CS6140: Machine Learning

Homework Assignment # 2

Assigned: 02/16/2021 *Due: 03/01/2021, 11:59pm, through Canvas*

Three problems, 100 points in total. Good luck!
Prof. Predrag Radivojac, Northeastern University

Problem 1. (20 points) Naive Bayes classifier. Consider a binary classification problem where there are eight data points in the training set. That is,

$$\mathcal{D} = \{(-1,-1,-1,-), (-1,-1,1,+), (-1,1,-1,+), (-1,1,1,-), (1,-1,-1,+), (1,-1,1,-), (1,1,-1,-), (1,1,1,+)\},$$

where each tuple (x_1, x_2, x_3, y) represents a training example with input vector (x_1, x_2, x_3) and class label y.

a) (10 points) Construct a naive Bayes classifier for this problem and evaluate its accuracy on the training set. Measure accuracy as the fraction of correctly classified examples.

b) (10 points) Transform the input space into a higher-dimensional space

$$(x_1, x_2, x_3, x_1x_2, x_1x_3, x_2x_3, x_1x_2x_3, x_1^2, x_2^2, x_3^2, x_1^2x_2, x_1x_2^2, x_1x_3^2, x_2^2x_3, x_2x_3^2)$$

and repeat the previous step.

Carry out all steps manually and show all your calculations. Discuss your main observations.

Problem 2. (25 points) Consider a binary classification problem in which we want to determine the optimal decision surface. A point $\mathbf{x}$ is on the decision surface if $P(Y=1|\mathbf{x}) = P(Y=0|\mathbf{x})$.

a) (10 points) Find the optimal decision surface assuming that each class-conditional distribution is defined as a two-dimensional Gaussian distribution:

$$p(\mathbf{x}|Y=i) = \frac{1}{(2\pi)^{d/2}|\mathbf{\Sigma}_i|^{1/2}} \cdot e^{-\frac{1}{2}(\mathbf{x}-\mathbf{m}_i)^T\mathbf{\Sigma}_i^{-1}(\mathbf{x}-\mathbf{m}_i)}$$

where $i \in \{0,1\}$, $\mathbf{m}_0 = (1,2)$, $\mathbf{m}_1 = (6,3)$, $\mathbf{\Sigma}_0 = \mathbf{\Sigma}_1 = \mathbf{I}_2$, $P(Y=0) = P(Y=1) = 1/2$, $\mathbf{I}_d$ is the d-dimensional identity matrix, and $|\mathbf{\Sigma}_i|$ is the determinant of $\mathbf{\Sigma}_i$.

b) (5 points) Generalize the solution from part (a) using $\mathbf{m}_0 = (m_{01}, m_{02})$, $\mathbf{m}_1 = (m_{11}, m_{12})$, $\mathbf{\Sigma}_0 = \mathbf{\Sigma}_1 = \sigma^2\mathbf{I}_2$ and $P(Y=0) \neq P(Y=1)$.

c) (10 points) Generalize the solution from part (b) to arbitrary covariance matrices $\mathbf{\Sigma}_0$ and $\mathbf{\Sigma}_1$. Discuss the shape of the optimal decision surface.

Problem 3. (55 points) Consider a multivariate linear regression problem of mapping $\mathbb{R}^d$ to $\mathbb{R}$, with two different objective functions. The first objective function is the sum of squared errors, as presented in class; i.e., $\sum_{i=1}^n e_i^2$, where $e_i = w_0 + \sum_{j=1}^d w_j x_{ij} - y_i$. The second objective function is the sum of square Euclidean distances to the hyperplane; i.e., $\sum_{i=1}^n r_i^2$, where r_i is the Euclidean distance between point (x_i, y_i) to the hyperplane $f(x) = w_0 + \sum_{j=1}^d w_j x_j$.

a) (10 points) Derive a gradient descent algorithm to find the parameters of the model that minimizes the sum of squared errors.

b) (20 points) Derive a gradient descent algorithm to find the parameters of the model that minimizes the sum of squared distances.

c) (20 points) Implement both algorithms and test them on 3 datasets. Datasets can be randomly generated, as in class, or obtained from resources such as UCI Machine Learning Repository. Compare the solutions to the closed-form (maximum likelihood) solution derived in class and find the R^2 in all cases on the same dataset used to fit the parameters; i.e., do not implement cross-validation. Briefly describe the data you use and discuss your results.

d) (5 points) Normalize every feature and target using a linear transform such that the minimum value for each feature and the target is 0 and the maximum value is 1. The new value for feature j of data point i can be found as

$$x_{ij}^{\text{new}} = \frac{x_{ij} - \min_{k\in\{1,2,\ldots,n\}} x_{kj}}{\max_{k\in\{1,2,\ldots,n\}} x_{kj} - \min_{k\in\{1,2,\ldots,n\}} x_{kj}},$$

where n is the dataset size. The new value for the target i can be found as

$$y_i^{\text{new}} = \frac{y_i - \min_{k\in\{1,2,\ldots,n\}} y_k}{\max_{k\in\{1,2,\ldots,n\}} y_k - \min_{k\in\{1,2,\ldots,n\}} y_k}.$$

Measure the number of steps towards convergence and compare with the results from part (c). Briefly discuss your results.

Directions and Policies

Submit a single package containing all answers, results and code. Your submission package should be compressed and named firstnamelastname.zip (e.g., predragradivojac.zip). In your package there should be a single pdf file named main.pdf that will contain answers to all questions, all figures, and all relevant results. Your solutions and answers must be typed[1] and make sure that you type your name and Northeastern username (email) on top of the first page of the main.pdf file. The rest of the package should contain all code that you used. The code should be properly organized in folders and subfolders, one for each question or problem. All code, if applicable, should be turned in when you submit your assignment as it may be necessary to demo your programs to the teaching assistants. Use Matlab, Python, R, Java, or C/C++. However, you are encouraged to use languages with good machine learning libraries (e.g., Matlab, Python, R), which may be handy in future assignments.

Unless there are legitimate circumstances, late assignments will be accepted up to 5 days after the due date and graded using the following rules:

on time: your score $\times$ 1

1 day late: your score $\times$ 0.9

2 days late: your score $\times$ 0.7

3 days late: your score $\times$ 0.5

4 days late: your score $\times$ 0.3

5 days late: your score $\times$ 0.1

For example, this means that if you submit 3 days late and get 80 points for your answers, your total number of points will be $80 \times 0.5 = 40$ points.

All assignments are individual, except when collaboration is explicitly allowed. **All the sources used for problem solution must be acknowledged**; e.g., web sites, books, research papers, personal communication with people, etc. Academic honesty is taken seriously! For detailed information see Office of Student Conduct and Conflict Resolution.

[1] We recommend Latex; in particular, TexShop-MacTeX combination for a Mac and TeXnicCenter-MiKTex combination on Windows. An easy way to start with Latex is to use the freely available Lyx. You can also use Microsoft Word or other programs that can display formulas professionally.