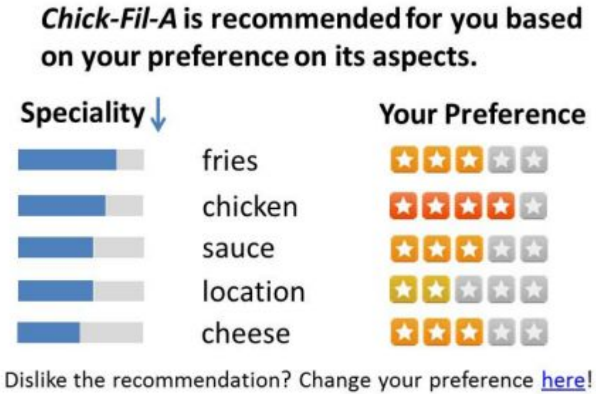
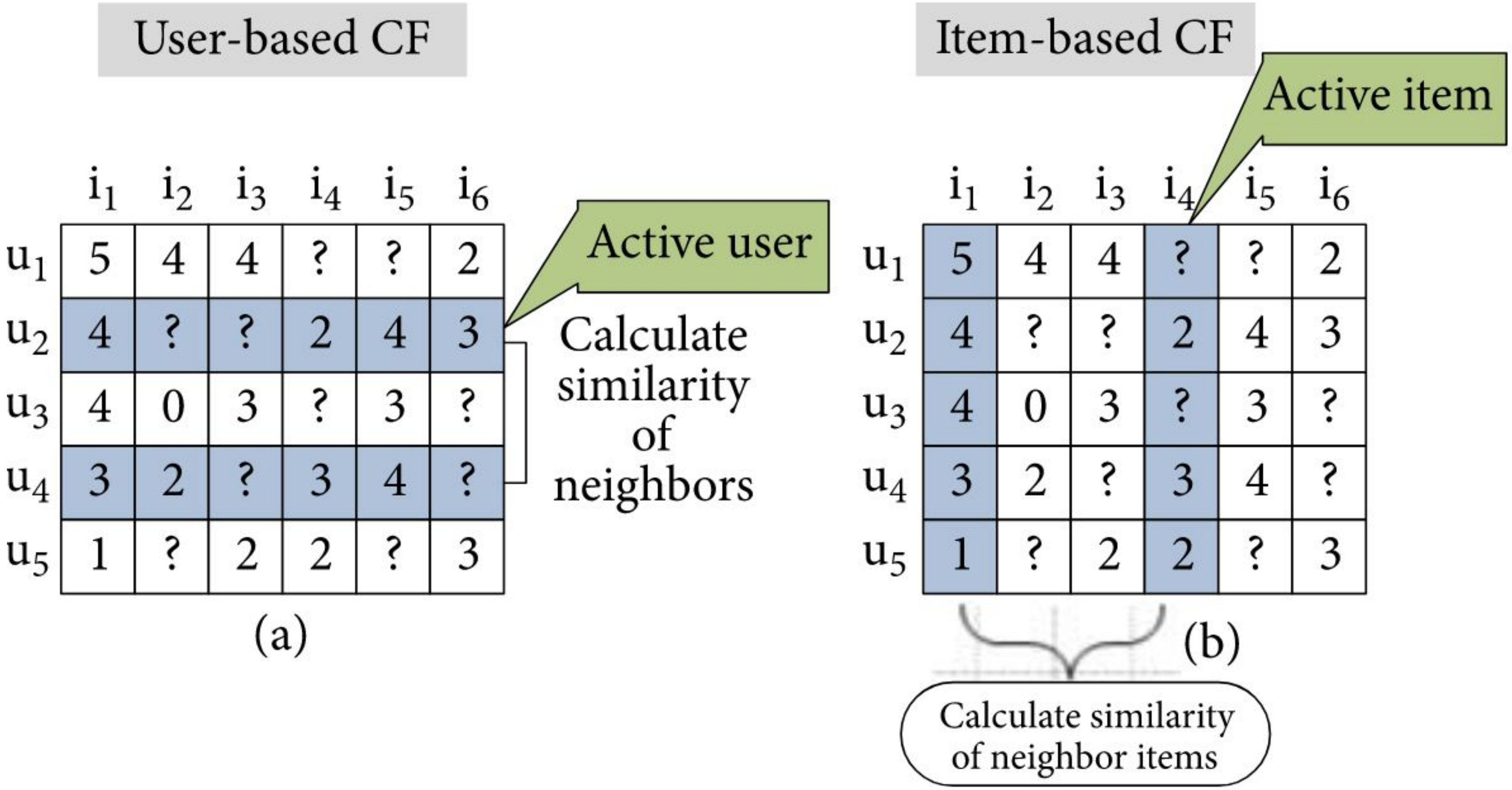
TriRank: Review-aware Explainable Recommendation by Modeling Aspects

