Assignment #2

Due: 23:55pm Weds March 6, 2019

Upload at: `https://dropbox.cs.princeton.edu/COS324_S2019/HW2`

Problem 1 (1pt)

Use the instructions above to put your name on the assignment when you compile this document.

Problem 2 (14pts)

Answer the following questions about a classification problem where:

1) there is a data set $\{\boldsymbol{x}_n, y_n\}_{n=1}^N$ where $\boldsymbol{x}_n \in \mathbb{R}^D$ and $y_n \in \{0, 1\}$,
2) we will use logistic regression to model the data,
3) we will use a squared L^2 norm on the weights as a regularizer.

A. Write the log likelihood, denoting the weight of the regularization penalty as λ and the regression weights as $\boldsymbol{w}$.
B. Prove that the penalized log likelihood is concave.

Problem 3 (15pts)

Imagine that we're doing basic linear regression with a probabilistic interpretation. We have data $\{\boldsymbol{x}_n, y_n\}_{n=1}^N$ where $\boldsymbol{x}_n \in \mathbb{R}^D$ and $y_n \in \mathbb{R}$. We're using a Gaussian likelihood where $y_n \mid \boldsymbol{x}_n, \boldsymbol{w}, v \sim \mathcal{N}(y_n \mid \boldsymbol{w}^T \boldsymbol{x}_n, v)$. We have some *a priori* knowledge about the data and so we want to make fairly strong assumptions about the weights. To reflect this knowledge we use a prior $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{w} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$. Derive the maximum *a posteriori* MAP estimate of $\boldsymbol{w}$ in terms of $\boldsymbol{X}, \boldsymbol{y}, v, \boldsymbol{\mu}$, and $\boldsymbol{\Sigma}$.

Problem 4 (20pts)

One way we can classify data is using a *generative* classifier. A generative classifier posits a distribution over the data in which the class is an unknown *latent variable*. The model then assumes that the data were generated from a class-conditional distribution. We learn such a classifier by fitting the parameters of each class by itself, along with the *a priori* weight between the two classes, and then using Bayes' theorem. Imagine that we have a binary classification problem with features in $\mathbb{R}^D$ and that we've already fit two Gaussians to the two classes:

$$\Pr(\boldsymbol{x} \mid \text{Class 1}) = \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_1, \boldsymbol{\Sigma}) \qquad \Pr(\boldsymbol{x} \mid \text{Class 2}) = \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_2, \boldsymbol{\Sigma}) \,. \tag{1}$$

So the class-conditional distributions have the same covariance, but different means. We have also already fit the marginal distributions over the two classes and found them to be equal:

$$\Pr(\text{Class 1}) = \Pr(\text{Class 2}) = \frac{1}{2} \,. \tag{2}$$

Classification can then be done for a new datum $\boldsymbol{x}'$ by asking about the conditional distribution $\Pr(\text{Class} \mid \boldsymbol{x}')$. Show that the decision boundary between these two classes is a hyperplane and give an equation for that plane.

Problem 5 (10pts)

A. (Scalar Case) Let X be a standard normal random variable, i.e., $X \sim \mathcal{N}(0, 1)$. Let Y be the linear transformation $Y = aX + b$ for some fixed scalars a and b. Prove that $Y \sim \mathcal{N}(b, a^2)$.

B- (Vector Case) Let $X \in \mathbb{R}^D$ be a standard normal random vector, i.e., $X \sim \mathcal{N}(0, \boldsymbol{I}_D)$, where $\boldsymbol{I}_D$ is the identity matrix in dimension D. Let Y be the linear transformation $Y = \boldsymbol{A}X + \boldsymbol{b}$ where $\boldsymbol{A} \in \mathbb{R}^{D \times D}$ is an invertible matrix and $\boldsymbol{b} \in \mathbb{R}^D$. Prove that $Y \sim \mathcal{N}(\boldsymbol{b}, \boldsymbol{A}\boldsymbol{A}^T)$.

Problem 6 (40pts)

In this problem, we'll apply logistic regression to predict whether a person will default on a home equity loan. There are two data files: `hmeq-train.csv` and `hmeq-test.csv` with 4000 and 1357 data, respectively. These are comma-delimited CSV files with the following columns:

- `BAD`: Value is 1 if the loan was bad, 0 if it was paid back. This is the quantity you will predict with your logistic regression model.
- `LOAN`: Amount of the loan.
- `MORTDUE`: Amount of existing mortgage.
- `VALUE`: Value of current property.
- `REASON`: {`DebtCon`, `HomeImp`, `Unknown`}
- `JOB`: {`Mgr`, `Office`, `Other`, `ProfExe`, `Self`, `Sales`, `Unknown`}
- `YOJ`: Years at present job. -1 if unknown.
- `DEROG`: Number of major derogatory reports. -1 if unknown.
- `DELINQ`: Number of delinquent credit lines. -1 if unknown.
- `CLAGE`: Age of oldest credit line in months. -1 if unknown.
- `NINQ`: Number of recent credit inquiries. -1 if unknown.
- `CLNO`: Number of credit lines. -1 if unknown.

You will build a logistic regression model to predict the `BAD` column, using the training data. You will need to load the CSV file (look into the `csv` python module) and turn the values into useful data that you can use as features. In particular, that will probably mean using a one-hot coding for the categorical variables. (Optionally, you may also want to replace the unknown -1 values with some other quantity, like the mean of that column; this is called *data imputation*.) In your first pass, don't do anything fancy with basis functions, just use the raw features and a bias term.

A. Write code to compute the training and test log likelihoods. It is likely that it will not return anything sensible. Explain why that might be.

B. Standardize the training data. That means take each of the continuous features, subtract its mean and divide by its standard deviation. This will make it zero-mean and have unit variance. Make sure to store the means and standard deviations of each column, as we'll need to transform the test data with the same values. It's bad hygiene to standardize the test data using its own mean and standard deviation.

C. Use full-batch gradient ascent to learn the weights of the logistic regression model. Try learning rates over a several orders of magnitude (all less than one) and report what works and how many iterations are necessary to reach convergence. Plot the log likelihoods (*training curves*) as a function of iteration for three different learning rates that you tried.

D. What final training log likelihood did you get with your best learning rate? What training accuracy?

E. Using the training set standardization values, transform the test set and evaluate the test log likelihood and the test accuracy. What values did you get?

F. Look at the weights that the model learned. Relate these to the underlying features and explain what you find. Any interesting associations between, e.g., job and bad debt?

G. Try out a more advanced idea to see if you can improve performance, e.g., add L^2 regularization or introduce some basis functions. Explain what you tried out, why you thought it might work, and whether or not you improved performance.

Changelog

- 21 February 2019 – Added a bullet to the last question on interpretation.
- 21 February 2019 – Initial version.