

typical a priori sample

# Some probability essentials

## Definitions

*Sample Space*  $\Omega$ : Space of possible outcomes  $\omega$  of a random experiment.

*Events*: (measurable) subsets of  $\Omega$ .

*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:

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

for a set  $S \in \mathcal{R}^2$ .<sup>1</sup>

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

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  $A_1, A_2, \dots$  are called *independent* if for any subset

$$\{A_{i_1}, A_{i_2}, \dots, A_{i_k}\} \quad P(A_{i_1} \cap \dots \cap A_{i_k}) = P(A_{i_1})P(A_{i_2}) \cdots P(A_{i_k}).$$

A family of random variables  $X_1, X_2, \dots$  are called *independent* if for any subset  $\{X_{i_1}, X_{i_2}, \dots, X_{i_k}\}$

$$P(X_{i_1}, X_{i_2}, \dots, X_{i_k}) = P(X_{i_1})P(X_{i_2}) \cdots P(X_{i_k}) = \prod_{j=1}^k P(x_{i_j}) \text{ (with an analogous definition for densities). Hence, if } X \text{ and } Y \text{ independent then}$$
$$P(x|y) = \frac{P(x,y)}{P(y)} = P(x).$$

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

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

- $\text{Var}\left(\sum_{i=1}^N X_i\right) = \sum_{i=1}^N \text{Var}(X_i).$

- Law of large numbers

Let  $X_1, X_2, \dots, X_N$ , i.i.d. with finite variance  $\sigma^2$  and

$S_N = \frac{1}{N} \sum_{i=1}^N X_i$ , then one can show that

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

Hence, when  $N$  large, with high probability we have

$$\frac{1}{N} \sum_{i=1}^N X_i \approx E(X).$$

The proof uses additivity 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  $\mathbf{x} = (x_1, \dots, x_d)^T$  and  $\boldsymbol{\mu} = (\mu_1, \dots, \mu_d)^T$

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

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

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

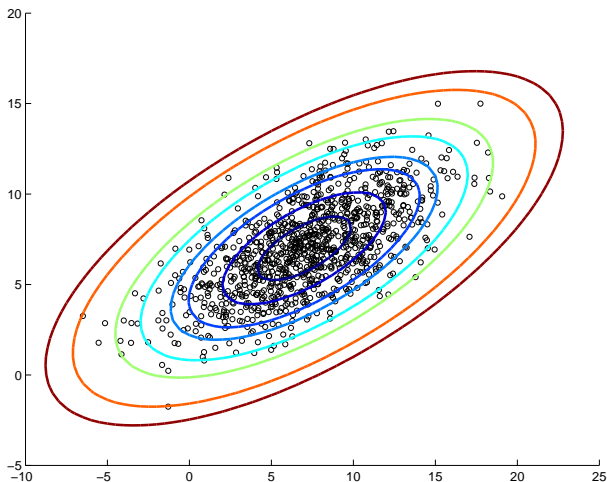
$$\Sigma_{ij} = E(x_i - \mu_i)(x_j - \mu_j)$$

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

### Example:

Lines of constant density and random data for a two dimensional Gaussian.

The mean is  $\mu = (7, 7)^T$  and the covariance matrix is  $\Sigma = \begin{pmatrix} 16.6 & 6.8 \\ 6.8 & 6.4 \end{pmatrix}$



# Eigenvalue problem for $\Sigma$

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

$$\Sigma \mathbf{u}_i = \lambda_i \mathbf{u}_i \quad (2)$$

with an eigenvector  $\mathbf{u}_i$  and eigenvalue  $\lambda_i$ , where  $i = 1, \dots, d$ .  $\Sigma$  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

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

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

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

$$\mathbf{\Sigma} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T \quad (4)$$

and

$$\mathbf{\Sigma}^{-1} = \mathbf{U}\mathbf{\Lambda}^{-1}\mathbf{U}^{-1} = \mathbf{U}\mathbf{\Lambda}^{-1}\mathbf{U}^T \quad (5)$$

$\mathbf{U}$  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

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

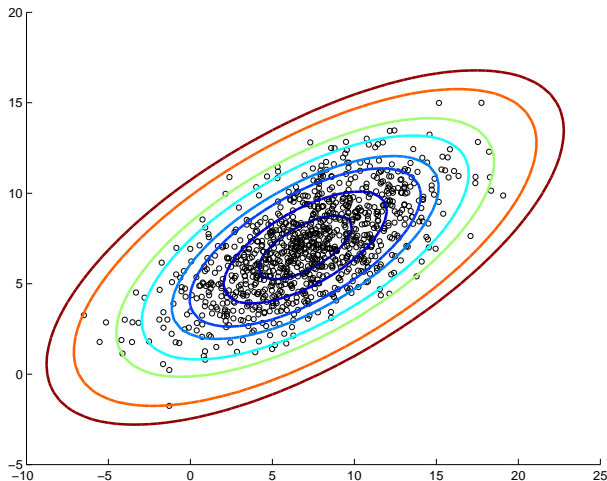
Using the new coordinate system, we see that

- 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*, i.e.

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

- We see that  $\Sigma$  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  $\Sigma = \begin{pmatrix} 16.6 & 6.8 \\ 6.8 & 6.4 \end{pmatrix}$ . 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 *independed* 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  $x \rightarrow x^2$ .

### 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 (Bernoulli model)

Consider a data sequence  $D = (x_1, x_2, \dots, x_n)$  of bits  $x_i \in \{0, 1\}$  which we believe are generated independently at random with the same probability. Call  $\theta$  the **unknown** probability 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  $\theta$ . We call it the **likelihood**.

To **estimate** the **true parameter**  $\theta$  of the model from which the data was generated we use the method of Maximum Likelihood choosing  $\hat{\theta} = \operatorname{argmax} P(D|\theta)$ . 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

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

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

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

## Example III: Gaussian noise and Linear Regression

Minimization with respect to  $\mu$  and  $\sigma^2$  leads to the *Maximum Likelihood Estimates*

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

Observe a set of input–output data  $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$  with  $x$  = input,  $y$  = target values. Try to fit a linear function  $y = w_0 + w_1x$  to the data. We represent this as a probabilistic model and assume that  $n$  observations are generated as

$$y_i = w_0 + w_1x_i + \text{noise}_i$$

for  $i = 1, \dots, n$ .



For independent Gaussian noise of variance  $\sigma^2$  we can write

$$p(y, x | \mathbf{w}) = p(y | x, \mathbf{w}) p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y - w_0 - w_1 x)^2}{2\sigma^2}} p(x)$$

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

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  $i = 1, \dots, N$ , with  $f(\cdot)$  unknown,  $\nu_i$  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

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

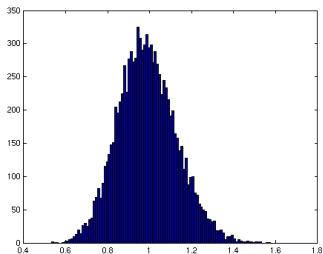
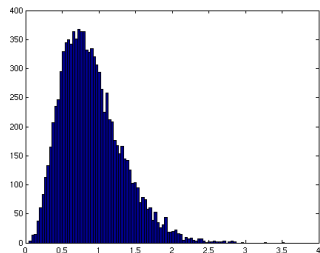
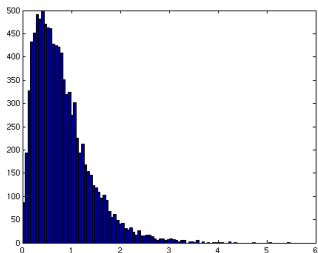
# Properties of Estimators

- Parameter estimates  $\hat{\theta}(D)$  are random variables with respect to the random drawing of the data. The *bias* of an estimator is defined as  $E_D(\hat{\theta}) - \theta$  and its *variance* as  $E_D \left( \hat{\theta} - E_D(\hat{\theta}) \right)^2$ , where the expectation  $E_D$  is over datasets which are drawn at random from a distribution with *true* parameter  $\theta$ .
- “Good” estimators should become asymptotically *consistent*, i.e. the estimates should converge to the *true* parameters as  $N \rightarrow \infty$ . This means that bias and variance must go to 0 as  $N \rightarrow \infty$ .
- ML estimators are consistent under rather general circumstances. Note that

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

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*<sup>†</sup>) **exponential families** which in their **canonical representation** are written as

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

$\boldsymbol{\psi}$  is the **natural parameter** and  $\boldsymbol{\phi}(x)$  the sufficient statistics.

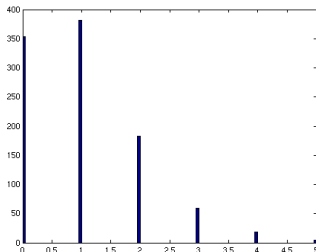
For a Gaussian, take  $\boldsymbol{\psi}(\boldsymbol{\theta}) = (\mu/\sigma^2, 1/2\sigma^2)$  and  $\boldsymbol{\phi}(x) = (x, -x^2)$ .

(<sup>†</sup> regular means that the range of the data  $x$  is independent of the parameter  $\theta$ ).

# Another exponential family: Poisson distributions

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

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



## Example: Multinomial family

Let  $\mathbf{n} = (n_1, \dots, n_K)$ , with  $n_j \in \mathbb{N}$  and  $\sum_j n_j = n$ , we define the Multinomial 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).

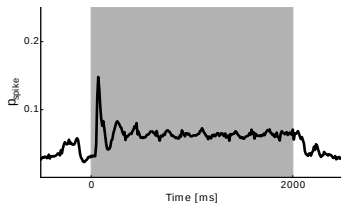
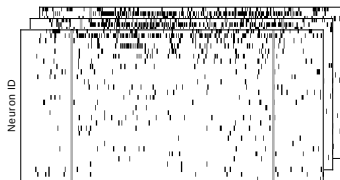
Sufficiency: Let  $p(\mathbf{x}|\theta)$  be a parametric family. A statistics  $T(\mathbf{x})$  of the sample  $\mathbf{x} = \{x_1, x_2, \dots, x_n\}$  is called **sufficient** if the conditional probability

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

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

For exponential families,  $\mathbf{T}(\mathbf{x}) = \sum_{i=1}^n \phi(x_i)$  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  $i = 1, \dots, N$ . Joint distribution of the variables is defined as (Markov random field)

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

- 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  $\theta_{ij}$  and  $\theta_i$  by gradient descent requires computation of  $E[x_i x_j]$  and  $E[x_i]$ . 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-log-likelihood*:

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

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

- Justification: Show that

$$E [\nabla_{\theta} \ln P(x_i | \mathbf{x}_{-i}, \theta)] = 0$$

when  $\theta$  is the true parameter vector.

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

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

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  $(z_1, \dots, z_k)$  (we specialise to **unbiased** estimators  $E(\hat{\theta}) = \theta$  for simplicity)

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

with the **Fisher Information** matrix

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

For  $z_i \geq 0$ , we can interpret the left hand side as a squared weighted average of the individual error components  $\hat{\theta}_i - \theta_i$ . 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)

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

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

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

**Note:** A different representation of the Fisher Information is

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

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

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

for  $n \rightarrow \infty$ . where

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

and

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

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

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

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  $\theta$  will be the estimate for  $\theta$ , depends only on the distance  $\|d\theta\|$ .



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

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

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:

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

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.

**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

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

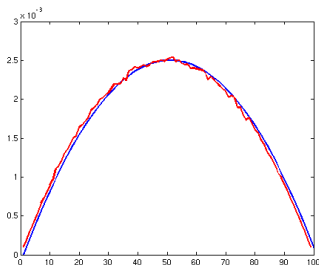
The differential operator  $\mathbf{J}^{-1}(\theta(n)) \nabla_{\theta}$  is termed natural gradient. For the choice  $\gamma_n = \frac{1}{n}$ , one can show that the online algorithm yields *asymptotically efficient* estimation.

# Example: Fisher Information

Bernoulli random variables

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

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



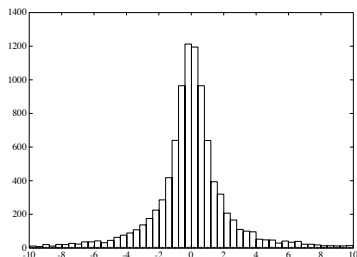
Cauchy density

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

# Estimating a Cauchy Density

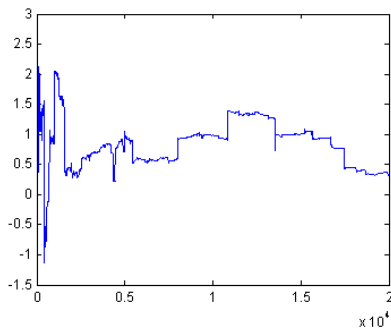
We consider the family of Cauchy densities given by

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

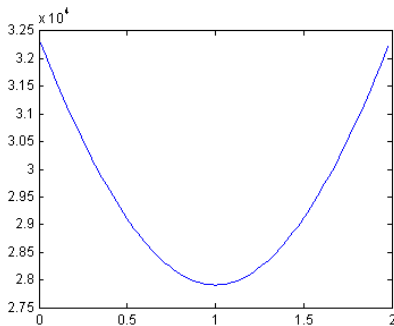


with location parameter  $\theta$ .

**Naive estimate**  $\hat{\theta} = \frac{1}{n} \sum_i x_i$   
 $-\ln p(D|\theta)$ . (true  $\theta = 1$ ).



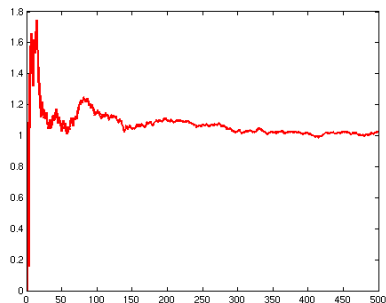
negative log-likelihood



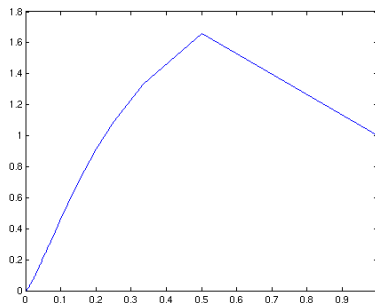
# Natural gradient

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

Prediction  $\theta_n$  (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.



