PRINCE: **PR**ovider-side **IN**terpretability with **C**ounterfactual **E**xplanations in Recommender Systems

Azin Ghazimatin, Oana Balalau, **Rishiraj Saha Roy**, and Gerhard Weikum

Max Planck Institute for Informatics, Saarbruecken, Germany

PRINCE: **PR**ovider-side **IN**terpretability with **C**ounterfactual **E**xplanations in Recommender Systems

aka

On the Feasibility and Usefulness of Action-based Explanations in Recommenders

Azin Ghazimatin, Oana Balalau, **Rishiraj Saha Roy**, and Gerhard Weikum

Max Planck Institute for Informatics, Saarbruecken, Germany

Problem: Why did I receive this recommendation?

rec

u

Problem: Why did I receive this recommendation?

u

rec

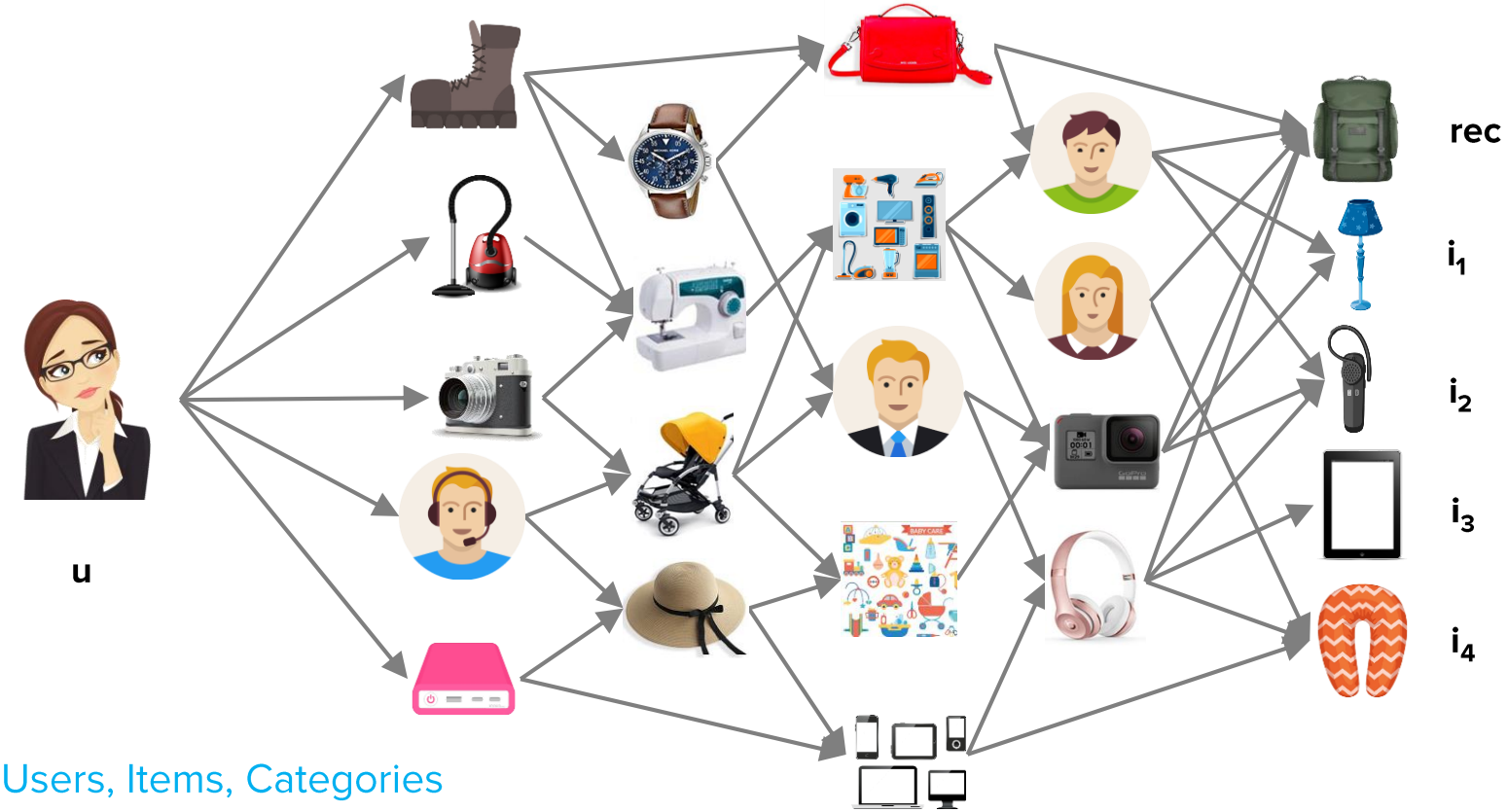
i_1

i_2

i_3

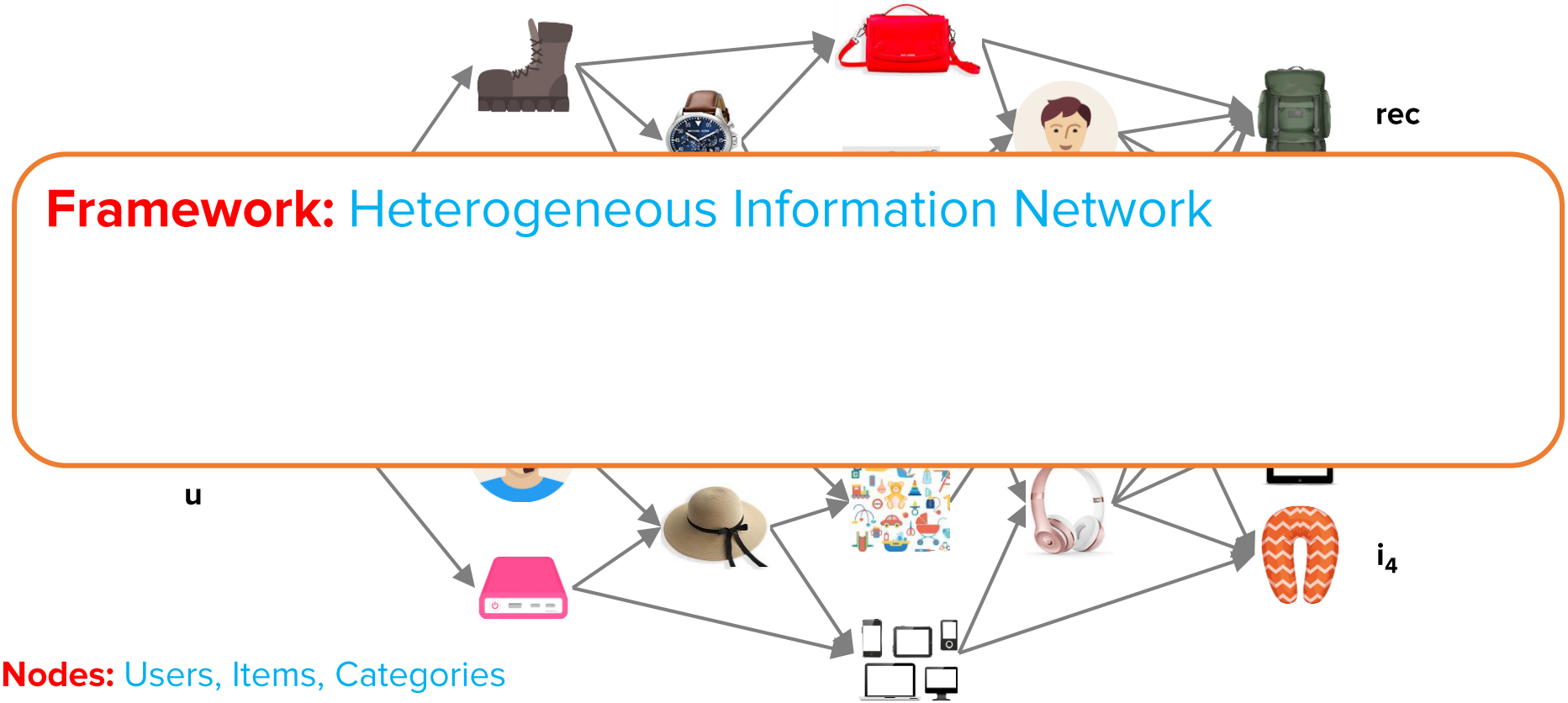
i_4

Problem Setup



Nodes: Users, Items, Categories
Edges: Actions or Similarity, Directed, Weighted

