Lecture 8

21/1/2022

Graph Construction / Graph Learning

- $\{x_1, x_2 \ldots x_n\}$

$\Rightarrow (\cancel{\{x_i\}},\ x\, y$

$(\{x_i, y_i\})$

f_1 f_2 f_3

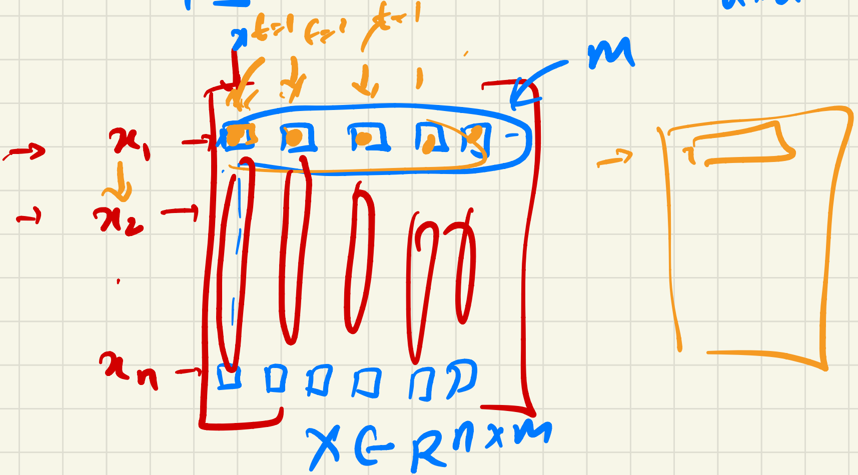
$$X = \begin{bmatrix} | & | & & | \\ x_1 & x_2 & \cdots & x_d \\ | & | & & | \end{bmatrix} \Longrightarrow \begin{bmatrix} & \\ & \end{bmatrix}_{d\times d}$$

$x_1 \leftarrow P_1$

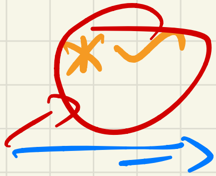
$x_2 \leftarrow P_2$

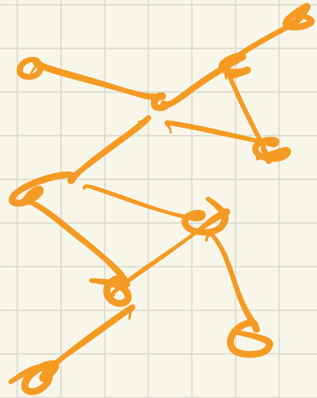
$\vdots$

$x_n \leftarrow P_n$



$$\begin{bmatrix} X \end{bmatrix}_{n \times m}$$





$\Rightarrow n$: is the no. of dim,

Graph learning from data.

* Graph-Based Semi-Supervised learning

  * Amarnag Subramanya

* Partha Pratim Talukdar

- Semi-Supervised learning with Graphs

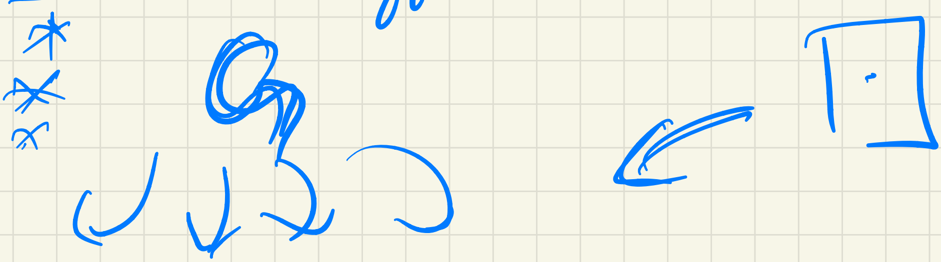
→ Thesis CMU

Graph Learning Problem

⇒

* Task dependent graph learning
  - → $\langle x_i, y_i \rangle$
* Task independent graph learning
  - $\langle x_i, \{x_i, \dots \} \rangle \rightarrow G$

