Homework Assignment 3

Tuesday, October 13, 2015 5:06 PM

James Lowrey

Data Mining Analysis and Concepts, M. Zaki and W. Meira (the authors have kindly made an online version available):
http://www.dataminingbook.info/uploads/book.pdf

13.1) Given the following points: 2,4,10,12,3,20,30,11,25. Assume k = 3, and that we randomly pick the initial means μ1 = 2, μ2 = 4 and μ3 = 6. Show the clusters obtained using K-means algorithm after one iteration, and show the new means for the next iteration.

After one iteration of K-Means algorithm:
Original mean (2):2,3
Original mean (4):4
Original mean (6):10,12,20,30,11,25

Original mean (2). New mean = 2.5
Original mean (4). New mean = 4
Original mean (6). New mean = 18

13.3)
(a)

Table 13.2. Dataset for Q3

	X_1	X_2
$\mathbf{x}_1$	0	2
$\mathbf{x}_2$	0	0
$\mathbf{x}_3$	1.5	0
$\mathbf{x}_4$	5	0
$\mathbf{x}_5$	5	2

Given the two-dimensional points in Table 13.2, assume that $k = 2$, and that initially the points are assigned to clusters as follows: $C_1 = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_4\}$ and $C_2 = \{\mathbf{x}_3, \mathbf{x}_5\}$. Answer the following questions:

(a) Apply the K-means algorithm until convergence, that is, the clusters do not change, assuming (1) the usual Euclidean distance or the L_2-*norm* as the distance between points, defined as $\|\mathbf{x}_i - \mathbf{x}_j\|_2 = \left(\sum_{a=1}^{d}(x_{ia} - x_{ja})^2\right)^{1/2}$, and (2) the Manhattan distance or the L_1-*norm* defined as $\|\mathbf{x}_i - \mathbf{x}_j\|_1 = \sum_{a=1}^{d}|x_{ia} - x_{ja}|$.

Euclidean Distance
Original clusters { [0,2] , [0,0] , [5,0] } { [1.5,0] , [5,2] } have means {1.6666666666666667,0.6666666666666666} {3.25,1}
Iteration 1 has clusters { [0,2] , [0,0] , [1.5,0] } { [5,0] , [5,2] } with means {0.5,0.6666666666666666} {5,1}
Iteration 2 has clusters { [0,2] , [0,0] , [1.5,0] } { [5,0] , [5,2] } with means {0.5,0.6666666666666666} {5,1}

Manhattan Distance
Original clusters { [0,2] , [0,0] , [5,0] } { [1.5,0] , [5,2] } have means {1.6666666666666667,0.6666666666666666} {3.25,1}
Iteration 1 has clusters { [0,2] , [0,0] , [1.5,0] } { [5,0] , [5,2] } with means {0.5,0.6666666666666666} {5,1}
Iteration 2 has clusters { [0,2] , [0,0] , [1.5,0] } { [5,0] , [5,2] } with means {0.5,0.6666666666666666} {5,1}

14.1)
a)

Table 14.3. Data for Q1

Point	X_1	X_2	X_3	X_4	X_5
$\mathbf{x}_1$	1	0	1	1	0
$\mathbf{x}_2$	1	1	0	1	0
$\mathbf{x}_3$	0	0	1	1	0
$\mathbf{x}_4$	0	1	0	1	0
$\mathbf{x}_5$	1	0	1	0	1
$\mathbf{x}_6$	0	1	1	0	0

The similarity between categorical data points can be computed in terms of the number of matches and mismatches for the different attributes. Let n_{11} be the number of attributes on which two points $\mathbf{x}_i$ and $\mathbf{x}_j$ assume the value 1, and let n_{10} denote the number of attributes where $\mathbf{x}_i$ takes value 1, but $\mathbf{x}_j$ takes on the value of 0. Define n_{01} and n_{00} in a similar manner. The contingency table for measuring the similarity is then given as

		$\mathbf{x}_j$	
		1	0
$\mathbf{x}_i$	1	n_{11}	n_{10}
	0	n_{01}	n_{00}

Define the following similarity measures:

- Simple matching coefficient: $SMC(X_i, X_j) = \frac{n_{11}+n_{00}}{n_{11}+n_{10}+n_{01}+n_{00}}$
- Jaccard coefficient: $JC(X_i, X_j) = \frac{n_{11}}{n_{11}+n_{10}+n_{01}}$
- Rao's coefficient: $RC(X_i, X_j) = \frac{n_{11}}{n_{11}+n_{10}+n_{01}+n_{00}}$

Find the cluster dendrograms produced by the hierarchical clustering algorithm under the following scenarios:

ALGORITHM 14.1. Agglomerative Hierarchical Clustering Algorithm

```
AGGLOMERATIVECLUSTERING(D, k):
C ← {C_i = {x_i} | x_i ∈ D} // Each point in separate cluster
Δ ← {δ(x_i, x_j): x_i, x_j ∈ D} // Compute distance matrix
repeat
    Find the closest pair of clusters C_i, C_j ∈ C
    C_ij ← C_i ∪ C_j // Merge the clusters
    C ← (C \ {C_i, C_j}) ∪ {C_ij} // Update the clustering
    Update distance matrix Δ to reflect new clustering
until |C| = k
```

(a) We use single link with RC.

{ [1,0,1,1,0] } { [1,1,0,1,0] } { [0,0,1,1,0] } { [0,1,0,1,0] } { [1,0,1,0,1] } { [0,1,1,0,0] } --- Original, 6 clusters
{ [1,0,1,1,0] } { [1,1,0,1,0] } { [0,0,1,1,0] } { [0,1,1,0,0] } { [0,1,0,1,0] , [1,0,1,0,1] } --- Next, the clusters that are closest are merged
{ [1,1,0,1,0] } { [0,0,1,1,0] } { [0,1,1,0,0] } { [1,0,1,1,0] , [{ [0,1,0,1,0] } { [1,0,1,0,1] }] } --- Repeat of above.
{ [0,0,1,1,0] } { [0,1,1,0,0] } { [1,1,0,1,0] , [{ [1,0,1,1,0] } { [0,1,0,1,0] , [1,0,1,0,1] }] }
{ [0,1,1,0,0] } { [0,0,1,1,0] , [{ [1,1,0,1,0] } { [1,0,1,1,0] , [{ [0,1,0,1,0] } { [1,0,1,0,1] }] }] }
{ [{ [0,1,1,0,0] } { [0,0,1,1,0] , [{ [1,1,0,1,0] } { [1,0,1,1,0] , [{ [0,1,0,1,0] } { [1,0,1,0,1] }] }] }] } ---final, 1 cluster

(b) We use complete link with SMC.

{ [1,0,1,1,0] } { [1,1,0,1,0] } { [0,0,1,1,0] } { [0,1,0,1,0] } { [1,0,1,0,1] } { [0,1,1,0,0] }
{ [0,0,1,1,0] } { [0,1,0,1,0] } { [1,0,1,0,1] } { [0,1,1,0,0] } { [1,0,1,1,0] , [1,1,0,1,0] }
{ [1,0,1,0,1] } { [0,1,1,0,0] } { [1,0,1,1,0] , [1,1,0,1,0] } { [0,0,1,1,0] , [0,1,0,1,0] }
{ [1,0,1,1,0] , [1,1,0,1,0] } { [0,0,1,1,0] , [0,1,0,1,0] } { [1,0,1,0,1] , [0,1,1,0,0] }
{ [1,0,1,0,1] , [0,1,1,0,0] } { [{ [1,0,1,1,0] } { [1,1,0,1,0] }] , [{ [0,0,1,1,0] } { [0,1,0,1,0] }] }
{ [{ [1,0,1,0,1] , [0,1,1,0,0] } { [{ [1,0,1,1,0] } { [1,1,0,1,0] }] , [{ [0,0,1,1,0] } { [0,1,0,1,0] }] }] }

(c) We use group average with JC.

{ [1,0,1,1,0] } { [1,1,0,1,0] } { [0,0,1,1,0] } { [0,1,0,1,0] } { [1,0,1,0,1] } { [0,1,1,0,0] }
{ [1,0,1,1,0] } { [1,1,0,1,0] } { [0,0,1,1,0] } { [0,1,1,0,0] } { [0,1,0,1,0] , [1,0,1,0,1] }
{ [1,1,0,1,0] } { [0,0,1,1,0] } { [0,1,1,0,0] } { [1,0,1,1,0] , [{ [0,1,0,1,0] } { [1,0,1,0,1] }] }
{ [0,0,1,1,0] } { [0,1,1,0,0] } { [1,1,0,1,0] , [{ [1,0,1,1,0] } { [0,1,0,1,0] , [1,0,1,0,1] }] }
{ [0,1,1,0,0] } { [0,0,1,1,0] , [{ [1,1,0,1,0] } { [1,0,1,1,0] , [{ [0,1,0,1,0] } { [1,0,1,0,1] }] }] }
{ [{ [0,1,1,0,0] } { [0,0,1,1,0] , [{ [1,1,0,1,0] } { [1,0,1,1,0] , [{ [0,1,0,1,0] } { [1,0,1,0,1] }] }] }] }

14.3)

Table 14.4. Dataset for Q3

	A	*B*	*C*	*D*	*E*
A	0	1	3	2	4
B		0	3	2	3
C			0	1	3
D				0	5
E					0

Using the distance matrix from Table 14.4, use the average link method to generate hierarchical clusters. Show the merging distance thresholds.

newDist = Ai*dist(Ci,Cr) + Aj*dist(Cj,Cr) where Ci and Cj are clusters that are being merged, and Cr is some cluster that is not being merged (Lance-Williams formula for group average clusters)

where Ai = |Ci| /(|Ci| + |Cj|) and Aj = |Cj| /(|Ci| + |Cj|)

The group average method defines the values in the distance matrix as the average distance between all the points of the clus ter and another cluster. Thus this distance should be minimized when merging clusters

BEGIN ALGORITHM

A,B are the closest ->merge!

	AB	C	D	E
AB	0	.5*1+.5*3 = 2	.5*2+.5*2=2	.5*4+.5*3=3.5
C		0	1	3
D			0	5
E				0

	AB	CD	E
AB	0	.5*2+.5*2=2	3.5
CD		0	.5*3+.5*5=4
E			0

	ABCD	E
ABCD	0	.5*3.5+.5*4=3.75
E		0

ABCDE

15.1)

Consider Figure 15.12 and answer the following questions, assuming that we use the Euclidean distance between points, and that $\epsilon = 2$ and $minpts = 3$

(a) List all the core points.

(b) Is a directly density reachable from d?

(c) Is o density reachable from i? Show the intermediate points on the chain or the point where the chain breaks.

(d) Is density reachable a symmetric relationship, that is, if x is density reachable from y, does it imply that y is density reachable from x? Why or why not?

(e) Is l density connected to x? Show the intermediate points that make them density connected or violate the property, respectively.

(f) Is density connected a symmetric relationship?

(g) Show the density-based clusters and the noise points.

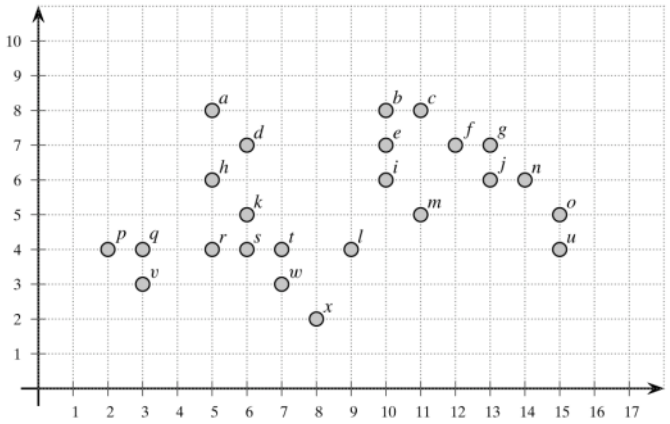

Figure 15.12. Dataset for Q1.

a) b, c, d, e, f, g, h, I, j, k, n, q, r, s, t, w
b) no
c) i->e->f->j->n->o
d) No. I is not density reachable from O as O is not a core point. Thus N is not directly density reachable from O and the relationship is not symmetric
e) x<-w<- t ->L
f) In one sense yes, as either endpoint can be reached by the central point. However, the direction of the paths cannot be reversed as density reachable is not symmetric. Also, the paths can be different lengths and thus the overall path lengths are not necessarily symmetric.
g) A density-based cluster is defined as a maximal set of density connected points. Noise points are non-core, non-border points.
   a. clusters: [p, q, v, r, s, t, w, x, l, k, h, d, a], [m, I, e, b, c, f, g, j, n, o]
   b. noise points: u

18.1)

Consider the dataset in Table 18.3. Classify the new point: (Age=23, Car=truck) via the full and naive Bayes approach. You may assume that the domain of Car is given as {sports, vintage, suv, truck}.

Table 18.3. Data for Q1

$\mathbf{x}_i$	Age	Car	Class
$\mathbf{x}_1$	25	sports	L
$\mathbf{x}_2$	20	vintage	H
$\mathbf{x}_3$	25	sports	L
$\mathbf{x}_4$	45	suv	H
$\mathbf{x}_5$	20	sports	H
$\mathbf{x}_6$	25	suv	H

Naïve Bayes Theorem:

P(truck) = (#trucks + 1)/(sizeCluster + #DiffCategoricalPossibilities)

P(age) = 1/(sqrt(2*pi) *stdDev) * e^([datapoint - mean]^2/[2*stdDev^2]) = normal probability density function

L: mean_age = 25, stdDev age = 0

P(car=truck) = (0+1)/(2+4) = 0.1667

P(age=23) = 0

P(age=23 & car=truck) = P(age=23) * P(car=truck) = 0.1667 *0 = 0

H: mean_age = 27.5, stdDev age = 10.3078

P(age=23) = 0.03519

P(car=truck) = (0+1)/(4+4) = 0.125

P(age=23 & car=truck) = P(age=23) * P(car=truck) = 0.03519 * 0.125 = 0.0044

Thus the class is ---- since it has a larger probability

18.2

use the KNN classifier (k=3) to classify the new point (T,F,1.0). Define a distance metric that equally weights all 3 attributes. You will need to use a form of min - max normalization for the third attribute to ensure equal weights.

Table 18.4. Data for Q2

$\mathbf{x}_i$	a_1	a_2	a_3	Class
$\mathbf{x}_1$	T	T	5.0	Y
$\mathbf{x}_2$	T	T	7.0	Y
$\mathbf{x}_3$	T	F	8.0	N
$\mathbf{x}_4$	F	F	3.0	Y
$\mathbf{x}_5$	F	T	7.0	N
$\mathbf{x}_6$	F	T	4.0	N
$\mathbf{x}_7$	F	F	5.0	N
$\mathbf{x}_8$	T	F	6.0	Y
$\mathbf{x}_9$	F	T	1.0	N

Let T=1 and F=0. Thus after min-max normalization the third data values become

x1	x2	x3	x4	x5	x6	x7	x8	x9
0.5714	0.8571	1	0.2857	0.8571	0.4286	0.5714	0.7143	0

The distance metric will be Euclidean distance where the T/F values have been substituted for 1,0 and the a3 value has been m in-max normalized with the existing dataset. The distance from the new point to all other points is below.

	x1	x2	x3	x4	x5	x6	x7	x8	x9
distance from (1,0,0)	1.1518	1.3171	1	1.0400	1.6537	1.4777	1.1518	0.7143	1.4142

Thus the 3 nearest neighbors are x8 (Y), x3 (N), and x4(Y). (Y) is the majority class amongst the new point's neighbors and t hus the new point will be labeled as (Y).

19.4

Table 19.4. Data for Q4

Instance	a_1	a_2	a_3	Class
1	T	T	5.0	Y
2	T	T	7.0	Y
3	T	F	8.0	N
4	F	F	3.0	Y
5	F	T	7.0	N
6	F	T	4.0	N
7	F	F	5.0	N
8	T	F	6.0	Y
9	F	T	1.0	N

(a) Show which decision will be chosen at the root of the decision tree using information gain [Eq. (19.5)], Gini index [Eq. (19.6)], and CART [Eq. (19.7)] measures. Show all split points for all attributes.

(b) What happens to the purity if we use Instance as another attribute? Do you think this attribute should be used for a decision in the tree?

(a). Attribute a1 will be chosen as root as the decision tree. As you can see from the entropy calculations, splitting on a1 results in the lowest entropy, thus the maximum information gain. The lowest GINI index is also achieved via splitting on a1, which means it is the best. The max CART value is also a1, further supporting it as the split point.

In the splits below for a1 and a2: True = 1, False =0. Splitting on 0 or 1 yields the same value as they result in the same s plit in reverse order (if there were more categories this wouldn't be the case).

Splitting using Entropy:
Splitting on attribute a1
Split values are 1,0

Split Val of 1 yields 0.7616392191414825
Split Val of 0 yields 0.7616392191414825
Splitting on attribute a2
Split values are 1,0
Split Val of 1 yields 0.9838614413637048
Split Val of 0 yields 0.9838614413637048
Splitting on attribute a3
Split values are 2,3.5,4.5,5,5.5,6.5,7,7.5
Split Val of 2 yields 0.8888888888888888
Split Val of 3.5 yields 0.9885107724710844
Split Val of 4.5 yields 0.9727652780181631
Split Val of 5 yields 0.9727652780181631
Split Val of 5.5 yields 0.9838614413637048
Split Val of 6.5 yields 0.9727652780181631
Split Val of 7 yields 0.9727652780181631
Split Val of 7.5 yields 0.8888888888888888

Splitting using GINI:
Splitting on attribute a1
Split values are 1,0
Split Val of 1 yields 0.34444444444444433
Split Val of 0 yields 0.34444444444444433
Splitting on attribute a2
Split values are 1,0
Split Val of 1 yields 0.4888888888888889
Split Val of 0 yields 0.4888888888888889
Splitting on attribute a3
Split values are 2,3.5,4.5,5,5.5,6.5,7,7.5
Split Val of 2 yields 0.4444444444444444
Split Val of 3.5 yields 0.49206349206349215
Split Val of 4.5 yields 0.48148148148148145
Split Val of 5 yields 0.48148148148148145
Split Val of 5.5 yields 0.4888888888888889
Split Val of 6.5 yields 0.48148148148148145
Split Val of 7 yields 0.48148148148148145
Split Val of 7.5 yields 0.4444444444444444

Splitting using CART:
Splitting on attribute a1
Split values are 1,0
Split Val of 1 yields 0.5432098765432098
Split Val of 0 yields 0.5432098765432098
Splitting on attribute a2
Split values are 1,0
Split Val of 1 yields 0.0987654320987654
Split Val of 0 yields 0.0987654320987654
Splitting on attribute a3
Split values are 2,3.5,4.5,5,5.5,6.5,7,7.5
Split Val of 2 yields 0.19753086419753085
Split Val of 3.5 yields 0.04938271604938271
Split Val of 4.5 yields 0.14814814814814814
Split Val of 5 yields 0.14814814814814814
Split Val of 5.5 yields 0.0987654320987654
Split Val of 6.5 yields 0.14814814814814814
Split Val of 7 yields 0.14814814814814814
Split Val of 7.5 yields 0.19753086419753085

(b) As the number of attributes increases there will be more splitting in the decision tree. If the clusters in the original (non-instance using) decision tree are not pure, and a new attribute is added further splitting can be completed resulting in clusters with potent ially increased homogeneity. Since clusters become more homogenous, they would be becoming more pure.
This attribute should not be used in a decision tree, as it increases model complexity and build time without indicating any real information about

the underlying object features. In the worst case the instances are organized such that classes are alternating, and thus the split would only separate 1 value which is not very helpful.

19.5

Consider Table 19.5. Let us make a nonlinear split instead of an axis parallel split, given as follows: $AB - B^2 \leq 0$. Compute the information gain of this split based on entropy (use $\log_2$, i.e., log to the base 2).

Table 19.5. Data for Q5

	A	B	Class
$\boldsymbol{x}_1$	3.5	4	H
$\boldsymbol{x}_2$	2	4	H
$\boldsymbol{x}_3$	9.1	4.5	L
$\boldsymbol{x}_4$	2	6	H
$\boldsymbol{x}_5$	1.5	7	H
$\boldsymbol{x}_6$	7	6.5	H
$\boldsymbol{x}_7$	2.1	2.5	L
$\boldsymbol{x}_8$	8	4	L

split classes
AB-B^2 <= 0 :: 1,2,4,5,7
AB-B^2 > 0 :: 3,6,8
Information Gain: 0.9544340029249649 - 0.7955659970750351 = 0.15886800584992988