Graph Convolutional Matrix Completion

Rianne van den Berg Thomas N. Kipf

Max Welling

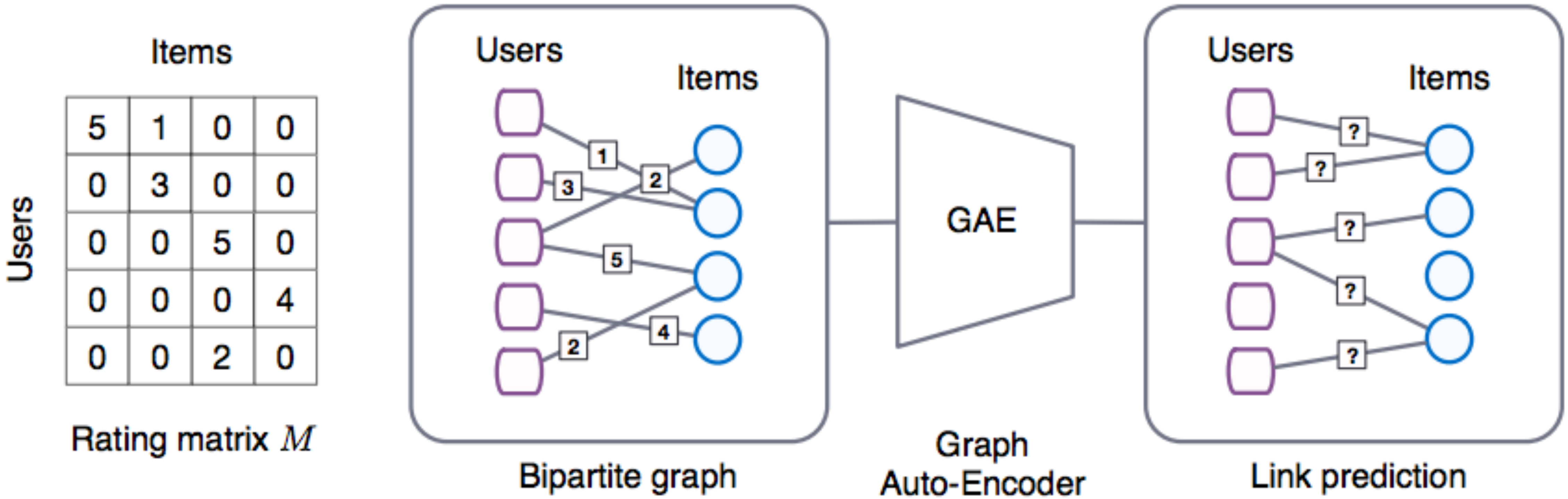
Presenter: Yevgeny Tkach

2019 Spring @
https://qdata.github.io/deep2Read/

Executive Summary

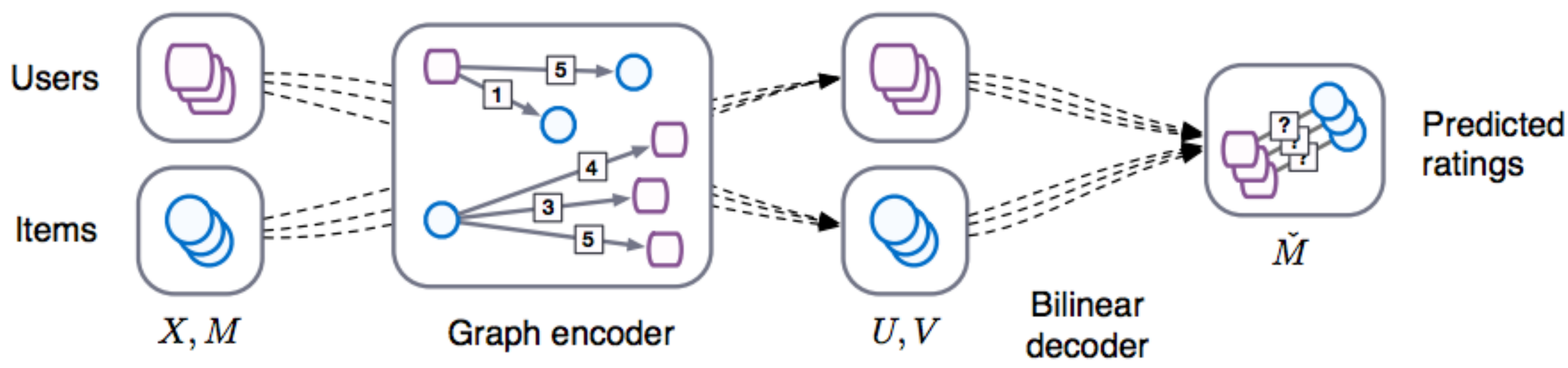
- Matrix completion for recommender systems is reduced to link prediction problem on bipartite user-item graphs with an auto-encoder framework
- This formulation naturally incorporates auxiliary item/user data (node features) in the form of graphs
- Each rating is associated with a different edge type and the predicted score is the <u>expected</u> edge type
- Authors present novel weight sharing strategy for different edge type prediction
- Experiment show competitive performance on standard collaborative filtering (CF) benchmarks and state-of-the-art on CF with user/item graphs

