Identifying Users behind Shared Accounts in Online Streaming Services

Jyun-Yu Jiang†, Cheng-Te Li‡, Yian Chen* and Wei Wang†

†University of California, Los Angeles (UCLA)
‡National Cheng Kung University (NCKU)
*KKBOX Inc.

July 9, 2018 (SIGIR)

Online Streaming Services

Online streaming services are popular nowadays.

However, they might not be free.

- Membership is usually not free.
  - Spotify charges $9.99 per month
  - Netflix charges $7.99 per month
  - Hulu charges $7.99 per month
  - Amazon charges $99 per year
  - $\cdots$
- Tendency to save money by sharing accounts

Some users may choose to share one account!

However, they might not be free.

- Membership is usually not free.
  - Spotify charges $9.99 per month
  - Netflix charges $7.99 per month
  - Hulu charges $7.99 per month
  - Amazon charges $99 per year
  - ...
- Tendency to save money by sharing accounts

Some users may choose to share one account!

However, they might not be free.

- Membership is usually not free.
  - Spotify charges $9.99 per month
  - Netflix charges $7.99 per month
  - Hulu charges $7.99 per month
  - Amazon charges $99 per year
  - . . .
- Tendency to save money by sharing accounts

Some users may choose to share one account!

Account sharing can be a serious issue!

Lost Revenue

- When n users share an account, $n-1$ fees are lost.
- Policy violation

Personalized Recommenders

- Transactions of an account are a mixture of multi-user activities.
- Unsatisfactory recommendations

Identifying users behind shared accounts is important!

Account sharing can be a serious issue!

Lost Revenue

- When n users share an account, $n-1$ fees are lost.
- Policy violation

Personalized Recommenders

- Transactions of an account are a mixture of multi-user activities.
- Unsatisfactory recommendations

Identifying users behind shared accounts is important!

Account sharing can be a serious issue!

Lost Revenue

- When n users share an account, $n - 1$ fees are lost.
- Policy violation

Personalized Recommenders

- Transactions of an account are a mixture of multi-user activities.
- Unsatisfactory recommendations

Identifying users behind shared accounts is important!

Account sharing can be a serious issue!

Lost Revenue

- When n users share an account, $n-1$ fees are lost.
- Policy violation

Personalized Recommenders

- Transactions of an account are a mixture of multi-user activities.
- Unsatisfactory recommendations

Identifying users behind shared accounts is important!

Motivation: Meta information of items tells stories

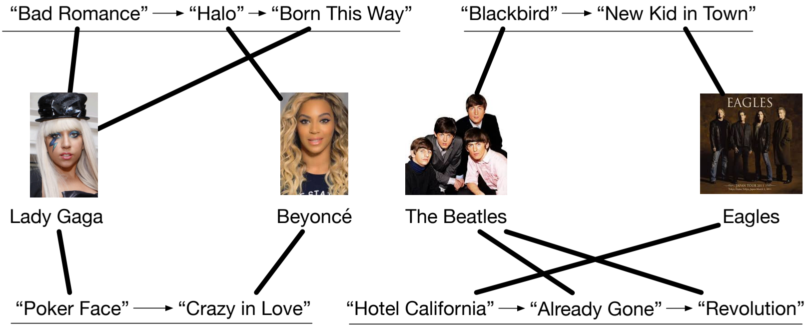
"Bad Romance" → "Halo" → "Born This Way"

"Blackbird" → "New Kid in Town"

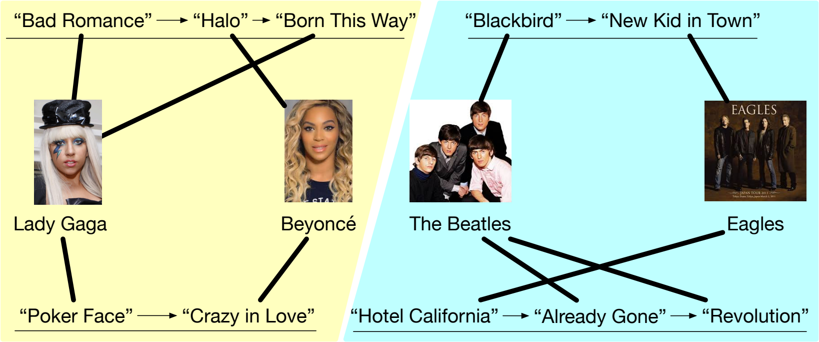
"Poker Face" → "Crazy in Love"

