Assignment 1

Homework assignments will be done individually: each student must hand in their own answers. Use of partial or entire solutions obtained from others or online is strictly prohibited. Electronic submission on Canvas is mandatory.

1. **Maximum Likelihood estimator** (10 points) Assuming data points are independent and identically distributed (i.i.d.), the probability of the data set given parameters: μ and σ^2 (the likelihood function):

$$P(\mathbf{x}|\mu,\sigma^2) = \prod_{n=1}^{N} \mathcal{N}(x_n|\mu,\sigma^2)$$

Please calculate the solution for μ and σ^2 using Maximum Likelihood (ML) estimator.

$$\ln P(\mathbf{x}|\mu,\sigma^2) = \ln \prod_{n=1}^{N} \mathcal{N}(x_n|\mu,\sigma^2) = \ln(\frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x_1-\mu)^2}{2\sigma^2}}) + \cdots + \ln(\frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x_n-\mu)^2}{2\sigma^2}})$$

$$= -\frac{1}{2}\ln(2\pi\sigma^2) - \frac{(x_1-\mu)^2}{2\sigma^2}\ln(e) - \cdots - \frac{1}{2}\ln(2\pi\sigma^2) - \frac{(x_n-\mu)^2}{2\sigma^2}\ln(e)$$

$$= -\frac{1}{2}\ln(2\pi) - \frac{1}{2}\ln(\sigma^2) - \frac{(x_1-\mu)^2}{2\sigma^2} - \cdots - \frac{1}{2}\ln(2\pi) - \frac{1}{2}\ln(\sigma^2) - \frac{(x_n-\mu)^2}{2\sigma^2}$$

$$= -\frac{1}{2}\ln(2\pi) - \ln(\sigma) - \frac{(x_1-\mu)^2}{2\sigma^2} - \cdots - \frac{1}{2}\ln(2\pi) - \ln(\sigma) - \frac{(x_n-\mu)^2}{2\sigma^2}$$

$$= -\frac{n}{2}\ln(2\pi) - n\ln(\sigma) - \frac{(x_1-\mu)^2}{2\sigma^2} - \cdots - \frac{(x_n-\mu)^2}{2\sigma^2}$$

$$\frac{\partial}{\partial\mu}\ln P(\mathbf{x}|\mu,\sigma^2) = -0 - 0 + \frac{2(x_1-\mu)}{2\sigma^2} + \cdots + \frac{2(x_n-\mu)}{2\sigma^2} = \frac{1}{\sigma^2}[(x_1+\cdots+x_n) - n\mu]$$

$$\frac{\partial}{\partial\sigma}\ln P(\mathbf{x}|\mu,\sigma^2) = -0 - \frac{n}{\sigma} + \frac{2(x_1-\mu)^2}{2\sigma^3} + \cdots + \frac{2(x_n-\mu)^2}{2\sigma^3} = -\frac{n}{\sigma} + \frac{1}{\sigma^3}[(x_1-\mu)^2 + \cdots + (x_n-\mu)^2]$$

$$0 = \frac{1}{\sigma^2}[(x_1+\cdots+x_n) - n\mu]$$

$$n\mu = (x_1+\cdots+x_n)$$

$$\mu = \frac{(x_1+\cdots+x_n)}{n}$$

$$0 = -\frac{n}{\sigma} + \frac{1}{\sigma^3}[(x_1-\mu)^2 + \cdots + (x_n-\mu)^2]$$

$$n = \frac{1}{\sigma^2}[(x_1-\mu)^2 + \cdots + (x_n-\mu)^2]$$

$$n\sigma^2 = (x_1-\mu)^2 + \cdots + (x_n-\mu)^2$$

$$\sigma^2 = \frac{(x_1-\mu)^2 + \cdots + (x_n-\mu)^2}{n}$$

2. **Maximum Likelihood** (10 points) We assume there is a true function $f(\mathbf{x})$ and the target value is given by $y = f(x) + \epsilon$ where ϵ is a Gaussian distribution with mean 0 and variance σ^2. Thus,

$$p(y|x, w, \beta) = \mathcal{N}(y|f(x), \beta^{-1})$$

where $\beta^{-1} = \sigma^2$.

Assuming the data points are drawn independently from the distribution, we obtain the likelihood function:

$$p(\mathbf{y}|\mathbf{x}, w, \beta) = \prod_{n=1}^{N} \mathcal{N}(y_n|f(x), \beta^{-1})$$

Please show that maximizing the likelihood function is equivalent to minimizing the sum-of-squares error function.

