

Probabilistic & Unsupervised Learning

Approximate Inference

Bayesian 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 2022

Model selection

Models (labelled by m) have parameters θ_m that specify the probability of data:

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

If model is known, learning θ_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, θ_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 [although Monte-Carlo methods may be able to sample from such a posterior.]

\Rightarrow Separate model selection step:

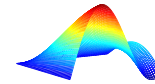
$$P(\theta_m, m|\mathcal{D}) = \underbrace{P(\theta_m|m, \mathcal{D})}_{\text{model-specific posterior}} \cdot \underbrace{P(m|\mathcal{D})}_{\text{model selection}}$$

Learning model structure

How many clusters in the data?



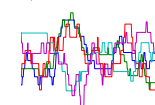
How smooth should the function be?



Is this input relevant to predicting that output?



What is the order of a dynamical system?



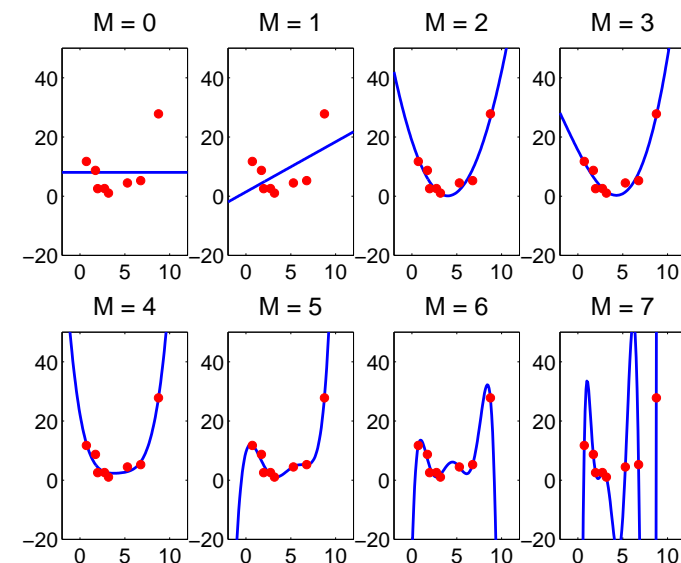
How many states in a hidden Markov model?

SVYDAAAQLTADVKKDLRDSWKVIGSDKKKNG

How many auditory sources in the input?



Model complexity and overfitting: a simple example



Model selection

Given models labeled by m with parameters θ_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.
- ▶ Tests often only exact asymptotically in data number.
- ▶ Conservative (N-P hypothesis tests are asymmetric by design).

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 $\theta_m^{\text{ML}} = \text{argmax} P(\mathcal{D}_{\text{tr}}|\theta)$. [Or, better, greatest $P(\mathcal{D}_{\text{vid}}|\mathcal{D}_{\text{tr}}, m)$.]
- ▶ **Consistent**; and selects most **useful** model, even if all are **incorrect**.
- ▶ May be biased towards simpler models; often high-variance.
- ▶ **Cross-validation** uses multiple partitions and averages likelihoods.

Bayesian model selection

- ▶ Choose **most likely** model: $\text{argmax} P(m|\mathcal{D})$.
- ▶ **Consistent**; probabilistically principled **if true model is in set being considered**, but sensitive to assumed priors etc.
- ▶ Posterior probabilities can **weight** models for combined predictions (avoiding selection).

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:

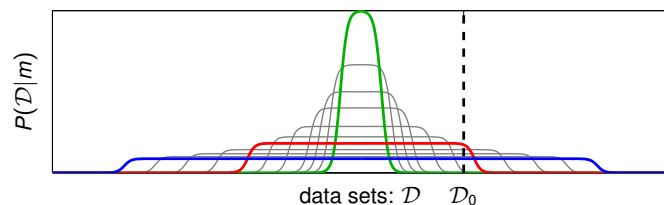
$$P(m|\mathcal{D}) = \frac{P(\mathcal{D}|m)P(m)}{P(\mathcal{D})}, \quad P(\mathcal{D}|m) = \int_{\Theta_m} P(\mathcal{D}|\theta_m, m)P(\theta_m|m) d\theta_m$$

$P(\mathcal{D}|m)$: 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 θ_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:

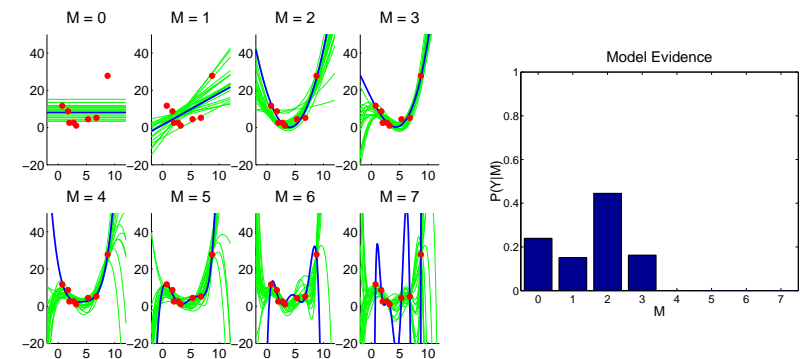
$$P(\mathcal{D}|m) = \int_{\Theta_m} P(\mathcal{D}|\theta_m, m)P(\theta_m|m) d\theta_m$$

(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 θ_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:

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

and so:

$$P(\mathcal{D}|m) = \int d\theta_m P(\mathcal{D}, \theta_m|m) = \frac{Z(\sum_i \mathbf{s}(\mathbf{x}_i) + \mathbf{s}_p, N + n_p)}{Z(\mathbf{s}_p, \rho)} \quad \leftarrow \begin{array}{l} \text{posterior volume} \\ \text{prior volume} \end{array}$$

While the intuition is general, tractability is a special case. Thus, we must 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), θ_m becomes more constrained $\Rightarrow P(\mathcal{D}, \theta_m|m) \propto P(\theta_m|\mathcal{D}, m)$ becomes concentrated on posterior mode θ_m^* .

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

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

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

Bethe approximations, Expectation propagation . . .

We will encounter many of these approaches later in the 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 $N = |\mathcal{D}| \rightarrow \infty$, $A \rightarrow NA_0 + \text{constant}$, for fixed PD matrix $A_0 = \langle -\nabla^2 \log P(\mathbf{x}|\theta^*, m) \rangle$. $\Rightarrow \log |A| \rightarrow \log |NA_0| = \log(N^d |A_0|) = d \log N + \log |A_0|$.

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 θ instead of the MAP estimate (= as $N \rightarrow \infty$).
- Related to the "Minimum Description Length" (MDL) criterion (Asst 2).
- 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 η , 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: $\Sigma_{\mathbf{w}} = \left(\frac{XX^T}{\sigma^2} + C^{-1}\right)^{-1}$; $\bar{\mathbf{w}} = \Sigma_{\mathbf{w}} \frac{XY^T}{\sigma^2}$.
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:

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

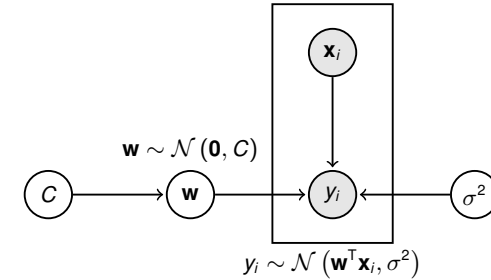
- ▶ For optimization, general forms for the gradients are available. If θ is a parameter in C :

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

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

Evidence optimisation in linear regression

Consider simple linear regression:



- ▶ Maximize

$$\begin{aligned} P(y_1 \dots y_N | \mathbf{x}_1 \dots \mathbf{x}_N, C, \sigma^2) &= \int d\mathbf{w} P(y_1 \dots y_N | \mathbf{x}_1 \dots \mathbf{x}_N, \mathbf{w}, \sigma^2) P(\mathbf{w}|C) \\ &= \int d\mathbf{w} \prod_i P(y_i | \mathbf{x}_i, \mathbf{w}, \sigma^2) P(\mathbf{w}|C) \end{aligned}$$

to find optimal values of C, σ .

