TriRank: Review-aware Explainable Recommendation by Modeling Aspects

Xiangnan He, Tao Chen, Min-Yen Kan, Xiao Chen

National University of Singapore

Presented by Xiangnan He
CIKM'15, Melbourne, Australia
Oct 22 2015

Recommender System – Multifaceted

- Accuracy
- Scalability
- Explainability
- Transparency
- Scrutability
- Online learning
- Privacy
- Diversity

……

Increase Users' **Trust & Satisfaction**

- Collaborative Filtering
  - Model-based
  - Memory-based
  - Graph-based
- Content Filtering
- Context-aware
  - Social
  - Temporal
  - Reviews

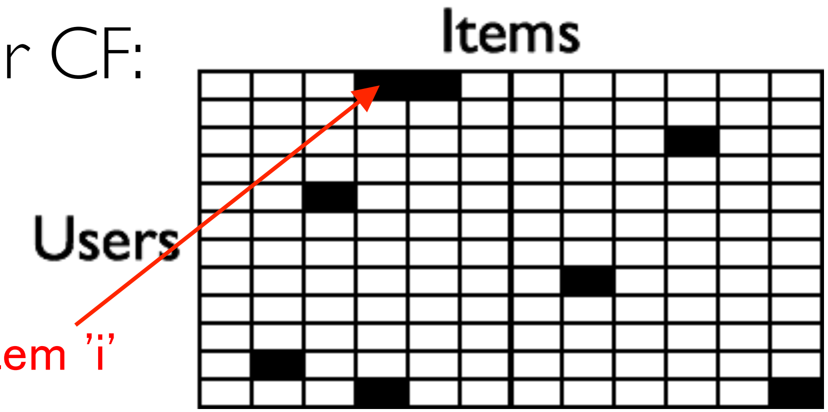
  ……
- Hybrid

Recap: Collaborative Filtering

- Predict the preference of a user by the similar users.
- Focus on the user-item feedback matrix.

E.g. matrix factorization model for CF:

Input: Given a sparse user–item feedback **matrix:**



User 'u' bought item 'i'

