Bayesian Kernel Methods

THE AUSTRALIAN NATIONAL UNIVERSITY

Unit 1: Bayes Rule, Approximate Inference, Hyperparameters

Unit 2: Gaussian Processes, Covariance Function, Kernel

Unit 3: GP: Regression

Unit 4: GP: Classification

Unit 5: Implementation: Laplace Approximation, Low Rank Methods

Unit 6: Implementation: Low Rank Methods, Bayes Committee Machine

Unit 7: Relevance Vector Machine: Priors on Coefficients

Unit 8: Relevance Vector Machine: Efficient Optimization and Extensions

http://mlg.anu.edu.au/~smola/summer2002/

Overview of Unit 6: Bayes Committee Machine

THE AUSTRALIAN NATIONAL UNIVERSITY

01: Splitting the Data
02: Bayes Committee Machine
03: Joining the Posterior
04: Proof
05: Sherman-Morrison-Woodbury
06: Predicting for Small Test Set
07: Generalized BCM

Splitting the Data

Idea

If we have too much data to minimize the log-posterior directly, we could simply use the following strategy:

- split into chunks
- optimize over each of the chunks independently
- average over the results

Problems

- how to average
- how to improve confidence ratings
- what is the form of the optimization problem on the chunks
- connection to the exact solution

Bayes Committee Machine (Tresp)

THE AUSTRALIAN NATIONAL UNIVERSITY

Basic Idea

Split data D into N chunks $D_1, \ldots, D_N$. By Bayes' rule we have

$$p(f|D_i, D_{i-1}) \propto p(D_i|f, D_{i-1})p(f|D_{i-1})$$

Approximation To be able to expand $p(f|D_1, \ldots, D_N)$ into terms of $p(f|D_i)$ we approximate

$$p(D_i|f, D_{i-1}) \approx p(D|f)$$

This would be true for function generating the data (given the underlying hypothesis, the individual data blocks are independent), in our case it is just an approximation.

Result

$$p(f|D_i) \propto \left(\prod_{i=1}^{N} p(D_i|f)\right) p(f) = \frac{\prod_{i=1}^{N} p(D_i|f)p(f)}{p^{N-1}(f)} \propto \frac{\prod_{i=1}^{N} p(f|D_i)}{p^{N-1}(f)}$$

Now we may approximate each of the $p(f|D_i)$ and combine the results.

Joining the Posterior

Laplace Approximation

We approximate each $p(f|D_i)p(f)$ by a normal distribution.

Combining Normal Distributions

Taking products of normal distributions with means μ_i and covariances Σ_i leads to an overall normal distribution with

$$\Sigma^{-1} = \sum_{i=1}^{N} \Sigma_i^{-1} \text{ and } \mu = \Sigma \sum_{i=1}^{N} \Sigma_i^{-1} \mu_i$$

For quotients (of densities) the signs are reversed.

Combined Posterior

Given the GP prior $p(f)$ with covariance matrix Σ_G we obtain

$$\Sigma^{-1} = (1-N)\Sigma_G^{-1} + \sum_{i=1}^{N} \Sigma_i^{-1} \text{ and } \mu = \Sigma \sum_{i=1}^{N} \Sigma_i^{-1} \mu_i$$

GP Regression

Estimate on Subset

For regression with normal additive noise we have

$$\mu_i = K^{mn}(K^{nn} + \sigma^2 \mathbf{1})^{-1}\mathbf{y} \text{ and } \Sigma_i = K^{mm} - K^{mn}(K^{nn} + \sigma^2 \mathbf{1})^{-1}\left(K^{mn}\right)^\top$$

where we labelled all the predictive part with m and the given part with n.

Combining Individual Predictions

$$\text{Covar } \Sigma^{-1} = (1-N)K^{mm} + \sum_{i=1}^{N}\left(K^{mm} - K_i^{mn}(K_i^{nn} + \sigma^2 \mathbf{1})^{-1}\left(K_i^{mn}\right)^\top\right)$$

$$\mu = \Sigma \sum_{i=1}^{N} \Sigma_i^{-1}\mu_i$$

Online-BCM

Idea

If we observe a new instance (x_{m+1}, y_{m+1}), we can make the approximation

$$p(f|X, Y, (x_{m+1}, y_{m+1})) \approx p(f|X, Y)\frac{p(f|y_i, x_i)}{p(f)}$$

and simply update mean and covariance according to the combination strategy.

$$\begin{aligned}\Sigma^{-1} &\leftarrow \Sigma^{-1} + (\Sigma_i^{-1} - \Sigma_G^{-1}) \\ \Sigma^{-1}\mu &\leftarrow \Sigma^{-1}\mu + (\Sigma_i^{-1} - \Sigma_G^{-1})\mu_i\end{aligned}$$

Advantage

We only need to store mean and covariance for updates. No need to remember the training data (for GP regression exact, since mean and variance are **sufficient statistics** of a Normal distribution).

General Case

Idea

For the posterior on the individual chunks X_i, Y_i we have

$$\begin{aligned} -\log p(f|Y_i, X_i) &= \sum_{i=1}^{m_i} -\log p(y_i|x_i, f(x_i)) - \log p(f) + c \\ &= \sum_{i=1}^{m_i} -\log p(y_i|x_i, f(x_i)) + \frac{1}{2}\mathbf{f}^\top \Sigma_G^{-1}\mathbf{f} + c \end{aligned}$$

The Laplace approximation at the mode of $p(f|Y_i, X_i)$ yields

$$\begin{aligned} \mu &= \Sigma_G \mathbf{c}' && \text{where } c_i' := \partial_{\mu_i} - \log(y_i|x_i, \mu_i) \\ \Sigma^{-1} &= \Sigma_G^{-1} + \operatorname{diag}(\mathbf{c}'') && \text{where } c_i'' := \partial_{\mu_i}^2 - \log(y_i|x_i, \mu_i) \end{aligned}$$

So, the curvature of the likelihood at the mean determines the confidence of the estimates.