Daniel Wang, Theo Rose, Rajiv Sarvepalli
University of Virginia - Information Retrieval CS 4780



Traditional Collaborative Filtering



3/9/2021

2 photos

This spot is a great place to grab a healthy and filling salad near Grounds. The service is friendly and fast. The prices are high and the portions can be different for each visit, but overall it's a great spot to eat.

10/16/2020

The Southern Bowl with BBQ tofu is the best vegetarian dish in C'Ville, full stop. The tofu is cooked to perfection, the BBQ sauce is great. I recommend substituting the tahini dressing for cilantro lime. My general experience with the Wertland pickup location has been good and has felt so safe during the pandemic.

Existing CF techniques only model user-rating, losing granularity on "aspects". Hard for the system to infer actual rationale for the rating.

Background in Review-aware Recommendation

- Aspect Extraction in Review Mining

- Prior Work:
    - Word-based: LDA + latent factor model (McAuley ‘13, Ling ‘14, Xu ‘14)
    - Sentiment-based: Rating + Sentiment in an integrated graphical model (Qiao ‘14)
    - Aspect-based: Factorized the user–item rating matrix by inserting aspects (Zhang ‘14)

- Limitations:
    - Rating Prediction rather than Top-K recommendation
    - Lack of support for online learning

      Retraining the system is often too expensive, and user cannot update preferences easily

Key Limitations of Recommender Systems

Feature-rich complementary data sourced left untapped by CF

10/22/2020

Eh, I've given it a few tries because I'm missing Sweetgreen and am looking for a replacement, but have been disappointed each time. You can barely taste the dressing (I had to search last time because I actually couldnt taste it at all and wondered if they forgot it). The chicken is super tough, I had to throw most of it away. I appreciate a healthy spot but its super mediocre, unfortunately.

Taste Replacement
Mediocre Tough

discusses properties justifying the rating known as aspects

Lack of explainability/transparency with LFM

Results in inability for users to scrutinize or interact with the recommendation system.

Existing Latent Factor Models (LFM) used on reviews provide opaque rationale with "single-shot" recommendation.

Graph Modeling

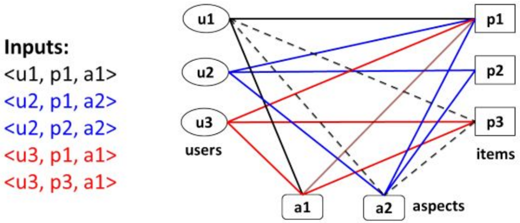
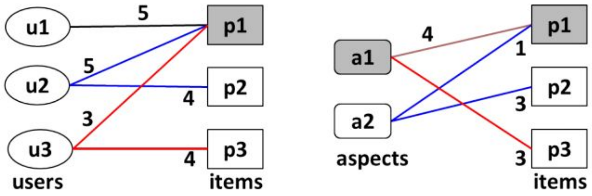


Figure 1: An example tripartite structure of the given inputs (the dashed line illustrates the additional input $< u_1, p_3, a_2 >$).



(a) User-Item structure (b) Item-Aspect structure

Figure 2: Smoothness constraints on decomposed graphs from Figure 1. Assume $u1$ previously rated item p_1 with mentioning aspect a_1 (shaded vertices).

1. Goal is, given a user, to map an item to real number representing that user's preference
2. Each input is a user mentioning an item with an aspect
    a. Triangular graph with weighted edges denoting the strength of the connection
3. Enforces smoothing and fitting constraints

Tripartite Graph Ranking Algorithm

The goal for item recommendation is to devise a ranking function $f : P \rightarrow R$, which maps each item in P to a real number such that the value reflects the target user u's (predicted) preference on the item

TriRank assigns the ranking score of vertices by enforcing the structural smoothness and fitting constraints of the graph.

Smoothness implies local consistency: that nearby vertices should not vary too much in their scores.

Fitting encodes prior belief: that the ranking function should not cause much deviation from the observations.

Since the regularization function is convex, it is **minimized using ALS** (alternate least squares). ALS is used as it does not need to set a learning rate, is easier to parallelize, and leads to faster convergence.

Semi-supervised learning process on graphs – with the prior preference as labeled data, the algorithm propagates the labels to other unlabeled vertices.

ML Graph Propagation via. Regularization Constraints

Combine the smoothness regularizer on each relation type with the fitting regularizer using different weights for each vertex type

α, β and γ: weight of smoothness on user–item, item–aspect and user–aspect relation

ηU , ηP and ηA: weight of fitting constraint on users, items and aspects

edge weight between u_i and p_j

final ranking scores

sum of edge weights for normalization

$$
\begin{aligned}
Q(f) = {} & \alpha \sum_{i,j} r_{ij} \left(\frac{f(u_i)}{\sqrt{d_i^u}} - \frac{f(p_j)}{\sqrt{d_j^p}} \right)^2 + \beta \sum_{j,k} x_{jk} \left(\frac{f(p_j)}{\sqrt{d_j^p}} - \frac{f(a_k)}{\sqrt{d_k^a}} \right)^2 \\
& + \gamma \sum_{i,k} y_{ik} \left(\frac{f(u_i)}{\sqrt{d_i^u}} - \frac{f(a_k)}{\sqrt{d_k^a}} \right)^2 + \eta_U \sum_i (f(u_i) - u_i^0)^2 \\
& + \eta_P \sum_j (f(p_j) - p_j^0)^2 + \eta_A \sum_k (f(a_k) - a_k^0)^2,
\end{aligned}
$$

Iterative Update Rules

$$\vec{u} = \frac{\alpha}{\alpha+\gamma+\eta_U} S_R \cdot \vec{p} + \frac{\gamma}{\alpha+\gamma+\eta_U} S_Y \cdot \vec{a} + \frac{\eta_U}{\alpha+\gamma+\eta_U} \vec{u}_0,$$

$$\vec{p} = \frac{\alpha}{\alpha+\beta+\eta_P} S_R^T \cdot \vec{u} + \frac{\beta}{\alpha+\beta+\eta_P} S_X \cdot \vec{a} + \frac{\eta_P}{\alpha+\beta+\eta_P} \vec{p}_0,$$

$$\vec{a} = \frac{\gamma}{\gamma+\beta+\eta_A} S_Y^T \cdot \vec{u} + \frac{\beta}{\gamma+\beta+\eta_A} S_X^T \cdot \vec{p} + \frac{\eta_A}{\gamma+\beta+\eta_A} \vec{a}_0, \tag{4}$$

p, p0: ranking vector and prior preference vector for items

Online Learning

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

Algorithm 1: TriRank for review-aware top-N item recommendation.

Input: User-Item interactions R and reviews.

Offline Training (for all users):
1. Extract aspects from reviews (Section 2).
2. Build item–aspect matrix X and user–aspect matrix Y.
3. TF term weighting: $X = X.tf(); Y = Y.tf()$.
4. Build symmetric normalized matrices S_R, S_X, S_Y.