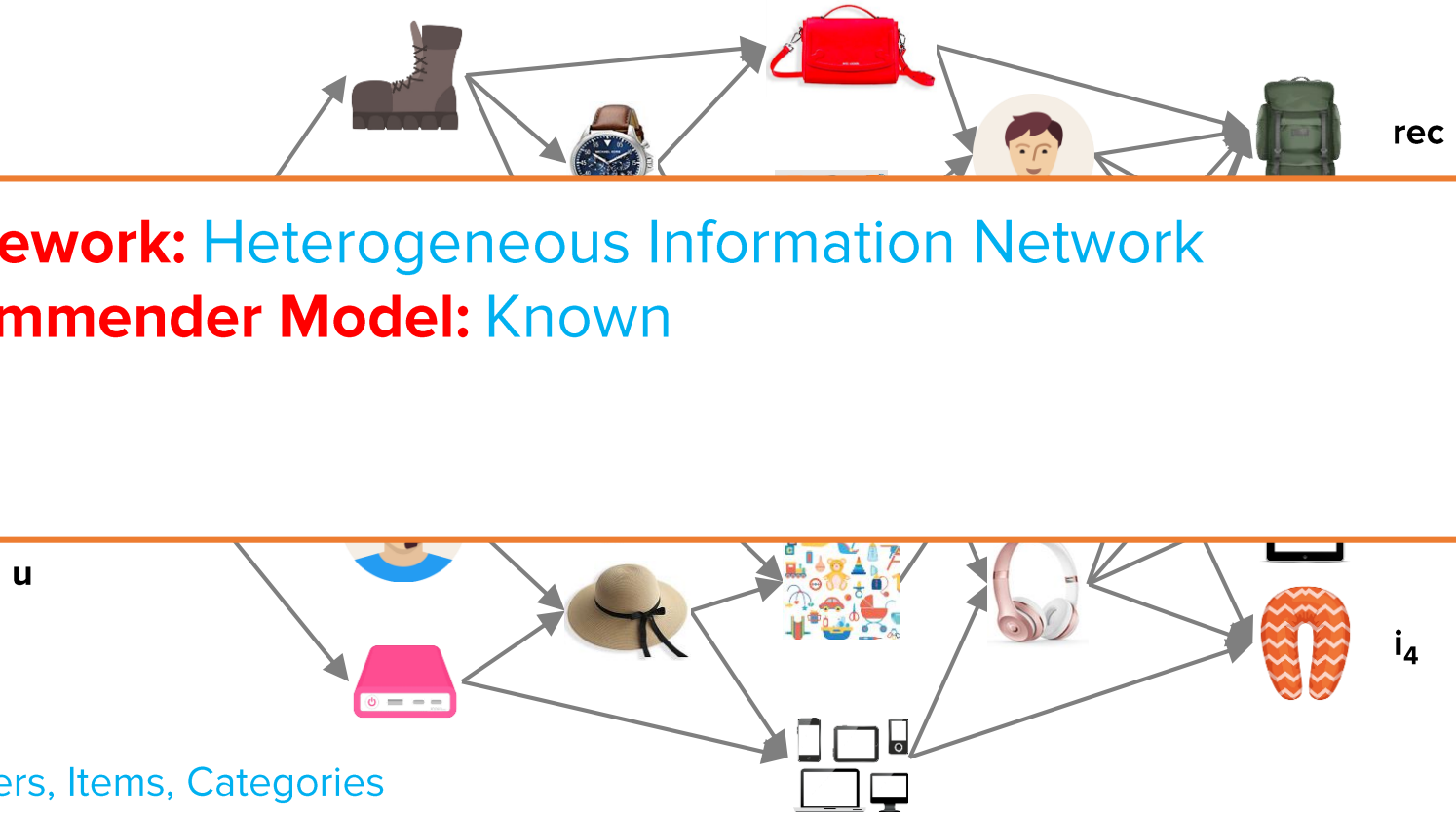
Problem Setup



Framework: Heterogeneous Information Network

Nodes: Users, Items, Categories
Edges: Actions or Similarity, Directed, Weighted

Problem Setup



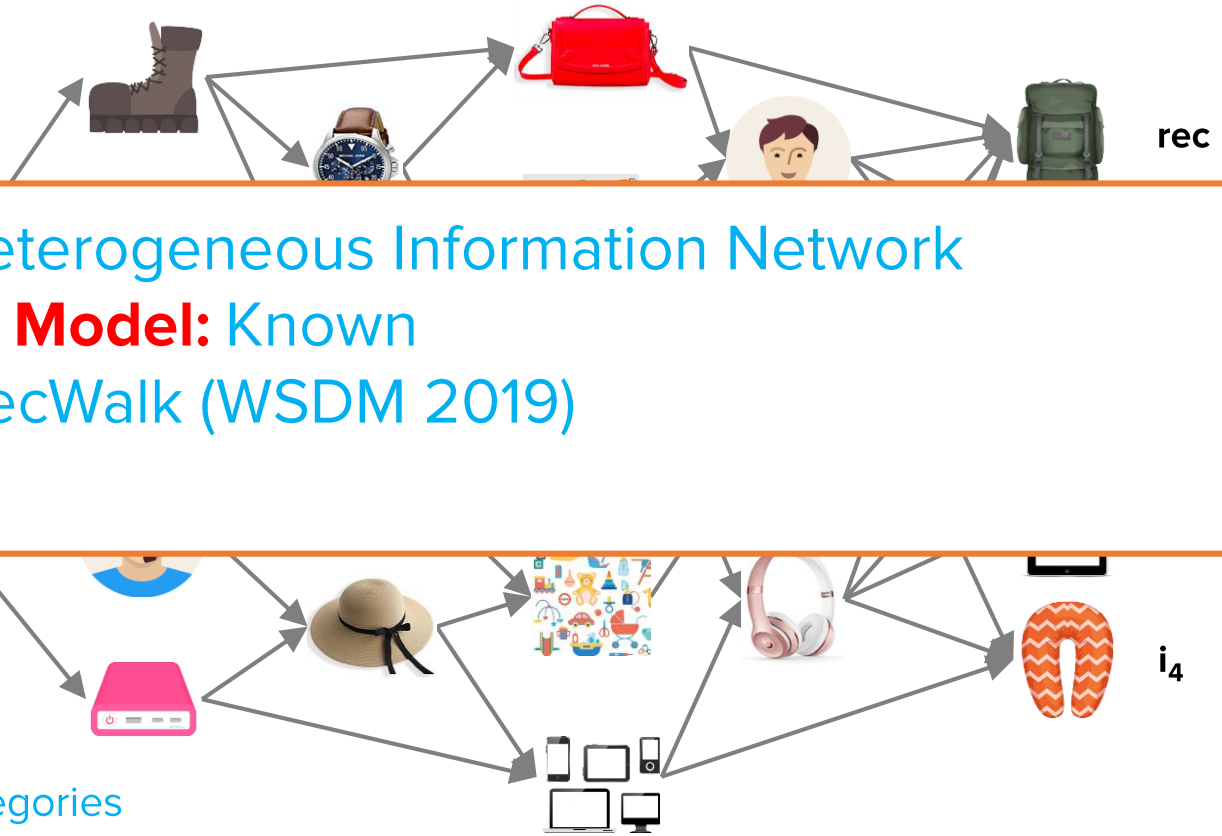
Framework: Heterogeneous Information Network

Recommender Model: Known

Nodes: Users, Items, Categories

Edges: Actions or Similarity, Directed, Weighted

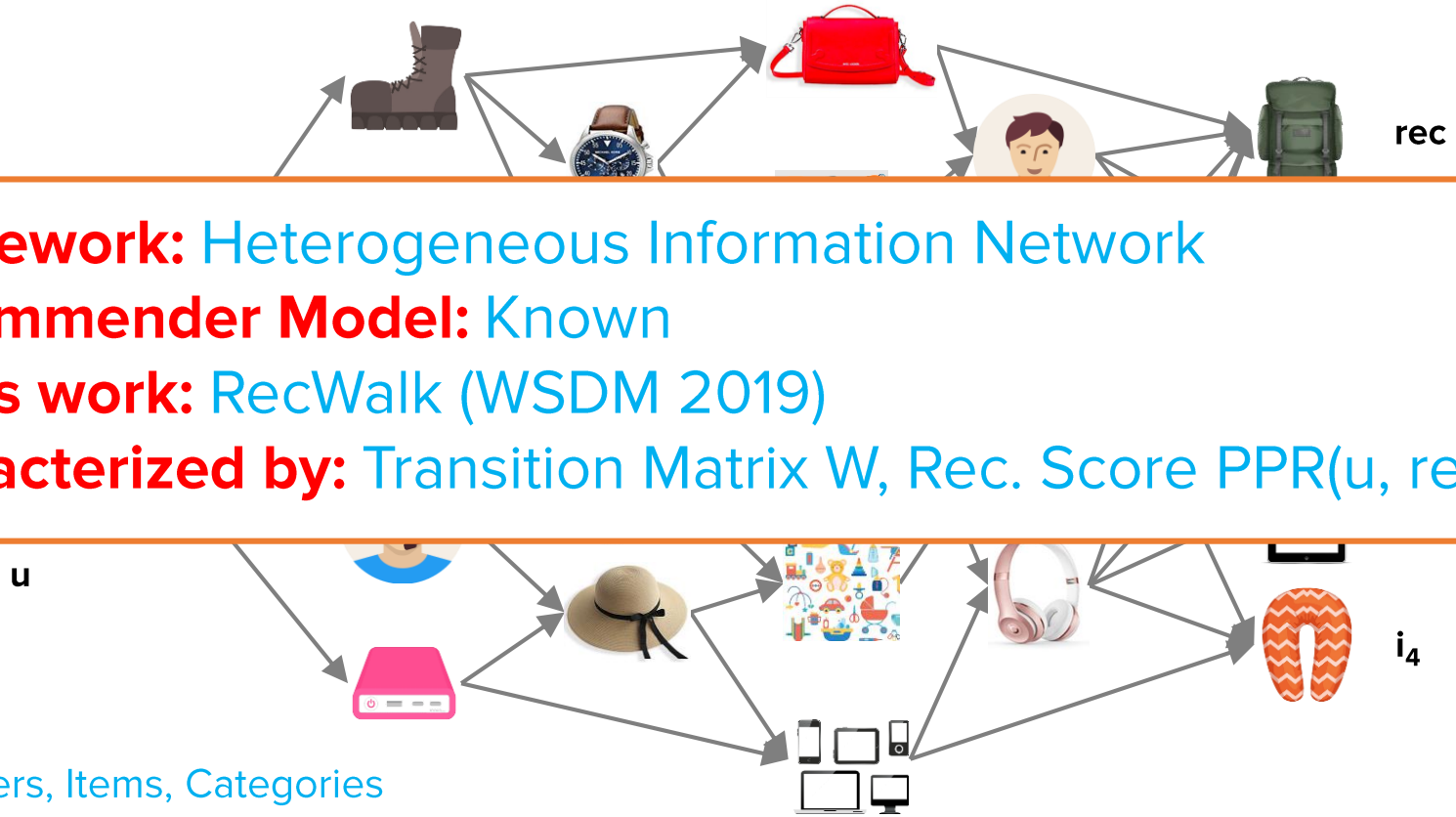
Problem Setup



Framework: Heterogeneous Information Network
Recommender Model: Known
In this work: RecWalk (WSDM 2019)

Nodes: Users, Items, Categories
Edges: Actions or Similarity, Directed, Weighted

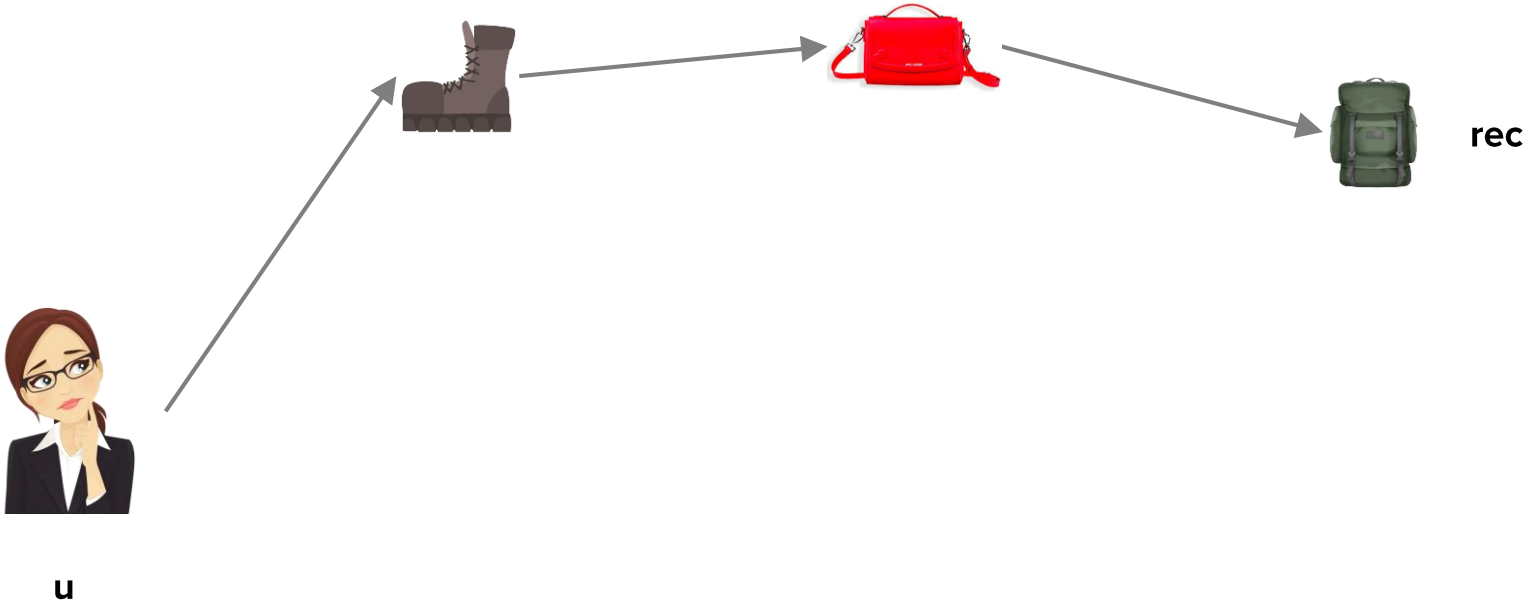
Problem Setup



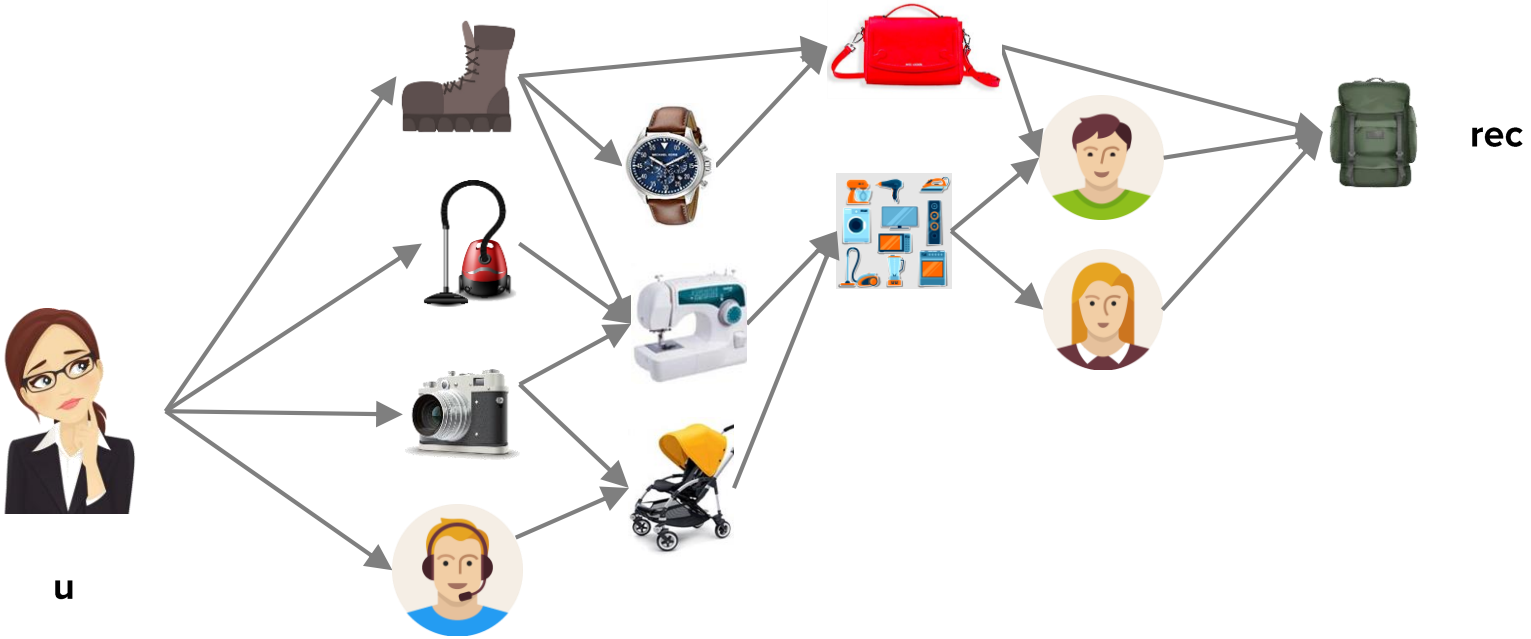
Framework: Heterogeneous Information Network
Recommender Model: Known
In this work: RecWalk (WSDM 2019)
Characterized by: Transition Matrix W, Rec. Score PPR(u, rec)

