AFFINITY PROPAGATION:

- We use a type of clustering algorithm where the complete data is viewed as a network with each data point being a node in the network.
- The entire algorithm is based on finding iteratively how well one point is suited to be a representative of another point (i.e., how suited a particular point is to be an exemplar to another point by gaining information about other prospective representatives in the data).
- Unlike clustering algorithms such as k-means or k-medoids, affinity propagation does not require the number of clusters to be determined or estimated before running the algorithm
- A dataset is described using a small number of exemplars, ‘exemplars’ are members of the input set that are representative of clusters. The messages sent between pairs represent the suitability for one sample to be the exemplar of the other, which is updated in response to the values from other pairs
- This updating happens iteratively until convergence, at that point the final exemplars are chosen, and hence we obtain the final clustering.

ALGORITHM:

- **The first step is to create the similarity matrix based on the given data .**
- **The similarity matrix is formed using the distance between two points. Similarity is the negative of the sum of squares of distance between the points.**
- **Then calculate the responsibility matrix using the similarity matrix and availability matrix. Availability matrix however is initialised to zero.**
- **Once the responsibility matrix is calculated we use it to calculate the availability matrix that was initialized to zero.**
- **Once we have both availability and responsibility matrix we add them to get the criterion matrix which decides the ultimate clusters.**

$$r(i,k) \leftarrow s(i,k) - \max_{k' \text{ such that } k' \neq k} \{a(i,k') + s(i,k')\}, \qquad (1)$$

$$a(k,k) \leftarrow \sum_{i' \text{ such that } i' \neq k} \max\{0, r(i',k)\}, \qquad (2)$$

$$a(i,k) \leftarrow \min\left\{0, r(k,k) + \sum_{i' \text{ such that } i' \notin \{i,k\}} \max\{0, r(i',k)\}\right\} \qquad (3)$$

,

$$c(i,k) \leftarrow r(i,k) + a(i,k). \qquad (4)$$

When updating the messages, it is important that they be damped to avoid numerical oscillations that arise in some circumstances. Each message is set to l times its value from the previous iteration plus 1 – l times its prescribed updated value, where the damping factor l is between 0 and 1. In all of our experiments (3), we used a default damping factor of l = 0.5

Affinity Propagation :

→ Basically a machine learning algorithm used for clustering.

→ Unique: The algorithm determines the optimal no. of clusters. No need of input (k) beforehands.

→ Theory: uses matrix → Similarity Matrix
Responsibility Matrix
Availability Matrix
Crieterion Matrix

Sets exemplers: objects that end up with same exempler are clustered together.

1. Similarity Matrix:

Contains values that correspond to how similar two objects are.

Diagnol values: fill them with lowest no. among all the cell.

Non-diagnol values:

Negation the sum of the squares of the difference between participants.

$$\boxed{S_{ik} = -\|x_i - x_k\|^2 \text{ for } i \neq k}$$

2. Responsibility Matrix:

Contains values that correspond to how responsible one object is for another.

Calculated by:

$$\boxed{r(i,k) \leftarrow s(i,k) - \max_{\substack{k' \text{ such that} \\ k' \neq k}} \{a(i,k') + s(i,k')\}}$$

3. Availability Matrix:

Contains values that correspond to how available one object is to be an exempler for another object.

Diagnol Values: Sum of all positive responsibities in the coloumn, excluding object's self-responsibility.

$$\boxed{a(k,k) \leftarrow \sum_{\substack{i' \text{ such that} \\ i' \neq k}} \max\{0, r(i',k)\}}$$

Exempler ⟹ Centroid of cluster

Non-diagnol values:

$$\boxed{a(i,k) \leftarrow \min\{0, r(k,k) + \sum_{\substack{i' \text{ such that} \\ i' \neq k}} \max\{0, r(i',k)\}\}}$$

4. Crieterion Matrix:

→ Each cell is simply the sum of the availability matrix and Responsibility matrix at that location.

→ The highest crieterion value of each row is then designated as an exemplar.

→ Rows that share the same exempler end up being in the same cluster.

