Probabilistic & Unsupervised Learning

Bayesian Treatment of Probabilistic Models
and Gaussian Processes

Yee Whye Teh

`ywteh@gatsby.ucl.ac.uk`

Gatsby Computational Neuroscience Unit
University College London

Term 1, Autumn 2011

Learning Model Structure

How many clusters in the data?

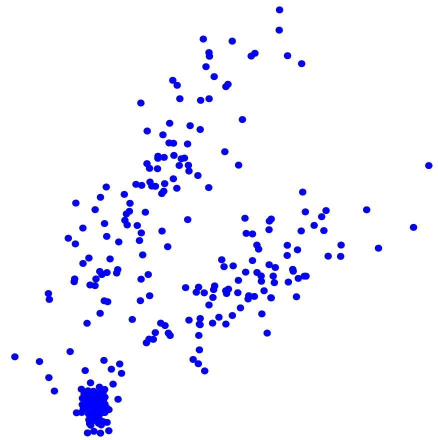
What is the intrinsic dimensionality of the data?

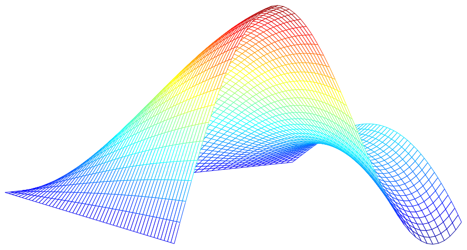
Is this input relevant to predicting that output?

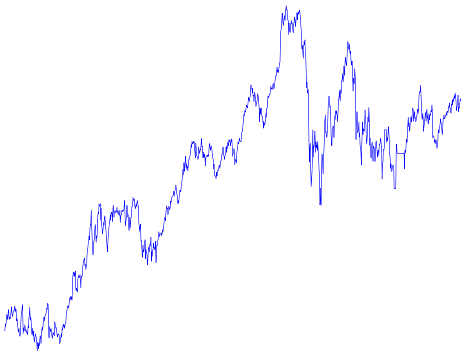
What is the order of a dynamical system?

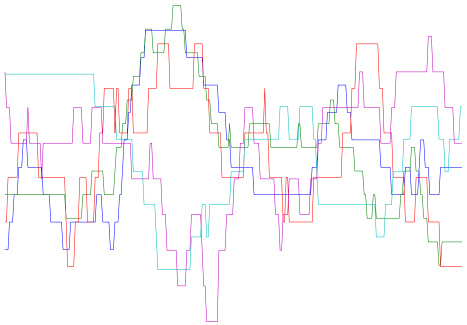
How many states in a hidden Markov model?

How many auditory sources in the input?



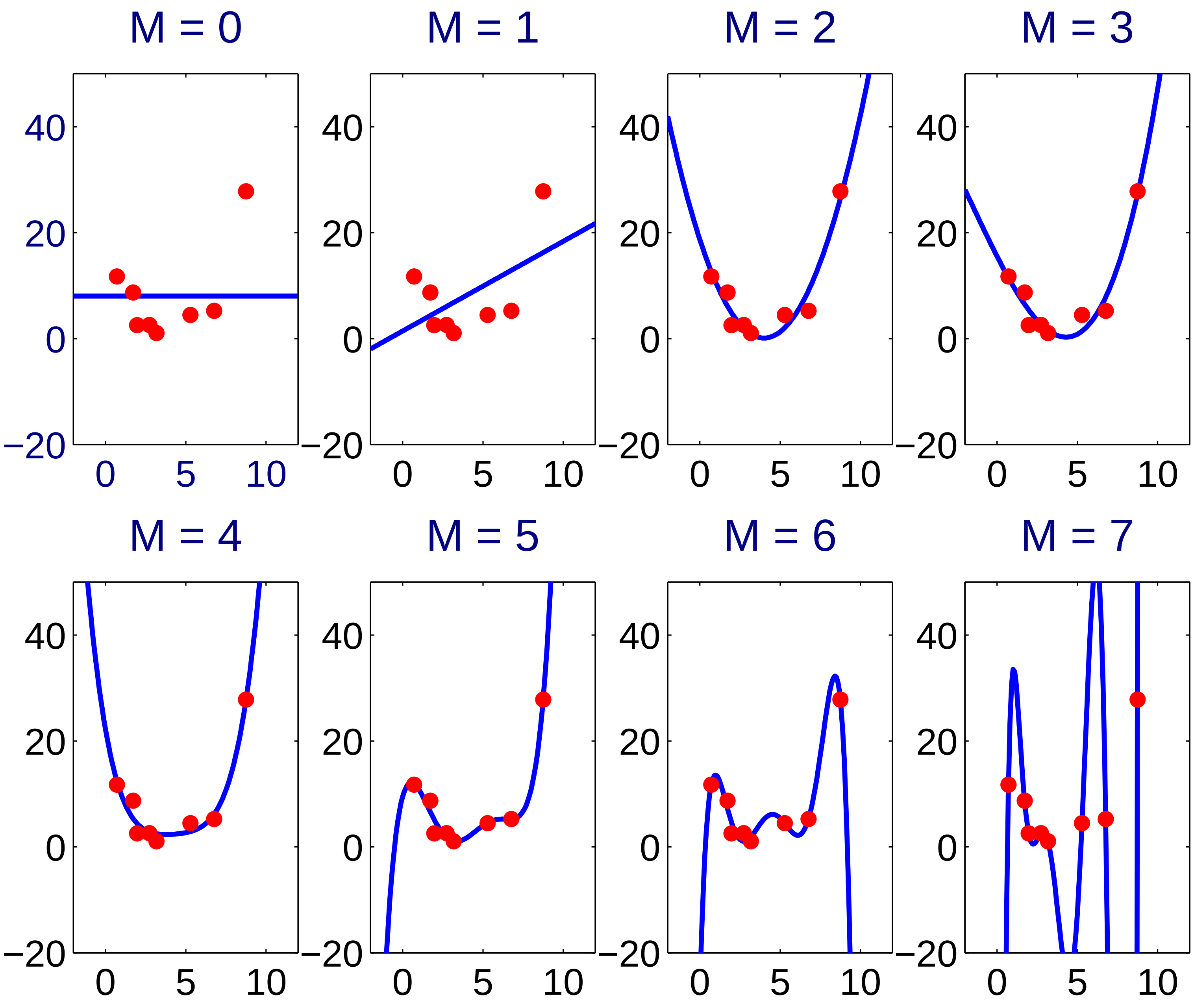






SVYDAAAQLTADVKKDLRDSWKVIGSDKKGNGVALMTTY

Model complexity and overfitting: a simple example



Learning Model Structure

Models labeled by m have parameters $\boldsymbol{\theta}_m$. Which model is correct?
ML (or MAP) has no good answer: $P(\mathcal{D}|\boldsymbol{\theta}_m^{\text{ML}})$ is always larger for more complex (nested) models.

Neyman-Pearson hypothesis testing

- For nested models. Starting with simplest model ($m = 1$), compare (e.g. by likelihood ratio test) null hypothesis m to alternative $m + 1$. Continue until $m + 1$ is rejected.
- Usually only valid asympotically in data number.
- Conservative (N-P hypothesis tests are asymmetric).

Likelihood validation

- Partition data into disjoint *training* and *validation* data sets $\mathcal{D} = \mathcal{D}_{\text{tr}} \cup \mathcal{D}_{\text{vld}}$. Choose model with greatest $P(\mathcal{D}_{\text{vld}}|\boldsymbol{\theta}_m^{\text{ML}})$, with $\boldsymbol{\theta}_m^{\text{ML}} = \text{argmax}\, P(\mathcal{D}_{\text{tr}}|\boldsymbol{\theta})$.
- Unbiased, but often high-variance.
- Cross-validation uses multiple partitions and averages likelihoods.