"Hotel California" → "Already Gone" → "Revolution"

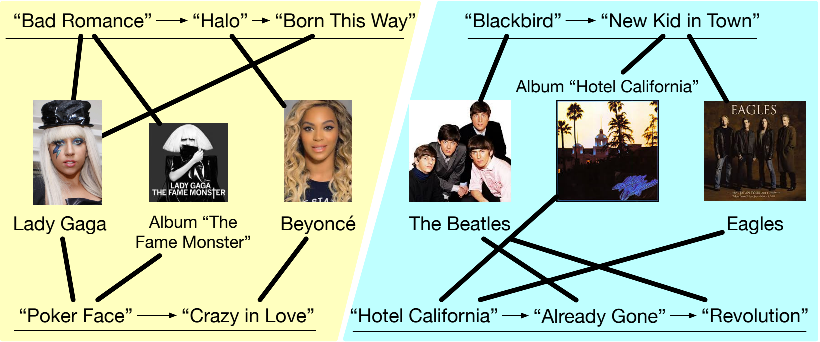
Motivation: Meta information of items tells stories



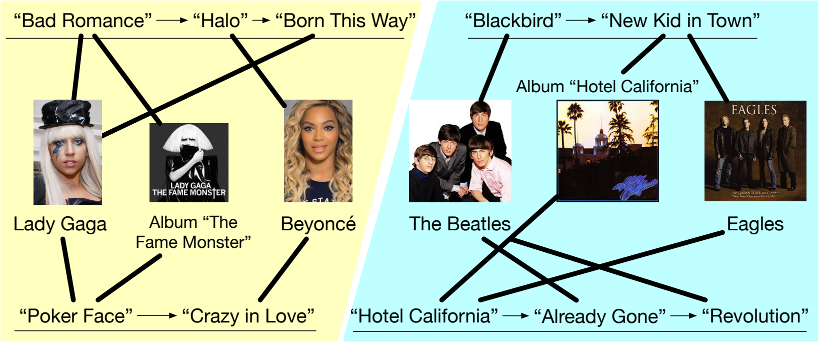
Motivation: Meta information of items tells stories



Motivation: Meta information of items tells stories



Motivation: Meta information of items tells stories



In this work, we exploit meta information of items to identify users.

Problem Definition

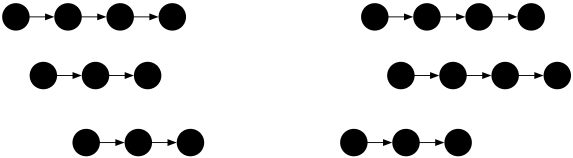
Given an account and its existing sessions, there are two goals.

Problem Definition

Given an account and its existing sessions, there are two goals.

Goal 1: User Identification as Session Clustering (UI-Past)

- Group the given sessions into clusters
  - so that each cluster represents a user.

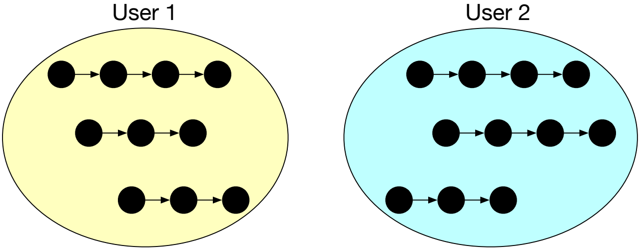


Problem Definition

Given an account and its existing sessions, there are two goals.

Goal 1: User Identification as Session Clustering (UI-Past)

- Group the given sessions into clusters
  - so that each cluster represents a user.

