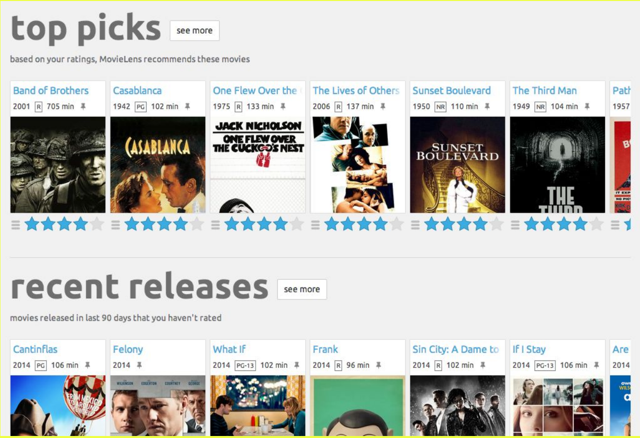
Movie Recommendation System On MovieLens Dataset

Tianyou Xiao, Ziyu Song

Table of Contents

How to predict a user’s preference based on the movie he/she reviewed?



Data Preprocessing

- MovieLens Dataset
- GroupLens Website

Two Categories of Recommendation System

Memory-Based Techniques

- Similarity Measures
  - Cosine Similarity
  - Pearson Correlation
  - Jaccard Coefficient
- Match Similar Users and Items
- Collaborative Filtering

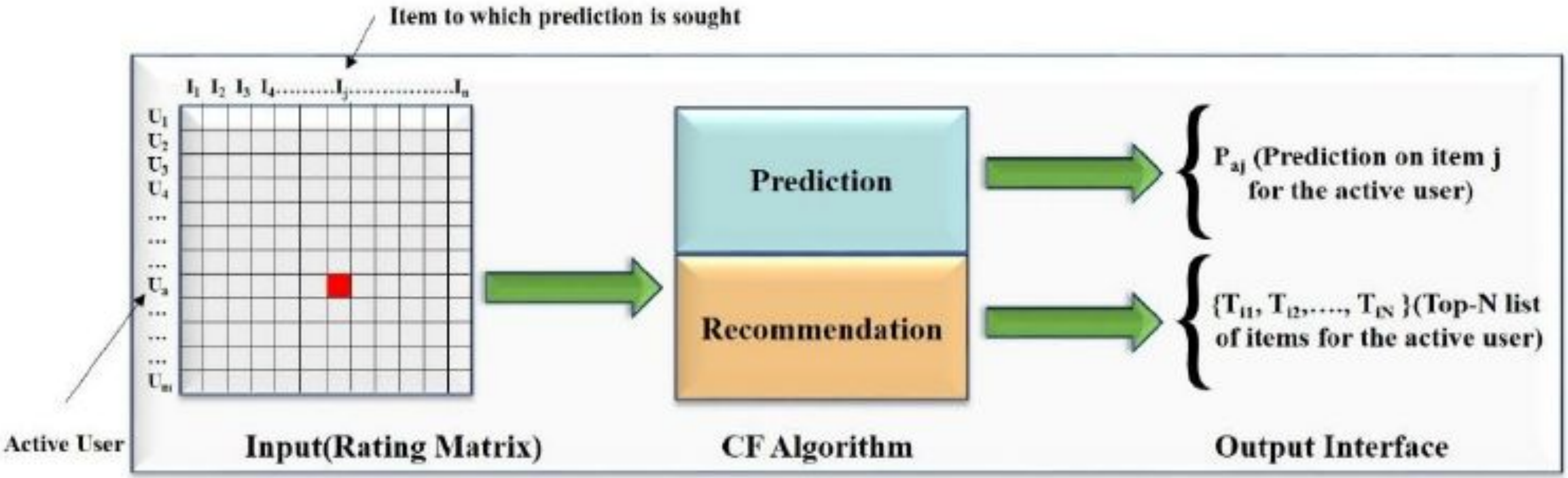
Model-Based Techniques

- Predict based on pre-trained models
- Typical model-based techniques:
  - Bayesian Networks
  - Singular Value Decomposition
  - Probabilistic Latent Semantic Analysis
  - Deep learning

