भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Department of Electrical Engineering
IIT Hyderabad
EE5610/AI5000 - PRML

Classtest - 1 Max. Marks: 40

1. Let the target variable t be regressed from the observed inputs $\mathbf{x}$ as $\hat{t} = y(\mathbf{x}, \mathbf{w}) = \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x})$. Let the model parameters be estimated from N training examples $(\mathbf{x}_{1:N}, t_{1:N})$, using least squares approach discussed in the class, and let us denote it with $\mathbf{w}^{(N)}$. Consider a scenario where we have access to M additional inputs points $(x_{N+1:N+M})$, i.e., the corresponding desired targets are not available for these M points. Let us use the model $\mathbf{w}^{(N)}$ trained on original N points to estimate the targets for the these M points, $(\hat{t}_{N+1:N+M})$. Let us mix the new inputs and their predicted targets $(x_{N+1:N+M}, \hat{t}_{N+1:N+M})$ with the original N points to create a bigger dataset with $N + M$ points. Let us use this augmented dataset to train a new model $\mathbf{w}^{(N+M)}$. Does the new model $\mathbf{w}^{(N+M)}$ perform better than or worse than or similar the original one $\mathbf{w}^{(N)}$. Provide mathematical justifications for your arguments. (10)

2. Suppose you are experimenting with L_1 and L_2 regularization. Further, imagine that you are running gradient descent and at some iteration your weight vector is $w = [1, \epsilon] \in \mathbb{R}^2$ where $\epsilon > 0$ is very small. With the help of this example explain why L_2 norm does not encourage sparsity i.e., it will not try to drive ϵ to 0 to produce a sparse weight vector. Give mathematical explanation.

3. When we perform least squares linear regression, we make certain idealized assumptions about the errors e_n, namely, that it is distributed $\mathcal{N}(0, \beta^{-1})$. In practice, departures from these assumptions occur. Particularly, in cases where the error distribution is heavier tailed than the Normal distribution (i.e. has more probability in tails than the Normal).

The least square loss is sensitive to outliers, and hence robust regression methods are of interest. The problem with the least square loss in the existence of outliers (i.e. when the noise term e_n can be arbitrarily large), is that it weighs each observation equally in getting parameter estimates. Robust methods, on the other hand, enable the observations to be weighted unequally. More specifically, observations that produce large residuals are down-weighted by a robust estimation method.

In this problem, you will assume that e_n are independent and identically distributed according to a Laplacian distribution, rather than Gaussian. That is, each $e_n \sim \text{Laplace}(0, b) = \frac{1}{2b}\exp(-\frac{e_n}{b})$.
(a) Provide the loss function $J_L(\mathbf{w})$ whose minimization is equivalent to finding the ML estimate under the Laplacian noise model.
(b) Suggest a method for optimizing the objective function in (a), and write the update equations.
(c) Why do you think that the above model provides a more robust fit to data compared to the standard model assuming Gaussian distribution of the noise terms?

4. When we have multiple independent outputs $\mathbf{t} = [t_1 t_2 \cdots t_K]^T$ in linear regression, the model becomes

$$p(\mathbf{t}|\mathbf{x}, \mathbf{W}) = \prod_{k=1}^{K} \mathcal{N}(t_k|\mathbf{w}_k^T \boldsymbol{\phi}_k(\mathbf{x}), \beta_k^{-1})$$

Since the likelihood factorizes across dimensions, so does ML estimates. Thus

$$\mathbf{W} = [\mathbf{w}_1 \mathbf{w}_2 \cdots \mathbf{w}_K]$$

where $\mathbf{w}_k$ can be estimated using the method discussed in the class from the $k-$th dimension of the data. Suppose we have the training data as follows

x	**t**
0	$[-1 \quad -1]^T$
0	$[-1 \quad -2]^T$
0	$[-2 \quad -1]^T$
1	$[1 \quad 1]^T$
1	$[1 \quad 2]^T$
1	$[2 \quad 1]^T$

Let each input x_n be embedded into a 2-D space using the following basis functions:

$$\boldsymbol{\phi}(0) = [1 \quad 0]^T \qquad \boldsymbol{\phi}(1) = [0 \quad 1]^T$$

The estimate of the output vector becomes

$$\hat{\mathbf{t}}_n = \mathbf{W}^T \boldsymbol{\phi}(x_n)$$

where **W** is a 2x2 matrix. Compute ML estimate for **W** from the above data.