Maximizing the likelihood function

$$\ln p(\mathbf{y}|\mathbf{x}, w, \beta) = \frac{N}{2}\ln(\beta) - \frac{N}{2}\ln(2\pi) - \frac{\beta}{2}\sum_{n=1}^{N}(y_n - f(x_n))^2$$

$$\frac{\partial}{\partial f(x)}\ln p(\mathbf{y}|\mathbf{x}, w, \beta) = 0 - 0 - \frac{-2\beta}{2}\sum_{n=1}^{N}(y_n - f(x_n)) = \beta\sum_{n=1}^{N}(y_n - f(x_n))$$

$$\frac{\partial}{\partial \beta}\ln p(\mathbf{y}|\mathbf{x}, w, \beta) = \frac{N}{2\beta} - 0 - \frac{\sum_{n=1}^{N}(y_n - f(x_n))^2}{2} = \frac{N\beta^{-1} - \sum_{n=1}^{N}(y_n - f(x_n))^2}{2}$$

$$0 = \beta(\mathbf{y} - f(\mathbf{x}))$$

$$0 = \mathbf{y} - f(\mathbf{x})$$

$$\mathbf{y} = f(\mathbf{x})$$

$$0 = \frac{N\beta^{-1} - \sum_{n=1}^{N}(y_n - f(x_n))^2}{2}$$

$$0 = N\beta^{-1} - \sum_{n=1}^{N}(y_n - f(x_n))^2$$

$$\sum_{n=1}^{N}(y_n - f(x_n))^2 = N\beta^{-}1$$

$$\frac{\sum_{n=1}^{N}(y_n - f(x_n))^2}{N} = \frac{1}{\beta}$$

Minimizing the sum-of-squares error function.

$$E_D(w) = \frac{1}{2}\sum_{n=1}^{N}(y_n - w^T x_n)^2$$

$$\frac{\partial}{\partial w}E_D(w) = \sum_{n=1}^{N}(y_n - w^T x_n)x_n$$

$$0 = (\mathbf{y} - w^T\mathbf{x})\mathbf{x}$$

$$0 = \mathbf{y} - w^T\mathbf{x}$$

$$w^T\mathbf{x} = \mathbf{y}$$

When maximizing the likelihood function, one of the solutions is $\mathbf{y} = f(x)$ which is the same as our minimization of the sum-of-squares error function $w^T\mathbf{x} = \mathbf{y}$.

3. **MAP estimator** (15 points) Given input values $\mathbf{x} = (x_1, ..., x_N)^T$ and their corresponding target values $\mathbf{y} = (y_1, ..., y_N)^T$, we estimate the target by using function $f(x, \mathbf{w})$ which is a polynomial curve. Assuming the target variables are drawn from Gaussian distribution:

$$p(y|x, \mathbf{w}, \beta) = \mathcal{N}(y|f(x, \mathbf{w}), \beta^{-1})$$

and a prior Gaussian distribution for $\mathbf{w}$:

$$p(\mathbf{w}|\alpha) = (\frac{\alpha}{2\pi})^{(M+1)/2} \exp(-\frac{\alpha}{2}\mathbf{w}^T\mathbf{w})$$

Please prove that maximum posterior (MAP) is equivalent to minimizing the regularized sum-of-squares error function. Note that the posterior distribution of $\mathbf{w}$ is $p(\mathbf{w}|\mathbf{x}, \mathbf{y}, \alpha, \beta)$. **Hint: use Bayes' theorem.**

$$p(\mathbf{w}|\mathbf{x}, \mathbf{y}, \alpha, \beta) = \frac{p(\mathbf{y}|\mathbf{w}, \beta)p(\mathbf{w}|\alpha)}{p(\mathbf{y})}$$

$$= (\frac{\alpha}{2\pi})^{M+1} e^{\frac{\alpha \mathbf{w}^T \mathbf{w}}{2}} \cdot \frac{1}{2\pi\sigma^2}^{\frac{n}{2}} e^{\frac{-\sum (x_i - f(\mathbf{x}, \mathbf{w}))^2}{2\sigma^2}}$$

$$\log(P(\mathbf{w}|\mathbf{x}, \mathbf{y}, \alpha, \beta)) = \log[(\frac{\alpha}{2\pi})^{M+1} e^{\frac{\alpha \mathbf{w}^T \mathbf{w}}{2}} \cdot \frac{1}{2\pi\sigma^2}^{\frac{n}{2}} e^{\frac{-\sum (x_i - f(\mathbf{x}, \mathbf{w}))^2}{2\sigma^2}}]$$

$$\log(P(\mathbf{w}|\mathbf{x}, \mathbf{y}, \alpha, \beta)) = (M+1)\log(\frac{\alpha}{2\pi}) + \frac{\alpha \mathbf{w}^T \mathbf{w}}{2} - \frac{n}{2}\log(2\pi\sigma^2) - \sum \frac{(y_i - f(x, w))^2}{2\sigma^2}$$

$$\frac{d(\log(P(\mathbf{w}|\mathbf{x}, \mathbf{y}, \alpha, \beta)))}{dw} = \alpha \mathbf{w} - \frac{d\sum (y_i - f(x, w))^2}{dw}$$

The result is the same as the minimizing of the regularized sum-of-squares error function.