Nodes: Users, Items, Categories
Edges: Actions or Similarity, Directed, Weighted

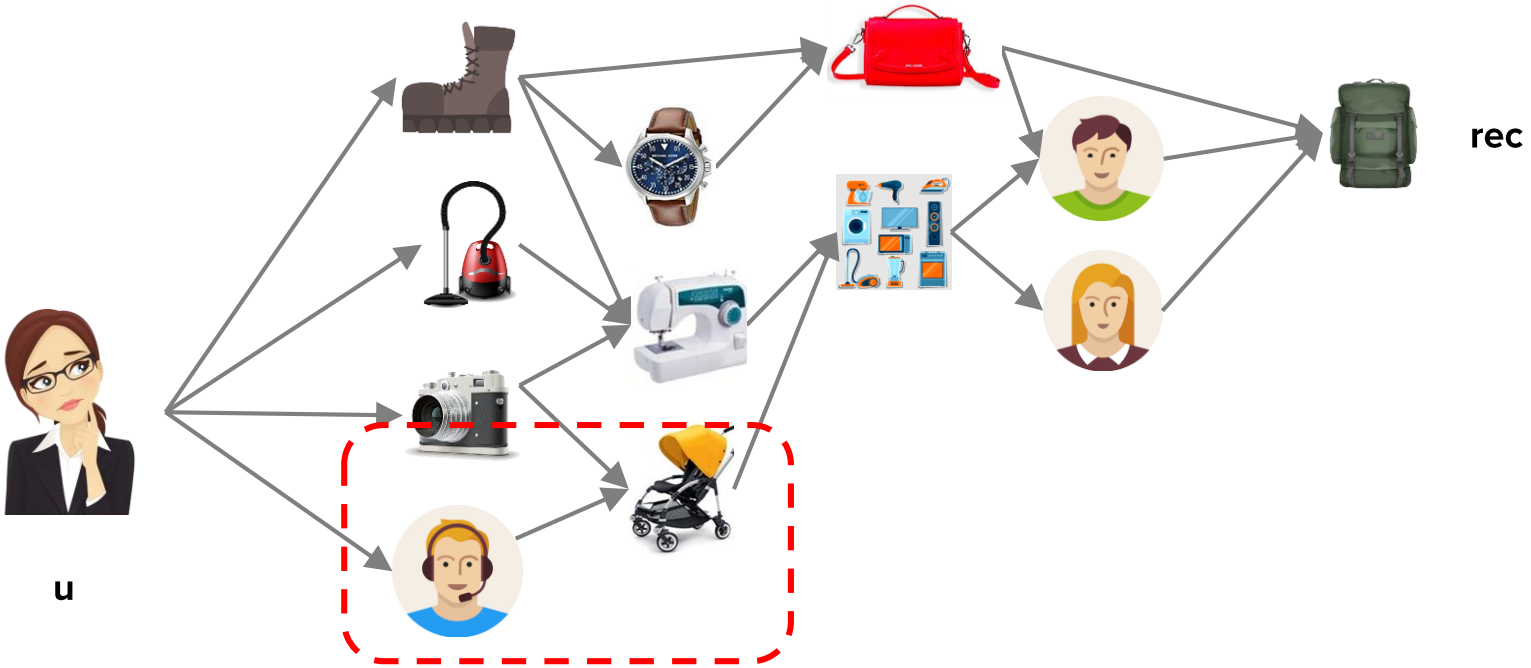
Existing approaches based on paths



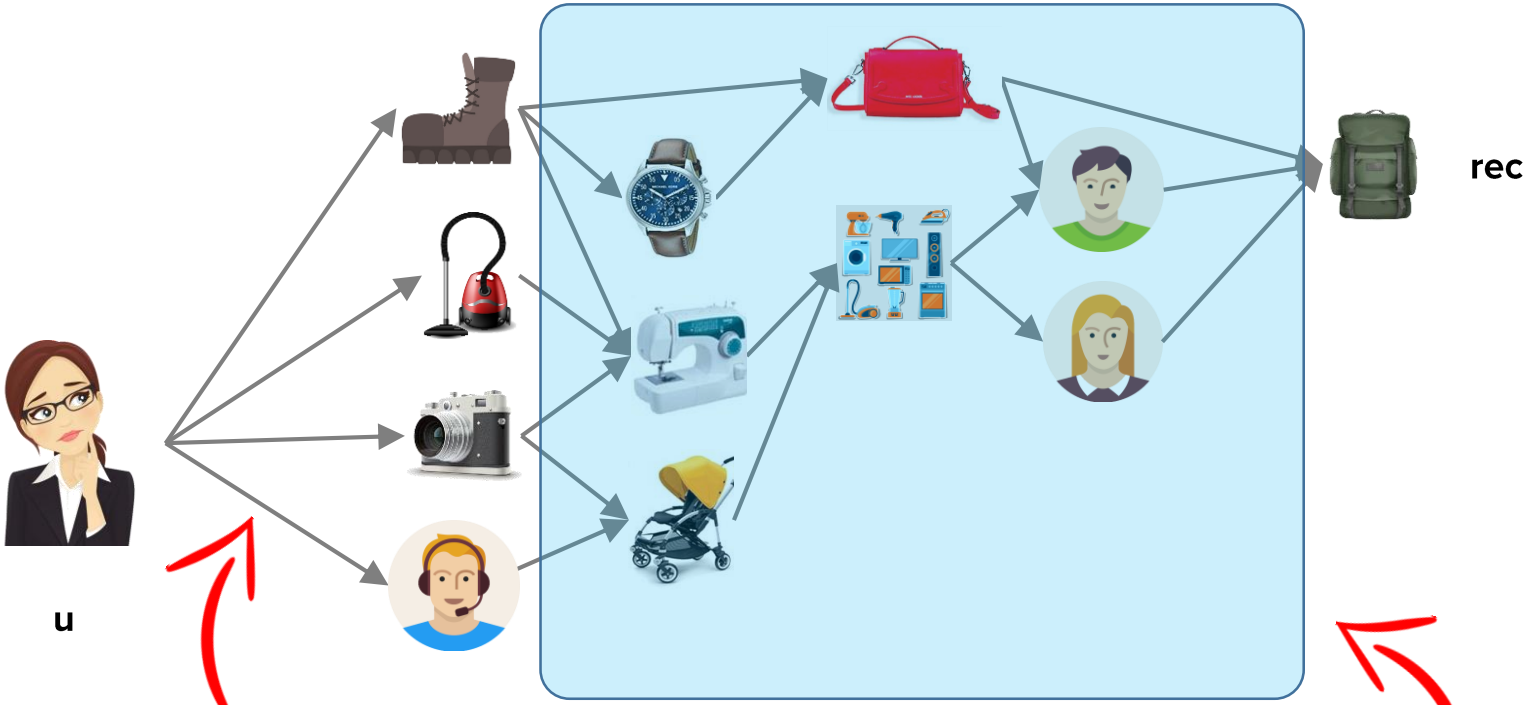
Concerns with paths: Too many



Concerns with paths: Potential privacy breaches



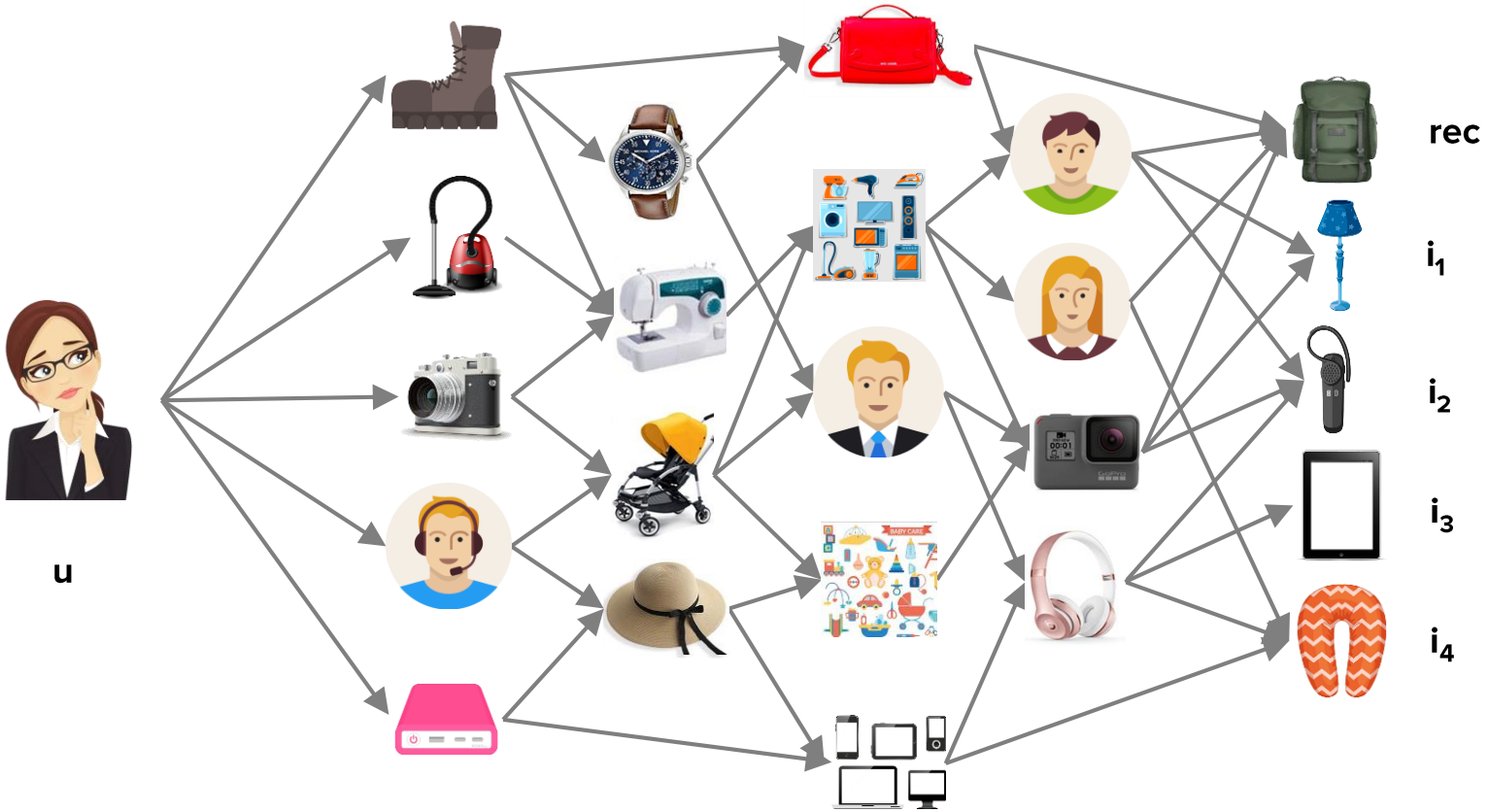
Concerns with paths: Not actionable



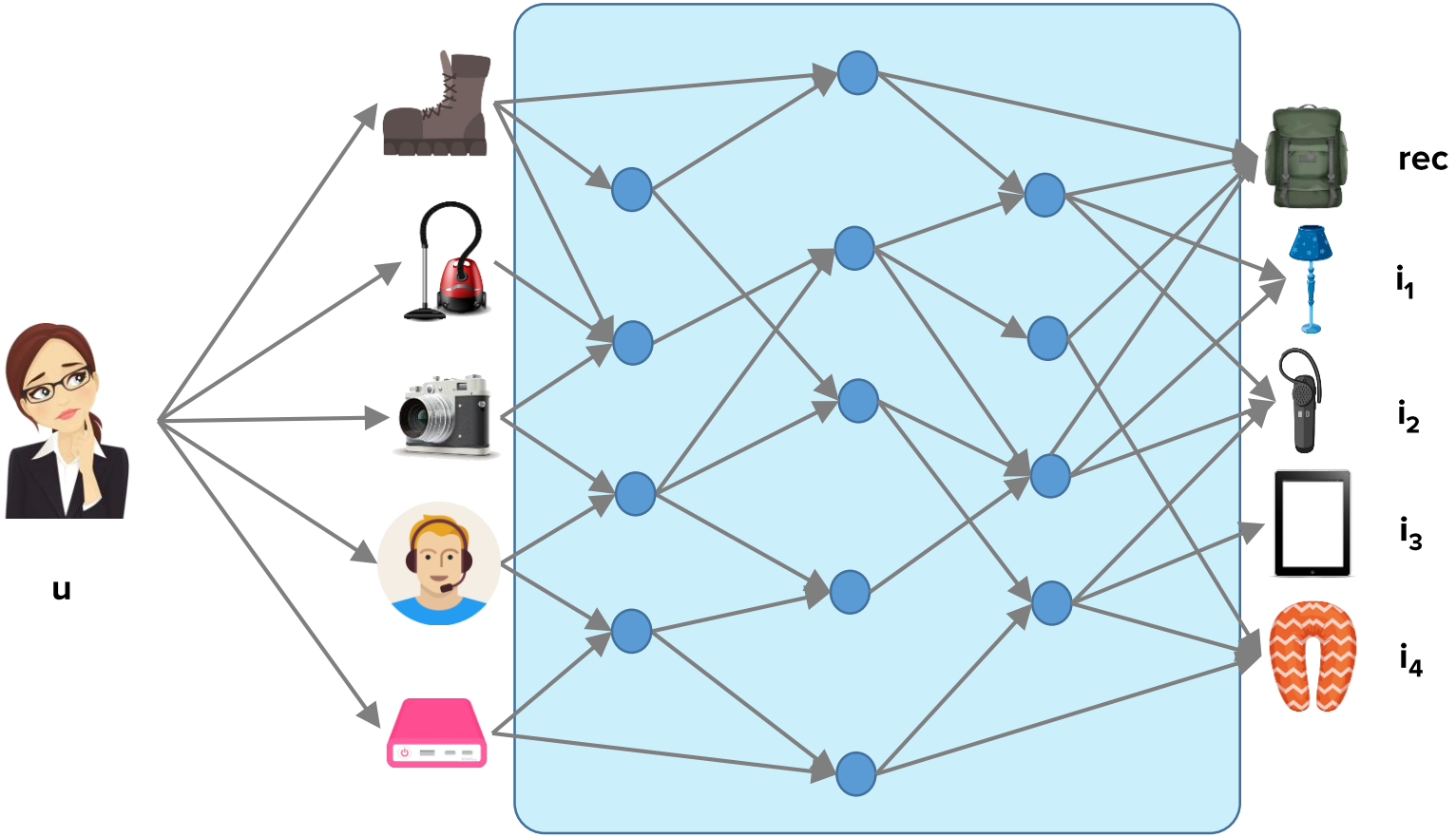
Not clear if **acting** upon these edges changes rec