- ▶ 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

$$\begin{aligned} \alpha_i^{\text{new}} &= \frac{1 - \alpha_i [\Sigma_{\mathbf{w}}]_{ii}}{\bar{\mathbf{w}}_i^2} \\ (\sigma^2)^{\text{new}} &= \frac{(Y - \bar{\mathbf{w}}^T X)(Y - \bar{\mathbf{w}}^T X)^T}{N - \sum_i (1 - [\Sigma_{\mathbf{w}}]_{ii} \alpha_i)} \end{aligned}$$

During optimization the α_i s meet one of two fates

$$\begin{aligned} \alpha_i \rightarrow \infty &\Rightarrow w_i = 0 && \text{irrelevant input } x_i \\ \alpha_i \text{ finite} &\Rightarrow w_i = \text{argmax } P(w_i | X, Y, \alpha_i) && \text{relevant input } x_i \end{aligned}$$

This procedure, **Automatic Relevance Determination** (ARD), yields **sparse** solutions that improve on ML regression. (cf. L₁-regression or LASSO).

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

Prediction averaging

Sometimes, our goal is not to learn the model structure or parameters (or their posteriors); but rather to predict the (conditional) density at a new data point.

The Bayesian approach in this case should **integrate out the parameters**:

- Density (unsupervised learning): $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$

$$p(\mathbf{x}|\mathcal{D}, m) = \int d\theta p(\mathbf{x}|\theta, m)p(\theta|\mathcal{D}, m)$$

- Predictions (supervised learning): $\mathcal{D} = \{(\mathbf{x}_1, y_1)(\mathbf{x}_2, y_2) \dots (\mathbf{x}_n, y_n)\}$

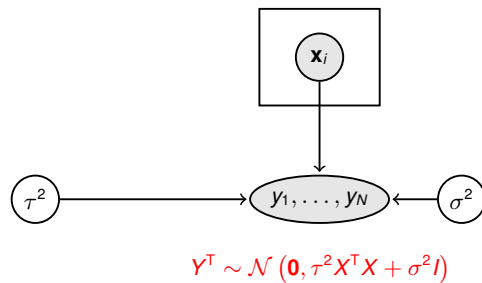
$$p(y|\mathbf{x}, \mathcal{D}, m) = \int d\theta p(y|\mathbf{x}, \theta, m)p(\theta|\mathcal{D}, m)$$

The integral naturally favours predictions associated with large volumes in the posterior, and so incorporates an Occam-like factor.

In principle, predictions may resist overfitting even with an infinitely complex model [or, put another way, the marginalised model has **no** parameters] \Rightarrow **Bayesian nonparametrics**.

Taking this approach to (non)linear regression leads to a powerful supervised learning method called **Gaussian process regression**.

Marginalised linear regression



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

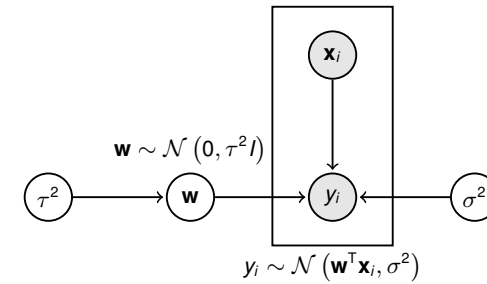
$$E[y_i] = E[\mathbf{w}^T \mathbf{x}_i] = \mathbf{0}^T \mathbf{x}_i = 0$$

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

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

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

Prediction averaging in linear regression



Let $X = [\mathbf{x}_1 \dots \mathbf{x}_N]$, $Y = [y_1 \dots y_N]$. Then (as we've seen)

$$\mathbf{w}|\mathcal{D} \sim \mathcal{N}(\bar{\mathbf{w}}, \Sigma_{\mathbf{w}})$$

where $\Sigma_{\mathbf{w}} = (\frac{1}{\sigma^2} X X^T + \frac{1}{\tau^2} I)^{-1}$ and $\bar{\mathbf{w}} = \frac{1}{\sigma^2} \Sigma_{\mathbf{w}} X Y^T$

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

$$y|\mathbf{x} \sim \mathcal{N}(\bar{\mathbf{w}}^T \mathbf{x}, \mathbf{x}^T \Sigma_{\mathbf{w}} \mathbf{x} + \sigma^2).$$

Added variance $\mathbf{x}^T \Sigma_{\mathbf{w}} \mathbf{x}$ comes from posterior uncertainty in \mathbf{w} (cf Factor Analysis).