* Pairwise Graph learning → correlation graph
* Model based graph learning
  - ⇒ PGM
  - → Diffusion based Graphical Model

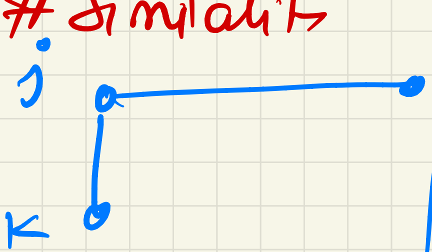


$$[X] \longrightarrow [D] \longrightarrow [W]$$

$n \times m$ $\qquad$ $n \times n$ $\qquad$ $n \times n$

caw
→ pai
→ Iwend.

similarity



$\Rightarrow d_{jk} < d_{ji}$

$[x_j, x_k]$ $\qquad$ x_j, x_i

$w_{jk} > w_{ji}$

- Similarity / Dissimilarity

$$X = [x_1 \in R^m, \ldots, x_n \in R^m]$$

① Cosine similarity

$$= \frac{x_i^T x_j}{\sqrt{x_i^T x_i}\ \sqrt{x_j^T x_j}}$$

② Pearson's correlation

③ Spearman's correlation

④ → Partial correlation

$$P_{ij,V \setminus \{i,j\}} = \frac{-\theta_{ij}}{\sqrt{\theta_{ii}\,\theta_{jj}}}$$

$\mathcal{N}(0, \Sigma)$, $\Sigma = \Theta^{-1}$

Θ_{ij}

(5) Simple matching coef.

$$\frac{1}{d} \sum_{k=1}^{M} \mathbb{I}\{x_{ik} = x_{jk}\}$$

(6) Hamming distance

(7) Jaccard distance

(8) Mahalanobis distance

$$-(x_i - x_j)^T M (x_i - x_j) , \quad M \succ 0$$

(9) Chi-square distance

$$d_M(x_i, x_j) = (x_i - x_j)^T M (x_i - x_j)$$

$$M = PP^T$$

$$= (x_i - x_j)^T P P^T (x_i - x_j)$$

$$\boxed{d_P(x_i, x_j) = \|Px_i - Px_j\|_2^2}$$

$[x] \longrightarrow [D] \longrightarrow [W]$

S_{sim}

* K- nearest Neighbour Graph

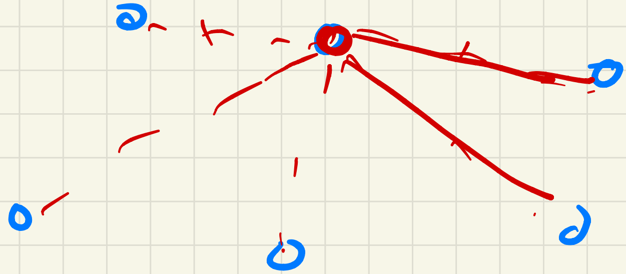
ε - Graph

$[X] \longrightarrow [sim(x_i, x_j)]$

$$W_{ij} = \begin{cases} sim(x_i, x_j), & \text{if } x_i \text{ is the nearest neighbours of } j \text{ or vice versa} \\ 0 & \text{o.w} \end{cases}$$

$$\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \longrightarrow \begin{bmatrix} d(x_1,x_1) & d(x_1,x_2) & \cdots \\ & & \\ & & \end{bmatrix}_{n \times n}$$

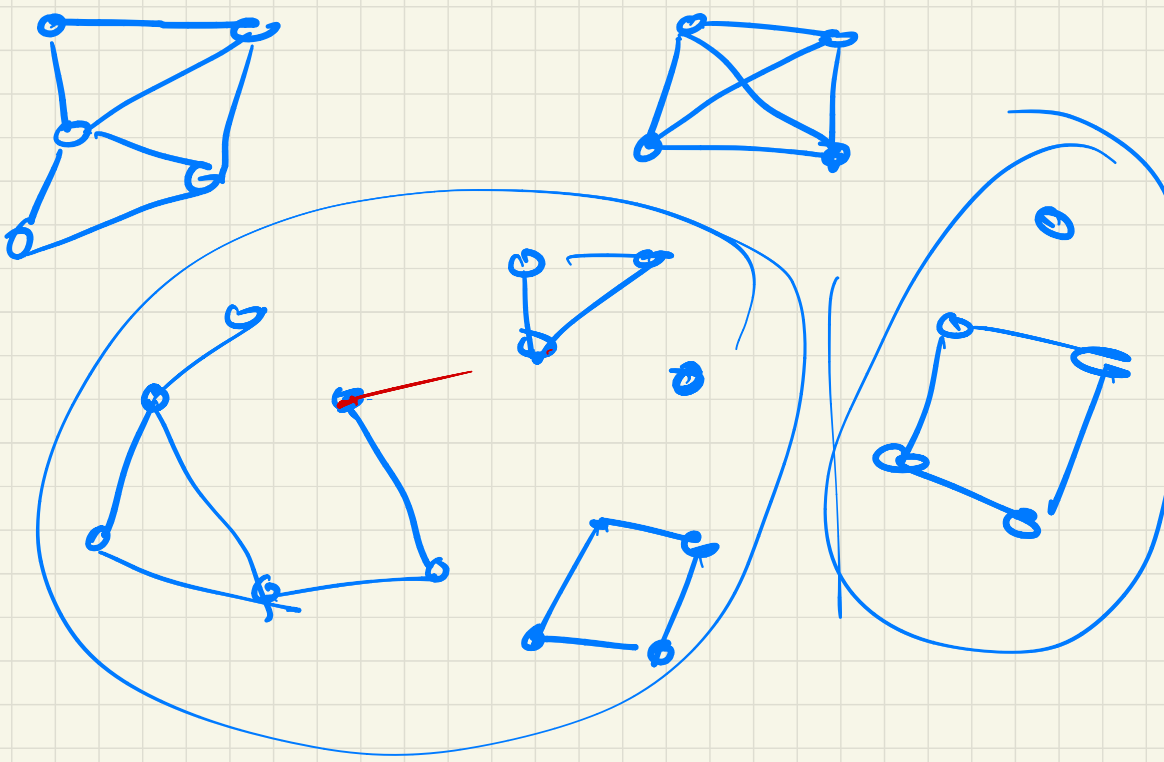
OO k- nn



ϵ

if $d_{ij} > \epsilon$ $\quad w_{ij} = 0$

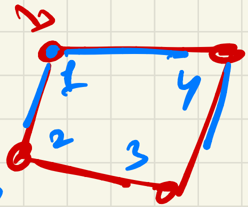
$d_{ij} \leq \epsilon$ $\quad w_{ij} \sim f(d_{ij})$



Regular Graph

$w_{ij} = \{0, 1\}$

$$W = \begin{bmatrix} w_{11} & \cdots & w_{1n} \\ w_{n1} & \cdots & w_{nn} \end{bmatrix}$$



$d_2 = w_{12} + w_{23}$

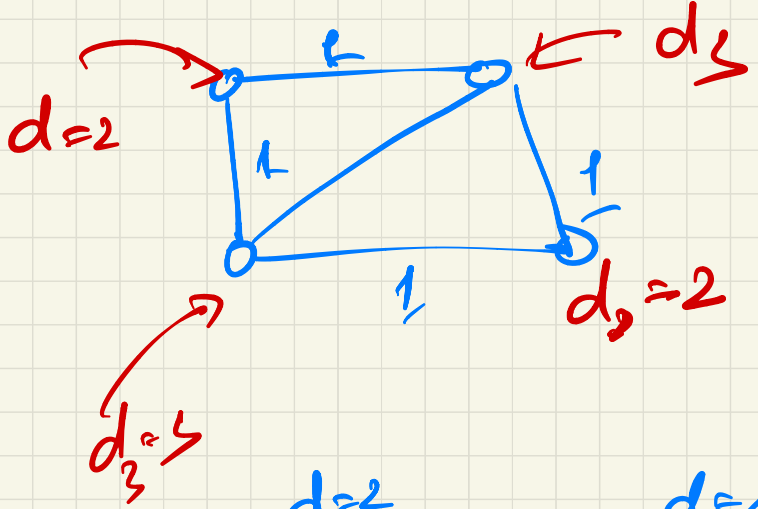
$$D = \begin{bmatrix} \sum_{i \neq j} w_{ij} & \\ & \ddots \end{bmatrix}$$

Regular Graphs

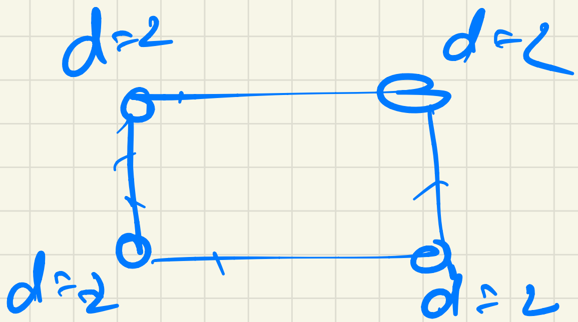
* All the nodes will have same degree

d

* for unweighted graph, regular graph means all nodes will have same no of neighbors



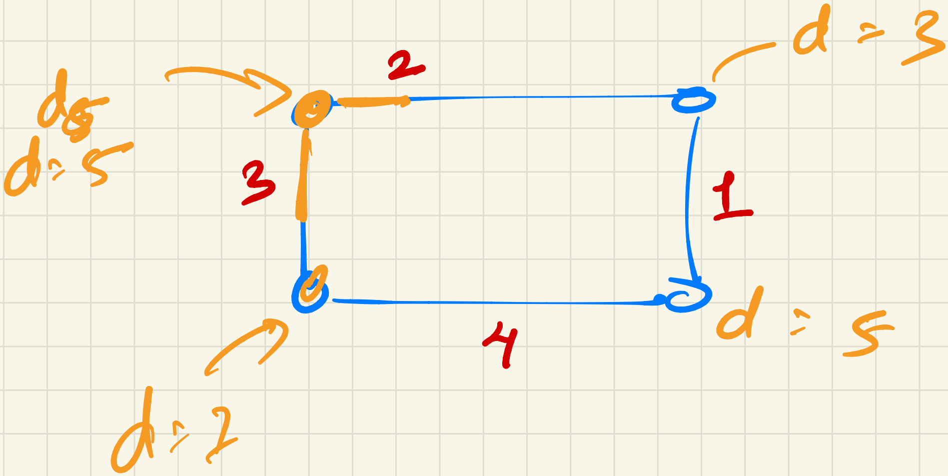
d=2
1
1
1
1
d_1
d_2=2
d_3=4



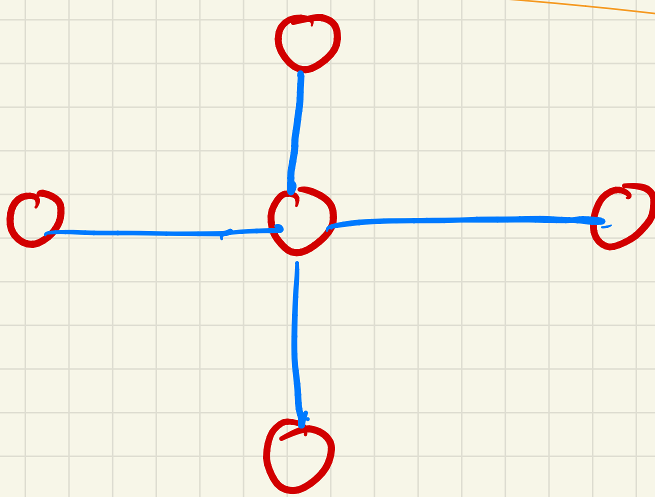
d=2
d=2
d=2
d=2

$$d_1 = \sum_{j \neq 1} w_{1j} \quad ,$$

$$d_i = \sum_{j \neq i} w_{ij}$$



k-NN



B matching

$\boxed{D} \Rightarrow \boxed{W}$

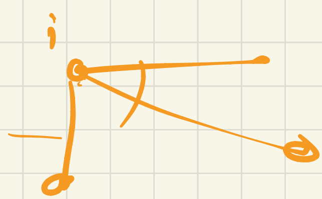
↳ Graph sparsification

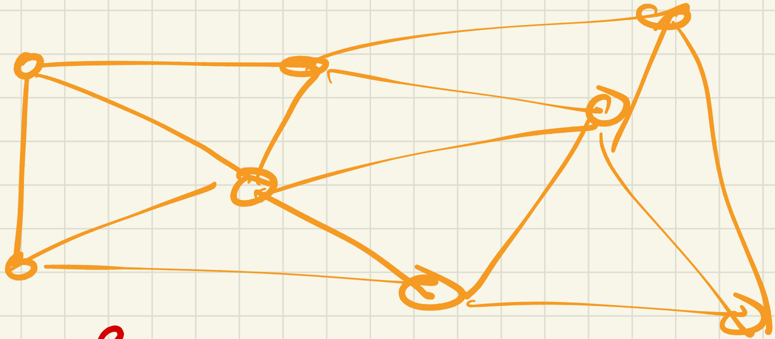
$D \simeq \Delta$

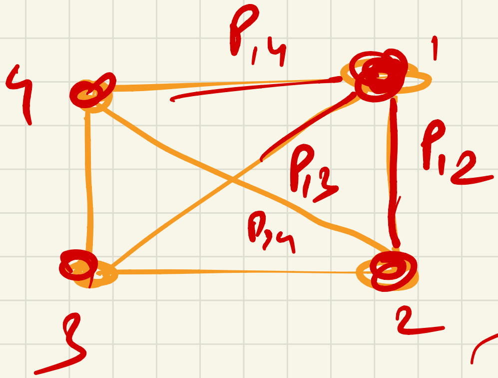
Optimization

$$\Delta = \begin{bmatrix} \delta_{12} & \delta_{13} & \cdots & \delta_{1n} \\ \delta_{n1} & & & \delta_{nn} \end{bmatrix}$$

$P_{ij} \in \{0,1\}$, Solve for P_{ij})