Collaborative Filtering

- A Common Memory-Based Strategy
- Collects Information From Many Users (Collebrating)
- Makes Automatic Predictions (Recommendation/Filtering)
- User-User Based and Item-Item Based

Collaborative Filtering



The Collaborative Filtering Process

Sarwar, B., Karypis, G., Konstan, J. & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. Proceedings of the 10th international conference on World Wide Web (p./pp. 285--295), New York, NY, USA: ACM. ISBN: 1-58113-348-0

Collaborative Filtering

User-User Based

- Predict with similarities between users

Item-Item Based

- Predict with similarities between items

Weighted Average

$$\hat{r}_{ui} = \frac{\sum_{u'} sim(u, u') r_{u'i}}{\sum_{u'} |sim(u, u')|}$$

Top-k CF

$$\hat{r}_{ui} = \frac{\sum_{u'_k} sim(u, u'_k) r_{u'_k i}}{\sum_{u'_k} |sim(u, u'_k)|}$$

Theoretically, they are dual methods and should produce similar result.

Similarity

Cosine Similarity

$$cos(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|} = \frac{\sum_{i=1}^{n} u_i v_i}{\sqrt{\sum_{i=1}^{n} u_i^2} \sqrt{\sum_{i=1}^{n} v_i^2}}$$

