Graph-based Hybrid Recommendation System

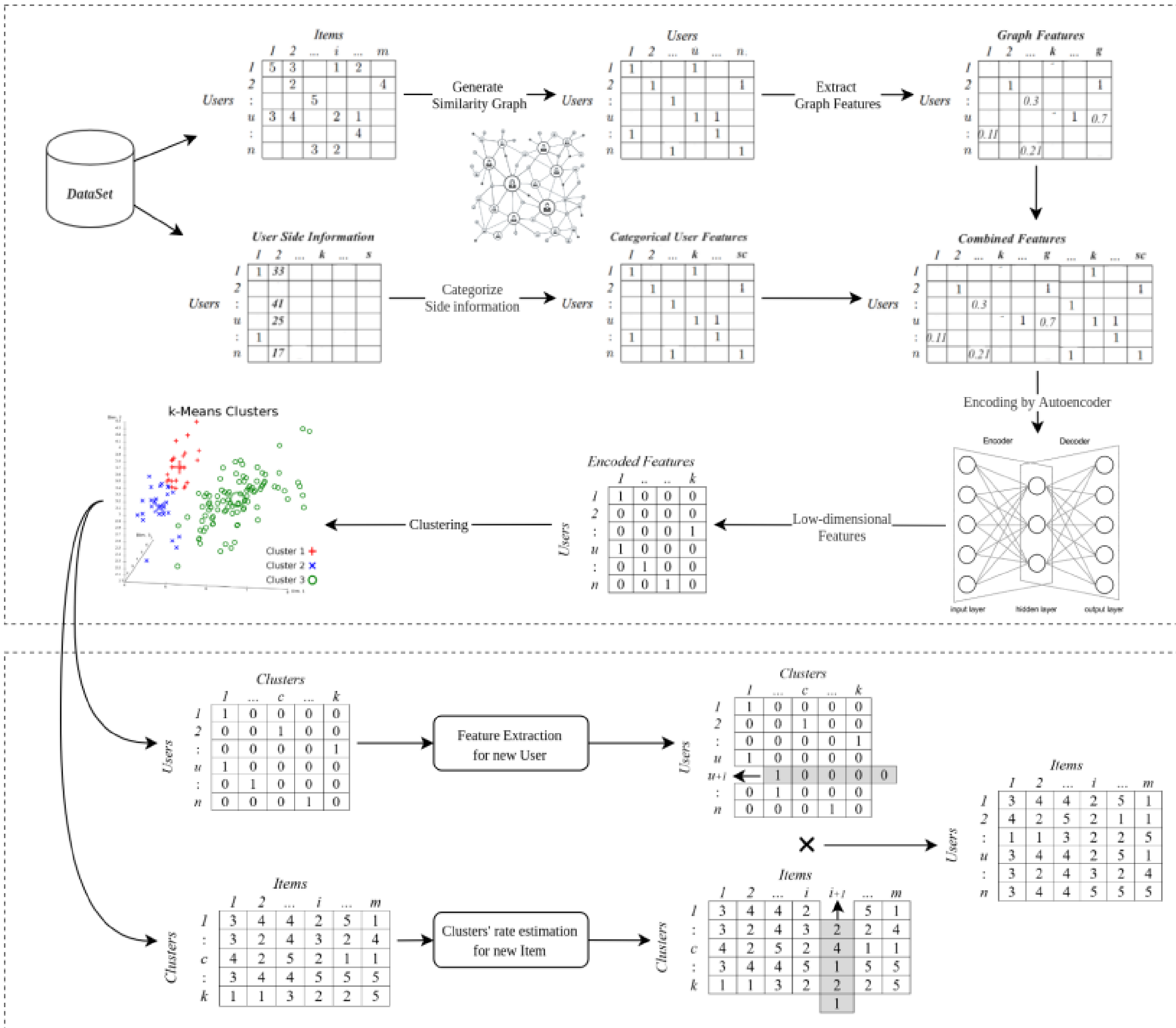

Figure 3: The framework of the proposed recommendation system. The method encodes the combined features with autoencoder and creates the model by clustering the users using the encoded features (upper part). At last, a preference-based ranking model is used to retrieve the predicted movie rank for the target user (lower part)

Algorithm 1 Proposed method detailed workflow

Input: U, I, R, F_u, F_i
Output: Estimated rates for user-item

Set alpha = percentage of items with similar ratings between two users
Compute the aggregated similarity between users base of α (the percentage of items which two users rated them similarly)
Construct the Similarity Graph and consider the users as nodes
F_g: Extracted graph-based features for users (nodes)
F_s: Preprocessed and categorized users' side information (demographic informations)
Combine F_g and F_s in a single feature vector F_t. Apply the Autoencoder on F_t and train the model with the best settings
Encode the F_t using the Autoencoder an extract the low dimensional feature vector F_e
Find the optimum clusters for clustering users with F_e
Perform user clustering using extracted features vector F_e and find clusters C
Generate the user-cluster matrix UC
Estimate clusters' ratings for items matrix CI:
if there are users rated the item i before in the cluster c **then**
    CI_{ci} = average (users' rates of the item i in the cluster c)
else if there are Similar Items to the item i, rated by users in the cluster c **then**
    CI_{ci} = average (users' rates of Similar Items in the cluster c)
else
    CI_{ci} = average (all users' rates in the cluster c)
end if
Estimate users' ratings' matrix $R' = UC \times CI$
Compute the recommendation list for target user u