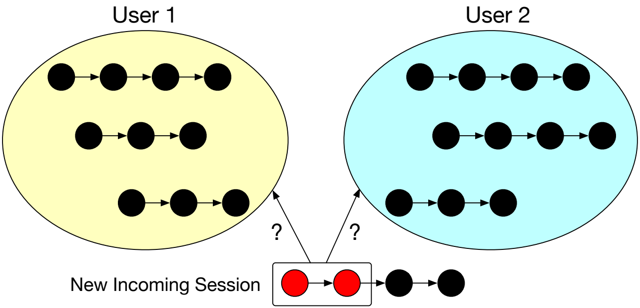


Problem Definition

Given an account and its existing sessions, there are two goals.

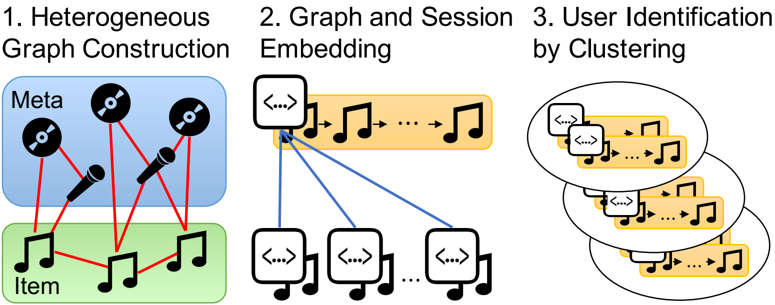
Goal 2: Identifying Users for New Sessions (UI-New)

- Identify the user using only a few preceding items of a new session
  - so that the we can identify the user as early as possible.



Framework Overview of SHE-UI

Session-based Heterogeneous graph Embedding for User Identification(SHE-UI)



For UI-Past (existing sessions)

Treat each cluster as a user

For UI-New (new sessions)

Find the closest cluster

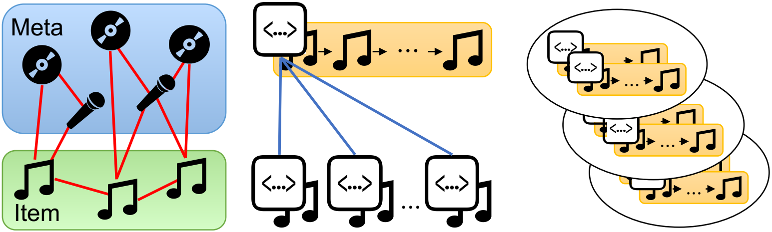
Framework Overview of SHE-UI

Session-based Heterogeneous graph Embedding for User Identification(SHE-UI)

1. Heterogeneous Graph Construction

2. Graph and Session Embedding

3. User Identification by Clustering



For UI-Past (existing sessions)

Treat each cluster as a user

For UI-New (new sessions)

Find the closest cluster

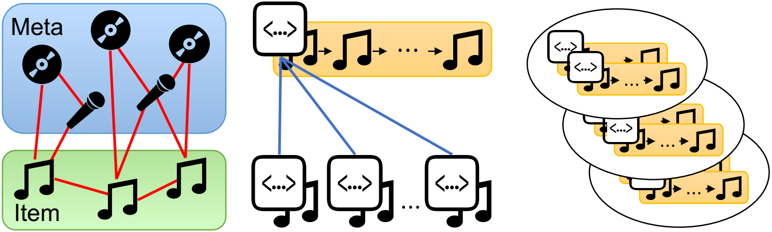
Framework Overview of SHE-UI

Session-based Heterogeneous graph Embedding for User Identification(SHE-UI)

1. Heterogeneous Graph Construction

2. Graph and Session Embedding

3. User Identification by Clustering



For UI-Past (existing sessions)

Treat each cluster as a user

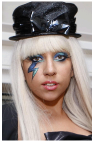
For UI-New (new sessions)

Find the closest cluster

Heterogeneous Graph Construction

- Items and their meta information can be represented by nodes.
- Relationships among items and meta are represented by edges.

