Last name (CAPTIALS): _____

First name (CAPITALS): _____

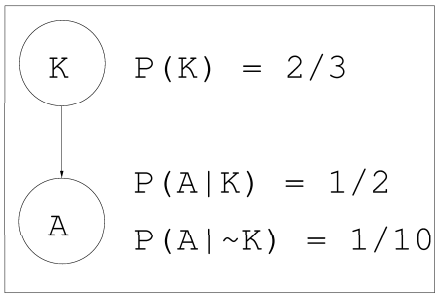
Andrew User ID (CAPITALS): (without the @andrew.cmu.edu bit): _____

15-781 Final Exam, Fall 2001

- You must answer any nine questions out of the following twelve. Each question is worth 11 points.
- You must fill out your name and your andrew userid clearly and in block capital letters on the front page. You will be awarded 1 point for doing this correctly.
- If you answer more than 9 questions, your best 9 scores will be used to derive your total.
- Unless the question asks for explanation, no explanation is required for any answer. But you are welcome to provide explanation if you wish.

1 Bayes Nets Inference

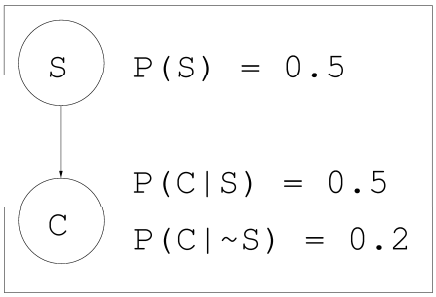
(a) **Kangaroos.**



Half of all kangaroos in the zoo are angry, and 2/3 of the zoo is comprised of kangaroos. Only 1 in 10 of the other animals are angry. What's the probability that a randomly-chosen animal is an angry kangaroo?

`P(A ^ K) = P(K) P(A|K) = 2/3 * 1/2 = 1/3`

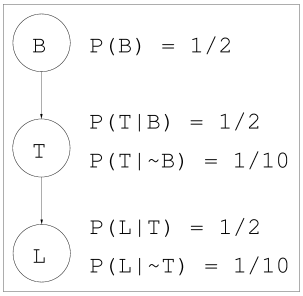
(b) **Stupidity.**



Half of all people are stupid. If you're stupid then you're more likely to be confused. A randomly-chosen person is confused. What's the chance they're stupid?

`P(S|C) = P(S^C) / [P(S^C) + P(~S ^ C)] = 1/4 / (1/4 + 1/10) = 5/7`

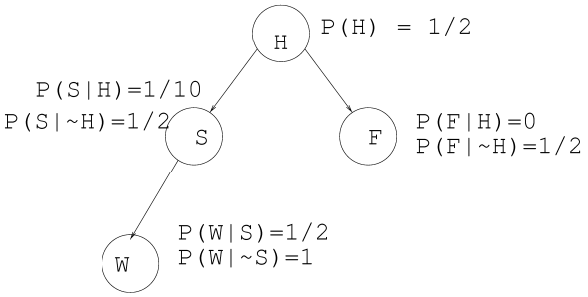
(c) **Potatoes.**



Half of all potatoes are big. A big potato is more likely to be tall. A tall potato is more likely to be lovable. What's the probability that a big lovable potato is tall?

`P(T|B^L) = P(T^B^L) / [P(T^B^L + P(~T^B^L)] = 1/8 / (1/8 + 1/40) = 5/6`

(d) **Final part.**



What's $P(W \wedge F)$?

```
P(W^F) = P(W^F^H) + P(W^F^~H)   but  P(W^F^H) = 0

P(W^F^~H) = P(F|~H)P(W^~H) =
            1/2 * (P(W ^ S ^ ~H) + P(W ^ ~S ^ ~H)) = 1/2 * ( 1/8 + 1/4) = 3/16
```

2 Bayes Nets and HMMs

(a) Let nbs(m) = the number of possible Bayes Network graph structures using m attributes. (Note that two networks with the same structure but different probabilities in their tables do not count as different structures). Which of the following statements is true?

- (i) $\text{nbs}(m) < m$
- (ii) $m \leq \text{nbs}(m) < \frac{m(m-1)}{2}$
- (iii) $\frac{m(m-1)}{2} \leq \text{nbs}(m) < 2^m$
- (iv) $2^m \leq \text{nbs}(m) < 2^{\frac{m(m-1)}{2}}$
- (v) $2^{\frac{m(m-1)}{2}} \leq \text{nbs}(m)$

```
Answer is (v) because the number of undirected graphs with n vertices
is 2^[m choose 2], and there are even more acyclic directed graphs
```

(b) Remember that $I < X, Y, Z >$ means

X is conditionally independent of Z given Y

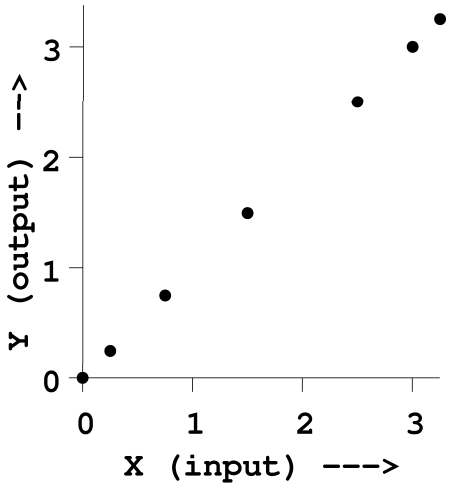
Assuming the conventional assumptions and notation of Hidden Markov Models, in which q_t denotes the hidden state at time t and O_t denotes the observation at time t, which of the following are true of all HMMs? Write "True" or "False" next to each statement.

- (i) $I < q_{t+1}, q_t, q_{t-1} >$
- (ii) $I < q_{t+2}, q_t, q_{t-1} >$
- (iii) $I < q_{t+1}, q_t, q_{t-2} >$
- (iv) $I < O_{t+1}, O_t, O_{t-1} >$
- (v) $I < O_{t+2}, O_t, O_{t-1} >$
- (vi) $I < O_{t+1}, O_t, O_{t-2} >$

```
(i) (ii) (iii) all TRUE
(iv) (v) (vi) all FALSE
```

3 Regression

(a) Consider the following data with one input and one output.



- (i) What is the mean squared training set error of running linear regression on this data (using the model $y = w_0 + w_1 x$)?

  0

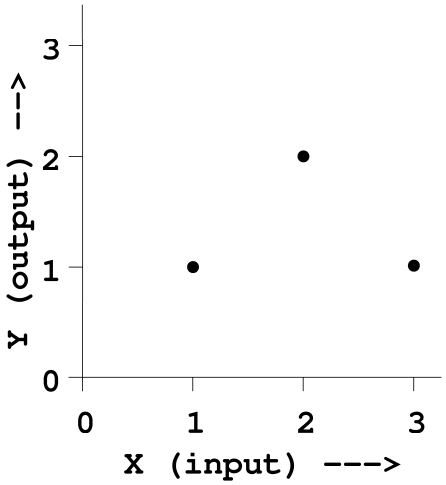
- (ii) What is the mean squared test set error of running linear regression on this data, assuming the rightmost three points are in the test set, and the others are in the training set.

  0

- (iii) What is the mean squared leave-one-out cross-validation (LOOCV) error of running linear regression on this data?

0

(b) Consider the following data with one input and one output.



X	Y
1	1
2	2
3	1

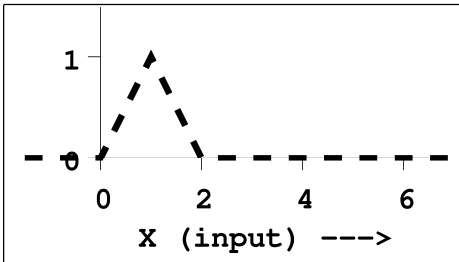
- (i) What is the mean squared training set error of running linear regression on this data (using the model $y = w_0 + w_1 x$)? (Hint: by symmetry it is clear that the best fit to the three datapoints is a horizontal line).

```
SSE = (1/3)^2 + (2/3)^2 + (1/3)^2 = 6/9
MSE = SSE/3 = 2/9
```

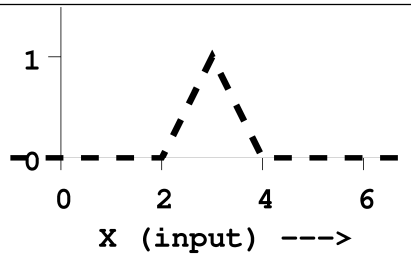
- (ii) What is the mean squared leave-one-out cross-validation (LOOCV) error of running linear regression on this data?

```
1/3 * (2^2 + 1^2 + 2^2) = 9/3 = 3
```

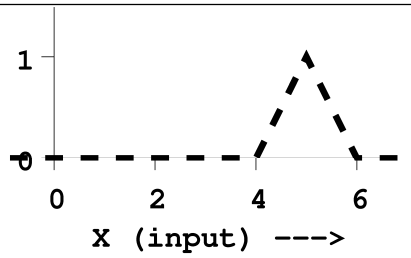
(c) Suppose we plan to do regression with the following basis functions:



$$\phi_1(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } 0 \le x < 1 \\ 2 - x & \text{if } 1 \le x < 2 \\ 0 & \text{if } 2 \le x \end{cases}$$



$$\phi_2(x) = \begin{cases} 0 & \text{if } x < 2 \\ x - 2 & \text{if } 2 \le x < 3 \\ 4 - x & \text{if } 3 \le x < 4 \\ 0 & \text{if } 4 \le x \end{cases}$$



$$\phi_3(x) = \begin{cases} 0 & \text{if } x < 4 \\ x - 4 & \text{if } 4 \le x < 5 \\ 6 - x & \text{if } 5 \le x < 6 \\ 0 & \text{if } 6 \le x \end{cases}$$

Our regression will be $y = \beta_1\phi_1(x) + \beta_2\phi_2(x) + \beta_3\phi_3(x)$.

Assume all our datapoints and future queries have $1 \le x \le 5$. Is this a generally useful set of basis functions to use? If "yes", then explain their prime advantage. If "no", explain their biggest drawback.

NO

They're forced to predict y=0 at x=2 and x=4 (and forced to be close to zero nearby) no matter what the values of beta.

4 Regression Trees

Regression trees are a kind of decision tree used for learning from data with a real-valued output instead of a categorical output. They were discussed in the "Eight favorite regression algorithms" lecture.

On the next page you will see pseudocode for building a regression tree in the special case where all the input attributes are boolean (they can have values 0 or 1).

The MakeTree function takes two arguments:

- D, a set of datapoints
- and A, a set of input attributes.

It then makes the best regression tree it can using only the datapoints and attributes passed to it. It is a recursive procedure. The full algorithm is run by calling MakeTree with D containing every record and A containing every attribute. *Note that this code does no pruning, and that it assumes that all input attributes are binary-valued.*

Now read the code on the next page, after which question (a) will ask you about bugs in the code.

```
MakeTree(D,A)
Returns a Regression Tree

1.     For each attribute a in the set A do...
1.1        Let D0 = { (xk,yk) in D such that xk[a] = 0 }
           // Comment: xk[a] denotes the value of attribute a in record xk
1.2        Let D1 = { (xk,yk) in D such that xk[a] = 1 }
           // Comment: Note that D0 union D1 == D
           // Comment: Note too that D0 intersection D1 == empty
1.3        mu0 = mean value of yk among records in D0
1.4        mu1 = mean value of yk among records in D1
1.5        SSE0 = sum over all records in D0 of (yk - mu0) squared
1.6        SSE1 = sum over all records in D1 of (yk - mu0) squared
1.7        Let Score[a] = SSE0 + SSE1

2.     // Once a score has been computed for each attribute, let...
       a* = argmax  Score[a]
              a

3.     Let D0 = { (xk,yk) in D such that xk[a*] = 0 }
4.     Let D1 = { (xk,yk) in D such that xk[a*] = 1 }
3.     Let LeftChild = MakeTree(D0,A - {a*})
       // Comment: A - {a*} means the set containing all elements of A except for a*
4.     Let RightChild = MakeTree(D1,A - {a*})
5.     Return a tree whose root tests the value of a*, and whose ``a* = 0''
       branch is LeftChild and whose ``a* = 1'' branch is RightChild.
```

(a) Beyond the obvious problem that there is no pruning, there are three bugs in the above code. They are all very distinct. One of them is at the level of a typographical error. The other two are more serious errors in logic. Identify the three bugs (remembering that the lack of pruning is not one of the three bugs), explaining why each one is a bug. It is not necessary to explain how to fix any bug, though you are welcome to do so if that's the easiest way to explain the bug.

Line 1.6 should use (yk - mu1)^2

Line 2 should use argmin

The algorithm is missing the base case of the recursion

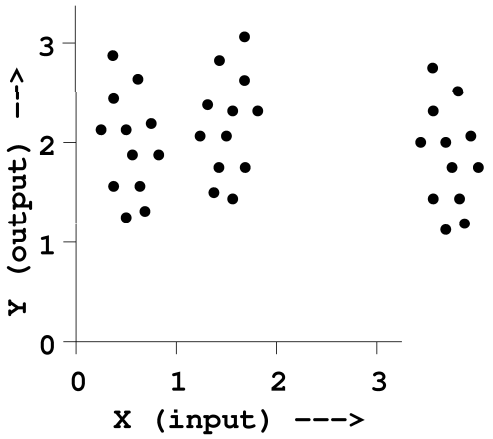
(b) Why, in the recursive calls to MakeTree, is the second argument "$A - \{a*\}$" instead of simply "A"?

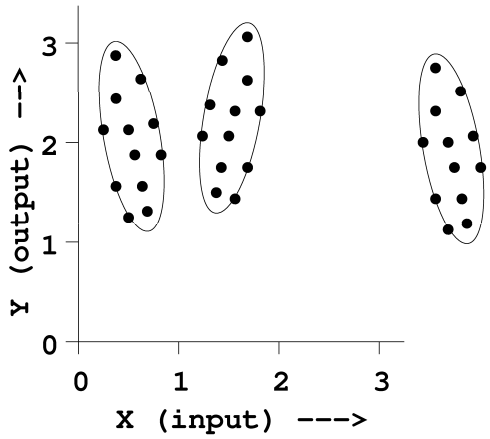
Because a* can't possibly be chosen in any recursive calls

5 Clustering

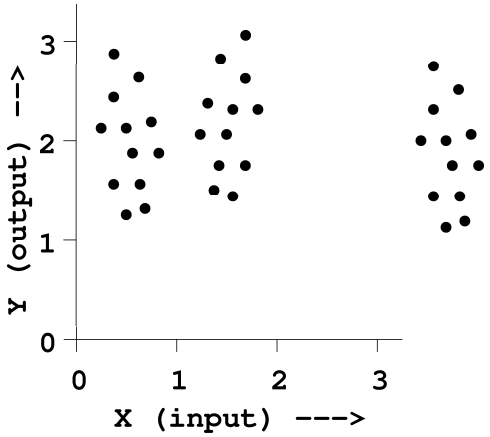
In the left of the following two pictures I show a dataset. In the right figure I sketch the globally maximally likely mixture of three Gaussians for the given data.

- Assume we have protective code in place that prevents any degenerate solutions in which some Gaussian grows infinitesimally small.
- And assume a GMM model in which all parameters (class probabilities, class centroids and class covariances) can be varied.

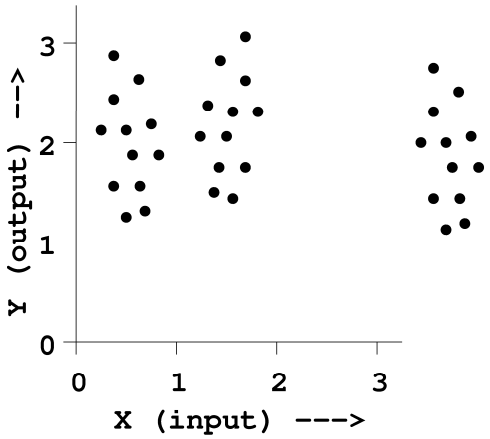




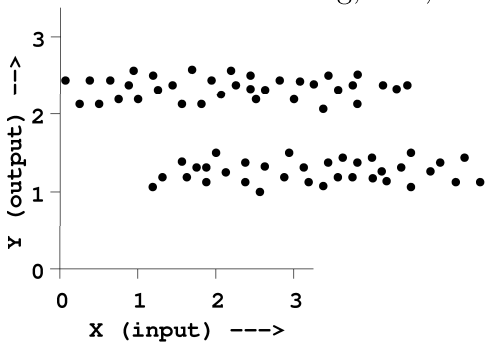
(a) Using the same notation and the same assumptions, sketch the globally maximally likely mixture of **two** Gaussians.



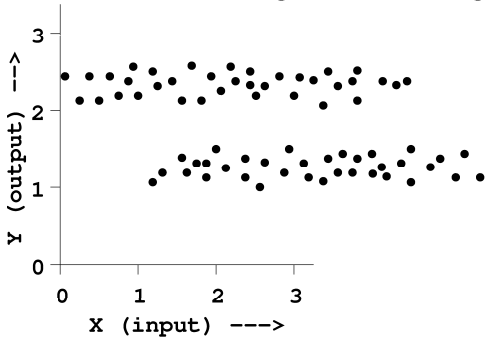
(b) Using the same notation and the same assumptions, sketch a mixture of three distinct Gaussians that is stuck in a suboptimal configuration (i.e. in which infinitely many more iterations of the EM algorithm would remain in essentially the same suboptimal configuration). *(You must not give an answer in which two or more Gaussians all have the same mean vectors—we are looking for an answer in which all the Gaussians have distinct mean vectors).*



(c) Using the same notation and the same assumptions, sketch the globally maximally likely mixture of two Gaussians in the following, new, dataset.



(d) Now, suppose we ran k-means with $k = 2$ on this dataset. Show the rough locations of the centers of the two clusters in the configuration with globally minimal distortion.



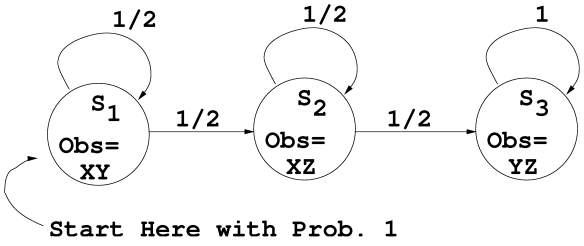
6 Regression algorithms

For each empty box in the following table, write in "Y" if the statement at the top of the column applies to the regression algorithm. Write "N" if the statement does not apply.

	No matter what the training data is, the predicted output is guaranteed to be a continuous function of the input. (i.e. there are no discontinuities in the prediction). If a predictor gives continuous but undifferentiable predictions then you should answer "Y".	The cost of training on a dataset with R records is at least $O(R^2)$: quadratic (or worse) in R. For iterative algorithms marked with (*) simply consider the cost of one iteration of the algorithm through the data.
Linear Regression	Y	N
Quadratic Regression	Y	N
Perceptrons with sigmoid activation functions (*)	Y	N
1-hidden-layer Neural Nets with sigmoid activation functions (*)	Y	N
1-nearest neighbor	N	N
10-nearest neighbor	N	N
Kernel Regression	Y	N
Locally Weighted Regression	Y	N
Radial Basis Function Regression with 100 Gaussian basis functions	Y	N
Regression Trees	N	N
Cascade correlation (with sigmoid activation functions)	Y	N
Multilinear interpolation	Y	N
MARS	Y	

7 Hidden Markov Models

Warning: this is a question that will take a few minutes if you really understand HMMs, but could take hours if you don't. Assume we are working with this HMM



$a_{11} = 1/2$	$a_{12} = 1/2$	$a_{13} = 0$	$b_1(X) = 1/2$	$b_1(Y) = 1/2$	$b_1(Z) = 0$	$\pi_1 = 1$
$a_{21} = 0$	$a_{22} = 1/2$	$a_{23} = 1/2$	$b_2(X) = 1/2$	$b_2(Y) = 0$	$b_2(Z) = 1/2$	$\pi_2 = 0$
$a_{31} = 0$	$a_{32} = 0$	$a_{33} = 1$	$b_3(X) = 0$	$b_3(Y) = 1/2$	$b_3(Z) = 1/2$	$\pi_3 = 0$

Where

$$\begin{aligned} a_{ij} &= P(q_{t+1} = S_j | q_t = S_i) \\ b_i(k) &= P(O_t = k | q_t = S_i) \end{aligned}$$

Suppose we have observed this sequence

XZXYYZYZZ

(in long-hand: $O_1 = X, O_2 = Z, O_3 = X, O_4 = Y, O_5 = Y, O_6 = Z, O_7 = Y, O_8 = Z, O_9 = Z$). Fill in this table with $\alpha_t(i)$ values, remembering the definition:

$$\alpha_i(t) = P(O_1 \wedge O_2 \wedge ...O_t \wedge q_t = s_i)$$

So for example,

$$\alpha_3(2) = P(O_1 = X \wedge O_2 = Z \wedge O_3 = X \wedge q_3 = S_2)$$

t	$\alpha_t(1)$	$\alpha_t(2)$	$\alpha_t(3)$
1	1/2		
2		1/8	
3		1/32	
4			1/128
5			1/256
6			1/512
7			1/1024
8			1/2048
9			1/4096

8 Locally Weighted Regression

Here's an argument made by a misguided practitioner of Locally Weighted Regression.

> Suppose you have a dataset with R_1 training points and another dataset with R_2 test points. You must predict the output for each of the test points. If you use a kernel function that decays to zero beyond a certain Kernel width then Locally Weighted Regression is computationally cheaper than regular linear regression. This is because with locally weighted regression you must do the following for each query point in the test set,
>
> - Find all the points that have non-zero weight for this particular query.
> - Do a linear regression with them (after having weighted their contribution to the regression appropriately).
> - Predict the value of the query.
>
> whereas with regular linear regression you must do the following for each query point:
>
> - take all the training set datapoints.
> - Do an unweighted linear regression with them.
> - Predict the value of the query.
>
> The locally weighted regression frequently finds itself doing regression on only a tiny fraction of the datapoints because most have zero weight. So most of the local method's queries are cheap to answer. In contrast, regular regression must use every single training point in every single prediction and so does at least as much work, and usually more.

This argument has a serious error. Even if it is true that the kernel function causes almost all points to have zero weight for each LWR query the argument is wrong. What is the error?

```
Linear regression only needs to learn its weights (i.e. do the
appropriate matrix inversion) once in total. LWR must do
a separate matrix inversion for each test point.
```

9 Nearest neighbor and cross-validation

At some point during this question you may find it useful to use the fact that if U and V are two independent real-valued random variables then $Var[aU + bV] = a^2\, Var[U] + b^2\, Var[V]$.

Suppose you have 10,000 datapoints $\{(x_k, y_k) : k = 1, 2, ..., 10000\}$. Your dataset has one input and one output. The kth datapoint is generated by the following recipe:

$$\begin{aligned} x_k &= k/10000 \\ y_k &\sim N(0, 2^2) \end{aligned}$$

So that y_k is all noise: drawn from a Gaussian with mean 0 and variance $\sigma^2 = 4$ (and standard deviation $\sigma = 2$). Note that its value is independent of all the other y values. You are considering two learning algorithms:

- **Algorithm NN:** 1-nearest neighbor.
- **Algorithm Zero:** Always predict zero.

(a) What is the expected Mean Squared Training Error for **Algorithm NN**?

`0`

(b) What is the expected Mean Squared Training Error for **Algorithm Zero**?

`4`

(c) What is the expected Mean Squared Leave-one-out Cross-validation Error for **Algorithm NN**?

`8 = E[(xk - x[x+1])^2]`

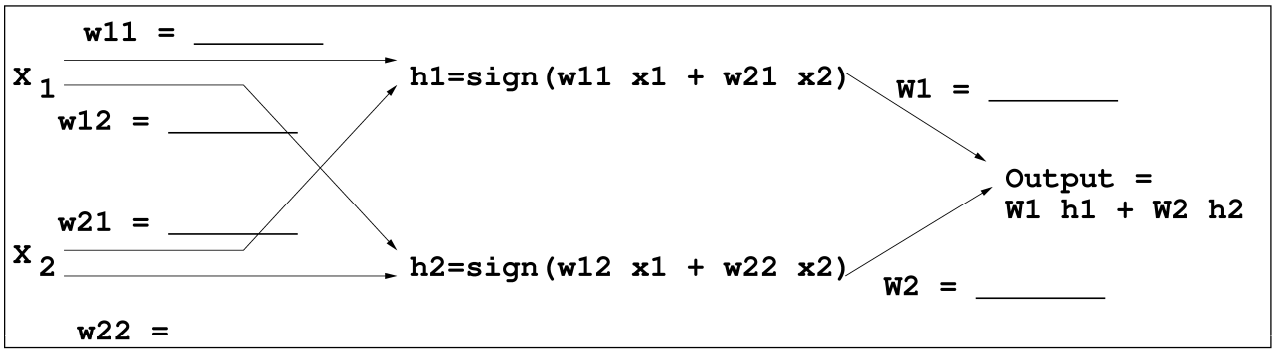
(d) What is the expected Mean Squared Leave-one-out Cross-validation Error for **Algorithm Zero**?

4

10 Neural Nets

(a) Suppose we are learning a 1-hidden-layer neural net with a sign-function activation

$$\text{Sign}(z) = \quad 1 \quad \text{if } z \geq 0$$
$$\text{Sign}(z) = \quad -1 \quad \text{if } z < 0$$



We give it this training set, which represents the exclusive-or function if you interpret -1 as false and +1 as true:

X_1	X_2	Y
1	1	-1
1	-1	1
-1	1	1
-1	-1	-1

On the diagram above you must write in six numbers: a set of weights that would give zero training error. (Note that constant terms are not being used anywhere, and note too that the output does not need to go through a sign function). Or..if it impossible to find a satisfactory set of weights, just write "impossible".

`Impossible`

(b) You have a dataset with one real-valued input x and one real-valued output y in which you believe

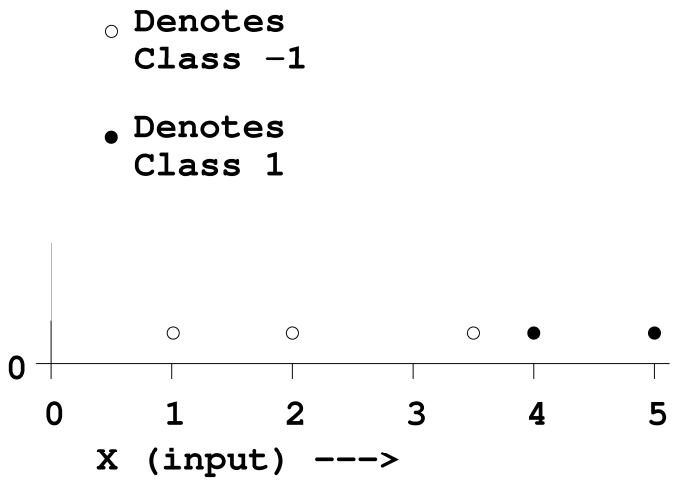
$$y_k = \exp(wx_k) + \epsilon_k$$

where (x_k, y_k) is the kth datapoint and ϵ_k is Gaussian noise. This is thus a neural net with just one weight: w.

Give the update equation for a gradient descent approach to finding the value of a that minimizes the mean squared error.

11 Support Vector Machines

Consider the following dataset. We are going to learn a linear SVM from it of the form $f(x) = \text{sign}(wx + b)$.



(a) What values for w and b will be learned by the linear SVM?