A common metric to measure the similarity between two users/items

calculate the cosine value of the angle between the two users'/items' vectors

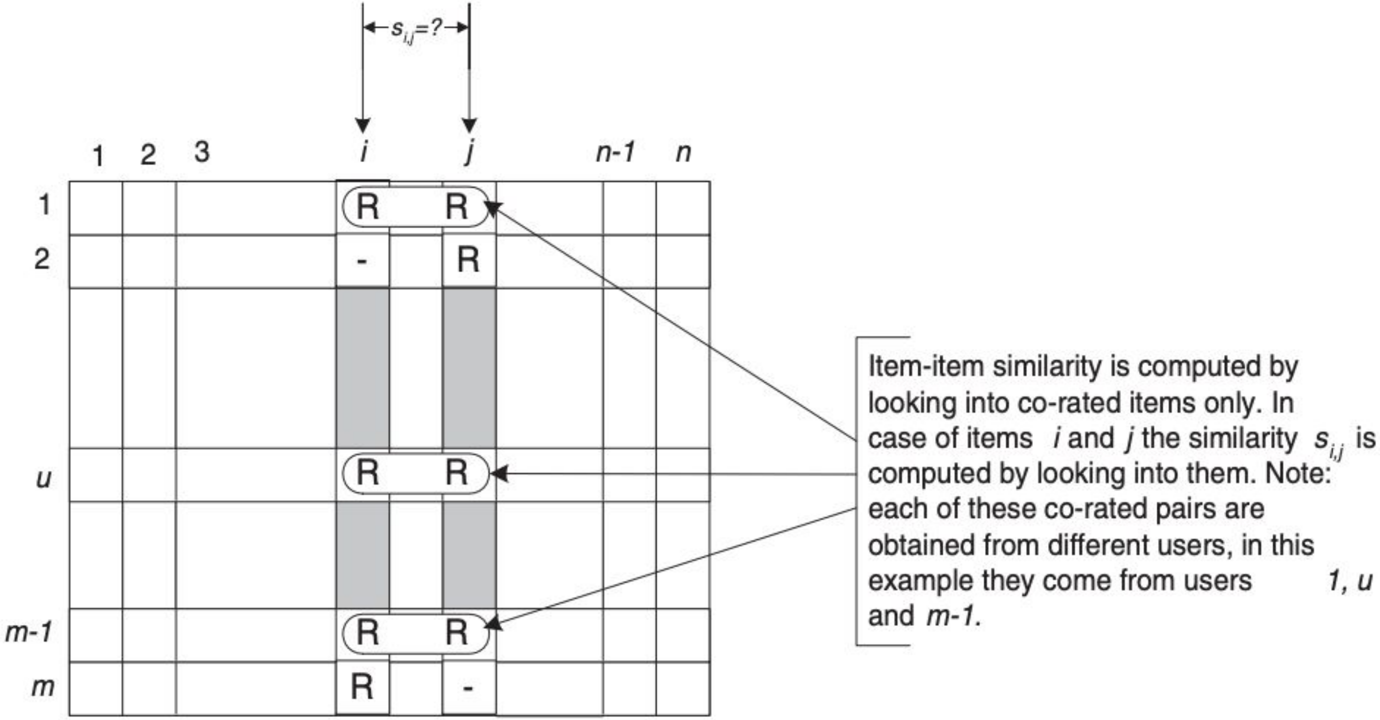
Euclidean Distance

$$d(\mathbf{u}, \mathbf{v}) = \sqrt{(r_{u1} - r_{v1})^2 + (r_{u2} - r_{v2})^2 + \cdots + (r_{up} - r_{vp})^2}$$

Euclidean distance can be used to measure how far two points/vectors are from each other.

Here we use reciprocal of Euclidean distance as another similarity metric.

Item-Item Similarity

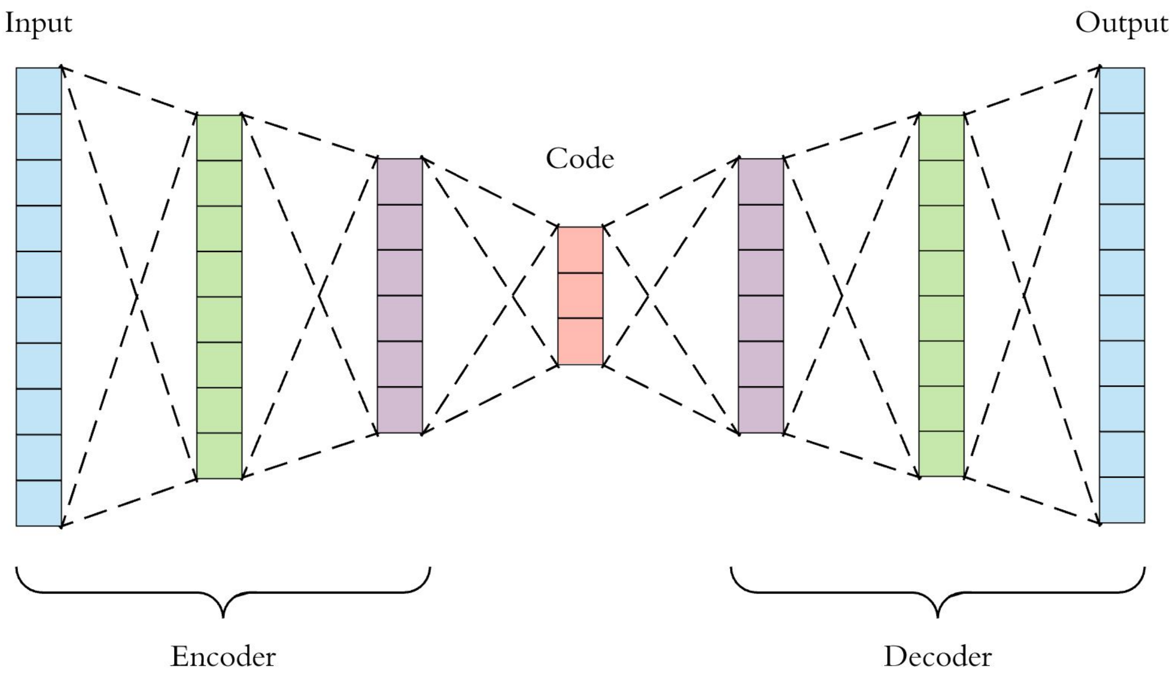


Sarwar, B., Karypis, G., Konstan, J. & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. Proceedings of the 10th international conference on World Wide Web (p./pp. 285--295), New York, NY, USA: ACM. ISBN: 1-58113-348-0

