Probabilistic & Unsupervised Learning

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 2012
Bayesian Linear Regression

Given observed data $D = \{X = [x_1 \ldots x_N], Y = [y_1 \ldots y_N]\}$, the posterior on $w$ is:

$$w|D \sim \mathcal{N}\left(\frac{1}{\sigma^2} \sum_w X Y^T, \left(\frac{1}{\sigma^2} X X^T + \frac{1}{\tau^2} I\right)^{-1}\right)$$

The Bayesian predictive distribution for $y'|x'$ is obtained by integrating out $w$:

$$p(y'|x', D) = \int dw \ p(y'|w, x')p(w|D)$$

$$= \int dw \ \mathcal{N}(y'|w^T x', \sigma^2) \ \mathcal{N}(w|\mu_w, \Sigma_w)$$

$$= \mathcal{N}(\mu_{w^T x'}, x'|\Sigma_w x + \sigma^2).$$
**Alternative View of Linear Regression**

\[ w \sim N(0, \tau^2 I) \]
\[ y_i \sim N(w^T x_i, \sigma^2) \]

\[ y \sim N(0, \tau^2 XX^T + \sigma^2 I) \]

Integrate out \( w \): the joint distribution of \( y_1, \ldots, y_N \) given \( x_1, \ldots, x_N \) is Gaussian. The means and covariances are:

\[
E[y_i] = E[w^T x_i] = 0^T x_i = 0
\]
\[
E[(y_i - \bar{y}_i)^2] = E[(x_i^T w)(w^T x_i)] + \sigma^2 = \tau^2 x_i^T x_i + \sigma^2
\]
\[
E[(y_i - \bar{y}_i)(y_j - \bar{y}_j)] = E[(x_i^T w)(w^T x_j)] = \tau^2 x_i^T x_j
\]

\[
\begin{bmatrix}
y_1 \\
y_2 \\
\vdots \\
y_N
\end{bmatrix}
\sim
\begin{bmatrix}
0 \\
0 \\
\vdots \\
0
\end{bmatrix}
, \quad
\begin{bmatrix}
\tau^2 x_1^T x_1 + \sigma^2 \\
\tau^2 x_1^T x_2 + \sigma^2 \\
\vdots \\
\tau^2 x_N^T x_N + \sigma^2
\end{bmatrix}
\]

\[ Y^T | X \sim N(0_N, \tau^2 XX^T + \sigma^2 I_N) \]
Alternative View of Linear Regression

Now, include the test input vector $x'$ and test output $y'$:

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

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

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

So

$$
y'|Y, X, x' \sim \mathcal{N} \left( \tau^2 x'^T X (\tau^2 X^T X + \sigma^2 I)^{-1} Y^T, \tau^2 x'^T x' + \sigma^2 - \tau^2 x'^T X (\tau^2 X^T X + \sigma^2 I)^{-1} \tau^2 X^T x' \right)
$$

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

Same answer as when we integrated posterior over $w$ to obtain predictive distribution over $y'$.

Similarly, evidence $P(Y|X)$ is just probability under Gaussian, and reduces to previous expression.

The point: Bayesian regression can be derived from a joint, parameter-free distribution on the outputs conditioned on the inputs.
Nonlinear Regression

What if we introduce a nonlinear mapping \( x \mapsto \phi(x) \)? Each element of \( \phi(x) \) is a (nonlinear) feature extracted from \( x \). May be many more features than elements on \( x \).

The regression function \( f(x) = w^T \phi(x) \) 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 \( \phi(x_i) \).

Proceeding as before, the predictive distribution over \( y' \) on a test input \( x' \) is:

\[
y'|Y, X, x' \sim \mathcal{N} \left( \tau^2 \phi(x')^T \Phi K^{-1} Y^T, \tau^2 \phi(x')^T \phi(x') + \sigma^2 - \tau^4 \phi(x)^T \Phi K^{-1} \Phi^T \phi(x') \right)
\]

\[
K = \tau^2 \Phi^T \Phi + \sigma^2 I
\]
The Covariance Kernel

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

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 \( x, x' \in \mathbb{X} \) are two input vectors with corresponding outputs \( y, y' \), then

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

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

The covariance kernel has two properties:

- **Symmetric**: \( K(x, x') = K(x', x) \) for all \( x, x' \).
- **Positive semidefinite**: the matrix \([K(x_i, x_j)]\) formed by any finite set of input vectors \( x_1, \ldots, x_N \) 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

\[
K(x, x') = \phi(x)^T \phi(x')
\]

The feature space \( \mathbb{H} \) can potentially be infinite dimensional.
Regression using the Covariance Kernel

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

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

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

where the $i, j$ entry in the covariance matrix $K(X, X)$ is $K(x_i, x_j)$.

