Name: **Andrew ID:**

Final Exam, 10701 Machine Learning, Spring 2009

- The exam is open-book, open-notes, no electronics other than calculators.
- The maximum possible score on this exam is 100. You have 3 hours.
- Don't spend too much time on any one problem. If you get stuck on any of the problems, move on to another one and come back to that problem if you have time.

Good luck!

Question	Max. Score	Score
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		

Question 1 – Short Questions

(1). Let X, Y and Z be random variables. X ~ Unif(0,1). Let $0 < a < b < 1$

$$Y = \begin{cases} 1 & if \ 0 \le X \le a \\ 0 & otherwise \end{cases}$$

$$Z = \begin{cases} 1 & if \ b \le X \le 1 \\ 0 & otherwise \end{cases}$$

Is Y & Z independent ? Show why / why not ?

Answer:

Y & Z are not independent. This is because P(Y,Z) != P(Y) * P(Z)

Y	Z	P(Y)	P(Z)	P(Y)P(Z)	P(Y,Z)
0	0	1-a	b	b-ab	b-a
0	1	1-a	1-b	1-a-b+ab	1-b
1	0	a	b	ab	a
1	1	a	1-b	a-ab	0

Find $E_Y (Y | Z = z)$ for each value of z.

Z = 0

E(Y| Z = 0) = 0 P(Y=0|Z=0) + 1 P(Y=1|Z=0)

= 0 + 1 * a / b = a / b

E(Y| Z = 1) = 0 P(Y=0|Z=0) + 1 P(Y=1|Z=0) = 0

(2). We are trying to learn regression parameters for a dataset which we know was generated from a polynomial of a certain degree, but we do not know what this degree is. Assume the data was actually generated from a polynomial of degree 5 with some added Gaussian noise (that is $y = w_0+w_1x+w_2x^2+w_3x^3+w_4x^4+w_5x^5 + \varepsilon$, ε~N(0,1)).

For training we have 100 $\{x,y\}$ pairs and for testing we are using an additional set of 100 $\{x,y\}$ pairs. Since we do not know the degree of the polynomial we learn two models from the data. Model A learns parameters for a polynomial of degree 4 and model B learns parameters for a polynomial of degree 6. Which of these two models is likely to fit the *test* data better?

Answer: Degree 6 polynomial. Since the model is a degree 5 polynomial and we have enough training data, the model we learn for a six degree polynomial will likely fit a very small coefficient for x^6 . Thus, even though it is a six degree polynomial it

will actually behave in a very similar way to a fifth degree polynomial which is the correct model leading to better fit to the data.

(3). What is the VC dimension for Linear Support Vector Machines in d-dimensional space?

Answer: d+1

(4). True or false? Any decision boundary that we get from a generative model with class-conditional Gaussian distributions could in principle be reproduced with an SVM and a polynomial kernel.

True! In fact, since class-conditional Gaussians always yield quadratic decision boundaries, they can be reproduced with an SVM with kernel of degree less than or equal to two.

(5). True or false? AdaBoost will eventually reach zero training error, regardless of the type of weak classifier it uses, provided enough weak classifiers have been combined.

False! If the data is not separable by a linear combination of the weak classifiers, AdaBoost can't achieve zero training error.

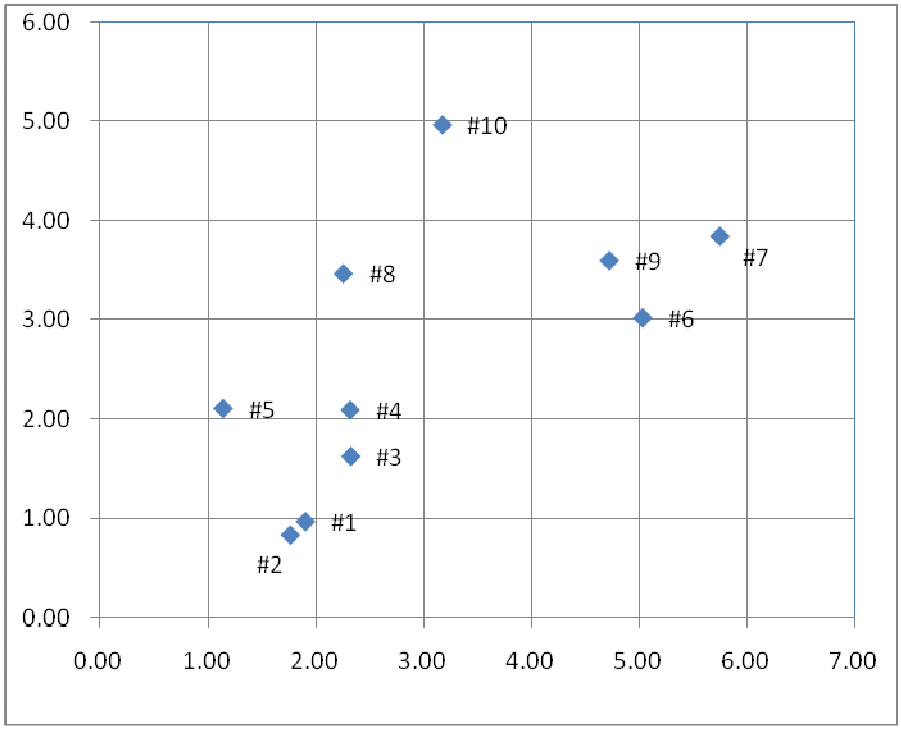
(6). How can we determine the appropriate number of *states* for a Hidden Markov model?

Answer:
One method is to use cross validation. While more states would fit the training data well, adding more states may reduce transition probabilities leading to lower likelihood for the test data.

(7). **[2 Points]** Let a configuration of the k means algorithm correspond to the k way partition (on the set of instances to be clustered) generated by the clustering at the end of each iteration. Is it possible for the k-means algorithm to revisit a configuration? Justify how your answer proves that the k means algorithm converges in a finite number of steps.