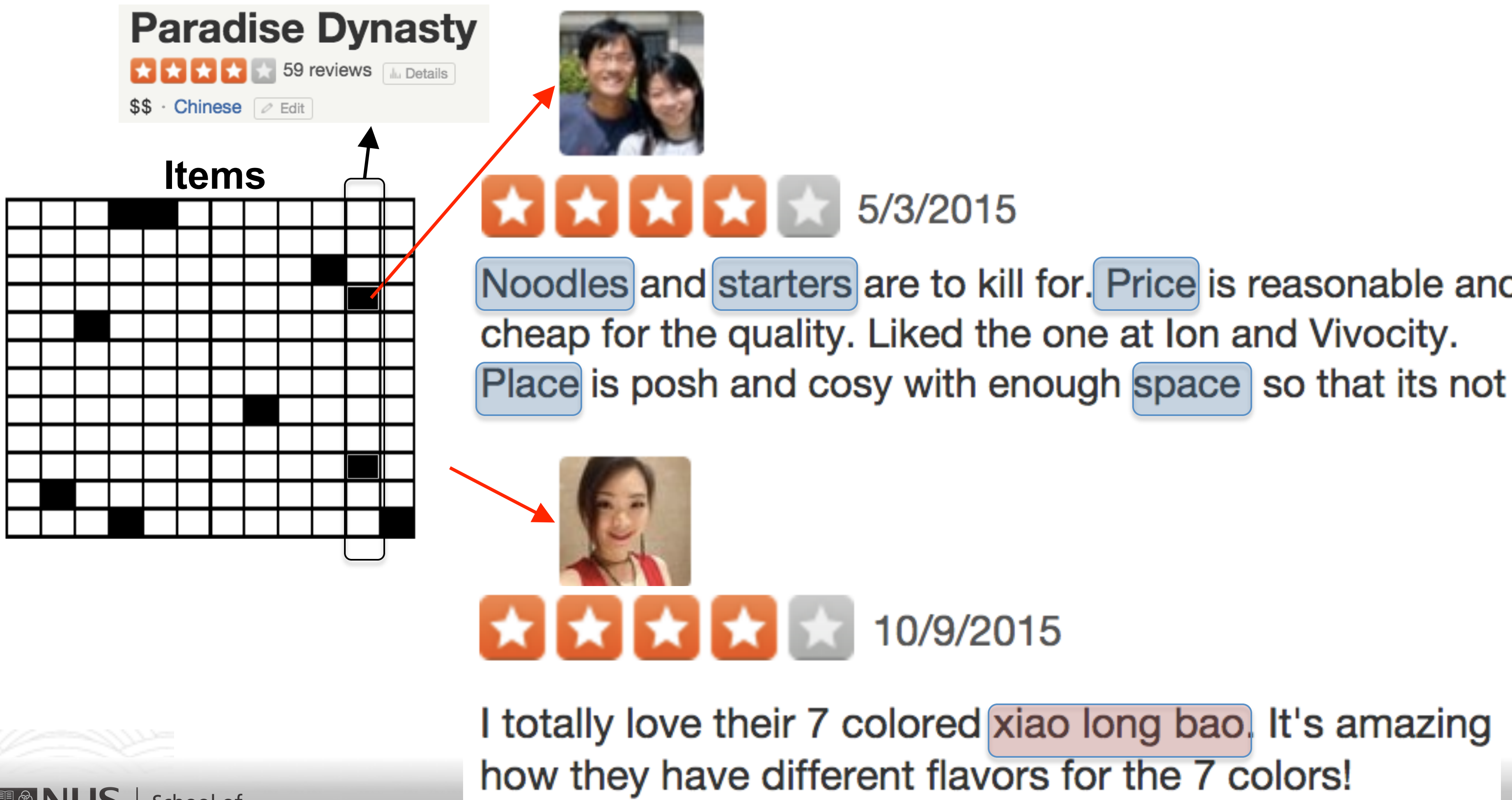
$1 \times K$

Learn latent vector **for each user, item:** v_u^U v_i^I

Affinity between user 'u' and item 'i': $\hat{y}_{ui} = < v_u, v_i >$

Main Limitation of CF

Hard to infer the actual rationale from the rating score only!



Example: Dilemma of CF

Inputs:
<u1, p1, 5>
<u2, p1, 5>
<u2, p2, 4>
<u3, p1, 5>
<u3, p3, 4>
<u4, p3, 4>
<u4, p4, 5>

Inputs (aspects):
<u1, p1, 5, seafood>
<u2, p1, 5, chicken>
<u2, p2, 4, chicken>
<u3, p1, 5, seafood>
<u3, p3, 4, seafood>
<u4, p3, 4, seafood>
<u4, p4, 5, seafood>

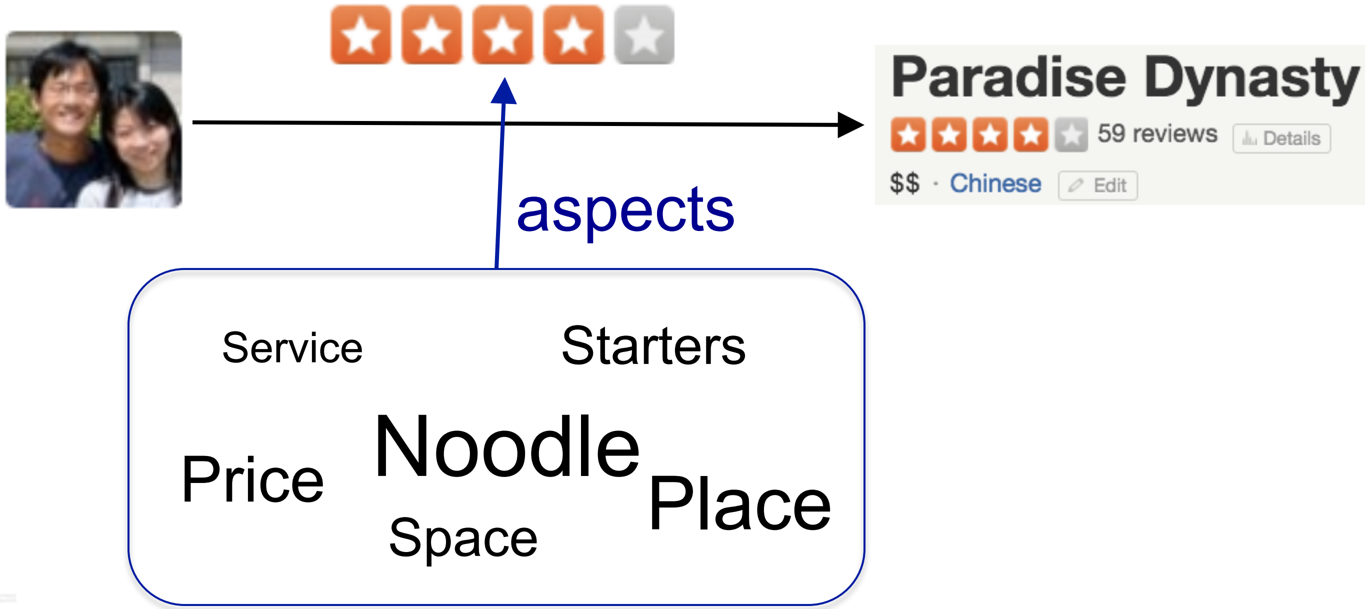
	p1	p2	p3	p4
u1	5	0	0	0
u2	5	4	0	0
u3	5	0	4	0
u4	0	0	4	5

Neighbors u2 and u3 have equal preference on p2 and p3

CF can not choose between p2 and p3!

Review-aware Recommendation

- Reviews justify a user’s rating:
  - by discussing the specific properties of items (aspects);
  - by revealing which aspects the user is most interested in.

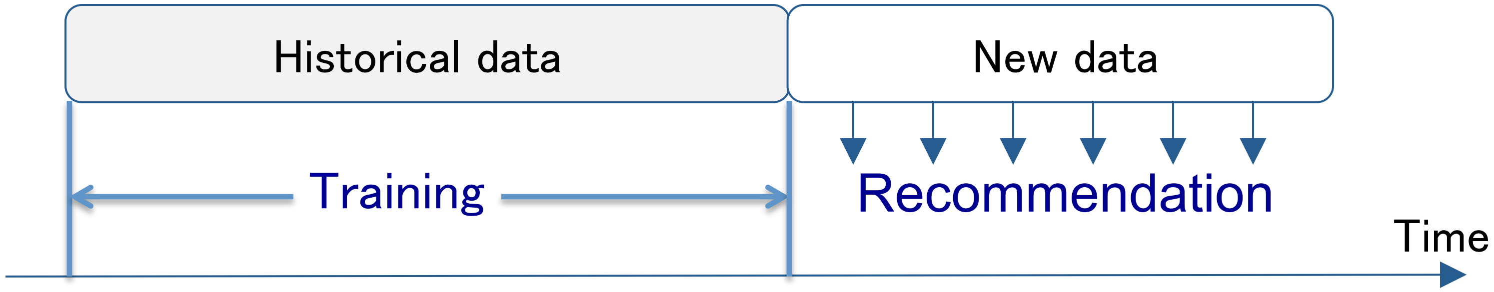


Existing Works

- Topic models on words + item latent factors:
  - McAuley and Leskovec, Recsys'13: LDA + MF
  - Ling *etc*, Recsys'14: LDA + PMF (full Bayesian treatment)
  - Xu *etc*, CIKM'14: LDA + PMF + user clusters (full Bayesian)
  - Bao *etc*, AAAI'14: NMF + MF
- Joint modeling of aspects and ratings:
  - Diao *etc*, KDD'14: graphical model
  - Zhang *etc*, SIGIR'14: collective NMF
  - Musat *etc*, IJCAI'13: build user topical profiles

Limitations of previous works

- Focused on rating prediction.
  - Top-K recommendation is more practical.
- Lack explainability and transparency.
  - Well-known drawback of latent factor model.
- Do not support online learning (instant personalization).
  - New data comes in (retraining is expensive).
  - User updates his/her preference (scrutability).



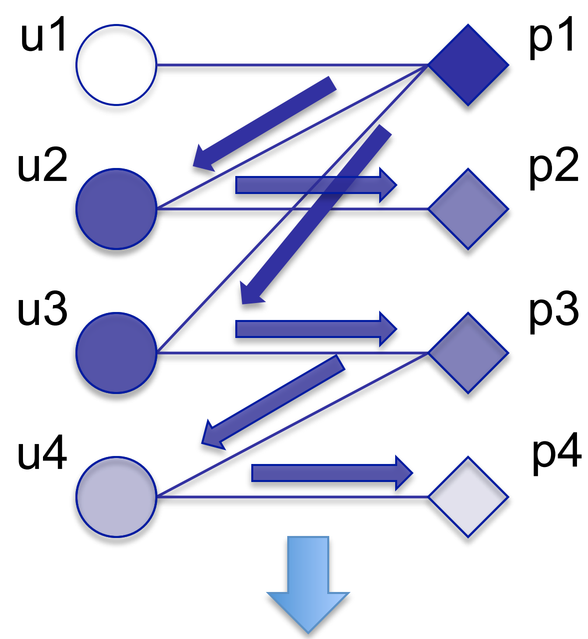
Our Solution - TriRank

- ✓ Review-aware recommendation.
- ✓ Graph-based method.
  - Top-K recommendation ➔ Vertex ranking.
- ✓ Good accuracy.
- ✓ Explainable.
- ✓ Transparent.
- ✓ Offline training + online learning.
  - Provide instant personalization without retraining.

Basic Idea: Graph Propagation

Inputs:
<u1, p1, 1>
<u2, p1, 1>
<u2, p2, 1>
<u3, p1, 1>
<u3, p3, 1>
<u4, p3, 1>
<u4, p4, 1>

Target user: u1



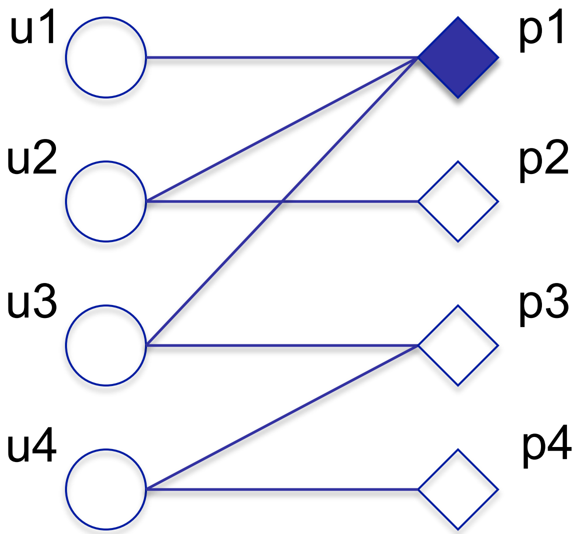
Item ranking: p2 ≈ p3 > p4
User ranking: u2 ≈ u3 > u4

Label propagation from the target user's historical item nodes captures the collaborative filtering.

How to encode that mathematically?