Users have no **control** on this part of the graph

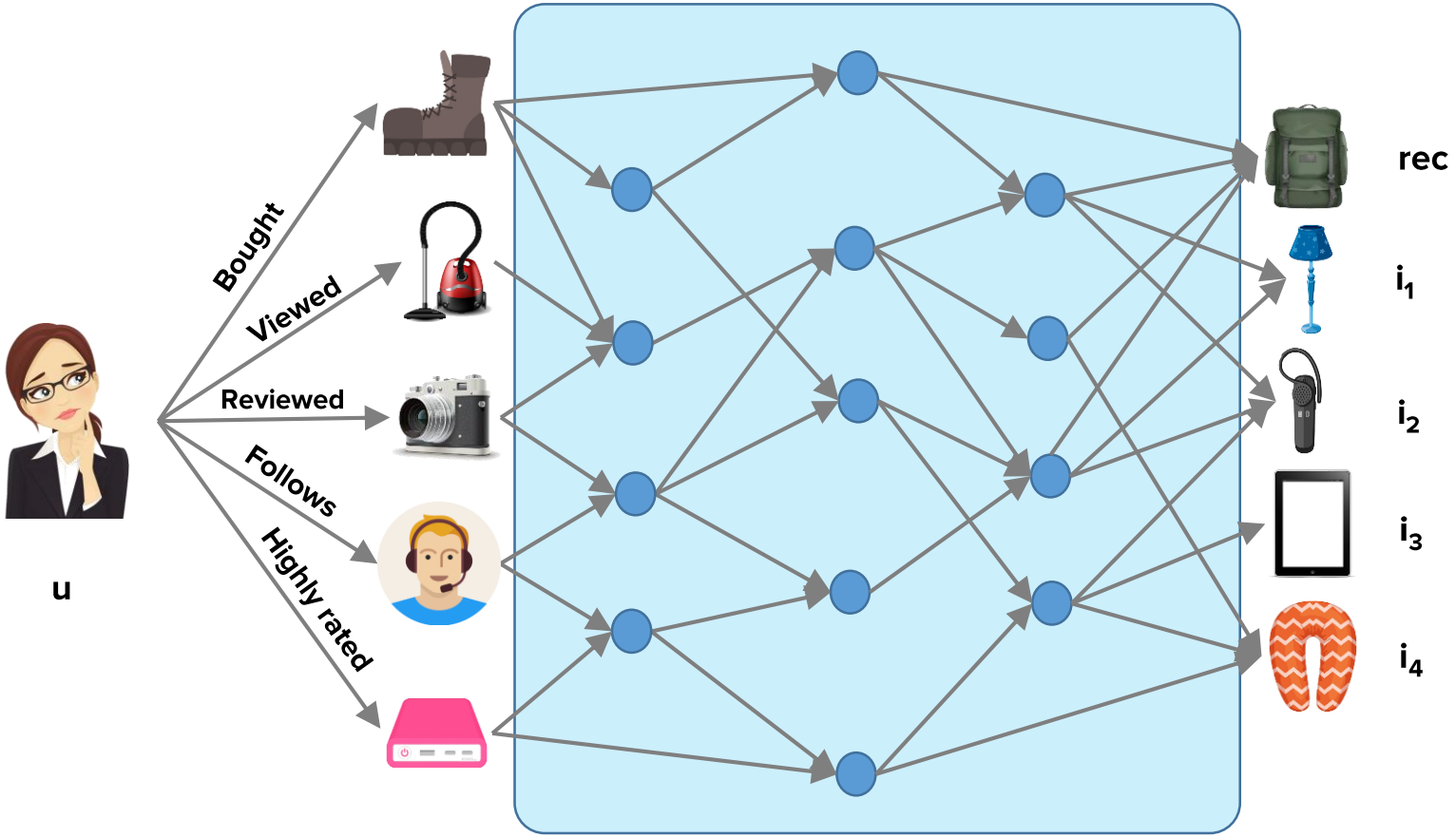
PRINCE: A fresh perspective on recommender explanations



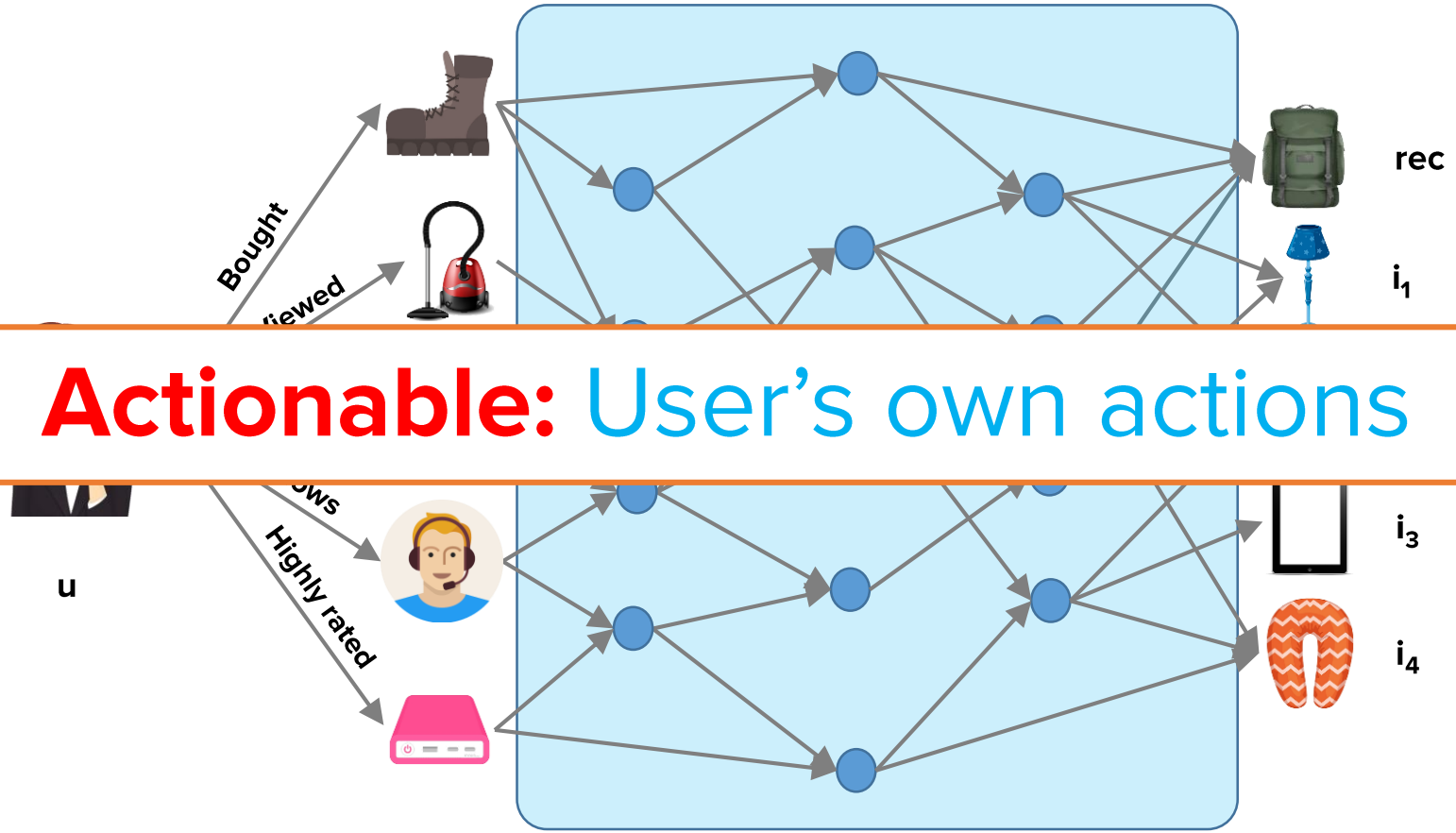
PRINCE: A fresh perspective on recommender explanations