This is called the \textit{kernel trick}.

\textbf{Prediction}: compute the predictive distribution of $y'$ conditioned on $Y$:

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

\textbf{Evidence}: this is just the Gaussian likelihood:

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

\textbf{Evidence optimisation}: the covariance kernel $K$ often has parameters, and these can be optimized by gradient ascent in $\log P(Y | X, K)$. 
A **Gaussian process** (GP) is a collection of random variables, any finite number of which have (consistent) Gaussian distributions.

In our regression setting, corresponding to each input vector \( x \) we have an output \( f(x) \). Given \( X = [x_1, \ldots, x_N] \), the joint distribution of the outputs \( F = [f(x_1), \ldots, f(x_N)] \) is:

\[
F \mid X, K \sim \mathcal{N}(0, K(X, X))
\]

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

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

\[
f(\cdot) \sim GP(m(\cdot), K(\cdot, \cdot))
\]

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

\[
F' \mid X', X, F, K \sim \mathcal{N}(K(X', X)K(X, X)^{-1}F^T, K(X', X') - K(X', X)K(X, X)^{-1}K(X, X'))
\]

thus the posterior over functions \( f(\cdot) \mid X, F \) is still a Gaussian process!
Regression with Gaussian Processes

We wish to model the joint distribution of outputs \( y_1, \ldots, y_N \) given inputs \( x_1, \ldots, x_N \).

Use a GP prior over functions:

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

Usually, instead of treating \( y_i \) as direct observation of the function value \( f(x_i) \), we add Gaussian observation noise:

\[
y_i | x_i, f(\cdot) \sim \mathcal{N}(f(x_i), \sigma^2)
\]

**Evidence**: again this is just a multivariate Gaussian likelihood,

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

**Posterior**: the posterior function is still a GP,

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

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

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

**Evidence Optimisation**: we can do this by gradient ascent in \( \log P(Y|X) \).
Samples from a Gaussian Process

We can draw sample functions from a GP by fixing a set of input vectors $x_1, \ldots, x_N$, and drawing a sample $f(x_1), \ldots, f(x_N)$ from the corresponding multivariate Gaussian. This can then be plotted.

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

Another approach is to

- sample $f(x_1)$ first,
- then $f(x_2) | f(x_1)$,
- and generally $f(x_n) | f(x_1), \ldots, f(x_{n-1})$ for $n = 1, 2, \ldots$. 
Sample from a 2D Gaussian Process
Examples of covariance Kernels

- **Polynomial:**
  \[ K(x, x') = (1 + x^T x')^m \]

- **Squared-exponential:**
  \[ K(x, x') = \theta^2 e^{-\frac{||x-x'||^2}{2\eta^2}} \]

- **Periodic (exp-sine):**
  \[ K(x, x') = \theta^2 e^{-\frac{2 \sin^2(\pi(x-x')/\tau)}{\eta^2}} \]

- **Rational Quadratic:**
  \[ K(x, x') = \left(1 + \frac{||x-x'||^2}{2\alpha\eta^2}\right)^{-\alpha} \]
  \( \alpha > 0 \)
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:** $K_1(\phi(x), \phi(x'))$ for some function $\phi$.

We say a covariance kernel is translation-invariant if

$$K(x, x') = h(x - x')$$

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

We say a covariance kernel is radial if

$$K(x, x') = h(||x - x'||)$$

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

But the Bayesian treatment integrates parameters out, so they cannot be adjusted to overfit the data! In the GP, the parameter is the function $f(x)$ which can be infinite-dimensional.

The Gaussian process is an example of a 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 “sub-model” with largest integral automatically.
- No explicit need for Occam’s razor, validation or added regularisation penalty.
Automatic relevance determination appeared in MacKay (1993) *Bayesian Methods for Back-
propagation Networks* and Neal (1993) *Bayesian Learning for Neural Networks.*
Gaussian processes can also be used in classification and latent variable models. We will
consider classification in the second half of course.

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

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

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