Predictions with marginalised regression

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

$$\begin{bmatrix} Y^T \\ y \end{bmatrix} \Bigg| X, \mathbf{x} \sim \mathcal{N} \left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \tau^2 X^T X + \sigma^2 I & \tau^2 X^T \mathbf{x} \\ \tau^2 \mathbf{x}^T X & \tau^2 \mathbf{x}^T \mathbf{x} + \sigma^2 \end{bmatrix} \right) = \mathcal{N} \left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \tilde{K}_{XX} & \tilde{K}_{Xx} \\ \tilde{K}_{xX} & \tilde{K}_{xx} \end{bmatrix} \right)$$

We can find $y|Y$ by the standard multivariate Gaussian result:

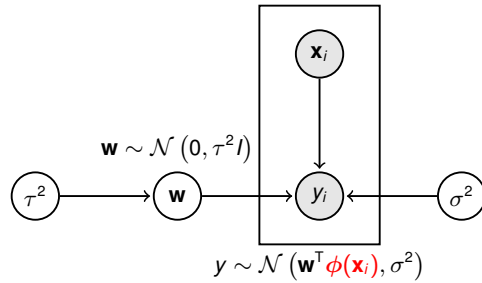
$$\begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix} \sim \mathcal{N} \left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} K_{AA} & K_{AB} \\ K_{BA} & K_{BB} \end{bmatrix} \right) \Rightarrow \mathbf{b}|\mathbf{a} \sim \mathcal{N}(K_{BA} K_{AA}^{-1} \mathbf{a}, K_{BB} - K_{BA} K_{AA}^{-1} K_{AB})$$

So

$$\begin{aligned} y|Y, X, \mathbf{x} &\sim \mathcal{N} \left(\tilde{K}_{xX} \tilde{K}_{XX}^{-1} Y^T, \tilde{K}_{xx} - \tilde{K}_{xX} \tilde{K}_{XX}^{-1} \tilde{K}_{Xx} \right) \\ &\sim \mathcal{N} \left(\tau^2 \mathbf{x}^T X (\tau^2 X^T X + \sigma^2 I)^{-1} Y^T, \tau^2 \mathbf{x}^T \mathbf{x} + \sigma^2 - \tau^2 \mathbf{x}^T X (\tau^2 X^T X + \sigma^2 I)^{-1} \tau^2 X^T \mathbf{x} \right) \\ &\sim \mathcal{N} \left(\underbrace{\mathbf{x}^T \Sigma_{\mathbf{w}} X Y^T}_{\bar{w}}, \underbrace{\tau^2 \mathbf{x}^T \mathbf{x} + \sigma^2 - \tau^2 \mathbf{x}^T X (\tau^2 X^T X + \sigma^2 I)^{-1} \tau^2 X^T \mathbf{x}}_{\Sigma_{\mathbf{w}}} \right) \quad \Sigma = \left(\frac{1}{\sigma^2} X X^T + \frac{1}{\tau^2} I \right)^{-1} \quad [\text{Matrix Inv. Lem.}] \end{aligned}$$

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

Nonlinear regression



Introduce 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 in \mathbf{x} .

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

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

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

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

$$y | \mathbf{x}, Y, X \sim \mathcal{N}(\tilde{K}_{\mathbf{x}X} \tilde{K}_{XX}^{-1} Y^T, \tilde{K}_{\mathbf{x}\mathbf{x}} - \tilde{K}_{\mathbf{x}X} \tilde{K}_{XX}^{-1} \tilde{K}_{X\mathbf{x}})$$

where, now $\tilde{K}_{XX} = \tau^2 \Phi^T \Phi + \sigma^2 I$; $\tilde{K}_{\mathbf{x}X} = \tau^2 \phi(\mathbf{x})^T \Phi$ and $\tilde{K}_{\mathbf{x}\mathbf{x}} = \tau^2 \phi(\mathbf{x})^T \phi(\mathbf{x}) + \sigma^2$.

Regression using the covariance kernel

For Bayesian regression, prediction depends on $\tilde{K}(\mathbf{x}, \mathbf{x})$ rather than explicitly on $\phi(\mathbf{x})$.

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

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

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

This is the **kernel trick**.

