

# Probabilistic & Unsupervised Learning

## Model selection, Hyperparameter optimisation, and Gaussian Processes

Maneesh Sahani

maneesh@gatsby.ucl.ac.uk

Gatsby Computational Neuroscience Unit, and  
MSc ML/CSML, Dept Computer Science  
University College London

Term 1, Autumn 2013

### Model selection

Models (labelled by  $m$ ) have parameters  $\theta_m$  that specify the probability of data:

$$P(\mathcal{D}|\theta_m, m).$$

If model is known, learning  $\theta_m$  means finding posterior or point estimate (ML, MAP, ...).

What if we need to learn the model too?

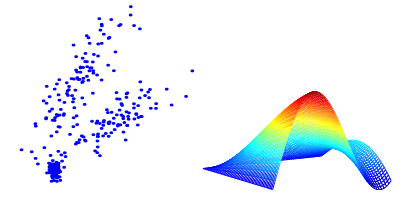
- ▶ Could combine models into a single “supermodel”, with composite parameter  $(m, \theta_m)$ .
  - ▶ ML learning will **overfit**: favours most flexible (nested) model with most parameters, even if the data actually come from a simpler one.
  - ▶ Density function on composite parameter space (union of manifolds of different dimensionalities) difficult to define  $\Rightarrow$  MAP learning ill-posed.
  - ▶ Joint posterior difficult to compute — dimension of composite parameter varies.

$\Rightarrow$  Separate model selection step:

$$P(\theta_m, m|\mathcal{D}) = \underbrace{P(\theta_m|m, \mathcal{D})}_{\text{model-specific posterior}} \cdot \underbrace{P(m|\mathcal{D})}_{\text{model selection}}$$

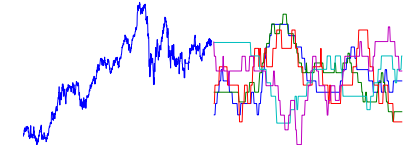
### Learning model structure

How many clusters in the data?



How smooth should the function be?

Is this input relevant to predicting that output?

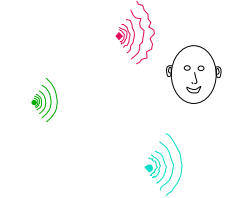


What is the order of a dynamical system?

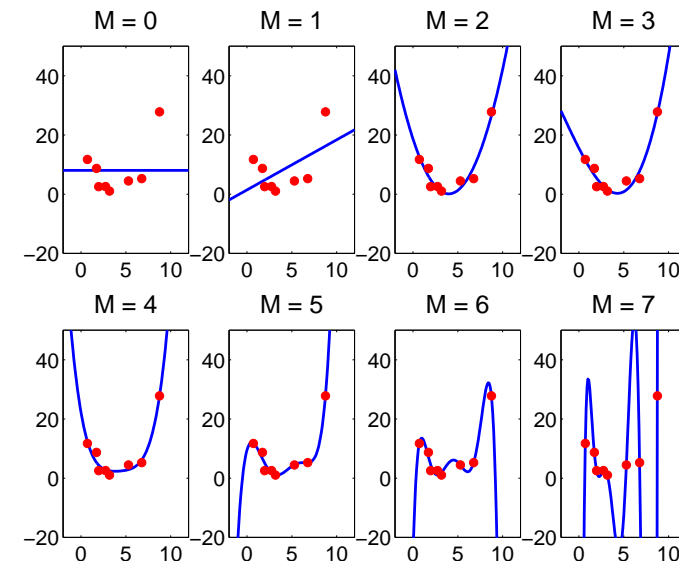
How many states in a hidden Markov model?

SVYDAAAQLTADVKKDLRDSWKVIGSDKKGNG

How many auditory sources in the input?



### Model complexity and overfitting: a simple example



## Model selection

Given models labeled by  $m$  with parameters  $\theta_m$ , identify the “correct” model for data  $\mathcal{D}$ .

ML/MAP has no good answer:  $P(\mathcal{D}|\theta_m^{\text{ML}})$  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 asymptotically in data number.
- ▶ Conservative (N-P hypothesis tests are asymmetric).

### Likelihood validation

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

### Bayesian model selection

- ▶ Choose most likely **model**:  $\text{argmax} P(m|\mathcal{D})$ .
- ▶ Principled from a probabilistic viewpoint—if true model is in set being considered—but sensitive to assumed priors etc.
- ▶ Can use posterior probabilities to **weight** models for combined predictions (no need to select at all).

