DATA20023 Bayesian Machine Learning

Exercise 4: Gaussian processes and gradient-based variational approximation
Due Feb 24

---

1 Gaussian process regression

Read in the data provided in Moodle as $N = 30$ pairs of univariate inputs x (first column) and outputs y. Your task is to fit **GP regression** model for this data and you can build your solution on the notebook if you wish. Use the kernel

$$k(x, x') = \lambda e^{-\frac{1}{2l^2}(x-x')^2}$$

and normal likelihood $p(y|f(x)) = \mathcal{N}(0, \sigma^2)$ where σ^2 is the variance.

(a) Implement a solution for selecting the three hyperparameters (λ, l and σ^2) by maximising the marginal likelihood $p(\boldsymbol{y}|\boldsymbol{x}, \lambda, l, \sigma^2)$ and report the optimal values. You can use any strategy, such as

- Grid search or some gradient-free optimization routine (e.g. Bayesian optimization)
- Gradient-based optimization using either manually computed derivatives (available in Barber's and Murphy's books) or automatic differentiation (PyTorch)
- Metropolis-Hastings that gives you a posterior over the hyperparameters

(b) Plot the posterior mean and an illustration of the uncertainty (samples from the posterior, or confidence intervals) for the function $f(x)$ over some grid for x using the optimal hyperparameters, overlaid on top of a scatter-plot of the training data. Can you model the data well? Draw a similar plot also using some other choice for the parameters (that has clearly worse marginal likelihood) and discuss the differences.

2 Non-conjugate GP

Read in the data provided in Moodle as pairs of $N = 40$ univariate inputs x (first column) and outputs y. Now y are integer values and your task is a to implement a model for this data using a GP prior for functions $f(x)$ and Poisson likelihood

$$p(y|f) = \text{Poisson}(e^f)$$

for the observations. That is, we assume $e^{f(x)}$ to be the rate parameter of a Poisson distribution. Use the same kernel as above, with $l = 0.5$ and $\lambda = 2.0$.

Implement **Laplace approximation** for inference, as briefly illustrated in lecture slide 23 and explained e.g. in Barber's book Section 19.5. Even though the example in the book is for classification, you can follow the same procedure by just changing the likelihood.

1. Use gradient-based optimization to learn $\hat{\boldsymbol{f}}$ that maximizes the joint log-likelihood. You can use analytic gradients or automatic differentiation.
2. Approximate the posterior with normal distribution with mean $\hat{\boldsymbol{f}}$ and covariance $\left(\boldsymbol{K}^{-1} + \boldsymbol{W}\right)^{-1}$, where $\boldsymbol{W}$ is a diagonal matrix with second derivatives of $-\log p(y_n|f_n)$ on the diagonal, evaluated at the mean.
3. Write code that provides samples from the posterior predictive distribution $p(\boldsymbol{y}|\boldsymbol{x}^*)$ for a vector of new inputs $\boldsymbol{x}^*$. You can do this by first sampling $\boldsymbol{f}^*$ from the approximation and then sampling the actual outcomes $\boldsymbol{y}$ from Poisson with rate $e^{\boldsymbol{f}^*}$.

Plot the function and its uncertainty together with a scatter-plot of the data. You can either plot $\log(\boldsymbol{y})$ (or perhaps $\log(\boldsymbol{y} + 1)$) and $\boldsymbol{f}$ in the same plot or use $\boldsymbol{y}$ and $e^{\boldsymbol{f}}$. If you use the latter, remember to think about how to present the uncertainty right.