Answer: Since the k means algorithm converges if the k way partition does not change in successive iterations, thus the k way partition has to change after every iteration. As the mean squared error monotonically decreases it is thus impossible to revisit a configuration. Thus eventually the k means algorithm will run out of configurations, and converge.
NB: The maximum no of iterations corresponds to the no of k way partitions possible on a set of n objects : S(n,k) where S are Stirling numbers of the 2^{nd} kind.

(8). Suppose you are given the following <x,y> pairs. You will simulate the k-means algorithm and Gaussian Mixture Models learning algorithm to identify TWO clusters in the data.



Data #	x	y
1	1.90	0.97
2	1.76	0.84
3	2.32	1.63
4	2.31	2.09
5	1.14	2.11
6	5.02	3.02
7	5.74	3.84
8	2.25	3.47
9	4.71	3.60
10	3.17	4.96

Suppose you are given initial assignment cluster center as {cluster1: #1}, {cluster2: #10} – the first data point is used as the first cluster center and the 10-th as the second cluster center. Please simulate the k-means (k=2) algorithm for ONE iteration. What are the

cluster assignments after ONE iteration? Assume k-means uses Euclidean distance. What are the cluster assignments until convergence? (Fill in the table below)

Data #	Cluster Assignment after One Iteration	Cluster Assignment after convergence
1	1	1
2	1	1
3	1	1
4	1	1
5	1	1
6	2	2
7	2	2
8	2	1
9	2	2
10	2	2

(9). Assume we would like to use spectral clustering to cluster *n* elements. We are using the *k* nearest neighbor method we discussed for generating the graph that would be used in the clustering procedure. Following this process:

What is the maximum number of nodes that a single node is connected to?

Answer: n-1

What is the minimum number of nodes that a single node is connected to?

Answer: k

(10). Can SVD and PCA produce the same projection result? Yes/No? If YES, under what condition they are the same? If NO, please explain why? (briefly)

Answer:

Yes. When the data has a zero mean vector, otherwise you have to center the data first before taking SVD.

(11). Let X_i be the independent variable, and Y_i be the dependent variable. We will use X_i to predict Y_i, using several models of regression.

M1 : $Y_i = aX_i + eta$

M2: $Y_i = aX_i + b + eta$

M3: $Y_i = aX_i^2 + bX_i + c + eta$

where we fit the constants a, b and c from data. We assume that Eta~N(0,σ^2) and σ^2 is estimated from the training data.

Let us choose a model from M1, M2 and M3 using the AIC.

a. How many degrees of freedom does each of the three models (M1, M2,) M3 have?

Answer:

Degrees of freedom for Model 1 = 2 (a and eta)

Degrees of freedom for Model 1 = 3 (a, b and eta)

Degrees of freedom for Model 1 = 4 (a,b, c and eta)

b. Let log likelihood(data| ML parameters of model M1) = -130.4,
log likelihood(data| ML parameters of model M2) = -108.1,
log likelihood(data| ML parameters of model M3) = -107.99

Based on the ML framework, which model should we choose ?

According to the ML framework, we should choose the model with the max likelihood, and also max log likelihood : M3

c.

Based on the AIC, which model should we choose? Does the choice change if we observe the same likelihoods, but learn that M1, M2 and M3 all had 2 more degrees of freedom ?

According to the ML framework, we should choose the model with the max likelihood, and also max log likelihood : M3

AIC score = log likelihood(data|model) – degrees of freedom

AIC score for M1 = -130.4 – 2

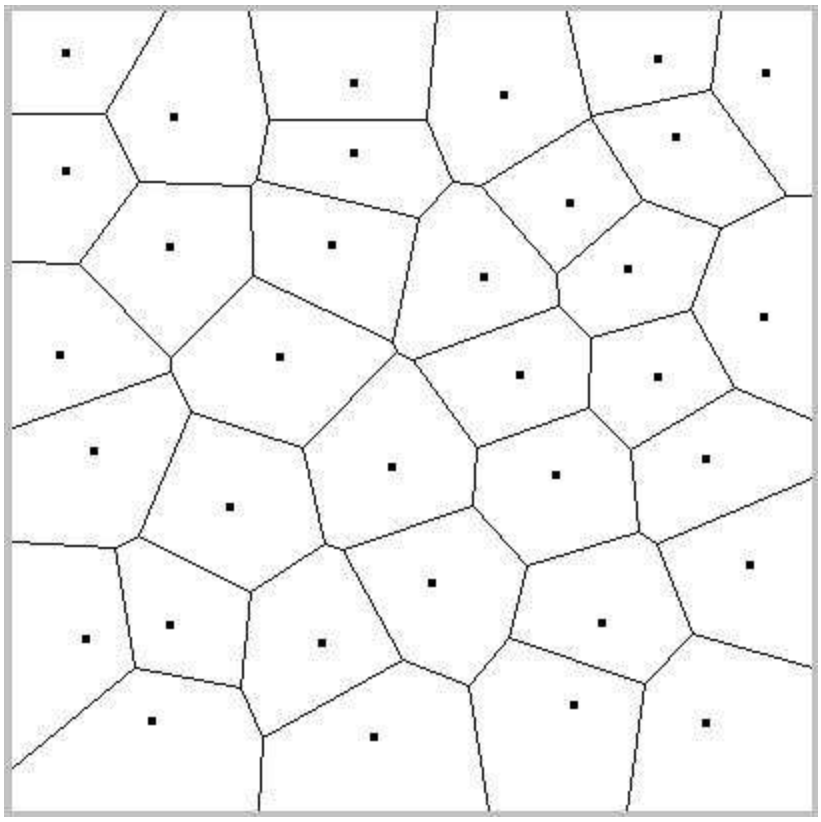
AIC score for M2 = -108.1 - 3

AIC score for M3 = -107.99 - 4

Thus, under maximizing AIC criterion, M2 should be chosen. If an equal number of degrees of freedom are added to every model, the model choice does not change if the likelihd is remains unchanged, as an equal number is decreased from each AIC score.

