Recommender system meets causality learning

王梦晗 CS PHD (4th-year)
Email： wangmengh@zju.edu.cn
Tel: +1 412-596-5649 (USA) 18768113799 (China)

Outline

1. Research Interests
2. Four Recent Works
3. Other Experiences
4. Current Works

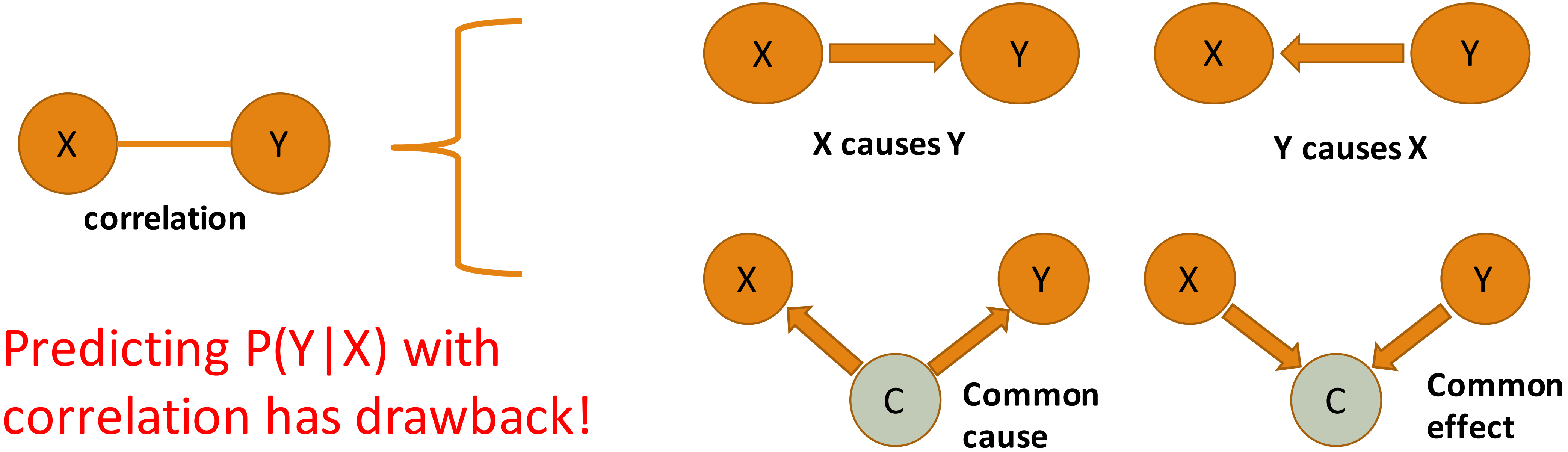
Research Interests

- Recommender systems
  - Widely applied in real applications to predict user behaviors.
  - A hot research area but many issues need further explored.

- Causal inference (causality learning)
  - Explore causation rather than correlation between variables.
  - An emerging and promising research direction in machine learning .

A Brief Intro to Causal Inference

- correlation $\neq$ causation.
  - Four types of relations that correlation can not tell.



Predicting P(Y|X) with correlation has drawback!

A Brief Intro to Causal Inference

- Characteristics:
  - prefer P(X,Y) than p(Y|X) because X may be domain specific, changing, or suffering from selection bias.
  - care more about the design of metrics.

- Barriers between theory to practice in big data.
  - Causal Markov assumption & causal faithfulness assumption -- too strong.
  - conditional independence -- computational Inefficiency.

Causal Thinking in Recommendations

- Some points are ignored by traditional RecSys:
  1. heterogeneous users.
  2. asymmetric relations among products.
  3. causal process of user behavior.
  4. what are the true effects of recommender systems?

Causal Thinking in Recommendations

Pros of Recommendations + Causal inference

- more reasonable and explainable.
- provide unbiased predictions.
- human prior knowledge can be incorporated.

My Four Recent Works

- My research methodology.
  - Explore the common potential issues behind the "main trends."
  - Combine causal knowledge into recommender systems to make them more reasonable and explainable.
  - Prefer extensible techniques & models.

1. Irrelevant Review Filtering for Recsys: Understanding the Consistency Between Ratings and Reviews.

- Background
  - Utilizing auxiliary information (text, pictures, social relations) to build Recsys becomes a trend.
  - However, most previous works ignore that the quality of data is uneven (e.g., ratings & reviews).

- Intuition
  - Consistency: reviews of a product with same ratings will share some similarities on semantic meaning and word usage.
  - Classify reviews in an unsupervised way.

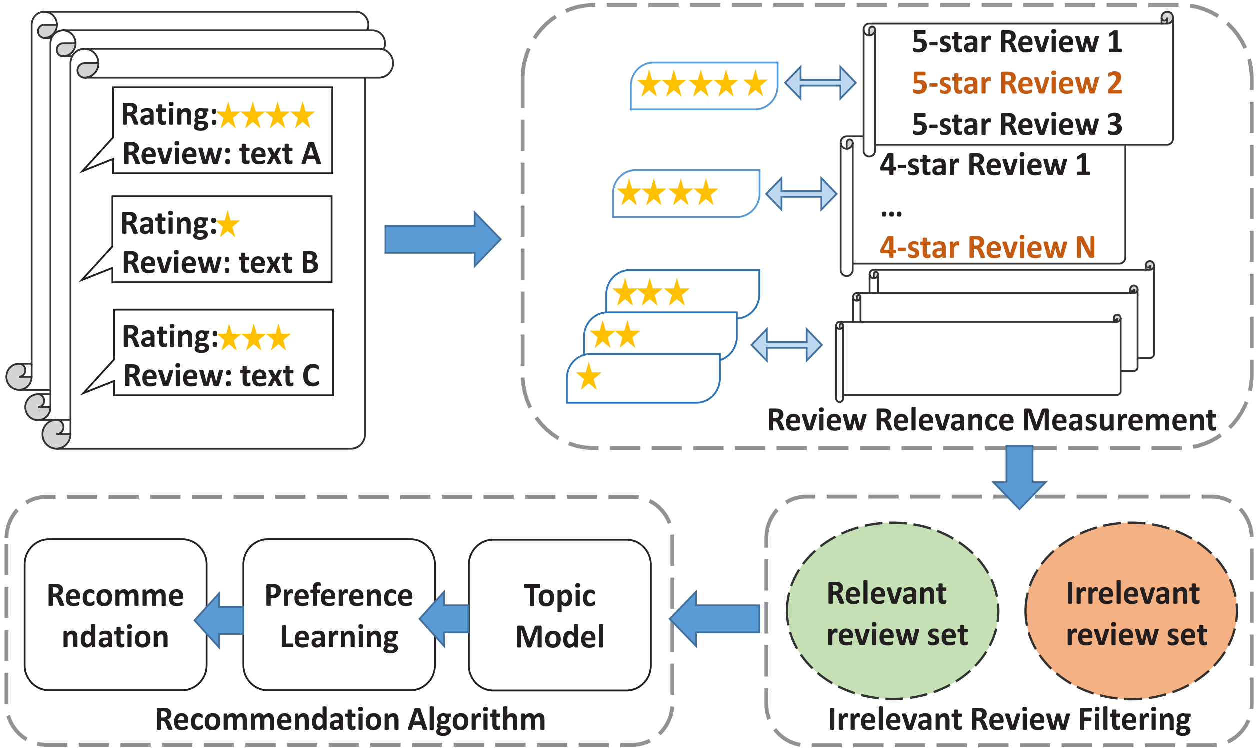
1. Irrelevant Review Filtering for Recsys: Understanding the Consistency Between Ratings and Reviews.

- IRF-Rec model:

We combine the ideas of text classification and latent factor model for recommendation.

- Main procedures:

a) Review relevance measurement
b) Irrelevant review filtering
c) Recommendation algorithm



1. Irrelevant Review Filtering for Recsys: Understanding the Consistency Between Ratings and Reviews.

a). Review relevance measurement

- Implemented via a combination of two Fisher Kernels: one for topic-level similarity and one for word-level similarity.
- Pros: can process variable length reviews.

b). Irrelevant review filtering

- One-class SVM with aforementioned Fisher Kernel.
- Pros: can filter different types of irrelevant reviews.

1. Irrelevant Review Filtering for Recsys: Understanding the Consistency Between Ratings and Reviews.

c). Recommendation algorithm

- Two strategies to utilize irrelevant reviews:
  1) utilize the reviews and ratings with a lower weight;
  2) abandon them.

- Reviews are learned for user and item features via topic modeling.
- Ratings are utilized as follows (the weight c):

$$min_{b^*,p^*,q^*} \sum_{(u,i)\in R^+} (r_{u,i}-\hat{r}_{u,i})^2 + \sum_{(u,i)\in R^-} c(r_{u,i}-\hat{r}_{u,i})^2 + \Omega.$$

$$\Omega = \lambda_1(b_u^2 + b_i^2) + \lambda_2(\|q_u\|^2 + \|p_i\|^2).$$

1. Irrelevant Review Filtering for Recsys: Understanding the Consistency Between Ratings and Reviews.

IRF-Rec effectiveness (tested on 4 Amazon datasets).

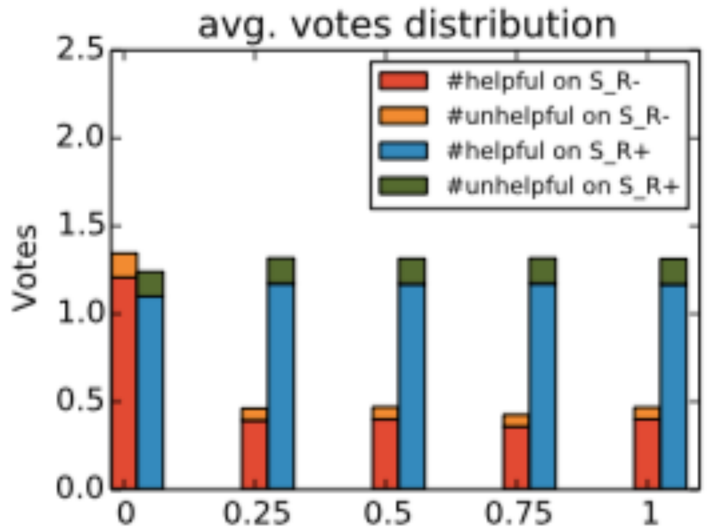
- Classification performance:

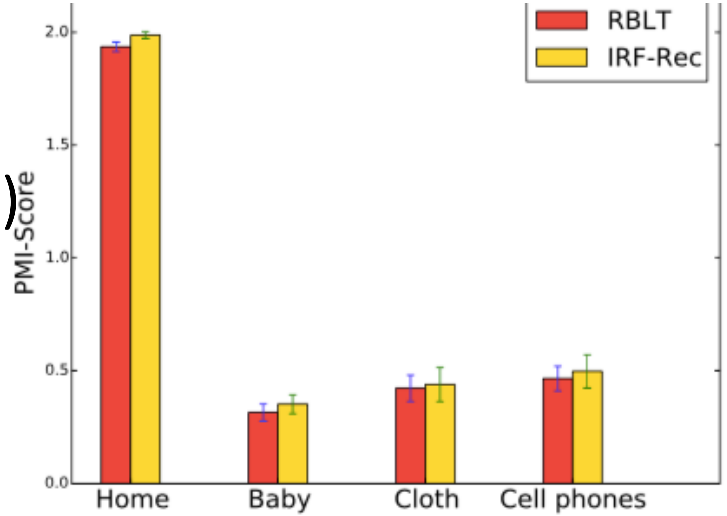
- Helpful votes. relevant : irrelevant reviews = 3.26:1
- Topic coherence. PMI-score improvement: + 5.4%

- Recommendation performance:

- Recall@10 IRF-Rec vs best baseline +14.03%.
- Recall@50 IRF-Rec vs best baseline +7.02%.

(Note that the best baseline assumes data is of equal quality)





1. Irrelevant Review Filtering for Recsys: Understanding the Consistency Between Ratings and Reviews.

➢ Conclusions.

- data quality prioritizes data quantity in some sense.
- unsupervised method

➢ Possible extensions.

- consistency among multi aspects of user generated content (pic, tag, link …).

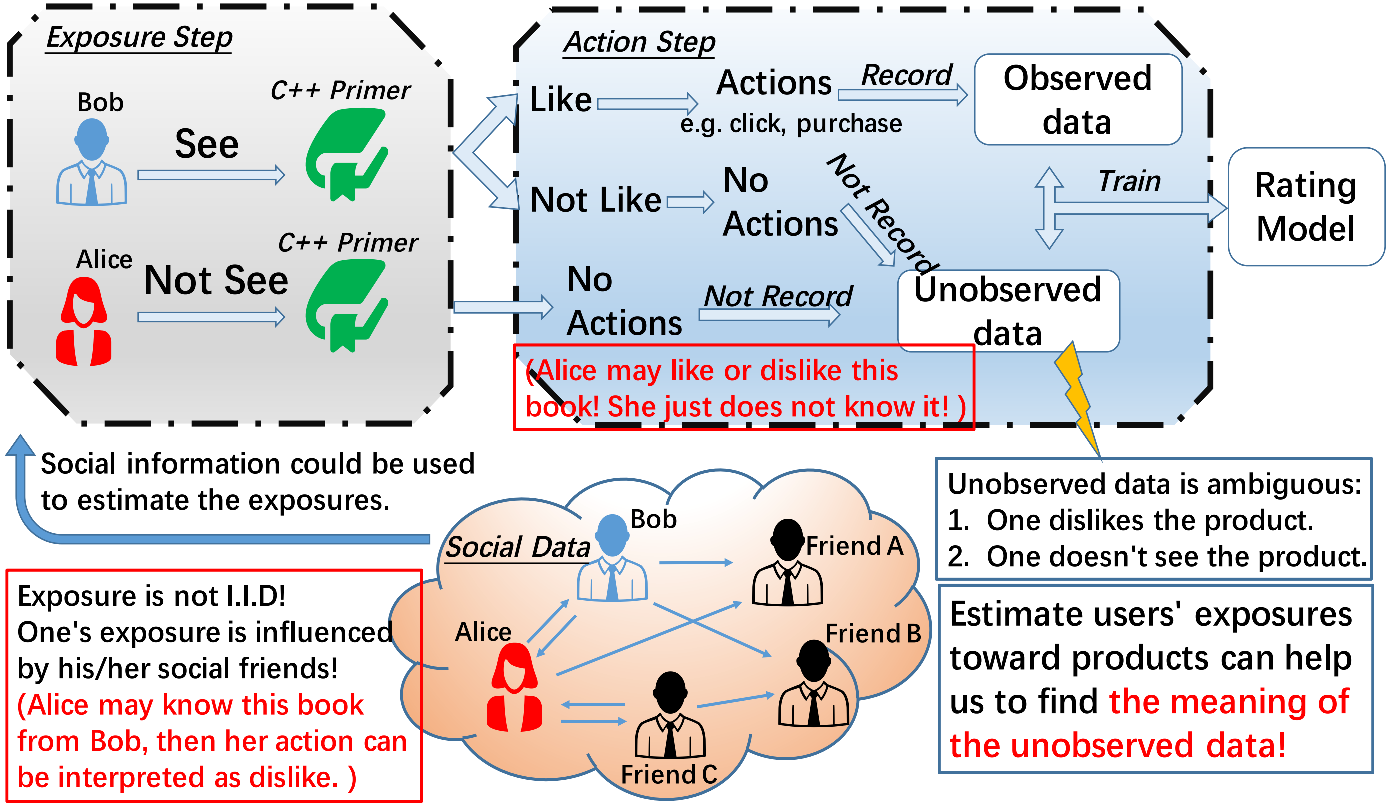
2. Collaborative Filtering with Social Exposure: A Modular Approach to Social Recommendation.