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

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

Goal:

Demix the signals and recover sources

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

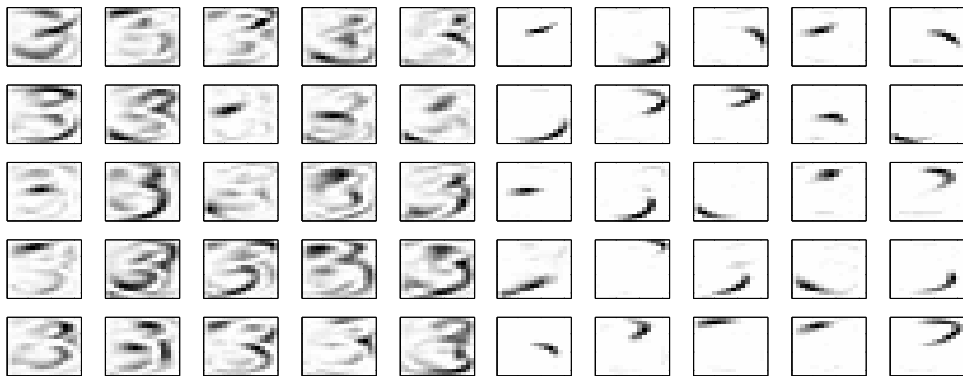
with  $\mathbf{W} = \mathbf{A}^{-1}$  for square matrices and no noise.

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

# 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



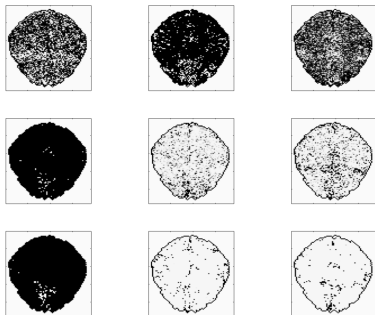
**left:** unconstrained      **right:** constrained (positive) mixing matrix  $\mathbf{A}$ .

$x_i(t)$  = sequence of 500 images (handwritten '3's).  $p(s) = e^{-s}$ ,  $s \geq 0$ .

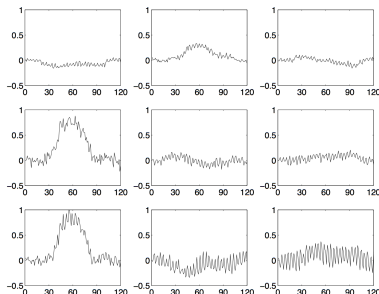
Shown are the  $m = 25$  columns  $\mathbf{A}_{\bullet j}$  of the matrix  $\mathbf{A}$ .

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

**left:** Posterior mean sources



**right:** responses  $A_{\bullet i}$  for  $i = 1, \dots, 9$ .



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

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

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.  $\Sigma \mathbf{U} = \mathbf{U} \Lambda$ . If we set  $\mathbf{W} = \Lambda^{-\frac{1}{2}} \mathbf{U}^T$ , then the vector

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

For Gaussian signals: decorrelated = independent!

BUT any  $\mathbf{QW}$  with orthogonal  $\mathbf{Q}$  (i.e.  $\mathbf{Q}\mathbf{Q}^T = \mathbf{I}$ ) 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**  $p_i(s) \propto \frac{1}{e^s + e^{-s}}$ .

$$\Delta W_{ij} = \eta \left[ n(\mathbf{W}^{-1})_{ji} + \sum_{t=1}^n X_{jt} \phi_i \left( \sum_{r=1}^d W_{ir} X_{rt} \right) \right] \quad (8)$$



$$\phi_i(s) = \frac{d}{ds} \ln p_i(s) \quad (9)$$

- Hyperbolic secant distribution:

$$P(s) = \frac{1}{Z \cosh(s)} \quad (10)$$

$$\phi(s) = -\tanh(s) \quad (11)$$

- Gain-Factor:

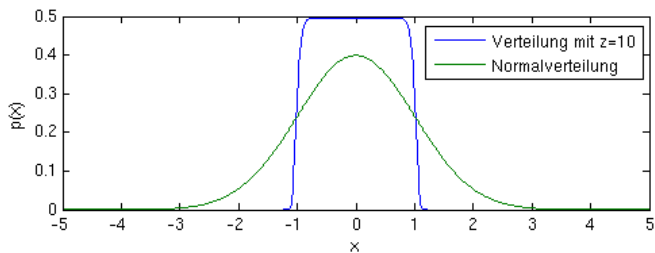
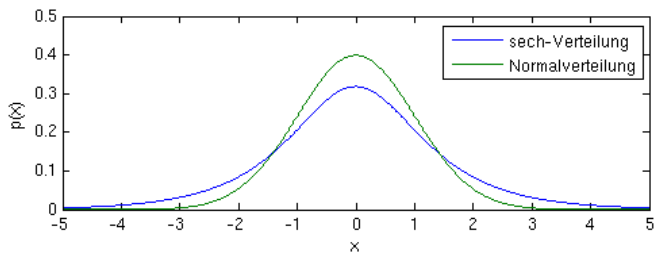
$$P(s) = \frac{1}{Z \cosh(s)^{1/z}} \quad (12)$$

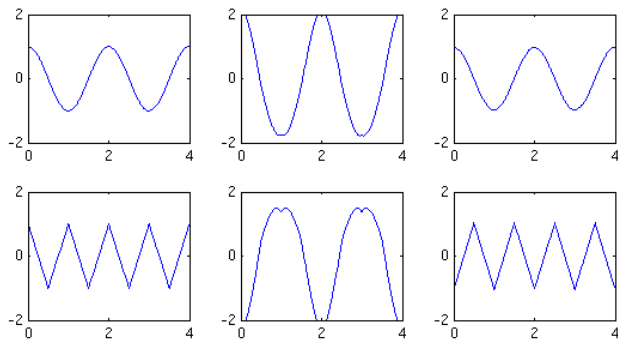
$$\phi(s) = -\tanh(zs) \quad (13)$$

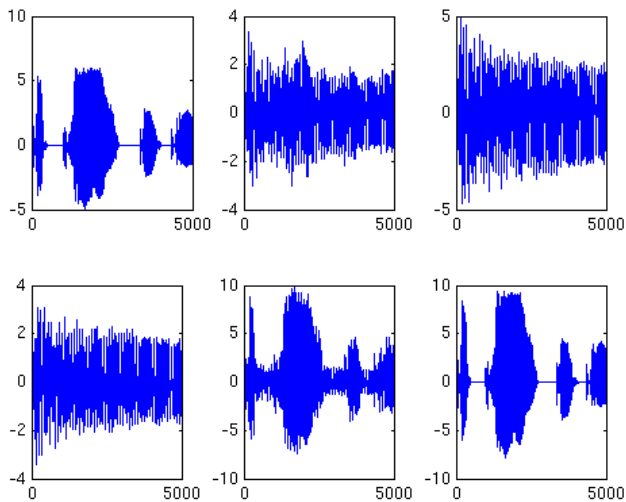
- Modified Gaussian:

$$P(s) = \frac{1}{Z} \exp(-|s|^{2z}/2) \quad (14)$$

$$\phi(s) = -zs|s|^{2(z-1)} \quad (15)$$







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.

- 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

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

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

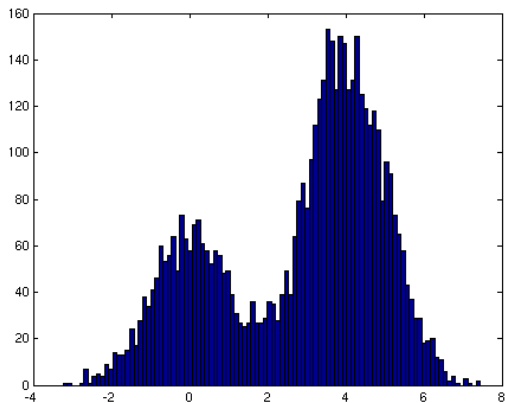
$y_i$  observed, component  $c_i$  hidden,

$\boldsymbol{\theta} = \{\mu_c, \sigma_c, p(c)\}_{c=1}^K$  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



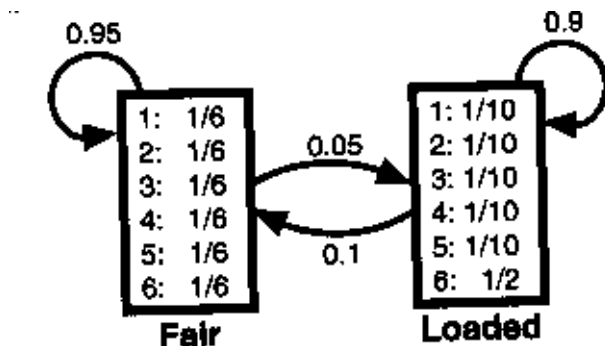
# Example II: Hidden Markov Models

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

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

- 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!!

# The Expectation–Maximisation (EM) Algorithm

- 1 Start with arbitrary  $\theta_0$

**Iterate:**

- 2 (E-Step): Compute the expectation

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

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

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

- 3 (M-Step) Maximise

$$\theta_{t+1} = \operatorname{argmax}_{\theta} \mathcal{L}(\theta, \theta_t)$$

**Claim:**  $\ln p(\mathbf{y}|\theta_{t+1}) \geq \ln p(\mathbf{y}|\theta_t)$  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

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

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  $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\theta}_t) = -F(q_t, \boldsymbol{\theta}) + \sum_{\mathbf{x}} q_t(\mathbf{x}) \ln q_t(\mathbf{x})$

Hence, the EM algorithm can be reformulated as:

- 1 E-Step: Minimise  $F(q, \boldsymbol{\theta}_t)$  w.r.t  $q \rightarrow q_t(\mathbf{x})$  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  $\boldsymbol{\theta}_{t+1} = \operatorname{argmax}_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\theta}_t)$



- Variation with respect to  $\mu_c$

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

- Variation with respect to  $\sigma_c^2$

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

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

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

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  $\theta$  **before** she/he has seen the data, should be represented by a prior distribution. Take eg

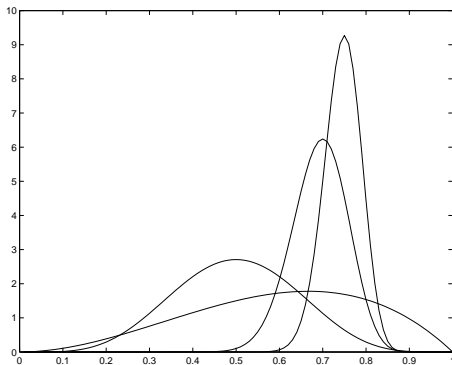
$$p(\theta) = 1 \quad \text{for } 0 \leq \theta \leq 1 .$$

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  $\theta$  **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  $\theta$  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  $\theta$ , ie the value which has the **Maximum Posterior** probability (density). For our choice of prior, this coincides with the ML value. Another estimator is the the **posterior** mean of  $\theta$  which is given by

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

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

$$L_2(\hat{\theta}) = \int (\hat{\theta} - \theta)^2 p(\theta|D) d\theta$$

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

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  $\sigma^2$  is known but  $\mu$  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

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

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



# Conjugate priors

For exponential families, conjugate priors allow for simple computations:

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

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

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

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

If we have a variety of models  $\mathcal{M}_1, \mathcal{M}_2, \dots$  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  $i = 1, \dots, N$ , with  $f(\cdot)$  unknown,  $\nu_i$  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

$$E[\mathbf{w}|D] = \left( \frac{\sigma^2}{\sigma_0^2} \mathbf{I}_{K+1} + \mathbf{X}^T \mathbf{X} \right)^{-1} \mathbf{X}^T \mathbf{y} \quad (16)$$

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:

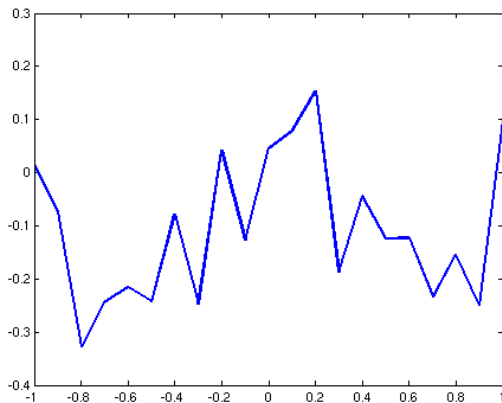
$$\ln p(D) = -\frac{N}{2} \ln 2\pi - \frac{1}{2} \ln |\boldsymbol{\Sigma}| - \frac{1}{2} \mathbf{y}^T \boldsymbol{\Sigma}^{-1} \mathbf{y} , \quad (17)$$

where

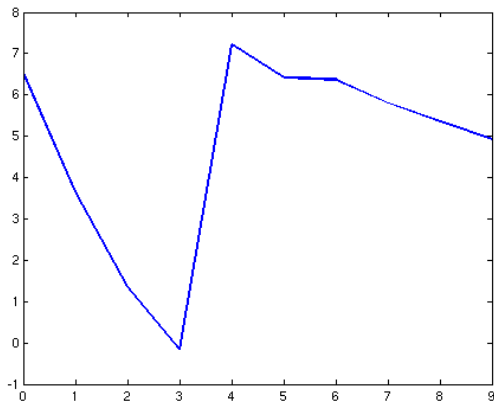
$$\boldsymbol{\Sigma} = \sigma_0^2 \mathbf{X} \mathbf{X}^T + \sigma^2 \mathbf{I}_N \quad (18)$$

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  $\sigma_0^2 = 1$ .

