The FacT: Taming Latent Factor Models for Explainability with Factorization Trees

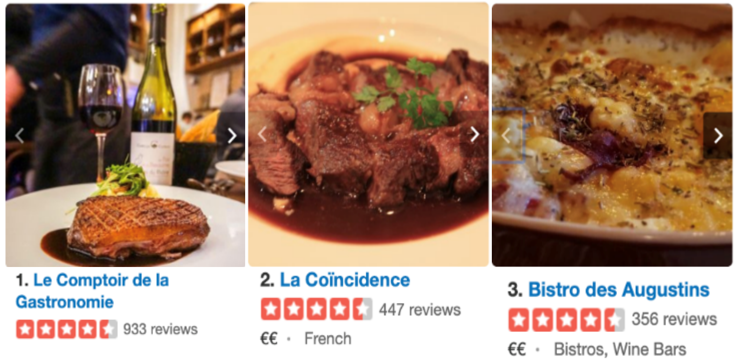
Yiyi Tao[1], **Yiling Jia**[2], Nan Wang[2], Hongning Wang[2]

[1]Peking University, [2]University of Virginia

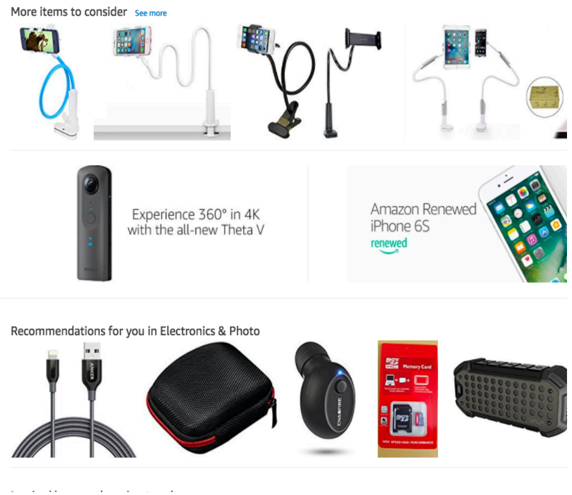
Explainable Recommendation

- Recommender systems have achieved great success in feeding the right content to the right users.

Best Restaurants in Paris, France



Online Shopping



- Transparency
  - System: how the customized results should be presented to a user?
  - User: why this item is recommended to me?

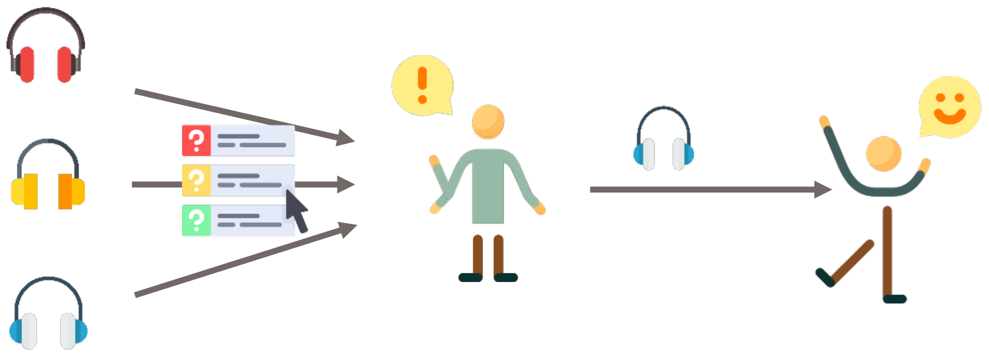
Explainable Recommendation

- Explanation in recommender system
  - Allow the users to make more informed and accurate decisions about which results to utilize.

Its comfort and sound performance matches your preference.

Its light weight, and fantastic appearance matches your preference.

Its low price, and great comfort matches your preference.



- The **fidelity of explanations** is a prerequisite for explainable recommendations to be useful in practice.
  - Improve transparency, increase recommendation effectiveness, user satisfaction and trust, etc.

Recommendation vs. Explanation

- **Recommendation quality** and **explanation fidelity** have long been considered **irreconcilable**[1]
  - Content-based collaborative filtering
    - Easy to explain, limited recommendation quality.
  - Latent factor models
    - Promising performance, hard to explain.

- **Neighbor-based**: similarity in learned latent space[2].

The latent space is not constructed for explanation!

Recommendation vs. Explanation

- **Recommendation quality** and **explanation fidelity** have long been considered **irreconcilable**[1]
  - Content-based collaborative filtering
    - Easy to explain, limited recommendation quality.
  - Latent factor models
    - Promising performance, hard to explain.

- **Neighbor-based**: similarity in learned latent space[2].
- **Feature-based**: incorporate sentiment analysis[3][4].

The feature representation learning is only a companion task of recommendation learning.

Recommendation quality *vs.* Explanation fidelity

Insight

- The tension between recommendation quality and explanation fidelity is not necessarily inevitable.

Rule-based Decision Making → Latent Factor Model Learning

Easy to perceive and justify

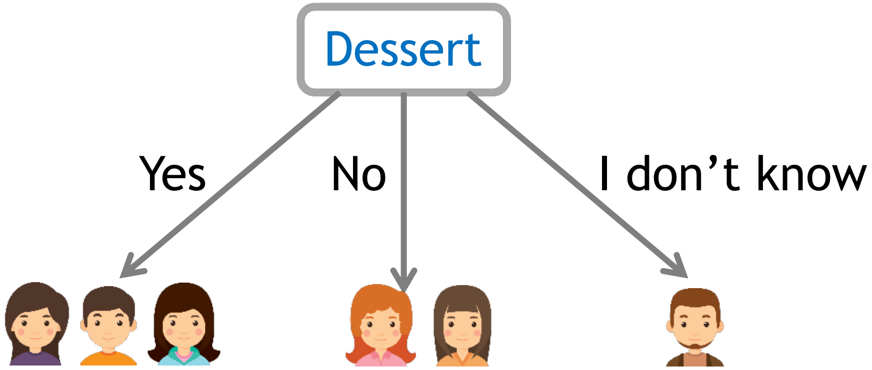
Effectiveness in recommendation

- **Treat the latent factors as a function of the rules**
  - Users who provide the same responses to the rules would share the same latent factors. Same for the items.

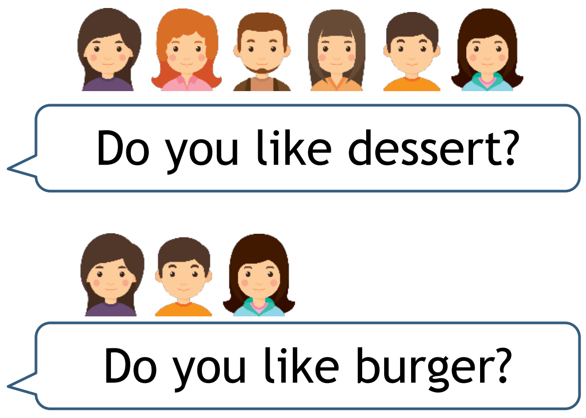
Users and items can be grouped according to the rules.

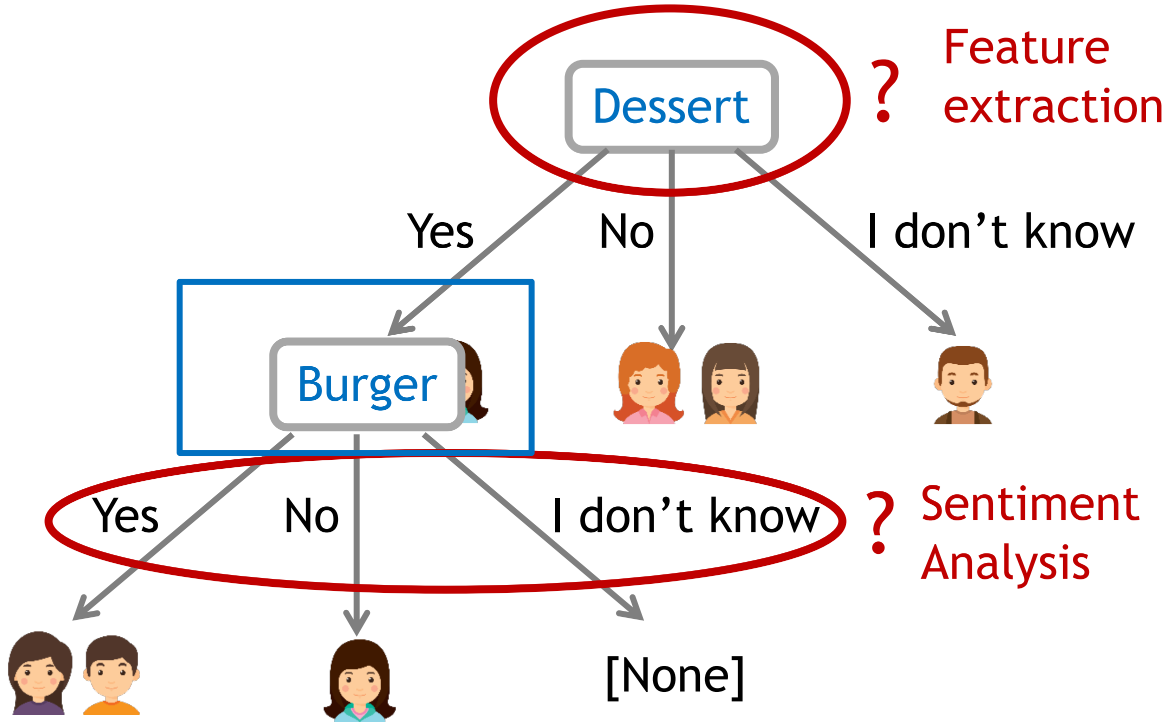
Share Similar characteristics

Insight



Insight





- Construct user tree and item tree.
- Explanation for the recommendation:
  - *We recommend [restaurant X] because it matches your preference on dessert and burger. And it performs well on cake.*

Sentiment Analysis in Reviews

Feature & Opinion Extraction

- User reviews provide a fine-grained understanding of a user's evaluation of an item.
- Feature-level sentiment analysis techniques can be readily applied to reviews[5][6].

Restaurant B

Customer A

The food is good, great burger, crispy potato fries.
But the service is awful and we waited for a long time to get the drink and they didn't come by ever to ask us if we need refill.

