CSci 8980: Advanced Topics in Graphical Models

Gaussian Processes

Instructor: Arindam Banerjee

November 15, 2007
Gaussian Processes

Outline

- Parametric Bayesian Regression
- Parameters to Functions
- GP Regression
- GP Classification

We will use
- Primary: Carl Rasmussen's GP tutorial slides (NIPS'06)
- Secondary: Hanna Wallach's slides on regression
Gaussian Processes

Outline

- Parametric Bayesian Regression
Gaussian Processes

Outline
- Parametric Bayesian Regression
- Parameters to Functions
Gaussian Processes

Outline

- Parametric Bayesian Regression
- Parameters to Functions
- GP Regression

We will use

Primary: Carl Rasmussen’s GP tutorial slides (NIPS’06)
Secondary: Hanna Wallach’s slides on regression
Gaussian Processes

Outline

- Parametric Bayesian Regression
- Parameters to Functions
- GP Regression
- GP Classification

We will use
Primary: Carl Rasmussen’s GP tutorial slides (NIPS’06)
Secondary: Hanna Wallach’s slides on regression
Gaussian Processes

Outline

- Parametric Bayesian Regression
- Parameters to Functions
- GP Regression
- GP Classification

We will use

Primary: Carl Rasmussen's GP tutorial slides (NIPS'06)
Secondary: Hanna Wallach's slides on regression
Gaussian Processes

Outline

- Parametric Bayesian Regression
- Parameters to Functions
- GP Regression
- GP Classification

We will use

- Primary: Carl Rasmussen’s GP tutorial slides (NIPS’06)
Gaussian Processes

Outline
- Parametric Bayesian Regression
- Parameters to Functions
- GP Regression
- GP Classification

We will use
- Primary: Carl Rasmussen’s GP tutorial slides (NIPS’06)
- Secondary: Hanna Wallach’s slides on regression
The Prediction Problem

\[ \text{CO}_2 \text{ concentration, ppm} \]

\[ \text{year} \]

Rasmussen (MPI for Biological Cybernetics)  
Advances in Gaussian Processes  
December 4th, 2006  
3 / 55
The Prediction Problem

![Graph showing CO$_2$ concentration from 1960 to 2020](chart.png)
The Prediction Problem

\[ \text{CO}_2 \text{ concentration, ppm} \]

year

1960 1980 2000 2020

320
340
360
380
400
420

Rasmussen (MPI for Biological Cybernetics)
Advances in Gaussian Processes
December 4th, 2006
Maximum likelihood, parametric model

Supervised parametric learning:

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

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:

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

Parameter prior:

$$p(w|M_i)$$

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

$$p(w|x, y, M_i) = \frac{p(w|M_i)p(y|x, w, M_i)}{p(y|x, M_i)}$$
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!
Bayesian Linear Regression (2)

- Likelihood of parameters is:
  \[ P(y|X, w) = \mathcal{N}(X^\top w, \sigma^2 I). \]

- Assume a Gaussian prior over parameters:
  \[ P(w) = \mathcal{N}(0, \Sigma_p). \]

- Apply Bayes’ theorem to obtain posterior:
  \[ P(w|y, X) \propto P(y|X, w)P(w). \]
Bayesian Linear Regression (3)

- Posterior distribution over $\mathbf{w}$ is:

$$P(\mathbf{w}|\mathbf{y}, \mathbf{X}) = \mathcal{N}(\frac{1}{\sigma^2}A^{-1}\mathbf{X}\mathbf{y}, A^{-1}) \text{ where } A = \Sigma_p^{-1} + \frac{1}{\sigma^2} \mathbf{X}\mathbf{X}^\top.$$ 

- Predictive distribution is:

$$P(f^*|\mathbf{x}^*, \mathbf{X}, \mathbf{y}) = \int f(\mathbf{x}^*|\mathbf{w})P(\mathbf{w}|\mathbf{X}, \mathbf{y})d\mathbf{w}$$

$$= \mathcal{N}(\frac{1}{\sigma^2}\mathbf{x}^{*\top}A^{-1}\mathbf{X}\mathbf{y}, \mathbf{x}^{*\top}A^{-1}\mathbf{x}^*).$$
Non-parametric Gaussian process models

