Recommendation System Review

95729: E-Commerce Tech Individual Project

MISM-BIDA Jiefei Xia

12/5/2019

Agenda

- Basket Analysis
- Recommendation Problem
- Collaborative Filtering
- Negative Matrix Factorization
- Boltzmann Machine
- Practice Tips

Before the recommendation system
Basket Analysis

- Data: Order – Item List
- Association rule

	Apriori	FP Growth
Data Structure	Apriori property	Frequency Pattern Tree
Algorithm	Breadth first search	Divide and conquer
Memory	Large	Small
Time Complexity[1]	$O(Tk + (1 - m^T)/(1 - m))$	$O(m^2)$

- Problem:
  - More personalized?
  - Basket level -> User level

Order ID	Item	User ID?
o_1	Banana, Apple	c_1
o_2	Beer, Diaper, Cookie	c_3
o_3	Beer, Chips	c_2
o_4	Potato, Tomato, Egg	c_1
o_5	Orange juice, Beer, Diaper	c_7
…	…	…
o_k	…	c_n

Note1: Suppose the number of input transactions is k, the threshold is T, number of unique elements is m.

Recommendation Problem

Aggerate Data to User -- Item

- User c (row) + profile
- Item s (column) + profile
- Utility u (value): View/Purchase/Review

	Item ID	s_1	s_2	s_3	…	s_m
User ID		p_{s_1}	p_{s_2}	p_{s_3}	…	p_{s_m}
c_1	p_{c_1}	u_{11}	u_{12}	u_{13}	…	u_{1m}
c_2	p_{c_2}	u_{21}	u_{22}	u_{23}	…	u_{2m}
c_3	p_{c_3}	u_{31}	u_{32}	u_{33}	…	u_{3m}
…	…	…	…	…	…	…
c_n	p_{c_n}	u_{n1}	u_{n2}	u_{n3}	…	u_{nm}

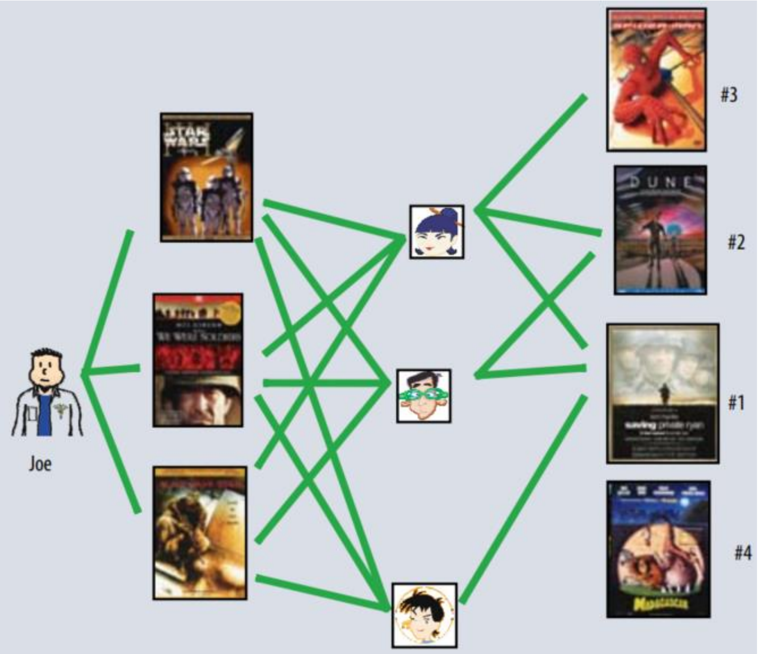
Introduction

- Some utility might be missing X…
- **Predict the missing utility**
- Given the very sparse matrix (Lots of 0)

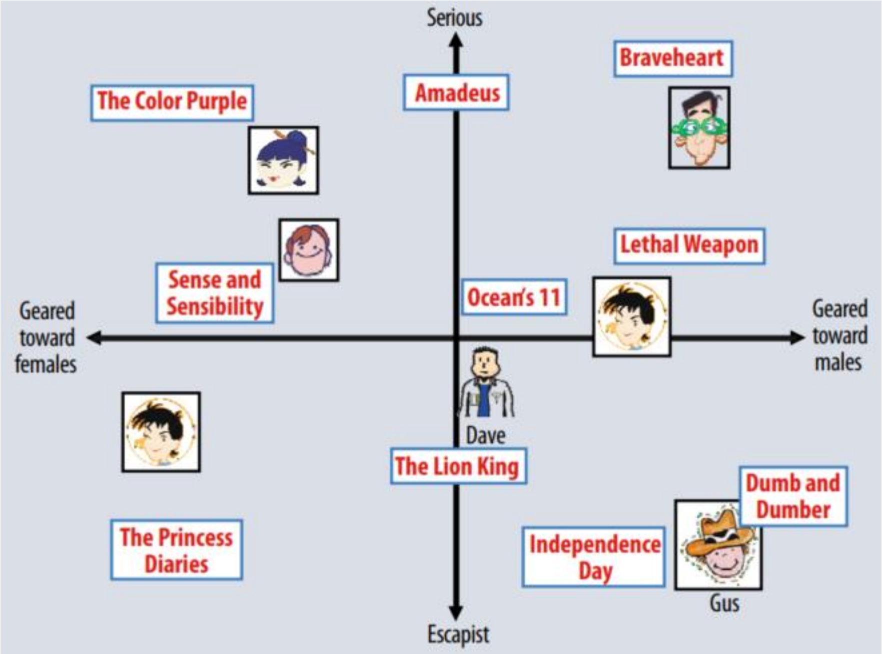
	Item ID	s_1	s_2	s_3	…	s_m
User ID		p_{s_1}	p_{s_2}	p_{s_3}	…	p_{s_m}
c_1	p_{c_1}	0	X	u_{13}	…	0
c_2	p_{c_2}	u_{21}	0	0	…	u_{2m}
c_3	p_{c_3}	0	u_{32}	X	…	0
…	…	…	…	…	…	…
c_n	p_{c_n}	u_{n1}	0	u_{n3}	…	u_{nm}

