Name and Student ID: Signature:

Machine Learning BLG527E, Jan 2, 2013, 120mins, 15:00-17:00, Final Exam.

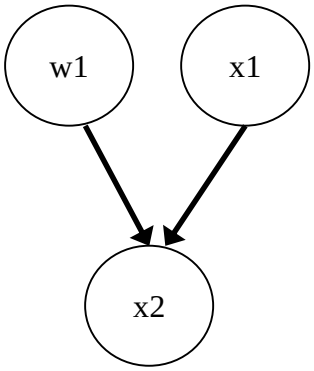
1 20	2 15	3 15	4 25	5 25	Total 100

Duration: 120 minutes.
Open books, closed notes. Write your answers neatly in the space provided for them. Write your name on each sheet. Good Luck!

QUESTIONS

QUESTION1) *[20 points]*
Let $x_1 \sim N(0, s^2)$, that is x_1 is Gaussian distributed with mean 0 and standard deviation s. Let $x_{t+1} = ax_t + w_t$ where $w_t \sim N(0, q^2)$ and the w's are uncorrelated with the x's, and a is given.
(a) [6points] What is the distribution of $x_2 + x_1$?
(b) [7points] What is the distribution of x_2 given x_1 ?
(c) [7points] Draw the Bayesian Network that shows the dependency between x_2 and x_1 and w_1.

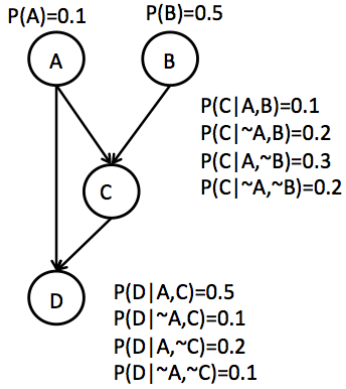
ANSWER1)
1a) $x_2 + x_1 = ax_1 + w_1 + x_1 = (a+1)x_1 + w_1 \sim N(0,(a+1)^2 + q^2)$
1b) $x_2 = ax_1 + w_1$, given x1, ax_1 term is deterministic, the only randomness is in w_1.
$x_2 \mid x_1 \sim N(ax_1, q^2)$
1c)



Name and Student ID: Signature:

1	2	3	4	5	Total
20	15	15	25	25	100

QUESTION2) *[15 points] Given the Bayes Network below, compute P(D|B).*



ANSWER2)
P(D|B)
=P(A,C,D|B)+ P(~A,C,D|B)+ +P(A,~C,D|B) +P(~A,~C,D|B)

P(A,C,D|B)=P(A,B,C,D)/P(B)
=P(A)P(B)P(C|A,B)P(D|A,B)/P(B)
= P(A)P(C|A,B) P(D|A,C)

Applying similar derivation for the other three terms:
P(D|B)
= P(A)P(C|A,B) P(D|A,C)+ P(~A) P(C|~A,B) P(D|~A,C)+ P(A) P(~C|A,B) P(D|A,~C)+ P(~A) P(~C|~A,B) P(D|~A,~C)
=0.1*0.1*0.5 + (1-0.1)*0.2*0.1+0.1*(1-0.1)*0.2+(1-0.1)*(1-0.2)*0.1
=0.113

Common mistakes: ,
You can not write P(D|B)=P(D|C)P(C|B), have to take A into account.

Sum rule was not remembered right by some people:
P(X|Y) = P(Z,X|Y)+P(~Z,X|Y)

QUESTION3) *[15 points]*
Let M be a neural network with 2 inputs and 2 sigmoidal hidden units and a linear output. Let w_{10}, w_{11}, w_{12} and w_{20}, w_{21}, w_{22} be the first layer weights, and v_0, v_1 and v_2 be the output layer weights. Let f(x,w,v) be a function that is implemented by M. Let M' be another neural network with tanh hidden units and let f(x,w,v) = f'(x,w',v') for all x. Compute the weights w' and v' of M' in terms of w an v.

ANSWER3)
For M, let the inputs to the sigmoid nonlinearity in the hidden units be:
$u_1=w_{10}+x_1 w_{11}+ x_2 w_{12}$
$u_2=w_{20}+x_1 w_{21}+ x_2 w_{22}$

Keep the first layer weights to be the same, so that **w'=w,**
Since the input and the first layer weights are the same, $u_1'=u_1$ and $u_2'=u_2$
Also note that for any u: tanh(u)=2sigm(u)-1

We need the outputs of both the sigmoidal (M) and tanh (M') neural networks to be the same.
f'(x,w',v')
$=v_0'+v_1'\tanh(u_1)+v_2'\tanh(u_2)=v_0'+v_1'(2\text{sigm}(u_1)-1)+v_2'(2\text{sigm}(u_2)-1)$
$=v_0'-v_1'-v_2'+2v_1'\text{sigm}(u_1)+2v_2'\text{sigm}(u_2)$
$=v_0 \quad +v_1\text{sigm}(u_1) \quad +v_2\text{sigm}(u_2)= f(x,w,v)$
Therefore, we need:
$v_1'= v_1/2$, $v_2'= v_2/2$
$v_0'-v_1'-v_2'= v_0$ which implies: **$v_0'= v_0+ v_1/2+ v_2/2$**