## The Bayesian Occam’s razor

**Occam’s Razor** is a principle of scientific philosophy: of two explanations adequate to explain the same set of observations, the simpler should always be preferred.

Bayesian inference formalises and *automatically* implements a form of Occam’s Razor.

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

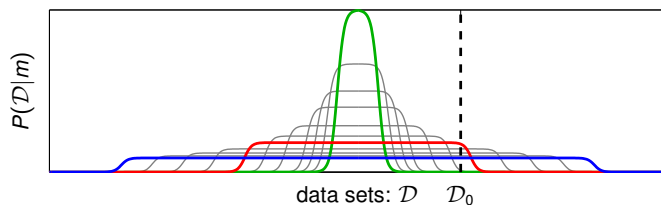
$$P(m|\mathcal{D}) = \frac{P(\mathcal{D}|m)P(m)}{P(\mathcal{D})}, \quad P(\mathcal{D}|m) = \int_{\Theta_m} P(\mathcal{D}|\theta_m, m)P(\theta_m|m) d\theta_m$$

$P(\mathcal{D}|m)$ : 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 observed 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.

Like Goldilocks, we favour a model that is **just right**.



## Bayesian model selection: some terminology

A **model class**  $m$  is a set of distributions parameterised by  $\theta_m$ , e.g. the set of all possible mixtures of  $m$  Gaussians.

The model implies both a **prior** over the parameters  $P(\theta_m|m)$ , and a **likelihood** of data given parameters (which might require integrating out latent variables)  $P(\mathcal{D}|\theta_m, m)$ .

The **posterior** distribution over parameters is

$$P(\theta_m|\mathcal{D}, m) = \frac{P(\mathcal{D}|\theta_m, m)P(\theta_m|m)}{P(\mathcal{D}|m)}$$

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

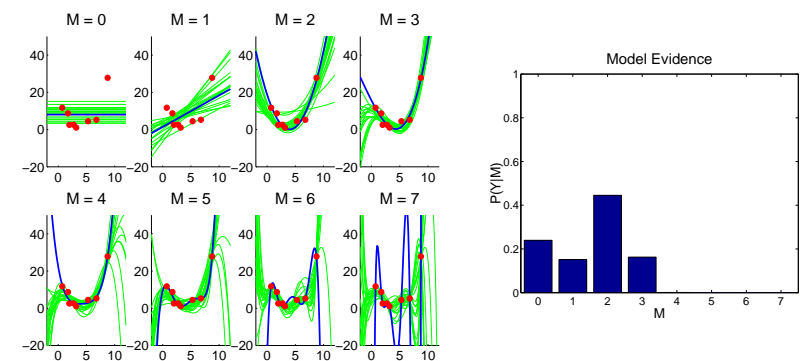
$$P(\mathcal{D}|m) = \int_{\Theta_m} P(\mathcal{D}|\theta_m, m)P(\theta_m|m) d\theta_m$$

(also called the **Bayesian evidence** for model  $m$ ).

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

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

## Bayesian model comparison: Occam’s razor at work



## Conjugate-exponential families (recap)

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

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

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

If our prior on  $\theta_m$  is **conjugate**:

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

then the joint is in the same family:

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

and so:

$$P(\mathcal{D}|m) = \int d\theta_m P(\mathcal{D}, \theta_m|m) = Z(\sum_i \mathbf{s}(\mathbf{x}_i) + \mathbf{s}_p, N + n_p) / Z(\mathbf{s}_p, p)$$

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

## Laplace approximation

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

As data size  $N$  grows (relative to parameter count  $d$ ),  $\theta_m$  becomes more constrained  
 $\Rightarrow P(\mathcal{D}, \theta_m|m) \propto P(\theta_m|\mathcal{D}, m)$  becomes concentrated on posterior mode  $\theta_m^*$ .

