CS189: Introduction to Machine Learning

Homework 4

Due: Sunday, October 18th, 2015, 11:59 pm

Submission Instructions

In your submission, include two separate files, submitted to **bCourses**:

1. A pdf writeup with answers to all the questions and your plots. Include in the pdf a copy of your code for each problem (code for problems 1, 3, 4).
2. A zip archive containing your code for each problem, and a README with instructions on how to run your code.

Submit **2 separate files to bCourses: a pdf and a zip of your code**.

You are also required to submit a text file with your best predictions for the examples in the test set (for both MNIST and spam) to Kaggle, just like Homework 1. There will be separate Kaggle competitions for each dataset. The Kaggle invite links and more instructional details will be posted on the course website.
Good luck!

Problem 1: Centering and Ridge Regression

In this problem we will return to predicting the median home value in a given Census area by extending linear regression. The data is in housing_data.mat and it comes from `http://lib.stat.cmu.edu/datasets/houses.zip`. There are only 8 features for each data point; you can read about the features in **housing_data_source.txt**.

a) You are given a training set $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$. Let $\mathbf{X}$ be the design matrix (i.e. the matrix whose i^{th} row is $\mathbf{x}_i$), and let $\mathbf{y}$ be the column vector whose i^{th} entry is y_i. Let $\mathbf{1}$ be a $n \times 1$ column vector of ones.

Define $\bar{\mathbf{x}} = \frac{1}{n}\sum_i \mathbf{x}_i$ and $\bar{y} = \frac{1}{n}\sum_i y_i$. Assume that the input data has been centered, so that $\bar{\mathbf{x}} = 0$. Show that the optimizer of the following loss function $J(\mathbf{w}, w_0)$

$$J(\mathbf{w}, w_0) = (\mathbf{y} - \mathbf{X}\mathbf{w} - w_0\mathbf{1})^\top(\mathbf{y} - \mathbf{X}\mathbf{w} - w_0\mathbf{1}) + \lambda\mathbf{w}^\top\mathbf{w}$$

is given by

$$\hat{w_0} = \bar{y}\ \mathbf{,}\ \hat{\mathbf{w}} = (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y}.$$

b) Using the result from part a,

i) Implement a ridge regression model with least squares. Include your code in the submission.

ii) Use 10-fold cross validation to tune the hyperparameter λ. Using the tuned value of λ, test your trained model on the validation set. What is the residual sum of squares (RSS) on the validation set? How does this compare to the result from HW3?

iii) Plot the regression coefficients, w. How do these compare to the result from HW3?

Problem 2: Derivation of the Bonferroni Correction Imagine you are playing a dice game with your friends. You roll 2 dice, and to win you must roll two sixes (anything else is a loss). You are a cheater and while your friends are not looking, you roll the dice 3 times to increase your chances.

a) What is the probability of a double 6 in 1 roll?

b) What is the probability of getting a double 6 **at least once** if you roll your dice 3 times?

c) Now, imagine you train a classifier that gets an error rate better than your competitor's error rate. You then test the significance of the results with the bootstrap method. In 1000 samples of your test set errors, 203 samples show an error rate worse than that of your competitor. What is the p-value of the test whose null hypothesis is that your performance is not different than that of your competitor (with alternative hypothesis that you are better)?

d) You are not satisfied with the result and you try 5 other classification methods. One of them gets a bootstrap p-value of 0.05. Can you assert that your result is "significantly better" than that of your competitor with 5% chance of being wrong (in the sense that we can reject the null hypothesis with risk 0.05)?

e) Derive the Bonferroni correction, m * p-value, where m is the number of models tried, for small p-values. Hint: the p-value is like the probability of your double six dice.

f) We have a classification problem of normal vs. cancer patients using gene expression data. The feature space has d=50,000 features (genes). We use each gene as a very simple classifier: the feature value is used to predict Y (cancer = +1, normal = -1) with the AUC. The p-value in this case can be computed in a closed form (this is called the Wilcoxon-Mann-Whitney test). You find a gene with pvalue 0.0001. Is this a significant gene (in the sense that it is predictive of the Y outcome)? How come the Bonferroni correction gives a value greater than 1?

Problem 3: Independence vs. Correlation

(a) Consider the random variables X and Y $\in \mathbb{R}$ with the following conditions.

(i) X and Y can take values [-1,0,1].

(ii) When X is 0, Y takes values 1 and -1 with equal probability ($\frac{1}{2}$). When Y is 0, X takes values 1 and -1 with equal probability ($\frac{1}{2}$).

(iii) Either X is 0 with probability ($\frac{1}{2}$), or Y is 0 with probability ($\frac{1}{2}$).

Are X and Y uncorrelated? Are X and Y independent? Prove your assertions. *Hint:* Graph these points onto the Cartesian Plane. What's each point's joint probability?

(b) Consider three Bernoulli random variables B_1, B_2, B_3 which take values $\{0, 1\}$ with equal probability. Lets construct the following random variables X, Y, Z: $X = B_1 \oplus B_2$, $Y = B_2 \oplus B_3$, $Z = B_1 \oplus B_3$, where $\oplus$ indicates the XOR operator. Are X, Y, and Z pairwise independent? Mutually independent? Prove it.

(c) Why are the questions above relevant to what we are learning in class? (Hint: under what condition is a problem learnable for various models we have learned about?) Describe a dataset upon which we cannot apply the methods learned in class.

Problem 4: Isocontours of Normal Distributions

Let $f(\mu, \Sigma)$ denote the density function of a Gaussian random variable. Plot isocontours of the following functions:

a) $f(\mu, \Sigma)$, where $\mu = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and $\Sigma = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$

b) $f(\mu, \Sigma)$, where $\mu = \begin{bmatrix} -1 \\ 2 \end{bmatrix}$ and $\Sigma = \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix}$

c) $f(\mu_1, \Sigma_1) - f(\mu_2, \Sigma_2)$, where $\mu_1 = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$, $\mu_2 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$ and $\Sigma_1 = \Sigma_2 = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$

d) $f(\mu_1, \Sigma_1) - f(\mu_2, \Sigma_2)$, where $\mu_1 = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$, $\mu_2 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$, $\Sigma_1 = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$ and $\Sigma_2 = \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix}$

e) $f(\mu_1, \Sigma_1) - f(\mu_2, \Sigma_2)$, where $\mu_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, $\mu_2 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$, $\Sigma_1 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$ and $\Sigma_2 = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$

Problem 5: Visualizing Eigenvectors of Gaussian Covariance Matrix

We have two one dimensional random variables $X_1 \sim \mathcal{N}(3, 9)$ and $X_2 \sim \frac{1}{2}X_1 + \mathcal{N}(4, 4)$, where $\mathcal{N}(\mu, \sigma^2)$ is a Gaussian distribution with mean μ and variance σ^2. In software, draw $N = 100$ random samples of X_1 and of X_2.

(a) Compute the mean of the sampled data.

(b) Compute the covariance matrix of the sampled data.

(c) Compute the eigenvectors and eigenvalues of this covariance matrix.

(d) On a two dimensional grid with a horizontal axis for X_1 ranging from $[-15, 15]$ and a vertical axis for X_2 ranging from $[-15, 15]$, plot the following:

  i) All $N = 100$ data points

  ii) Arrows representing both covariance eigenvectors. The eigenvector arrows should originate from the mean and have magnitude equal to their corresponding eigenvalues.

(e) By placing the eigenvectors of the covariance matrix into the columns of a matrix $U = [v_1 \; v_2]$, where v_1 is the eigenvector corresponding to the largest eigenvalue, we can use U' as a rotation matrix to rotate each of our sampled points from our original (X_1, X_2) coordinate system to a coordinate system aligned with the eigenvectors (without the transpose, U can rotate back to the original axes). Center your data points by subtracting the mean and then rotate each point by U', specifically $x_{rotated} = U'(x - \mu)$. Plot these rotated points on a new two dimensional grid with both axes ranging from [-15,15].

Problem 6: Covariance Matrixes and Decompositions

As described in lecture, a covariance matrix $\Sigma \in \mathbb{R}^{N,N}$ for a random variable $X \in \mathbb{R}^N$ with the following values, where $cov(X_i, X_j) = E[(X_i - \mu_i)(X_j - \mu_j)]$ is the covariance between the ith and jth elements of the random vector X:

$$\Sigma = \begin{bmatrix} cov(X_1, X_1) & ... & cov(X_1, X_n) \\ \cdots & & \cdots \\ cov(X_n, X_1) & ... & cov(X_n, X_n) \end{bmatrix} \tag{1}$$

For now, we are going to leave the formal definition of covariance matrices aside and focus instead on some transformations and properties. The motivating example we will use is the N dimensional Multivariate Gaussian Distribution defined as follows:

$$f(x) = \frac{1}{\sqrt{(2\pi)^N |\Sigma|}} e^{-\frac{1}{2}((x-\mu)^\top \Sigma^{-1}(x-\mu)} \tag{2}$$

(a) We usually assume that Σ^{-1} exists, but in many cases it will not. Describe the conditions for which Σ_X^{-1} corresponding to random variable X will not exist. Which linear transfer allows us to convert variable X into a new random variable X' (without loss of information), which has an invertible covariance matrix?

(b) Consider a data point x drawn from a zero mean Multivariate Gaussian Random Variable $X \in \mathbb{R}^N$ like shown above. Prove that there exists matrix $A \in R^{N,N}$ such that $x^\top \Sigma^{-1} x = \|Ax\|_2^2$ for all vectors x. What is the matrix A?

