

Unsupervised Learning in Data Analytics

Introduction

- Describe the nature of the data to be analyzed.
- Unsupervised learning is also known as descriptive analytics
- There is no real target variable (pure exploration)
- Explore the relation of the data to the underlying population.

3 common types of descriptive analytics

- Association Rules
- Sequence Rules
- Clustering

* Note that statistical analysis such as aggregation, mean, variance analysis, outlier detection etc. fall under descriptive analytics as well.

1. Association Rules

Explanation: Detect frequently occurring patterns between items.

Example:

- i. Detecting what products are frequently purchased together in a supermarket context. (aka Market Basket Analysis)
 - ii. Detecting what words frequently co-occur in a text document.
 - iii. Detecting what elective courses are frequently selected together in a university setting.
- Often used as a recommender system (Amazon, Netflix)

Association Rule Mathematical Notation

$$X \Rightarrow Y$$

Event X implicates event Y

whereby $X \subset I$, $Y \subset I$ and $X \cap Y = \emptyset$

X belongs to a
universal population

Y belongs to a
universal population

Intersection of X and Y is
not null. (100% happens)

- X is referred to as the rule antecedent. (Cause)
- Y is referred to as the rule consequent. (Effect)

Support and Confidence

- Key measures to quantify the strength of an association rule.
- The support of an item set is the percentage of total transactions that contains the item set. (Frequency of an item)
- The confidence measures the strength of the association.

$$\text{support}(X \cup Y) = \frac{\text{no of transactions supporting}(X \cup Y)}{\text{total number of transactions}}$$

$$\text{confidence}(X \rightarrow Y) = P(X|Y) = \frac{\text{support}(X \cup Y)}{\text{support}(X)}$$

Support Calculation Example

Transaction Identifier	Items
1	
2	Coke, beer, diapers
3	Cigarettes, diapers, baby food
4	Chocolates, diapers, milk, apples
5	Tomatoes, water, apples, beer
6	
7	Water, beer, baby food
8	Diapers, baby food, spaghetti
9	
10	Apples, wine, baby food

$$\text{support}(X \cup Y) = \frac{\text{no of transactions supporting}(X \cup Y)}{\text{total number of transactions}}$$

The association rule:

Baby Food & Diapers \Rightarrow Beer

Support:

= no of transactions supporting (baby food, diaper and beer) / 10 transactions
= 3/10 or 30%



When a customer buys baby food and diapers, the customer tends to also buy beer 30% of the time.

~Walmart beer and diapers

Confidence Calculation Example

Transaction Identifier	Items
1	
2	Coke, beer, diapers
3	
4	Chocolates, diapers, milk, apples
5	Tomatoes, water, apples, beer
6	
7	Water, beer, baby food
8	
9	
10	Apples, wine, baby food

$$\text{confidence}(X \rightarrow Y) = P(X|Y) = \frac{\text{support}(X \cup Y)}{\text{support}(Y)}$$

The association rule:

Baby Food & Diapers \Rightarrow Beer

Minimum confidence (*minconf*):

= support (Baby Food & Diapers & Beer) / support (Baby Food & Diapers)
(baby food & diapers) = 3/5 or **60%**



When a customer buys baby food and diapers, I am 60% positive (confident) that the customer will also buy beer.

~Walmart beer and diapers

Association Rule Mining

- Association rule mining follows a 2-step procedure:
 - i. Identification of all item sets having support above *minsup* (i.e. “frequent” item sets)
 - ii. Discovery of all derived association rules having confidence above *minconf*.

Minsup Identification

- Performed using Apriori Algorithm.
- Explanation: Every subset of a frequent item set is frequent as well and vice-versa.
- For each frequent item set k , generate all non-empty subsets of k
- For every non-empty subset s of k , output the rule $s \Rightarrow k-s$ if the confidence $> \text{minconf}$

Apriori Algorithm Example

Database



minsup = 50%

4 itemsets
(based on 4 transactions)

Various combinations is
computed. Discard
association rule < minsup

C_2

Itemsets	Support
{1, 2}	1/4
{1, 3}	2/4
{1, 5}	1/4
{2, 3}	2/4
{2, 5}	3/4
{3, 5}	2/4

C_3

Itemsets	Support
{2, 3, 5}	2/4

{1,3} and {2,3} give
{1,2,3}, but because {1,2}
is not frequent, you do not
have to consider it!

Minconf Discovery

- Confidence can be easily computed using the support values that were obtained during the frequent item set mining.
- For the frequent item set {baby food, diapers, beer}, the following association rules can be derived:

diapers, beer \Rightarrow baby food [*conf* = 75%]

baby food, beer \Rightarrow diapers [*conf* = 75%]

baby food, diapers \Rightarrow beer [*conf* = 60%]

beer \Rightarrow baby food and diapers [*conf* = 50%]

baby food \Rightarrow diapers and beer [*conf* = 43%]

diapers \Rightarrow baby food and beer [*conf* = 43%]

Lift Measure

	Tea	Not Tea	Total
Coffee	150	750	900
Not coffee	50	50	100
Total	200	800	1,000

- Consider the association rule $\text{tea} \Rightarrow \text{coffee}$.
- Support = $100/1,000$ (10%)
- Confidence = $150/200$ (75%) [Wow! Seems interesting]
- Closer inspection reveals that the prior probability of buying coffee equals $900/1,000$ (90%)
- Hence, a customer who buys tea is less likely to buy coffee than a customer about whom we have no information. (Random target)

Lift Measure Calculation

$$Lift(X \rightarrow Y) = \frac{support(X \cup Y)}{support(X) \cdot support(Y)}$$

- Lift value less than 1 indicates a negative dependence/substitution effect.
- Lift value larger than 1 indicates a positive dependence/complementary effect.
- In our example, the lift value equals 0.89, which indicates the expected substitution effect between coffee and tea.

2. Sequence Rules

Explanation: Detect sequences of events.

Example:

- i. Detecting sequences of purchase behavior in a supermarket context.
- ii. Detecting sequences of web page visits in a web mining context (Clickstream analysis).
- iii. Detecting sequences of words in a text document.