Machine Learning for Graph Propagation (Graph Regularization)

[He etc, SIGIR 2014]



Input:
- Graph structure (matrix *Y*)
- Initial labels to propagate (vectors p^0)

Output:
- Scores for each vertex (vectors *u*, *p*)

Smoothness kernel (propagation):

-Nearby vertices should not vary too much:

$$\sum_{i \in U} \sum_{j \in P} y_{ij} \left(\frac{u_i}{\sqrt{d_i}} - \frac{p_j}{\sqrt{d_j}}\right)^2$$

Fitting constraint (initial labels):

-Ranking scores should adhere to the initial labels:

$$\sum_{j \in P} (p_j - p_j^0)^2$$

Optimization (coordinate descent):

$$\mathbf{p} = S_Y \mathbf{u} + \mathbf{p}^0$$

$$\mathbf{u} = S_Y^T \mathbf{p}, \quad \text{where } S_Y = \left[\frac{y_{ui}}{\sqrt{d_u d_i}}\right]$$

, which exactly mimic the propagation process!

X. He, M. Gao, M.-Y. Kan, Y. Liu, and K. Sugiyama. Predicting the popularity of web 2.0 items based on user comments. In *Proc. SIGIR '14*

Connection to CF models

- Recap: ranking loss function (for a target user):

$$\boxed{\sum_{j\in P}(p_j - p_j^0)^2} + \lambda \sum_{i\in U}\sum_{j\in P} y_{ij}\left(\frac{u_i}{\sqrt{d_i}} - \frac{p_j}{\sqrt{d_j}}\right)^2$$

Prediction loss Regularizations

- Traditional machine learning-based CF models:
  1. Prediction model:

     E.g., matrix factorization: $\hat{y}_{ui} = < v_u, v_i >$
  2. Loss function:

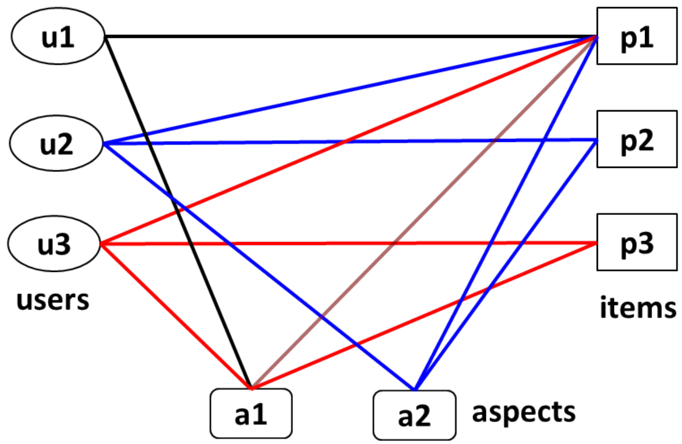
$$\sum_{u\in U}\boxed{\sum_{i\in I}(y_{ui} - \hat{y}_{ui})^2}$$

Prediction loss on all items (include imputations).
(important for top-K recommendation)

TriRank Solution

- Graph propagation in the tripartite graph:

Inputs:
<u1, p1, a1>
<u2, p1, a1>
<u2, p2, a1>
<u3, p1, a2>
<u3, p3, a2>



$$\mathbf{u} = \mathbf{u}^0 + \lambda_1 UP \cdot \mathbf{p} + \lambda_2 UA \cdot \mathbf{a}$$
$$\mathbf{p} = \mathbf{p}^0 + \lambda_3 PU \cdot \mathbf{u} + \lambda_4 PA \cdot \mathbf{a}$$
$$\mathbf{a} = \mathbf{a}^0 + \lambda_5 AP \cdot \mathbf{p} + \lambda_6 AU \cdot \mathbf{u}$$

Initial labels should encode:
- Target user's preference on aspects/items/users:
  - a_0: reviewed aspects.
  - p_0: ratings on items.
  - u_0: similarity with other users (friendship).

Online Learning

- Offline Training:
  1. Extract aspects from user reviews
  2. Build the tripartite graph model (edge weights)
  3. Label propagation from each vertex and save the scores.
     - *i.e.* store a $|V|\times|V|$ matrix $\mathbf{f}(\mathbf{v_i}, \mathbf{v_j})$.

     (to save space, we can save top scores for each vertex)
- Online Learning (new data and updated preference applies):
  1. Build user profile (*i.e.*, $\mathbf{L_u}$ vertices to propagate from).
  2. Average the scores of the $\mathbf{L_u}$ vertices:

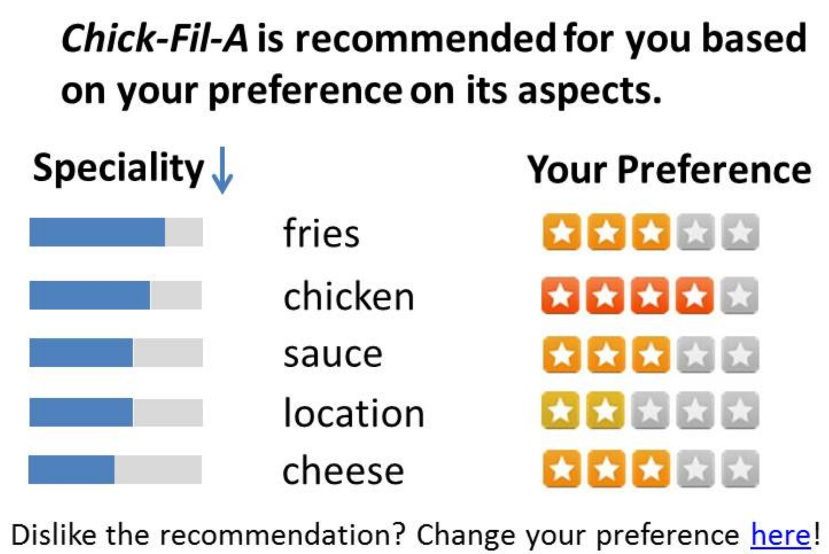
$$y_j = \frac{1}{|L_u|} \sum_{v_u \in L_u} f(v_u, v_j)$$

Complexity: $O(\mathbf{L_u})$, almost constant!

