Problem Set 4

10-601 Fall 2012

Due: Friday Nov. 9, at 4 pm

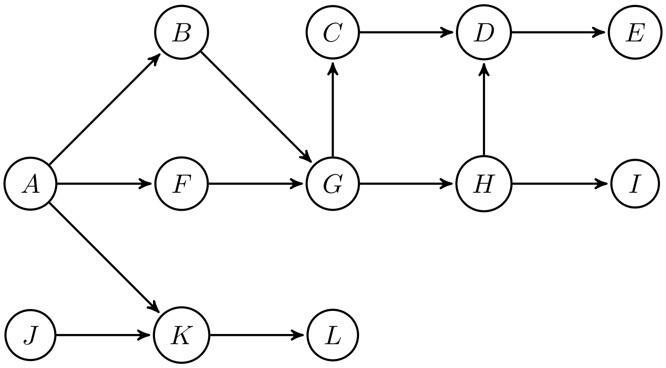
TA: Daegun Won (daegunw@cs.cmu.edu)

Due Date

This is due at **Friday Nov. 9, at 4 pm**. Hand in a hard copy to Sharon Cavlovich, GHC 8215.

1 Bayesian Network

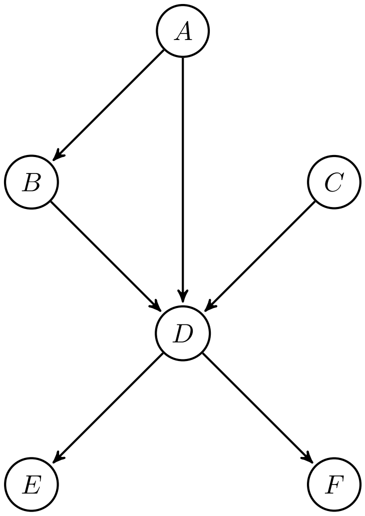
1.a d-separation



Which of the following statements are true given the Bayesian network below? For false statements, show one active trail.

1. $P(H,J) = P(H)P(J)$ [*Solution:* true]
2. $P(H,J|L) = P(H|L)P(J|L)$ [*Solution:* false. There's an active trail JKAFGH]
3. $P(C,I|F) = P(C|F)P(I|F)$ [*Solution:* false. CGHI is an active trail.]
4. $P(C,I|G,E) = P(C|G,E)P(I|G,E)$ [*Solution:* false. CDHI]
5. $P(A,D|B) = P(A|B)P(D|B)$ [*Solution:* false. AFGHD]
6. $P(B,F) = P(B)P(F)$ [*Solution:* false BAF]
7. $P(C,K|B,F) = P(C|B,F)P(K|B,F)$ [*Solution:* true]
8. $P(E,K|L) = P(E|L)P(K|L)$ [*Solution:* false. KAFGHDE]

1.b Variable Elimination



$$P(A=T)=0.6, P(C=T)=0.8$$

$$P(B=T|A=T)=0.5, P(B=T|A=F)=0.1$$

$$P(D=T|A=T,B=T,C=T)=0.6, P(D=T|A=F,B=T,C=T)=0.3$$
$$P(D=T|A=T,B=T,C=F)=0.9, P(D=T|A=F,B=T,C=F)=0.5$$
$$P(D=T|A=T,B=F,C=T)=0.1, P(D=T|A=F,B=F,C=T)=0.7$$
$$P(D=T|A=T,B=F,C=F)=0.1, P(D=T|A=F,B=F,C=F)=0.6$$

$$P(E=T|D=T)=0.5, P(E=T|D=F)=0.6$$
$$P(F=T|D=T)=0.9, P(F=T|D=F)=0.8$$

1. Using variable elimination, compute $P(A=T,B=T,C=T,E=T,F=T)$. Show your work. *[Solution:* 0.11088*]*

2. From your work above, compute $P(E=T,F=T)$ by removing B, A, and C in order. Show your work. *[Solution:* 0.465408*]*

3. Compute $P(E=T,F=T)$ again but using elimination order of A, B, C and then D. *[Solution:* 0.465408*]*

4. Would you say the order of variables matter in terms of final result? How about in terms of computational efficiency? *[Solution:* It shouldn't affect the final value, but it affects the computational efficiency.*]*

1.c Constructing a Network

Let X, Y, Z be binary variables. After observing many instances of X, Y, Z, you summarized the data with the following joint distribution.

