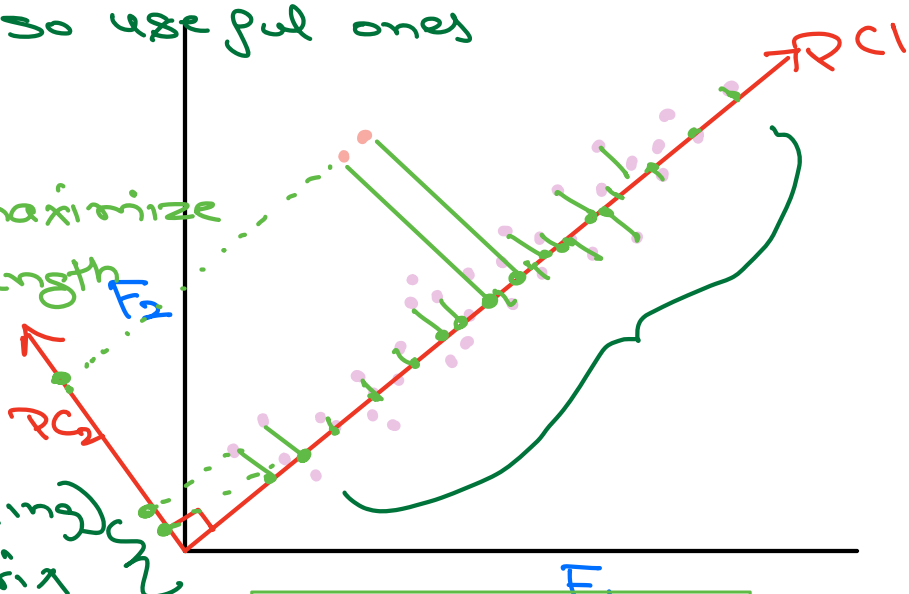
- Recap
- Issues with PCA
- t-SNE intro
- Preserving Relative Distances
- Inverse of Distances as PDF
- Normalized probability with ND
- Perplexity as an alternative
- Crowding problem
- t-distribution
- KL-Divergence for comparing Distributions

Recap of PCA

* **Goal** Find new Feature and remove some Not so useful ones

* **How?**

By trying to maximize Avg Projection Length



Steps for PCA

1. Mean centering (Standard Scaling)
2. Covariance Matrix $V \Rightarrow X^T X / n$
3. Find Eigen Values and Eigen Vector of V
4. Sort w.r.t Eigen Values and remove Eigen Vectors with Small E.Values

$$\max \frac{1}{n} \sum_{i=1}^{n} \frac{\vec{x}_i \cdot \vec{w}}{\|\vec{w}\|}$$

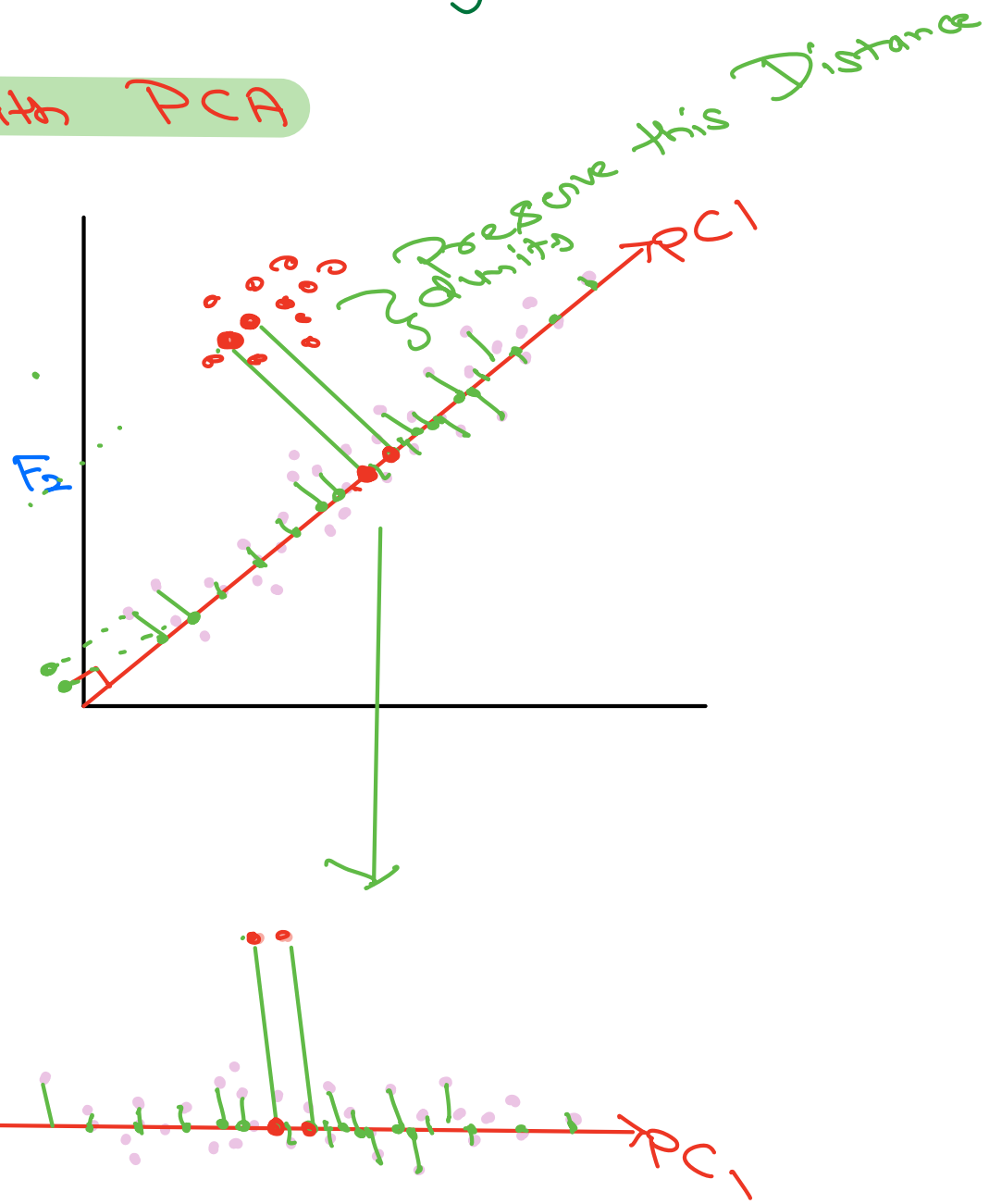
$$V\vec{v} = \lambda\vec{v}$$

⑤ Project your Data to Eigen Vectors

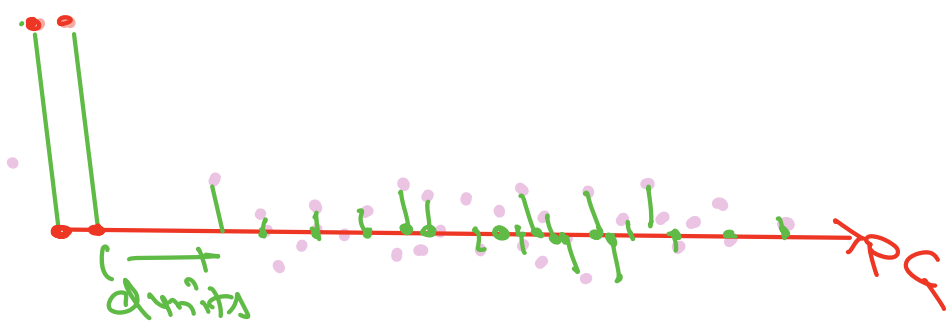
$X \cdot EV$

⑥ Plot or perform ML training

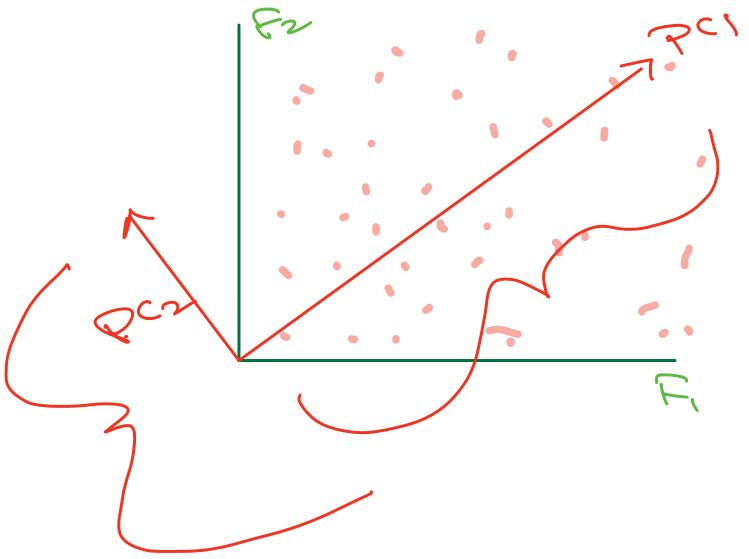
⑦ Issue with PCA

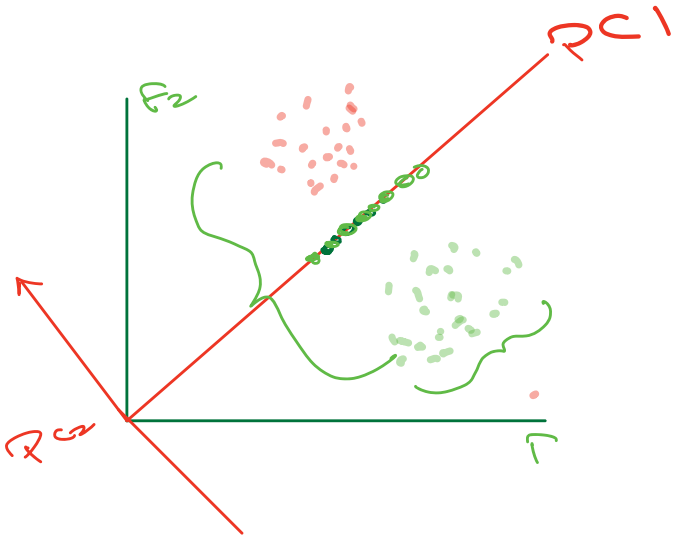


Ideal Scenario:

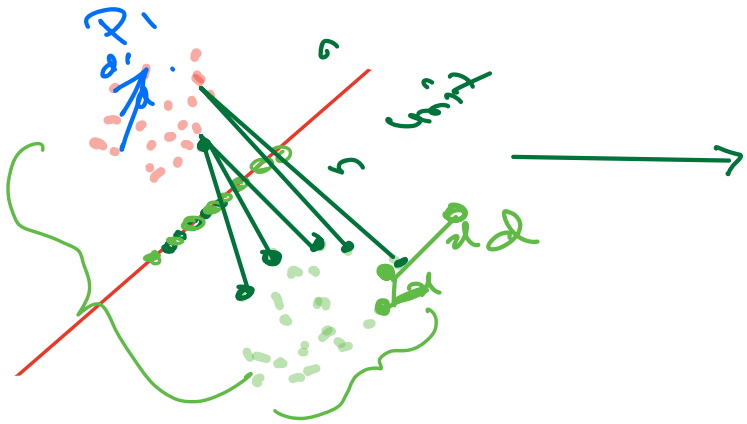


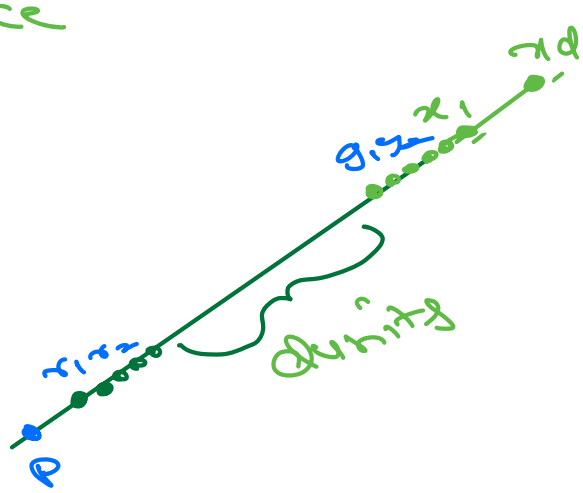
Can i drop any of the PC's



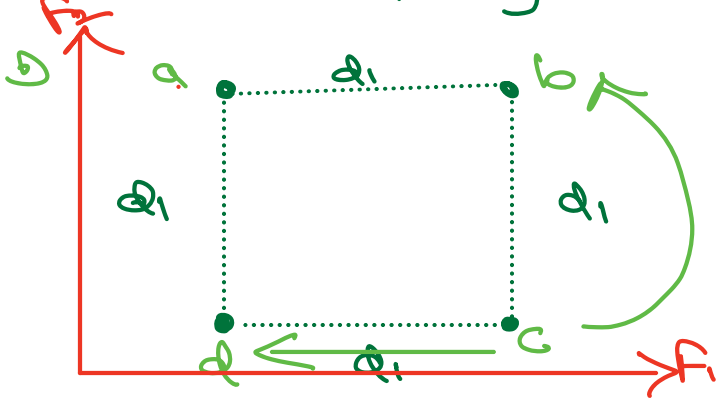


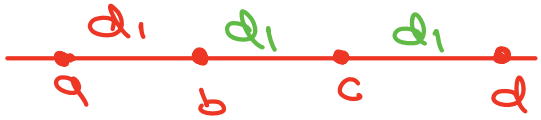
Idea-1 : Preserve the Relative Distance





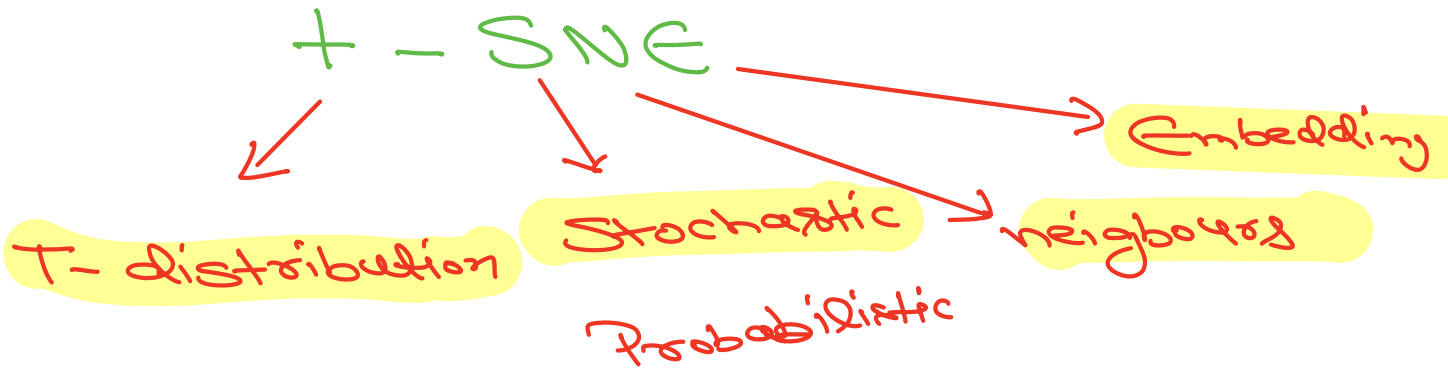
Drawback of this distance approach





* Impossible without any violation

Idea 2: Preserve probability of points as Neighbours



Embedding

F_1	F_2
10	30
25	35
.	.
.	.
.	.

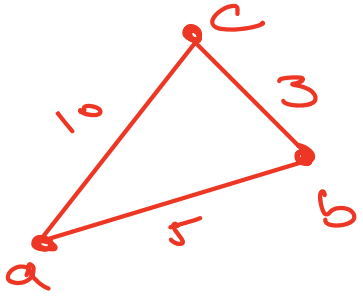
PC_1	PC_2
0.9	0.1
—	—
—	—
—	—

$PC_1 \propto \alpha F_1 + \beta F_2$

Projected feature space is also called Embedding

How can we convert distance into probability

$d(a,c) \gg d(a,b) \Rightarrow$ Prob score



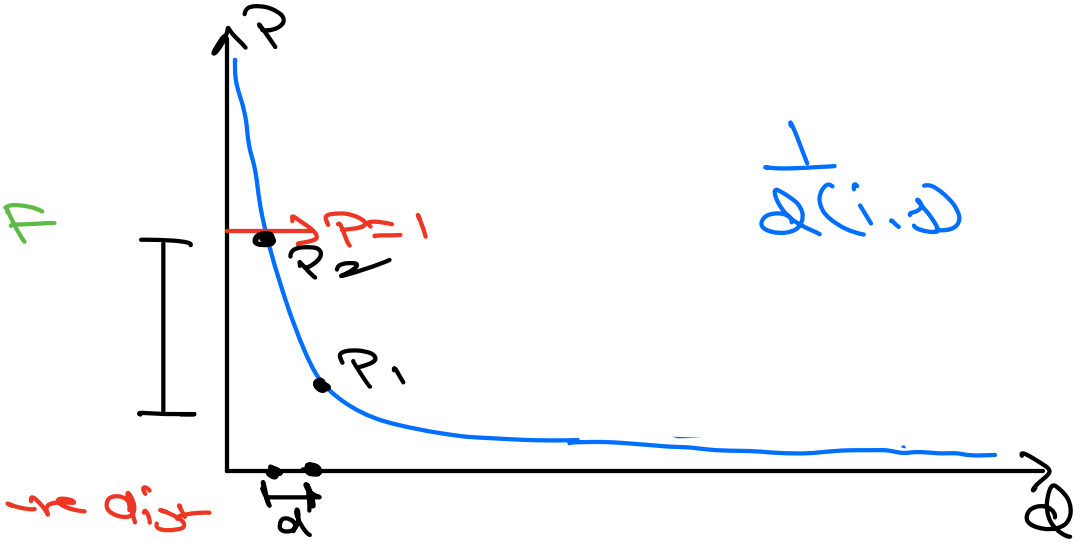
$P(A,B)$ being Neighbours can be quantified as $\frac{1}{d(A,B)} \propto \frac{1}{\|A,B\|}$

$P(a,b) \propto \frac{1}{d(a,b)} \propto \frac{1}{5} \propto 0.2$

$P(b,c) \propto \frac{1}{d(b,c)} \propto \frac{1}{3} \propto 33.33$

$P(c,a) \propto \frac{1}{d(c,a)} \propto \frac{1}{10} \propto 0.1$

PDF

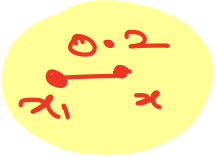


$\frac{1}{d(i,j)}$

Issues with PDF $\propto \frac{1}{d(i,j)}$

1) When $d \to 0, p \to \infty$

2) When $d \to 0.2, p \to 5$

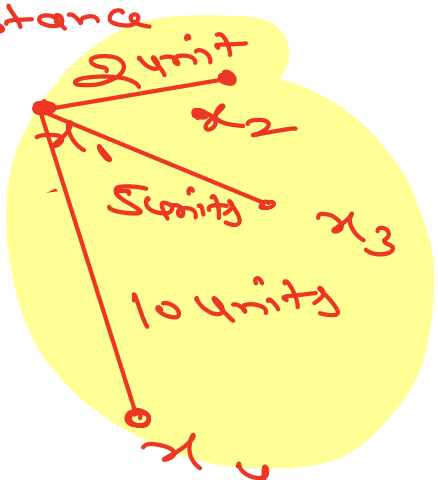


3) Very sensitive to Small Changes in Distance

$P(x_1, x_2) \propto \frac{1}{2} \propto 0.5$

$P(x_1, x_3) \propto \frac{1}{5} \propto 0.2$

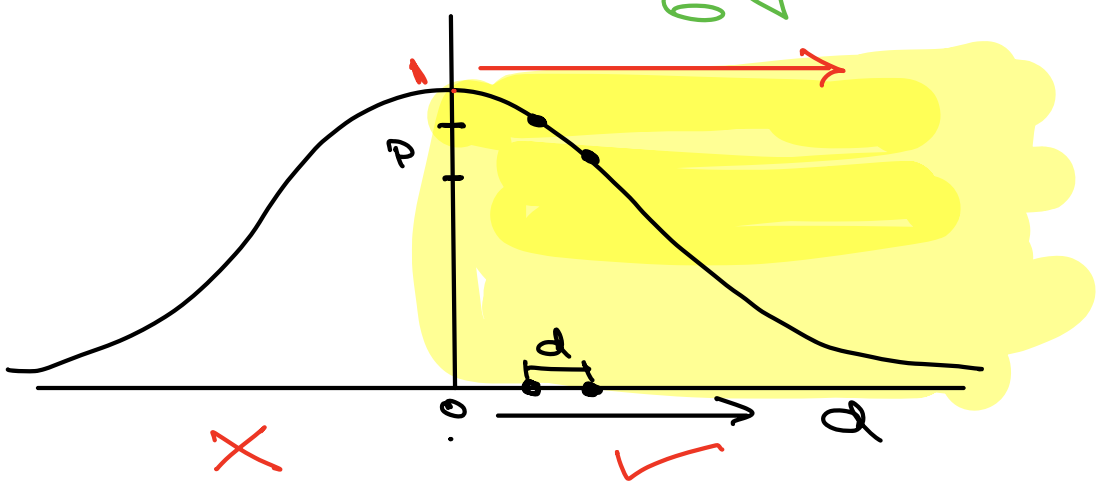
$P(x_1, x_4) \propto \frac{1}{10} \propto 0.10$



$\frac{1}{100} \propto 0.01$

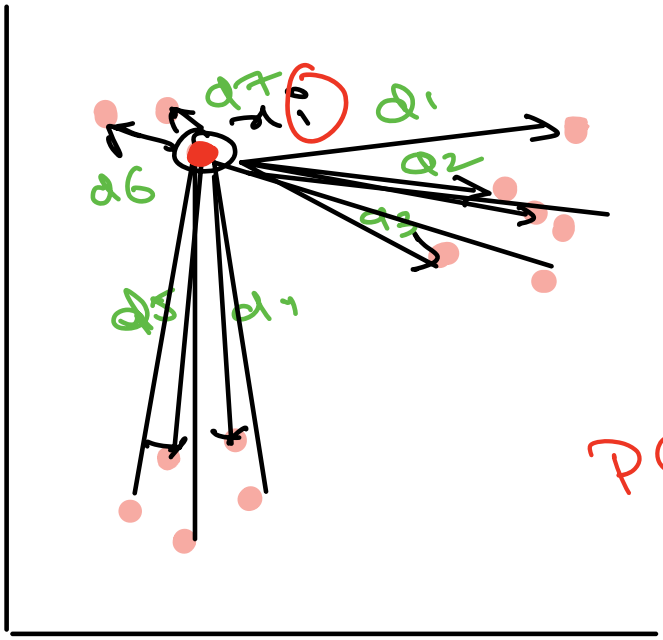
Can we use Normal Distribution instead

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



① $d \to 0$, $P \to 1$

② Small increase in distance does not lead to High decrease in prob (less Sensitive than $1/d(i,j)$)



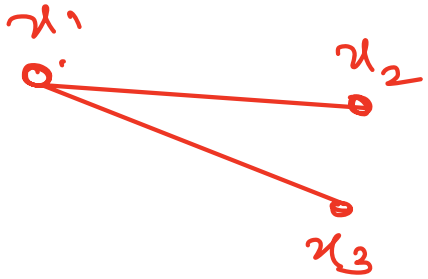
$P \propto \frac{1}{d(i,j)}$

with N.D as PDF

$$P(i,j) \propto \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-y)^2}{2\sigma^2}}$$

$x_j - y_i \Rightarrow (x_j - x_i)$

x_i

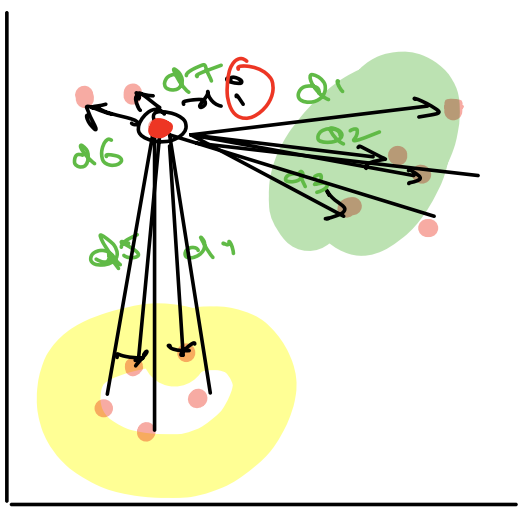


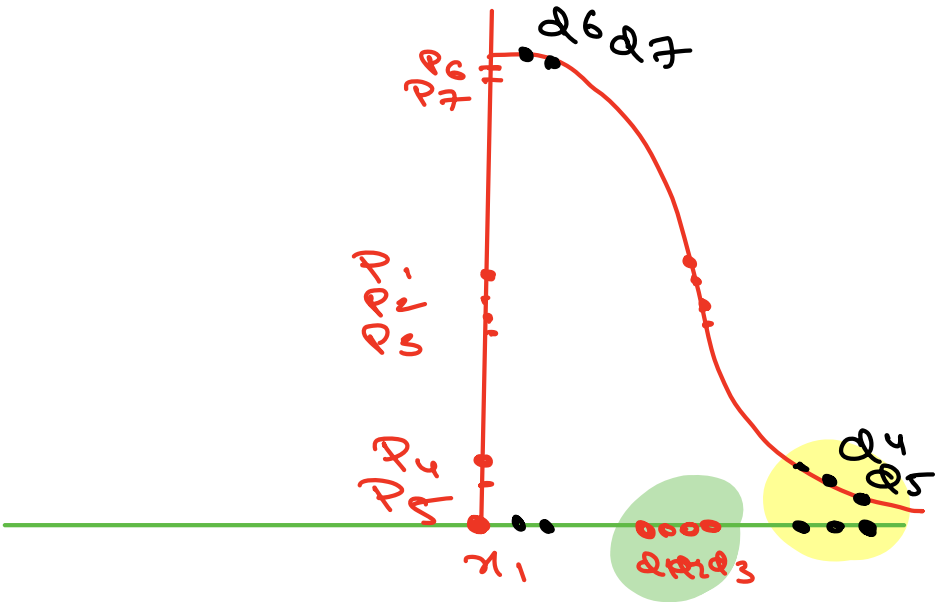
$$P(x_1, x_2) \quad \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_1-x_2)^2}{2\sigma^2}}$$

$d(i,j)$

$$P(x_1, x_3) \quad \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_1-x_3)^2}{2\sigma^2}}$$

$$\boxed{P(i,j) \propto \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}}}$$





Can we Normalize to make

Sum of $P_1 + P_2 + \cdots + P_n \to 1$

$$P(i,j) \to \frac{\cancel{\frac{1}{2\pi\sigma}} e^{-\frac{\|x_i - x_j\|^2}{2\sigma}}}{\sum_i \sum_j \cancel{\frac{1}{2\pi\sigma}} e^{-\frac{\|x_i - x_j\|^2}{2\sigma}}}$$

Normalize P.D.F

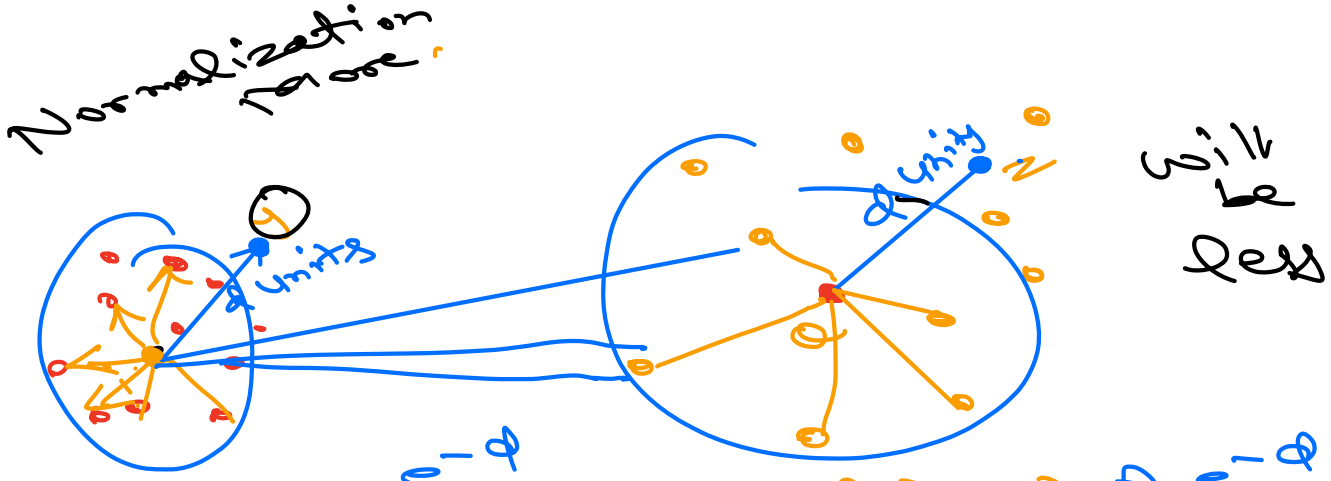
$$P(i,j) \to \frac{e^{-\|x_i - x_j\|^2 / 2\sigma_i^2}}{\sum_i \sum_{j \neq i} e^{-\|x_i - x_j\|^2 / 2\sigma^2}}$$

6. $\propto [d_1, d_2, d_3, d_4 \ldots d_n]$.std()

$$P(i,j) \propto \frac{e^{-\|x_i - x_j\|/2\sigma^2}}{\sum_{k \neq i} e^{-\|x_i - x_k\|^2/2\sigma^2}}$$

$$P(i,j) = \frac{e^{-\|x_i - x_j\|}}{\sum_{k \neq i} e^{-\|x_i - x_k\|^2}}$$

for calculating probability of Neighbours in Original feature space



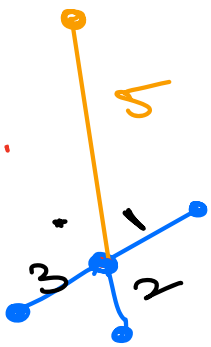
$$P(i,j) = \frac{e^{-\phi}}{P(\text{all points})} \qquad P(\Theta, \Rightarrow) \propto \frac{e^{-\phi}}{P(\text{all points})}$$

$$P(i,j) \propto \frac{e^{\|x_i - x_j\|}}{\sum_{k \neq i} e^{\|x_i - x_k\|^2}}$$

$k \neq i$ → is all the point in Dataset

we take only k nearest neighbors for Normalization

$$P(i,j) \propto \frac{e^{-d}}{\sum_{k=1}^{k} e^{-d}}$$



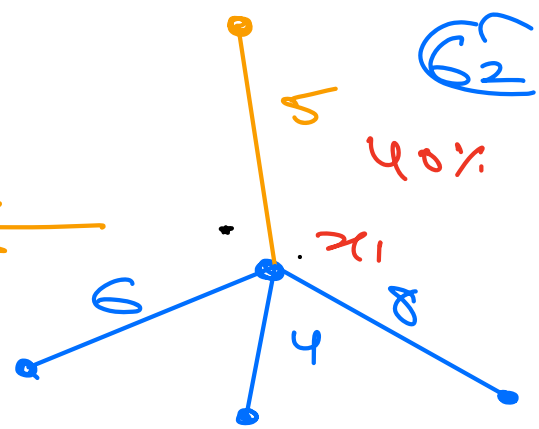
$n=3$

1.2%

$\propto \frac{e^{-5}}{e^{-1}+e^{-2}+e^{-3}}$ $\propto \frac{0.006}{0.36+0.13+0.04}$ $\propto 0.012$

G_1

$\frac{e^{-5}}{e^{-4}+e^{-6}+e^{-8}} \propto$



40% $\frac{0.006}{0.01+0.002+0.003}$ ←

- So for normalize we fix a hyper-parmeter n-neighbours to Handle different Densities

- Perplexity is a replacement of n-neighbors

  n_neighbors : int

  perplexity : float

Perplexity $= 2^{H(P_{i|j})}$

Where

$$H(P_{i|j}) = -\sum_{j} P_{i|j} \log_2 P_{i|j}$$

Shanon's Entropy

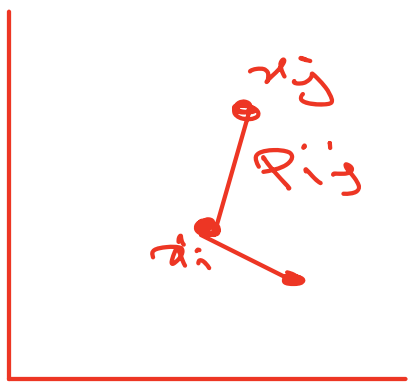
Perplexity : 20.0 - Dynamic n_neighbors

Probability of pairs being Neighbours in Original / High Dimension

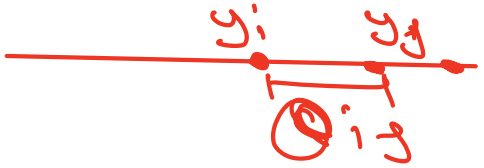
$$P_{ij} \propto \frac{e^{\|x_i - x_j\|^2}}{\sum_{k=1}^{n} e^{\|x_i - x_k\|^2}}$$

Can we use same Distribution and calculate probability in Lower / Transformed Dimension?

$$Q_{ij} \propto \frac{e^{\|y_i - y_j\|^2}}{\sum_{k=1}^{n} e^{\|y_i - y_k\|^2}}$$

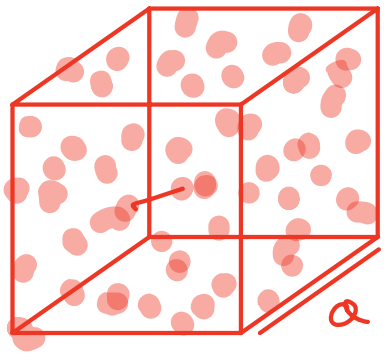


2D



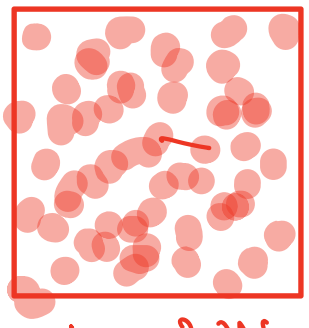
Crowding Problem

3d High



1 million D.P.'s

Low Dim 2d



1 million D.P.'s

① To avoid Crowding : $a' > a$

② $P_{ij} \approx Q_{ij}$

if i use same distribution for represent both P and Q

$P_{ij} \approx Q_{ij}$ will not be possible

$$P_{ij} = \frac{e^{-\|x_i - x_j\|^2}}{\sum_{k=1}^{n} e^{-\|x_i - x_j\|^2}}$$

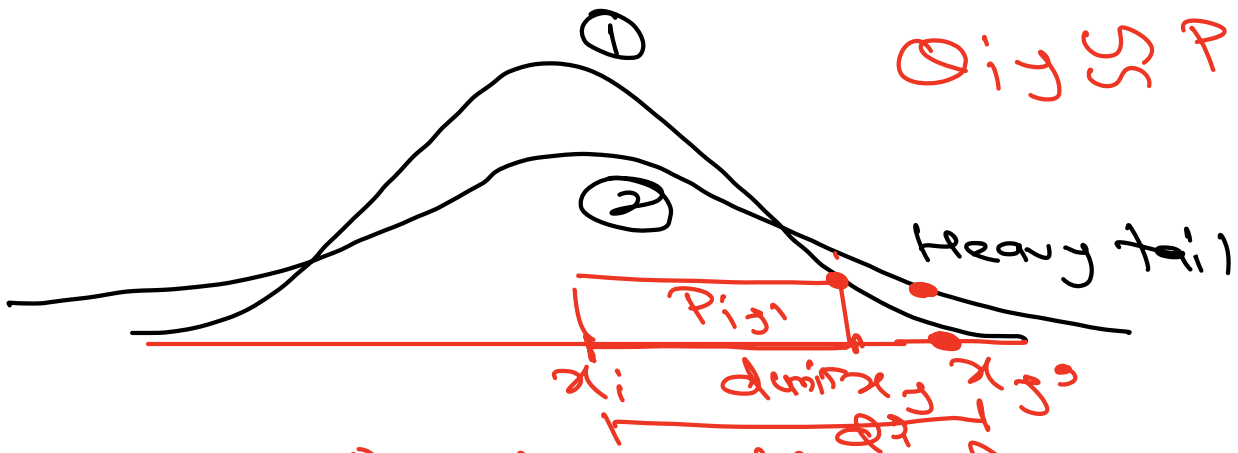
$$Q_{ij} = \frac{e^{-\|y_i - y_j\|^2}}{\sum_{k=1}^{n} e^{-\|y_i - y_j\|^2}}$$

$a < a'$

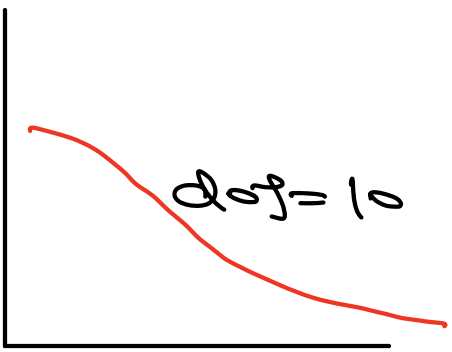
Hence for Lower Dimension we can use relaxed Normal Distribution

$Q \gg P$

$Q_{ij} \approx P_{ij}$



T - distribution with Degree of freedom = 1



$$q_{ij} = \frac{e^{\|y_i - y_j\|^2}}{\sum_{k=1}^{n} e^{\|y_i - y_k\|^2}} \Rightarrow \frac{\left(1 + \|y_i - y_j\|^2\right)^{-1}}{\sum_{k=1}^{n}\left(1 + \|y_i - y_k\|^2\right)^{-1}}$$

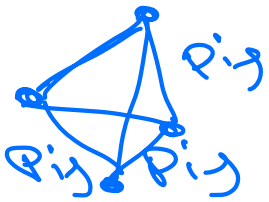
N. Dist

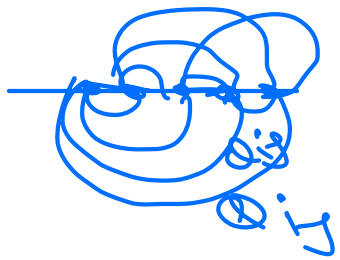
T - Dist

We have

In high dim $\Rightarrow P_{ij} \rightarrow$ Normal Dist

In low dim $\Rightarrow Q_{ij} \rightarrow$ T-Distribution (Cauchy Dist)





How do i Compare all the probabilities

↓

How do i compare and say that $P_{ij} \leq Q_{ij}$

[Normal Dist] [T Dist]

A new Loss function is required

- We use KL Divergence

$$KL_{div}(P, Q) = \sum_i \sum_j P_{ij} \log\left(\frac{P_{ij}}{Q_{ij}}\right)$$

- we use KL_{div} as Loss function

Case 1:

i and j overlap

$KL(P, Q) \to 0$ as $P_{ij} \to 0$

Case 2:

$P_{ij} > Q_{ij}$

$KL(P, Q) = \log\left(\frac{P_{ij}}{Q_{ij}}\right) \to > 1$

$= (\log(>1)) \times P_{ij}$

Case 3:

$P_{ij} = Q_{ij}$

$\log\left(\frac{P_{ij}}{Q_{ij}}\right) \times P_{ij} = P_{ij} \log(1) = 0$

- To solve and find optimal axis we can use KL-Div as loss function and apply Gradient Descent.

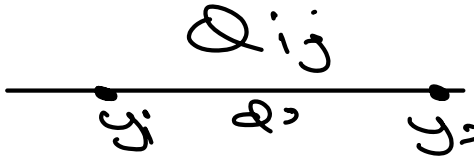
Drawback:

- T-SNE is very time intensive
- Hyperparameter: Perplexity Tuning

+ T-SNE is generally used for Visualization

⇒ Hence n_components → 2, 3

- Umap and Umap



$x_i, x_j \rightarrow P_{ij} \ni 0$ $\quad Q_{ij} = P_{ij}$ ⊙ Same prob scores