Neural-Network-Based Autoencoder

- Generates a Model To Predict The Missing Ratings
- Learn The Relations Between Values of Inputs By Recreating Ratings of Every User

Autoencoder



$$\phi : \mathcal{X} \rightarrow \mathcal{F}$$

$$\psi : \mathcal{F} \rightarrow \mathcal{X}$$

$$\phi, \psi = \underset{\phi,\psi}{\operatorname{argmax}} \|X - (\psi \circ \phi)X\|^2$$

the encoder and the decoder,

defined as transitions ϕ and ψ,

$\mathcal{F}$ is the feature space

$\mathcal{X}$ is the input space.

Evaluation

- Rooted Mean Squared Error (RMSE)
- Mean Absolute Error (MAE)

Evaluation Metrics

Rooted Mean Squared Error (RMSE)

A frequently used measure of differences between values predicted by a model or an estimator and the values observed

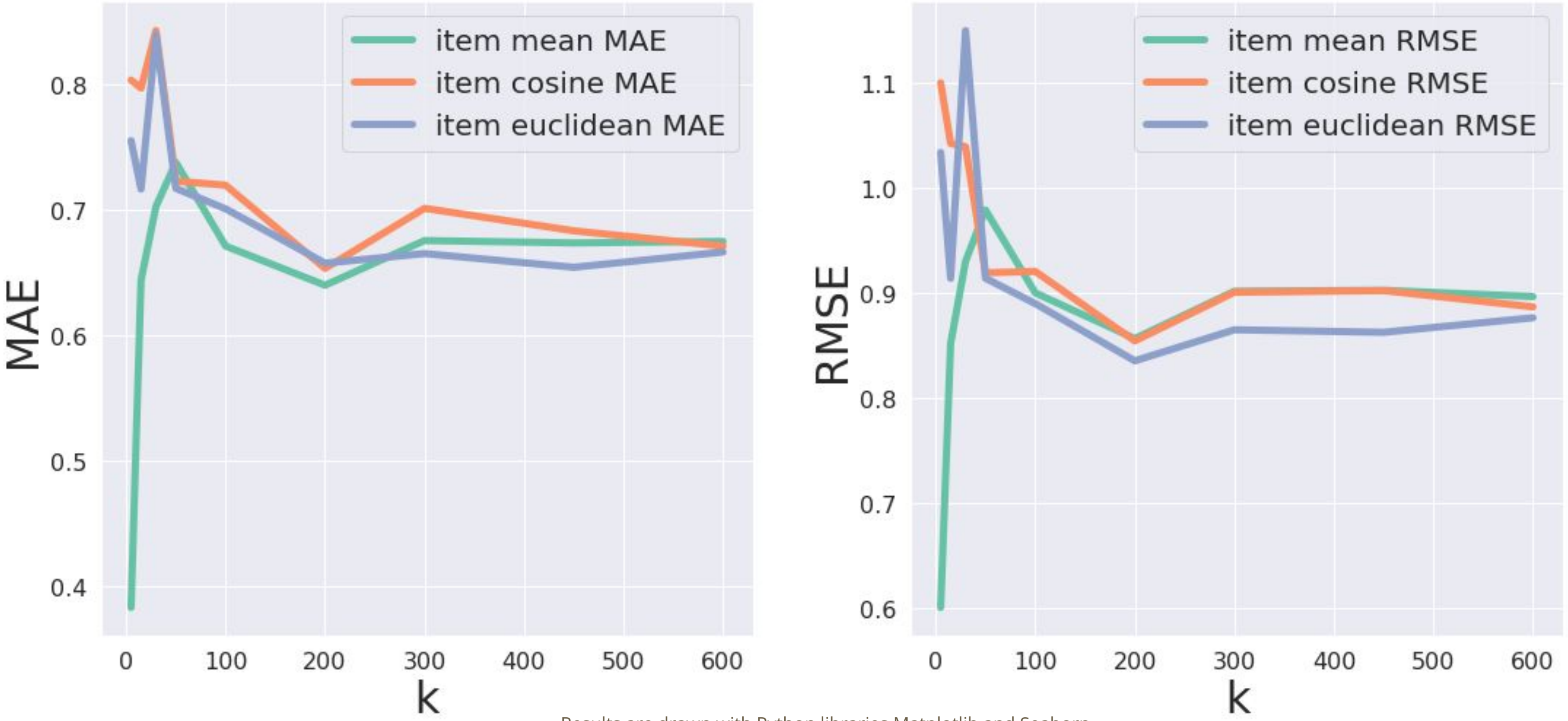
$$RMSE = \sqrt{\frac{\sum_{i=1}^{m}\sum_{j=1}^{n}(\hat{r_{ij}} - r_{ij})^2}{m * n}}$$

