AutoML: Gaussian Processes

Gaussian Process Classification

Bernd Bischl Frank Hutter Lars Kotthoff
Marius Lindauer Joaquin Vanschoren

Training a GP via the Maximum Likelihood I

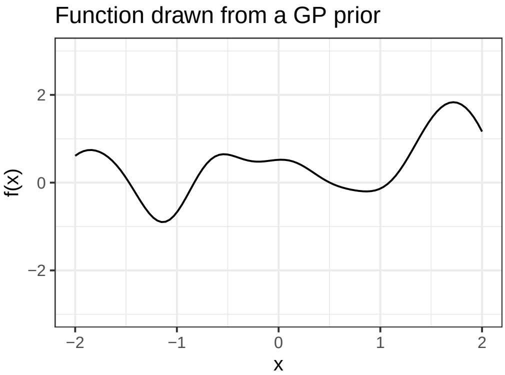
- Consider a binary classification problem, in which we want to learn $h : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{Y} = \{0, 1\}$.

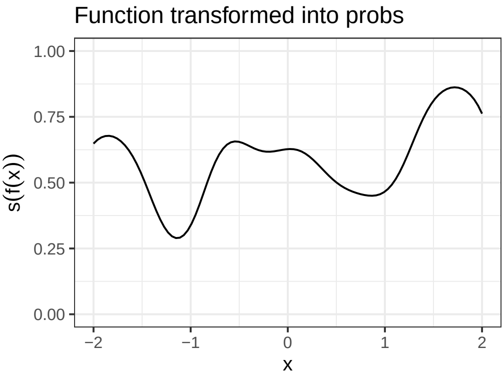
- The idea behind Gaussian process classification is straightforward: a GP prior is placed over the score function $f(\mathbf{x})$ and then transformed to a class probability via a sigmoid function $s(t)$:
$$p(y = 1 \mid f(\mathbf{x})) = s(f(\mathbf{x})).$$

- Since this is a non-Gaussian likelihood, we need to use approximate inference methods, such as Laplace approximation, expecation propagation, MCMC.

- For more details see *Rasmussen, Gaussian Processes for Machine Learning, Chapter 3*.

Training a GP via the Maximum Likelihood II





Training a GP via the Maximum Likelihood III

- According to Bayes' rule, the posterior of the score function $\boldsymbol{f}$ takes the following form:

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

  where, the denominator is independent of $\boldsymbol{f}$ and hence has been dropped.

- Frm the GP assumption, we can assert that $p(\boldsymbol{f} \mid \mathbf{X}) \sim \mathcal{N}(0, \boldsymbol{K})$. Hence, we have:

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

Training a GP via the Maximum Likelihood IV

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

- It can be seen that the Gaussian process classification corresponds to the **kernel Bayesian logistic regression**!

Comparison: GP vs. SVM I

- For the SVM, we have:

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

- Plugging that in, the optimization objective would be:

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

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

- From the representer theorem: $\boldsymbol{\theta} = \sum_{i=1}^{n} \beta_i\, y^{(i)} k\left(\mathbf{x}^{(i)}, \cdot\right)$, and thus:

$$\boldsymbol{\theta}^\top \boldsymbol{\theta} = \beta^\top \boldsymbol{K} \beta = \boldsymbol{f}^\top \boldsymbol{K}^{-1} \boldsymbol{f}$$

.

Comparison: GP vs. SVM II

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

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

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

Comparison: GP vs. SVM III

- 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 decays slowly.
- It is this flat part of the Hinge function that gives rise to the sparsity of the SVM solution.
- The SVM classifier can be construed as a "sparse" GP classifier.