2020·06

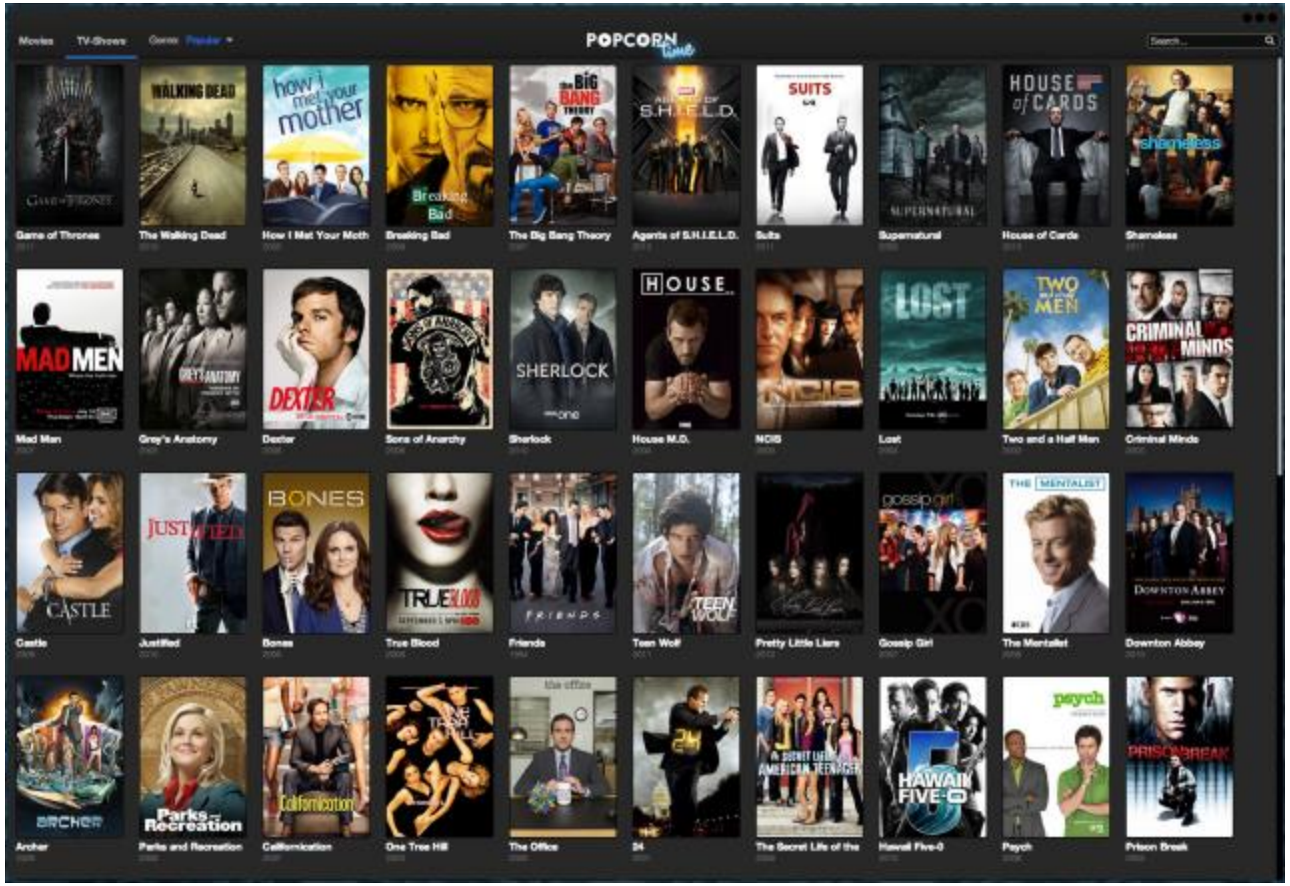
以DAE向量雜訊移除為基礎之新進使用者冷啟動推薦

A Cold Start Recommendation Method for New Users Based on DAE Vector Noise Removal

指導教授：陳建錦　　學生：吳承翰

Why is Recommendation System Important ?

For user, there are tons of movies



Why is Recommendation System Important ?

For platforms, well personalization can reduce subscriber churn and benefit from:

1. **Increase the lifetime value of the existing subscribers**

2. **Reduce the effort of acquiring new subscribers.**

Carlos A. Gomez-Uribe, Neil Hunt. 2015 [1]
(ACM Transactions on Management Information Systems)
"The Netflix Recommender System: Algorithms, Business Value, and Innovation"

Agenda

01 研究主題
Research Topic

02 文獻回顧
Literature Review

03 研究方法
Research Method

04 研究結果分析
Result & Analysis

05 結論與未來展望
Conclusion & Future Work

1. Research Topic: Cold start problem in recommendation system

- What’s the difficulty of cold start problem
- Apply deep learning to solve cold start problem

1 Research Topic

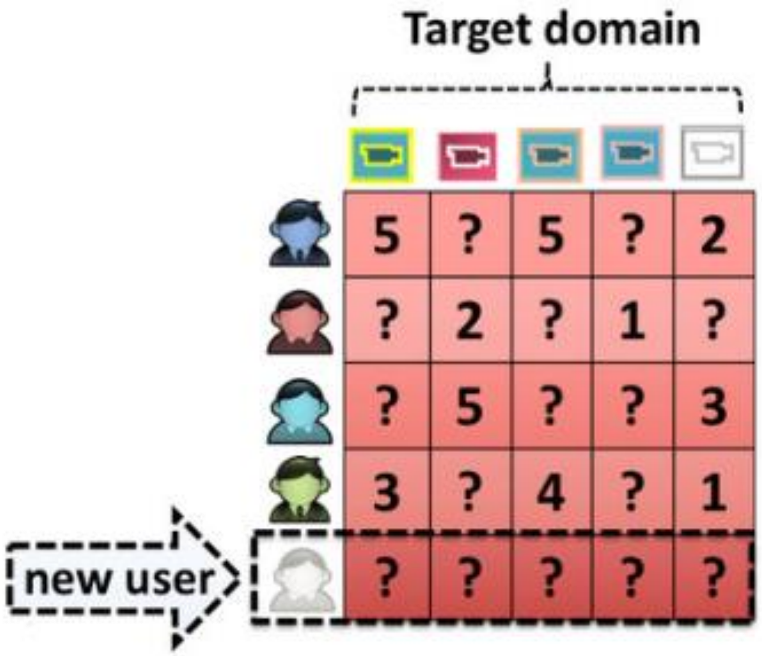
What is the difficulty of cold start problem

Non Cold Start Context

We can use rating similarity between two people to identify potential items a person would like.

Cold Start Context

New user do not have ratings overlap with other people.



1

Research Topic

Apply deep learning to solve cold start problem

Why deep learning?

Because deep learning based methods have two major advantages:

Nonlinear relationship simulation

Different kinds of nonlinear activation function

- Sigmoid
- ReLU
- Tanh

Effective feature extraction

Automatically extract feature from

- Text
- Image
- Video / Audio

Application Examples

In the industry

Covington, et al., 2016 [6] use deep neuron network to model Youtube recommendation.

Cheng, et al., 2016 [7] apply deep neuron network to model Google Play APP recommendation.

In Academia

ACM RecSys included a deep learning based recommendation system as one of the conference themes since 2016.

Apply deep learning to solve cold start problem

Which type of deep learning model will we use? And why?

We will use Denoising Autoencoder (DAE), because DAE has two appealing features:

Denoising

In the field of image processing, DAE is often used to restore images disturbed by noise

This feature meets our needs !

Dimension reduction

DAE is good at learning top-quality latent representation of data

What is the relation between cold start problem and noise?

