**TriRank: Review-aware Explainable Recommendation**

**by Modeling Aspects**

1. INTRODUCTION

TriRank ranks vertices by accounting for both the structural smoothness (encoding collaborative filtering and aspect filtering effects) and fitting constraints (encoding personalized preferences).

2. ASPECT EXTRACTION

Aspect extraction, also termed as feature or attribute ex- traction, has a long history in review mining (see [31]). Aspects can be seen as the components, attributes, or properties of an item.

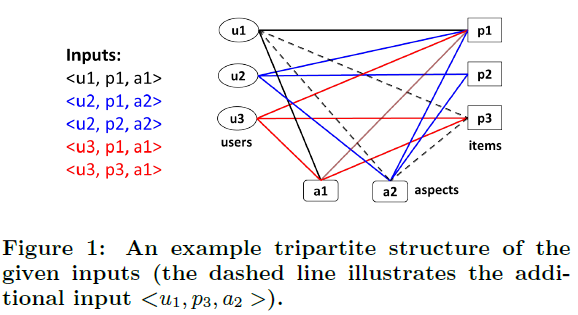
Aside from the unsupervised rule-based methods, supervised sequence labeling techniques such as the Conditional Random Field have been adopted to learn aspects. As each feature is a noun word or phrase, representing the item's property that a user comments on, we can directly use it as an aspect. Aspects, on the other hand, describe specific attributes of items, and are implicitly extracted from free-text reviews.

3. PROPOSED METHOD

3.1 Data Model and Notation

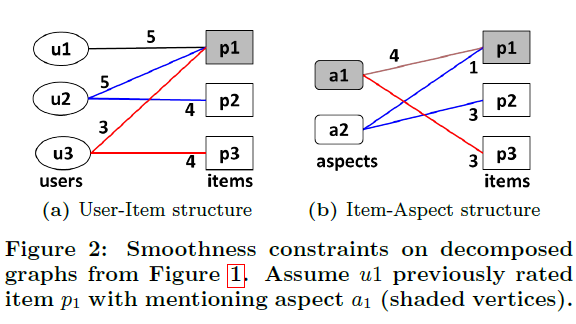
Each edge carries a weight to denote the strength of two connected vertices; edges with higher weight denote stronger more significant relations between vertices.

3.2 Tripartite Graph Ranking (TriRank)



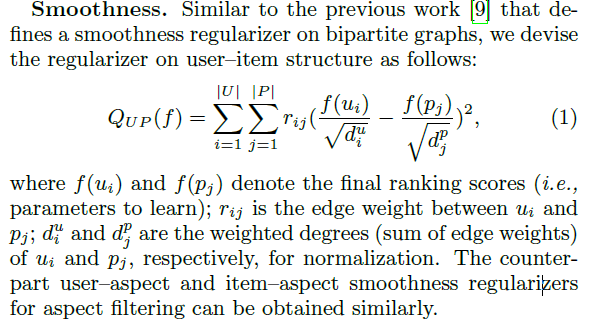
Smoothness implies local consistency: that nearby vertices should not vary too much in their scores.

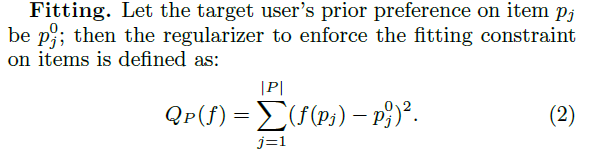
Fitting encodes prior belief: that the ranking function should not cause much deviation from the observations.