`w = 4, b = -15`

(b) What is the training set error of the above example? (expressed as the percentage of training points misclassified)

`0`

(c) What is the leave-one-out cross-validation error of the above example? (expressed as the percentage of left-out points misclassified)

`2 wrong => 40%`

(d) **True or False:** Even with the clever SVM Kernel trick it is impossibly computationally expensive, even on a supercomputer, to do the following:

Given a dataset with 200 datapoints and 50 attributes learn an SVM classifier with full 20th-degree-polynomial basis functions and then apply what you've learned to predict the classes of 1000 test datapoints.

`FALSE`

12 VC Dimension

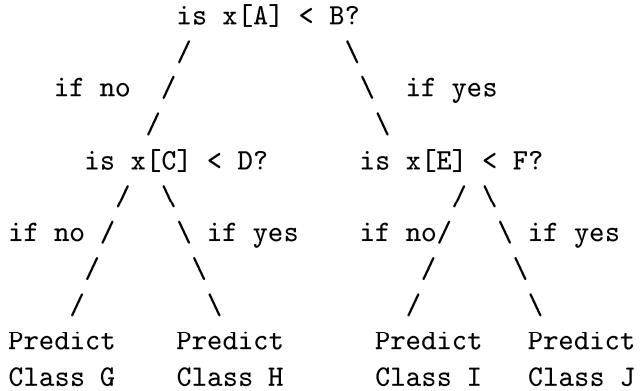
(a) Suppose we have one input variable x and one output variable y. We are using the machine $f_1(x, \alpha) = \text{sign}(x + \alpha)$. What is the VC dimension of f_1?

1

(b) Suppose we have one input variable x and one output variable y. We are using the machine $f_2(x, \alpha) = \text{sign}(\alpha x + 1)$. What is the VC dimension of f_2?

1

(c) Now assume our inputs are m-dimensional and we use the following two-level, two-choice decision tree to make our classification:



Where the machine has 10 parameters

$$\begin{aligned} A &\in \{1, 2, ..., m\} \\ B &\in \Re \\ C &\in \{1, 2, ..., m\} \\ D &\in \Re \\ E &\in \{1, 2, ..., m\} \\ F &\in \Re \\ G &\in \{-1, 1\} \\ H &\in \{-1, 1\} \\ I &\in \{-1, 1\} \\ J &\in \{-1, 1\} \end{aligned}$$

What is the VC-dimension of this machine?

4

15-781 Final Exam, Fall 2002

1. Write your name and your andrew email address below.

   Name: Andrew Moore

   Andrew ID:

2. There should be 17 pages in this exam (excluding this cover sheet).

3. If you need more room to work out your answer to a question, use the back of the page and clearly mark on the front of the page if we are to look at what's on the back.

4. You should attempt to answer all of the questions.

5. You may use any and all notes, as well as the class textbook.

6. All questions are worth an equal amount. They are not all equally difficult.

7. You have 3 hours.

8. Good luck!

1 Computational Learning Theory

1.1 PAC learning for Decison Lists

A decision list is a list of if-then rules where each condition is a literal (a variable or its negation). It can be thought of as a decision tree with just one path. For example, say that I like to go for a walk if it's warm or if it's snowing and I have a jacket, as long as it's not raining. We could describe this as the following decision list:

```
if rainy then no
else if warm then yes
else if not(have-jacket) then no
else if snowy then yes
else no.
```

(a) Describe an algorithm to learn DLs given a data set, for example

a	b	c	class
1	0	0	+
0	1	1	-
1	1	1	+
0	0	0	-
1	1	0	+

Your algorithm should have the characteristic that it should always classify examples that it has already seen correctly (ie, it should be consistent with the data).If it's not possible to continue to produce a decision list that's consistent with the data, your algorithm should terminate and announce that it has failed.

(b) Find the size of the hypothesis space, $|H|$, for decisions lists of k attributes.

(c) Find an expression for the number of examples needed to learn a decision list of k attributes with error at most .10 with probability 90%.

(d) What if the learner is trying to learn a decision list, but the representation that it is using is a conjunction of k literals? Find the expression for the number of examples needed to learn the decision list with error at most .10 with 90% probability.

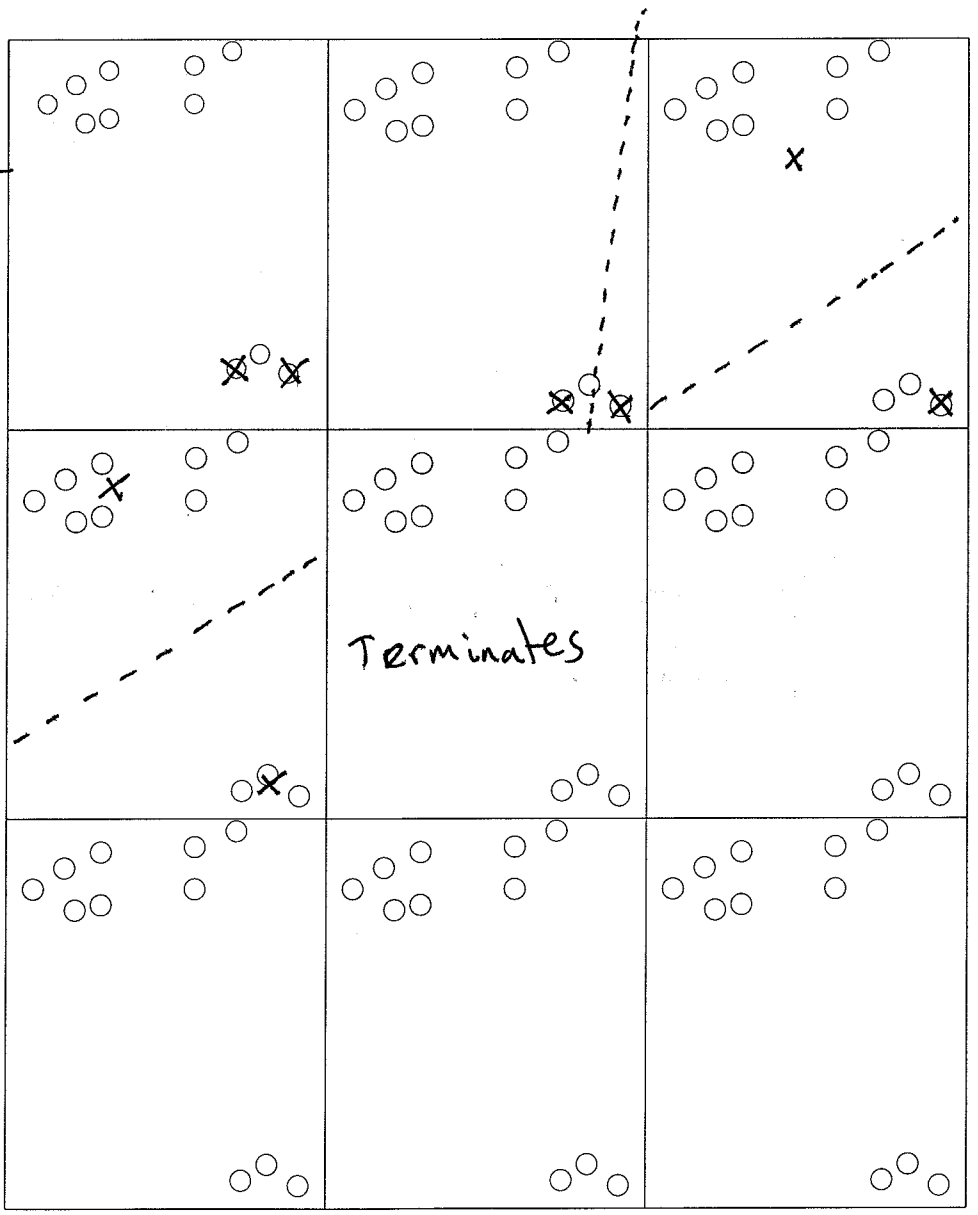
2 K-means and Gaussian Mixture Models

(a) What is the effect on the means found by k-means (as opposed to the true means) of overlapping clusters?

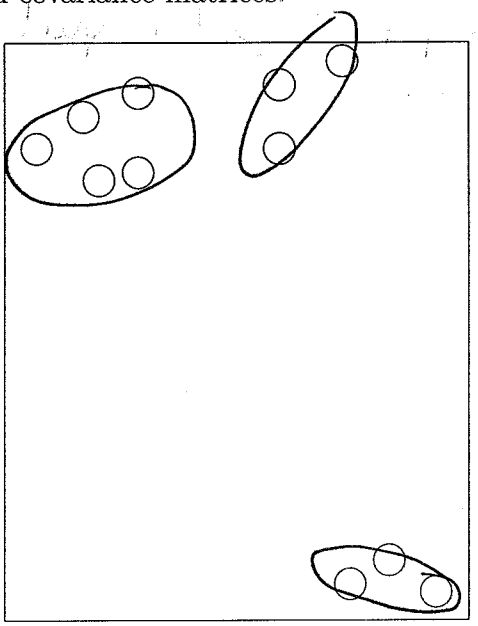
They are pushed further apart than the true means would be.

(b) Run k-means manually for the following dataset. Circles are data points and squares are the initial cluster centers. Draw the cluster centers and the decision boundaries that define each cluster. Use as many pictures as you need until convergence.
Note: Execute the algorithm such that if a mean has no points assigned to it, it stays where it is for that iteration.

I don't see any squares so I'll start somewhat arbitrary



(c) Now draw (approximately) what a Gaussian mixture model of three gaussians with the same initial centers as for the k-means problem would converge to. Assume that the model puts no restrictions on the form of the covariance matrices and that EM updates both the means and covariance matrices.



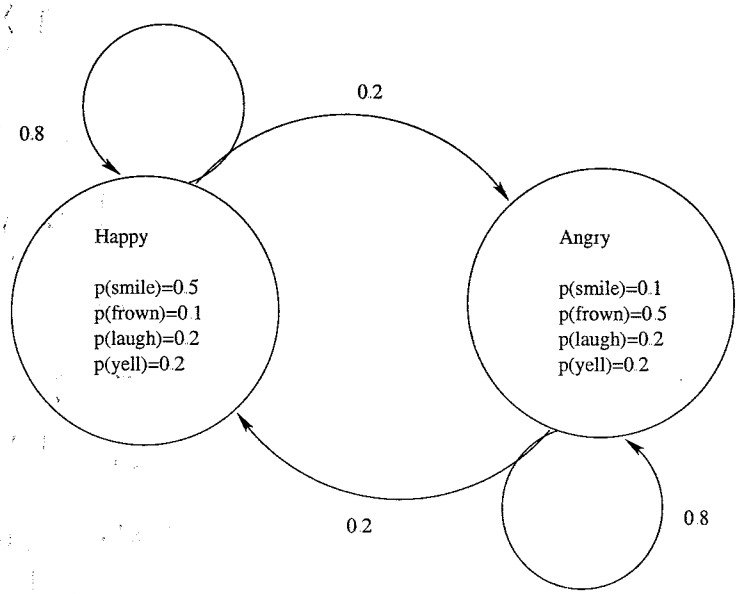
This is the result you'd get if no local optima

(d) Is the classification given by the mixture model the same as the classification given by k-means? Why or why not?

I'd answer if I knew the start locations.

3 HMMs

Andrew lives a simple life. Some days he's Angry and some days he's Happy. But he hides his emotional state, and so all you can observe is whether he smiles, frowns, laughs, or yells. We start on day 1 in the Happy state, and there's one transition per day.



Definitions:
q_t = state on day t.
O_t = observation on day t.

(a) What is $P(q_2 = Happy)$? 0.8

(b) What is $P(O_2 = frown)$? $\frac{8}{10} \times \frac{1}{10} + \frac{2}{10} \times \frac{1}{2} = \frac{8}{100} + \frac{10}{100} = \frac{18}{100}$

(c) What is $P(q_2 = Happy | O_2 = frown)$? $\frac{P(O_2 = frown \mid q_2 = H)\, P(q_2 = H)}{P(O_2 = frown)}$

$= \frac{\frac{1}{10} \times \frac{8}{10}}{\left(\frac{18}{100}\right)} = \frac{4}{9}$

(d) What is $P(O_{100} = yell)$?

$P(O_{100} = yell) = P(O_{100} = yell \mid q_{100} = H) P(q_{100} = H) + P(O_{100} = yell \mid q_{100} = A) P(q_{100} = A)$

$= \frac{2}{10} \times \left(P(q_{100} = H) + P(q_{100} = A)\right) = \frac{2}{10} \times 1 = \frac{2}{10}$

(e) Assume that $O_1 = frown$, $O_2 = frown$, $O_3 = frown$, $O_4 = frown$, and $O_5 = frown$. What is the most likely sequence of states?

HAAAA

4 Bayesian Inference

(a) Consider a dataset over 3 boolean attributes, X, Y, and Z.

Of these sets of information, which are sufficient to specify the joint distribution? Circle all that apply.

A. $P(\sim X|Z)\ P(\sim X|\sim Z)\ P(\sim Y|X\wedge Z)\ P(\sim Y|X\wedge\sim Z)$
$P(\sim Y|\sim X\wedge Z)\ P(\sim Y|\sim X\wedge\sim Z)\ P(Z)$

Yes because this implements

see Bayes Net lecture

B. $P(\sim X|\sim Z)\ P(X|\sim Z)\ P(Y|X\wedge Z)\ P(Y|X\wedge\sim Z)$
$P(Y|\sim X\wedge Z)\ P(Y|\sim X\wedge\sim Z)\ P(Z)$

No because can't deduce $P(X|Z)$

C. $P(X|Z)\ P(X|\sim Z)\ P(Y|X\wedge Z)\ P(Y|X\wedge\sim Z)$
$P(Y|\sim X\wedge Z)\ P(\sim Y|\sim X\wedge\sim Z)\ P(\sim Z)$

Yes, like A (and can get $P(\sim Z)$ from $P(Z)$)

D. $P(X|Z)\ P(X|\sim Z)\ P(Y|X\wedge Z)\ P(Y|X\wedge\sim Z)$
$P(\sim Y|\sim X\wedge\sim Z)\ P(Y|\sim X\wedge\sim Z)\ P(Z)$

No, can't get $P(Y|\sim X\wedge Z)$

Given this dataset of 16 records:

A	B	C
0	0	1
0	0	1
0	0	1
0	1	0
0	1	1
0	1	1
0	1	1
0	1	1
1	0	0
1	0	0
1	0	0
1	0	0
1	0	0
1	1	0
1	1	0
1	1	1

(b) Write down the probabilities needed to make a joint density bayes classifier

$P(A \wedge B \mid C)$ $\quad P(A \wedge B \mid \sim C)$

$P(A \wedge \sim B \mid C)$ $\quad P(A \wedge \sim B \mid \sim C)$ $\quad P(C) = \frac{1}{2}$

$\frac{1}{2} =$ $P(\sim A \wedge B \mid C)$ $\quad P(\sim A \wedge B \mid \sim C) = \frac{1}{8}$

$P(\sim A \wedge \sim B \mid C)$ $\quad P(\sim A \wedge \sim B \mid \sim C)$ ← Actually redundant

(c) Write down the probabilities needed to make a naive bayes classifier.

$P(A \mid C) = \frac{7}{8}$ $\quad P(C) = \frac{1}{2}$ $\quad P(A \mid \sim C) = \frac{7}{8}$

$P(B \mid C) = \frac{5}{8}$ $\quad P(B \mid \sim C) = \frac{3}{8}$

(d) Write the classification that the joint density bayes classifier would make for C given A=0,B=1.

$$P(C \mid \sim A \wedge B) = \frac{P(\sim A \wedge B \mid C) P(C)}{P(\sim A \wedge B \mid C) P(C) + P(\sim A \wedge B \mid \sim C) P(\sim C)}$$

$$= \frac{\frac{1}{2} \times \frac{1}{2}}{\frac{1}{2} \times \frac{1}{2} + \frac{1}{8} \times \frac{1}{2}} = \frac{\frac{1}{2}}{\frac{1}{2} + \frac{1}{8}} = \frac{(4/8)}{(5/8)} = 4/5$$

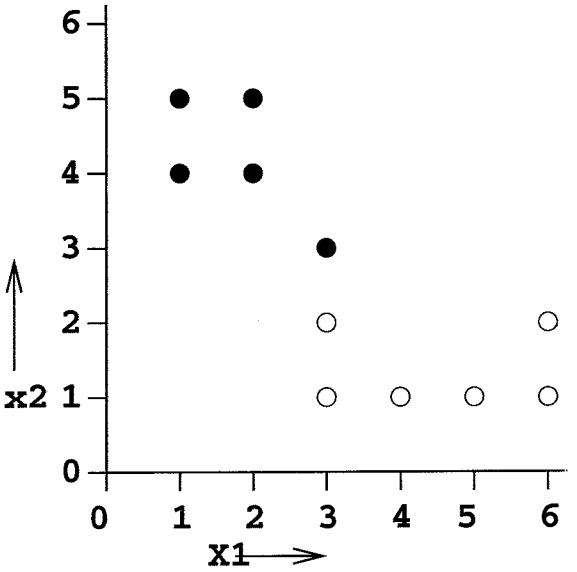
(e) Write the classification that the naive bayes classifier would make for C given A=0,B=1.

$$P(C \mid \sim A \wedge B) = \frac{P(\sim A \wedge B \mid C) P(C)}{P(\sim A \wedge B \mid C) P(C) + P(\sim A \wedge B \mid \sim C) P(\sim C)} = \frac{P(\sim A \mid C) P(B \mid C) P(C)}{\left[P(\sim A \mid C) P(B \mid C) P(C) + P(\sim A \mid \sim C) P(B \mid \sim C) P(\sim C) \right]}$$

$$= \frac{7}{8} \frac{5}{8} \frac{1}{2} \Big/ \left[\frac{7}{8} \frac{5}{8} \frac{1}{2} + \frac{1}{8} \frac{3}{8} \frac{1}{2} \right] = \frac{35}{35+3} = \frac{35}{38}$$

5 Support Vector Machines

This picture shows a dataset with two real-valued inputs (x_1 and x_2) and one categorical output class. The positive points are shown as solid dots and the negative points are small circles.



(a) Suppose you are using a linear SVM with no provision for noise (i.e. a Linear SVM that is trying to maximize its margin while ensuring all datapoints are on their correct sides of the margin). Draw three lines on the above diagram, showing the classification boundary and the two sides of the margin. Circle the support vector(s).

(b) Using the familiar LSVM classifier notation of class = sign($\mathbf{w}.\mathbf{x} + b$), calculate the values of $\mathbf{w}$ and b learned for part (a).

(c) Assume you are using a noise-tolerant LSVM which tries to minimize

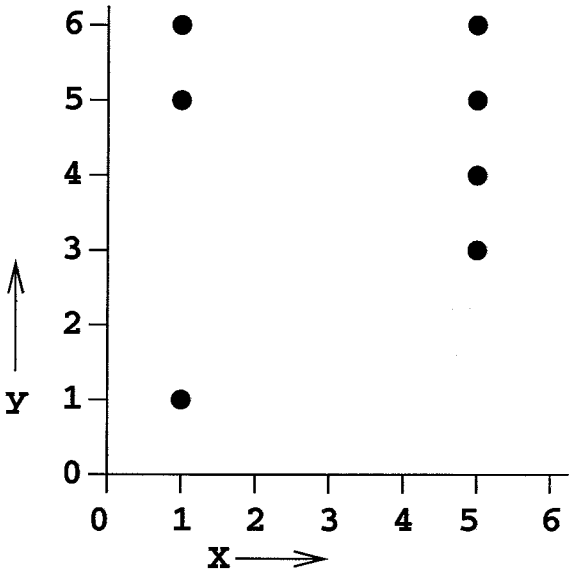
$$\frac{1}{2}\mathbf{w}.\mathbf{w} + C\sum_{k=1}^{R}\epsilon_k \tag{1}$$

using the notation of your notes and the Burges paper.

Question: is it possible to invent a dataset and a positive value of C in which (a) the dataset is linearly separable but (b) the LSVM would nevertheless misclassify at least one training point? If it is possible to invent such an example, please sketch the example and suggest a value for C. If it is not possible, explain why not.

6 Instance-based learning

This picture shows a dataset with one real-valued input x and one real-valued output y. There are seven training points.



Suppose you are training using kernel regression using some unspecified kernel function. The only thing you know about the kernel function is that it is a monotonically decreasing function of distance that decays to zero at a distance of 3 units (and is strictly greater than zero at a distance of less than 3 units).

(a) What is the predicted value of y when $x = 1$?

mean $(6, 5, 1) = \frac{12}{3} = 4$ (rightmost points will be ignored)

(b) What is the predicted value of y when $x = 3$?

Since same distance for all seven datapoints, all points weighted equally in weighted average $\frac{(1+5+6+3+4+5+6)}{7}$

$= 30/7$

(c) What is the predicted value of y when $x = 5$?

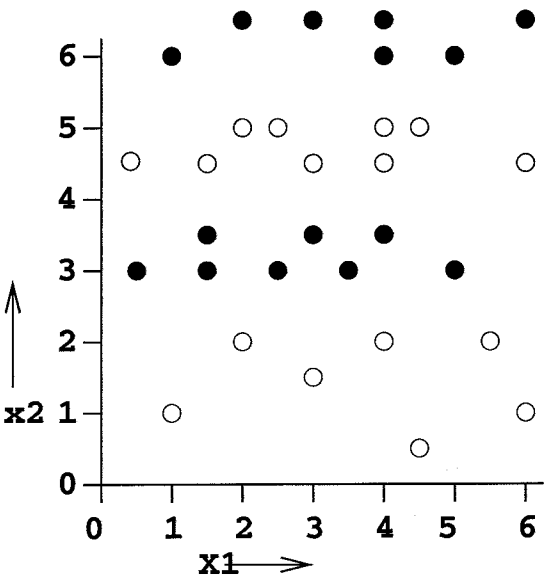
$\frac{3+4+5+6}{4} = 4\frac{1}{2}$ (ignores left)

(d) What is the predicted value of y when $x = 6$?

$4\frac{1}{2}$ (ignores left)

The final two parts of this question concern 1-nearest neighbor used as a classifier.

The following dataset has two real valued inputs and one binary categorical output. The class is denoted by the color of the datapoint.



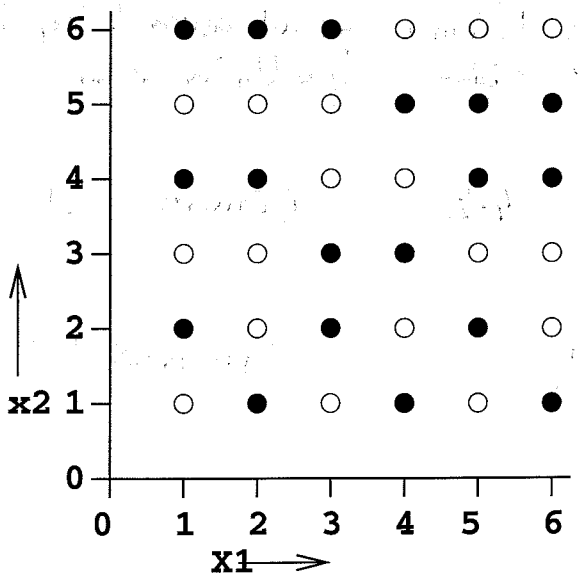
(e) Does there exist a choice of Euclidian distance metric for which 1-nearest-neighbor would achieve zero training set error on the above dataset?

Yes, eg. $\text{Distance}(a,b) = 0\times(a.x1-b.x1)^2 + 1\times(a.x2-b.x2)^2$

(i.e. ignore x_1)

But in fact, it's a trick question: ANY METRIC will get 0 training error

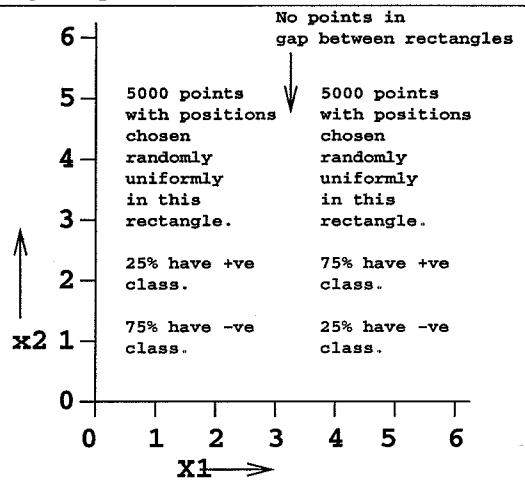
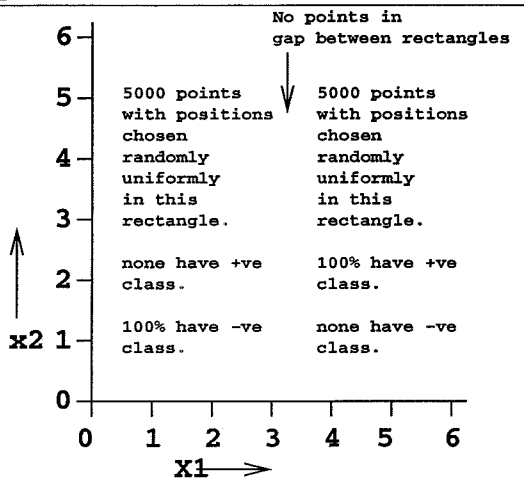
Now let's consider a different dataset:



(f) Does there exist a choice of Euclidian distance metric for which 1-nearest-neighbor would achieve zero training set error on the above dataset?

Yes, any metric

7 Nearest Neighbor and Cross-Validation

Recipe for making training set of 10,000 datapoints with two real-valued inputs and one binary output class:	Recipe for making test set of 10,000 datapoints with two real-valued inputs and one binary output class:
	

Using the above recipes for making training and test sets you will see that the training set is noisy: in either region, 25% of the data comes from the minority class. The test set is noise-free.

In each of the following questions, circle the answer that most closely defines the expected error rate, expressed as a fraction.

(a) What is the expected training set error using one-nearest-neighbor?

(0) 1/8 1/4 3/8 1/3 1/2 5/8 2/3 3/4 7/8 1

(b) What is the expected leave-one-out cross-validation error on the training set using one-nearest-neighbor? $\frac{1}{4} \times \frac{3}{4} + \frac{3}{4} \times \frac{1}{4} = \frac{3}{8}$

0 1/8 1/4 (3/8) 1/3 1/2 5/8 2/3 3/4 7/8 1

(c) What is the expected test set error if we train on the training set, test on the test set, and use one-nearest-neighbor?

0 1/8 1/4 (3/8) 1/3 1/2 5/8 2/3 3/4 7/8 1

(d) What is the expected training set error using 21-nearest-neighbor?

0 1/8 (1/4) 3/8 1/3 1/2 5/8 2/3 3/4 7/8 1

(e) What is the expected leave-one-out cross-validation error on the training set using 21-nearest-neighbor?

0 1/8 (1/4) 3/8 1/3 1/2 5/8 2/3 3/4 7/8 1

(f) What is the expected test set error if we train on the training set, test on the test set, and use 21-nearest-neighbor?

0 1/8 (1/4) 3/8 1/3 1/2 5/8 2/3 3/4 7/8 1

8 Learning Bayes Net Structure

For each of the following training sets, draw the structure and CPTs that a Bayes Net Structure learner should learn, assuming that it tries to account for all the dependencies in the data as well as possible while minimizing the number of unnecessary links. In each case, your Bayes Net will have three nodes, called A B and C. Some or all of these questions have multiple correct answers...you need only supply one answer to each question.