- Background
  - Previous social recommendations assume “similar preferences among social friends.”
  - The assumption is too strong in reality.

- Intuition
  - We get information through friends.
  - They influence our exposures towards products.

2. Collaborative Filtering with Social Exposure: A Modular Approach to Social Recommendation.

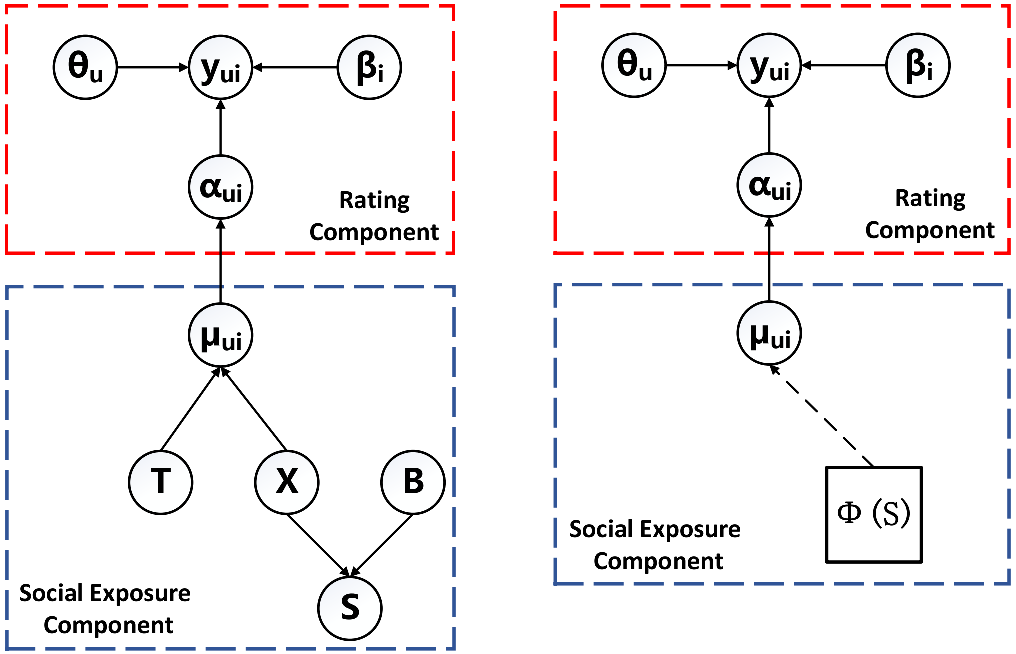
- A causal procedure for Rec
  - User have to first seen the products. (Exposure step)
  - Then they have a chance to take action on the products. (Action step)
- Usage of social information
  - utilize it in exposure step rather than action step.



2. Collaborative Filtering with Social Exposure: A Modular Approach to Social Recommendation.

➢ SERec Model

- two components: 1) modeling social exposure 2) predict ratings.
- two implementations of social exposure:
  1) social regularization and
  2) *social boosting*.
- modular and scalable.



(a) SERec with social regularization

(b) SERec with social boosting

2. Collaborative Filtering with Social Exposure: A Modular Approach to Social Recommendation.

- **Rating Component:**

the log joint probability of exposures α_{ui} and click y_{ui} for user u and item i is:

$$\begin{aligned} &\log p(\alpha_{ui}, y_{ui} | \mu_{ui}, \theta_u, \beta_i, \lambda_y^{-1}) \\ &= Bernoulli(\alpha_{ui}|\mu_{ui}) + \alpha_{ui} \log \mathcal{N}(y_{ui}|\theta_u^T \beta_i, \lambda_y^{-1}) \\ &\quad + (1-\alpha_{ui}) \log \mathrm{II}\left[y_{ui}=0\right] \end{aligned}$$

- **Social Regularization:** set the X as a shared variable.

$$\begin{aligned} \mathcal{L}_{sr} = &\sum_{ui}(X_u^T T_i + \gamma_i - \mu_{ui})^2 + \lambda_{sr} \sum_{uk}(X_u^T B_k - S_{uk})^2 \\ &+ \lambda_x \|X_u\|^2 + \lambda_t \|T_i\|^2 + \lambda_b \|B_k\|^2 + \lambda_\gamma \|\gamma_i\|^2, \end{aligned}$$

- **Social Boosting**: a consumer's knowledge of products is boosted by his/her friends.

$$\mu_{ui} = e_{ui} + \Phi(S),$$

$$\Phi(S) = \sum_{f \in Friends(u)} s \cdot \mu_{fi},$$

where e_{ui} is the inner exposure; Φ(S) is the social exposure.

