Bayesian Inference and Gaussian Processes

MLSS 2007, Tübingen

Carl Edward Rasmussen

Department of Engineering, University of Cambridge
Max Planck Institute for Biological Cybernetics, Tübingen

August 20-31, 2007
Outline

• motivation
• Classical and Bayesian probability
• inference
• Bernoulli example
• Occam’s Razor and model selection
• testing
• summary
Why probabilistic inference in Machine Learning?

- *probabilities* provide a language for representing uncertainty
- in *learning systems* knowledge is typically uncertain
- observed data provide evidence for and against explanations
- a learner must be able to “weigh the evidence”
- probabilities are used to
  - represent knowledge
  - infer the new state of knowledge in the light of new observations
  - make optimal decisions
- probabilistic models are equivalent to other views of learning:
  - information theoretic
  - physical analogies: minimizing free energy
Interpretations of Probability

“The probability of rolling a 6 with this die is 1/6.”

What does it mean?

• classical/frequentist: probability is defined in terms of long run frequencies in repeatable experiments

“The probability that the last US soldier has left Iraq in 10 years is pretty low.”

What does it mean?

• Bayesian: probability reflects a subjective degree of belief

Depending on your interpretation, you may do inference in different ways, but the rules of probability are always the same.

Bayesian probability is used to describe all kinds of uncertainty (intrinsically stochastic things, subjective views, as well as consequences of ignorance).
Notation for probabilities

Probabilities are non-negative \( p(x) \geq 0 \)

Probabilities normalize: \( \sum_x p(x) = 1 \) (discrete) or \( \int_{-\infty}^{\infty} p(x) \, dx = 1 \) (continuous).

The joint probability of \( x \) and \( y \) is \( p(x, y) \).

The marginal probability of \( x \) is \( p(x) = \int p(x, y) \, dy \).

The conditional probability of \( x \) given \( y \) is

\[
p(x|y) = \frac{p(x, y)}{p(y)}.
\]

Bayes Rule is given by

\[
p(x, y) = p(x|y)p(y) = p(y|x)p(x) \Rightarrow p(y|x) = \frac{p(x|y)p(y)}{p(x)} \propto p(x|y)p(y).
\]
The likelihood function

The **likelihood** is the probability of the observed data given the parameters.

Example: independent coin flips.

Parameter: $\pi$ (probability of heads), tail: $x = 0$, heads: $x = 1$.

Bernoulli likelihood (for single outcome):

$$p(x|\pi) = \pi^x(1-\pi)^{1-x}.$$ 

Observations: $D = \{x_i|i = 1, \ldots, n\}$, $k$ heads.

Likelihood (using independence):

$$p(D|\pi) = \pi^k(1-\pi)^{n-k}.$$ 

Notice: the likelihood function is a probability distribution over *observations*, not over *parameters*.
Classical inference: maximum likelihood

Classical inference is based on the use of estimators.

One commonly used estimator is the maximum likelihood estimator.

Idea: if we don’t know the value of the parameter, then let’s guess the value which would maximize the probability of the data we actually observed.

In our Bernoulli experiment

$$p(\mathcal{D}|\pi) = \pi^k (1 - \pi)^{n-k}.$$ 

Maximum likelihood estimation (negative log likelihood is a cost function):

$$\frac{\partial \log p(\mathcal{D}|\pi)}{\partial \pi} = 0 \implies \pi = \frac{k}{n}.$$ 

When is this a good answer?
Bayesian Inference

Likelihood: probability of observations given parameters $p(\mathcal{D}|\pi)$

Prior: knowledge or assumptions about the parameters before we start $p(\pi)$

Posterior: knowledge about parameters after observations are made $p(\pi|\mathcal{D})$

Applying Bayes Rule:

$$p(\pi|\mathcal{D}) = \frac{p(\mathcal{D}|\pi)p(\pi)}{p(\mathcal{D})},$$

where $p(\mathcal{D})$ is called the marginal likelihood (or evidence).

Make predictions:

$$p(x^*|\mathcal{D}) = \int p(x^*|\pi)p(\pi|\mathcal{D})d\pi.$$ 

Note:
in contrast to the maximum likelihood predictions, we *average* over possible parameter settings.
Bayesian inference - model selection

If we want to compare multiple models, we can work out the posterior probability over models:

Likelihood: $p(D|\pi, H_i)$

Prior: $p(\pi|H_i)$

Posterior: $p(\pi|D, H_i)$

Applying Bayes Rule:

$$p(\pi|D, H_i) = \frac{p(D|\pi, H_i)p(\pi|H_i)}{p(D|H_i)} ,$$

where $p(D|H_i)$ is called the marginal likelihood (or evidence).

Bayes rule at the model level:

$$p(H_i|D) = \frac{p(D|H_i)p(H_i)}{p(D)}.$$
A Coin Toss Example: Comparing two Learners

Let’s compare two different models (priors):

**Learner A** believes that the coin is fair: \( \pi = \frac{1}{2} \)

**Learner B** believes that all possible values for \( \pi \) are equally plausible: \( p(\pi) = 1 \).

We can write Learner B’s beliefs using a Beta distribution, with \( \alpha = \beta = 1 \):

\[
p(\pi|\alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \pi^{\alpha-1}(1 - \pi)^{\beta-1}.
\]
Bernoulli example: observations and predictions

We observe some data: T T H H T T T H T T

What will the two learners predict?

• Learner A: Predictions: $p(x_\ast = 1|\mathcal{D}, \mathcal{H}_A) = \frac{1}{2}$
• Learner B: Posterior: $p(\pi|\mathcal{D}, \mathcal{H}_B) = \frac{\Gamma(12)}{\Gamma(4)\Gamma(8)} \pi^3 (1 - \pi)^7$.

Predictions: $p(x_\ast |\mathcal{D}, \mathcal{H}_B) = \int \pi p(\pi|\mathcal{D}, \mathcal{H}_B) d\pi = \frac{\Gamma(12)}{\Gamma(4)\Gamma(8)} \frac{\Gamma(5)\Gamma(8)}{\Gamma(13)} = \frac{4}{12} = \frac{1}{3}$. 