(a)

A	B	C
0	0	0
0	0	0
0	0	1
0	0	1
0	1	0
0	1	0
0	1	1
0	1	1
1	0	0
1	0	0
1	0	1
1	0	1
1	1	0
1	1	0
1	1	1
1	1	1

(b)

A	B	C
0	0	0
0	0	0
0	0	0
0	1	1
0	1	1
0	1	1
1	1	0
1	1	0
1	1	0
1	0	1
1	0	1
1	0	1

(c)

A	B	C
0	0	0
0	0	0
0	0	0
1	0	1
1	0	1
1	0	1
1	1	0
1	1	0
1	1	0

9 Markov Decision Processes

Consider the following MDP, assuming a discount factor of $\gamma = 0.5$. Note that the action "Party" carries an immediate reward of +10. The action "Study" unfortunately carries no immediate reward, except during the senior year, when a reward of +100 is provided upon transition to the terminal state "Employed".

(a) What is the probability that a freshman will fail to graduate to the "Employed" state within four years, even if they study at every opportunity?

(b) Draw the diagram for the Markov Process (not the MDP, the MP) that corresponds to the policy "study whenever possible."

(c) What is the value associated with the state "Junior" under the "study whenever possible" policy?

(d) Exactly how rewarding would parties have to be during junior year in order to make it advisable for a junior to party rather than study (assuming, of course, that they wish to optimize their cumulative discounted reward)?

(e) Answer the following true or false. If true, give a one-sentence argument. If false, give a counterexample.

- **(True or False?)** If partying during junior year an optimal action when it is assigned reward r, then it will also be an optimal action for a freshman when assigned reward r.

- **(True or False?)** If partying during junior year is an optimal action when it is assigned reward r, then it will also be an optimal action for a freshman when assigned reward r.

10 Q Learning

Consider the robot grid world shown below, in which actions have deterministic outcomes, and for which the discount factor $\gamma = 0.5$. The robot receives zero immediate reward upon executing its actions, except for the few actions where an immediate reward has been written in on the diagram. Note the state in the upper corner allows an action in which the robot remains in that same state for one time tick.

IMPORTANT: Notice the immediate reward for the state-action pair $< C, South >$ is -100, not +100.

(a) Write in the Q value for each state-action pair, by writing it next to the corresponding arrow.

(b) Write in the $V^*(s)$ value for each state, by writing its value inside the grid cell representing that state.

(c) Write down an equation that relates the $V^*(s)$ for an arbitrary state s to the $Q(s, a)$ values associated with the same state.

(d) Describe one optimal policy, by circling only the actions recommended by this policy

(e) Hand execute the deterministic Q learning algorithm, assuming the robot follows the trajectory shown below. Show the sequence of Q estimates (describe which entry in the Q table is being updated at each step):

state	action	next-state	immediate-reward	updated-Q-estimates
A	East	B	0	
B	East	C	10	
C	Loop	C	0	
C	South	F	-100	
F	West	E	0	
E	North	B	0	
B	East	C	10	

(f) Propose a change to the immediate reward function that results in a change to the Q function, but not to the V function.

11 Short Questions

(a) Describe the difference between a *maximum likelihood* hypothesis and a *maximum a posteriori* hypothesis.

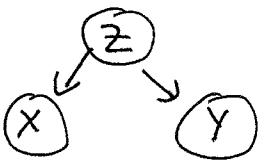
MLE = maximize $P(\text{data} \mid \text{parameters})$ by searching over parameters

MAP = Maximize $P(\text{parameters} \mid \text{data})$ by searching over params, and accounting for prior over params

(b) Consider a learning problem defined over a set of instances X. Assume the space of possible hypotheses, H, consists of all possible disjunctions over instances in X. I.e., the hypothesis $x_1 \vee x_6$ labels these two instances positive, and no others. What is the VC dimension of H?

(c) Consider a naive Bayes classifier with 2 boolean input variables, X and Y, and one boolean output, Z.

- Draw the equivalent Bayesian network.



- How many parameters must be estimated to train such a naive Bayes classifier?

5 $P(z)$ $P(x \mid z)$ $P(y \mid z)$ $P(x \mid \sim z)$ $P(y \mid \sim z)$

- How many parameters would have to be estimated if the naive Bayes assumption is not made, and we wish to learn the Bayes net for the joint distribution over X, Y, and Z?

7 : $P(z)$ and $P(X=i, Y=j \mid Z=k)$

$\forall i,j,k \in \{0,1\}^3$

except $P(X=1, Y=1 \mid Z=k)$ can be deduced from the other 3 cases

True or False? If true, explain why in at most two sentences. If false, explain why or give a brief counterexample.

- **(True or False?)** The error of a hypothesis measured over the training set provides a pessimistically biased estimate of the true error of the hypothesis.

- **(True or False?)** Boosting and the Weighted Majority algorithm are both methods for combining the votes of multiple classifiers.

- **(True or False?)** Unlabeled data can be used to detect overfitting.

- **(True or False?)** Gradient descent has the problem of sometimes falling into local minima, whereas EM does not.

- **(True or False?)** HMM's are a special case of MDP's.

ANDREW ID (CAPITALS): _____

NAME (CAPITALS): _____

10-701/15-781 Final, Fall 2003

- You have 3 hours.
- There are 10 questions. If you get stuck on one question, move on to others and come back to the difficult question later.
- The maximum possible total score is 100.
- Unless otherwise stated there is no need to show your working.
- Good luck!

1 Short Questions (16 points)

(a) Traditionally, when we have a real-valued input attribute during decision-tree learning we consider a binary split according to whether the attribute is above or below some threshold. Pat suggests that instead we should just have a multiway split with one branch for each of the distinct values of the attribute. From the list below choose the single biggest problem with Pat's suggestion:

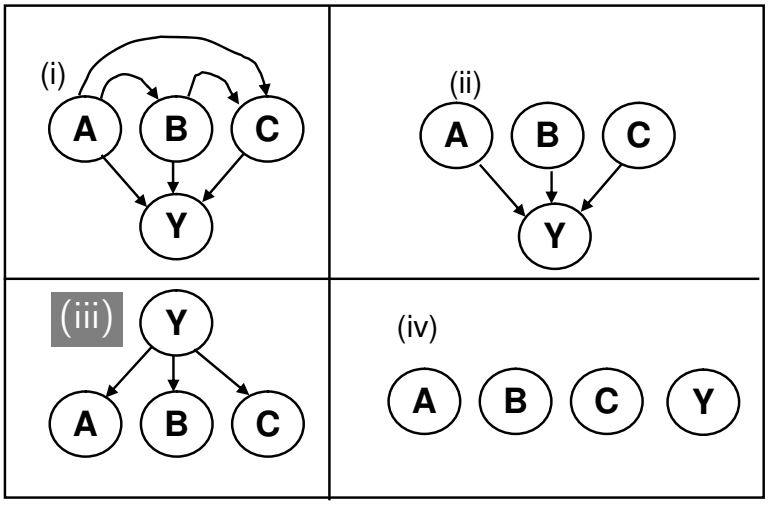
(i) It is too computationally expensive.

(ii) It would probably result in a decision tree that scores badly on the training set and a testset.

(iii) It would probably result in a decision tree that scores well on the training set but badly on a testset.

(iv) It would probably result in a decision tree that scores well on a testset but badly on a training set.

(b) You have a dataset with three categorical input attributes A, B and C. There is one categorical output attribute Y. You are trying to learn a Naive Bayes Classifier for predicting Y. Which of these Bayes Net diagrams represents the naive bayes classifier assumption?



(c) For a neural network, which one of these structural assumptions is the one that most affects the trade-off between underfitting (i.e. a high bias model) and overfitting (i.e. a high variance model):

(i) The number of hidden nodes

(ii) The learning rate

(iii) The initial choice of weights

(iv) The use of a constant-term unit input

(d) For polynomial regression, which one of these structural assumptions is the one that most affects the trade-off between underfitting and overfitting:

(i) The polynomial degree

(ii) Whether we learn the weights by matrix inversion or gradient descent

(iii) The assumed variance of the Gaussian noise

(iv) The use of a constant-term unit input

(e) For a Gaussian Bayes classifier, which one of these structural assumptions is the one that most affects the trade-off between underfitting and overfitting:

(i) Whether we learn the class centers by Maximum Likelihood or Gradient Descent

(ii) Whether we assume full class covariance matrices or diagonal class covariance matrices

(iii) Whether we have equal class priors or priors estimated from the data.

(iv) Whether we allow classes to have different mean vectors or we force them to share the same mean vector

(f) For Kernel Regression, which one of these structural assumptions is the one that most affects the trade-off between underfitting and overfitting:

(i) Whether kernel function is Gaussian versus triangular versus box-shaped

(ii) Whether we use Euclidian versus L_1 versus L_∞ metrics

(iii) The kernel width

(iv) The maximum height of the kernel function

(g) **(True or False)** Given two classifiers A and B, if A has a lower VC-dimension than B then A almost certainly will perform better on a testset.

(h) $P(\textit{Good Movie} \mid \textit{Includes Tom Cruise}) = 0.01$
$P(\textit{Good Movie} \mid \textit{Tom Cruise absent}) = 0.1$
$P(\textit{Tom Cruise in a randomly chosen movie}) = 0.01$

What is $P(\textit{Tom Cruise is in the movie} \mid \textit{Not a Good Movie})$?

$T \sim \textit{Tom Cruise is in the movie}$

$G \sim \textit{Good Movie}$

$$P(T|\tilde{}G) = \frac{P(T, \tilde{}G)}{P(\tilde{}G)}$$

$$= \frac{P(\tilde{}G|T)P(T)}{P(\tilde{}G|T)P(T) + P(\tilde{}G|\tilde{}T)P(\tilde{}T)}$$

$$= \frac{0.01 \times (1 - 0.01)}{0.01 \times (1 - 0.01) + (1 - 0.1) \times (1 - 0.01)}$$

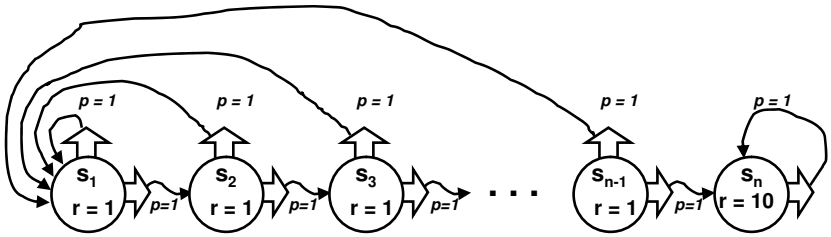
$$= 1/91 \approx 0.01099$$

2 Markov Decision Processes (13 points)

For this question it might be helpful to recall the following geometric identities, which assume $0 \leq \alpha < 1$.

$$\sum_{i=0}^{k} \alpha^i = \frac{1-\alpha^{k+1}}{1-\alpha} \qquad \sum_{i=0}^{\infty} \alpha^i = \frac{1}{1-\alpha}$$

The following figure shows an MDP with N states. All states have two actions (North and Right) except S_n, which can only self-loop. Unlike most MDPs, all state transitions are deterministic. Assume discount factor γ.



For questions (a)–(e), express your answer as a finite expression (no summation signs or . . . 's) in terms of n and/or γ.

(a) What is $J^*(S_n)$?

$$J^*(S_n) = 10 + \gamma \cdot J^*(S_n) \Longrightarrow J^*(S_n) = \frac{10}{1-\gamma}$$

(b) There is a unique optimal policy. What is it?

$A_i =$ Right $(i = 1, \ldots, n)$

(c) What is $J^*(S_1)$?

$$J^*(S_1) = 1 + \gamma + \cdots + \gamma^{n-2} + J^*(S_n) \cdot \gamma^{n-1} = \frac{1 + 9\gamma^{n-1}}{1-\gamma}$$

(d) Suppose you try to solve this MDP using value iteration. What is $J^1(S_1)$?

$J^1(S_1) = 1$

(e) Suppose you try to solve this MDP using value iteration. What is $J^2(S_1)$?

$J^2(S_1) = 1 + \gamma$

(f) Suppose your computer has exact arithmetic (no rounding errors). How many iterations of value iteration will be needed before all states record their exact (correct to infinite decimal places) J^* value? Pick one:

(i) Less than $2n$

(ii) Between $2n$ and n^2

(iii) Between $n^2 + 1$ and 2^n

(iv) It will never happen

It's a limiting process.

(g) Suppose you run policy iteration. During one step of policy iteration you compute the value of the current policy by computing the exact solution to the appropriate system of n equations in n unknowns. Suppose too that when choosing the action during the policy improvement step, ties are broken by choosing North.

Suppose policy iteration begins with all states choosing North.

How many steps of policy iteration will be needed before all states record their exact (correct to infinite decimal places) J^* value? Pick one:

(i) Less than $2n$

(ii) Between $2n$ and n^2

(iii) Between $n^2 + 1$ and 2^n

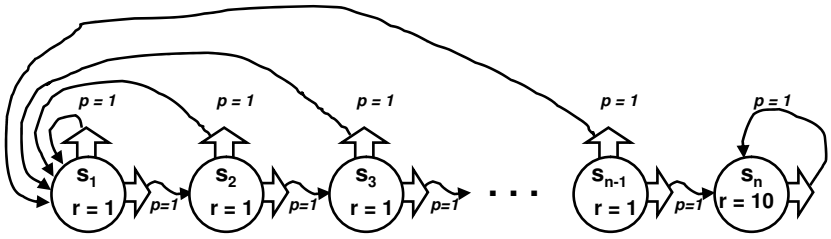
(iv) It will never happen

After i policy iterations, we have

$$Action(S_j) = \begin{cases} Right & \text{if } n - i < j < n \\ North & \text{otherwise.} \end{cases}$$

3 Reinforcement Learning (10 points)

This question uses the same MDP as the previous question, repeated here for your convenience. Again, assume $\gamma = \frac{1}{2}$.



Suppose we are discovering the optimal policy via Q-learning. We begin with a Q-table initialized with 0's everywhere:

$Q(S_i, North) = 0$ for all i
$Q(S_i, Right) = 0$ for all i

Because the MDP is determistic, we run Q-learning with a learning rate $\alpha = 1$. Assume we start Q-learning at state S_1.

(a) Suppose our exploration policy is to always choose a random action. How many steps do we expect to take before we first enter state S_n?

(i) $O(n)$ steps
(ii) $O(n^2)$ steps
(iii) $O(n^3)$ steps
(iv) $O(2^n)$ steps
(v) It will certainly never happen

You are expected to visit S_i twice before entering S_{i+1}.

(b) Suppose our exploration is greedy and we break ties by going North:

Choose North if $Q(S_i, North) \geq Q(S_i, Right)$
Choose Right if $Q(S_i, North) < Q(S_i, Right)$

How many steps do we expect to take before we first enter state S_n?

(i) $O(n)$ steps
(ii) $O(n^2)$ steps
(iii) $O(n^3)$ steps
(iv) $O(2^n)$ steps
(v) It will certainly never happen

The exploration sequence is $S_1 S_1 S_1 \ldots$

(c) Suppose our exploration is greedy and we break ties by going Right:

Choose North if $Q(S_i, North) > Q(S_i, Right)$
Choose Right if $Q(S_i, North) \leq Q(S_i, Right)$

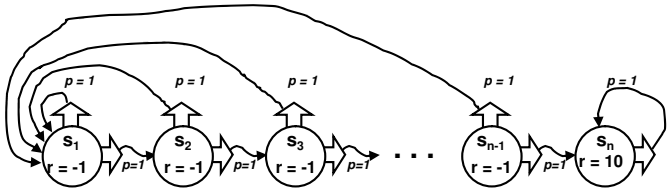
How many steps do we expect to take before we first enter state S_n?

(i) $O(n)$ steps
(ii) $O(n^2)$ steps
(iii) $O(n^3)$ steps
(iv) $O(2^n)$ steps
(v) It will certainly never happen

The exploration sequence is $S_1 S_2 S_3 \ldots S_{n-1} S_n$.

WARNING: Question (d) is only worth 1 point so you should probably just guess the answer unless you have plenty of time.

(d) In this question we work with a similar MDP except that each state other than S_n has a punishment (-1) instead of a reward (+1). S_n remains the same large reward (10). The new MDP is shown below:



Suppose our exploration is greedy and we break ties by going North:

Choose North if $Q(S_i, North) \geq Q(S_i, Right)$
Choose Right if $Q(S_i, North) < Q(S_i, Right)$

How many steps do we expect to take before we first enter state S_n?

(i) $O(n)$ steps
(ii) $O(n^2)$ steps
(iii) $O(n^3)$ steps
(iv) $O(2^n)$ steps
(v) It will certainly never happen

(ii) or (iii).

Each time a new state S_i is visited, we have to go North and jump back to S_1. So the sequence should be longer than $S_1 S_{1:2} S_{1:3} \ldots S_{1:n}$, i.e. it takes at least $O(n^2)$ steps.

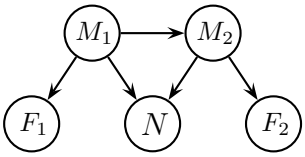
The jump from S_j to S_1 happens more than once because $Q(S_j, Right)$ keeps increasing. But the sequence should be shorter than $\{S_1\}\{S_1 S_{1:2}\}\{S_1 S_{1:2} S_{1:3}\} \ldots \{S_1 S_{1:2} \cdots S_{1:n}\}$, i.e. it takes at most $O(n^3)$ steps.

4 Bayesian Networks (11 points)

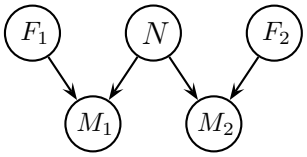
Construction. Two astronomers in two different parts of the world, make measurements M_1 and M_2 of the number of stars N in some small regions of the sky, using their telescopes. Normally, there is a small possibility of error by up to one star in each direction. Each telescope can be, with a much smaller probability, badly out of focus (events F_1 and F_2). In such a case the scientist will undercount by three or more stars or, if N is less than three, fail to detect any stars at all.

For questions (a) and (b), consider the four networks shown below.

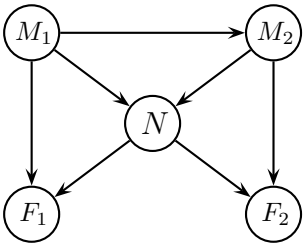
(i)



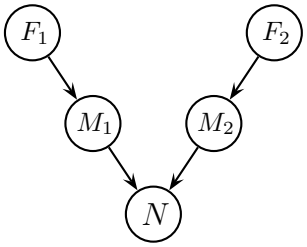
(ii)



(iii)



(iv)



(a) Which of them correctly, but not necessarily efficiently, represents the above information? **Note that there may be multiple answers**.

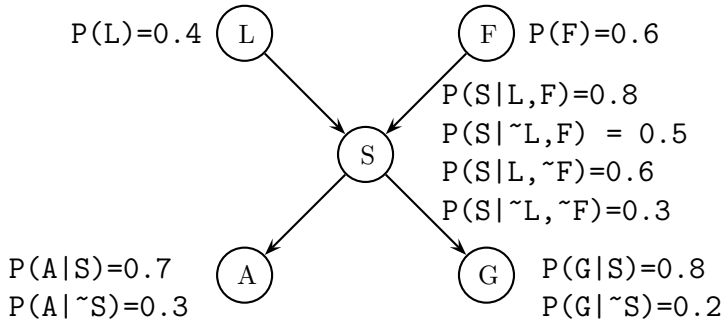
(ii) and (iii).

(ii) can be constructed directly from the physical model. (iii) is equivalent to (ii) with a different ordering of variables. (i) is incorrect because F_i and N cannot be conditionally independent given M_i. (iv) is incorrect because M_1 and M_2 cannot be independent.

(b) Which is the best network?

(ii). Intuitive and easy to interpret. Less links thus less CPT entries. Easier to assign the values of CPT entries.

Inference. A student of the Machine Learning class notices that people driving SUVs (S) consume large amounts of gas (G) and are involved in more accidents than the national average (A). He also noticed that there are two types of people that drive SUVs: people from Pennsylvania (L) and people with large families (F). After collecting some statistics, he arrives at the following Bayesian network.



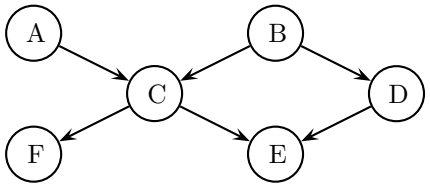
(c) What is $P(S)$?

$$\begin{aligned} P(S) &= P(S|L,F)P(L)P(F) + P(S|\tilde{}L,F)P(\tilde{}L)P(F) + \\ &\quad P(S|L,\tilde{}F)P(L)P(\tilde{}F) + P(S|\tilde{}L,\tilde{}F)P(\tilde{}L)P(\tilde{}F) \\ &= 0.4 \cdot 0.6 \cdot 0.8 + 0.6 \cdot 0.6 \cdot 0.5 + 0.4 \cdot 0.4 \cdot 0.6 + 0.6 \cdot 0.4 \cdot 0.3 \\ &= 0.54 \end{aligned}$$

(d) What is $P(S|A)$?

$$P(S|A) = \frac{P(S,A)}{P(A|S)P(S) + P(A|\tilde{}S)P(\tilde{}S)} = \frac{0.54 \cdot 0.7}{0.54 \cdot 0.7 + 0.46 \cdot 0.3} = 0.733$$

Consider the following Bayesian network. State whether the given conditional independences are implied by the net structure.



(f) **(True or False)** I<A,{},B>

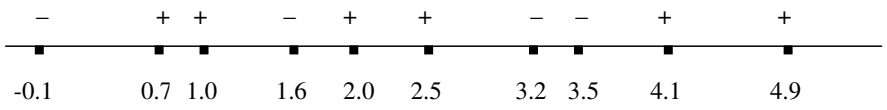
(g) **(True or False)** I<A,{E},D>

(h) **(True or False)** I<A,{F},D>

5 Instance Based Learning (8 points)

Consider the following dataset with one real-valued input x and one binary output y. We are going to use k-NN with unweighted Euclidean distance to predict y for x.

X	Y
-0.1	-
0.7	+
1.0	+
1.6	-
2.0	+
2.5	+
3.2	-
3.5	-
4.1	+
4.9	+



(a) What is the leave-one-out cross-validation error of 1-NN on this dataset? Give your answer as the number of misclassifications.

4

(b) What is the leave-one-out cross-validation error of 3-NN on this dataset? Give your answer as the number of misclassifications.

8

Consider a dataset with N examples: $\{(x_i, y_i)|1 \leq i \leq N\}$, where both x_i and y_i are real valued for all i. Examples are generated by $y_i = w_0 + w_1 x_i + e_i$ where e_i is a Gaussian random variable with mean 0 and standard deviation 1.

(c) We use least square linear regression to solve w_0 and w_1, that is $\{w_0^*, w_1^*\} = \arg\min_{\{w_0,w_1\}} \sum_{i=1}^N (y_i - w_0 - w_1 x_i)^2$.

We assume the solution is unique. Which one of the following statements is true?

(i) $\sum_{i=1}^N (y_i - w_0^* - w_1^* x_i) y_i = 0$

(ii) $\sum_{i=1}^N (y_i - w_0^* - w_1^* x_i) x_i^2 = 0$

(iii) $\sum_{i=1}^N (y_i - w_0^* - w_1^* x_i) x_i = 0$

(iv) $\sum_{i=1}^N (y_i - w_0^* - w_1^* x_i)^2 = 0$

(d) We change the optimization criterion to include local weights, that is $\{w_0^*, w_1^*\} = \arg\min_{\{w_0,w_1\}} \sum_{i=1}^N \alpha_i^2 (y_i - w_0 - w_1 x_i)^2$

where α_i is a local weight. Which one of the following statements is true?

(i) $\sum_{i=1}^N \alpha_i^2 (y_i - w_0^* - w_1^* x_i)(x_i + \alpha_i) = 0$

(ii) $\sum_{i=1}^N \alpha_i (y_i - w_0^* - w_1^* x_i) x_i = 0$

(iii) $\sum_{i=1}^N \alpha_i^2 (y_i - w_0^* - w_1^* x_i)(x_i y_i + w_1^*) = 0$

(iv) $\sum_{i=1}^N \alpha_i^2 (y_i - w_0^* - w_1^* x_i) x_i = 0$

6 VC-dimension (9 points)

Let H denote a hypothesis class, and $VC(H)$ denote its VC dimension.

(a) **(True or False)** If there exists a set of k instances that *cannot* be shattered by H, then $VC(H) < k$.

(b) **(True or False)** If two hypothesis classes H_1 and H_2 satisfy $H_1 \subseteq H_2$, then $VC(H_1) \leq VC(H_2)$.

(c) **(True or False)** If three hypothesis classes H_1, H_2 and H_3 satisfy $H_1 = H_2 \cup H_3$, then $VC(H_1) \leq VC(H_2) + VC(H_3)$.

A counter example:

$H_2 = \{h\}, h = 0$ and $H_3 = \{h'\}, h' = 1$. Apparently $VC(H_2) = VC(H_3) = 0$.

$H_1 = H_2 \cup H_3 = \{h, h'\}$.

So $VC(H_1) = 1 > VC(H_2) + VC(H_3) = 0$.

For questions (d)–(f), give $VC(H)$. No explanation is required.

(d) $H = \{h_\alpha | 0 \leq \alpha \leq 1, h_\alpha(x) = 1$ iff $x \geq \alpha$ otherwise $h_\alpha(x) = 0\}$.

1

(e) H is the set of all perceptrons in 2D plane, i.e.
$H = \{h_{\mathbf{w}} | h_{\mathbf{w}} = \theta(w_0 + w_1 x_1 + w_2 x_2)$ where $\theta(z) = 1$ iff $z \geq 0$ otherwise $\theta_z = 0\}$.

3

(f) H is the set of all circles in 2D plane. Points inside the circles are classified as 1 otherwise 0.

3

7 SVM and Kernel Methods (8 points)

(a) Kernel functions implicitly define some mapping function $\phi(\cdot)$ that transforms an input instance $\mathbf{x} \in \mathbb{R}^d$ to a high dimensional feature space Q by giving the form of dot product in Q: $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$.

Assume we use radial basis kernel function $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{1}{2}\|\mathbf{x}_i - \mathbf{x}_j\|^2)$. Thus we assume that there's some implicit unknown function $\phi(\mathbf{x})$ such that

$$\phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j) = K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{1}{2}\|\mathbf{x}_i - \mathbf{x}_j\|^2)$$

Prove that for any two input instances $\mathbf{x}_i$ and $\mathbf{x}_j$, the squared Euclidean distance of their corresponding points in the feature space Q is less than 2, i.e. prove that $\|\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)\|^2 < 2$.

$$\begin{aligned}
&\|\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)\|^2 \\
=&(\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)) \cdot (\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)) \\
=&\phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_i) + \phi(\mathbf{x}_j) \cdot \phi(\mathbf{x}_j) - 2 \cdot \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j) \\
=&2 - 2\exp(-\frac{1}{2}\|\mathbf{x}_i - \mathbf{x}_j\|^2) \\
<&2
\end{aligned}$$

(b) With the help of a kernel function, SVM attempts to construct a hyper-plane in the feature space Q that maximizes the margin between two classes. The classification decision of any $\mathbf{x}$ is made on the basis of the sign of

$$\hat{\mathbf{w}}^T \phi(\mathbf{x}) + \hat{w}_0 = \sum_{i \in SV} y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) + \hat{w}_0 = f(\mathbf{x}; \alpha, \hat{w}_0),$$

where $\hat{\mathbf{w}}$ and $\hat{w}_0$ are parameters for the classification hyper-plane in the feature space Q, SV is the set of support vectors, and α_i is the coefficient for the support vector.