Question 2 – Nearest Neighbors [8 Points]

Let us try and classify data points in 2D Euclidean space. We are given n instances of such points P1, P2, …, Pn and the corresponding category for each point C1, C2, Cn [where C1, C2, .. Cn take values from the set of all possible class labels]. Under the k nearest neighbors classification scheme, each new element Q is simply categorized by a majority vote among its k nearest neighbors in instance space. The 1-NN is a simple variant of this which divides up the input space for classification purposes into a convex region (see figure below for the 1NN decision boundaries under the Euclidean distance measure), each corresponding to a point in the instance set.



(1). Is it possible to build a decision tree (with decisions at each node of the form "is x > a", "is x < b", "is y > c", or "is y < d" for any real constants a,b,c,d) which classifies exactly according to the 1-NN scheme using the Euclidean distance measure ? If so, explain how. If not, explain why not.

No. the decision boundaries for 1 - NN correspond to the cell boundaries of each point (see image) and are not necc parallel to the coordinate axes. The decision tree boundaries would always be parallel to coordinate axes based on the kinds of qns asked at each node of the decision tree. To approximate a gradient by decision trees

could take an arbitrary (uncountable) number of decisions, not possible in a decision tree.

(2). Now assume that the distance measure is not explicitly specified to you. Instead, you are given a "black box" where you input a set of instances P1, P2, .. Pn and a new example Q, and the black box outputs the nearest neighbor of Q, say Pi and its corresponding class label Ci. Is it possible to construct a k-NN classification algorithm based on this black box alone? If so, how and if not, why not?

Yes. First use the 1NN algorithm on the instance set for the new example, note the nearest neighbor and it s class and throw it out of the instance set. Use 1NN now with the reduced instance set and the example to be classified and again note the nearest neighbor and its class and throw it out of the instance set. Repeat this process k times, take a majority vote among the noted classes, and classify the new example accordingly.

(3). If the black box returns the j nearest neighbors (and their corresponding class labels) instead of the single most nearest neighbor (assume j != k) , is it possible to construct a k-NN classification algorithm based on the black box ? If so how, and if not why not ?

If j < k, then use the algorithm floor(k/j) times to obtain the j * floor(k/j) nearest neighbors and their classes. To obtain the remaining k – j * floor(k/j) nearest neighbors use the j NN one more time and note the final batch of j nearest neighbors. Now to order the last set of j nearest neighbors and choose the top k – j * floor(k/j) nearest neighbors from them.
To do this, we merely need to construct an instance set using the final batch of j nearest neighbors and any j – (k – j*floor(k/j)) nearest neighbors from the set of the top j nearest neighbors. The (k – j*floor(k/j)) elements from the last batch which get picked as the j nearest neighbors are thus the top k – j *floor(k/j) elements in the last batch of j nearest neighbors that we needed to identify.

If j > k, we cannot do k-NN using the j-NN algorithm black box.

Question 3 –Decision Trees

We would like to construct a decision tree for n vectors each with m attributes.

(1). Assume that there exists *i* and *j* such that for ALL vectors X in our training data, these attributes are equal ($x_i=x_j$ for all vectors where x_i is the i'th entry in the vector X). Assume that we break ties between them by using x_i (that is, if both lead to the same conditional entropy we would use x_i). Can removing attribute *j* from our training data change the decision tree we learn for this dataset? Explain briefly.

Answer: No, removing attribute *j* would not change the decision tree we learn. Since *i* and *j* are the same attribute *j* cannot add any information that is not already used in attribute *i* .

(2). Assume we have two equal vectors X and Z in our training set (that is, all attributes of X and Z including the labels are exactly the same). Can removing Z from our training data change the decision tree we learn for this dataset? Explain briefly.

Answer: Yes, the decision tree can change. The conditional entropy in each split depends on the set of samples and copying a vector twice may change the distribution leading to a selection of a different attribute to split on.

For the next set of questions consider a dataset with continuous attributes. For such attributes we said in class we can use threshold splits to determine the best partition for a set of vectors. Assume we are at the root and we have ***n*** vectors, all with different (continuous) values for attribute *x1*.

(3). Assume we would like to use binary splits. For such splits we need to choose a value *a* and split the data by propagating all vectors with $x1<a$ to the left and those with $x1 \geq a$ to the right. For any value of *a* we consider we would like to have at least one vector assigned to each of the two branches of the split. How many values of *a* do we need to consider?

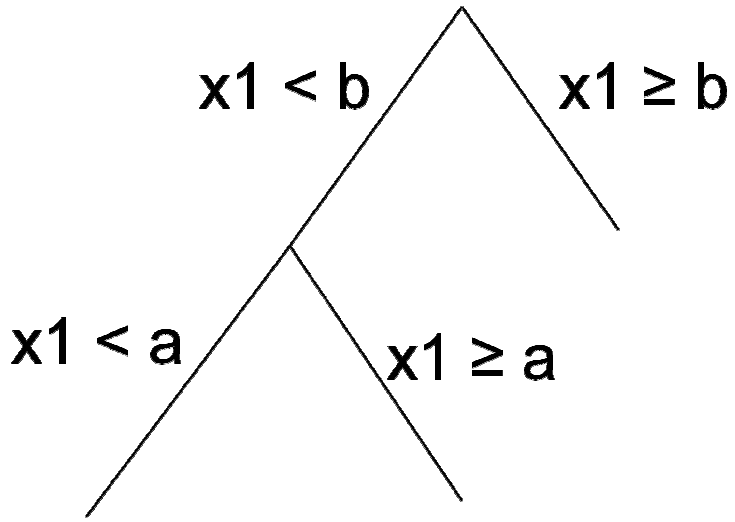
Answer: n-1. We order the values for attribute *x1* and test all values between two points in the ordering.

