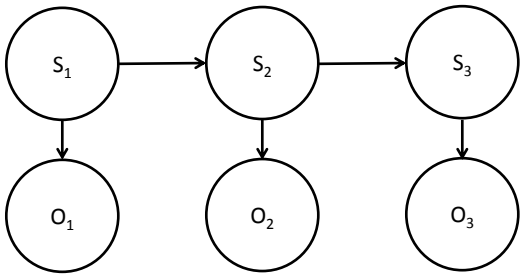
10-601 Machine Learning, Fall 2011: Homework 5

Machine Learning Department
Carnegie Mellon University

Due: Nov 21, 5pm

Instructions There are **3** questions on this assignment. Please submit your completed homework to Sharon Cavlovich (GHC 8215) by **5 pm on Nov 21**. Submit your homework as **3 separate** sets of pages, one for each question. Please staple or paperclip all pages from a single question together, but **DO NOT** staple pages from different questions together. This will make it much easier for the TA's to split up your homework for grading. Include your name and email address on each set.

1 Hidden Markov Models [Shing-hon Lau, 40 points]



1. [5 pts]Assume that we have the Hidden Markov Model (HMM) depicted in the figure above. If each of the states can take on k different values and a total of m different observations are possible for each state, how many parameters are required to fully define this HMM? Justify your answer.

2. [5 pts]What conditional independences hold in this HMM? Justify your answer.

Suppose that we have binary states (labeled A and B) and binary observations (labeled 0 and 1) and the initial, transition, and emission probabilities are as given in the table.

3. [7 pts]Using the forward algorithm, compute the probability that we observe the sequence $O_1 = 0$, $O_2 = 1$, and $O_3 = 0$. Show your work (i.e., show each of your alphas).

State	$P(S_1)$
A	0.99
B	0.01

(a) Initial probs.

S_1	S_2	$P(S_2\|S_1)$
A	A	0.99
A	B	0.01
B	A	0.01
B	B	0.99

(b) Transition probs.

S	O	$P(O\|S)$
A	0	0.8
A	1	0.2
B	0	0.1
B	1	0.9

(c) Emission probs.

4. [7 pts]Using the backward algorithm, compute the probability that we observe the aforementioned sequence ($O_1 = 0$, $O_2 = 1$, and $O_3 = 0$). Recall that $P(\{O_t\}_{t=1}^T) = \sum_k \alpha_1^k \beta_1^k$. Again, show your work (i.e., show each of your betas).

5. [2 pts]Do your results from the forward and backward algorithm agree?

6. [3 pts]Using the forward-backward algorithm, compute (and report) the most likely setting for each state. Hint: you already have the alphas and betas from the above sub-problems.

7. [9 pts]Use the Viterbi algorithm to compute (and report) the most likely sequence of states. Show your work (i.e., show each of your Vs).

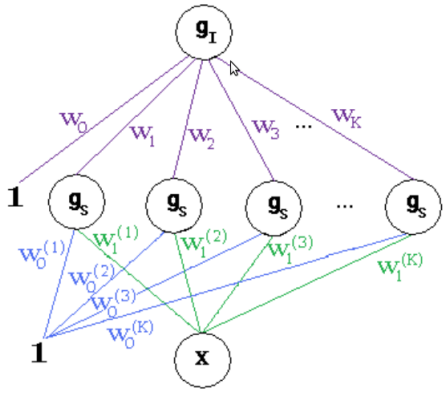
8. [2 pts]Is the most likely sequence of states the same as the sequence comprised of the most likely setting for each individual state? Provide a 1-2 sentence justification for your answer.

2 Neural Networks [Mladen Kolar, 20 points]

In this problem, we will consider neural networks constructed using the following two types of activation functions (instead of sigmoid functions):

- identity
$$g_I(x) = x$$
- step function
$$g_S(x) = \begin{cases} 1 & \text{if } x \geq 0, \\ 0 & \text{otherwise.} \end{cases}$$

For example, the following figure represents a neural network with one input x, a single hidden layer with



K units having step function activations, and a single output with identity activation. The output can be written as

$$out(x) = g_I(w_0 + \sum_{k=1}^{K} w_k g_S(w_0^{(k)} + w_1^{(k)} x))$$

Now you will construct some neural networks using these activation functions.

1. [5 pts] Consider the step function
$$u(x) = \begin{cases} y & \text{if } x \geq a, \\ 0 & \text{otherwise.} \end{cases}$$
Construct a neural network with one input x and one hidden layer, whose response is $u(x)$. Draw the structure of the neural network, specify the activation function for each unit (either g_I or g_s), and specify the values for all weights (in terms of a and y).

2. [5 pts] Now consider the indicator function

$$\mathbb{I}_{[a,b)}(x) = \begin{cases} 1 & \text{if } x \in [a,b), \\ 0 & \text{otherwise.} \end{cases}$$

Construct a neural network with one input x and one hidden layer, whose response is $y\,\mathbb{I}_{[a,b)}(x)$, for given real values y, a and b. Draw the structure of the neural network, specify the activation function for each unit (either g_I or g_s), and specify the values for all weights (in terms of a, b and y).