$$\sum P_{12} + P_{13} + P_{14}$$
$$= P_{12} + P_{24} + P_{23}$$

$$\boxed{\sum_j P_{ij} = k}$$

$X \to D \to W$

$\Delta \to P_{ij}$?

s.t. $\sum_j P_{ij} = k$, $P_{ii} = 0$

$$\min_{P \in \{0,1\}^{n \times n}} \sum P_{ij} \, \delta_{ij}$$

D

$$\frac{P_{ij}}{P_{ji}}$$

Subject to $\sum_j P_{ij} = k$, $P_{ii} = 0$

$1 \le i, j \le n$

$$\sum P_{ij} = k, \quad P_{ii} = 0$$

$$P_{ij} = P_{ji}, \quad \forall\ 1 \le i, j \le n$$

$$D = \begin{bmatrix} & 0 & & \\ 0 & & 0 & 2 \\ & c & & d \end{bmatrix} \qquad \begin{bmatrix} 0 & 0 & 1 & 0 \\ & & & \\ & & & \end{bmatrix}$$

$\downarrow$

$\boxed{W}$

$d_{ij} < d_{ik}$

$w_{ij} > w_{ik}$

Edge-Reweighting

① $W = P$ (Binary graph)

② Gaussian kernel (GK):

$$w_{ij} = P_{ij} \exp\left(-\frac{d(x_i, x_j)}{2\sigma^2}\right)$$

* Fitting a Graph to Vector Data