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

$$\begin{aligned} \int P(\mathcal{D}, \theta_m|m) d\theta_m &= \int e^{\log P(\mathcal{D}, \theta_m|m)} d\theta_m \\ &= \int e^{\log P(\mathcal{D}, \theta_m^*|m) + \underbrace{\nabla \log P(\mathcal{D}, \theta_m^*|m)}_{=0} \cdot (\theta_m - \theta_m^*) + \frac{1}{2} (\theta_m - \theta_m^*)^\top \underbrace{\nabla^2 \log P(\mathcal{D}, \theta_m^*|m)}_{=-A} (\theta_m - \theta_m^*)} d\theta_m \\ &= \int P(\mathcal{D}, \theta_m^*|m) e^{-\frac{1}{2} (\theta_m - \theta_m^*)^\top A (\theta_m - \theta_m^*)} d\theta_m \\ &= P(\mathcal{D}|\theta_m^*, m) P(\theta_m^*|m) (2\pi)^{\frac{d}{2}} |A|^{-\frac{1}{2}} \end{aligned}$$

$A = -\nabla^2 \log P(\mathcal{D}, \theta_m^*|m)$  is the negative Hessian of  $\log P(\mathcal{D}, \theta|m)$  evaluated at  $\theta_m^*$ .

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

## Practical Bayesian approaches

### ▶ Laplace approximation:

- ▶ Approximate posterior by a Gaussian centred at the maximum *a posteriori* parameter estimate.

### ▶ Bayesian Information Criterion (BIC)

- ▶ an asymptotic ( $N \rightarrow \infty$ ) approximation.

### ▶ Variational Bayes

- ▶ Lower bound on the marginal probability.
- ▶ Biased estimate.
- ▶ Easy and fast, and often better than Laplace or BIC.

### ▶ Monte Carlo methods:

- ▶ (Annealed) Importance sampling: estimate evidence using samples  $\theta^{(i)}$  from arbitrary  $f(\theta)$ :

$$\sum_i \frac{P(\mathcal{D}|\theta^{(i)}, m) P(\theta^{(i)}|m)}{f(\theta^{(i)})} \rightarrow \int d\theta f(\theta) \frac{P(\mathcal{D}, \theta|m)}{f(\theta)} = P(\mathcal{D}|m)$$

- ▶ "Reversible jump" Markov Chain Monte Carlo: sample from posterior on composite  $(m, \theta_m)$ . # samples for each  $m \propto p(m|\mathcal{D})$ .
- ▶ Both exact in the limit of infinite samples, but may have high variance with finite samples.

Not an exhaustive list (Bethe approximations, Expectation propagation, . . .)

We will discuss Laplace and BIC now, leaving the rest for the second half of course.

## Bayesian Information Criterion (BIC)

BIC can be obtained from the Laplace approximation:

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

We have

$$A = \nabla^2 \log P(\mathcal{D}, \theta^*|m) = \nabla^2 \log P(\mathcal{D}|\theta^*, m) + \nabla^2 \log P(\theta^*|m)$$

So as the number of iid data  $N \rightarrow \infty$ ,  $A$  grows as  $NA_0 + \text{constant}$  for a fixed matrix  $A_0$ .  
 $\Rightarrow \log |A| \rightarrow \log |NA_0| = \log(N^d |A_0|) = d \log N + \log |A_0|$ .

Retaining only terms that grow with  $N$  we get:

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

Properties:

- ▶ Quick and easy to compute.
- ▶ Does not depend on prior.
- ▶ We can use the ML estimate of  $\theta$  instead of the MAP estimate (= as  $N \rightarrow \infty$ ).
- ▶ Related to the "Minimum Description Length" (MDL) criterion.
- ▶ 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).
- ▶ Neglects multiple modes (e.g. permutations in a MoG).

## Hyperparameters and Evidence optimisation

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

$$P(\mathcal{D}|\eta) = \int P(\mathcal{D}|\theta)P(\theta|\eta) d\theta$$

↑  
hyperparameters

This choice can be made by ascending the gradient in:

- ▶ the exact evidence (if tractable).
- ▶ the approximated evidence (Laplace, EP, Bethe, ...)
- ▶ a free-energy bound on the evidence (Variational Bayes)

or by placing a **hyperprior** on the hyperparameters  $\eta$ , and sampling from the posterior

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

using Markov chain Monte Carlo sampling.