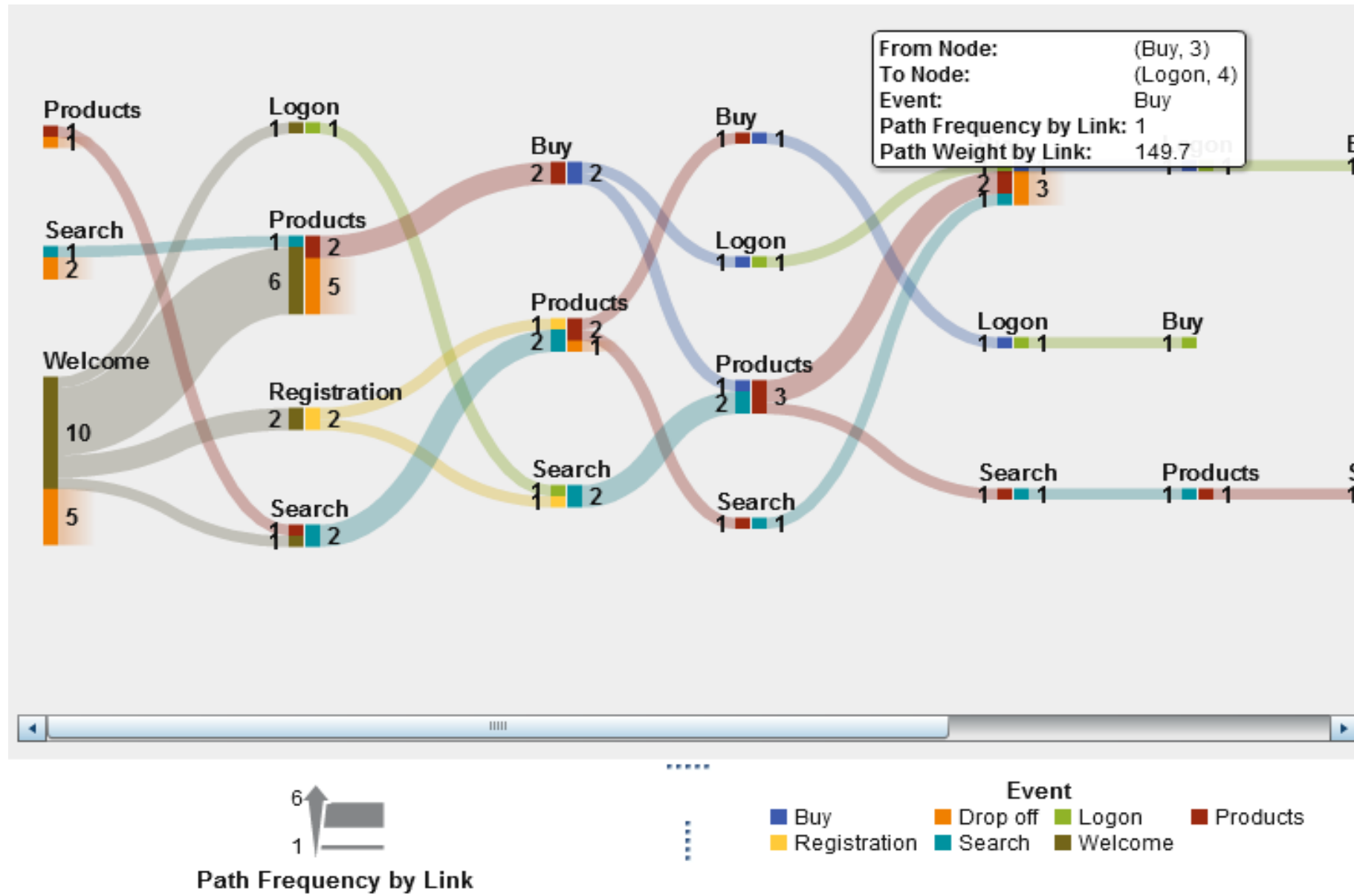
What is Sequence Rules?

- Given a set of customer transactions, the goal of mining sequential rules is to find the maximum sequences among all sequences that have certain minimum support and confidence.

Home page \Rightarrow Electronics \Rightarrow Cameras and Camcorders \Rightarrow Digital
Cameras \Rightarrow Shopping cart \Rightarrow Order confirmation \Rightarrow Return to shopping

- Effectively represented using Sankey diagram.
- Also mined using Apriori algorithm.

Sankey Diagram



Important things to note:

- Transaction time
 - How long a user spends viewing a page
 - How many clicks a user performed to successfully make a transaction etc.
- Sequence field
 - Starting point of transaction (eg. Welcome page, redirected from search engine etc.)
 - Dropout point (eg. After adding to cart, after searching for a particular item etc)

Apriori in Sequence Rules

Session ID	Page	Sequence
2	B	1
2	C	2
4	A	1
4	B	2
4	D	3
5	D	1
5	C	1
5	A	1

A sequential version can then be obtained as follows:

Session 1: A, B, C

Session 2: B, C

Session 3: A, C, D

Session 4: A, B, D

Session 5: D, C, A

Question: Calculate the support & confidence for sequence rule $A \Rightarrow C$.

1st approach: A may lead to B then only C.

Support = 2/5 (40%)

Confidence = 2/4 (50%)

2nd approach: A leads to C directly.

Support = 1/5 (20%)

Confidence = 1/4 (25%)

3. Clustering (Segmentation)

Explanation:

- Detects homogeneous (identical) segments of observations.
- Maximize homogeneity within segment (cohesive)
- Maximize heterogeneity between segments (separation)

Example:

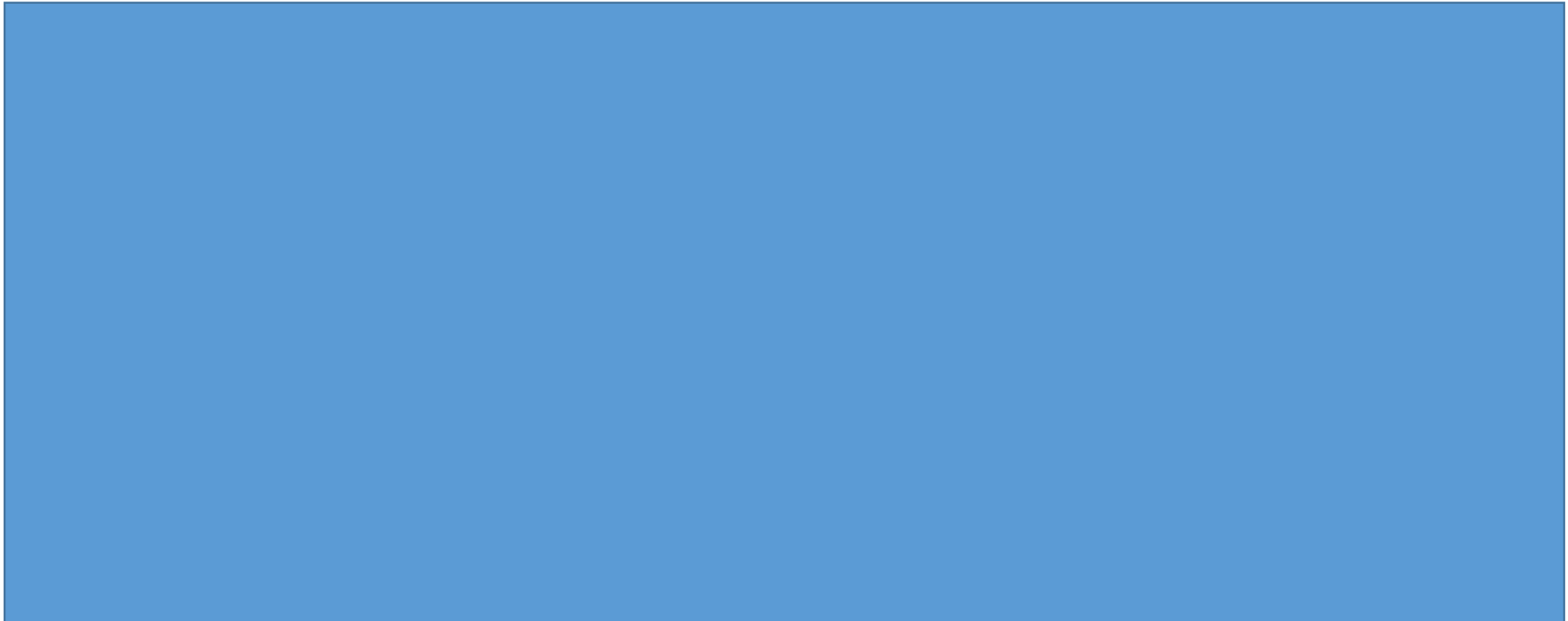
- i. Differentiate between brands in a marketing portfolio
- ii. Segment customer population for targeted marketing

Various data types of clustering data

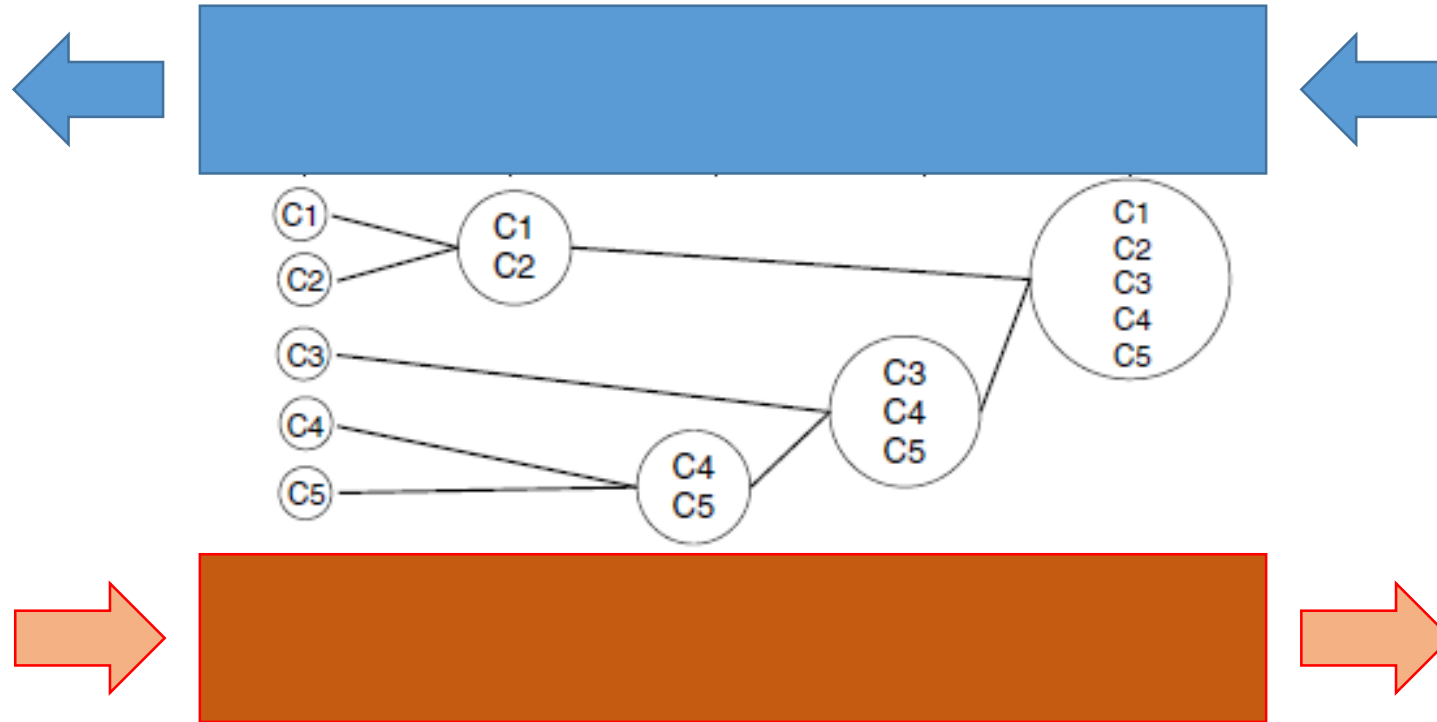
- Demographic data (age, gender, population, income, education etc.)
- Lifestyle (Standard of living data i.e. spending habits, living expenses etc.)
- Attitudinal (Likes and Dislikes toward a particular product/service)
- Behavioral (Information produced as a result of actions i.e sites visited, the apps downloaded, or games played)
- RFM [Recency, Frequency, Monetary] – How recent, how frequent, and how much customers spend on transaction.
- Acquisitional (Data collected/acquired from sensors)
- Social Network (Friends, activities, etc)
- Etc.

