Stony Brook University
CSE512 – Machine Learning – Fall 19
Homework 2, Due: Sep 20 at midnight 11:59pm
Version 1 - Updated 6 Sep 2019

This homework contains 3 questions. The second question is optional. The last question requires programming. The maximum number of points is 100 plus 20 bonus points. Bonus points will not be used to fit the grade curve.

For Question 2 (and it is a good idea in general), we will use the following notations. Bold uppercase letters denote matrices (e.g. $\mathbf{D}$), bold lowercase letters denote column vectors (e.g. $\mathbf{d}$). $\mathbf{d}_i$ represents the i^{th} column of the matrix $\mathbf{D}$. d_{ij} denotes the scalar in the row j^{th} and column i^{th} of the matrix $\mathbf{D}$; it is also the j^{th} entry of column vector $\mathbf{d}_i$. Non-bold letters represent scalar variables. $\mathbf{1}_k \in \Re^{k\times 1}$ is a column vector of ones. $\mathbf{0}_k \in \Re^{k\times 1}$ is a column vector of zeros. $\mathbf{I}_k \in \Re^{k\times k}$ is the identity matrix. We use **comma** to denote horizontal concatenation of two vectors (or matrices) (e.g., $[\mathbf{a}, \mathbf{b}]$). We use **semicolon** to denote vertical concatenation of two vectors (e.g., $[\mathbf{a}; \mathbf{b}]$).

1 Question 1 – Parameter estimation (45 points)

This question uses a discrete probability distribution known as the Poisson distribution. A discrete random variable X follows a Poisson distribution with parameter λ if

$$p(X = k|\lambda) = \frac{\lambda^k}{k!}e^{-\lambda} \qquad k \in \{0, 1, 2, \dots\}$$

The Poisson distribution is a useful discrete distribution which can be used to model the number of occurrences of something per unit time. For example, it can be used to model the number of customers arriving at a bank per hour, or the number of calls handled by a telephone operator per minute.

1.1 Question 1.1 – MLE (15 points)

Assume the wait time for calling an Uber car is Poisson distributed (i.i.d) with parameter λ. You used Uber seven times and record the wait times in each trip.

Trip	1	2	3	4	5	6	7
Wait time	4	12	3	5	6	9	10

Let $\mathbf{X} = (X_1, \dots, X_n)$ be a random vector where X_i is the number of delay minutes for your i^{th} trip:

1. *(5 points)* Give the log-likelihood function of $\mathbf{X}$ given λ.
2. *(5 points)* Compute the MLE for λ in the general case.
3. *(5 point)* Compute the MLE for λ using the observed $\mathbf{X}$.

1.2 Question 1.2 – MAP (15 points)

Now let's be Bayesian and put a prior over the parameter λ. You talk to your party-goer friends, who tell you about the expected delays that they experience. You plot the distribution of the expected delays and your

extensive experience in statistics tells you that the plot resembles a Gamma distribution pdf. So you believe a good prior distribution for λ may be a Gamma distribution. The Gamma distribution has pdf:

$$p(\lambda|\alpha,\beta) = \frac{\beta^\alpha}{\Gamma(\alpha)}\lambda^{\alpha-1}e^{-\beta\lambda}, \lambda > 0. \tag{1}$$

$\Gamma(\alpha)$ is the Gamma function, which is the normalization factor that is introduced to have a proper probability density function (i.e., sum to 1). Do not worry if you don't know the explicit format of $\Gamma(\alpha)$; it is not important for this question. Also, if $\lambda \sim \Gamma(\alpha,\beta)$, then it has mean α/β and the mode is $(\alpha-1)/\beta$ for $\alpha > 1$. Assume the prior distribution of λ is $\Gamma(\lambda|\alpha,\beta)$.

1. *(8 points)* Compute the posterior distribution over λ. Hint:

   $$\lambda^{\sum_{i=1}^n X_i+\alpha-1}e^{-\lambda(n+\beta)}$$

   looks like a Gamma distribution! Is the rest of the expression constant with respect to λ? Working out a messy integral can lead to the answer but it is unnecessary.

2. *(7 points)* Derive an analytic expression for the maximum a posterior (MAP) estimate of λ.

1.3 Question 1.3 – Estimator Bias (15 points)

In class, we saw that the maximum likelihood estimator for the variance of a Gaussian distribution:

$$\hat{\sigma}^2_{MLE} = \frac{1}{N}\sum_{i=1}^{N}(x_i-\hat{\mu})^2$$

is biased, and that an unbiased estimator of variance is:

$$\hat{\sigma}^2_{unbiased} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i-\hat{\mu})^2$$

For the Gaussian distribution, these estimators give similar results for large enough N, and it is unclear whether one estimator is preferable to the other. In this problem, we will explore an example in which the maximum likelihood estimate is dramatically superior to any unbiased estimator.

Suppose we are not quite interested in estimating λ. We are more interested in estimating a quantity that is a *nonlinear* function of λ, namely $\eta = e^{-2\lambda}$. Suppose we want to estimate η from a single observation $X \sim Poisson(\lambda)$.

1. [5pts] Let $\hat{\eta} = e^{-2X}$. Show that $\hat{\eta}$ is the maximum likelihood estimate of η.
2. [5pts] Show that the bias of $\hat{\eta}$ is $e^{-(1-1/e^2)\lambda} - e^{-2\lambda}$. The following Taylor expansion may be useful:

   $$e^x = \sum_{k=0}^{\infty}\frac{x^k}{k!}$$

3. [5pts] It turns out that $(-1)^X$ is the *only* unbiased estimate of η. Prove that it is indeed unbiased and briefly explain why this is a bad estimator to use.

2 Question 2 – Ridge Regression and LOOCV (Optional - 10 bonus points)

In class, you learned about using cross validation as a way to estimate the true error of a learning algorithm. The preferred solution is *Leave-One-Out Cross Validation* (LOOCV), which provides an almost unbiased estimate of this true error, but it can take a really long time to compute. In this problem, you will derive a formula for efficiently computing the LOOCV error for ridge regression.

Given a set of n data points and associated labels $\{\mathbf{x}_i, y_i | \mathbf{x}_i \in \Re^k, y_i \in \Re\}_{i=1}^n$. Ridge regression find the weight vector $\mathbf{w}$ and a bias term b to optimize the following:

$$\underset{\mathbf{w},b}{\text{minimize}} \ \lambda||\mathbf{w}||^2 + \sum_{i=1}^{n}(\mathbf{w}^T\mathbf{x}_i + b - y_i)^2. \tag{2}$$

2.1

Let $\overline{\mathbf{w}} = [\mathbf{w}; b], \overline{\mathbf{X}} = [\mathbf{X}; \mathbf{1}_n^T], \bar{\mathbf{I}} = [\mathbf{I}_k, \mathbf{0}_k; \mathbf{0}_k^T, 0]$, $\mathbf{C} = \overline{\mathbf{X}}\overline{\mathbf{X}}^T + \lambda\bar{\mathbf{I}}$, and $\mathbf{d} = \overline{\mathbf{X}}\mathbf{y}$. Show that the solution of Ridge regression is:

$$\overline{\mathbf{w}} = \mathbf{C}^{-1}\mathbf{d} \tag{3}$$

2.2

Now suppose we remove $\mathbf{x}_i$ from the training data, let $\mathbf{C}_{(i)}, \mathbf{d}_{(i)}, \overline{\mathbf{w}}_{(i)}$ be the corresponding matrices for removing $\mathbf{x}_i$. Express $\mathbf{C}_{(i)}$ in terms of $\mathbf{C}$ and $\mathbf{x}_i$. Express $\mathbf{d}_{(i)}$ in terms of $\mathbf{d}$ and $\mathbf{x}_i$.

2.3

Express $\mathbf{C}_{(i)}^{-1}$ in terms of $\mathbf{C}^{-1}$ and $\mathbf{x}_i$. Hint: use the Sherman-Morrison formula:

$$(\mathbf{A} + \mathbf{u}\mathbf{v}^T)^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{u}\mathbf{v}^T\mathbf{A}^{-1}}{1 + \mathbf{v}^T\mathbf{A}^{-1}\mathbf{u}} \tag{4}$$

2.4

Show that

$$\overline{\mathbf{w}}_{(i)} = \overline{\mathbf{w}} + (\mathbf{C}^{-1}\overline{\mathbf{x}}_i)\frac{-y_i + \overline{\mathbf{x}}_i^T\overline{\mathbf{w}}}{1 - \overline{\mathbf{x}}_i^T\mathbf{C}^{-1}\overline{\mathbf{x}}_i} \tag{5}$$

2.5

Show that the leave-one-out error for removing the i^{th} training data is:

$$\overline{\mathbf{w}}_{(i)}^T\overline{\mathbf{x}}_i - y_i = \frac{\overline{\mathbf{w}}^T\mathbf{x}_i - y_i}{1 - \overline{\mathbf{x}}_i^T\mathbf{C}^{-1}\overline{\mathbf{x}}_i} \tag{6}$$

2.6

The LOOCV is defined as: $\sum_{i=1}^{n}(\overline{\mathbf{w}}_{(i)}^{T}\overline{\mathbf{x}}_i - y_i)^2$. What is the algorithmic complexity of computing LOOCV error using the formula given in Question 2.5? How is it compared with the usual way of computing LOOCV? Note that the complexity of inverting a $k \times k$ matrix is $O(k^3)$.

3 Question 3 – Programming [55 points + 10 bonus]

3.1 Implement LOOCV Ridge Regression

Implement a function to perform Ridge Regression with LOOCV as explained in Question 2. You can use either Matlab or any other programming language for this assignment. If you use Matlab, your function should have the following signature:

$$[\mathbf{w}, b, obj, cvErrs] = ridgeReg(\mathbf{X}, \mathbf{y}, \lambda)$$

Inputs:

- $\mathbf{X}$: a $k \times n$ data matrix, for n data points, each has k dimensions.
- $\mathbf{y}$: a $n \times 1$ label vector. This is not necessarily binary vector.
- λ: the weight for regularization term of Ridge regression

Outputs:

- $\mathbf{w}$: $k \times 1$ vector
- b: a scalar value for the bias term
- obj: the value of the objective function, i.e., $obj = \lambda||\mathbf{w}||^2 + \sum_{i=1}^{n}(\mathbf{w}^T\mathbf{x}_i + b - y_i)^2$
- $cvErrs$: $n \times 1$ vector for the cross validation errors. $cvErrs(i)$ is the leave-one-out error for removing the i^{th} training data.

Other programming language should have similar function signature. We recommend you use Matlab. Matlab is a very useful language, and you should find that Matlab achieves reasonable performance for this problem. Matlab has many toolboxes, and you cannot use any existing `ridge` function for the Statistics toolboxes (similarly for other languages).

Before you get started, here are some hints that you may find helpful:

- The only matrix operations required are: multiplying a matrix and a vector, adding vectors, computing dot product between two vectors, point-wise matrix operations (e.g., power of 2), calculate the sum of columns or rows or a matrix. Try to use as much vector/matrix computation as possible.
- You need to implement the efficient method for computing the LOOCV errors. But start by implementing the ridge regression solution without LOOCV. Implement the 'inefficient' way of computing the LOOCV errors by actually removing each training data in turn and recomputing the solution. Use this inefficient method to validate the results provided by the efficient method.

Finally here are some pointers toward useful parts of Matlab

- **B**= `repmat(`**A**`, [m, n])`: replicate the matrix **A**.

- Use `sum(A, dim)` to compute the sum of row or column of a matrix $\mathbf{A}$.

3.2 Predicting goodness points of a wine given its review [55 points + 10 bonus]

We will now put the Ridge Regression to work on some real data. Your task is to predict the review points a wine get based on its review text alone.

For this task, you are free to use any of the features given, you can even generate your own features from the data given. For instance, you could square the features, normalize some features, combine features, etc. to improve your results.

The data was scraped from WineEnthusiast during the week of June 15th, 2017 and can be found on the analytics website Kaggle (`https://www.kaggle.com/zynicide/wine-reviews`). This dataset provides a variety of features, the points, description (review), variety, country, province etc. For this homework we will be using the points and description features. Points are ratings within range 1- 100. However, the dataset contains description for wines rated in range 80-100. We already provide you the feature vectors for each review. You can download the data for this homework from
`https://www.kaggle.com/c/hw2-predict-wine-goodness-from-review/data`

The following files will be available for download on the project webpage:

`trainData.csv`	Training data matrix for predicting number of stars
`trainLabels.csv`	Training data matrix for predicting number of stars
`valData.csv`	Validation data
`valLabels.csv`	Validation labels
`testData.csv`	Test data
`featureTypes.txt`	List of features used

The data matrices are stored in csv format. Each line is a wine instance, containing various feature values. Description for each feature is provided in `featureTypes.txt`.

In Matlab, you can use the following code to load train data and labels:

```
D = csvread('trainData.csv',1,0);
trLb = csvread('trainLabels.csv',1,0);
```

Don't forget, the first column is IDs, which you might want to ignore for all the computation which follows.

For this part of the problem, you have the following tasks:

1. *(20 points)* Solve Ridge to predict the number of points a Wine will receive. Run Ridge on the training set, with $\lambda = 0.01, 0.1, 1, 10, 100, 1000$. At each solution, record the root-mean-squared-error (RMSE) on training, validation and leave-one-out-cross-validation data.

   Plot the train, validation and leave-one-out-cross-validation RMSE values together on a plot against λ. Label each curve in the plot.

2. *(10 points)* What λ achieve the best best LOOCV performance? For the model using this λ, report the objective value, the sum of square errors (on training data), the value of the regularization term.

3. *(10 points)* Using the model that you computed using λ that achieves best LOOCV performance, list the top 10 most important features and the top 10 least important features. Comment if the weights make sense intuitively.

4. *(15 points + 10 bonus points)* Use your model to predict the points for the reviews in test data. Save the predicted values on a CSV file `predTestLabels.csv` (see 4.2 for the format). The order of predicted values should correspond to the order of the reviews in `testData.csv` (counting starts from 0).

   Submit this `predTestLabels.csv` file and enter our Kaggle competition for fame. You must use your Stony Brook email to register on Kaggle. We will maintain a leader board, and the top three entries on the private leader board at the end of the competition (due date) will receive 10 bonus points. The ranking is based on RMSE. The points for this questions will be given in the proportion of your rank on public leader board on Kaggle. Report the RMSE.

   To prevent exploiting test data, you are allowed to make a maximum of 3 submissions per 24 hours. Your submission will be evaluated immediately and the leader board will be updated.

   You are allowed to create new features from the existing set of features. You are allowed to perform feature normalization on existing and new features. You can use both training and validation data to train your classifier. You can also use Lasso regression instead of Ridge. If you use Lasso regression, you must implement it yourself. You cannot use Multilayer Perceptron or any Deep Learning techniques for prediction.

4 What to submit?

4.1 Blackboard submission

You will need to submit both your code and your answers to questions on Blackboard. Put the answer file and your code in a folder named: SUBID_FirstName_LastName (e.g., *10947XXXX_barack_obama*). Zip this folder and submit the zip file on Blackboard. Your submission must be a zip file, i.e, SUBID_FirstName_LastName.zip. The answer file should be named: answers.pdf. The first page of the answers.pdf should be the filled cover page at the end of this homework. The remaining of the answer file should contain:

1. Answers (and derivations) to Question 1.
2. Optional: answers and derivations to Question 2.
3. Answers and plots to Question 3.2.1
4. Answers to Questions 3.2.2, 3.2.3 and 3.2.4

You can use Latex if you wish, but it is not compulsory.

4.2 Kaggle submission

For Question 3.2.4, you must submit a CSV file to get the RMSE from the competition site `https://www.kaggle.com/t/342c03e816ee45d09cde62892c7091b2`. A submission file should contain two columns: Id and Prediction. The file should contain a header and have the following format.

$$\begin{array}{ll} Id, & Prediction \\ 0, & 82000 \\ 1, & 106000 \\ \ldots & \ldots \end{array} \tag{7}$$

A sample submission file is available from the competition site and our handout.

5 Cheating warnings

Don't cheat. You must do the homework yourself, otherwise you won't learn. You cannot ask and discuss with students from previous years. You cannot look up the solution online.

Cover page for answers.pdf
CSE512 Fall 2019 - Machine Learning - Homework 2

Your Name:

Solar ID:

NetID email address:

Names of people whom you discussed the homework with: