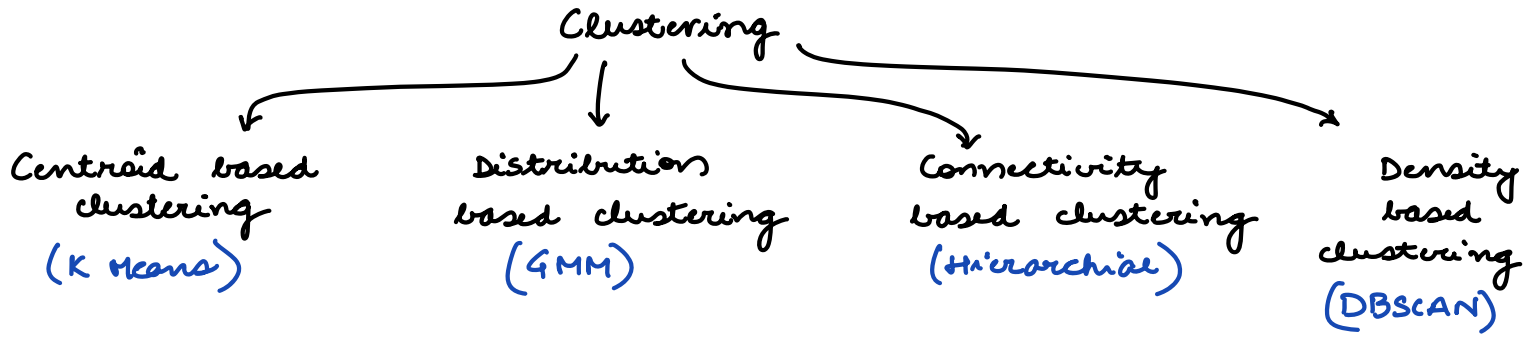
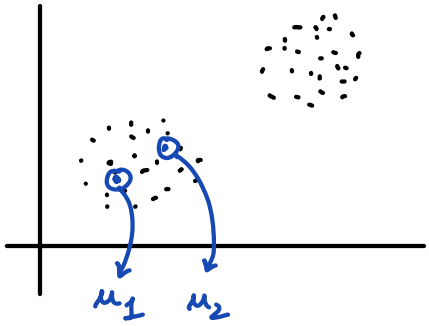


CENTROID BASED CLUSTERING :

Illustration

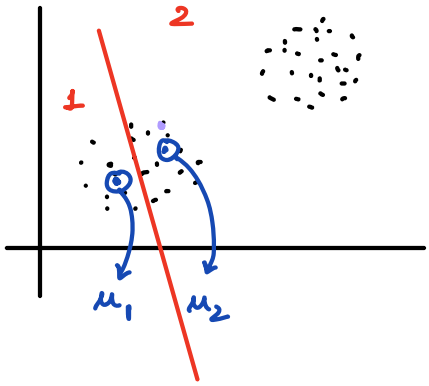


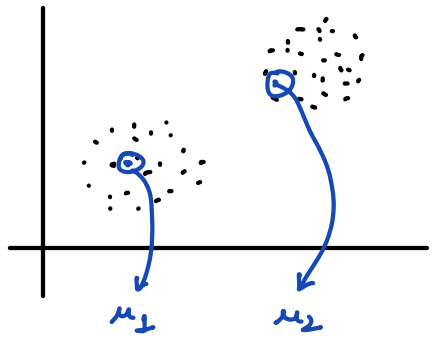
Initialization

Iteration 1

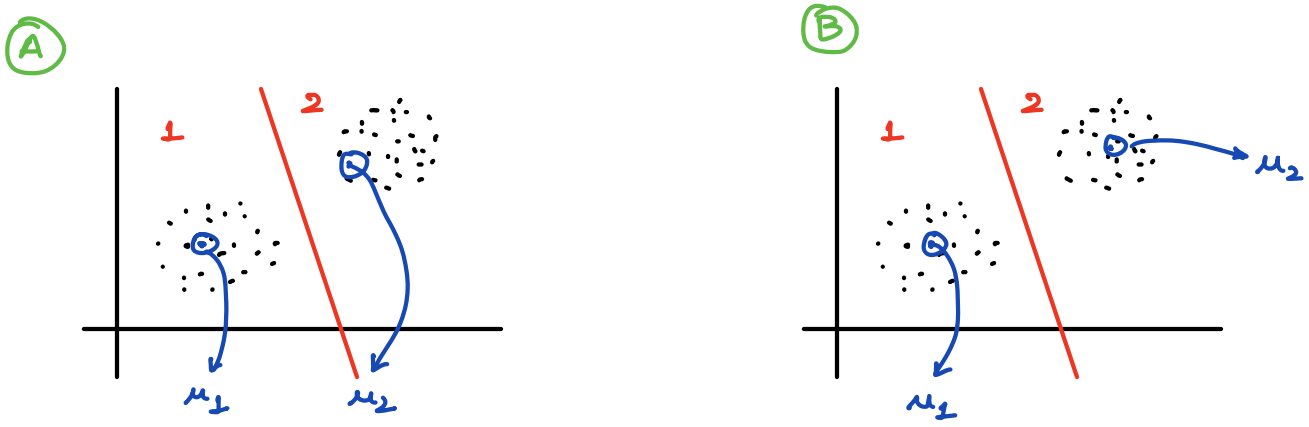
(A) find cluster assignment (assign each point to closest center)

(B) Recompute μ

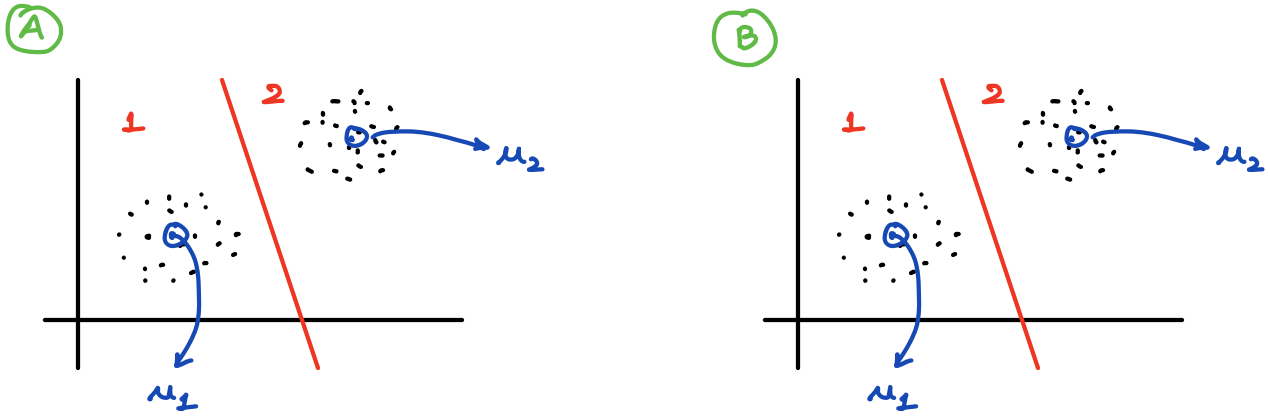




Iteration 2 :

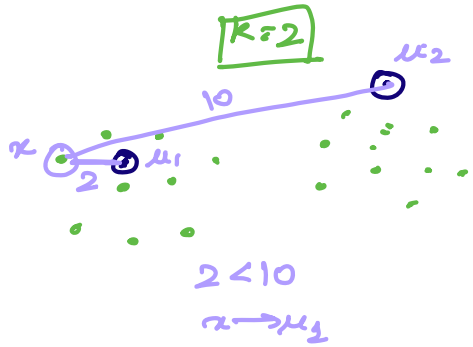


Iteration 3 : Algo converged (Assignment and μ doesnot change)



E Step

M Step



Code (E Step)

- go to every data point → m
  - calculate the distance of this → k·n
    data point from μ of every cluster
  - then find out the minimum distance
  - Assign data point to the cluster with min distance

↳ $m(nk + k + 1) = O(mnk)$

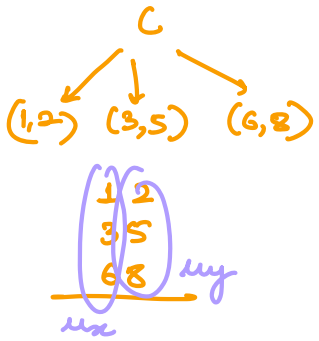
Code (M Step)

- Iterate over all clusters

Iterate over all points present in the cluster and take the mean

m: total no. of datapoints

$$\frac{m}{2}\cdot n + \frac{m}{4}\cdot n + \frac{m}{4}\cdot n = m\cdot n$$



Time Complexity:

$x^{(i)} \in R^n$

K: no of clusters

m = # points

$$O(m\cdot k\cdot n + n\cdot m)$$

$m\cdot k\cdot n$ → E Step

$n\cdot m$ → M Step

↳ for a particular cluster: $n\cdot m_k$

m_k → no. of points in k cluster

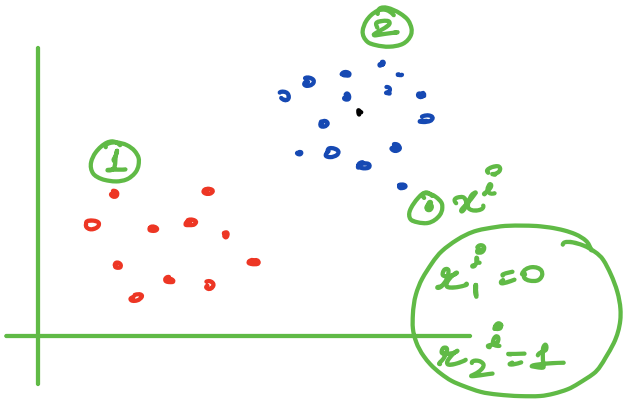
$$\sum_{k=1}^{K} n m_k : n m_1 + n m_2 + \cdots n m_k = n\cdot m$$

Maths:

$$x = \{x^1, x^2, x^3 \cdots x^m\}$$

$$\mu = \{\mu_1, \mu_2 \cdots \mu_k\}$$

K = clusters



Loss/Inertia

$$J = \sum_{i=1}^{m} \sum_{k=1}^{K} r_k^{(i)} \left(x^{(i)} - \mu_k\right)^2$$

$\sum_{i=1}^{m}$ → all datapoints

$\sum_{k=1}^{K}$ → all clusters

$\left(x^{(i)} - \mu_k\right)^2$ → distance

} compute the distance of every point from every cluster.

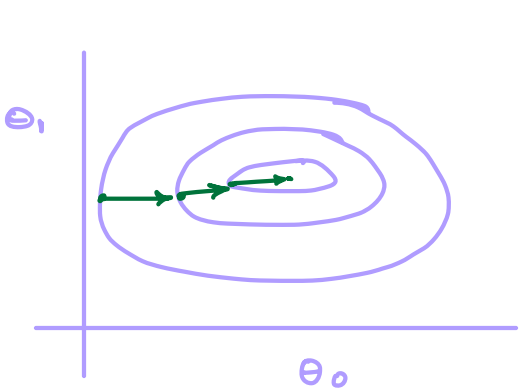
$$r_k^{(i)} = \begin{cases} 1 & \text{if } i\text{th point} \in k\text{th cluster} \\ 0 & \text{otherwise} \end{cases}$$

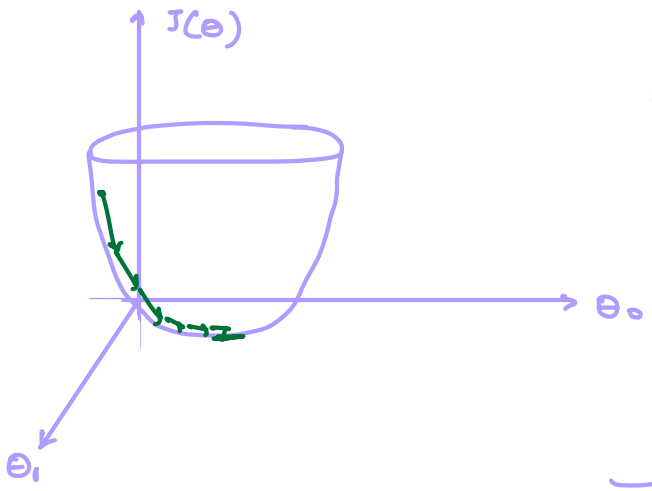
We want to learn $r_k^{(i)}$ and μ_k

This is done by using COORDINATE DESCENT.

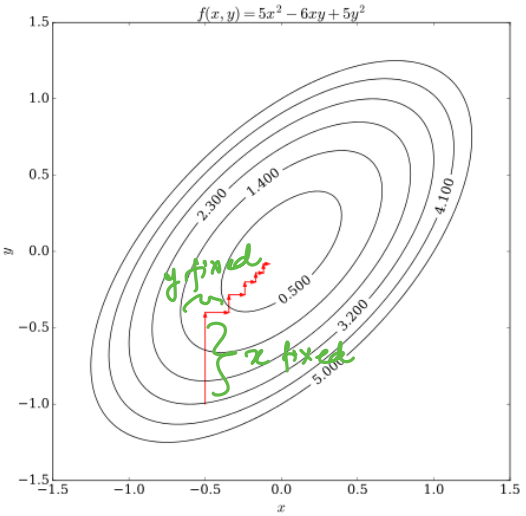
↓

optimise one set of variable at a time while keeping others fixed.





GRADIENT DESCENT



Minimize your loss wrt only one variable at a time.

$$\min(10, 20, 30, 40, 5, 50) = 5$$

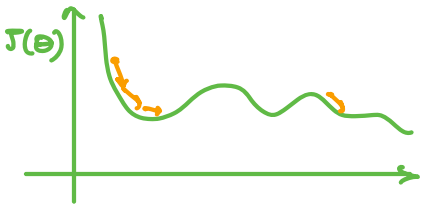
$$\text{argmin}(10, 20, 30, 40, 5, 50) = 4$$

for what value of j $\|x^{(i)} - \mu_j\|^2$ is minimum

$$r_k^{(i)} = \begin{cases} 1 & \text{if } k = \underbrace{\arg\min_j \|x^{(i)} - \mu_j\|^2}_{= 4} \\ 0 & \text{otherwise} \end{cases}$$

(40, 5, 50) (3, 4, 5)

$$J = \sum_{i=1}^{m} \sum_{k=1}^{K} r_k^{(i)} \left(x^{(i)} - \mu_k\right)^2$$



K Means loss fxⁿ is non convex.

$$2 \sum_{i=1}^{m} r_k^{(i)} \left(x^{(i)} - \mu_k\right) = 0$$

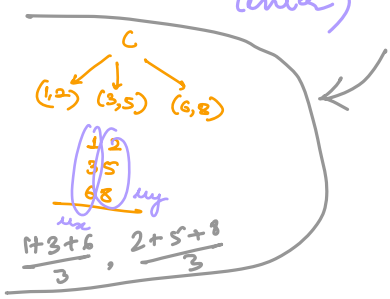
$$\sum_{i=1}^{m} r_k^{(i)} x^{(i)} - \mu_k \sum_{i=1}^{m} r_k^{(i)} = 0$$

$$\mu_k = \frac{\sum_{i=1}^{m} r_k^{(i)} x^{(i)}}{\sum_{i=1}^{m} r_k^{(i)}}$$

→ points in kth cluster.

M Step

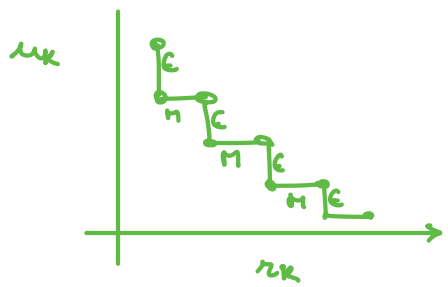
(Cluster center) ← μ_k



no of points in kth cluster

M Step: find μ_k keeping $r_k^{(i)}$ fixed

E Step: find $r_k^{(i)}$ keeping μ_k as fixed.



it will converge to local minima.

Algo:

initialize cluster centroids $\mu_1, \mu_2 \cdots \mu_k$ $\quad \mu \in \mathbb{R}^n$

Repeat until convergence:

- for every data point i, set

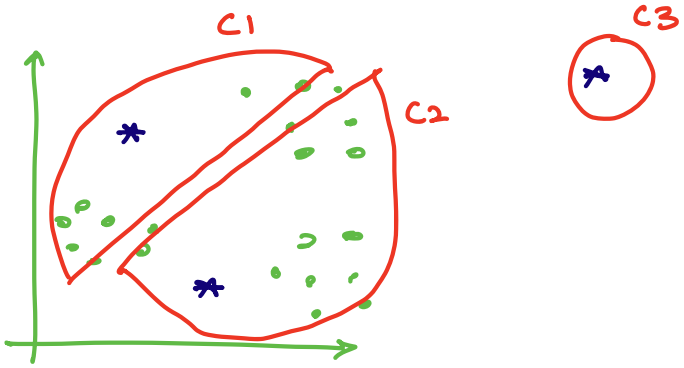
$$r_k^{(i)} = \begin{cases} 1 & \text{if } k = \arg\min_j \|x^{(i)} - \mu_j\|^2 \\ 0 & \text{otherwise} \end{cases}$$ — E Step

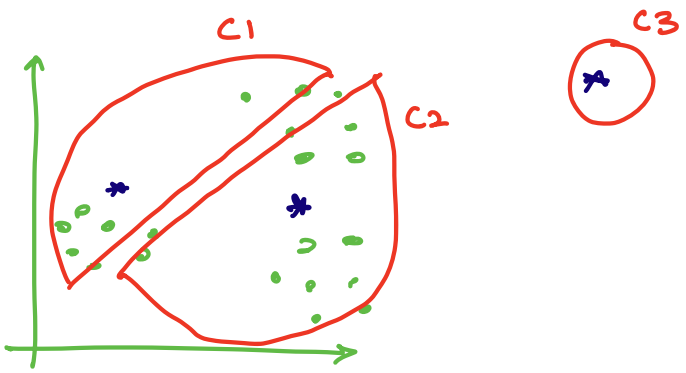
- for each k, set $\mu_k = \dfrac{\sum_{i=1}^{m} r_k^{(i)} x^{(i)}}{\sum_{i=1}^{m} r_k^{(i)}}$ — M Step

When k-means doesnot work ?

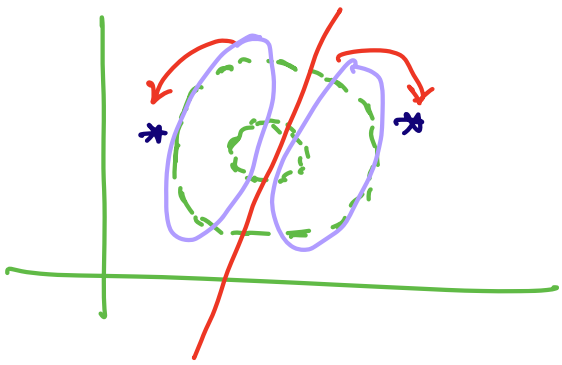
- random initialisation (empty)
- non linearly separable (O)
- one cluster is large as compared to other.

Random Initialization

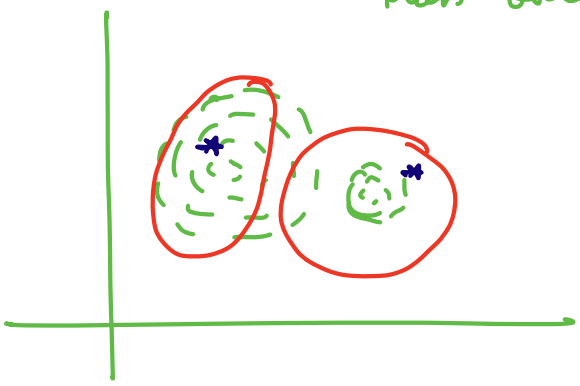




Non linearly Separable



one cluster is longer than other

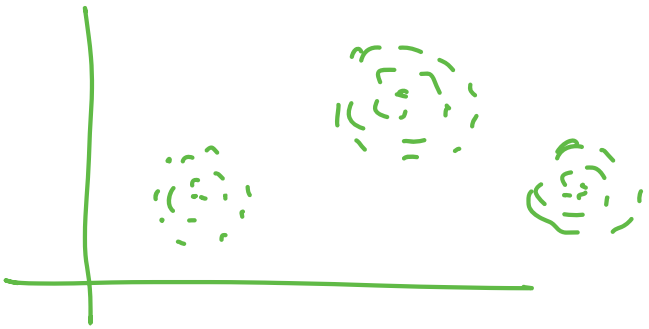


k = no of clusters.

↳ hyperparameter

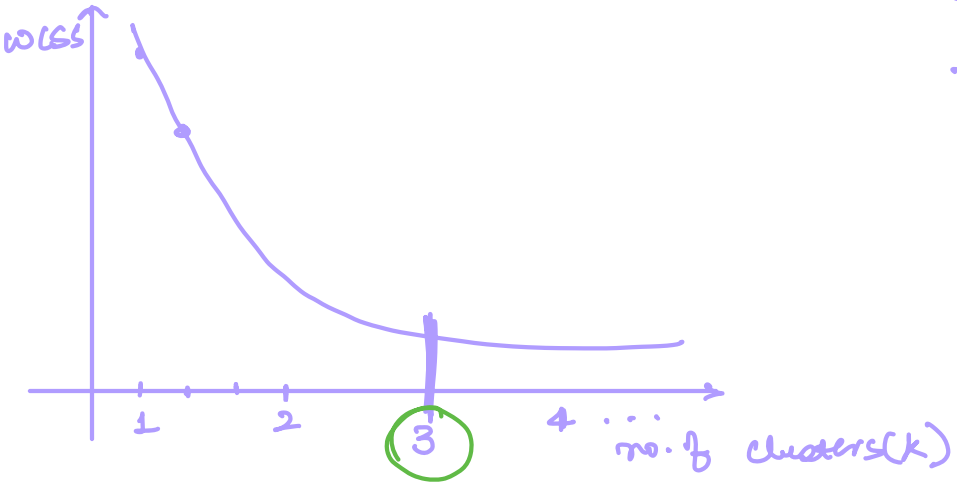
k = 1
2
3
⋮

Within cluster sum of Squares (WCSS)



$$WCSS = \sum_{i \text{ in } C1} \text{distance}(x^{(i)}, \mu_1) + \sum_{i \text{ in } C2} \text{distance}(x^{(i)}, \mu_2) \quad \cdots\cdots$$

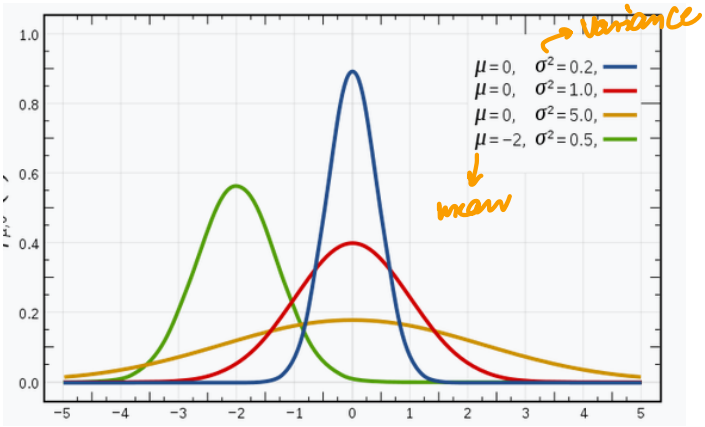
Elbow Method

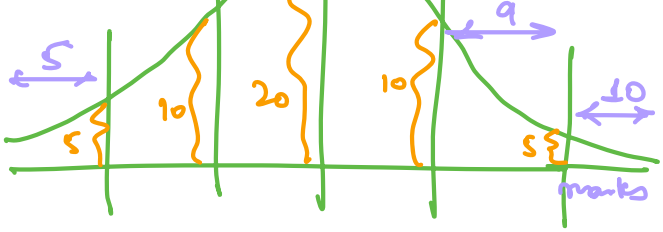


- no. of clusters = 2
- k means algo converge
- WCSS

Gaussian Mixture Models

→ Normal Distribution

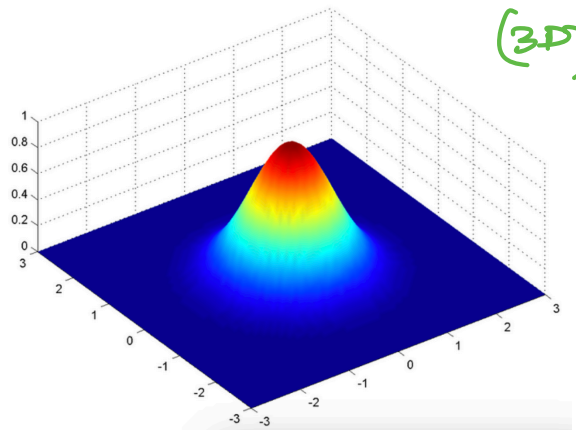


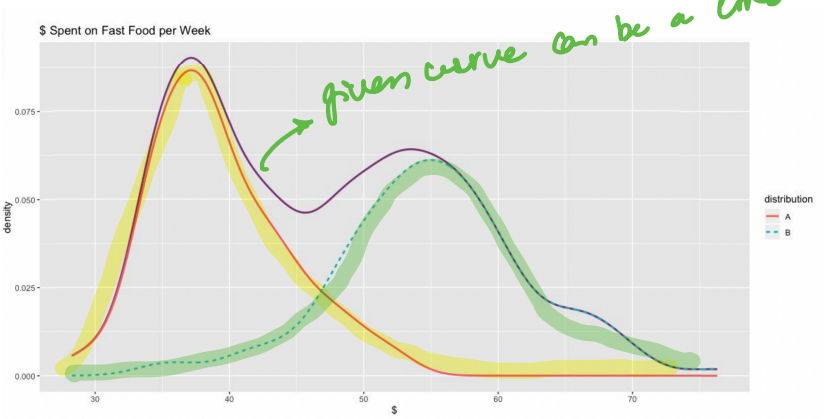


$$f(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$\sigma^2 = \frac{\Sigma(x_i - \bar{x})^2}{n-1}$$

(3D)





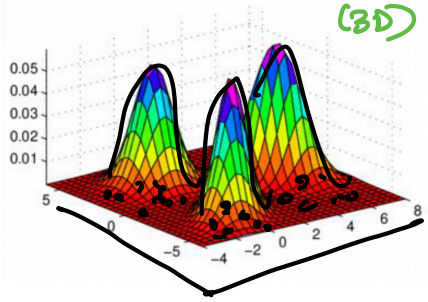
given curve can be a combination of 2 different normal distribution curves.

Multivariate Normal Distributions

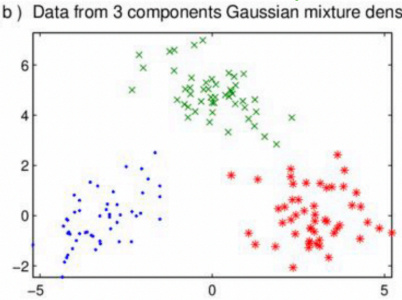
(2D)

(a) 3 components Gaussian mixture density

(3D)



(b) Data from 3 components Gaussian mixture density

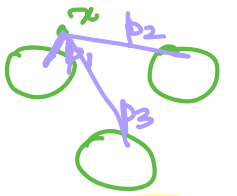


K means

- Hard Assignment
- All Variances the Same
- Roughly same # of datapoints

GMM

- Soft (probabilistic) Assignment
- Variances can be different
- Explicitly model # of data points



K-Means Algorithm

1. Choose **k** random points to be cluster centers
2. For each data point, assign it to the cluster whose center is closest
3. Using these assignments, recalculate the **centers**
4. Repeat 2 and 3 until either:
   a. Cluster membership does not change
   b. Centers change only a tiny amount

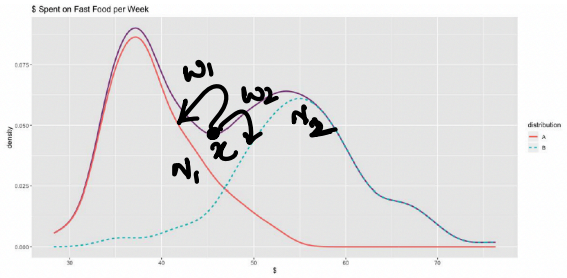
Gaussian Mixture Model (EM Algorithm)

1. Choose **k** random points to be cluster centers (**or estimate using k-means...etc)**
2. For each data point, calculate the **probability** of belonging to each cluster
3. Using these probability weights, recalculate the **means + variances** (and weights)
4. Repeat 2 and 3 until **distributions converge.**

$$p(x) = \sum_{k=1}^{K} w_k \, p_k(x)$$

$$p(x) = \sum_{k=1}^{K} w_k \, p_k(x)$$

$$p(x) = \sum_{k=1}^{K} w_k \, \mathcal{N}(x|\mu_k, \Sigma_k)$$



$$p(x) = \sum_{k=1}^{K} w_k \, \mathcal{N}(x|\mu_k, \Sigma_k)$$

Ref : https://www.youtube.com/watch?v=sAjOZ9hkn84

Probability of being in group k

Likelihood of seeing x in group k

$$p(x) = \sum_{k=1}^{K} w_k \, \mathcal{N}(x|\mu_k, \Sigma_k)$$

Posterior Probabilities

Prior probability of being in cluster k

Likelihood of seeing x in cluster k

$$p(\text{cluster } k|x) = \frac{w_k \, \mathcal{N}(x|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} w_j \, \mathcal{N}(x|\mu_j, \Sigma_j)}$$

Posterior probability of being in cluster k

$r_k^{(i)}$

Maximum Likelihood Estimation

Probability of being in group k

Likelihood of seeing x in group k

$$p(x) = \sum_{k=1}^{K} w_k \, \mathcal{N}(x|\mu_k, \Sigma_k)$$

$$p(\mathbf{X}|\mathbf{w}, \mu, \mathbf{\Sigma}) = p(x_1, x_2, ..., x_n|\mathbf{w}, \mu, \mathbf{\Sigma}) = \prod_{n=1}^{N} \sum_{k=1}^{K} w_k \, \mathcal{N}(x_n|\mu_k, \Sigma_k)$$

$$log(p(\mathbf{X}|\mathbf{w}, \mu, \mathbf{\Sigma})) = \sum_{n=1}^{N} log \left\{ \sum_{k=1}^{K} w_k \, \mathcal{N}(x_n|\mu_k, \Sigma_k) \right\}$$

Goal: choose w, μ, Σ that maximize the log likelihood

mean

covariance

Formulas (E-Step)

$$r_{nk} = \frac{w_k N(x_n|\mu_k, \Sigma_k)}{\sum_j w_j N(x_n|\mu_j, \Sigma_j)}$$

$$p(\text{cluster } k|x) = \frac{w_k \mathcal{N}(x|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} w_j \mathcal{N}(x|\mu_j, \Sigma_j)}$$

Prior probability of being in cluster k

Likelihood of seeing x in cluster k

Posterior probability of being in cluster k

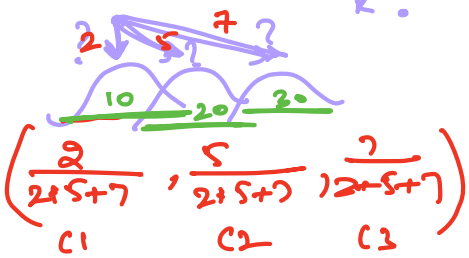
Responsibilities are the posterior probability of a data point being in cluster k

GMM

E step:

$w_1 = \frac{10}{10+20+30}$

x_k^i ?



$\left(\frac{2}{2+5+7}, \frac{5}{2+5+7}, \frac{7}{2+5+7}\right)$

C1 C2 C3

Formulas (M-Step)

$$\mu_k = \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} x_n$$

N → data points

$$N_k = \sum_{n=1}^{N} r_{nk}$$

$$\Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} (x_n - \mu_k)(x_n - \mu_k)^T$$

$w_2 = \frac{20}{10+20+30}$

$w_3 = \frac{30}{10+20+30}$

$w_k = \frac{N_k}{N}$

- GMM does **soft assignment**, every data point belongs to every cluster with some probability
- Data points that are more likely to be in a cluster have **more influence** over its parameters
- GMM uses the EM algorithm to iteratively update the cluster distributions. It first assigning a responsibility to each data point (**E-step**), and then using them to calculate weighted means and variances for each cluster (**M-step**)
- Responsibilities measure the **probability of a data point being in each cluster** (technically the **posterior probability**).
- Responsibilities contain information about **how common a cluster is** (w_k) as well as the **likelihood of a data point belonging to that cluster**

Normal Distribution eqn.

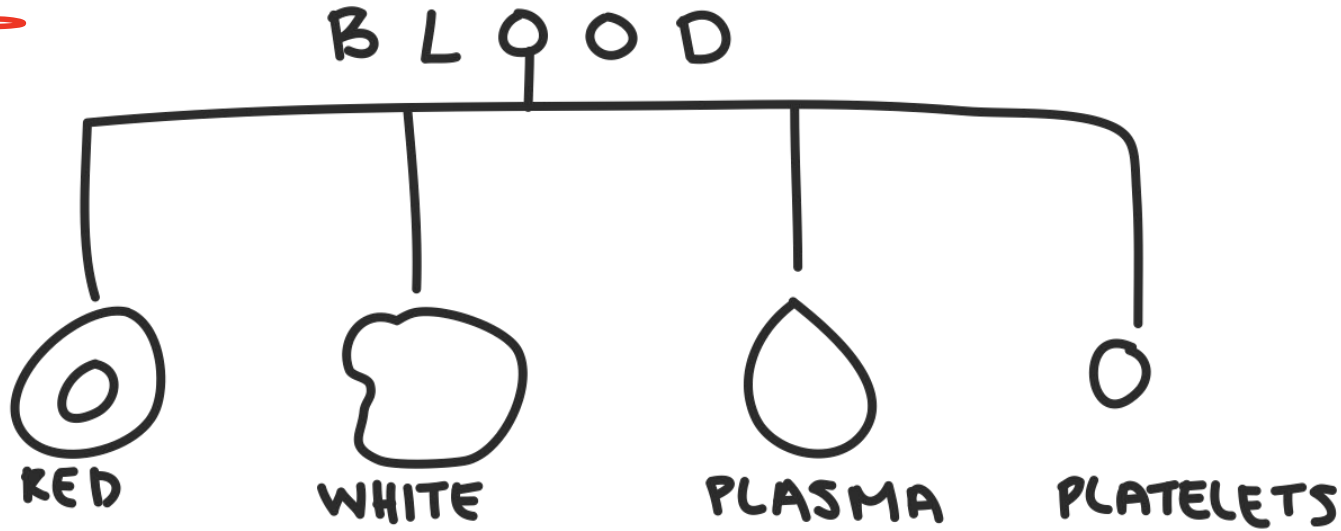
GMM

Hierarchical Based Clustering → Agglomerative (Bottom Up)

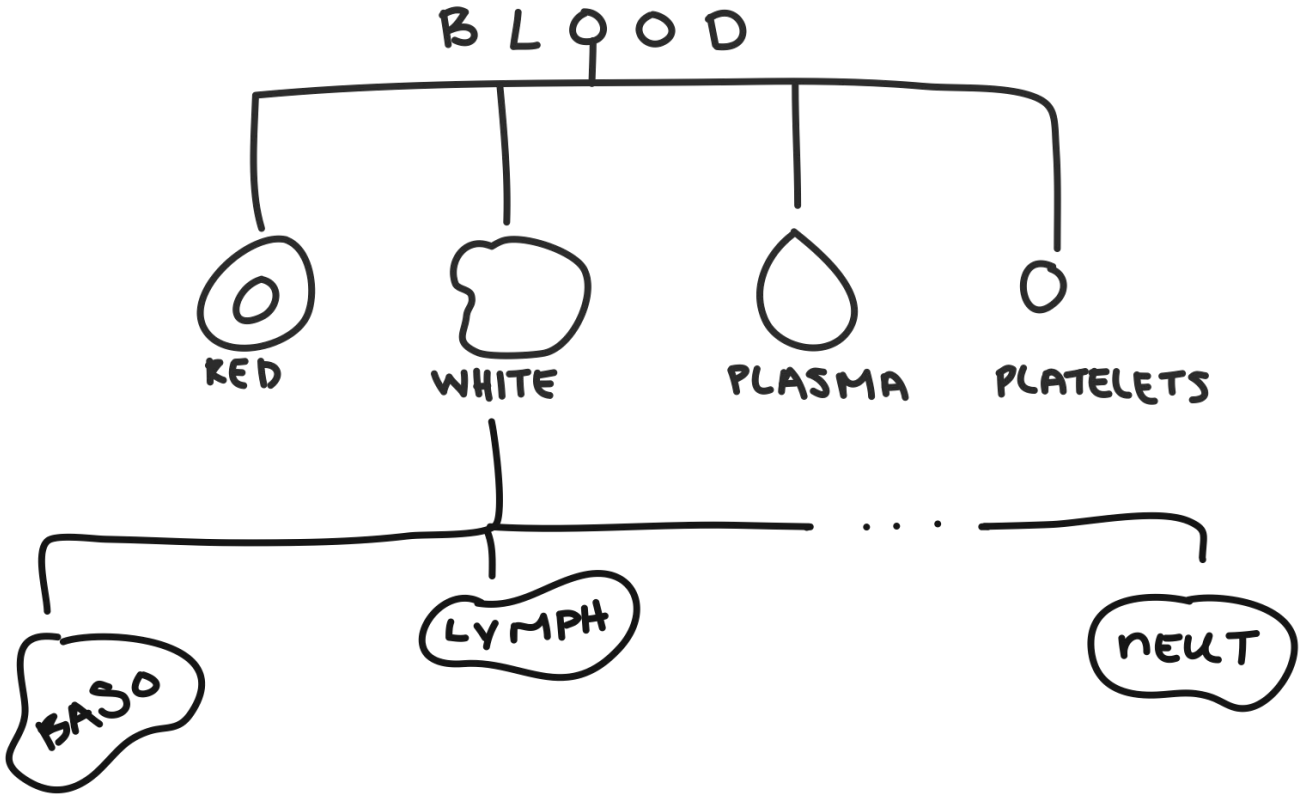
→ Divisive (Top Down)

HAC

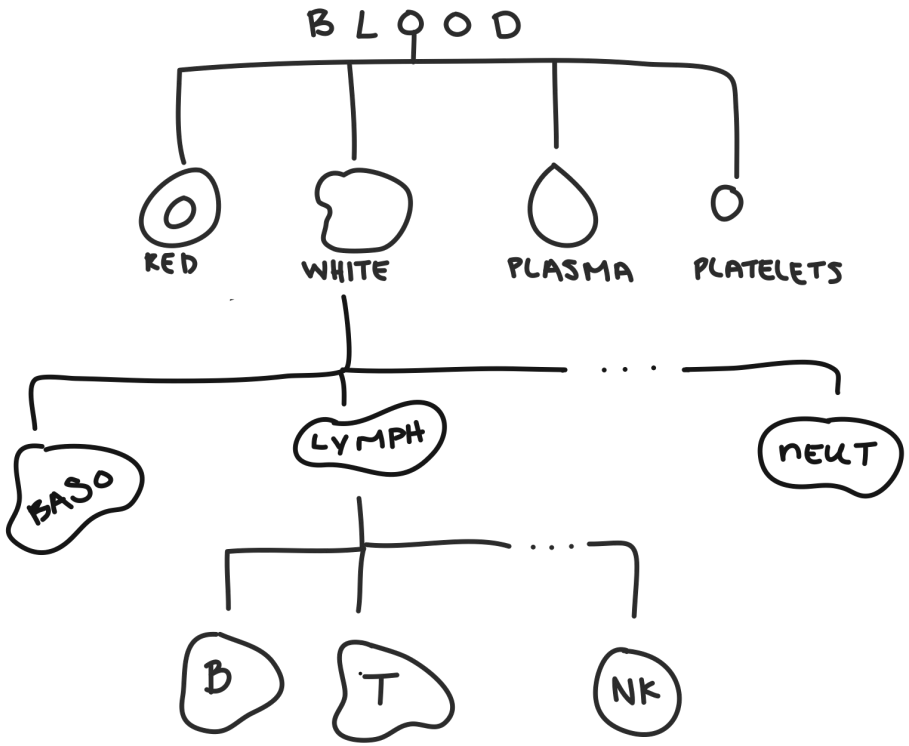
Divisive →



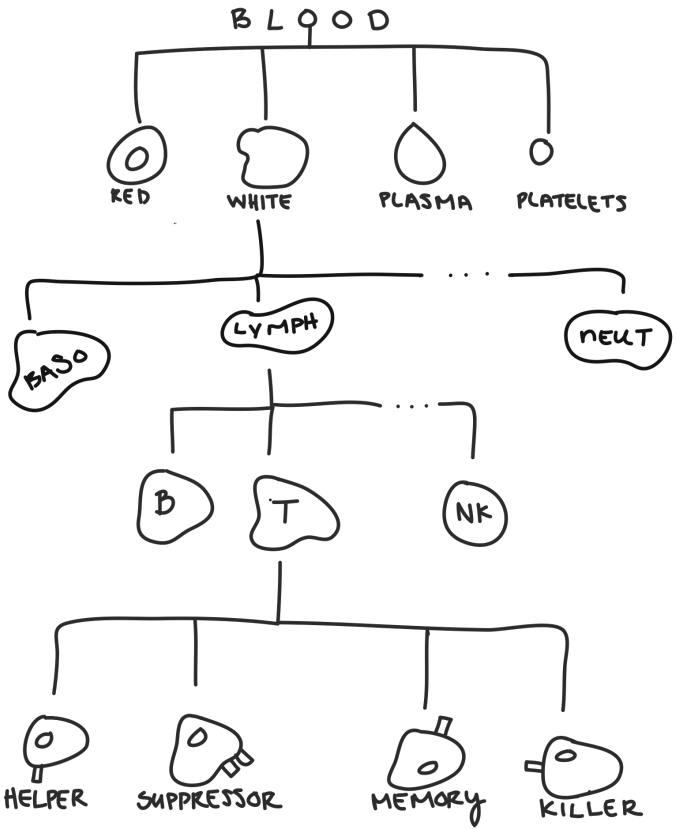
HAC



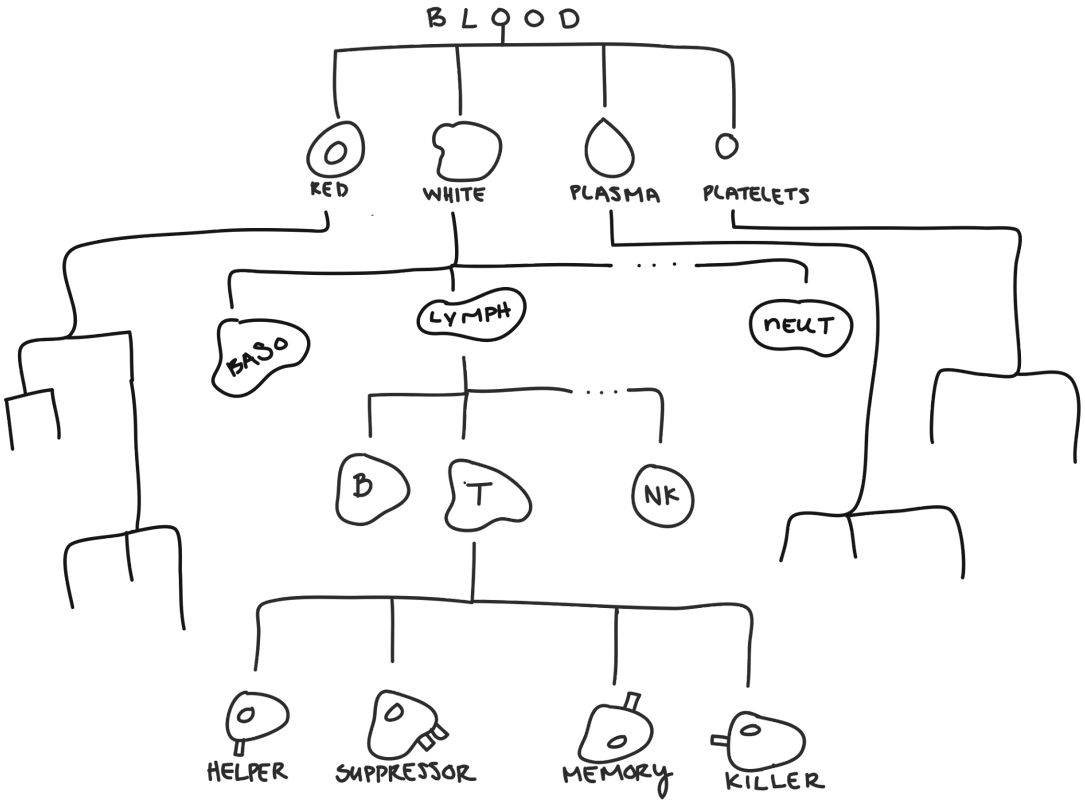
HAC



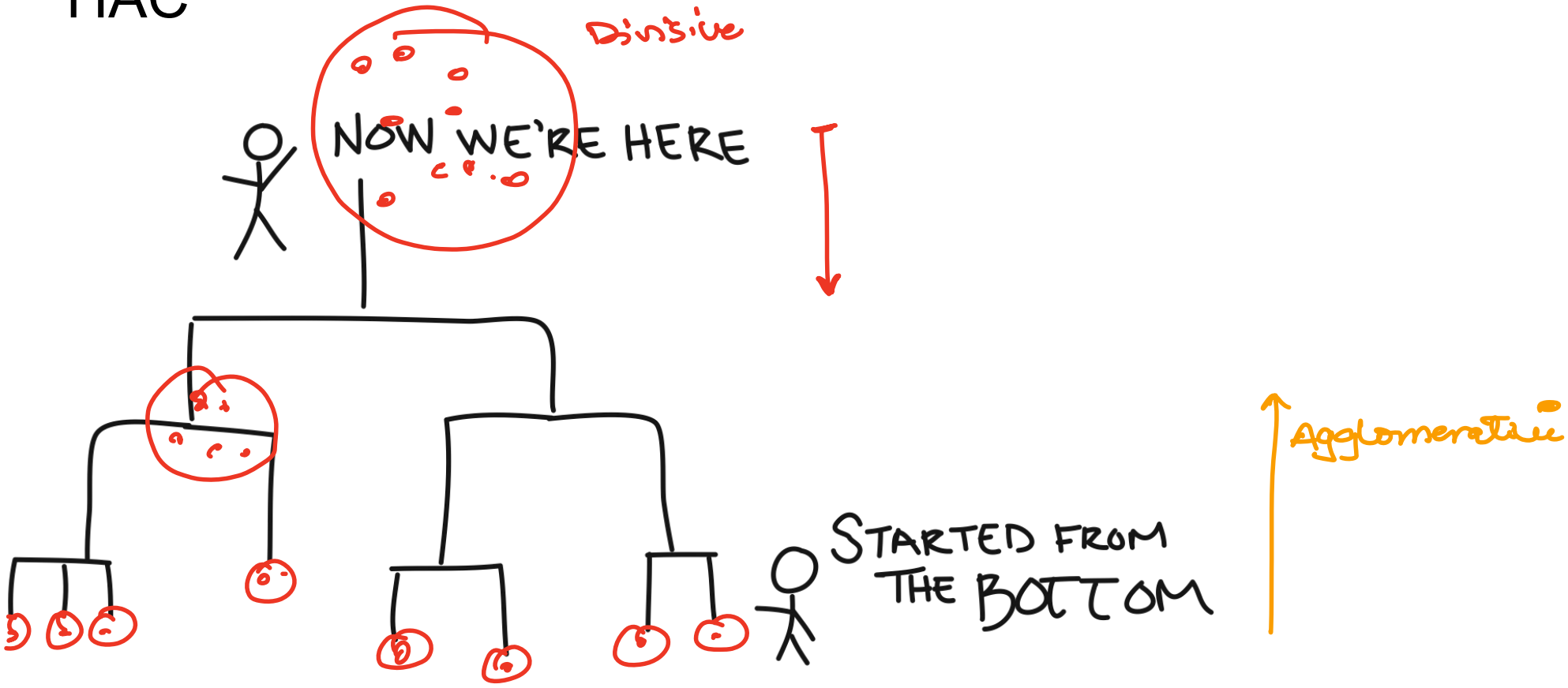
HAC


BLOOD
RED
WHITE
PLASMA
PLATELETS
BASO
LYMPH
NEUT
B
T
NK
HELPER
SUPPRESSOR
MEMORY
KILLER

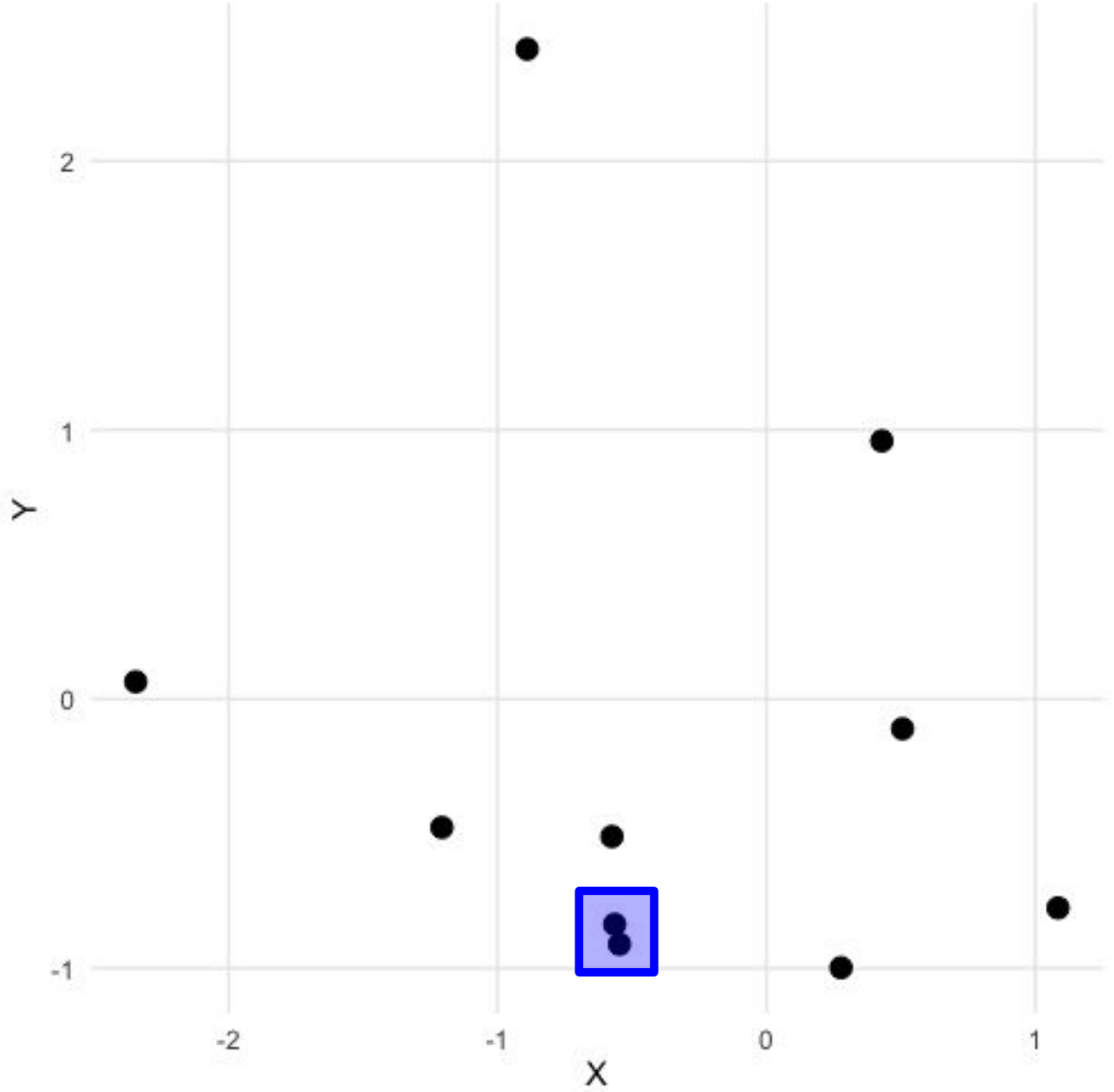
HAC


BLOOD
RED
WHITE
PLASMA
PLATELETS
BASO
LYMPH
NEUT
B
T
NK
HELPER
SUPPRESSOR
MEMORY
KILLER

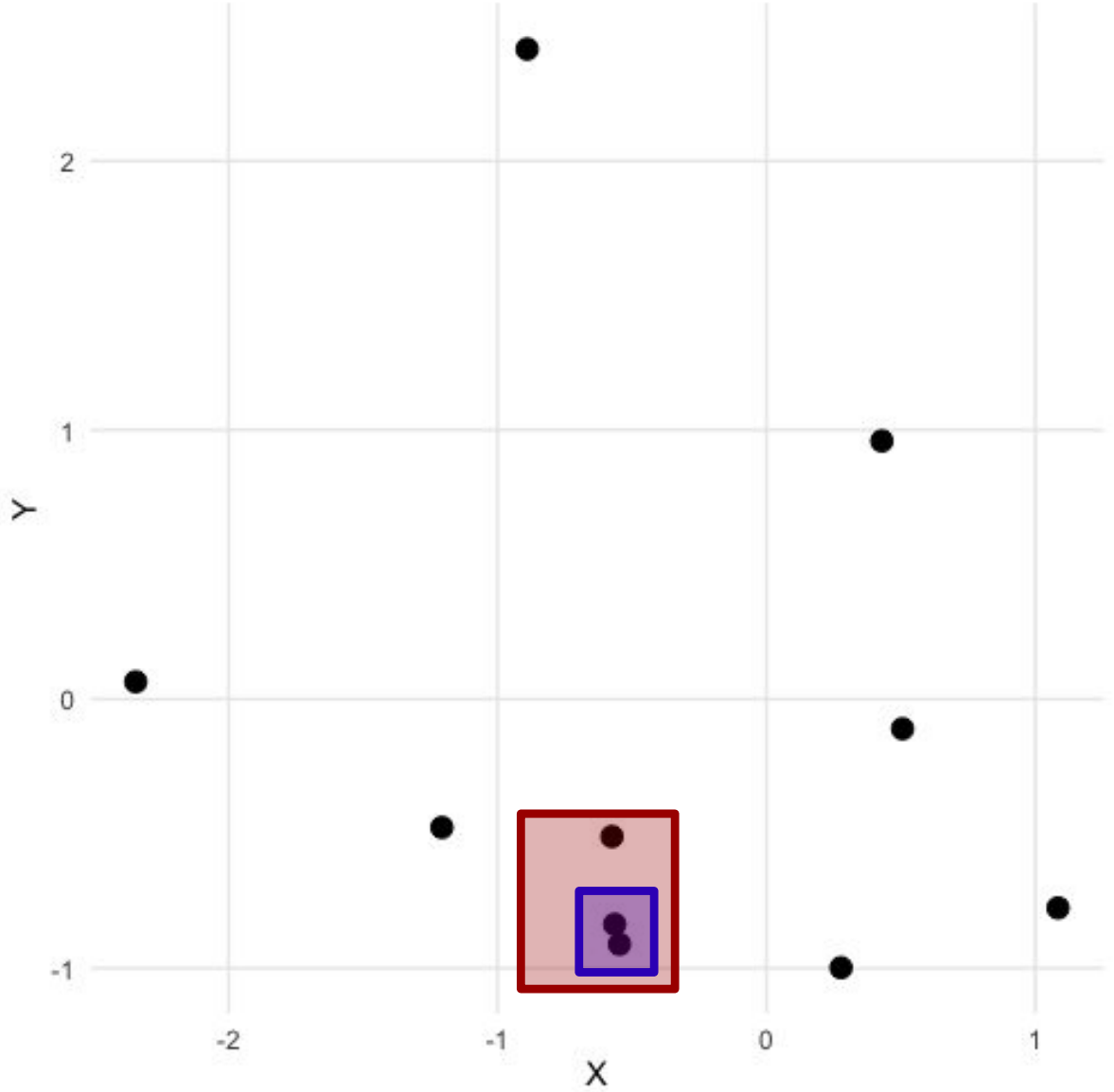
HAC


Divisive
NOW WE'RE HERE
STARTED FROM THE BOTTOM
Agglomerative

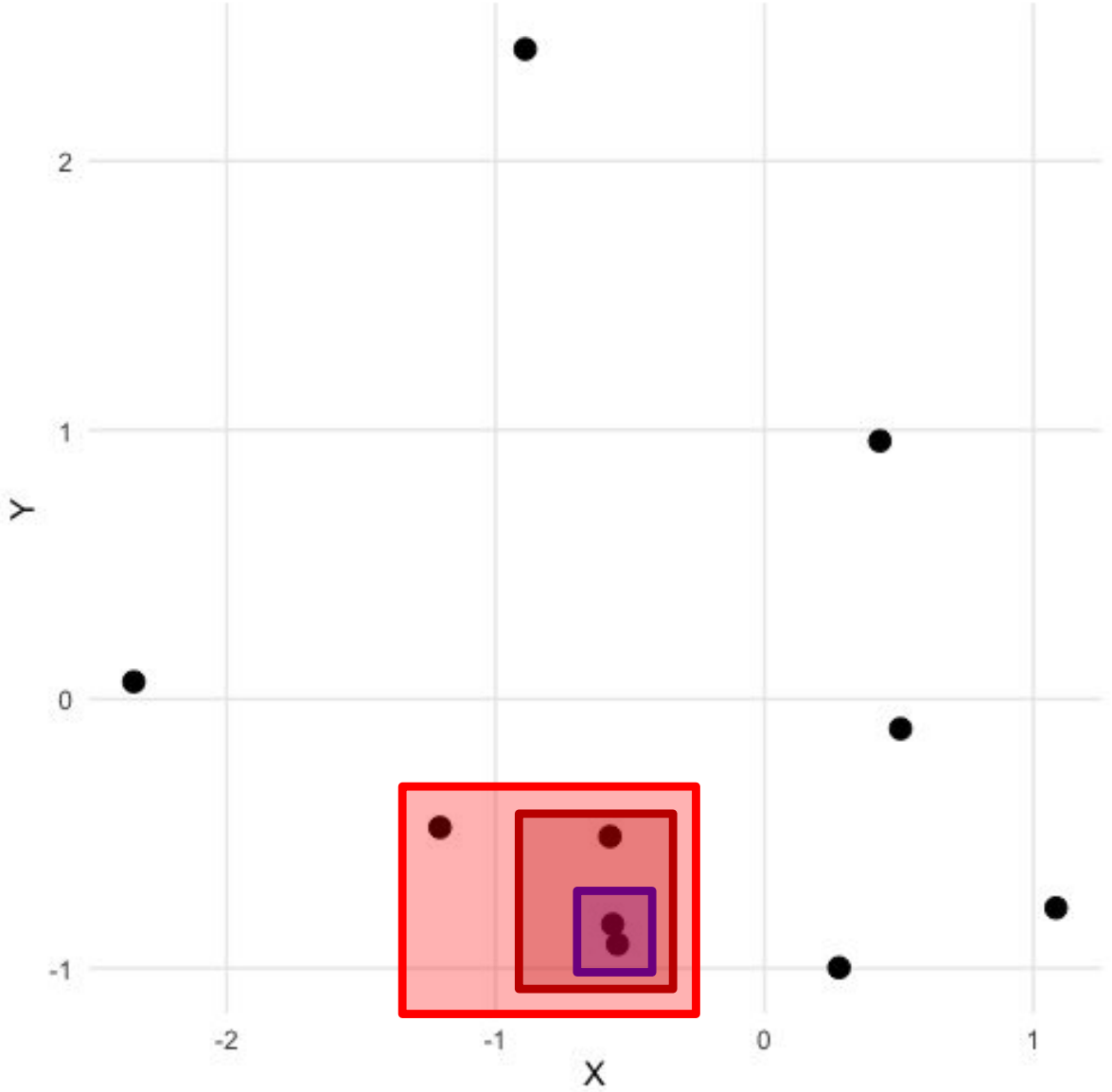
Algorithm



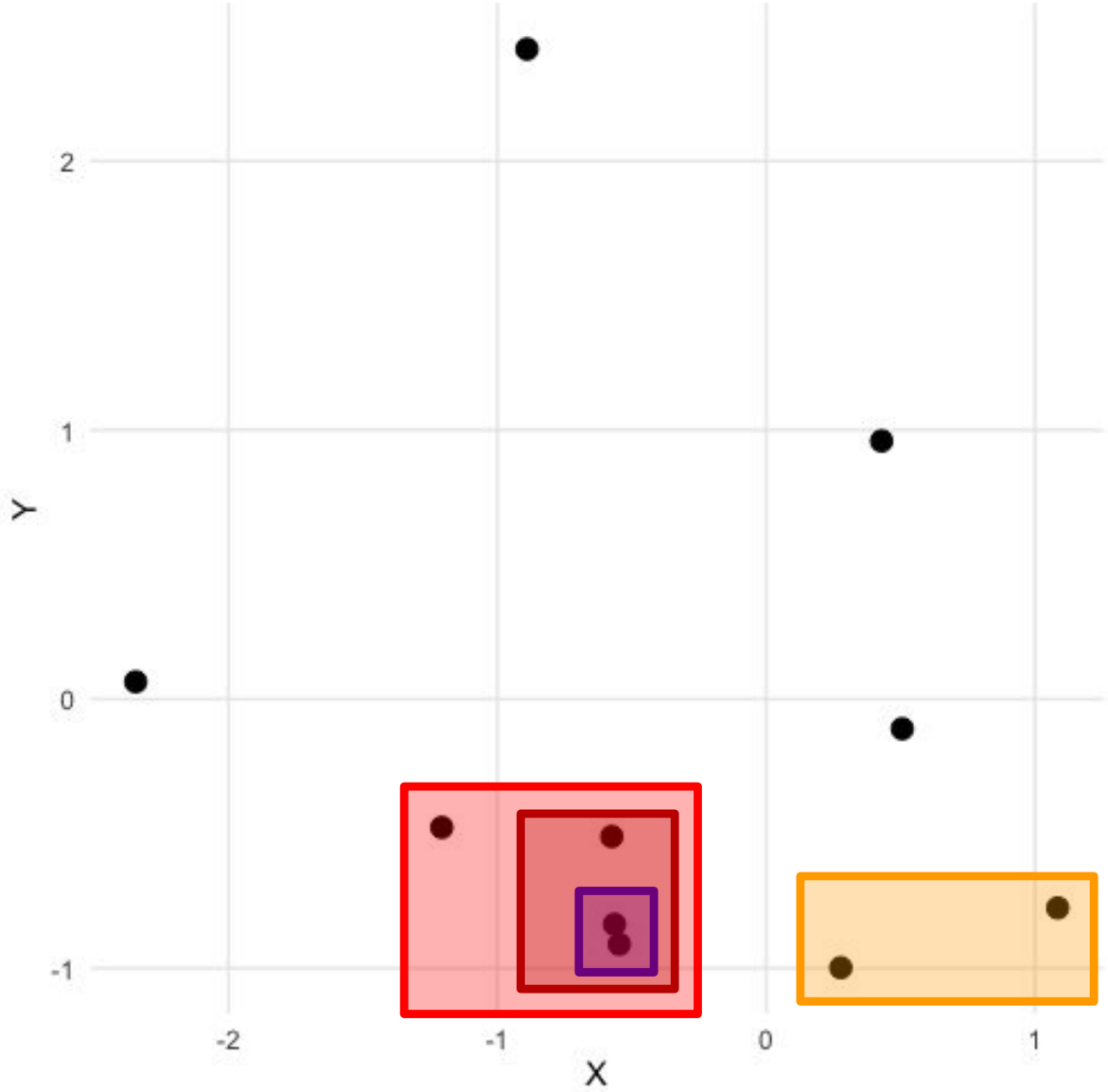
Algorithm



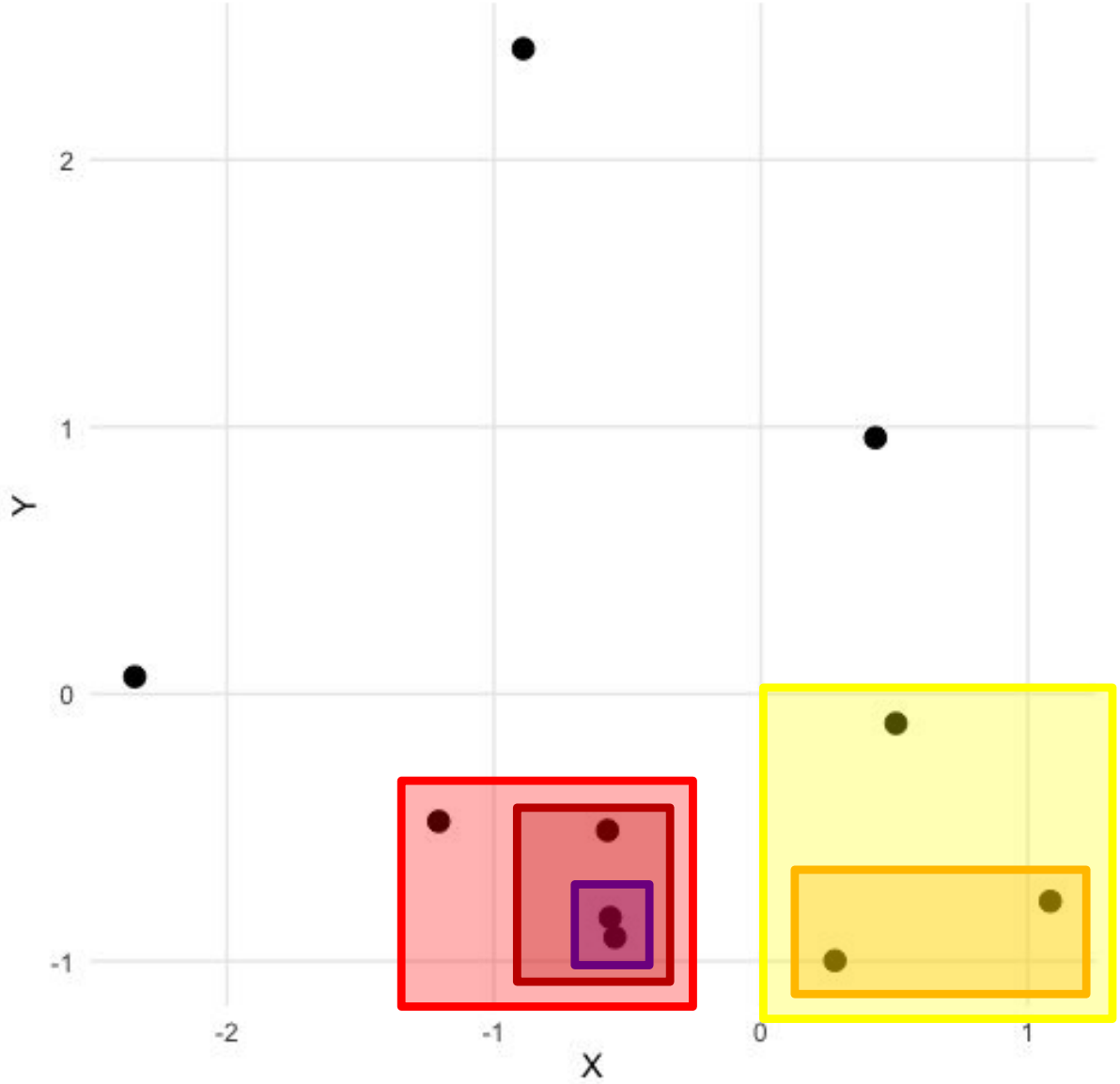
Algorithm



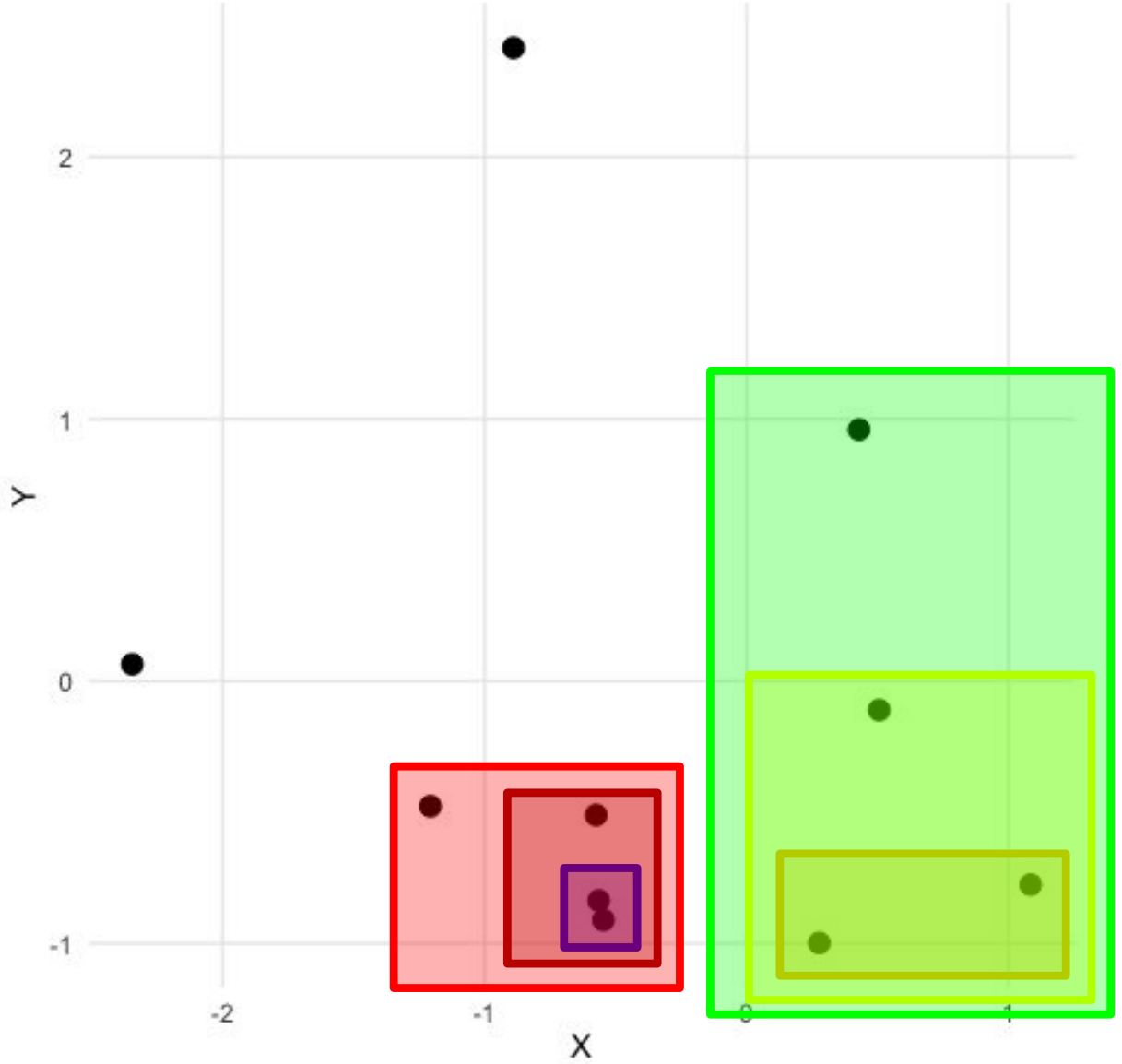
Algorithm



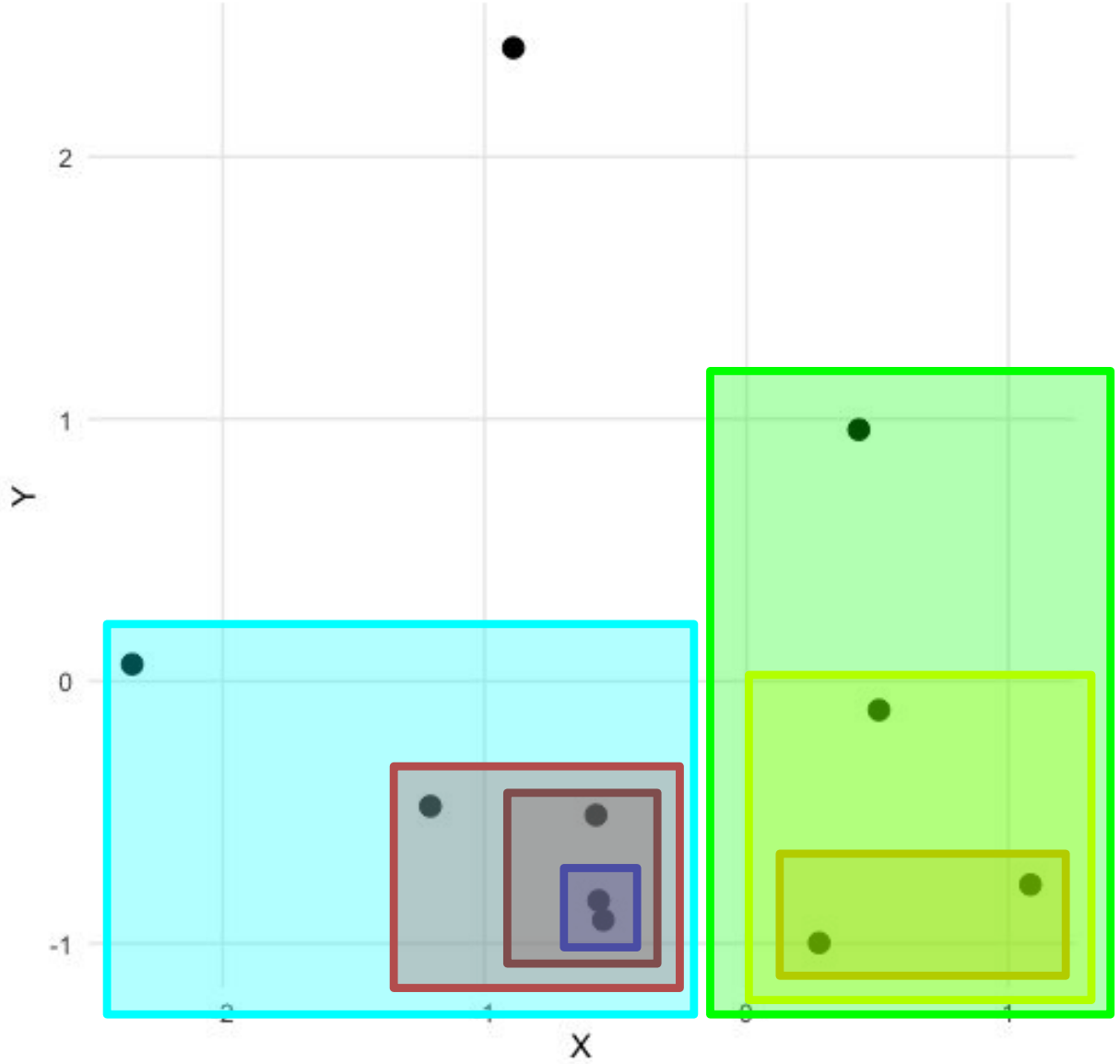
Algorithm



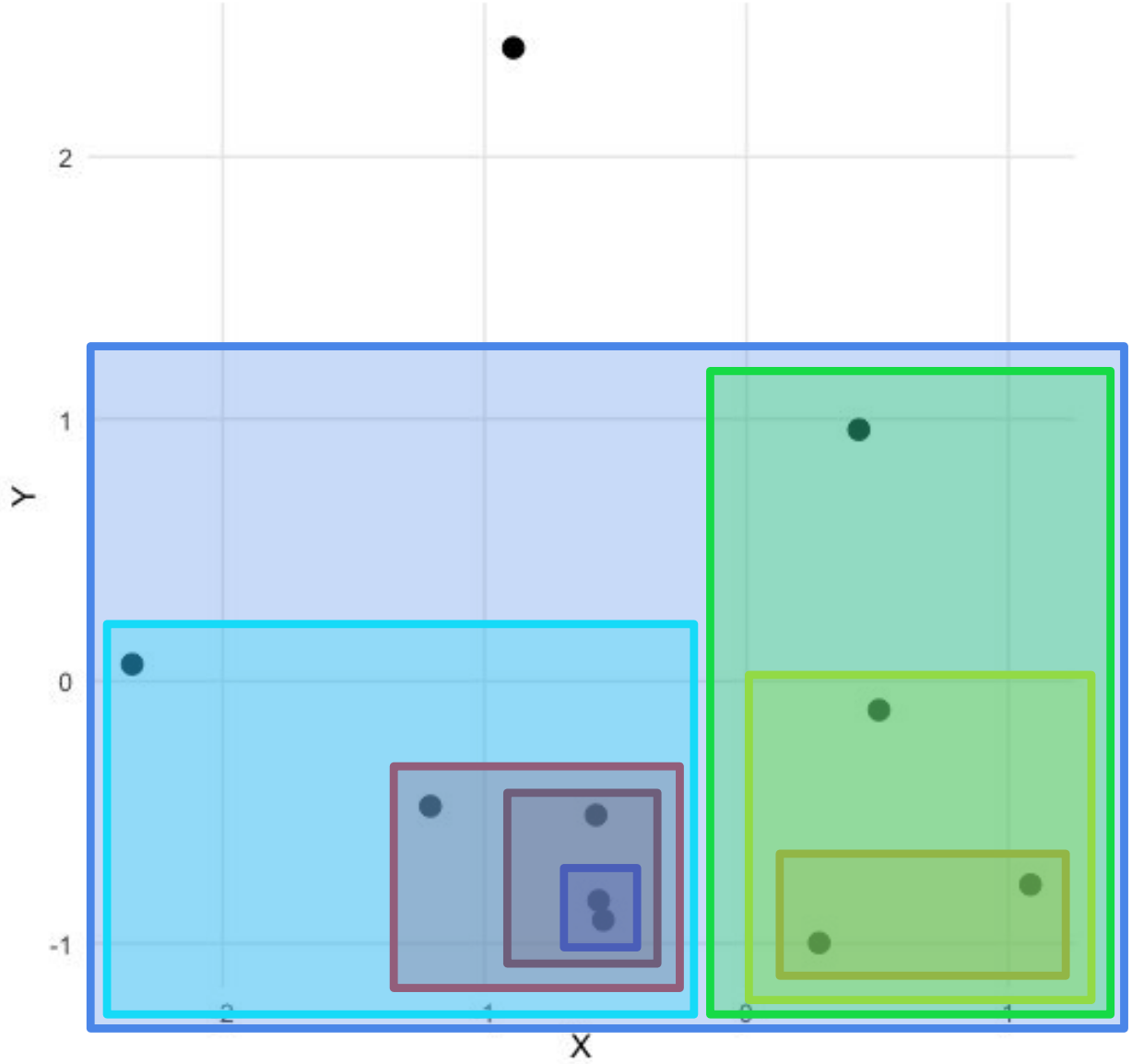
Algorithm



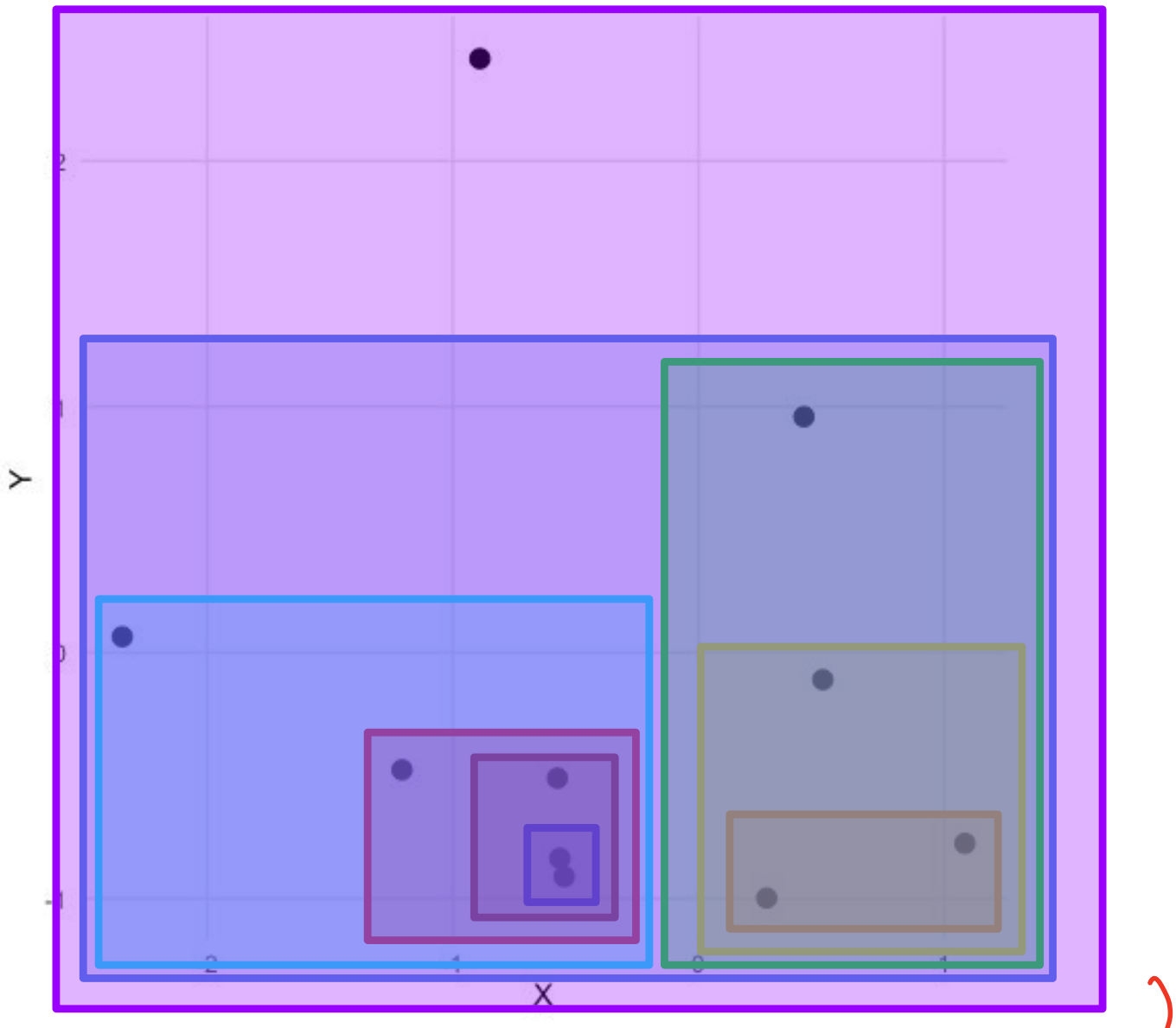