GCMC



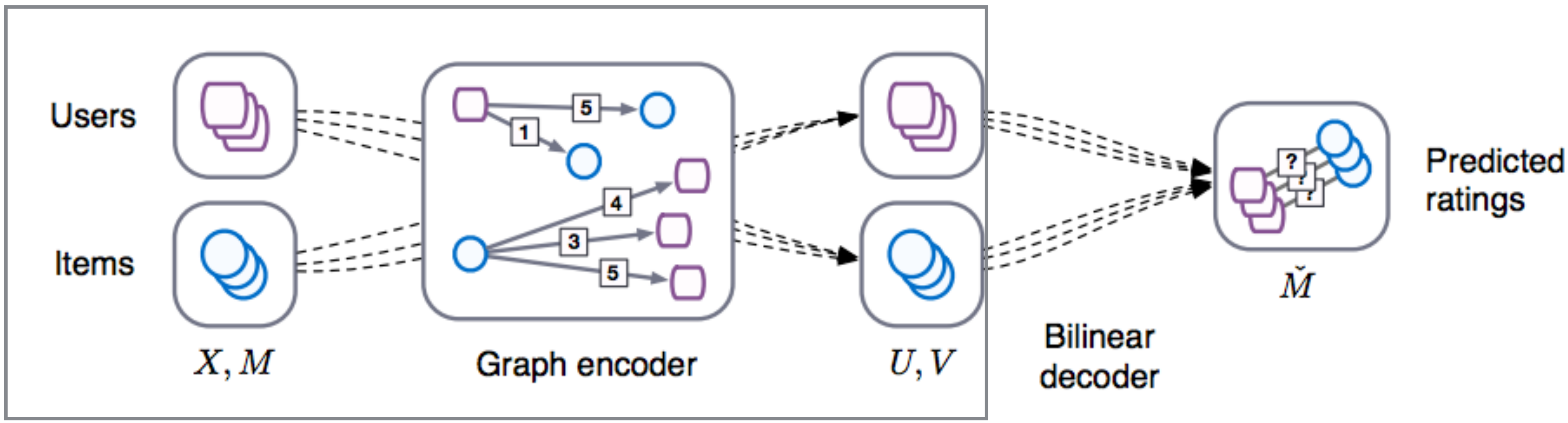
- Rating matrix *M* of shape $N_u \times N_v$, N_u - # of users, N_v - # of items
- a bipartite *G=(W,E,R)* is defined from the matrix such that:
  - *U*∪ *V* = *W* where *U, V* - are the user and item nodes, respectively.
  - Edges $(u_i, r, v_j) \in E$ where *r* are ordinal rating levels, $(u_i, r, v_j) \in E$
  - *R* adjacency matrices are defined $M_1, \ldots, M_R$ where $M_r \in \{0,1\}^{N_u \times N_v}$

Graph Auto-Encoder



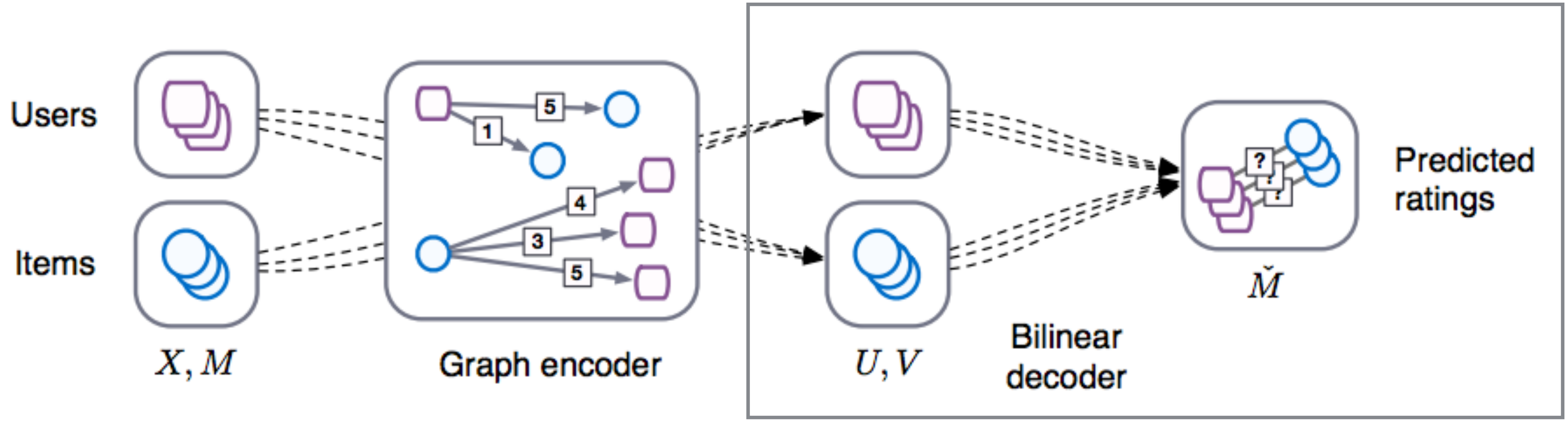
- Here *X* is a feature matrix of shape *N×D*
- Encoder produces $[U, V] = f(X, M_1, \ldots, M_R)$ where U and V are metrics of user and item embeddings respectively of shape $N_u \times E$ and $N_v \times E$
- The decoder tries to reconstruct the rating matrix $\check{M} = g(U, V)$ based on the per of user and item embeddings
- $\check{M}$ is calculated by the expectation over the rank specific adjacency matrices