Beta(4,8)
Bernoulli example: model comparison

Which learner is the best model?

- **Learner A**: \( p(\mathcal{D}|\mathcal{H}_A) = \left(\frac{1}{2}\right)^{10} \approx 0.00098 \)

- **Learner B**: 
  \[
  p(\mathcal{D}|\mathcal{H}_B) = \int \pi^3 (1 - \pi)^7 d\pi = \frac{\Gamma(4)\Gamma(8)}{\Gamma(12)} \approx 0.00076.
  \]

- **Maximum likelihood estimate**: \( \pi_{\text{ML}} = \frac{3}{10} \).
  
  Maximum Likelihood Probability: 
  \[
  p(\mathcal{D}|\pi_{\text{ML}}) = \left(\frac{3}{10}\right)^3 \left(\frac{7}{10}\right)^7 \approx 0.0022.
  \]

Is this reasonable?
Why, in principle, does Bayesian Inference work? Occam’s Razor

![Graph illustrating Occam’s Razor](image-url)

- **P(Y|M_i)**
- **All possible data sets**
- **"just right"**
- **too simple**
- **too complex**
Classical testing

Example: We want to do a statistical test as to whether a new drug is helpful in treating high blood pressure.

In a clinical trial, we measure the difference in blood pressure before and after treatment, with the new drug and a placebo (randomized, double blind, etc).

Q: does the data cause us to reject the null hypothesis, $\mathcal{H}_0$, that the drug is no more efficient than the placebo, ie $\Delta = \Delta_{\text{drug}} - \Delta_{\text{placebo}} = 0$?

The classical $p$-value measures how probable are the observations, or something more extreme, given that the null hypothesis is true?

Note: in the classical test, the observations are considered random, and the unknown parameter assumed to be fixed.
Bayesian testing

Find the posterior distribution of $\Delta = \Delta_{\text{drug}} - \Delta_{\text{placebo}}$.

Report the probability that the drug works better than placebo: $p(\Delta > 0)$.

Note: in the Bayesian test, the observed data are considered deterministic, but the unknown parameter considered random.
Some vocabulary

Bayesian inference is **subjective** as it depends on **prior** information.

Bayesian inference obeys the **likelihood principle**: conclusions depend only on the likelihood of the observations (and the explicit model assumptions).

The result of Bayesian inference is the **posterior distribution**, which captures everything you know.

To make **decisions**, minimize the expected value of the **loss function** averaged over the posterior.

**Non-informative** priors are sometimes used in an attempt to construct “objective Bayes”, but this may be philosophically questionable, and often leads to technical problems.
The negative log probability is sometimes called a cost function; eg a Gaussian likelihood corresponds to a squared error loss.

Mimizing loss is then equivalent to maximizing likelihood.

Cost functions often have penalty terms or Regularizers, which can be thought analogous to a log prior.

Minimizing such a cost is sometimes called Maximum A Posteri (MAP) estimates. A better name would be penalized maximum likelihood. It is a non-Bayesian procedure.
In learning theory learning minimizes the empirical risk (loss). Clever bounds are used to upper bound the generalization error by the sum of the empirical risk and a complexity penalty.

Note: there is no notion of likelihood function here. Don’t confuse the negative log likelihood with the loss.
Bayesian inference summary

Assumptions are made explicit in the form of a prior

Predictions are made by averaging over the posterior, taking all possible interpretations of the data into account

Bayesian inference does not involve any maximization so there is no possibility of over-fitting. Instead, parameters are integrated out.

Bayesian inference is usually difficult, because it is difficult to do integrals.
The Prediction Problem

CO$_2$ concentration, ppm

year

1960 1980 2000 2020
320 340 360 380 400 420

Rasmussen (Engineering, Cambridge) Bayesian Inference and Gaussian Processes August 20-31, 2007
The Prediction Problem

CO₂ concentration, ppm

year

1960 1980 2000 2020
The Prediction Problem
The Prediction Problem

<table>
<thead>
<tr>
<th>Year</th>
<th>CO₂ concentration, ppm</th>
</tr>
</thead>
<tbody>
<tr>
<td>1960</td>
<td>320</td>
</tr>
<tr>
<td>1980</td>
<td>340</td>
</tr>
<tr>
<td>2000</td>
<td>360</td>
</tr>
<tr>
<td>2020</td>
<td>380</td>
</tr>
</tbody>
</table>

Rasmussen (Engineering, Cambridge)
The Prediction Problem

Ubiquitous questions:

- Model fitting
  - how do I fit the parameters?
  - what about overfitting?
- Model Selection
  - how to I find out which model to use?
  - how sure can I be?
- Interpretation
  - what is the accuracy of the predictions?
  - can I trust the predictions, even if
    - …I am not sure about the parameters?
    - …I am not sure of the model structure?

Gaussian processes solve some of the above, and provide a practical framework to address the remaining issues.
Outline

Part I: foundations

• What is a Gaussian process
  • from distribution to process
  • distribution over functions
  • the marginalization property
• Inference
  • Bayesian inference
  • posterior over functions
  • predictive distribution
  • marginal likelihood
  • Occam’s Razor
  • automatic complexity penalty

Part II: advanced topics

• Example
  • priors over functions
  • hierarchical priors using hyperparameters
  • learning the covariance function
• Approximate methods for classification
• Gaussian Process latent variable models
• Sparse methods
The Gaussian distribution is given by

\[
p(x|\mu, \Sigma) = N(\mu, \Sigma) = (2\pi)^{-D/2}|\Sigma|^{-1/2} \exp \left( -\frac{1}{2}(x - \mu)^\top \Sigma^{-1}(x - \mu) \right)
\]

where \( \mu \) is the mean vector and \( \Sigma \) the covariance matrix.
Both the **conditionals** and the **marginals** of a joint Gaussian are again Gaussian.
A Gaussian process is a generalization of a multivariate Gaussian distribution to infinitely many variables.