Explainability

- Transparency:
  - Collaborative filtering + Aspect filtering ➔ Item Ranking / Aspect Ranking / User Ranking

    (Similar users also choose the item) (Reviewed aspects match with the item)
  - An example of reasoned recommendation:



Experimental Settings

- Public datasets (filtering threshold at 10):
  - Yelp Challenge
  - Amazon electronics

Dataset	Review#	Item#	User#
Yelp	114,316	4,043	3,835
Amazon	55,677	14,370	2,933

- Sort reviews in chronological order for each user:
  - Split: 80% training + 10% validation + 10% test
- Top-K evaluation:
  - For each test user, we output K items as a ranking list:

  Recall-based measure: $$Hit\ Ratio = \frac{\#hits@K}{|Test|}$$

  Ranking-based measure: $$NDCG = \sum_{i=1}^{K} \frac{2^{r_i} - 1}{log_2(i+1)}$$

Aspect Extraction

- A well studied task in review mining [survey: Zhang and Liu, 2014]:
  - Unsupervised rule-based methods:
    - [Hu and Liu, KDD'04; Zhang *etc.* COLING'10]: phrase/sentence patterns.
  - Supervised sequence labeling methods:
    - [Jin and Ho, ICML'09; Jakob etc. EMNLP'10]: HMM, CRF …
- We adopt a tool developed by Tsinghua IR group

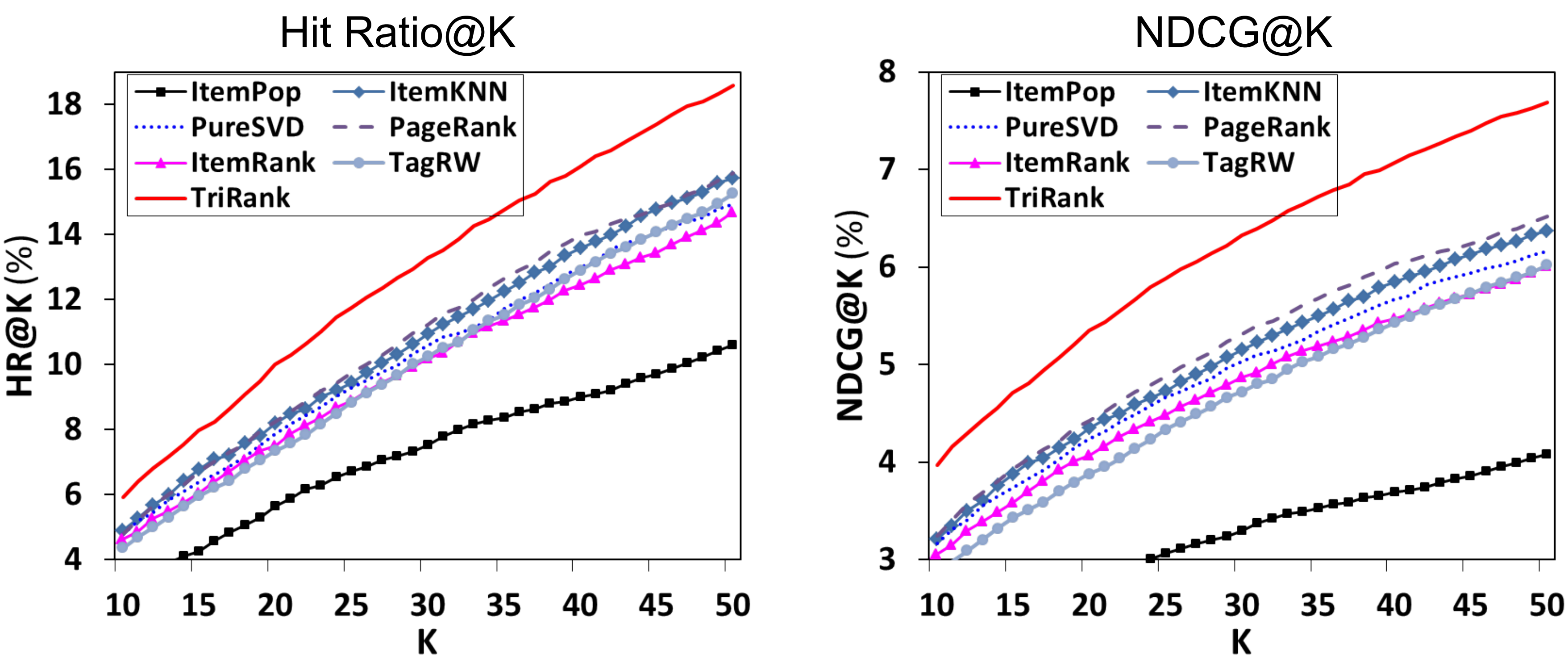
[Zhang *etc.* SIGIR'14]: rule-based system:

Dataset	#Aspect	Density (U-A)	Density (I-A)	Top aspects (good examples)	Noisy aspects
Yelp	6,025	3.05%	2.29%	bar, salad, chicken, sauce, cheese, fries, bread, sandwich	restaurants, food, ive (I've), 150
Amazon	1,617	3.80%	1.44%	camera, quality, sound, price, battery, screen, size, lens	product, features, picturemy

Baselines

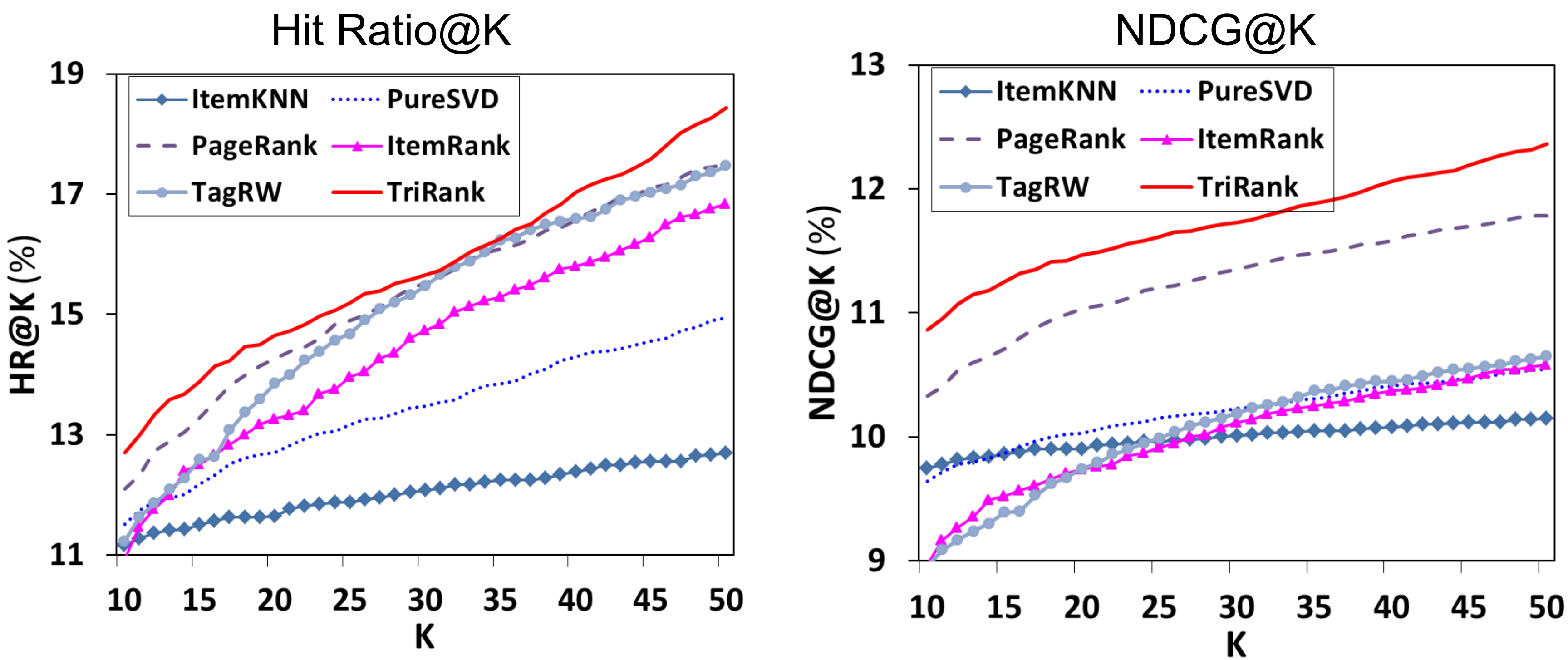
- Item Popularity (ItemPop)
- ItemKNN [Sarwar *etc.* 2001]
  - Item-based collaborative filtering
- PureSVD [Cremonesi *etc.* 2010]
  - Matrix factorization with imputations
  - Best factor number is 30. Large factors lead to overfitting.
- PageRank [Haveliwala *etc.* 2002]
  - Personalized with user preference vector
- ItemRank [Gori *etc.* 2007]
  - Personalized PageRank on item-item correlation graph
- TagRW [Zhang *etc.* 2013]
  - Integrate tags by converting to user-user and item-item graph.