2. Collaborative Filtering with Social Exposure: A Modular Approach to Social Recommendation.

- Model Performance
  - significant improvements (tested on four public datasets).
    - SERec vs best baseline: Recall@10 +28.60%, NDCG@100 +24.36%
    - SERec vs traditional social recsys: Recall@10 +111.42% , NDCG@100 +102.37%

Effectiveness of models								
Dataset	Metrics	baseMF	RSTE	TrustMF	WMF	ExpoMF	$\text{SERec}_{regular}$	SERec_{boost}
Lastfm	recall@10	0.0004	0.0045	0.0731	0.1206	0.1483	0.2117	**0.2159**
	recall@50	0.0013	0.0199	0.1898	0.3081	0.3357	0.3990	**0.4381**
	MAP@100	0.0001	0.0018	0.0208	0.0406	0.0366	0.0501	**0.0527**
	NDCG@100	0.0008	0.0131	0.1261	0.2385	0.2200	0.2937	**0.3102**

2. Collaborative Filtering with Social Exposure: A Modular Approach to Social Recommendation.

- Conclusions
  - Online social friends probably share information in exposures rather than preferences.
  - SERec is modular and scalable.
  - User exposure modeling is a promising approach to the selection bias. (i.e., the missing problem.)

3. Modeling Dynamic Missingness of Implicit Feedback for Recommendation .

- Background
  - The data in Recsys is *missing not at random (MNAR)*.
  - The missing data is a mix of negative and unknown feedback.
  - Current missingness mechanisms are static.

- Intuition
  - Data is missing dynamically.
  - The asymmetric relations among products can be utilized.

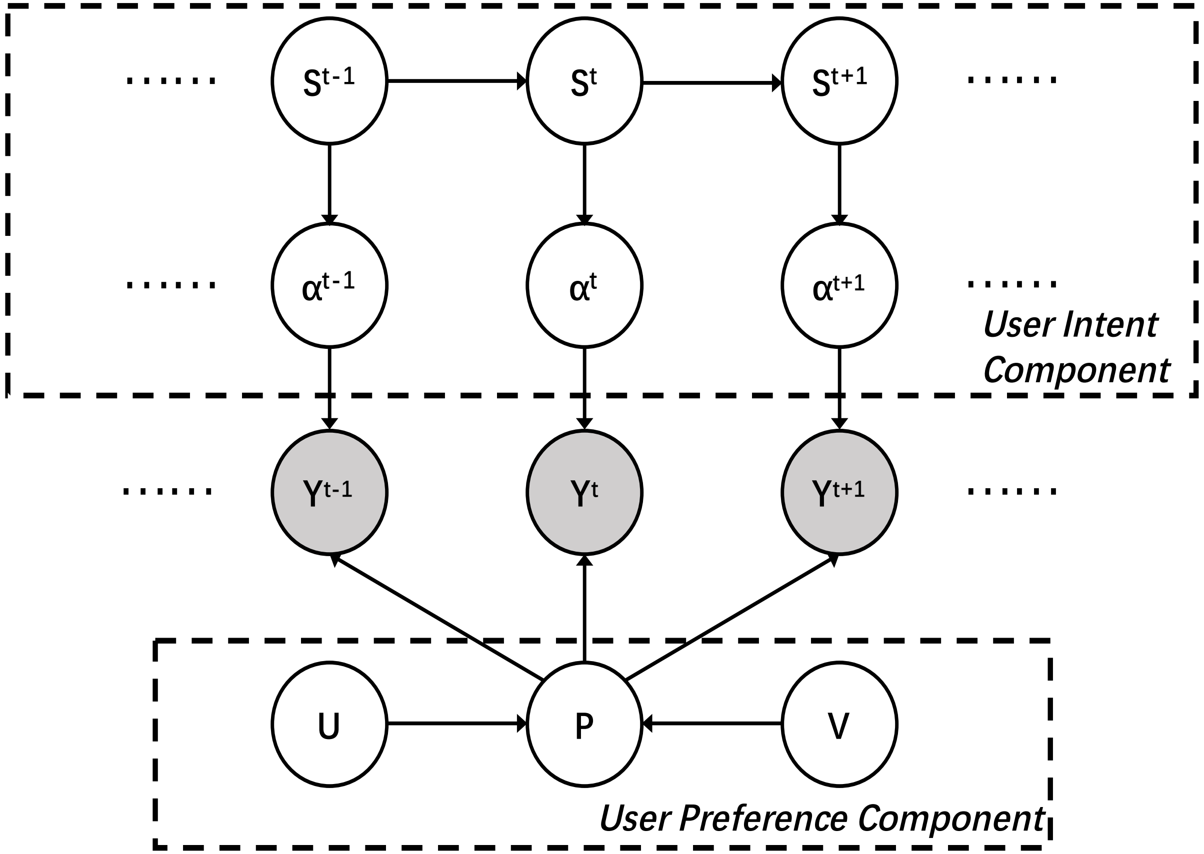
3. Modeling Dynamic Missingness of Implicit Feedback for Recommendation .