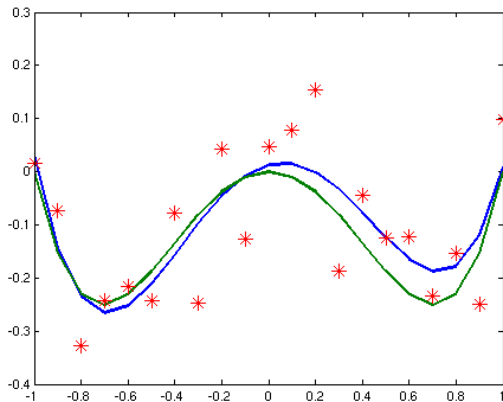
Typical observations



## Log-evidence as function of $K$

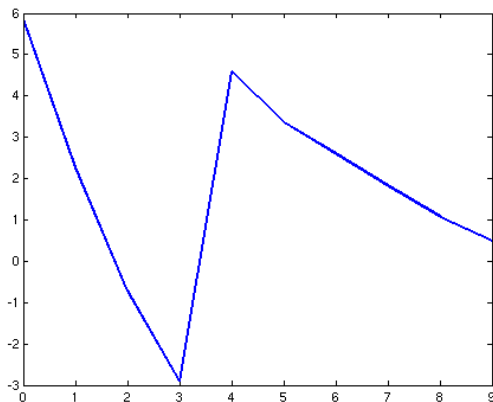


## Reconstruction using posterior mean $E[\mathbf{w}|D] = \int d\mathbf{w} p(\mathbf{w}|D) f_{\mathbf{w}}(x)$



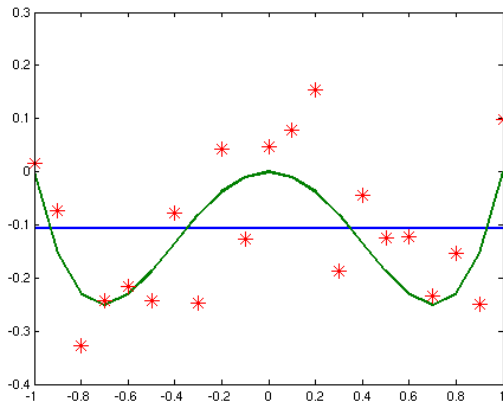
The same, but now with a different prior  $\sigma_0 = 2$

Log-evidence as function of  $K$



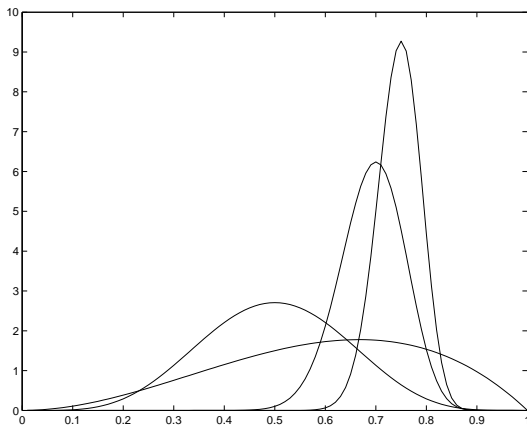


Reconstruction using posterior mean  $E[\mathbf{w}|D] = \int d\mathbf{w} p(\mathbf{w}|D) f_{\mathbf{w}}(x)$




# Computational tools I: Laplace approximation

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



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

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


with

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

Hence, in the posterior, the dominating term

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

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

For finite dimensional parametric models with continuous priors we have

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

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

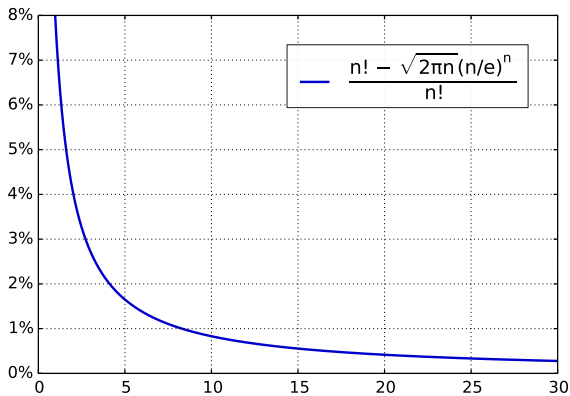
# Laplace approximation

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

$$\begin{aligned}\int e^{-h(\mathbf{z})} d\boldsymbol{\theta} &\approx e^{-h(\hat{\mathbf{z}})} \int \exp \left[ -\frac{1}{2}(\mathbf{z} - \hat{\mathbf{z}})^T \mathbf{A}(\mathbf{z} - \hat{\mathbf{z}}) \right] d\mathbf{z} \\ &= e^{-h(\hat{\mathbf{z}})} \frac{(2\pi)^{K/2}}{|\mathbf{A}|^{1/2}}\end{aligned}$$

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

## Application: Stirling approximation



## Approximating the evidence

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

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

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

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

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



# Application: Bayesian Neural Networks

Consider neural network input-output

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

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

Probabilistic model:

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

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

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

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

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  $\lambda_k$  are eigenvalues of  $\beta \nabla^2 E_D^{MP}$ .  
Start with some values of  $\alpha$  and  $\beta$ , optimise  $\hat{\mathbf{w}}$  and re-estimate

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

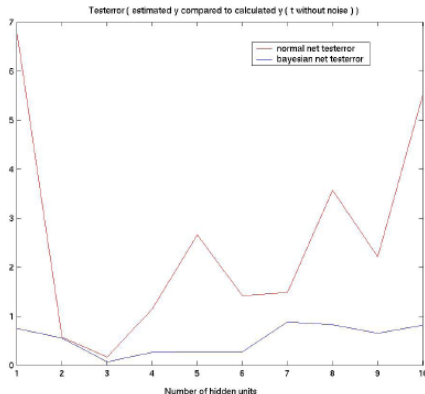
optimise  $\hat{\mathbf{w}}$  and repeat until convergence.

ARD: The method can be extended to separate  $\alpha_k$ s for each input neuron.  
Large  $\alpha_k$  leads to a 'shut off' for the corresponding weights.

# Example

Artificial data set: *Friedman data* generated as

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



ARD:  $\alpha_i$  for network inputs  $x_i$ :

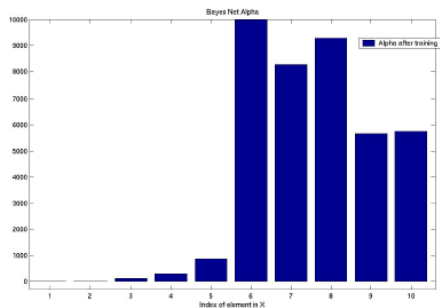


Figure 5: 5 hidden units, 200 training samples, 100 training loops, 50 evidence-iterations, zoomed into diagram

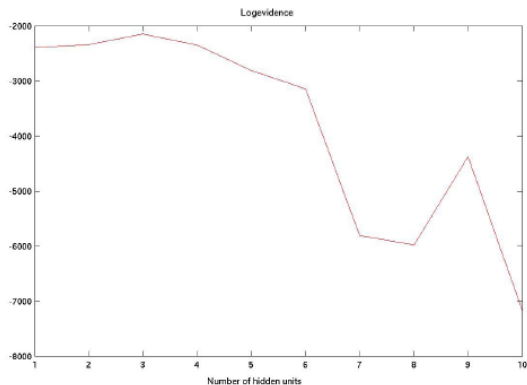


Figure 7: 200 training samples, 30 training loops, 30 evidence-iterations

# 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.



**Goal:** Represent probability distributions by random samples.

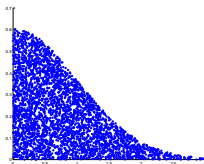
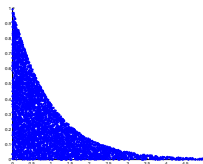
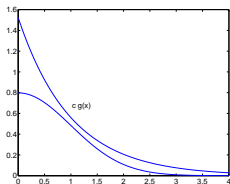
Hence, we have to be able to generate (usually dependent!) samples from a given distribution  $p(x)$ . In the application to Bayesian models case  $x$  is set of parameters and  $p$  the posterior.

# 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  $Z$ .

- It is 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  $1 - A(x'; x)$



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

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

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

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

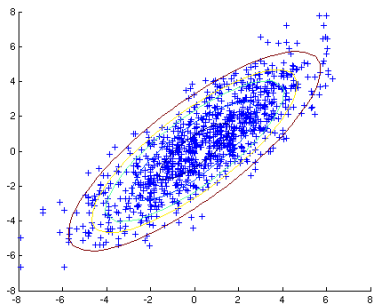
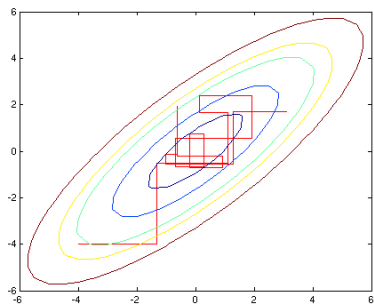
$$\begin{aligned} p(x)T(x'|x) &= p(x)q(x'|x) \min \left( 1, \frac{p(x')q(x|x')}{p(x)q(x'|x)} \right) = \\ &= \min (q(x'|x)p(x), q(x|x')p(x')) = p(x')q(x|x')A(x; x') = p(x')T(x|x') \end{aligned}$$

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  $q$ .

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

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



# Gibbs sampler from Metropolis Hastings

Consider the Gibbs proposal  $q(\mathbf{x}'|\mathbf{x}) = p(x'_i|\mathbf{x}_{-i})\delta_{\mathbf{x}_{-i}}(\mathbf{x}'_{-i})$  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  $i \in \{1, 2, \dots, n\}$ . 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  $\lambda_1$  to  $\lambda_2$  at unknown **change point**  $K \in \{1, 2, \dots, n\}$ .

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  $\eta_{1,2} \sim \text{Gamma}(b_{1,2}, c_{1,2})$  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  $D = (x_1, \dots, x_n)$  over  $n$  years, draw samples from the **posterior distribution**  $p(K, \eta, \lambda|\mathbf{x})$ .

- Joint distribution

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

$$\begin{aligned} & \prod_{i=1}^K e^{-\lambda_1} \frac{\lambda_1^{x_i}}{x_i!} \times \prod_{K+1}^n e^{-\lambda_2} \frac{\lambda_2^{x_i}}{x_i!} \times \\ & \times \frac{\eta_1^{a_1}}{\Gamma(a_1)} \lambda_1^{a_1-1} e^{-\eta_1 \lambda_1} \times \frac{\eta_2^{a_2}}{\Gamma(a_2)} \lambda_2^{a_2-1} e^{-\eta_2 \lambda_2} \times \\ & \times \frac{c_1^{b_1}}{\Gamma(b_1)} \eta_1^{b_1-1} e^{-c_1 \eta_1} \times \frac{c_2^{b_2}}{\Gamma(b_2)} \eta_2^{b_2-1} e^{-c_2 \eta_2} \times \\ & \times p(K) \end{aligned}$$

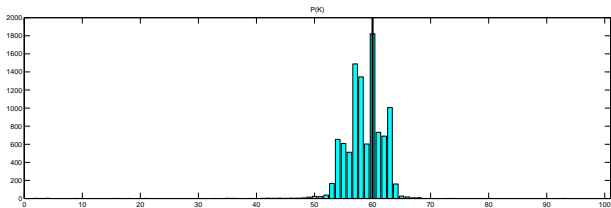
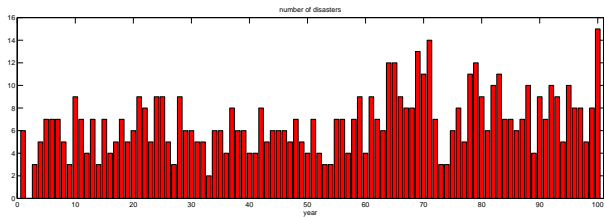
- Conditional distributions for Gibbs sampler

$$\lambda_2 | \lambda_1, \eta_{1,2}, K, \mathbf{x} \sim \text{Gamma}(a_2 + \sum_{K+1}^n x_i, n - K + \eta_2)$$

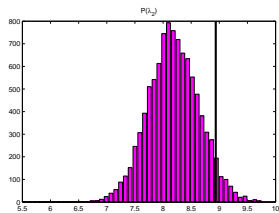
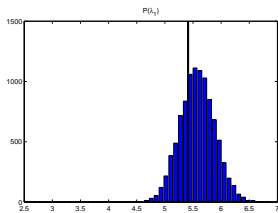
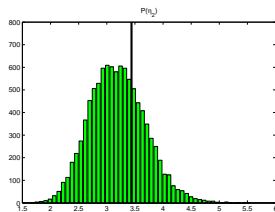
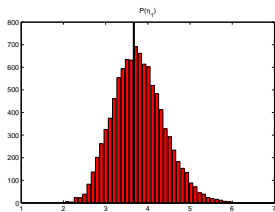
$$\eta_1 | \lambda_{1,2}, \eta_2, K, \mathbf{x} \sim \text{Gamma}(a_1 + b_1, \lambda_1 + c_1)$$

$$K | \lambda_{1,2}, \eta_{1,2}, \mathbf{x} \sim \text{const} \times p(K) e^{-K(\lambda_1 - \lambda_2)} (\lambda_1 / \lambda_2)^{\sum_{i=1}^K x_i}$$

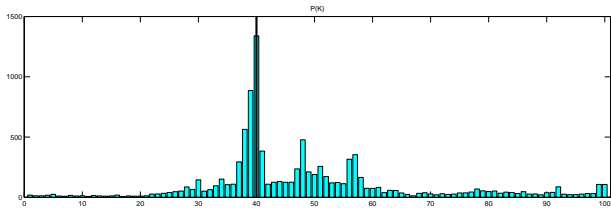
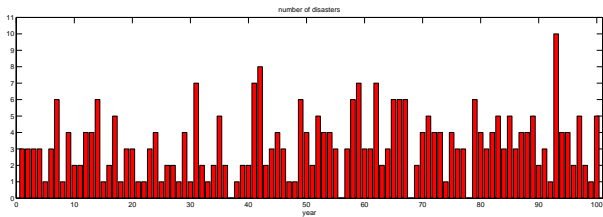
# Simulations

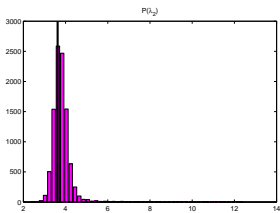
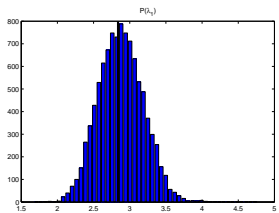
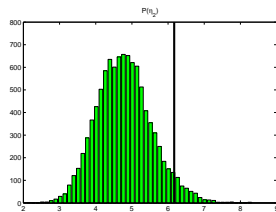
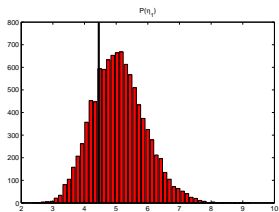






with somewhat more similar  $\lambda_{12}$





# More Metropolis-Hastings: Random walk sampler

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

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

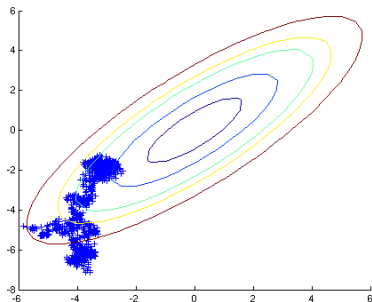
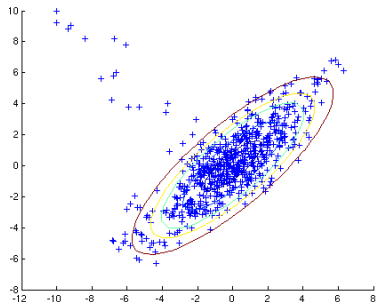
where  $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$ . This is a **symmetric proposal** with  $q(\mathbf{x}'|\mathbf{x}) = q(\mathbf{x}|\mathbf{x}')$ . The acceptance probability is then

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

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

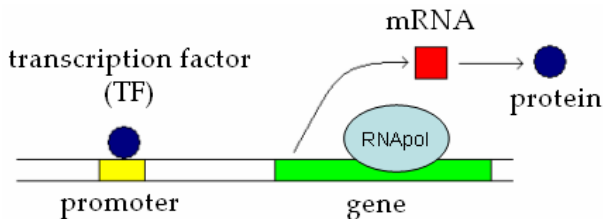
The choice of  $\rho$  is important. For large  $\rho$  acceptance will be unlikely. Small  $\rho$  will lead to high acceptance rates but too a very slow **diffusion**.

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



# Stochastic processes as a prior for unobserved dynamics :

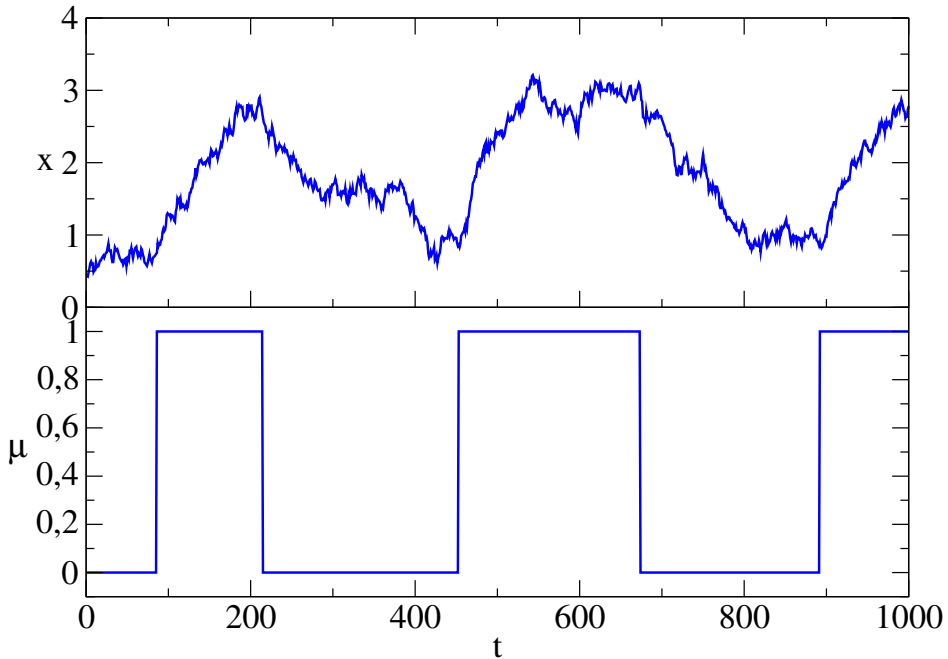
## Model for transcriptional regulation:

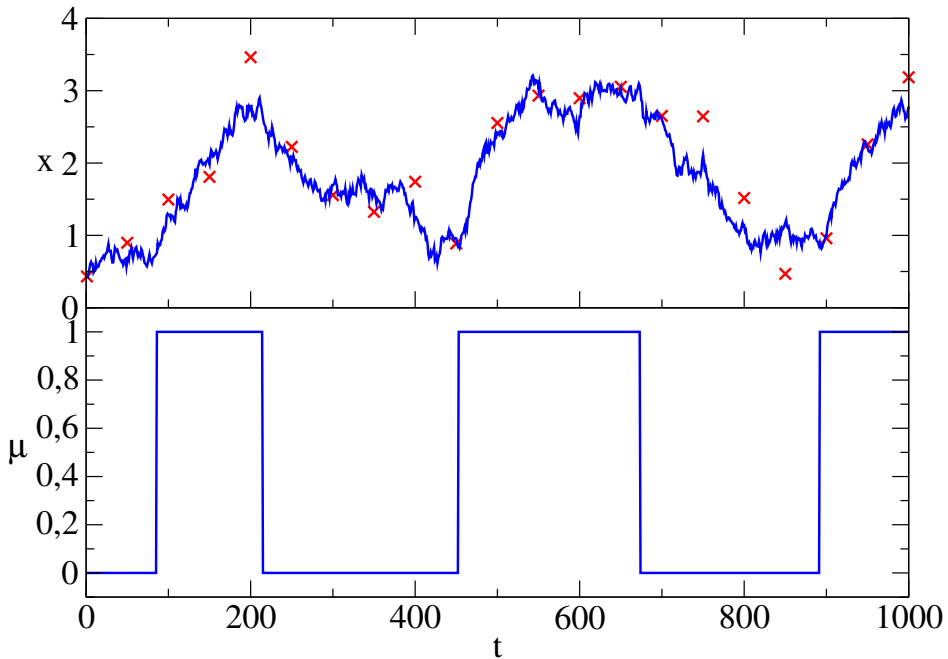


- $X(t)$  = mRNA concentration of target gene. modelled by an Ornstein - Uhlenbeck process

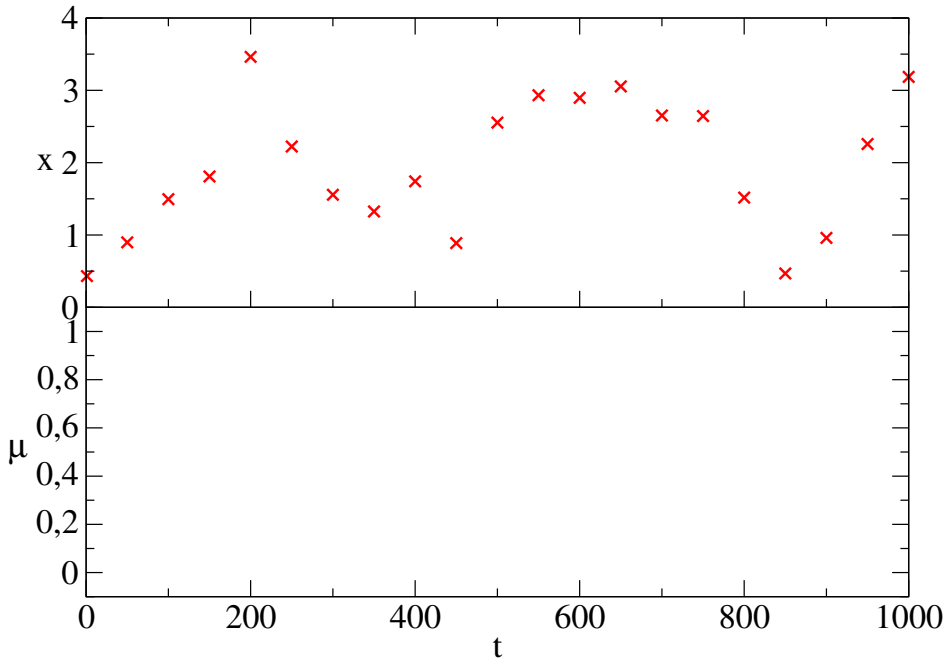
$$dX(t) = (A(t) + b - \lambda X(t))dt + \sigma dW(t)$$

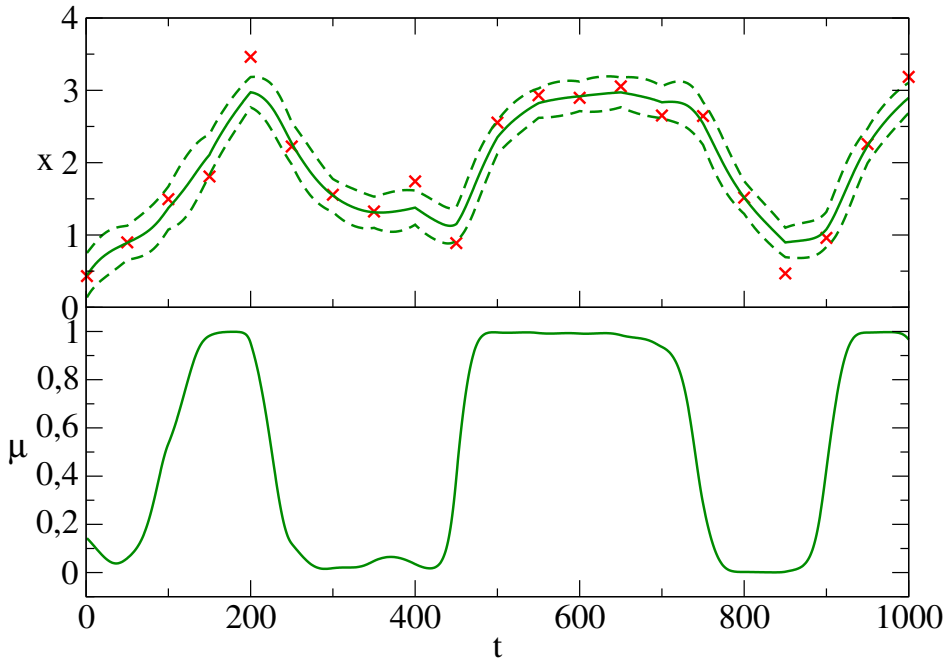
- $A(t)$  = fast switching transcription factor activity (unobserved) modelled by  $A(t) \sim \mathcal{TP}(f_{\pm})$  a **random telegraph process**.





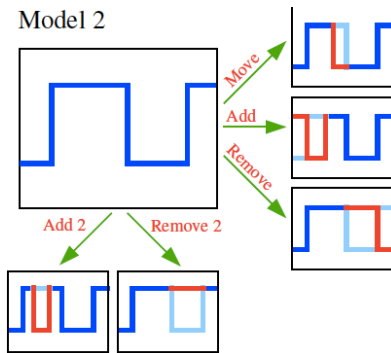






- Joint process  $X(t), A(t)$  Markov, but hard to sample from  $p(X(0 : T), A(0 : T) | \mathbf{Y})$ .
- Integrate out simple process  $X(0 : T)$  analytically given observations  $\mathbf{Y}$  and  $A(0 : T)$ .
- Sampling from  $p(A(0 : T) | \mathbf{Y})$  efficient if number of jumps small: Use **Metropolis–Hastings sampler**: Generate proposal changes of  $A(0 : T)$  (piecewise constant) and accept/reject with appropriate probabilities.

Model 2



# More Metropolis-Hastings: Independence sampler

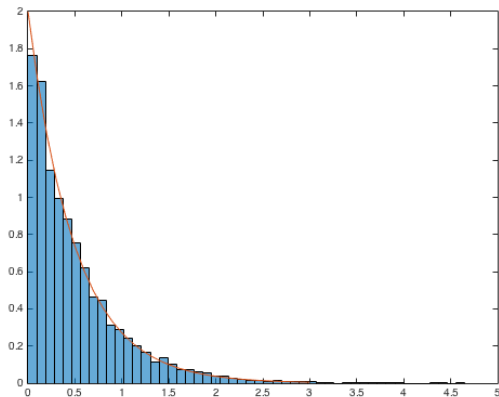
Let  $q(x'|x) = q(x')$  independent of  $x$  in the Metropolis method.  
Then the acceptance probability is

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

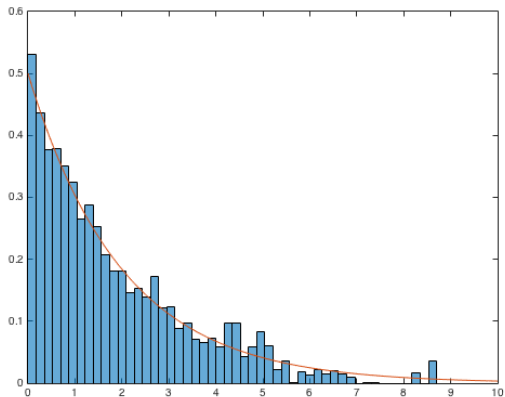
This is similar to a rejection method, but samples are dependent. Again,  $q$  should be similar to  $p$  to achieve good acceptance rates.

Target  $p(x) = \lambda e^{-\lambda x}$ ,  $x \geq 0$ .

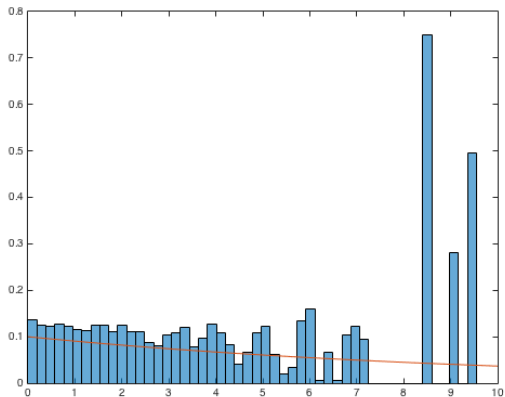
Proposal  $q(x) = e^{-x}$ ,  $x \geq 0$ . 10000 MCMC steps with  $\lambda = 2$ :



$\lambda = 0.5$ :

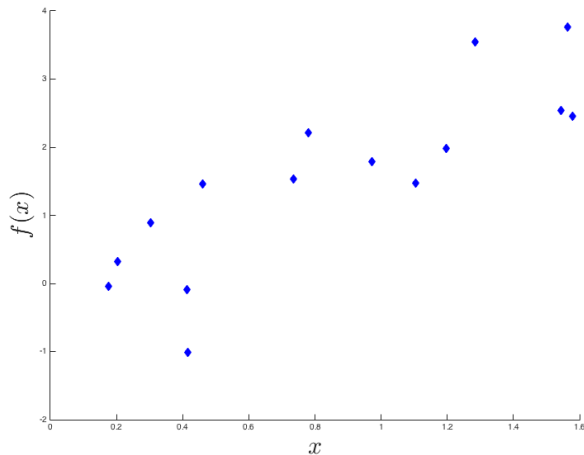


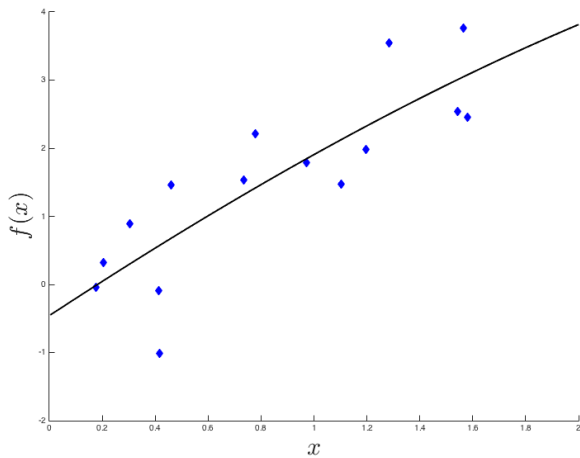
$\lambda = 0.1$ :

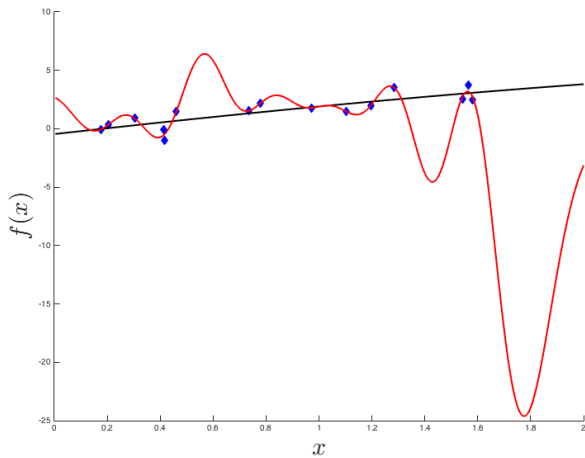




# From curve fitting to Gaussian processes







# A statistical model for curve fitting

## Generative model:

$$y_i = f_{\theta}(x_i) + \nu_i$$

- $\nu_i$  independent Gaussian noise of variance  $\sigma^2$
- $\theta$  unknown parameter of 'true' function  $f_{\theta}(x_i)$ .

# A statistical model for curve fitting

## Generative model:

$$y_i = f_{\theta}(x_i) + \nu_i$$

- $\nu_i$  independent Gaussian noise of variance  $\sigma^2$
- $\theta$  unknown parameter of 'true' function  $f_{\theta}(x_i)$ .
- Introduce likelihood

$$p(\text{Data}|\theta) = \prod_{i=1}^n p(y_i|\theta)$$

- and prior distribution  $p(\theta)$

# Regression with Gaussian noise

- Gaussian noise:  $p(y|\theta) \propto \exp\left(-\frac{1}{2\sigma^2}(y - f_\theta(x))^2\right)$
- Simple linear parametric form

$$f_\theta(x) = \theta x$$

# Regression with Gaussian noise

- Gaussian noise:  $p(y|\theta) \propto \exp\left(-\frac{1}{2\sigma^2}(y - f_\theta(x))^2\right)$
- Simple linear parametric form

$$f_\theta(x) = \theta x$$

- for Gaussian prior  $p(\theta)$ , the posterior  $p(\theta|\text{Data})$  is also a Gaussian density

# Nonlinear functions ?

- Consider

$$f_{\theta}(x) = \sum_{l=1}^K \theta_l \phi_l(x)$$

with *nonlinear functions*  $\phi_l(x)$  of  $x$ .

- This is *linear in the parameters*  $\theta_l$ !



# Nonlinear functions ?

- Consider

$$f_{\theta}(x) = \sum_{l=1}^K \theta_l \phi_l(x)$$

with *nonlinear functions*  $\phi_l(x)$  of  $x$ .

- This is *linear in the parameters*  $\theta_l$ !
- Power series

$$f_{\theta}(x) = \sum_{l=1}^K \theta_l x^l$$

- or Fourier series

$$f_{\theta}(x) = \sum_{l=1}^K \{ \theta_l \sin(2\pi lx) + \theta'_l \cos(2\pi lx) \}$$

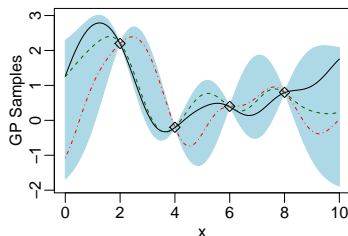
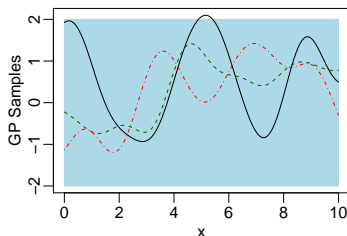
- with Gaussian prior  $p(\theta)$  the posterior  $p(\theta|\text{Data})$  is still Gaussian.

# The limit $K \rightarrow \infty$ ?

- Transition to random functions.
- Infinitely many parameters  $\theta_l$  not easy to handle !

# The limit $K \rightarrow \infty$ ?

- Transition to random functions.
- Infinitely many parameters  $\theta_l$  not easy to handle !
- Nonparametric:  $f(\cdot) \sim \mathcal{GP}(0, K)$



# Gaussian Process (GP) priors over functions

- Family of random variables  $f(x), x \in T$ . For any finite collection  $\{f(x_1), f(x_2), \dots, f(x_n)\}$  the joint distribution is Gaussian.
- For  $E[f(x)] = 0$  for all  $x$ , the process is characterised by covariance **kernel**

$$K(x, x') = E[f(x)f(x')]$$

# Gaussian Process (GP) priors over functions

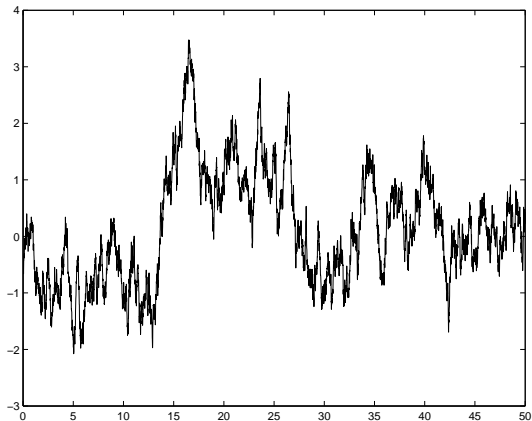
- Family of random variables  $f(x), x \in T$ . For any finite collection  $\{f(x_1), f(x_2), \dots, f(x_n)\}$  the joint distribution is Gaussian.
- For  $E[f(x)] = 0$  for all  $x$ , the process is characterised by covariance **kernel**

$$K(x, x') = E[f(x)f(x')]$$

- Kernel functions encode prior beliefs (or knowledge) about smoothness or 'wiggleness' of functions  $f(x)$ .
- Example: Stationary kernels  $K(x - x')$  constructed from  $K(x) = \int_{-\infty}^{\infty} e^{i\omega x} \hat{K}(\omega) d\omega$  with non-negative  $\hat{K}(\omega) > 0$ .

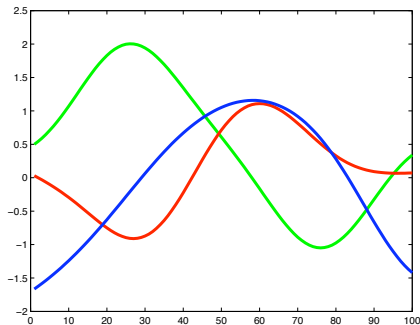
# Samples from the GP prior

Samples from a GP with  $K(x, x') = e^{-|x-x'|}$  (Ornstein–Uhlenbeck process).

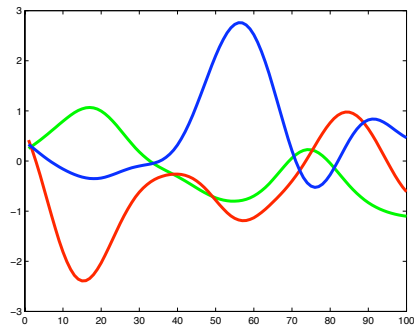


# Samples from the GP prior

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



$K(x, x') = e^{-10(x-x')^2}$



# GPs on higher dimensional spaces

Kernels can be constructed for  $d$  dimensional inputs  $\mathbf{x} = (x(1), x(2), \dots, x(d))$  where  $x(i)$  is the  $i$ -th coordinate of  $\mathbf{x}$ . A popular choice is the radial basis function (RBF) kernel.

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

allowing for different *hyperparameters* (lengthscales)  $\lambda_k$ .



# How to make predictions ?

Let  $\mathbf{y} = (y_1, \dots, y_n)$ ,  $\mathbf{z} = (f(x_1), \dots, f(x_n))$  and  $v = f(x)$ . Then

$$p(v|\mathbf{y}) = \int p(v|\mathbf{z})p(\mathbf{z}|\mathbf{y})d\mathbf{z}$$

- In general, we have

$$p(\mathbf{z}|\mathbf{y}) \propto \prod_{i=1}^n p(y_i|z_i) \exp \left[ -\frac{1}{2} \mathbf{z}^\top \mathbf{K}^{-1} \mathbf{z} \right]$$

with the kernel matrix  $K_{ij} \doteq K(x_i, x_j)$ .

- For the Gaussian noise model,  $y_i = f(x_i) + \nu_i$ , and  $\nu_i = \mathcal{N}(0, \sigma^2)$  we get the Gaussian posterior

$$p(\mathbf{z}|\mathbf{y}) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}, \mathbf{S})$$

with  $\mathbf{S} = (\mathbf{K}^{-1} + \frac{1}{\sigma^2} \mathbf{I})^{-1}$  and  $\boldsymbol{\mu} = \frac{1}{\sigma^2} \mathbf{S} \mathbf{y}$

## Conditioning

Let

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

with the information matrix  $\Omega = \begin{pmatrix} \Omega_{vv} & \Omega_{vz} \\ \Omega_{zv} & \Omega_{zz} \end{pmatrix}$  and  $\xi = (\xi_v \ \xi_z)^\top$ .

The conditional density is

$$p(v|z) \propto \exp \left[ -\frac{1}{2} v^\top \Omega_{vv} v + v^\top (\xi_v - \Omega_{vy} z) \right]$$

For the GP model we have  $\xi = 0$ . This yields  $E(v|z) = -(\Omega_{vv})^{-1} \Omega_{vz} z$  and  $\text{Cov}(v|z) = (\Omega_{vv})^{-1}$

# 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

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

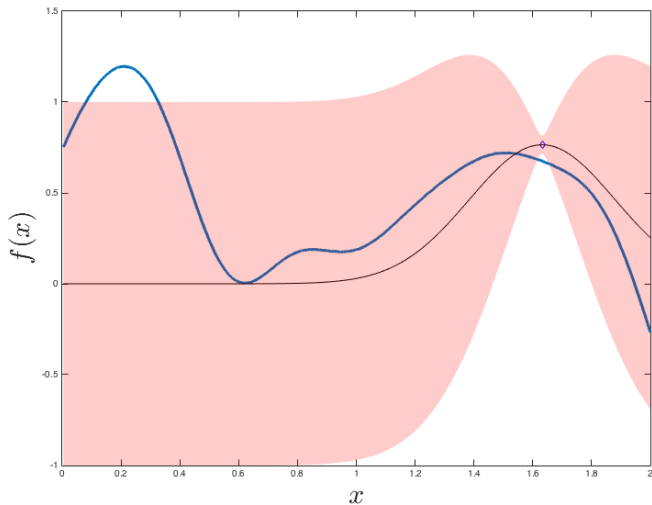
## Details for regression with Gaussian noise:

- $p(v|\mathbf{z}) = \mathcal{N}(v|m, s)$  with  $m \doteq E[v|\mathbf{z}] = \mathbf{k}_x^\top \mathbf{K}^{-1} \mathbf{z}$  and  $s \doteq \text{VAR}[v|\mathbf{z}] = K(x, x) - \mathbf{k}_x^\top \mathbf{K}^{-1} \mathbf{k}_x$  where  $\mathbf{k}_x \doteq (K(x, x_1), \dots, K(x, x_n))^\top$ .
- Mean Prediction:

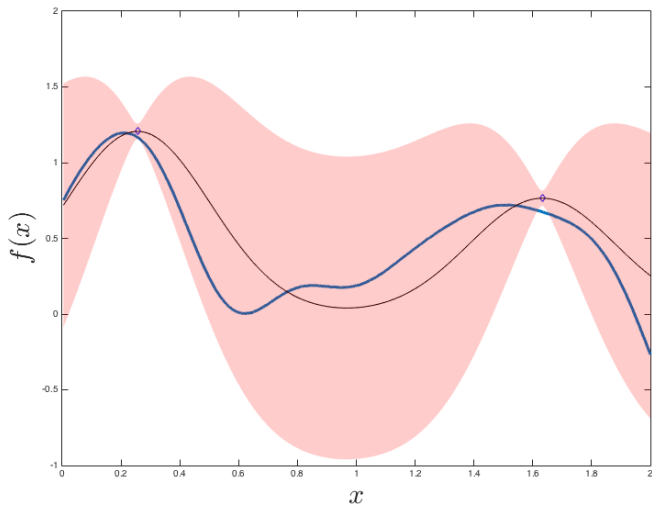
$$\begin{aligned} E[v|\mathbf{y}] &= \int E[v|\mathbf{z}] p(\mathbf{z}|\mathbf{y}) d\mathbf{z} = \mathbf{k}_x^\top \mathbf{K}^{-1} \int \mathbf{z} p(\mathbf{z}|\mathbf{y}) d\mathbf{z} = \\ \mathbf{k}_x^\top \mathbf{K}^{-1} E[\mathbf{z}|\mathbf{y}] &= \frac{1}{\sigma^2} \mathbf{k}_x^\top \mathbf{K}^{-1} \mathbf{y} = \frac{1}{\sigma^2} \mathbf{k}_x^\top \mathbf{K}^{-1} \left( \mathbf{K}^{-1} + \frac{1}{\sigma^2} \mathbf{I} \right)^{-1} \mathbf{y} \\ &= \mathbf{k}_x^\top (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{y} \end{aligned}$$

- Uncertainty  $\text{VAR}[v|\mathbf{y}] = K(x, x) - \mathbf{k}_x^\top (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{k}_x$  independent of  $\mathbf{y}$ .

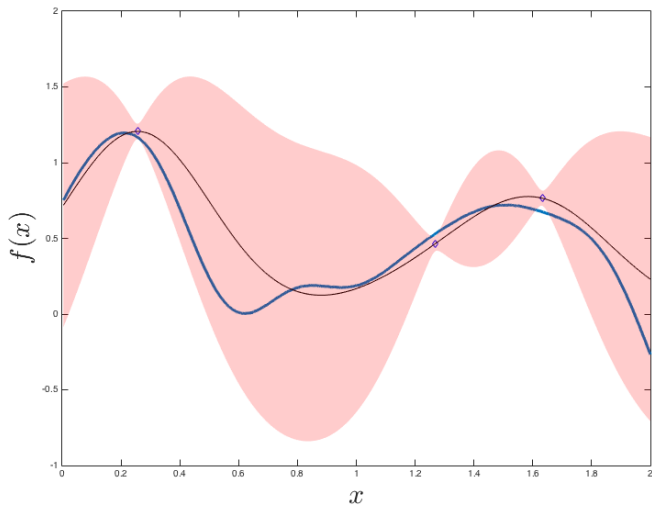
# 1 observation



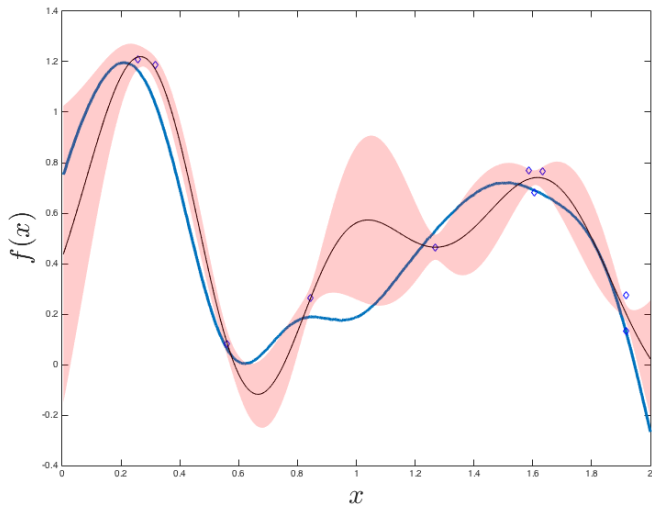
## 2 observations



## 3 observations

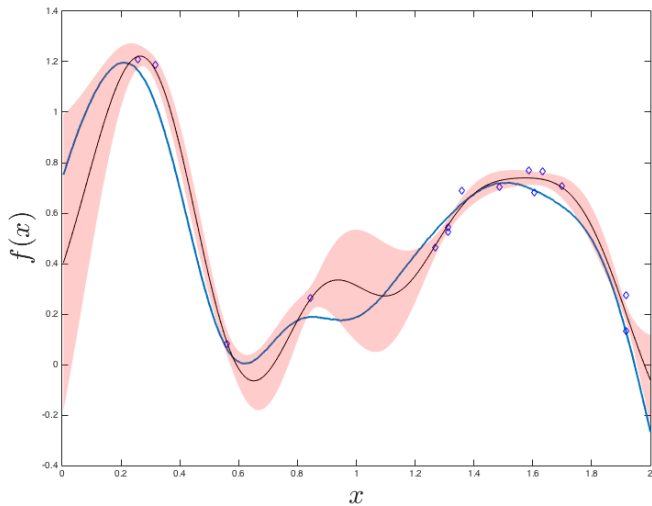


## 10 observations

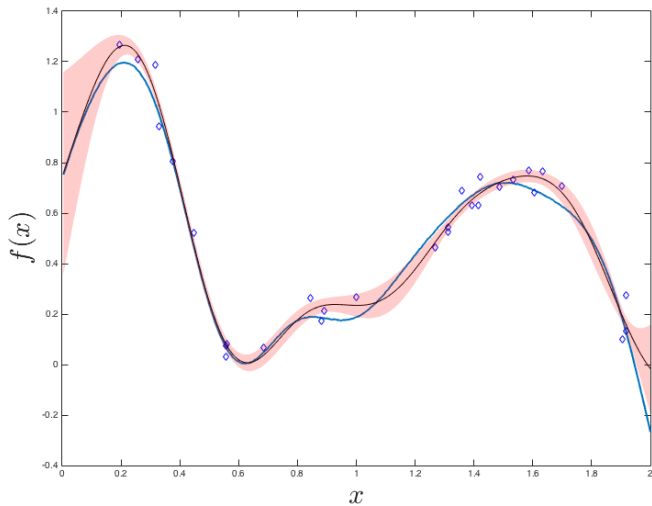




## 15 observations



## 30 observations

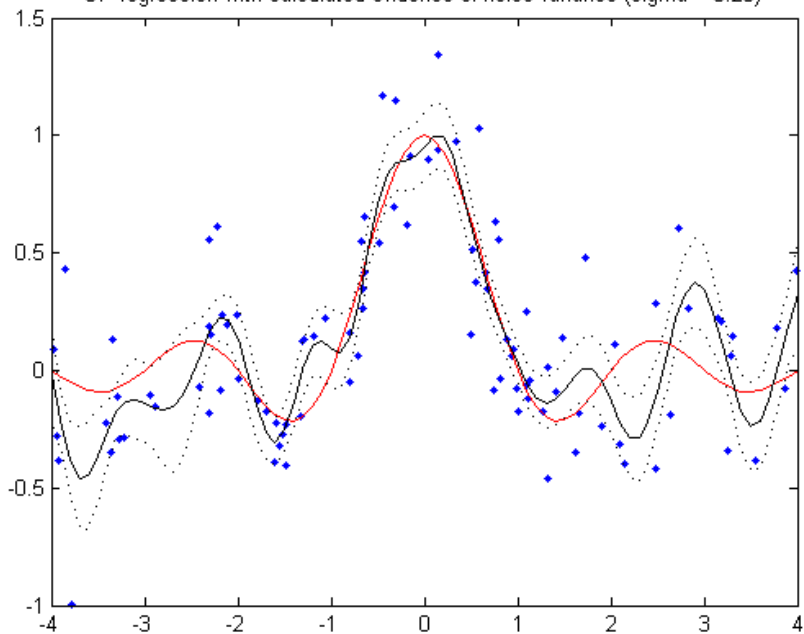


# Model selection using the evidence

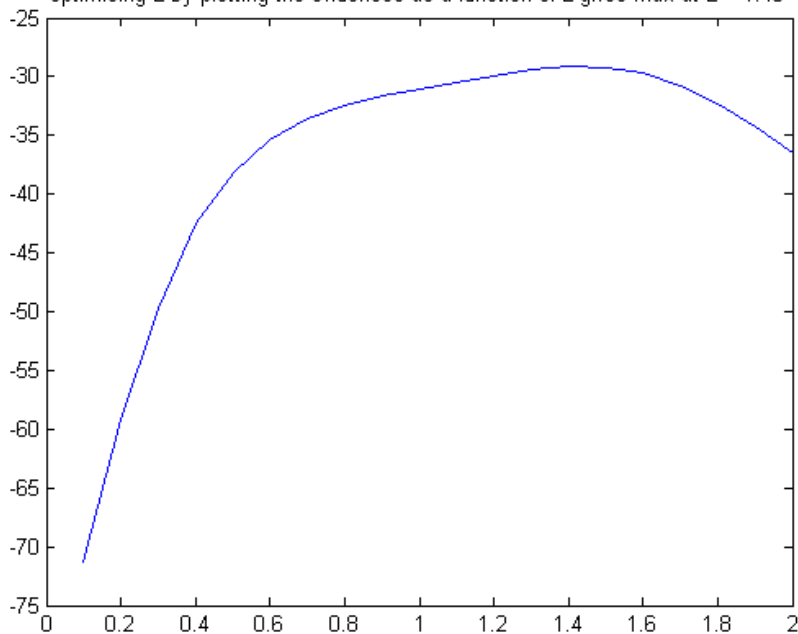
Sensible values for **kernel hyperparameters** and noise  $\sigma^2$  can be obtained by numerically maximising the evidence (Maximum Likelihood II)

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

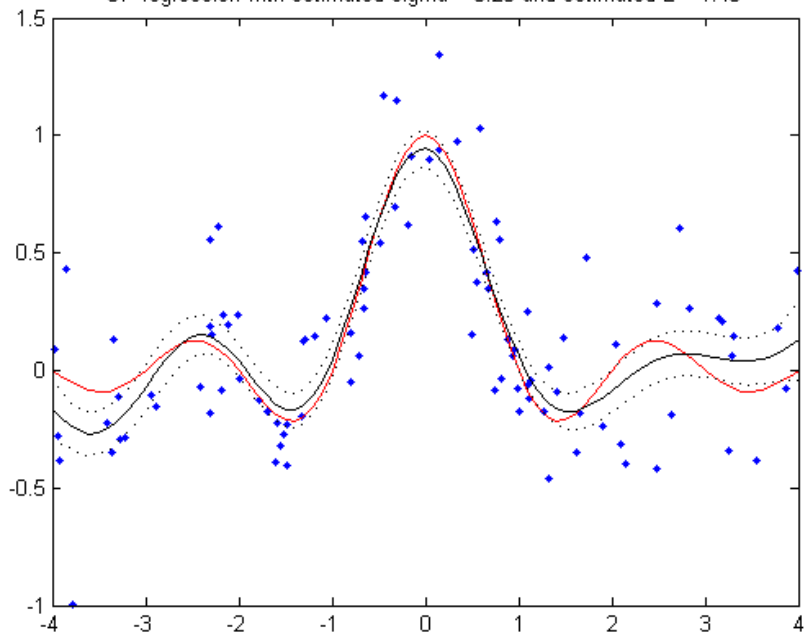
GP regression with calculated evidence of noise variance ( $\sigma = 0.26$ )



optimising  $L$  by plotting the evidences as a function of  $L$  gives max at  $L = 1.40$



GP regression with estimated sigma = 0.26 and estimated L = 1.40



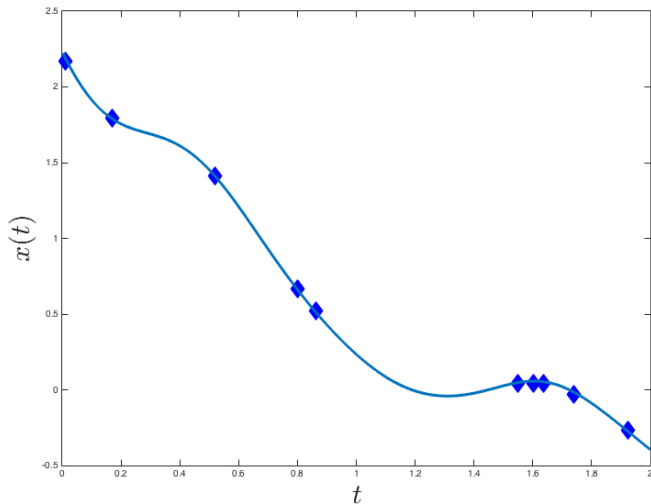
# Linear ordinary differential equations

- Linear operations on GPs leads to GPs !
- A dynamical model

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

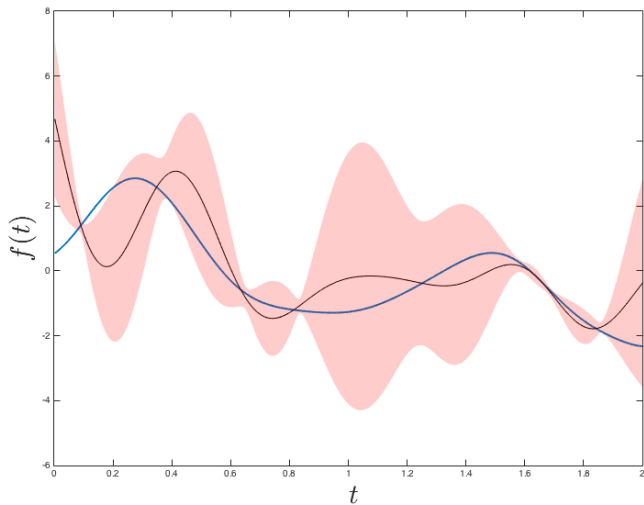
- $f(t)$  is an unknown function to be estimated.
- Use a GP prior over  $f(\cdot)$ .

# With 10 observations ...

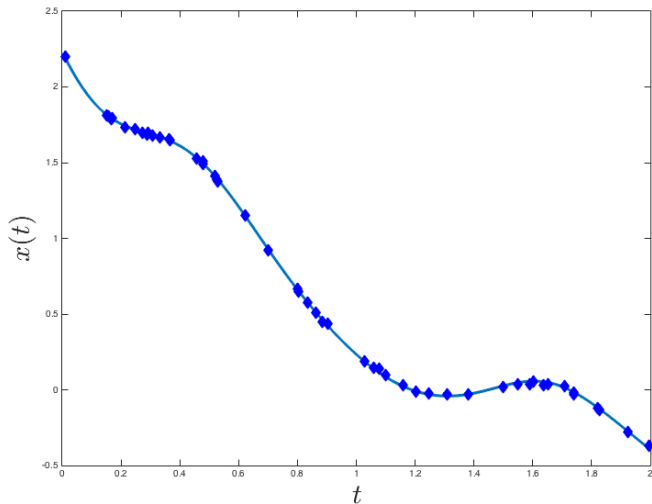




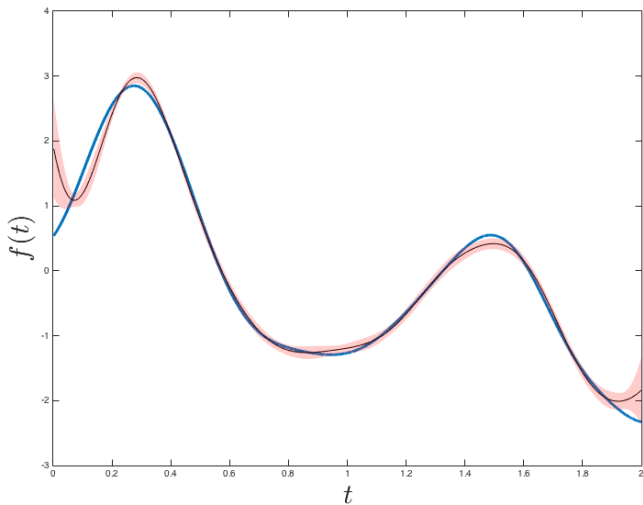
# With 10 observations ...



# With 50 observations ...



## With 50 observations ...



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.

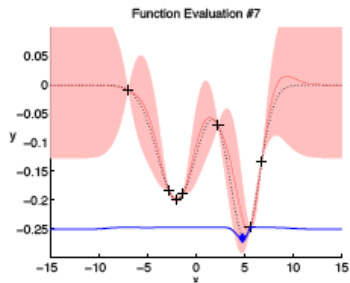
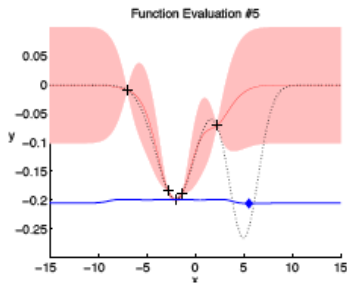
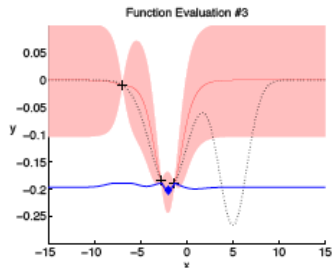
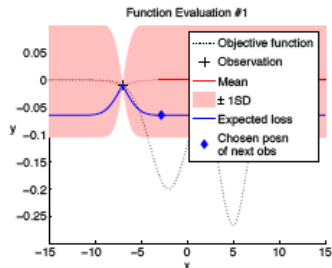
**Sensitivity analysis:** Changes of outputs under small input changes.

**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.

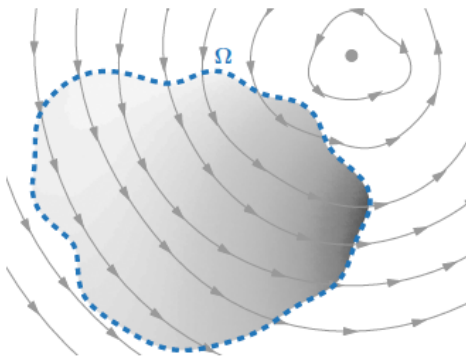


# Modeling and interpolation of the ambient magnetic field

A. Solin et al arXiv:1509.04634v1

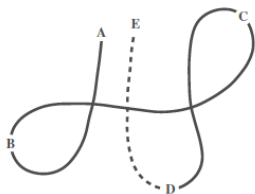
- Goal: Use maps of magnetic fields in buildings for localisation.
- Magnetic fields are vectors  $\mathbf{H}(\mathbf{x})$  which fulfil  $\nabla \times \mathbf{H}(\mathbf{x})$ . Thus we have the scalar field representation  $\mathbf{H}(\mathbf{x}) = -\nabla\phi(\mathbf{x})$ .
- Observation model

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

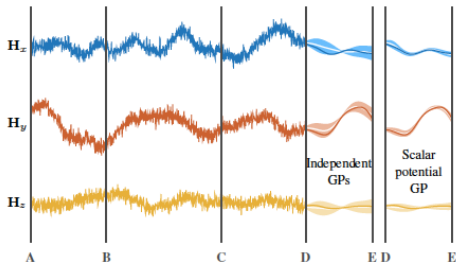


**Figure 2.** Illustration of a vector field with non-zero curl. The vortex point makes it non-curl-free as the vector field curls around it. However, the subset  $\Omega$  excludes the vortex point and the vector field is curl-free in this region. To this region a scalar potential  $\varphi$  can be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



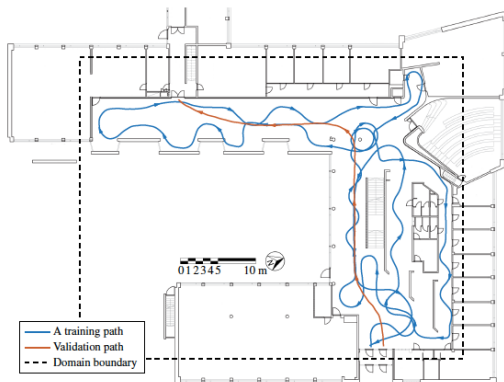


(a) The route

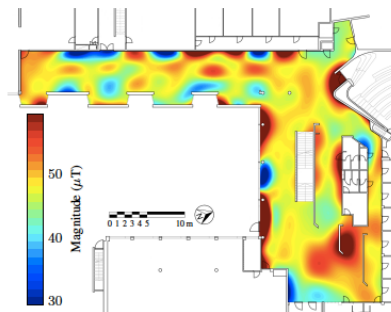


(b) The data and the GP prediction for D-E

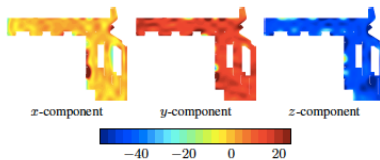
**Figure 3.** A simulated example of the interpolation problem. (a) Training data has been collected along the route A-D, but the magnetic field between D-E is unknown. (b) The noisy observations of the magnetic field between A-D, and GP predictions with 95% credibility intervals. Both the independent 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 ( $\|f\|$ ) interpolated by the scalar potential GP model. (b) The separate field components of the estimate.

# Computational tools III: Variational approximation

- Observations  $\mathbf{y} \equiv (y_1, \dots, y_K)$  ("**data**")
- Latent, unobserved variables  $\mathbf{z} \equiv (z_1, \dots, z_N)$
- Likelihood  $p(\mathbf{y}|\mathbf{z})$  **forward model**
- Prior distribution  $p(\mathbf{z})$

# Computational tools III: Variational approximation

- Observations  $\mathbf{y} \equiv (y_1, \dots, y_K)$  ("data")
- Latent, unobserved variables  $\mathbf{z} \equiv (z_1, \dots, z_N)$
- Likelihood  $p(\mathbf{y}|\mathbf{z})$  **forward model**
- Prior distribution  $p(\mathbf{z})$
- **Inverse problem:** Make predictions on  $x$  given observations using **Bayes rule:**

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

# Computational tools III: Variational approximation

- Observations  $\mathbf{y} \equiv (y_1, \dots, y_K)$  ("data")
- Latent, unobserved variables  $\mathbf{z} \equiv (z_1, \dots, z_N)$
- Likelihood  $p(\mathbf{y}|\mathbf{z})$  **forward model**
- Prior distribution  $p(\mathbf{z})$
- **Inverse problem:** Make predictions on  $x$  given observations using **Bayes rule:**

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

- Easy ?

# Not quite ...

- Often easy to write down the posterior of **all** hidden variables

$$p(z_1, \dots, z_N | \text{data}) = \frac{p(\text{data} | z_1, \dots, z_N) p(z_1, \dots, z_N)}{p(\text{data})}$$

# Not quite ...

- Often easy to write down the posterior of **all** hidden variables

$$p(z_1, \dots, z_N | \text{data}) = \frac{p(\text{data} | z_1, \dots, z_N) p(z_1, \dots, z_N)}{p(\text{data})}$$

- But what we really need are **marginal distributions** eg.

$$p(z_i | \text{data}) = \int dz_1 \dots dz_{i-1} dz_{i+1} \dots dz_N \frac{p(\text{data} | z_1, \dots, z_N) p(z_1, \dots, z_N)}{p(\text{data})}$$

- and

$$p(\text{data}) = \int dz_1 \dots dz_N p(\text{data} | z_1, \dots, z_N) p(z_1, \dots, z_N)$$



## Reminder: Gaussian processes

- The posterior density of the unknown function  $f(x)$  at an input  $x$  is

$$p(f(x)|\mathbf{y}) = \int p(f(x)|z_1, \dots, z_n) p(z_1, \dots, z_n|\mathbf{y}) dz_1, \dots, dz_n$$

- For a Gaussian noise model  $p(y_i|z_i)$  the integrals can be performed analytically.

# Non-Gaussian observation models

GP appears as a latent function in more complicated models:

- $y_i = f(x) + \text{non-Gaussian noise}$
- Binary classification  $y_i \in \{0, 1\}$  with  $p(y = 1|f(x)) = \text{sigmoid}[f(x)]$ .
- ...

**Approximations are necessary !**

# Simplest type of dependencies

$$p(z_1 \dots, z_N | \text{data}) = \prod_i \psi_i(z_i) \prod_{i < j} \psi_{ij}(z_i, z_j)$$

# Simplest type of dependencies

$$p(z_1 \dots, z_N | \text{data}) = \prod_i \psi_i(z_i) \prod_{i < j} \psi_{ij}(z_i, z_j)$$

even simpler

$$p(z_1 \dots, z_N | \text{data}) = \prod_i \psi_i(z_i) \exp \left[ \sum_{i < j} A_{ij} z_i z_j \right]$$

compare to

$$\prod_{i=1}^n p(y_i | z_i) \exp \left[ -\frac{1}{2} \mathbf{z}^\top \mathbf{K}^{-1} \mathbf{z} \right] = \prod_{i=1}^n p(y_i | z_i) \exp \left[ -\frac{1}{2} \sum_{i,j} z_i (\mathbf{K}^{-1})_{ij} z_j \right]$$

# The KL divergence

- For two distributions  $q(\mathbf{z})$  and  $p(\mathbf{z})$  one can show (Jensen's inequality) that the **Kullback–Leibler divergence**

$$D(q\|p) = E_q \left[ \ln \frac{q(\mathbf{z})}{p(\mathbf{z})} \right] = \int q(\mathbf{z}) \ln \frac{q(\mathbf{z})}{p(\mathbf{z})} d\mathbf{z} \geq 0$$

# The KL divergence

- For two distributions  $q(\mathbf{z})$  and  $p(\mathbf{z})$  one can show (Jensen's inequality) that the **Kullback–Leibler divergence**

$$D(q\|p) = E_q \left[ \ln \frac{q(\mathbf{z})}{p(\mathbf{z})} \right] = \int q(\mathbf{z}) \ln \frac{q(\mathbf{z})}{p(\mathbf{z})} d\mathbf{z} \geq 0$$

- Equality = 0 if and only if  $p = q$  almost everywhere.