PRINCE: A fresh perspective on recommender explanations

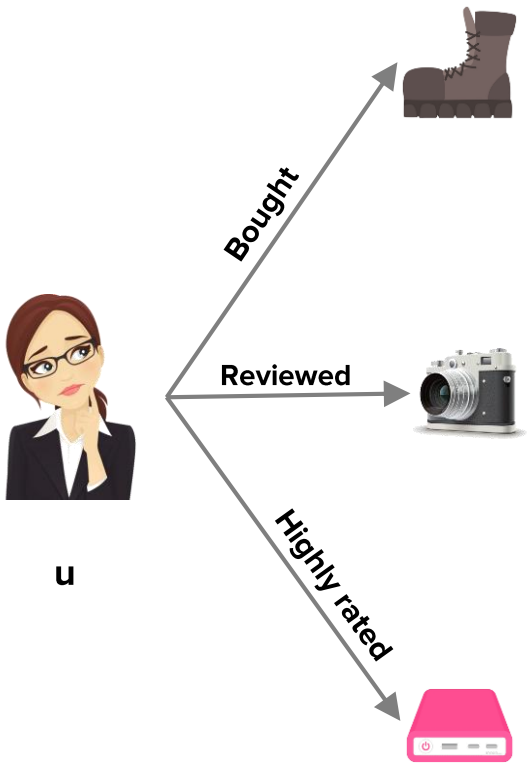


PRINCE: A fresh perspective on explanations

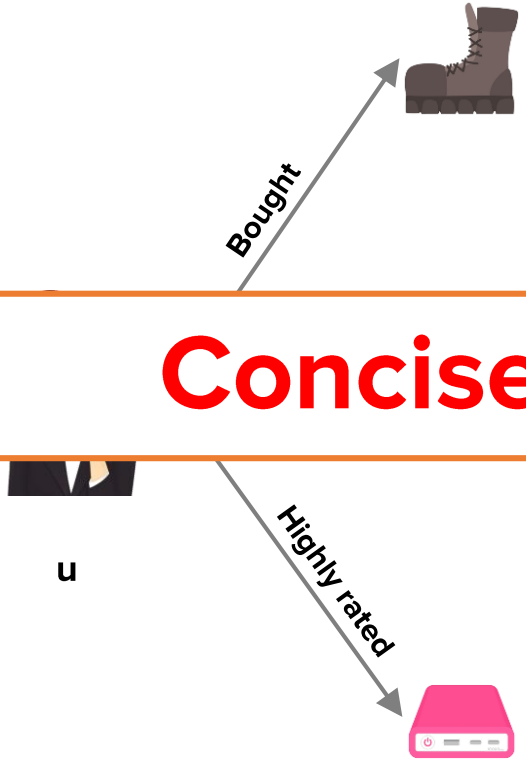


Actionable: User's own actions

PRINCE: A fresh perspective on recommender explanations

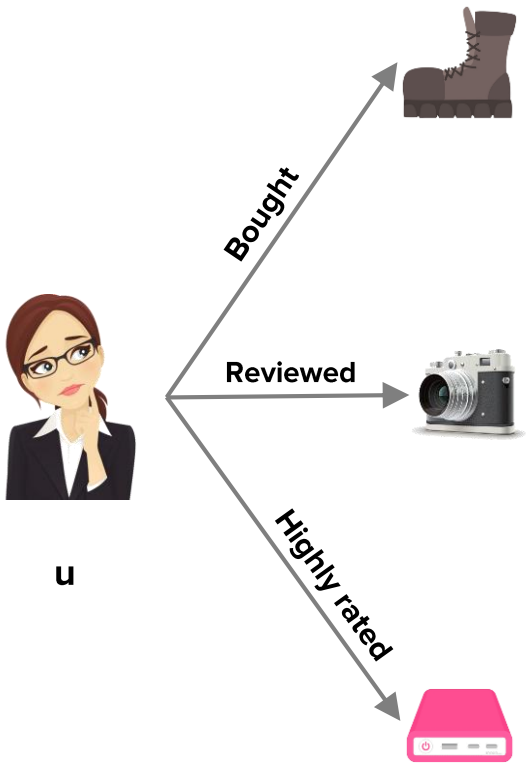


PRINCE: A fresh perspective on explanations

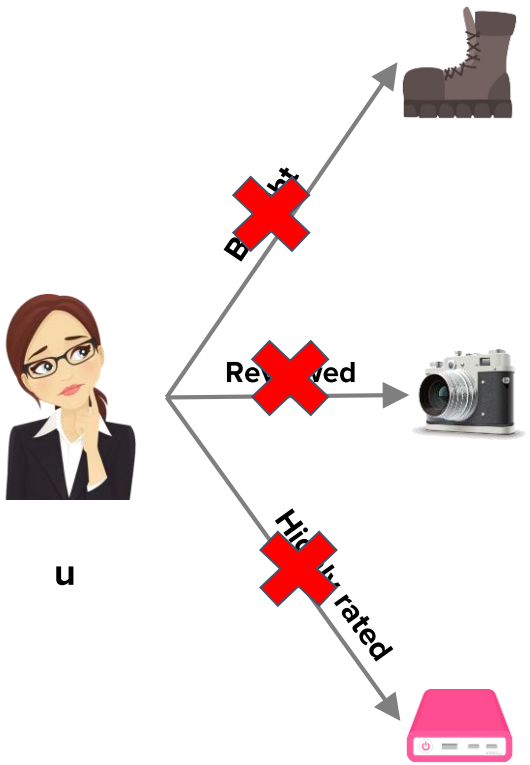


Concise: Use minimal sets

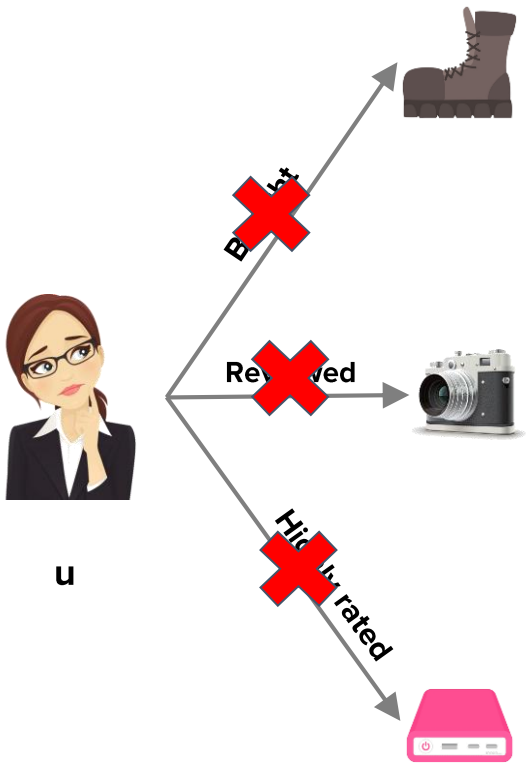
PRINCE: A fresh perspective on recommender explanations



PRINCE: A fresh perspective on recommender explanations



PRINCE: A fresh perspective on recommender explanations



PRINCE: A fresh perspective on recommender explanations

Scrutable: Use counterfactual setup

PRINCE: A fresh perspective on recommender explanations

- ★ **Actionable:** Ground explanations in user's own **actions**
- ★ **Concise:** Use **minimal** sets of actions for presentation
- ★ **Scrutable:** Use a **counterfactual** setup for derivation

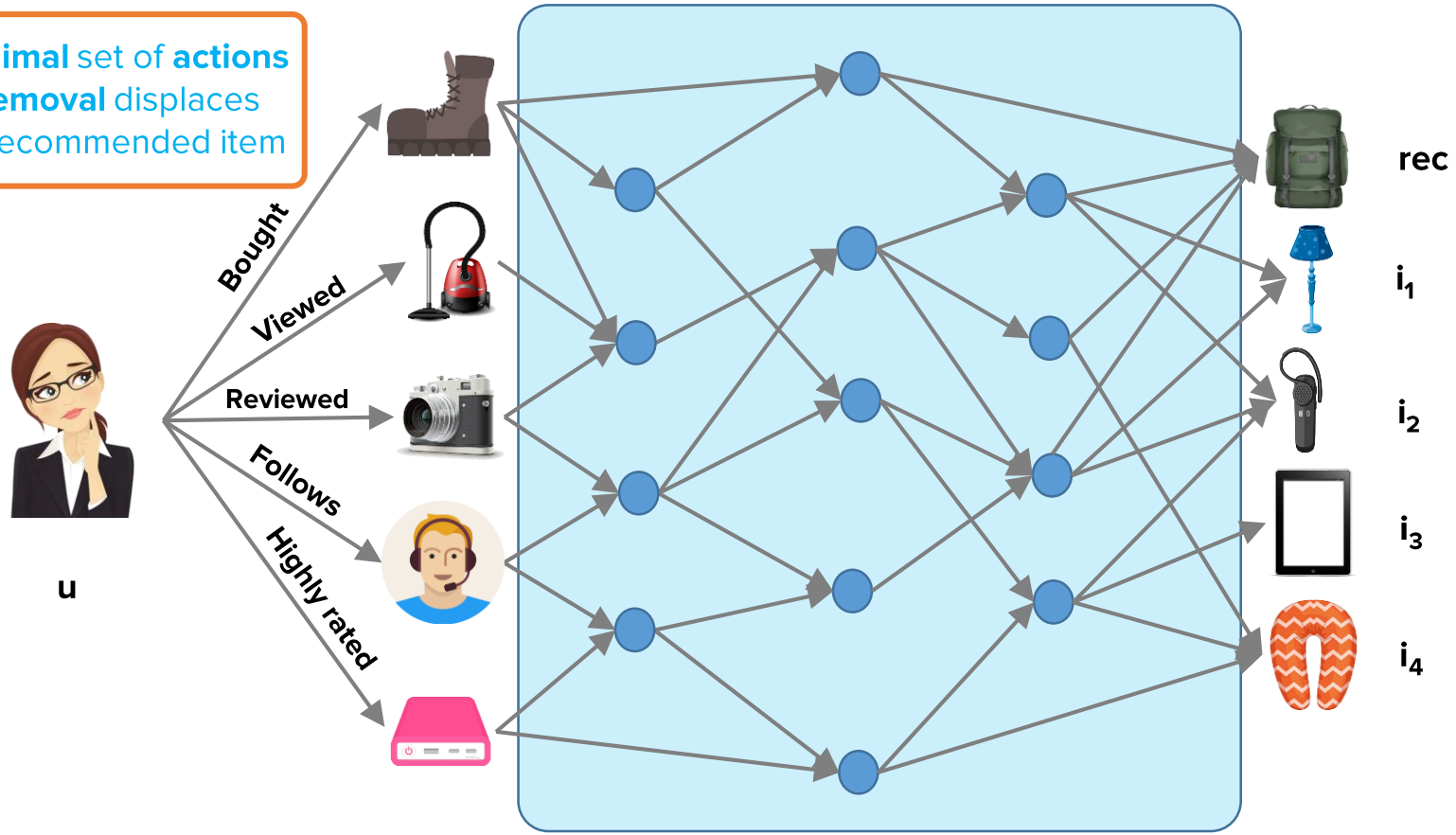
User u: Why did I receive this recommendation "Jack Wolfskin backpack"?

PRINCE: You **bought** "Adidas Hiking Shoes";
You **reviewed** "Nikon Coolpix Camera" with "Sleek! Handy on hikes!";
You **rated** "Intenso Travel Power Bank" highly.

If you **had not** done these actions:
"iPad Air" **would have replaced** "Jack Wolfskin backpack".

Formal problem statement for counterfactual explanation

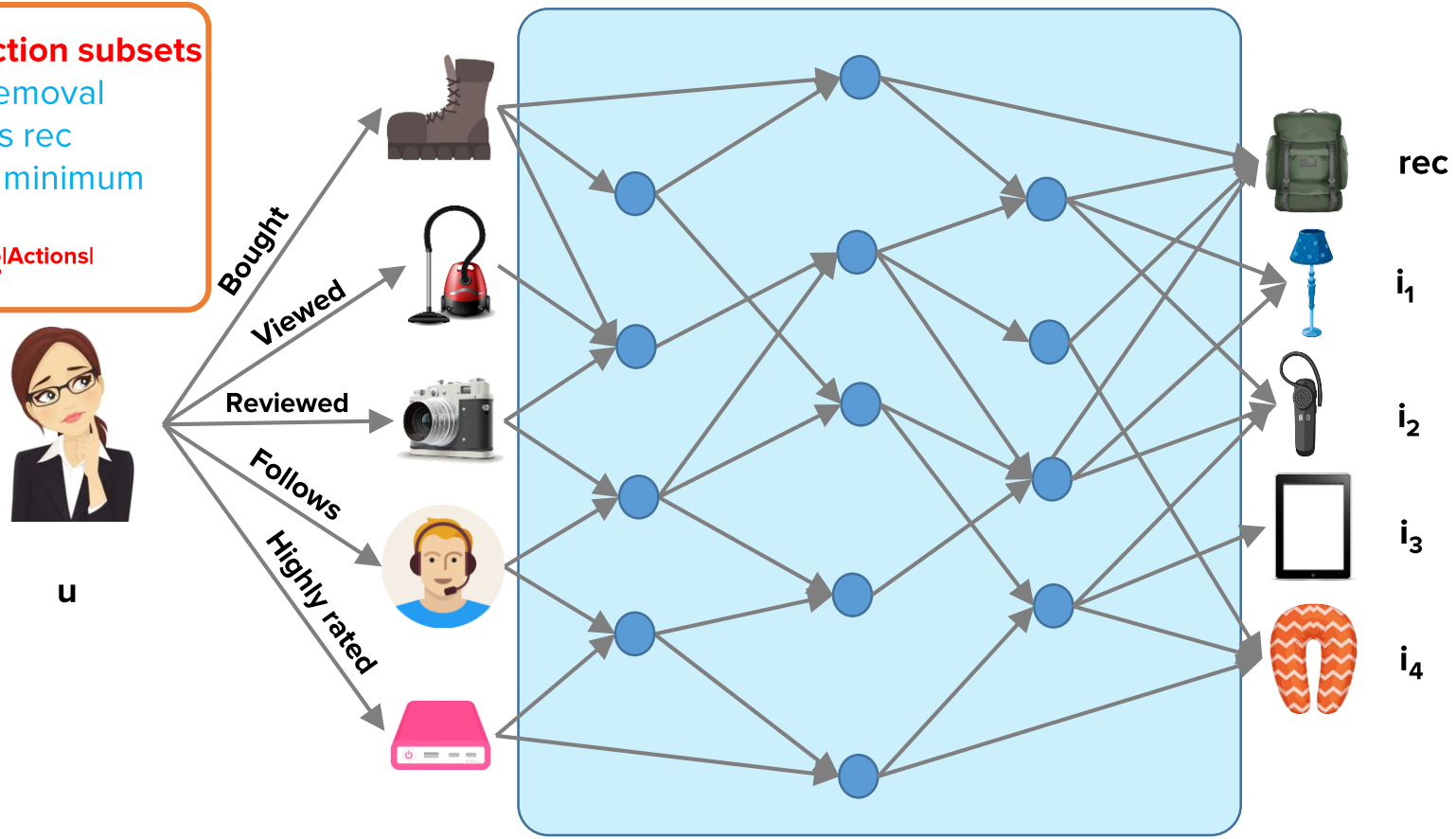
Find **minimal** set of **actions** whose **removal** displaces the top recommended item



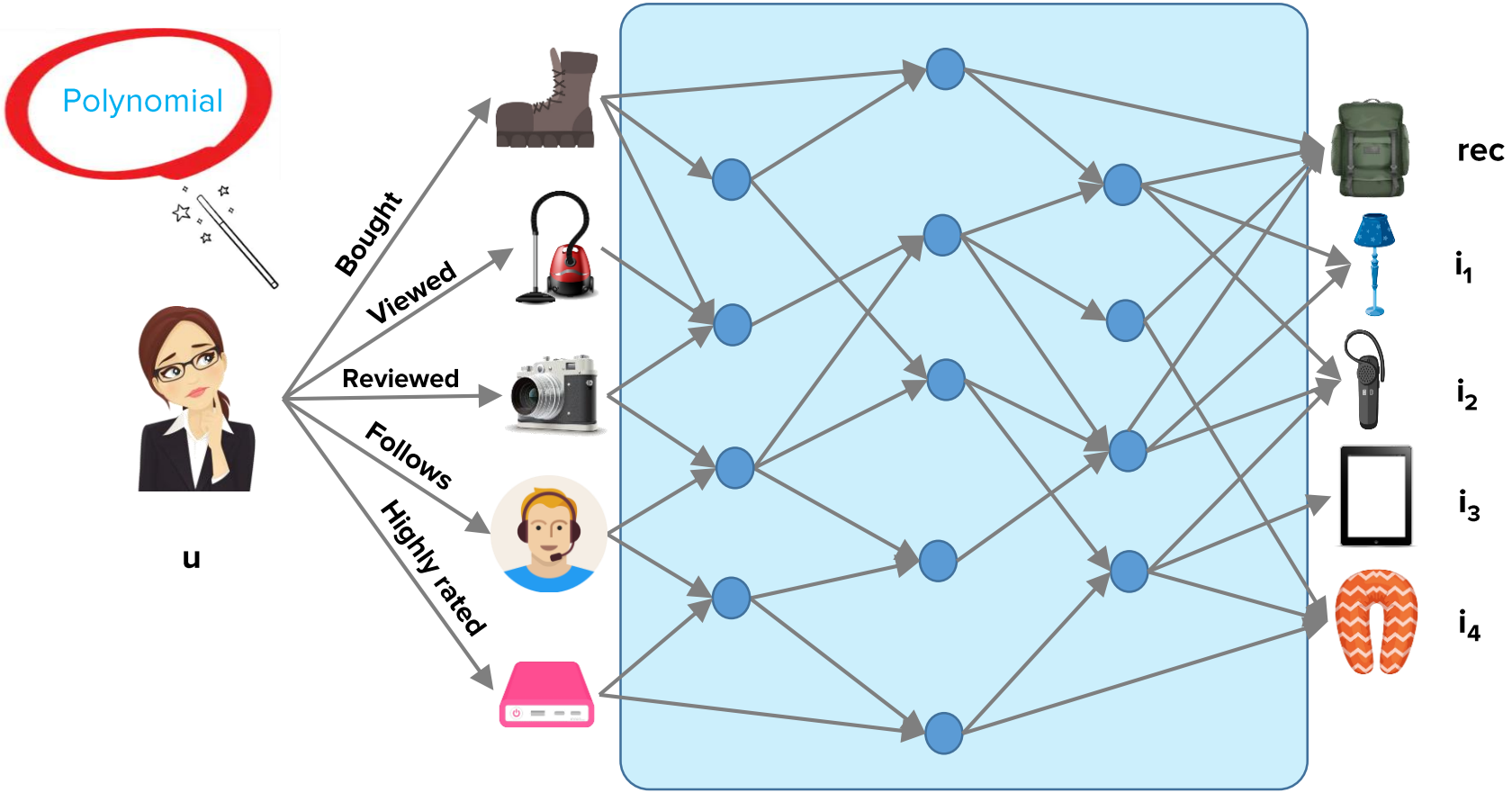
Naïve algorithm: Exponential time complexity!!

Try **all action subsets** whose removal displaces rec and find minimum

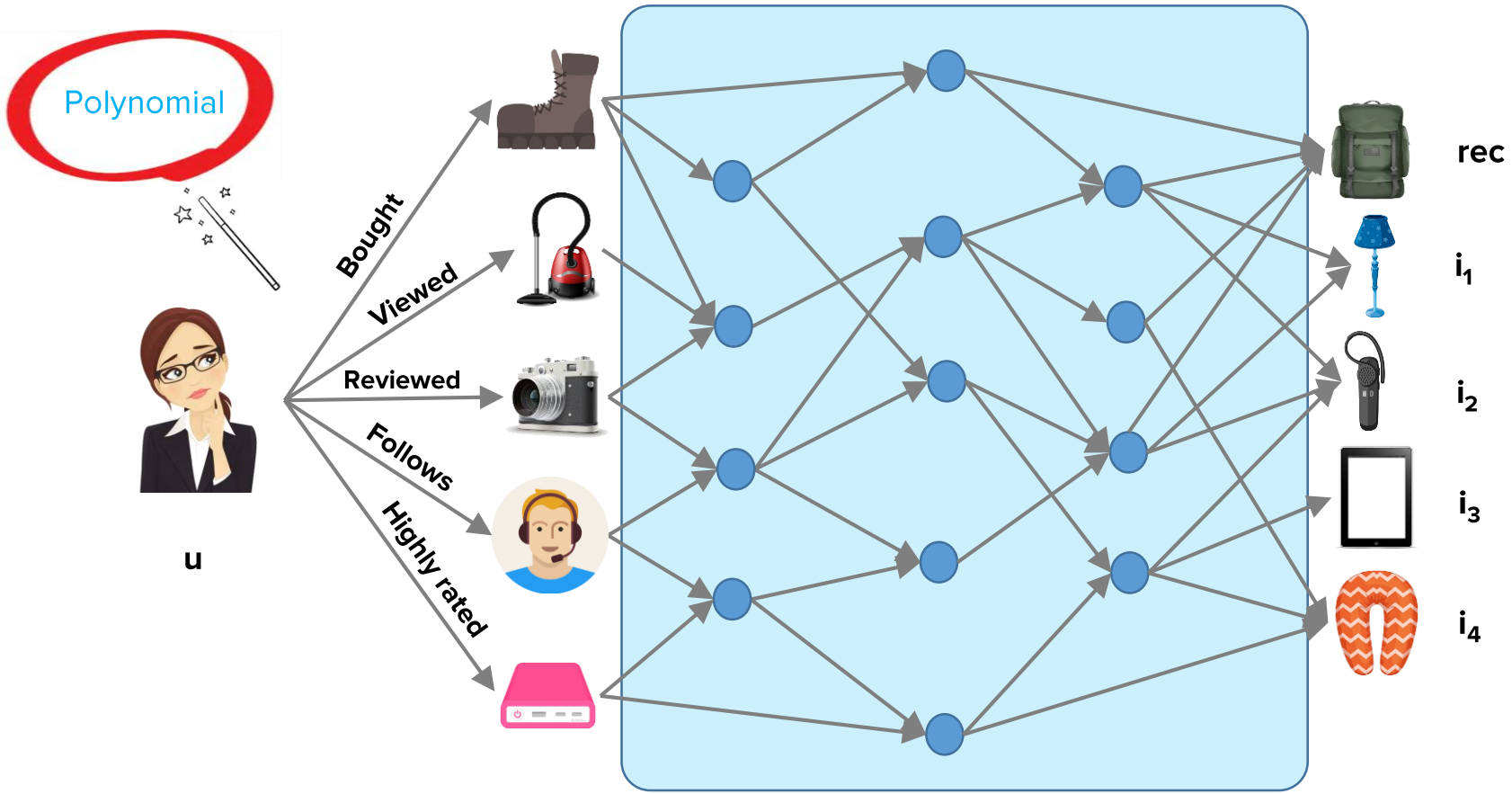
In time $2^{|Actions|}$



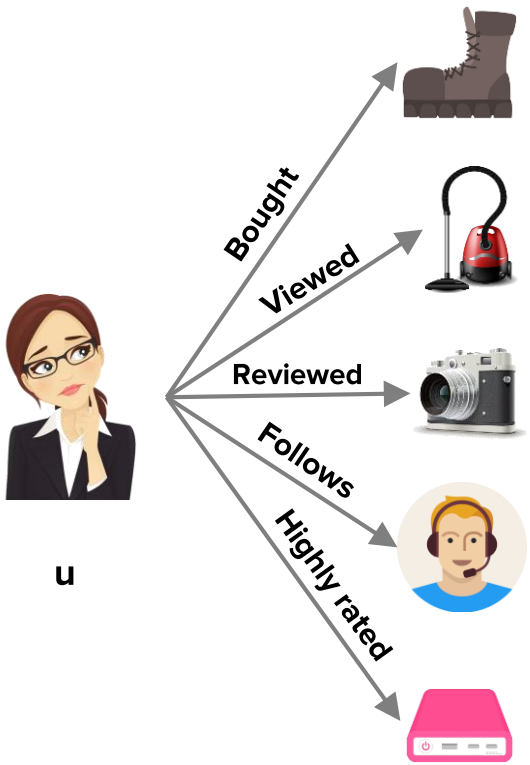
PRINCE has polynomial time complexity: How?



PRINCE iterates over items and not action subsets

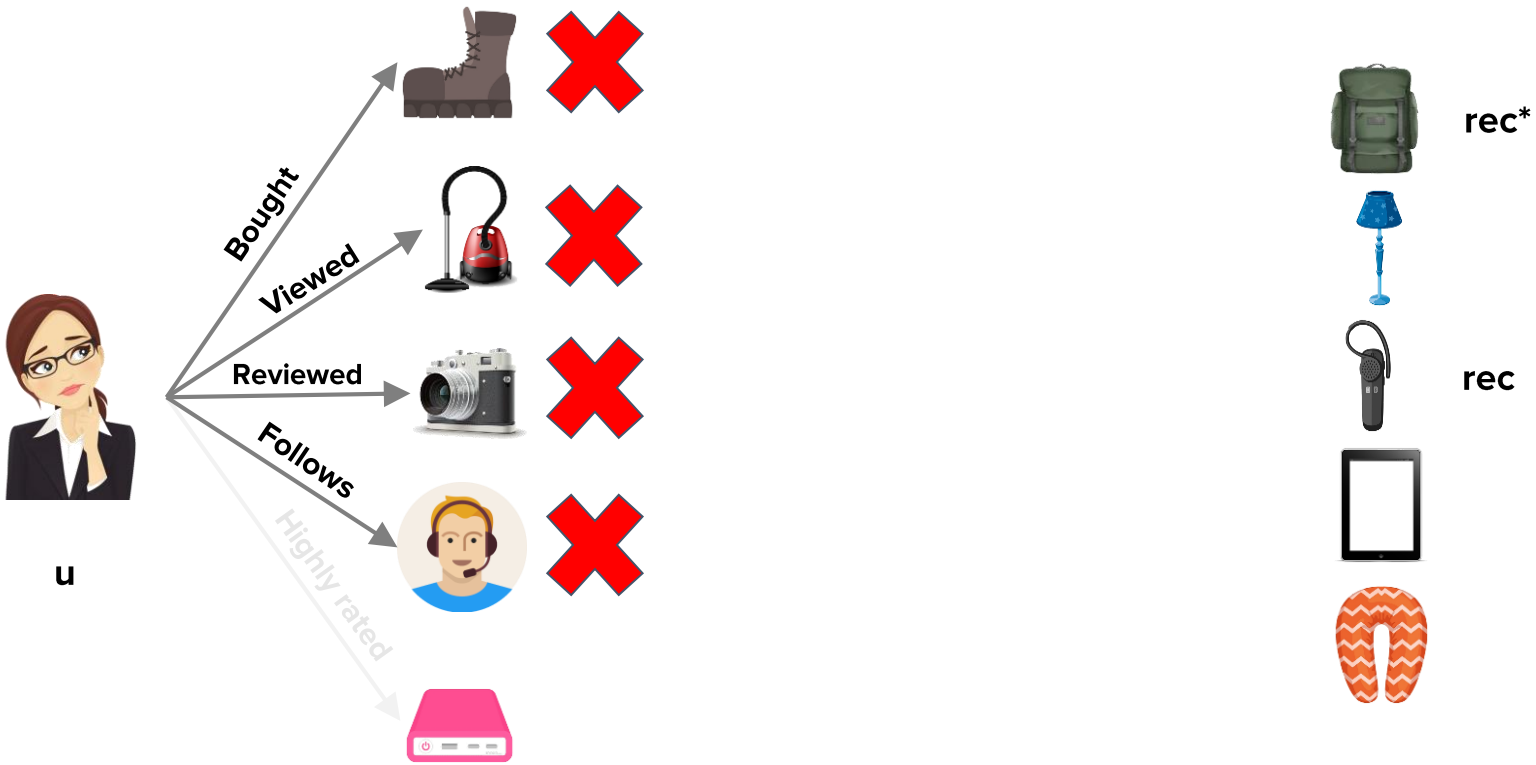


PRINCE iterates over items and not action subsets

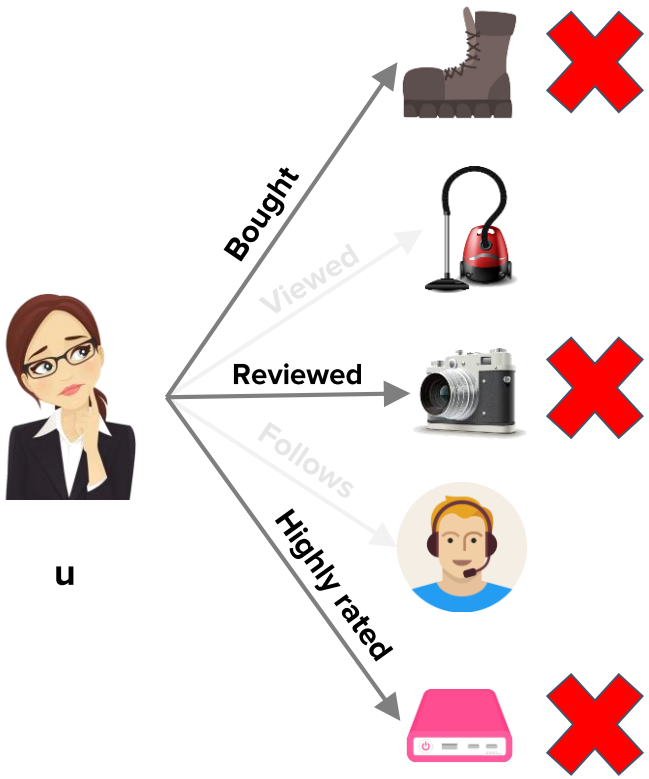


PRINCE iterates over items and not action subsets: Item i_1

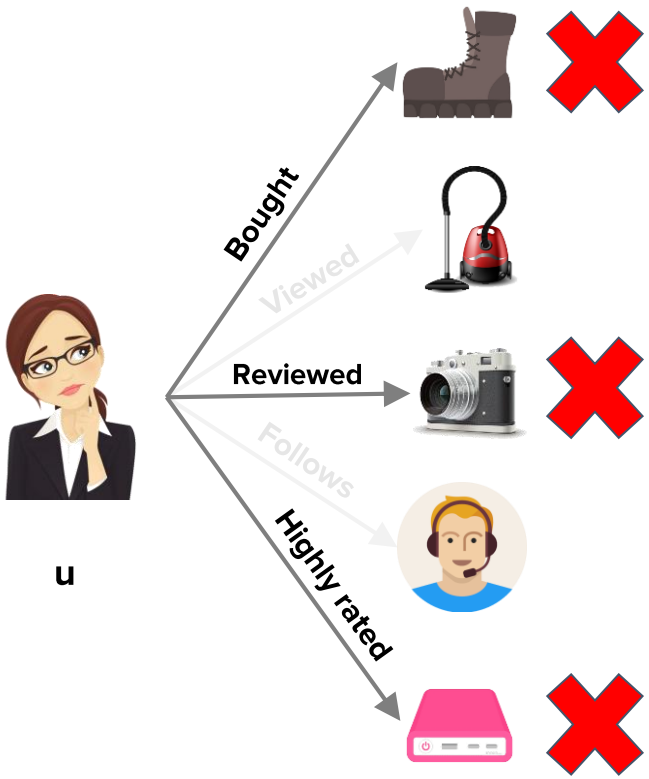
PRINCE iterates over items and not action subsets: Item i_2



PRINCE iterates over items and not action subsets: Item i_3 ...



PRINCE iterates over items and not action subsets: Item i_3 ...

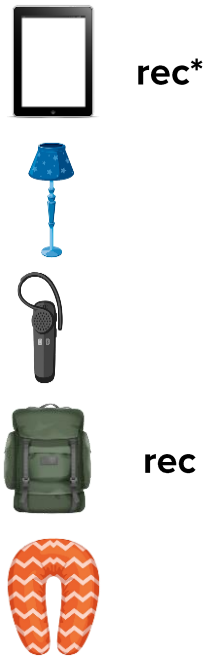


Iterate over items to find actions for **swapping** rec and rec*

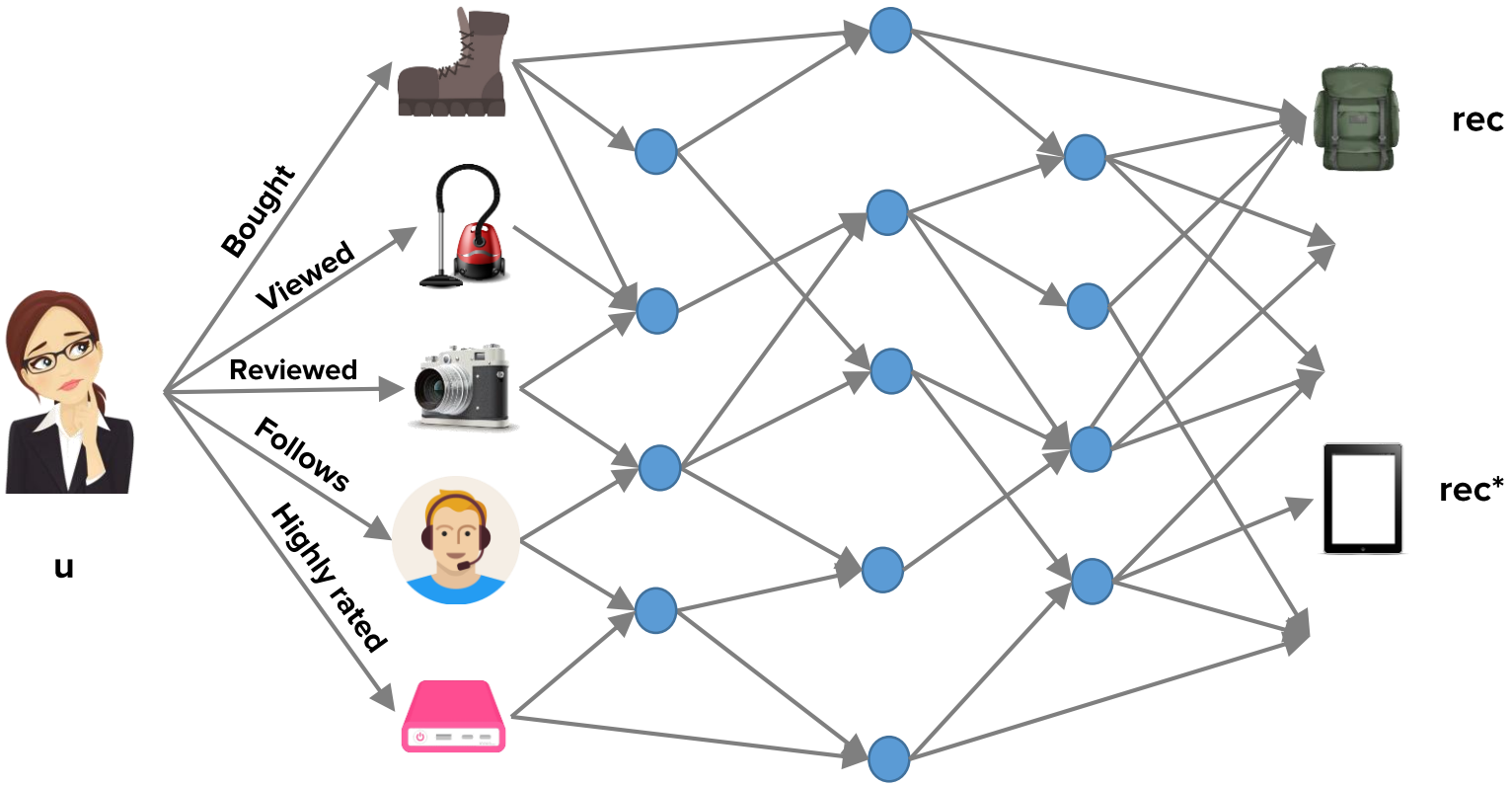
Sort actions by contribution within each iteration

Report minimal set

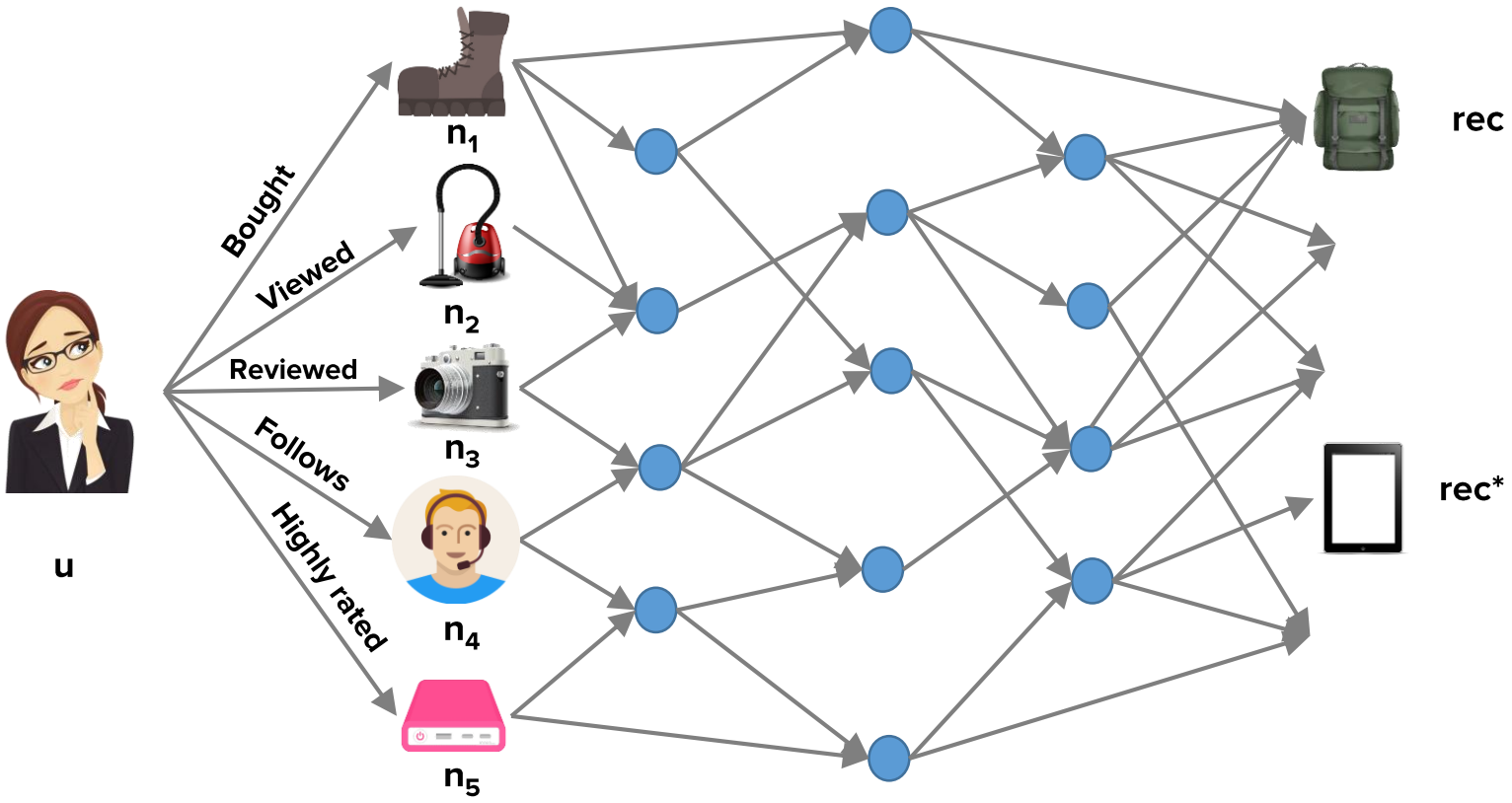
O (|Items| x |Actions| x log |Actions|)



Core of PRINCE: How do we swap orders of rec and rec*?



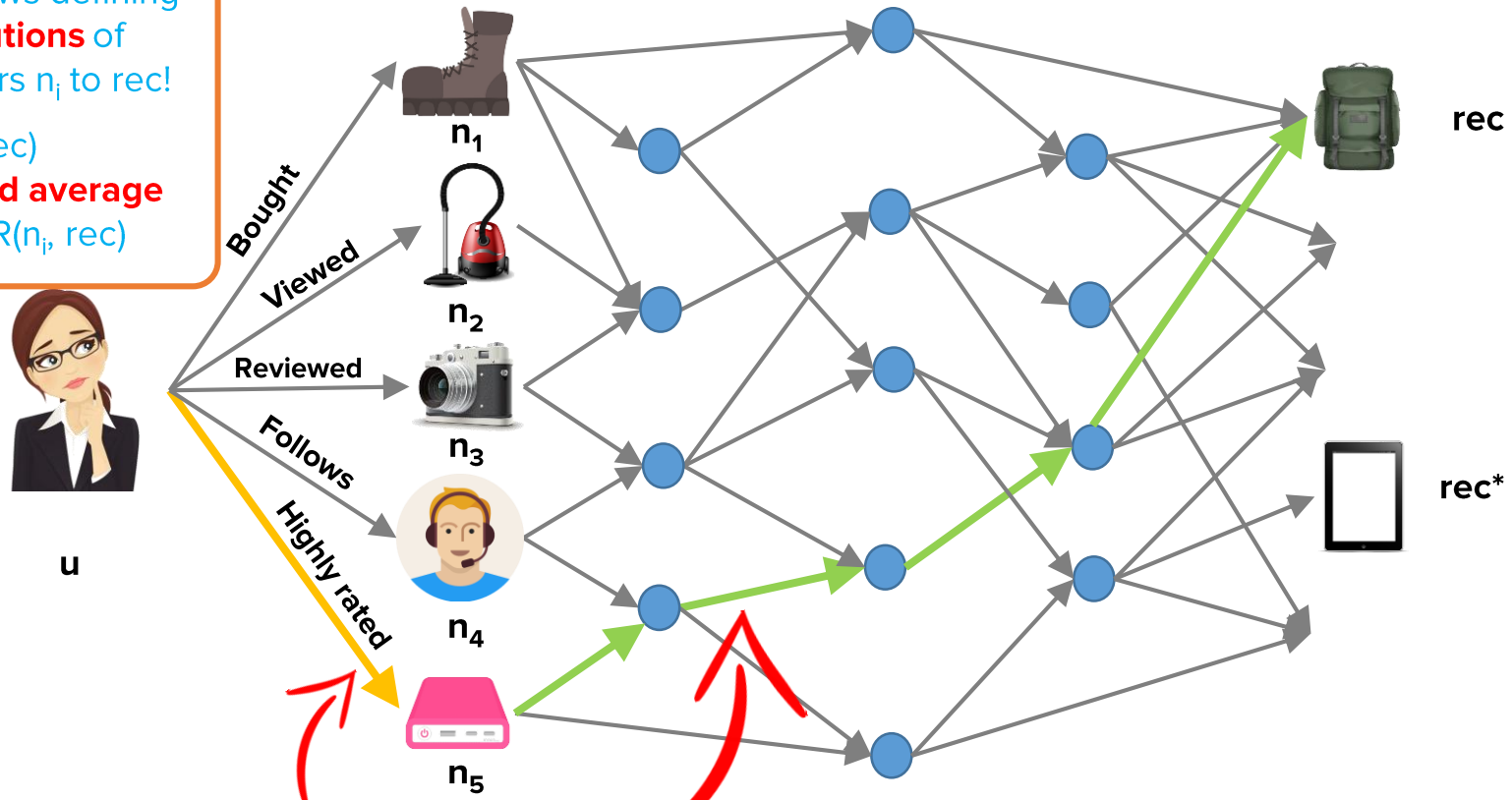
Swapping items: Actions connect to user u to neighbors n_i