User A → Item B

Feature, Opinion, Sentiment Polarity

(food, good, +1)
(burger, great, +1)
(potato fries, crispy, +1)
(service, awful, -1)
(wait time, long, -1)

→

With all reviews

Feature-level user profile

(User, Feature, Opinion)

Tree Construction: Rule Induction

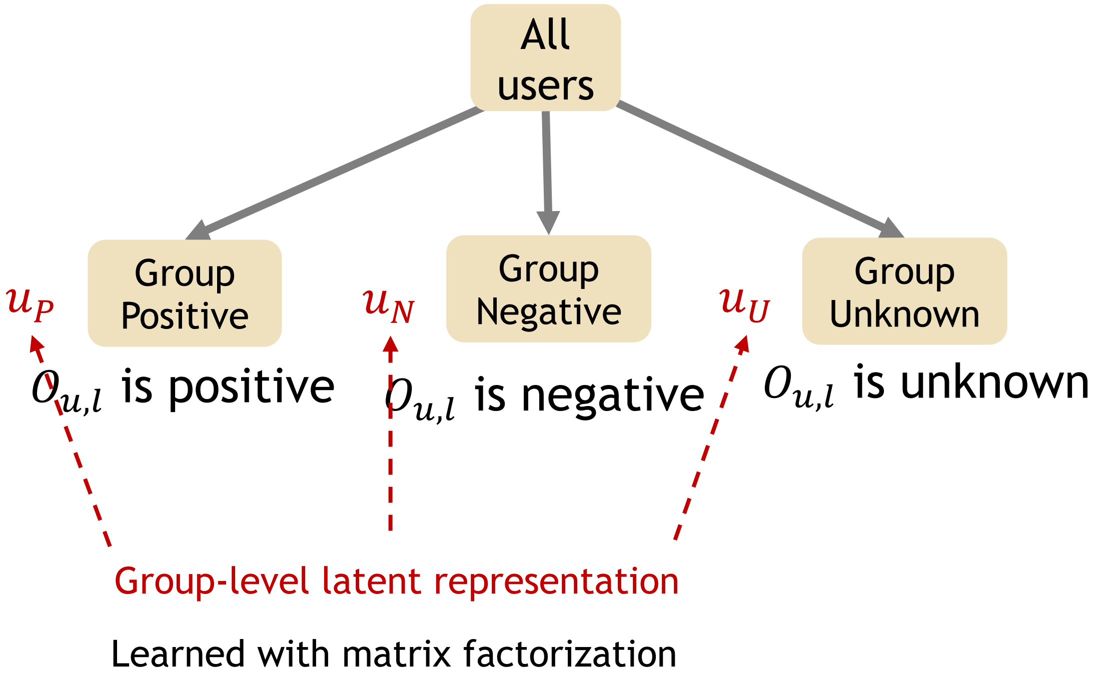
How to select the features?

- **Treat the latent factors as a function of the rules.**
  - Users who provide the same responses to the rules would share the same latent factors. Same for the items.

Find the best feature to divide the users!

Generate the minimum reconstruction error.

For feature l



Tree Construction: Rule Induction

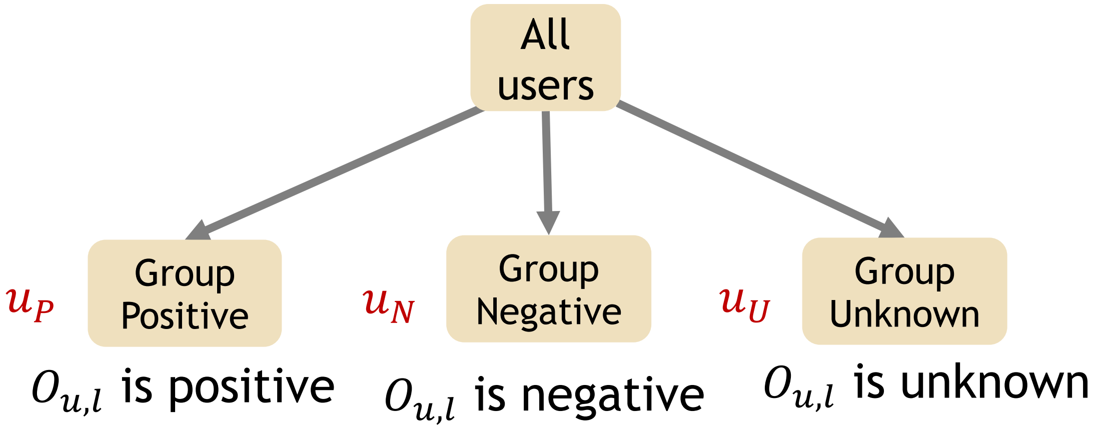
How to select the features?

- **Treat the latent factors as a function of the rules.**
  - Users who provide the same responses to the rules would share the same latent factors. Same for the items.

Find the best feature to divide the users!

Generate the minimum reconstruction error.

For feature l



$$
\begin{aligned}
&Error(l,) \\
&= L(u_L, V, R_L) + L(u_R, V, R_R) + L(u_E, V, R_E) \\
&- \lambda_b\big(B(u_L, V, R_L) + B(u_R, V, R_R) + B(u_E, V, R_E)\big) + \lambda_u(||u_L|| + ||u_R|| + ||u_E||)
\end{aligned}
$$

Tree Construction: Rule Induction

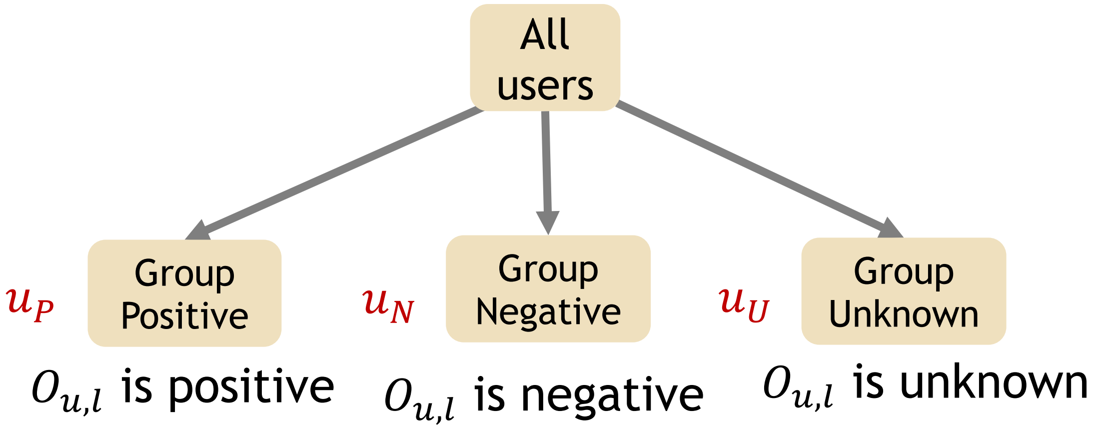
How to select the features?

- **Treat the latent factors as a function of the rules.**
  - Users who provide the same responses to the rules would share the same latent factors. Same for the items.

Find the best feature to divide the users!

Generate the minimum reconstruction error.

For feature l



$$Error(l) = \boxed{L(u_L, V, R_L) + L(u_R, V, R_R) + L(u_E, V, R_E)} - \lambda_b\big(B(u_L, V, R_L) + B(u_R, V, R_R) + B(u_E, V, R_E)\big) + \lambda_u(||u_L|| + ||u_R|| + ||u_E||)$$

Pointwise loss (MSE)

Tree Construction: Rule Induction

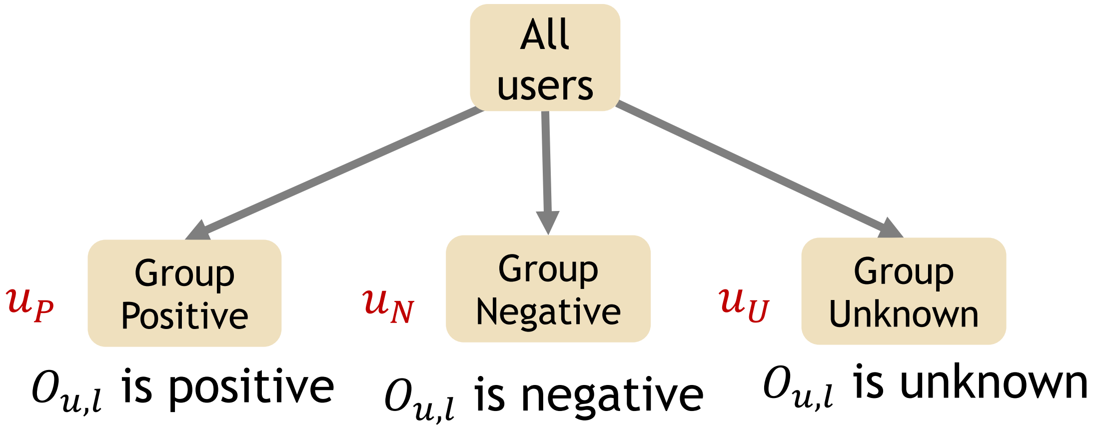
How to select the features?

- **Treat the latent factors as a function of the rules.**
  - Users who provide the same responses to the rules would share the same latent factors. Same for the items.

Find the best feature to divide the users!

Generate the minimum reconstruction error.

For feature l



$$
\begin{aligned}
&Error(l) \\
&= L(u_L, V, R_L) + L(u_R, V, R_R) + L(u_E, V, R_E) \\
&- \underbrace{\boxed{\lambda_b\big(B(u_L, V, R_L) + B(u_R, V, R_R) + B(u_E, V, R_E)\big)}}_{\text{Pairwise loss (BPR)}} + \lambda_u(||u_L|| + ||u_R|| + ||u_E||)
\end{aligned}
$$

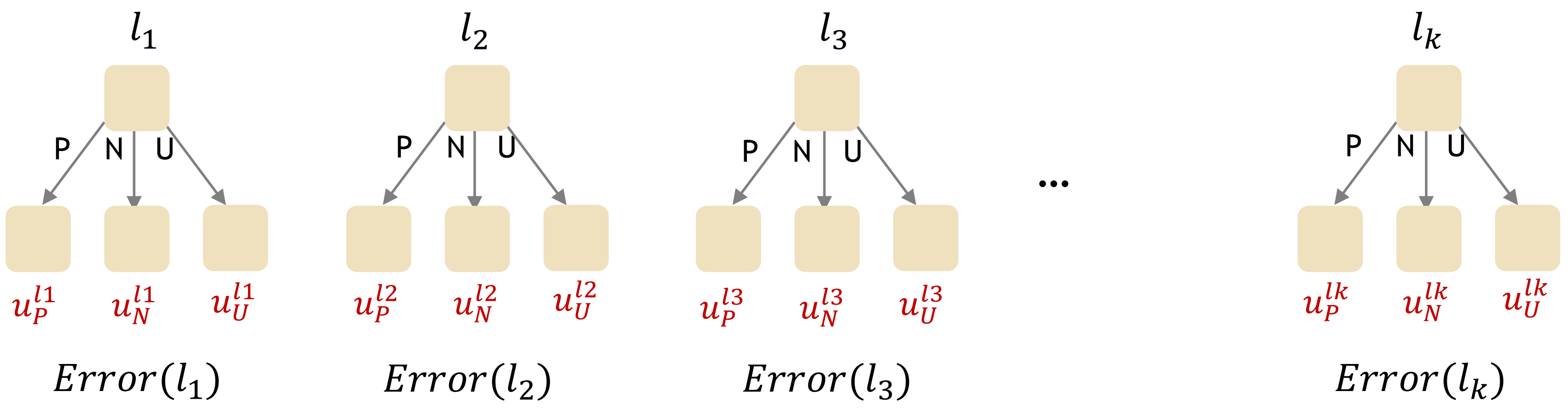
Tree Construction: Rule Induction

How to select the features?

- **Treat the latent factors as a function of the rules.**
  - Users who provide the same responses to the rules would share the same latent factors. Same for the items.

Find the best feature to divide the users!

Generate the minimum reconstruction error.