Online Recommendation (for target user u_i):
5. Build u_i's prior preference vectors $\vec{p}_0$, $\vec{a}_0$ and $\vec{u}_0$.
6. L_1 norm on $\vec{p}_0$, $\vec{a}_0$ and $\vec{u}_0$.
7. Randomly initialize ranking vectors $\vec{p}$, $\vec{a}$ and $\vec{u}$.
8. Iteratively run update rules Eq. (4), until convergence.
9. Recommend top ranked items to u_i, and explain the recommendation using top ranked aspects.

- Extract aspects from user review
- Build the graph (edge labels), and propagate vertex labels
- Computing user profile is almost constant time

Aspect Extraction Used in TriRank

Utilize rule-based tool built from studies in review mining techniques to extract aspects from each review. [Zhang *etc.* SIGIR'14]

Table 1: Top automatically extracted aspects.

Yelp	bar, salad, menu, chicken, sauce, restaurant, rice, cheese, fries, bread, sandwich, drinks, patio
Amazon	camera, quality, sound, price, product, battery, pictures, features, screen, size, memory, lens

ranked by tf x idf

Table 2: Statistics of aspects extracted from reviews.

		User–Aspect		Item–Aspect	
Dataset	# of Aspects	Avg. # of A / User	Density	Avg. # of A / Item	Density
Yelp	6,025	183.8	3.05%	138.0	2.29%
Amazon	1,617	61.4	3.80%	23.2	1.44%

The aspect extraction is fairly noisy, although the impact is muted on TriRank's performance as they occur less frequently in reviews.

Domain-specific stop words also distribute evenly across users and items, and thus do not change relative rankings between items.

Negative Examples - *"picturemy"*, *"restaurants"*, *"features"*, *etc.*

Experimental Settings of TriRank

Public Datasets - 2013 Yelp Challenge, Amazon Electronics Category

Yelp - 49.6% of users only made one review   *Amazon - 77.9% of users only made one review*

- Users and items with less than 10 reviews are filtered out
- Reviews are sorted in chronological order for each user
- 80% training, remaining 20% split between validation and testing set

Top-K evaluation with Hit Ratio @ K and NDCG @ K

$$HR@K = \frac{Number\ of\ Hits\ @K}{|GT|}.$$

$$NDCG@K = Z_K \sum_{i=1}^{K} \frac{2^{r_i} - 1}{log_2(i+1)},$$

Baselines In Comparison

Item Popularity (ItemPop) -Items are ranked by their popularity judged by number of ratings.

ItemKNN - Item-based CF, used commercially by Amazon. TriRank uses cosine similarity to measure the similarity among items.

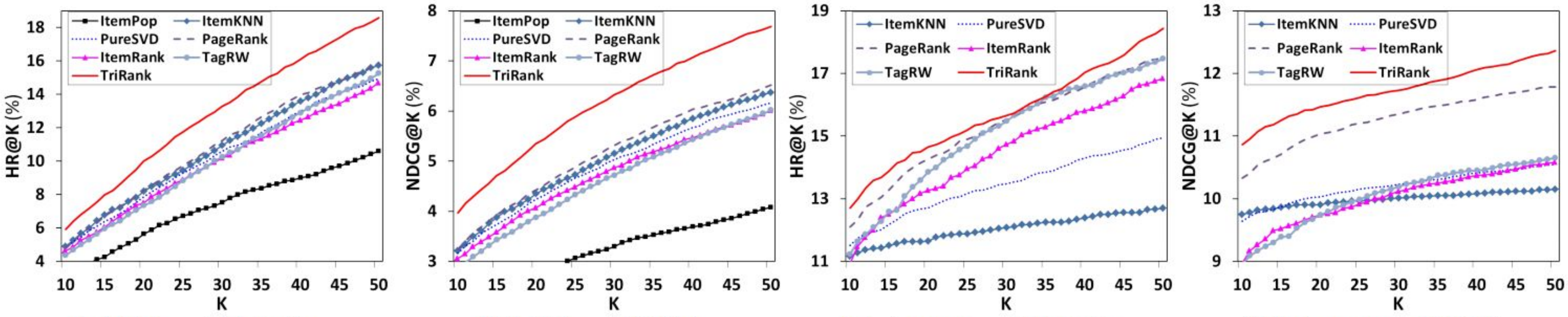
PureSVD - A state-of-the-art for top-N recommendation, which performs Singular Value Decomposition on the whole matrix, thus directly considering all instances. unlike other LFM that optimize against error only on rated instances.