Bayesian model selection

- Choose most likely model: argmax $P(m|\mathcal{D})$.
- Principled (from a probabilistic viewpoint), but dependent on assumed priors etc.
- Can use posterior probabilities to weight models for combined predictions (no need to select at all).

Bayesian Treatment of Probabilistic Models: Terminology

A model class m is a set of distributions parameterised by $\boldsymbol{\theta}_m$, e.g. the set of all possible mixtures of m Gaussians.

We have a prior over the parameters $P(\boldsymbol{\theta}_m|m)$, and a likelihood of data given parameters (this might involve integrating out latent variables) $P(\mathcal{D}|\boldsymbol{\theta}_m, m)$.
The posterior distribution over parameters is

$$P(\boldsymbol{\theta}_m|\mathcal{D}, m) = \frac{P(\mathcal{D}|\boldsymbol{\theta}_m, m)P(\boldsymbol{\theta}_m|m)}{P(\mathcal{D}|m)}.$$

The marginal probability of the data under model class m is:

$$P(\mathcal{D}|m) = \int_{\Theta_m} P(\mathcal{D}|\boldsymbol{\theta}_m, m)P(\boldsymbol{\theta}_m|m)\, d\boldsymbol{\theta}_m.$$

This is also known as the Bayesian evidence for model m.

The ratio of two marginal probabilities (or sometimes its log) is known as the Bayes factor:

$$\frac{P(\mathcal{D}|m)}{P(\mathcal{D}|m')}$$

The Bayesian Occam's Razor

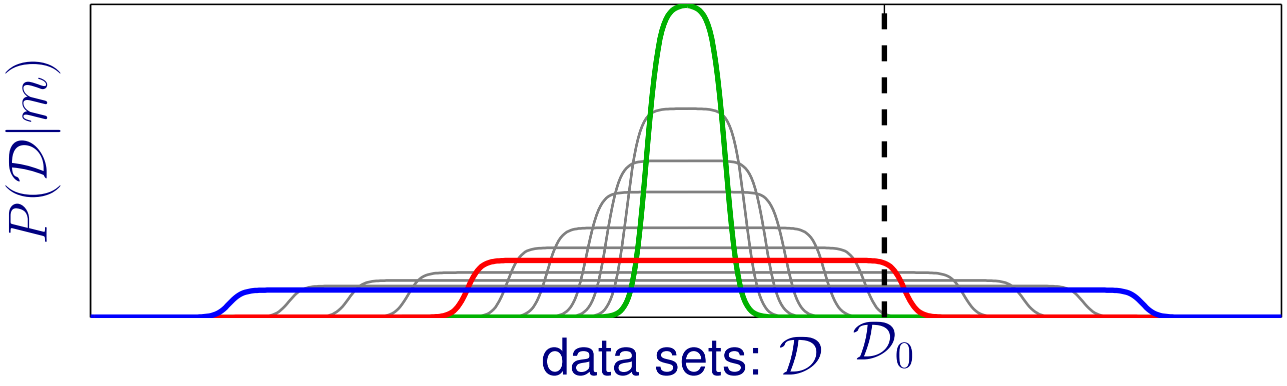
The Occam's Razor principle is, roughly speaking, that one should prefer simpler explanations than more complex explanations. Bayesian inference formalises and *automatically* implements the Occam's Razor principle.

Compare model classes m using their posterior probability given the data:

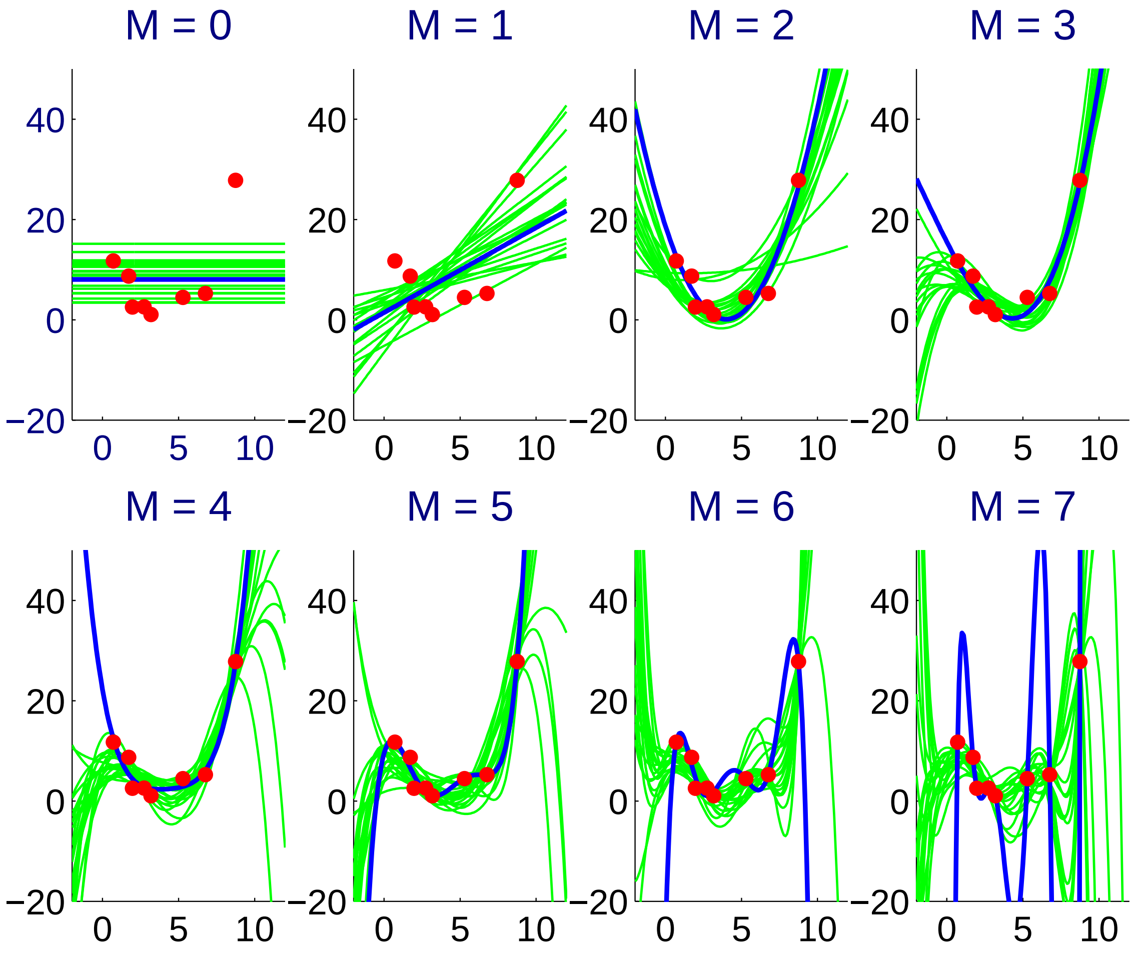
$$P(m|\mathcal{D}) = \frac{P(\mathcal{D}|m)P(m)}{P(\mathcal{D})}, \qquad P(\mathcal{D}|m) = \int_{\Theta_m} P(\mathcal{D}|\boldsymbol{\theta}_m, m)P(\boldsymbol{\theta}_m|m)\, d\boldsymbol{\theta}_m$$