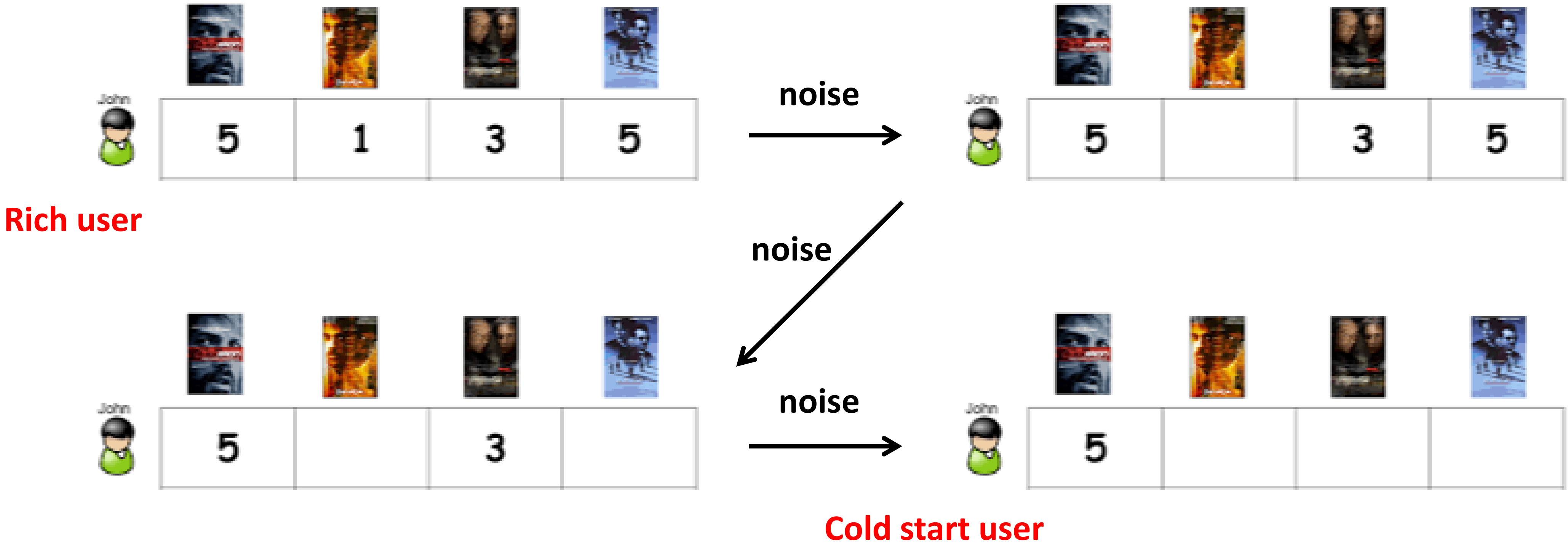
1

Research Topic

Apply deep learning to solve cold start problem

Relation between cold start problem and noise?

We can regard a cold start user as the user state that comes from a rich user disturbed by noise.



1

Research Topic

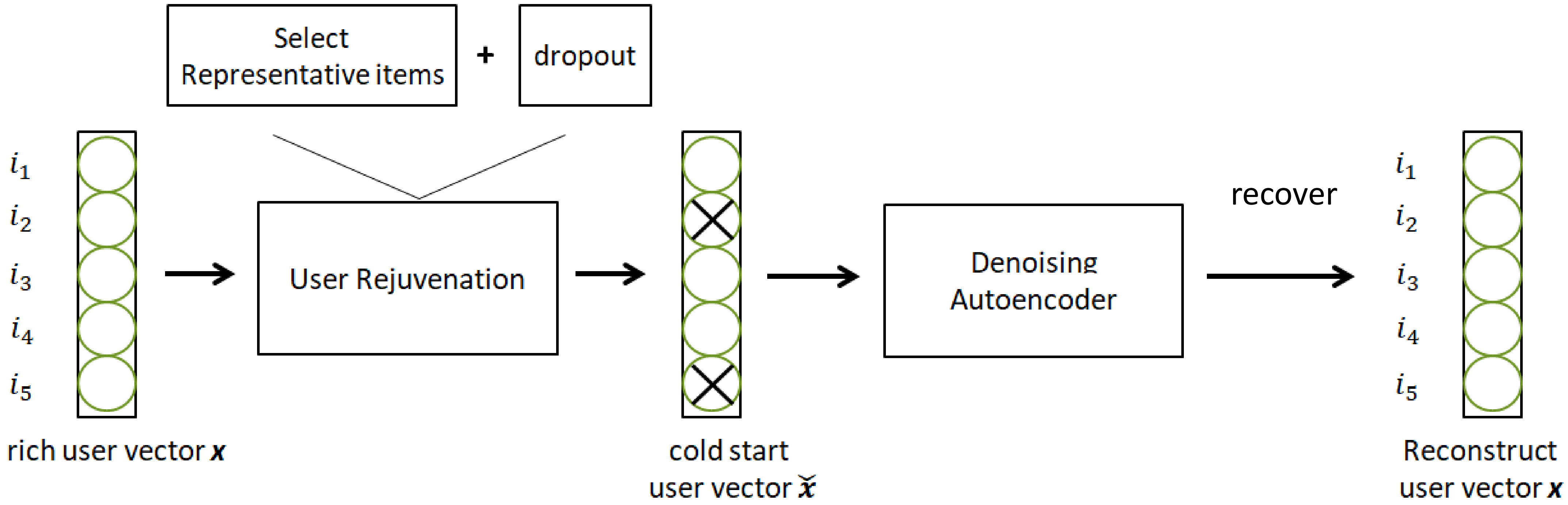
Apply deep learning to solve cold start problem

Relation between DAE and cold start problem?

We design a mechanism called **"user rejuvenation"** which turns rich user back to cold start user state.

After getting cold start users, we can use DAE to recover them back to the rich users' state.

To the best of our knowledge, we are the first to propose noise generating mechanism combine with DAE for solving cold start problem in the field of recommendation system



2. Literature Review

- References
- Deep learning methods for solving cold start problem (1)
- Representative items mining (2)

References

Deep learning methods for recommendation system		
Author	**Paper**	**index**
Fu,M., et al.	A novel deep learning-based collaborative filtering model for recommendation system	[11]
He, X., et al.	Neural collaborative filtering	[12]
Covington, P., J. Adams, and E. Sargin	Deep neural networks for youtube recommendations	[6]

Autoencoder based methods for recommendation system		
Zhuang, F., et al.	Representation learning via Dual-Autoencoder for recommendation	[14]
Vincent, P., et al.	Stacked denoising autoencoders: Learning useful representation in a deep network with a local denoising criterion	[8]
Wu, Y., et al.	Collaborative denoising auto-encoders for top-n recommender systems	[9]
Majumdar, A. and A.Jain	Cold-start, warm-start and everything in between: an autoencoder based approach to recommendation	[13]

Model Performance Comparison

References

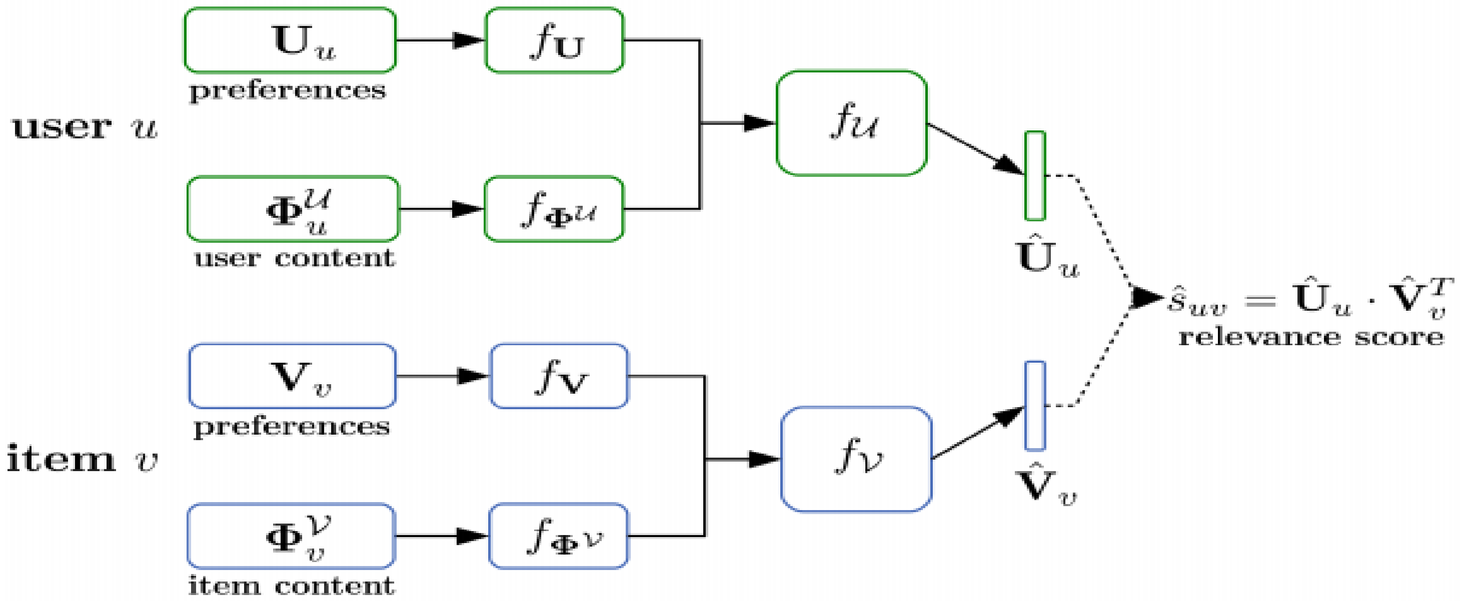
Combine the ideas of these three papers

Deep learning methods for solving cold start problem		
Author	**Paper**	**index**
1 Volkovs, M., G.Yu, and T. Poutanen	Dropoutnet: Addressing cold start in recommender systems.	[15]
Shi, S., et al.	Attention-based adaptive model to unify warm and cold starts recommendation	[16]

