Probabilistic Inference & Learning

Exercise Sheet #6

Generalized Linear Models

1. **Newton Optimization** Consider a real-valued function $L : \mathbb{R}^d \to \mathbb{R}$ with real-vector-valued inputs $f \in \mathbb{R}^d$. Let

$$\nabla L(f) = \left[\frac{\partial L(f)}{\partial f_i}\right]_{i=1,\dots,d} \in \mathbb{R}^d \qquad \text{and} \qquad B(f) = \left[\frac{\partial^2 L(f)}{\partial f_i \partial f_j}\right]_{i,j} \text{ for } i,j \in [1,\dots,n]$$

denote the *gradient* and *Hessian matrix* of f, respectively. Assume that $B(f)$ is symmetric positive definite for all $f \in \mathbb{R}^d$.

(a) Now consider the function L in the vicinity of a specific location $f_0 \in \mathbb{R}^d$. Use Taylor's expansion to construct a local *quadratic* approximation $\tilde{L}(f_0 + \delta)$ to $L(f_0 + \delta)$ for a small $\delta \in \mathbb{R}^n$. Show that the minimum of this approximation lies at

$$f_1 := f_0 - B^{-1}(f_0)\nabla L(f_0).$$

This is the basic Newton optimization step from x_0 to x_1. *20 points*

(b) Assume that the input vector f to L can be written in the form $f(w) = \phi^\intercal w$ with some feature matrix $\phi : \mathbb{R}^m \to \mathbb{R}^d$ and a weight vector $w \in \mathbb{R}^m$. Re-write the second-order expansion of $L(w) = L(f = \phi^\intercal w)$ analagous to 1.(a), around $w = w_0 + \epsilon$. What is the Newton step in w? *15 points*

2. **Logistic Link Functions** Consider the logistic link function

$$\sigma(f) = \frac{1}{1 + \exp(-f)} \qquad \text{for } f \in \mathbb{R}.$$

Show the following two identities used in Lecture 10:

$$\frac{\partial \log \sigma(y \cdot f)}{\partial f} = \left(\frac{y+1}{2} - \sigma(f)\right) \qquad \text{and} \qquad \frac{\partial^2 \log \sigma(y \cdot f)}{(\partial f)^2} = -\sigma(f)(1 - \sigma(f))$$

15 points

3. **Basic Properties of Gaussian Processes** Use an ipython notebook (you can use the basic scaffold available as `GP_basics.ipynb` on Ilias) to solve the following questions

(a) draw and plot 3 sample functions from $p(f) = \mathcal{GP}(f; m, k)$ with $m(x) \equiv 0 \forall x$ and kernel $k(a,b) = (1 + (a-b)^2)^{-1/2}$ (known as the *rational quadratic* kernel) on the plotting grid given by `X = numpy.linspace(-8,8,100)`.

10 points

(b) draw and plot 3 sample functions as in 3.(a), but use the mean function $m(x) = x^2$. *5 points*

(c) draw and plot 3 sample functions as in 3.(a), but use the kernel $\tilde{k}(a,b) = 100 \cdot k(a,b)$

5 points

(d) draw and plot 3 sample functions as in 3.(a), but use the kernel $\hat{k}(a,b) = k(\phi(a), \phi(b))$ with $\phi(x) = ((x + 8.1)/10)^{3/2}$ *10 points*

(e) draw $n = 10^4$ sample functions and compute their *empirical covariance matrix* (use the library function `numpy.cov`). Note that this should give a matrix of size 100×100, not $10^4 \times 10^4$. Make a plot of both the empirical covariance and the kernel matrix k_{XX} next to each other and comment on why they look similar

20 points