(4). Assume we would like to use three way splits. For such splits we need to chose values *a* and *b* such that $a < b$ and split the data into three sets: $x1 < a$, $a \leq x1 < b$, $x1 \geq b$. Again we require that for any value of *a* and *b* that we consider at least one vector would be assigned to each of the three branches. How many $\{a,b\}$ do we need to consider?

Answer: Any two of the values selected in (3) would lead to a three way split that has at least one vector assigned to each leaf. So the answer is $\binom{n-1}{2} = \frac{(n-1)(n-2)}{2}$

(5). For a given three way split at the root (parameterized by an $\{a,b\}$ pair) can we reconstruct the same split with a tree that uses only binary splits?
If no briefly explain why.
If yes, show the tree that leads to the same set of leaves with the same nodes in each leaf as the three way split.

Answer: Yes. See figure.

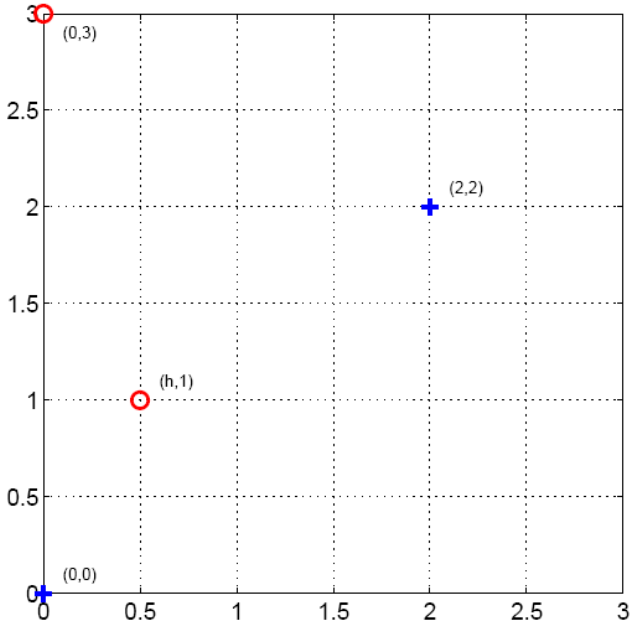


(6). If you answered 'no' to 5 explain which type of data can be correctly separated (classified) by a three way split tree but cannot be correctly classified using a binary split tree.
If you answered 'yes' to 5 explain why it may still be beneficial to learn a three way split tree rather than a binary split tree.

Answer: The binary split search for the best initial split and only then looks for the second split. Thus, it's a greedy procedure. The three way splits evaluates all possible pairs together leading to a better three way split. Of course, this comes at the expense of a larger run time.

Question 4 – Support Vector Machine

Suppose we only have four training examples in two dimensions (see figure above):



positive examples at x1 = [0, 0] , x2 = [2, 2] and negative examples at x3 = [h, 1] , x4 = [0, 3], where we treat $0 \leq h \leq 3$ as a parameter.

(1). How large can $h \geq 0$ be so that the training points are still linearly separable?

Up to (excluding) h=1

(2). Does the orientation of the maximum margin decision boundary change as a function of h when the points are separable (Y/N)?

No, because x1, x2, x3 remain the support vectors.

(3). What is the margin achieved by the maximum margin boundary as a function of h? [Hint : It turns out that the margin as a function of h is a linear function.]

m(h)=0 for h=1 and m(h)=sqrt(2)/2 for h=0 m(h)=sqrt(2)/2 – sqrt(2)/2 * h

(4). Assume that we can only observe the second component of the input vectors. Without the other component, the labeled training points reduce to (0,y = 1), (2,y = 1), (1,y = -1), and (3,y =-1). What is the lowest order p of polynomial kernel that would allow us to correctly classify these points?

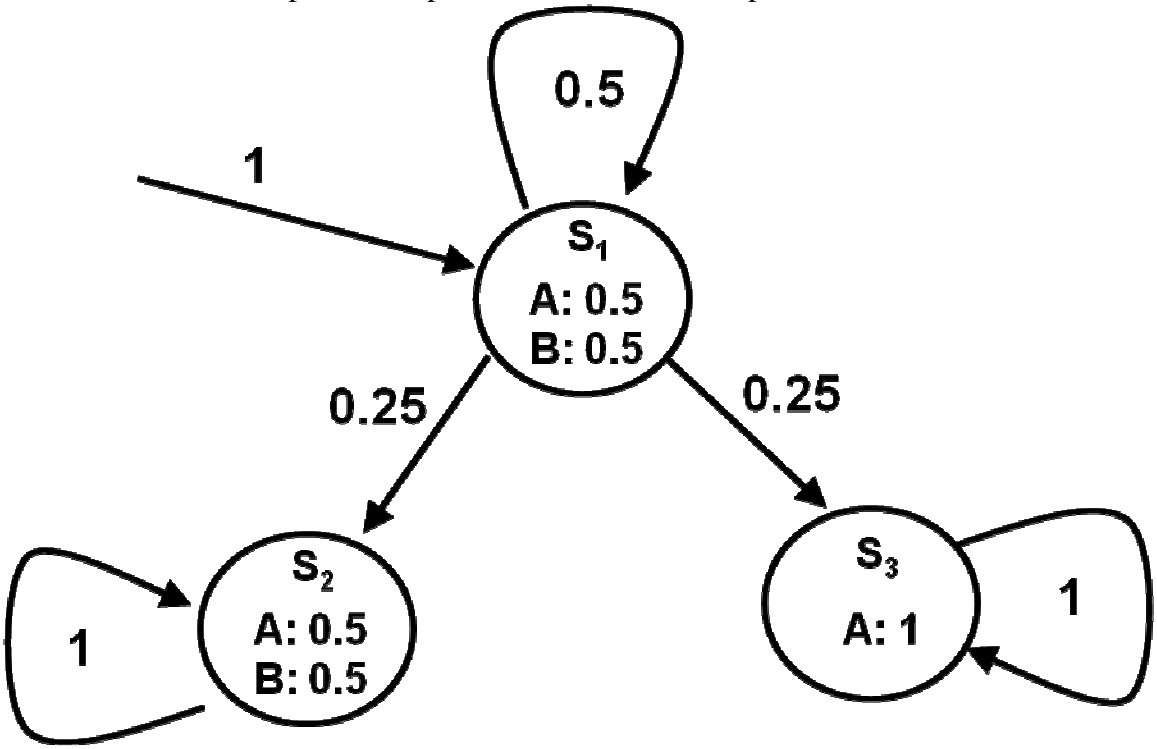
The classes of the points on the x2-projected line observe the order 1,-1,1,-1. Therefore, we need a cubic polynomial.