Representative items mining		
Author	**Paper**	**index**
2 Liu, N.N., et al.	Wisdom of the better few: cold start recommendation via representative based rating elicitation.	[3]
3 Shi, L., W.X. Zhao, and Y.-D.J.A.T.o.I.S. Shen	Local representative-based matrix factorization for cold start recommendation	[5]
Georgiou, O. and N. Tsapatsoulis.	The importance of similarity metrics for representative users identification in recommender systems.	[4]

1 DropoutNet: Addressing cold start in recommender systems. [Volkovs, M.,G. Yu,and T. Poutanen]

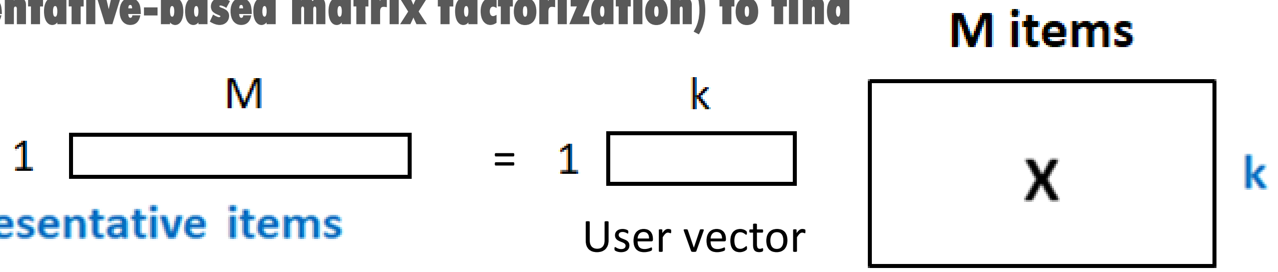
Combine both preference data and content information as model inputs. While training the model, if the item or user is under cold start scenario, we set the corresponding item preference or user preference to zero vector, called “dropout”.

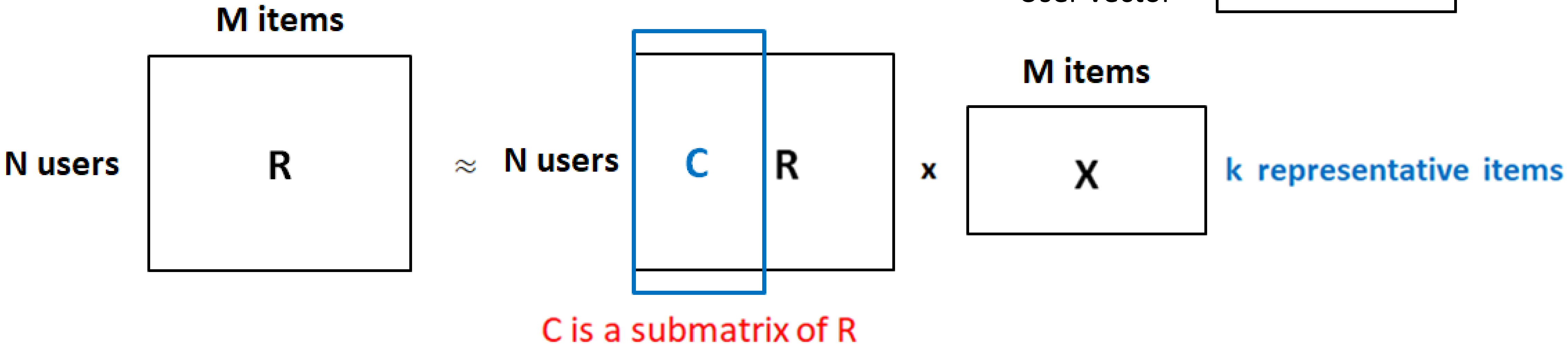


Inspiration : We can dropout the unimportant information while training the model !

2 Wisdom of the better few: cold start recommendation via representative based rating elicitation. [Nathan Liu, Xiangrui Meng, Chao Liu, Qiang Yang]

Propose a method called RBMF (representative-based matrix factorization) to find representative items from item set.



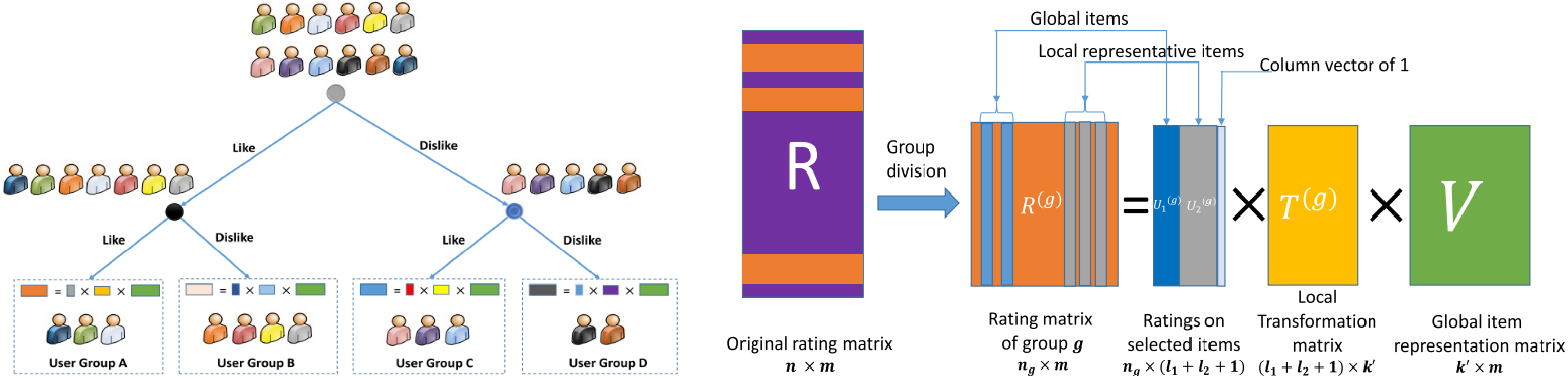


Inspiration : We can use RBMF to choose representative items!

3 Local representative-based matrix factorization for cold-start recommendation.

[Lei Shi, Wayne Xin Zhao, Yi-Dong Shen]

LRBMF tries to improve the RBMF by grouping people and find "global representative items" and "local representative items" respectively for a group of users.



Inspiration : We can emulate LRBMF to split users into different groups and apply the RBMF.

3. Research Method

- Introduction of DAE
- User rejuvenation
- Cold start user recommendation

Introduction of DAE

Autoencoder

Autoencoder is a deep learning model composed of **encoder** and **decoder**:

- **Encoder** is used to compress high dimension input vector into low dimension latent vector.

$$f_\theta(\mathrm{x}) = \sigma(\mathrm{Wx} + \mathrm{b})$$

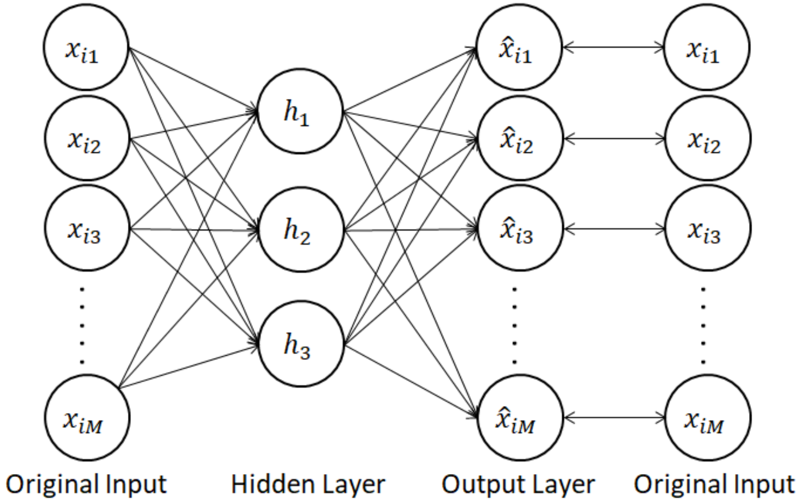
- **Decoder** is used to recover the low dimension latent vector back to input dimension.

$$g_\varphi(\mathrm{x}) = \sigma(\mathrm{W'x} + \mathrm{b'})$$

- The goal of training Autoencoder is to minimize the reconstruction error between input vector and output vector.