An intuitive example:

- The relation between phone and phone accessories is asymmetric.
- A high chance of purchase indicates a low missingness probability

3. Modeling Dynamic Missingness of Implicit Feedback for Recommendation .

- H4MF framework
  - model dynamic missingness via latent variables “user intent” (S)
  - HMM + MF



3. Modeling Dynamic Missingness of Implicit Feedback for Recommendation .

➢ Further constraints on items

- Currently all missingness variables share the symmetric Beta priors
- Inner constraint and outer constraint
- Goal:

$$a^d_{\mathrm{new}} \leftarrow a^d_{\mathrm{ini}} + \sigma^d_j \lambda_{\mathrm{Inner}} + \omega^d_j \lambda_{\mathrm{Outer}}, \quad b^d_{\mathrm{new}} \leftarrow b^d_{\mathrm{ini}} + \lambda_{\mathrm{Inner}} + \lambda_{\mathrm{Outer}}, d \in \{1, ..., D\}$$

$$\omega^d_j = \frac{\text{\#records of item } j \text{ under user intent } d}{\text{\#total records of item } j}$$

$$\sigma^d_j = \frac{\text{\#records of item } j \text{ under user intent } d}{\text{\#total records under user intent } d}$$

3. Modeling Dynamic Missingness of Implicit Feedback for Recommendation.

Parameter Inference

- EM-MAP
- Gibbs sampling for {S}
- M-step: gradient descent

Experimental Result

- $H4MF_c$ vs best baseline: HR@10 +16.74%, NDCG@10 +19.33%

Effectiveness of models							
Dataset	Metrics	PMF	FPMC	WMF	ExpoMF	H4MF	$H4MF_c$
MovieLens-100K	HR@10	0.0031	0.0021	0.1251	0.1230	0.1317	**0.1569**
	HR@50	0.0296	0.0212	0.3968	0.3478	0.3990	**0.4347**
	NDCG@10	0.0011	0.0007	0.0501	0.0616	0.0583	**0.0779**
	NDCG@50	0.0066	0.0046	0.1203	0.1101	0.1205	**0.1367**

3. Modeling Dynamic Missingness of Implicit Feedback for Recommendation.

- Interpretations of user intents
  - Currently all missingness variables share the symmetric Beta priors

User Intent 1		User Intent 2	
Movie Name	Genres	Movie Name	Genres
1. Pulp Fiction	Crime, Drama	1. Little City	Comedy, Romance
2. Fargo	Crime, Drama, Thriller	2. The Whole Wide World	Drama
3. Star Wars	Action, Adventure, Sci-Fi, War	3. Maya Lin: A Strong Clear Vision	Documentary
4. The Full Monty	Comedy	4. Savage Nights	Drama
5. Contact	Drama, Sci-Fi	5. Beat the Devil	Comedy, Drama
6. The English Patient	Drama, Romance, War	6. Ill Gotten Gains	Drama
7. Four Weddings and a Funeral	Comedy, Romance	7. Withnail and I	Comedy
8. The Fugitive	Action, Thriller	8. The Inkwell	Comedy, Drama
9. The Princess Bride	Action, Adventure, Romance	9. Fast, Cheap & Out of Control	Documentary
10. Raiders of the Lost Ark	Action, Adventure	10. Carrington	Drama, Romance

Table 3: Top 10 recommendations for one user on *Movielens-100K* under two user intents.

3. Modeling Dynamic Missingness of Implicit Feedback for Recommendation.

- Conclusions
  - H4MF framework is interpretable.
  - reveal more understandings of user behaviors.
  - user exposure modeling is a promising method to selection bias.

4. User-Sensitive Recommendation Ensemble with Clustered Multi-Task Learning.

- Background
  - Current recommendation ensembles train a global model.
  - User heterogeneity is ignored, which may lead to user skewed problem.
  - Personalized ensemble is costly.

- Intuition
  - learning ensemble for one user can be seen as a task.
  - Clustered Multi-task learning.

4. User-Sensitive Recommendation Ensemble with Clustered Multi-Task Learning.

- User skewed prediction problem.
  - The model with the best average predictive accuracy will leave meaningful subsets of users/items modeled significantly worse than other subsets.
  - An experimental trial on MovieLens-100.