Graph-based Features

- **Page Rank:** Measures the transitive influence or connectivity of nodes. Computed by iteratively distributing one node's rank (based on the degree) over the neighbors.
- **Degree Centrality:** Measures the number of incoming and outgoing relationships from a node. The Degree Centrality algorithm can be used to find the popularity of individual nodes.
- **Closeness Centrality:** A way to detect nodes that can spread information efficiently through a graph.
- **Betweenness Centrality:** A factor which we use to detect the amount of influence a node has over the flow of information in a graph.

$$C_B(u) = \frac{\sigma(s,t|u)}{\sum_{s,t\in V}\sigma(s,t)} \quad (2)$$

where V is the set of nodes, $\sigma(s,t)$ is the number of shortest path between (s,t), and $\sigma(s,t|u)$ is the number of those paths passing through some node u other than s and t.

- **Load Centrality:** The fraction of all shortest paths that pass through that node.
- **Average Neighbor Degree:** Returns the average degree of the neighborhood of each node.

$$AND(u) = \frac{1}{N(u)} \sum_{v\in N(u)} k_v \quad (3)$$

where $N(u)$ are the neighbors of node u and k_v is the degree of node v which belongs to $N(u)$. For weighted

Experiments

Table 4
Comparison with other methods

Model	MovieLens 100K	MovieLens 1M
Collaborative Topic Regression (Wang and Blei, 2011)	-	0.896
Collaborative Deep Learning (Wang et al., 2015)	-	0.887
Convolutional Matrix Factorization (Kim et al., 2016)	-	0.853
Convolutional Matrix Factorization+ (Kim et al., 2016)	-	0.854
Robust Convolutional Matrix Factorization (Kim et al., 2017)	-	0.847
RippleNet (Wang et al., 2018)	-	0.863
Imputed Singular Value Decomposition (Yuan et al., 2019)	-	0.85
Genetic Algorithm and Gravitational Emulation (Mohammadpour et al., 2019)	-	1.087
Noise Correction Based RS (Bag et al., 2019)	-	1.7
DST-HRS (Khan et al., 2020)	-	0.846
Autoencoder COFILS (Barbieri et al., 2017)	**0.885**	0.838
Baseline COFILS (Barbieri et al., 2017)	0.892	0.848
Kernel PCA COFILS (Barbieri et al., 2017)	0.898	-
Slope One (Lemire and Maclachlan, 2005)	0.937	0.9
Regularized SVD (Paterek, 2007)	0.989	0.96
Improved Regularized SVD (Paterek, 2007)	0.954	0.907
SVD++ (Koren, 2008)	0.903	0.856
Non-Negative Matrix Factorization (Lee and Seung, 2001)	0.944	0.912
Bayesian Probabilistic Matrix Factorization (Salakhutdinov and Mnih, 2008)	0.901	0.84
RBM-CF (Salakhutdinov et al., 2007)	0.936	0.872
AutoRec (Sedhain et al., 2015)	0.887	0.844
Mean Field (Langseth and Nielsen, 2015)	0.903	0.856
GHRS (Proposed Method)	0.887	**0.833**

Cold-start performance

By removing the rating information for a random percentage of users, we can assess the model's predictions using only the user's side information.

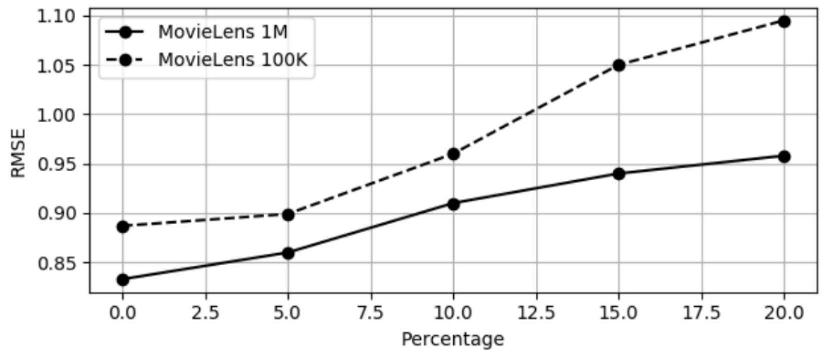

Figure 13: GHRS method RMSE result versus the percentage of users which have been randomly removed from the user-item rating matrix.