Math 156: Machine Learning

Instructions. Please submit a pdf with neatly written solutions. See the syllabus for instructions on coding problems.

1. Book Exercise 1.1

2. Suppose that we are performing linear regression for sampled data points $(\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2), ..., (\mathbf{x}_N, t_N)$ and that while sampling, along with the data point $(\mathbf{x}_i, t_i)$, we are given the scalar value $r_i > 0$ which represents how certain we are about the target value t_i associated to $\mathbf{x}_i$. We can alter the least-squares error function to use this information as

$$E_D(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N} r_n\{t_n - \mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_n)\}^2.$$

   Show that the solution $\mathbf{w}^*$ which minimizes this error function is

$$\mathbf{w}^* = \left(\sum_{n=1}^{N} r_n\boldsymbol{\phi}(\mathbf{x}_n)\boldsymbol{\phi}(\mathbf{x}_n)^T\right)^{-1}\left(\sum_{n=1}^{N} r_n\boldsymbol{\phi}(\mathbf{x}_n)t_n\right).$$

3. Recall the polynomial curve fitting model which fits data with functions of the form $y(x, \mathbf{w}) = w_0 + w_1x + w_2x^2 + ... + w_Mx^M$. Assume that the target t associated to x is distributed as $t \sim \mathcal{N}(y(x, \mathbf{w}), \beta^{-1})$, and that there is i.i.d. training data $x_1, x_2, ..., x_N$ with targets $t_1, t_2, ..., t_N$. We now introduce a prior distribution over the polynomial coefficients $\mathbf{w}$:

$$p(\mathbf{w} \mid \alpha) = \mathcal{N}\left(\mathbf{w} \mid \mathbf{0}, \alpha^{-1}\mathbf{I}\right) = \left(\frac{\alpha}{2\pi}\right)^{(M+1)/2} \exp\left\{-\frac{\alpha}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}\right\}$$

   Show that the maximum of the posterior is given by the minimum of

$$\frac{\beta}{2}\sum_{n=1}^{N}\left\{y\left(x_n, \mathbf{w}\right) - t_n\right\}^2 + \frac{\alpha}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}.$$

   What does this tell us about the MAP vs MLE approach in this case?

4. Linear Regression on Wine Dataset:
   (a) Download the ".csv" file for the red wines in the "Wine Quality" dataset.
   (b) Split the dataset into train, validation, and test sets. You can use scikit-learn's `train_test_split` function.
   (c) Write a program for training a simple linear regression model with sum-of-squares error function using the closed-form solution (do not use a built-in function).
   (d) For the train data, generate a plot of the actual target values vs predicted target values. How do we interpret this plot?
   (e) Report the root-mean-square on the train and test sets.

(f) Implement the least-mean-squares (LMS) algorithm for linear regression with random initialization of $\mathbf{w}^{(0)}$ and stepsize(s) of your choice (you can experiment with that).

(g) Report the root-mean-square on the train and test sets.

Suggested problems (not for submission)

1. Show that (3.28) is the minimizer of (3.27) in section 3.1.4 of the book.
2. Book Exercises 3.2, 3.4.
3. (Extra) Read section 3.3 on Bayesian Linear Regression. Book Exercises 3.8, 3.11.