Informally: infinitely long vector $\simeq$ function

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

A Gaussian distribution is fully specified by a mean vector, $\mu$, and covariance matrix $\Sigma$:

$$f = (f_1, \ldots, f_n) \sim \mathcal{N}(\mu, \Sigma), \text{ indexes } i = 1, \ldots, n$$

A Gaussian process is fully specified by a mean function $m(x)$ and covariance function $k(x, x')$:

$$f(x) \sim \mathcal{GP}(m(x), k(x, x')), \text{ indexes: } x$$
The marginalization property

Thinking of a GP as a Gaussian distribution with an infinitely long mean vector and an infinite by infinite covariance matrix may seem impractical...

...luckily we are saved by the *marginalization property*:

Recall:

\[ p(x) = \int p(x, y) dy. \]

For Gaussians:

\[ p(x, y) = \mathcal{N}\left( \begin{bmatrix} a \\ b \end{bmatrix}, \begin{bmatrix} A & B \\ B^\top & C \end{bmatrix} \right) \implies p(x) = \mathcal{N}(a, A) \]
Random functions from a Gaussian Process

Example one dimensional Gaussian process:

\[ p(f(x)) \sim \mathcal{GP}(m(x) = 0, \ k(x, x') = \exp(-\frac{1}{2}(x - x')^2)). \]

To get an indication of what this distribution over functions looks like, focus on a finite subset of function values \( f = (f(x_1), f(x_2), \ldots, f(x_n))^\top \), for which

\[ f \sim \mathcal{N}(0, \Sigma), \]

where \( \Sigma_{ij} = k(x_i, x_j) \).

Then plot the coordinates of \( f \) as a function of the corresponding \( x \) values.
Some values of the random function

<table>
<thead>
<tr>
<th>input, x</th>
<th>output, f(x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-5</td>
<td>-5</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1.5</td>
</tr>
<tr>
<td>-1.5</td>
<td>-1.5</td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>-0.5</td>
<td>-0.5</td>
</tr>
<tr>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1.5</td>
<td>1.5</td>
</tr>
</tbody>
</table>
Sequential Generation

Factorize the joint distribution

\[ p(f_1, \ldots, f_n|x_1, \ldots, x_n) = \prod_{i=1}^{n} p(f_i|f_{i-1}, \ldots, f_1, x_i, \ldots, x_1), \]

and generate function values sequentially.

What do the individual terms look like? For Gaussians:

\[ p(x, y) = \mathcal{N}( \begin{bmatrix} a \\ b \end{bmatrix}, \begin{bmatrix} A & B \\ B^\top & C \end{bmatrix}) \implies p(x|y) = \mathcal{N}(a+BC^{-1}(y-b), A-BC^{-1}B^\top) \]

Do try this at home!
Function drawn at random from a Gaussian Process with Gaussian covariance.
Maximum likelihood, parametric model

Supervised parametric learning:

- data: \( x, y \)
- model: \( y = f_w(x) + \varepsilon \)

Gaussian likelihood:

\[
p(y|x, w, M_i) \propto \prod_c \exp\left(-\frac{1}{2}(y_c - f_w(x_c))^2 / \sigma_{\text{noise}}^2\right).
\]

Maximize the likelihood:

\[
w_{ML} = \arg\max_w p(y|x, w, M_i).
\]

Make predictions, by plugging in the ML estimate:

\[
p(y^*|x^*, w_{ML}, M_i)
\]
Bayesian Inference, parametric model

Supervised parametric learning:

- data: $x, y$
- model: $y = f_w(x) + \varepsilon$

Gaussian likelihood:

$$p(y|x, w, M_i) \propto \prod_c \exp\left(-\frac{1}{2} (y_c - f_w(x_c))^2 / \sigma_{\text{noise}}^2 \right).$$

Parameter prior:

$$p(w|M_i)$$

Posterior parameter distribution by Bayes rule $p(a|b) = p(b|a)p(a)/p(b)$:

$$p(w|x, y, M_i) = \frac{p(w|M_i)p(y|x, w, M_i)}{p(y|x, M_i)}$$
Bayesian Inference, parametric model, cont.

Making predictions:

\[ p(y^* | x^*, x, y, M_i) = \int p(y^* | w, x^*, M_i) p(w | x, y, M_i) dw \]

Marginal likelihood:

\[ p(y | x, M_i) = \int p(w | M_i) p(y | x, w, M_i) dw. \]

Model probability:

\[ p(M_i | x, y) = \frac{p(M_i) p(y | x, M_i)}{p(y | x)} \]

Problem: integrals are intractable for most interesting models!
Non-parametric Gaussian process models

In our non-parametric model, the “parameters” is the function itself!

Gaussian likelihood:

\[ y | x, f(x), M_i \sim \mathcal{N}(f, \sigma^2_{\text{noise}} I) \]

(Zero mean) Gaussian process prior:

\[ f(x) | M_i \sim \mathcal{GP}(m(x) \equiv 0, \ k(x, x')) \]

Leads to a Gaussian process posterior

\[
\begin{align*}
  f(x) | x, y, M_i & \sim \mathcal{GP}(m_{\text{post}}(x) = k(x, x)[K(x, x) + \sigma^2_{\text{noise}} I]^{-1} y, \\
  k_{\text{post}}(x, x') & = k(x, x') - k(x, x)[K(x, x) + \sigma^2_{\text{noise}} I]^{-1} k(x, x').
\end{align*}
\]

And a Gaussian predictive distribution:

\[
\begin{align*}
  y^* | x^*, x, y, M_i & \sim \mathcal{N}(k(x^*, x) \top [K + \sigma^2_{\text{noise}} I]^{-1} y, \\
  k(x^*, x^*) + \sigma^2_{\text{noise}} - k(x^*, x) \top [K + \sigma^2_{\text{noise}} I]^{-1} k(x^*, x))
\end{align*}
\]
Prior and Posterior

Predictive distribution:

\[ p(y^*|x^*, x, y) \sim \mathcal{N}(k(x^*, x)^\top [K + \sigma_{\text{noise}}^2 I]^{-1} y, \]
\[ k(x^*, x^*) + \sigma_{\text{noise}}^2 - k(x^*, x)^\top [K + \sigma_{\text{noise}}^2 I]^{-1} k(x^*, x)) \]
Square nodes are observed (clamped), round nodes stochastic (free).