Again we use the radial basis kernel function. Assume that the training instances are linearly separable in the feature space Q, and assume that the SVM finds a margin that perfectly separates the points.

(True or False) If we choose a test point $\mathbf{x}_{far}$ which is far away from any training instance $\mathbf{x}_i$ (distance here is measured in the original space $\mathbb{R}^d$), we will observe that $f(\mathbf{x}_{far}; \alpha, \hat{w}_0) \approx \hat{w}_0$.

$$\begin{aligned}
&\|\mathbf{x}_{far} - \mathbf{x}_i\| \gg 0, \ \forall i \in SV \\
\Longrightarrow &K(\mathbf{x}_{far}, \mathbf{x}_i) \approx 0, \ \forall i \in SV \\
\Longrightarrow &\sum\nolimits_{i \in SV} y_i \alpha_i K(\mathbf{x}_{far}, \mathbf{x}_i) \approx 0 \\
\Longrightarrow &f(\mathbf{x}_{far}; \alpha, \hat{w}_0) \approx \hat{w}_0
\end{aligned}$$

(c) **(True or False)** The SVM learning algorithm is guaranteed to find the globally optimal hypothesis with respect to its object function.

See Burges' tutorial.

(d) **(True or False)** The VC dimension of a Perceptron is smaller than the VC dimension of a simple linear SVM.

Both Perceptron and linear SVM are linear discriminators (i.e. a line in 2D space or a plane in 3D space ...), so they should have the same VC dimension.

(e) **(True or False)** After being mapped into feature space Q through a radial basis kernel function, a Perceptron may be able to achieve better classification performance than in its original space (though we can't guarantee this).

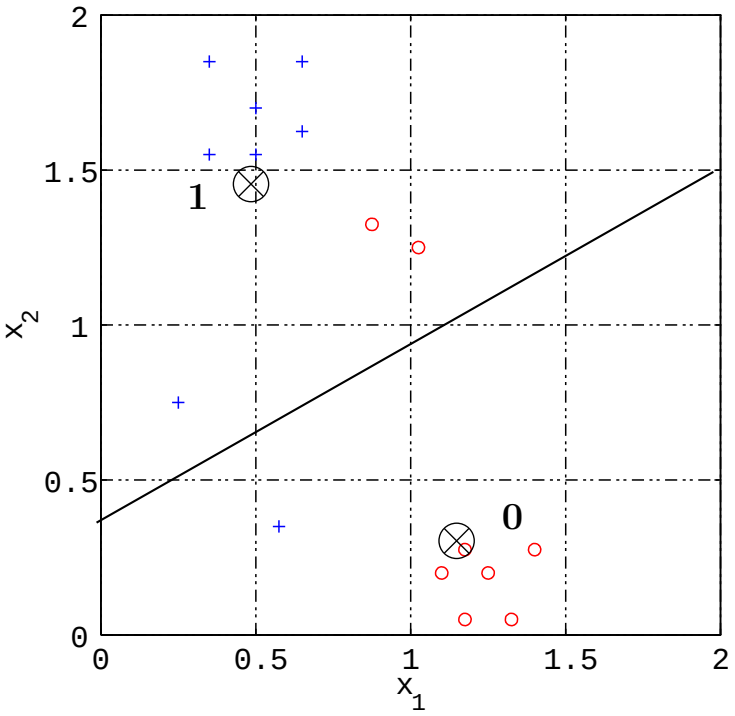
Sometimes it isn't sufficient for a given learning algorithm to work in the input space because the assumption behind the algorithm doesn't match the real pattern of the data. For example, SVM and Perceptron require the data are linearly separable. When the assumption isn't held, we may apply some kind of transformation to the data, mapping them to a new space where the learning algorithm can be used. Kernel function provides us a means to define the transformation. You may have read some papers that report improvements on classification performance using kernel function. However, the improvements are usually obtained from careful selection and tuning of parameters. Namely, we can't guarantee the improvements are always available.

(f) **(True or False)** After mapped into feature space Q through a radial basis kernel function, 1-NN using unweighted Euclidean distance may be able to achieve better classification performance than in original space (though we can't guarantee this).

Suppose $\mathbf{x}_i$ and $\mathbf{x}_j$ are two neighbors for the test instance $\mathbf{x}$ such that $\|\mathbf{x}-\mathbf{x}_i\| < \|\mathbf{x}-\mathbf{x}_j\|$. After mapped to feature space, $\|\phi(\mathbf{x}) - \phi(\mathbf{x}_i)\|^2 = 2 - 2\exp(-\frac{1}{2}\|\mathbf{x} - \mathbf{x}_i\|^2) < 2 - 2\exp(-\frac{1}{2}\|\mathbf{x} - \mathbf{x}_j\|^2) = \|\phi(\mathbf{x}) - \phi(\mathbf{x}_j)\|^2$. So, if $\mathbf{x}_i$ is the nearest neighbor of $\mathbf{x}$ in the original space, it will also be the nearest neighbor in the feature space. Therefore, 1-NN doesn't work better in the feature space. Please note that k-NN using non-Euclidean distance or weighted voting may work.

8 GMM (8 points)

Consider the classification problem illustrated in the following figure. The data points in the figure are labeled, where "o" corresponds to class 0 and "+" corresponds to class 1. We now estimate a GMM consisting of 2 Gaussians, one Gaussian per class, with the constraint that the covariance matrices are identity matrices. The mixing proportions (class frequencies) and the means of the two Gaussians are free parameters.



(a) Plot the maximum likelihood estimates of the means of the two Gaussians in the figure. Mark the means as points "x" and label them "0" and "1" according to the class.

The means of the two Gaussians should be close to the center of mass of the points.

(b) Based on the learned GMM, what is the probability of generating a new data point that belongs to class 0?

0.5

(c) How many data points are classified *incorrectly*?

3

(d) Draw the decision boundary in the same figure.

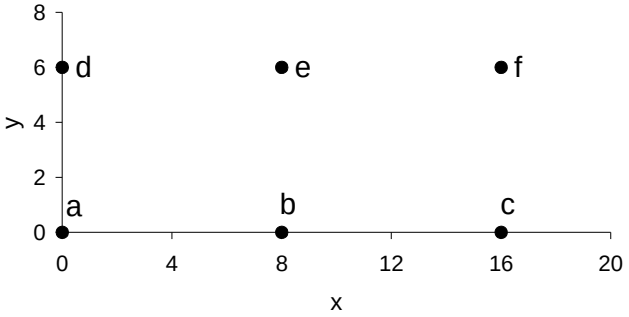
Since the two classes have the same number of points and identical covariance matrices, the decision boundary should be a straight line, which is also the orthogonal bisector of the line segment connecting the class means.

9 K-means Clustering (9 points)

There is a set S consisting of 6 points in the plane shown as below, $a = (0, 0)$, $b = (8, 0)$, $c = (16, 0)$, $d = (0, 6)$, $e = (8, 6)$, $f = (16, 6)$. Now we run the k-means algorithm on those points with $k = 3$. The algorithm uses the Euclidean distance metric (i.e. the straight line distance between two points) to assign each point to its nearest centroid. Ties are broken in favor of the centroid to the left/down. Two definitions:

- A **k-starting configuration** is a subset of k starting points from S that form the initial centroids, e.g. $\{a, b, c\}$.
- A **k-partition** is a partition of S into k non-empty subsets, e.g. $\{a, b, e\}, \{c, d\}, \{f\}$ is a 3-partition.

Clearly any k-partition induces a set of k centroids in the natural manner. A k-partition is called *stable* if a repetition of the k-means iteration with the induced centroids leaves it unchanged.



(a) How many 3-starting configurations are there? (Remember, a 3-starting configuration is just a subset, of size 3, of the six datapoints).

$C_6^3 = 20$

(b) Fill in the following table:

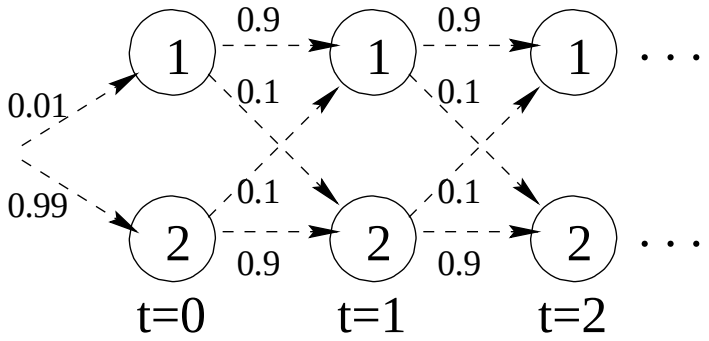
3-partition	Stable?	An example 3-starting configuration that can arrive at the 3-partition after 0 or more iterations of k-means (or write "none" if no such 3-starting configuration exists)	# of unique 3-starting configurations that arrive at the 3-partition
$\{a, b, e\}, \{c, d\}, \{f\}$	N	none	0
$\{a, b\}, \{d, e\}, \{c, f\}$	Y	{b, c, e}	4
$\{a, d\}, \{b, e\}, \{c, f\}$	Y	{a, b, c}	8
$\{a\}, \{d\}, \{b, c, e, f\}$	Y	{a, b, d}	2
$\{a, b\}, \{d\}, \{c, e, f\}$	Y	none	0
$\{a, b, d\}, \{c\}, \{e, f\}$	Y	{a, c, f}	1

10 Hidden Markov Models (8 points)

Consider a hidden Markov model illustrated as the figure shown below, which shows the hidden state transitions and the associated probabilities along with the initial state distribution. We assume that the state dependent outputs (coin flips) are governed by the following distributions

$P(x = heads|s = 1) = 0.51$
$P(x = heads|s = 2) = 0.49$
$P(x = tails|s = 1) = 0.49$
$P(x = tails|s = 2) = 0.51$

In other words, our coin is slightly biased towards *heads* in state 1 whereas in state 2 *tails* is a somewhat more probable outcome.



(a) Now, suppose we observe three coin flips all resulting in *heads*. The sequence of observations is therefore $heads; heads; heads$. What is the most likely state sequence given these three observations? (It is not necessary to use the Viterbi algorithm to deduce this, nor any subsequent questions).

2,2,2

The probabilities of outputting head are nearly identical in two states and it is very likely that the system starts from state 2 and stay there. It loses a factor of 9 in probability if it ever switchs to state 1.

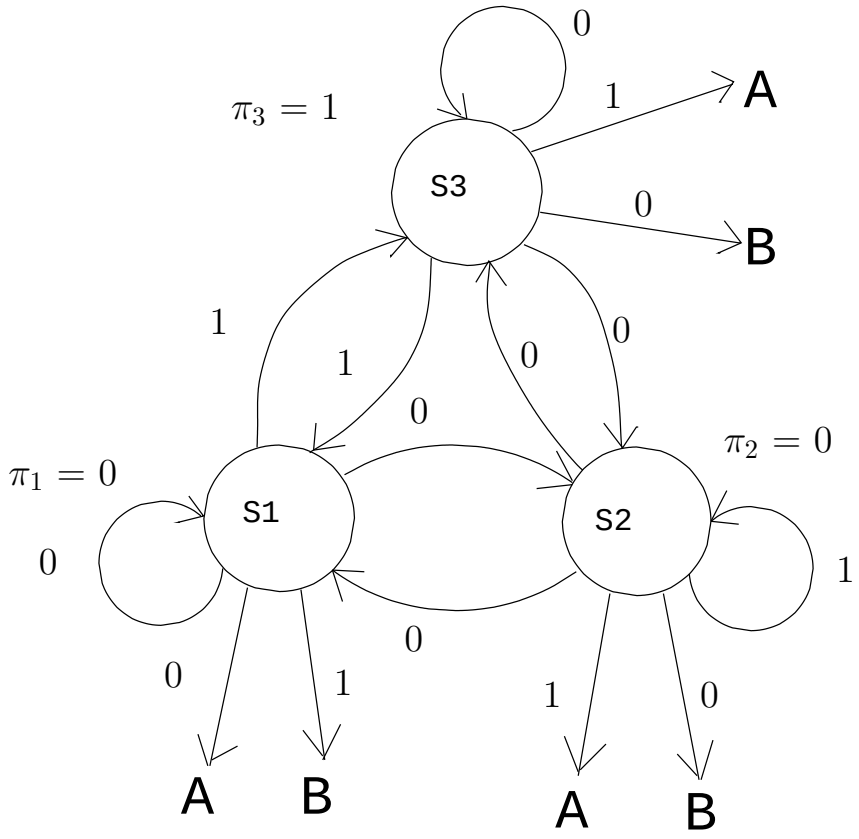
(b) What happens to the most likely state sequence if we observe a long sequence of all heads (e.g., 10^6 heads in a row)?

2,1,1,1,...

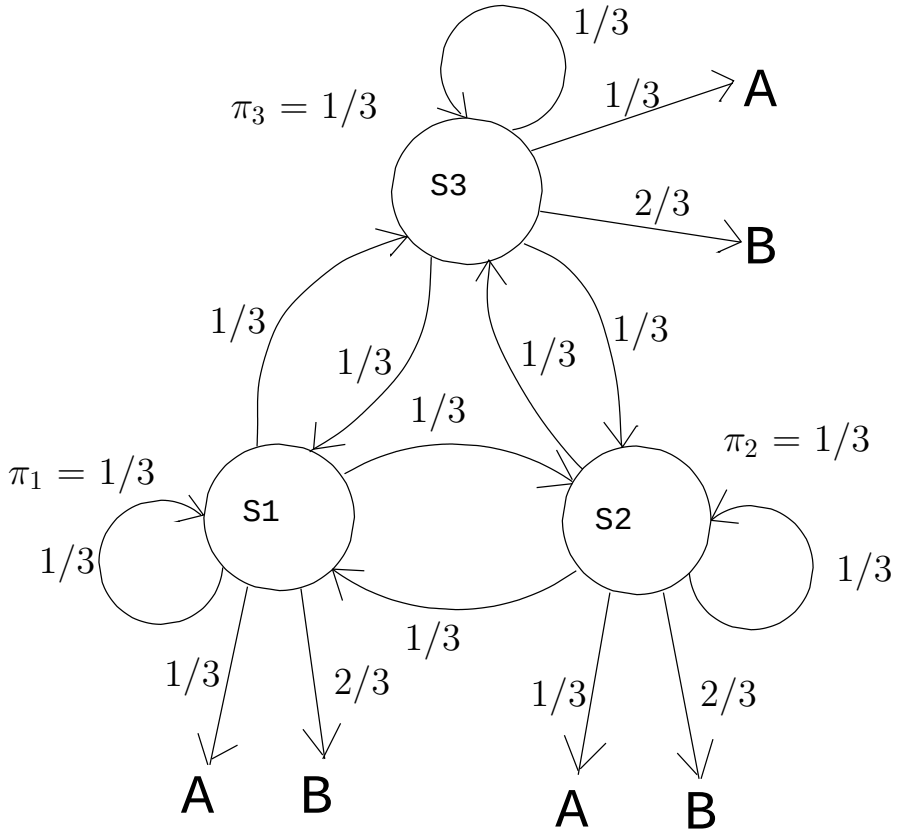
When the number of continuous observations of heads increases, the pressure for the system to switch to state 1 also increases, as state 1 has a slight advantage per observation. Eventually the switch will take place and then there's no benefit from ever switching back to state 2. The cost of the transition switching from state 2 to state 1 is the same regardless of when it takes place. But switching earlier is better than later, since the likelihood of observing the long sequence of all heads is greater. However, it is somewhat better to go via state 2 initially and switch right after (0.99*0.49*0.1 ...) rather than start from state 1 to begin with (0.01*0.51*0.9 ...).

(c) Consider the following 3-state HMM, π_1, π_2 and π_3 are the probabilities of starting from each state $S1$, $S2$ and $S3$. Give a set of values so that the resulting HMM maximizes the likelihood of the output sequence ABA.

There are many possible solutions, and they are all correct as long as they output ABA with probability 1, and the parameter settings of the models are sound. Here is one possible solution:



(d) We're going to use EM to learn the parameters for the following HMM. Before the first iteration of EM we have initialized the parameters as shown in the following figure. **(True or False)** For these initial values, EM will successfully converge to the model that maximizes the likelihood of the training sequence ABA.



Note the symmetry of the initial set of values over S_1, S_2 and S_3. After each EM iteration, the transition matrix will keep the same ($a_{ij} = 1/3$). The observation matrix may change, but the symmetry still holds ($b_i(A) = b_j(A)$).

(e) **(True or False)** In general when are trying to learn an HMM with a small number of states from a large number of observations, we can almost always increase the training data likelihood by permitting more hidden states.

To model any finite length sequence, we can increase the number of hidden states in an HMM to be the number of observations in the sequence and therefore (with appropriate parameter choices) generate the observed sequence with probability 1. Given a fixed number of finite sequences (say n), we would still be able to assign probability $1/n$ for generating each sequence. This is not useful, of course, but highlights the fact that the complexity of HMMs is not limited.

Solution of Final Exam : 10-701/15-781 Machine Learning

Fall 2004

Dec. 12th 2004

Your Andrew ID in capital letters:

Your full name:

- There are 9 questions. Some of them are easy and some are more difficult. So, if you get stuck on any one of the questions, proceed with the rest of the questions and return back at the end if you have time remaining.
- The maximum score of the exam is 100 points
- If you need more room to work out your answer to a question, use the back of the page and clearly mark on the front of the page if we are to look at what's on the back.
- You should attempt to answer all of the questions.
- You may use any and all notes, as well as the class textbook.
- You have 3 hours.
- Good luck!

Problem 1. Assorted Questions (16 points)

(a) [*3.5 pts*] Suppose we have a sample of real values, called x_1, x_2, ..., x_n. Each sampled from p.d.f. $p(x)$ which has the following form:

$$f(x) = \begin{cases} \alpha e^{-\alpha x}, & if \quad x \geq 0 \\ 0, & otherwise \end{cases} \tag{1}$$

where α is an unknown parameter. Which one of the following expressions is the maximum likelihood estimation of α? (Assume that in our sample, all x_i are large than 1.)

1). $\frac{\sum_{i=1}^{n} log(x_i)}{n}$ 2). $\frac{\max_{i=1}^{n} log(x_i)}{n}$

3). $\frac{n}{\sum_{i=1}^{n} log(x_i)}$ 4). $\frac{n}{\max_{i=1}^{n} log(x_i)}$

5). $\frac{\sum_{i=1}^{n} x_i}{n}$ 6). $\frac{\max_{i=1}^{n} x_i}{n}$

7). $\frac{n}{\sum_{i=1}^{n} x_i}$ 8). $\frac{n}{\max_{i=1}^{n} x_i}$

9). $\frac{\sum_{i=1}^{n} x_i^2}{n}$ 10). $\frac{\max_{i=1}^{n} x_i^2}{n}$

11). $\frac{n}{\sum_{i=1}^{n} x_i^2}$ 12). $\frac{n}{\max_{i=1}^{n} x_i^2}$

13). $\frac{\sum_{i=1}^{n} e^{x_i}}{n}$ 14). $\frac{\max_{i=1}^{n} e^{x_i}}{n}$

15). $\frac{n}{\sum_{i=1}^{n} e^{x_i}}$ 16). $\frac{n}{\max_{i=1}^{n} e^{x_i}}$

Answer: Choose [7].

(b) . [*7.5 pts*] Suppose that X_1, ..., X_m are categorical input attributes and Y is categorical output attribute. Suppose we plan to learn a decision tree without pruning, using the standard algorithm.

b.1 (**True or False** -*1.5 pts*) : If X_i and Y are independent in the distribution that generated this dataset, then X_i will not appear in the decision tree.

Answer: False (because the attribute may become relevant further down the tree when the records are restricted to some value of another attribute) (e.g. XOR)

b.2 (**True or False** -*1.5 pts*) : If $IG(Y|X_i) = 0$ according to the values of entropy and conditional entropy computed from the data, then X_i will not appear in the decision tree.

Answer: False for same reason

b.3 (**True or False** -*1.5 pts*) : The maximum depth of the decision tree must be less than m+1 .

Answer: True because the attributes are categorical and can each be split only once

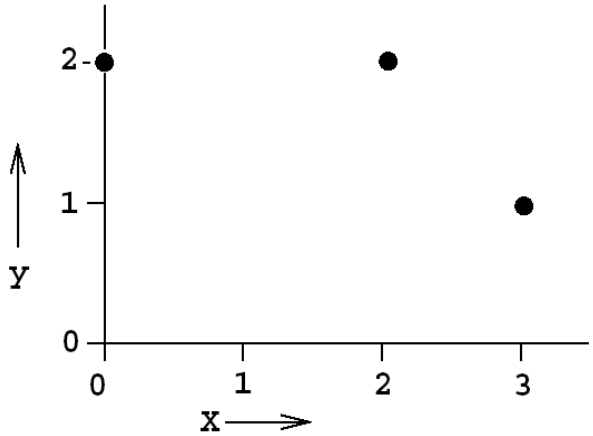
b.4 (**True or False** -*1.5 pts*) : Suppose data has R records, the maximum depth of the decision tree must be less than $1 + log_2 R$

Answer: False because the tree may be unbalanced

b.5 (**True or False** -*1.5 pts*) : Suppose one of the attributes has R distinct values, and it has a unique value in each record. Then the decision tree will certainly have depth 0 or 1 (i.e. will be a single node, or else a root node directly connected to a set of leaves)

Answer: True because that attribute will have perfect information gain. If an attribute has perfect information gain it must split the records into "pure" buckets which can be split no more.

(c) [*5 pts*] Suppose you have this data set with one real-valued input and one real-valued output:



x	y
0	2
2	2
3	1

(c.1) What is the mean squared leave one out cross validation error of using linear regression ? (i.e. the mode is $y = \beta_0 + \beta_1 x + noise$)

Answer: $\frac{2^2+(2/3)^2+1^2}{3} = 49/27$

(c.2) Suppose we use a trivial algorithm of predicting a constant $y = c$. What is the mean squared leave one out error in this case? (Assume c is learned from the non-left-out data points.)

Answer: $\frac{0.5^2+0.5^2+1^2}{3} = 1/2$

Problem 2. Bayes Rule and Bayes Classifiers (12 points)

Suppose you are given the following set of data with three Boolean input variables a, b, and c, and a single Boolean output variable K.

a	b	c	K
1	0	1	1
1	1	1	1
0	1	1	0
1	1	0	0
1	0	1	0
0	0	0	1
0	0	0	1
0	0	1	0

For parts (a) and (b), assume we are using a naive Bayes classifier to predict the value of K from the values of the other variables.

(a) [*1.5 pts*] According to the naive Bayes classifier, what is $P(K = 1|a = 1 \wedge b = 1 \wedge c = 0)$?

Answer: 1/2.

$$\begin{aligned} P(K = 1|a = 1 \wedge b = 1 \wedge c = 0) &= P(K = 1 \wedge a = 1 \wedge b = 1 \wedge c = 0)/P(a = 1 \wedge b = 1 \wedge c = 0) \\ &= P(K{=}\,1) \cdot P(a = 1|K = 1) \cdot P(b = 1|K = 1) \cdot P(c = 0|K = 1)/ \\ &\qquad P(a = 1 \wedge b = 1 \wedge c = 0 \wedge K = 1) + P(a = 1 \wedge b = 1 \wedge c = 0 \wedge K = 0). \end{aligned}$$

(b) [*1.5 pts*] According to the naive Bayes classifier, what is $P(K = 0|a = 1 \wedge b = 1)$?

Answer: 2/3.

$$\begin{aligned} P(K = 0|a = 1 \wedge b = 1) &= P(K = 0 \wedge a = 1 \wedge b = 1)/P(a = 1 \wedge b = 1) \\ &= P(K{=}\,0) \cdot P(a = 1|K = 0) \cdot P(b = 1|K = 0)/ \\ &\qquad P(a = 1 \wedge b = 1 \wedge K = 1) + P(a = 1 \wedge b = 1 \wedge K = 0). \end{aligned}$$

Now, suppose we are using a joint Bayes classifier to predict the value of K from the values of the other variables.

(c) [*1.5 pts*] According to the joint Bayes classifier, what is $P(K = 1|a = 1 \wedge b = 1 \wedge c = 0)$?

Answer: 0.
Let $num(X)$ be the number of records in our data matching X. Then we have $P(K = 1|a = 1 \wedge b = 1 \wedge c = 0) = num(K$

(d) [*1.5 pts*] According to the joint Bayes classifier, what is $P(K = 0|a = 1 \wedge b = 1)$?

Answer: 1/2.
$P(K = 0|a = 1 \wedge b = 1) = num(K = 0 \wedge a = 1 \wedge b = 1)/num(a = 1 \wedge b = 1) = 1/2$.

In an unrelated example, imagine we have three variables X, Y, and Z.

(e) [*2 pts*] Imagine I tell you the following:

$$P(Z|X) = 0.7$$
$$P(Z|Y) = 0.4$$

Do you have enough information to compute $P(Z|X \wedge Y)$? If not, write "not enough info". If so, compute the value of $P(Z|X \wedge Y)$ from the above information.

Answer: Not enough info.

(f) [*2 pts*] Instead, imagine I tell you the following:

$$P(Z|X) = 0.7$$
$$P(Z|Y) = 0.4$$
$$P(X) = 0.3$$
$$P(Y) = 0.5$$

Do you now have enough information to compute $P(Z|X \wedge Y)$? If not, write "not enough info". If so, compute the value of $P(Z|X \wedge Y)$ from the above information.

Answer: Not enough info.

(g) [*2 pts*] Instead, imagine I tell you the following (falsifying my earlier statements):

$$P(Z \wedge X) = 0.2$$
$$P(X) = 0.3$$
$$P(Y) = 1$$

Do you now have enough information to compute $P(Z|X \wedge Y)$? If not, write "not enough info". If so, compute the value of $P(Z|X \wedge Y)$ from the above information.

Answer: 2/3.
$P(Z|X \wedge Y) = P(Z|X)$ since $P(Y) = 1$. In this case, $P(Z|X \wedge Y) = P(Z \wedge X)/P(X) = 0.2/0.3 = 2/3$.

Problem 3. SVM (9 points)

(a) (**True/False** - *1 pt*) Support vector machines, like logistic regression models, give a probability distribution over the possible labels given an input example.

Answer: False

(b) (**True/False** - *1 pt*) We would expect the support vectors to remain the same in general as we move from a linear kernel to higher order polynomial kernels.

Answer: False (There are no guarantees that the support vectors remain the same. The feature vectors corresponding to polynomial kernels are non-linear functions of the original input vectors and thus the support points for maximum margin separation in the feature space can be quite different.)

(c) (**True/False** - *1 pt*) The maximum margin decision boundaries that support vector machines construct have the lowest generalization error among all linear classifiers.

Answer: False (The maximum margin hyperplane is often a reasonable choice but it is by no means optimal in all cases.)

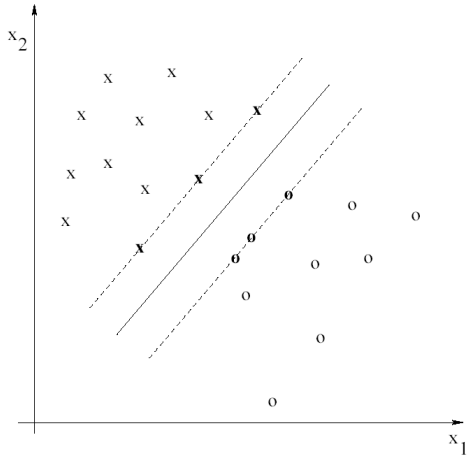
(d) (**True/False** - *1 pt*) Any decision boundary that we get from a generative model with class-conditional Gaussian distributions could in principle be reproduced with an SVM and a polynomial kernel of degree less than or equal to three.

