Graph Convolutional Networks for recommender systems

Paula Gómez Duran
paulagomezduran@gmail.com

PhD student

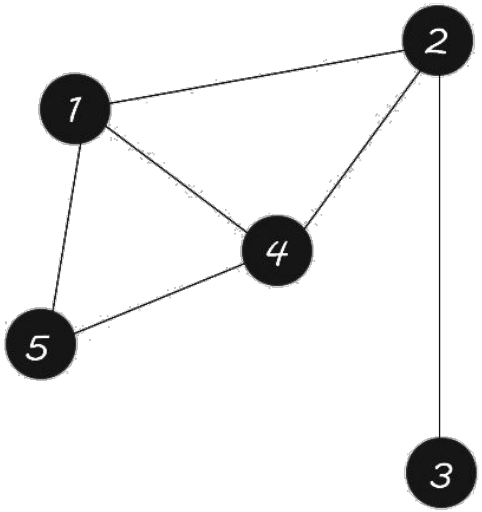
UNIVERSITAT DE BARCELONA

What is a GRAPH?

A Graph is anything with nodes connected by edges.

We focus on **Graph Convolutional Networks** (GCN).

Graph Convolutional Network



$$A = \begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 0 & 1 & 0 & 1 & 1 \\ 2 & 1 & 0 & 1 & 1 & 0 \\ 3 & 0 & 1 & 0 & 0 & 0 \\ 4 & 1 & 1 & 0 & 0 & 1 \\ 5 & 1 & 0 & 0 & 1 & 0 \end{array}$$

$\boldsymbol{X}$ = *input features*

GCN equation

Defined in: https://arxiv.org/pdf/1609.02907.pdf

D = degree matrix
A = adjacency mx
or
rating mx (R)

Symmetric normalization

New layer → H(1)

Previous layer → H(0) = X

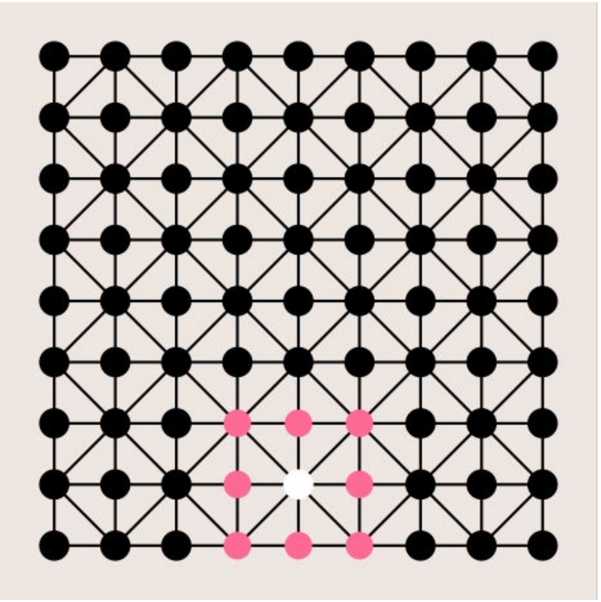
$$\boxed{H^{(l+1)}} = \sigma\left(\boxed{\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}}\boxed{H^{(l)}}\boxed{W(l)}\right)$$