Graph Convolutional Encoder



- User embeddings:
  - edge type specific message from item j to user i $\mu_{j\to i,r} = \frac{1}{c_{ij}} W_r x_j$
  - c_{ij} normalization factor $|\mathcal{N}_i|$ or $\sqrt{|\mathcal{N}_i||\mathcal{N}_j|}$
  - Weight sharing setup: $W_r = \sum_{s=1}^{r} T_s$
  - Graph Convolution layer: $h_i = \sigma\left[\mathrm{accum}\left(\sum_{j\in\mathcal{N}_{i,1}} \mu_{j\to i,1}, \ldots, \sum_{j\in\mathcal{N}_{i,R}} \mu_{j\to i,R}\right)\right]$
  - Dense layer: $u_i = \sigma(W h_i)$
  - If user info available x_i^f , then: $u_i = \sigma(W h_i + W_2^f f_i) \quad \text{with} \quad f_i = \sigma(W_1^f x_i^f + b)$

Bilinear decoder



- rank prediction:
  - $p(\check{M}_{ij} = r) = \frac{e^{u_i^T Q_r v_j}}{\sum_{s \in R} e^{u_i^T Q_s v_j}}$ where:
  - $Q_r = \sum_{s=1}^{n_b} a_{rs} P_s$ is a trainable parameter matrix

- Loss function: Negative log likelihood

$$\mathcal{L} = - \sum_{i,j;\boldsymbol{\Omega}_{ij}=1} \sum_{r=1}^{R} I[r = M_{ij}] \log p(\check{M}_{ij} = r)$$

Experiments

- **MovieLens 100K -** user-item rating matrix that's accompanied by user and item side information.
- **MovieLens 1M and 10M -** Same as the 100K dataset with more items, users and possible ratings, and without side information. This is the current standard for comparing matrix completion algorithms.
- **Flixster, Douban and YahooMusic -** These datasets include user/item side information in the form of graphs (i.e. connecting user-user and item-item)**.** For GC-MC side information is represented as the relevant row from the side-graph adjacency matrix normalized by the degree.

Dataset	Users	Items	Features	Ratings	Density	Rating levels
Flixster	3,000	3,000	Users/Items	26,173	0.0029	0.5, 1, ..., 5
Douban	3,000	3,000	Users	136,891	0.0152	1, 2, ..., 5
YahooMusic	3,000	3,000	Items	5,335	0.0006	1, 2, ..., 100
MovieLens 100K (ML-100K)	943	1,682	Users/Items	100,000	0.0630	1, 2, ..., 5
MovieLens 1M (ML-1M)	6,040	3,706	—	1,000,209	0.0447	1, 2, ..., 5
MovieLens 10M (ML-10M)	69,878	10,677	—	10,000,054	0.0134	0.5, 1, ..., 5

Table 1: Number of users, items and ratings for each of the MovieLens datasets used in our experiments. We further indicate rating density and rating levels.