Input vector Output vector

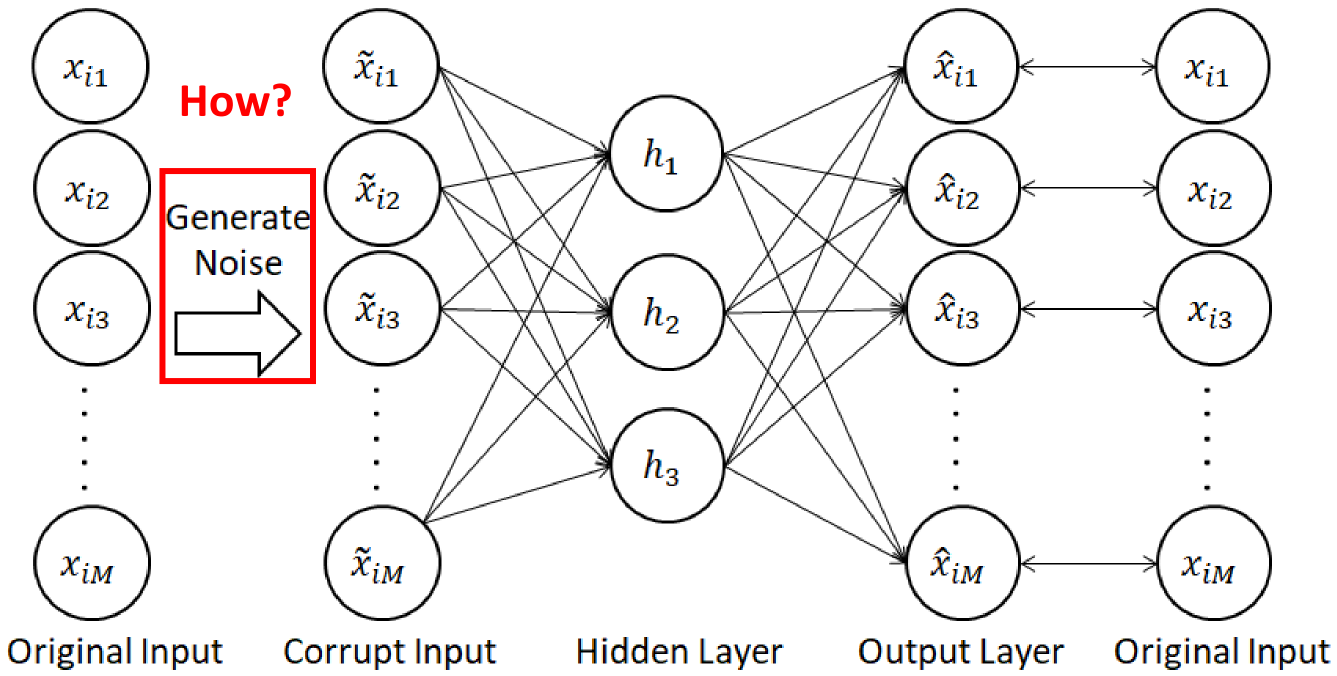
$$\underset{\theta,\varphi}{\operatorname{argmin}} \sum_{i=1}^{N} ||x^{(i)} - \hat{x}^{(i)}||^2 = ||\underline{x^{(i)}} - \underline{g_\varphi(f_\theta(\hat{x}^{(i)}))}||^2$$



Denoising Autoencoder

Denoising Autoencoder is a variant of Autoencoder, which will generate noise in some dimensions of the input vector before training the following Autoencoder.

The reason behind the noise? **It can make the trained model more robust.**



Noise generation methods for Denoising Autoencoder

1. **Gaussian Noise**

   Generate noise according to normal distribution, and add the noise to the dimension of input vector.

2. **Masking Noise**

   Randomly set a part of the dimension value in input vector to zero.

3. **Salt-and-pepper Noise**

   Randomly set a part of the dimension value in input vector to the maximum or minimum allowable value for the dimension.

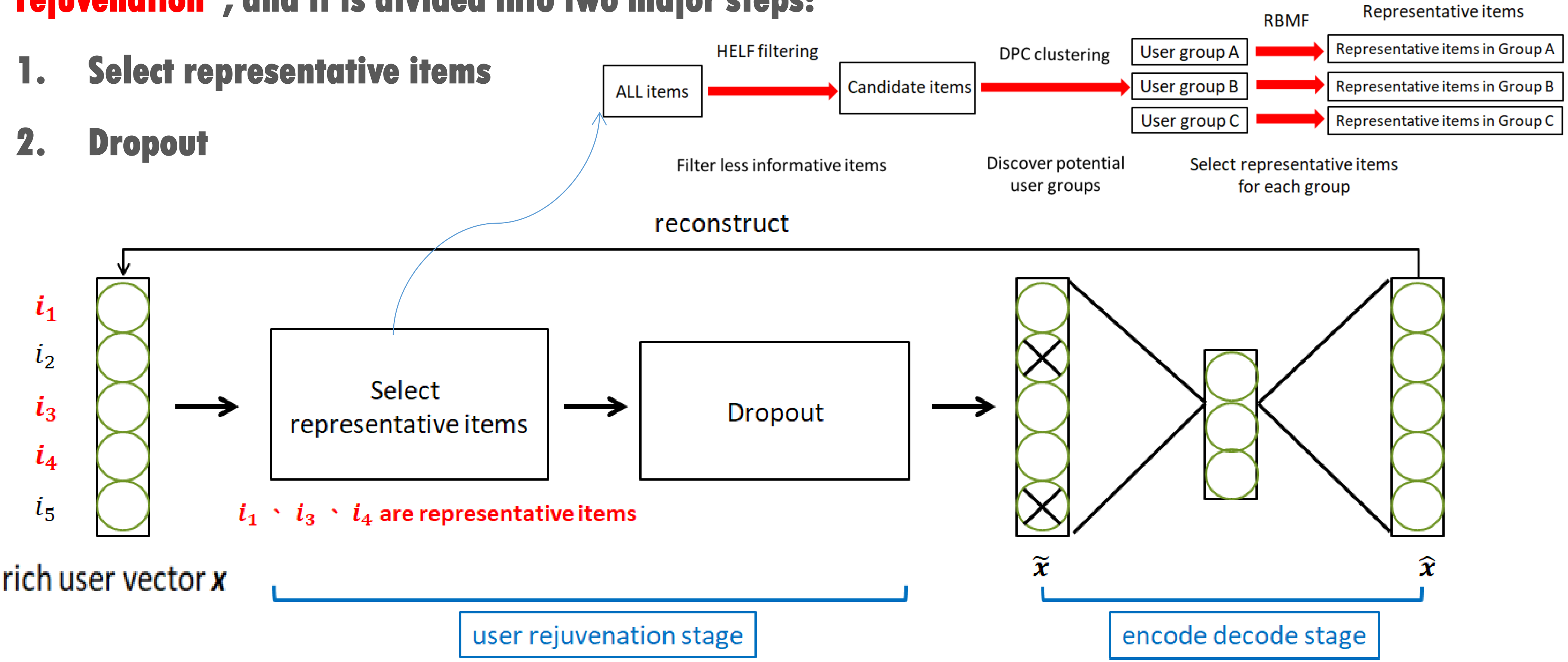
They are suitable for the image processing and speech processing.

But may not for recommendation task and low interpretability.

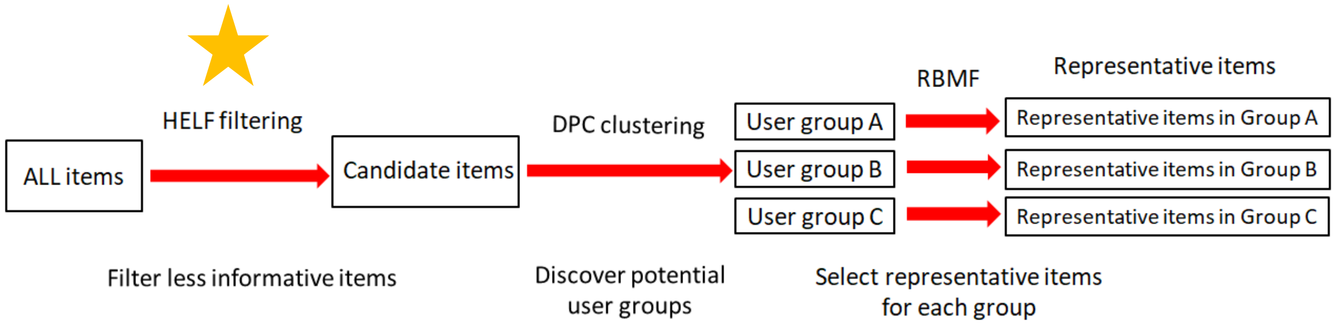
User rejuvenation

We design a noise generation method especially for the recommendation task, called **"user rejuvenation"**, and it is divided into two major steps:

1. Select representative items
2. Dropout



Harmonic mean of Entropy and Logarithm of Frequency (HELF)



When selecting candidate items, we will consider two characteristics of an item:

1. **Item popularity:**

   If more users rate the item, the item will have higher popularity.