"Bad Romance" "Halo" "Born This Way"



Lady Gaga

Album "The Fame Monster"

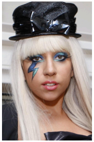
Beyoncé

"Poker Face" "Crazy in Love"

Heterogeneous Graph Construction

- Items and their meta information can be represented by nodes.
- Relationships among items and meta are represented by edges.

"Bad Romance" — "Halo" — "Born This Way"



Lady Gaga

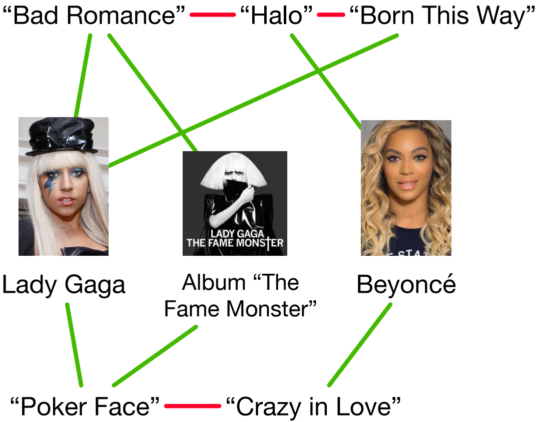
Album "The Fame Monster"

Beyoncé

"Poker Face" — "Crazy in Love"

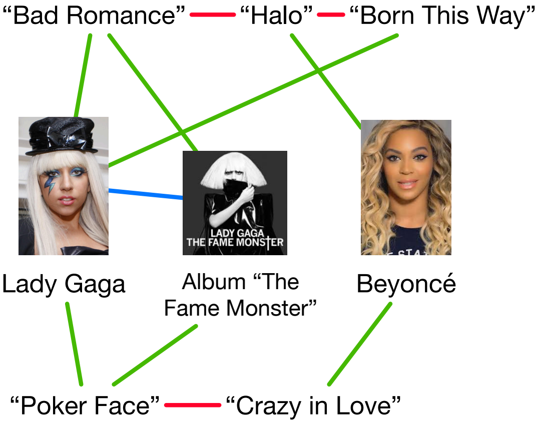
Heterogeneous Graph Construction

- Items and their meta information can be represented by nodes.
- Relationships among items and meta are represented by edges.



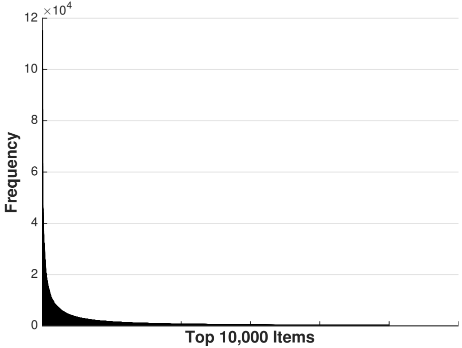
Heterogeneous Graph Construction

- Items and their meta information can be represented by nodes.
- Relationships among items and meta are represented by edges.



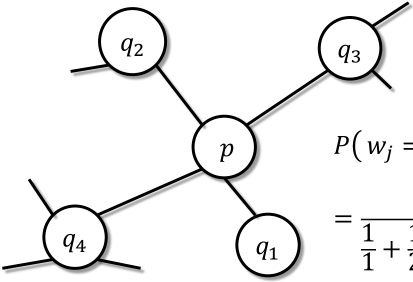
Graph and Session Embedding

- Random walks are commonly utilized for node embedding.
- However, their popularity has a large variance.
  - i.e., some items will be over-optimized.



Normalized Random Walk for Node Embedding

- Normalize probabilities with node degrees

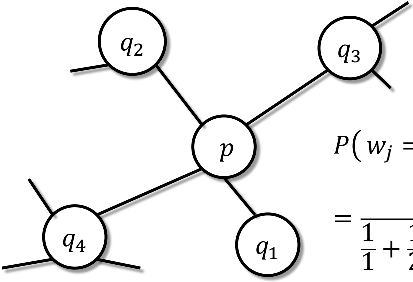


$$P(w_j = q_2 \mid w_{j-1} = p) = \frac{\frac{1}{2}}{\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4}} = 0.24$$

Skip-gram architectures such as DeepWalk can then be applied to learn node embeddings.