Answer: True (A polynomial kernel of degree two suffices to represent any quadratic decision boundary such as the one from the generative model in question.)

(e) (**True/False** - *1 pts*) The values of the margins obtained by two different kernels $K_1(x, x_0)$ and $K_2(x, x_0)$ on the same training set do not tell us which classifier will perform better on the test set.

Answer: True (We need to normalize the margin for it to be meaningful. For example, a simple scaling of the feature vectors would lead to a larger margin. Such a scaling does not change the decision boundary, however, and so the larger margin cannot directly inform us about generalization.)

(f) (*2 pts*) What is the leave-one-out cross-validation error estimate for maximum margin separation in the following figure ? (we are asking for a number)



Answer: 0

Based on the figure we can see that removing any single point would not chance the resulting maximum margin separator. Since all the points are initially classified correctly, the leave-one-out error is zero.

(g) (*2 pts*) Now let us discuss a SVM classifier using a second order polynomial kernel. The first polynomial kernel maps each input data x to $\Phi_1(x) = [x, x^2]^T$. The second polynomial kernel maps each input data x to $\Phi_2(x) = [2x, 2x^2]^T$.

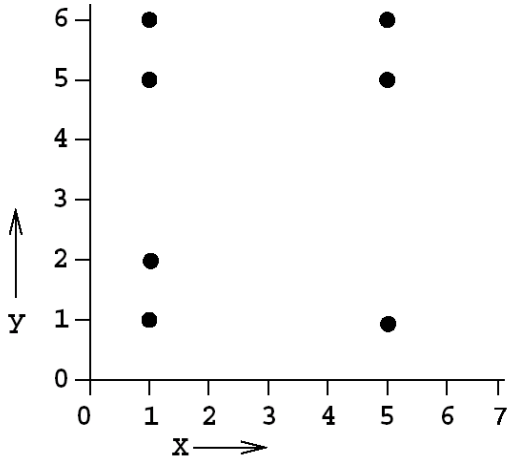
In general, is the margin we would attain using $\Phi_2(x)$

A. ( ) greater
B. ( ) equal
C. ( ) smaller
D. ( ) any of the above

in comparison to the margin resulting from using $\Phi_1(x)$?
Answer: A.

Problem 4. Instance based learning (8 points)

The following picture shows a dataset with one real-valued input x and one real-valued output y. There are seven training points.



Suppose you are training using kernel regression using some unspecified kernel function. The only thing you know about the kernel function is that it is a monotonically decreasing function of distance that decays to zero at a distance of 3 units (and is strictly greater than zero at a distance of less than 3 units).

(a) (*2 pts*) What is the predicted value of y when x = 1?

Answer: $\frac{1+2+5+6}{4} = 3.5$

(b) (*2 pts*) What is the predicted value of y when x = 3?

Answer: $\frac{1+2+5+6+1+5+6}{7} = 26/7$

(c) (*2 pts*) What is the predicted value of y when x = 4?

Answer: $\frac{1+5+6}{3} = 4$

(d) (*2 pts*) What is the predicted value of y when x = 7?

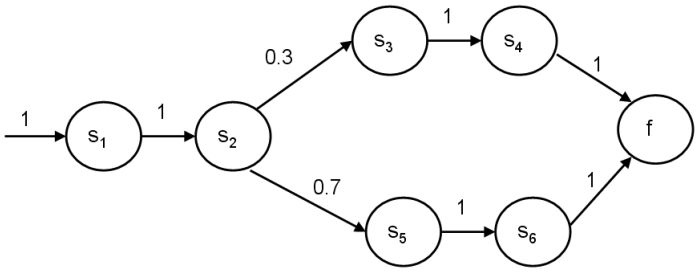
Answer: $\frac{1+5+6}{3} = 4$

Problem 5. HMM (12 points)

Consider the HMM defined by the transition and emission probabilities in the table below. This HMM has six states (plus a start and end states) and an alphabet with four symbols (A,C, G and T). Thus, the probability of transitioning from state S_1 to state S_2 is 1, and the probability of emitting A while in state S_1 is 0.3.

	0	S_1	S_2	S_3	S_4	S_5	S_6	f	A	C	G	T
0	0	1	0	0	0	0	0	0				
S_1	0	0	1	0	0	0	0	0	0.5	0.3	0	0.2
S_2	0	0	0	0.3	0	0.7	0	0	0.1	0.1	0.2	0.6
S_3	0	0	0	0	1	0	0	0	0.2	0	0.1	0.7
S_4	0	0	0	0	0	0	0	1	0.1	0.3	0.4	0.2
S_5	0	0	0	0	0	0	1	0	0.1	0.3	0.3	0.3
S_6	0	0	0	0	0	0	0	1	0.2	0.3	0	0.5

Here is the state diagram:



For each of the pairs belows, place $<$, $>$ or $=$ between the right and left components of each pair. (*2 pts each*):

(a) $P(O_1 = A, O_2 = C, O_3 = T, O_4 = A, q_1 = S_1, q_2 = S_2)$
$P(O_1 = A, O_2 = C, O_3 = T, O_4 = A | q_1 = S_1, q_2 = S_2)$

Below we will use a shortened notation. Specifically we will us $P(A, C, T, A, S_1, S_2)$ instead of $P(O_1 = A, O_2 = C, O_3 = T, O_4 = A, q_1 = S_1, q_2 = S_2)$, $P(A, C, T, A)$ instead of $P(O_1 = A, O_2 = C, O_3 = T, O_4 = A)$ and so forth.

Answer: $=$
$P(A, C, T, A, S_1, S_2) = P(A, C, T, A | S_1, S_2) P(S_1, S_2) = P(A, C, T, A | S_1, S_2)$, since $P(S_1, S_2) = 1$

(b) $P(O_1 = A, O_2 = C, O_3 = T, O_4 = A, q_3 = S_3, q_4 = S_4)$
$P(O_1 = A, O_2 = C, O_3 = T, O_4 = A | q_3 = S_3, q_4 = S_4)$

Answer: $<$
As in (b), $P(A, C, T, A, S_3, S_4) = P(A, C, T, A | S_3, S_4) P(S_3, S_4)$ however, since $P(S_3, S_4) = 0.3$, then the right hand side is bigger.

(c) $P(O_1 = A, O_2 = C, O_3 = T, O_4 = A, q_3 = S_3, q_4 = S_4)$
$P(O_1 = A, O_2 = C, O_3 = T, O_4 = A, q_3 = S_5, q_4 = S_6)$

Answer: $<$
The first two emissions (A and C) do not matter since they are the same. Thus, the right hand side translates to $P(O_3 = T, O_4 = A, q_3 = S_3, q_4 = S_4) = P(O_3 = T, O_4 = A | q_3 = S_3, q_4 = S_4) P(S_3, S_4) = 0.7 * 0.1 * 0.3 = 0.021$ while the right hand side is $0.3 * 0.2 * 0.7 = 0.042$.

(d) $P(O_1 = A, O_2 = C, O_3 = T, O_4 = A)$ $P(O_1 = A, O_2 = C, O_3 = T, O_4 = A, q_3 = S_3, q_4 = S_4)$

Answer: $>$

Here the left hand side is: $P(A, C, T, A, S_3, S_4) + P(A, C, T, A, S_5, S_6)$. The right side of the summation is the right hand side above. Since the left side of the summation is greater than 0, the left hand side is greater.

(e) $P(O_1 = A, O_2 = C, O_3 = T, O_4 = A)$ $P(O_1 = A, O_2 = C, O_3 = T, O_4 = A | q_3 = S_3, q_4 = S_4)$

Answer: $<$

As mentioned for (e) the left hand side is: $P(A, C, T, A, S_3, S_4) + P(A, C, T, A, S_5, S_6) = P(A, C, T, A | S_3, S_4)P(S_3, S_4) + P(A, C, T, A | S_5, S_6)P(S_5, S_6)$. Since $P(A, C, T, A | S_3, S_4) > P(A, C, T, A | S_5, S_6)$ the left hand side is lower from the right hand side.

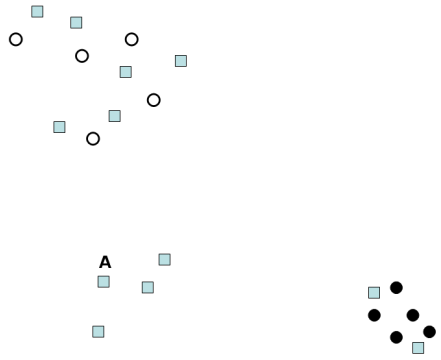
(f) $P(O_1 = A, O_2 = C, O_3 = T, O_4 = A)$ $P(O_1 = A, O_2 = T, O_3 = T, O_4 = G)$

Answer: $<$

Since the first and third letters are the same, we only need to worry about the second and fourth. The left hand side is: $0.1 * (0.3 * 0.1 + 0.7 * 0.2) = 0.017$ while the right hand side is: $0.6 * (0.7 * 0 + 0.3 * 0.4) = 0.072$.

Problem 6. Learning from labeled and unlabeled data (10 points)

Consider the following figure which contains labeled (class 1 black circles class 2 hollow circles) and unlabeled (squares) data. We would like to use two methods discussed in class (re-weighting and co-training) in order to utilize the unlabeled data when training a Gaussian classifier.

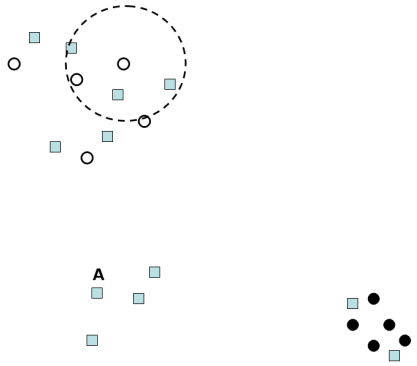


(a) (*2 pts*) How can we use co-training in this case (what are the two classifiers) ?

Answer:

Co-training partitions thew feature space into two separate sets and uses these sets to construct independent classifiers. Here, the most natural way is to use one classifier (a Gaussian) for the x axis and the second (another Gaussian) using the y axis.

(b) We would like to use re-weighting of unlabeled data to improve the classification performance. Re-weighting will be done by placing a the dashed circle on each of the labeled data points and counting the number of unlabeled data points in that circle. Next, a Gaussian classifier is run with the new weights computed.



(b.1). (*2 pts*) To what class (hollow circles or full circles) would we assign the unlabeled point A is we were training a Gaussian classifier using **only** the labeled data points (with no re-weighting)?

Answer:
Hollow class. Note that the hollow points are much more spread out and so the Gaussian learned for them will have a higher variance.

(b.2). (*2 pts*) To what class (hollow circles or full circles) would we assign the unlabeled point A is we were training a classifier using the re-weighting procedure described above?

Answer:
Again, the hollow class. Re-weighting will not change the result since it will be done independently for each of the two classes, and will produce very similar class centers to the ones in 1 above.

(c) (*4 pts*) When we handle a polynomial regression problem, we would like to decide what degree of polynomial to use in order to fit a test set. The table below describes the dis-agreement between the different polynomials on unlabeled data and also the disagreement with the labeled data. Based on the method presented in class, which polynomial should we chose for this data? **Which of the two tables do you prefer?**

	Disagreement on unlabled data					Disagreement on training data
Degree	1	2	3	4	5	
1	0	0.3	0.5	0.6	0.7	0.4
2		0	0.2	0.4	0.5	0.2
3			0	0.2	0.5	0.1
4				0	0.3	0
5					0	0

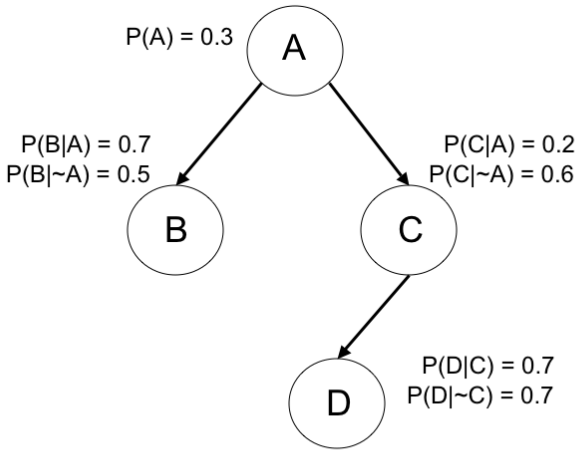
Answer:

The degree we would select is 3. Based on the classification accuracy, it is beneficial to use higher degree polynomials. However, as we said in class these might overfit. One way to test if they do or don't is to check consistency on unlabeled data by requiring that the triangle inequality will hold for the selected degree. For a third degree this is indeed the case since $u(2,3) = 0.2 \leq l(2) + l(3) = 0.2 + 0.1$ (where $u(2,3)$ is the disagreement between the second and third degree polynomials on the unlabeled data and $l(2)$ is the disagreement between degree 2 and the labeled data). Similarly, $u(1,3) = 0.5 \leq l(1) + l(3) = 0.4 + 0.1$. In contrast, this does not hold for a fourth degree polynomial since $u(3,4) = 0.2 > l(3) + l(4) = 0.1$.

Problem 7. Bayes Net Inference (10 points)

For (a) through (c), compute the following probabilities from the Bayes net below.

Hint: These examples have been designed so that none of the calculations should take you longer than a few minutes. If you find yourself doing dozens of calculations on a question sit back and look for shortcuts. This can be done easily if you notice a certain special property of the numbers on this diagram.



(a) (*2 pts*) $P(A|B) =$

Answer: 3/8.
$P(A|B) = P(A \wedge B)/P(B) = P(B|A) \cdot P(A)/(P(B|A) \cdot P(A) + P(B|\ A) \cdot P(\ A)) = 0.21/(0.21 + 0.35) = 3/8$.

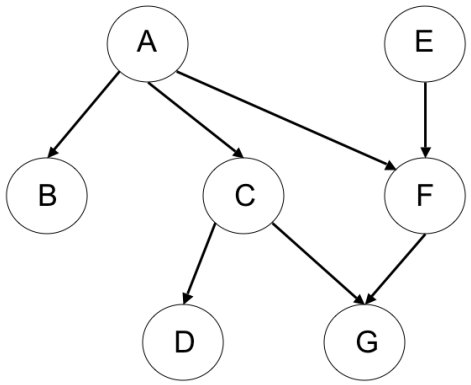
(b) (*2 pts*) $P(B|D) =$

Answer: 0.56.
$P(D|C) = P(D|\ C)$ so D is independent of C, and is not influencing the Bayes net. So $P(B|D) = P(B)$, which we calculated in (a) to be 0.56.

(c) (*2 pts*) $P(C|B) =$

Answer: $5/11$.
$P(C|B) = (P(A \wedge B \wedge C) + P(\ A \wedge B \wedge C))/P(B) = (P(A) \cdot P(B|A) \cdot P(C|A) + P(\ A) \cdot P(B|\ A) \cdot P(C|\ A))/P(B) = (0.042 + 0.21)/0.56\ = 5/11$.

For (d) through (g), indicate whether the given statement is **TRUE** or **FALSE** in the Bayes net given below.



(d) [**T/F** - (*1 pt*)] I$<A, \{\}, E>$
Answer: TRUE.

(e) [**T/F** - (*1 pt*)] I$<A, G, E>$
Answer: FALSE.

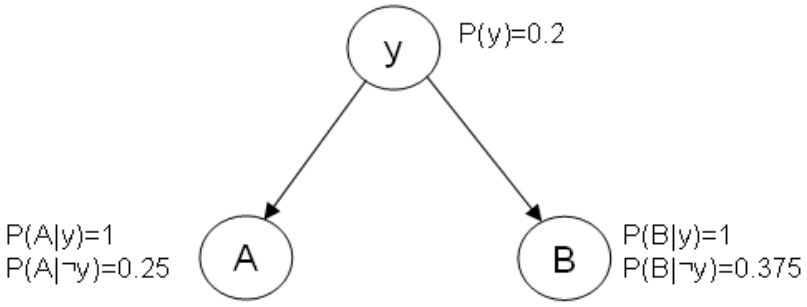
(f) [**T/F** - (*1 pt*)] I$<C, \{A, G\}, F>$
Answer: FALSE.

(g) [**T/F** - (*1 pt*)] I$<B, \{C, E\}, F>$
Answer: FALSE.

Problem 8. Bayes Nets II (12 points)

(a) *(4 points)* Suppose we use a naive Bayes classifier to learn a classifier for $y = A \wedge B$, where A, B are independent of each other boolean random variables with $P(A) = 0.4$, $P(B) = 0.5$. Draw the Bayes net that represents the independence assumptions of our classifier and fill in the probability tables for the net.

Answer:

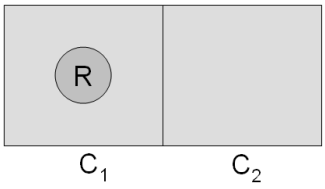


In computing the probabilities for the Bayes net we use the following Boolean table with corresponding probabilities for each row:

A	B	y	P
0	0	0	0.6*0.5=0.3
0	1	0	0.6*0.5=0.3
1	0	0	0.4*0.5=0.2
1	1	1	0.4*0.5=0.2

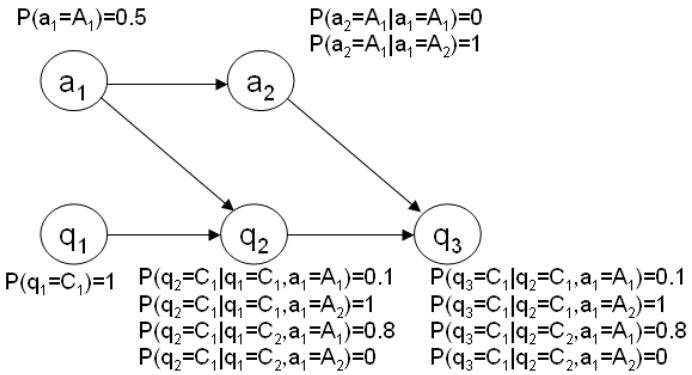
Using the table we can compute the probabilities for the Bayes net: $P(y) = 0.2$
$P(B|y) = \frac{P(B,y)}{P(y)} = 1$
$P(B|\neg y) = \frac{P(B,\neg y)}{P(\neg y)} = \frac{0.3}{0.8} = 0.375$
$P(A|y) = \frac{P(A,y)}{P(y)} = 1$
$P(A|\neg y) = \frac{P(A,\neg y)}{P(\neg y)} = \frac{0.2}{0.8} = 0.25$

(b) *(8 points)* Consider a robot operating in the two-cell gridworld shown below. Suppose the robot is initially in the cell C_1. At any point of time the robot can execute any of the two actions: A_1 and A_2. A_1 is "to move to a neighboring cell". If the robot is in C_1 the action A_1 succeeds (moves the robot into C_2) with the probability 0.9 and fails (leaves the robot in C_1) with the probability 0.1. If the robot is in C_2 the action A_1 succeeds (moves the robot into C_1) with the probability 0.8 and fails (leaves the robot in C_2) with the probability 0.2. The action A_2 is "to stay in the same cell", and when executed it keeps the robot in the same cell with probability 1. The first action the robot executes is chosen at random (with an equal probability between A_1 and A_2). Afterwards, the robot alternates the actions it executes. (for example, if the robot executed action A_1 first, then the sequence of actions is $A_1, A_2, A_1, A_2, \dots$). Answer the following questions.



(b.1) *(4 points)* Draw the Bayes net that represents the cell the robot is in during the first two actions the robot executes (e.g, initial cell, the cell after the first action and the cell after the second action) and fill in the probability tables. (Hint: The Bayes net should have five variables: q_1 - the initial cell, q_2, q_3 - the cell after the first and the second action, respectively, a_1, a_2 - the first and the second action, respectively).

Answer:



(b.2) *(4 points)* Suppose you were told that the first action the robot executes is A_1. What is the probability that the robot will appear in cell C_1 after it executes close to infinitely many actions?

Answer: Since actions alternate and the first action is A_1 the transition matrix for any odd action is:

$P(a_{odd}) = \begin{pmatrix} 0.1 & 0.9 \\ 0.8 & 0.2 \end{pmatrix}$,

where p_{ij} element is a probability of transitioning into cell j as a result of an execution of an odd action given that the robot is in cell i before executing this action.

Similarly, the transition matrix for any even action is: $P(a_{even}) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$.

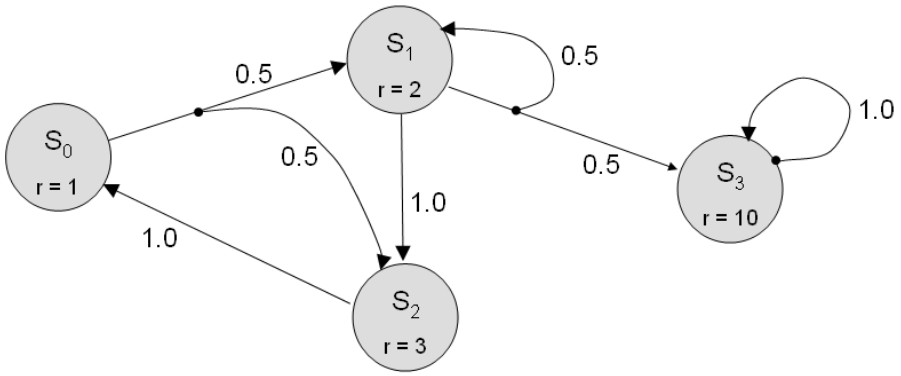
If we consider the pair of actions as one "meta-action", then we have a Markov chain with the transition probability matrix:

$P = P(a_{odd}) * P(a_{even}) = \begin{pmatrix} 0.1 & 0.9 \\ 0.8 & 0.2 \end{pmatrix}$.

At $t = \infty$, the state distribution satisfies $P(q_t) = P^T * P(q_t)$. So,
$P(q_t = C_1) = 0.1 * P(q_{t-1} = C_1) + 0.8 * P(q_{t-1} = C_2)$.
Since there are only two cells possible we have:
$P(q_t = C_1) = 0.1 * P(q_{t-1} = C_1) + 0.8 * (1 - P(q_{t-1} = C_1))$.
Solving for $P(q_t = C_1)$ we get:
$P(q_t = C_1) = 0.8/1.7 = 0.4706$.

Problem 9. Markov Decision Processes (11pts)

(a) *(8 points)* Consider the MDP given in the figure below. Assume the discount factor $\gamma = 0.9$. The r-values are rewards, while the numbers next to arrows are probabilities of outcomes. Note that only state S_1 has two actions. The other states have only one action for each state.



(a.1) *(4 points)* Write down the numerical value of $J(S_1)$ after the first and the second iterations of Value Iteration.

Initial value function: $J^0(S_0) = 0$; $J^0(S_1) = 0$; $J^0(S_2) = 0$; $J^0(S_3) = 0$;

$J^1(S_1) =$

$J^2(S_1) =$

Answer:

$J^1(S_1) = 2$

$$\begin{aligned} J^2(S_1) &= \max(2 + 0.9(0.5 * J^1(S_1) + 0.5 * J^1(S_3)), 2 + 0.9 * J^1(S_2)) \\ &= \max(2 + 0.9(0.5 * 2 + 0.5 * 10), 2 + 0.9 * 3) \\ &= 7.4 \end{aligned}$$

(a.2) *(4 points)* Write down the optimal value of state S_1. There are few ways to solve it, and for one of them you may find useful the following equality: $\sum_{i=0}^{\infty} \alpha^i = \frac{1}{1-\alpha}$ for any $0 \leq \alpha < 1$.

$J^*(S_1) =$

Answer:

It is pretty clear from the given MDP that the optimal policy from S_1 will involve trying to move from S_1 to S_3 as this is the only state that has a large reward. First, we compute optimal value for S_3:
$J^*(S_3) = 10 + 0.9 * J^*(S_3)$
$J^*(S_3) = \frac{10}{0.1} = 100$
We can now compute optimal value for S_1:
$J^*(S_1) = 2 + 0.9(0.5 * J^*(S_1) + 0.5 * J^*(S_3)) = 2 + 0.9(0.5 * J^*(S_1) + 50);$
Solving for $J^*(S_1)$ we get:
$J^*(S_1) = \frac{47}{0.55} = 87.\overline{45}$

(b) *(3 points)* A general MDP with N states is guaranteed to converge in the limit for Value Iteration as long as $\gamma < 1$. In practice one cannot perform infinitely many value iterations to guarantee convergence. Circle **all** the statements below that are **true**.

- (1) Any MDP with N states converges after N value iterations for $\gamma = 0.5$;

  Answer: False

- (2) Any MDP converges after the 1st value iteration for $\gamma = 1$;
  Answer: False

- (3) Any MDP converges after the 1st value iteration for a discount factor $\gamma = 0$;

  Answer: True, since all the converged values will be just immediate rewards.

- (4) An acyclic MDP with N states converges after N value iterations for any $0 \leq \gamma \leq 1$.

  Answer: True, since there are no cycles and therefore after each iteration at least one state whose value was not optimal before is guaranteed to have its value set to an optimal value (even when $\gamma = 1$), unless all state values are already converged.

- (5) An MDP with N states and no stochastic actions (that is, each action has only one outcome) converges after N value iterations for any $0 \leq \gamma < 1$.

  Answer: False. Consider a situation where there are no absorbing goal states.

- (6) One usually stops value iterations after iteration $k+1$ if: $\max_{0 \leq i \leq N-1} |J^{k+1}(S_i) - J^k(S_i)| < \xi$, for some small constant $\xi > 0$.

  Answer: True.

10-701/15-781, Fall 2006, Final

Dec 15, 5:30pm-8:30pm

- There are 9 questions in this exam (15 pages including this cover sheet).
- If you need more room to work out your answer to a question, use the back of the page and clearly mark on the front of the page if we are to look at what's on the back.
- This exam is open book and open notes. Computers, PDAs, cell phones are not allowed.
- You have 3 hours. Best luck!

Name:
Andrew ID:

Q	Topic	Max. Score	Score
1	Short Questions	20	
2	Instance-Based Learning	7	
3	Computational Learning Theory	9	
4	Gaussian Mixture Models	10	
5	Bayesian Networks	10	
6	Hidden Markov Models	12	
7	Dimensionality Reduction	8	
8	Graph-Theoretic Clustering	8	
9	MDPs and Reinforcement Learning	16	
Total		100	

1 Short Questions (20pts, 2pts each)

(a) **True or False** The ID3 algorithm is guaranteed to find the optimal decision tree.

(b) **True or False** Consider a continuous probability distribution with density $f()$ that is nonzero everywhere. The probability of a value x is equal to $f(x)$.

(c) **True or False**. In a Bayesian network, the inference results of the junction tree algorithm are the same as the inference results of variable elimination.

(d) **True or False** If two random variable X and Y are conditionally independent given another random variable Z, then in the corresponding Bayesian network, the nodes for X and Y are d-separated given Z.

(e) **True or False**. Besides EM, gradient descent can be used to perform inference or learning on a Gaussian mixture model.

