10701/15781 Machine Learning, Fall 2007: Homework 5

Due: Wednesday, December 5, to Monica by 3pm

Instructions

There are 4 questions on this assignment. Problem 3 involves coding. Do not attach your code to the writeup. Instead, copy your implementation to

`/afs/andrew.cmu.edu/course/10/701/Submit/your_andrew_id/HW5`

To write in this directory, you need a kerberos instance for andrew, or you can log into, for example, unix.andrew.cmu.edu.

Please submit each problem seperately with your name and Andrew ID on each problem. Refer to the webpage for policies regarding collaboration, due dates, and extensions.

IMPORTANT: We will not accept homework turned in after 3pm on Saturday December 8th (even for partial credit). We want to be able to post the solutions for this homework as early as possible.

1 [20 Points] Online Learning (Steve)

1. We saw in class that for any finite hypothesis class $\mathbb{C}$ of binary classifiers, the optimal mistake bound is at most $\log_2 |\mathbb{C}|$. That is, there is an algorithm that, regardless of what sequence $x_1, x_2, \ldots$ an adversary chooses, as long as there is some $f \in \mathbb{C}$ such that the true label of every x_i is $f(x_i)$, if the algorithm is told whether it makes a mistake immediately after each prediction, the algorithm will make at most $\log_2 |\mathbb{C}|$ mistakes on the entire sequence. In particular, the Halving algorithm is an example of such an algorithm.

   Prove that for every hypothesis class $\mathbb{C}$, the optimal mistake bound for $\mathbb{C}$ is at least $VC(\mathbb{C})$, the VC-dimension of $\mathbb{C}$. In other words, you need to prove that for any (deterministic) algorithm, there exists a sequence $x_1, x_2, \ldots$ and a target function $f \in \mathbb{C}$ such that if the true label of every x_i is $f(x_i)$, then the algorithm will make at least $VC(\mathbb{C})$ mistakes on the sequence.

   *H*int: Because it's a deterministic algorithm, it suffices to specify a sequence $x_1, x_2, \ldots$, and describe how an adversary would label any example in the sequence in response to the algorithm's prediction. Then briefly explain why this forces the algorithm to make $VC(\mathbb{C})$ mistakes, while still keeping all of the true labels consistent with some $f \in \mathbb{C}$.

2. Instead of binary classification, suppose we are interested in classifying into k categories. That is, a hypothesis class $\mathbb{C}$ contains functions h such that for any example x, $h(x) \in \{1, 2, \ldots, k\}$. In this modification of the mistake bound model, in each round the learning algorithm observes an example x, then makes a prediction $y \in \{1, 2, \ldots, k\}$, and then is told the true label $f(x) \in \{1, 2, \ldots, k\}$, (where $f \in \mathbb{C}$ is a fixed target function).

   Prove that for any hypothesis class $\mathbb{C}$ of this type, the optimal mistake bound is still at most $\log_2 |\mathbb{C}|$.

2 [25 points] Expectation Maximization [Sue Ann]

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 and) 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:

$$P(Z \text{ missing}|Y = 1) = .08, \quad P(Z = 1|Y = 1) = .92$$
$$P(Z \text{ missing}|Y = 0) = .02, \quad P(Z = 0|Y = 0) = .98$$

1. 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. 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. We would like to learn the best choice of parameters for $P(Y), P(X_1|Y), P(X_2|Y)$, and $P(X_3|Y)$. Assume $Y, X_1|Y, X_2|Y, X_3|Y$ are all Bernoulli variables and let us denote the parameters as[1]

$$\theta_{Y=y} = P(Y = y), \quad \theta_{X_1=x_1|Y=y} = P(X_1 = x_1|Y = y),$$
$$\theta_{X_2=x_2|Y=y} = P(X_2 = x_2|Y = y), \quad \theta_{X_3=x_3|Y=y} = P(X_3 = x_3|Y = y).$$

   Write the log-probability of X, Y and Z given θ, in terms of θ, and $P(Z|Y)$, 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 $i = 1, ..., n$.

4. Provide the E-step and M-step for performing expectation maximization of θ for this problem.

   In the E-step, compute the distribution $Q_{t+1}(Y|Z, X)$ using

$$Q_{t+1}(Y = 1|Z, X) = E[Y|Z, X_1, X_2, X_3, \theta_t]$$

   using your Bayes net from part 1 and conditional probability from part 2 for the unobserved class label Y of a single example.

   In the M-step, compute

$$\theta_{t+1} = \text{argmax}_\theta \sum_{i=1}^{n} \sum_y Q(Y^i = y|Z^i, X^i) \log P(X_1^i, X_2^i, X_3^i, Y^i, Z^i|\theta)$$

   using all of the examples $(X_1^1, X_2^1, X_3^1, Y^1, Z^1), ..., (X_1^n, X_2^n, X_3^n, Y^n, Z^n)$. Note: it is OK to leave your answers in terms of $Q(Y|Z, X)$.

3 [30 points] Clustering and Dimensionality Reduction

In this problem we will again 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 $60,000 \times 784$ matrix. The $60,000$ dimensional vector Y contains the true number for each image. Please submit include in your write-up a copy of all plots for this problem.

[1] We only need $\theta_Y = P(Y = 1), \theta_{X_i|Y=y} = P(X_i = 1|Y = y), ...$ since $\theta_{Y=0} = 1 - \theta_{Y=1}, ...$, but the set of θs defined here should help you notationally.

Update: There is now a smaller data set availabe. In your results please say what data set you use and use only one data set throughout the problem. The smaller data set consists of 5000 10×10pixel images each represented as 100 dimensional row vector. When asked to use the first 1, 2, 5, 10, 21, 44, 94, 200, and 784 principal components instead use the first 1, 2, 3, 6, 10, 18, 32, 56, and 100.

3.1 [15 Points] Principal Components Analysis

A very common technique for dimensionality reduction is principal components analysis typically referred to as PCA. In the first part of this problem you will need to implement PCA. Because this is a relatively large data set, consider using the functions `cov` and `eig` or `eigs`.

1. [5 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. [5 Points] Plot the eigenvalues in decreasing order. From the plot, how many eigenvectors do you believe are necessary to approximately represent the images.

3. [5 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?

3.2 [15 Points] Gaussian Mixture Model

For this problem you will need to implement Expectation Maximization (EM) for the axis aligned Gaussian mixture model. Recall that the axis aligned gaussian mixture model uses the Gaussian Niave bayes assumption that given the class all the features are conditionally independent Gaussians. The specific form of the model is given in Equation 1.

Update: Due to numerical issues on this data set, you may use K-means clustering instead of a Gaussian mixture model. If you have a clever solution to the numerical problems please describe the solution in your write-up for extra credit.

$$
\begin{aligned}
Z_i &\sim Multinomial(p_1, \ldots, p_K) \\
X_i | Z_i = z &\sim N\left(\begin{bmatrix} \mu_1^{(z)} \\ \mu_2^{(z)} \\ \vdots \\ \mu_d^{(z)} \end{bmatrix}, \begin{bmatrix} (\sigma_1^{(z)})^2 & 0 & \cdots & 0 \\ 0 & (\sigma_2^{(z)})^2 & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & (\sigma_d^{(z)})^2 \end{bmatrix} \right)
\end{aligned} \tag{1}
$$

If you find that running your algorithm on the full data set takes too long try using the first $5,000$ digits instead of all $60,000$. You will want to avoid looping over all $n = 60,000$ examples.

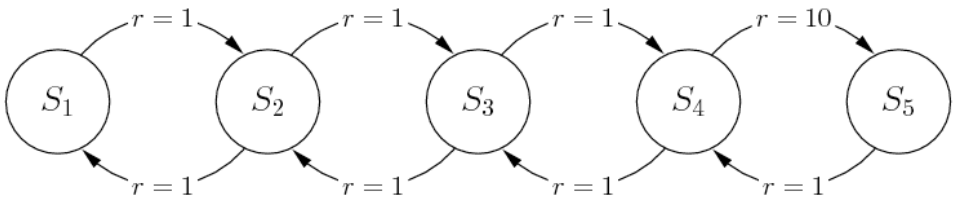
1. [5 Points] Run EM to fit a Gaussian mixture model with 16 gaussians on the digits data without applying PCA. Plot each of the means using `subplot(4,4,i)` to save paper.

2. [5 Points] Again run EM clustering with $K = 16$ using only the first 100 principal components. Again, plot each of the means. Do you notice a difference in the output or running time?

3. [5 Points] Evaluating clustering performance is difficult. However, because we have "truth" data, we can roughly assess clustering performance. One possible metric is to label each cluster with the majority label for that cluster using the truth data. Then, for each point we predict the cluster label and measure the mean 0/1 loss. For the digits data set, what score would we get if we placed all the points in one class? Using 1, 2, 5, 10, 21, 44, 94, 200, and 784 principal components what scores does $k = 16$ means clustering obtain?

4 [25 points] Reinforcement Learning [Jingrui]

Consider the following Markov Decision Process:



We have states S_1, S_2, S_3, S_4, and S_5. We have actions Left and Right, and the chosen action happens with probability 1. In S_1 the only option is to go back to S_2, and similarly in S_5 we can only go back to S_4. The reward for taking any action is $r = 1$, except for taking action Right from state S_4, which has a reward $r = 10$. For all parts of this problem, assume that $\gamma = 0.8$.

1. What is the optimal policy for this MDP?
2. What is $V^*(S_5)$? It is acceptable to state it in terms of γ, but not in terms of state values.
3. Consider executing Q-learning on this MDP. Assume that the Q values for all (state,action) pairs are initialized to 0, that $\alpha = 0.5$, and that Q-learning uses a greedy exploration policy, meaning that it always chooses the action with maximum Q value. The algorithm breaks ties by choosing Left. What are the first 10 (state, action) pairs if our robot learns using Q-learning and starts in state S_3 (e.g. $(S_3, Left), (S_2, Right), (S_3, Right), \dots$)?
4. Now consider executing R_{max} on this MDP. Assume that we trust an observed $P(x'|x, a)$ transition probability after a single observation, that the value of $R_{max} = 100$, and that we update our policy each time we observe a transition. Also, assume that R_{max} breaks ties by choosing a policy of Left. What are the first 10 (state, action) pairs if our robot learns using R_{max} and starts in state S_3 (e.g. $(S_3, Left), (S_2, Right), (S_3, Right), \dots$)?