Yelp Results



Hit Ratio (recall): TriRank > PageRank > ItemKNN > TagRW > PureSVD > ItemRank
NDCG (ranking): TriRank > PageRank > ItemKNN > PureSVD > ItemRank > TagRW

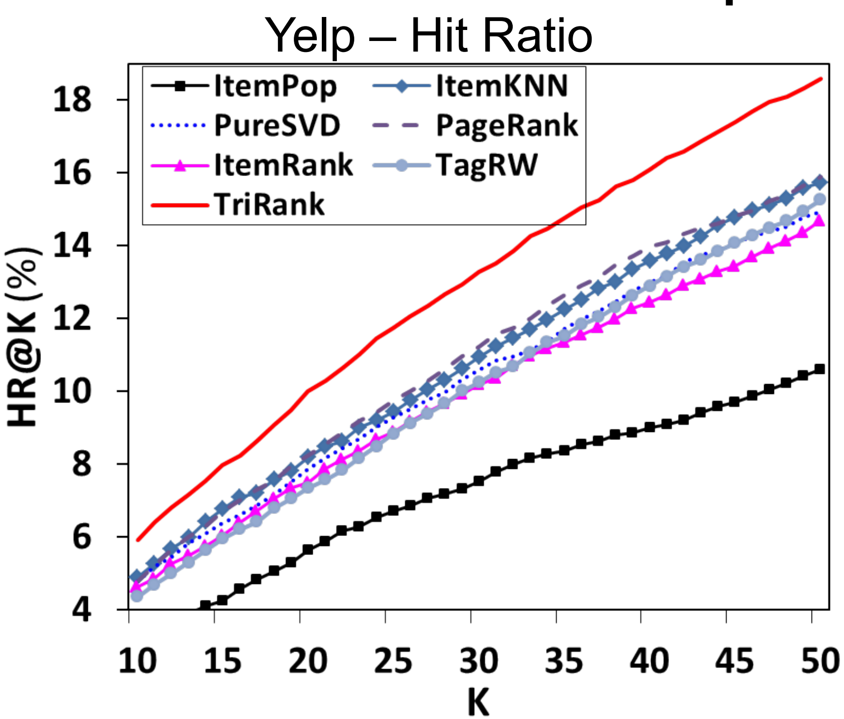
Amazon Results

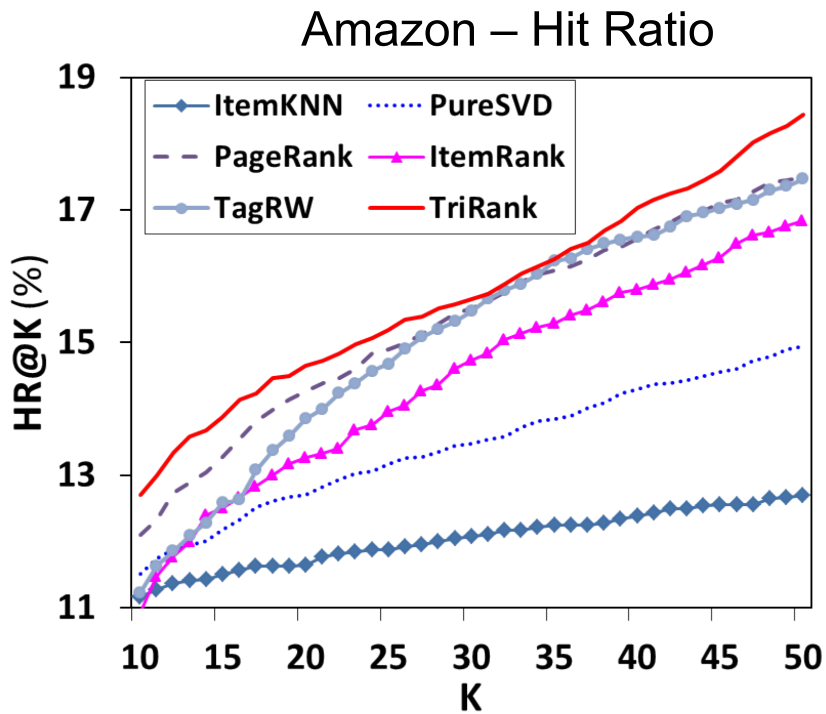


The discrepancy between HR and NDCG is more obvious:

- TagRW is strong for HR, but weak for NDCG;

Yelp VS Amazon





1. ItemKNN is strong for Yelp, but weak for Amazon

- Amazon dataset is more sparse (#reviews/item: 28 vs 4)

2. PageRank performs better than ItemRank (both are Personalized PageRank)

- Converting user-item graph to item-item graph leads to signal loss.

Utility of Aspects

Dataset	Yelp		Amazon	
Settings (@50)	HR	NDCG	HR	NDCG
All Set	**18.58**	**7.69**	**18.44**	**12.36**
No item-aspect	17.05	6.91	16.23	11.31
No user-aspect	**18.52**	**7.68**	**18.40**	**12.36**
No aspects	17.00	6.90	15.97	11.16
No user-item	11.67	4.84	10.32	5.08