→ No. of exemplers = No. of clusters.

$$\boxed{C(i,k) \leftarrow r(i,k) + a(i,k)}$$

Example:

Preferences of five participants:

Participant	Tax Rate	Fee	Interest Rate	Quantity limit	Price limit
Alice	3	4	3	2	1
Bob	4	3	5	1	1
Cary	3	5	3	3	3
Doug	2	1	3	3	2
Edna	1	1	3	2	3

In an experimental exersize participants are asked to indicate on a 5-point scale their preferences wrt tax rate, fee, interest rate, quantity limit and price limit.

Calculating off-diagnol elements in Similarity Matrix:

$$\boxed{S_{i,k} = -\|x_i - x_k\|^2 ; i \neq k}$$

Similarity b/w alice & Bob:

$$(3-4)^2 + (4-3)^2 + (3-5)^2 + (2-1)^2 + (1-1)^2 = 7$$

Similarity (Alice, Bob) = (-7)

Similarly calculate Similarity for all other non-diagnal elements.

Similarity b/w Edna and bob comes out to be the least.

$$-S(E,B) = (4-1)^2 + (3-1)^2 + (5-3)^2 + (1-2)^2 + (1-3)^2 = 22$$

$\boxed{S(E,B) = -22}$ → least.

The least non-diagnol element is placed in the diagnol positions-(All).

Date
Page

Similarity Matrix:

Participant	Alice	Bob	Cary	Doug	Edna
Alice	(-22)	-7	-6	-12	-17
Bob	-7	(-22)	-17	-17	-22
Cary	-6	-17	(-22)	-18	-21
Doug	-12	-17	-18	(-22)	-3
Edna	-17	-22	-21	-3	(-22)

Reason for placing (-22) in diagnol values:

The algorithm will converge around a small no. of clusters if a smaller value is chosen.
Here (-22) is the smallest off-diagnol value, so placing this value in every one of the diagnol elements directs the algorithm to converge onto a small no. of clusters

Responsibility Matrix:

Responsibility of Bob (coloumn) to Alice (row) is similarity of Bob to Alice (= -7) minus the maximum of the remaining similarities of Alice's row (6).

$$\boxed{s(i,k) - \max\{a(i,k') + s(i,k')\} = r(i,k)}$$

Initially all the elements of availability matrix are set to zero.

$$a(i,k') = 0 \cdot (All).$$

$R(B,A) = S(B,A) - \max$ (0, row remaining similarities of Alice)

$R(B,A) = (-7) - (-6) = (-1)$

$R(A,A) = (-22) - (-6) = (-16)$

Similarly others.

$R(A, B) = S(A,B) - \max(0,$ remaining similarities in Bob row$)$

$R(A,B) = (-7) - (-17) = 10$ · (Bob row)

Responsibility Matrix:

Participant	Alice	Bob	Cary	Doug	Edna
Alice	-16	-1	1	-6	-11
Bob	10	-15	-10	-10	-15
Cary	11	-11	-16	-12	-15
Doug	-9	-14	-15	-19	9
Edna	-14	-19	-18	14	-19

Availability Matrix

Following Eqⁿ(s) are used to update availability matrix.

$$\boxed{a(k,k) \leftarrow \sum_{i' \text{ such that } i' \neq k} \max\{0, r(i',k)\}}$$ → Diagnol elements → excluding self-responsibility

$$\boxed{a(i,k) \leftarrow \min\{0, r(k,k) + \sum_{i' \text{ such that } i' \neq k} \max\{0, r(i',k)\}\}}$$ → Non-diagnol elements.

Self-availabity of Alice is Sum of positive responsibilities of Alice coloumn excluding Alice's self-responsibility

$10 + 11 = 21$ · (Diagnol Value).

Availability of Bob (coloumn) to Alice (row) is bob's self responsibility plus the sum of the remaining positive responsibilities of bob's coloumn excluding the responsibility of bob to Alice

$(-15 + 0 + 0 + 0) = (-15)$.