4. **Linear model** (20 points) Consider a linear model of the form:

$$f(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{i=1}^{D} w_i x_i$$

together with a sum-of-squares error/loss function of the form:

$$L_D(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{f(\mathbf{x}_n, \mathbf{w}) - y_n\}^2$$

Now suppose that Gaussian noise ϵ_i with zero mean and variance σ^2 is added independently to each of the input variables x_i. By making use of $\mathbb{E}[\epsilon_i] = 0$ and $\mathbb{E}[\epsilon_i \epsilon_j] = \delta_{ij}\sigma^2$, show that minimizing L_D averaged over the noise distribution is equivalent to minimizing the sum-of-squares error for noise-free input variables with the addition of a weight-decay regularization term, in which the bias parameter w_0 is omitted from the regularizer.

With the added noise:

$$\hat{f(\mathbf{x}_n}, \mathbf{w}) = w_0 + \sum_{i=1}^{D} w_i(x_{ni} + \epsilon_{ni}) = w_0 + \sum_{i=1}^{D} w_i x_{ni} + \sum_{i=1}^{D} w_i \epsilon_{ni} = f(\mathbf{x}_n, \mathbf{w}) + \sum_{i=1}^{D} w_i \epsilon_{ni}$$

New error function:

$$\hat{L_D(\mathbf{w})} = \frac{1}{2} \sum_{n=1}^{N} \{\hat{f(\mathbf{x}_n}, \mathbf{w}) - y_n\}^2$$

$$= \frac{1}{2} \sum_{n=1}^{N} \{f(\mathbf{x}_n, \mathbf{w}) + \sum_{i=1}^{D} w_i \epsilon_{id} - y_n\}^2$$

$$= \frac{1}{2} \sum_{i=1}^{N} \left\{ (f(\mathbf{x}_n, \mathbf{w}) - y_n)^2 + 2(f(\mathbf{x}_n, \mathbf{w}) - y_n) \left(\sum_{i=1}^{D} w_i \epsilon_{ni} \right) + \left(\sum_{i=1}^{D} w_i \epsilon_{ni} \right)^2 \right\}$$

Taking the expectation of this and using the linearity of expectation, we get

$$\mathbb{E}\left[\hat{L_D(\mathbf{w})}\right] = \frac{1}{2} \sum_{n=1}^{N} \left\{ (f(\mathbf{x}_n, \mathbf{w}) - y_n)^2 + 2(f(\mathbf{x}_n, \mathbf{w}) - y_n) \left(\sum_{i=1}^{D} w_i \mathbb{E}\left[\epsilon_{ni}\right] \right) + \mathbb{E}\left[\left(\sum_{i=1}^{D} w_i \epsilon_{ni} \right)^2 \right] \right\}$$

$\mathbb{E}[\epsilon_{id}]$ is 0, so the second term disappers. Now we look at the third term

$$\mathbb{E}\left[\left(\sum_{i=1}^{D} w_i \epsilon_{id} \right)^2 \right] = \mathbb{E}\left[\sum_{i=1}^{D} \sum_{i'=1}^{D} w_i w_{i'} \epsilon_{ni} \epsilon_{ni'} \right] = \sum_{i=1}^{D} \sum_{i'=1}^{D} w_i w_{i'} \mathbb{E}[\epsilon_{ni} \epsilon_{ni'}]$$

$$= \sum_{i=1}^{D} \sum_{i'=1}^{D} w_i w_{i'} \delta_{ii'} = \sum_{i=1}^{D} w_i^2$$

Using these results, we get

$$\mathbb{E}[\hat{L_D(\mathbf{w})}] = \frac{1}{2} \sum_{n=1}^{N} \left\{ (f(\mathbf{x}_n, \mathbf{w}) - y_n)^2 + \sum_{i=1}^{D} w_d^2 \right\} = L_D(\mathbf{w}) + \frac{N}{2} \sum_{i=1}^{D} w_i^2$$

and we see that we get a L_2 regularization term without the bias parameter w_0.

5. **Linear regression** (45 points) Please choose **one** of the below problems. You will need to **submit your code**.

   a) UCI Machine Learning: Facebook Comment Volume Data Set

   Please implement a Ridge regression model and use mini-batch gradient descent to train the model on this dataset for predicting the number of comments in next H hrs (H is given in the feature). You do not need to use all the features. Use K-fold cross validation and report the mean squared error (MSE) on the testing data. You need to write down every step in your experiment.

   a) UCI Machine Learning: Bike Sharing Data Set

   Please write a Ridge regression model and use mini-batch gradient descent to train the model on this dataset for predicting the count of total rental bikes including both casual and registered. You do not need to use all the features. Use K-fold cross validation and report the mean squared error (MSE) on the testing data. You need to write down every step in your experiment.

   The first step is to get the data and process it by doing feature selection. By using sklearn's package we can see that 1 of the features is not needed in predicting the comments in the next H hours. Then we need to use K-fold cross validation to tune the hyperparameter α for the regularization term. Once that has been tuned, we run the training data set through the Rigde regression model. Once the model has been trained we then use the test dataset to get the predictions and use mean squared error to see how well the model did.
   Since this model isn't using a package, we can expect the results to be higher then if we used the package due to package optimizations amoung other things. In my case, we get a MSE of 3,000 which is very high. This is primarly due to how well the hyperparameter was tuned as well as how well the model does in learning.