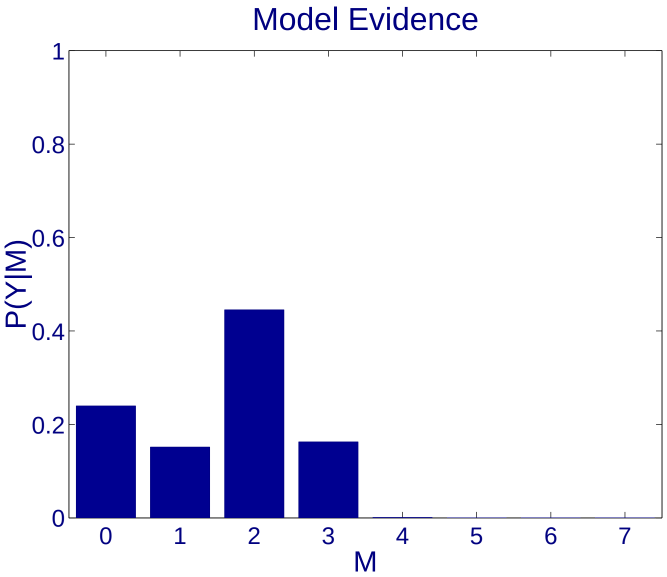
Interpretation of $P(\mathcal{D}|m)$: The probability that *randomly selected* parameter values from the model class would generate data set $\mathcal{D}$.

Model classes that are too simple are unlikely to generate the data set.
Model classes that are too complex can generate many possible data sets, so again, they are unlikely to generate that particular data set at random.



Bayesian Model Comparison: Occam’s Razor at Work





e.g. for quadratic (M=2): $y = a_0 + a_1 x + a_2 x^2 + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \tau)$ and $\boldsymbol{\theta}_2 = [a_0 \; a_1 \; a_2 \; \tau]$

Conjugate-Exponential Families

Can we compute $P(\mathcal{D}|m)$? . . . . . . Sometimes.

Suppose $P(\mathcal{D}|\boldsymbol{\theta}_m, m)$ is a member of the exponential family:

$$P(\mathcal{D}|\boldsymbol{\theta}_m, m) = \prod_{i=1}^{N} P(\mathbf{x}_i|\boldsymbol{\theta}_m, m) = \prod_{i=1}^{N} e^{\mathbf{s}(\mathbf{x}_i)^\top \boldsymbol{\theta}_m - A(\boldsymbol{\theta}_m)}.$$

If our prior on $\boldsymbol{\theta}_m$ is conjugate:

$$P(\boldsymbol{\theta}_m|m) = e^{\mathbf{s}_p^\top \boldsymbol{\theta}_m - n_p A(\boldsymbol{\theta}_m)} / Z(\mathbf{s}_p, n_p)$$

then the joint is in the same family:

$$P(\mathcal{D}, \boldsymbol{\theta}_m|m) = e^{\left(\sum_i \mathbf{s}(\mathbf{x}_i) + \mathbf{s}_p\right)^\top \boldsymbol{\theta}_m - (N+n_p)A(\boldsymbol{\theta}_m)} / Z(\mathbf{s}_p, p)$$

and so:

$$P(\mathcal{D}|m) = \int d\boldsymbol{\theta}_m \; P(\mathcal{D}, \boldsymbol{\theta}_m|m) = Z\left(\textstyle\sum_i \mathbf{s}(\mathbf{x}_i) + \mathbf{s}_p, N + n_p\right) / Z(\mathbf{s}_p, p)$$

But this is a special case. In general, we need to approximate . . .

Practical Bayesian approaches

- Laplace approximations:
  - Makes a Gaussian approximation about the maximum *a posteriori* parameter estimate.
- Bayesian Information Criterion (BIC)
  - an asymptotic approximation.
- Markov chain Monte Carlo methods (MCMC):
  - In the limit are guaranteed to converge, but:
  - There is often high variance in the estimated integrals.
  - Many samples required to ensure accuracy.
  - Sometimes hard to assess convergence.
- Variational approximations
  - Lower bound on the marginal probabilities.
  - Biased estimate.
  - Easy and fast, and often better than Laplace or BIC.

This list is not exhaustive. There are a number of other deterministic approximations, including those based on, e.g. Bethe approximations and expectation propagation.
We will discuss Laplace and BIC in this lecture, but the rest in second half of course.

Laplace Approximation

We want to find $P(\mathcal{D}|m) = \int P(\mathcal{D}, \boldsymbol{\theta}_m|m)\ d\boldsymbol{\theta}_m$.

As data size N grows (relative to number of parameter d), $\boldsymbol{\theta}_m$ becomes more constrained $\Rightarrow P(\mathcal{D}, \boldsymbol{\theta}_m|m) \propto P(\boldsymbol{\theta}_m|\mathcal{D}, m)$ becomes concentrated on MAP mode $\boldsymbol{\theta}_m^*$.

Idea: approximate **log**$P(\mathcal{D}, \boldsymbol{\theta}_m|m)$ to second-order around $\boldsymbol{\theta}^*$.

$$
\begin{aligned}
\int P(\mathcal{D}, \boldsymbol{\theta}_m|m) d\boldsymbol{\theta}_m &= \int e^{\log P(\mathcal{D}, \boldsymbol{\theta}_m|m)}\ d\boldsymbol{\theta}_m \\
&= \int e^{\log P(\mathcal{D}, \boldsymbol{\theta}_m^*|m) + \nabla \log P(\mathcal{D}, \boldsymbol{\theta}_m^*|m) \cdot (\boldsymbol{\theta}_m - \boldsymbol{\theta}_m^*) + \frac{1}{2}(\boldsymbol{\theta}_m - \boldsymbol{\theta}_m^*)^{\mathsf{T}} \nabla^2 \log P(\mathcal{D}, \boldsymbol{\theta}^*|m)(\boldsymbol{\theta}_m - \boldsymbol{\theta}_m^*)}\ d\boldsymbol{\theta}_m \\
&= \int P(\mathcal{D}, \boldsymbol{\theta}_m^*|m) e^{-\frac{1}{2}(\boldsymbol{\theta}_m - \boldsymbol{\theta}_m^*)^{\mathsf{T}} A (\boldsymbol{\theta}_m - \boldsymbol{\theta}_m^*)}\ d\boldsymbol{\theta}_m \\
&= P(\mathcal{D}|\boldsymbol{\theta}_m^*, m) P(\boldsymbol{\theta}_m^*|m) (2\pi)^{\frac{d}{2}} |A|^{-\frac{1}{2}}
\end{aligned}
$$

with $A = -\nabla^2 \log P(\mathcal{D}, \boldsymbol{\theta}_m^*|m)$ the negative of the Hessian matrix of $\log P(\mathcal{D}, \boldsymbol{\theta}|m)$ evaluated at $\boldsymbol{\theta}_m^*$. Note that we use the fact that the gradient at the mode vanishes.

This is equivalent to approximating the posterior by a Gaussian: an approximation which is asymptotically correct.