## The evidence for linear regression

- ▶ The posterior on  $\mathbf{w}$  is normal, with variance  $\Sigma = (\frac{XX^T}{\sigma^2} + C^{-1})^{-1}$  and mean  $\mu = \Sigma \frac{XY^T}{\sigma^2}$ .  
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:

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

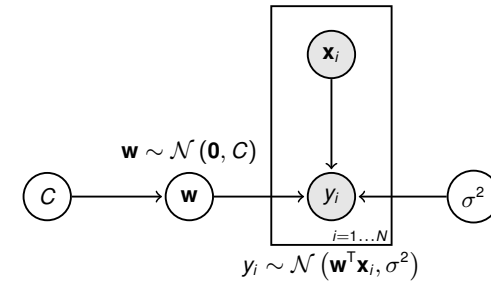
- ▶ For optimization, general forms for the gradients are available. If  $\theta$  is a parameter in  $C$ :

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

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

## Evidence optimisation in linear regression

Consider simple linear regression:



- ▶ Maximize

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

to find optimal values of  $C, \sigma$ .

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

## Automatic Relevance Determination

The most common form of evidence optimization for regression (due to MacKay and Neal) takes  $C^{-1} = \text{diag}(\alpha)$  (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

$$\alpha_i^{\text{new}} = \frac{1 - \alpha_i \Sigma_{ii}}{\mu_i^2}$$

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

During optimization the  $\alpha_i$ s meet one of two fates

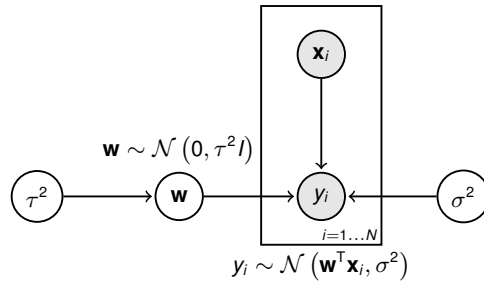
$$\alpha_i \rightarrow \infty \Rightarrow w_i = 0 \quad \text{irrelevant input } x_i$$

$$\alpha_i \text{ finite} \Rightarrow w_i = \text{argmax } P(w_i | X, Y, \alpha_i) \quad \text{relevant input } x_i$$

This procedure, **Automatic Relevance Determination** (ARD), yields **sparse** solutions that improve on ML regression. (cf. L<sub>1</sub>-regression or LASSO).

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

## Prediction averaging



Linear regression predicts output  $y$  given input vector  $\mathbf{x}$  by:

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

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

Given a new input vector  $\mathbf{x}'$ , the predicted output  $y'$  is (integrating out  $\mathbf{w}$ ):

$$y' | \mathbf{x}' \sim \mathcal{N}(\mu^T \mathbf{x}', \mathbf{x}'^T \Sigma \mathbf{x}' + \sigma^2)$$

the additional variance term  $\mathbf{x}'^T \Sigma \mathbf{x}'$  comes from the posterior uncertainty in  $\mathbf{w}$ .

## Predictions with marginalised regression

Now, include the test input vector  $\mathbf{x}'$  and test output  $y'$ :

$$\begin{bmatrix} Y^T \\ y' \end{bmatrix} | X, \mathbf{x}' \sim \mathcal{N} \left( \begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}, \begin{bmatrix} \tau^2 X^T X + \sigma^2 I & \tau^2 X^T \mathbf{x}' \\ \tau^2 \mathbf{x}'^T X & \tau^2 \mathbf{x}'^T \mathbf{x}' + \sigma^2 \end{bmatrix} \right)$$

We can find  $y' | Y, X, \mathbf{x}'$  by the standard multivariate Gaussian result:

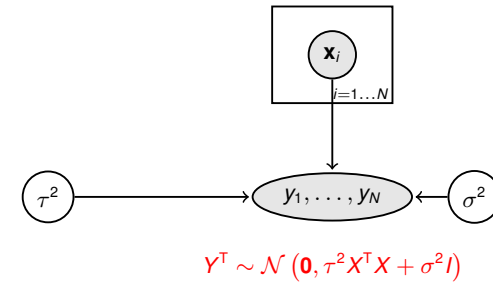
$$\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix} \sim \mathcal{N} \left( \begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}, \begin{bmatrix} A & C \\ C^T & B \end{bmatrix} \right) \Rightarrow \mathbf{b} | \mathbf{a} \sim \mathcal{N} \left( C^T A^{-1} \mathbf{a}, B - C^T A^{-1} C \right)$$

So

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

- ▶ Same answer as obtained by integrating wrt **posterior** over  $\mathbf{w}$ .
- ▶ Evidence  $P(Y|X)$  is just probability under joint Gaussian; also reduces to expression found previously.
- ▶ Thus, Bayesian linear regression can be derived from a **joint, parameter-free** distribution on all the outputs conditioned on all the inputs.