Question 5 –HMM

In class we used the forward term (which we defined as α) to compute the following probability for a set of observed outputs:

$P(O_1 \ldots O_T)$

In this problem we will use a different term for deriving this probability and will use this new derivation to compute some probabilities for an example HMM.



(1). Let $v_i^t = p(O_1 \ldots O_T \mid q_t = s_i)$. Write a formula for $P(O_1 \ldots O_T)$ using *only* v_i^t and $p_t(i)$ which we defined in class (in class we defined $p_t(i) = p(q_t = s_i)$).

Answer:

$$\begin{aligned} p(O_1 \ldots O_T) &= \sum_i p(O_1 \ldots O_T, q_t = s_i) \\ &= \sum_i p(O_1 \ldots O_T \mid q_t = s_i) p(q_t = s_i) \\ &= \sum_i v_i^t p_t(i) \end{aligned}$$

For the next two questions consider the HMM in figure 1. Initial and transition probabilities are listed next to the corresponding edges. Emission probabilities and the states' names are listed inside each node. For example, for state S_2 the emission probabilities are: 0.5 for A and 0.5 for B.

(2). Use only v_i^t and $p_t(i)$ as in (1) to compute $p(O_1 = B,\ldots, O_{200} = B)$ (the probability of observing 200 B's in a row). You need to write an appropriate *t* for this computation and then explicitly derive the values of v_i^t and $p_t(i)$ for the *t* that you have chosen and show how you can use these values to compute the probability of this output.

Hint: for computing $p_t(i)$ note that the transitions to and from S_2 and S_3 are symmetric and so for any *t*, $p_t(S_2) = p_t(S_3)$.

Answer: We will use t=200. For this t we have $p(O_1 = B,\ldots, O_{200} = B \mid S_3) = 0$ and $p(O_1 = B,\ldots, O_{200} = B \mid S_1) = p(O_1 = B,\ldots, O_{200} = B \mid S_2) = (1/2)^{200}$
Also, $p_{200}(1)=(1/2)^{199}$ and, based on the hint $p_{200}(2) = (1- (1/2)^{199})/2$

Putting this together we get: $(1/2)^{399} + (1/2)^{201} (1- (1/2)^{199})$

(3). Use only v_i^t and $p_t(i)$ as in (1) to compute $p(O_1 = A,\ldots, O_{200} = A)$ (the probability of observing 200 A's in a row). Again, you would need to find an appropriate *t* for this computation. However, for this part you can use v_1^t for the *t* that you have chosen in your solution (that is, you do not need to derive the value of v_1^t). Note that this applies only to v_1^t. You would still need to derive the actual values of v_2^t and v_3^t and $p_t(i)$ (for all i) for the *t* that you have chosen and show how you can use these values to compute the probability of this output

Hint – the *t* for (3) may be different from the *t* you selected for (2).

Answer: We will select t=2 for this part. For this t we have
$p(O_1 = A,\ldots, O_{200} = A \mid q_2=S_2) = (1/2)^{200}$
$p(O_1 = A,\ldots, O_{200} = A \mid q_2=S_3) = (1/2)$

$p(O_1 = B, \ldots, O_{200} = B \mid q_2 = S_1) = v_1^2$

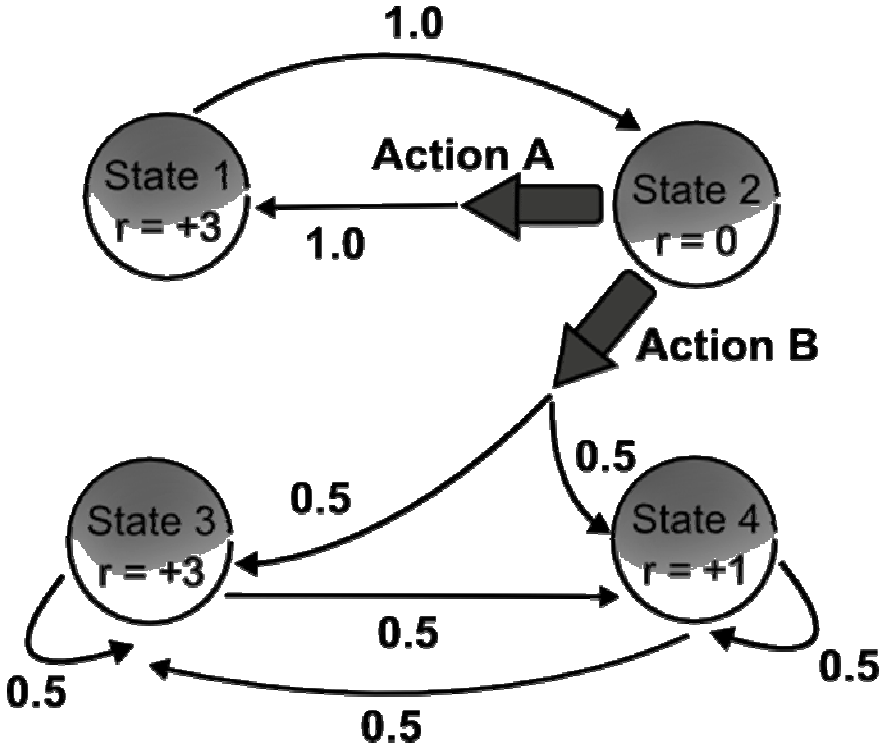
and
$p_2(1) = 0.5$, $p_2(2) = p_2(3) = 0.25$