Tree Construction: Rule Induction

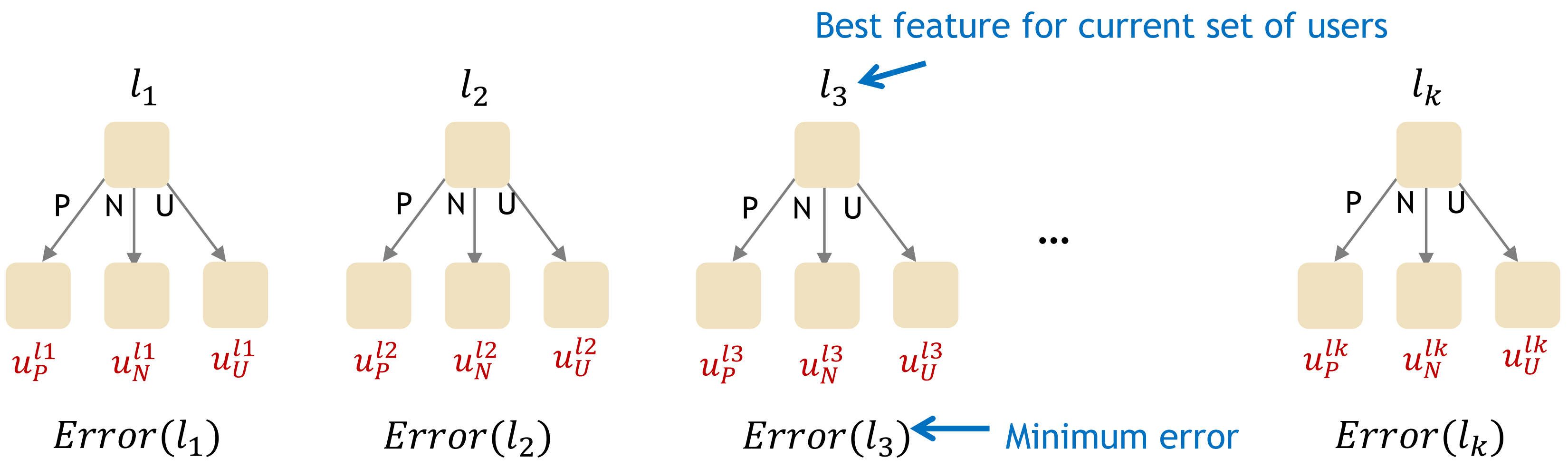
How to select the features?

- **Treat the latent factors as a function of the rules.**
  - Users who provide the same responses to the rules would share the same latent factors. Same for the items.

Find the best feature to divide the users!

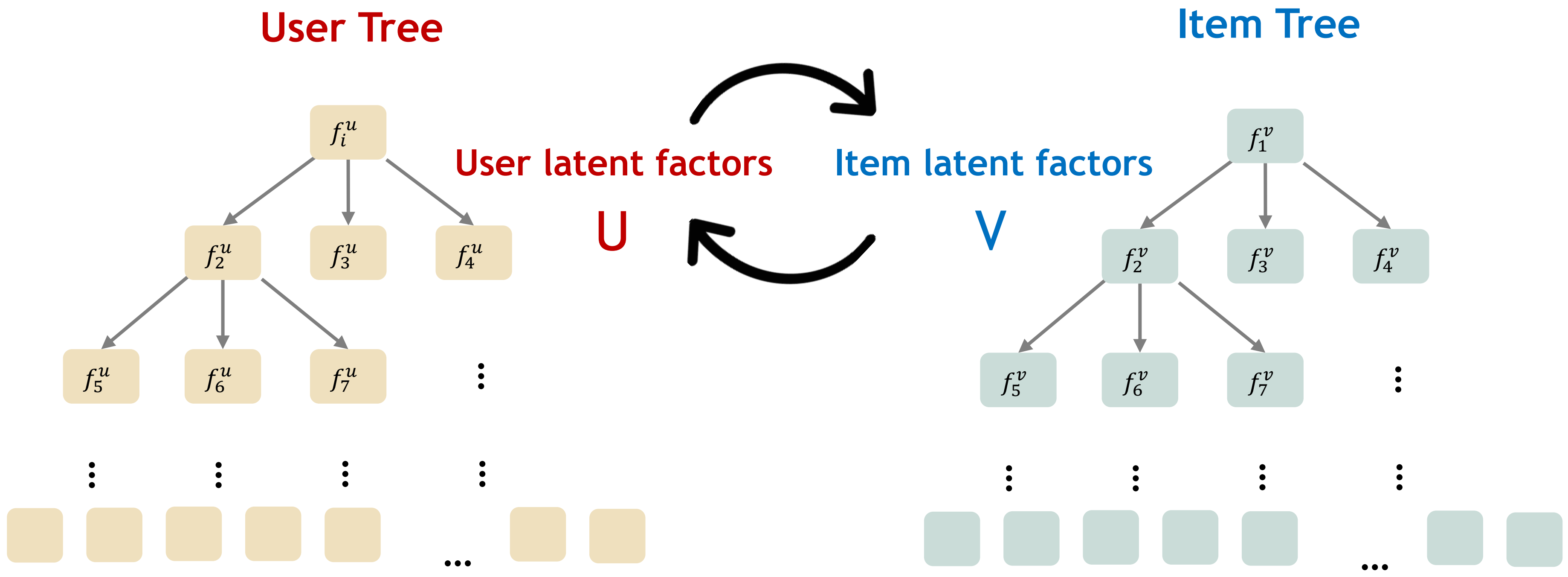
Generate the minimum reconstruction error.

$$l_{best} = argmin_{l \in F} Error(l)$$



Tree Construction: Alternative Optimization

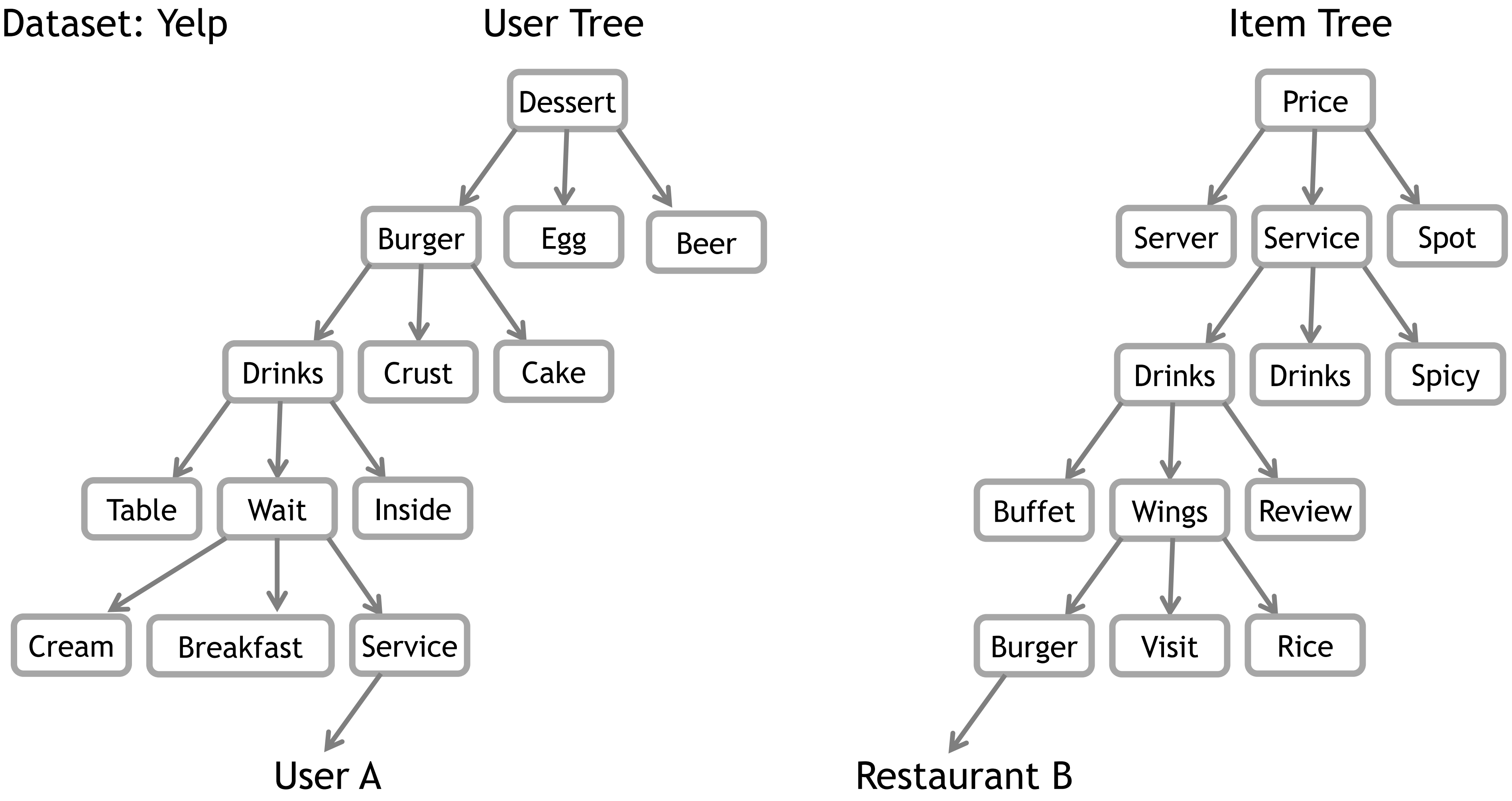
- Initialization
  - Perform a plain matrix factorization to obtain the initial item factors V_0.



- Factorization Tree (FacT)
  - Alternate the optimization of explanation rule construction and latent factor learning under a recommendation quality based metric.