2. **Rating entropy:**

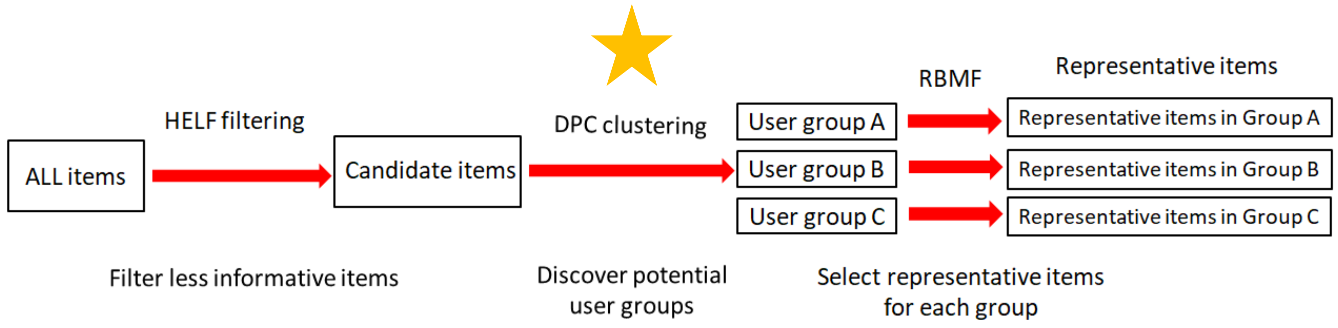
   If the users' preference for an item is more inconsistent, the item will have higher rating entropy.

Thus, we can calculate a metric, called HELF, to help us filter out less informative items.

$$HELF = \frac{2 * popularity * rating\ entropy}{popularity + rating\ entropy}$$

Density Peaks Clustering

After filtering out less informative items by HELF, the remaining items are called “candidate items”.



Each user is represented by candidate items, and we can use DPC to cluster users into groups.

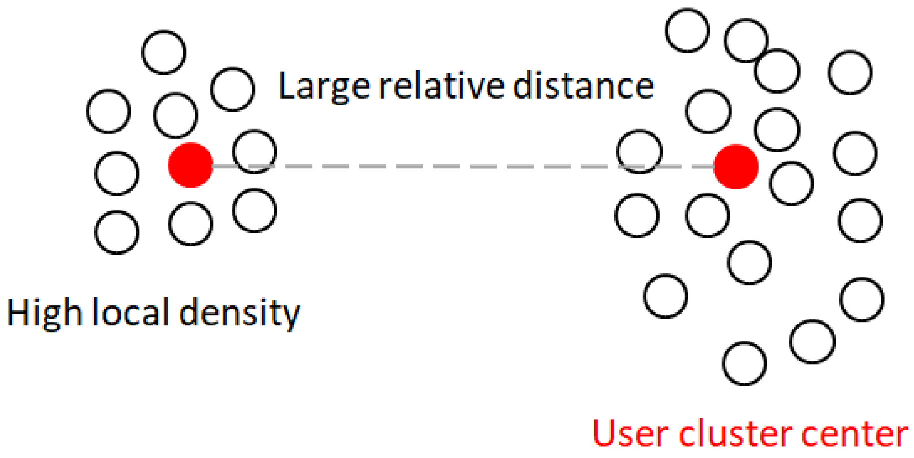
The cluster centers found by the DPC have two characteristics:

1. **High local density**

   There will be many users near the cluster center.

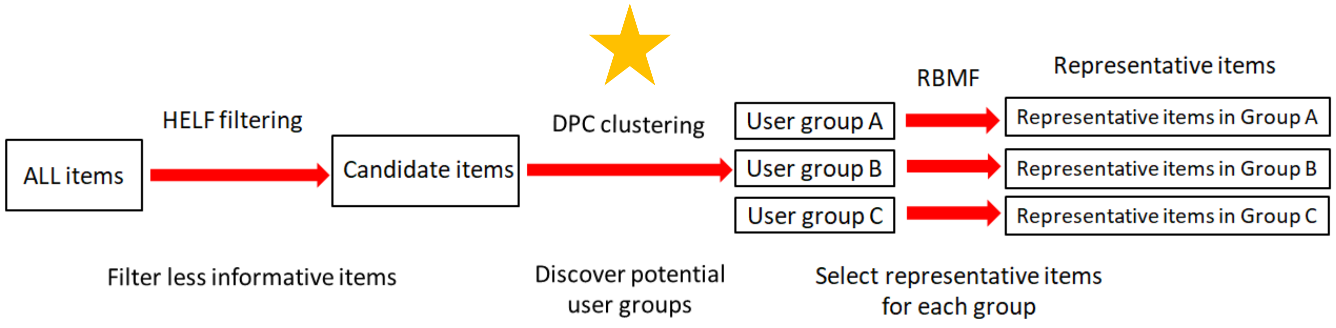
2. **Large relative distance**

   The distance between cluster centers are large.



Density Peaks Clustering

1. Set a pre-define distance d_c



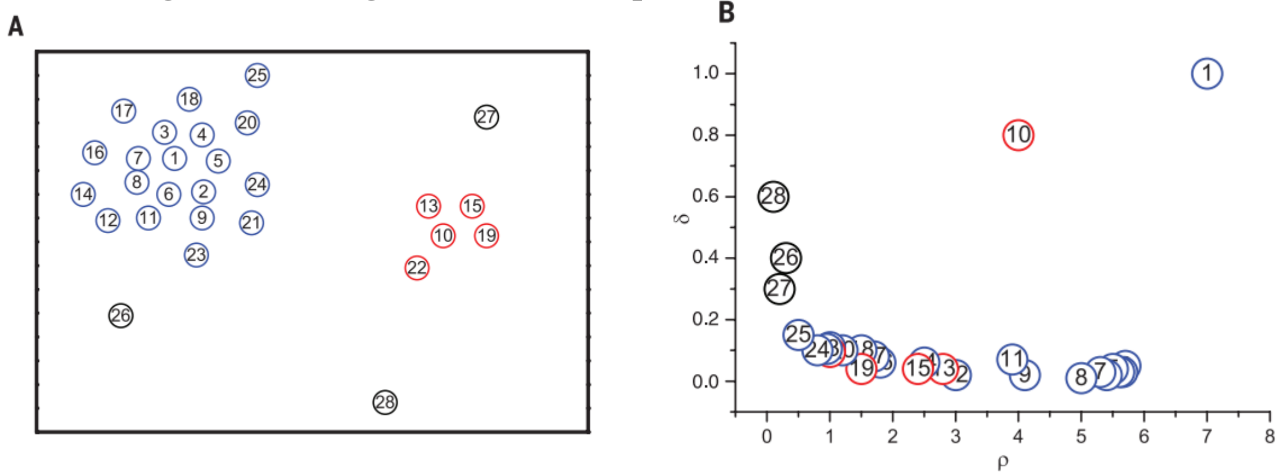
2. Calculate local density of all users.

$$\rho_i = \sum_j \chi(d_{ij} - d_c),\ \ \chi(x) = \begin{cases} 1\ ,\ x < 0 \\ 0\ ,\ x \geq 0 \end{cases}$$

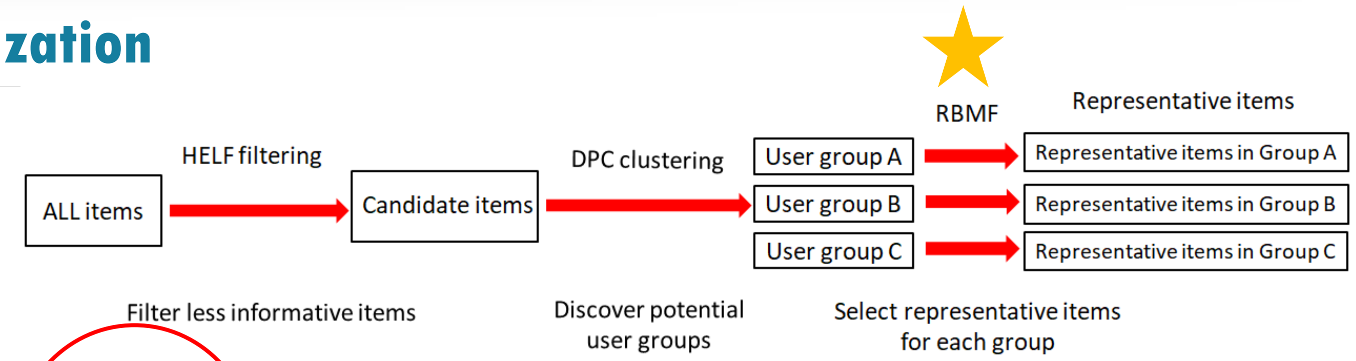
3. Calculate relative distance of all users.