Clustering Techniques

- Hierarchical and non-hierarchical



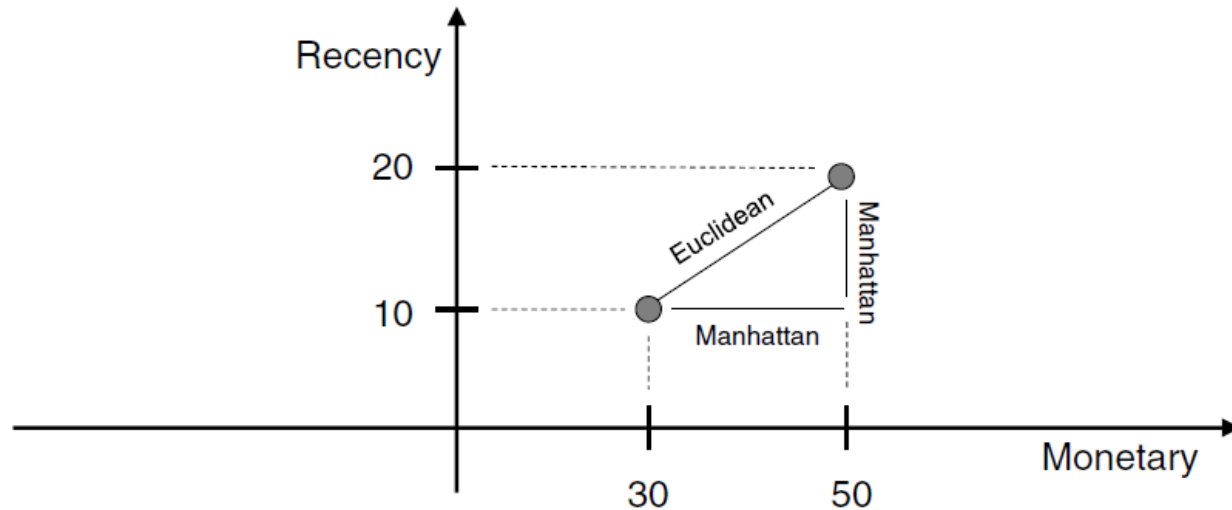
Hierarchical Clustering



- Divisive clustering
 - Initializes at the complete dataset level
 - Breaks up the data into smaller clusters until one observation per cluster remains.
- Agglomerative clustering
 - Merges data from smaller clusters based on similarity until they make up one cluster.

Similarity Measurement

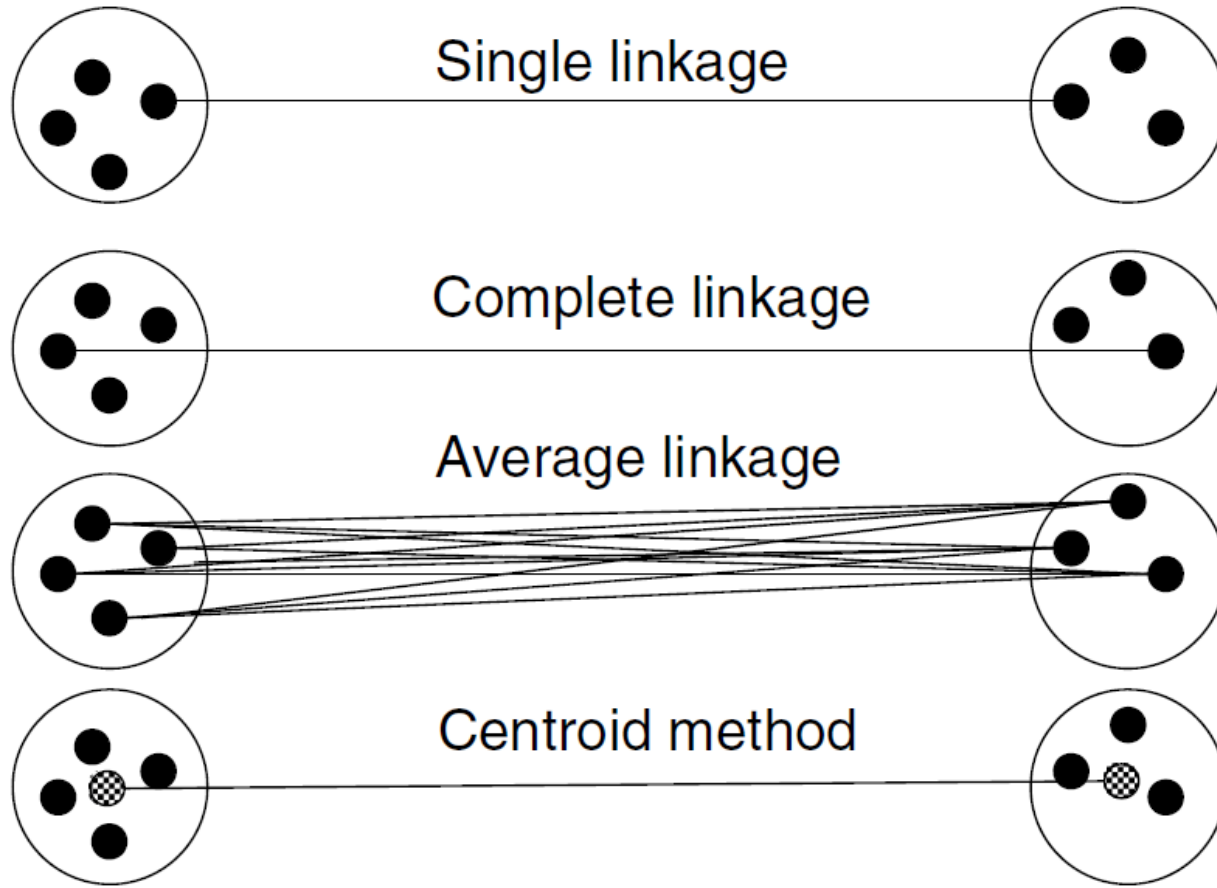
- Similarity rule is needed to decide on merging/splitting
- Euclidean distance and Manhattan distance are examples of some popular distance measurement.