Explanation Generation

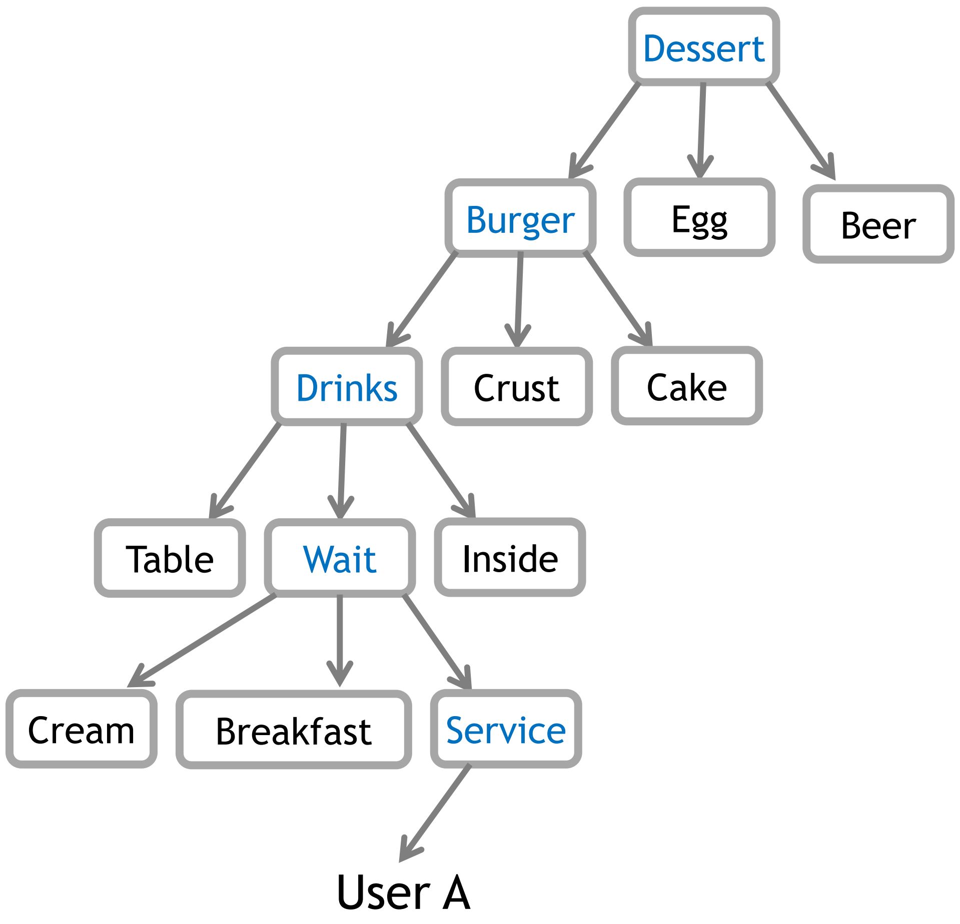
Dataset: Yelp



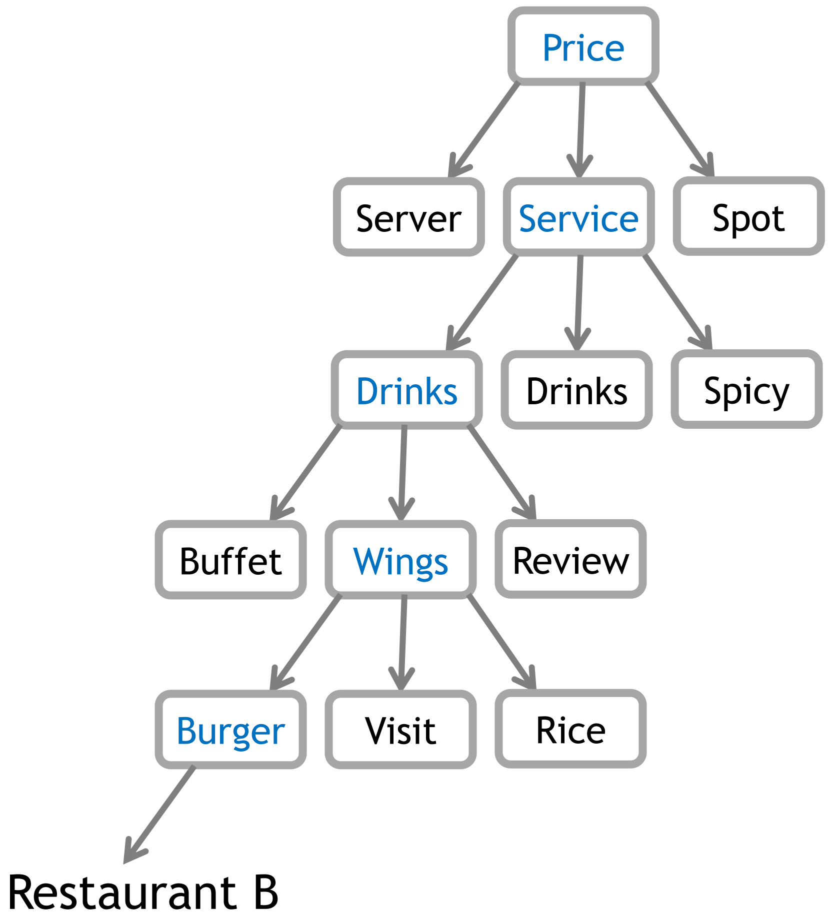
Explanation Generation

Dataset: Yelp

User Tree

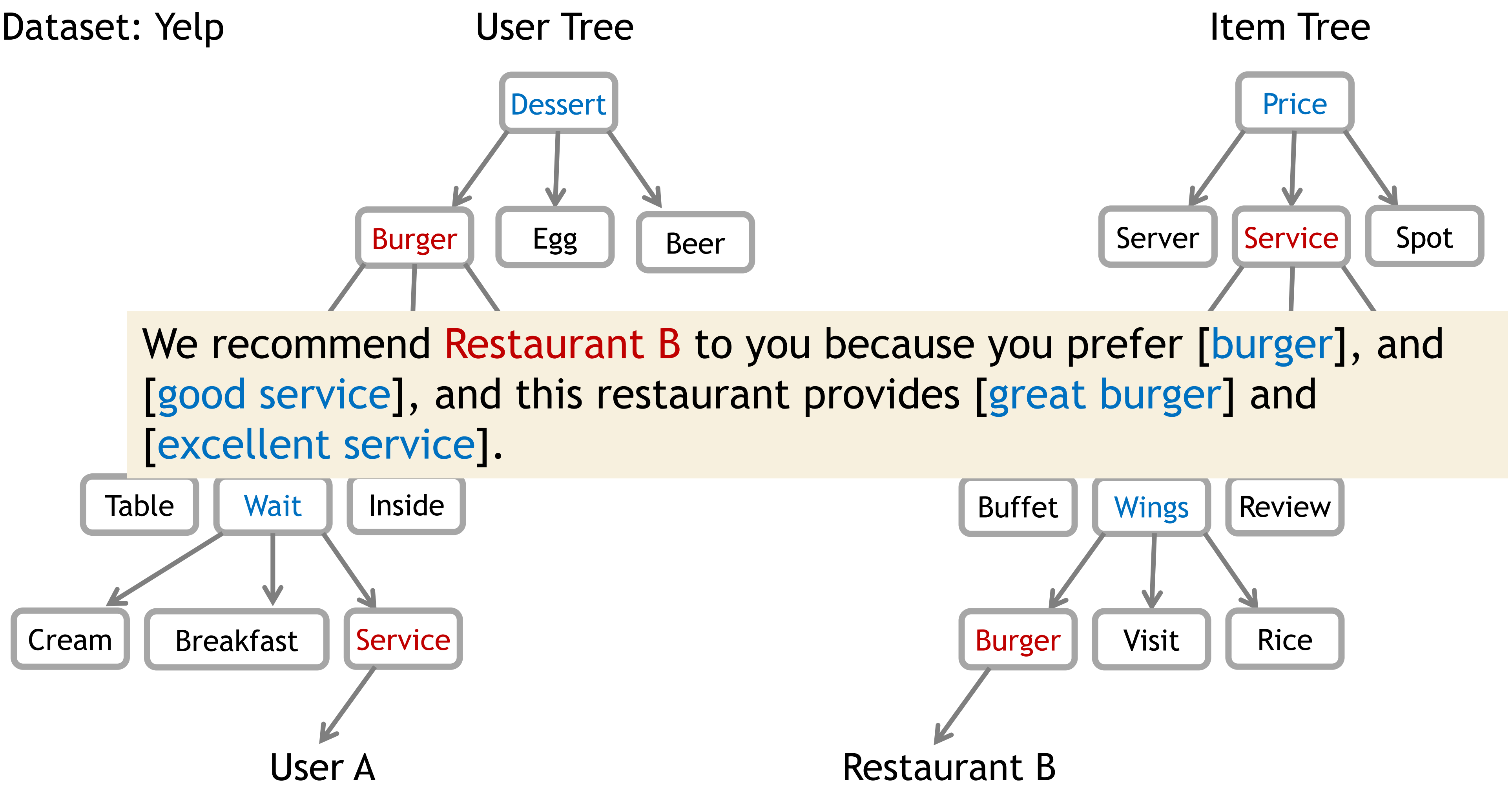


Item Tree



Explanation Generation

Dataset: Yelp



We recommend Restaurant B to you because you prefer [burger], and [good service], and this restaurant provides [great burger] and [excellent service].

Experiment: Setup

Statistic of evaluation datasets:

Dataset	#users	#items	#features	#opinions	#reviews
Amazon	6,285	12,626	101	591	55,388
Yelp	10,719	10,410	104	1,019	285,346

Baselines:

- **Most Popular (MP):** Rank items by popularity.
- **NMF**: Non-negative Matrix Factorization[7].
- **BPRMF**: Bayesian Personalized Raking (BPR) optimization for Matrix Factorization[8].
- **JMARS**: Jointly models aspects, ratings, and sentiments by collaborative filtering and topic modeling[9].
- **EFM**: Explicit Factor Models[10].
- **FMF**: Functional Matrix Factorization[11].
- **MTER**: A multi-task learning model that integrates user preference modeling and opiniated content modeling via a joint tensor factorization[12].

Quantitative Evaluation

- Top-K recommendation
- NDCG: items ranked higher should be more relevant to a user's preference.
- Depth = 6, latent dimension = 20

Table 2: Comparison of recommendation performance.

NDCG @K	Amazon								Improvement best v.s. second best
	FMF	MP	NMF	BPRMF	JMARS	EFM	MTER	FacT	
10	0.1009	0.0961	0.0649	0.1185	0.1064	0.1109	0.1351	**0.1482**	9.70%*
20	0.1331	0.1310	0.0877	0.1490	0.1348	0.1464	0.1653	**0.1795**	8.59%*
50	0.1976	0.1886	0.1601	0.2070	0.1992	0.2056	0.2234	**0.2367**	5.95%*
100	0.2529	0.2481	0.2144	0.2669	0.2575	0.2772	0.2803	**0.2869**	2.35%*
NDCG @K	Yelp								Improvement best v.s. second best
	FMF	MP	NMF	BPRMF	JMARS	EFM	MTER	FacT	
10	0.0931	0.1060	0.0564	0.1266	0.1155	0.1071	0.1380	**0.1499**	8.62%*
20	0.1243	0.1333	0.0825	0.1643	0.1553	0.1354	0.1825	**0.1991**	9.10%*
50	0.1871	0.1944	0.1345	0.2214	0.2111	0.1903	0.2365	**0.2488**	5.20%*
100	0.2509	0.2502	0.2175	0.2668	0.2575	0.2674	0.2783	**0.2867**	3.02%*

* p-value < 0.05

Traditional factorization methods.
No explanation.

Quantitative Evaluation

- Top-K recommendation
- NDCG: items ranked higher should be more relevant to a user's preference.
- Depth = 6, latent_dimension = 20

Table 2: Comparison of recommendation performance.

NDCG	Amazon								Improvement
@K	FMF	MP	NMF	BPRMF	JMARS	EFM	MTER	FacT	best v.s. second best
10	0.1009	0.0961	0.0649	0.1185	0.1064	0.1109	0.1351	**0.1482**	9.70%*
20	0.1331	0.1310	0.0877	0.1490	0.1348	0.1464	0.1653	**0.1795**	8.59%*
50	0.1976	0.1886	0.1601	0.2070	0.1992	0.2056	0.2234	**0.2367**	5.95%*
100	0.2529	0.2481	0.2144	0.2669	0.2575	0.2772	0.2803	**0.2869**	2.35%*
NDCG	Yelp								Improvement
@K	FMF	MP	NMF	BPRMF	JMARS	EFM	MTER	FacT	best v.s. second best
10	0.0931	0.1060	0.0564	0.1266	0.1155	0.1071	0.1380	**0.1499**	8.62%*
20	0.1243	0.1333	0.0825	0.1643	0.1553	0.1354	0.1825	**0.1991**	9.10%*
50	0.1871	0.1944	0.1345	0.2214	0.2111	0.1903	0.2365	**0.2488**	5.20%*
100	0.2509	0.2502	0.2175	0.2668	0.2575	0.2674	0.2783	**0.2867**	3.02%*

* p-value < 0.05

State-of-the-art explainable recommendation methods

Quantitative Evaluation

- Top-K recommendation
- NDCG: items ranked higher should be more relevant to a user's preference.
- Depth = 6, latent_dimension = 20

Table 2: Comparison of recommendation performance.