Putting it together we get:

$0.5\, v_1^2 + (1/2)^{202} + (1/2)^3$

Question 6 – Markov Decision Process

You are given the following Markov decision process, where r denotes the reward at each state :



1. Which action, A or B, maximizes our expected reward on the following turn, for the starting state State 2 ?

Action A

2. Which action from State 2 maximizes the total expected discounted future reward, with a discount factor γ of 0.9? What is the expected discounted future reward for each action?

Action B:

$R(A) = \text{sum\_i=0\^infty } 3*\gamma^{2i+1} + 0*\gamma^{2i} = 3\gamma \text{ sum\_k=0\^infty } (\gamma^2)^k = 3\gamma/(1-\gamma^2) = 14.21$

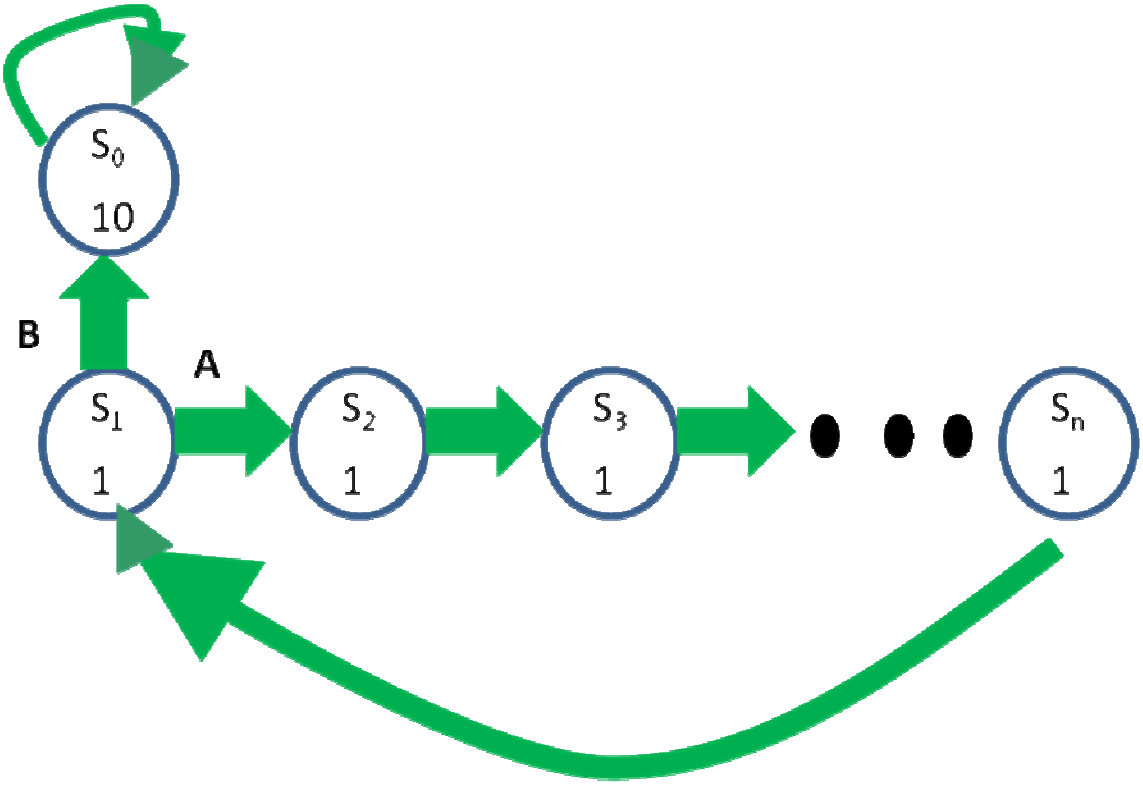
$R(B) = \text{sum\_i=1\^infty } .5[3\gamma^i + 1\gamma^i] = 2 \text{ sum\_i=1\^infty } \gamma^i = 2/(1-\gamma) - 2 = 18$

3. For what value of γ does the expected discounted future reward for each action from State 2 become equal ?

$3\gamma/(1-\gamma^2) = 2/(1-\gamma) - 2$ iff $3\gamma = 2(1+\gamma) - 2(1+\gamma)(1-\gamma)$ iff $0 = -\gamma + 2\gamma^2$ iff $\gamma = 1/2$ (since $0 < \gamma < 1$)

Question 7 –Reinforcement Learning

Consider the RL model in Figure 1 below. Assume $n > 1000$. All states accept S_1 have only one possible action with a deterministic outcome (transition probability of 1). State S_1 has two possible actions, A and B, each with a deterministic outcome (A always leads to S_2 and B always leads to S_0). Assume a discount factor of $\gamma = 0.5$.
To learn this model we will use Q learning with $\alpha=1$. All Q functions for all states are initialized to 0. Whenever we reach state S_1 we use our current Q function estimate to chose the action leading to the highest long term pay. We break ties by choosing action A. We start at state S_1 .



(1). After 1 step, what are $Q(S_1, A)$ and $Q(S_1, B)$?