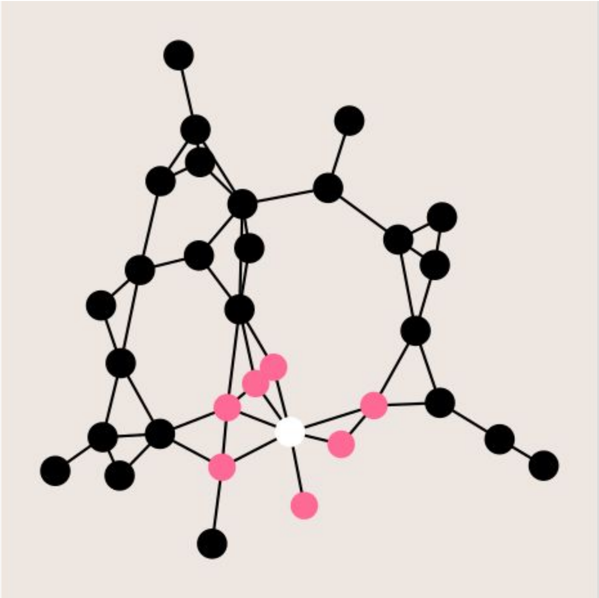
Weights we train

GOAL: Have, in each node, information not just about the node itself but also about correlated nodes.

GCN layer equation

$$H^{(l+1)} = \sigma\left(\boxed{\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}}\,\boxed{H^{(l)}}\,\boxed{W(l)}\right)$$





GCN equation defined in: https://arxiv.org/pdf/1609.02907.pdf

GCN equation

$$EMBEDDINGS = \hat{D}^{-\frac{1}{2}} \hat{A} \hat{D}^{-\frac{1}{2}} \; X \; W(l)$$

$$g(x) = H(1)$$

We will use just one GCN layer to extract embeddings

because

more than one GCN layer is prone to overfit!

How can GCN improve embeddings?

No side information → X=I

GCN equation: $$H^{(l+1)} = \boxed{\hat{D}^{-\frac{1}{2}} \hat{A} \hat{D}^{-\frac{1}{2}}} \; \mathrm{I} \; W(l)$$

≈ A

Usual embedding = W
GCN embedding = Â * W

RANDOM

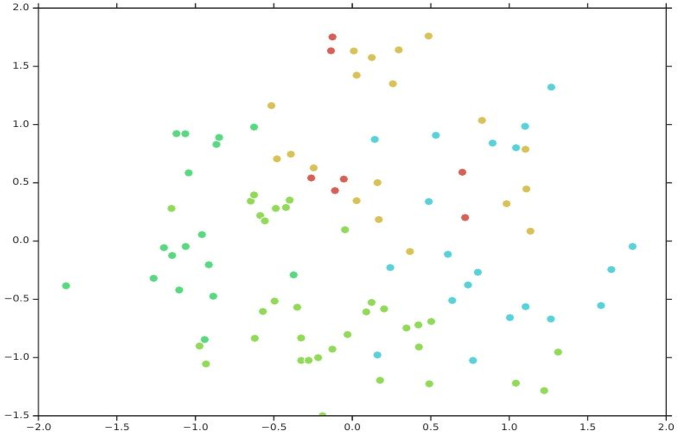
TOPOLOGY * RANDOM

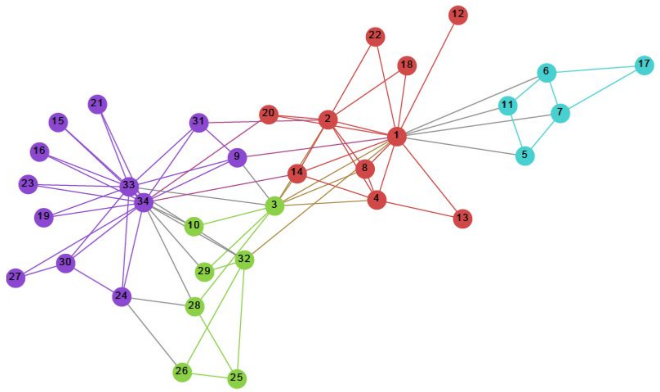
GCN embeddings lead to best performance even without training (1st epoch) !

