EECS 189 Introduction to Machine Learning
Fall 2020

DIS3

This discussion was released **Friday, September 18**.

This discussion material will cover parts of Problem 2 and 4 in HW3. It aims to give you a better understanding of an interpretation of regression as a probabilistic model, and it should get you started on the homework problems. We will first start with this Jupyter notebook and play around with multivariate Gaussian distribution (2D). We will consider a linear regression problem from a probabilistic view and examine the likelihood function and the posterior function.

As a reminder, if you have questions, we will answer them via the queue at oh.eecs189.org. Once you complete the Jupyter notebook, please return to this worksheet and get started on the following exercises extracted from Problem 2 and 4 of HW3.

1 Jupyter Notebook

Solution:

Solution of the discussion questions from the Jupyter notebook can be found via this **[link]**.

2 Probabilistic Model of Linear Regression

Part (g)

(Multi-dimensional ridge regression) Consider the following setup (a bit more simplified version of the Jupyter notebook): $Y_i = \mathbf{w}^\top \mathbf{x}_i + Z_i$, where $Y_i \in \mathbb{R}, \mathbf{w}, \mathbf{x}_i \in \mathbb{R}^d$, and $Z_i \sim N(0,1)$ iid standard normal random variables. Now we treat $\mathbf{w}$ as a random vector and assume a prior knowledge about its distribution. In particular, we use the prior information that the random variables W_j are i.i.d. $\sim N(0, \sigma^2)$ for $j = 1, 2, \ldots, d$. **Derive the posterior distribution of w given all the $\mathbf{x}_i, Y_i$ pairs. What is the mean of the posterior distribution of the random vector w?** Hint: Compute the posterior up-to proportionality, i.e. you may discard terms that do not depend on $\mathbf{w}$ to simplify the algebra. After a few steps, you should be able to identify the family of the distribution of the posterior. Then, you can determine the mean and the variance by completing the square. We find that it is simpler to work in matrix and vector format: $\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^\top \\ \vdots \\ \mathbf{x}_n^\top \end{bmatrix}$ and $\mathbf{Y} = \begin{bmatrix} Y_1 \\ \vdots \\ Y_n \end{bmatrix}$

3 Orthonormalization of features

Note: This problem has a companion Jupyter notebook. Most of the parts have a corresponding demo/visualization in the Jupyter notebook but for parts (c) and (g) you are required to fill part of the code in the notebook. Consider the problem setting of learning a 1-dimensional function $f(x)$ from noisy samples $(x_i, y_i)_{i=1,\ldots,n}$ like in Problem 8 of HW 1. We perform linear regression in feature space with the featurization $\phi : \mathbb{R} \to \mathbb{R}^d$ to learn coefficients $\mathbf{w}$. On a given test point x, we predict

$$\hat{f}(x) = \phi(x)^\top \mathbf{w}.$$

To evaluate our learned predictor we care about its "test performance", the average performance on data coming from our test distribution. Next we will formalize this.
Let X, Y denote the random variables corresponding to the test data. Further suppose that $Y = f(X)$, i.e each randomly sampled test point comes along with its true function value (there is no additional randomness in the Y at test time due to noise). Let X have the probability density function p, i.e $X \sim p$. Thus the test data is $(X, f(X))$ and its distribution is determined by p. On the random test point X, we make the prediction $Y_{\text{pred}} = \hat{f}(X) = \phi(X)^\top \mathbf{w}$. Note: The test data distribution can be different than the train data distribution. One such case is if the training points x_i were not sampled from the same distribution p as the test point. Even if the training points were sampled from p, the training labels y_i may not be the true function value at points x_i due to the presence of noise and thus the test and train data distributions typically vary in practice. For this question we assume that the training points and consequently the learned coefficients $\mathbf{w}$ are fixed and not random. We define the prediction error/test error as,

$$\mathcal{E}_{\text{pred}} = \mathbb{E}(Y_{\text{pred}} - Y)^2 = \mathbb{E}(\phi(X)^\top \mathbf{w} - f(X))^2 = \int_x (\phi(x)^\top \mathbf{w} - f(x))^2 p(x) dx$$

Note: The expectation is over the randomness due to the test data point X. Nothing else is random at test time here. We are using squared loss here because it is convenient and has nice geometry. By changing the loss function we can have other measures of prediction performance.

(a) **Prediction error vs parameter estimation error.**

Suppose the true function can be exactly represented in the feature space, $f(X) = \phi(X)^\top \mathbf{w}^*$ for some $\mathbf{w}^* \in \mathbb{R}^d$. In this case we can define another metric of test performance, the parameter estimation error given by

$$\mathcal{E}_{\text{est}} = \mathbb{E}\left\|\mathbf{w} - \mathbf{w}^*\right\|_2^2 = \left\|\mathbf{w} - \mathbf{w}^*\right\|_2^2,$$

since neither $\mathbf{w}$ nor $\mathbf{w}^*$ vary based on the test point X. Estimation error measures how well we are estimating the paramters of the function in feature space while test error measures how close the values predicted by the learned function are to the true function values.

Show that if $\mathbb{C} = \mathbb{E}[\phi(X)\phi(X)^\top] = \mathbf{I}_d$, then $\mathcal{E}_{\text{pred}} = \mathcal{E}_{\text{est}}$. In this case, the features are orthonormal with respect to the test distribution.

(b) The entries of the "covariance" matrix are given by,

$$C_{ij} = \mathbb{E}[\phi_i(X)^\top \phi_j(X)] = \int_x \phi_i(x)^\top \phi_j(x) p(x) dx,$$

where $\phi(x) = [\phi_1(x), \phi_2(x), \ldots, \phi_d(x)]^\top$.
Consider a 3-dimensional polynomial featurization $\phi(x) = [1, x, x^2]^\top$ and let $p = \text{Uniform}[-1, 1]$.
Calculate C. Are the features orthonormal?

Contributors:

- Anant Sahai
- Chawin Sitawarin
- Peter Wang
- Raaz Dwivedi
- Rahul Arya
- Vignesh Subramanian