Probabilistic Machine Learning

Exercise Sheet #7

Binary Classification with GPs

1. **EXAMple Question — The original Laplace approximation** Consider a single scalar latent number $r \in [0, 1]$, assumed to be distributed according to the Beta distribution (which you know from Exercise Sheet #2)

$$p_r(r) = \mathcal{B}(r; a, b) = \frac{1}{B(a,b)} r^{a-1}(1-r)^{b-1} \qquad \text{with} \qquad B(a,b) := \int_0^1 r^{a-1}(1-r)^{b-1}\, dr. \tag{1}$$

We will construct a Gaussian approximation $p_r(r) \approx q(r) = \mathcal{N}(r; \mu, \sigma^2)$ to $p(r)$. A variant of this computation was used by Pierre-Simon Laplace to approximate the Eulerian Beta function $B(a, b)$. Thus, consider the unnormalized function $\tilde{p}(r) = r^{a-1}(1-r)^{b-1}$.

   (a) Write down the first and second derivatives

$$g(r) := \frac{\partial \log \tilde{p}(r)}{\partial r} \qquad \text{and} \qquad \psi(r) := \frac{\partial^2 \log \tilde{p}(r)}{(\partial r)^2}$$

   (b) Find the maximum $\hat{r} = \arg\max_{r\in[0,1]} \log p_r(r)$ by setting $g(r) = 0$ and thus construct **the Laplace approximation** $q(r) = \mathcal{N}(r; \hat{r}, -\psi^{-1}(\hat{r}))$. For this to work, you will have to make certain restricting assumptions on the value of the parameters a and b. What are they?

   (c) Compute an approximation to the Beta function by using the Gaussian integral (which was known to Laplace) $\int e^{-x^2}\, dx = \sqrt{\pi}$ to get

$$\begin{aligned} B(a,b) &\approx \int_{-\infty}^{\infty} \exp\left(\log \tilde{p}(\hat{r}) - \frac{(r-\hat{r})^2}{-2\psi^{-1}}\right) dr \\ &= \hat{r}^{(a-1)} \cdot (1-\hat{r})^{(b-1)} \cdot \sqrt{2\pi(-\psi^{-1}(\hat{r}))} = Z(a,b). \end{aligned}$$

   You may note that it would be preferable to do the integral over the domain $[0, 1]$ instead of the whole real line. But Laplace did not have access to the error function, so we incur a small approximation error by extending the integration domain. What is your explicit form of $Z(a, b)$?

2. **Theory Question — Stirling's approximation** As you know from Exercise Sheet #2, the Beta Function can be written as

$$B(a,b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)} \quad \text{with} \quad \Gamma(z) = \int_0^\infty t^{z-1} e^{-t}\, dt,$$

   where Γ denotes the Eulerian Gamma function (The use of the greek letter is due to Legendre, 1809). By this definition, the Gamma function is evidently (up to a transformation), the normalization constant of the **Gamma distribution**

$$p(t \mid a, b) = \Gamma(t; a, b) := \frac{b^a}{\Gamma(a)} t^{a-1} e^{-bt} \quad \text{for } t > 0, a \geq 1, b \geq 0 \in \mathbb{R}$$

   If we fix $b = 1$, then we can use almost the exact same procedure as in Exercise 1 to construct the Laplace approximation of $\Gamma(t; a, 1)$, and thus the factorial! To do so,

(a) as in 1 above, consider the unnormalized density $\tilde{p}(t \mid a, b) = b^a \cdot t^{a-1} e^{-bt}$. Compute the first and second derivative of $\log \tilde{p}(t \mid a, b)$, and find its mode $\hat{t}$ and Hessian $\Psi(\hat{t})$ there.

(b) As in 1.(c), this yields a Taylor approximation

$$\log \tilde{p}(t \mid a, b) \approx \log \tilde{p}(\hat{t}; a, b) - \frac{1}{2} \frac{(t - \hat{t})^2}{-\psi^{-1}(\hat{t})}.$$

Use this to estimate for the normalization constant $\Gamma(a)$ of $\tilde{p}(t \mid a)$ as

$$\Gamma(a) \approx \tilde{p}(\hat{t} \mid a, b) \cdot \int_{-\infty}^{\infty} \exp\left(-\frac{1}{2} \frac{(t - \hat{t})^2}{-\psi^{-1}(\hat{t})} \right) dt = \tilde{p}(\hat{t} \mid a, b) \sqrt{2\pi(-\psi^{-1}(\hat{t}))}$$

Here we have again added a small positive approximation error by extending the integral to the whole real line. Write out the explicit form as a function of a, b to get full points. If you plug in your results for $\hat{t}$ and $\psi(\hat{t})$ and use $\Gamma(n+1) = n!$, you will recover a famous approximation to the factorial function, **Stirling's approximation**:

$$n! \approx \sqrt{2\pi n} \left(\frac{n}{e} \right)^n.$$

3. **Practical Question**

The practical part consists of two tasks. In the first you have to play around with an off-the-shelf GP classification package. In the second you will open the black box and implement the binary GP classification yourself. Even if you don't plan on doing the second part, you should check out the first to get a feeling for how different kernels behave.

On ILIAS you can find a `jupyter` notebook that describes the exercise in more detail.