1st epoch (no training yet):

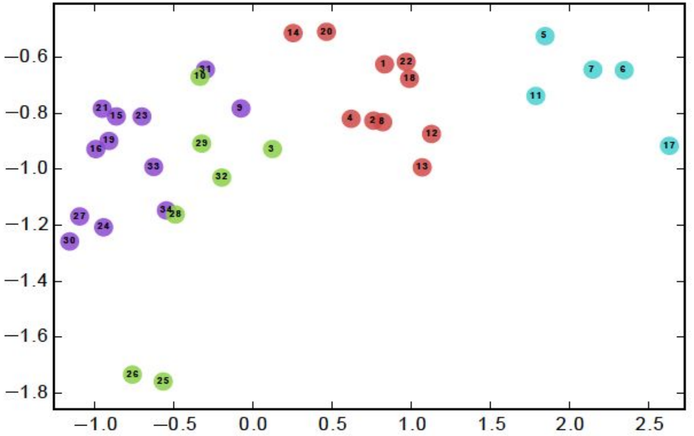
	o_ENE	o_ESE	o_East	o_NE	o_NNE	o_NNW	o_NW	o_SW	o_South	o_Variable	o_WSW
0	1	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	1	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	1	0
4	0	0	0	0	1	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0	0	0	0
6	0	0	0	0	1	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	1	0	0	0	0	0	0	0

RANDOM

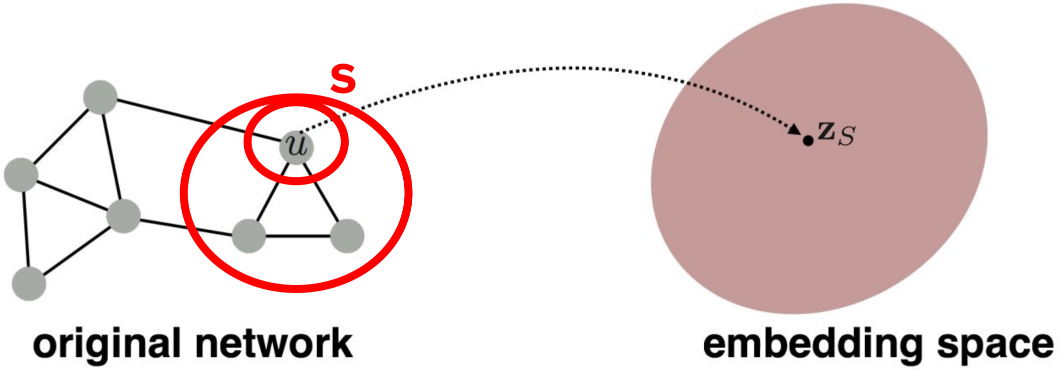




LEVERAGE STRUCTURE



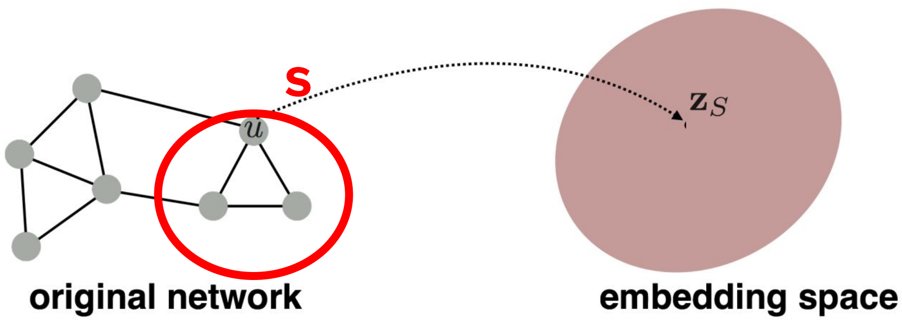
USUAL EMBEDDING vs GCN EMBEDDING:

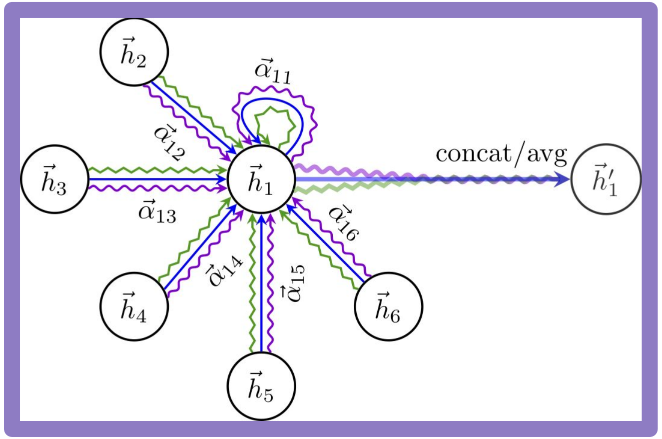


GCN vs Graph Attention Network (GAT)

Paper here

$$h_i^{(l+1)} = \sigma\left(\sum_{j\in\mathcal{N}(i)} \frac{1}{c_{ij}} W^{(l)} h_j^{(l)}\right) \Rightarrow h_i^{(l+1)} = \sigma\left(\sum_{j\in\mathcal{N}(i)} \boxed{\alpha_{ij}^{(l)}} W^{(l)} h_j^{(l)}\right)$$





FM PROBLEM!

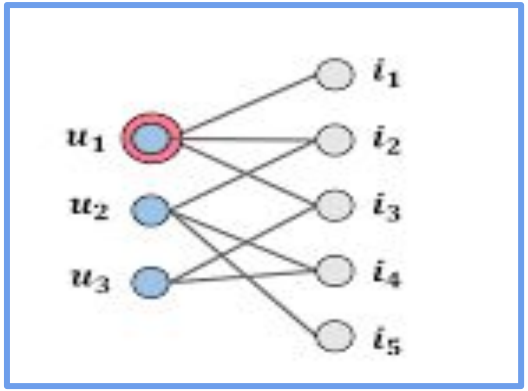
NF … D

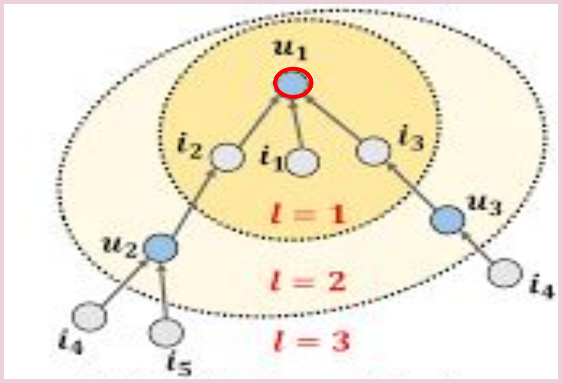
SOLUTION WITH FM:

GCN for capturing embeddings!

FM captures interactions of only second order! *($\ell = 1$)*

We might want to capture high order interactions ....