NDCG @K	Amazon								Improvement best v.s. second best
	FMF	MP	NMF	BPRMF	JMARS	EFM	MTER	FacT	
10	0.1009	0.0961	0.0649	0.1185	0.1064	0.1109	0.1351	**0.1482**	9.70%*
20	0.1331	0.1310	0.0877	0.1490	0.1348	0.1464	0.1653	**0.1795**	8.59%*
50	0.1976	0.1886	0.1601	0.2070	0.1992	0.2056	0.2234	**0.2367**	5.95%*
100	0.2529	0.2481	0.2144	0.2669	0.2575	0.2772	0.2803	**0.2869**	2.35%*
NDCG @K	Yelp								Improvement best v.s. second best
	FMF	MP	NMF	BPRMF	JMARS	EFM	MTER	FacT	
10	0.0931	0.1060	0.0564	0.1266	0.1155	0.1071	0.1380	**0.1499**	8.62%*
20	0.1243	0.1333	0.0825	0.1643	0.1553	0.1354	0.1825	**0.1991**	9.10%*
50	0.1871	0.1944	0.1345	0.2214	0.2111	0.1903	0.2365	**0.2488**	5.20%*
100	0.2509	0.2502	0.2175	0.2668	0.2575	0.2674	0.2783	**0.2867**	3.02%*

* p-value < 0.05

Tree-based factorization.
Use item to group users.

Quantitative Evaluation

- Top-K recommendation
- NDCG: items ranked higher should be more relevant to a user's preference.
- Depth = 6, latent_dimension = 20

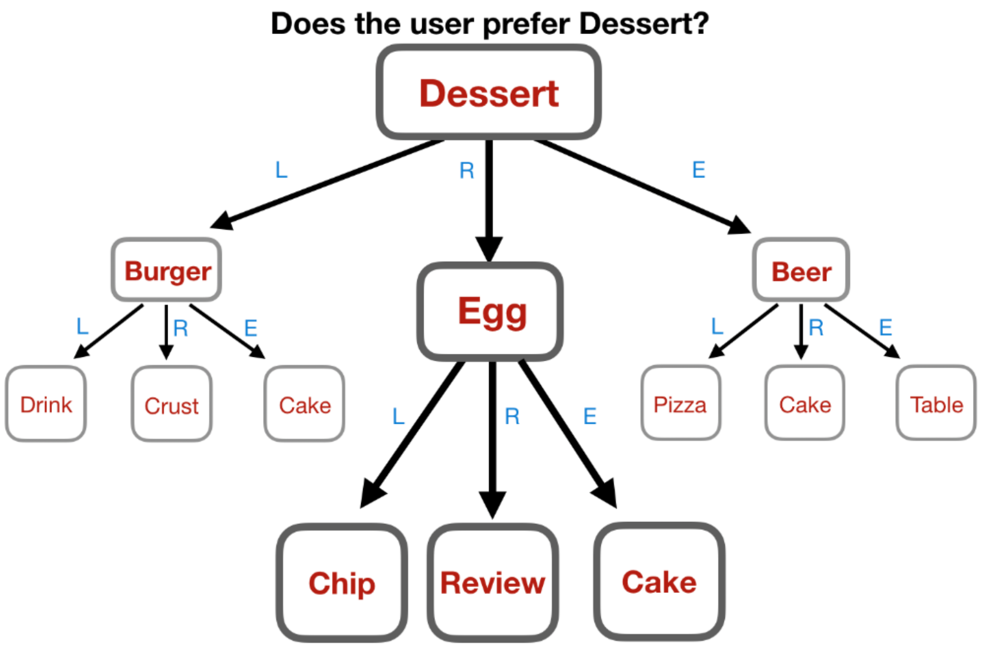
Table 2: Comparison of recommendation performance.

NDCG @K	Amazon FMF	MP	NMF	BPRMF	JMARS	EFM	MTER	FacT	Improvement best v.s. second best
10	0.1009	0.0961	0.0649	0.1185	0.1064	0.1109	0.1351	**0.1482**	9.70%*
20	0.1331	0.1310	0.0877	0.1490	0.1348	0.1464	0.1653	**0.1795**	8.59%*
50	0.1976	0.1886	0.1601	0.2070	0.1992	0.2056	0.2234	**0.2367**	5.95%*
100	0.2529	0.2481	0.2144	0.2669	0.2575	0.2772	0.2803	**0.2869**	2.35%*
NDCG @K	Yelp FMF	MP	NMF	BPRMF	JMARS	EFM	MTER	FacT	Improvement best v.s. second best
10	0.0931	0.1060	0.0564	0.1266	0.1155	0.1071	0.1380	**0.1499**	8.62%*
20	0.1243	0.1333	0.0825	0.1643	0.1553	0.1354	0.1825	**0.1991**	9.10%*
50	0.1871	0.1944	0.1345	0.2214	0.2111	0.1903	0.2365	**0.2488**	5.20%*
100	0.2509	0.2502	0.2175	0.2668	0.2575	0.2674	0.2783	**0.2867**	3.02%*

* p-value < 0.05

Cold-start Recommendation

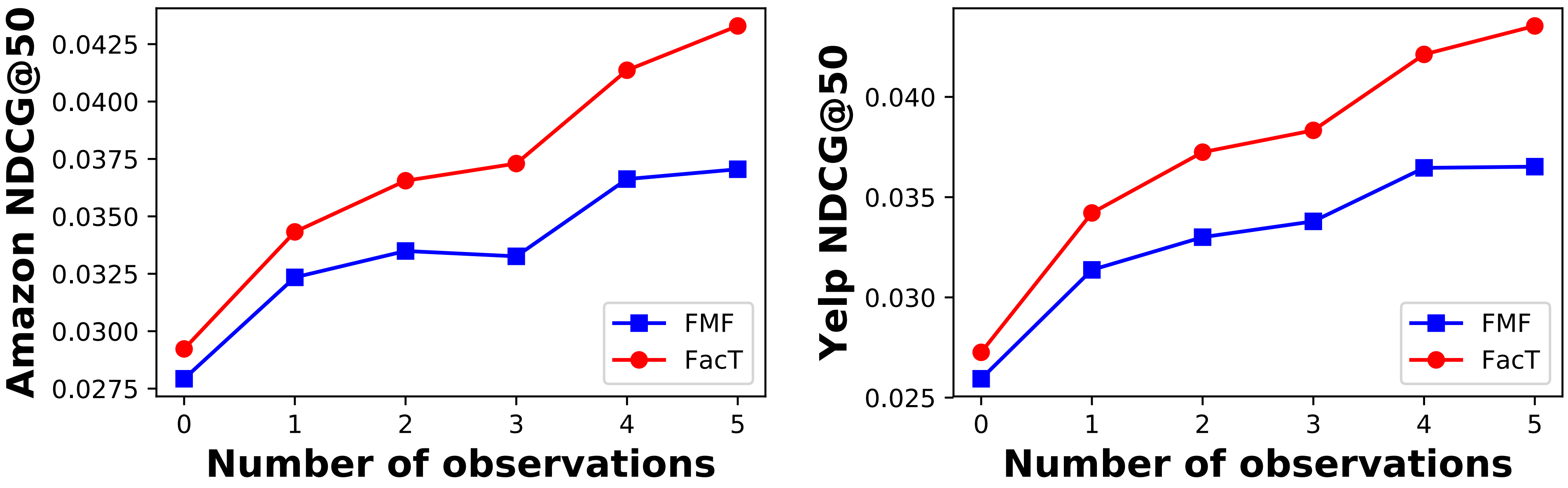
- Cold-start problem: without sufficient information about new users, it's hard to provide recommendation with high quality
- A by-product of FacT:
  - Rules: a set of interview questions to solicit user preference



- Training: 95% users
  - Build user tree and item tree
- Test: 5% users
  - Use first k reviews to construct user profile.

Cold-start Recommendation

- Cold-start problem: without sufficient information about new users, it's hard to provide recommendation with high quality
- A by-product of FacT:
  - Rules: a set of interview questions to solicit user preference



User Study: Setup

- Dataset: Amazon and Yelp
- Recruit participants on Amazon Mechanical Turk
- Settings
  - Warm-start users: the ratings and reviews are known to the system beforehand.
  - Cold-start users: totally new to the system.

User Study: Warm-start users

Participant

- Select a user in the training set.

User A
Review1
Review2
Review3
…

- Recommendation and explanation

Recommendation
Rec1, Explanation1
Rec2, Explanation2
Rec3, Explanation3
…

- Evaluate the model.

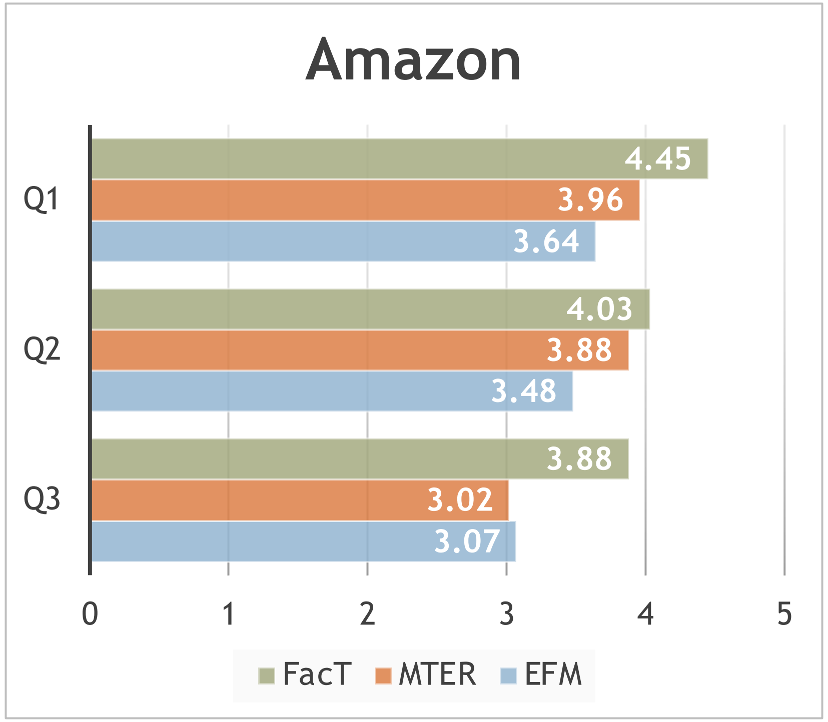
Evaluation
Q1, Rating1
Q2, Rating2
Q3, Rating3
…

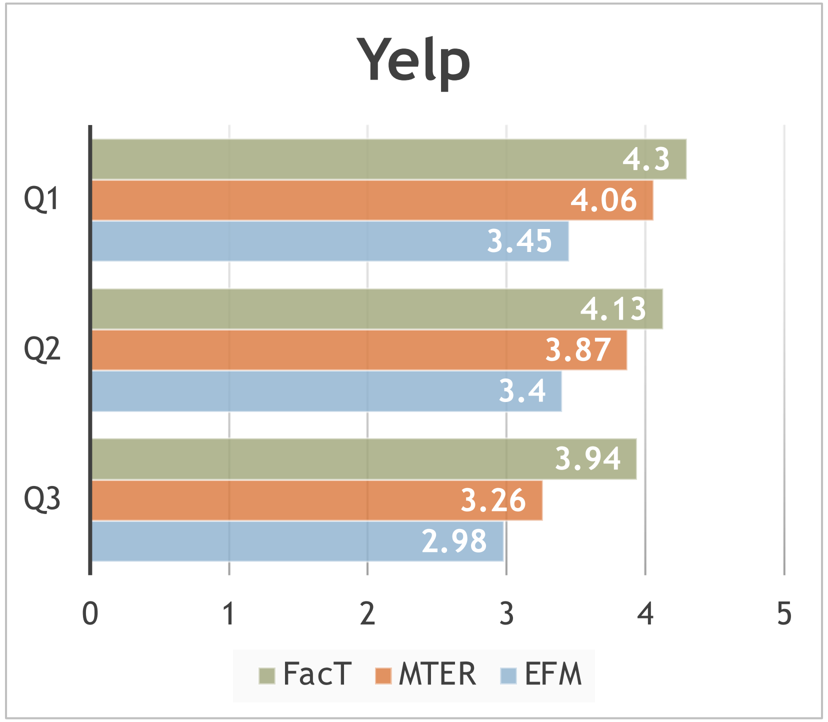
- Baseline: EFM, MTER (both can provide textual explanations)
- A/B test: ensure the evaluation is unbiased.
- Valid response: 300

User Study: Warm-start users

Q1: Generally, are you satisfied with our recommendations?
Q2: Do the explanations presented to you really match your preference?
Q3: Do you have any idea about how we make recommendations for you?