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

Gaussian likelihood:

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

(Zero mean) Gaussian process prior:

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

Leads to a Gaussian process posterior

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

And a Gaussian predictive distribution:

\[
\begin{align*}
  y^*|\mathbf{x}^*, \mathbf{x}, y, M_i & \sim \mathcal{N}(k(\mathbf{x}^*, \mathbf{x})^\top [K + \sigma^2_{\text{noise}} I]^{-1}y, \\
  k(\mathbf{x}^*, \mathbf{x}^*) + \sigma^2_{\text{noise}} - k(\mathbf{x}^*, \mathbf{x})^\top [K + \sigma^2_{\text{noise}} I]^{-1}k(\mathbf{x}^*, \mathbf{x})).
\end{align*}
\]
The Gaussian distribution is given by

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

where \( \mu \) is the mean vector and \( \Sigma \) the covariance matrix.
Both the **conditionals** and the **marginals** of a joint Gaussian are again Gaussian.
What is a Gaussian Process?

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 \( \Sigma \):

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

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

\[
f(x) \sim \mathcal{GP}(m(x), k(x, x')) , \quad \text{indexes: } x
\]
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:

\[ p(x, y) = \mathcal{N}(\begin{bmatrix} a \\ b \end{bmatrix}, \begin{bmatrix} A & B \\ B^T & C \end{bmatrix}) \implies p(x) = \mathcal{N}(a, A) \]
Example one dimensional Gaussian process:

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

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

-5 0 5

-1.5 -1 -0.5 0 0.5 1 1.5

input, x

output, f(x)
Non-parametric Gaussian process models

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

Gaussian likelihood:

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

(Zero mean) Gaussian process prior:

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

Leads to a Gaussian process posterior

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

And a Gaussian predictive distribution:

\[
\begin{align*}
  y^*|\mathbf{x}^*, \mathbf{x}, y, M_i & \sim \mathcal{N}(k(\mathbf{x}^*, \mathbf{x})^T[K + \sigma^2_{\text{noise}}\mathbf{I}]^{-1}y, \\
  k(\mathbf{x}^*, \mathbf{x}^*) + \sigma^2_{\text{noise}} - k(\mathbf{x}^*, \mathbf{x})^T[K + \sigma^2_{\text{noise}}\mathbf{I}]^{-1}k(\mathbf{x}^*, \mathbf{x})
\end{align*}
\]
Predictive distribution:

\[
p(y^*|x^*, x, y) \sim \mathcal{N}(k(x^*, x)^	op [K + \sigma_{\text{noise}}^2 I]^{-1} y, k(x^*, x^*) + \sigma_{\text{noise}}^2 - k(x^*, x)^	op [K + \sigma_{\text{noise}}^2 I]^{-1} k(x^*, x))
\]
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:

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

The mean is linear in two ways:

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

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

The variance is the difference between two terms:

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

the first term is the 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

\[ P(Y|M_i) \]

All possible data sets

"just right"

too simple

too complex

Rasmussen (MPI for Biological Cybernetics)  Advances in Gaussian Processes  December 4th, 2006  25 / 55
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:

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

We can compute the mean function:

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

and covariance function:

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

\[ = \int a^2xx'p(a)\,da + \int b^2p(b)\,db + (x + x')\int abp(a)p(b)\,dadb = \alpha xx' + \beta.\]
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!
Model Selection in Practice; 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:

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

- \( v_1 = v_2 = 1 \)
- \( v_1 = v_2 = 0.32 \)
- \( v_1 = 0.32 \) and \( v_2 = 1 \)
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

$$p(y|f) = \prod_{i=1}^{n} p(y_i|f_i) = \prod_{i=1}^{n} \Phi(y_if_i).$$
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:

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

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:

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

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

\[
\mu_* = \mathbf{k}_*^{\top}K^{-1}\mathbf{m} \\
\sigma_*^2 = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^{\top}(K^{-1} - K^{-1}AK^{-1})\mathbf{k}_*.
\]

Using this approximation with the cumulative Gaussian likelihood

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

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

**Laplace’s method**: Find the Maximum A Posteriori (MAP) latent values $f_{\text{MAP}}$, 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].
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