$$\text{Euclidean : } \sqrt{(50 - 30)^2 + (20 - 10)^2} = 22$$

$$\text{Manhattan : } |50 - 30| + |20 - 10| = 30$$

Distance calculation between two clusters



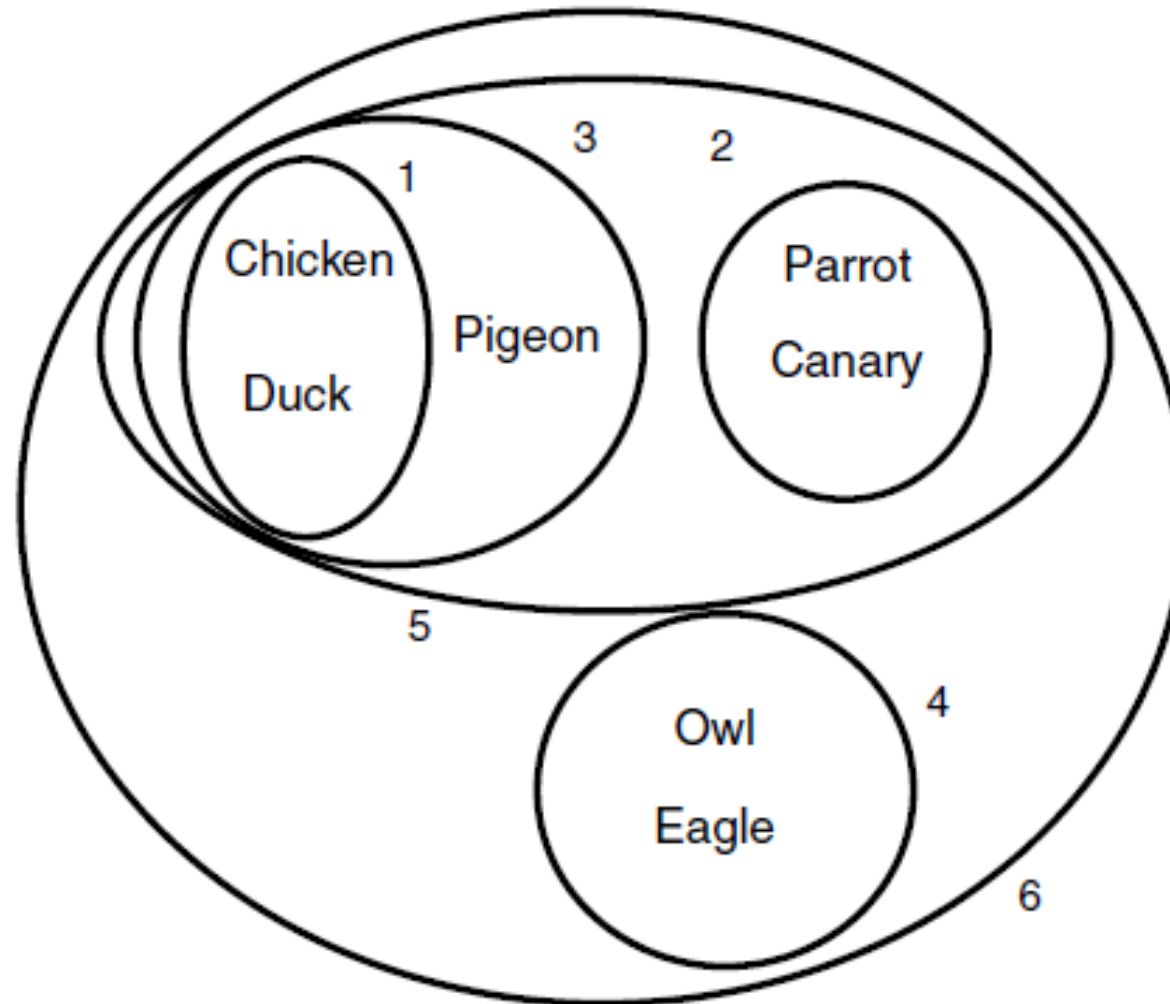
The single linkage method defines the distance between two clusters as the shortest possible distance, or the distance between the two most similar objects.

The complete linkage method defines the distance between two clusters as the biggest distance, or the distance between the two most dissimilar objects.

The average linkage method calculates the average of all possible distances.

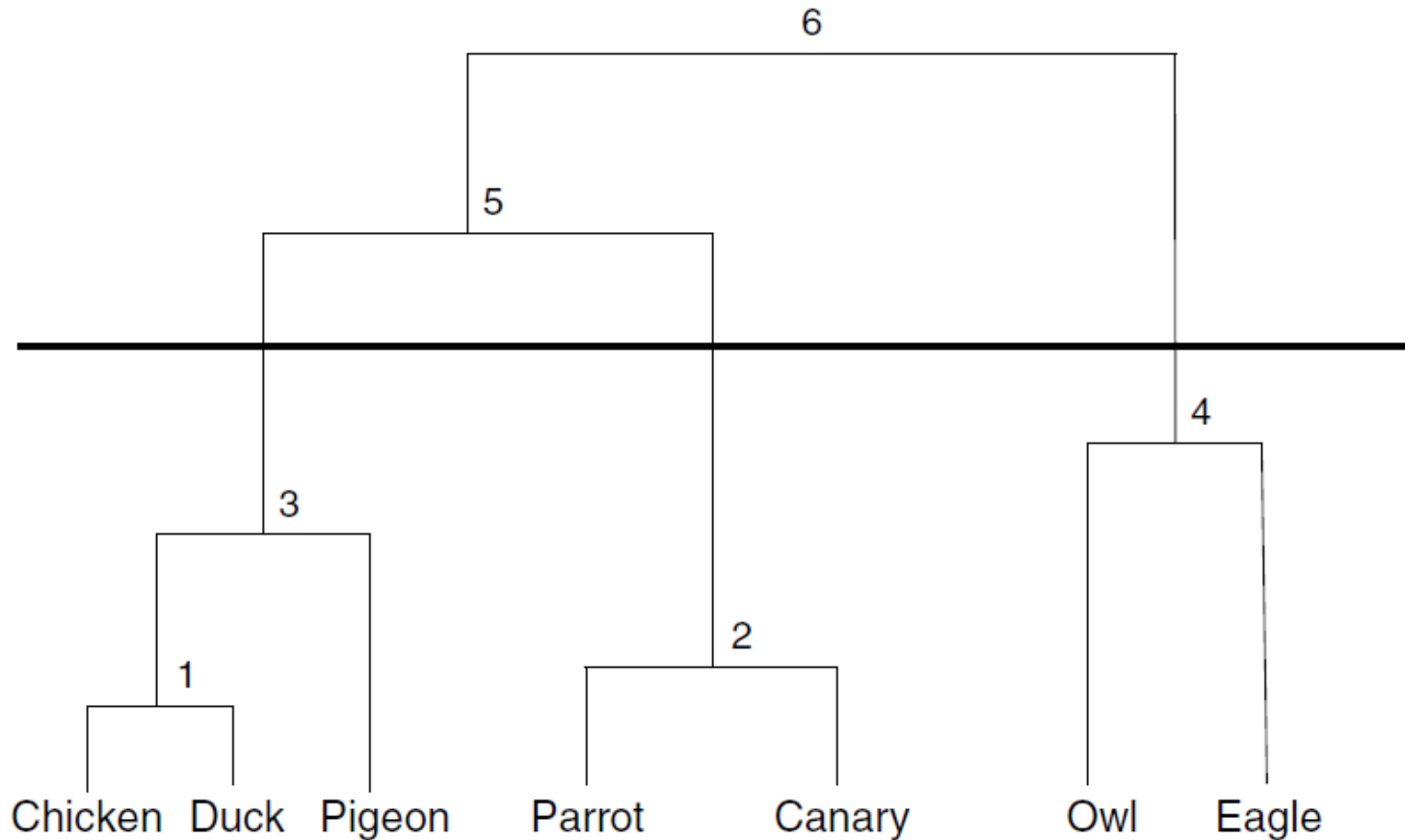
The centroid method calculates the distance between the centroids of both clusters.