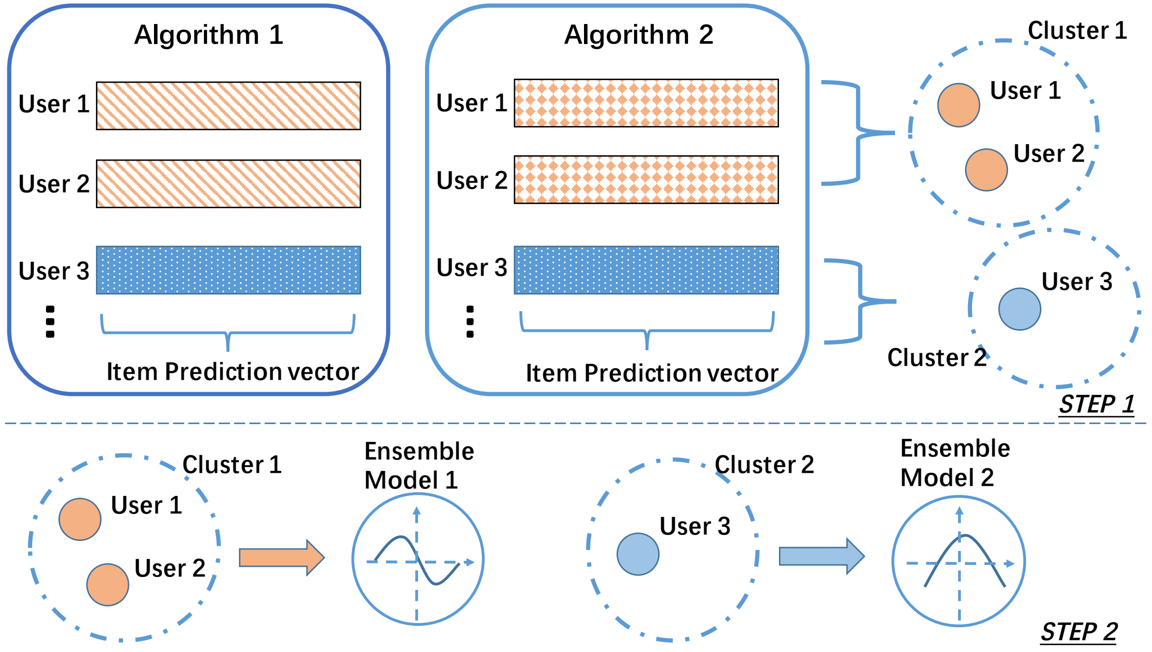
Traditional evaluation	MSE (All users)
KNN	0.9310
MF	0.8806

MSE (user level)	KNN wins MF (46.8% users)	MF wins KNN (53.2% users)
KNN	0.7157 ± 0.3187	1.1791 ± 0.5190
MF	1.1992 ± 0.5431	0.6705 ± 0.2729

4. User-Sensitive Recommendation Ensemble with Clustered Multi-Task Learning.

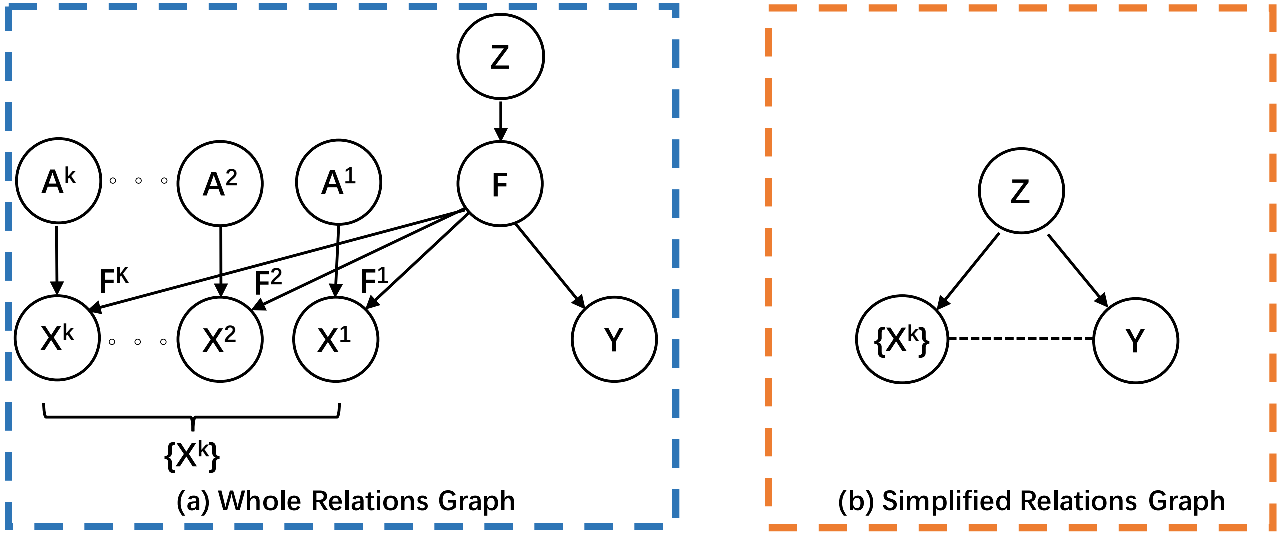
- Basic assumption
  - User heterogeneity is ignored, which may lead to Simpson's paradox.
  - Personalized ensemble is costly.

- Intuition
  - learning ensemble for one user can be seen as a task.
  - Clustered Multi-task learning.