Bayesian Information Criterion (BIC)

BIC can be obtained from the Laplace approximation:

$$\log P(\mathcal{D}|m) \approx \log P(\boldsymbol{\theta}^*_m|m) + \log P(\mathcal{D}|\boldsymbol{\theta}^*_m, m) + \frac{d}{2}\log 2\pi - \frac{1}{2}\log|A|$$

in the large sample limit $(N \to \infty)$ where N is the number of data points.
A grows as NA_0 for some fixed matrix A_0, so $\log|A| \to \log|NA_0| = \log(N^d|A_0|) = d\log N + \log|A_0|$. Retaining only terms that grow in N we get:

$$\log P(\mathcal{D}|m) \approx \log P(\mathcal{D}|\boldsymbol{\theta}^*_m, m) - \frac{d}{2}\log N$$

Properties:

- Quick and easy to compute.
- It does not depend on the prior.
- We can use the ML estimate of θ instead of the MAP estimate
- It is related to the "Minimum Description Length" (MDL) criterion.
- It assumes that in the large sample limit, all the parameters are well-determined (i.e. the model is identifiable; otherwise, d should be the number of well-determined parameters).
- **Danger:** counting parameters can be deceiving!

Hyperparameters and Evidence Optimisation

In some cases, we need to choose between a family of continuously parameterised models.

$$P(\mathcal{D}|\eta) = \int P(\mathcal{D}|\boldsymbol{\theta}) P(\boldsymbol{\theta}|\underset{\substack{\uparrow \\ \text{hyperparameters}}}{\eta})\; d\boldsymbol{\theta}$$

This can often be done by gradient ascent in:

- The exact evidence (if tractable).
- Approximated evidence (Laplace, EP, Bethe, . . .)
- Free-energy bound on the evidence (VB)

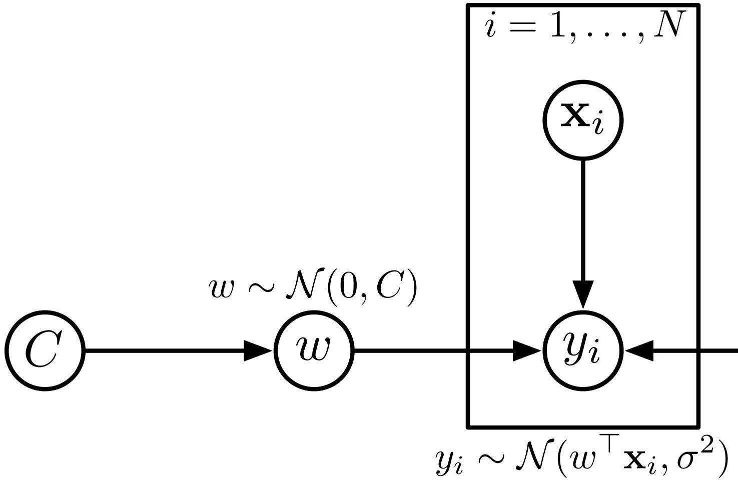
Another possibility: to place a hyperprior on the hyperparameters η, and obtain samples from the posterior

$$P(\eta|\mathcal{D}) = \frac{P(\mathcal{D}|\eta)P(\eta)}{P(\mathcal{D})}$$

using Markov chain Monte Carlo sampling.

Evidence Optimisation in Linear Regression

Consider simple linear regression:



$i = 1, \ldots, N$
$\mathbf{x}_i$
$w \sim \mathcal{N}(0, C)$
C
w
y_i
σ^2
$y_i \sim \mathcal{N}(w^\top \mathbf{x}_i, \sigma^2)$

- Maximize

$$P(y_1 \ldots y_N | \mathbf{x}_1 \ldots \mathbf{x}_N, C, \sigma^2) = \int P(y_1 \ldots y_N | \mathbf{x}_1 \ldots \mathbf{x}_N, \mathbf{w}, \sigma^2) P(\mathbf{w}|C) \, d\mathbf{w}$$

  to find optimal C, σ^2.

- Compute the posterior $P(\mathbf{w}|y_1 \ldots y_N, \mathbf{x}_1 \ldots \mathbf{x}_N, C, \sigma^2)$ given these optimal values.

The Evidence for Linear Regression

The posterior on **w** is normal, with variance $\Sigma = (\frac{XX^{\mathsf{T}}}{\sigma^2} + C^{-1})^{-1}$ and mean $\mu = \Sigma \frac{XY^{\mathsf{T}}}{\sigma^2}$. Note: X is a matrix where columns are input vectors, and Y is a row vector of corresponding predicted outputs.

The evidence, $\mathcal{E}(C, \sigma^2) = \int P(Y|X, \mathbf{w}, \sigma^2) P(\mathbf{w}|C)\, d\mathbf{w}$, is given by:

$$\mathcal{E}(C, \sigma^2) = \sqrt{\frac{|2\pi\Sigma|}{|2\pi\sigma^2 I|\,|2\pi C|}} \exp\left(-\frac{1}{2} Y \left(\frac{I}{\sigma^2} - \frac{X^{\mathsf{T}}\Sigma X}{\sigma^4}\right) Y^{\mathsf{T}}\right)$$

For optimization, general forms for the gradients are available. If θ is a parameter in C:

$$\frac{\partial}{\partial\theta} \log \mathcal{E}(C, \sigma^2) = \frac{1}{2} \mathsf{Tr}\left[(C - \Sigma - \mu\mu^{\mathsf{T}}) \frac{\partial}{\partial\theta} C^{-1}\right]$$

$$\frac{\partial}{\partial\sigma^2} \log \mathcal{E}(C, \sigma^2) = \frac{1}{\sigma^2}\left(-N + \mathsf{Tr}\left[I - \Sigma C^{-1}\right] + \frac{1}{\sigma^2}(Y - \mu^{\mathsf{T}} X)(Y - \mu^{\mathsf{T}} X)^{\mathsf{T}}\right)$$

Automatic Relevance Determination

The standard form of evidence optimization for regression (due to MacKay and Neal [3]) takes $C^{-1} = \text{diag}(\boldsymbol{\alpha})$ (i.e. $w_i \sim \mathcal{N}(0, \alpha_i^{-1})$) and then optimizes the precisions $\{\alpha_i\}$. Setting the gradients to 0 and solving gives

$$\begin{aligned} \alpha_i^{\text{new}} &= \frac{1 - \alpha_i \Sigma_{ii}}{\mu_i^2} \\ (\sigma^2)^{\text{new}} &= \frac{(Y - \mu^{\mathsf{T}} X)(Y - \mu^{\mathsf{T}} X)^{\mathsf{T}}}{N - \sum_i (1 - \Sigma_{ii}\alpha_i)} \end{aligned}$$

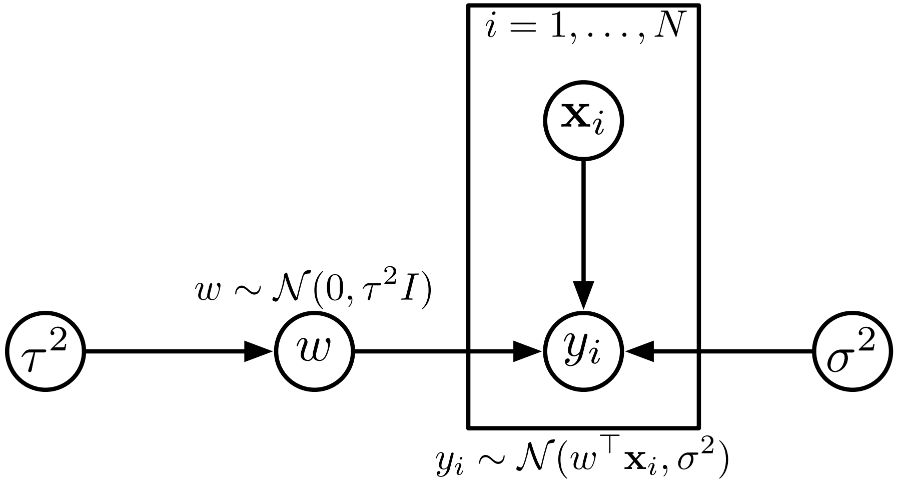
During optimization the α_i's meet one of two fates

$$\begin{array}{rcll} \alpha_i \to \infty & \Rightarrow & w_i = 0 & \text{irrelevant feature } i \\ \alpha_i \text{ finite} & \Rightarrow & w_i = \text{argmax P}\left(w_i \mid X, Y, \alpha_i\right) & \text{relevant feature } i \end{array}$$

This procedure, Automatic Relevance Determination (ARD), yields sparse solutions that improve on ML regression.