Collaborative filtering

- Neighborhood method: KNN, Cosine Similarity
- Latent factor model: Matrix factorization



Neighborhood Method



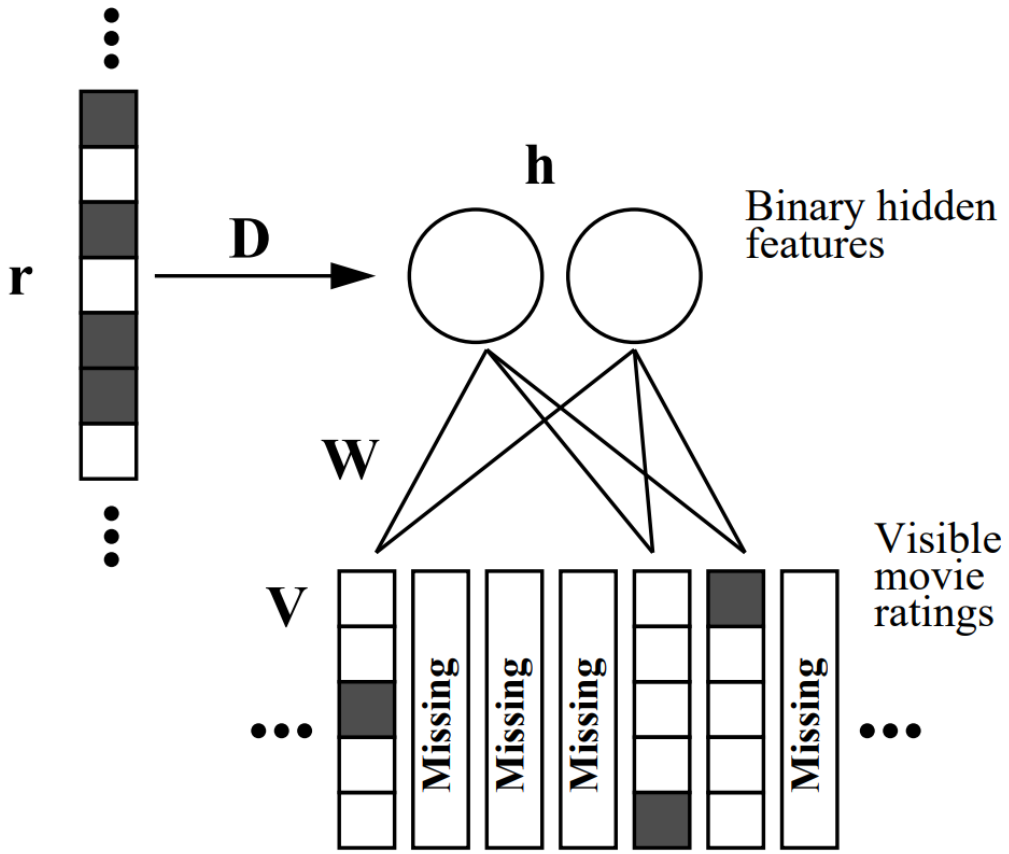
Latent Factor Model

Negative Matrix Factorization

- Map both users and items to a joint latent factor
  - $q_s \in f$ measure the extent to which the item s possesses those factors.
  - $q_c \in f$ measure the extent of interest the user c has in items that are high on the corresponding factors.
- Approximates user c's utility of item s
  - $\widehat{u_{cs}} = q_c^T \cdot p_u$

Restricted Boltzmann Machine

- RBM
  - Learn a representation in hidden units
- Modification from Salakhutdinov
  - Contrastive Divergence to speed up
  - Use a special Neuron to represent missing value
  - Use SoftMax to get probability of rating



Experiment

Dataset: Instacart grocery data (https://www.instacart.com/datasets/grocery-shopping-2017)
Size: 3.4m orders, 206k users, 50k products.

Algorithm	Train Time[1]	Test Time	Performance[2]
Apriori	28 min	3 min	0.08 hit
FP Growth	5 min	3 min	0.08 hit
KNN (Cosine Similarity)	17 min	12 min	0.007 MAE
Singular Value Decomposition	Not feasible on sparse matrix		
Negative Matrix Factorization	8 min	≈0	0.006 MAE
Restricted Boltzmann Machine	21 min	≈0	0.003 MAE

NOTE1: Experiment is run on AWS P3xlarge computing instance.
NOTE2: Performance is measured on test dataset.

Insights

- Objective definition is important in real machine learning problem.
- Use small math modification to fit new algorithm to new application
  - E.g. (RBM + SoftMax) -> Recommendation System = $1M (Netflix Prize)
- Offline computation is preferred in real case.
- Data Science spend 80% of time in preparing data

Thank you!
Q&A

Star https://github.com/jiefeixia/recommendation_system_review

Reference

- Adomavicius, G. &. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge & Data Engineering, 734-749.
- J.B. Schafer, J. K. (2001). E-Commerce Recommendation Applications. Data Mining and Knowledge Discovery, 115-153.
- Koren, Y. R. (2009). Matrix factorization techniques for recommender systems. pp. 30-37.
- Salakhutdinov, R. A. (2007). Restricted Boltzmann machines for collaborative filtering. Proceedings of the 24th international conference on Machine learning. ACM.