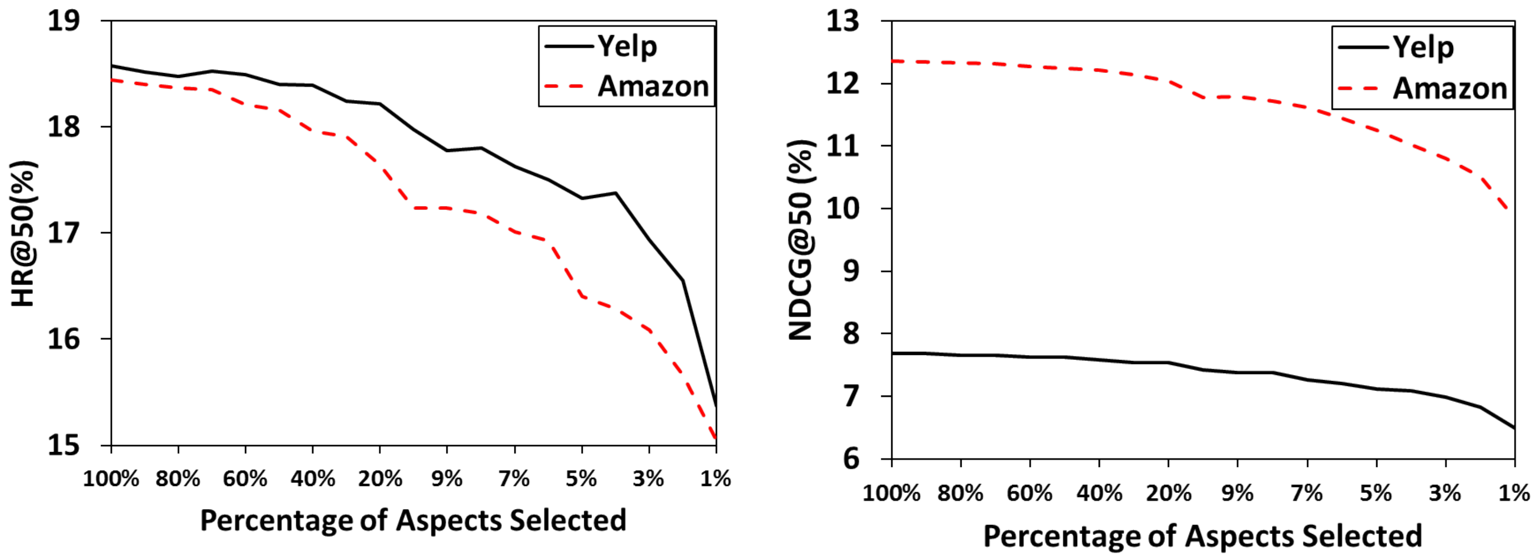
1. **Item-aspect** relation is more important than **user-aspect** relation.

2. **Aspects filtering** is complementary to collaborative filtering.

3. **User-item** relation is still fundamental to model and most important!

Aspect Filtering

- How does the noisy aspects impact the performance?
  - Ranking aspects by their TF-IDF score in item-aspect matrix.



Insensitivity to noisy aspects:

- Filtering out low TF-IDF aspects (e.g. stop words or quirks) do not improve.

High TF-IDF aspects carry more useful signal for recommendation.

- Filtering out high TF-IDF aspects hurt performance significantly.

Case Study

Training reviews of a sampled Yelp user.

★★★☆☆ 20/11/2012

Basically it was was grilled chicken with a few green onions and sesame seeds. Teriyaki with no teriyaki sauce? Strange.

★★☆☆☆ 18/10/2012

Unfortunately, find my picture and see that I'm reviewing the food and wait time. It was a 15-20 minute wait for two chicken strip baskets.

★★★★☆ 13/7/2012

This is usually my take out place of choice. It's quick, inexpensive, close, and delicious. I usually get the shrimp lo mein.

★★★★☆ 11/7/2011

I'm still breaking in my sushi palate, but I'll still review the place as I see it. Happy hour specials make my addiction to their tempura shrimp a little easier on the wallet!

Rank list by TriRank:

. . .

3rd: *Red Lobster*

. . .

6th: *Chick-Fil-A*

. . .

Although the test set doesn't contain *Red Lobster*, we found she actually reviewed it later. (outside of the Yelp dataset)

Conclusion

- Tripartite graph ranking solution for review-aware recommendation:
  - Explainable and transparent
  - Robust to noisy aspects
  - Online learning and instant personalization without retraining.
- Future work:
  - Combine with factorization model (more effective to sparse data)
  - Personalized (regularization) parameter settings
  - More contexts to model: temporal, taxonomy and sentiment.

Thank you!

Thank SIGIR Student Travel Grant!

Reference

X. He, M. Gao, M.-Y. Kan, Y. Liu, and K. Sugiyama. Predicting the popularity of web 2.0 items based on user comments. In Proc. SIGIR '14, pages 233–242, 2014.

J. McAuley and J. Leskovec. Hidden factors and hidden topics: Understanding rating dimensions with review text. In Proc. of RecSys'13, pages 165–172, 2013.

G. Ling, M. R. Lyu, and I. King. Ratings meet reviews, a combined approach to recommend. In Proc. of RecSys '14, pages 105–112, 2014.

Y. Xu, W. Lam, and T. Lin. Collaborative filtering incorporating review text and co-clusters of hidden user communities and item groups. In Proc. of CIKM '14, pages 251–260, 2014.

Q. Diao, M. Qiu, C.-Y. Wu, A. J. Smola, J. Jiang, and C. Wang. Jointly modeling aspects, ratings and sentiments for movie recommendation (jmars). In Proc. of KDD '14, pages 193–202, 2014.

Y. Zhang, M. Zhang, Y. Zhang, Y. Liu, and S. Ma. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In Proc. of SIGIR '14, pages 83–92, 2014.

C.-C. Musat, Y. Liang, and B. Faltings. Recommendation using textual opinions. In Proc. of IJCAI '13, pages 2684–2690, 2013.