Mean Absolute Error (MAE)

Measures the average magnitude of the errors in a set of predictions, without considering their direction

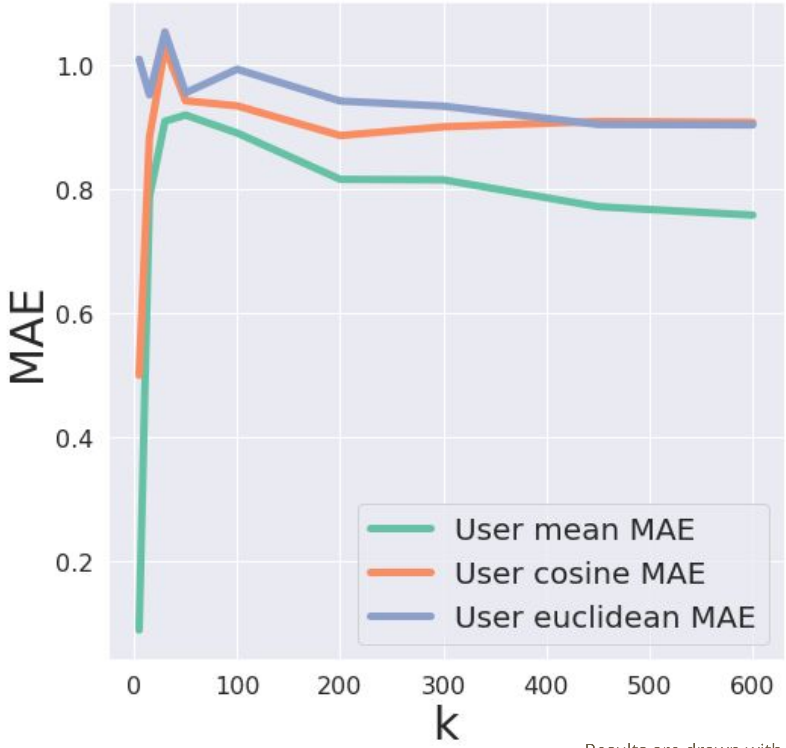
$$MAE = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}|r_{ij} - \hat{r_{ij}}|}{m * n}$$

