Prediction with Gaussian Processes: Basic Ideas

Chris Williams

School of Informatics, University of Edinburgh, UK

Overview

- Bayesian Prediction
- Gaussian Process Priors over Functions
- GP regression
- GP classification

Bayesian prediction

- Define a prior over functions
- Observe data, obtain a posterior distribution over functions
  \[ P(f|D) \propto P(f)P(D|f) \]
  posterior \( \propto \) prior \( \times \) likelihood
- Make predictions by averaging predictions over the posterior \( P(f|D) \)
- Averaging mitigates overfitting

Bayesian Linear Regression

\[ f(x) = \sum_i w_i \phi_i(x) \quad w \sim N(0, \Sigma) \]

Samples from the prior
Gaussian Processes: Priors over functions

- For a stochastic process \( f(x) \), mean function is
  \[ \mu(x) = E[f(x)]. \]
  Assume \( \mu(x) \equiv 0 \ \forall x \)

- Covariance function
  \[ k(x, x') = E[f(x)f(x')]. \]

- Forget those weights! We should be thinking of defining priors over functions, not weights.

- Priors over function-space can be defined directly by choosing a covariance function, e.g.
  \[ k(x, x') = \exp(-w|x - x'|) \]

- Gaussian processes are stochastic processes defined by their mean and covariance functions.

Examples of GPs

- \( \sigma_0^2 + \sigma_1^2 xx' \)
- \( \exp -|x - x'| \)
- \( \exp -(x - x')^2 \)

Connection to feature space

A Gaussian process prior over functions can be thought of as a Gaussian prior on the coefficients \( w \sim N(0, \Lambda) \) where

\[
  f(x) = \sum_{i=1}^{N_f} w_i \phi_i(x) = w^T \Phi(x)
\]

\[
  \Phi(x) = \begin{pmatrix}
    \phi_1(x) \\
    \phi_2(x) \\
    \vdots \\
    \phi_{N_f}(x)
  \end{pmatrix}
\]

In many interesting cases, \( N_f = \infty \)

Choose \( \Phi(\cdot) \) as eigenfunctions of the kernel \( k(x, x') \) wrt \( p(x) \) (Mercer)

\[
  \int k(x, y)p(x)\phi_i(x) \, dx = \lambda_i \phi_i(y)
\]

Gaussian process regression

Dataset \( \mathcal{D} = (x_i, y_i)_{i=1}^n \), Gaussian likelihood \( p(y_i|f_i) \sim N(0, \sigma^2) \)

\[
  \bar{f}(x) = \sum_{i=1}^{n} \alpha_i k(x, x_i)
\]

where

\[
  \alpha = (K + \sigma^2 I)^{-1} y
\]

\[
  \text{var}(x) = k(x, x) - \bar{k}(x)(K + \sigma^2 I)^{-1} \bar{k}(x)
\]

in time \( O(n^3) \), with \( \bar{k}(x) = (k(x, x_1), \ldots, k(x, x_n))^T \).
After 1 observation:

After 2 observations:

Approximation methods can reduce $O(n^3)$ to $O(mn^2)$ for $m \ll n$

GP regression is competitive with other kernel methods (e.g. SVMs)

Can use non-Gaussian likelihoods (e.g. Student-t)

Adapting kernel parameters

$$k(x^i, x^j) = v_0 \exp \left( -\frac{1}{2} \sum_{l=1}^{d} w_l (x^i_l - x^j_l)^2 \right)$$

For GPs, the marginal likelihood (aka Bayesian evidence) $\log P(y|\theta)$ can be optimized wrt the kernel parameters $\theta = (v_0, w)$

For GP regression $\log P(y|\theta)$ can be computed exactly

$$\log P(y|\theta) = -\frac{1}{2} \log |K + \sigma^2 I| - \frac{1}{2} y^T (K + \sigma^2 I)^{-1} y - \frac{n}{2} \log 2\pi$$
Regularization

- $\bar{f}(x)$ is the (functional) minimum of
  
  $$J[f] = \frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \frac{1}{2} \|f\|_{H}^2$$

  (1st term = $-\log$-likelihood, 2nd term = $-\log$-prior)

- However, the regularization framework does not yield predictive variance or marginal likelihood

Previous work

- Wiener-Kolmogorov prediction theory (1940's)
- Splines (Kimeldorf and Wahba, 1971; Wahba 1990)
- ARMA models for time-series
- Kriging in geostatistics (for 2-d or 3-d spaces)
- Regularization networks (Poggio and Girosi, 1989, 1990)
- Design and Analysis of Computer Experiments (Sacks et al, 1989)
- Infinite neural networks (Neal, 1995)

GP prediction for classification problems

- Likelihood
  
  $$-\log P(y_i | f_i) = \log(1 + e^{-y_i f_i})$$

  - Integrals can’t be done analytically
    - Find maximum a posteriori value of $P(f|x)$ (Williams and Barber, 1997)
    - Expectation-Propagation (Minka, 2001; Opper and Winther, 2000)
    - MCMC methods (Neal, 1997)

Squash through logistic (or erf) function
MAP Gaussian process classification

To obtain the MAP approximation to the GPC solution, we find \( \hat{f} \) that maximizes the convex function

\[
\psi(y) = -\sum_{i=1}^{n} \log(1 + e^{-y_i}) - \frac{1}{2} y^T K^{-1} y + c
\]

The optimization is carried out using the Newton-Raphson iteration

\[
f_{\text{new}} = K (I + W K)^{-1} (W f + (t - \pi))
\]

where \( W = \text{diag}(\pi_1(1 - \pi_1), ..., \pi_n(1 - \pi_n)) \) and \( \pi_i = \sigma(f_i) \). Basic complexity is \( O(n^3) \)

For a test point \( x_\text{test} \), we compute \( \hat{f}(x_\text{test}) \) and the variance, and make the prediction as

\[
P(\text{class 1}|x_\text{test}, D) = \int \sigma(f) p(f|y) df,
\]

This is a quadratic programming problem. Can be solved in many ways, e.g. with interior point methods, or special purpose algorithms such as SMO.

Basic complexity is \( O(n^3) \).

- Define \( g_\sigma(z) = \log(1 + e^{-z}) \)

- SVM classifier is similar to GP classifier, but with \( g_\sigma \) replaced by \( g_{SV M}(z) = [1 - z]_+ \) (Wahba, 1999)

SVMs

1-norm soft margin classifier has the form

\[
f(x) = \sum_{i=1}^{n} y_i \alpha_i^* k(x, x_i) + w_0^*
\]

where \( y_i \in \{-1, 1\} \) and \( \alpha^* \) optimizes the quadratic form

\[
Q(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} t_i t_j \alpha_i \alpha_j k(x_i, x_j)
\]

subject to the constraints

\[
\sum_{i=1}^{n} y_i \alpha_i = 0
\]

\[
C \geq \alpha_i \geq 0, \quad i = 1, \ldots, n
\]

- Note that the MAP solution using \( g_\sigma \) solution is not sparse, but gives a probability output