- The posterior is obtained as

$$p(\mathbf{z}|\mathbf{y}) = \arg \min_q \left\{ E_q \left[ \ln \frac{q(\mathbf{z})}{p(\mathbf{z}, \mathbf{y})} \right] \right\}$$

- The minimum is

$$\min_q \left\{ E_q \left[ \ln \frac{q(\mathbf{z})}{p(\mathbf{z}, \mathbf{y})} \right] \right\} = -\ln p(\mathbf{y})$$

# The Mean Field Method: 2 Variables

- Approximate

$$p(z_1, z_2 | \mathbf{y})$$

by distributions from simpler family  $\mathcal{F}$  of factorising distributions

$$q(z_1, z_2) \doteq q_1(z_1)q_2(z_2)$$

- Find the best  $q$  by solving

$$q^{opt}(z_1, z_2) = \arg \min_{q \in \mathcal{F}} \left\{ E_q \left[ \ln \frac{q(z_1, z_2)}{p(z_1, z_2, \mathbf{y})} \right] \right\}$$

- The solution is

$$q_1^{opt}(z_1) \propto \exp \left\{ E_{q_2^{opt}} [\ln p(z_1, z_2 | \mathbf{y})] \right\}$$
$$q_2^{opt}(z_2) \propto \exp \left\{ E_{q_1^{opt}} [\ln p(z_1, z_2 | \mathbf{y})] \right\}$$



# The Mean Field Method: Many variables

- Approximate  $p(\mathbf{z}|\mathbf{y})$  by the best factorising distribution  
 $q(\mathbf{z}) = \prod_{i=1}^N q_i(z_i)$
- The optimal solution is:

$$q_i^{opt}(z_i) = \frac{1}{Z_i} \exp \{ E_{\setminus i} [\ln p(\mathbf{z}, \mathbf{y})] \}$$

with  $E_{\setminus i}[\dots]$  the average over all variables except  $z_i$ .

# Work this out for the ...

- ... latent Gaussian variable model

$$p(z_1 \dots, z_N | \text{data}) = \prod_i \psi_i(z_i) \exp \left[ \sum_{i < j} A_{ij} z_i z_j \right]$$

- Optimal solution

$$q_i(z) \propto \psi_i(z) \exp \left[ z \sum_{j \neq i} A_{ij} m_j \right]$$

- with

$$m_j = E_q[Z_j] = \frac{\int \psi_i(z) \exp \left[ z \sum_{k \neq j} A_{jk} m_k \right] z \, dz}{\int \psi_i(z) \exp \left[ z \sum_{k \neq j} A_{jk} m_k \right] \, dz}$$

# A simple classifier

- No noise, binary class labels  $y_i = \pm 1$ :

$$p(y_i|z_i) = I_{y_i z_i > 0}$$

- We have

$$\psi_i(z) = p(y_i|z_i) \exp \left[ -\frac{1}{2} (\mathbf{K}^{-1})_{ii} z_i^2 \right]$$

- and  $A_{ij} = (\mathbf{K}^{-1})_{ij}$

# Gaussian variational approximation

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

The variational objective is

$$\mathcal{F}[q] = E_q \left[ \ln \frac{q(\mathbf{z})}{p(\mathbf{y}, \mathbf{z})} \right] = -\frac{N}{2} \log 2\pi - \frac{1}{2} \log |\boldsymbol{\Sigma}| - \frac{N}{2} - E_q[\log p(\mathbf{y}, \mathbf{z})]$$

# Gaussian variational approximation

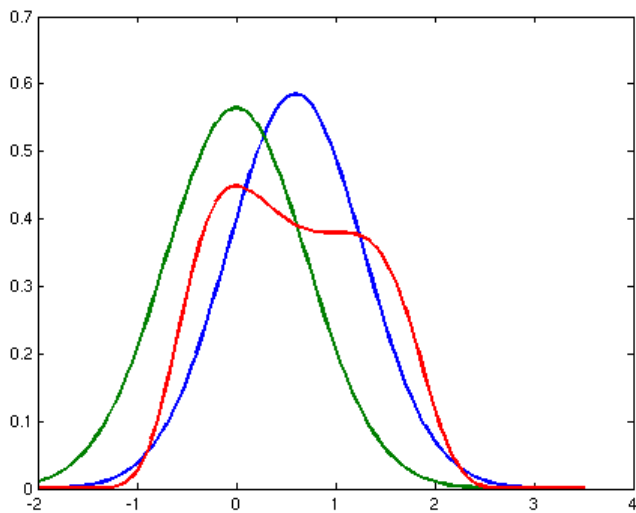
$$q(\mathbf{z}) = (2\pi)^{-N/2} |\boldsymbol{\Sigma}|^{-1/2} \exp \left( -\frac{1}{2} (\mathbf{z} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{z} - \boldsymbol{\mu}) \right).$$

The variational objective is

$$\mathcal{F}[q] = E_q \left[ \ln \frac{q(\mathbf{z})}{p(\mathbf{y}, \mathbf{z})} \right] = -\frac{N}{2} \log 2\pi - \frac{1}{2} \log |\boldsymbol{\Sigma}| - \frac{N}{2} - E_q[\log p(\mathbf{y}, \mathbf{z})]$$

Taking derivatives w.r.t. variational parameters

$$\begin{aligned} 0 &= E_q [\nabla_{\mathbf{z}} \log p(\mathbf{y}, \mathbf{z})] \\ (\boldsymbol{\Sigma}^{-1})_{ij} &= -E_q \left[ \frac{\partial^2 \log p(\mathbf{y}, \mathbf{z})}{\partial z_i \partial z_j} \right] \end{aligned}$$



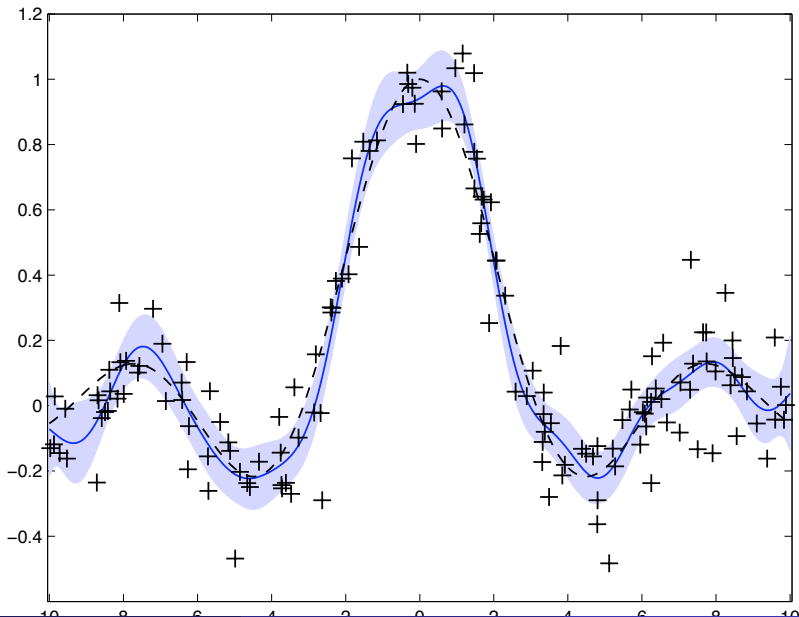
$$p(\mathbf{z}, \mathbf{y}) = \frac{1}{Z_0} \exp \left( - \sum_n V_n(y_n, z_n) - \frac{1}{2} \mathbf{z}^T \mathbf{K}^{-1} \mathbf{z} \right),$$

Covariance

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

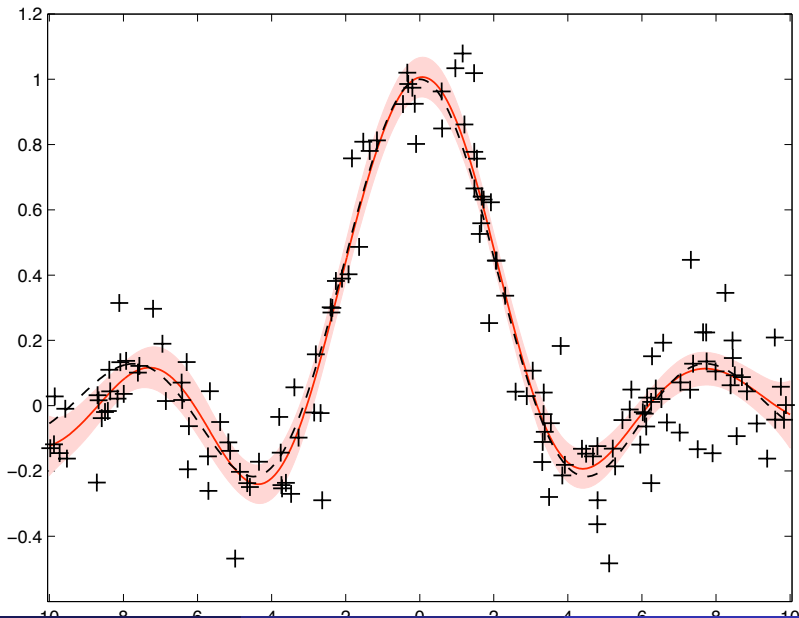
is parametrised by  $N$  elements!

## Function with Cauchy noise added (GP with Gaussian likelihood)





# Function with Cauchy noise added (Var - Approx. with Cauchy likelihood)



# Variational Optimal Sparsity for GPs

L. Csato (2002), M. Titsias (2009)

- Assume a split of  $\mathbf{z} = (s, B)$  into small set  $s$  and rest  $B$ .
- True joint probability

$$p(s, B, \mathbf{y}) = p_0(s, B)L(s, B)$$

- Approximate posterior with sparse likelihood

$$q(s, B) = p_0(s, B)\hat{L}(s)$$

- Minimise

$$\mathcal{F}[q] = E_q \left[ \ln \frac{p_0(s, B)\hat{L}(s)}{p_0(s, B)L(s, B)} \right]$$

# Variational Optimal Sparsity for GPs

L. Csato (2002), M. Titsias (2009)

- Assume a split of  $\mathbf{z} = (s, B)$  into small set  $s$  and rest  $B$ .
- True joint probability

$$p(s, B, \mathbf{y}) = p_0(s, B)L(s, B)$$

- Approximate posterior with sparse likelihood

$$q(s, B) = p_0(s, B)\hat{L}(s)$$

- Minimise

$$\mathcal{F}[q] = E_q \left[ \ln \frac{p_0(s, B)\hat{L}(s)}{p_0(s, B)L(s, B)} \right]$$

- The optimal likelihood is

$$\hat{L}(s) \propto \exp \left[ \int \ln L(B, s) p(B|s) dB \right]$$

Consider

$$\log L(\mathbf{z}) \propto \mathbf{a}^T \mathbf{z}^T - \frac{1}{2} \mathbf{z}^T \mathbf{A} \mathbf{z}$$

- $\hat{L}(s)$  is obtained from  $L$  by replacing  $\mathbf{z} \rightarrow E_0[\mathbf{z}|\mathbf{z}_s]$  where

$$\mathbf{z}_s = \{f(x)\}_{x \in \text{sparse set of inputs}}$$

Consider

$$\log L(\mathbf{z}) \propto \mathbf{a}^T \mathbf{z} - \frac{1}{2} \mathbf{z}^T \mathbf{A} \mathbf{z}$$

- $\hat{L}(s)$  is obtained from  $L$  by replacing  $\mathbf{z} \rightarrow E_0[\mathbf{z}|\mathbf{z}_s]$  where

$$\mathbf{z}_s = \{f(x)\}_{x \in \text{sparse set of inputs}}$$

- An explicit calculation shows that

$$E_0[\mathbf{z}|\mathbf{z}_s] = \mathbf{K}_{Bs} \mathbf{K}_{ss}^{-1} \mathbf{z}_s$$

- Variational approach is based on formulation of inference as optimisation problem
- For (simple) GP models, factorising (MF) and Gaussian approximations reduce integrations to one-dimensional integrals
- For MF approximation, the optimisation can be performed by coordinate descent.
- What about more complex models ?

# Black box approaches

(see e.g. Ranganath et al, 2014)

- Main idea: Perform expectations

$$\mathcal{F} = E_q \left[ \ln \frac{q_\phi(\mathbf{z})}{p(\mathbf{z}, \mathbf{y})} \right]$$

by **sampling** from parametric variational distribution  $q_\phi$ .  $\phi$  is set of parameters.

- Use gradient descent as algorithm

$$\phi_{t+1} = \phi_t - \gamma \nabla_\phi \mathcal{F}(\phi_t)$$

- Problem: Gradient is noisy (stochastic gradient descent).
- Represent gradient  $\nabla_{\phi} \left\{ E_q \left[ \ln \frac{q_{\phi}(\mathbf{z})}{p(\mathbf{z}, \mathbf{y})} \right] \right\}$  as an expectation (unbiased estimator)



- Problem: Gradient is noisy (stochastic gradient descent).
- Represent gradient  $\nabla_{\phi} \left\{ E_q \left[ \ln \frac{q_{\phi}(\mathbf{z})}{p(\mathbf{z}, \mathbf{y})} \right] \right\}$  as an expectation (unbiased estimator)
- Use *reparametrisation trick* e.g. if  $q_{\phi}(\mathbf{z})$  is Gaussian with parameters  $\phi = (\mu, \sigma)$ , we can set  $\mathbf{z} = \mu + \sigma \mathbf{u}$ , where  $\mathbf{u} \sim \mathcal{N}(0, 1)$

# Why Kullback–Leibler ?

- Try something else: Evaluate

$$-E_q \left[ \left( \frac{p(\mathbf{z}, \mathbf{y})}{q_\phi(\mathbf{z})} \right)^{1-\alpha} \right]$$

using MC sampling and minimise.

- Bias for estimating  $p(\mathbf{y})$  disappears for  $\alpha \rightarrow 0$ .

# Why Kullback–Leibler ?

- Try something else: Evaluate

$$-E_q \left[ \left( \frac{p(\mathbf{z}, \mathbf{y})}{q_\phi(\mathbf{z})} \right)^{1-\alpha} \right]$$

using MC sampling and minimise.

- Bias for estimating  $p(\mathbf{y})$  disappears for  $\alpha \rightarrow 0$ .
- The ratio has typically huge fluctuations for  $\alpha < 1$  ! KL is recovered in the limit  $\alpha \rightarrow 1$ .