OLX GROUP

ADAM MICKIEWICZ UNIVERSITY POZNAŃ

Part 5: Graph-based models

Robert Kwieciński

OLX Group and Adam Mickiewicz University

May 22, 2020

Different types of graph models

There exists many recommendation models utilizing graphs.

Depending of what is the node in the graph we distinguish three basic categories:

- user-user graphs where nodes are users
- item-item graphs where nodes are items
- user-item graphs where nodes are both users and items

It is also possible to treat other objects as nodes - for example some features of items or users like tags or categories.

User-user graphs

Example approach is described in [1], pages 63-66.
The idea is as follows:

- each node of a graph is a user,

User-user graphs

Example approach is described in [1], pages 63-66.
The idea is as follows:

- each node of a graph is a user,
- a directed edge between users u and v exists if there exists a linear transformation which transforms ratings of one user onto the ratings of another accurately (only movies rated by both users considered),

User-user graphs

Example approach is described in [1], pages 63-66.
The idea is as follows:

- each node of a graph is a user,
- a directed edge between users u and v exists if there exists a linear transformation which transforms ratings of one user onto the ratings of another accurately (only movies rated by both users considered),
- a prediction of a rating $\hat{r}_{ui}^{(v)}$ is a rating of r_{vi} under the composition of linear transformations of edges on the shortest path connecting u and v,

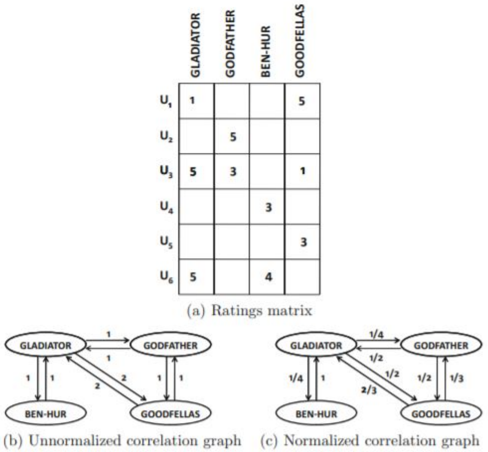
User-user graphs

Example approach is described in [1], pages 63-66.
The idea is as follows:

- each node of a graph is a user,
- a directed edge between users u and v exists if there exists a linear transformation which transforms ratings of one user onto the ratings of another accurately (only movies rated by both users considered),
- a prediction of a rating $\hat{r}_{ui}^{(v)}$ is a rating of r_{vi} under the composition of linear transformations of edges on the shortest path connecting u and v,
- a final prediction of a rating $\hat{r}_{ui}$ is an average of $\hat{r}_{ui}^{(v)}$ over distinct v's

Item-item graphs

Item-item graph is an alternative way of defining similarity measure over items and might be used by other algorithms (for example kNN).



(a) Ratings matrix

(b) Unnormalized correlation graph (c) Normalized correlation graph

Figure: Item-item graph. Source: [1]

User-item bipartite graph

In user-item graphs:

- each user and item is represented by a node,
- the edge between nodes exists if there was an interaction between given user and item,
- it is possible to add weights between edges.

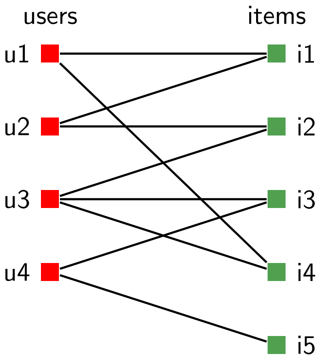


Figure: Bipartite graph of 5 users and 5 items

Adjacency matrix

Adjacency matrix of a graph of n nodes is a $n \times n$ matrix where $a_{i,j}$ is the number of edges connecting nodes i and j.
In our example we have

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

Note that $A^s(i,j)$ is the number of paths connecting vertices i and j in exactly s steps.

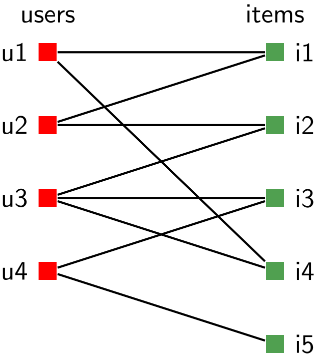


Figure: Bipartite graph of 5 users and 5 items

3-Paths

3-Paths model

For a given user u we can recommend items with the greatest number of paths of a given order starting at node u.

The smallest order we can take is 3 and usually performance is not improved with a greater order.

3-Paths

3-Paths model

For a given user u we can recommend items with the greatest number of paths of a given order starting at node u.