4. User-Sensitive Recommendation Ensemble with Clustered Multi-Task Learning.

- UREC Model.
  - user hidden types **Z**, user features **F**, rec- algorithms **A**, and user-item rating matrix **Y**.
  - For heterogeneous users, **$P(Y|X^K,z)$** may be different for different **z**.
  - Procedures: 1) divide users in groups. 2) learn ensemble strategies in groups.



4. User-Sensitive Recommendation Ensemble with Clustered Multi-Task Learning.

- One implentation.
  - clustering: K-means with a customized distance func:

Let $\mathbf{m}^k_i$ be the perforn

$$d_{Final}(i,q) = \frac{\sum_{k=1}^{K} \frac{m_i^k + m_q^k}{2} d_{iq}^k}{\sum_{k=1}^{K} \frac{m_i^k + m_q^k}{2}}.$$

on user **i**.

users within each group share similar parameters

$$L(W) = \sum_{i=1}^{N} l(w_i^T X_{i*}^K, Y_{i*}) + \Omega(W),$$

$$\Omega(W) = \alpha \sum_{z=1}^{Z} \sum_{v \in \mathbb{I}_z} \|w_v - \bar{w}_z\|^2 + \beta \sum_{z=1}^{Z} \|\bar{w}_z\|^2$$

4. User-Sensitive Recommendation Ensemble with Clustered Multi-Task Learning.

- An approximation implementation. (handle new users)
  - Relax the regularization component with spectral functions of matrices and transforms to a convex problem.

$$\Omega(W,F) = \alpha(\mathrm{Tr}(W^TW) - \mathrm{Tr}(F^TW^TWF)) - \beta\,\mathrm{Tr}(W^TW),$$

$$\Downarrow$$

$$\Omega_{appr}(W,M) = \alpha\eta(1+\eta)\,\mathrm{Tr}(W(\eta I + M)^{-1}W^T),$$

  - optimize parameters with alternating optimization algorithm.

4. User-Sensitive Recommendation Ensemble with Clustered Multi-Task Learning.

- Experimental results.
  - UREC vs Stacking_{one}: MAE -1.56%, RMSE -1.86%
  - UREC vs SVD++: MAE -6.26%, RMSE -6.10%

Effectiveness of models								
Dataset	Metrics	KNN	LFM	SVD++	Stacking_{one}	Stacking_{user}	UREC_{appr}	UREC
MovieLens-100K	MAE	0.7594	0.7329	0.7272	0.7274	0.7340	0.7257	**0.7228**
	RMSE	0.9649	0.9289	0.9212	0.9214	0.9322	0.9226	**0.9190**
MovieLens-1M	MAE	0.7325	0.6869	0.7006	0.6038	0.6027	0.5975	**0.5837**
	RMSE	0.9221	0.8735	0.8854	0.7779	0.7706	0.7657	**0.7514**
Epinions	MAE	0.8570	0.8399	0.8219	0.8124	0.8157	0.8116	**0.8064**
	RMSE	1.1466	1.1192	1.0832	1.0723	1.0705	1.0677	**1.0514**

4. User-Sensitive Recommendation Ensemble with Clustered Multi-Task Learning.

- A real-life example of Simpson's paradox with kidney stone treatments.

- Current recommendation ensembles train a global model.
  - User heterogeneity is ignored, which may lead to Simpson's paradox.
  - Personalized ensemble is costly.

- user skewed prediction
  - learning ensemble for one user can be seen as a task.
  - Clustered Multi-task learning.

	Treatment A	Treatment B
Small stones	*Group 1* **93% (81/87)**	*Group 2* 87% (234/270)
Large stones	*Group 3* **73% (192/263)**	*Group 4* 69% (55/80)
Both	78% (273/350)	**83% (289/350)**

4. User-Sensitive Recommendation Ensemble with Clustered Multi-Task Learning.

- Conclusions.
  - UREC vs $\text{Stacking}_{\text{one}}$: MAE -1.56%, RMSE -1.86%
  - UREC vs SVD++: MAE -6.26%, RMSE -6.10%

4. User-Sensitive Recommendation Ensemble with Clustered Multi-Task Learning.

Possible extensions/insights to industry:

factorize user behavior into user preference + user intent.

A/B testing, more reasonable metrics

Gains beyond Papers

- Efficiency.
  - Code choice: Python -> Cython -> C++ (openmp)
  - speed 30X faster
  - optimization algorithm sgd, compressed

- Learning from other domains.
  - Explore the common potential issues behind the "main trends".

Other Experiences

- Project.
  - Explore the common potential issues behind the "main trends".
  - Combine causal knowledge into recommender a.
- Data mining competitions.
  - Explore the common potential issues behind the "main trends".

Current Works

- New metrics.
  - Explore the common potential issues behind the "main trends".
- Find asymmetric relations among products.
  - Explore the common potential issues behind the "main trends"
- Reinforcement learning + personalized ranking
  - Explore the common potential issues behind the "main trends".

Thanks for Listening

Questions?

Contact：

Menghan Wang (王梦晗)

Email: wangmengh@zju.edu.cn

Wechat & Tel: 18768113799

Twitter: