April 16, 2020

Last time: graph convolutional networks

$$\left({}_n\overset{n}{\boxed{A}}, {}_n\overset{d}{\boxed{F}} \right) \xrightarrow{f} {}_n\overset{k}{\boxed{X}} \xrightarrow{g} {}_n\overset{1}{\boxed{\hat{Y}}} \xrightarrow{\ell} (\hat{y}_H, \ell_H)$$

$$\hat{y} = \begin{bmatrix} \hat{y}_H \\ \hat{y}_u \end{bmatrix}$$ predictions on unlabeled nodes

Smoothed features:

ith column $\quad x_i \approx \underset{z}{\operatorname{argmin}} \; f(z) + \lambda \|z - f_i\|_2^2$

$z^T L z \quad z^T N z \quad z^T (I - P) z$

Each GD steps:

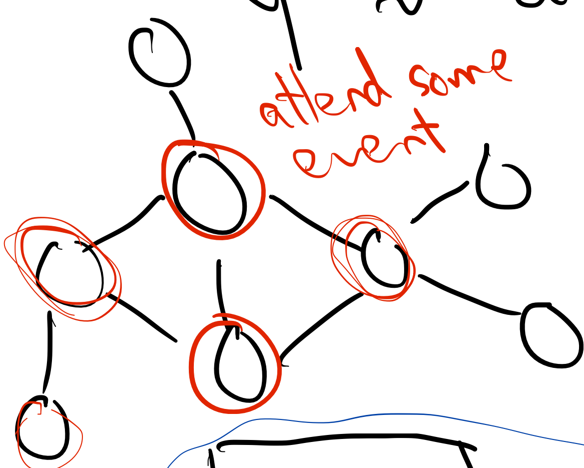
$$x_i^{(t+1)} \approx \Big[\underbrace{\alpha P + (1-\alpha) I}_{M}\Big] x_i^{(t)}$$

$$X^{(t+1)} \approx M X^{(t)} \overset{\text{GCN}}{\Longrightarrow} X^{(t+1)} = \sigma\left(M X^{(t)} W^{(t)} + b^{(t)}\right)$$

$W^{(t)}, b^{(t)}$ couples embedding and prediction task

Smoothed labels

$$\underset{n \times 1}{y} \approx \arg\min_{z} f(z) + \lambda \left\| \begin{pmatrix} y_H \\ y_u \end{pmatrix} - \begin{pmatrix} l_H \\ 0 \end{pmatrix} \right\|_2^2 \quad (!)$$



location, history, etc.

seems like we should be using the label

$$\left(\underset{n \times n}{\boxed{A}},\ \underset{n \times d}{\boxed{F}},\ \underset{n \times 1}{\boxed{\begin{matrix} l_H \\ 0 \end{matrix}}} \right)$$

label as input feature!

$$X^{(t+1)} = M X^{(t)} = [\alpha P + (1-\alpha) I] X^{(t)}$$

$$P = D^{-1} A \quad \text{row stochastic}$$

i^{th} row

$$\overbrace{x_i^{(t+1)}}^{\text{next rep.}} = \alpha \overbrace{\underbrace{\left[\frac{1}{d_i} \sum_{(i,j) \in E} x_j^{(t)} \right]}_{X_{N(i)}^{(t)}}}^{\text{average of neighbors}} + (1-\alpha) \overbrace{\left[x_i^{(t)} \right]}^{\text{current rep}}$$

$$x_i^{(t+1)} = \phi\left(X_{N(i)}^{(t)}, \; x_i^{(t)} \right)$$

one example: $1 \; \underset{X_{N(i)}}{\overset{k}{\boxed{\qquad\qquad}}} \quad 1 \; \underset{x_i}{\overset{k}{\boxed{\qquad}}} \quad 2k \; \overset{k}{\boxed{W}} \quad \begin{bmatrix} \alpha \\ \hline 1-\alpha \end{bmatrix}$

$$X^{(t+1)} = \sigma\Big(\underbrace{\big[\, \underset{X_{N(i)}^{(t)}}{PX^{(t)}} \;\; X^{(t)} \,\big]}_{n \times 2k} \; \underbrace{W^{(t)}}_{2k \times k} + \mathbf{1}\, \underbrace{b^{(t)}}_{1 \times k}{}^{T} \Big)$$

i^{th} row: $PX^{(t)} = \frac{1}{d_i} \sum_{(i,j)\in E} x_j^{(t)} = X_{N(i)}^{(t)}$

GraphSAGE (Hamilton et al. 2017)