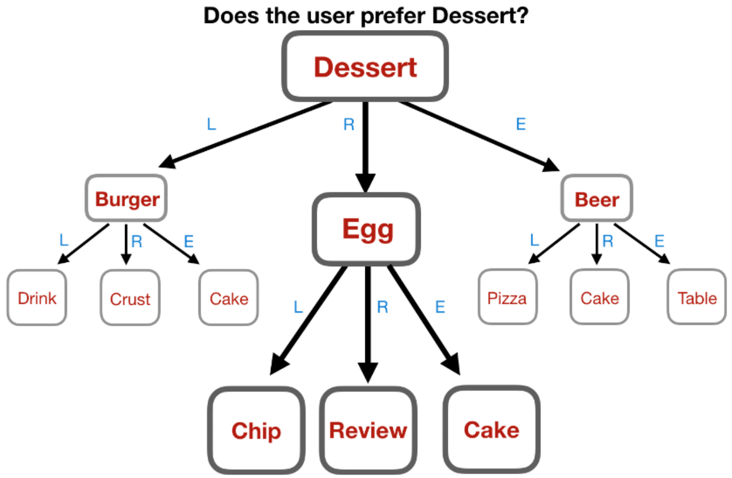
Score:
1.Strongly negative 2.Negative 3.Neutral 4. Positive 5.Strongly positive



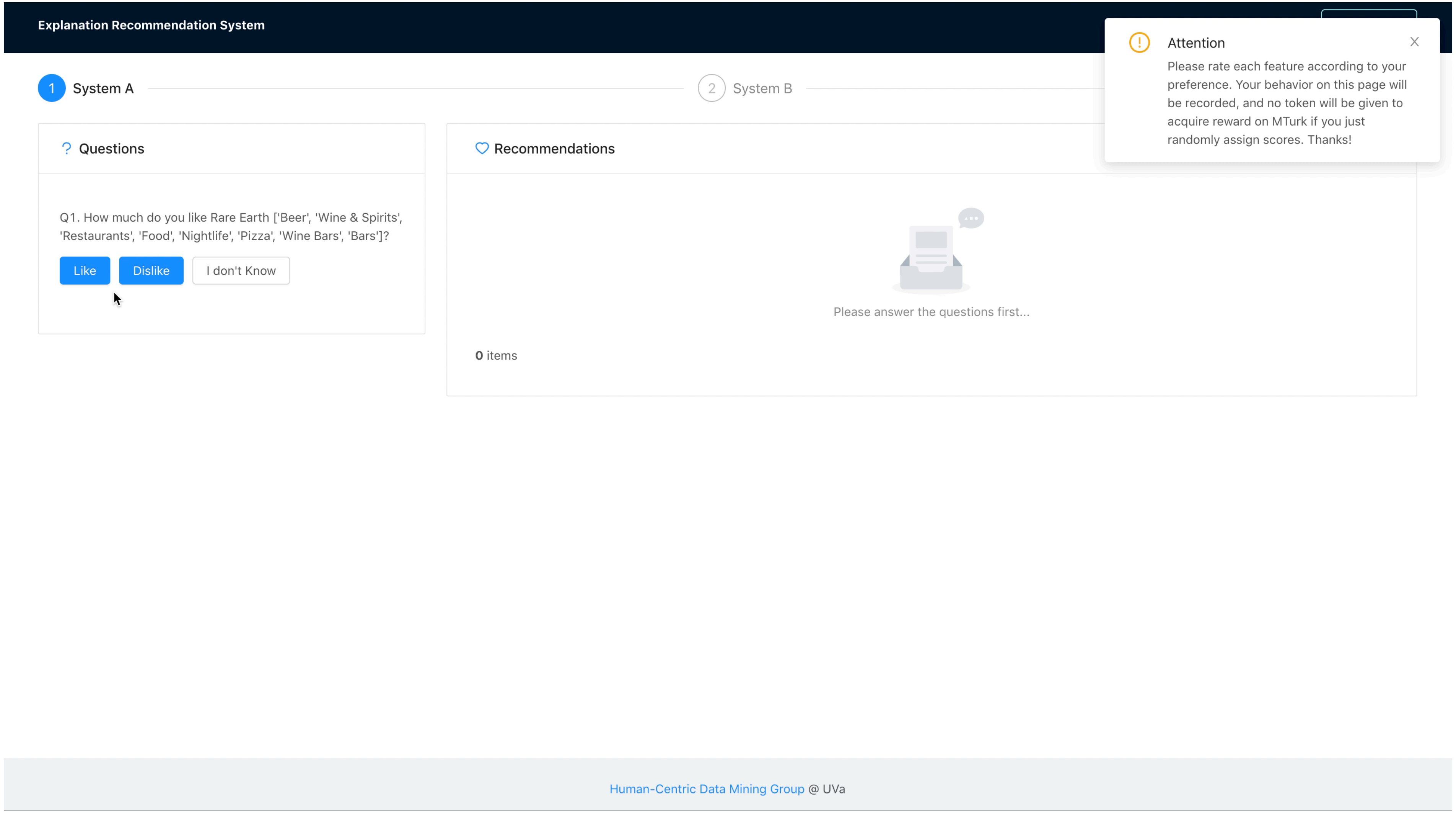


User Study: Cold-start users

- No review history for cold-start users
  - We progressively query user responses through an interview process.
  - Develop the user profile according to the responses.
- Baseline: FMF
  - Address the cold-start problem.
  - Use items to construct the tree.
- Interleaved test
  - Participants were asked to interact with two models one after the other in a random order.



User Study: Cold-start users



https://aobo-y.github.io/explanation-recommendation/

User Study: Cold-start users

Q1: Generally, between system A and system B, whose **recommendations** are you more satisfied with?
Q2: Between system A and B, whose **explanations** do you think can better help you understand the recommendation?
Q3: Between system A and B, whose **explanations** can better help you make a more informed decision?

- Valid response: 100

Table 5: Results of cold-start interleaved test.

number of votes	Amazon		Yelp	
	FMF	FacT	FMF	FacT
Q1	44	**63***	40	**64***
Q2	43	**64***	34	**70***
Q3	45	**62**	33	**71***

* p-value < 0.05

Conclusion

Conclusion

- We seamlessly integrate latent factor learning with explanation rule learning for explainable recommendation.
  - The fidelity of explanation is optimized.
  - The quality of recommendation is ensured.
- Both offline experiments and user studies have shown the effectiveness of our model in recommendation and explanation.

Future Work

- Use more complex forms of the threshold predicates, such as nonlinear function, for better explainability.
- Develop other hybrid factorization models to integrate sentiment analysis with rules.
- Use features as key words and retrieve sentences from items' reviews to generate more natural explanation.

Acknowledgement

- Thank the conference for awarding the travel grant.
- Thank the NSF grant IIS-1553568 and CPS-1646501 for supporting this work.

References

1. Abdollahi, Behnoush, and Olfa Nasraoui. "Using explainability for constrained matrix factorization." *Proceedings of the Eleventh ACM Conference on Recommender Systems*. ACM, 2017.
2. Sharma, Amit, and Dan Cosley. "Do social explanations work?: studying and modeling the effects of social explanations in recommender systems." *Proceedings of the 22nd international conference on World Wide Web*. ACM, 2013.
3. Bilgic, Mustafa, and Raymond J. Mooney. "Explaining recommendations: Satisfaction vs. promotion." *Beyond Personalization Workshop, IUI*. Vol. 5. 2005.
4. Tintarev, Nava. "Explanations of recommendations." *Proceedings of the 2007 ACM conference on Recommender systems*. ACM, 2007.
5. Yue Lu, Malu Castellanos, Umeshwar Dayal, and ChengXiang Zhai. 2011. Automatic Construction of a Context-aware Sentiment Lexicon: An Optimization Approach. In Proceedings of the 20th International Conference on World Wide Web.
6. Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. 2014. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In Proceedings of the 37th ACM SIGIR. 83–92.
7. D. D. Lee and H. S. Seung. Algorithms for Non-negative Matrix Factorization. Proc. NIPS, 2001.
8. Steffen Rendle, Christoph Freudenthaler, ZenoGantner and LarsSchmidt-Thieme. 2012. BPR: Bayesian Personalized Ranking from Implicit Feedback. CoRR (2012).
9. Qiming Diao, Minghui Qiu, Chao Yuan Wu, Alexander J. Smola, Jing Jiang, and Chong Wang. 2014. Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS). In ACM SIGKDD.
10. Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. 2014. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. Proc. SIGIR. 83–92
11. Zhou, Ke, Shuang-Hong Yang, and Hongyuan Zha. "Functional matrix factorizations for cold-start recommendation." *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, 2011.
12. Wang, Nan, et al. "Explainable recommendation via multi-task learning in opinionated text data." *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 2018.

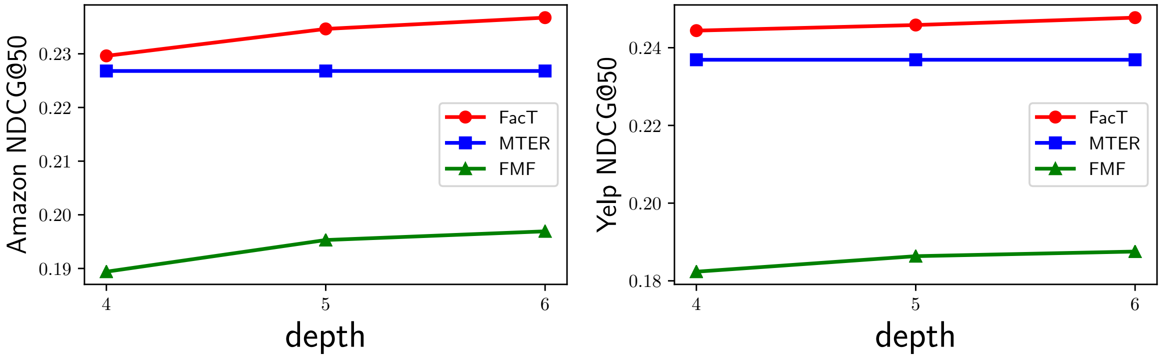
Thanks!
Q & A

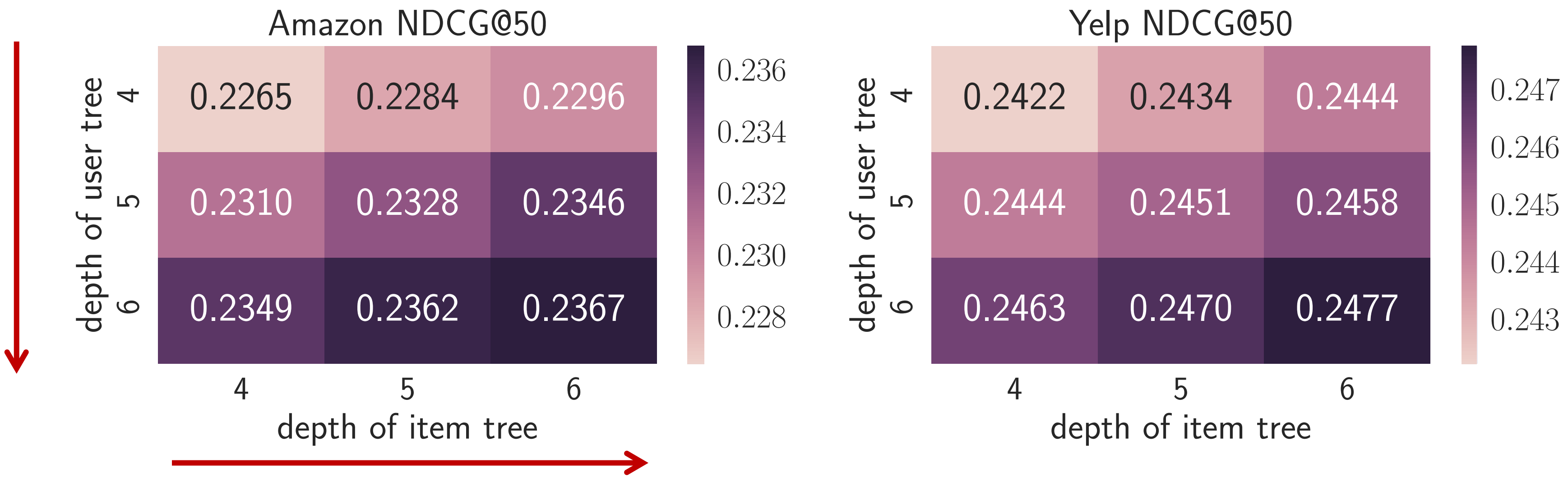
The FacT
https://github.com/yilingjia/TheFacT

Backup Slides

Quantitative Evaluation

Maximum Tree Depth





Sentiment Analysis in Reviews

Obtain user opinion

Feature, Opinion, Sentiment Polarity

(food, good, +1)
(burger, great, +1)
(potato fries, crispy, +1)
(service, awful, -1)
(wait time, long, -1)

→

Count the number of positive and negative sentiment polarity in the review r for user A, and item B.

$$p_{r,A,food} = 1, p_{r,B,food} = 1$$
$$p_{r,A,burger} = 1, p_{r,B,burger} = 1$$
$$\dots$$
$$n_{r,A,wait\ time} = 1, n_{r,B,wait\ time} = 1$$

Sentiment Analysis in Reviews

Obtain user opinion

Feature, Opinion, Sentiment Polarity

(food, good, +1)
(burger, great, +1)
(potato fries, crispy, +1)
(service, awful, -1)
(wait time, long, -1)

$$F_{il}^{v} = \begin{cases} \emptyset, & if\ p_{il}^{u} = n_{il}^{u} = 0 \\ p_{il}^{v} - n_{il}^{v}, & otherwise \end{cases}$$

From all the reviews

For feature l:
p_{Al}^{u}: #positive sentiment polarity.

n_{Al}^{v}: #negative sentiment polarity.

For user $$F_{il}^{u} = \begin{cases} \emptyset, & if\ p_{il}^{u} = n_{il}^{u} = 0 \\ \boxed{p_{il}^{u} + n_{il}^{u},} & otherwise \end{cases}$$

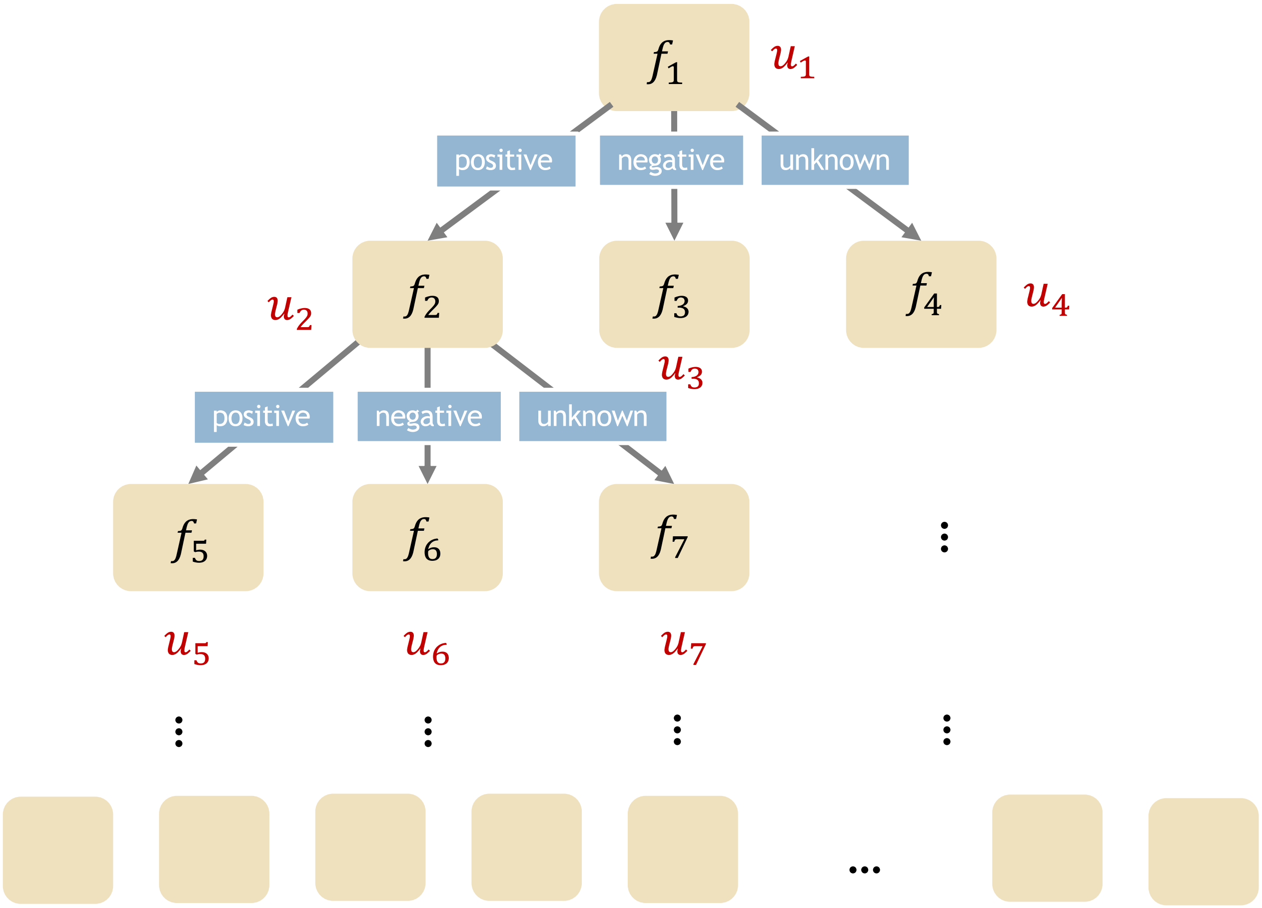
Frequency: capture the relative emphasis that the user i has given to the feature f_l.

For item $$F_{il}^{v} = \begin{cases} \emptyset, & if\ p_{il}^{u} = n_{il}^{u} = 0 \\ \boxed{p_{il}^{v} - n_{il}^{v},} & otherwise \end{cases}$$

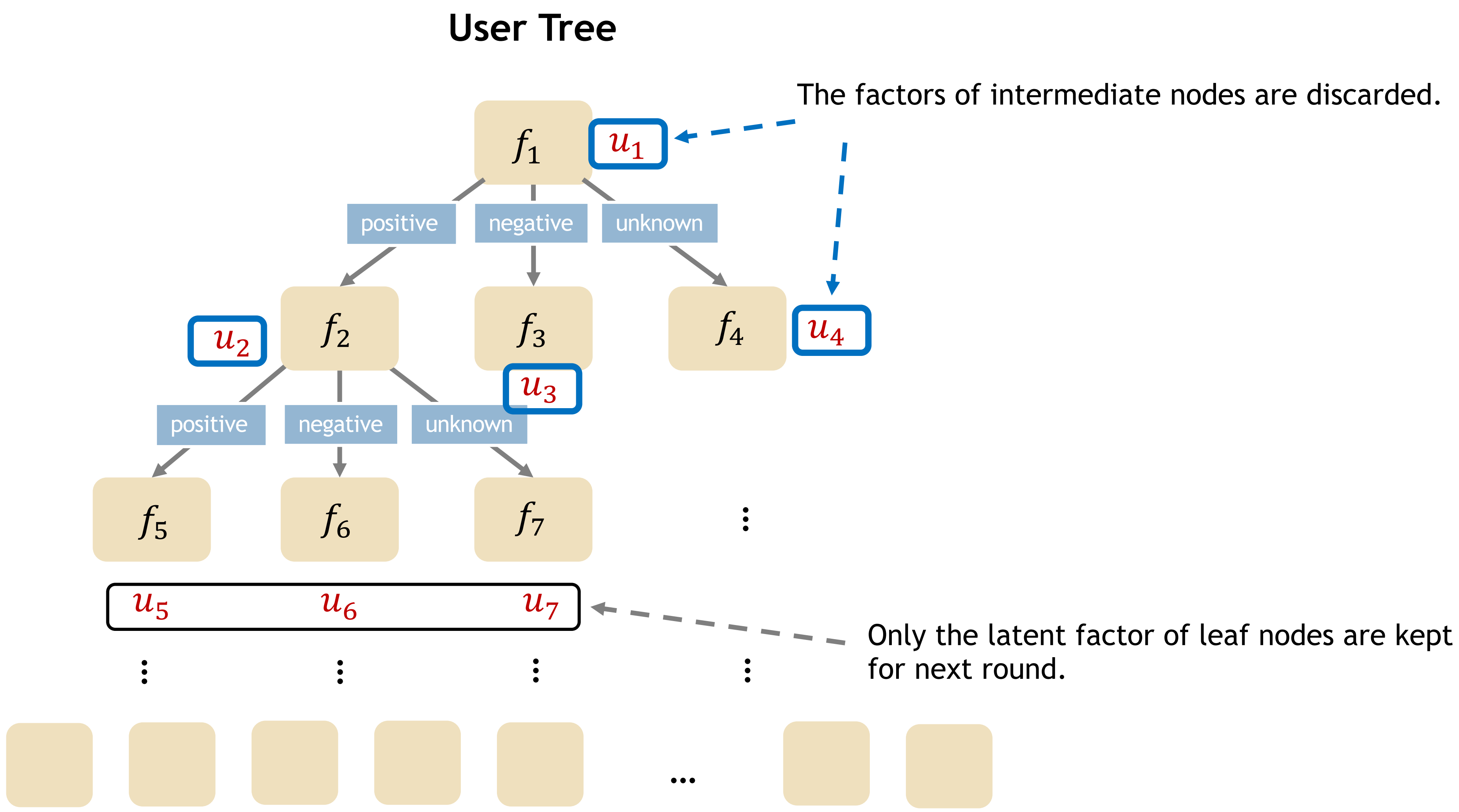
Sentiment opinion: reflect the aggregated user sentiment evaluation about feature f_l of item j.

