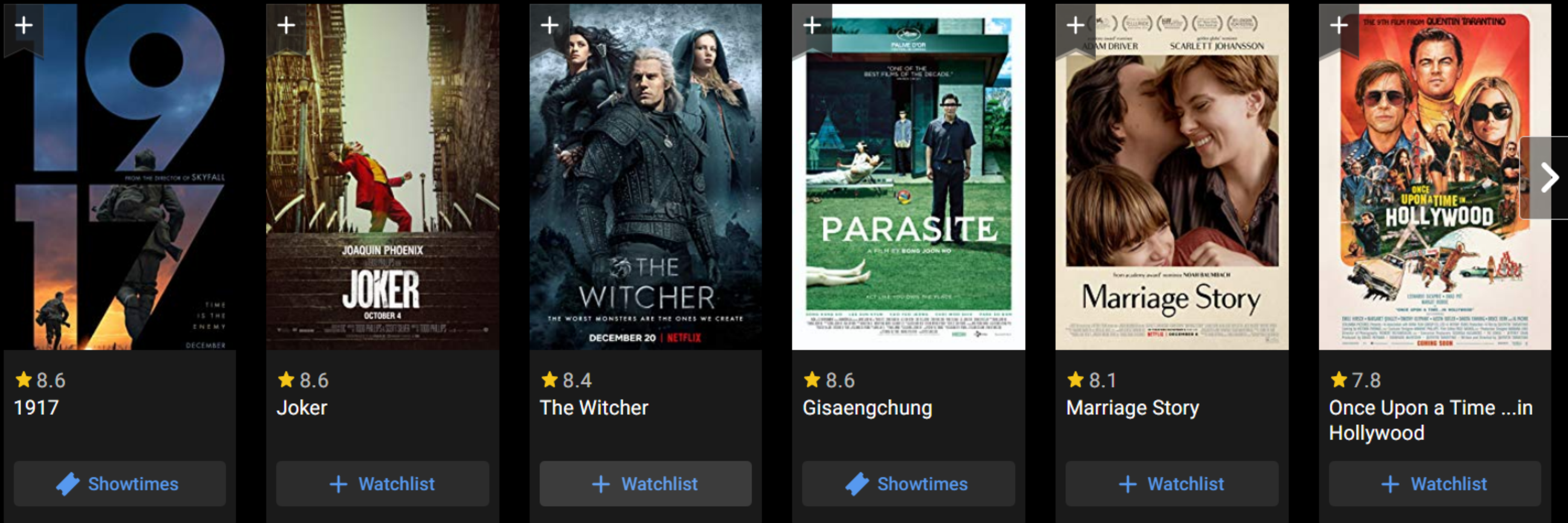


Graph-based Recommender System

Group 33

Yuxuan LONG, Shengzhao LEI

Futong LIU, Mingbo CUI

Dataset Description

Dataset: Movielens-100k[1]

a subset of Movielens movie recommendation system database

100,000 ratings

943 Users

1686 Movies
(18 repeated ones)

Users: age, gender, occupation, zip code and etc.

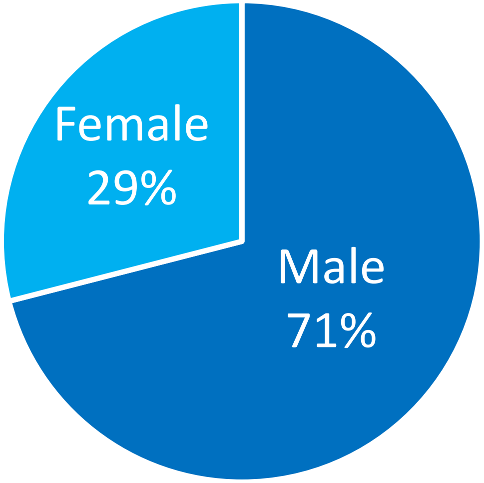
Movies: release date, genre, movie title and etc.

[1]:Harper, F. Maxwell, and Joseph A. Konstan. "The movielens datasets: History and context." Acm transactions on interactive intelligent systems (tiis) 5.4 (2015): 1-19.

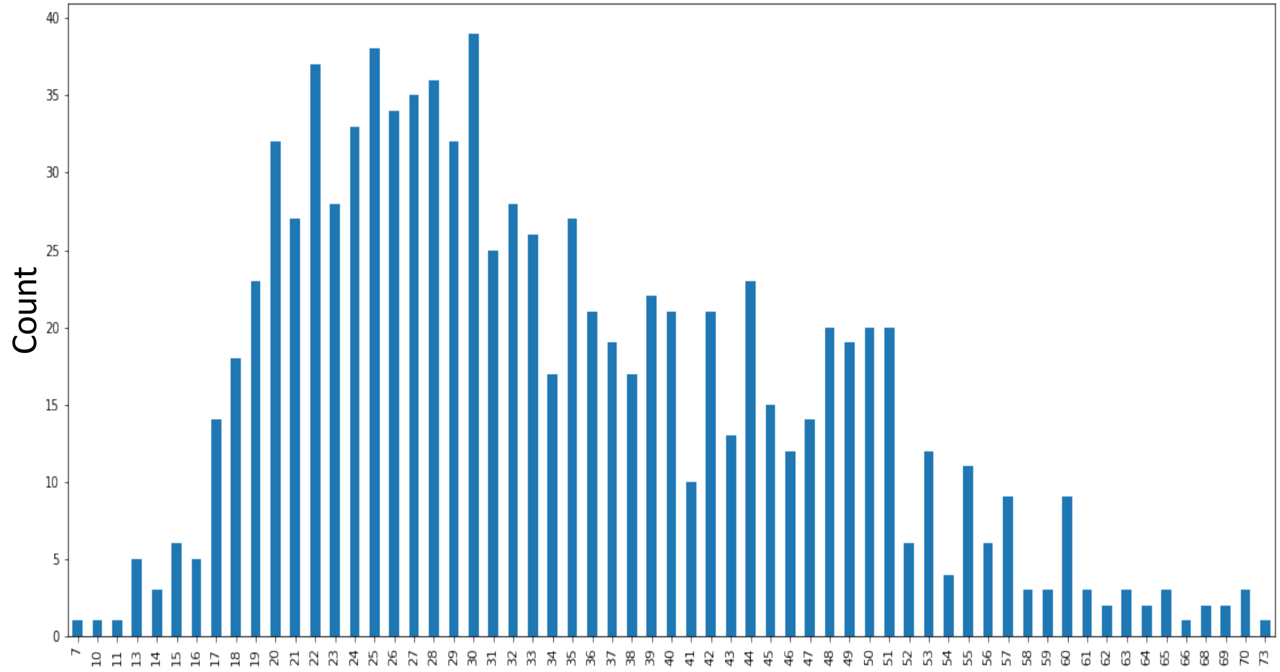
Pipeline


Raw Data
Data visualization
Data preprocessing
Model building and evaluation

Distribution of User Gender and User Age

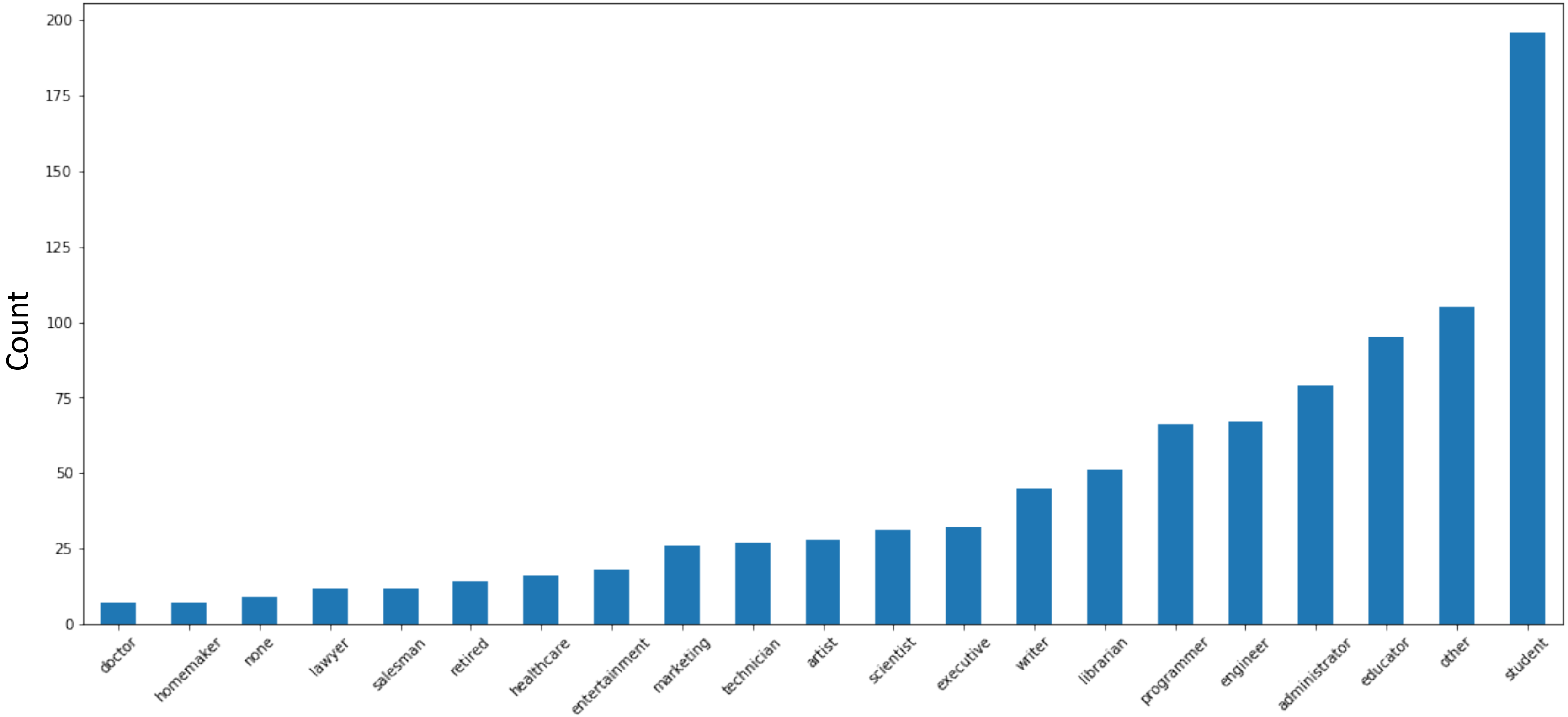


User gender distribution



User age distribution

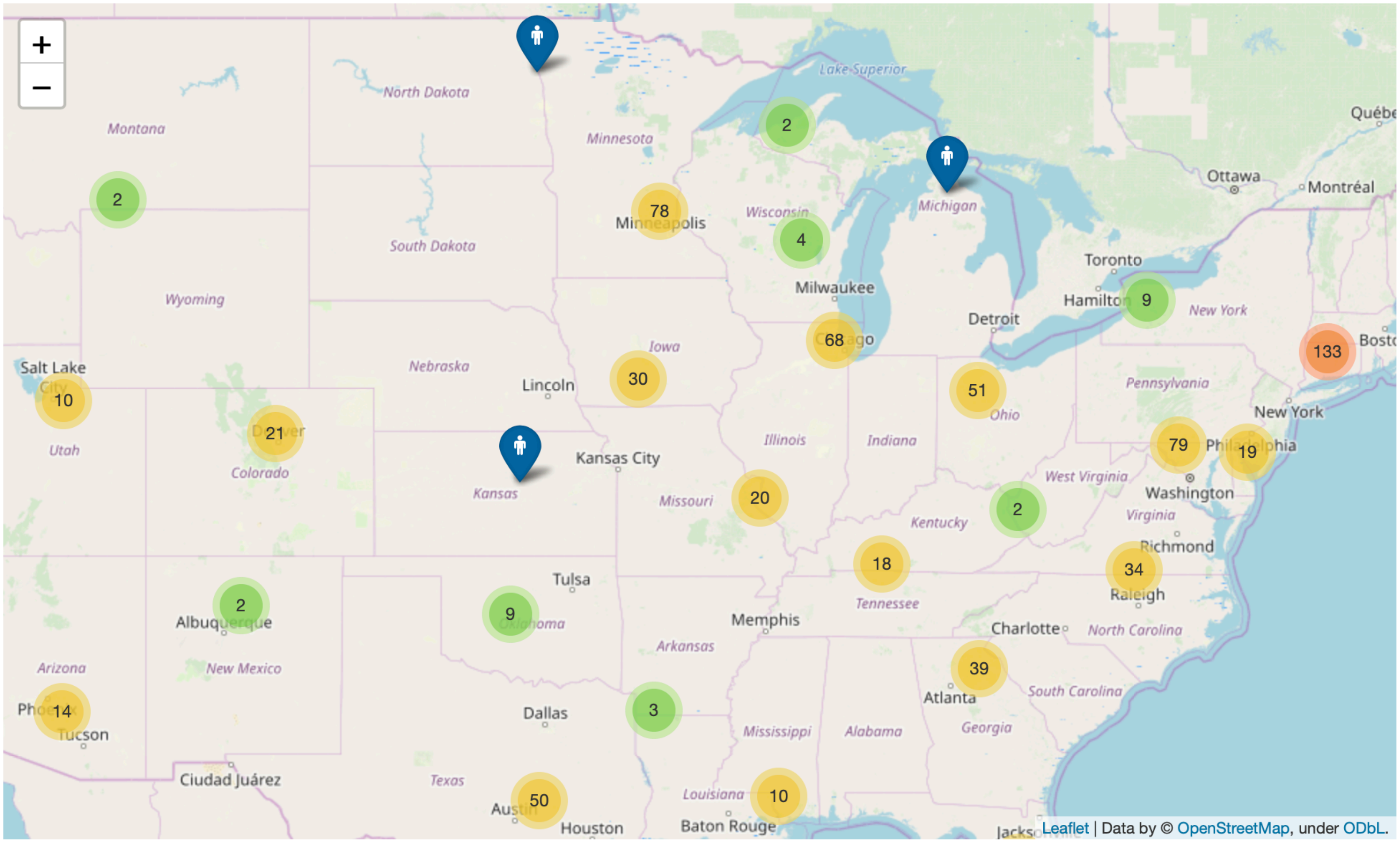
Distribution of User Occupation



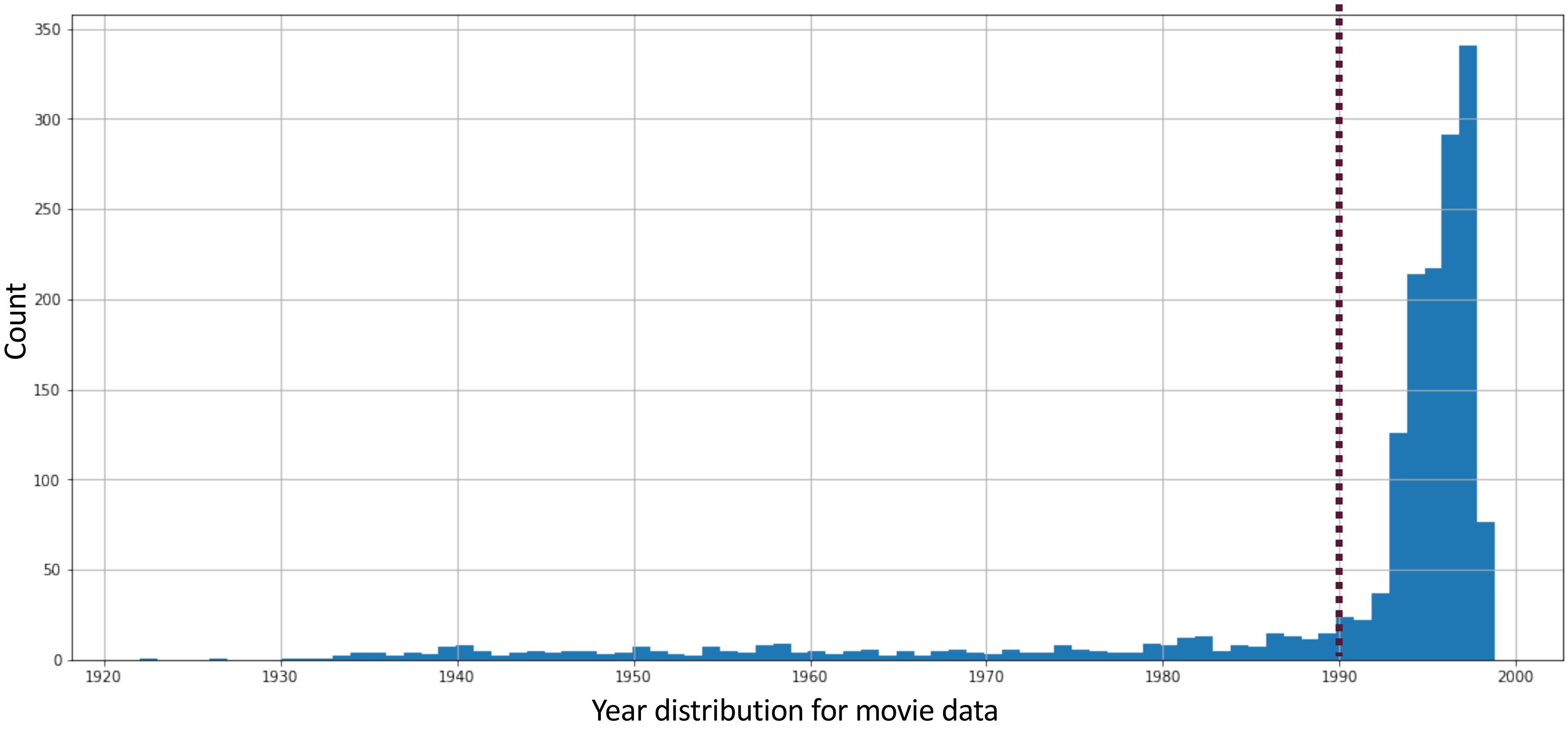
User occupation distribution

Geographical Distribution of Users

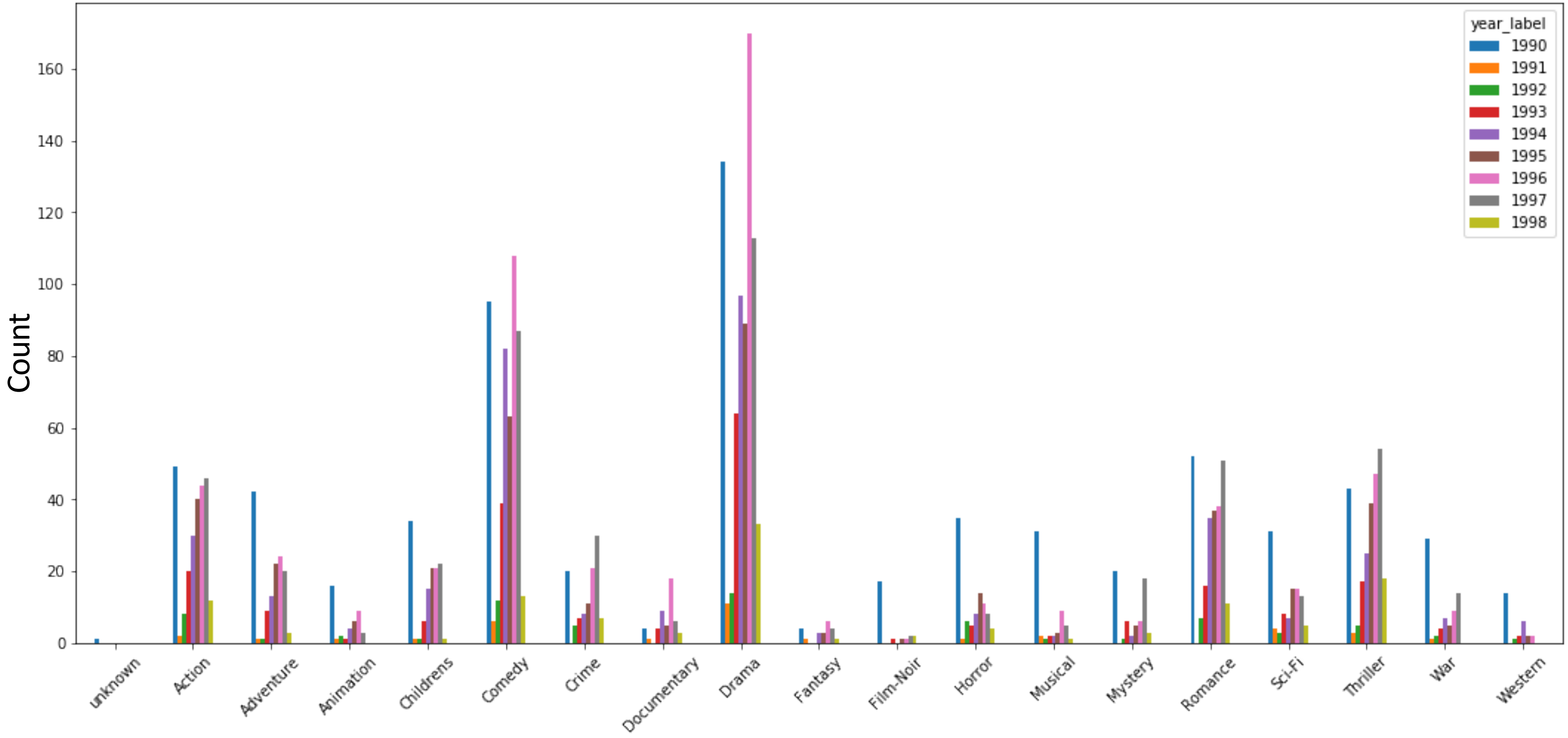
Geographical distribution of user in America



Distribution of Movie Release Years



Distribution of Movie Categories



Genre distribution in different year for movie data

User Graph

Undirected Weighted Graph

Graph Construction

Nodes: Users

Edge Weights:

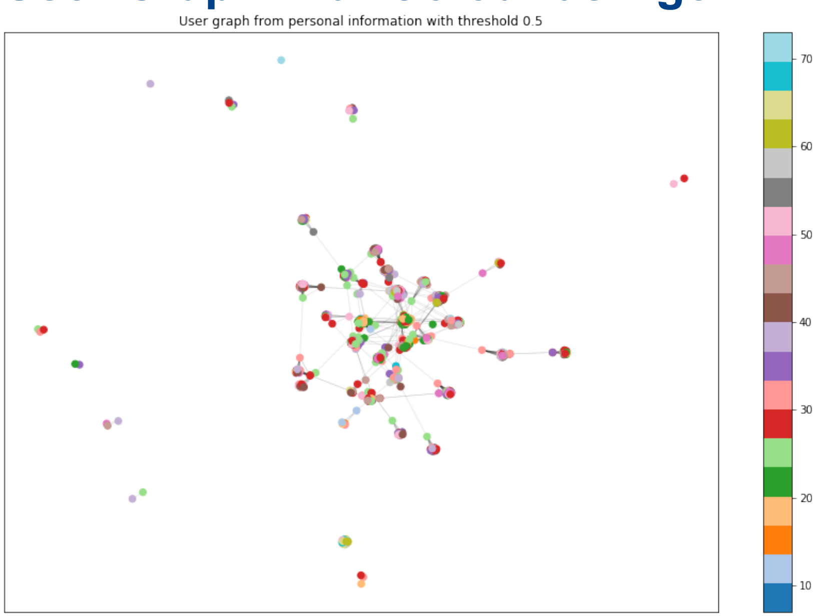
Method 1:

Edge weight as similarity between users' information such as age, gender, occupation and location;

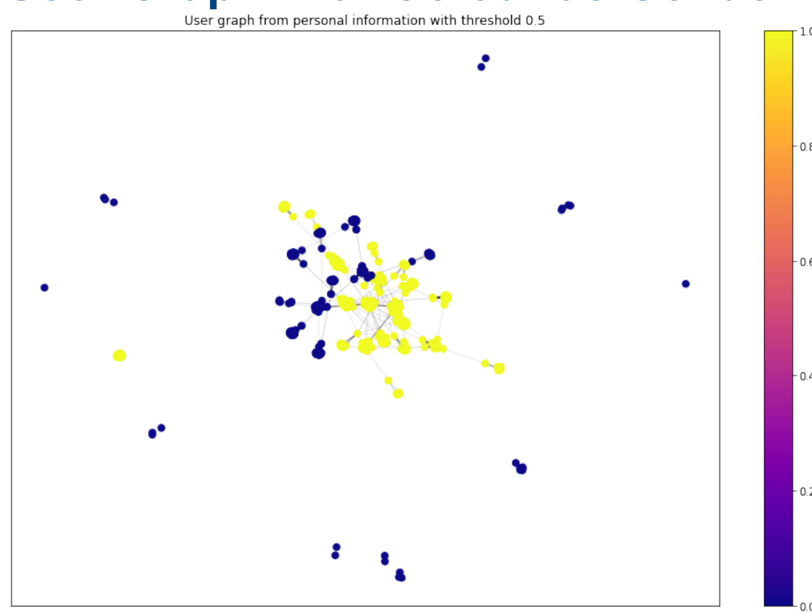
Min-max-normalize the age and calculate the cosine similarity between them.

Gender, occupation and location are one-hot encoded.

User Graph with Colour as Age

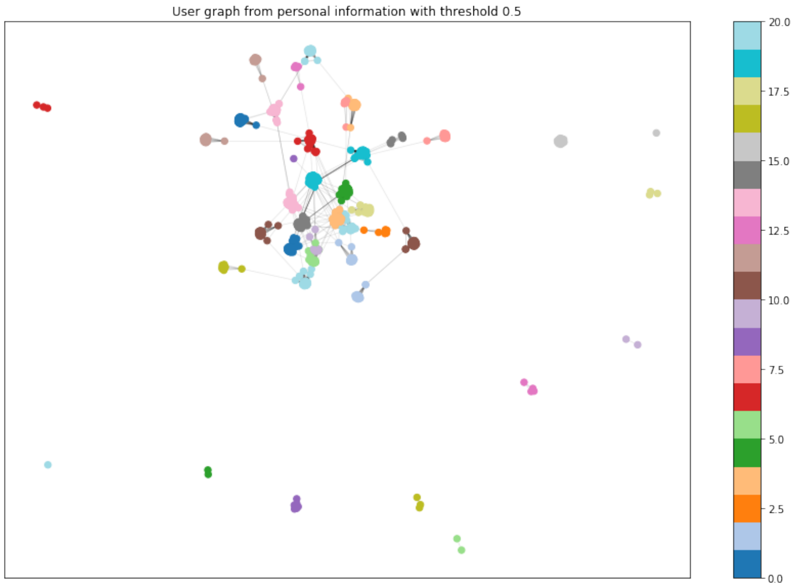


User Graph with Colour as Gender



User Graph

User Graph with Colour as Occupation



Method 2:

Based on Method 1 but with ZCA whitening[2] applied to de-correlate all the features.

There might be correlations between features, for example ”age” and “occupation”.

We wish to decorrelate the data while preserving information as much as possible.

<u>ZCA Whitening</u>: a linear transformation for decorrelating data (identity covariance matrix). It is optimal if the goal is to keep the transformed random vector as similar as possible to the original one. The whitened

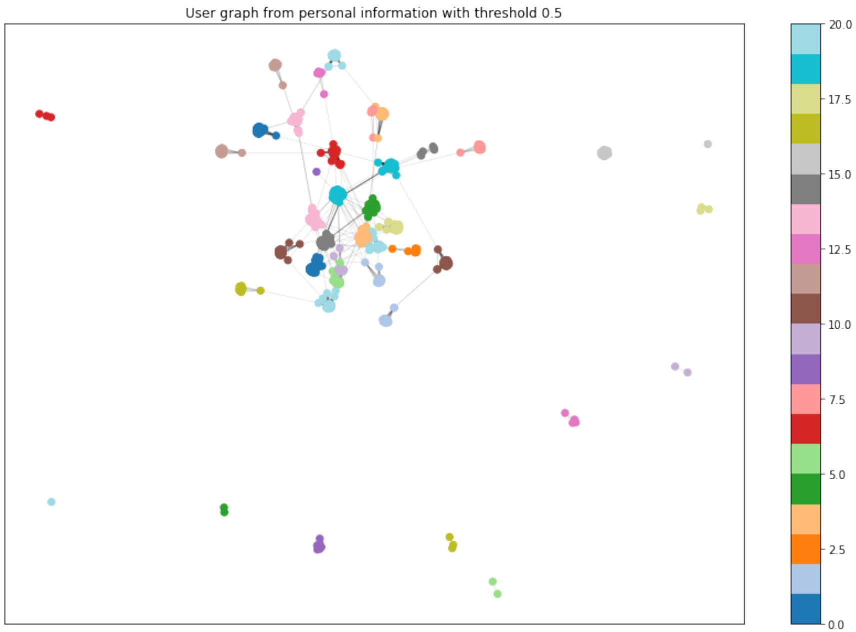
$$\mathbf{z} = \mathbf{W}^{\mathrm{ZCA}}\mathbf{x} = \mathbf{U}\boldsymbol{\Lambda}^{-\frac{1}{2}}\mathbf{U}^T\mathbf{x}$$

[2]. A. Kessy, A. Lewin, and K. Strimmer, “Optimal whitening and decorrelation,”The American Statistician, vol. 72, no. 4, pp.309–314, 2018.

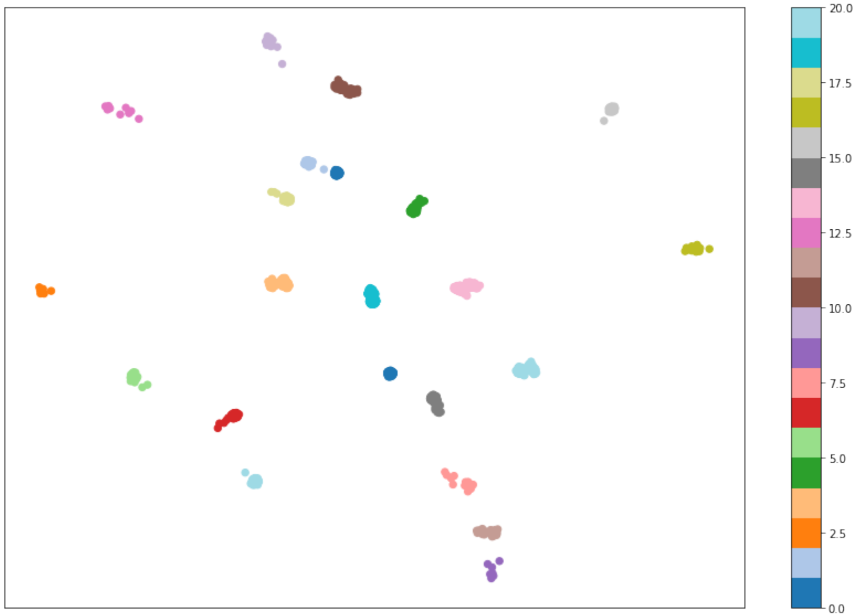
User Graph

Colour as Occupation

User Graph Before Whitening (Method 1)



User Graph After Whitening (Method 2)



After whitening, the clusters (connected components) in the user graph is clearer and more balanced, which might improve the recommender system model later.

User Graph - Comparisons

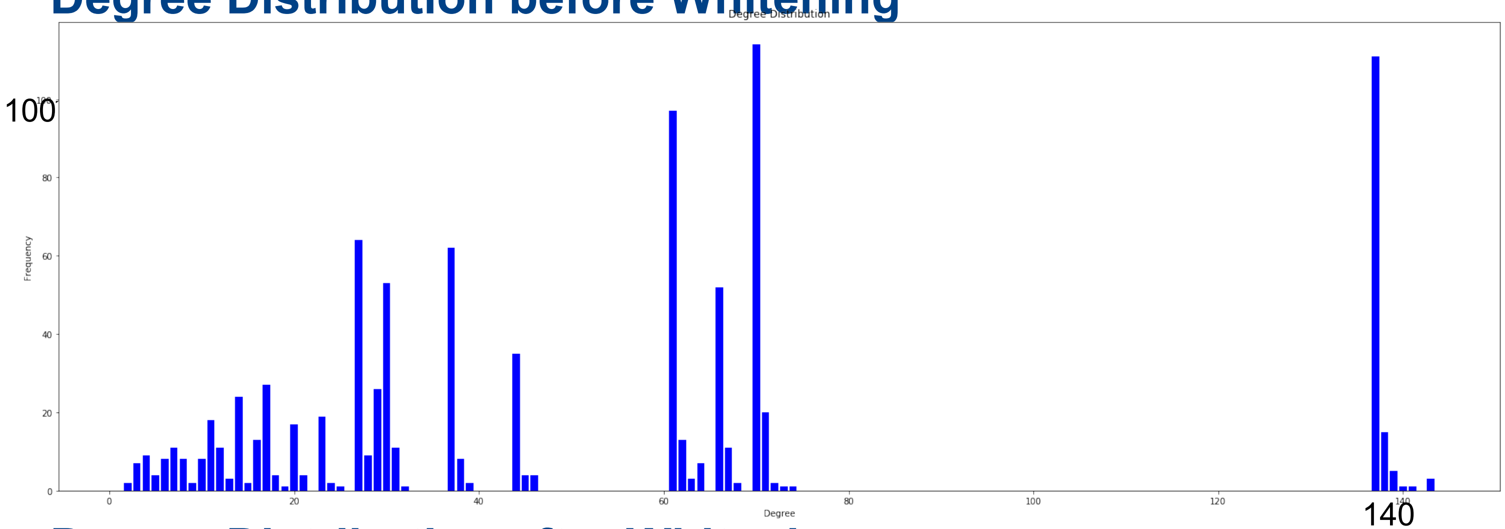
Graph Properties Comparison

	Before Whitening	After Whitening
Number of Nodes	943	943
Number of Edges	26055	26653
Average Degree	55.2598	56.5281
Number of Connected Components	12	21 (Number of occupations: 21)
Ratio of the Largest Connected Component	95.76 %	20.78 %
Sparsity	0.0587	0.0600
Global Clustering Coefficient (Transitivity)	0.9929	0.8559

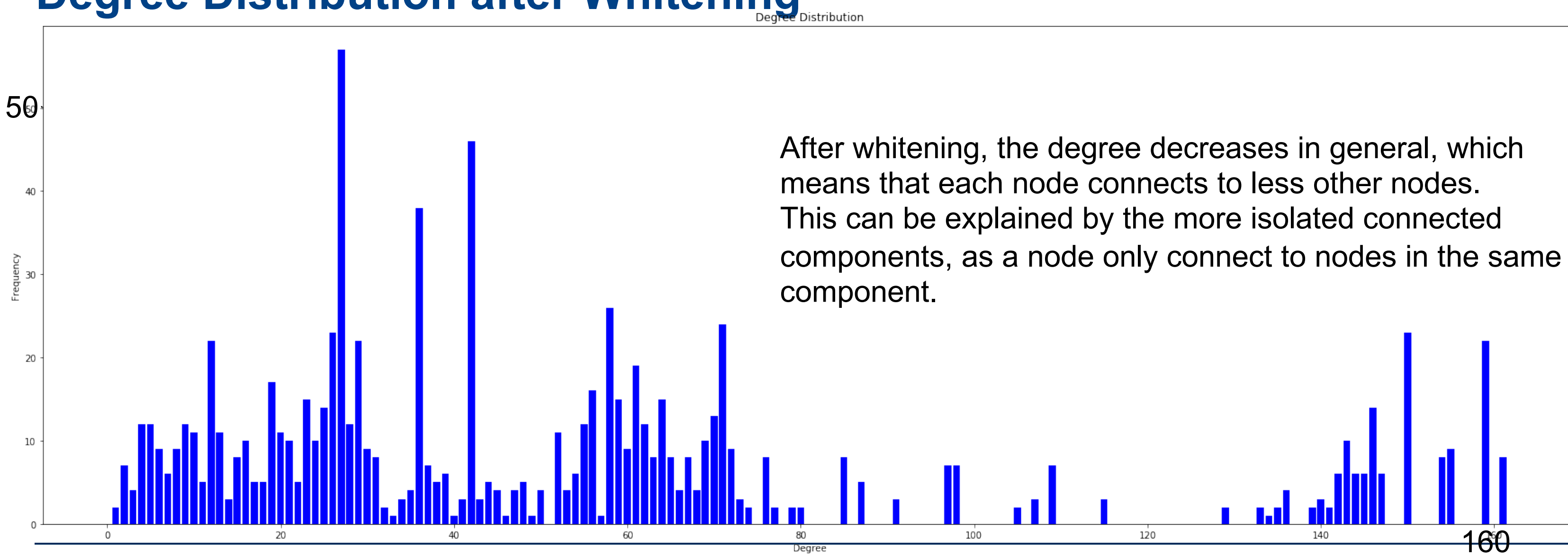
After whitening, the number of connected components increase to **21**, the exact number of occupations. The largest connected component takes less ratio in the entire graph, indicating a more balanced graph. The sparsity increases as we decorrelated the features with whitening.

User Graph - Comparisons

Degree Distribution before Whitening



Degree Distribution after Whitening



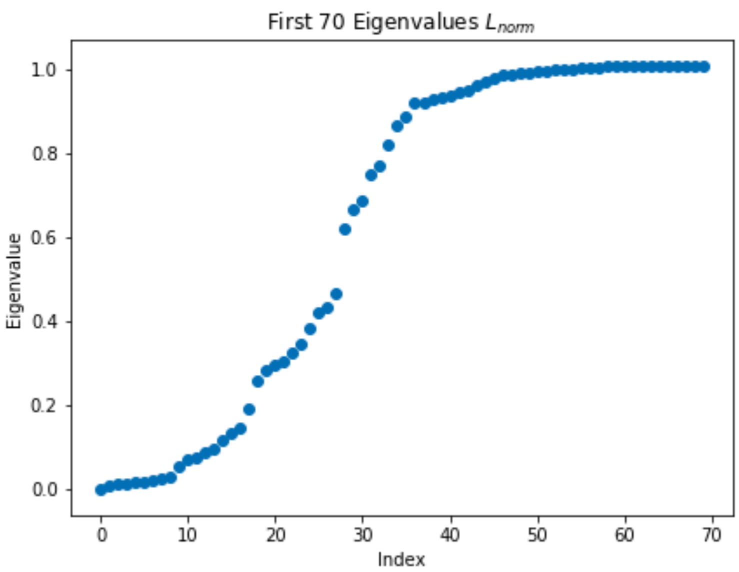
After whitening, the degree decreases in general, which means that each node connects to less other nodes. This can be explained by the more isolated connected components, as a node only connect to nodes in the same component.

User Graph - Comparisons

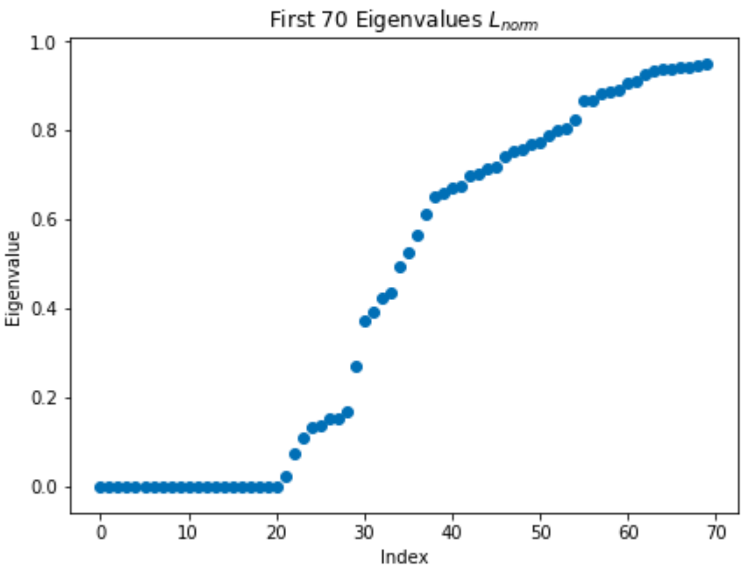
Laplacian Spectrum

If the data has K clear clusters, there will be a sharp gap in the Laplacian spectrum after the K-th eigenvalue.

Before Whitening (First 70 Eigenvalues)



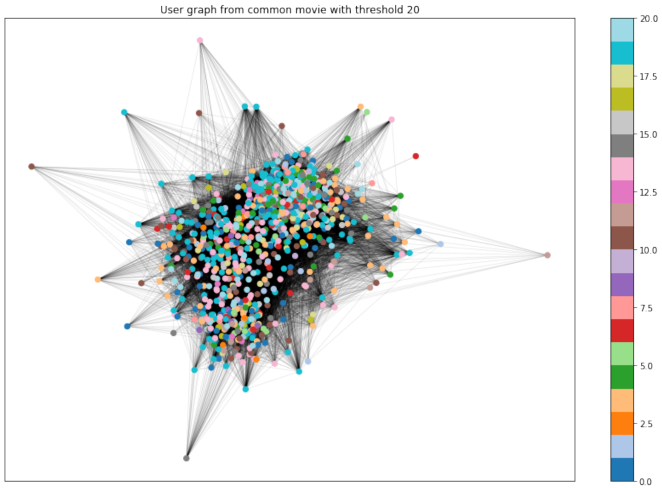
After Whitening (First 70 Eigenvalues)



User Graph – An Alternative

Edge weight as how many common movies two users have rated

Colour as Occupation



Colour as Average Rating

Colour as Age

Movie Graph

Undirected Weighted Graph

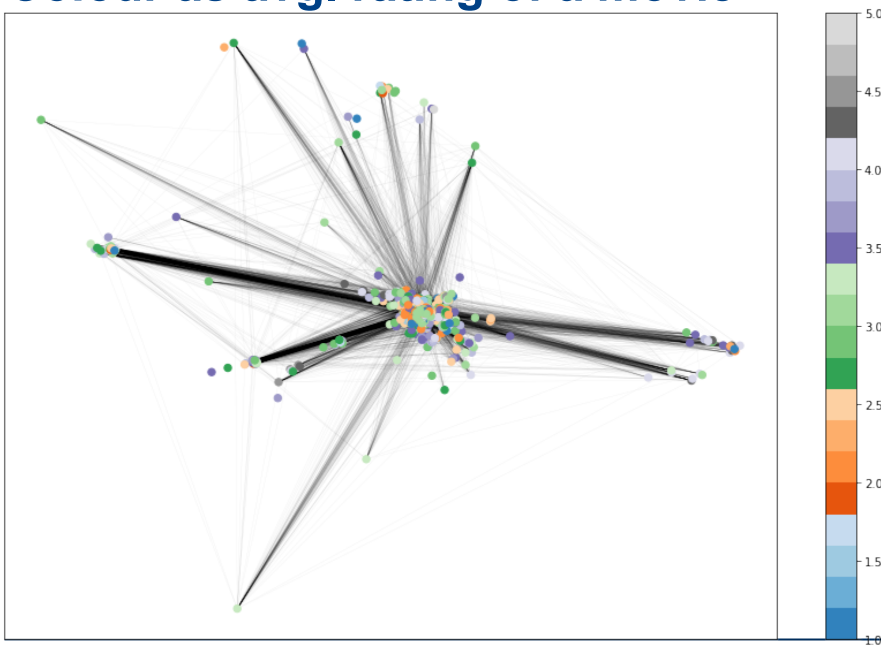
Nodes: Movies

Edges weights:

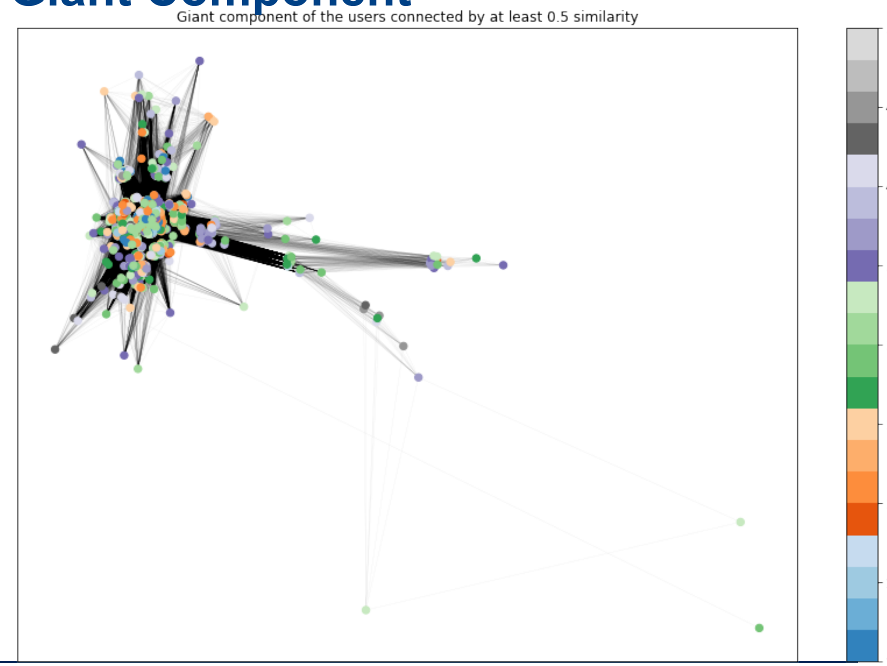
Method 1:

Edge weight as the Euclidean distance between two movies' one-hot encoded genre vectors after whitening.

Colour as avg. rating of a movie



Giant Component



Movie Graph

Undirected Weighted Graph

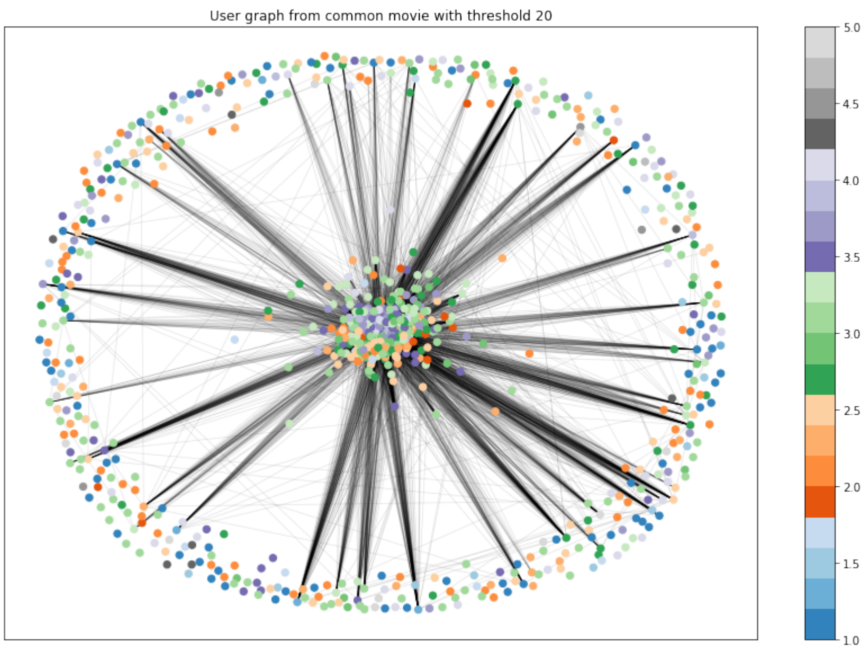
Nodes: Movies

Edges weights:

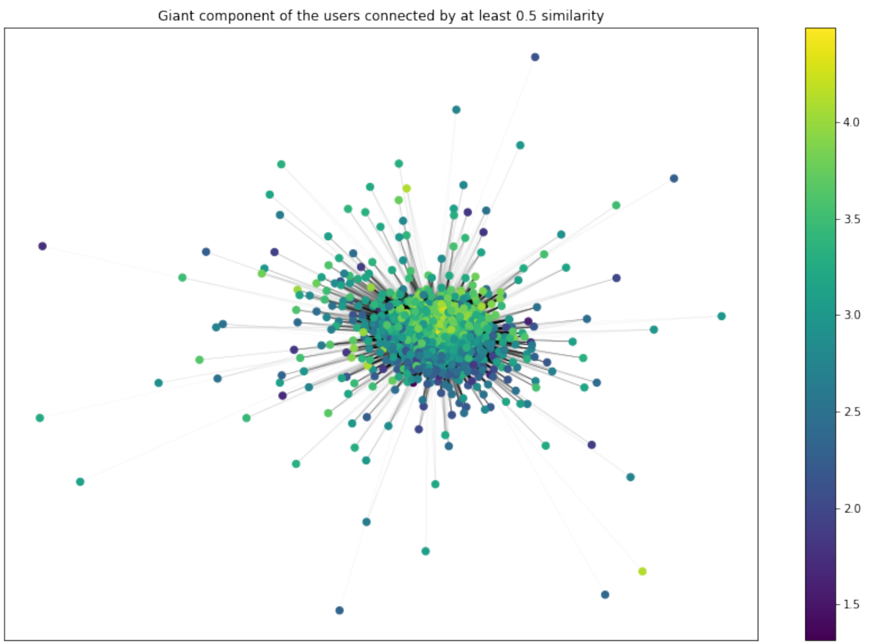
Method 2:

Edge Weight as the number of common users who have rated them.

Colour as avg. rating of a movie



Giant Component



Model

Unweighted Bipartite Graph[3]

For each rating r, we have a bipartite graph $\mathcal{G}_r$. An edge $e = (u_i, v_j)$ in the graph $\mathcal{G}_r$ is connected if the user u_i gives the item v_j a rating of r.

$\boldsymbol{M} \in \mathbb{R}^{N_u \times N_v}$, rating matrix

$\boldsymbol{M}_r \in \mathbb{R}^{N_u \times N_v}$, adjacency matrix for the bipartite graph at rating level r.

For example, two users and three items (i.e. movies), rating $r \in \mathcal{R} = \{1,2,3,4,5\}$.

$$\boldsymbol{M}_1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\boldsymbol{M}_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

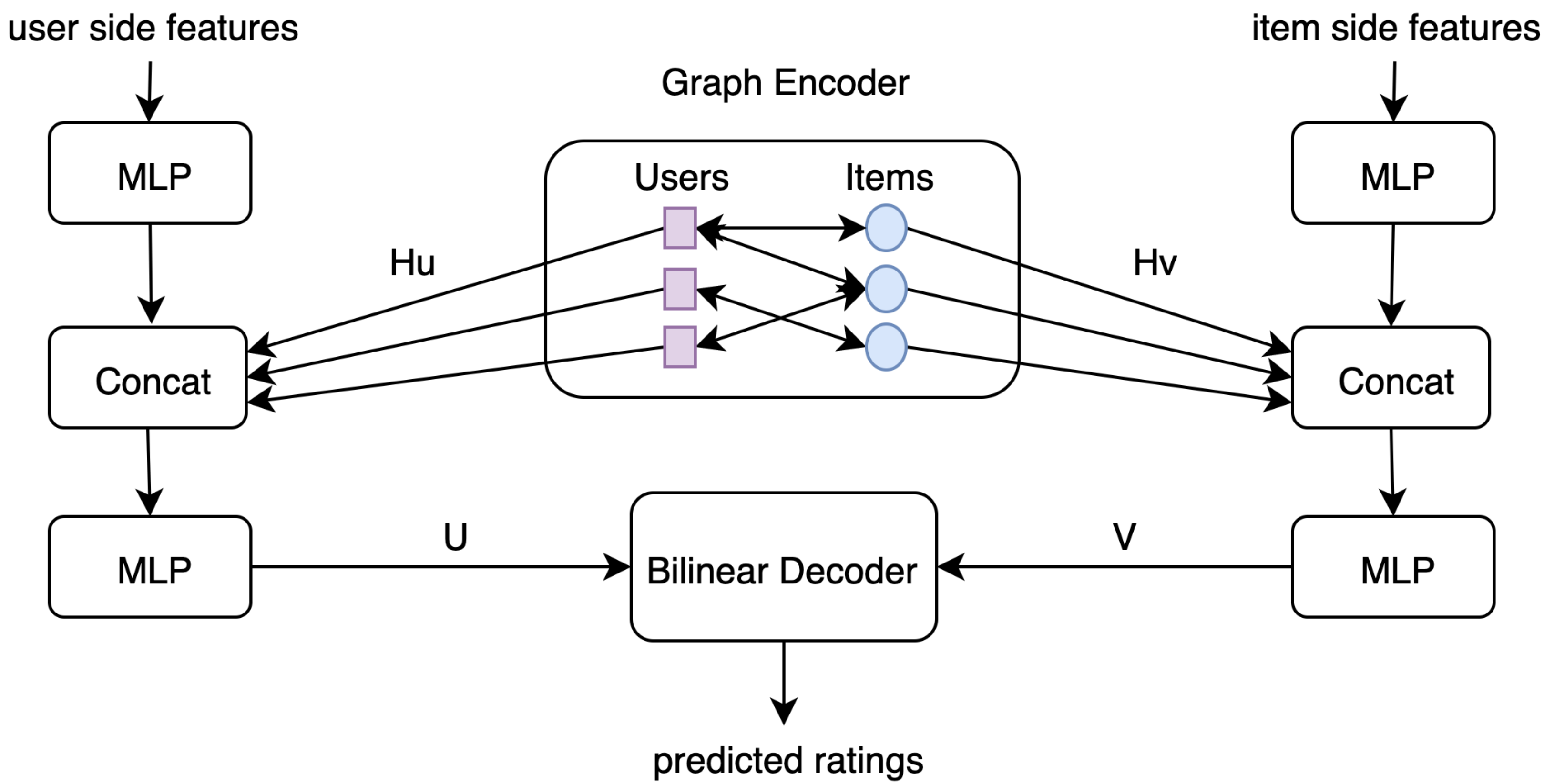
$$\boldsymbol{M} = \begin{bmatrix} 3 & 1 & 5 \\ 4 & 2 & 4 \end{bmatrix} \Rightarrow \boldsymbol{M}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\boldsymbol{M}_4 = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{M}_5 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

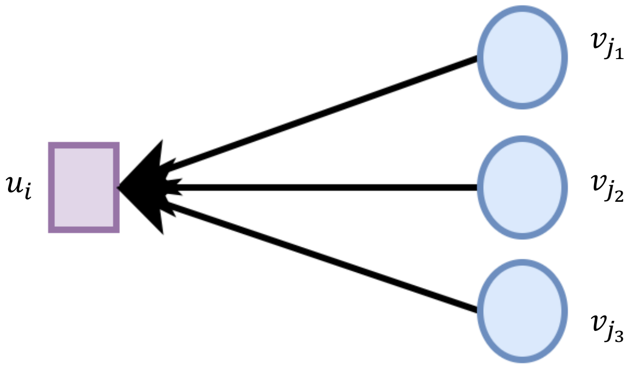
[3]. R. v. d. Berg, T. N. Kipf, and M. Welling, “Graph convolutional matrix completion,” arXiv preprint arXiv:1706.02263, 2017.

Model Architecture



Model

Message passing:



$$\mathbf{m}_{u_i,r} = \frac{1}{|\mathcal{N}_{i,r}|}\sum_{j\in\mathcal{N}_{i,r}} \mathbf{m}_{v_j\rightarrow u_i,r} = \frac{1}{|\mathcal{N}_{i,r}|}((\mathbf{M}_r)_{i,:}\mathbf{X}_v\mathbf{W}_r) \quad (1)$$

where $\boldsymbol{X}_u$ user feature matrix
$\boldsymbol{X}_v$ item feature matrix,
$\boldsymbol{W}_r$ trainable matrix

Message concatenation:

$$(\boldsymbol{H}_u)_{i,:} = concat(\sigma\left(\{\boldsymbol{m}_{u_i,r}\}_{r\in\mathcal{R}}\right), \{(\boldsymbol{X}_u\boldsymbol{W}_r)_{i,:}\}_{r\in\mathcal{R}}) \quad (2)$$

where $\sigma(.)$ is the activation function

Bilinear Decoder

Confidence map:

$$(\mathbf{P}_r)_{i,j} = \frac{\exp\left(\mathbf{u}_i^T \mathbf{Q}_r \mathbf{v}_j\right)}{\sum_{s \in \mathcal{R}} \exp\left(\mathbf{u}_i^T \mathbf{Q}_s \mathbf{v}_j\right)} \tag{3}$$

where $\boldsymbol{u}_i$: embedded user feature
$\boldsymbol{v}_j$: embedded item feature,
$\boldsymbol{Q}_r$: trainable matrix at the rating level r

Predicted ratings:

$$\hat{\mathbf{M}} = \sum_{r \in \mathcal{R}} r \mathbf{P}_r \tag{4}$$

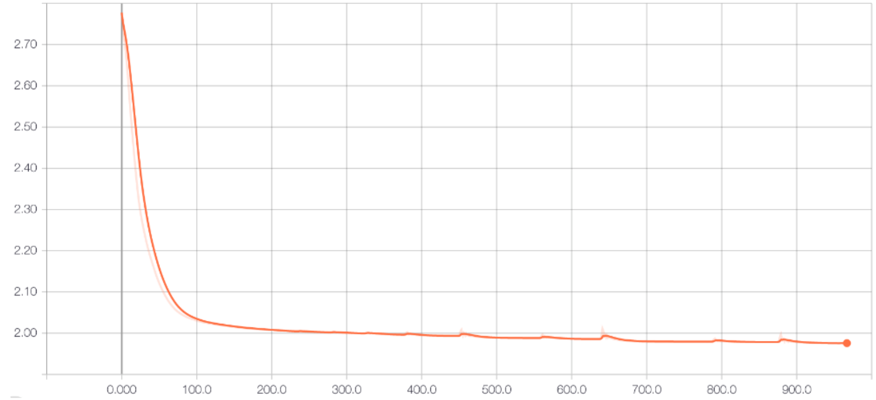
Evaluation metric and Loss function

Evaluation metric:

$$RMSE = \sqrt{\frac{1}{|\Omega|}\sum_{(i,j)\in\Omega}\left((\hat{\mathbf{M}}_r)_{i,j} - (\mathbf{M})_{i,j}\right)^2} \quad (5)$$

Loss function:

$$\mathcal{L} = \lambda \cdot \mathcal{L}_{Laplacian} + (1-\lambda)\cdot(\mathcal{L}_{CE} + \mathcal{L}_{RMSE}) \quad (6)$$



Rapid convergence of loss

Detailed introduction of Laplacian loss

Dirichlet norm[4]:

$$||\hat{\mathbf{M}}||^2_{L_u} = trace(\hat{\mathbf{M}}^T \mathbf{L}_u \hat{\mathbf{M}}) \quad (7)$$

$$||\hat{\mathbf{M}}||^2_{L_v} = trace(\hat{\mathbf{M}} \mathbf{L}_v \hat{\mathbf{M}}^T) \quad (8)$$

Laplacian Loss:

$$\mathcal{L}_{Laplacian} = \frac{1}{N_u^2}||\hat{\mathbf{M}}||^2_{L_u} + \frac{1}{N_v^2}||\hat{\mathbf{M}}||^2_{L_v} \quad (9)$$

Reasoning:

- Consider one column vector c of $\hat{\mathrm{M}}$;
- $C^T L_u C$ will be small if our prediction matrix $\hat{\mathrm{M}}$ aligns with the user graph;
- Same reasoning for row vector of $\hat{\mathrm{M}}$.

[4] F. Monti, M. Bronstein, and X. Bresson, "Geometric matrix completion with recurrent multi-graph neural networks," in Advances in Neural Information Processing Systems, 2017, pp. 3697–3707.

Ablation study

Ablation			RMSE
$\neg S$	D	L	0.942
$\neg S$	$\neg D$	L	0.941
$\neg S$	D	$\neg L$	0.948
$\neg S$	$\neg D$	$\neg L$	0.948
S	$\neg D$	L	0.947
S	$\neg D$	$\neg L$	0.945
S	D	$\neg L$	0.957
S	D	L	**0.938**

Table I. Ablation study: S refers **Using side feature**, D refers to **Data whitening** and L refers to **Laplacian loss,** ¬ refers to NOT used

Comparison with Baseline models

Model	RMSE
IMC[1]	1.653
GMC[2]	0.996
MC[3]	0.973
GRALS[4]	0.945
OUR MODEL	**0.938**
sRGCNN[5]	0.929
GC-MC[6]	0.905

Table II. Comparisons with other published models for the MovieLens-100K task on a 80/20 training/test set split.

Summary

Limitations of our recommender system:

- Grid search for hyperparameters will be time consuming;
- No experiment on other dataset;
- Some information in the dataset are discard, e.g. the URL and location information, but no one else has used these information in published paper.

Highlights:

- Construct the recommender system from the perspective of graph;
- Incorporate the Laplacian loss into loss function which improves the model performance


THE END