Prediction: compute the predictive distribution of y conditioned on Y as before:

$$y | \mathbf{x}, X, Y, \tilde{K} \sim \mathcal{N}(\tilde{K}_{\mathbf{x}X}^T \tilde{K}_{XX}^{-1} Y^T, \tilde{K}_{\mathbf{x}\mathbf{x}} - \tilde{K}_{\mathbf{x}X}^T \tilde{K}_{XX}^{-1} \tilde{K}_{X\mathbf{x}})$$

where now $[\tilde{K}_{XX}]_{ij} = \tilde{K}(\mathbf{x}_i, \mathbf{x}_j)$; $[\tilde{K}_{\mathbf{x}X}]_i = \tilde{K}(\mathbf{x}_i, \mathbf{x})$ and $\tilde{K}_{\mathbf{x}\mathbf{x}} = \tilde{K}(\mathbf{x}, \mathbf{x})$.

Evidence: given by the Gaussian likelihood:

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

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

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

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

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

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

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

The feature space \mathbb{H} might be an infinite-dimensional Hilbert space.

The Gaussian process

A covariance kernel $K(\mathbf{x}, \mathbf{x}')$ (and mean function $m(\mathbf{x})$) defined on a domain \mathbb{X} defines a **Gaussian process** (GP): a stochastic process (ie collection of random variables) on \mathbb{R} indexed by $\mathbf{x} \in \mathbb{X}$, any finite subset of which have (consistent) Gaussian distributions.

Let $f(\mathbf{x})$ be the random variable indexed by \mathbf{x} . Then a draw from the whole GP (ie for all \mathbf{x}) is a random **function** $f : \mathbb{X} \mapsto \mathbb{R}$. We write

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

where the (\cdot) 's emphasise that these are all functions.

The GP is defined such that, given a finite list of points $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$, the joint distribution of the function values $\mathbf{f} = [f(\mathbf{x}_1), \dots, f(\mathbf{x}_N)]^T$ is:

$$\mathbf{f} | X, K \sim \mathcal{N}(\mathbf{m}, K_{XX})$$

where, as usual, $[K_{XX}]_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$ and $[\mathbf{m}]_i = m(\mathbf{x}_i)$. If we enlarge or reduce the set of \mathbf{x} 's then the means and covariance matrices produced by fixed functions marginalise correctly.

For nonlinear regression, $f(\cdot)$ could instead be defined by drawing the weight vector $\mathbf{w} \in \mathbb{H}$ from the prior. But \mathbb{H} may be infinite dimensional \Rightarrow need an infinite-size object to make even a single prediction. In the GP view, each $f(\mathbf{x})$ can be drawn separately.

Regression with Gaussian processes

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

Instead of assuming a specific form, 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 are taken to be noisy versions of the (almost surely continuous) latent $f(\mathbf{x}_i)$:

$$y_i | \mathbf{x}_i, f(\cdot) \sim \mathcal{N}(f(\mathbf{x}_i), \sigma^2) \quad [\text{so } Y \sim \mathcal{N}(0, \tilde{K}_{XX}) \text{ with } \tilde{K}_{XX} = K_{XX} + \sigma^2 I]$$

Evidence: given by the multivariate Gaussian likelihood:

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

Posterior: on latent f is also a GP:

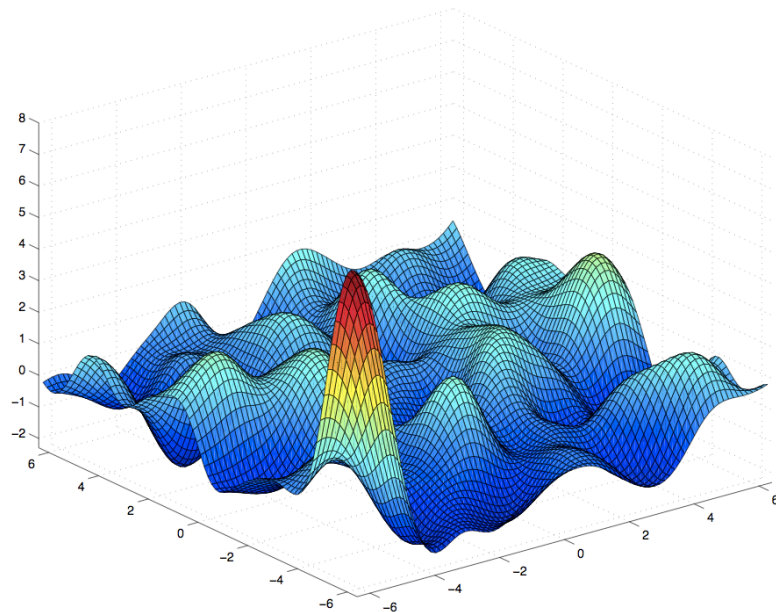
$$f(\cdot) | X, Y \sim \mathcal{GP}(K_{.x}(K_{XX} + \sigma^2 I)^{-1} Y^T, K(\cdot, \cdot) - K_{.x}(K_{XX} + \sigma^2 I)^{-1} K_{.x})$$