(c) In the context of Multivariate Gaussians from the previous problem, what is the intuitive meaning of $x^\top \Sigma^{-1} x$ when we transform it into $\|Ax\|_2^2$?

(d) Lets constrain $\|x\|_2 = 1$. In other words, the L2 norm (or magnitude) of vector x is 1. In this case, what is the maximum and minimum value of $\|Ax\|_2^2$? If we have $X_i \perp\!\!\!\perp X_j \ \forall i, j$, then what is the intuitive meaning for the maximum and minimum value of $\|Ax\|_2^2$? To maximize the probability of $f(x)$, which x should we choose?

Problem 7: Gaussian Classifiers for Digits

In this problem we will build Gaussian classifiers for digits in MNIST. More specifically, we will model each digit class as a Gaussian distribution and make our decisions on the basis of posterior probabilities. This is a generative method for classifying images where we are modelling the class conditional probabilities as normal distributions. The steps mentioned below should be done for each training set in `train.mat` and you need to plot a curve of error rate vs no. of training examples upon evaluating on the test set in `test.mat`. Submit your predicted class labels for the `test.mat` dataset on the Kaggle competition website. Please use do not use the datasets that we provided in the HW1.zip folder, and only use the datasets provided in the current HW4.zip folder. We have randomized the MNIST test and training sets.

a) Taking raw pixel values as features, fit a Gaussian distribution to each digit class using maximum likelihood estimation. This involves finding the means and covariance matrices for each digit class. Say we have i.i.d observations $X_1 ... X_n$, what are the maximum likelihood estimates for the mean and covariance matrix of a Gaussian distribution?
Tip: It is a good idea to contrast normalize images before using the raw pixel values. One way of normalization is to divide the pixel values of an image by the l_2 norm of its pixel values.

b) How would you model the prior distribution for each class? Compute prior probabilities for all classes.

c) Visualize the covariance matrix for a particular class. Do you see any kind of structure in the matrix? What does this symbolize?

d) We will now classify digits in the test set on the basis of posterior probabilities using two different approaches:

   i) Define $\Sigma_{overall}$ to be the average of the covariance matrices of all the classes. We will use this matrix as an estimate of the covariance of all the classes. This amounts to modelling class conditionals as Gaussians ($\sim \mathcal{N}(\mu_i, \Sigma_{overall})$) with different means and the same covariance matrix. Using this form of class conditional probabilities, classify the images in the test set into one of the 10 classes assuming 0-1 loss and compute the error rate and plot it over the following number of randomly chosen training data points [100, 200, 500, 1000, 2000, 5000, 10000, 30000, 60000]. Expect some variance in your error rate for low training data scenarios. What is the form of the decision boundary in this case? Why?

   ii) We can also model class conditionals as $\mathcal{N}(\mu_i, \Sigma_i)$, where Σ_i is the estimated covariance matrix for the i^{th} class. Classify images in the test set using this form of the conditional probability (assuming 0-1 loss) and compute the error rate and plot it over the following number of randomly chosen training data points [100, 200, 500, 1000, 2000, 5000, 10000, 30000, 60000]. What is the form of the decision boundary in this case?

iii) Compare your results in parts *i* and *ii*. What do you think is the source of difference in the performance?

iv) Train your best classifier using `train.mat` and classify the images in `test.mat`. Submit your labels to the online Kaggle competition and record your optimum prediction rate. If you used an additional featurizer, please describe your implementation. Please only use any extra "image featurizer" code on this portion of the assignment.

Note: In your submission, you need to include learning curves (error-rate vs no. of training examples) and actual error-rate values for the above two cases and short explanations for the all the questions. Also, the covariance matrices you compute using MLE might be singular (and thus non-invertible). In order to make them non-singular and positive definite, you can add a small weight to their diagonals by setting $\Sigma_i = \Sigma_i + \alpha I$, where α is the weight you want to add to the diagonals. You may want to use k-fold cross validation to see what the optimum "small weight" is.

e) Now that you have developed Gaussian classification for digits, lets apply this to spam. Use the training and testing data located in `spam.mat` to generate a set of test labels that you will submit to the online Kaggle competition and record your optimum prediction rate. If you used an additional featurizer, please describe your implementation.

Optional: If you use the default feature set, you may obtain relatively low classification rates. The TA's suggest using a bag of words model. You may download 3rd party packages if you wish. Also, normalizing your vectors like before may help

f) *Extra for Experts:* Using the `training_data` and `training_labels` in `spam.mat,` identify 10 words in your features set corresponding to the maximum and minimum variances. Use k-fold cross validation to train your classifier only using 10 variance maximum words and record your average classification rate. Do the same with the 10 minimum variance words. What do you notice? Can you tie this in with what you proved in part 6.d)? Will the assumption of independence between words hold here? For more information: **PCA, Courtesy of Professor Laurent El Ghaoui**