$$\delta_i = max_j(d_{ij}) \qquad \delta_i = min_{j:\rho_j > \rho_i}(d_{ij})$$

4. Plot the decision diagram using local density and relative distance, and select cluster centers.

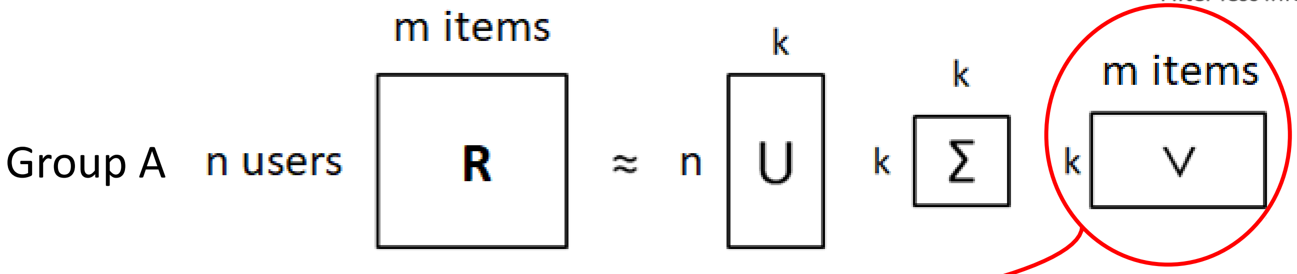


Representative-based matrix factorization

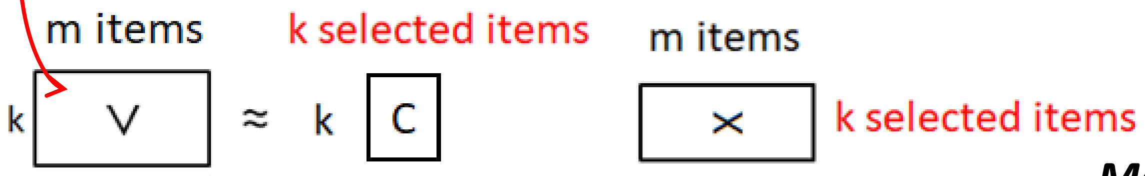


RBMF composed of two steps:

1. **SVD dimension reduction**



2. **Basis selection (use maximal volume algorithm)**



(C is submatrix of V)

Maximal Volume Algorithm:

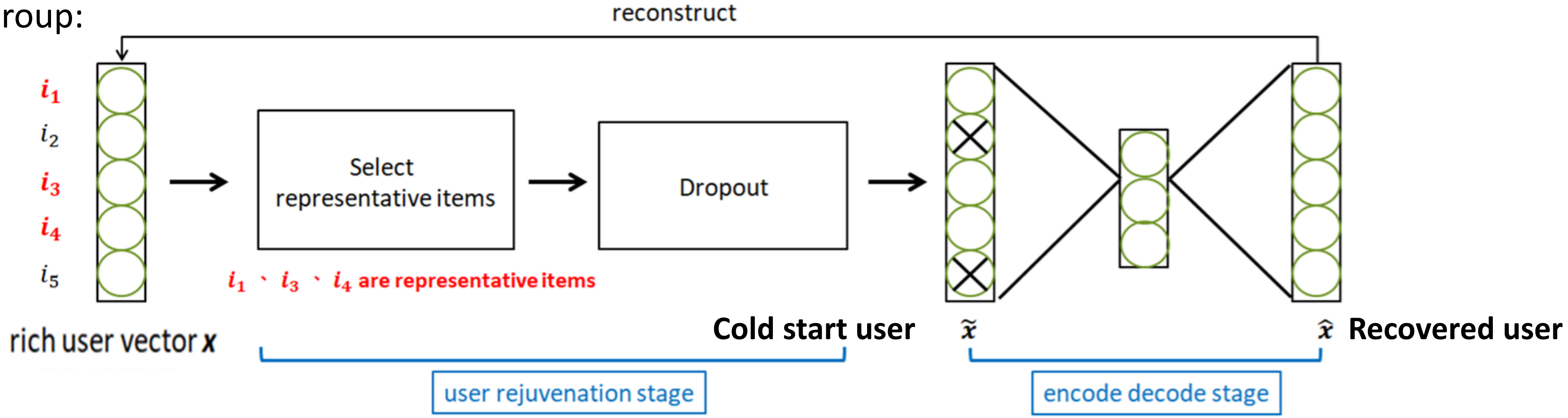
If all the entries of VC^{-1} are smaller than 1 in absolute value.
Select the corresponding items as representative items

3 Research Method

Cold start user recommendation

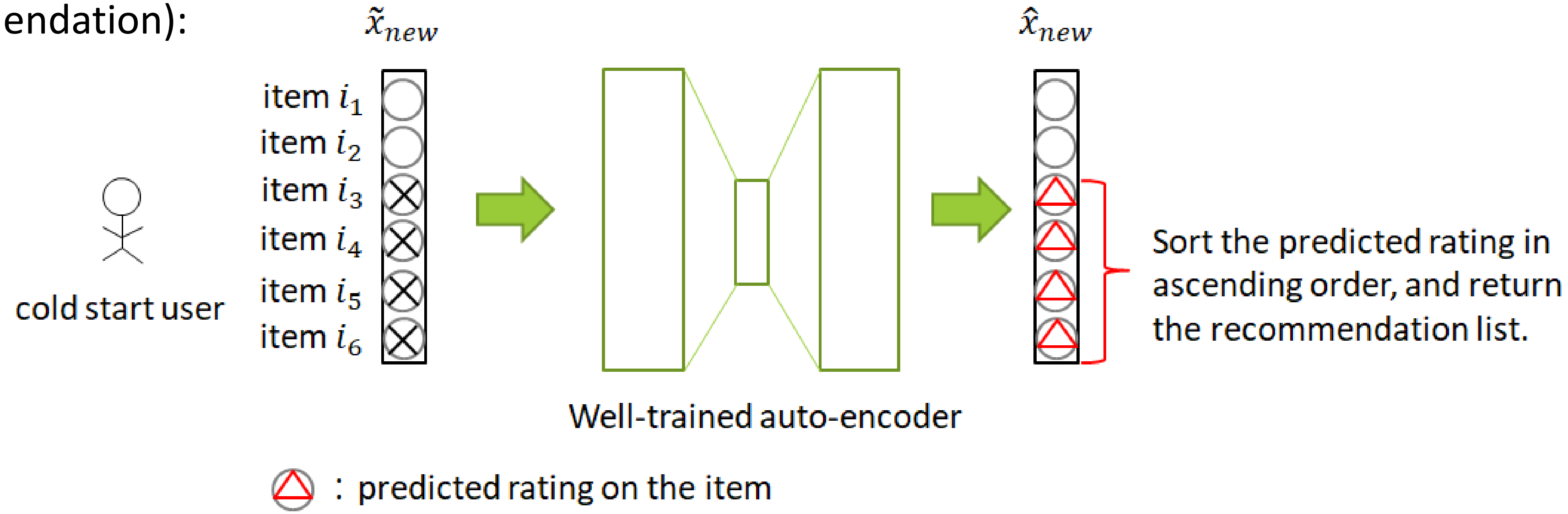
We train a DAE for each group of users.

Training Phase for a group:



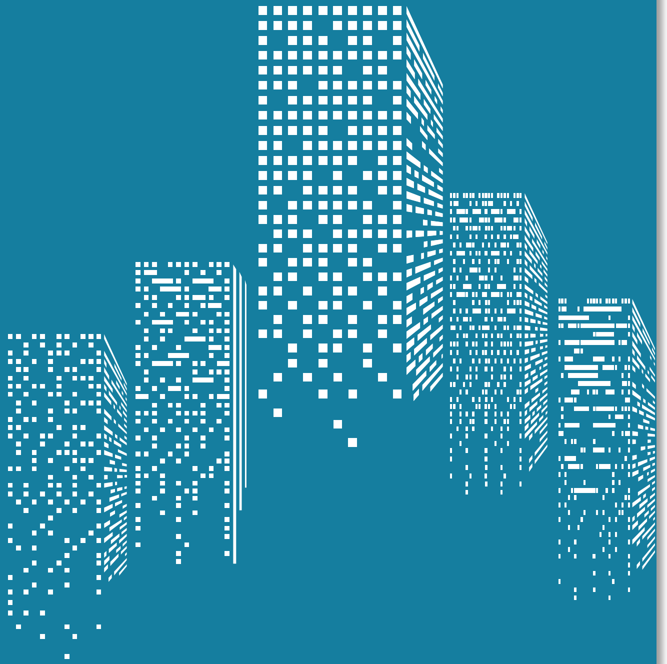
A test user is assigned to the DAE corresponding to the cluster center closest to him.

Test Phase (Recommendation):



4. Result & Analysis

- Dataset and Experiment Settings
- Evaluation Metrics
- Experiment Result Analysis



Dataset and Experiment Settings

Dataset

MovieLens 1M dataset is the movie rating data provided by MovieLens users, and we organize the ratings information of users in the table below.

	Num of ratings	Num of movies	Num of users
MovieLens 1M	1,000,209	3,900	6,040

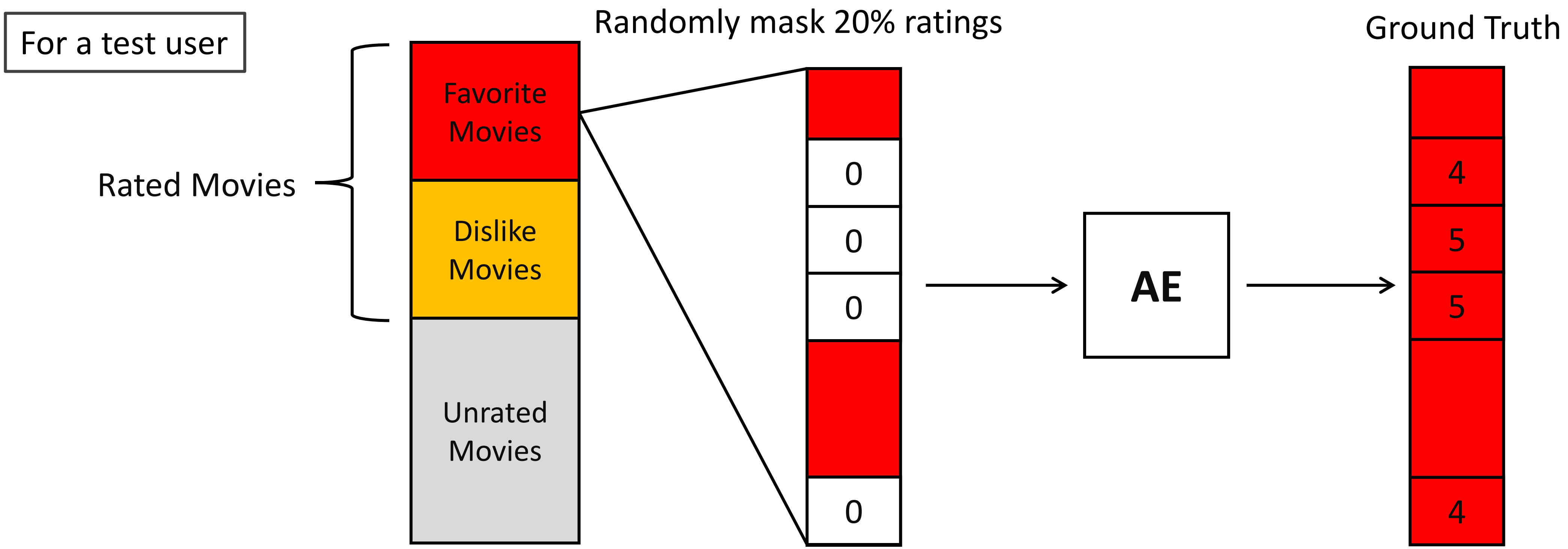
Experiment Settings (Train-Test split)

The user with more than 25 user ratings are regarded as training users, and the users with less than 25 ratings are regarded as test users.

Training set		Test set	
Num of users	Num of movies	Num of users	Num of movies
5549	3702	491	1995

Experiment Settings

- When training the model, use the users in the training set to train the parameters of the Autoencoder.
- When evaluating the effectiveness trained model, the 20% of the movie ratings that each test user likes are randomly covered to 0, which is regarded as the ground truth to be verified.

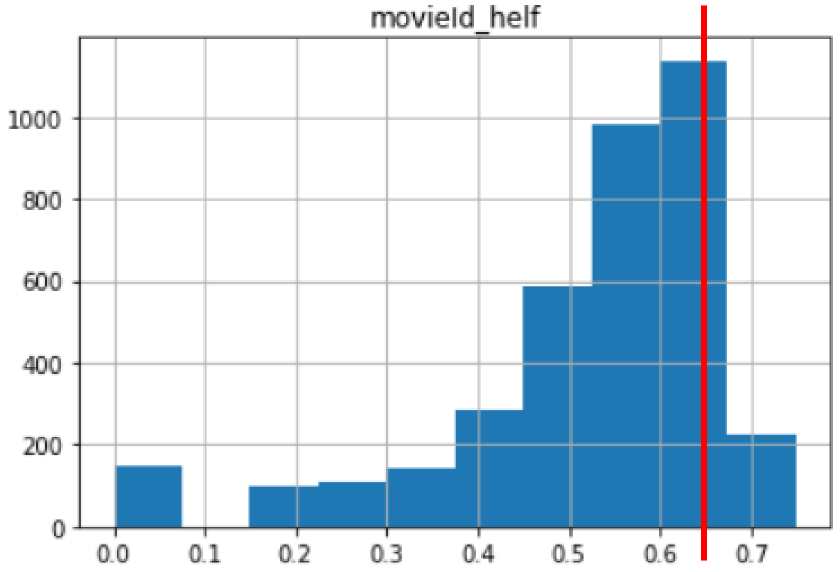


Experiment Settings (Hyper-parameters)

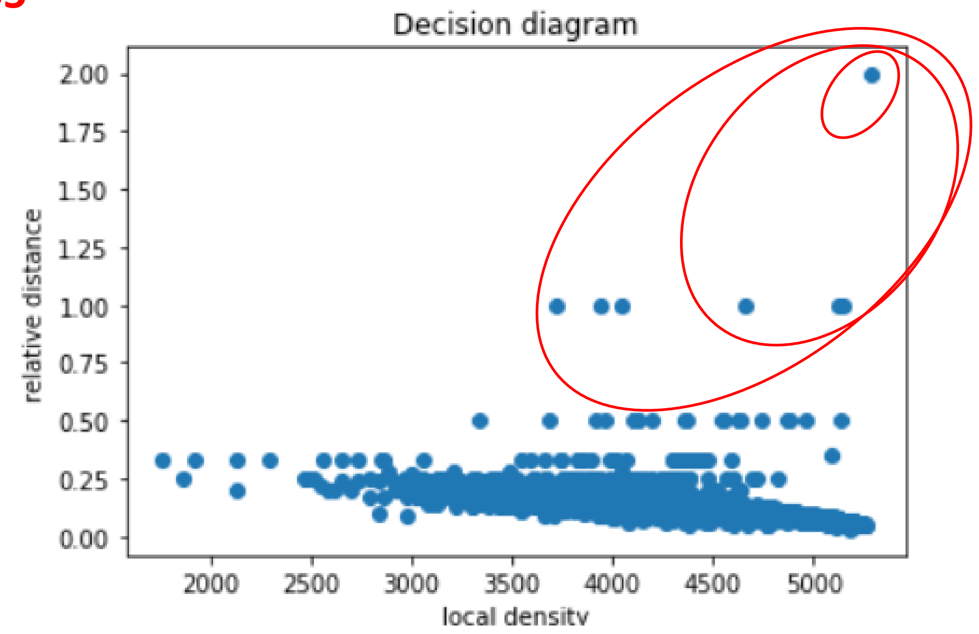
Our method has two parts of hyper-parameters to be determined:

1. **We filter out movies with insufficient information, which means movies with small HELF value. When grouping all users, we will select the appropriate number of groups according to Decision Diagram.**

Set HELF threshold to 0.65

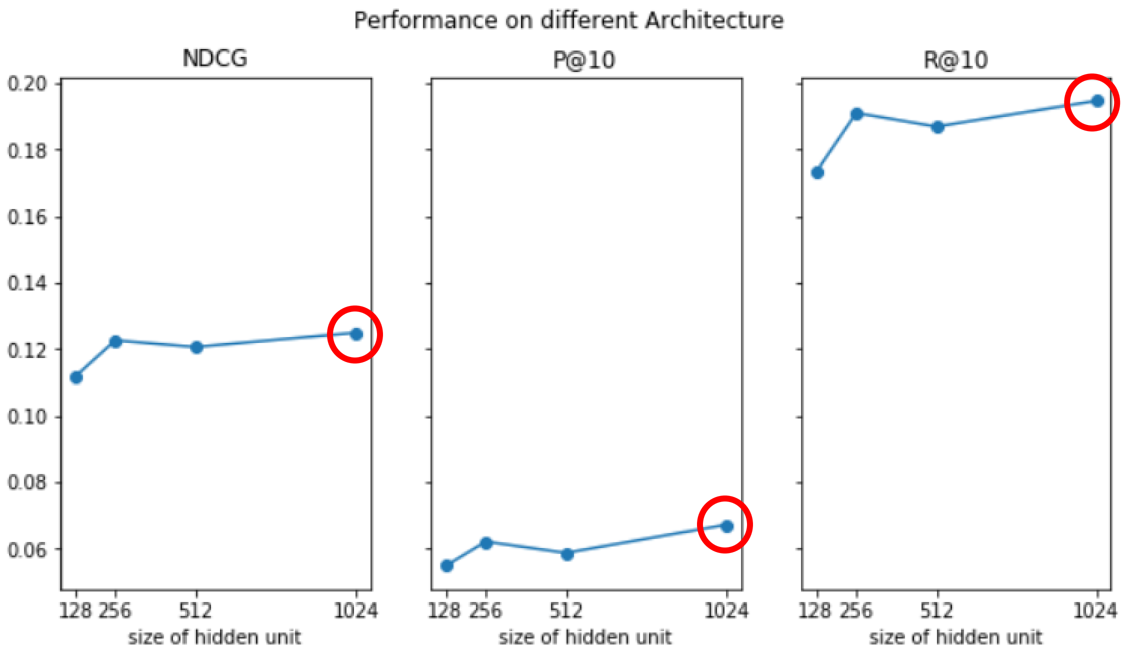


We will choose 1, 4, 7 clusters in our experiment



Experiment Settings (Hyper-parameters)

2. Architecture of the Autoencoder model, and the dropout ratio of representative movies and non-representative dropout ratio. Eventually, **we found that the low dropout rate of representative movies and the high dropout rate of non-representative movies can get better results.**



Choose 1024 as our hidden dimension for Autoencoder

Evaluation Metrics (NDCG@k)

1. **NDCG is equal to DCG divided by IDCG.**

$$\mathrm{NDCG@k} = \frac{DCG}{IDCG}$$

DCG: used to measure the relevance of an item based on its position in the recommendation list.

$$\mathrm{DCG@k} = \frac{1}{k}\sum_{j=1}^{k} \frac{rel_j}{\log(1+j)}$$

IDCG: is the maximum DCG score that can be obtained when the recommendation system can perfectly sort the recommendation results.