Name and Student ID: Signature:

QUESTION4) [25 points]

At your latest visit to Las Vegas, your friends started to play a die game in the casino. The die game is very simple. Player pays N dollars at each turn to play and wins 5N dollars if the die comes up 6 and gets nothing otherwise. As a machine learning and probability and statistics expert you don't play since you know that gambling is for losers and house always wins in the long run. ☺

While you observe your friends' game you noticed that croupier (the casino employee managing the game) uses two dice and is switching dice during the game once in a while. After your long observations you realized that the switching frequency of the croupier is 10% from either die and he starts with the 1st die 80 percent of the time. When you investigate the two dice during a break, you realize that 1st die is fair and the 2nd one is loaded. The loaded die has the probability 1/16 for showing a 6 and the rest of the faces have equal probability.

a) [7 points] Is this a fair game (expected earnings is the same for both the player and the casino) if the croupier does not use a loaded die?

b) [7 points] Draw the HMM diagram that explains this game and write down the HMM parameters.

c) [11 points] Given the sequence of rolls [2 5 3 5 2 4 3 1 5 3] and the HMM you constructed in b, what is the probability that croupier made use of the loaded die at least once.

a) Expected earnings for the player:

$$\frac{1}{6}5N - N = -\frac{N}{6}$$

Expected earnings for the house:

$$N - \frac{1}{6}5N = \frac{N}{6}$$

So it is not fair. Player will lose in the long run. House always wins ;)

b) There are two states, say F and L, representing fair die and loaded die, respectively. For observables, you can model the system for 6 observables, namely the faces of the die. Alternatively, you can simply model the observables as getting a 6 or not, as non-6 faces are identical for the sake of game result. We accepted both answers.

Π={F=0.8, L=0.2}, $A = \begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix}$ $B = \begin{matrix} 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 3/6 & 3/16 & 3/16 & 3/16 & 3/16 & 1/16 \end{matrix}$

c) Since we know that this sequence is observed, the probability of generating the given sequence with this HMM visiting state L at least once is 1 – Px/Pa where Pa stands for the probability of this sequence being generated by this HMM (in any way possible) and Px stands for the probability of generating this sequence without ever visiting L.

$$Px = 0.8\left(\frac{1}{6}\right)^{10}(0.9)^9$$

$$Pa = \alpha_{10}(F) + \alpha_{10}(L)$$

Name and Student ID: Signature:

QUESTION5) [25 points]
You are required to build a two-class (C_1 and C_2) protein classifier.
Proteins can be classified based on their **sequence similarity** to other proteins or their **physicochemical properties**.
You have a function $S(x,x_t)$ that calculates the sequence similarity score between proteins x and x_t . You can assume that this similarity calculation is a proper kernel and can be converted into a distance metric using for example $1/(1+ S(x,x_t))$.
You have a function F(x) that returns a 10 dimensional vector of physicochemical properties for a given protein x.
Based on the protein sequence, you can compute $p_a(x)$, the probability of protein x containing an α-helix. You know that proteins that contain an α-helix structure are best classified by using their sequence similarity to members of the classes. For proteins that do not contain an α-helix structure, their physicochemical properties are more useful.
You have a training dataset that contains proteins and whether they belong to C_1 or C_2.

a) [7 points] Design a classifier that can classify a new protein x, based on the sequence similarity.
b) [7 points] Design a classifier that can classify a new protein x, based on physicochemical properties.
c) [11 points] Design a kernel that can be used by a support vector machine that combines these tools ($S(x,x_t)$, F(x) and $p_a(x)$) and information in a way you think that will optimize the classification accuracy.

a) Many possibilities. Since $S(x,x_t)$ is a proper kernel we can directly train a support vector machine classifier with this kernel S and the training data. Or, given a new protein x_n, we can use S to find the most similar k sequences in the training data and classify it using the labels of these k most similar sequences.
b) Since F(x) gives us 10 dimensional vectors, we can use almost any classification algorithm we know. We can use parametric methods with a Gaussian distribution assumption on the classes or we can use naïve bayes, k-means, multi-layer perceptron. Or we can use any vectorial kernel (such as rbf, polynomial, etc.) with a support vector machine.
c) Since we are asked to define a single kernel, we can combine the sequence similarity based kernel S with the kernel of our choice on the vectors of physicochemical properties given by F(x). We know that any linear combination of kernels gives us another proper kernel. So one such kernel can be:

$$K(x,x_t) = S(x,x_t) + F(x)^T F(x_t)$$

But this kernel gives equal importance to both kernels and does not use the prior information that proteins with α-helix structure is better classified by using sequence similarity. A better approach would be giving more weight to sequence similarity kernel when the probability of existence of an α-helix in the new protein is higher. Such as:

$$K(x,x_t) = p_a(x)\, S(x,x_t) + (1-p_a(x))\, F(x)^T F(x_t)$$

Note that for the physicochemical properties any vectorial kernel mentioned in part (b) can be used.