Evidence optimisation is also called maximum marginal likelihood or ML-2 (Type 2 maximum likelihood).

Linear Regression Revisited



$i = 1, \ldots, N$
$\mathbf{x}_i$
$w \sim \mathcal{N}(0, \tau^2 I)$
τ^2
w
y_i
σ^2
$y_i \sim \mathcal{N}(w^\top \mathbf{x}_i, \sigma^2)$

Linear regression predicts output y given input vector **x** by:

$$y \sim \mathcal{N}(\mathbf{w}^\top \mathbf{x}, \sigma^2)$$

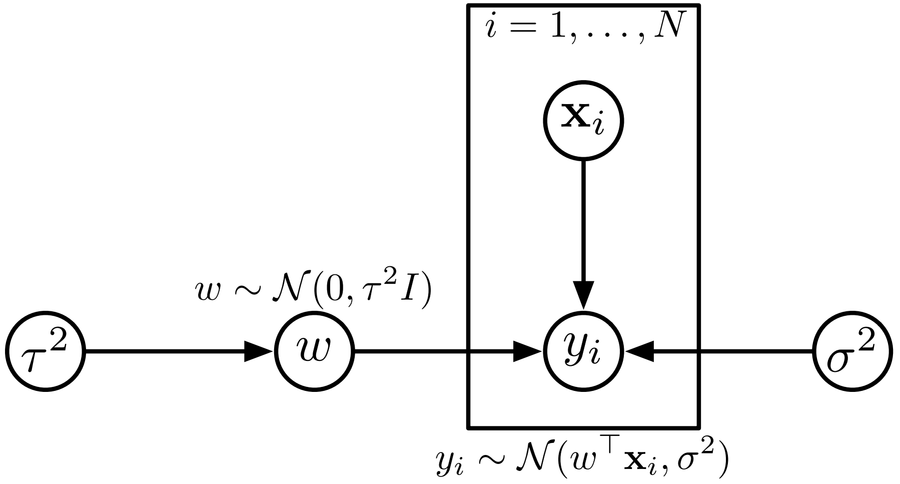
Posterior over **w** is Gaussian with covariance $\Sigma = (\frac{1}{\sigma^2} X X^\top + \frac{1}{\tau^2} I)^{-1}$ and mean $\mu = \frac{1}{\sigma^2} \Sigma X Y^\top$ (where X is matrix with columns being input vectors, Y is row vector of outputs).

Given a new input vector **x**′, the predicted output y' is (integrating out **w**):

$$y' | \mathbf{x}' \sim \mathcal{N}(\mu^\top \mathbf{x}', \mathbf{x}'^\top \Sigma \mathbf{x}' + \sigma^2)$$

the additional variance term $\mathbf{x}'^\top \Sigma \mathbf{x}'$ results from the posterior uncertainty in **w**.

Alternative View of Linear Regression


$i = 1, \ldots, N$
$\mathbf{x}_i$
$w \sim \mathcal{N}(0, \tau^2 I)$
τ^2
w
y_i
σ^2
$y_i \sim \mathcal{N}(w^\top \mathbf{x}_i, \sigma^2)$

Integrating out $\mathbf{w}$, the joint distribution of $y_1, \ldots, y_N$ given $\mathbf{x}_1, \ldots, \mathbf{x}_N$ is Gaussian. The means and covariances are:

$$\begin{aligned}
E[y_i] &= E[\mathbf{w}^\mathsf{T}\mathbf{x}_i] = 0^\mathsf{T}\mathbf{x}_i = 0 \\
E[(y_i - 0)^2] &= E[(\mathbf{x}_i^\mathsf{T}\mathbf{w})(\mathbf{w}^\mathsf{T}\mathbf{x}_i)] + \sigma^2 = \tau^2\mathbf{x}_i^\mathsf{T}\mathbf{x}_i + \sigma^2 \\
E[(y_i - 0)(y_j - 0)] &= E[(\mathbf{x}_i^\mathsf{T}\mathbf{w})(\mathbf{w}^\mathsf{T}\mathbf{x}_j)] = \tau^2\mathbf{x}_i^\mathsf{T}\mathbf{x}_j
\end{aligned}$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \Bigg| \mathbf{x}_1, \ldots, \mathbf{x}_N \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} \tau^2\mathbf{x}_1^\mathsf{T}\mathbf{x}_1 + \sigma^2 & \tau^2\mathbf{x}_1^\mathsf{T}\mathbf{x}_2 & \cdots & \tau^2\mathbf{x}_1^\mathsf{T}\mathbf{x}_N \\ \tau^2\mathbf{x}_2^\mathsf{T}\mathbf{x}_1 & \tau^2\mathbf{x}_2^\mathsf{T}\mathbf{x}_2 + \sigma^2 & & \tau^2\mathbf{x}_2^\mathsf{T}\mathbf{x}_N \\ \vdots & & \ddots & \vdots \\ \tau^2\mathbf{x}_N^\mathsf{T}\mathbf{x}_1 & \tau^2\mathbf{x}_N^\mathsf{T}\mathbf{x}_2 & \cdots & \tau^2\mathbf{x}_N^\mathsf{T}\mathbf{x}_N + \sigma^2 \end{bmatrix} \right)$$

$$Y^\mathsf{T}|X \sim \mathcal{N}(0_N, \tau^2 X^\mathsf{T} X + \sigma^2 I_N)$$

Alternative View of Linear Regression

If we also include the test input vector $\mathbf{x}'$ and test output y':

$$\begin{bmatrix} Y^\top \\ y' \end{bmatrix} \Bigg| X, \mathbf{x}' \sim \mathcal{N}\left(\begin{bmatrix} 0_N \\ 0 \end{bmatrix}, \begin{bmatrix} \tau^2 X^\top X + \sigma^2 I & \tau^2 X^\top \mathbf{x}' \\ \tau^2 \mathbf{x}'^\top X & \tau^2 \mathbf{x}'^\top \mathbf{x}' + \sigma^2 \end{bmatrix} \right)$$

Conditioning on the observed output values of Y, the distribution of y' can be worked out using standard results of multivariate Gaussian distributions,

$$y' | Y, X, \mathbf{x}' \sim \mathcal{N}\left(\tfrac{1}{\sigma^2} \mathbf{x}'^\top \Sigma X Y^\top, \mathbf{x}'^\top \Sigma \mathbf{x}' + \sigma^2 \right) \qquad \Sigma = \left(\tfrac{1}{\sigma^2} X X^\top + \tfrac{1}{\tau^2} I \right)^{-1}$$

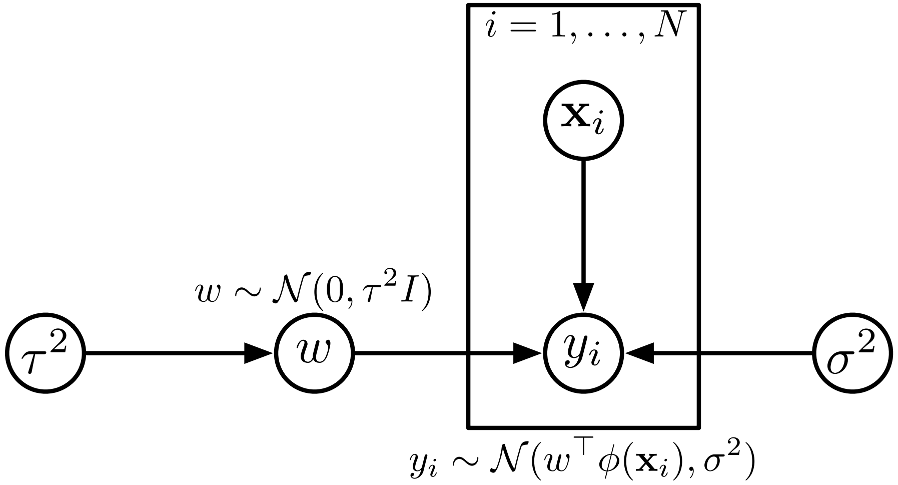
The above result is exactly the same as when we computed the posterior for $\mathbf{w}$, then the predictive distribution over y'.

Similarly, the evidence $P(Y|X)$ can be computed and will be equal to what we obtained previously.

The point: we can do regression if we can express the joint distribution over all outputs Y given all inputs as a big Gaussian, regardless of the functional form involved.

Next: nonlinear regression.

Nonlinear Regression



$i = 1, \ldots, N$
$\mathbf{x}_i$
$w \sim \mathcal{N}(0, \tau^2 I)$
τ^2
w
y_i
σ^2
$y_i \sim \mathcal{N}(w^\top \phi(\mathbf{x}_i), \sigma^2)$

Introduce a nonlinear mapping $\mathbf{x} \mapsto \phi(\mathbf{x})$.
Each entry in $\phi(\mathbf{x})$ is understood as a (nonlinear) feature extracted from $\mathbf{x}$.

The resulting function $f(\mathbf{x}) = \mathbf{w}^\mathsf{T}\phi(\mathbf{x})$ is nonlinear, but outputs Y still jointly Gaussian!

$$Y^\mathsf{T}|X \sim \mathcal{N}(0_N, \tau^2\Phi^\mathsf{T}\Phi + \sigma^2 I_N)$$

where the i^{th} column of matrix Φ is $\phi(\mathbf{x}_i)$.

Proceeds as before, e.g. the predictive distribution over y' on a test input $\mathbf{x}'$ is:

$$\begin{aligned} y'|Y, X, \mathbf{x}' &\sim \mathcal{N}\left(\tau^2\phi(\mathbf{x}')^\mathsf{T}\Phi K^{-1}Y^\mathsf{T}, \tau^2\phi(\mathbf{x}')^\mathsf{T}\phi(\mathbf{x}') + \sigma^2 - \tau^4\phi(\mathbf{x})^\mathsf{T}\Phi K^{-1}\Phi^\mathsf{T}\phi(\mathbf{x}')\right) \\ K &= \tau^2\Phi^\mathsf{T}\Phi + \sigma^2 I \end{aligned}$$

The Covariance Kernel

$$Y^{\mathsf{T}}|X \sim \mathcal{N}(0_N, \tau^2\Phi^{\mathsf{T}}\Phi + \sigma^2 I_N)$$

The covariance of the output vector Y plays a central role in the development of the theory of Gaussian processes.

Define the covariance kernel K as follows. If $\mathbf{x}, \mathbf{x}'$ are two input vectors with corresponding outputs y, y', then

$$K(\mathbf{x}, \mathbf{x}') = \mathrm{Cov}[y, y'] = E[yy'] - E[y]E[y']$$

In the nonlinear regression example we have $K(\mathbf{x}, \mathbf{x}') = \tau^2\phi(\mathbf{x})^{\mathsf{T}}\phi(\mathbf{x}') + \sigma^2\delta_{\mathbf{x}=\mathbf{x}'}$.

The covariance kernel has two properties:

- Symmetric: $K(\mathbf{x}, \mathbf{x}') = K(\mathbf{x}', \mathbf{x})$ for all $\mathbf{x}, \mathbf{x}'$.
- Positive semidefinite: the matrix $[K(\mathbf{x}_i, \mathbf{x}_j)]$ formed by any finite set of input vectors $\mathbf{x}_1, \ldots, \mathbf{x}_N$ is positive semidefinite.

Theorem: A covariance kernel K is symmetric and positive semidefinite if and only if there is a feature map ϕ such that

$$K(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^{\mathsf{T}}\phi(\mathbf{x}')$$

The feature map $\phi(\mathbf{x})$ can potentially be infinite dimensional.

Gaussian Process Regression

Let K be a covariance kernel. Simply define the joint distribution over outputs Y given inputs X by

$$Y|X, K \sim \mathcal{N}(0_N, K(X, X))$$

where the i, j entry in the covariance matrix $K(X, X)$ is $K(\mathbf{x}_i, \mathbf{x}_j)$.

By the previous theorem this is equivalent to implicitly using a (potentially infinite-dimensional) feature map $\phi(\mathbf{x})$. This is called the kernel trick.

Prediction: compute the predictive distribution of y' condition on Y:

$$y'|\mathbf{x}', X, Y, K \sim \mathcal{N}(\underbrace{K(\mathbf{x}', X)K(X, X)^{-1}Y^\top}_{\text{mean}}, \underbrace{K(\mathbf{x}', \mathbf{x}') - K(\mathbf{x}', X)K(X, X)^{-1}K(X, \mathbf{x}')}_{\text{variance}})$$

Evidence: this is just the Gaussian likelihood:

$$P(Y|X, K) = |2\pi K(X, X)|^{-\frac{1}{2}} e^{-\frac{1}{2} Y K(X,X)^{-1} Y^\top}$$

Evidence optimisation: the covariance kernel K often has parameters, and these can be optimized by gradient ascent in $\log P(Y|X, K)$.

The Gaussian Process

A **Gaussian process** (GP) is a collection of random variables, any finite number of which have (consistent) Gaussian distributions.

In our regression setting, corresponding to each input vector $\mathbf{x}$ we have an output $f(\mathbf{x})$. Given $X = [\mathbf{x}_1, \dots, \mathbf{x}_N]$, the joint distribution of the outputs $F = [f(\mathbf{x}_1), \dots, f(\mathbf{x}_N)]$ is:

$$F|X, K \sim \mathcal{N}(0, K(X, X))$$

Thus the random function $f(\mathbf{x})$ (as a collection of random variables, one $f(\mathbf{x})$ for each $\mathbf{x}$) is a Gaussian process.

In general, a Gaussian process is parametrized by a mean function $m(\mathbf{x})$ and covariance kernel $K(\mathbf{x}, \mathbf{x}')$, and we write

$$f(\cdot) \sim \mathcal{GP}(m(\cdot), K(\cdot, \cdot))$$

Posterior Gaussian process: on observing X and F, the conditional joint distribution of $F' = [f(\mathbf{x}'_1), \dots, f(\mathbf{x}'_M)]$ on another set of input vectors $\mathbf{x}'_1, \dots, \mathbf{x}'_M$ is still Gaussian:

$$F'|X', X, F, K \sim \mathcal{N}(K(X', X)K(X, X)^{-1}F^{\mathsf{T}}, K(X', X') - K(X', X)K(X, X)^{-1}K(X, X'))$$

thus the posterior over functions $f(\cdot)|X, F$ is still a Gaussian process!