Why Does It Work?

THE AUSTRALIAN NATIONAL UNIVERSITY

Idea

In general we want to minimize the negative log-posterior. This can be written as

$$\begin{aligned} -\log p(f|X,Y) &= \left[\sum_{i=1}^{N} \underbrace{-\log p(f|X_i,Y_i)}_{:=g_i(f)}\right] \underbrace{-\log p(f)}_{:=g_0(f)} + c \\ &= \sum_{i=1}^{N} \left[\underbrace{-\log p(f|X_i,Y_i) - \log p(f)}_{g_0(f)+g_i(f)}\right] + \underbrace{(N-1)\log p(f)}_{-(N-1)g_0(f)} + c \end{aligned}$$

Reformulation

Given $g_0, g_1, \ldots, g_N : \mathbb{R}^n \to \mathbb{R}$ we want to minimize $g(\alpha) := g_0(\alpha) + \sum_{i=1}^{N} g_i(\alpha)$.

Instead, we **minimize each** $\tilde{g}_i := g_0 + g_i$ **separately**, compute a quadratic approximation q_i of $\tilde{g}_i$ at its minimum, and minimize $q := \sum_{i=1}^{N} q_i - (N-1)g_0$.

Why Does It Work (part II)?

THE AUSTRALIAN NATIONAL UNIVERSITY

General Observation

If all g_i are quadratic functions, the procedure is exact. Otherwise, it is a good first approximation.

GP Regression with Normal Noise

For GP regression with Normal noise the posterior is a **quadratic function**. For each of the partial negative log-posteriors the approximation is exact, hence the overall estimate is exact.

Prediction

For prediction on a small test test, we can use the predictive means and variances on the subsets. Again, for GP regression and normal additive noise the estimate is exact.

Note: This also holds if we would have to invert a **large covariance matrix** for full prediction instead, since we only predict on a low dimensional subspace.

Iterative Extension

A Simple Idea

Use the quadratic approximations q_i to improve the estimates at the next iteration:

- Find initial approximations q_i by minimizing $g_i + g_0$.
- Repeat

  minimize $g_i + \sum_{j=1,j\neq i}^{N} q_j$

  compute new quadratic approximation q_i at minimum
- Until converged

When to use

- If we have a simple minimization algorithm which cannot deal with $g = \sum_i f_i$ simultaneously (too much data).
- If we have a ready-made optimizer for the subproblems.
- Otherwise, Newton method should be better (after all, we need an algorithm to minimize each of the auxiliary functions).

Beyond Bayes: Combining Predictors

THE AUSTRALIAN NATIONAL UNIVERSITY

Problem

Assume, we are given N predictors f_i (with $1 \leq i \leq N$) which we would like to combine such that

1. the combined predictor is unbiased
2. the variance of the prediction is minimized.

More specifically, the following conditions hold:

1. The predictors f_i are **unbiased**.
2. We have the liberty of finding different linear combinations **for each test point**.
3. We know the covariance matrices $\Omega_{ij} = \mathrm{Cov}(f_i, f_j)$ between all predictors.

Ansatz

- Prediction via $f = A[f_1, \ldots, f_N]$
- To ensure unbiasedness we require that $AI = \mathbf{1}$, where $I = [\mathbf{1}, \ldots, \mathbf{1}]$.

Combining Predictors, Part II

THE AUSTRALIAN NATIONAL UNIVERSITY

Recall

- $f = A[f_1, \ldots, f_N]$
- $AI = \mathbf{1}$, where $I = [\mathbf{1}, \ldots, \mathbf{1}]$.

Variance

$$\mathbf{E}\left[f^\top f\right] = \mathbf{E}\left[\left(A[f_1, \ldots, f_N]\right)^\top \left(A[f_1, \ldots, f_N]\right)\right] = \text{tr } A\Omega A^\top$$

Constrained Optimization Problem

$$\begin{aligned} \text{minimize} \quad & \tfrac{1}{2}\text{tr } A\Omega A^\top \\ \text{subject to} \quad & AI = \mathbf{1} \end{aligned}$$

Lagrange Function

$$L(A, \Lambda) = \frac{1}{2}\text{tr } A\Omega A^\top + \text{tr } \Lambda(AI - \mathbf{1})$$

We obtain that $A = -\Lambda^\top I^\top \Omega^{-1}$ is the saddlepoint value.

After some more algebra, this leads to $A = (I^\top \Omega^{-1} I)^{-1} I^\top \Omega^{-1}$.

Combining Predictors, Part III

Special Case

Predictors are independent (e.g., they were obtained on independent blocks of the data). In this case

$$\Omega = \begin{bmatrix} \Omega_{11} & 0 & \dots & 0 \\ 0 & \Omega_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \Omega_{NN} \end{bmatrix}$$

and hence

$$A = \Sigma \left[\Omega_{11}^{-1}, \dots, \Omega_{NN}^{-1}\right] \text{ where } \Sigma^{-1} = \sum_{i=1}^{N} \Omega_{ii}^{-1}.$$

Prediction and Variance

This leads to $f = \Sigma \sum_{i=1}^{m} \Omega_{ii}^{-1} f_i$ and $\text{Cov}[f] = (I^\top \Omega^{-1} I)^{-1} = \Sigma$.

In other words, the averaging method is identical, except that we ignored the prior (to be expected for a ML fit).