Predictions: posterior on f , plus observation noise:

$$y | X, Y, \mathbf{x} \sim \mathcal{N}(E[f(\mathbf{x}) | X, Y], \text{Var}[f(\mathbf{x}) | X, Y] + \sigma^2) = \mathcal{N}(K_{xx} \tilde{K}_{XX}^{-1} Y, K_{xx} \tilde{K}_{XX}^{-1} K_{xx} + \sigma^2)$$

Evidence Optimisation: gradient ascent in $\log P(Y|X)$.

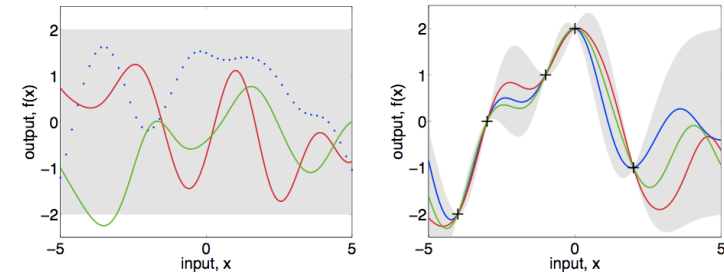
Sample from a 2D Gaussian process



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.

Example prior and posterior GPs:



Another approach is to

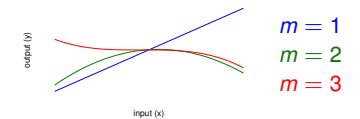
- ▶ sample $f(\mathbf{x}_1)$ first,
- ▶ then $f(\mathbf{x}_2) | f(\mathbf{x}_1)$,
- ▶ and generally $f(\mathbf{x}_n) | f(\mathbf{x}_1), \dots, f(\mathbf{x}_{n-1})$ for $n = 1, 2, \dots$

Examples of covariance kernels

- Polynomial:

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

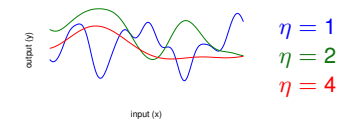
f is inhomogeneous polynomial of degree m



- Squared-exponential (or exponentiated-quadratic):

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

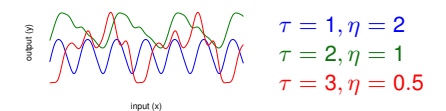
f is smooth (C^∞ almost surely) on length scale η



- Periodic (exp-sine):

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

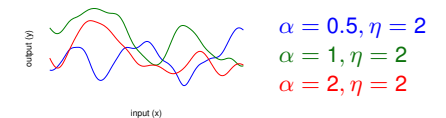
f is smooth and periodic



- Rational Quadratic:

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

f is smooth over multiple scales



Forms of 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 .

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 $f(\mathbf{x})$ (or "weights" in non-linear feature space) 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.
- ▶ Examples include the Dirichlet process (infinite mixtures), Infinite Binary Prior (infinite binary factor models), Infinite HMM . . .

GP methods

- ▶ With suitable kernels, combinations of kernels, and hyperparameter learning, GPs can identify a wide range of functional dependence. (The "automated statistician" project starts with GPs).
- ▶ With approximation, the mapping from f to y may be taken to be non-Gaussian, allowing GP classification, ordinal regression, domain-specific noise and more.
- ▶ Functions in more complex hierarchical models may be drawn from GP priors:
 - ▶ GP latent variable model (GPLVM)
 - ▶ Stacked GPs
 - ▶ Deep GP networks
- ▶ Inference and learning require inversion of K_{xx} : scales as N^3 . [Sparse](#) approximate methods reduce this to order N .
- ▶ State-of-the-art approach, particularly when data are limited.