

10-601 Assignment 4

Tanay Gavankar

tgavanka@andrew.cmu.edu

Due: 11/9/12

1.a: Bayesian Network: d-separation

- (1) True, they are d-separated because Z is empty and K is a collider node (rule 1), so they are conditionally independent.
- (2) True, they are d-separated because the only path between them has a collider (rule 2), so they are conditionally independent.
- (3) False, they are d-connected because the path CGHI does not have a collider nor a member of Z (rule 2), so they are not conditionally independent.
- (4) True, they are d-separated because all paths between C and I contain either a collider (D) or a member of Z (G) (rule 2), so they are conditionally independent.
- (5) False, they are d-connected because the path AFGHD does not contain a member of Z or a collider (rule 2), so they are not conditionally independent.
- (6) False, they are d-connected because the path BAF does not contain a collider and Z is empty (rule 1), so they are not conditionally independent.
- (7) True, they are d-separated because all paths between C and K must pass through either B or F , both of which are in Z (rule 2), so they are conditionally independent).
- (8) False, they are d-separated because the path EDHGFAK does not have any colliders or members of Z (rule 2), so they are not conditionally independent.

1.b: Bayesian Network: Variable Elimination

(1)

$$\begin{aligned}
P(A, B, C, D, E, F) &= P(A)P(B|A)P(C)P(D|A, B, C)P(E|D)P(F|D) \\
f_D(A, B, C, E, F) &= \sum_D P(D|A, B, C)P(E|D)P(F|D) \\
P(A, B, C, E, F) &= P(A)P(B|A)P(C)f_D(A, B, C, E, F) \\
&= P(A)P(B|A)P(C)P(D|A, B, C)P(E|D)P(F|D) \\
&\quad + P(A)P(B|A)P(C)P(\neg D|A, B, C)P(E|\neg D)P(F|\neg D) \\
&= P(A)P(B|A)P(C)(P(D|A, B, C)P(E|D)P(F|D) \\
&\quad + P(\neg D|A, B, C)P(E|\neg D)P(F|\neg D)) \\
&= P(A)P(B|A)P(C)(P(D|A, B, C)P(E|D)P(F|D) \\
&\quad + (1 - P(D|A, B, C))P(E|\neg D)P(F|\neg D)) \\
&= (0.6)(0.5)(0.8)((0.6)(0.5)(0.9) + (1 - (0.6))(0.6)(0.8)) \\
&= 0.11088
\end{aligned}$$

(2)

$$\begin{aligned}
f_D(A, B, C, E, F) &= \sum_D P(D|A, B, C)P(E|D)P(F|D) \\
f_D(A, B, C, E, F) &= 0.6 * 0.5 * 0.9 + 0.4 * 0.6 * 0.8 = 0.462 \\
f_D(A, B, \neg C, E, F) &= 0.9 * 0.5 * 0.9 + 0.1 * 0.6 * 0.8 = 0.453 \\
f_D(A, \neg B, C, E, F) &= 0.1 * 0.5 * 0.9 + 0.9 * 0.6 * 0.8 = 0.477 \\
f_D(A, \neg B, \neg C, E, F) &= 0.1 * 0.5 * 0.9 + 0.9 * 0.6 * 0.8 = 0.477 \\
f_D(\neg A, B, C, E, F) &= 0.3 * 0.5 * 0.9 + 0.7 * 0.6 * 0.8 = 0.471 \\
f_D(\neg A, B, \neg C, E, F) &= 0.5 * 0.5 * 0.9 + 0.5 * 0.6 * 0.8 = 0.465 \\
f_D(\neg A, \neg B, C, E, F) &= 0.7 * 0.5 * 0.9 + 0.3 * 0.6 * 0.8 = 0.459 \\
f_D(\neg A, \neg B, \neg C, E, F) &= 0.6 * 0.5 * 0.9 + 0.4 * 0.6 * 0.8 = 0.462
\end{aligned}$$

$$P(A, B, C, E, F) = P(A)P(B|A)P(C)f_D(A, B, C, E, F)$$

$$f_B(A, C, E, F) = \sum_B P(B|A)f_D(A, B, C, E, F)$$

$$f_B(A, C, E, F) = 0.5 * 0.462 + 0.5 * 0.477 = 0.4695$$

$$f_B(\neg A, C, E, F) = 0.1 * 0.471 + 0.9 * 0.459 = 0.4602$$

$$f_B(A, \neg C, E, F) = 0.5 * 0.453 + 0.5 * 0.477 = 0.465$$

$$f_B(\neg A, \neg C, E, F) = 0.1 * 0.465 + 0.9 * 0.462 = 0.4623$$

$$P(A, C, E, F) = P(A)P(C)f_B(A, C, E, F)$$

$$f_A(C, E, F) = \sum_A P(A)f_B(A, C, E, F)$$

$$f_A(C, E, F) = 0.6 * 0.4695 + 0.4 * 0.4602 = 0.46578$$

$$f_A(\neg C, E, F) = 0.6 * 0.465 + 0.4 * 0.4623 = 0.46392$$

$$P(C, E, F) = P(C)f_A(C, E, F)$$

$$\begin{aligned} f_C(E, F) &= \sum_C P(C)f_A(C, E, F) \\ &= 0.8 * 0.46578 + 0.2 * 0.46392 = 0.465408 \end{aligned}$$

$$\begin{aligned} P(E, F) &= f_C(E, F) \\ &= 0.465408 \end{aligned}$$

So, we have performed elimination in the given order and solved for $P(E, F)$.

(3)

$$\begin{aligned}
P(A, B, C, D, E, F) &= P(A)P(B|A)P(C)P(D|A, B, C)P(E|D)P(F|D) \\
f_A(B, C, D) &= \sum_A P(A)P(B|A)P(D|A, B, C) \\
f_A(B, C, D) &= 0.6 * 0.5 * 0.6 + 0.4 * 0.1 * 0.3 = 0.192 \\
f_A(B, C, \neg D) &= 0.6 * 0.5 * 0.4 + 0.4 * 0.1 * 0.7 = 0.148 \\
f_A(B, \neg C, D) &= 0.6 * 0.5 * 0.9 + 0.4 * 0.1 * 0.5 = 0.29 \\
f_A(B, \neg C, \neg D) &= 0.6 * 0.5 * 0.1 + 0.4 * 0.1 * 0.5 = 0.05 \\
f_A(\neg B, C, D) &= 0.6 * 0.5 * 0.1 + 0.4 * 0.9 * 0.7 = 0.282 \\
f_A(\neg B, C, \neg D) &= 0.6 * 0.5 * 0.9 + 0.4 * 0.9 * 0.3 = 0.378 \\
f_A(\neg B, \neg C, D) &= 0.6 * 0.5 * 0.1 + 0.4 * 0.9 * 0.6 = 0.246 \\
f_A(\neg B, \neg C, \neg D) &= 0.6 * 0.5 * 0.9 + 0.4 * 0.9 * 0.4 = 0.414
\end{aligned}$$

$$P(B, C, D, E, F) = P(C)P(E|D)P(F|D)f_A(B, C, D)$$

$$\begin{aligned}
f_B(C, D) &= \sum_B f_A(B, C, D) \\
f_B(C, D) &= 0.192 + 0.282 = 0.474 \\
f_B(C, \neg D) &= 0.148 + 0.378 = 0.526 \\
f_B(\neg C, D) &= 0.29 + 0.246 = 0.536 \\
f_B(\neg C, \neg D) &= 0.05 + 0.414 = 0.464
\end{aligned}$$

$$P(C, D, E, F) = P(C)P(E|D)P(F|D)f_B(C, D)$$

$$\begin{aligned}
f_C(D) &= \sum_C P(C)f_B(C, D) \\
f_C(D) &= 0.8 * 0.474 + 0.2 * 0.536 = 0.4864 \\
f_C(\neg D) &= 0.8 * 0.526 + 0.2 * 0.464 = 0.5136
\end{aligned}$$

$$P(D, E, F) = P(E|D)P(F|D)f_C(D)$$

$$\begin{aligned}
f_D(E, F) &= \sum_D P(E|D)P(F|D)f_C(D) \\
&= 0.5 * 0.9 * 0.4864 + 0.6 * 0.8 * 0.5136 = 0.465408
\end{aligned}$$

$$\begin{aligned}
P(E, F) &= f_D(E, F) \\
&= 0.465408
\end{aligned}$$

(4) The order of elimination does not affect the final result. However, the order of the variables does affect computational efficiency. For example, in the second ordering, there were no multiplications (only additions) in the B-elimination step. Note that in the first version, every elimination had

both multiplication and addition.

1.c: Bayesian Network: Constructing a Network



There are 3 different nets that satisfy this joint probability table. To figure this out, marginal probabilities were calculated to see if there were any conditional independences found in the data. It was determined that X is conditionally independent from Y . So, the net above was generated and checked to have the following values.

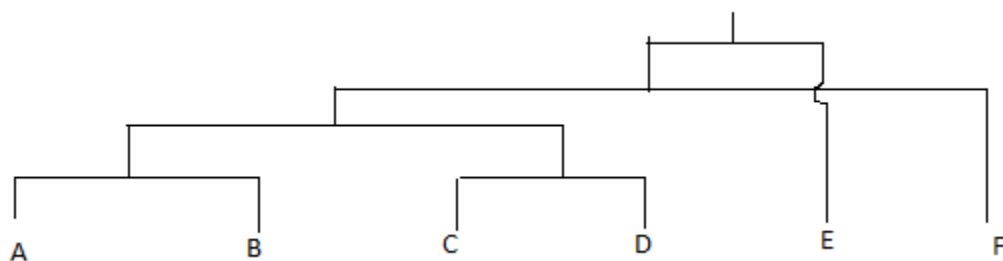
Y	P(Y)
0	0.7
1	0.3

Y	Z	P(Z Y)
0	0	0.26
1	0	0.5
0	1	0.74
1	1	0.5

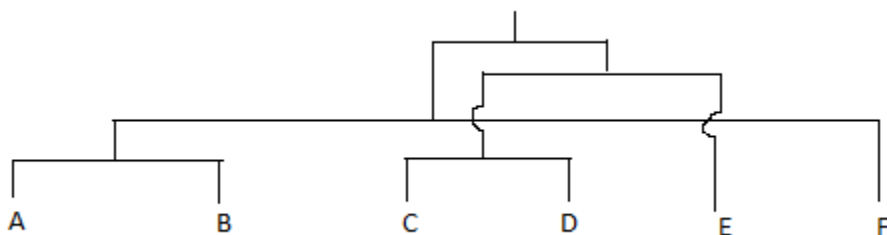
Y	Z	X	P(X Y,Z)
0	0	0	0.230769
0	1	0	0.72973
1	0	0	0.36
1	1	0	0.84
0	0	1	0.769231
0	1	1	0.27027
1	0	1	0.64
1	1	1	0.16

2.a: Clustering: Heirarchical Clustering

(1)



(2)



(3) Swap the distances for AD and BF.

2.b: Clustering: Which clustering method should we use?

(1) This looks like hierarchical clustering with single link. Points close to each other are clustered together, and since the clusters are long and skinny, they are more likely to be single link.

(2) This looks like K-means clustering. Points are clustered in a circular pattern, and k-means attempts to minimize the distance between the points and the cluster centroids.

(3) This looks like GMM. Because the two clusters seem to cross each other but still have a distinct shape, it seems like they follow a two-dimensional probability distribution, as is the case in GMM.

3.a: Semi-Supervised Learning

(1) To show that $d(h_1, h_2)$ is a distance metric, we must show that it is symmetric, non-negative and satisfies the triangle inequality.

Symmetry:

$$\begin{aligned}
d(h_1, h_2) &= \int |h_1(x) - h_2(x)|p(x)dx \\
d(h_2, h_1) &= \int |h_2(x) - h_1(x)|p(x)dx \\
&= \int |-(h_1(x) - h_2(x))|p(x)dx \text{ by } a - b = -(b - a) \text{ factoring} \\
&= \int |h_1(x) - h_2(x)|p(x)dx \text{ by def of absolute value} \\
\Rightarrow d(h_2, h_1) &= d(h_1, h_2)
\end{aligned}$$

and

Assume $h_1 = h_2$.

$$\begin{aligned}
d(h_1, h_2) &= d(h_1, h_1) \\
&= \int |h_1(x) - h_1(x)|p(x)dx \\
&= \int 0 * p(x)dx \\
&= 0 \\
\Rightarrow h_1 = h_2 &\Rightarrow d(h_1, h_2) = 0
\end{aligned}$$

Assume $d(h_1, h_2) = 0$.

$$\begin{aligned}
&\Rightarrow p(x) = 0 \quad \text{or} \quad |h_1(x) - h_2(x)| = 0 \\
&\quad \text{Case } p(x) = 0, \text{ trivial} \\
&\text{Case } |h_1(x) - h_2(x)| = 0 \\
&\Rightarrow h_1(x) - h_2(x) = 0 \\
&\Rightarrow h_1(x) = h_2(x) \\
&\Rightarrow d(h_1, h_2) = 0 \Rightarrow h_1 = h_2
\end{aligned}$$

Together, these two proofs show prove that $h_1 = h_2 \iff d(h_1, h_2) = 0$.

Non-negative:

We know that an integral is essentially a sum of, and so we know that if all the terms of the sum are non-negative, the total sum will also be non-negative. So, we can show that $d(h_1, h_2)$ is non-negative by showing each term is non-negative. In particular, we want to show that $|h_1(x) - h_2(x)|p(x) \geq 0$. By looking at the terms, we know that $p(x)$ is a probability distribution function, and it is impossible for a probability to be negative, and so that term is non-negative. Furthermore, by definition of absolute value, we know that the term $|h_1(x) - h_2(x)|$ must be non-negative. Therefore, we have a product of two non-negative terms, which we know must also be non-negative. So, the entire integral is non-negative. QED.

Triangle Inequality:

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

So, $d(h_1, h_2)$ is a distance metric because it satisfies all the criteria.

(2) You can estimate $d(h_1, f)$ by using L and U with h_1 and seeing how different it is from the true labels (as they would be predicted by f). Then, you can estimate $d(h_1, h_2)$ by using the triangle inequality and splitting it into $d(h_1, f) + d(f, h_2)$ and using the first strategy.

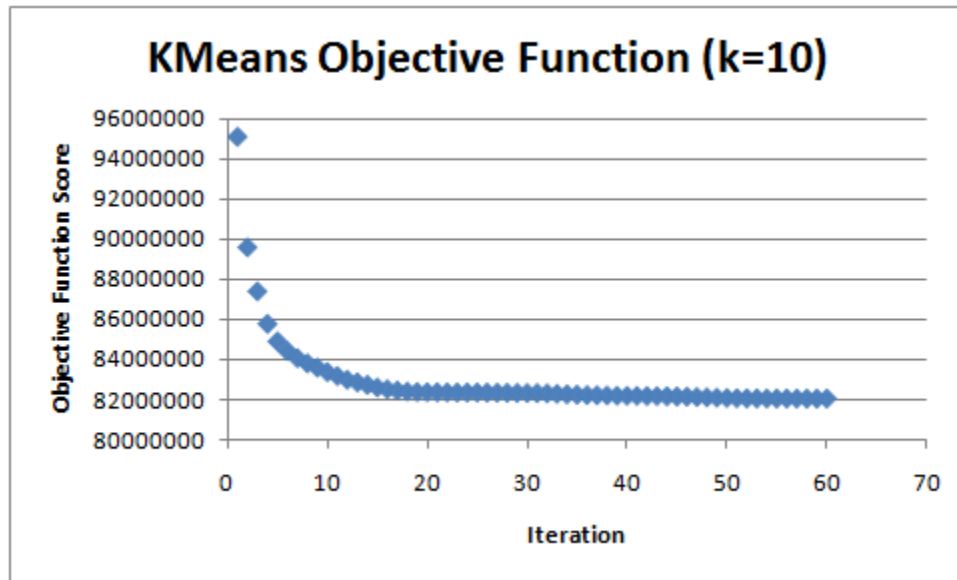
3.b: Semi-Supervised Learning

(1) I would pick $n = 2$ because all the hypotheses after that are overfitting. I wrote a Python script that outputted the squared error for each hypothesis, and got the following values for hypotheses h_1 through h_6 (respectively): 207, 205, 234661, 821984, 3, 1. Furthermore, I also used this script to calculate additional distance metrics (in particular, $\hat{d}(h_i, f)$), which in this case turned out to be 1 for $i \in \{1, 2, 3, 4, 5, 6\}$. This means that since we have the heuristic of continue training until $\hat{d}(h_i, h_{i-1})$ fails to satisfy the triangle inequality, we should train only until $\hat{d}(h_i, h_{i-1}) \leq \hat{d}(h_i, f) + \hat{d}(h_{i-1}, f)$ no longer holds. This occurs when $i = 3$, because $\hat{d}(h_3, h_2) = 3.244 \not\leq 2.0$, and so the triangle inequality no longer holds. Thus, I pick $n = 2$.

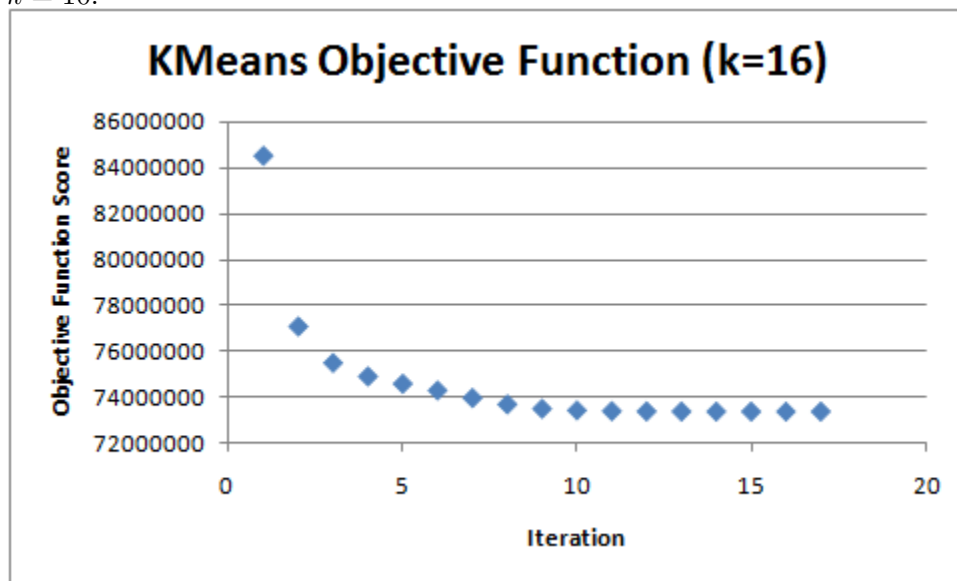
4: Programming: K-Means

(1) Since we have the true labels, we can evaluate the clustering performance by seeing the precision of the run, and so when starting with randomly selected centroids, the "best" results would be the ones with the highest precision. (In the case that we did not have the true labels, the lowest value of the objective function of K-means could have been used. Note that it is possible to have the lowest objective function value but still not the best precision.)

(2) The objective function is monotonically decreasing as the number of iterations increases.



(3) The algorithm converges much, much earlier (17 iterations instead of 60) than when $k = 10$, and also the overall objective function score asymptotes to a lower value when $k = 16$ than when $k = 10$.



(4)

K	Accuracy	Objective Function Score
1	0.114	1276236239
5	0.449	955013392
10	0.6346	828455714
16	0.6882	732872757
20	0.733	695375664

(5) By looking at the true labels, it seems like there were only 10 clusters (as there were only 10

labels, 0-9). The labels of the clusters when $k = 10$ were relatively evenly distributed (0-9, with the exception of label 5 being replaced with 7). At higher k values, there were many more duplicates that were not evenly distributed, meaning it seemed that the data was overfit and it was splitting an existing large cluster into smaller clusters to account for the larger value of k . So, it seems like 10 clusters would be the optimal number, providing that good starting points were chosen. However, because of the way we actually label the clusters after clustering (using the highest count true label to label the entire cluster), we get better accuracy as the number of clusters increase (most likely due to overfitting on the training data - which we want to avoid). Also, it is logical to assume that the final objective function score will decrease as the number of clusters increase because, with more centroids, in general, the distance between a point and its corresponding cluster's centroid will decrease. This means that the overall objective function will also decrease.