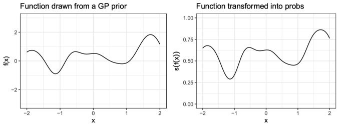
Introduction to Machine Learning

Gaussian Process Classification



Learning goals

- XXX
- XXX

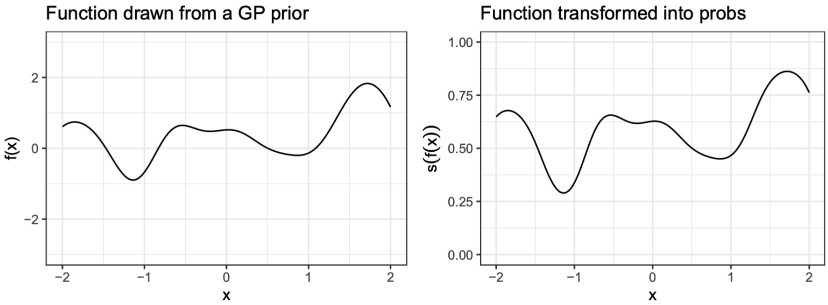
GAUSSIAN PROCESS CLASSIFICATION

- Consider a binary classification problem where we want to learn $h : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{Y} = \{0, 1\}$.
- The idea behind Gaussian process classification is very simple: a GP prior is placed over the score function $f(\mathbf{x})$ and then transformed to a class probability via a sigmoid function $s(t)$

$$p(y = 1 \mid f(\mathbf{x})) = s(f(\mathbf{x})).$$

- This is a non-Gaussian likelihood, so we need to use approximate inference methods, e.g. Laplace approximation, expectation propagation, MCMC
- For more details see *Rasmussen, Gaussian Processes for Machine Learning, Chapter 3*.

GAUSSIAN PROCESS CLASSIFICATION



GAUSSIAN PROCESS CLASSIFICATION

According to Bayes' rule, the posterior (of the score function $\boldsymbol{f}$)

$$p(\boldsymbol{f} \mid \mathbf{X}, \mathbf{y}) = \frac{p(\mathbf{y} \mid \boldsymbol{f}, \mathbf{X}) \cdot p(\boldsymbol{f} \mid \mathbf{X})}{p(\mathbf{y} \mid \mathbf{X})} \propto p(\mathbf{y} \mid \boldsymbol{f}) \cdot p(\boldsymbol{f} \mid \mathbf{X})$$

(the denominator is independent of $\boldsymbol{f}$ and thus dropped).

Since $p(\boldsymbol{f} \mid \mathbf{X}) \sim \mathcal{N}(0, \boldsymbol{K})$ by the GP assumption, we have

$$\log p(\boldsymbol{f} \mid \mathbf{X}, y) \propto \log p(\mathbf{y} \mid \boldsymbol{f}) - \frac{1}{2}\boldsymbol{f}^\top \boldsymbol{K}^{-1} \boldsymbol{f} - \frac{1}{2} \log |\boldsymbol{K}| - \frac{n}{2} \log 2\pi.$$

GAUSSIAN PROCESS CLASSIFICATION

If the kernel is fixed, the last two terms are fixed. To obtain the maximum a-posteriori estimate (MAP) we minimize

$$\frac{1}{2}\boldsymbol{f}^\top \boldsymbol{K}^{-1}\boldsymbol{f} - \sum_{i=1}^{n} \log p(y^{(i)} \mid f^{(i)}) + C.$$

Note that $-\sum_{i=1}^{n} \log p(y^{(i)} \mid f^{(i)})$ is the logistic loss. We can see that Gaussian process classification corresponds to **kernel Bayesian logistic regression**!

COMPARISON: GP VS. SVM

The SVM

$$\frac{1}{2}\|\boldsymbol{\theta}\|^2 + C\sum_{i=1}^{n} L\left(y^{(i)}, f\left(\mathbf{x}^{(i)}\right)\right),$$

where $L(y, f(\mathbf{x})) = \max\{0, 1 - f(\mathbf{x}) \cdot y\}$ is the Hinge loss.

By the representer theorem we know that $\boldsymbol{\theta} = \sum_{i=1}^{n} \beta_i y^{(i)} k\left(\mathbf{x}^{(i)}, \cdot\right)$ and thus $\boldsymbol{\theta}^\top\boldsymbol{\theta} = \beta^\top \boldsymbol{K}\beta = \boldsymbol{f}^\top\boldsymbol{K}^{-1}\boldsymbol{f}$, as $\boldsymbol{K}\beta = \boldsymbol{f}$. Plugging that in, the optimization objective is

$$\frac{1}{2}\boldsymbol{f}^\top\boldsymbol{K}^{-1}\boldsymbol{f} + C\sum_{i=1}^{n} L\left(y^{(i)}, f\left(\mathbf{x}^{(i)}\right)\right).$$

COMPARISON: GP VS. SVM

For log-concave likelihoods $\log p(\mathbf{y} \mid \boldsymbol{f})$, there is a close correspondence between the MAP solution of the GP classifier

$$\arg\min_{f} \frac{1}{2}\boldsymbol{f}^\top \boldsymbol{K}^{-1}\boldsymbol{f} - \sum_{i=1}^{n} \log p(y^{(i)} \mid f^{(i)}) + C \quad \text{(GP classifier)}$$

and the SVM solution

$$\arg\min_{f} \qquad \frac{1}{2}\boldsymbol{f}^\top \boldsymbol{K}^{-1}\boldsymbol{f} + C\sum_{i=1}^{n} L\left(y^{(i)}, f\left(\mathbf{x}^{(i)}\right)\right) \quad \text{(SVM classifier).}$$

COMPARISON: GP VS. SVM

- Both the Hinge loss and the Bernoulli loss are monotonically decreasing with increasing margin $yf(\mathbf{x})$.
- The key difference is that the hinge loss takes on the value 0 for $yf(\mathbf{x}) \geq 1$, while the Bernoulli loss just decays slowly.
- It is this flat part of the hinge function that gives rise to the sparsity of the SVM solution.
- We can see the SVM classifier as a “sparse” GP classifier.