Normalized Random Walk for Node Embedding

- Normalize probabilities with node degrees



$$P(w_j = q_2 \mid w_{j-1} = p) = \frac{\frac{1}{2}}{\frac{1}{1}+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}} = 0.24$$

Skip-gram architectures such as DeepWalk can then be applied to learn node embeddings.

Item-based Session Embedding

- Session embedding can be computed by aggregating item embeddings.
- But repeated items in a session may cause issues.
  - 100 play counts v.s. 20 play counts, 1 play count v.s. 2 play counts

Occurrence-Preference Assumption (Gopalan et al., NIPS'14)

The item occurrences is proportional to the square of the preference score.

- The features of the session s can be computed as:

$$f(s) = \frac{1}{\sum_{i \in U(s)} \sqrt{C(s,i)}} \sum_{i \in U(s)} \sqrt{C(s,i)} \cdot f(i).$$

We then cluster the sessions in the item-based session embedding space.

Item-based Session Embedding

- Session embedding can be computed by aggregating item embeddings.
- But repeated items in a session may cause issues.
  - 100 play counts v.s. 20 play counts, 1 play count v.s. 2 play counts

Occurrence-Preference Assumption (Gopalan et al., NIPS'14)

The item occurrences is proportional to the square of the preference score.

- The features of the session s can be computed as:

$$f(s) = \frac{1}{\sum_{i \in U(s)} \sqrt{C(s,i)}} \sum_{i \in U(s)} \sqrt{C(s,i)} \cdot f(i).$$

We then cluster the sessions in the item-based session embedding space.

Item-based Session Embedding

- Session embedding can be computed by aggregating item embeddings.
- But repeated items in a session may cause issues.
  - 100 play counts v.s. 20 play counts, 1 play count v.s. 2 play counts

Occurrence-Preference Assumption (Gopalan et al., NIPS'14)

The item occurrences is proportional to the square of the preference score.

- The features of the session s can be computed as:

$$f(s) = \frac{1}{\sum_{i \in U(s)} \sqrt{C(s,i)}} \sum_{i \in U(s)} \sqrt{C(s,i)} \cdot f(i).$$

We then cluster the sessions in the item-based session embedding space.

Item-based Session Embedding

- Session embedding can be computed by aggregating item embeddings.
- But repeated items in a session may cause issues.
  - 100 play counts v.s. 20 play counts, 1 play count v.s. 2 play counts

Occurrence-Preference Assumption (Gopalan et al., NIPS'14)

The item occurrences is proportional to the square of the preference score.

- The features of the session s can be computed as:

$$f(s) = \frac{1}{\sum_{i \in U(s)} \sqrt{C(s,i)}} \sum_{i \in U(s)} \sqrt{C(s,i)} \cdot f(i).$$

We then cluster the sessions in the item-based session embedding space.

Experimental Settings

- Two datasets
  - Real-world KKBOX dataset
  - Synthetic Last.fm dataset
- Segment logs into sessions with a 30-minute threshold
- Remove inactive accounts and short sessions

(a) Session Information

	Last.fm	KKBOX
existing sessions	209,313	10,783,556
new sessions	209,925	10,782,507
accounts	370	88,399
unique users	922	343,723
items	314,763	564,164

(b) metadata

Last.fm	
artists	60,410
KKBOX	
artists	43,157
albums	253,896
published years	77
genres	48

Baseline Methods of User Identification

Item-based Clustering (Items as features)

- K-Means++ (KM)
- Subspace Clustering (SS)
- Affinity Propagation (AP)

Embedding-based Clustering (Embedding as features)

- word2vec (W2V)
- LINE
- DeepWalk (DW)