Availability Matrix:

Participant	Alice	Bob	Cary	Doug	Edna
Alice	21	-15	-16	-5	-10
Bob	-5	0	-15	-5	-10
Cary	-6	-15	1	-5	-10
Doug	0	-15	-15	14	-19
Edna	0	-15	-15	-19	9

Crieterion Matrix:

Eqn used to calculate crieterion matrix:

$$\boxed{c(i,k) = r(i,k) + a(i,k)}$$

hence, crietorion value of Bob (coloumn) to Alice (row) is the sum of the responsibility and availability of bob to Alice.

$$(-1) + (-15) = (-16).$$

Crietorion matrix:

Participant	Alice	bob	Cary	Doug	Edna
Alice	5	-16	-15	-11	-21
bob	5	-15	-25	-15	-25
cary	5	-26	-15	-17	-25
Doug	-9	-29	-30	-5	-10
Edna	-14	-34	-33	-5	-10

The coloumn with highest crietcrion value for each row identifies the exemplar for that item of that row.

Rows that share the same exemplar are in same cluster.

hence, two clusters appear.

cluster 1: {Alice, bob, cary}

cluster 2: {Doug, Edna}.

⇒ Repeated Application of Equations 1−4 do not change the solution, so here the first soln is the soln.

→ This was one iteration, we went through all 4 eqn. In our first iteration the availability matrix is set 0 initially.

Now in second iteration, we keep the value of $a(i,k)$ obtained from first iteration and calculate criterion matrix.

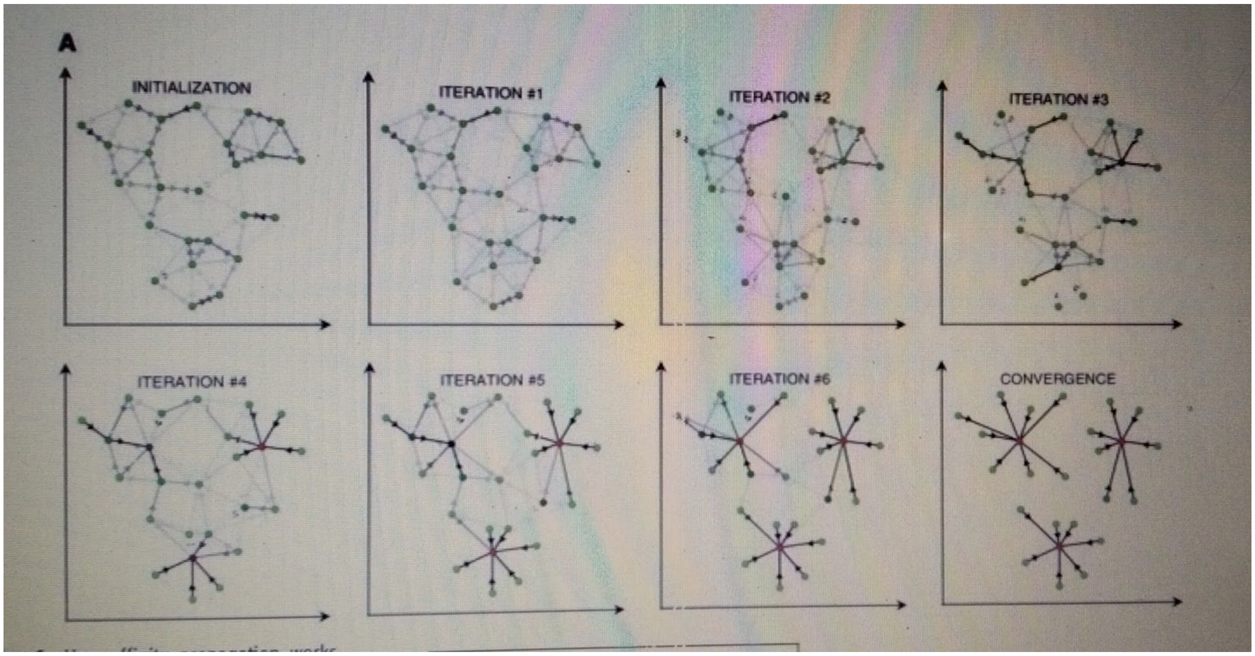
⇒ To assure convergence a tiny bit of random noise is added to the similarity matrix.