Regression with Gaussian Processes

We wish to model the joint distribution of outputs $y_1, \ldots, y_N$ given inputs $\mathbf{x}_1, \ldots, \mathbf{x}_N$.
Use a GP prior over functions:

$$f(\cdot) \sim \mathcal{GP}(0, K(\cdot, \cdot))$$

Usually, instead of treating y_i as direct observation of the function value $f(\mathbf{x}_i)$, we add Gaussian observation noise:

$$y_i | \mathbf{x}_i, f(\cdot) \sim \mathcal{N}(f(\mathbf{x}_i), \sigma^2)$$

Evidence: again this is just a multivariate Gaussian likelihood,

$$P(Y|X) = |2\pi(K(X,X) + \sigma^2 I)|^{-\frac{1}{2}} e^{-\frac{1}{2} Y (K(X,X)+\sigma^2 I)^{-1} Y^\mathsf{T}}$$

Posterior: the posterior function is still a GP,

$$f(\cdot)|X,Y \sim \mathcal{GP}(K(\cdot,X)(K(X,X)+\sigma^2 I)^{-1} Y^\mathsf{T}, K(\cdot,\cdot) - K(\cdot,X)(K(X,X)+\sigma^2 I)^{-1} K(X,\cdot))$$

Prediction: the predictive distribution is just posterior plus observation noise:

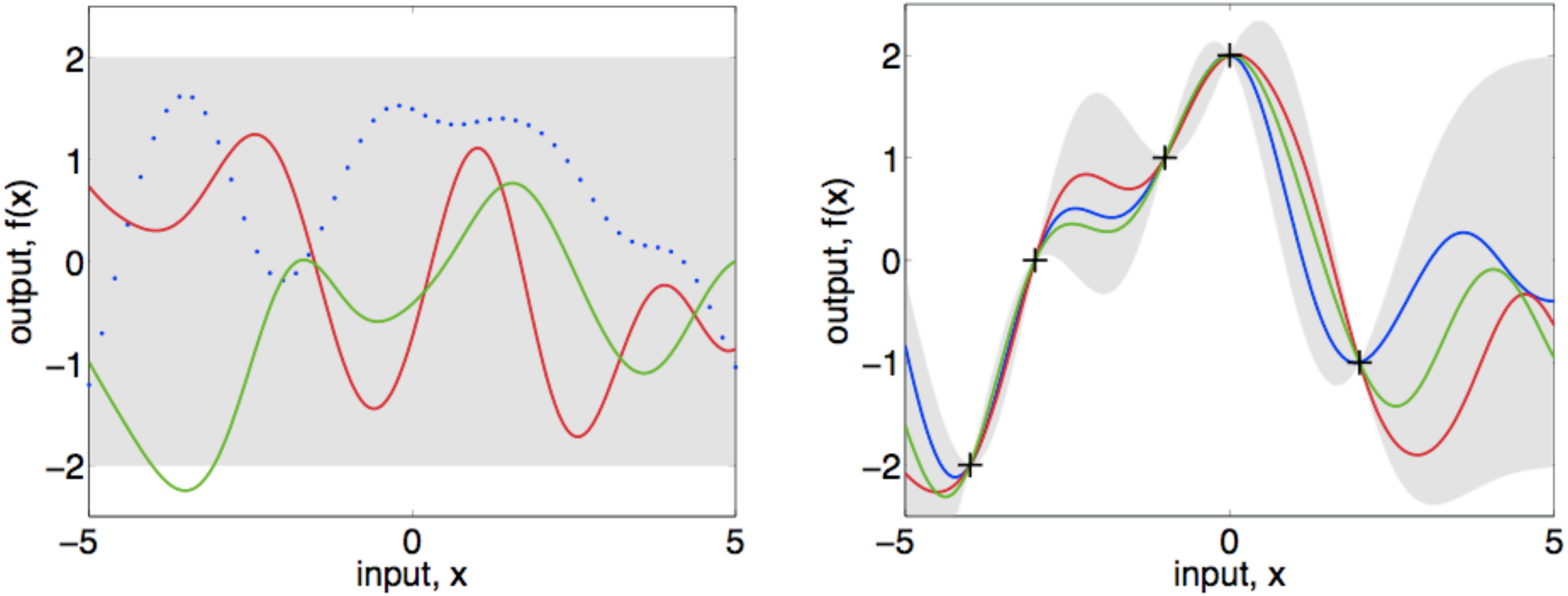
$$y' | X, Y, \mathbf{x}' \sim \mathcal{N}(E[f(\mathbf{x}')|X,Y], \mathrm{Var}[f(\mathbf{x}')|X,Y] + \sigma^2)$$

Evidence Optimisation: we can do this by gradient ascent in $\log P(Y|X)$.

Samples from a Gaussian Process

We can draw sample functions from a GP by fixing a set of input vectors $\mathbf{x}_1, \ldots, \mathbf{x}_N$, and drawing a sample $f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N)$ from the corresponding multivariate Gaussian. This can then be plotted.

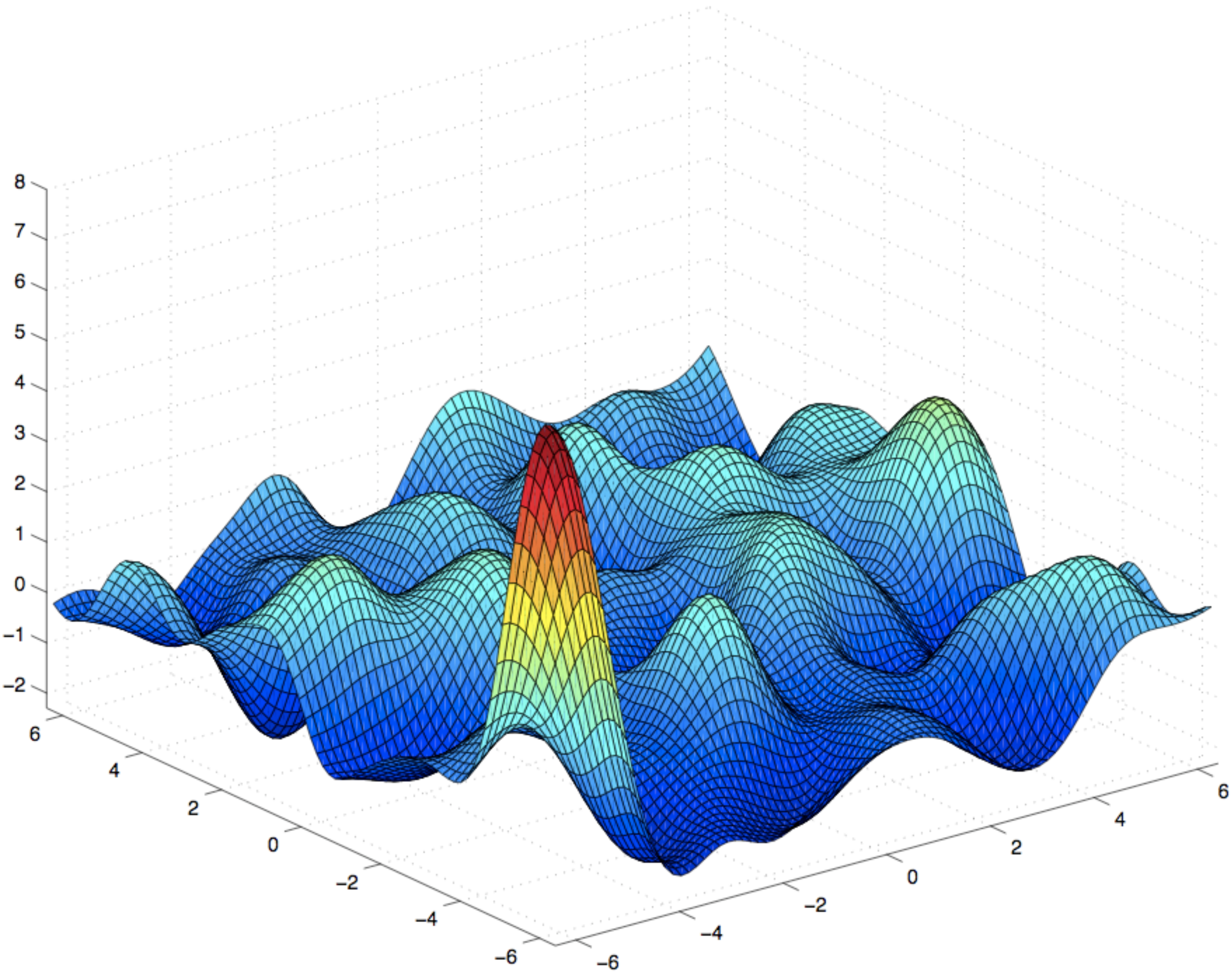
Below we plot samples from an example prior and corresponding posterior GP.