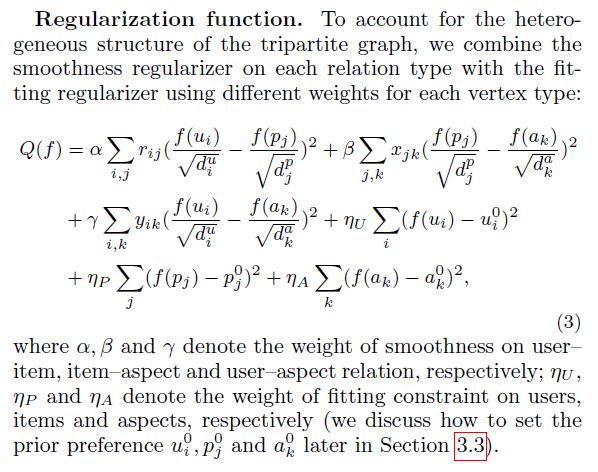


3.2.1 Illustrating Regularization Constraints

3.2.2 Regularization on Tripartite Graph







3.2.3 Optimizing the Regularization Function  
For this scenario, we adopt ALS over SGD as the objective function can be analytically solved for each parameter, and importantly, it does not need to set the learning rate, which is crucial to SGD's effectiveness. Additionally, it usually yields a faster convergence and is easier to parallelize than SGD.

3.3 Personalized Recommendation

Given the general TriRank algorithm, we need to cover how we obtain the initial graph (specifically, edge weights and the target user's prior preference) to concretize the generic algorithm for our review-based recommendation scenario.

**Edge weights**

**Prior preference**

The overall solution can be seen as a semi-supervised learning process on graphs-with the prior preference as labeled data, the algorithm propagates the labels to other unlabeled vertices.

3.4 Discussion

There are three properties of TriRank that merit a more detailed discussion: explainability, insensitivity to noisy aspects, and structural ambiguity.

4. EXPERIMENTS

4.1 Performance Study

4.2 Utility of Aspects

4.2.1 Aspect Importance Study

4.2.2 Aspect Quality Study

There are natural issues about aspects that we also wish to address:

1. How do the aspect-related components (e.g., item-aspect and user{aspect) contribute to the performance?

2. How does the quality of aspects impact the performance? Can TriRank handle the inherent noise in automatically extracted aspects well?

For the second question, we first rank aspects by their tf-idf score in the item-aspect matrix, and then select top scoring aspects to build the tripartite graph and inspect TriRank's performance.

4.3 Case Studies

While macro-level empirical analysis are useful, it is also instructive to examine actual results to better understand the outputs of TriRank. To this end, we give two case studies drawn from the Yelp dataset to demonstrate its explainability and scrutability.

4.3.1 Explainability

4.3.2 Scrutability

5. RELATED WORK

5.1 Review-aware Recommendation

Regardless of domain, we can categorize the approaches based on how reviews are integrated into the recommender: 1) word-based, 2) sentiment-based, and 3) aspect-based methods.

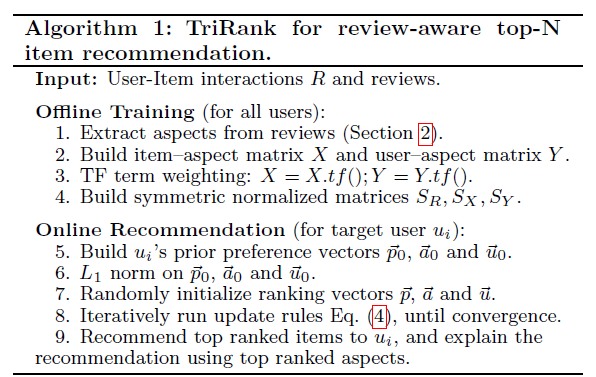
**Word-based** These approaches directly factorize the re- view words into CF.

**Sentiment-based** These approaches utilize the user's explicitly mentioned opinions on items.

**Aspect-based** Our work falls into this category.

Several hybrid methods have also integrated aspect and sentiment as they are closely related.

Compared to the above review-aware works, our method explores a graph model to integrate aspects, which has not been previously been investigated. Moreover, our proposed TriRank affords the recommender a finer degree of user interaction | aspect preference | allowing for both more accurate and transparent recommendation.



5.2 Graph-based Recommendation

Graphs form a natural representation for modeling the relationship among data objects. A typical workflow is first representing items as vertices of a graph, and then admitting recommendation as a vertex-ranking problem.

6. CONCLUSION