Example of Clustering Birds



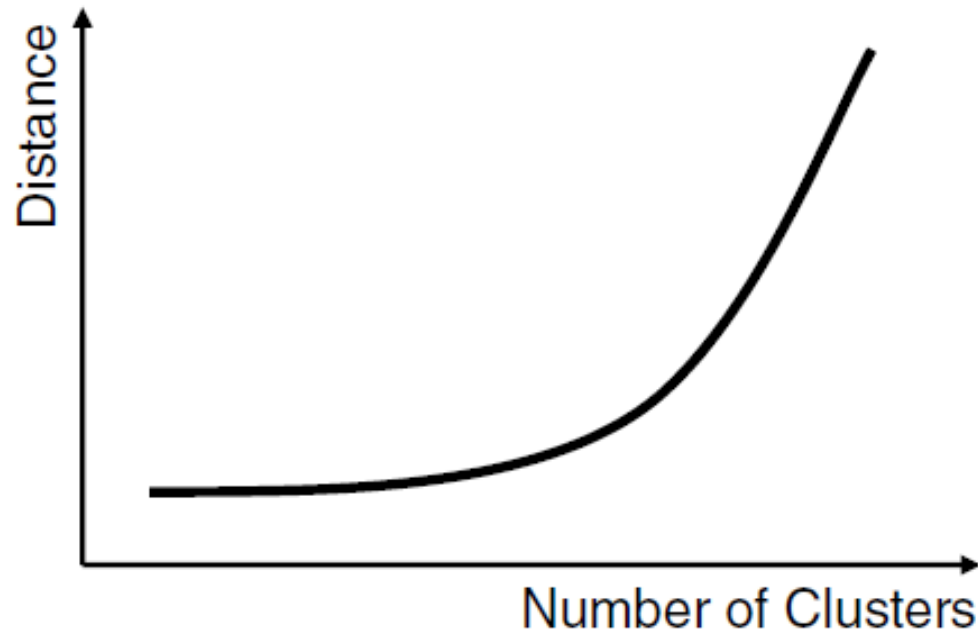
The number indicates clustering steps.

Dendrogram



- Used to determine the number of clusters.
- Records the sequence of merges
- Vertical/horizontal scale gives the distance between two clusters
- Dendrogram can be 'cut' at desired level to determine optimal clustering.

Scree Plot



- Plot of distance at which clusters are merged
- Elbow point indicates optimal clustering

K-means Clustering Algorithm

- A non-hierarchical procedure that works as such:

Step 1. Select k observations as initial cluster centroids (**seeds*).

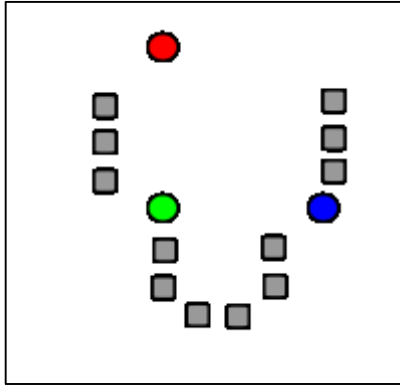
Step 2. Assign each observation to the cluster that has the closest centroid

Step 3. When all observations have been assigned, recalculate the positions of the k centroids.

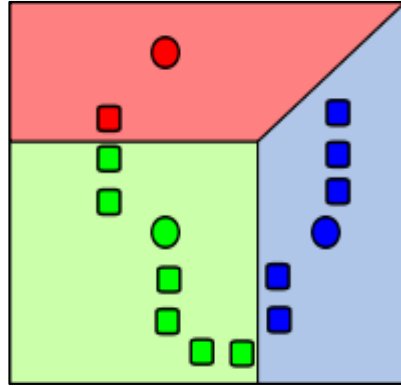
Step 4. Repeat until the cluster centroids no longer change.

*It is recommended to try out different seeds to verify the stability of the clustering solution.

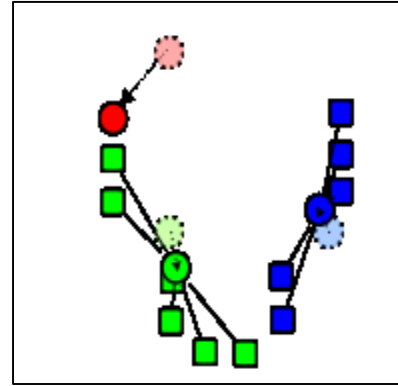
Demonstration of the k-means clustering algorithm



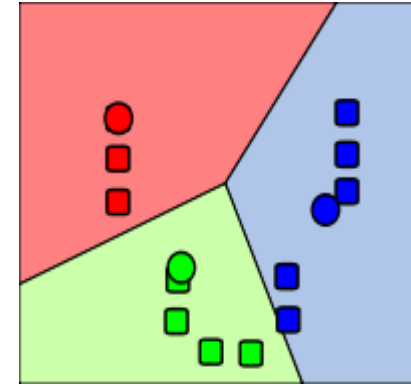
1. k initial “seed” (in this case $k=3$) are randomly generated within the data domain (shown in color).



2. k clusters are created by associating every observation with the nearest mean.