Top 3 recommendation: IDCG Ground Truth: [5,3,1] $\frac{1}{3}\left(\frac{5}{\log_2(1+1)} + \frac{3}{\log_2(1+2)} + \frac{1}{\log_2(1+3)}\right) = 2.4642$

DCG Recommend: [1,3,5] $\frac{1}{3}\left(\frac{1}{\log_2(1+1)} + \frac{3}{\log_2(1+2)} + \frac{5}{\log_2(1+3)}\right) = 1.7975$

Evaluation Metrics (precision@k & recall@k)

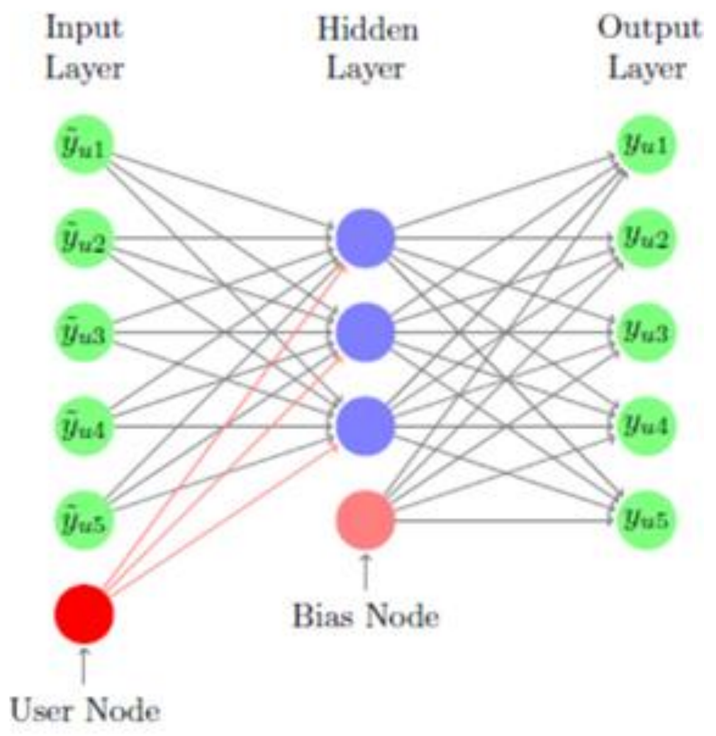
In the context of the recommendation system, we will divide the movie into two categories according to the rating given by the user. If a movie's rating is higher than the user's average rating, we will regard the movie as the user's favorite movie; otherwise, we will regard it as a movie the user don't like. Then, we can use the below formula to calculate **precision@k** and **recall@k** :

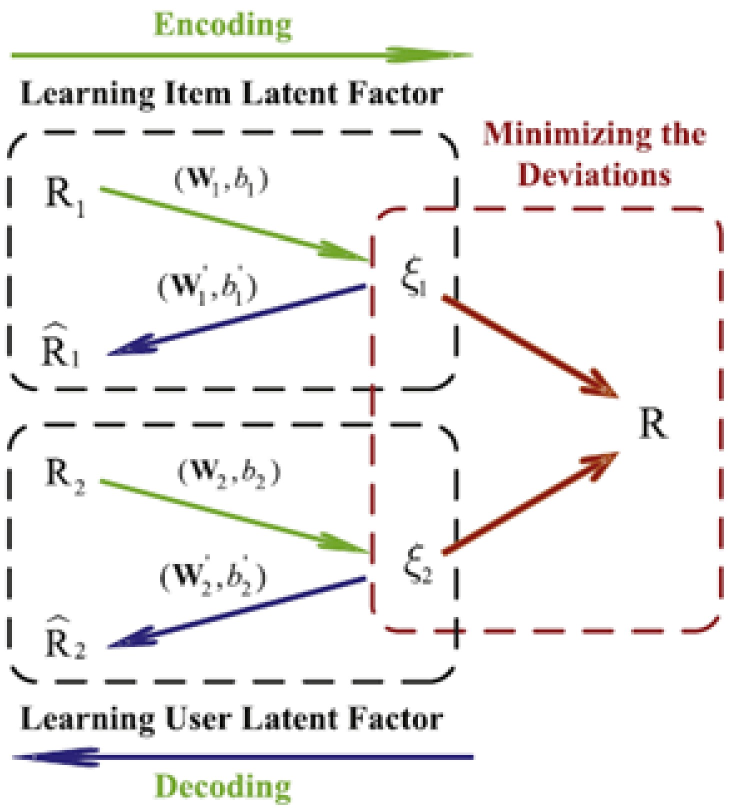
$$\mathrm{P@k} = \frac{num\ of\ recommended\ items\ at\ k\ that\ user\ likes}{k}$$

$$\mathrm{R@k} = \frac{num\ of\ recommended\ items\ at\ k\ that\ user\ likes}{total\ num\ of\ items\ that\ user\ likes}$$

All model performance comparison

- Autoencoder
- Autoencoder + random mask noise
- CDAE
- Dual Autoencoder





All model performance comparison

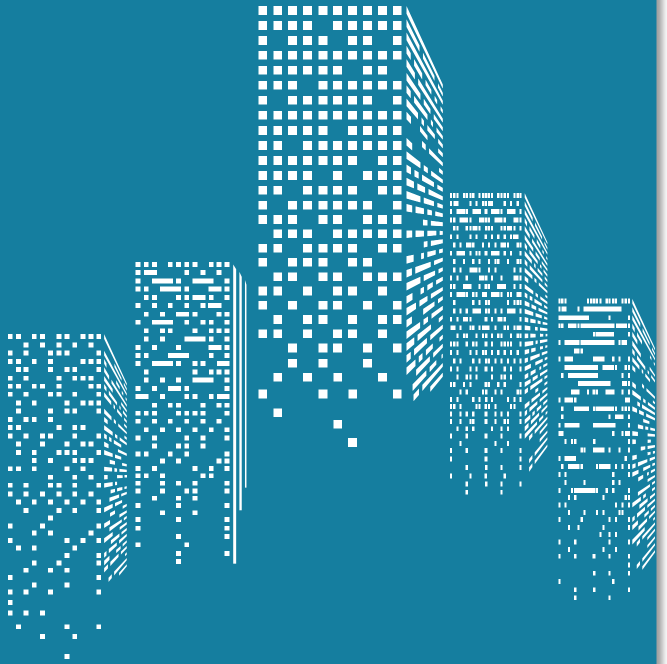
Models	NDCG@10	P@10	R@10
Autoencoder	0.1196**	0.0573	0.1934**
Random Autoencoder	0.1087**	0.0510**	0.1754**
CDAE	0.1120**	0.0493**	0.1865**
Dual Autoencoder	0.0875**	0.0395**	0.1457**
Our Method	0.1321	0.0689	0.2029

Analysis:

- It seems that Autoencoders with random masking noise, like Random Autoencoder and CDAE, perform not well.
  - The reason may be that random masking noise is not suitable for training Autoencoder in the context of recommendation system.
- Dual Autoencoder has the worst performance.
  - The reason may be that the output of the Dual Autoencoder simply do a inner product which cannot effectively capture the non-linear relationship between the user and the item.

5. Conclusion & Future Work

- Conclusion
- Future Work



5 Conclusion & Future Work

Our contributions

1. We are the first to propose the concept that combines noise generation and Denoising Autoencoder to solve the cold start problem in the field of recommendation system.

2. Our “user rejuvenation” is helpful for training the Denoising Autoencoder.

Future Work

1. Try the more complex model architecture to increase the capacity of the model.

2. Experiment on other dataset, not just on the MovieLen 1M.

2020.06

Thanks for listening !