Another approach is to

- sample $f(\mathbf{x}_1)$ first,
- then $f(\mathbf{x}_2)|f(\mathbf{x}_1)$,
- and generally $f(\mathbf{x}_n)|f(\mathbf{x}_1), \ldots, f(\mathbf{x}_{n-1})$ for $n = 1, 2, \ldots$.

Sample from a 2D Gaussian Process



Covariance Kernels

Examples of covariance kernels:

- Polynomial:

$$K(\mathbf{x},\mathbf{x}') = (1+\mathbf{x}^{\mathsf{T}}\mathbf{x}')^m \qquad m = 1, 2, \ldots$$

- Squared-exponential:

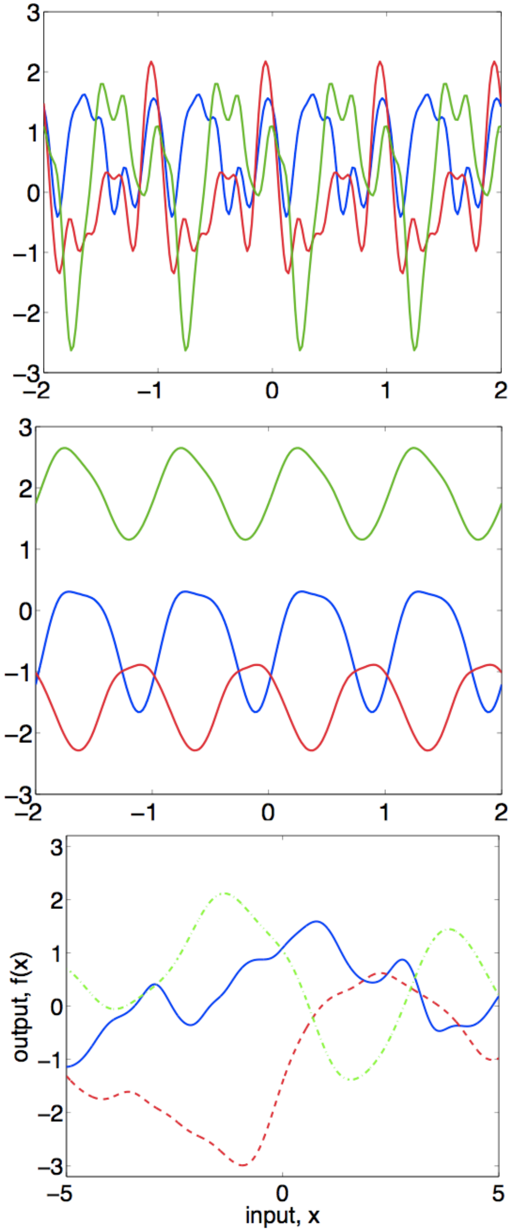
$$K(\mathbf{x},\mathbf{x}') = \theta^2 e^{-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\eta^2}}$$

- Periodic:

$$K(x,x') = \theta^2 e^{-\frac{2\sin^2(\pi(x-x')/\tau)}{\eta^2}}$$

- Rational Quadratic:

$$K(\mathbf{x},\mathbf{x}') = \left(1+\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\alpha\eta^2}\right)^{-\alpha} \qquad \alpha > 0$$



Covariance Kernels

If K_1 and K_2 are covariance kernels, then so are:

- Rescaling: αK_1 for $\alpha > 0$.
- Addition: $K_1 + K_2$
- Elementwise product: $K_1 K_2$
- Mapping: $K_1(\phi(\mathbf{x}), \phi(\mathbf{x}'))$ for some function ϕ.

We say a covariance kernel is translation-invariant if

$$K(\mathbf{x}, \mathbf{x}') = h(\mathbf{x} - \mathbf{x}')$$

A GP with a translation-invariant covariance kernel is stationary: if $f(\cdot) \sim \mathcal{GP}(0, K)$, then so is $f(\cdot - \mathbf{x}) \sim \mathcal{GP}(0, K)$ for each $\mathbf{x}$.

We say a covariance kernel is radial if

$$K(\mathbf{x}, \mathbf{x}') = h(\|\mathbf{x} - \mathbf{x}'\|)$$

A GP with a radial covariance kernel is stationary with respect to translations, rotations, and reflections of the input space.

Nonparametric Bayesian Models and Occam's Razor Revisited

We motivated the need for model comparison by showing how models can overfit to training data if they contain too many parameters.

In the Bayesian treatment of probabilistic models, all parameters are integrated out, so *none* of them can overfit to data! For example, in Gaussian processes, the parameter is the function $f(\mathbf{x})$ itself, which can be infinite-dimensional.

The Gaussian process is an example of **nonparametric Bayesian models**, which are models with an infinite number of parameters.

Nonparametric Bayesian models can often be constructed as the infinite limit of a nested family of finite models. As opposed to the usual Occam's Razor argument, the nonparametric Bayesian paradigm says that since overfitting is avoided by integrating out all parameters, we should simply use the infinite (nonparametric) models.

This sidesteps the need for model selection. But Occam's Razor is still around: there is often hyperparameters which govern the complexity of the nonparametric model, and we will need to choose a good hyperparameter setting. However this can be achieved by optimizing the usual marginal likelihood (too much complexity is automatically penalized).

Thus the nonparametric Bayesian paradigm replaces model selection by hyperparameter optimization (usually easier) and no validation set or extra penalty terms required.

End Notes

Automatic relevance determination appeared in MacKay (1993) Bayesian Methods for Backpropagation Networks and Neal (1993) Bayesian Learning for Neural Networks.
Gaussian processes can also be used in classification and latent variable models. We will consider classification in the second half of course.

Many of the figures have been copied from a Gaussian process tutorial by Carl Rasmussen (MLSS 2007) at http://agbs.kyb.tuebingen.mpg.de/wikis/mlss07/CarlERasmussen

An excellent text book on Gaussian processes is Gaussian processes for Machine Learning by Rasmussen and Williams, available online at http://www.gaussianprocess.org/gpml/

The original paper on Gaussian process latent variable models is by Neil Lawrence (NIPS 2004) at http://www.cs.man.ac.uk/∼neill/