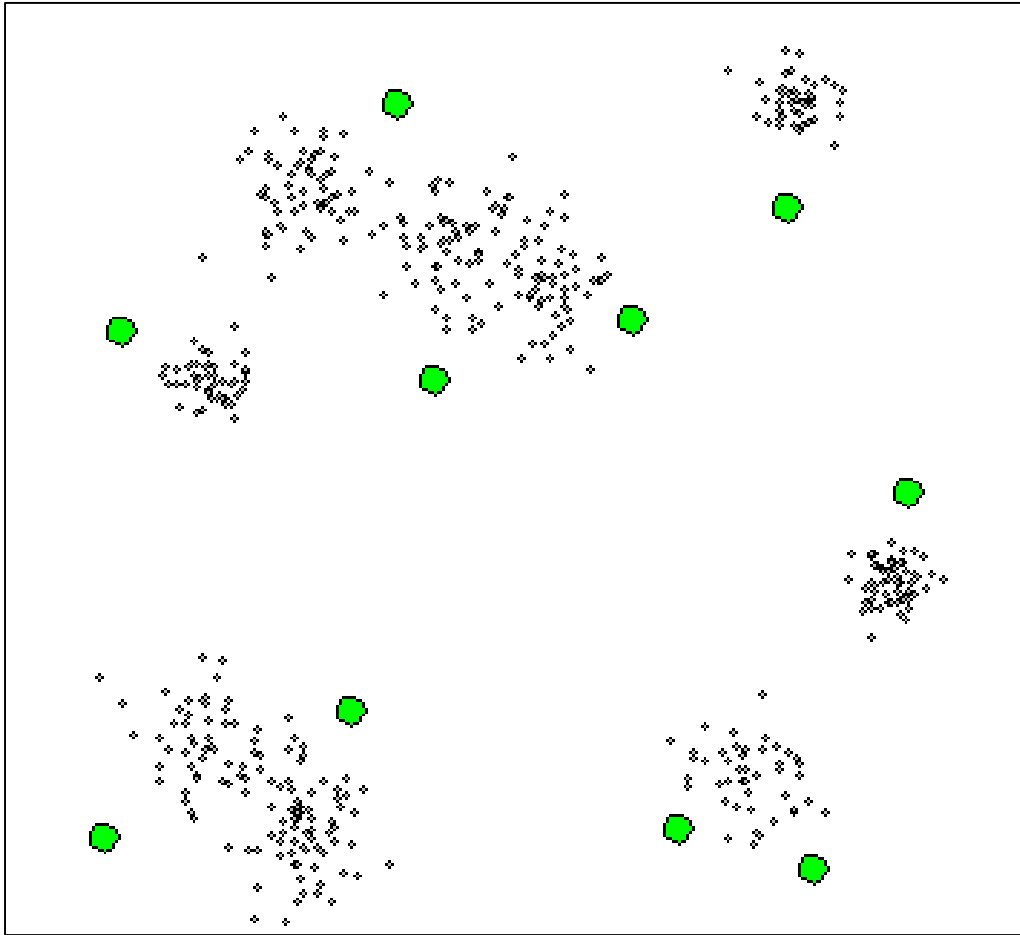


3. The centroid of each of the k clusters becomes the new mean.



4. Steps 2 and 3 are repeated until convergence has been reached.


Competitive learning



- Competitive learning is useful for clustering input patterns into a discrete set of output clusters.
- **Unit** is a special type of data point (an artificial particle)
- **Unit** points are dynamic

Competitive Learning

- The position of the unit for each data point can be expressed as follow:



$p(t+1) = a(p(t) - x) d(p(t), x)$

Where

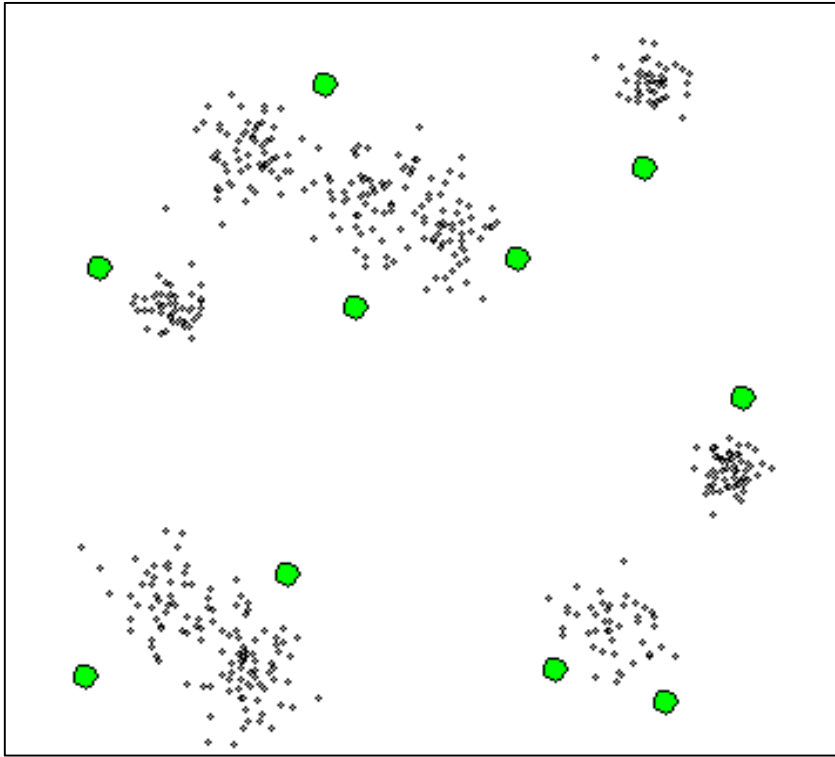
a = learning rate

$d(p, x)$ = distance scaling function

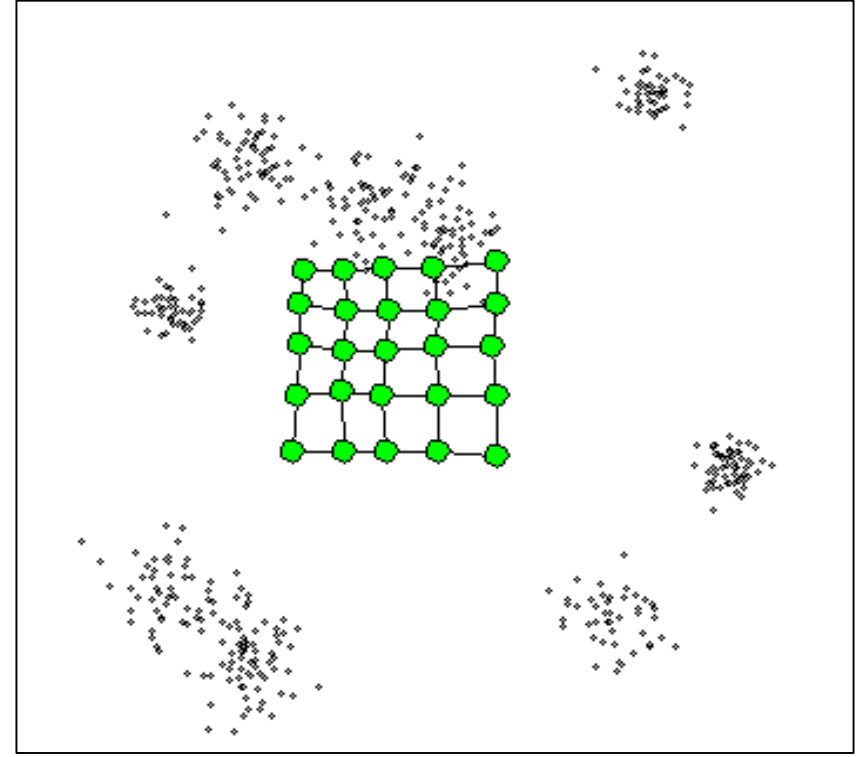
Self-Organizing Maps

- A 2-layer feedforward neural network that is trained using unsupervised learning.
- Enables the visualization and clustering of high-dimensional data on a low-dimensional grid of neurons (Allows data of ≥ 3 dimensional to be mapped onto 2-dimensional grid)
- Neurons of the output layer are usually ordered in two-dimensional rectangular/hexagonal grid.
- It is based on Competitive Learning
- Difference is that the units are all interconnected in a **grid**.

SOM



Competitive Learning



SOM

SOM

- The unit closest to the input vector is call Best Matching Unit (BMU).
- The BMU and other units will adjust its position toward the input vector.
- The update formula is

$$W_v(t + 1) = W_v(t) + \Theta(v, t) \alpha(t)(D(t) - W_v(t))$$



where

$Wv(t)$ = weight vector

$\alpha(t)$ = monotonically decreasing learning coefficient

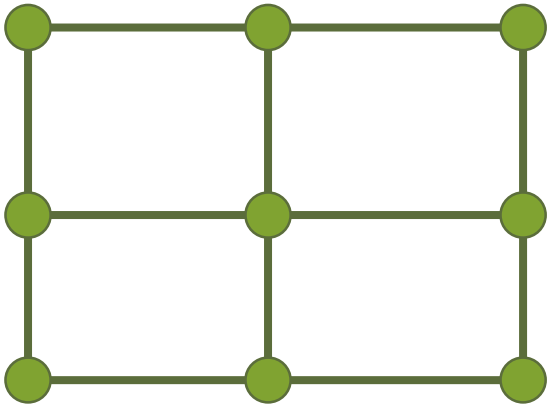
$D(t)$ = the input vector

$\Theta(v, t)$ = neighborhood function

- This process is repeated for each input vector for a number of cycles.

SOM Algorithm

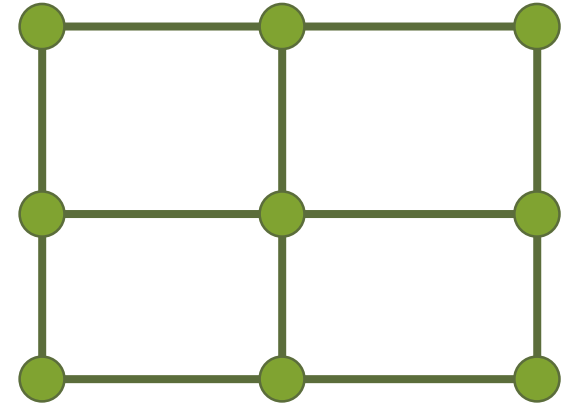
1. Randomize the map's nodes' weight vectors



2. Pump in input vector

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ \vdots \\ \vdots \\ x_n \end{bmatrix}$$

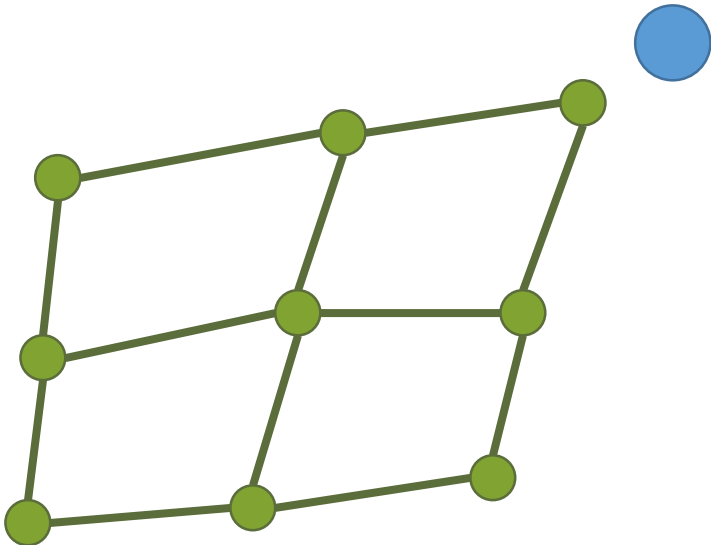
3. Traverse each node in the map



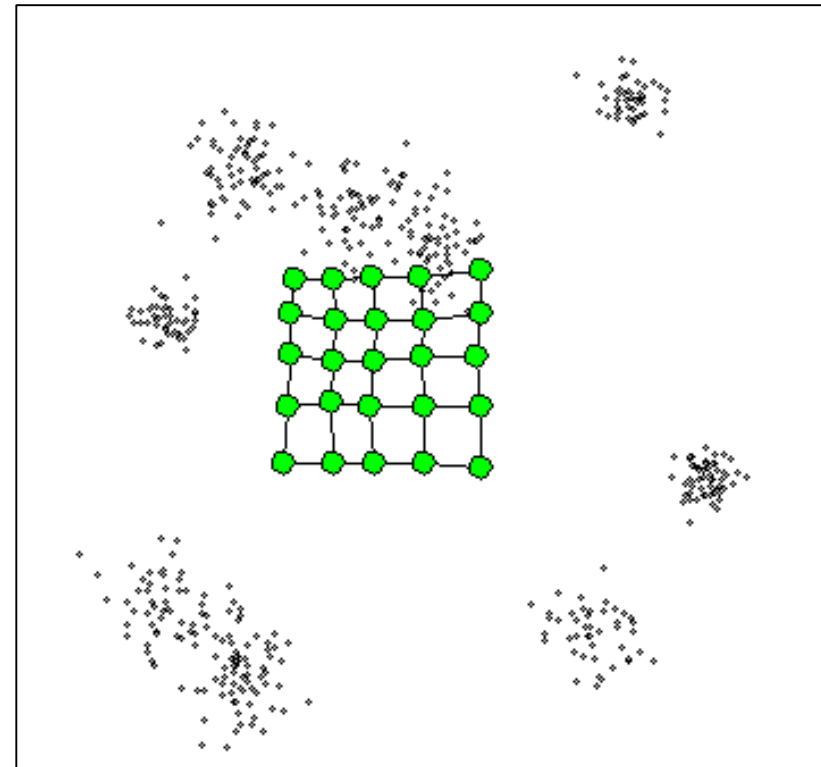
SOM Algorithm

4. Update nodes in the neighborhood by pulling them closer to the input vector

$$W_v(t + 1) = W_v(t) + \Theta(v, t) \alpha(t)(D(t) - W_v(t))$$



5. Increment t and repeat the process until termination criteria



Visualization with SOM