Answer: $Q(S_1, A)=1$ and $Q(S_1, B) = 0$

(2). After 5 steps, what are $Q(S_1, A)$ and $Q(S_1, B)$?

Answer: $Q(S_1, A)=1$ and $Q(S_1, B) = 0$

(3). After n+5 steps, what are $Q(S_1, A)$ and $Q(S_1, B)$?

Answer: $Q(S_1, A)=1.5$ and $Q(S_1, B) = 0$

(4). When our Q learning converges, what are the convergence values for $Q(S_1, A)$ and $Q(S_1, B)$? What is the convergence value for $Q(S_2, \text{right})$?

Answer: Note, since we are breaking ties by going right all states are symmetric so when we converge we have: $Q^*(S_1, A)=1+ 0.5Q^*(S_2, \text{right})=1+0.5\ Q^*(S_1, A)$ and so:

$Q(S_1, A)=2$, $Q(S_1, B) = 0$, $Q(S_2, \text{right})=2$

(5). Now, lets treat this as a MDP and compute J* using value iteration. What is $J^*(S_1)$?

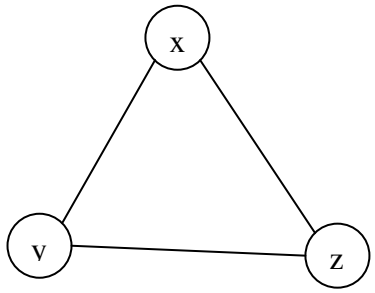
Answer: We first compute $J^*(S_0)$. $J^*(S_0) = 10+0.5J^*(S_0)$ and so $J^*(S_0) = 20$. Now, $J^*(S_1) = 1 + 0.5\ J^*(S_0) = 11$

(6). If we use $\varepsilon = 0.01$ how many iteration do we need for J* to converge for all states?

Answer: In this case the state that we converge the slowest is S_0 since it receives the largest increase at each iteration (S_1) will converge at a similar rate and all other states would converge at least as fast, most faster). In iteration k of the value iteration we increase $J^*(S_0)$ by $(10/2^k)$. So in order to determine how many iteration we need for $\varepsilon = 0.01$ we need to find the smallest k s.t. $(10/2^k) < (1/100)$ which leads to k=10.

Question 8 –Graphical Models

Consider the following undirected Graphical Model on x, y and z. All three variables can take values in {-1, 1}.



The associated potential functions are given as:

$$\psi(x,y) = 2xy + x^3$$
$$\psi(x,z) = xz + z$$
$$\psi(y,z) = yz^3$$

(1). What is the posterior marginal probability $P(y|x=1)$?

Answer:

$$P(x,y,z) = \frac{1}{Z}\exp(2xy + x^3 + xz + z + yz^3)$$

$$P(y|x=1) = \sum_z P(y,z|x=1) = \sum_z \frac{P(y,z,x=1)}{P(x=1)}$$

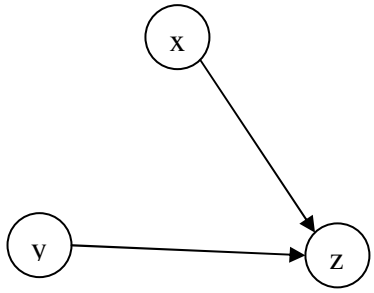
$$P(y=1|x=1) = \frac{P(y=1,z=1,x=1) + P(y=1,z=-1,x=1)}{P(x=1)} = \frac{e^6 + e^0}{P(x=1)}$$

$$P(y=-1|x=1) = \frac{P(y=-1,z=1,x=1) + P(y=-1,z=-1,x=1)}{P(x=1)} = \frac{e^0 + e^{-2}}{P(x=1)}$$

$$P(y=1|x=1) = 0.997$$

$$P(y=-1|x=1) = 0.003$$

(2). Consider the following directed Graphical Model



Is there an undirected Graphical Model for x, y and z, with the same set of probabilistic dependency/independency? If YES, please draw the model. If NO, please explain briefly which dependency/independency cannot be modeled using an undirected model?

Answer:

NO.

Because the independency of x and y given z cannot be represented in undirected graphical model.

Question 9 – Clustering [8 Points]

(1). Consider the dataset :
{0, 4, 5, 20,25, 39, 43, 44}
Suppose we want the two top level clusters from this dataset. What will single link, complete link and average link output as the two clusterings ? If single link and complete link give the same 2 clusters, does it follow that average link will output the same 2 clusters ? Explain.

Single Link : {0,4,5} {20,25,39,43,44}
Complete Link : {0,4,5} {20,25,39,43,44}
Avg Link : {0,4,5} {20,25,39,43,44} or {0,4,5,20,25}{39,43,44} – as both clusterings at the final step correspond to the distance of 117/6 between clusters. This shows that even if single link and complete link produce the same cluster, avg link might behave differently for clustering.

(2). We would like to clusters the numbers from 1 to 1024 using hierarchical clustering. We will use Euclidian distance as our distance measure. We break ties by combining the two clusters in which the lowest number resides. For example, if the distance between clusters A and B is the same as the distance between clusters C and D we would chose A and B as the next two clusters to combine if min{A,B} < min{C,D} where {A,B} are the set of numbers assigned to A and B.

We would like to compare the results of the three linkage methods discussed in class for this dataset. For each of the three methods, specify the number of elements (numbers) assigned to *each of the two clusters* defined by the root (that is, what are the sizes of the two clusters if we cut the hierarchical clustering tree at the root or in other words what are the sizes of the last two clusters that we combine).

Single link: 1023 + 1 , clustering (((1,2), 3), 4 ….