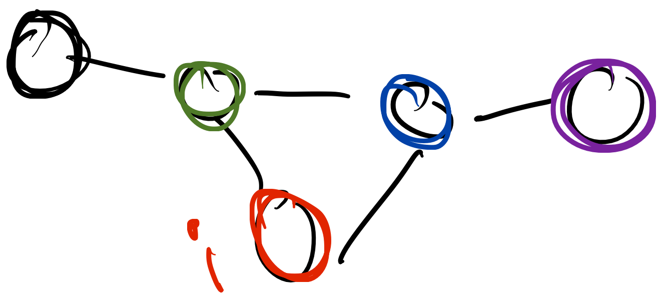
Idea: think of other "aggregation" methods

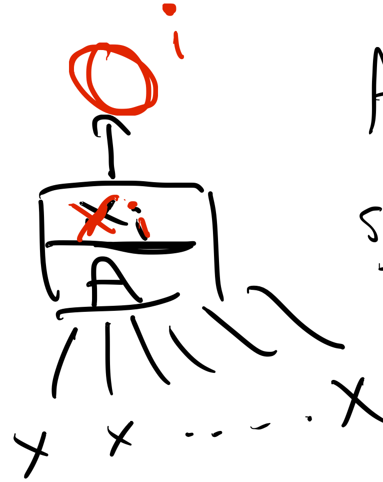
$$X_{N(i)}^{(t+1)} = \frac{1}{d_i} \sum_{(i,j)} X_j^{(t)} \quad \text{(mean)}$$

$$X_{N(i)}^{(t+1)} = \frac{1}{\sqrt{d_i}} \sum_{(i,j)} X_j^{(t)} \Big/ \sqrt{d_j}$$

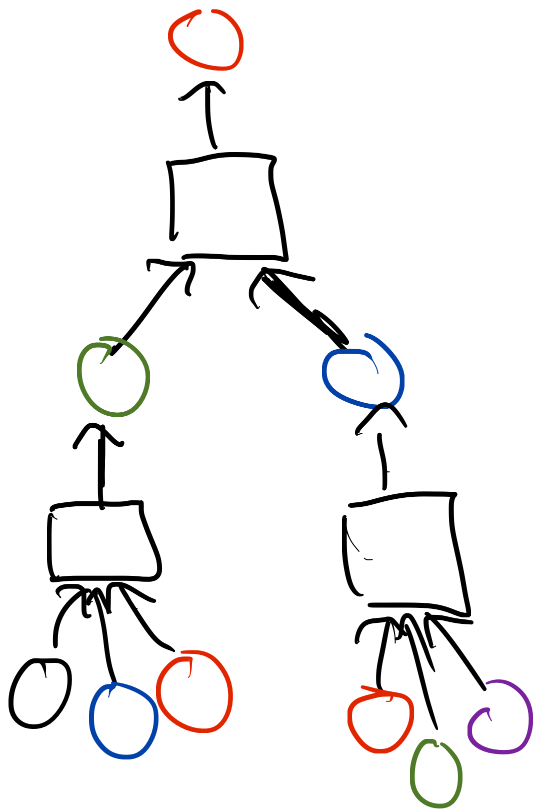
$$X_{N(i)}^{(t+1)} = \max\left(\{x_j^{(t)}\} \mid j \in N(i)\right\}$$

(max-pooling) entry-wise

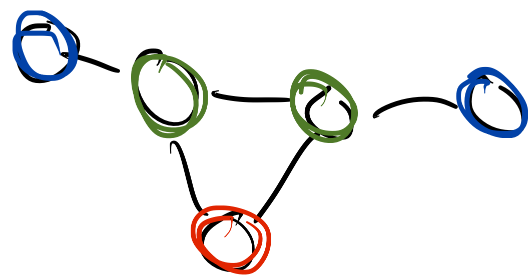




A: aggregation
set function



Extreme case:
all nodes had same features
$\Rightarrow$ learn nothing



Augment features?
degree?

Combine with spectral, node2vec
cost?

Types of problems:

(1) clustering / node predict

(2) link prediction

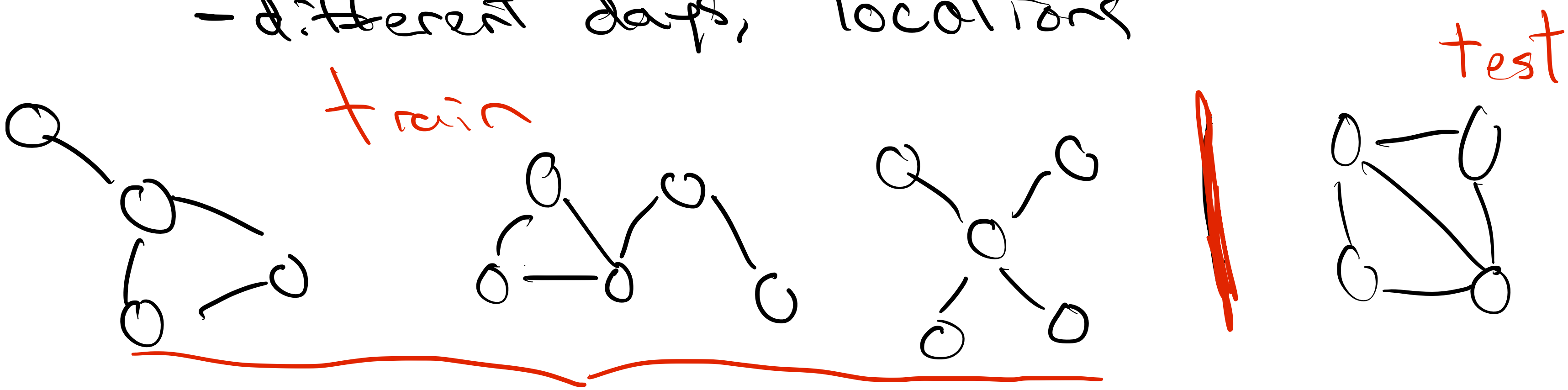
Might want to do this across graphs

PPI graphs : predict function, links

- different tissues
- different species

Traffic networks: predict flow

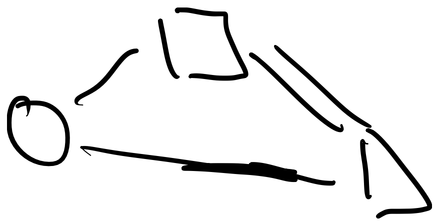
- different days, locations



one big graph, disjoint components

Another task: graph classification

Molecular graphs:

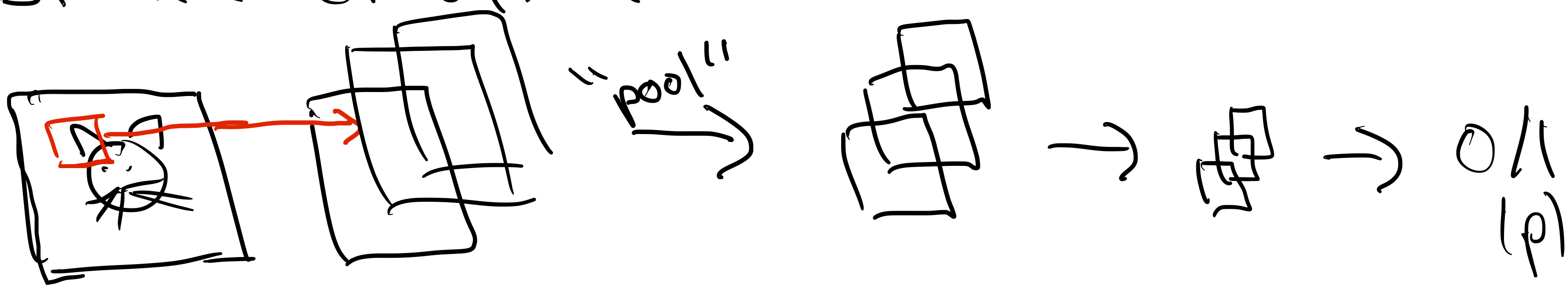


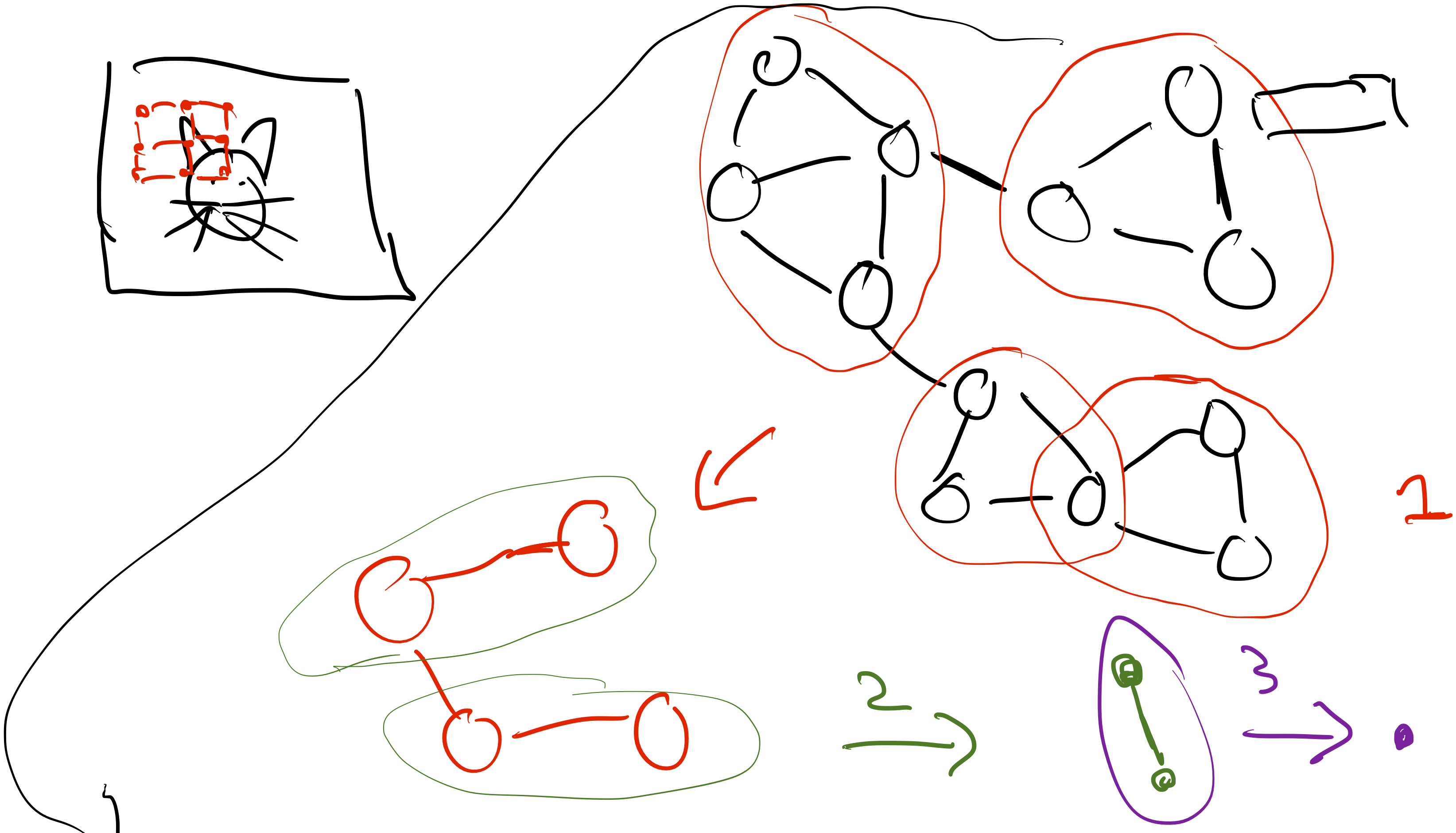
taste good?

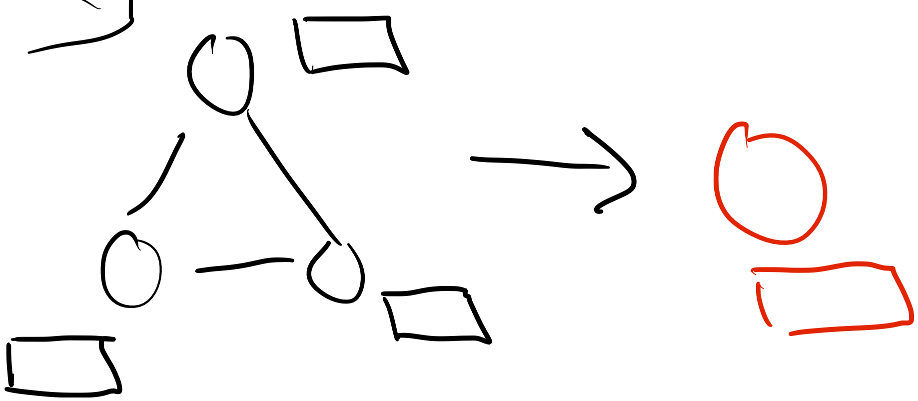
effective drug?

$$\left\{ \left({}_{n}\overset{n}{\boxed{A_i}} \, , \; {}_{n}\overset{d}{\boxed{F_i}} \right) \right\}_i \longrightarrow \{0/1\}_i \;\; (p)$$

Bruna et al. (2014)







Apply some "filters"
(learn from data)
then pool

$$\left({}_{n}\overset{n}{\boxed{A}}\,,\ {}_{n}\overset{d}{\boxed{F}} \right) \xrightarrow{f} {}_{n}\overset{k}{\boxed{X}} \xrightarrow{\text{pool}} z^T \to \text{logit}(z^T \beta)$$

Idea: pool rows of X

Set function $\{x_i\} \to z$

$$z = \frac{1}{n} \sum x_i$$

$$z = \max(\{x_i\}) \text{ entrywise}$$

$$z \quad {}_{k}\overset{1}{\boxed{}}$$