3. [10 points] A neural network with a single hidden layer can provide an arbitrarily close approximation to any 1-dimensional bounded smooth function. This question will guide you through the proof. Let $f(x)$ be any function whose domain is $[C, D)$, for real values $C < D$. Suppose that the function is Lipschitz continuous, that is,

$$\forall x, x' \in [C, D),\ |f(x') - f(x)| \leq L|x' - x|,$$

for some constant $L \geq 0$. Use the building blocks constructed in the previous part to construct a neural network with one hidden layer that approximates this function within $\epsilon > 0$, that is, $\forall x \in [C, D),\ |f(x) - out(x)| \leq \epsilon$, where $out(x)$ is the output of your neural network given input x. Your network should use only the activation functions g_I and g_S given above. You need to specify the number K of hidden units, the activation function for each unit, and a formula for calculating each weight w_0, w_k, $w_0^{(k)}$, and $w_1^{(k)}$, for each $k \in \{1, 2, \ldots, K\}$. These weights may be specified in terms of C, D, L and ϵ, as well as the values of $f(x)$ evaluated at a finite number of x values of your choosing (you need to explicitly specify which x values you use). You do not need to explicitly write the $out(x)$ function. Why does your network attain the given accuracy ϵ?

3 Principle Component Analysis [William Bishop, 10 points]

1. [5 pts] The fourth slide presented in class on November 10th states that PCA finds principle component vectors such that the projection onto these vectors yields minimum mean squared reconstruction error. In the case of a set of sample vectors $\vec{x}_1 \ldots, \vec{x}_n$ and finding only the *first* principle component, $\vec{v}$, this amounts to finding $\vec{v}$ which minimizes the objective:

$$J(\vec{v}) = \sum_{i=1}^{n} ||\vec{x}_i - (\vec{v}^T \vec{x}_i)\vec{v}||^2$$

To ensure that this problem has a meaningful solution we require that that $||\vec{v}|| = 1$, where $||\cdot||$ denotes the length of the vector.

In class we also learned we can view PCA as maximizing the variance of the data after it has been projected onto $\vec{v}$. If we assume the sample vectors $\vec{x}_1, \ldots, \vec{x}_n$ have zero mean, we can write this as:

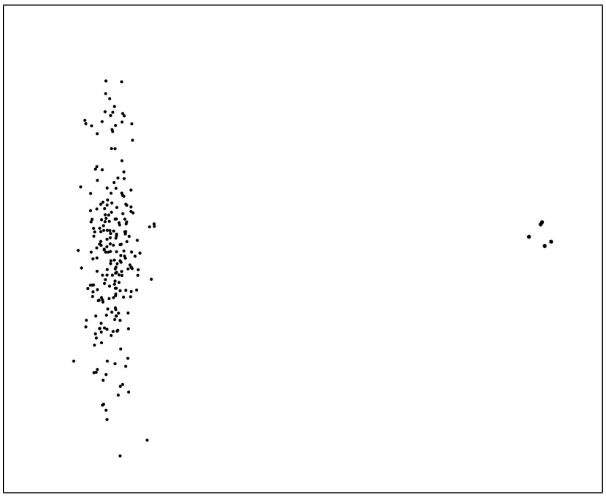
$$Q(\vec{v}) = \sum_{i=1}^{n} (\vec{v}^T \vec{x}_i)^2 = \vec{v}^T X X^T \vec{v}$$

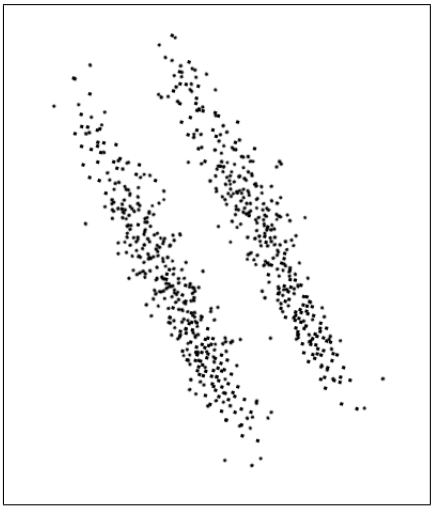
where we again require $||\vec{v}|| = 1$. In the equation on there right, the columns of the matrix X are made up of the sample vectors, i.e., $X = [\vec{x}_1, \ldots, \vec{x}_n]$.

Please show that minimizing $J(\vec{v})$ is equivalent to maximizing $Q(\vec{v})$.

Hint: It is sufficient to show that the objective function J reduces to Q; you do not need to find the actual solution.

2. [5 pts] Draw the first two principle components for the following two datasets.





Based on this, comment on a limitation of PCA.