CS189: Introduction to Machine Learning

Homework 3

Due: Thursday March 3rd, 2016, 11:59 pm
Homework Parties:
Tuesday, February 23rd, 2016 (3:30-5:00 PM , Wozniak Lounge)
Tuesday, March 1st, 2016 (6:30-8:30 PM , 521 Cory)

Submission Instructions

You will submit this assignment to both **bCourses** and **Gradescope**. There will also be a **Kaggle** competition.

In your submission to **Gradescope**, include:

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 2, 3, 5, and 6).

In your submission to **bCourses**, include:

2. A zip archive containing your code for each problem, and a README with instructions on how to run your code. Please include the pdf writeup in this zip archive as well.

Submitting to **Kaggle**:

3. Submit a csv file with your best predictions for the examples in the test set (for **both** MNIST and Spam) to Kaggle, just like Homework 1.

Note: There will be separate Kaggle competitions for each dataset. The Kaggle invite links and more instructional details will be posted on Piazza.
Good luck!

Problem 1: 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. Let's 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.

Problem 2: 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 3: 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^T 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^T, specifically $x_{rotated} = U^T(x - \mu)$. Plot these rotated points on a new two dimensional grid with both axes ranging from $[-15, 15]$.

Problem 4: 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:

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

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 transformation 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) Let's constrain $\|x\|_2 = 1$. In other words, the L_2 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 5: 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**
(`https://inst.eecs.berkeley.edu/~ee127a/book/login/l_senate_pca.html`)

Problem 6: Linear Regression

In this problem we will try to predict the median home value in a given Census area by using 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.

1. Implement a linear regression model with least squares. Include your code in the submission.

   You should add a constant term to the training data (e.g. add another dimension to each data point, with the value of 1). This is same as adding the bias term to linear regression (see discussion 4 question 1). Note that each data point $\mathbf{x}^{(i)^T}$ is a row of the training data matrix X.

2. Test your trained model on the validation set. What is the residual sum of squares (RSS) on the validation set? What is the range of predicted values (min, max)? Do they make sense?

3. Plot the regression coefficients $\boldsymbol{w}$ (plot the value of each coefficient against the index of the coefficient). Be sure to exclude the coefficient corresponding to the constant offset you added earlier.

4. Plot a histogram of the residuals of the training data. What distribution does this resemble? **Note:** the residual corresponding to point i is $f(\mathbf{x}^{(i)}) - y^{(i)}$.

NOTE: You may not use any library routine for linear regression or least squares solving. You may use any other linear algebra routines.