X	Y	Z	$P(X, Y, Z)$
0	0	0	0.042
0	0	1	0.378
0	1	0	0.054
0	1	1	0.126
1	0	0	0.140
1	0	1	0.140
1	1	0	0.096
1	1	1	0.024

Draw a Bayes net that can represent the above distribution with as few edges as possible. How many such networks are there? Show your work.

[*Solution:* If you look for marginal dependencies, only X and Y are marginally dependent. Thus the graph has to be in a form such as X-Z-Y. There are 4 possible BNs with two edges, but only $X \to Z \leftarrow Y$ preserves the marginal dependence between X and Y.]

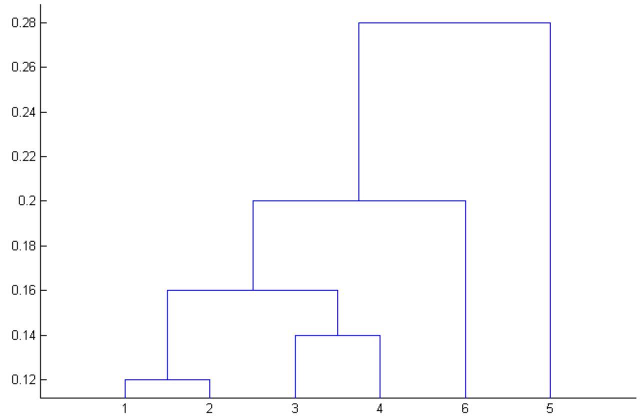
2 Clustering

The table below is a distance matrix for 6 objects.

	A	B	C	D	E	F
A	0					
B	0.12	0				
C	0.51	0.25	0			
D	0.84	0.16	0.14	0		
E	0.28	0.77	0.70	0.45	0	
F	0.34	0.61	0.93	0.20	0.67	0

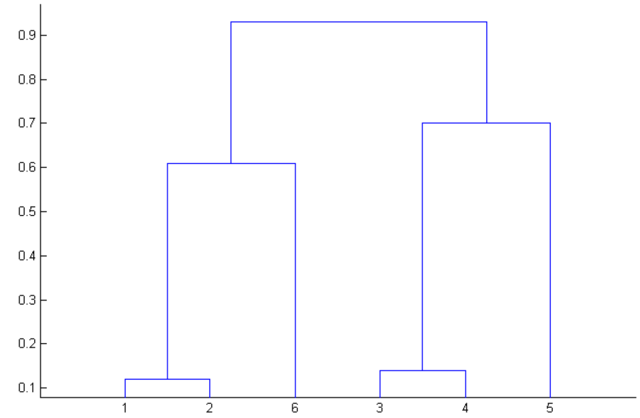
2.a Hierarchical clustering

1. Show the final result of hierarchical clustering with single link by drawing a dendrogram.



[Solution: *]*

2. Show the final result of hierarchical clustering with complete link by drawing a dendrogram.



[Solution: *]*

3. Change **two** values from the matrix so that your answer to the last two question would be same.
[Solution: There is more than one way possible, but one way would be the following:

The first step that the complete link clustering differs from the single link clustering is where AB and F are grouped together by dist(AB,F)=dist(B,F)=0.61. We'd want dist(AB, CD)=dist(A,D) to be smaller than this value, such as 0.53.

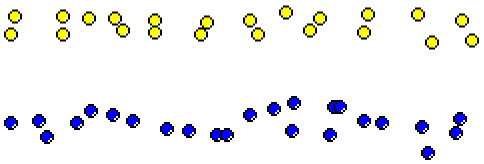
Then we want dist(ABCD,F) = dist(C, F) = 0.93 to be the smallest so that ABCD and F are grouped together. We set this value to 0.63. After these changes both dendrograms become identical.]

2.b Which clustering method should we use?

Which clustering method(s) is most likely to produce the following results at $k = 2$? Choose the most likely method(s) and briefly explain why it/they will work better where others will not in **at most 3 sentences**. Here are the five clustering methods you can choose from:

- Hierarchical clustering with single link
- Hierarchical clustering with complete link
- Hierarchical clustering with average link
- K-means
- GMM (with no assumption on the covariance matrices)

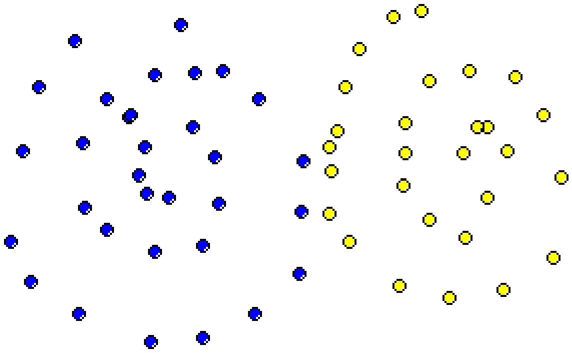
1.



[Solution: Hierarchical clustering with single link is most likely to well. GMM can also produce a decision boundary that can produce such clustering result, but depending on initialization it might converge to a different set of clusters (left half vs. right half).

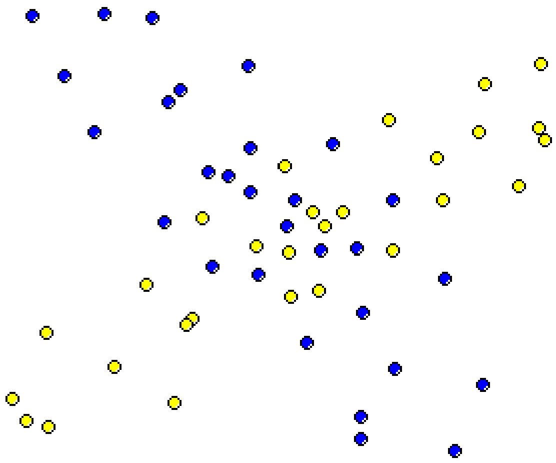
Other hierarchical clusterings won't really work well because at some point, two intermediate clusters from different true cluster will have shorter cluster distance than two from the same true cluster.]

2.



[Solution: K-means or GMM is most likely. Hierarchical clustering wouldn't work since the early few steps will group instances near the decision boundary (note some of them are very close).]

3.



[Solution: Among the five methods, only GMM has the capability of handling overlapping clusters. So GMM is the only method that would result in such clusters. *]*

3 Semi-supervised learning

Let H be the set of all polynomials. Consider the following function $d(h_1, h_2) : H \times H \to \mathbb{R}$:

$$d(h_1, h_2) = \int |h_1(x) - h_2(x)| p(x) dx$$

3.a

1. Show that $d(h_1, h_2)$ is a distance metric.
   [Solution:
   -Non-negativity: Both the absolute value and the pdf $p(x)$ is nonnegative, thus the integrated value has to be nonnegative
   -Symmetry: $|h_1(x) - h_2(x)| = |h_2(x) - h_1(x)|$, thus the value being integrated will be the same for $d(h_1, h_2)$ and $d(h_2, h_1)$
   -Triangle inequality: We know that $|a + b| \leq |a| + |b|$, so

$$\begin{aligned} d(h_1, h_2) &= \int |h_1(x) - h_2(x)| p(x) dx \\ &= \int |h_1(x) - h_3(x) + h_3(x) - h_2(x)| p(x) dx \\ &\leq \int (|h_1(x) - h_3(x)| + |h_3(x) - h_2(x)|) p(x) dx \\ &= \int |h_1(x) - h_3(x)| p(x) dx + \int |h_3(x) - h_2(x)| p(x) dx \\ &= d(h_1, h_3) + d(h_3, h_2) \end{aligned}$$

   ]

2. Let L be a set of labeled instances, U be a set of unlabeled instances, and f be the true classifier. How would you estimate $d(h_1, f)$ and $d(h_1, h_2)$?

[Solution:
$d(h_1, h_2) = \frac{1}{|U|} \sum_{x \in U} |h_1(x) - h_2(x)|$
$d(h_1, f) = \frac{1}{|L|} \sum_{x \in L} |h_1(x) - f(x)|$
]

3.b

Suppose you made the following observations from $[0, 1] \times \mathbb{R}$:

x	0.1	0.2	0.4	0.5	0.6	0.8	0.9	1.0
y	7.72	8.13	6.39	3.35	3.09	12.26	17.73	0.80

n	h_n	$\hat{d}(h_n, h_{n-1})$
1	$2.158x + 6.220$	
2	$6.498x^2 - 5.013x + 7.598$	0.451
3	$-175.6x^3 + 293.7x^2 - 133.9x + 20.98$	3.244
4	$-864.1x^4 + 1769x^3 - 1170x^2 + 278.5x - 11.14$	4.553
5	$-2297x^5 + 5417x^4 - 4477x^3 + 1570x^2 - 230.5x + 19.1$	3.315
6	$-2812x^6 + 6920x^5 - 6289x^4 + 2763x^3 - 671.8x^2 + 87.49x + 3.477$	1.171

1. Let H be the set of all polynomials and h_n be your hypothesis of degree n minimizing the squared error (i.e. $\sum_{(x,y)}(h_n(x) - y)^2$). Which n would you choose? Show your work. You may want to write a short Matlab program to do this part (you do not need to submit the code)

[Solution:
Using the answer to 3.a.2,

n	$\hat{d}(h_n, h_{n-1})$	$\hat{d}(h_n, f) + \hat{d}(h_{n-1}, f)$
t2	0.451	8.0934
3	3.244	7.5618
4	4.553	5.5267
5	3.315	2.3702
6	1.171	

At $n = 5$, the triangle inequality does not hold, which indicates overfitting. So $n = 4$. *]*

4 Programming (K-means)

In this problem we will implement K-means clustering. The data provided is a Matlab file of image data of 5000 handwritten digits. Each digit is a greyscale image of 10 x 10 pixels and is represented as a row vector of length 100. The variable X contains all the images in a 5000 x 100 matrix, and the vector Y contains the true label of each image.

1. Implement K-means algorithm. For initial cluster centers, use random points. Repeat the random start 10 times for each clustering run. After getting the K-means result with 10 different initializations, how can you determine the best starting point? For the following questions, use the best initialization for your final result.

   [Solution: The best starting point is one whose final clustering result has the smallest objective value.*]*

2. We define the objective function of K-means as the sum of the squared distances of each point to its cluster centers, $\sum_{k=1}^{K} \sum_{i=1}^{n_k} (x_{ki} - \mu_k)^2$. Run your program with $K = 10$ and plot the values of objective function against iterations. Is it monotonically decreasing?

   [*Solution:* It should be monotonically decreasing]

3. Try running it with $K = 16$ and plot the objective function again. How is the behavior of the objective function different from when $K = 10$?

   [*Solution:* The objective function converges at a lower value. Also takes more iterations to converge]

4. Clustering performance is hard to evaluate. However, since we have the true labels, we can use the following heursitics. For each cluster C, we find the most frequent (true) label Y_C and label the instances in that cluster with the majority label Y_C. Report your precision (number of correctly labeled instances / number of all instances) and final value of the objective function for $K = 1, 5, 10, 16, 20$.

   [*Solution:*

K	Precision	Obj. Func
1	0.1140	1.2762e+09
5	0.4482	9.5501e+08
10	0.5912	8.1849e+08
16	0.6842	7.3375e+08
20	0.7170	6.9961e+08

   ]

5. Among the five values you tried above, what would you choose to be the optimal number of clusters and why?

   [*Solution:* Using knee/elbow finding, it looks like 10 (or 5) should be the optimal number. Some said that there is no clear knee/elbow, that's also acceptable given the graph.]