## Marginalised linear regression



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

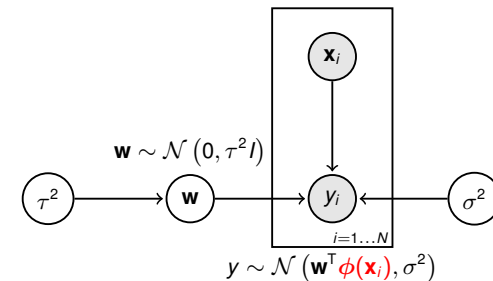
$$E[y_i] = E[\mathbf{w}^T \mathbf{x}_i] = \mathbf{0}^T \mathbf{x}_i = 0$$

$$E[(y_i - \bar{y}_i)^2] = E[(\mathbf{x}_i^T \mathbf{w})^2] + \sigma^2 = \tau^2 \mathbf{x}_i^T \mathbf{x}_i + \sigma^2$$

$$E[(y_i - \bar{y}_i)(y_j - \bar{y}_j)] = E[(\mathbf{x}_i^T \mathbf{w})(\mathbf{x}_j^T \mathbf{w})] = \tau^2 \mathbf{x}_i^T \mathbf{x}_j$$

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

## Nonlinear regression



We can also introduce a nonlinear mapping  $\mathbf{x} \mapsto \phi(\mathbf{x})$ ? Each element of  $\phi(\mathbf{x})$  is a (nonlinear) feature extracted from  $\mathbf{x}$ . May be many more features than elements on  $\mathbf{x}$ .

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

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

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

Proceeding as before, the predictive distribution over  $y'$  for a test input  $\mathbf{x}'$  is:

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

## The covariance kernel

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

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

Define the **covariance kernel** function  $K : \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{R}$  such that if  $\mathbf{x}, \mathbf{x}' \in \mathbb{X}$  are two input vectors with corresponding outputs  $y, y'$ , then

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

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

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  $\mathbf{x}_1, \dots, \mathbf{x}_N$  is positive semidefinite.

**Theorem:** A covariance kernel  $K : \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{R}$  is symmetric and positive semidefinite if and only if there is a feature map  $\phi : \mathbb{X} \rightarrow \mathbb{H}$  such that

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

The feature space  $\mathbb{H}$  can potentially be infinite dimensional.

## 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, for 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:

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

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

## Regression using the covariance kernel

For non-linear regression, all operations depended on  $K(\mathbf{x}, \mathbf{x}')$  rather than explicitly on  $\phi(\mathbf{x})$ .

So we can define the joint in terms of  $K$  *implicitly* using a (potentially infinite-dimensional) feature map  $\phi(\mathbf{x})$ .

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

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

This is called the **kernel trick**.

**Prediction:** compute the predictive distribution of  $y'$  conditioned on  $Y$ :

$$y' | \mathbf{x}', X, Y, K \sim \mathcal{N}(\underbrace{K(\mathbf{x}', X)K(X, X)^{-1}Y^T}_{\text{mean}}, \underbrace{K(\mathbf{x}', \mathbf{x}') - K(\mathbf{x}', X)K(X, X)^{-1}K(X, \mathbf{x}')}_{\text{variance}})$$

**Evidence:** this is just the Gaussian likelihood:

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

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

## Regression with Gaussian processes

We seek to learn the function that maps inputs  $\mathbf{x}_1, \dots, \mathbf{x}_N$  to outputs  $y_1, \dots, y_N$ .

Instead of assuming a specific form, we consider a random function drawn from a GP prior:

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

Any function is possible (no restriction on support) but some are (much) more likely *a priori*.

Observations  $y_i$  usually taken to be noisy versions of latent  $f(\mathbf{x}_i)$ :

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

**Evidence:** given by the multivariate Gaussian likelihood:

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

**Posterior:** also a GP:

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

**Predictions:** posterior on  $f$ , plus observation noise:

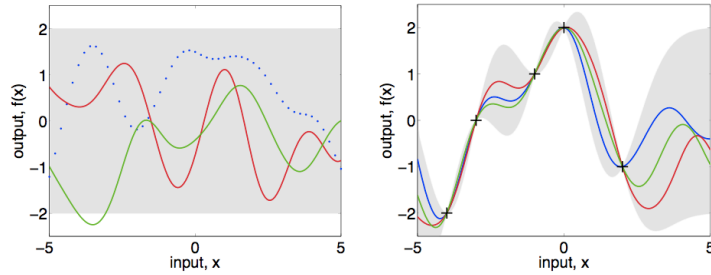
$$y' | X', Y, \mathbf{x}' \sim \mathcal{N}(E[f(\mathbf{x}') | X, Y], \text{Var}[f(\mathbf{x}') | X, Y] + \sigma^2)$$

**Evidence Optimisation:** 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  $\mathbf{x}_1, \dots, \mathbf{x}_N$ , and drawing a sample  $f(\mathbf{x}_1), \dots, f(\mathbf{x}_N)$  from the corresponding multivariate Gaussian. This can then be plotted.

