Datamining Assignment 4: Clustering

Arve Nygård

27.03.2014

1 Repetition: Apriori Algorithm

ID	Transaction
1	A, B, C
2	A, C
4	A, D
5	B, E, F

Table 1: Shopping basket

Set	Support Count	Support	Frequent
A	3	0.75	√
B	2	0.5	√
C	2	0.5	√
D	1	0.25	x
E	1	0.25	x
F	1	0.25	x
—	—	—	—
AB	1	0.25	x
AC	2	0.5	√
BC	1	0.25	x

Table 2: Frequent item sets. We only generate 2-tuples from frequent singlets. We stop when we observe that AC is the only frequent 2-tuple.

Candidate Rule	Confidence	Valid rule?
A->C	$2/3 = 0.3$	No
C->A	$2/2 = 1$	Yes

Table 3: Rule generation

2 Clustering

2.1 *k*-Means Clustering

ID	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
X	3	1	2	4	7	9	6	9	6	8

Distance calculated using the formula $Distance = |X_{centroid} - X_{point}|$

2.1.1 - Two Centroids:

ID	Position	Distance to A(2)	Distance to B(5)	Class
P1	3	1	2	A
P2	1	1	4	A
P3	2	0	3	A
P4	4	2	1	B
P5	7	5	2	B
P6	9	7	4	B
P7	6	4	1	B
P8	9	7	4	B
P9	6	4	1	B
P10	8	6	3	B

Table 5: Two centroids: First iteration

Update Centroids:
$A = (3 + 1 + 2)/3 = 2$
$B = (4 + 7 + 9 + 6 + 9 + 6 + 8)/7 = 7$

ID	Position	Distance to A(2)	Distance to B(7)	Class
P1	3	1	4	A
P2	1	1	6	A
P3	2	0	5	A
P4	4	2	3	A
P5	7	5	0	B
P6	9	7	2	B
P7	6	4	1	B
P8	9	7	2	B
P9	6	4	1	B
P10	8	6	1	B

Table 6: Two centroids: Second Iteration

Update Centroids:
$A = (3 + 1 + 2 + 4)/4 = 2.5$
$B = (7 + 9 + 6 + 9 + 6 + 8)/7.5$

ID	Position	Distance to A(2.5)	Distance to B(7.5)	Class
P1	3	.5	4.5	A
P2	1	1.5	6.5	A
P3	2	.5	5.5	A
P4	4	1.5	3.5	A
P5	7	4.5	.5	B
P6	9	6.5	1.5	B
P7	6	3.5	1.5	B
P8	9	6.5	1.5	B
P9	6	3.5	1.5	B
P10	8	5.5	.5	B

Table 7: Two centroids: Third Iteration.

No points changed class between the $2^n d$ and $3^r d$ iteration. -> No change in centroids. Stop.

2.1.2 - Three Centroids:

ID	Position	Distance to A(2)	Distance to B(6)	Distance to C(8)	Class
P1	3	1	3	5	A
P2	1	1	5	7	A
P3	2	0	4	6	A
P4	4	2	2	4	A
P5	7	5	1	1	B
P6	9	7	3	1	C
P7	6	4	0	2	B
P8	9	7	3	1	C
P9	6	4	0	2	B
P10	8	6	2	0	C

Table 8: Three centroids: First iteration

Update Centroids:
$ A = (3+1+2+4) / 4 = 2.5$
$ B = (7+6+6) / 3 = 6.33$
$ C = (9+9+8) / 3 = 8.67$

ID	X	Distance to A(2.5)	Distance to B(6.33)	Distance to C(8.67)	Class
P1	3	0.5	3.33	5.67	A
P2	1	1.5	5.33	7.67	A
P3	2	0.5	4.33	6.67	A
P4	4	1.5	2.33	4.67	A
P5	7	4.5	0.67	1.67	B
P6	9	6.5	2.67	0.33	C
P7	6	3.5	0.33	2.67	B
P8	9	6.5	2.67	0.33	C
P9	6	3.5	0.33	2.67	B
P10	8	5.5	1.67	0.67	C

Table 9: Three centroids: Second iteration

No change in classifications -> Stop.

2.2 Hierarchical Agglomerative Clustering (HAC)

2.2.1 Explain the hierarchical clustering and the difference between MIN-link and MAX-link

- Hierarchical clustering: Sett of nested clusters organized in a tree.
- MIN-link / MAX-link: Two ways of deciding closeness of clusters. In MIN-link the distance between two clusters is tefined as the distance between the closest pair of points, having 1 point from each cluster. For MAX-link, the distance between two clusters is defined as the distance between the farthest pair of points where a pair consists of a point from each cluster.

