10-701 Machine Learning: Assignment 4

Due on April 27, 2014 at 11:59am

Barnabas Poczos, Aarti Singh

Instructions: Failure to follow these directions may result in loss of points.

- Your solutions for this assignment need to be in a pdf format and should be submitted to the blackboard and the webpage `http://barnabas-cmu-10701.appspot.com` for peer-reviewing.
- We are NOT going to use Autolab in this assignment.
- DO NOT include any identification (your name, andrew id, or email) in both the content and filename of your submission.

Principal Component Analysis (Dani; 10 Points)

For this homework problem, you will implement Principal Component Analysis on the Iris dataset. The Iris dataset contains classifications of iris plants based on four features: sepal length, sepal width, petal length, and petal width. There are three classes of iris plants on this dataset: Iris Setosa, Iris Versicolor, and Iris Virginica. You can download the dataset from `http://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data`.

Implement Principal Component Analysis using any language of your choice. Use the first and second principal components to plot this dataset in 2 dimensions. Use different colors for each of the three classes in your plot.

ICA (Prashant; 10 Points)

The purpose of this problem is to experiment with ICA tool boxes and to see how powerful ICA can be. Firstly, you will need to download an ICA package. I recommend FastICA which you can get here : $http://research.ics.aalto.fi/ica/fastica$. It is available in several languages and you can pick your favorite one.

- We are going to warm up with some synthetic data. Go ahead and make two signals, one sine wave and one sawtooth wave. The code to do this in Matlab would look something like

  ```
  signal1 = sin(linspace(0,50, 1000));
  signal2 = sawtooth(linspace(0,37, 1000));
  ```

  Start out by plotting your two signals. Now, generate two mixtures with random coefficients. An example set of mixtures would be

  ```
  mix1 = signal1 - 2*signal2;
  mix2 = 1.73*signal1 + 3.41*signal2;
  ```

Plot both of these mixtures. Now, use the FastICA package to get back the original signals. Plot the two components. Along with your plots, turn in the code you used to generate them (probably won't be longer than a few lines).

Explain why the recovered components might be scaled differently.

- Now we are going to take a look at the cocktail party problem. When in a party, we hear a mixture of sounds coming from the many sources near us. ICA can help break this signal up into its components. To generate back 3 of the source signals, 3 separate inputs must be observed.

  For the homework, we are going to combine two mono samples of sound and then try to recover the original samples. Go ahead and download the zip file along with the homework with two wav files in it.

  Using the programming language of your choice, load the two wav files. Generate two random mixes of these samples just like we did for the first part. Listen to the mixes if you can. These two mixes will simulate the inputs from which we want to recover the signals. Think of it as two different microphones placed in the party.

  Now recover the two signals using the FastICA package. Since they may be scaled differently, divide each signal by the max value in that signal. Listen to the output and make sure it sounds alright. Each signal may have a whisper of the other signal but it should, on the whole, sound like the original.

  Plot the two original wav files, plot the mixtures you generated, and plot the output signals after normalization.

  Note that in practice, this won't work nearly as well. ICA assumes that the two signals mix with the same delay in each input. However, usually, each signal will reach each microphone at different times.

Boosting (Jit; 10 Points)

Suppose we have a hypothesis class $H \subseteq \{h : \mathcal{X} \to \{-1, 1\}\}$, and some unknown target function $f : \mathcal{X} \to \{-1, 1\}$ such that for any distribution μ over $\mathcal{X}$, there exists an $h \in H$ such that its classification error is at most $\frac{1}{2} - \gamma$, for some $\gamma > 0$:

$$\mathbb{P}_{x \sim \mu}(h(x) \neq f(x)) \leq \frac{1}{2} - \gamma.$$

Let $\text{WeightedMaj}_n(H)$ be the class of weighted majority vote functions consisting of n hypotheses, that is

$$WeightedMaj_n(H) = \{w(x) = sgn[\sum_{i=1}^{n} \alpha_i h_i(x)]\},$$

where data point $x \in \mathcal{X}$, $h_i \in H$, $\alpha_i \geq 0$ and $\sum_i \alpha_i = 1$. Argue that there exists a hypothesis in WeightedMaj_n with training error at most ϵ for $n = O(\frac{1}{\gamma^2} \log(\frac{1}{\epsilon}))$.

Haussler's Bound (Pulkit; 10 points)

In this question we are going to explore Haussler's bound. As we know, Haussler's bound states that for a finite hypothesis space H, m training examples, and any $0 \leq \epsilon \leq 1$, the probability that the true error of any hypothesis, that is consistent with the training data, is bigger than ϵ is

$$P \leq |H| e^{-\epsilon m}$$

Consider a hypothesis space of simple regular expressions that can be learned to model a company's item SKU codes. We are provided with 5 character strings, and the model needs to classify whether this string is a valid SKU code or not. Of course we don't know what the correct format of the code is but we will be provided a set of labeled training examples of valid and invalid SKU codes.

The regular expression comes from a limited subset and has the following characters [c, d, s and .]. (c - characters a-z, d - digits 0-9, s - special symbols and . - any character). The function to be learnt is of the form:
if $\langle X_1, X_2, X_3, X_4, X_5 \rangle$ matches c, ., c, d, s then SKU = VALID, else SKU = INVALID

1. How many training examples would be needed at the least to ensure that with a probability of at least 0.98, a consistent learner would output a hypothesis with true error at most 0.05? [Hint: Think about the size of the possible hypothesis space.] (3 points)
2. What is the maximum error ϵ we can guarantee for a hypothesis learnt from 1000 training examples, with a probability of 0.99? (3 points)
3. Plot the minimum number of training examples required for error bounds in the range [0.01, 0.2]. Do this for confidence probabilities of (0.9, 0.99, 0.999). Submit a plot for this. (We expect three curves of m against ϵ for each probability value on the same graph.) (5 points)

VC Dimension (Pengtao; 10 Points)

1 (2 pts)

Let a function be defined as

$$f(x) = \begin{cases} 1 & \text{if x} \in (a_1, a_2) \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $a_1, a_2 \in \mathcal{R}$, and $a_1 < a_2$. By varying a_1, a_2, we can get a function set. What is the VC dimension of this function set?

2 (2pts)

Now we extend the definition to two dimensions. Let a function be defined as

$$f(x, y) = \begin{cases} 1 & \text{if x} \in (a_1, a_2) \text{ and y} \in (b_1, b_2) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where $a_1, a_2, b_1, b_2 \in \mathcal{R}$ and $a_1 < a_2$, $b_1 < b_2$. Similarly, we can obtain a function set by varying a_1, a_2, b_1, b_2. What is the VC dimension of this function set?

3 (3pts)

Finally we extend the definition to three dimensions. Let a function be defined as

$$f(x, y, z) = \begin{cases} 1 & \text{if x} \in (a_1, a_2) \text{ and y} \in (b_1, b_2) \text{ and z} \in (c_1, c_2) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where $a_1 < a_2$, $b_1 < b_2$, $c_1 < c_2$, and $a_1, a_2, b_1, b_2, c_1, c_2 \in \mathcal{R}$. What is the VC dimension of this function set?

4 (3pts)

Let a function be defined as

$$f(x) = \begin{cases} 1 & \text{if x} \in (a_1, a_2) \text{ or x} \in (a_3, a_4) \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where $a_1, a_2, a_3, a_4 \in \mathcal{R}$ $a_1 < a_2 < a_3 < a_4$. What is the VC dimension of this function set?