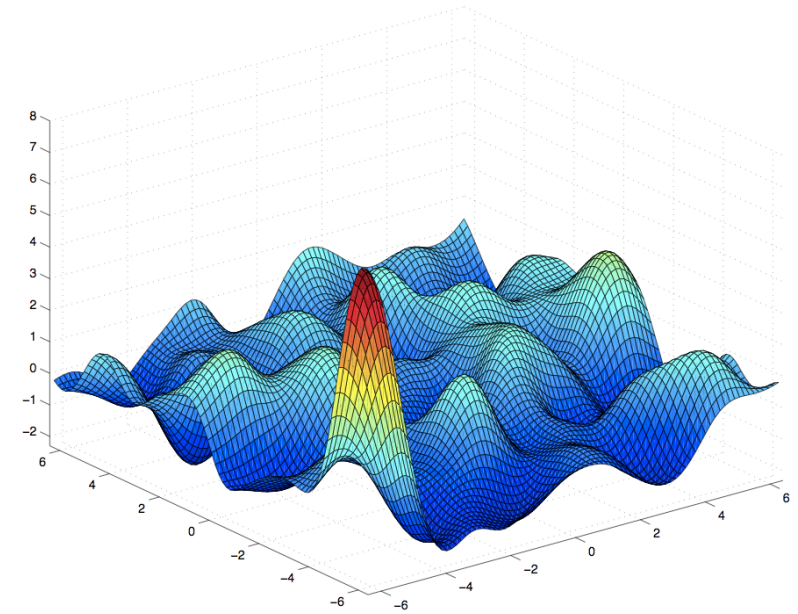
Example prior and posterior GPs:



Another approach is to

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

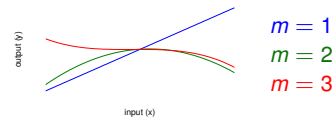
## Sample from a 2D Gaussian process



## Examples of covariance kernels

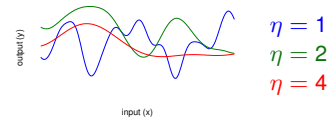
- Polynomial:

$$K(\mathbf{x}, \mathbf{x}') = (1 + \mathbf{x}^T \mathbf{x}')^m \quad m = 1, 2, \dots$$



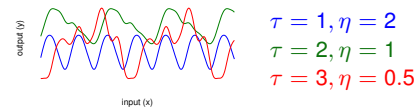
- Squared-exponential (or exponentiated-quadratic):

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



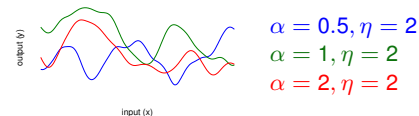
- Periodic (exp-sine):

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



- Rational Quadratic:

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



## Forms of kernels

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

- ▶ Rescaling:  $\alpha K_1$  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  $\phi$ .

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  $\mathbf{x}$ .

A covariance kernel is **radial** or **radially symmetric** 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

Overparameterised models can **overfit**. In the GP, the parameter is the function  $f(\mathbf{x})$  which can be infinite-dimensional.

However, the Bayesian treatment integrates over these parameters: we never identify a single "best fit"  $f$ , just a posterior (and posterior mean). So  $f$  cannot be adjusted to overfit the data.

The GP is an example of the larger class of **nonparametric Bayesian models**.

- ▶ Infinite number of parameters.
- ▶ Often constructed as the infinite limit of a nested family of finite models (sometimes equivalent to infinite model averaging).
- ▶ Parameters integrated out, so effective number of parameters to overfit is zero or small (hyperparameters).
- ▶ No need for model selection. Bayesian posterior on parameters will concentrate on "submodel" with largest integral automatically.
- ▶ No explicit need for Occam's razor, validation or added regularisation penalty.