User Identification Performance

Dataset	**Synthetic Last.fm**						**Real Data from KKBOX**					
	UI-Past			UI-New			UI-Past			UI-New		
Metric	NMI	MAF	MIF	NMI	MAF	MIF	NMI	MAF	MIF	NMI	MAF	MIF
Known Numbers of Users												
KM	0.2956	0.6109	0.7400	0.2802	0.6106	0.7400	0.3640	0.5710	0.6516	0.3286	0.5644	0.6592
SS	0.2954	0.6109	0.7405	0.2793	0.6105	0.7403	0.3627	0.5707	0.6612	0.3258	0.5642	0.6585
W2V	0.4865	0.7022	0.7982	0.4428	0.6823	0.7769	0.3828	0.5855	0.6524	0.3571	0.5739	0.6488
LINE	0.2667	0.5611	0.6544	0.2622	0.5724	0.6768	0.3830	0.5874	0.6463	0.3456	0.5634	0.6183
DW	0.5597	0.7372	0.8162	0.5148	0.7161	0.7947	0.3995	0.5976	0.6656	0.3587	0.5775	0.6419
SHE-UI	**0.6108**	**0.7613**	**0.8393**	**0.5718**	**0.7455**	**0.8236**	**0.4281**	**0.6111**	**0.6804**	**0.3880**	**0.5948**	**0.6625**
Unknown Numbers of Users												
AP	0.1677	0.3413	0.3474	0.1546	0.4825	0.5408	0.1884	0.4828	0.4978	0.1783	0.5225	0.5569
KM	0.1189	0.5842	**0.7003**	0.1061	0.5622	**0.6697**	0.1856	0.5264	0.5849	0.1516	0.5041	0.5642
SS	0.1518	0.5838	0.6856	0.1312	0.5616	0.6582	0.1927	0.5312	0.5904	0.1841	0.5151	0.5851
W2V	0.2981	0.6413	0.6587	0.2560	0.6148	0.6347	0.2081	0.5337	0.6025	0.1807	0.5149	0.5818
LINE	0.0813	0.5641	0.6687	0.0964	0.5546	0.6552	0.1955	0.5365	0.6083	0.1010	0.4782	0.5394
DW	0.3053	0.6286	0.6557	0.2669	0.5966	0.6244	0.2158	0.5508	0.6249	0.1941	0.5322	0.6024
SHE-UI	**0.3375**	**0.6563**	0.6782	**0.3214**	**0.6323**	0.6568	**0.2426**	**0.5610**	**0.6309**	**0.2218**	**0.5451**	**0.6117**

Application: User-level Recommendation

- Traditional systems can only provide account-level recommendation
  - Represented as $Z_A(a, i)$ for the account a and the item i
- With user identification, user-level recommendation is available.
  - Separately trained for each individual user
  - Denoted as $Z_U(a, i)$
- Two models can further be combined for better performance.

$$Z_C(a, u, i) = (1 - \alpha) \cdot \overline{Z_A}(a, i) + \alpha \cdot \overline{Z_U}(u, i),$$

- α is the parameter to control the weights of two systems.

Application: User-level Recommendation

- Traditional systems can only provide account-level recommendation
  - Represented as $Z_A(a, i)$ for the account a and the item i
- With user identification, user-level recommendation is available.
  - Separately trained for each individual user
  - Denoted as $Z_U(a, i)$
- Two models can further be combined for better performance.

$$Z_C(a, u, i) = (1 - \alpha) \cdot \overline{Z_A}(a, i) + \alpha \cdot \overline{Z_U}(u, i),$$

- α is the parameter to control the weights of two systems.

Application: User-level Recommendation

- Traditional systems can only provide account-level recommendation
  - Represented as $Z_A(a, i)$ for the account a and the item i
- With user identification, user-level recommendation is available.
  - Separately trained for each individual user
  - Denoted as $Z_U(a, i)$
- Two models can further be combined for better performance.

$$Z_C(a, u, i) = (1 - \alpha) \cdot \overline{Z_A}(a, i) + \alpha \cdot \overline{Z_U}(u, i),$$

- α is the parameter to control the weights of two systems.