2.2.2 Perform hierarchical agglomerative clustering on the dataset of Table 3 and show the resulting dendrogram. Perform both Min-link and MAX-link.

Point	X	Y
P1	1	11
P2	1	9
P3	1	5
P4	1	2
P5	6	7
P6	11	7

Table 10: Data for hierarchical clustering

We start by computing the distance between each point in the data set. We use the following distance formula: $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$.

Cluster	P1	P2	P3	P4	P5	P6
P1		**2**	6	9	6.4	10.8
P2			4	7	5.4	10.2
P3				3	5.4	10.2
P4					7.1	11.2
P5						5
P6						

Table 11: Distance matrix

We then proceed to iteratively pick the closest pair of points/clusters, using MIN-linking. This consists of merging two rows and doing table lookups to decide the new distance value in our matrix.

Cluster	Members
C1	{p1, p2}

Cluster	C1	P3	P4	P5	P6
C1		4	7	5.4	10.2
P3			**3**	5.4	10.2
P4				7.1	11.2
P5					5
P6					

`merge P3, P4`

Cluster	Members
C1	{p1, p2}
C2	{p3, p4}

Cluster	C1	C2	P5	P6
C1		**4**	5.4	10.2
C2			5.4	10.2
P5				5
P6				

`merge C1, C2`

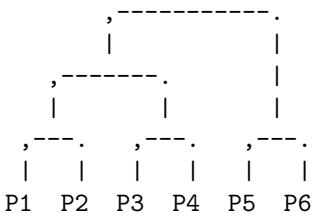
Cluster	Members
C1	{{p1, p2}, {p3, p4}}

Cluster	C1	P5	P6
C1		5.4	10.2
P5			**5**
P6			

```
merge P5,P6
```

Cluster	Members
C1	{{p1, p2}, {p3, p4}}
C2	{p5, p6}

Finally, no more calculations are necessary - we can directly merge the last two sets:
C_{Final} = {{{p1, p2}, {p3, p4}}, {p5, p6}}




```
MIN-Link = {{{p1, p2}, {p3, p4}}, {p5, p6}}
```

For MAX-link, we get the following:

`First iteration the same as before`

Cluster	Members
C1	{p1, p2}

Cluster	C1	P3	P4	P5	P6
C1		6	9	6.4	10.8
P3			**3**	5.4	10.2
P4				7.1	11.2
P5					5
P6					

`merge P3, P4`

Cluster	Members
C1	{p1, p2}
C2	{p3, p4}

Cluster	C1	C2	P5	P6
C1		9	6.4	10.8
C2			7.1	11.2
P5				**5**
P6				

`merge P5, P6`

Cluster	Members
C1	{p1, p2}
C2	{p3, p4}
C3	{p5, p6}

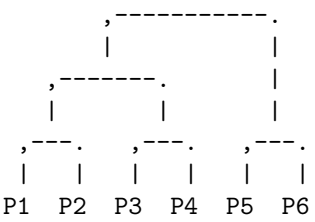
Cluster	C1	C2	C3
C1		**9**	10.8
C2			11.2
C3			

`merge C1, C2`

Cluster	Members
C1	{{p1, p2}, {p3, p4}}
C3	{p5, p6}

`finally, merge C1, C3`

C_{Final} = {{{p1, p2}, {p3, p4}}, {p5, p6}}.



`MAX-Link = {{{p1, p2}, {p3, p4}}, {p5, p6}}`

Note that a proper drawing of the dendrogram should show that the distances between the merged clusters in MAX-link are greater, by having taller "legs". This is pretty hard to draw in ASCII though, and I think by mentioning it here, I have shown that i understand dendrograms :-)

2.3 Clustering Methods

Given the following three descriptions of datasets, decide what algorightm to use and argue why.

2.3.1 Text collection (100.000 documents, 30.000 dimensions, i.e. 30.000 distinct words)

Large collection, many dimensions. K-means.

2.3.2 Noisy data collection (200 instances, 3 dimensions)

Density based clustering, because of all the noise.

2.3.3 Data collection with only little noise, with taxonomy-like relation in between some of the instances (~400 instances, ~ 20 dimensions)

Because of the existing taxonomy in the collection, go with HAC.

2.3.4 Images.

1. K-means should be able to handle the spherical clusters.
2. DBSCAN - because of the odd shapes
3. DBSCAN - because of the odd shapes
4. HAC or k-Means.