Tree Construction

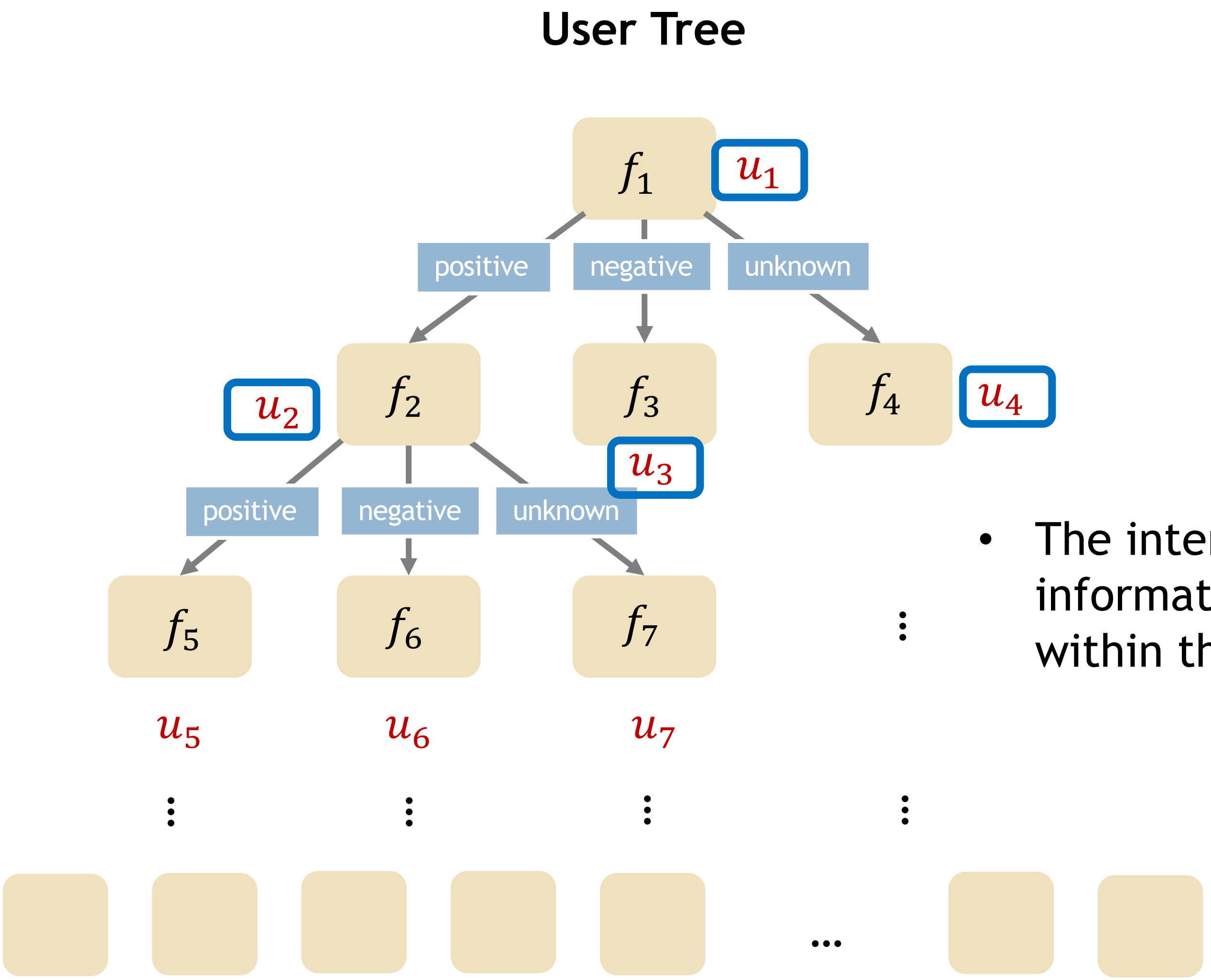
User Tree



Tree Construction

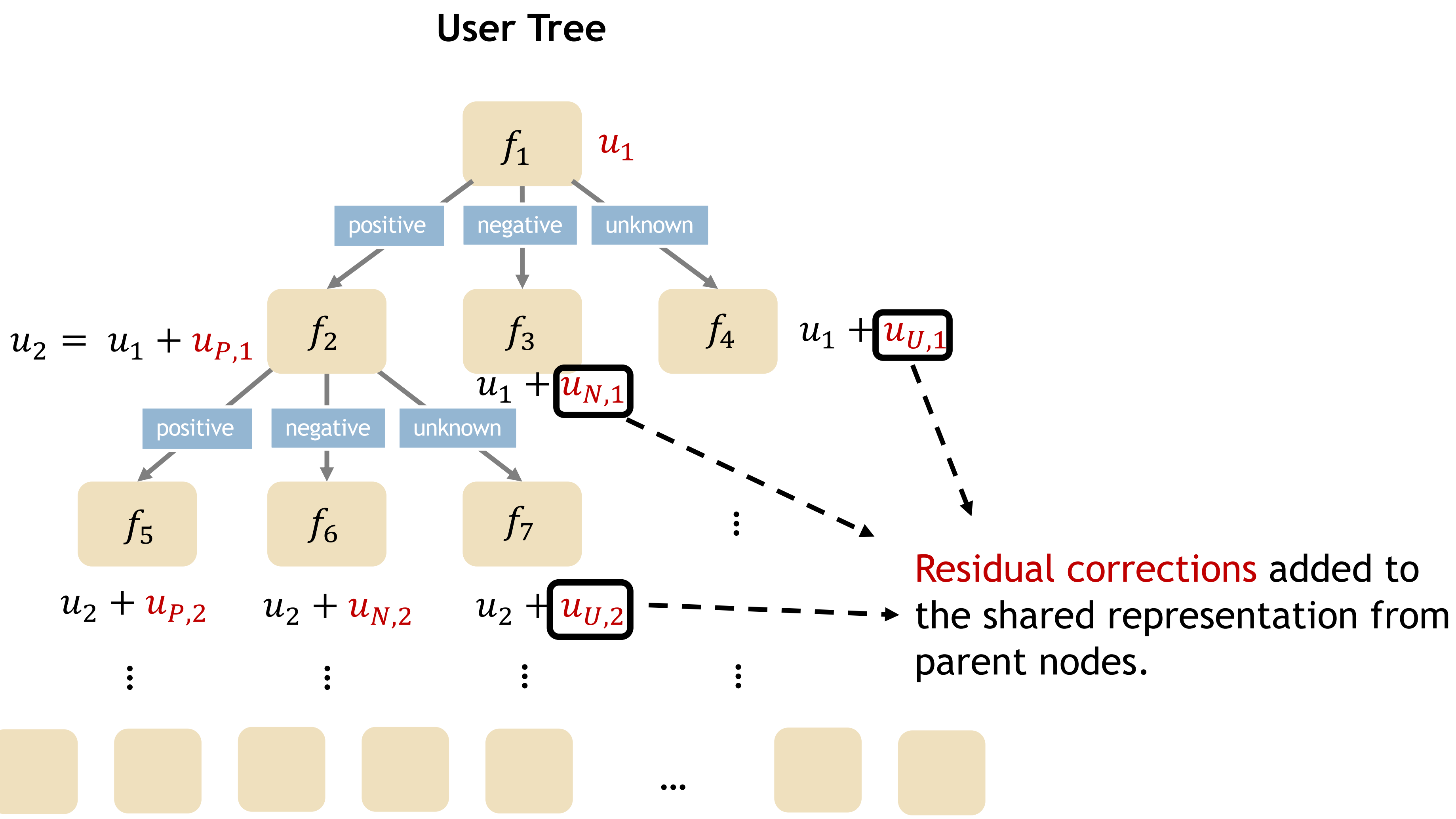


Tree Construction

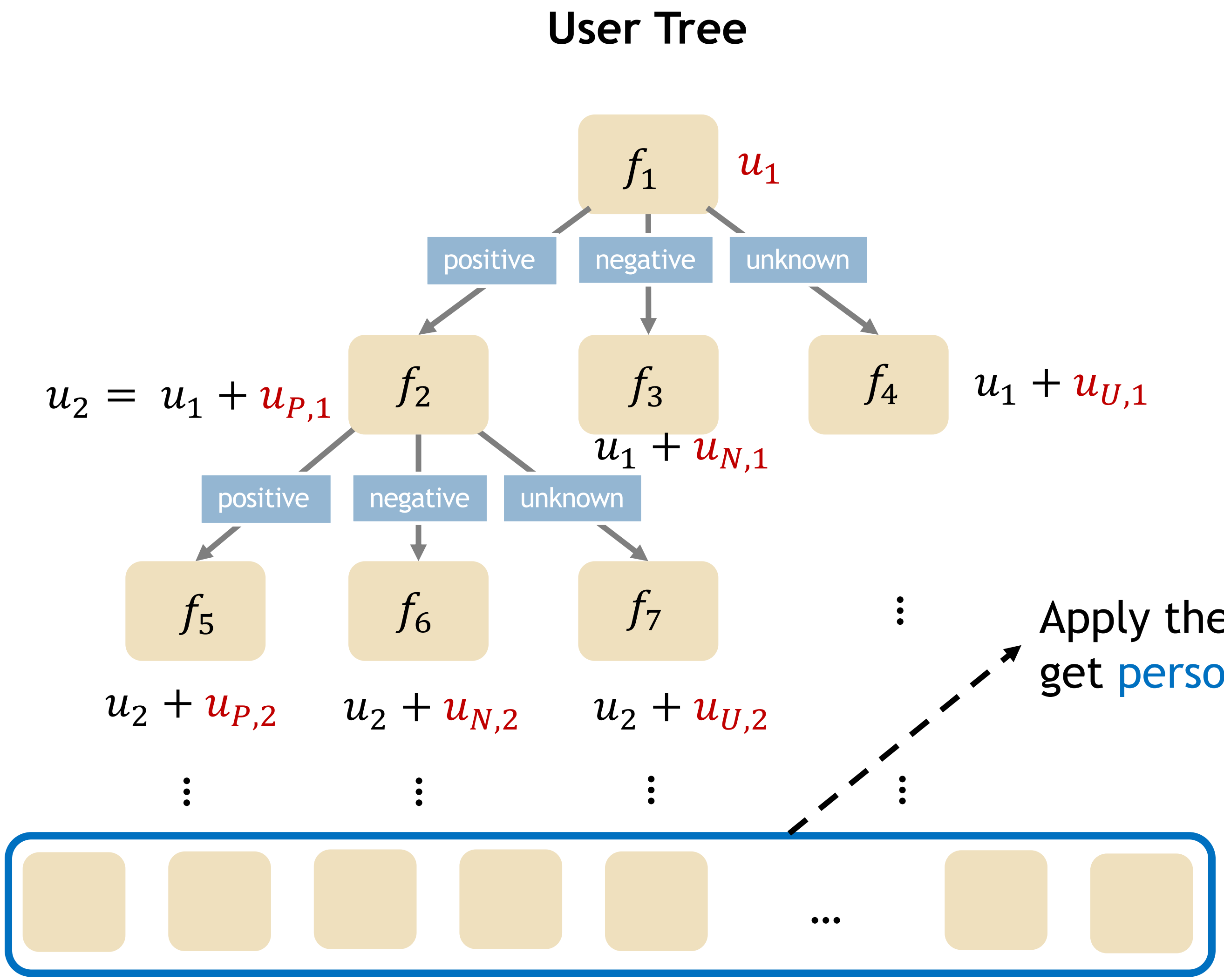


- The intermedia nodes capture the information about homogeneity within the identified user cluster

Tree Construction



Tree Construction



Apply the residual corrections to get personalized representation

Quantitative Evaluation

Inclusion of factors from parent nodes.
PF: Parent Factor