Homework 1 Solutions
Probability, Maximum Likelihood Estimation (MLE), Bayes Rule, kNN

CMU 10-701: Machine Learning (Fall 2016)
https://piazza.com/class/is95mzbrvpn63d
OUT: September 13th
DUE: September 26th, 11:59 PM

START HERE: Instructions

- **Collaboration policy:** Collaboration on solving the homework is allowed, after you have thought about the problems on your own. It is also OK to get clarification (but not solutions) from books or online resources, again after you have thought about the problems on your own. There are two requirements: first, cite your collaborators fully and completely (e.g., "Jane explained to me what is asked in Question 3.4"). Second, write your solution *independently*: close the book and all of your notes, and send collaborators out of the room, so that the solution comes from you only.
- **Late Submission Policy:** You will be allowed 2 total late days without penalty for the entire semester. You may be late by 1 day on two different homeworks or late by 2 days on one homework. Weekends and holidays are also counted as late days. Late submissions are automatically considered as using late days.
  Once those days are used, you will be penalized according to the following policy:
  - Homework is worth full credit before the deadline.
  - It is worth half credit for the next 24 hours.
  - It is worth zero credit after that.

  You must turn in at least $n-1$ of the n homeworks, even if for zero credit, in order to pass the course.
- **Submitting your work:** Assignments should be submitted as PDFs using Gradescope unless explicitly stated otherwise. Each derivation/proof should be completed on a separate page. Submissions can be handwritten, but should be labeled and clearly legible. Else, submissions can be written in LaTeX. Upon submission, label each question using the template provided by Gradescope.
- **Programming**: All programming portions of the assignments should be submitted to Autolab. We will not be using this for autograding, but rather for plagiarism detection, meaning you may use any language which you like to submit.

Problem 1: Background Material [20pts] (Devendra & Hyun-Ah)

1.1 Probability review (Devendra)

Consider a naive classifier which stochastically assigns labels to data points; specifically, for each label l, it assigns that label with probability proportional to the number of times that label appears in the training dataset. Let the size of training set be N, total number of classes be C and n_i be number of datapoints of class i ($\sum_i^C n_i = N$), then the probability of labelling a datapoint with class i is n_i/N.

1) Consider a training set with $N = 100$, $C = 2$, $n_1 = 50$, $n_2 = 50$ and a test set with, $N = 100$, $C = 2$, $n_1 = 30$, $n_2 = 70$.
a) What is the expected accuracy of the classifier on the training set?
b) What is the expected accuracy of the classifier on the test set?

2) Now consider a training set with $N = 100$, $C = 3$, $n_1 = 50$, $n_2 = 20$, $n_3 = 30$ and a test set with, $N = 100$, $C = 3$, $n_1 = 20$, $n_2 = 20$, $n_3 = 60$.
a) What is the expected accuracy of the classifier on the training set?
b) What is the expected accuracy of the classifier on the test set?

This kind of analysis can be used to calculate a baseline accuracy for various Machine Learning algorithms.

Problem 1.2: Bayes Rule [10 pts] (Hyun-Ah)

Smith owns a retail store for selling phones. The phones are manufactured at three different factories: A, B, C. Factory A, B, and C produces $20\%, 30\%$, and 50% of the phone being sold at Smith's store. The probabilities of the defective phones from stores A, B, and C are $2\%, 1\%$, and 0.05%, respectively. The total number of phones being sold at Smith's store is 10000. One day, a customer walks up to Smith's store, and ask for a refund for a defective phone.

a) What is the probability of a phone being defective?

b) What is the probability that this defective phone is manufactured at factory A?

c) What is the probability that this defective phone is manufactured at factory B?

d) What is the probability that this defective phone is manufactured at factory C?

Problem 2: kNN [30 pts] (Hemank)

In this problem, you will implement K-Nearest Neighbor approach on a synthetic dataset and on the provided dataset[1]. We will figure out how does various parameters affect the performance of K-NN.

k-Nearest Neighbor

Implement the k-nn algorithm and answer the following questions. We will use Euclidean distance for our implementation. For validation, we use N-fold cross validation($N = 10$).

Input Data: We will generate synthetic input data. The input data will be generated using the code provided in the folder. The code is provided in R, Matlab and Python syntax. The code takes as input the following:

n is the number of samples, p is the dimensionality of the data and $sigma$ determines the amount of noise.

1. Why is it suitable to choose odd-k?
2. Generate $n = 1000$ points with $p = 10$ and $sigma = 0.001$. For different values of k, ranging from 1 to 50 (in increments of 5), plot the cross-validation train-error and test-error. What can you say about the effect of k? What is the optimal value of k, which you can infer from the plot? How does performance changes when you move from smaller values of k to larger values?
3. Generate $n = 400$ points with $sigma = 0.001$. Vary p in range of 2 to 50. Run k-NN with $k = 9$. Plot the cross-validation test-error and comment about effect of dimensionality on the algorithm.
4. Generate $n = 1000$ points with $sigma = 0.001$ and $p = 10$. Run k-NN with $k = 9$. Vary the N for N-fold validation from 2 to 10 and plot the cross-validation test error. What do you observe from the plot?
5. On the given iris dataset, plot with varying $K = 1, 3, 5, 7$, the N-fold cross-validation test-error.

[1] http://archive.ics.uci.edu/ml/datasets/Iris

Problem 3: Ridge Regression [40 pts] (Hyun-Ah & Petar)

3.1 MLE (Petar)

We are given covariates $\mathbf{X} \in \mathbb{R}^{n \times d}$ and responses $\mathbf{y} \in \mathbb{R}^{n \times 1}$, such that:

$$\mathbf{y} = \mathbf{X}\beta + \epsilon \tag{1}$$

where $\beta \in \mathbb{R}^{d \times 1}$ is a vector of predictors and ϵ a vector of Gaussian noise components given by: $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$.

a) Derive the conditional probability $p(y_i|\mathbf{x_i}, \beta, \sigma)$, where y_i is the i-th component of $\mathbf{y}$, and $\mathbf{x_i}$ is the i-th row of $\mathbf{X}$.

b) Assume we have n i.i.d samples. Derive the log-likelihood of the data $\ell(\mathbf{y}|\beta)$.

c) Show that maximizing the log-likelihood is equivalent to the following criterion:

$$\min_{\beta} ||\mathbf{y} - \mathbf{X}\beta||_2^2 \tag{2}$$

d) Derive the MLE $\hat{\beta}$.

We have worked through the linear regression problem. Now, we will explore linear regression problem with regularization.

3.2 Ridge regression

Continued from Problem 1, we are given a matrix of n training samples with p dimensions, $\mathbf{X} \in \mathbb{R}^{n \times p}$, a parameter vector $\beta \in \mathbb{R}^p$, and a output vector $\mathbf{y} \in \mathbb{R}^n$. L2 penalized linear regression problem shown below penalizes the L2 norm of the parameter vector. This is also known as the ridge regression problem.

$$\min_{\beta} \frac{1}{2}||\mathbf{y} - \mathbf{X}\beta||_2^2 + \frac{\lambda}{2}||\beta||_2^2$$

a) Show that the solution to the ridge regression problem is $\beta^* = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$.

b) How does the solution differ from the solution to the ordinary linear regression problem? Explain how the solution changes as $\lambda \to 0$, and $\lambda \to \infty$.

Problem 4: Naive Bayes: Theory [20 pts] (Brynn)

Recall the difference in the modeling assumptions in a discriminative and a generative classifier. In a generative model, $P(X, Y)$ is estimated by initially modeling the conditional $P(X|Y)$, since, $P(X, Y) = P(Y)P(X|Y)$. The joint probability with Bayes rule is then used to calculate $P(Y|X)$ for each class label. On the other hand, a discriminative classifier directly estimates $P(Y|X)$. In this problem, we will explore the relation between Naive Bayes and logistic regression classifiers, by focusing on the class conditional $P(Y|X)$.

