Machine Learning and Data Mining I (DS 4400)
Homework 3

Instructor: Ehsan Elhamifar

Analytical: Due Date: March 20, 2019, 1:35pm
Programming: Due Date: March 23, 2019, 11:59pm

1. MAP estimation. Consider a Bernoulli random variable x with $p(x = 1) = \theta$. Given a dataset $D = \{x_1, \ldots, x_N\}$, assume N_1 is the number of trials where $x_i = 1$, N_0 is the number of trials where $x_i = 0$ and $N = N_0 + N_1$ is the total number of trials. Consider the following prior, that believes the experiments is biased:

$$p(\theta) = \begin{cases} 0.2 & \text{if } \theta = 0.6 \\ 0.8 & \text{if } \theta = 0.8 \\ 0 & \text{otherwise} \end{cases}$$

1. Write down the likelihood function, i.e., $p(D|\theta)$. What is the maximum likelihood solution for θ (we already have derived this in the class)?

2. Consider maximizing the posterior distribution, $p(D|\theta) \times p(\theta)$, that takes advantage of the prior. What is the MAP estimation?

2. Naive Bayes Classifier. Assume you have the following training set with two binary features x_1 and x_2, and a binary response/output y. Suppose you have to predict y using a naive Bayes classifier.

x_1	x_2	y
1	0	0
0	1	0
0	0	0
1	0	1
0	0	1
0	1	1
1	1	1

1. Compute the Maximum Likelihood Estimates (MLE) for θ_j^y for $j = 0, 1$ as well as $\theta_{\bar{x}_\ell|j}^{x_\ell|y}$ for $j = 0, 1$ and $\bar{x}_\ell = 0, 1$ and for $\ell = 1, 2$.

2. After learning via MLE is complete, what would be the estimate for $P(y = 0|x_1 = 0, x_2 = 1)$.

3. What would be the solution of the previous part without the naive Bayes assumption?

3. Constrained Optimization. Consider the regression problem on a dataset $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$, where $\boldsymbol{x}_i \in \mathbb{R}^k$ denotes the input and y_i denotes the output/response. Let $\boldsymbol{X} = \begin{bmatrix}\boldsymbol{x}_1 & \dots & \boldsymbol{x}_N\end{bmatrix}^\top$ and $\boldsymbol{y} = \begin{bmatrix}\boldsymbol{y}_1 & \dots & \boldsymbol{y}_N\end{bmatrix}^\top$. Consider the following regression optimization

$$\min_{\boldsymbol{\theta}} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}\|_2^2$$
$$\text{s.t.} \quad \boldsymbol{w}^\top \boldsymbol{\theta} = b,$$

where $\boldsymbol{w}$ and b are given and indicate the parameters of a hyperplane on which the desired parameter vector, $\boldsymbol{\theta}$, lies.
a) Assume that $\boldsymbol{X}^\top \boldsymbol{X} = \boldsymbol{I}_k$, where $\boldsymbol{I}_k$ denotes the identity matrix. Find the closed-form solution of the above regression problem.
b) Verify if your obtained solutionn $\boldsymbol{\theta}^*$ satisfies the constraint $\boldsymbol{w}^\top \boldsymbol{\theta}^* = b$.
c) What you have been the solution of this optimization, if the constraint $\boldsymbol{w}^\top \boldsymbol{\theta} = b$ was not present?

4. Logistic Regression Implementation. For this exercise, you can use Scikit-Learn in Python.
a) Write down a code in Python whose input is a training dataset $\{(\boldsymbol{x}^1, y^1), \dots, (\boldsymbol{x}^N, y^N)\}$ and its output is the weight vector $\boldsymbol{w}$ and bias b in the logistic regression model $y = \sigma(\boldsymbol{w}^\top \boldsymbol{x} + b)$.
b) Download the dataset of the HW03 on the piazza page. Run the code on the training dataset to compute $\boldsymbol{w}, b$ and evaluate on the test dataset. Report the classification error on the training set and classification error on the test set. Plot the data (use different colors for data in different classes) and plot the decision boundary found by the logistic regressions.

5. SVM Implementation. For this exercise, you can use Scikit-Learn in Python.
a) Write a code in Python that takes as input a training dataset $\{(\boldsymbol{x}_i, y_i)\}_{i=1}N$, a testing set $\{(\boldsymbol{x}_i^n)\}_{i=1}N'$, and outputs the parameters of the hyperplane and classification of the training data and test data in to the class $\{-1, +1\}$. Here, we only use linear SVM.
b) Download the dataset of the HW03 on the piazza page. Run the code on the training dataset to compute the hyperplane and evaluate on the test dataset. Report classification error on the training set and classification error on the test set. Plot the data (use different colors for data in different classes) and plot the decision boundary found by the SVM.

Homework Submission Instructions: Please submit the analytical part of the homework in class (put a paper copy of the HW on the instructor's desk before the class starts). Handwritten or printed documents generated by Latex/Word are both acceptable. Please submit all your plots and your Python code (.py file) via email, by the DEADLINE. To submit, please send an email to the instructor and cc the TA (huynh.dat [AT] husky [Dot] neu [Dot] edu).

– The title of your email must be "DS4400: HW03:Your-Last-Name".

– Please attach a single zip file to your email that contains all python codes and plots and a readme file on how to run your files.

– Please name your zip file as "HW03:Your-Last-Name".