Algorithm



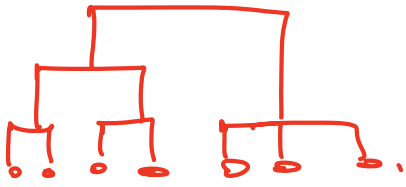
Algorithm

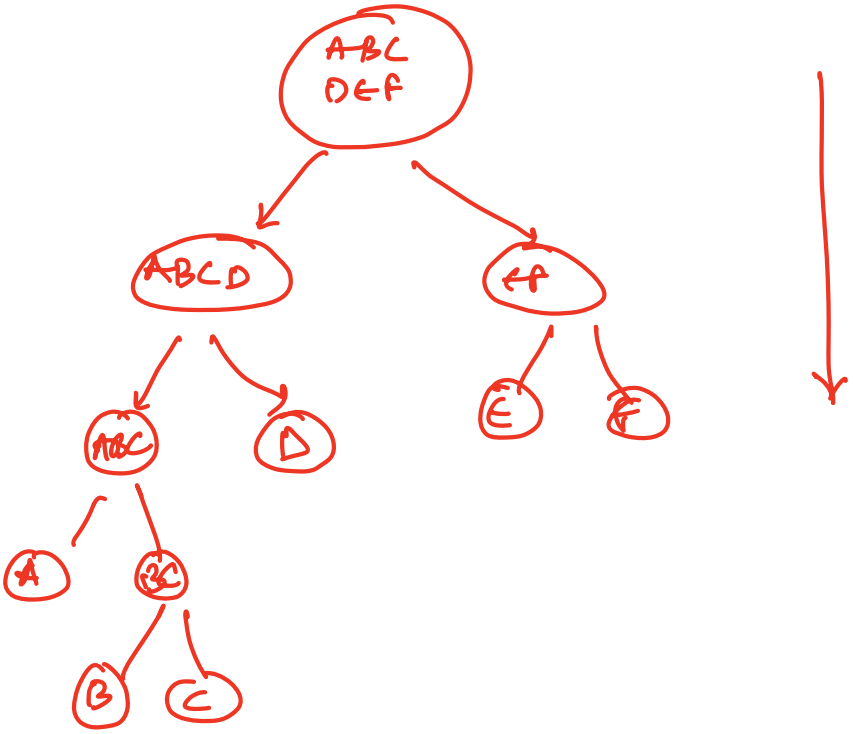


Algorithm



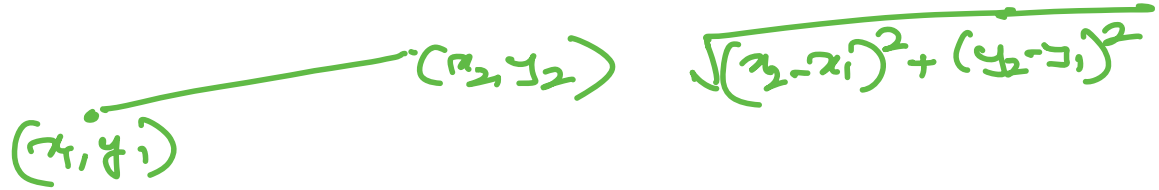
Agglomerative → BU

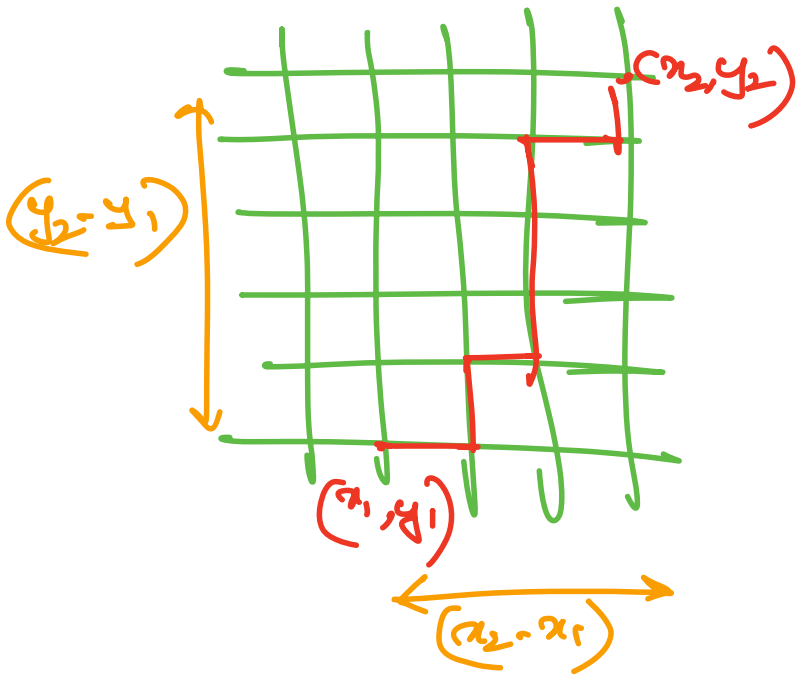


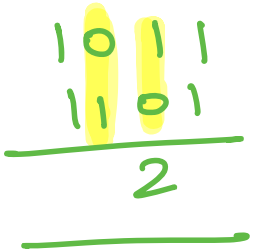


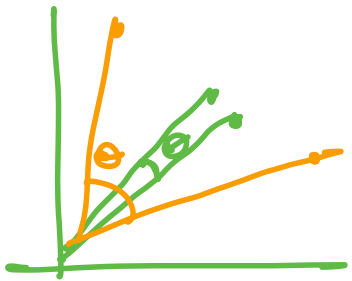
Distance Metrics

- **Euclidean**: Continuous Data
- **Manhattan**: High Dimensions
- **Hamming**: Categories
- **Cosine**: Word Counts



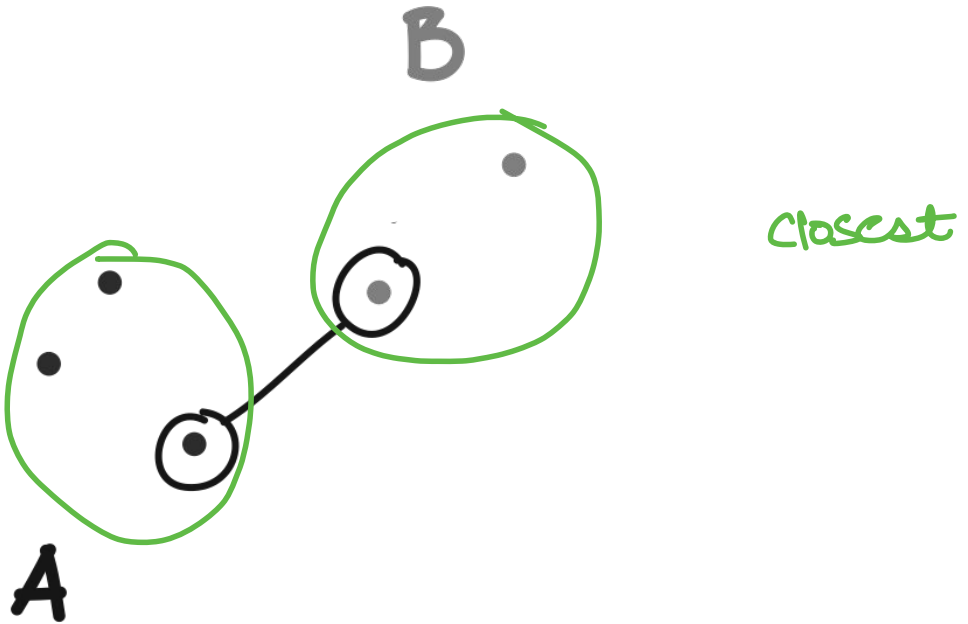






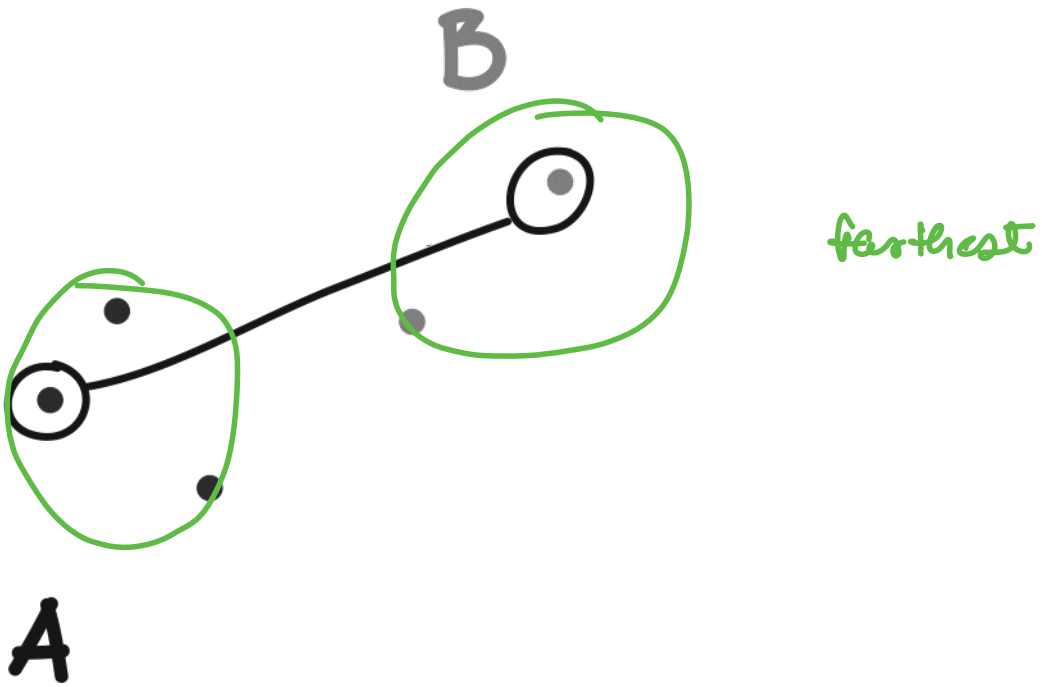
Linkage Criteria

(Distance b/w 2 clusters)



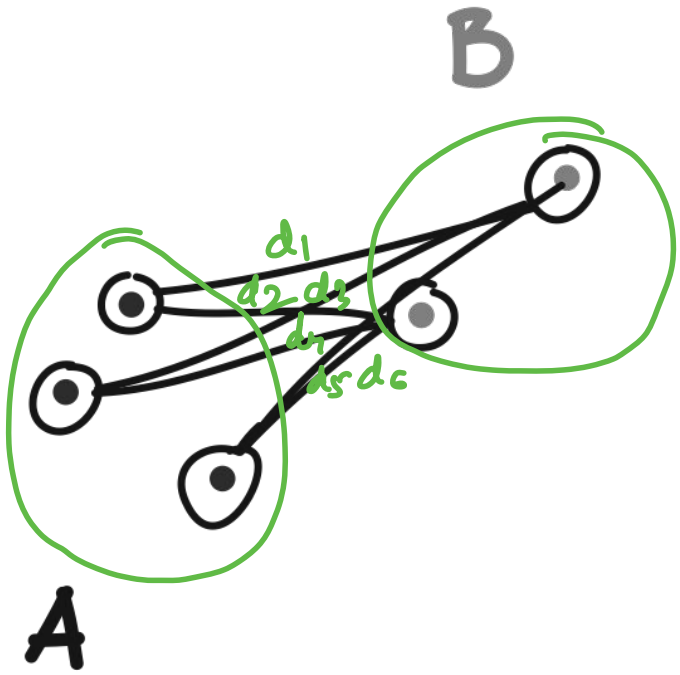
Single

Linkage Criteria



Complete

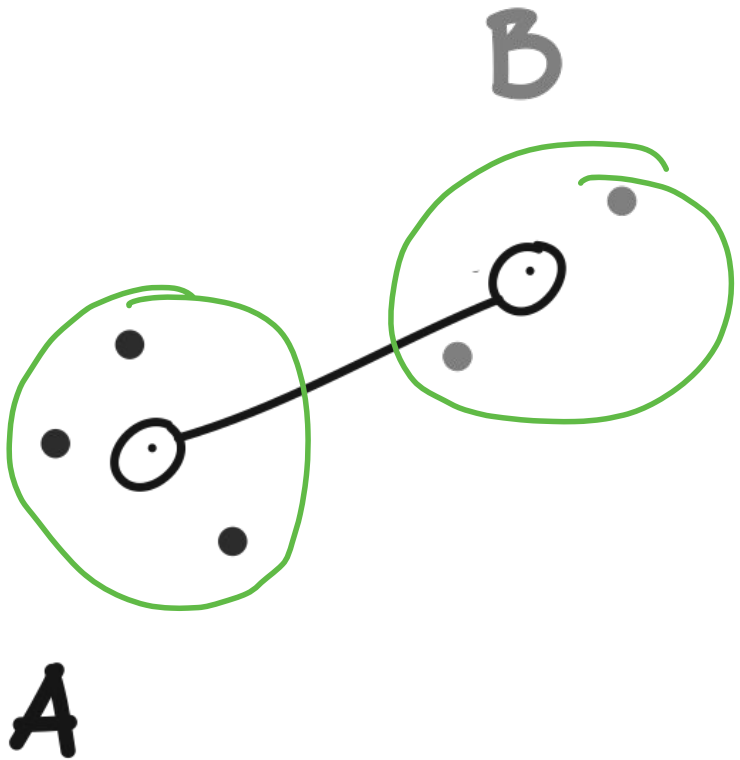
Linkage Criteria



$$\frac{d_1+d_2+d_3+d_4+d_5+d_6}{6}$$

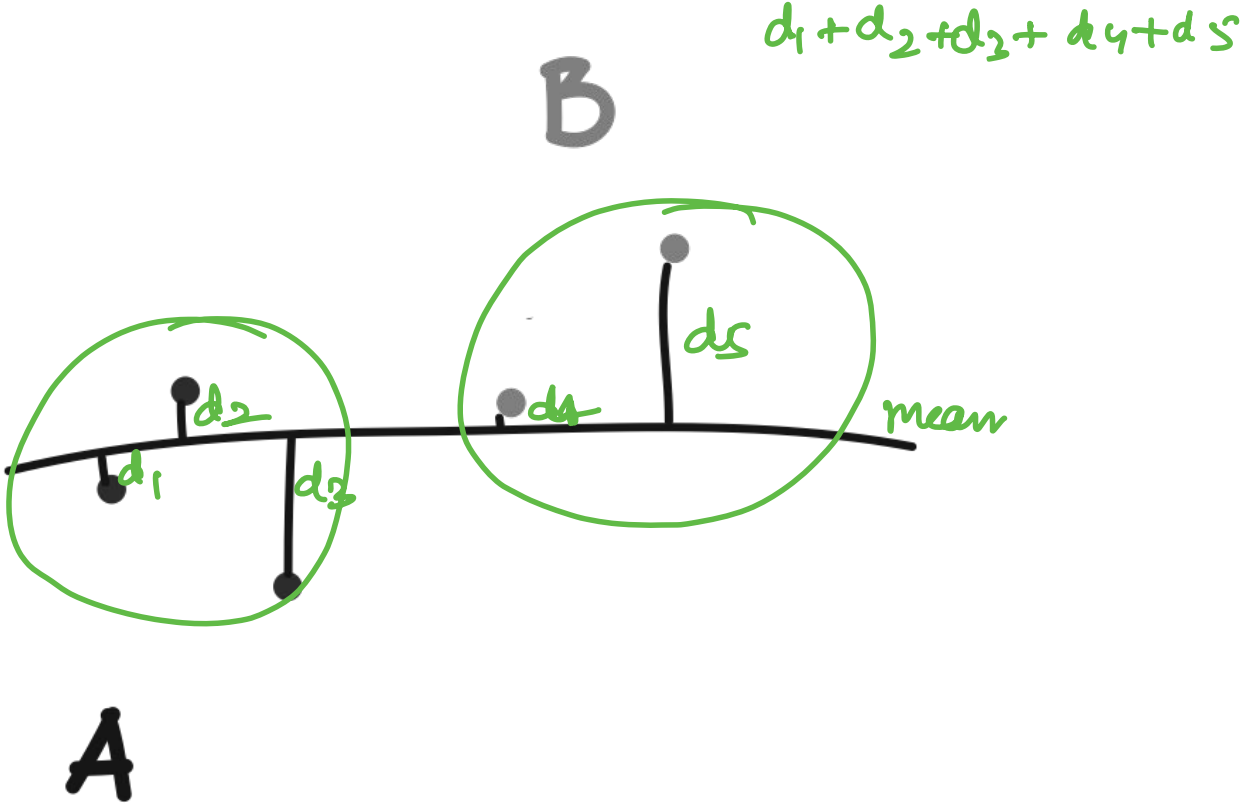
Average

Linkage Criteria



Centroid

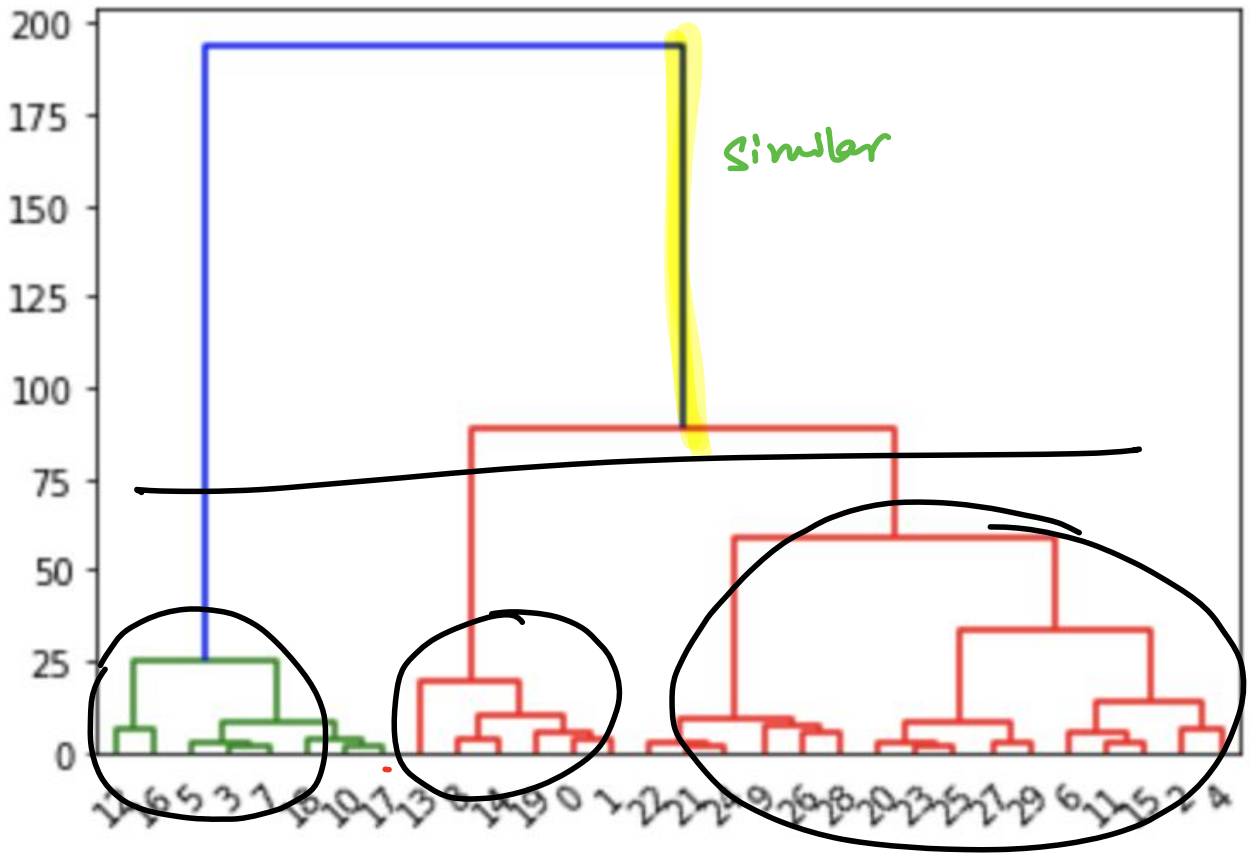
Linkage Criteria



(Variance Based)

Ward's

Reading a Dendrogram



DBSCAN:

Feature Selection:

10 features → all features are not relevant.

Feature Selection ?

Curse of Dimensionality

↳ lot of features, model performance ↓

Predict Salary

Humidity Temp. College CGPA Branch Exp.

Humidity, Temp. → irrelevant ✗

features

Irrelevant Redundant

ht (cm)	wt (kg)	wt (m)	age	dob
	✗	✗		✗

Curse of Dimensionality

↓

Dimensionality Reduction

↓

Feature Selection

abc ——— z } 26

top 5 features: b, f, i, 2, e

Feature Extraction

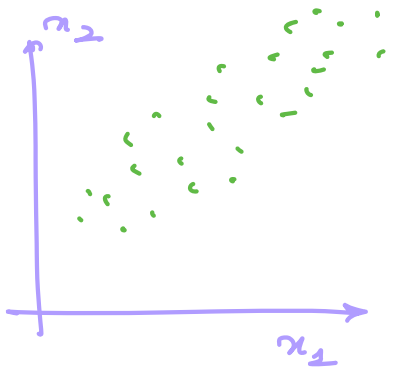
abc ——— z } 26 features

5 features:

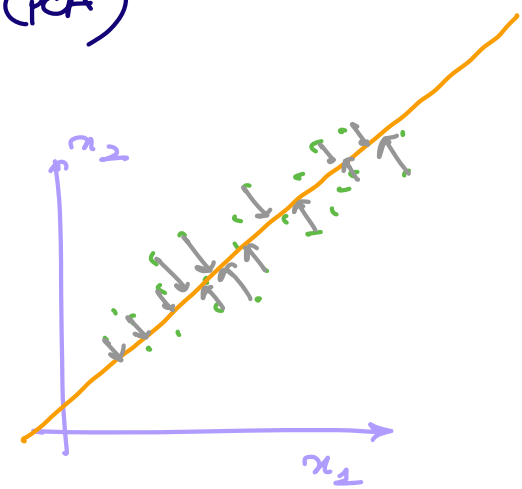
a+d b**e g+z*4

Algorithm: PCA

Principle Component Analysis (PCA)



(2D)



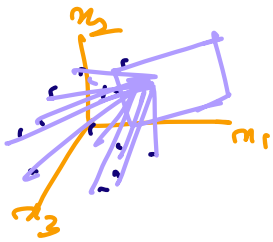
Goal: find Orange line

(2D)

(1D)

Applications:

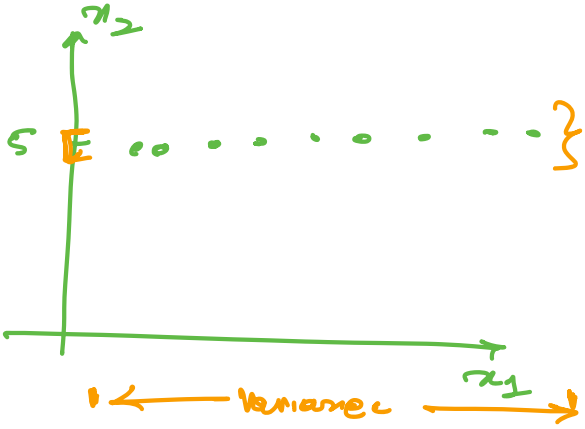
- Data Compression



- Data visualization
- Speed up Computation

Variance = Spread

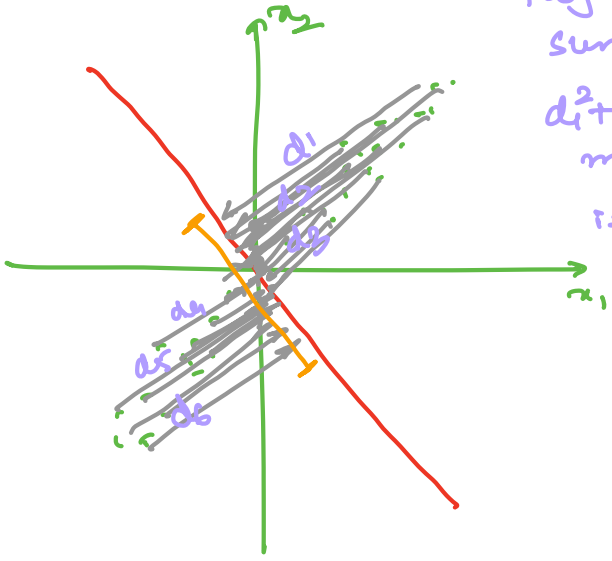
Eg:



ignore x_2
choose x_1

$$\text{Variance} = \frac{\Sigma(x-\bar{x})}{N}$$

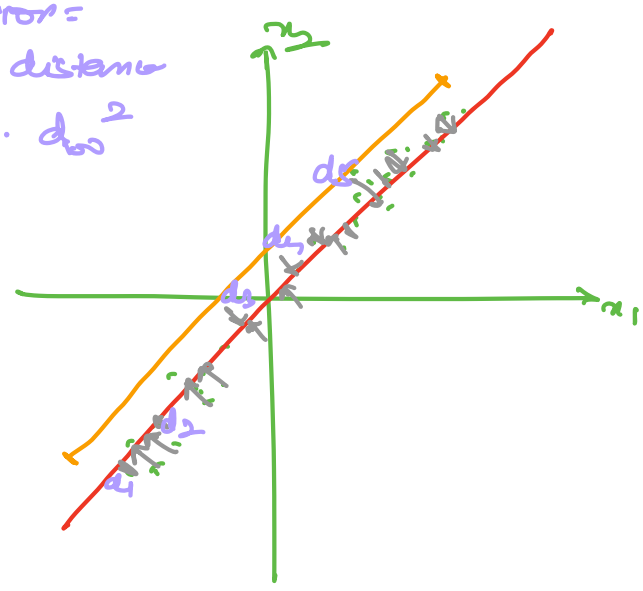
Eg:



Projection Error =
sum of ⊥ distance

$$d_1^2 + d_2^2 + \cdots d_{100}^2$$

$$\sum_{i=1}^{m} d_i^2$$

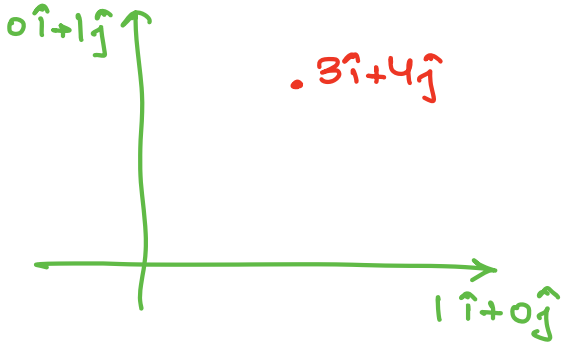


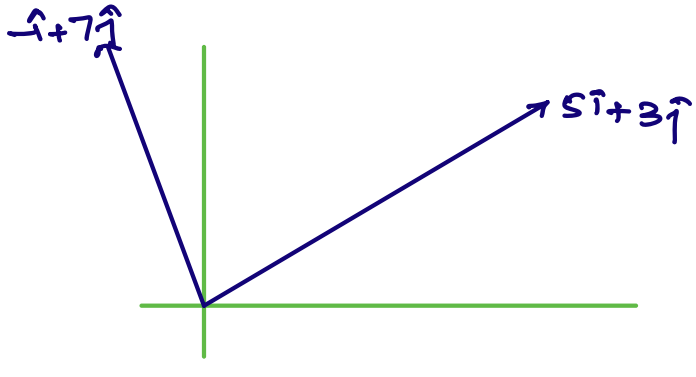
Spread less
Variance low
(Projection Error high)

Spread high
Variance high
PE low

Vectors:

— Original axis
— transformed axis
— Data point





Transformed = 3 (Transformed x) + 4 (Transformed y)

$\begin{bmatrix} 5 \\ 3 \end{bmatrix}$ $\begin{bmatrix} -1 \\ 7 \end{bmatrix}$

$= 3\begin{bmatrix} 5 \\ 3 \end{bmatrix} + 4\begin{bmatrix} -1 \\ 7 \end{bmatrix} = \begin{bmatrix} 15 \\ 9 \end{bmatrix} + \begin{bmatrix} -4 \\ 28 \end{bmatrix} = \begin{bmatrix} 11 \\ 37 \end{bmatrix}$

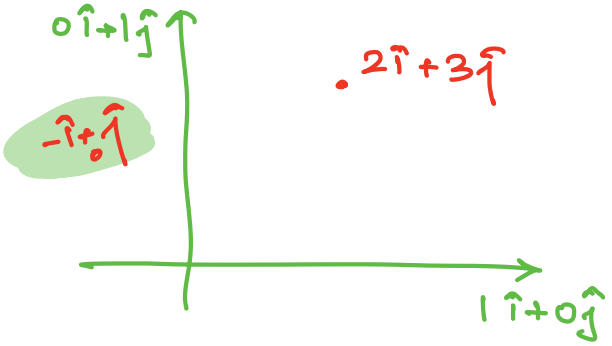
$11\hat{i} + 37\hat{j}$

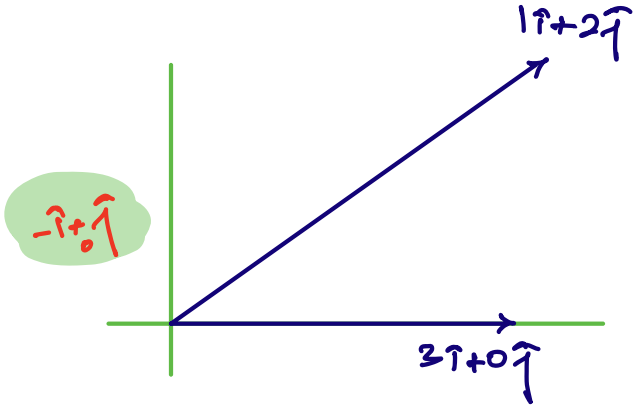
Transformation Matrix

$\begin{bmatrix} 5 & -1 & 3 \\ 3 & 7 & 4 \end{bmatrix} = \begin{bmatrix} 15-4 \\ 9+28 \end{bmatrix} = \begin{bmatrix} 11 \\ 37 \end{bmatrix}$

new x, new y, coordinates of vector

Eigen vector & Eigen values





$2\hat{i}+3\hat{j}$ $\begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 6+3 \\ 0+6 \end{bmatrix} = \begin{bmatrix} 1 \\ 6 \end{bmatrix} = 3\begin{bmatrix} 3 \\ 2 \end{bmatrix}$

$-\hat{i}+\hat{j}$ $\begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{bmatrix} -3+1 \\ 2 \end{bmatrix} = \begin{bmatrix} -2 \\ 2 \end{bmatrix} = 2\begin{bmatrix} -1 \\ 1 \end{bmatrix}$

$A \cdot \overline{v} = \lambda \cdot \overline{v}$

Transformation matrix — Eigen vector — Eigen value (scalar)

$$A\bar{v} = \lambda \cdot I \cdot \bar{v}$$

$$(A - \lambda I)\bar{v} = 0$$

$$|A - \lambda I| = 0$$

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\lambda I = \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix}$$

$$\begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} = 0$$

$$\begin{bmatrix} 3-\lambda & 1 \\ 0 & 2-\lambda \end{bmatrix} = 0$$

$$(3-\lambda)(2-\lambda) = 0$$

$\lambda = 2, 3$ → Eigen Values

$$\begin{bmatrix} cov(x_1,x_1) & cov(x_1,x_2) \\ cov(x_2,x_1) & cov(x_2,x_2) \end{bmatrix}$$

Covariance Matrix = $\begin{pmatrix} 9 & 4 \\ 4 & 3 \end{pmatrix}$

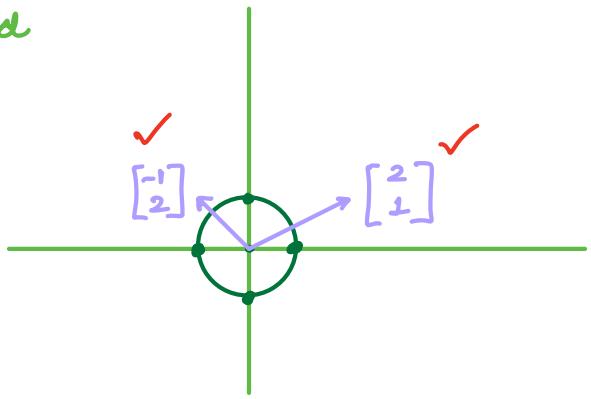
$cov(x_1, x_1)$, $cov(x_1, x_2)$, $cov(x_2, x_1)$, $cov(x_2, x_2)$

$$cov(X,Y) = \sum_{i=1}^{n} \frac{(x-\bar{x})(y-\bar{y})}{n-1}$$

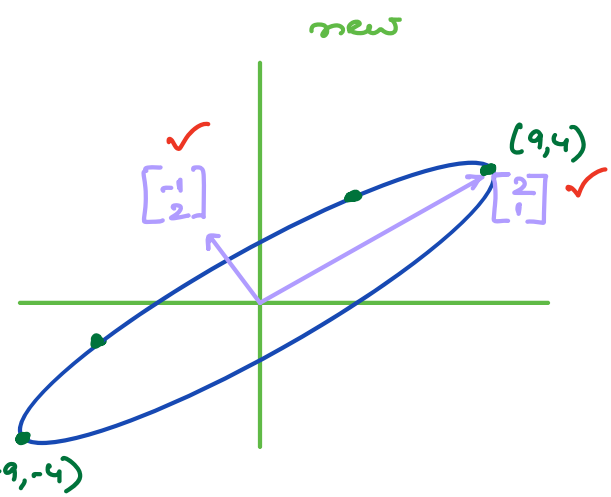
$$cov(X,X) = variance(X) = \sum_{i=1}^{n} \frac{(x-\bar{x})^2}{n-1}$$

- Symmetric
- Used as transformation matrix

old



new



Transformation matrix

$$\begin{pmatrix} 9 & 4 \\ 4 & 3 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = x\begin{bmatrix} 9 \\ 4 \end{bmatrix} + y\begin{bmatrix} 4 \\ 3 \end{bmatrix} = \begin{bmatrix} 9x+4y \\ 4x+3y \end{bmatrix} \checkmark$$

$$(x,y) \longrightarrow (9x+4y,\ 4x+3y)$$

$$(0,0) \longrightarrow (0,0)$$

Circle:
- $(1,0) \longrightarrow (9,4)$
- $(0,1) \longrightarrow (4,3)$
- $(-1,0) \longrightarrow (-9,-4)$
- $(0,-1) \longrightarrow (-4,-3)$

} plot → ellipse

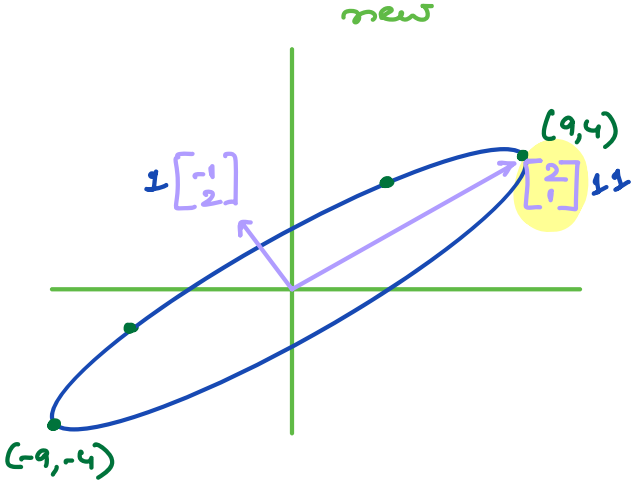
$(2,1) \longrightarrow (22,11) = 11(2,1)$

$(-1,2) \longrightarrow 1(-1,2)$

} Eigen vector

eigen value

In PCA, goal was to find new axis.



$\begin{bmatrix} -1 \\ 2 \end{bmatrix}$ and $\begin{bmatrix} 2 \\ 1 \end{bmatrix}$ are going to be the new axis.

These are the lines having highest spread.

Highest variance will be given by Eigen vector.

EV1 → 11
EV2 → 1

} Eigen value → magnitude

Spread

We will project all points on Eigen vector 1 bcz its magnitude is larger.

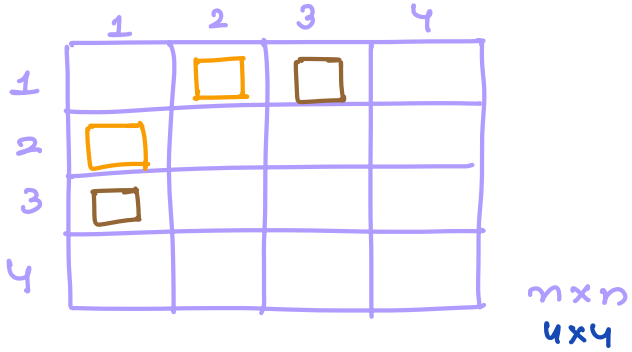
Steps:

1. Load Data
2. Standardization

$\mu = 0$ (mean = 0)
$\sigma = 1$ (std = 1)

3. Covariance Matrix : how are 2 features related to each other.

n features
4 4



n x n
4 x 4

4. Eigen Values & Eigen Vector

(4) { evec1 → evalue 1
evec2 → evalue 2
evec3 → evalue 3
evec4 → evalue 4

5. Larger eigen value means spread/variance is more along that vector.

evec1 → 50
evec2 → 30
evec3 → 60
evec4 → 10
} eigen values

Choose the axis along which spread is larger.
Spread is given by eigen value.

Sort them acc. to eigen value

vec 3 → 60
vec 1 → 50
vec 2 → 30
vec 4 → 10

} Sorting } $k \to 2$ vectors.

6. Pick top k eigen vectors.
↓
new dimension.

7. Projection