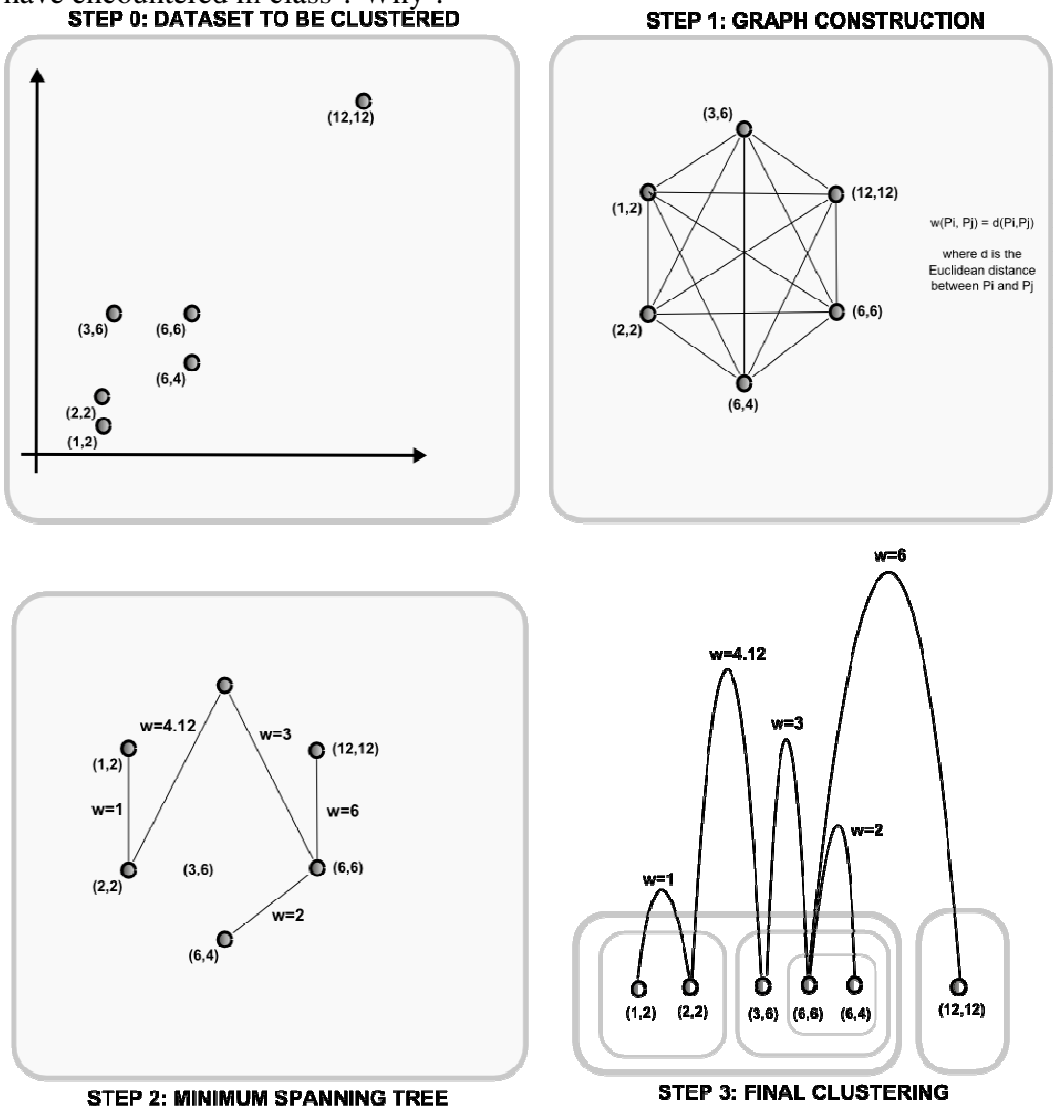
Complete link: 512 + 512 (((1,2), (3,4)) …

Average link: 512 + 512 (((1,2), (3,4)) …

(3). Hierarchical clustering may be bottom up or top down. Can a top down algorithm be exactly analogous to a bottom up algorithm ? Consider the following top down algorithm

1. Calculate the pairwise distance d(Pi, Pj) between every two object Pi and Pj in the set of objects to be clustered and build a complete graph on the set of objects with edge weights = corresponding distances
2. Generate the Minimum Spanning Tree of the graph I.e. Choose the subset of edges E' with minimum sum of weights such that G' = (P,E') is a single connected tree.
3. Throw out the edge with the heaviest weight to generate two disconnected trees corresponding to two top level clusters.
4. Repeat this step recursively on the lower level clusters to generate a top down clustering on the set of n objects

Does this top down algorithm perform analogously to any bottom up algorithm that you have encountered in class ? Why ?



The clustering corresponds to single-link bottom-up clustering. The edges used to calculate the cluster distances for the single link bottom up clustering correspond to the edges of the MST (since all points must be clustered, and the cluster distance is single link and chooses the min wt edge joining together two so far unconnected clusters). Thus, the heaviest edge in the tree corresponds to the top most cluster, and so on. See example above.

Question 10 – Dimensionality Reduction

You have the following data:

data #	x	y
1	5.51	5.35
2	20.82	24.03
3	-0.77	-0.57
4	19.30	19.38
5	14.24	12.77
6	9.74	9.68
7	11.59	12.06
8	-6.08	-5.22

You want to reduce the data into a single dimension representation. You are given the first principal component (0.694, 0.720).

(1). What is the representation (projected coordinate) for data #1 (x=5.51, y=5.35) in the first principal space?

Answer: (-5.74 or -5.75)

(2). What are the xy coordinates in the original space reconstructed using this first principal representation for data #1 (x=5.51, y=5.35)?

Answer: (5.31, 5.55)

(3). What is the representation (projected coordinate) for data #1 (x=5.51, y=5.35) in the second principal space?

Answer: 0.28
(±0.28, ±0.25 are accepted.)

(4). What is the reconstruction error if you use two principal components to represent original data?

Answer: 0