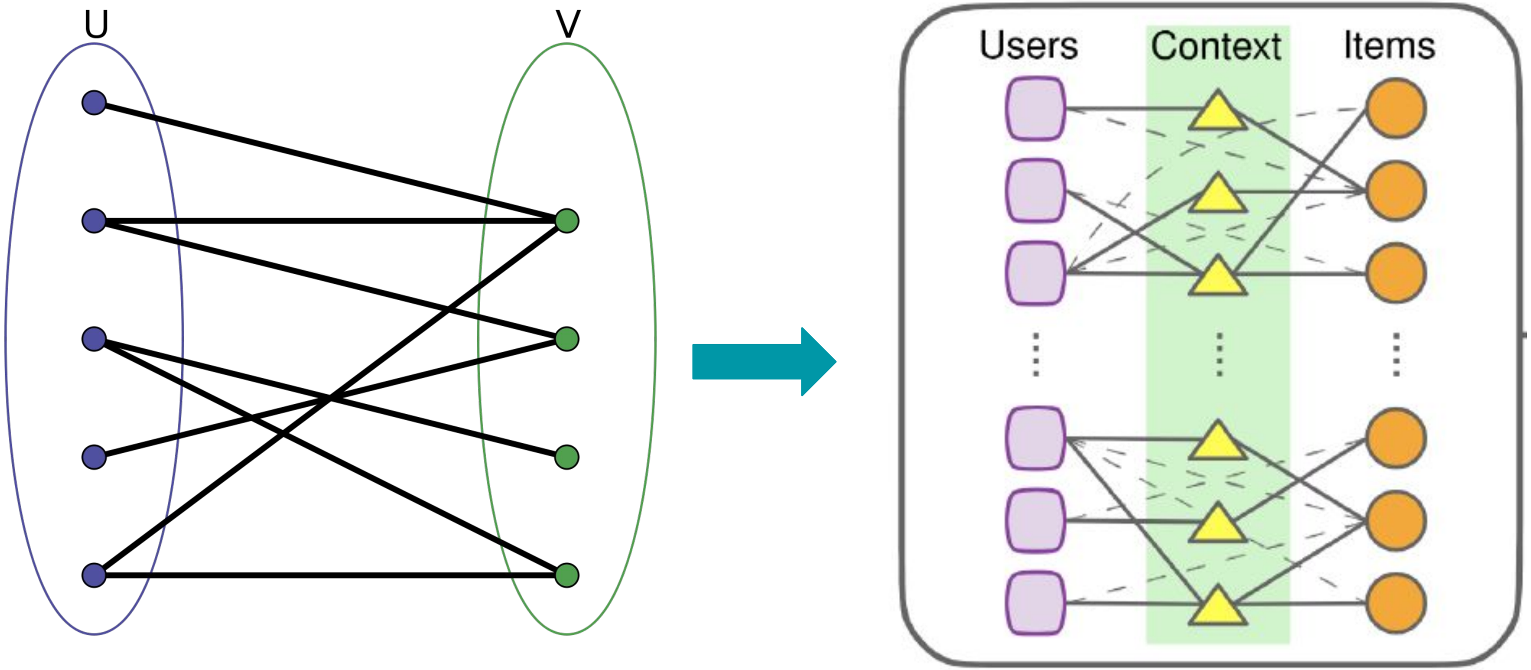




SO ....

HOW CAN WE USE GCN IN RECOMMENDATION?

To incorporate GCN we should extend GCN



N-Partite Graph

Extended message passing

$$\boxed{\mu_{j\rightarrow i} = \frac{1}{s_{ji}} W_u \, h_j}$$

$$\boxed{\mu_{j,e,\ldots,o\rightarrow i} = \frac{1}{s_{ji}} W_u h_j + \frac{1}{s_{ei}} W_{c_1} h_e + \cdots + \frac{1}{s_{oi}} W_{c_{N-2}} h_o}$$

- Sij ➜ symmetric normalization / left normalization
- Wu ➜ trainable weight matrix for the users factors; Wc1 for the first context factor, etc...
- hj ➜ input embedding for node **j** ➜ **first layer hj == Xj (feature vector)**

Extended message passing

	User_0 …. User_N_u	Item_0 … Item_N_i	Context_0 … Context_N_C
User_0 … User_N_u	0	Interactions user-item	Interactions user-context
Item_0 … Item_N_i	Interactions user-item	0	Interactions item-context
Context_0 … Context_N_C	Interactions user-context	Interactions item-context	0

How to use GCN for embedding generation:

$$\boxed{g(x) = \hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}} X \; W(l)}$$

$$\hat{y}(\mathbf{x}) := w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \boxed{\langle \boldsymbol{g}(x_i), \boldsymbol{g}(x_j) \rangle} x_i x_j$$

RECAP:

$$\text{FM} = \frac{1}{2}\sum_{f=1}^{k}\left(\left(\sum_{i=1}^{n} v_{i,f}\, x_i\right)^2 - \sum_{i=1}^{n} v_{i,f}^2\, x_i^2\right)$$