Application: User-level Recommendation

- Traditional systems can only provide account-level recommendation
  - Represented as $Z_A(a, i)$ for the account a and the item i
- With user identification, user-level recommendation is available.
  - Separately trained for each individual user
  - Denoted as $Z_U(a, i)$
- Two models can further be combined for better performance.

$$Z_C(a, u, i) = (1 - \alpha) \cdot \overline{Z_A}(a, i) + \alpha \cdot \overline{Z_U}(u, i),$$

- α is the parameter to control the weights of two systems.

User-level Recommendation

Baseline Methods

- Most Popular Recommendation (PopRec)
- Maximum Margin Matrix Factorization (MMMF)
- Bayesian Personalized Ranking Matrix Factorization (BPRMF)
- Collaborative Less-is-More Filtering (CLiMF)

Evaluation Method

- Rank all items and consider occurred items as relevant instances for each testing session.
- Sparse and pretty difficult

Performance of User-level Recommendation

Our approach is combined with BPRMF.

	PopRec	MMMF	BPRMF	CLiMF	Ours ($\alpha = 0.6$)
MRR	0.1242	0.1421	0.1353	0.1400	**0.1727 (+23.30%)**
MAP	0.0317	0.0331	0.0330	0.0337	**0.0439 (+30.03%)**
P@1	0.0597	0.0608	0.0577	0.0597	**0.0846 (+41.88%)**

Conclusions

- Focused on a novel task of user identification behind shared accounts
- Proposed an approach based on heterogeneous graph embedding
- Proposed to improve recommenders using user identification
- Extensive experiments on both synthetic and real-world datasets
- Outperformed several competitive baselines
- See our paper for more detailed parameter sensitivity experiments

Thanks for your attention! Questions?

Conclusions

- Focused on a novel task of user identification behind shared accounts
- Proposed an approach based on heterogeneous graph embedding
- Proposed to improve recommenders using user identification
- Extensive experiments on both synthetic and real-world datasets
- Outperformed several competitive baselines
- See our paper for more detailed parameter sensitivity experiments

Thanks for your attention! Questions?

Conclusions

- Focused on a novel task of user identification behind shared accounts
- Proposed an approach based on heterogeneous graph embedding
- Proposed to improve recommenders using user identification
- Extensive experiments on both synthetic and real-world datasets
- Outperformed several competitive baselines
- See our paper for more detailed parameter sensitivity experiments

Thanks for your attention! Questions?

Conclusions

- Focused on a novel task of user identification behind shared accounts
- Proposed an approach based on heterogeneous graph embedding
- Proposed to improve recommenders using user identification
- Extensive experiments on both synthetic and real-world datasets
- Outperformed several competitive baselines
- See our paper for more detailed parameter sensitivity experiments

Thanks for your attention! Questions?

Conclusions

- Focused on a novel task of user identification behind shared accounts
- Proposed an approach based on heterogeneous graph embedding
- Proposed to improve recommenders using user identification
- Extensive experiments on both synthetic and real-world datasets
- Outperformed several competitive baselines
- See our paper for more detailed parameter sensitivity experiments

Thanks for your attention! Questions?

Conclusions

- Focused on a novel task of user identification behind shared accounts
- Proposed an approach based on heterogeneous graph embedding
- Proposed to improve recommenders using user identification
- Extensive experiments on both synthetic and real-world datasets
- Outperformed several competitive baselines
- See our paper for more detailed parameter sensitivity experiments

Thanks for your attention! Questions?

Conclusions

- Focused on a novel task of user identification behind shared accounts
- Proposed an approach based on heterogeneous graph embedding
- Proposed to improve recommenders using user identification
- Extensive experiments on both synthetic and real-world datasets
- Outperformed several competitive baselines
- See our paper for more detailed parameter sensitivity experiments

Thanks for your attention! Questions?

Conclusions

- Focused on a novel task of user identification behind shared accounts
- Proposed an approach based on heterogeneous graph embedding
- Proposed to improve recommenders using user identification
- Extensive experiments on both synthetic and real-world datasets
- Outperformed several competitive baselines
- See our paper for more detailed parameter sensitivity experiments

Thanks for your attention! Questions?