All pairs of latent variables are connected.

Predictions $y^*$ depend only on the corresponding single latent $f^*$.

Notice, that adding a triplet $x_m^*, f_m^*, y_m^*$ does not influence the distribution. This is guaranteed by the marginalization property of the GP.

This explains why we can make inference using a finite amount of computation!
Some interpretation

Recall our main result:

\[ f_\ast|X_\ast, X, y \sim \mathcal{N}(K(X_\ast, X)[K(X, X) + \sigma_n^2 I]^{-1}y, \\
K(X_\ast, X_\ast) - K(X_\ast, X)[K(X, X) + \sigma_n^2 I]^{-1}K(X, X_\ast)) \].

The mean is linear in two ways:

\[ \mu(x_\ast) = k(x_\ast, X)[K(X, X) + \sigma_n^2 I]^{-1}y = \sum_{c=1}^{n} \beta_c y^{(c)} = \sum_{c=1}^{n} \alpha_c k(x_\ast, x^{(c)}). \]

The last form is most commonly encountered in the kernel literature.

The variance is the difference between two terms:

\[ V(x_\ast) = k(x_\ast, x_\ast) - k(x_\ast, X)[K(X, X) + \sigma_n^2 I]^{-1}k(X, x_\ast), \]

the first term is the \textit{prior variance}, from which we subtract a (positive) term, telling how much the data \( X \) has explained. Note, that the variance is independent of the observed outputs \( y \).
The marginal likelihood

Log marginal likelihood:

$$\log p(y|x, M_i) = -\frac{1}{2} y^\top K^{-1} y - \frac{1}{2} \log |K| - \frac{n}{2} \log(2\pi)$$

is the combination of a data fit term and complexity penalty. Occam’s Razor is automatic.

Learning in Gaussian process models involves finding

• the form of the covariance function, and
• any unknown (hyper-) parameters $\theta$.

This can be done by optimizing the marginal likelihood:

$$\frac{\partial \log p(y|x, \theta, M_i)}{\partial \theta_j} = \frac{1}{2} y^\top K^{-1} \frac{\partial K}{\partial \theta_j} K^{-1} y - \frac{1}{2} \text{trace}(K^{-1} \frac{\partial K}{\partial \theta_j})$$
Example: Fitting the length scale parameter

Parameterized covariance function: \[ k(x, x') = v^2 \exp\left( -\frac{(x - x')^2}{2\ell^2} \right) + \sigma_n^2 \delta_{xx'} \].

The mean posterior predictive function is plotted for 3 different length scales (the green curve corresponds to optimizing the marginal likelihood). Notice, that an almost exact fit to the data can be achieved by reducing the length scale – but the marginal likelihood does not favour this!
Why, in principle, does Bayesian Inference work? Occam’s Razor

A simple model (green) is preferred over a more complex model (blue) because it is more likely to generate the observed data (Y) given the model (Mi).

"Just right" model (red) achieves a balance between simplicity and complexity, providing the best fit to the data.

P(Y|M_i) represents the probability of observing the data Y given model M_i.
An illustrative analogous example

Imagine the simple task of fitting the variance, $\sigma^2$, of a zero-mean Gaussian to a set of $n$ scalar observations.

The log likelihood is

$$\log p(y|\mu, \sigma^2) = -\frac{1}{2} \sum (y_i - \mu)^2 / \sigma^2 - \frac{n}{2} \log(\sigma^2) - \frac{n}{2} \log(2\pi)$$
Consider the class of linear functions:

\[ f(x) = ax + b, \quad \text{where } a \sim \mathcal{N}(0, \alpha), \quad \text{and } b \sim \mathcal{N}(0, \beta). \]

We can compute the mean function:

\[ \mu(x) = E[f(x)] = \int \int f(x)p(a)p(b)da\,db = \int axp(a)da + \int bp(b)db = 0, \]

and covariance function:

\[ k(x, x') = E[(f(x) - 0)(f(x') - 0)] = \int \int (ax + b)(ax' + b)p(a)p(b)da\,db \]

\[ = \int a^2 xx' p(a)da + \int b^2 p(b)db + (x + x') \int abp(a)p(b)da\,db = \alpha xx' + \beta. \]
From random functions to covariance functions II

Consider the class of functions (sums of squared exponentials):

\[
f(x) = \lim_{n \to \infty} \frac{1}{n} \sum_{i} \gamma_i \exp(-(x - i/n)^2), \quad \text{where } \gamma_i \sim \mathcal{N}(0, 1), \ \forall i
\]

\[
= \int_{-\infty}^{\infty} \gamma(u) \exp(-(x - u)^2)du, \quad \text{where } \gamma(u) \sim \mathcal{N}(0, 1), \ \forall u.
\]

The mean function is:

\[
\mu(x) = E[f(x)] = \int_{-\infty}^{\infty} \exp(-(x - u)^2) \int_{-\infty}^{\infty} \gamma p(\gamma) d\gamma du = 0,
\]

and the covariance function:

\[
E[f(x)f(x')] = \int \exp \left( -(x - u)^2 - (x' - u)^2 \right) du
\]

\[
= \int \exp \left( -2(u - \frac{x + x'}{2})^2 + \frac{(x + x')^2}{2} - x^2 - x'^2 \right) du \propto \exp \left( -\frac{(x - x')^2}{2} \right).
\]

Thus, the squared exponential covariance function is equivalent to regression using infinitely many Gaussian shaped basis functions placed everywhere, not just at your training points!
Using finitely many basis functions may be dangerous!
Model Selection in Practise; Hyperparameters

There are two types of task: *form* and *parameters* of the covariance function.

Typically, our prior is too weak to quantify aspects of the covariance function. We use a **hierarchical model** using hyperparameters. Eg, in ARD:

$$k(x, x') = v_0^2 \exp\left(-\sum_{d=1}^{D} \frac{(x_d - x'_d)^2}{2v_d^2}\right), \quad \text{hyperparameters } \theta = (v_0, v_1, \ldots, v_d, \sigma^2_n).$$

- $v_1=v_2=1$
- $v_1=v_2=0.32$
- $v_1=0.32$ and $v_2=1$
Rational quadratic covariance function

The *rational quadratic* (RQ) covariance function:

\[
k_{\text{RQ}}(r) = \left(1 + \frac{r^2}{2\alpha \ell^2}\right)^{-\alpha}
\]

with \(\alpha, \ell > 0\) can be seen as a *scale mixture* (an infinite sum) of squared exponential (SE) covariance functions with different characteristic length-scales.

Using \(\tau = \ell^{-2}\) and \(p(\tau|\alpha, \beta) \propto \tau^{\alpha-1} \exp\left(-\frac{\alpha \tau}{\beta}\right)\):

\[
k_{\text{RQ}}(r) = \int p(\tau|\alpha, \beta) k_{\text{SE}}(r|\tau) d\tau \\
\propto \int \tau^{\alpha-1} \exp\left(-\frac{\alpha \tau}{\beta}\right) \exp\left(-\frac{\tau r^2}{2}\right) d\tau \propto \left(1 + \frac{r^2}{2\alpha \ell^2}\right)^{-\alpha},
\]
The limit $\alpha \to \infty$ of the RQ covariance function is the SE.
Matérn covariance functions

Stationary covariance functions can be based on the Matérn form:

\[
k(x, x') = \frac{1}{\Gamma(\nu)2^{\nu-1}} \left[ \frac{\sqrt{2\nu}}{\ell} |x - x'| \right]^\nu K_{\nu} \left( \frac{\sqrt{2\nu}}{\ell} |x - x'| \right),
\]

where \( K_{\nu} \) is the modified Bessel function of second kind of order \( \nu \), and \( \ell \) is the characteristic length scale.

Sample functions from Matérn forms are \( \lfloor \nu - 1 \rfloor \) times differentiable. Thus, the hyperparameter \( \nu \) can control the degree of smoothness.

Special cases:

- \( k_{\nu=\frac{1}{2}}(r) = \exp(-\frac{r}{\ell}) \): Laplacian covariance function, Browninan motion (Ornstein-Uhlenbeck)
- \( k_{\nu=\frac{3}{2}}(r) = \left(1 + \frac{\sqrt{3}r}{\ell}\right) \exp\left(-\frac{\sqrt{3}r}{\ell}\right) \) (once differentiable)
- \( k_{\nu=\frac{5}{2}}(r) = \left(1 + \frac{\sqrt{5}r}{\ell} + \frac{5r^2}{3\ell^2}\right) \exp\left(-\frac{\sqrt{5}r}{\ell}\right) \) (twice differentiable)
- \( k_{\nu \to \infty} = \exp(-\frac{r^2}{2\ell^2}) \): smooth (infinitely differentiable)
Matérn covariance functions II

Univariate Matérn covariance function with unit characteristic length scale and unit variance:

![Graph showing covariance function and sample functions.](image)

The graph on the left illustrates the covariance function for different values of \( \nu \) (\( \nu = 1/2, 1, 2, \nu \rightarrow \infty \)) as a function of the input distance. The graph on the right shows sample functions corresponding to these covariance functions.
Periodic, smooth functions

To create a distribution over periodic functions of \( x \), we can first map the inputs to \( u = (\sin(x), \cos(x))^\top \), and then measure distances in the \( u \) space. Combined with the SE covariance function, which characteristic length scale \( \ell \), we get:

\[
k_{\text{periodic}}(x, x') = \exp\left(-2 \sin^2\left(\pi(x - x')\right)/\ell^2\right)
\]

Three functions drawn at random; left \( \ell > 1 \), and right \( \ell < 1 \).
The Prediction Problem

![Graph showing the\nCO\textsubscript{2} concentration, ppm, over\nthe years 1960 to 2020.](image)

- **Rasmussen (Engineering, Cambridge)**
- **Bayesian Inference and Gaussian Processes**
- **August 20-31, 2007**
Covariance Function

The covariance function consists of several terms, parameterized by a total of 11 hyperparameters:

- long-term smooth trend (squared exponential)
  \[ k_1(x, x') = \theta_1^2 \exp(- (x - x')^2 / \theta_2^2), \]

- seasonal trend (quasi-periodic smooth)
  \[ k_2(x, x') = \theta_3^2 \exp \left( -2 \sin^2(\pi(x - x')) / \theta_5^2 \right) \times \exp \left( -\frac{1}{2} (x - x')^2 / \theta_4^2 \right), \]

- short- and medium-term anomaly (rational quadratic)
  \[ k_3(x, x') = \theta_6^2 \left( 1 + \frac{(x - x')^2}{2 \theta_8 \theta_7^2} \right)^{-\theta_8} \]

- noise (independent Gaussian, and dependent)
  \[ k_4(x, x') = \theta_9^2 \exp \left( -\frac{(x - x')^2}{2 \theta_{10}^2} \right) + \theta_{11}^2 \delta_{xx'}. \]

\[ k(x, x') = k_1(x, x') + k_2(x, x') + k_3(x, x') + k_4(x, x') \]

Let’s try this with the gpml software (http://www.gaussianprocess.org/gpml).
Long- and medium-term mean predictions

CO₂ concentration, ppm

320 340 360 380 400

year

−1 −0.5 0 0.5 1

CO₂ concentration, ppm

Rasmussen (Engineering, Cambridge)
Seasonal component: magnitude $\theta_3 = 2.4$ ppm, decay-time $\theta_4 = 90$ years.

Dependent noise, magnitude $\theta_9 = 0.18$ ppm, decay $\theta_{10} = 1.6$ months. Independent noise, magnitude $\theta_{11} = 0.19$ ppm.

Optimize or integrate out? See MacKay [5].
Spline models

One dimensional minimization problem: find the function $f(x)$ which minimizes:

$$
\sum_{i=1}^{c} (f(x^{(i)}) - y^{(i)})^2 + \lambda \int_0^1 (f''(x))^2 \, dx,
$$

where $0 < x^{(i)} < x^{(i+1)} < 1$, $\forall i = 1, \ldots, n - 1$, has as solution the **Natural Smoothing Cubic Spline**: first order polynomials when $x \in [0; x^{(1)}]$ and when $x \in [x^{(n)}; 1]$ and a cubic polynomical in each $x \in [x^{(i)}; x^{(i+1)}]$, $\forall i = 1, \ldots, n - 1$, joined to have continuous second derivatives at the knots.

The identical function is also the mean of a Gaussian process: Consider the class of a functions given by:

$$
f(x) = \alpha + \beta x + \lim_{n \to \infty} \frac{1}{\sqrt{n}} \sum_{i=0}^{n-1} \gamma_i (x - \frac{i}{n})_+,
$$

where $(x)_+ = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$

with Gaussian priors:

$$
\alpha \sim \mathcal{N}(0, \xi), \quad \beta \sim \mathcal{N}(0, \xi), \quad \gamma_i \sim \mathcal{N}(0, \Gamma), \quad \forall i = 0, \ldots, n - 1.
$$
Spline models, continued

The covariance function becomes:

\[ k(x, x') = \xi + xx'\xi + \Gamma \lim_{n \to \infty} \frac{1}{n} \sum_{i=0}^{n-1} (x - \frac{i}{n}) + (x' - \frac{i}{n}) + \]

\[ = \xi + xx'\xi + \Gamma \int_0^1 (x - u)_+ (x' - u)_+ du \]

\[ = \xi + xx'\xi + \Gamma \left( \frac{1}{2} |x - x'| \min(x, x')^2 + \frac{1}{3} \min(x, x')^3 \right). \]

In the limit \( \xi \to \infty \) and \( \lambda = \sigma_n^2 / \Gamma \) the posterior mean is the natural cubic spline.

We can thus find the hyperparameters \( \sigma^2 \) and \( \Gamma \) (and thereby \( \lambda \)) by maximising the marginal likelihood in the usual way. Defining \( h(x) = (1, x)^\top \) the posterior predictions with mean and variance:

\[
\tilde{\mu}(X_*) = H(X_*)^\top \beta + K(X, X_*)[K(X, X) + \sigma_n^2 I]^{-1}(y - H(X)^\top \beta)
\]

\[
\tilde{\Sigma}(x_*) = \Sigma(X_*) + R(X, X_*)^\top A(X)^{-1} R(X, X_*)
\]

\[
\beta = A(X)^{-1}H(X)[K + \sigma_n^2 I]^{-1}y,
\]

\[
A(X) = H(X)[K(X, X) + \sigma_n^2 I]^{-1}H(X)^\top
\]

\[
R(X, X_*) = H(X_*) - H(X)[K + \sigma_n^2 I]^{-1}K(X, X_*)
\]
Cubic Splines, Example

Although this is not the fastest way to compute splines, it offers a principled way of finding hyperparameters, and uncertainties on predictions.

Note also, that although the posterior mean is smooth (piecewise cubic), posterior sample functions are not.
Gaussian Processes for Monte Carlo

Evaluation of integrals in an important problem, especially in Bayesian statistics and machine learning:

\[ \bar{f}_p = \int f(x)p(x)dx, \]

where \( p(x) \) is a probability (density). Examples:

- **Prediction:**
  \[ p(y|\mathcal{D}) = \int p(y|\mathcal{D}, \theta)p(\theta|\mathcal{D})d\theta \]

- **Marginal likelihood (evidence) for model comparison:**
  \[ p(\mathcal{D}|\mathcal{M}) = \int p(\mathcal{D}|\theta, \mathcal{M})p(\theta|\mathcal{M})d\theta \]

- **Integration over latent or missing variables:**
  \[ p(y|\theta) = \int p(y|x, \theta)p(x|\theta)dx \]

Monte Carlo is often used for analytically intractable integrals.
Classical Monte Carlo

Problem: Evaluate

\[ \bar{f}_p = \int f(x)p(x)dx, \]

where \( f(x) \) is a function and \( p(x) \) is a probability (density).
Assume that we are able to: 1) evaluate \( f(x) \) at any \( x \) and 2) either evaluate \( p(x) \) or generate randomly from \( p(x) \).

In simple Monte Carlo we use:

\[ \bar{f}_p \approx \frac{1}{T} \sum_{t=1}^{T} f(x^{(t)}), \quad \text{where } x^{(t)} \sim p(x). \]

or in Importance Sampling

\[ \bar{f}_p \approx \frac{1}{T} \sum_{t=1}^{T} f(x^{(t)}) \frac{p(x^{(t)})}{q(x^{(t)})}, \quad \text{where } x^{(t)} \sim q(x). \]

These Monte Carlo procedures are purely frequentist!
Monte Carlo is based on the arbitrary choice of sampling distribution. Given the 
\textit{same} points $x$, i.e. \textit{same} information about $f$ and $p$, you get different results 
depending on $q(x)$!

Monte Carlo ignores relevant information: the $x$ values.
Problems with classical Monte Carlo II

Monte carlo makes no assumptions about $f$, not even smoothness or continuity. Hard to incorporate prior knowledge.

“Monte Carlo is fundamentally unsound” [O’Hagan 1987]
The Bayesian Interpretation

Classically, we think of $f_p$ as a fixed but unknown quantity.

A Bayesian describes all kinds of uncertainty using probability:

- $f_p$ is unknown, therefore treated as a random variable
- $f(x = x_0)$ is also unknown until we evaluate it

Recipe:

- Specify a prior over functions $p(f)$,
- combine it with observations:
  $$\mathcal{D} = \{ x^{(c)}, f(x^{(c)}) | c = 1, \ldots, n \}$$
- to get the posterior over functions, $p(f|\mathcal{D})$,
- this in turn implies a distribution over the desired $f_p$. 

Rasmussen (Engineering, Cambridge) Bayesian Inference and Gaussian Processes August 20-31, 2007 65 / 86
If \( f \) is a GP then, since the integral \( \int f(x)p(x)dx \) is just an infinite dimensional projection, \( p(f|\mathcal{D}) \) is Gaussian, with mean \( \mu \) and variance \( \sigma^2 \):

\[
\mu = \int \left[ \int f(x)p(x)dx \right] p(f|\mathcal{D})df = \int \left[ \int f(x)p(f|\mathcal{D})df \right] p(x)dx = \int \bar{f}_\mathcal{D}(x)p(x)dx,
\]

\[
\sigma^2 = \int \left[ \int f(x)p(x)dx - \int \bar{f}_\mathcal{D}(x)p(x)dx \right]^2 p(f|\mathcal{D})df = \int \int \left[ f(x) - \bar{f}(x) \right] \left[ f(x') - \bar{f}(x') \right] p(f|\mathcal{D})dfp(x)p(x')dx dx' = \int \int \text{Cov}_\mathcal{D}(f(x), f(x')) p(x)p(x') dx dx',
\]
where posterior mean and covariance are given by standard results for GPs:

$$
\tilde{f}_D(x) = Q(x, x)Q(x, x)^{-1}f
$$

$$
\text{Cov}_D(f(x), f(x')) = Q(x, x') - Q(x, x)Q(x, x)^{-1}Q(x, x').
$$

For some common choices of $p(x)$ and $Q(x, x')$ (Gaussian or mixtures of Gaussians) these integrals are analytically tractable.

A similar approach was explored in “Bayes-Hermite Quadrature” [O’Hagan 1992].
Some Details

For the case where:
\[ p(x) \sim \mathcal{N}(b, B), \]
and the covariance function a squared exponential:
\[ Q(x, x') \propto \mathcal{N}(x', A = \text{diag}(\theta_1, \ldots, \theta_D)), \]
we have:
\[ \mu = E_{f|\mathcal{D}}[\tilde{f}_p] = z^\top Q^{-1}f, \]
\[ \sigma^2 = V_{f|\mathcal{D}}[\tilde{f}_p] = w_0 \left| 2A^{-1}B + I \right|^{-1/2} - z^\top Q^{-1}z, \]
where we have defined:
\[ z = w_0 |A^{-1}B + I|^{-1/2} \exp[-0.5(a - b)^\top (A + B)^{-1}(a - b)]. \]
Bayesian Monte Carlo Example

- Bayesian inference
- Simple Monte Carlo
- Optimal importance

Sample size vs. average squared error.
importance sampling:

\[
\hat{f}_T = \frac{1}{T} \sum_i \frac{f(x^{(t)})p(x^{(t)})}{q(x^{(t)})}
\]

where \( x^{(t)} \sim q(x) \),

where \( q(x) > 0 \) wherever \( p(x) > 0 \). The variance of this estimator is:

\[
V(\hat{f}_T) = \frac{1}{T} \left[ \int \frac{f(x)^2p(x)^2}{q(x)} dx - \bar{f}_p^2 \right]
\]

Simple calculus of variations shows that the optimal (minimum variance) importance distribution is

\[
q^*(x) = \frac{|f(x)p(x)|}{\int |f(x')p(x')| dx'}
\]
The class probability is related to the latent function, $f$, through:

$$p(y = 1|f(x)) = \pi(x) = \Phi(f(x)),$$

where $\Phi$ is a sigmoid function, such as the logistic or cumulative Gaussian. Observations are independent given $f$, so the likelihood is

$$p(y|f) = \prod_{i=1}^{n} p(y_i|f_i) = \prod_{i=1}^{n} \Phi(y_i f_i).$$
Prior and Posterior for Classification

We use a Gaussian process prior for the latent function:

\[ f|X, \theta \sim \mathcal{N}(0, K) \]

The posterior becomes:

\[
p(f|D, \theta) = \frac{p(y|f) p(f|X, \theta)}{p(D|\theta)} = \frac{\mathcal{N}(f|0, K)}{p(D|\theta)} \prod_{i=1}^{m} \Phi(y_if_i),
\]

which is non-Gaussian.

The latent value at the test point, \( f(x^*) \) is

\[
p(f^*|D, \theta, x^*) = \int p(f^*|f, X, \theta, x^*) p(f|D, \theta) df,
\]

and the predictive class probability becomes

\[
p(y^*|D, \theta, x^*) = \int p(y^*|f^*) p(f^*|D, \theta, x^*) df^*,
\]

both of which are intractable to compute.
Gaussian Approximation to the Posterior

We approximate the non-Gaussian posterior by a Gaussian:

\[ p(f|D, \theta) \simeq q(f|D, \theta) = \mathcal{N}(m, A) \]

then \( q(f_*|D, \theta, x_*) = \mathcal{N}(f_*|\mu_*, \sigma_*^2) \), where

\[
\begin{align*}
\mu_* &= k_*^\top K^{-1} m \\
\sigma_*^2 &= k(x_*, x_*) - k_*^\top (K^{-1} - K^{-1}AK^{-1})k_*.
\end{align*}
\]

Using this approximation with the cumulative Gaussian likelihood

\[
q(y_* = 1|D, \theta, x_*) = \int \Phi(f_*) \mathcal{N}(f_*|\mu_*, \sigma_*^2) df_* = \Phi\left(\frac{\mu_*}{\sqrt{1 + \sigma_*^2}}\right)
\]
Laplace’s method and Expectation Propagation

How do we find a good Gaussian approximation $\mathcal{N}(\mathbf{m}, \mathbf{A})$ to the posterior?

**Laplace’s method**: Find the Maximum A Posteriori (MAP) latent values $\mathbf{f}_{\text{MAP}}$, and use a local expansion (Gaussian) around this point as suggested by Williams and Barber [10].

**Variational bounds**: bound the likelihood by some tractable expression. A local variational bound for each likelihood term was given by Gibbs and MacKay [1]. A lower bound based on Jensen’s inequality by Opper and Seeger [7].

**Expectation Propagation**: use an approximation of the likelihood, such that the moments of the marginals of the approximate posterior match the (approximate) moment of the posterior, Minka [6].

Laplace’s method and EP were compared by Kuss and Rasmussen [3].
Gaussian process latent variable models

GP’s can be used for non-linear dimensionality reduction (unsupervised learning).

Observed (high-dimensional) data $Y_{dc}$, where $1 \leq d \leq D$ indexes dimensions and $1 \leq c \leq n$ indexes dimensions.

Assume that each visible coordinate, $y_d$, is modeled by a separate GP using some latent (low dimensional) inputs $x$.

Find the best latent inputs by maximizing the marginal likelihood under the constraint that all visible variables must share the same latent values.

Computationally, this isn’t too expensive, as all dimensions are modeled using the same covariance matrix $K$.

This is the GPLVM model proposed by Lawrence [4].
Gaussian process latent variable models

Finding the latent variables is a high-dimensional, non-linear, optimization problem with local optima.

GPLVM defines a map from latent to observed space, not a generative model.

Mapping new latent coordinates to (distributions over) observations is easy.

Finding the latent coordinates (pre-image) for new cases is difficult.

Motion capture example, representing 102-D data in 2-D, borrowed from Neil Lawrence.
Sparse Approximations

Recall the graphical model for a Gaussian process. Inference is expensive because the latent variables are fully connected.

Exact inference: $\mathcal{O}(n^3)$.

Sparse approximations: solve a smaller, sparse, approximation of the original problem.

Algorithm: Subset of data.

Are there better ways to sparsify?
Inducing Variables

Because of the marginalization property, we can introduce more latent variables without changing the distribution of the original variables.

The \( \mathbf{u} = (u_1, u_2, \ldots)^\top \) are called inducing variables.

The inducing variables have associated inducing inputs, \( \mathbf{s} \), but no associated output values.

The marginalization property ensures that

\[
p(f, f^*) = \int p(f, f^*, u) du
\]
The Central Approximations

In a unifying treatment, Candela and Rasmussen [2] assume that training and test sets are *conditionally independent* given $u$.

Assume: $p(f, f_*) \simeq q(f, f_*)$, where

$$q(f, f_*) = \int q(f_* | u) q(f | u) p(u) du.$$  

The inducing variables *induce* the dependencies between training and test cases.

Different sparse algorithms in the literature correspond to different
- choices of the inducing inputs
- further approximations
Training and test conditionals

The exact training and test conditionals are:

\[
p(f|u) = \mathcal{N}(K_{f,u}K_{f,f}^{-1}u, K_{f,f} - Q_{f,f})
\]

\[
p(f_*|u) = \mathcal{N}(K_{f_*,u}K_{f,f}^{-1}u, K_{f_*,f_*} - Q_{f_*,f_*}),
\]

where \(Q_{a,b} = K_{a,u}K_{u,u}^{-1}K_{u,b}\).

These equations are easily recognized as the usual predictive equations for GPs.

The effective prior is:

\[
q(f, f_*) = \mathcal{N}(0, \begin{bmatrix} K_{f,f} & Q_{*,f} \\ Q_{f,*} & K_{*,*} \end{bmatrix})
\]
Example: Subset of Regressors

Replace both training and test conditionals by *deterministic* relations:

\[
q(f|u) = \mathcal{N}(K_{f,u}K^{-1}_{f,f}u, 0)
\]

\[
q(f_*|u) = \mathcal{N}(K_{f_*,u}K^{-1}_{f,f}u, 0).
\]

The effective prior becomes

\[
q_{SOR}(f, f_*) = \mathcal{N}(0, \begin{bmatrix} Q_{f,f} & Q_{f_*,f} \\ Q_{f,*} & Q_{*,*} \end{bmatrix}),
\]

showing that SOR is just a GP with (degenerate) covariance function \( Q \).
Example: Sparse parametric Gaussian processes

Snelson and Ghahramani [8] introduced the idea of sparse GP inference based on a pseudo data set, integrating out the targets, and optimizing the inputs.

Equivalently, in the unifying scheme:

\[
q(f|u) = \mathcal{N}(K_{f,u}K_{f,f}^{-1}u, \text{diag}[K_{f,f} - Q_{f,f}])
\]

\[
q(f_*|u) = p(f_*|u).
\]

The effective prior becomes

\[
q_{\text{FITC}}(f, f_*) = \mathcal{N}(0, \begin{bmatrix} Q_{f,f} - \text{diag}[Q_{f,f} - K_{f,f}] & Q_{*,f} \\ Q_{f,*} & K_{*,*} \end{bmatrix}),
\]

which can be computed efficiently.

The Bayesian Committee Machine [9] uses block diag instead of diag, and the inducing variables to be the test cases.
Sparse approximations

Most published sparse approximations can be understood in a single graphical model framework.

The *inducing inputs* (or expansion points, or support vectors) may be a subset of the training data, or completely free.

The approximations are understood as exact inference in a modified model (rather than approximate inference for the exact model).
Conclusions

Complex non-linear inference problems can be solved by manipulating plain old Gaussian distributions

- Bayesian inference is tractable for GP regression and
- Approximations exist for classification
- Predictions are probabilistic
- Compare different models (via the marginal likelihood)

GPs are a simple and intuitive means of specifying prior information, and explaining data, and equivalent to other models: RVM’s, splines, closely related to SVMs.

Outlook:

- New interesting covariance functions
- Application to structured data
- Better understanding of sparse methods
More on Gaussian Processes

Rasmussen and Williams
Gaussian Processes for Machine Learning,

Gaussian process web (code, papers, etc): http://www.GaussianProcess.org
A few references


