COEN 4890/EECE 5890: Developments in Computing: Machine Learning for Image Processing, Fall 2018: Homework 2

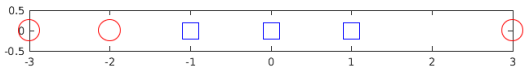
Instructor: Dong Hye Ye

Due: Tuesday, December 4^{th}, beginning of class

Instructions There are 3 written questions, plus 1 programming question. Submit your written answers as papers and your codes to D2L.

1 Feature Maps, Kernels, and SVM [30 points]

You are given a data set D in below figure with data from a single feature X_1 and corresponding label $Y \in \{+, -\}$. The data set contains three positive examples at $X_1 = \{-3, -2, 3\}$ and three negative examples at $X_1 = \{-1, 0, 1\}$. (Red circle and blue square represent + and −, respectively.)



1.1 Finite Features and SVMs

1. *(2 points)* Can this data set (in its current feature space) be perfectly separated using a linear separator? Why or why not?

2. *(2 points)* Lets define the simple feature map $\phi(u) = (u, u^2)$ which transform the data into two-dimensional space. Apply ϕ to the data and plot the points in the new two-dimensional feature space.

3. *(2 points)* Can a linear separator perfectly separate the points in the new two-dimensional feature space induced by ϕ? Why or why not?

4. *(6 points)* Construct a maximum-margin separating hyper plane. This hyper-plane will be a line, which can be parameterized by its normal equation, i.e. $w_1 Y_1 + w_2 Y_2 + c = 0$ for appropriate choices of w_1,w_2, c. Here, $(Y_1, Y_2) = \phi(X_1)$ is the result of applying the feature map ϕ to the original feature X_1. Give the values for w_1,w_2, c. Also, explicitly compute the margin for your hyper-plane. You do not need to solve a quadratic program to find the maximum margin hyper-plane. Instead, let your geometric intuition guide you.

5. *(2 points)* On the plot of the transformed points, plot the separating hyperplane and the margin, and circle the support vectors.

6. *(2 points)* Draw the decision boundary separating of the separating hyper-plane, in the original one-dimensional feature space.

1.2 Infinite Features Spaces and Kernel Magic

Lets define a new (infinitely) more complicated feature transformation ϕ_∞.

$$\phi_\infty = \{e^{-x^2/2}, e^{-x^2/2}x, \frac{e^{-x^2/2}x^2}{\sqrt{2}}, \ldots, \frac{e^{-x^2/2}x^i}{\sqrt{i!}}, \ldots\} \tag{1}$$

You can think of this feature transformation as taking some finite feature vector and producing an infinite dimensional feature vector rather than the simple two dimensional feature vector used in the earlier part of this problem.

1. *(2 points)* Can we directly apply this feature transformation to the data. Put another way, can we explicitly construct $\phi_\infty(x)$? (This is nearly rhetorical and not a trick question.)

2. *(6 points)* We know that we can express a linear classifier using only inner products of support vectors in the transformed feature space. It would be great if we could some how use the feature space obtained by the feature transformormation ϕ_∞. However, to do this we must be able to compute the inner product of examples in this infinite vector space. Lets define the inner product between two infinite vectors $a = \{a_1, \ldots, a_i, \ldots\}$ and $b = \{b_1, \ldots, b_i, \ldots,\}$ as the infinite sum given in the following equation.

$$k(a, b) = a \cdot b = \sum_{i=1}^{\infty} a_i b_i \tag{2}$$

   Can we explicitly compute $k(a, b)$? What is the explicit form of $k(a, b)$? (Hint: you may want to use the Taylor series expansion of e^x which is given in the following equation.)

$$e^x = \lim_{n\to\infty} \sum_{i=0}^{n} \frac{x^i}{i!} \tag{3}$$

3. *(2 points)* With such a high dimensional feature space should we be concerned about overfitting?

4. *(4 points)* Suppose we translate my inputs $x' = x + x_0$ for some arbitrary x_0 before using the infinite kernel above in an SVM. Will my predictions change? Explain why or why not?

2 Kernel Regression and Locally Weighted Regression [15 points]

Given a set of n examples, (x_j, y_j) (x_j is the input, and y_j is output), $j = 1, \ldots, n$, a linear smoother is defined as follows. For any x, there exists a vector $l(x) = (l_1(x), \ldots, l_n(x))^T$ such that the estimated output $\hat{y}$ of x is $\hat{y} = \sum_{j=1}^{n} l_j(x) y_j =< l(x), Y >$, where Y is a $n \times 1$ vector, $Y_j = y_j$, and $< a, b >$ is the inner product between two vectors a and b.

1. *(2 points)* Recall that in linear regression, we assume the data are generated from the linear regression model $y_j = \sum_{i=1}^{k} w_i h_i(x_j) + \epsilon_j$. The least squares estimate for the coefficient vector w is given by $w^* = (H^T H)^{-1} H^T Y$, where H is a $n \times k$ matrix, $H_{ji} = h_i(x_j)$. Given an input x, what is the estimated output $\hat{y}$? (Matrix form solution is required. You may want to use the $k \times 1$ vector $h(x) = [h_1(x), \ldots, h_k(x)]^T$)

2. *(3 points)* Based on part 1, prove that linear regression is a linear smoother. Give the vector $l(x)$ for a given input x.

3. *(2 points)* In kernel regression, given an input x, what is the estimated output $\hat{y}$?

4. *(3 points)* Based on part 3, prove that kernel regression is a linear smoother. Give the vector $l(x)$ for a given input x.

5. *(2 points)* In local weighted regression, given an input x, what is the estimated output $\hat{y}$?

6. *(3 points)* Based on part 5, prove that locally weighted regression is a linear smoother. Give the vector $l(x)$ for a given input x.

3 Naïve Bayes Classifier [15 points]

You are running a Naïve Bayes classifier for a classification problem with one (unobserved) binary class variable Y (e.g. whether it's too hot for your dog in here) and 3 binary feature variables X_1,X_2,X_3. The class value is never directly seen but approximately observed using a sensor (e.g. you see your dog panting). Let Z be the binary variable representing the sensor values. One morning (your dog is out to play) you realize the sensor value is missing in some of the examples. From the sensor specifications (that come with your dog), you learn that the probability of missing values is four times higher when $Y = 1$ than when $Y = 0$. More specifically, the exact values from the sensor specifications are:

$$P(Z \text{ missing}|Y = 1) = 0.08,\ P(Z = 1|Y = 1) = 0.92$$
$$P(Z \text{ missing}|Y = 0) = 0.02,\ P(Z = 0|Y = 0) = 0.98$$

1. *(5 points)* Draw a Bayes net that represents this problem with a node Y that is the unobserved label, a node Z that is either a copy of Y or has the value "missing", and the three features X_1,X_2,X_3.

2. *(5 points)* What is the probability of the unobserved class label being 1 given no other information, i.e., $P(Y = 1|Z =$ missing)? Derive the quantity using the Bayes rule and write your final answer in terms of $\theta_{Y=1}$, our estimate of $P(Y = 1)$.

3. *(5 points)* We would like to learn the best choice of parameters for $P(Y), P(X_1|Y), P(X_2|Y)$, and $P(X_3|Y)$. Write the log- joint probability of X, Y and Z given your Bayes net, first for a single example $(X_1 = x_1, X_2 = x_2, X_3 = x_3, Z = z, Y = y)$, then for n i.i.d. examples $(X_1^i = x_1^i, X_2^i = x_2^i, X_3^i = x_3^i, Z^i = z^i, Y^i = y^i)$ for $i = 1, \ldots, n$.

4 Principal Components Analysis [40 points]

In this problem we will be working with the digits data set. The data file we provide contains 60,000 hand written digits between 0 and 9. Each digit is a 28×28 grayscale image represented as a 784 dimensional vector. The variable X is a 60,000×784 matrix. The 60,000 dimensional vector Y contains the true number for each image.

A very common technique for dimensionality reduction is principal components analysis typically referred to as PCA. Here you will need to implement PCA. Because this is a relatively large data set, consider using the functions `cov` and `eig` or `eigs`. Please submit in your write-up a copy of all plots for this problem.

1. *(15 points)* Plot the first 9 principal components as images. You will probably need the functions `image`, `colormap('Gray')`, `subplot`, and `reshape(v,28,28)`. To plot the principal components rescale the vector so that its values range between 0 and 255.

2. *(10 points)* Plot the eigenvalues in decreasing order. From the plot, how many eigenvectors do you believe are necessary to approximately represent the images.

3. *(15 points)* Using the first 1, 2, 5, 10, 21, 44, 94, 200, and 784 principal components plot the reconstruction of the the first 2 digits in the data set. Use `subplot(3,3,i)` to save tree of the natural kind. Does the approximation get better with increasing principal components?