The smallest order we can take is 3 and usually performance is not improved with a greater order.

Problems

The most popular items (with the greatest degree of nodes) will influence the results.

3-Paths

3-Paths model

For a given user u we can recommend items with the greatest number of paths of a given order starting at node u.

The smallest order we can take is 3 and usually performance is not improved with a greater order.

Problems

The most popular items (with the greatest degree of nodes) will influence the results.

Possible solution

Instead of using adjacency matrix we can use transition matrix (our process here is a Markov chain).
To receive transition matrix it is enough to divide each row by its sum.

P3 model

P3 model

For a given user u we can recommend items with the greatest probability of transition from u within exactly 3 steps.

For convenience for will consider user and item transition matrices. In our case we will have

$$Pui = \begin{bmatrix} \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 \\ 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix}$$

$$Piu = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Note that for the matrix $P3 = Pui \cdot Piu \cdot Pui$, $P3(u, i)$ contains a probability that a random walk of the order 3 starting from u will finish at i.

P3alpha

In [2] Cooper et al. introduced a simple modification of P3 model. Basically they have taken each entry of a transition matrix to the power *alpha*, where *alpha* is an hiperparameter.

P3alpha model

For a given user u we can recommend items ordered by $P3alpha = (Pui)^{\alpha} \cdot (Piu)^{\alpha} \cdot (Pui)^{\alpha}$, where $(Pui)^{\alpha}$ is a matrix (Pui) taken elementwise to the power α.

Note that:

- $P3alpha$ is no longer a transition matrix,
- with increasing *alpha* we decrease the impact of popular nodes,
- $P3$ is a special case of $P3alpha$ for $alpha = 1$.

RP3beta

In [3] Paudel et al. introduced another simple improvement of P3 (and of P3alpha).

RP3beta model

For a given user u we can recommend items ordered by $P3alpha$ divided by item popularity to the power $beta$.

Note that:

- $RP3beta$ reduces to $P3alpha$ for $\beta = 0$,
- with increasing $beta$ we recommend popular items less often.

Matrix multiplication in case of large datasets is not possible.
Several ways ([2], [3]) introduced effective way of estimating graph based approaches by using random walk sampling.

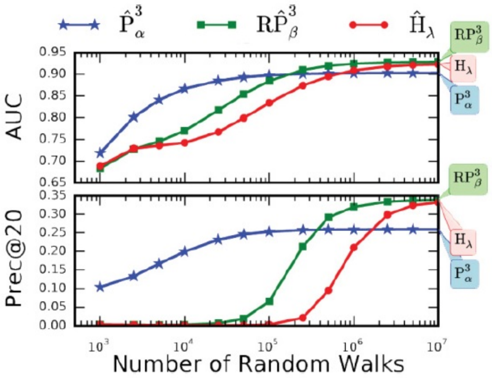


Figure: AUC and Prec@20 for different number of random walks per user generated on MovieLens-M dataset [3]

Python implementation

To do (especially for absent students):

- Go through - *P5. Graph-based* notebook to:
    - go through the implementation of RP3Beta
    - save recommendations of P3 model
    - observe evaluation measures
    - optimize hiperparameters
    - **project task 6: generate recommendations of RP3Beta for hiperparameters found to optimize recall**
    - **project task 7 (optional): implement graph-based model of your choice (for example change length of paths in RP3beta)**
    - observe sample recommendations

Project tasks

- project task 1: implement TopRated
- project task 2: implement self-made BaselineIU
- project task 3: implement some other evaluation measure
- project task 4: use a version of your choice of Surprise KNN algorithm
- project task 5: implement SVD on top baseline (as it is in Surprise library)
- project task 6: generate recommendations of RP3Beta for hiperparameters found to optimize recall
- project task 7 (optional): implement graph-based model of your choice (for example change length of paths in RP3beta)

OLX GROUP

ADAM MICKIEWICZ UNIVERSITY POZNAŃ

References I

[1] C. Aggarwal, "Recommender systems,", Jan. 2016, https://link.springer.com/book/10.1007/978-3-319-29659-3.

[2] C. Cooper, S. H. Lee, T. Radzik, and Y. Siantos, "Random walks in recommender systems: Exact computation and simulations,", WWW '14 Companion, pp. 811–816, 2014.

[3] B. Paudel, F. Christoffel, C. Newell, and A. Bernstein, "Updatable, accurate, diverse, and scalable recommendations for interactive applications," *ACM Transactions on Interactive Intelligent Systems*, vol. 7, pp. 1–34, Dec. 2016.