COMS W4721: Machine Learning for Data Science

Columbia University, Spring 2019

Homework 2: Due Wednesday, March 6, 2019 by 11:59PM

Please read these instructions to ensure you receive full credit on your homework. Submit the written portion of your homework as a *single* PDF file through Courseworks (less than 5MB). In addition to your PDF write-up, submit all code written by you in their original extensions through Courseworks (e.g., .m, .r, .py, .c). Any coding language is acceptable, but do not submit notebooks, do not wrap your files in .rar, .zip, .tar and do not submit your write-up in .doc or other file type. When resubmitting homeworks, be sure to resubmit *all files*. Your grade will be based on the contents of one PDF file and the original source code. Additional files will be ignored. We will not run your code, so everything you are asked to show should be put in the PDF file. Show all work for full credit.

Late submission policy: Late homeworks will have 0.1% deducted from the final grade for each minute late. *Your homework submission time will be based on the time of your **last** submission to Courseworks. I will not revert to an earlier submission!* Therefore, do not re-submit after midnight on the due date unless you are confident the new submission is significantly better to overcompensate for the points lost. Submission time is non-negotiable and will be based on the time you submitted your last file to Courseworks. The number of points deducted will be rounded to the nearest integer.

Problem 1 – 30 points

In this problem you will derive a naive Bayes classifier. For a labeled set of data $(y_1, x_1), \dots, (y_n, x_n)$, where for this problem $y \in \{0, 1\}$ and x is a D-dimensional vector of counts, the Bayes classifier observes a new x_0 and predicts y_0 as

$$y_0 \;=\; \arg\max_y \; p(y_0 = y|\pi) \textstyle\prod_{d=1}^{D} p(x_{0,d}|\lambda_{y,d}).$$

The distribution $p(y_0 = y|\pi) = \text{Bernoulli}(y|\pi)$. What is "naive" about this classifier is the assumption that all D dimensions of x are independent. Assume that each dimension of x is Poisson distributed with a Gamma prior. The full generative process is

$$\text{Data: } y_i \overset{iid}{\sim} \text{Bern}(\pi), \quad x_{i,d}|y_i \sim \text{Pois}(\lambda_{y,d}),\; d = 1, \dots, D \qquad \text{Prior: } \lambda_{y,d} \overset{iid}{\sim} \text{Gamma}(2, 1)$$

Derive the solution for π and each $\lambda_{y,d}$ by maximizing

$$\widehat{\pi},\, \widehat{\lambda}_{y,d} \;=\; \arg\max_{\pi, \lambda_{y,d}} \; \sum_{i=1}^{n} \ln p(y_i|\pi) + \sum_{d=1}^{D} \left(\ln p(\lambda_{y,d}) + \sum_{i=1}^{n} \ln p(x_{i,d}|\lambda_{y,d}) \right).$$

Please separate your derivations as follows:

(a) Derive $\widehat{\pi}$ using the objective above.

(b) Derive $\widehat{\lambda}_{y,d}$ using the objective above. Derive this leaving y arbitrary.

Problem 2 – 45 points

In this problem you will implement the naive Bayes classifier derived in Problem 1, as well as the k-NN algorithm and logistic regression algorithm. The data consists of examples of spam and non-spam emails, of which there are 4600 labeled examples. The feature vector x is a 54-dimensional vector extracted from the email and $y = 1$ indicates a spam email.[1]

In every experiment below, *randomly* partition the data into 10 groups and run the algorithm 10 different times so that each group is held out as a test set one time. The final result you show should be the cumulative result across these 10 groups.

(a) Implement the naive Bayes classifier described above. In a 2×2 table, write the number of times that you predicted a class y data point (ground truth) as a class y' data point (model prediction) in the (y, y')-th cell of the table, where y and y' can be either 0 or 1. There should be four values written in the table in your PDF. Next to your table, write the prediction accuracy—the sum of the diagonal divided by 4600. (The sum of all entries in the table should be 4600.)

(b) In one figure, show a stem plot (`stem()` in Matlab) of the 54 Poisson parameters for each class averaged across the 10 runs. (This average is only used for plotting purposes on this homework. In practice you would relearn these parameters using the entire data set to find their final values.) Use the README file to make an observation about dimensions 16 and 52.

(c) Implement the k-NN classifier for $k = 1, \dots, 20$. Use the ℓ_1 distance for this problem. Plot the prediction accuracy as a function of k.

Finally, you will run logistic regression on the same data set. ***Set every*** $y_i = 0$ ***to*** $y_i = -1$ ***for this part. Also, be sure to add a dimension equal to*** $+1$ ***to each data point.***

(d) Implement the steepest ascent algorithm discussed in class. Use a step size $\eta = \frac{0.01}{4600}$. Run your algorithm for 1,000 iterations and plot the logistic regression objective training function $\mathcal{L}$ per iteration for each of the 10 training runs. Plot this in the same figure.

(e) Finally, implement an algorithm called "Newton's method" for logistic regression as follows: At iteration t, approximate the function

$$\mathcal{L}(w) \approx \mathcal{L}'(w) \equiv \mathcal{L}(w_t) + (w - w_t)^T \nabla \mathcal{L}(w_t) + \frac{1}{2}(w - w_t)^T \nabla^2 \mathcal{L}(w_t)(w - w_t)$$

Then set $w_{t+1} = \arg\max_w \mathcal{L}'(w)$. Derive the update for w_{t+1} for the logistic regression problem and implement and run this algorithm. Plot the objective function $\mathcal{L}$ on the training data as a function of $t = 1, \dots, 100$ for each of the 10 training runs. Plot this in the same figure.

(f) In a 2×2 table, show the testing results using Newton's method in the same way as shown in Problem 2(a).

[1] I've preprocessed the data. The original data is at `https://archive.ics.uci.edu/ml/datasets/Spambase`. More information about the meanings of the 54 dimensions of the data is provided in two accompanying files.