When Y is Boolean and $X = \langle X_1 ... X_n \rangle$ is a vector of continuous variables, and each $P(X_i|Y = y_k)$ is modelled with a Gaussian distribution, then the assumptions of the Gaussian Naive Bayes classifier imply that $P(Y \mid X)$ is given by the logistic function with appropriate parameters $w_0, w_1, ..w_n$. In particular:

$$P(Y = 1 \mid X) = \frac{1}{1 + exp(w_0 + \sum_{i=1}^{n} w_i X_i)}$$

and

$$P(Y = 0 \mid X) = \frac{exp(w_0 + \sum_{i=1}^{n} w_i X_i)}{1 + exp(w_0 + \sum_{i=1}^{n} w_i X_i)}$$

1. Consider instead the case where Y is Boolean and $X = \langle X_1 ... X_n \rangle$ is a vector of Boolean variables. Show that the derived expression of $P(Y|X)$ has the same form as that in the Logistic Regression classifier model (by getting an appropriate expression, which can be substituted for weights in the $P(Y|X)$ equation for Logistic Regression).

   Hints

   (a) Simple notation will help. Since the X_i are Boolean variables, you need only one parameter to define $P(X_i \mid Y = y_k)$. Define $\theta_{i1} \equiv P(X_i = 1 \mid Y = 1)$, in which case $P(X_i = 0 \mid Y = 1) =$ (1-θ_{i1}). Similarly, use θ_{i0} to denote $P(X_i = 1|Y = 0)$.

   (b) Notice with the above notation you can represent $P(Xi \mid Y = 1)$ as follows

   $$P(X_i \mid Y = 1) = \theta_{i1}^{(X_i)}(1 - \theta_{i1})^{(1-X_i)}$$

   Note when $X_i = 1$ the second term is equal to 1 because its exponent is zero. Similarly, when $X_i = 0$ the first term is equal to 1 because its exponent is zero.

Problem 5: Naive Bayes: Implementation [30 pts] (Siddharth)

As part of this problem, you will implement a Naive Bayes classifier for classifying movie reviews as positive or negative. The dataset that you will be using is the IMDB Large Movie Review dataset (Maas et. al, ACL 2011). The processed dataset can be found *here*. The task is to estimate appropriate parameters using the training data, and use it to predict reviews from the test data, and classify each of them as either positive or negative.

We employ the Multinomial Naive Bayes model for modeling each $P(X_i|Y = y_k)$ $(i = 1..n)$, with appropriate word counts (Note n is the number of dimensions).

Please use Matlab, Python, R, C/C++ or Java for your implementation. Note that you will have to submit your codes in Autolab, and provide the answers to the questions in the below subsections in your report.

Preprocessing

The dataset is partitioned into 2 folders: 'train' and 'test', each of which contains 2 subfolders ('pos' and 'neg', for positive and negative samples respectively). The content of each file has to be converted to a bag-of-words representation. So the first task is to go through all the files in the 'train' folder, and construct the vocabulary V of all unique words. Please ignore all the stop-words as given in the file 'sw.txt' (provided along with the dataset). The words from each file (both in training and testing phase) must be extracted by splitting the raw text only with whitespace characters and converting them to lowercase characters.

The next step is to get counts of each individual words for the positive and the negative classes separately, to get $P(word|class)$.

Classification

In this step, you need to go through all the negative and positive samples in the test data, and classify each sample according to the parameters learned earlier. The classification should be done by comparing the log-posterior (un-normalized), which is given by $\log(P(X|Y)P(Y))$, for both the classes.

Laplace smoothing

An issue with the original Naive Bayes setup is that if a test sample contains a word which is not present in the dictionary, the $P(word|label)$ goes to 0. To mitigate this issue, one solution is to employ Laplace smoothing (it has a parameter α). Augment your $P(word|class)$ calculations by including the appropriate terms for doing Laplace smoothing.

Report the confusion matrix and overall accuracy of your classifier on the test dataset with $\alpha = 1$. Recall that the confusion matrix for such 2-class classification problem, is a matrix of the number of true positives (positive samples correctly classified as positive), number of true negatives (negative samples correctly classified as negative), number of false positives (negative samples incorrectly classified as positive), and number of false negatives (positive samples incorrectly classified as negative). The accuracy is the ratio of sum of true positives and true negatives, and the total number of samples (in the test dataset).

Now vary the value of α from 0.0001 to 1000 (by multiplying α with 10 each time), and report a plot of the accuracy on the test dataset for the corresponding values of α. (The x-axis should represent α values and use a log scale for the x-axis).

Why do you think the accuracy suffers when α is too high or too low?