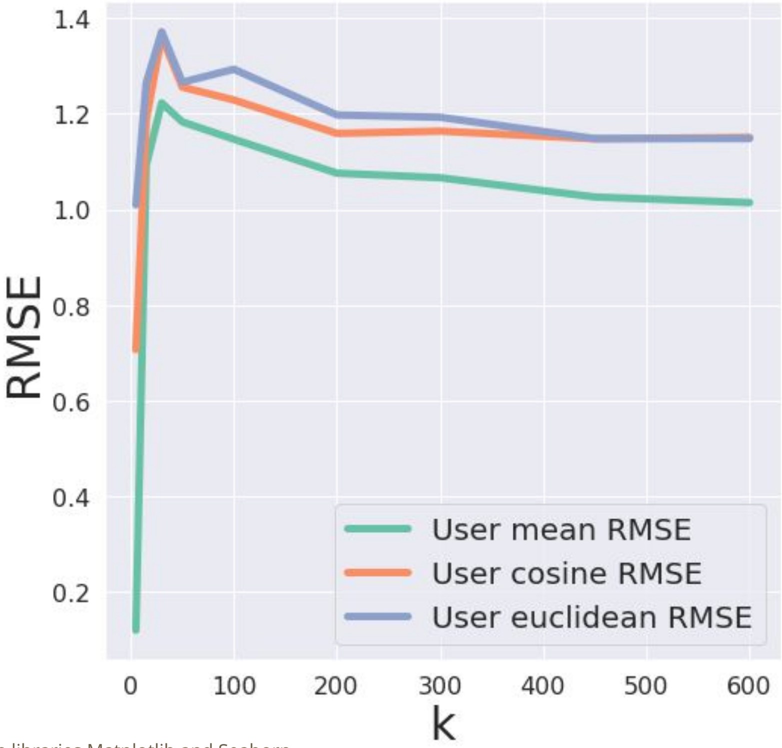
CF: Item-Item MAE and RMSE



Results are drawn with Python libraries Matplotlib and Seaborn

CF: User-User MAE and RMSE





Results are drawn with Python libraries Matplotlib and Seaborn

Sample Rating Prediction – CF

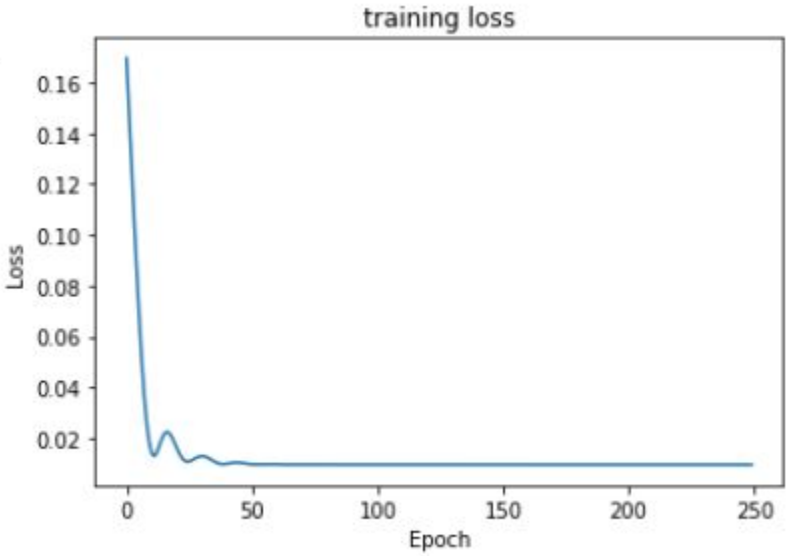
Sample rating prediction for user 1, top 50 similarity, merged with movie information

	movieId	title	genres	predicted_rating
42	924	2001: A Space Odyssey (1968)	Adventure|Drama|Sci-Fi	5.000000
8	111	Taxi Driver (1976)	Crime|Drama|Thriller	4.833511
69	2324	Life Is Beautiful (La Vita è bella) (1997)	Comedy|Drama|Romance|War	4.750258
102	58559	Dark Knight, The (2008)	Action|Crime|Drama|IMAX	4.749980
17	318	Shawshank Redemption, The (1994)	Crime|Drama	4.738419
40	904	Rear Window (1954)	Mystery|Thriller	4.699766
83	4226	Memento (2000)	Mystery|Thriller	4.666927
50	1206	Clockwork Orange, A (1971)	Crime|Drama|Sci-Fi|Thriller	4.625423
110	116797	The Imitation Game (2014)	Drama|Thriller|War	4.625171
52	1219	Psycho (1960)	Crime|Horror	4.624546