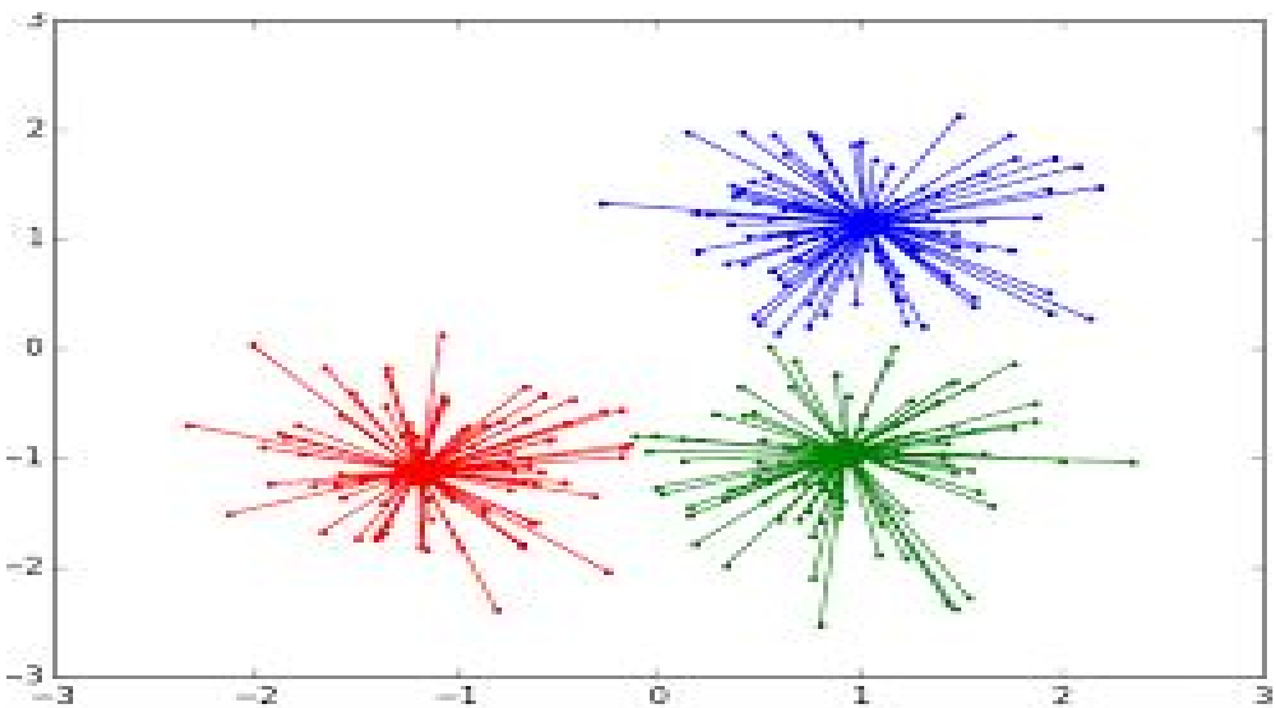
Damping updates of the availability and responsibility matrices by 50%.

$\{r_t', a_t'\}$ → Undamped updates of responsibility and availability matrices at iteration 't'.

Then damped updates are

$$\boxed{\begin{aligned} r_t &= 0.5 r_{t-1} + 0.5 r'_t \\ a_t &= 0.5 a_{t-1} + 0.5 a'_t \end{aligned}}$$


A
INITIALIZATION
ITERATION #1
ITERATION #2
ITERATION #3
ITERATION #4
ITERATION #5
ITERATION #6
CONVERGENCE


3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3

implementation:
https://colab.research.google.com/drive/1CfVKsR1MJkrx85f7SA62X9JryH4e5-Xa?usp=sharing

https://colab.research.google.com/drive/1KUBixlLWRcA6R75gSMd-FXKIaDG4v291?usp=sharing

References:

http://utstat.toronto.edu/reid/sta414/frey-affinity.pdf
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.490.7628&rep=rep1&type=pdf