Personalized PageRank - Widely used graph method for top-N recommendation,.

ItemRank - Graph based method that recommends based on the item–item correlation graph. Similar to Personalized PageRank

TagRW - State-of-the-art method to model tags for top-N item recommendation.

Experimental Results



(a) Yelp – Hit Ratio (b) Yelp – NDCG (c) Amazon – Hit Ratio (d) Amazon – NDCG

Figure 4: Performance evaluated by Hit Ratio and NDCG from position 10 to 50 (*i.e.*, K).

Table 5: Performance of TriRank with different parameter settings at rank 50.

Dataset	Yelp		Amazon	
Settings	**HR**	**NDCG**	**HR**	**NDCG**
0. All set	**18.58**	**7.69**	**18.44**	**12.36**
1. $\beta = 0$ (no item–aspect)	17.05	6.91	16.23	11.31
2. $\gamma = 0$ (no user–aspect)	**18.52**	**7.68**	**18.40**	**12.36**
3. $\eta_A = 0$ (no aspect query)	18.21	7.51	17.62	12.10
4. $\beta, \gamma, \eta_A = 0$ (no aspects)	17.00	6.90	15.97	11.16
5. $\alpha = 0$ (no user–item)	11.67	4.84	10.32	5.08

TriRank's Enhanced Explainability

Figure 6: Training reviews of a sampled Yelp user.

Reasoned Recommendation =

Collaborative Filtering *(Similar users also chose)* +

Aspect Filtering *(Reviewed aspects match the target)*

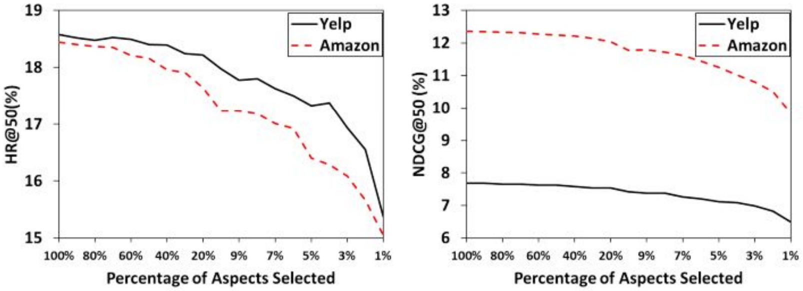
TriRank was able to correctly infer user went to Red Lobster (K = 3)

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

- User-item relationship is the most fundamental to the system
- Item-aspect relationship **>** User-aspect relationship
- *User-aspect relationship provides the less utility to the model*

Aspect Filtering - How do Noisy Aspects Influence?



(a) Hit Ratio versus Aspects (b) NDCG versus Aspects

Figure 5: TriRank performance with respect to percentage of top aspects selected.

Rank aspects by their TF-IDF score in item-aspect matrix.

Shows high TF-IDF aspects carry more useful signals for recommendation, but filtering out low TF-IDF aspects does not improve the system significantly.

Conclusion and Further Works

TriRank is a tripartite graph ranking solution for review-aware recommendation.

- Online Learning with Instant Updates without Retraining
- Explainable and Transparent
- Robust to Noisy Aspects in Reviews

Future Work to Expand the System

- Combine with factorization model to handle sparse review data environment
- Personalized regularization hyperparameter tuning
- Adding sentiment / context of review in addition to aspects