Experiment results

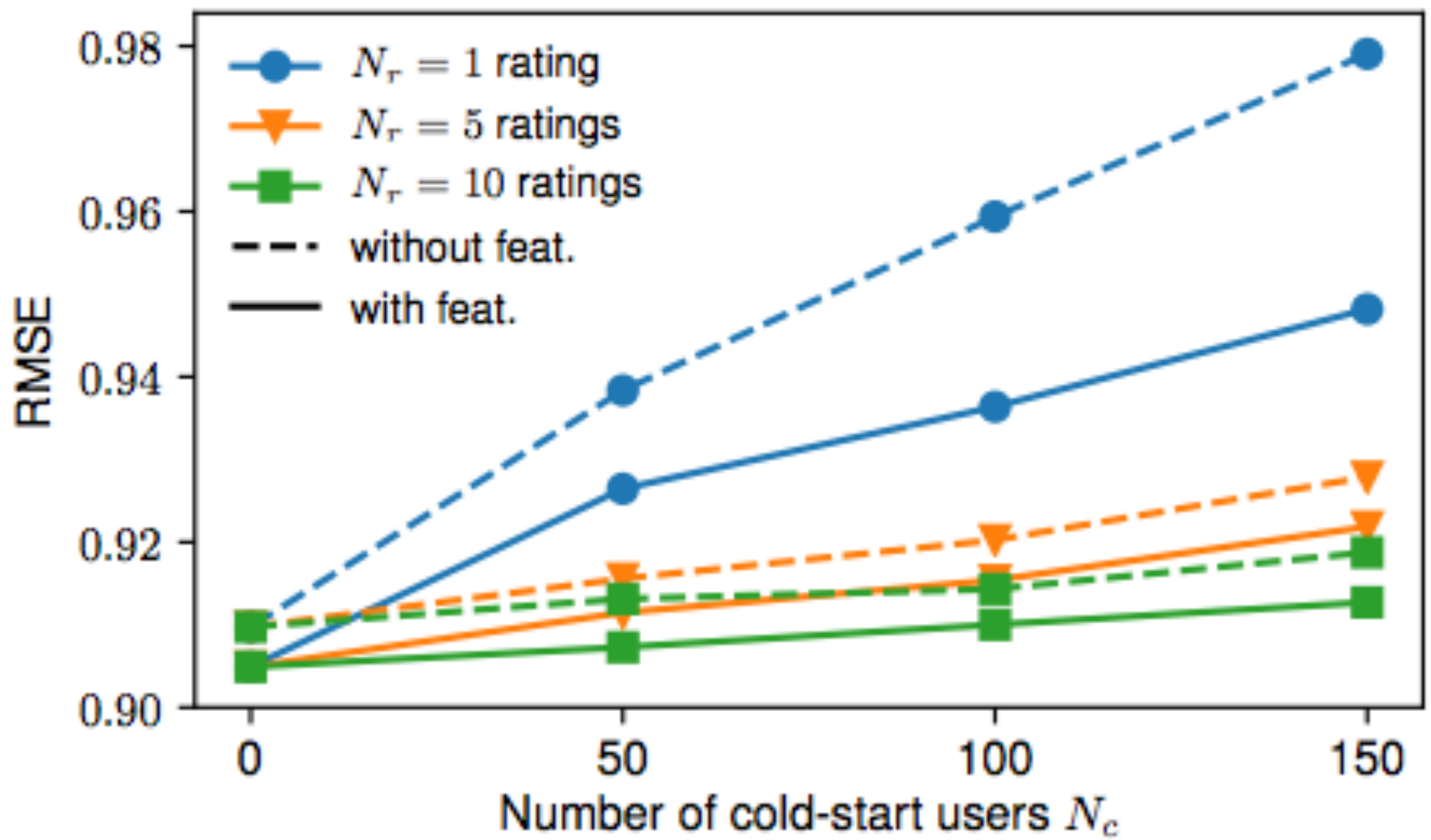
Model	ML-100K + Feat
MC [3]	0.973
IMC [11, 31]	1.653
GMC [12]	0.996
GRALS [25]	0.945
sRGCNN [22]	0.929
GC-MC (Ours)	0.910
GC-MC+Feat	**0.905**

Model	ML-1M	ML-10M
PMF [20]	0.883	–
I-RBM [26]	0.854	0.825
BiasMF [16]	0.845	0.803
NNMF [7]	0.843	–
LLORMA-Local [17]	0.833	0.782
I-AUTOREC [27]	0.831	0.782
CF-NADE [32]	**0.829**	**0.771**
GC-MC (Ours)	0.832	0.777

Model	Flixster	Douban	YahooMusic
GRALS	1.313/1.245	0.833	38.0
sRGCNN	1.179/0.926	0.801	22.4
GC-MC	**0.941/0.917**	**0.734**	**20.5**

Experiment results

- **Cold-start analysis -** The experiment looks at MovieLens100K dataset and tries to verify the added value of side information by masking the available ratings of *Nc* users to be no more than *Nr*. In the original dataset each user had at least 20 available ratings.



Discussion

- The authors present a novel interpretation of the matrix completion task as a bipartite graph edge prediction problem.
- This approach outperforms other benchmarks in incorporating auxiliary information about the users and provide almost state of the art results for large matrix completion problems.
- This problem formation is more flexible and allows to incorporate naturally different types of data and structure together.
- The authors offer a simple and clean approach for weight sharing which can be extended to different edges types as well