(f) In one sentence, characterize the differences between *maximum likelihood* and *maximum a posteriori* approaches.

maximum likelihood finds parameters to maximizes the likelihood function, while MAP maximizes the posterior probability

(g) In one sentence, characterize the differences between classification and regression.

classification maps inputs to discrete outputs
regression maps inputs to continuous outputs

(h) Give one similarity and one difference between feature selection and PCA.

similarity: reduce the dimension of data
difference: feature selection finds a subset of features, while PCA produces a smaller, new set

(i) Give one similarity and one difference between HMM and MDP.

similarity: Marov assumptions
difference: The Markov chain in HMM is hidden; in MDP, the states are fully observed

(j) For each of the following datasets, is it appropriate to use HMM? Provide a brief reasoning for your answer.

- (•) Gene sequence dataset.
- A database of movie reviews (eg., the IMDB database).
- (•) Stock market price dataset.
- (•) Daily precipitation data from the Northwest of the US.

Time-series data; Markov assumption may be reasonable

2 Instance-Based Learning (7pts)

1. Consider the following training set in the 2-dimensional Euclidean space:

x	y	Class
-1	1	$-$
0	1	$+$
0	2	$-$
1	-1	$-$
1	0	$+$
1	2	$+$
2	2	$-$
2	3	$+$

Figure 1 shows a visualization of the data.

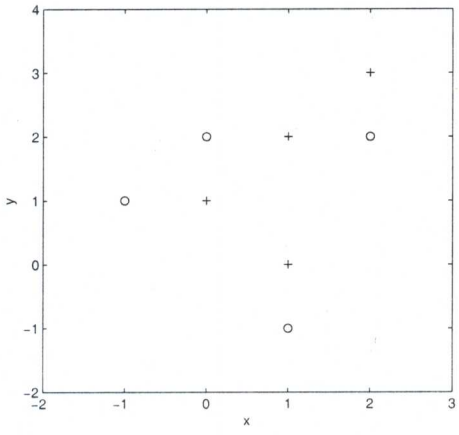


Figure 1: Dataset for Problem 2

(a) (1pt) What is the prediction of the 3-nearest-neighbor classifier at the point (1,1)?

+

(b) (1pt) What is the prediction of the 5-nearest-neighbor classifier at the point (1,1)?

+

(c) (1pt) What is the prediction of the 7-nearest-neighbor classifier at the point (1,1)?

−

2. Consider the two-class classification problem. At a data point x, the true conditional probability of a class $k, k \in \{0, 1\}$ is $p_k(x) = P(C = k|X = x)$.

   (a) (2pts) The Bayes error is the probability that an optimal Bayes classifier will misclassify a randomly drawn example. In terms of $p_k(x)$, what is the Bayes error E^* at x?

   $$1 - \max_{k \in \{0,1\}} P_k(x)$$

   or

   $$\min_{k \in \{0,1\}} P_k(x)$$

   (b) (2pts) In terms of $p_k(x)$ and $p_k(x')$ when x' is the nearest neighbor of x, what is the 1-nearest-neighbor error $E_{1\text{NN}}$ at x?

   $$P_0(x)P_1(x') + P_0(x')P_1(x)$$

Note that asymptotically as the number of training examples grows, $E^* \leq E_{1\text{NN}} \leq 2E^*$.

3 Computational Learning Theory (9pts, 3pts each)

In class we discussed different formula to provide a bound on the number of training examples sufficient for successful learning under different learning models.

$$m \geq \frac{1}{\epsilon}(\ln(1/\delta) + \ln|H|) \tag{1}$$

$$m \geq \frac{1}{2\epsilon^2}(\ln(1/\delta) + \ln|H|) \tag{2}$$

$$m \geq \frac{1}{\epsilon}(4\log_2(2/\delta) + 8VC(H)\log_2(13/\epsilon)) \tag{3}$$

Pick the appropriate one of the above formula to estimate the number of training examples needed for the following machine learning tasks. Briefly explain your choice.

1. Consider instances X containing 5 Boolean variables, $\{X_1, X_2, X_3, X_4, X_5\}$, and responses Y are $(X_1 \wedge X_4) \vee (X_2 \wedge X_3)$. We try to learn the function $f : X \rightarrow Y$ using a 2-layered neural network.

(3). $|H| = \infty$, $Y \in H$.

2. Consider instances X containing 5 Boolean variables, $\{X_1, X_2, X_3, X_4, X_5\}$, and responses Y are $(X_1 \wedge X_4) \vee (X_2 \wedge X_3)$. We try to learn the function $f : X \rightarrow Y$ using a "depth-2 decision trees". A "depth-2 decision tree" is a tree with four leaves, all distance 2 from the root.

(2). $|H| < \infty$, $Y \notin H$.

3. Consider instances X containing 5 Boolean variables, $\{X_1, X_2, X_3, X_4, X_5\}$, and responses Y are $(X_1 \wedge X_4) \vee (\neg X_1 \wedge X_3)$. We try to learn the function $f : X \rightarrow Y$ using a "depth-2 decision trees". A "depth-2 decision tree" is a tree with four leaves, all distance 2 from the root.

(1). $|H| < \infty$, $Y \in H$.

4 Gaussian Mixture Model (10pts)

Consider the labeled training points in Figure 2, where '+' and 'o' denote positive and negative labels, respectively. Tom asks three students (Yifen, Fan and Indra) to fit Gaussian Mixture Models on this dataset.

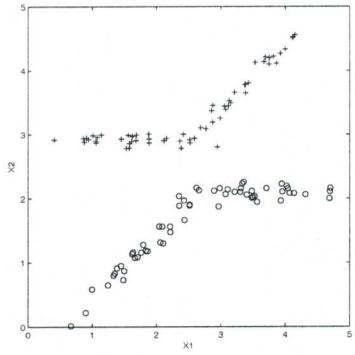
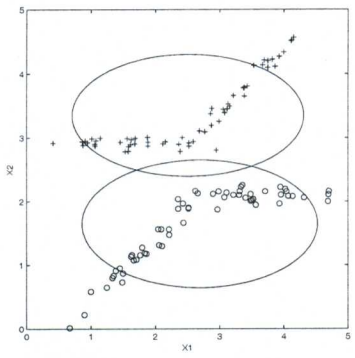
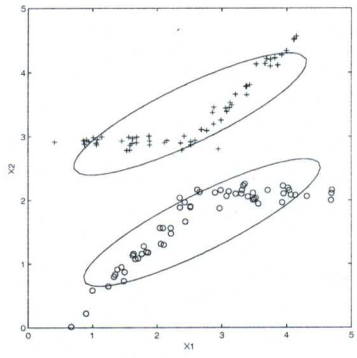


Figure 2: Dataset for Gaussian Mixture Model

1. (4pts) Yifen and Fan decide to use one Gaussian distribution for positive examples and one distribution for negative examples. The darker ellipse indicates the positive Gaussian distribution contour and the lighter ellipse indicates the negative Gaussian distribution contour.



Yifen's model

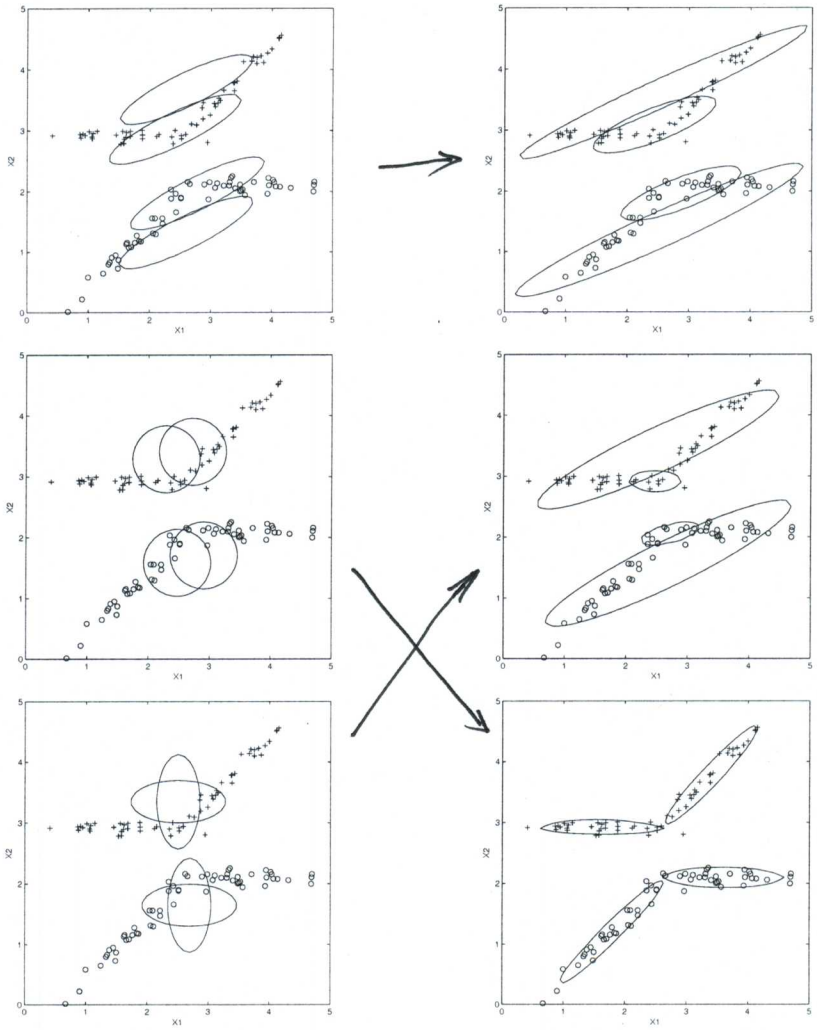


Fan's model

Whose model would you prefer for this dataset? What causes the difference between these two models?

Fan's model.
Yifen's model constrains the covariance matrixes to be diagonal, while Fan's model does not.

2. (6pts) Indra decides to use two Gaussian distributions for positive examples and two Gaussian distributions for negative examples. He uses EM algorithm to iteratively update parameters and also tries different initializations. The left column of Figure 3 shows 3 different initializations and the right column shows 3 possible models after the first iteration. For each initialization on the left, draw an arrow to the model on the right that will result after the first EM iteration. Your answer should consist of 3 arrows, one from each initialization.

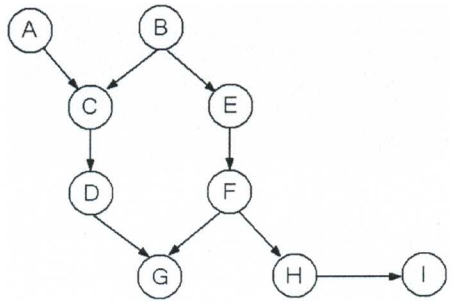


(a) Initialization (b) After first iteration

Figure 3: Three different initializations and models after the first iteration.

5 Bayesian Networks (10pts)

The figure below shows a Bayesian network with 9 variables, all of which are binary.



1. (3pts) Which of the following statements are always true for this Bayes net?
   (a) $P(A, B|G) = P(A|G)P(B|G)$;
   (b) $P(A, I) = P(A)P(I)$;
   (c) $P(B, H|E, G) = P(B|E, G)P(H|E, G)$;
   (d) $P(C|B, F) = P(C|F)$.

2. (2pts) What is the number of independent parameters in this graphical model?

   20

3. (3pts) The computational complexity of a graph elimination algorithm is determined by the size of the maximal elimination clique produced in the elimination process. What is the minimum size of such maximal elimination clique when we choose a perfect elimination order to compute $P(C = 1)$ using the graph elimination algorithm?

   3

4. (2pts) We would like to compute

$$\mu = \frac{P(F = 1|A, B, C, D, E, G, H, I)}{P(F = 0|A, B, C, D, E, G, H, I)}$$

   The value of μ depends on the values of all the variables other than F. What is the maximum possible number of different values of μ?

   16

   *Given the value of μ, as in the setting of Gibbs sampling, we could draw the random variable F from a Bernoulli distribution: $F \sim \text{Bernoulli}[1/(1 + \mu^{-1})]$.

6 Hidden Markov Models (12pts)

Consider an HMM with states $Y_t \in \{S_1, S_2, S_3\}$, observations $X_t \in \{A, B, C\}$, and parameters

$\pi_1 = 1$	$a_{11} = 1/2$	$a_{12} = 1/4$	$a_{13} = 1/4$	$b_1(A) = 1/2$	$b_1(B) = 1/2$	$b_1(C) = 0$
$\pi_2 = 0$	$a_{21} = 0$	$a_{22} = 1/2$	$a_{23} = 1/2$	$b_2(A) = 1/2$	$b_2(B) = 0$	$b_2(C) = 1/2$
$\pi_3 = 0$	$a_{31} = 0$	$a_{32} = 0$	$a_{33} = 1$	$b_3(A) = 0$	$b_3(B) = 1/2$	$b_3(C) = 1/2$

(a) (3pts) What is $P(Y_5 = S_3)$?

$$1 - P(Y_5 = S_1) - P(Y_5 = S_2)$$
$$= 1 - \frac{1}{16} - 4 \times \frac{1}{32}$$
$$= \frac{13}{16}$$

For 6(b)-(d), suppose we observe $AABCABC$, starting at time point 1.

(b) (2pts) What is $P(Y_5 = S_3 | X_{1:7} = AABCABC)$?

0

(c) (4pts) Fill in the following table assuming the observation $AABCABC$. The α's are values obtained during the forward algorithm: $\alpha_t(i) = P(X_1, \ldots, X_t, Y_t = i)$.

t	$\alpha_t(1)$	$\alpha_t(2)$	$\alpha_t(3)$
1	$\frac{1}{2}$	0	0
2	$\frac{1}{8}$	$\frac{1}{16}$	0
3	$\frac{1}{32}$	0	$\frac{1}{32}$
4	0	$\frac{1}{2^8}$	$\frac{5}{2^8}$
5	0	$\frac{1}{2^{10}}$	0
6	0	0	$\frac{1}{2^{12}}$
7	0	0	$\frac{1}{2^{13}}$

(d) (3pts) Write down the sequence of $Y_{1:7}$ with the maximal posterior probability assuming the observation $AABCABC$. What is that posterior probability?

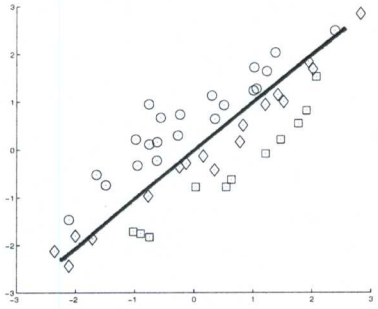
$S_1 S_1 S_1 S_2 S_2 S_3 S_3$

posterior probability : 1

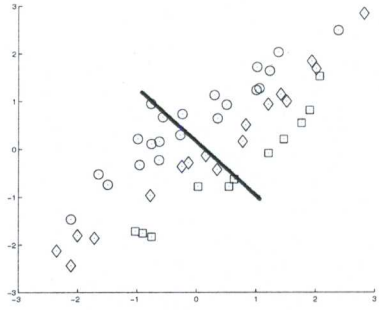
7 Dimensionality Reduction (8pts)

In this problem four linear dimensionality reduction methods will be discussed. They are principal component analysis (PCA), linear discriminant analysis (LDA), canonical correlation analysis (CCA), non-negative matrix factorization (NMF).

1. (3pts) LDA reduces the dimensionality given labels by *maximizing the overall interclass variance relative to intraclass variance.* Plot the directions of the *first* PCA and LDA components in the following figures respectively.

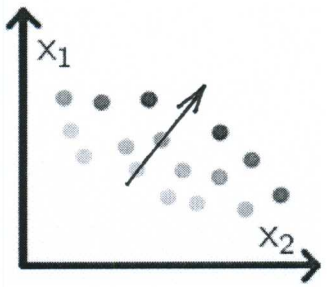


1(a) First PCA component

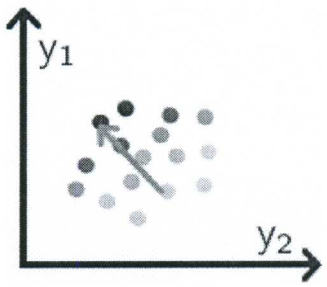


1(b) First LDA component

2. (2pts) In practice, each data point may have multiple vector-valued properties, e.g. a gene has its expression levels as well as the position on the genome. The goal of CCA is to reduce the dimensionality of the properties jointly. Suppose we have data points with two properties $\mathbf{x}$ and $\mathbf{y}$, each of which is a 2-dimension vector. This 4-dimensional data is shown in the pair of figures below; different data points are shown in different gray scales. CCA *finds* $(\mathbf{u}, \mathbf{v})$ *to maximize the correlation* $\widehat{corr}(\mathbf{u}^T\mathbf{x})(\mathbf{v}^T\mathbf{y})$. In figure 2(b) we have given the direction of vector $\mathbf{v}$, plot the vector $\mathbf{u}$ in figure 2(a).

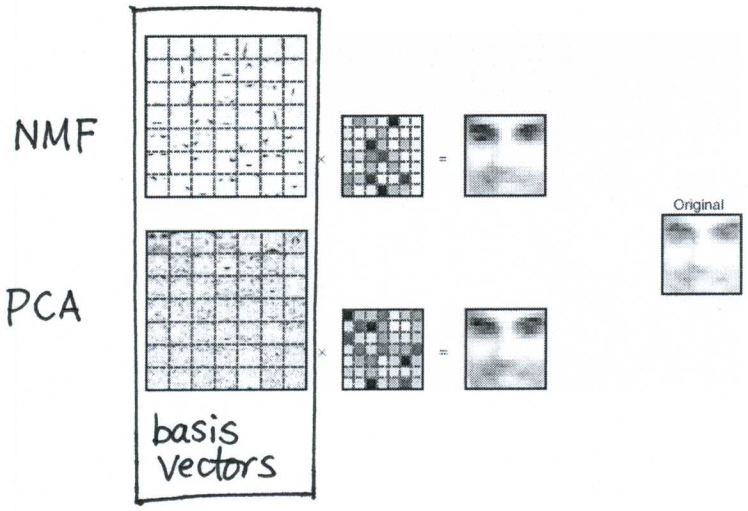


2(a)



2(b)

3. (3pts) The goal of NMF is to reduce the dimensionality given non-negativity constraints. That is, we would like to find principle components $\mathbf{u}_1, \ldots, \mathbf{u}_r$, each of which is of dimension $d > r$, such that the d-dimensional data $\mathbf{x} \approx \sum_{i=1}^{r} z_i \mathbf{u}_i$, and all entries in $\mathbf{x}, \mathbf{z}, \mathbf{u}_{1:r}$ are non-negative. NMF tends to find sparse (usually small L1 norm) basis vectors $\mathbf{u}_i$'s . Below is an example of applying PCA and NMF on a face image. Please point out the basis vectors in the equations and give them correct labels (NMF or PCA).



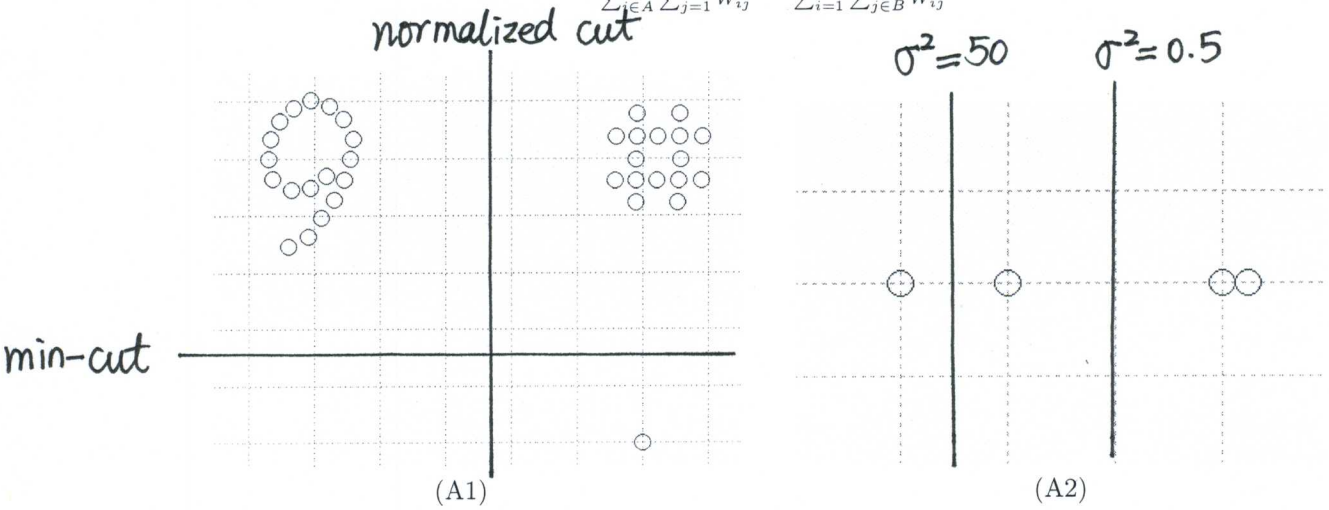
(* Figures in 7-2, 7-3 are originally from `http://www.eecs.berkeley.edu/~asimma/294-fall06/lectures/dimension/talk-maximal-1x2.pdf`.)

8 Graph-Theoretic Clustering (8pts)

Part A. Min-Cut and Normalized Cut

In this problem, we consider the 2-clustering problem, in which we have N data points $\mathbf{x}_{1:N}$ to be grouped in two clusters, denoted by A and B. Given the N by N affinity matrix W,

- Min-Cut: minimizes $\sum_{i\in A}\sum_{j\in B} W_{ij}$;
- Normalized Cut: minimizes $\frac{\sum_{i\in A}\sum_{j\in B} W_{ij}}{\sum_{i\in A}\sum_{j=1}^{N} W_{ij}} + \frac{\sum_{i\in A}\sum_{j\in B} W_{ij}}{\sum_{i=1}^{N}\sum_{j\in B} W_{ij}}$.



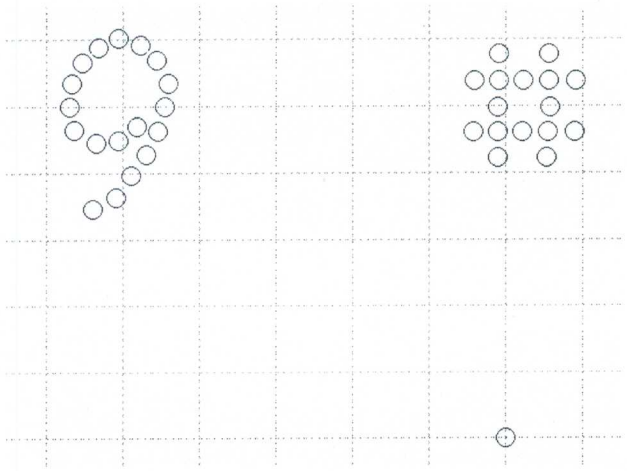
(A1) (A2)

A1. (2pts) The data points are shown in Figure (A1) above. The grid unit is 1. Let $W_{ij} = e^{-\|\mathbf{x}_i-\mathbf{x}_j\|_2^2}$, give the clustering results of min-cut and normalized cut respectively (You may show your work in the figure directly).

A2. (2pts) The data points are shown in Figure (A2) above. The grid unit is 1. Let $W_{ij} = e^{\frac{-\|\mathbf{x}_i-\mathbf{x}_j\|_2^2}{2\sigma^2}}$, describe the clustering results of min-cut algorithm for $\sigma^2 = 50$ and $\sigma^2 = 0.5$ respectively.

Part B. Spectral Clustering

Now back to the setting of the 2-clustering problem A1. The grid unit is 1.



B1. (2pts) If we use Euclidean distance to construct the affinity matrix W as follows:

$$W_{ij} = \begin{cases} 1 & \text{if } \| \mathbf{x}_i - \mathbf{x}_j \|_2^2 \leq \sigma^2 \\ 0 & \text{otherwise} \end{cases}$$

What σ^2 value would you choose? Briefly explain.

$\sigma^2 = 9 \sim 16$. W_{ij} should be 1 for every pair of points within "9", "#"; it should be 0 for other ~~cases~~ cases.

B2. (2pts) The next step is to compute the $k = 2$ dominant eigenvectors of the affinity matrix W. For the value of σ^2 you chose in the previous question, can you compute analytically eigenvalues corresponding to the first two eigenvectors? If yes, compute and report the eigenvalues. If not, briefly explain.

$$W = \begin{bmatrix} 1_{18\times 18} & 0 & 0 \\ 0 & 1_{16\times 16} & \\ 0 & 0 & 1 \end{bmatrix}$$

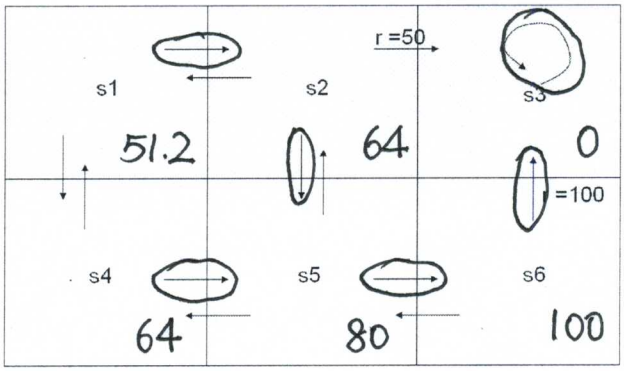
First two eigenvalues: 18, 16.

B3. *(1 Extra Credit, please try this question after you finished others!)
Suppose the data is of very high dimension so that it is impossible to visualize them and pick a good value as we did in Part B1. Suggest a heuristic that could find an appropriate σ^2.

9 MDPs and Reinforcement Learning [16pts]

Part A. [10pts]

Consider the following deterministic Markov Decision Process (MDP), describing a simple robot grid world. Notice the values of the *immediate rewards* are written next to transitions. Transitions with no value have an immediate reward of 0. **Assume the discount factor** $\gamma = 0.8$.



A1. (2pts) For each state s, write the value for $V^*(s)$ inside the corresponding square in the diagram.

A2. (2pts) Mark the state-action transition arrows that correspond to one *optimal* policy. If there is a tie, always choose the state with the smallest index.

A3. (2pts) Give a different value for γ which results in a different optimal policy and the number of changed policy actions should be minimal. Give your new value for γ, and describe the resulting policy by indicating which $\pi(s)$ values (i.e., which policy actions) change.

New value for γ: 0.7

Changed policy actions: $\pi(S_2) = S_3$

For the remainder of this question, assume again that $\gamma = 0.8$.

A4. (2pts) How many complete loops (iterations) of value iteration are sufficient to guarantee finding the optimal policy for this MDP? Assume that values are initialized to zero, and that states are considered in an arbitrary order on each iteration.

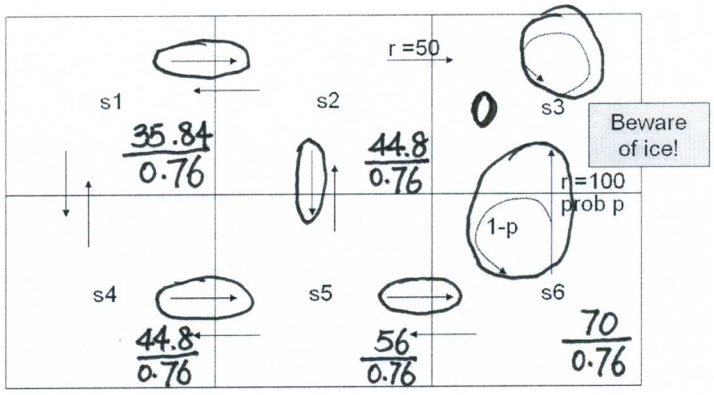
4

A5. (2pts) Is it possible to change the immediate reward function so that V^* changes but the optimal policy π^* remains unchanged? If yes, give such a change, and describe the resulting change to V^*. Otherwise, explain in at most 2 sentences why this is impossible.