Neural-Network-Based Autoencoder

```
Epoch:  0 | Step:  0 | train loss:  0.16952997
Epoch:  1 | Step:  0 | train loss:  0.14799552
Epoch:  2 | Step:  0 | train loss:  0.12715906
Epoch:  3 | Step:  0 | train loss:  0.10669372
Epoch:  4 | Step:  0 | train loss:  0.08686035
Epoch:  5 | Step:  0 | train loss:  0.068255045
Epoch:  6 | Step:  0 | train loss:  0.05148868
Epoch:  7 | Step:  0 | train loss:  0.03726017
Epoch:  8 | Step:  0 | train loss:  0.026145343
Epoch:  9 | Step:  0 | train loss:  0.01843946
Epoch:  10 | Step:  0 | train loss:  0.0141458465
Epoch:  11 | Step:  0 | train loss:  0.012979386
Epoch:  12 | Step:  0 | train loss:  0.01425484
Epoch:  13 | Step:  0 | train loss:  0.016896825
Epoch:  14 | Step:  0 | train loss:  0.019705849
Epoch:  15 | Step:  0 | train loss:  0.021741267
Epoch:  16 | Step:  0 | train loss:  0.022530701
Epoch:  17 | Step:  0 | train loss:  0.022037413
Epoch:  18 | Step:  0 | train loss:  0.02052057
```

(a) part of training process



(b) training loss graph over epochs

Our model achieves an MAE loss of 0.0096 with *learning rate = 0.01, epochs = 250,* and *batchsize = 64*, running for 417 seconds.

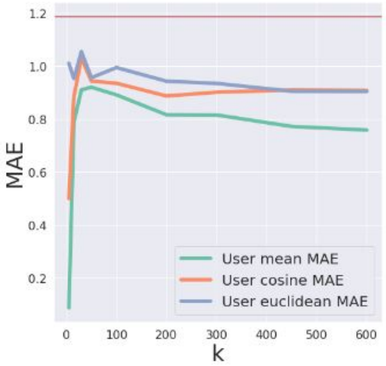
Sample Rating Prediction – Autoencoder

```
tensor([[[3.9682, 3.4754, 3.1795,  ..., 3.5001, 3.5000, 4.0001],
         [3.9682, 3.4753, 3.1795,  ..., 3.5000, 3.5000, 4.0000],
         [3.9665, 3.4739, 3.1785,  ..., 3.4983, 3.4985, 3.9982],
         ...,
         [3.9682, 3.4754, 3.1795,  ..., 3.5001, 3.5000, 4.0001],
         [3.9682, 3.4753, 3.1795,  ..., 3.5000, 3.5000, 4.0000],
         [3.9682, 3.4754, 3.1795,  ..., 3.5001, 3.5000, 4.0001]]],
       grad_fn=<AddBackward0>)
```

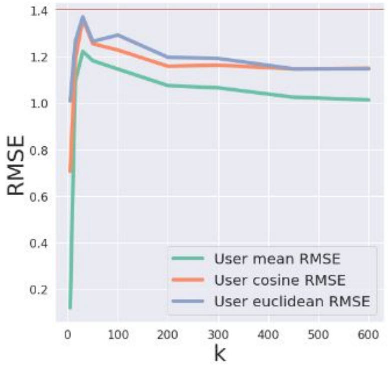
Sample rating prediction from autoencoder strategy

Conclusion

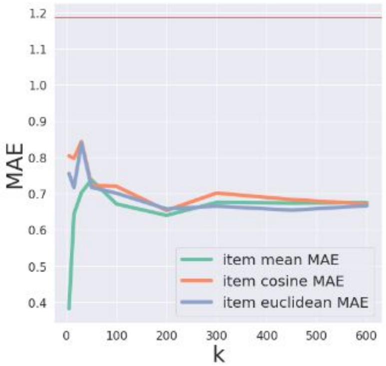
Comparison between memory-based and model-based performance



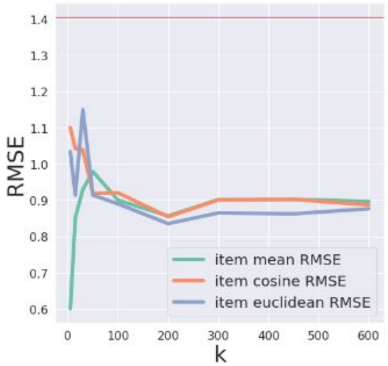
(a) User MAE and Autoencoder



(b) User RMSE and Autoencoder



(c) Item MAE and Autoencoder



(d) Item RMSE and Autoencoder

Thank You!

Date Presented: December 13, 2018