Yes. Double each immediate reward. Then V^* is also doubled; π^* remains unchanged.

Part B. (6pts)

It is December. Unfortunately for our robot, a patch of ice has appeared in its world, making one of its actions non-deterministic. The resulting MDP is shown below. Note that now the result of the action "go north" from state $s6$ results in one of two outcomes. With probability p the robot succeeds in transitioning to state $s3$ and receives immediate reward 100. However, with probability $(1-p)$ it slips on the ice, and remains in state $s6$ with zero immediate reward. **Assume the discount factor** $\gamma = 0.8$.



B1. (4pts) Assume $p = 0.7$. Write in the values of V^* for each state, and circle the actions in the optimal policy.

$$V_6^* = 100p + \gamma(1-p)V_6^*$$

B2. (2pts) How bad does the ice have to get before the robot will prefer to completely avoid it? Answer this question by giving a value for p below which the optimal policy chooses actions that completely avoid the ice, even choosing the action "go west" over "go north" when the robot is in state s6.

$$50\gamma^2 = V_6^* = \frac{100p}{1-\gamma(1-p)}$$

$$p = \frac{\gamma^2-\gamma^3}{2-\gamma^3} = \frac{8}{93}$$

10-701 Final Exam, Spring 2007

1. Personal info:
   - Name:
   - Andrew account:
   - E-mail address:
2. There should be 16 numbered pages in this exam (including this cover sheet).
3. You can use any material you brought: any book, class notes, your print outs of class materials that are on the class website, including my annotated slides and relevant readings, and Andrew Moore's tutorials. You cannot use materials brought by other students. Calculators are allowed, but no laptops, PDAs, phones or Internet access.
4. If you need more room to work out your answer to a question, use the back of the page and clearly mark on the front of the page if we are to look at what's on the back.
5. Work efficiently. Some questions are easier, some more difficult. Be sure to give yourself time to answer all of the easy ones, and avoid getting bogged down in the more difficult ones before you have answered the easier ones.
6. Note there are extra-credit sub-questions. The grade curve will be made without considering students' extra credit points. The extra credit will then be used to try to bump your grade up without affecting anyone else's grade.
7. You have 180 minutes.
8. Good luck!

Question	Topic	Max. score	Score
1	Short questions	21 + 0.911 extra	
2	SVM and slacks	16	
3	GNB	8	
4	Feature Selection	10	
5	Irrelevant Features	14 + 3 extra	
6	Neural Nets	16 + 5 extra	
7	Learning theory	15	

1 [Points] Short Questions

The following short questions should be answered with at most two sentences, and/or a picture. For the **(true/false)** questions, answer true or false. If you answer true, provide a short justification, if false explain why or provide a small counterexample.

1. [points] Your billionaire friend needs your help. She needs to classify job applications into good/bad categories, and also to detect job applicants who lie in their applications using density estimation to detect outliers. To meet these needs, do you recommend using a discriminative or generative classifier? Why?

   for density estimation, need $p(x|y)$

2. [points] Your billionaire friend also wants to classify software applications to detect bug-prone applications using features of the source code. This pilot project only has a few applications to be used as training data, though. To create the most accurate classifier, do you recommend using a discriminative or generative classifier? Why?

   based on convergence properties and some experimental observations

3. [points] Finally, your billionaire friend also wants to classify companies to decide which one to acquire. This project has lots of training data based on several decades of research. To create the most accurate classifier, do you recommend using a discriminative or generative classifier? Why?

4. [points] Assume that we are using some classifier of fixed complexity. Draw a graph showing two curves: test error vs. the number of training examples and cross-validation

error vs. the number of training examples.

5. [points] Assume that we are using an SVM classifier with a Gaussian kernel. Draw a graph showing two curves: training error vs. kernel bandwidth and test error vs. kernel bandwidth

6. [points] Assume that we are modeling a number of random variables using a Bayesian Network with n edges. Draw a graph showing two curves: Bias of the estimate of the joint probability vs. n and variance of the estimate of the joint probability vs. n.

7. [points]

   (a) Both PCA and linear regression can be thought of as algorithms for minimizing a sum of squared errors. Explain which error is being minimized in each algorithm.

8. [points] A long time ago there was a village amidst hundreds of lakes. Two types of fish lived in the region, but only one type in each lake. These types of fish both looked exactly the same, smelled exactly the same when cooked, and had the exact same delicious taste - except one was poisonous and would kill any villager who ate it. The only other difference between the fish was their effect on the pH (acidity) of the lake they occupy. The pH for lakes occupied by the non-poisonous type of fish was distributed according to a Gaussian with unknown mean (μ_{safe}) and variance (σ^2_{safe})

and the pH for lakes occupied by the poisonous type was distributed according to a different Gaussian with unknown mean (μ_{deadly}) and variance (σ^2_{deadly}). (Poisonous fish tended to cause slightly more acidic conditions).

Naturally, the villagers turned to machine learning for help. However, there was much debate about the right way to apply EM to their problem. For each of the following procedures, indicate whether it is an accurate implementation of Expectation-Maximization and will provide a reasonable estimate for parameters μ and σ^2 for each class.

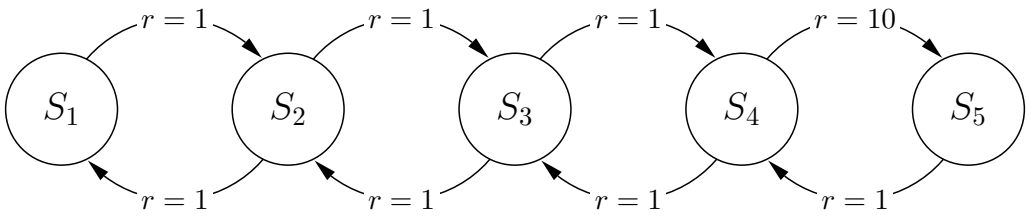
(a) Guess initial values of μ and σ^2 for each class. (1) For each lake, find the most likely class of fish for the lake. (2) Update the μ and σ^2 values using their maximum likelihood estimates based on these predictions. Iterate (1) and (2) until convergence.

(b) For each lake, guess an initial probability that it is safe. (1) Using these probabilities, find the maximum likelihood estimates for the μ and σ values for each class. (2) Use these estimates of μ and σ to reestimate lake safety probabilities. Iterate (1) and (2) until convergence.

(c) Compute the mean and variance of the pH levels across all lakes. Use these values for the μ and σ^2 value of each class of fish. (1) Use the μ and σ^2 values of each class to compute the belief that each lake contains poisonous fish. (2) Find the maximum likelihood values for μ and σ^2. Iterate (1) and (2) until convergence.

2 [points] Reinforcement Learning

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.* $(S_3, Left), (S_2, Right), (S_3, Right), \ldots$)?

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.* $(S_3, Left), (S_2, Right), (S_3, Right), \ldots$)?

3 [Points] Bayes Net Structure Learning

Finding the most likely Bayes Net structure given data is generally intractable. However, if certain restrictions are imposed on the structure, the most likely one can be found efficiently. One such restriction imposes a fixed ordering on the variables of the Bayes Net. This ordering restricts all edges to be directed forward in the ordering. For example, an edge $X \rightarrow Y$ can only exist if X comes before Y in the ordering.

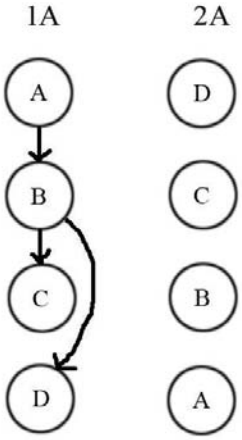
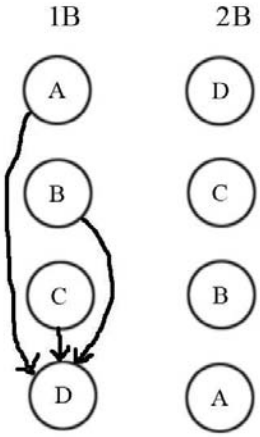
1. We'll now explore the effect that the ordering has on the number of parameters and independence assumptions of Bayes Nets. In each box you are given a Bayes Net that obeys a fixed ordering ABCD (1A and 1B).

   Draw a Bayes Net (part 2A) for the fixed ordering DCBA that can model the same distribution as the Bayes Net of part 1A. It should have no additional independence assumptions that are not present in part 1A, but also no unnecessary edges. Repeat for 1B and 2B.

   Count the number of parameters in each Bayes Net. Each variable is **binary** - it can take on 2 values.

   Identify an independence assumption of Bayes Net 1A that doesn't exist in Bayes Net 2A, if such an independence assumption exists. Repeat for Bayes Nets 1B and 2B.

   Hint: Pay close attention to V-structures - both existing ones and ones you create!!!

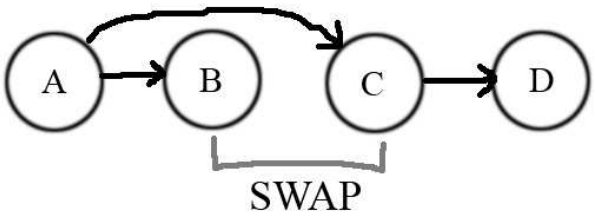
1A 2A	1B 2B
	
Number of parameters for Bayes Net 1A _____ Number of parameters for Bayes Net 2A _____ List an independence assumption of 1A not present in 2A (if there is one) _____	Number of parameters for Bayes Net 1B _____ Number of parameters for Bayes Net 2B _____ List an independence assumption of 1B not present in 2B (if there is one) _____

2. Given a fixed ordering over variables: $X_1, X_2, X_3, ..., X_n$, show that the choice of parents π_n is independent of the choice of other parents $\pi_1, ...\pi_{n-1}$. In other words, show that:

   $\max_{\pi_1,...,\pi_n} \log P(X_1, ..., X_n|\pi_1, ..., \pi_n) =$
   $\max_{\pi_n} f(X_1, ..., X_n, \pi_n) + \max_{\pi_1,...,\pi_{n-1}} g(X_1, ..., X_n, \pi_1, ..., \pi_{n-1})$

   for some functions f and g.

3. For fixed orderings with a limit of k on the number of parents for each node, the best structure can be obtained by combinatoric search. For each variable, all subsets of variables from earlier in the ordering of size k or less are considered. For each set, $\log P(child|parents)$ is computed. We saw in part 1 of this question that the ordering can change the number of parameters required to model the joint probability. In this question we'll consider the efficiency of modifying the ordering. This approach can be used to greedily search for a good ordering of variables.



Consider the scenario where you are given a fixed ordering and the most likely Bayes Net structure for that fixed ordering. We would like to find the most likely structure after we switch two **adjacent** variables in the ordering. How many calculations of $\log P(child|parent)$ would this require in the worst case? Explain.

Local swapping of variables is prone to getting stuck in local maxima. Instead, let's consider changing the fixed ordering so that the two variables we swap have j **variables in between**. How many $\log P(child|parent)$ calculations are required to find the most likely structure for this new ordering in the worst case? Explain.

4 [Points] Decision Trees

In class, we discussed greedy algorithms for learning decision trees from training data. These algorithms partition the feature space into labeled regions by greedily optimizing some metric (information gain) in hope of producing simple trees that partition the feature space into regions that perfectly classify the training data. As with most greedy approaches, if we consider finding a good tree with a limited depth, this approach is not guaranteed to produce the set of regions that best maximize this metric.

We can always be less greedy. Instead of greedily making one decision and then greedily making the next decision, we can consider the outcome of all possible pairs of those two decisions and choose the best of those. We'll now explore the benefits and costs of being less greedy.

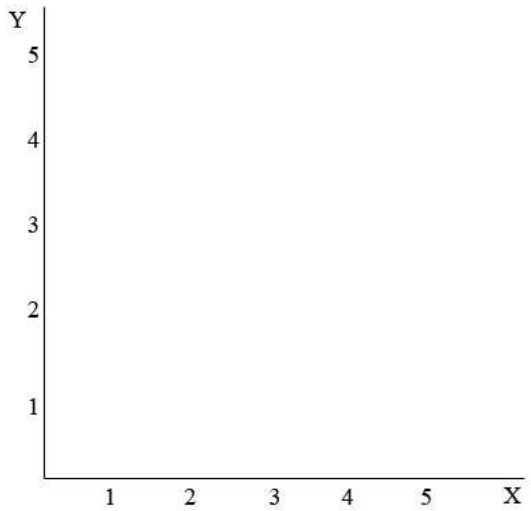
In a **standard decision tree**, each level of the recursion will find one decision boundary (e.g., X=3) that partitions the feature space into two regions (e.g., $X > 3$, $X \leq 3$) so to maximize the metric. Each region is then partitioned recursively using the same procedure.

In a **point-based look-ahead decision tree**, the feature space is partitioned into four regions by a single point (e.g., $X, Y = (3, 4)$ gives regions $[X > 3, Y > 4], [X > 3, Y \leq 4], [X \leq 3, Y > 4]$, and $[X \leq 3, Y \leq 4]$.

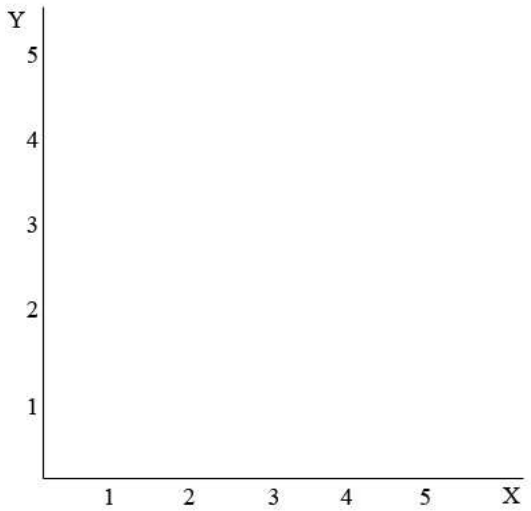
In a **boundary-based look-ahead decision tree**, three decision boundaries are considered in each level of the recursive decision-tree construction. The first decision boundary splits the feature space into two regions, and the two additional decision boundaries split those two regions for a total of 4 regions (e.g., $X = 3, Y = 4$ for $X < 3, Y = 2$ for $X > 3$) which yields regions $[X > 3, Y > 4], [X > 3, Y \leq 4], [X \leq 3, Y > 2]$, and $[X \leq 3, Y \leq 2]$.

1. Draw a dataset on the following 2 plots so that a standard decision tree with two levels (4 regions) will poorly classify the data, but a point-based look-ahead decision tree with one level (4 regions) will perfectly classify the data. Use '+' and '-' to indicate the class of each point and draw in the decision region boundaries of each decision tree.

Standard Decision Tree

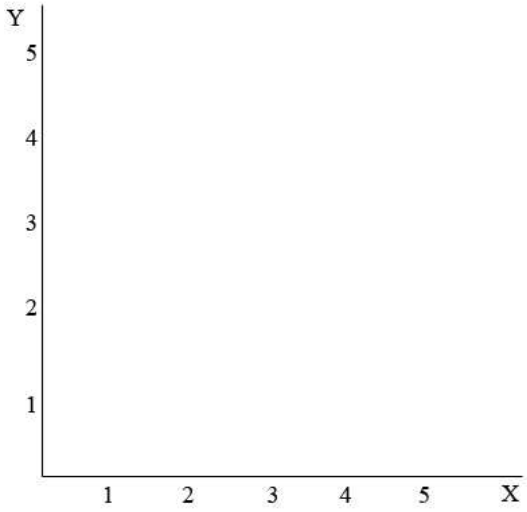


Point-based Look-ahead Decision Tree

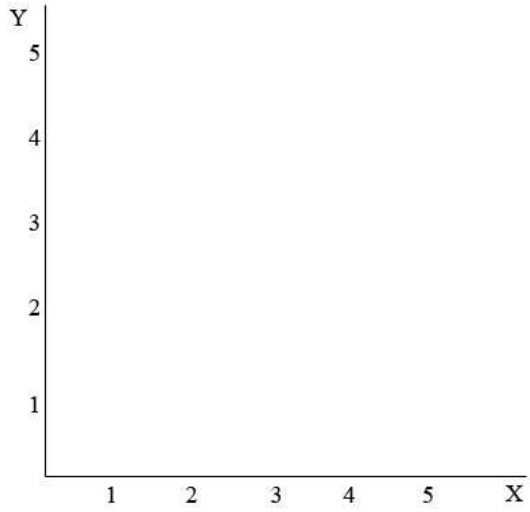


2. Now draw a dataset on the following 2 plots so that a point-based look-ahead decision tree with one level (4 regions) will poorly classify the data, but a boundary-based look-ahead decision tree with one level (4 regions) will perfectly classify the data. Use '+' and '-' to indicate the class of each point and draw in the decision region boundaries of each decision tree.

Point-based Look-ahead Decision Tree



Boundary-based Look-ahead Decision Tree



3. Now provide the running time required for one level of the partitioning in the various decision tree variants. Assume there are D points in the training set all with unique X and Y values. Explain your reasoning.

Standard Decision Tree

Point-Based Look-ahead Decision Tree

Boundary-Based Look-ahead Decision Tree

5 Neural Networks

Recall the two types of Neural Network activation functions from Homework 2, the linear activation function and the hard threshold:

- linear $y = w_0 + \sum_i w_i x_i$,
- hard threshold

$$y = \begin{cases} 1 & \text{if } w_0 + \sum_i w_i x_i \geq 0, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

1. Which of the following functions can be exactly represented by a neural network with one hidden layer which uses linear and/or hard threshold activation functions? For each case, justify your answer.

   (a) polynomials of degree one

   (b) hinge loss (h(x) = max(1-x,0))

   (c) polynomials of degree two

   (d) piecewise constant functions

6 [points] VC Dimentia

Given a hypothesis class $\mathcal{H}$, the VC dimension, $VC(\mathcal{H})$ is defined to be the size of the largest set that is shattered by $\mathcal{H}$. If $\mathcal{H}$ can shatter arbitrarily large sets, then we say that $VC(\mathcal{H}) = \infty$.

1. It is sometimes useful to think of VC dimension as being related to the number of parameters needed to specify an element of $\mathcal{H}$. For example, what is the VC dimension of the set of hypotheses of the following form?

$$h_\alpha(x) = \begin{cases} 1 & \text{if } \alpha_d x^d + \alpha_{d-1} x^{d-1} + \cdots + \alpha_0 > 0 \\ 0 & \text{otherwise} \end{cases}$$

   Justify your answer.

   Hint: think polynomial basis functions

2. Despite the result from part (1), the VC dimension is not always so nicely related to the number of parameters. For any positive integer M, can you come up with a hypothesis class which takes M parameters but has VC dimension 1?

   Hint: Think of how you might encode several parameters with just one parameter.

3. Consider the class of hypotheses of the form:

$$h_\alpha(x) = \begin{cases} 1 & \text{if } \sin(\alpha x) > 0 \\ 0 & \text{otherwise} \end{cases}$$

You will show that this one-parameter hypothesis class has infinite VC dimension.

To do this, show that given the datapoints $X = \{x_i = 10^{-i}, \quad i = 1, \ldots, n\}$, any set of labels $y_i \in \{0, 1\}$ can be realized by h_α by setting

$$\alpha = \left(1 + \sum_{i=1}^{n} (1 - t_i) 10^i\right) \cdot \pi$$

For example, if $n = 5$ and $y_i = (1, 1, 1, 1, 0)$, then $\alpha = (100001)\pi$.

Hint: On intervals of the form $(m\pi, (m+1)\pi)$, the sine function takes positive values if m is even and negative values if m is odd.

Machine Learning (10-701) Fall 2008

Final Exam

Professor: Eric Xing *Date: December 8, 2008*

. There are 9 questions in this exam (18 pages including this cover sheet)

. Questions are not equally difficult.

. This exam is open to book and notes. Computers, PDAs, Cell phones are not allowed.

. You have three hours.

. Good luck!

Last Name:			
First Name:			
Andrew ID:			
Q	Topic	Max. Score	Score
1	**Assorted Questions**	**20**	
2	**SVM**	**10**	
3	**PCA**	**10**	
4	**Linear Regression**	**12**	
5	**Sampling**	**8**	
6	**EM**	**10**	
7	**Learning Theory**	**10**	
8	**Hidden Markov Models**	**10**	
9	**Bayesian Networks**	**10**	
Total		100	

1 Assorted Questions [20 points]

1. (**True or False**, 2 pts) PCA and Spectral Clustering (such as Andrew Ng's) perform eigen-decomposition on two different matrices. However, the size of these two matrices are the same.

   Solutions: F

2. (**True or False**, 2 pts) The dimensionality of the feature map generated by polynomial kernel (e.g., $K(x, y) = (1 + x \cdot y)^d$)is polynomial wrt the power d of the polynomial kernel.

   Solutions: T

3. (**True or False**, 2 pts) Since classification is a special case of regression, logistic regression is a special case of linear regression.

   Solutions: F

4. (**True or False**, 2 pts) For any two variables x and y having joint distribution $p(x, y)$, we always have $H[x, y] \geq H[x] + H[y]$ where H is entropy function.

   Solutions: F

5. (**True or False**, 2 pts) The Markov Blanket of a node x in a graph with vertex set X is the smallest set Z such that $x \perp X/\{Z \cup x\}|Z$

   Solutions: T

6. (**True or False**, 2 pts) For some directed graphs, moralization decreases the number of edges present in the graph.

   Solutions: F

7. (**True or False**, 2 pts) The L_2 penalty in a ridge regression is equivalent to a Laplace prior on the weights.

   Solutions: F

8. (**True or False**, 2 pts) There is *at least one* set of 4 points in $\Re^3$ that can be shattered by

the hypothesis set of all 2D planes in $\Re^3$.

Solutions: T

9. (**True or False**, 2 pts) The log-likelihood of the data will *always* increase through successive iterations of the expectation maximation algorithm.

Solutions: F

10. (**True or False**, 2 pts) One disadvantage of Q-learning is that it can only be used when the learner has prior knowledge of how its actions affect its environment.

Solutions: F

2 Support Vector Machine(SVM) [10 pts]

1. **Properties of Kernel**

   1.1. (2 pts) Prove that the kernel $K(x_1, x_2)$ is symmetric, where x_i and x_j are the feature vectors for i^{th} and j^{th} examples.

   hints: Your proof will not be longer than 2 or 3 lines.

   Solutions: Let $\Phi(x_1)$ and $\Phi(x_2)$ be the feature maps for x_i and x_j, respectively. Then, we have $K(x_1, x_2) = \Phi(x_1)'\Phi(x_2) = \Phi(x_2)'\Phi(x_1) = K(x_2, x_1)$

   1.2. (4 pts) Given n training examples $(x_i, x_j)(i, j = 1, ..., n)$, the kernel matrix $\mathbf{A}$ is an $n \times n$ square matrix, where $\mathbf{A}(i, j) = K(x_i, x_j)$. Prove that the kernel matrix $\mathbf{A}$ is semi-positive definite.

   hints: (1) Remember that an $n \times n$ matrix $\mathbf{A}$ is semi-positive definite iff. for any n dimensional vector $\mathbf{f}$, we have $\mathbf{f'Af} \geq 0$. (2) For simplicity, you can prove this statement just for the following particular kernel function: $K(x_i, x_j) = (1 + x_i x_j)^2$.

   Solutions: Let $\Phi(x_i)$ be the feature map for the i^{th} example and define the matrix $\mathbf{B} = [\mathbf{\Phi(x_1)}, ..., \mathbf{\Phi(x_n)}]$. It is easy to verify that $\mathbf{A} = \mathbf{B'B}$. Then, we have $\mathbf{f'Af} = (\mathbf{Bf})'\mathbf{Bf} = \|\mathbf{Bf}\|^{\mathbf{2}} \geq \mathbf{0}$

2. **Soft-Margin Linear SVM**. Given the following dataset in 1-d space (Figure 1), which consists of 4 positive data points $\{0, 1, 2, 3\}$ and 3 negative data points $\{-3, -2, -1\}$. Suppose that we want to learn a soft-margin linear SVM for this data set. Remember that the soft-margin linear SVM can be formalized as the following constrained quadratic optimization problem. In this formulation, C is the regularization parameter, which balances the size of margin (i.e., smaller $w^t w$) vs. the violation of the margin (i.e., smaller $\sum_{i=1}^{m} \epsilon_i$).

$$\begin{aligned} \text{argmin}_{\{w,b\}} \ & \frac{1}{2} w^t w + C \sum_{i=1}^{m} \epsilon_i \\ \text{Subject to :} & y_i(w^t x_i + b) \geq 1 - \epsilon_i \\ & \epsilon_i \geq 0 \ \forall i \end{aligned}$$

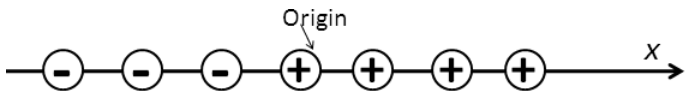


Figure 1: Dataset

2.1 (2 pts) if $C = 0$, which means that we only care the size of the margin, how many support vectors do we have?

Solutions: 7

2.2 (2 pts) if $C \to \infty$, which means that we only care the violation of the margin, how many support vectors do we have?

Solutions: 2

3 Principle Component Analysis (PCA) [10 pts]

1.1 (3 pts) **Basic PCA**

Given 3 data points in 2-d space, $(1,1)$, $(2,2)$ and $(3,3)$,

(a) (1 pt) what is the first principle component?

Solutions: $pc = (1/\sqrt{2}, 1/\sqrt{2})' = (0.707, 0.707)'$, (the negation is also correct)

(b) (1 pt) If we want to project the original data points into 1-d space by principle component you choose, what is the variance of the projected data?

Solutions: $4/3 = 1.33$

(c) (1 pt) For the projected data in (b), now if we represent them in the original 2-d space, what is the reconstruction error?

Solutions: 0

1.2 (7 pts) **PCA and SVD**

Given 6 data points in 5-d space, $(1,1,1,0,0)$, $(-3,-3,-3,0,0)$, $(2,2,2,0,0)$, $(0,0,0,-1,-1)$, $(0,0,0,2,2)$, $(0,0,0,-1,-1)$. We can represent these data points by a 6×5 matrix X, where each row corresponds to a data point:

$$X = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ -3 & -3 & -3 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & -1 & -1 \end{bmatrix}$$

(a) (1 pt) What is the sample mean of the data set?

Solutions: $[0,0,0,0,0]$

(b) (3 pts) What is SVD of the data matrix X you choose?

hints: The SVD for this matrix must take the following form, where $a, b, c, d, \sigma_1, \sigma_2$ are the parameters you need to decide.

$$X = \begin{bmatrix} a & 0 \\ -3a & 0 \\ 2a & 0 \\ 0 & b \\ 0 & -2b \\ 0 & b \end{bmatrix} \times \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \times \begin{bmatrix} c & c & c & 0 & 0 \\ 0 & 0 & 0 & d & d \end{bmatrix}$$

Solutions: $a = \pm 1/\sqrt{14} = \pm 0.267$, $b = \pm 1/\sqrt{6} = \pm 0.408$,
$\sigma_1 = 1/(a \cdot c) = \sqrt{42} = 6.48$, $\sigma_2 = 1/(b \cdot d) = \sqrt{12} = 3.46$,
$c = \pm 1/\sqrt{3} = \pm 0.577$,$d = \pm 1/\sqrt{2} = \pm 0.707$.

(c) (1 pt) What is first principle component for the original data points?

Solutions: pc=$\pm[c, c, c, 0, 0] = \pm[0.577, 0.577, 0.577, 0, 0]$ (Intuition: First, we want to notice that the first three data points are co-linear, and so do the last three data points. And also the first three data points are orthogonal to the rest three data points. Then, we want notice that the norm of the first three are much bigger than the last three, therefor, the first pc has the same direction as the first three data points)

(d) (1 pt) If we want to project the original data points into 1-d space by principle component you choose, what is the variance of the projected data?

Solutions: var=$\sigma_1^2/6 = 7$ (Intuition: we just the keep the first three data points, and set the rest three data points as $[0, 0, 0, 0, 0]$ (since they are orthogonal to pc), and then compute the variance among them)

(e) (1 pt) For the projected data in (d), now if we represent them in the original 5-d space, what is the reconstruction error?

Solutions: var=$\sigma_2^2/6 = 2$[1](Intuition, since the first three data points are orthogonal with the rest three, here the rerr is the just the sum of the norm of the last three data points (2+8+2=12), and then divided by the total number (6) of data points, if we use average definition

[1]if you give an answer var=$\sigma_2^2 = 12$, that is also correct. In this case, the reconstruction error is just the sum (not average) among all the data points, which is the definition used in Carlos' lecture notes. But in Bishop's book, he uses the average definition.

4 Linear Regression [12 Points]

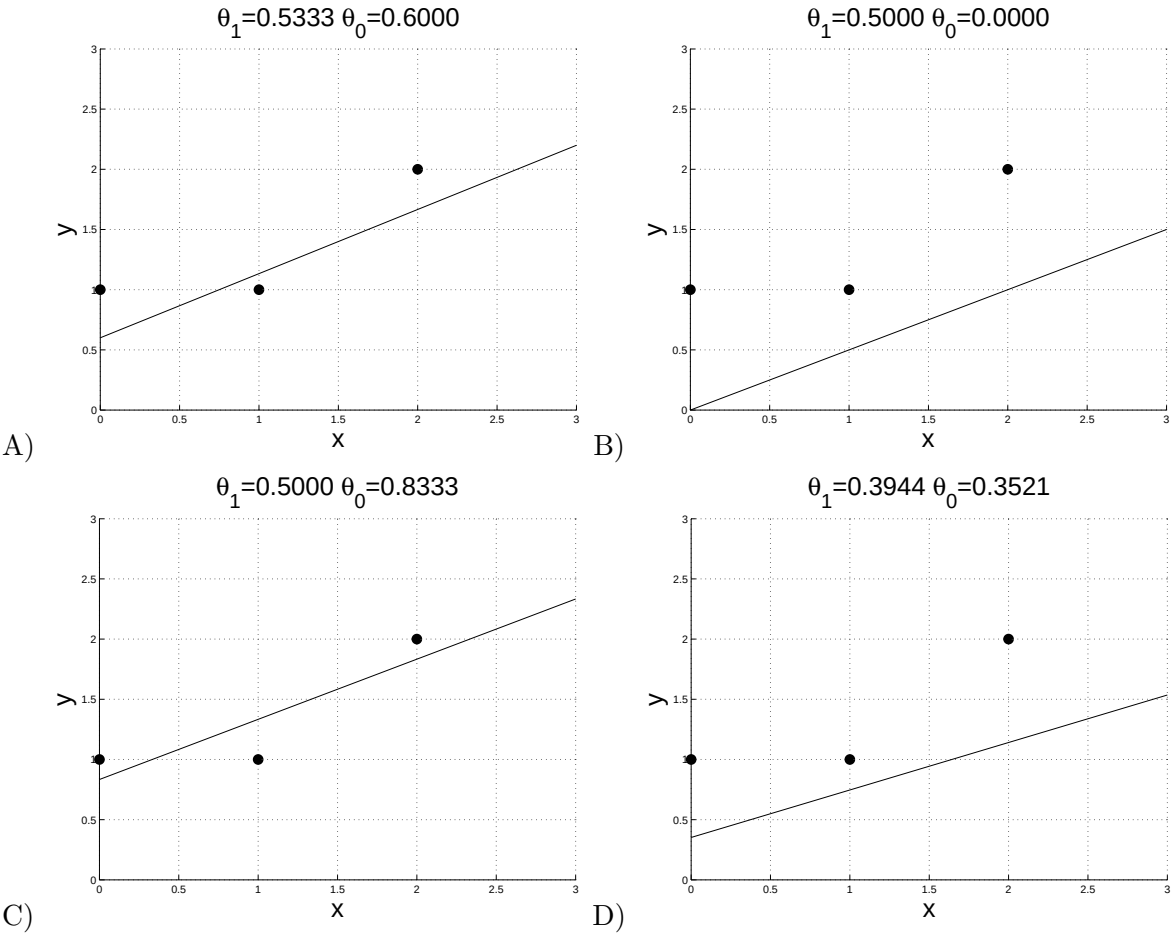


Figure 2: Plots of linear regression results with various regularization

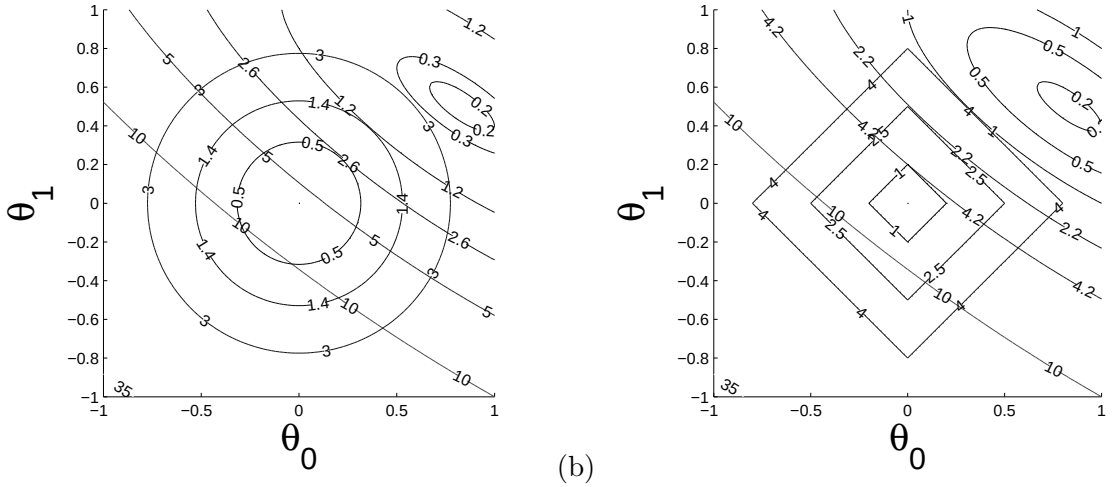
Background: In this problem we are working on linear regression with regularization on points in a 2-D space. Figure 2 plots linear regression results on the basis of three data points, (0,1), (1,1) and (2,2), with different regularization penalties.

As we all know, solving a linear regression problem is about to solve a minimization problem. That is to compute

$$\arg\min_{\theta_0,\theta_1} \sum_{i=1}^{n} (y_i - \theta_1 x_i - \theta_0)^2 + R(\theta_0, \theta_1)$$

where R represents a regularization penalty which could be L-1 or L-2. In this problem, $n = 3$, $(x_1, y_1) = (0, 1)$, $(x_2, y_2) = (1, 1)$, and $(x_3, y_3) = (2, 2)$. $R(\theta_0, \theta_1)$ could either be $\lambda(|\theta_1| + |\theta_0|)$ or $\lambda(\theta_1^2 + \theta_0^2)$.

However, in stead of computing the derivatives to get a minimum value, we could adopt a geometric method. In this way, rather than letting the square error term and the regularization penalty term vary simultaneously as a function of θ_0 and θ_1, we can fix one and only let the other vary at a time. Having a upper-bound, r, on the penalty, we can replace $R(\theta_0, \theta_1)$ by r, and solve a minimization



(a) (b)

Figure 3: Contour plots of the decomposition for the linear regression problem with (a) L-2 regularization or (b) L-1 regularization where the ellipsis correspond to the square error term, and circles/squares correspond to the regularization penalty term.

problem on the square error term for any non-negative value of r. Finally, we get the minimum value by enumerating over all possible value of r. That is,

$$\min_{\theta_0,\theta_1} \sum_{i=1}^{n} (y_i - \theta_1 x_i - \theta_0)^2 + R(\theta_0, \theta_1) = \min_{r \geq 0} \left\{ \min_{\theta_0,\theta_1} \left\{ \sum_{i=1}^{n} (y_i - \theta_1 x_i - \theta_0)^2 \mid R(\theta_0, \theta_1) \leq r \right\} + r \right\}$$

. The value of (θ_0, θ_1) corresponding to the minimum value of the object function can be got at the same time.

In Figure 3, we plot the square error term, $\sum_{i=1}^{n} (y_i - \theta_1 x_i - \theta_0)^2$, by ellipse contours. The circle contours in Fig 3(a) plots a L-2 penalty with $\lambda = 5$, whereas the square contours in Fig 3(b) plots a L-1 penalty with $\lambda = 5$.

To further explain how it works, the solution to

$$\min_{\theta_0,\theta_1} \left\{ \sum_{i=1}^{n} (y_i - \theta_1 x_i - \theta_0)^2 \mid R(\theta_0, \theta_1) \leq r \right\}$$

is the height of the smallest ellipse contour that is tangent with (or contained in) the contour that depict $R(\theta_0, \theta_1) = r$. The desired (θ_0, θ_1) are the coordinates of the tangent point.

Question:

1. Please assign each plot in Figure 2 to one (and only one) of the following regularization methods. You can get some help from Figure 3. Please answer A, B, C or D.

   (a) (2 pts) No regularization (or regularization parameter equals to 0).

   $$\sum_{i=1}^{3} (y_i - \theta_1 x_i - \theta_0)^2$$

Solution: C

(b) (3 pts) L-2 regularization with λ being 5.

$$\sum_{i=1}^{3}(y_i - \theta_1 x_i - \theta_0)^2 + \lambda(\theta_1^2 + \theta_0^2) \text{ where } \lambda = 5$$

Solution: D

(c) (3 pts) L-1 regularization with λ being 5.

$$\sum_{i=1}^{3}(y_i - \theta_1 x_i - \theta_0)^2 + \lambda(|\theta_1| + |\theta_0|) \text{ where } \lambda = 5$$

Solution: B

(d) (2 pts) L-2 regularization with λ being 1.

$$\sum_{i=1}^{3}(y_i - \theta_1 x_i - \theta_0)^2 + \lambda(\theta_1^2 + \theta_0^2) \text{ where } \lambda = 1$$

Solution: A

2. (2 pts) If we have much more features (that is more x_i's) and we want to perform feature selection while solving the LR problem, which kind of regularization method do we want to use? (Hint: L-1 or L-2? What about λ?)

Solution: We will choose L-1, and we will use bigger λ when we want fewer effective features.

5 Sampling [8 Points]

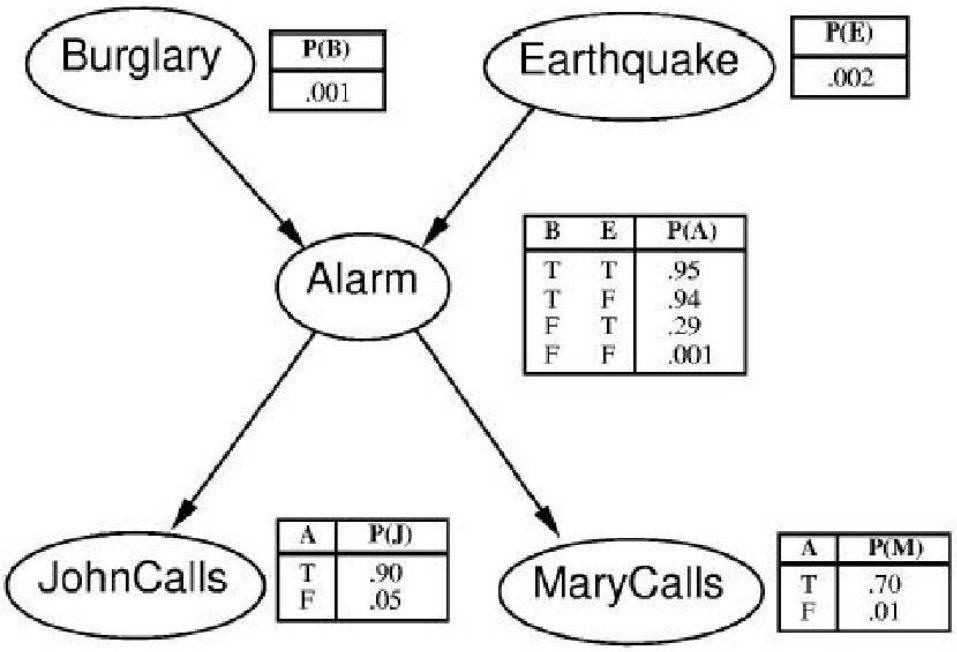


Figure 4: A Bayesian Network for studying sampling

1. (2 pts) Suppose we want to compute $P(B1|E1)$ using the naive sampling method. We draw 1,000,000 sample records in total. How many useful samples do we expect to see? (Hint: $B1$ means Burglary is true.)

   Solution: Records with $E = 1$ are useful samples. So, $1000000 * 0.002 = 2000$.

2. (1 pts) Suppose we want to compute $P(B1|J1)$ using the Gibbs sampling algorithm. How many different states of (B,E,A,J,M) will we observe during the process?

   Solution: There are four variables (B,E,A,M), each of which has two states, so we can observe $2^4 = 16$ different states.

3. (3 pts) Suppose we want to compute $P(B1|J1)$ using the Gibbs sampling algorithm, and we start with state (B1,E0,A0,J1,M0). We choose variable E in the first step. What are the possible states after the first step, and what are their probability of occurrence respectively?

   Solution: The next possible states are (B1,E0,A0,J1,M0) and (B1,E1,A0,J1,M0), because only E may change.

$$P(E1|B1, A0) = \frac{P(E1, B1, A0)}{P(B1, A0)} = \frac{P(E1, B1, A0)}{P(E1, B1, A0) + P(E0, B1, A0)}$$
$$P(E1, B1, A0) = P(E1) * P(B1) * P(A0|E1, B1) = 10^{-7}$$
$$P(E0, B1, A0) = P(E0) * P(B1) * P(A0|E0, B1) = 5.988 * 10^{-5}$$
$$\text{So, } P(E1|B1, A0) = 0.0017$$
$$P(E0|B1, A0) = 1 - P(E1|B1, A0) = 0.9983$$

With probability 0.9983 it will become (B1,E0,A0,J1,M0), and with probability 0.0017 it will become (B1,E1,A0,J1,M0).

4. (2 pts) In Markov Chain Monte Carlo (MCMC), is choosing the transition probabilities to satisfy the property of detailed balance a necessary condition for ensuring that a stationary distribution exists? Please answer Yes or No.

   Solution: No. It is a sufficient condition.

6 Expectation Maximization [10 Points]

Imagine a machine learning class where the probability that a student gets an "A" grade is $\mathbb{P}(A) = 1/2$, a "B" grade $\mathbb{P}(B) = \mu$, a "C" grade $\mathbb{P}(C) = 2\mu$, and a "D" grade $\mathbb{P}(D) = 1/2 - 3\mu$. We are told that c students get a "C" and d students get a "D". We don't know how many students got exactly an "A" or exactly a "B". But we do know that h students got either an a or b. Therefore, a and b are unknown values where $a + b = h$. Our goal is to use expectation maximization to obtain a maximum likelihood estimate of μ.

1. (4 pts) Expectation step: Which formulas compute the expected values of a and b given μ? Circle your answers.

$$\widehat{a} = \frac{1/2}{1/2+h}\mu \qquad \widehat{b} = \frac{\mu}{1/2+h}\mu$$
$$**** \widehat{a} = \frac{1/2}{1/2+\mu}h \qquad \widehat{b} = **** \frac{\mu}{1/2+\mu}h$$
$$\widehat{a} = \frac{\mu}{1/2+\mu}h \qquad \widehat{b} = \frac{1/2}{1/2+\mu}h$$
$$\widehat{a} = \frac{1/2}{1+\mu^2}h \qquad \widehat{b} = \frac{\mu}{1+\mu^2}h$$

   Solution: Marked with ****

2. (4 pts) Maximization step: Given the expected values of a and b which formula computes the maximum likelihood estimate of μ? Circle your answer. *Hint:* Compute the MLE of μ assuming unobserved variables are replaced by their expectation.

$$**** \widehat{\mu} = \frac{h-a+c}{6(h-a+c+d)}$$
$$\widehat{\mu} = \frac{h-a+d}{6(h-2a-d)}$$
$$\widehat{\mu} = \frac{h-a}{6(h-2a+c)}$$
$$\widehat{\mu} = \frac{2(h-a)}{3(h-a+c+d)}$$

   Solution: Marked with ****

3. (True/False, 2 pts) Iterating between the E-step and M-step will *always* converge to a local optimum of μ (which may or may not also be a global optimum)? Explain in 1-2 sentences. **Solution:** True, the lower bound increases on each iteration.

7 VC-Dimension and Learning Theory [10 Points]

1. (True/False, 2 pts) Can the set of all rectangles in the 2D plane (which includes non axis-aligned rectangles) shatter a set of 5 points? Explain in 1-2 sentences.

   Solution: True, can shatter 5 points along a circle.

2. (2 pts) What is the VC-dimension of k-Nearest Neighbour classifier when $k = 1$? Explain in 1-2 sentences.

   Solution: Infinity since it can shatter an arbitrary training set.

3. (2 pts) Consider the classifier $f(a) = 1$ if $a > 0$ and $f(a) = 0$ otherwise. What is the VC-dimension of $f(sin(\alpha x))$ when α is an adjustable parameter? Explain in 1-2 sentences.

   Solution: Infinity since it can shatter an arbitrary training set.

Consider the following formulas that bound the number of training examples necessary for successful learning:

$$
\begin{aligned}
m &\geq \frac{1}{\epsilon}(\ln(1/\delta) + \ln|H|) \\
m &\geq \frac{1}{2\epsilon^2}(\ln(1/\delta) + \ln|H|) \\
m &\geq \frac{1}{\epsilon}(4\log_2(2/\delta) + 8VC(H)\log_2(13/\epsilon))
\end{aligned}
$$

For each of the below questions, **pick the formula** you would use to estimate the number of examples you would need to learn the concept. You do not need to do any computation or plug in any numbers. Explain your answer.

1. (2 pts) Consider instances with two Boolean variables $\{X_1, X_2\}$, and responses Y are given by the XOR function. We try to learn the function $f : X \rightarrow Y$ using a *2-layer neural network*.

   Solution: Eq (3) since the hypothesis space is infinite.

2. (2 pts) Consider instances with two Boolean variables $\{X_1, X_2\}$, and responses Y are given by the XOR function. We try to learn the function $f : X \rightarrow Y$ using a *depth-two decision tree*. This tree has four leaves, all distance two from the top.

   Solution: Eq(1) because the hyptothesis space is finite and $Y \in H$

8 Hidden Markov Models with continuous emissions (10 points)

In this question, we will study hidden markov models with continuous emissions. We will use the notation used in class, with x^i denoting the output at time i, and y_i denoting the corresponding hidden state. The HMM has K states $\{1 \dots K\}$. The output for state k is obtained by sampling a Gaussian distribution parameterized by mean μ_k and standard deviation σ_k. Thus, we can write the emission probabilitye as $p(x_i|y_i = k, \theta) = \mathcal{N}(x_i|\mu_k, \sigma_k)$. θ is the set of parameters of the HMM, which includes the initial probabilities π, transition probability matrix A and the means and standard deviations $\{\mu_1, \dots, \mu_K, \sigma_1, \dots, \sigma_k\}$.

8.1 Log-likelihood (1 point)

Write down the log-likelihood for a sequence of observations of the emissions $\{x_1, \dots, x_n\}$ when the states (also observed) are $\{y_1, \dots, y_n\}$.

Solution:

$$\begin{aligned} \log p(x_1, \dots, x_n|y_1, \dots, y_n) &= \log \prod_i p(x_i|y_i) && (1) \\ &= \sum_i \log(\mathcal{N}(x_i|\mu_{y_i}, \sigma_{y_i})) && (2) \end{aligned}$$

8.2 Forward and backward updates (2 points)

Write the forward and backward update equations for this HMM. Explain in a single line how they are different from the updates we studied in class.

Solution:

$$\begin{aligned} \alpha_t^k &= \mathcal{N}(x_t|\mu_k, \sigma_k) \sum_i \alpha_{t-1}^i a_{i,k} && (3) \\ \beta_t^k &= \sum_i a_{k,i} \beta_{t+1}^i \mathcal{N}(x_t|\mu_i, \sigma_i) && (4) \end{aligned}$$

The equations are similar in form. But in this case, the output probabilities are gaussian rather than multinomial. The outputs are also continuous rather than discrete.

8.3 Supervised parameter learning

We are given a sequence of observations $X = \{x_1, \dots, x_n\}$ and the corresponding hidden states $Y = \{y_1, \dots, y_n\}$. We want to find the parameters θ for the HMM.

1. Are the update equations for A_{ij} and π_i different from the ones obtained for the HMM we studied in class? Explain why or why not (2 points).

   Solution: The update equations for A_{ij} and π_i are the same. They involve only the state transition counts and so are independent of the form chosen for emission probabilities.

2. What are the update equations for the Gaussian parameters μ_k and σ_k ? (Hint: You do not need to derive them. Given the hidden states, the outputs are all independent of each other, and each is sampled from one out of K gaussians.) (2 points)

   Solution:

$$\mu_k = \frac{\sum_i \mathcal{I}[y_i = k] x_i}{\sum_i \mathcal{I}[y_i = k]} \tag{5}$$

$$\sigma_k^2 = \frac{\sum_i \mathcal{I}[y_i = k](x_i - \mu_k)^2}{\sum_i \mathcal{I}[y_i = k]} \tag{6}$$

8.4 Unsupervised parameter learning

Now, we are only given a sequence of observations $X = \{x_1, \ldots, x_n\}$. We want to find the parameters θ for the HMM. (Slide 47 and 48 for the HMM lecture describe the unsupervised learning algorithm for the HMM discussed in class)

8.5 Objective function

The unsupervised learning algorithm optimizes the expected complete log-likelihood. Why is that a reasonable choice for the objective function? (1 point)

Solution: The expected complete log-likelihood is a lower bound to the complete log-likelihood. It is guaranteed to converge to a local optimum of the complete likelihood. Hence it is a reasonable choice for the objective function.

8.5.1 Expected complete LL

What is the expected complete log-likelihood ($\langle l_c(\theta; x, y) \rangle$) for the HMM with continuous gaussian emissions? Just write the expression, a derivation is not necessary.(1 point)

Solution:

$$\langle l_c(\theta; x, y) \rangle = \sum_n \left(\left\langle y_{n,1}^i \right\rangle \log \pi_i \right) + \sum_n \sum_{t=2}^{T} \left(\left\langle y_{n,t-1}^i y_{n,t}^j \right\rangle \log a_{i,j} \right) + \sum_n \sum_{t=1}^{T} \left(\left\langle y_{n,t}^i \right\rangle \log \mathcal{N}(x_n, t | \mu_i, \sigma_i) \right) \tag{7}$$

8.5.2 Gaussian Parameter estimation

Suppose you want to find ML estimates $\hat{\mu}_k$ and $\hat{\sigma}_k$ for parameters μ_k and σ_k . Will the ML expressions have the same form as those obtained for the means and variances in a mixture of gaussians? Explain in one line. (Hint: Write down the terms in $\langle l_c(\theta; x, y) \rangle$ that are relevant to the optimization (i.e, contain μ_k and σ_k))(1 point)

Solution: Yes, the ML expressions will have same form. The relevant term in $\langle l_c(\theta; x, y) \rangle$ is only the last term, which closely resembles the expected complete log-likelihood for a gaussian mixture

model. In this case the $p(y = 1|x)$ term is computed using the forward backward algorithm rather than by simply using Bayes rule (as is done for a mixture of gaussians).

9 Bayesian Networks (10 points)

Consider the Bayesian network shown in Figure 5. All the variables are boolean.

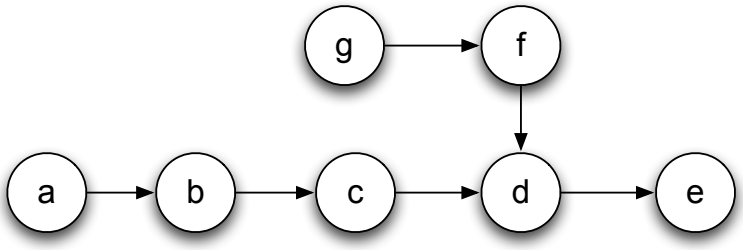


Figure 5: Bayesian network for Question 9.2 and 9.3

9.1 Likelihood

Write the expression for the joint likelihood of the network in its factored form. (2 points)

Solution: $p(a, b, c, d, e, f, g) = p(a)p(b|a)p(c|b)p(d|c, f)p(e|d)p(f|g)p(g)$

9.2 D-separation

1. Let $X = \{c\}, Y = \{b, d\}, Z = \{a, e, f, g\}$. Is $X \perp Z|Y$? If yes, explain why. If no, show a path from X to Z that is not blocked. (2 points)

   Solution: No, $X \not\perp Z|Y$. The path $c \rightarrow d \rightarrow f$ is not blocked since the v-structure at d is observed.

2. Suppose you are allowed to choose a set W such that $W \subset Z$. Then define $Z^* = Z/W$ and $Y^* = Y \cup W$. What is the smallest set W such that $X \perp Z^*|Y^*$ is true? (2 points)

   Solution: $W = \{f\}$ is the smallest subset that satisfies the requirement. Y^* then is the Markov Blanket of node c.

9.3 Conditional Independence

From the graph, we can see that $a \perp c, d|b$. Prove using the axioms of probability that this implies $a \perp c|b$. (2 points)

Solution: $a \perp c, d|b$ means $P(a, c, d|b) = P(a|b)P(c, d|b)$. We want to prove $a \perp c|b$ using the

axioms of probability.

$$
\begin{aligned}
P(a, c|b) &= \sum_d P(a, c, d|b) \text{ by the axiom of additivity for disjoint events} && (8)\\
&= \sum_d P(a|b)P(c, d|b) && (9)\\
&= P(a|b) \sum_d P(c, d|b) && (10)\\
&= P(a|b)P(c|b) && (11)
\end{aligned}
$$

Hence $a \perp c|b$. Other proofs are also accepted.

9.4 Structure learning

Suppose that we do not know the directionality of the edges $a - b$ and $b - c$, and we are trying to learn that by observing the conditional probability $p(a|b, c)$. Some of the entries in the table are observed and noted. Fill in the rest of the conditional probability table so that we obtain the directionality that we see in the graph, i.e, $a \rightarrow b$ and $b \rightarrow c$. (2 points)

$\mathrm{P}(a = 1\|b = 0, c = 0)$	0.8
$\mathrm{P}(a = 1\|b = 0, c = 1)$	**0.8**
$\mathrm{P}(a = 1\|b = 1, c = 0)$	0.4
$\mathrm{P}(a = 1\|b = 1, c = 1)$	**0.4**

Solution: We want $a \perp c|b$, i.e $P(a|b, c) = P(a|b)$. So we want that $P(a|b, c)$ should be the same for all values of c.