Swapping items: Define contributions of neighbors n_i to rec

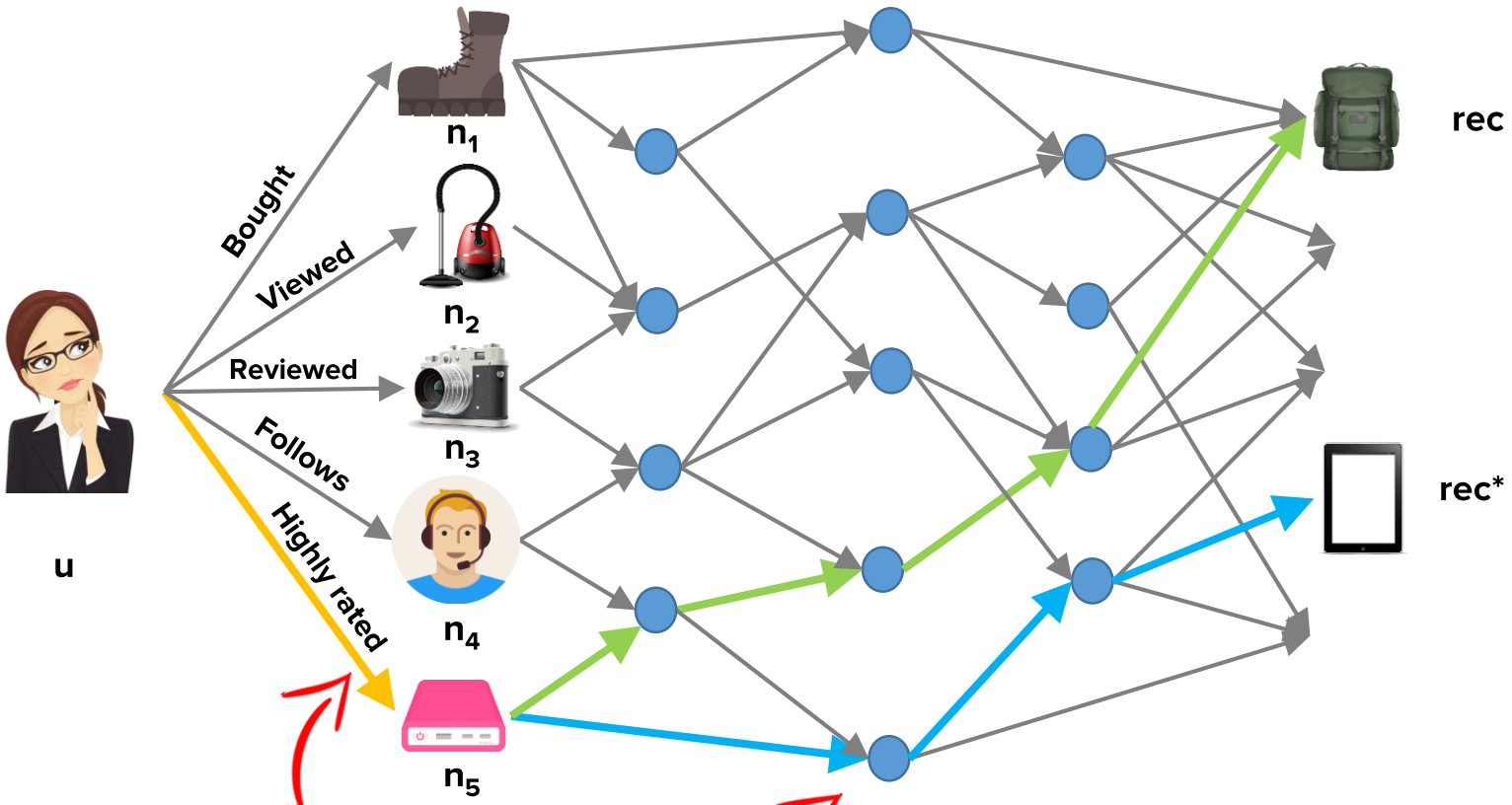
PPR allows defining **contributions** of neighbors n_i to rec!

PPR(u, rec) **weighted average over** PPR(n_i, rec)



Contrib. of n_i to rec: $W(u, n_i) \cdot PPR(n_i, rec)$

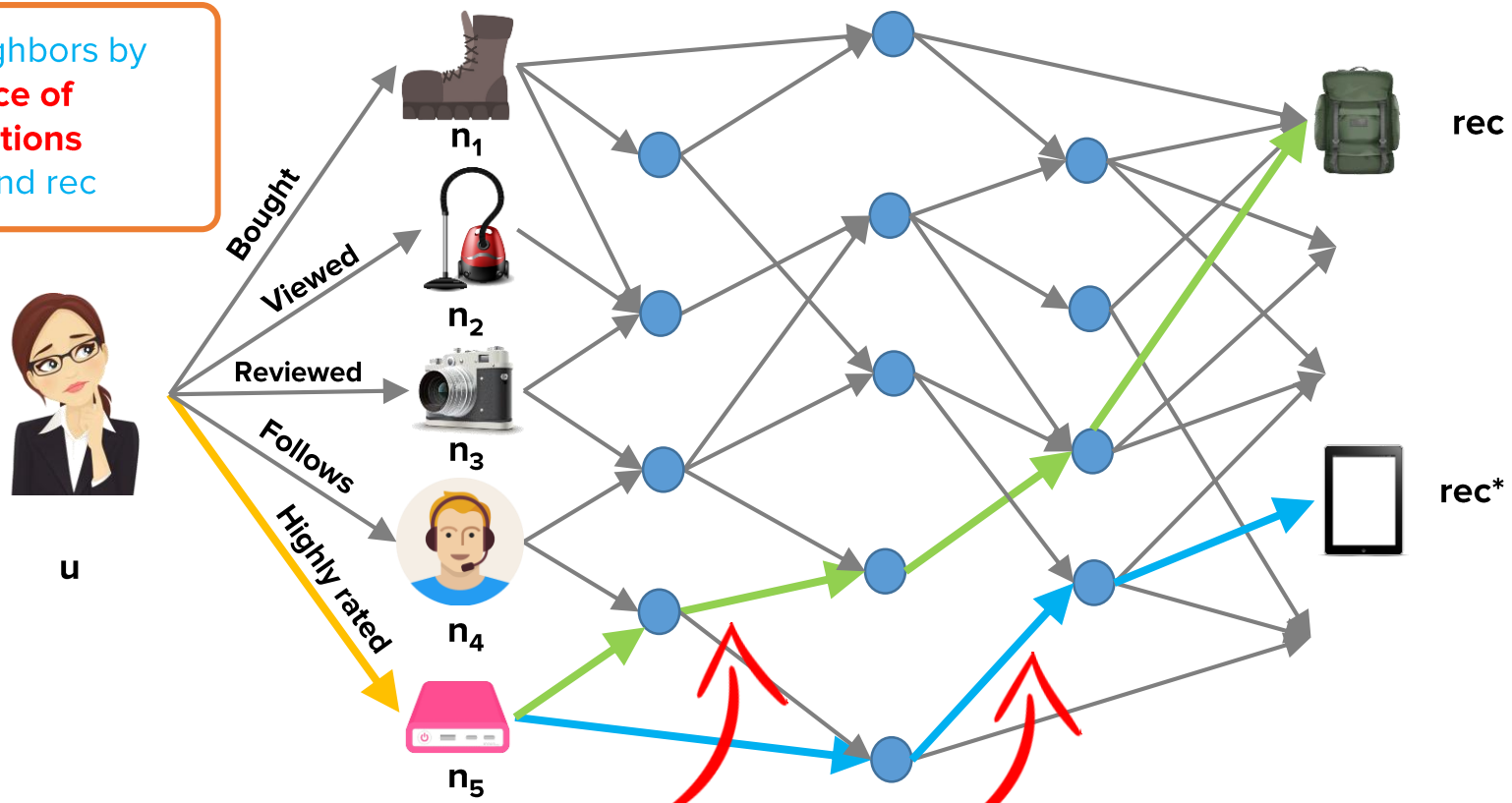
Swapping items: Define contributions of neighbors n_i to rec*



Contrib. of n_i to rec*: $W(u, n_i) \cdot PPR(n_i, rec^*)$

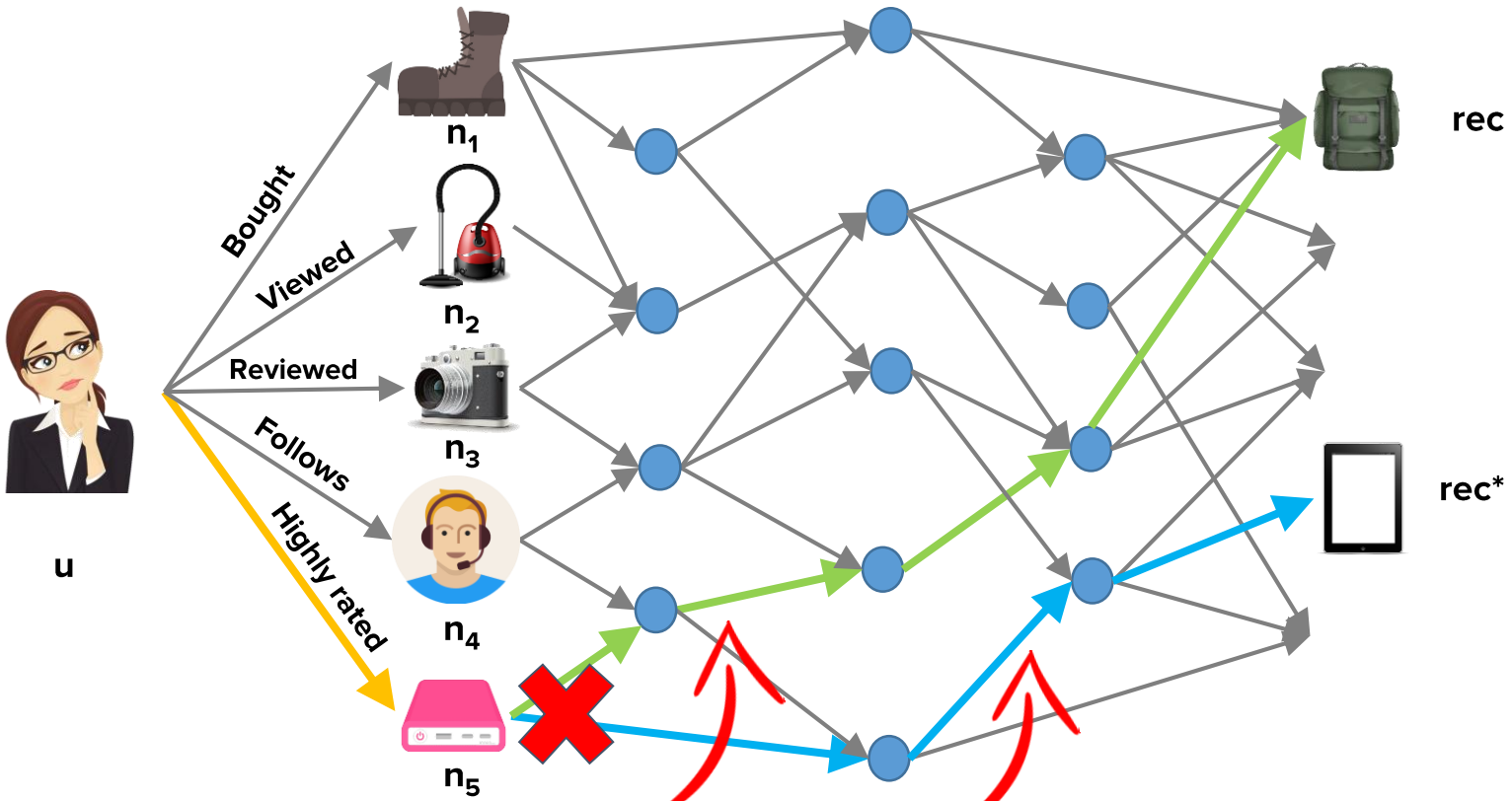
Sort neighbors by highest contribution to rec and least to rec*

Sort neighbors by **difference of contributions** to rec* and rec



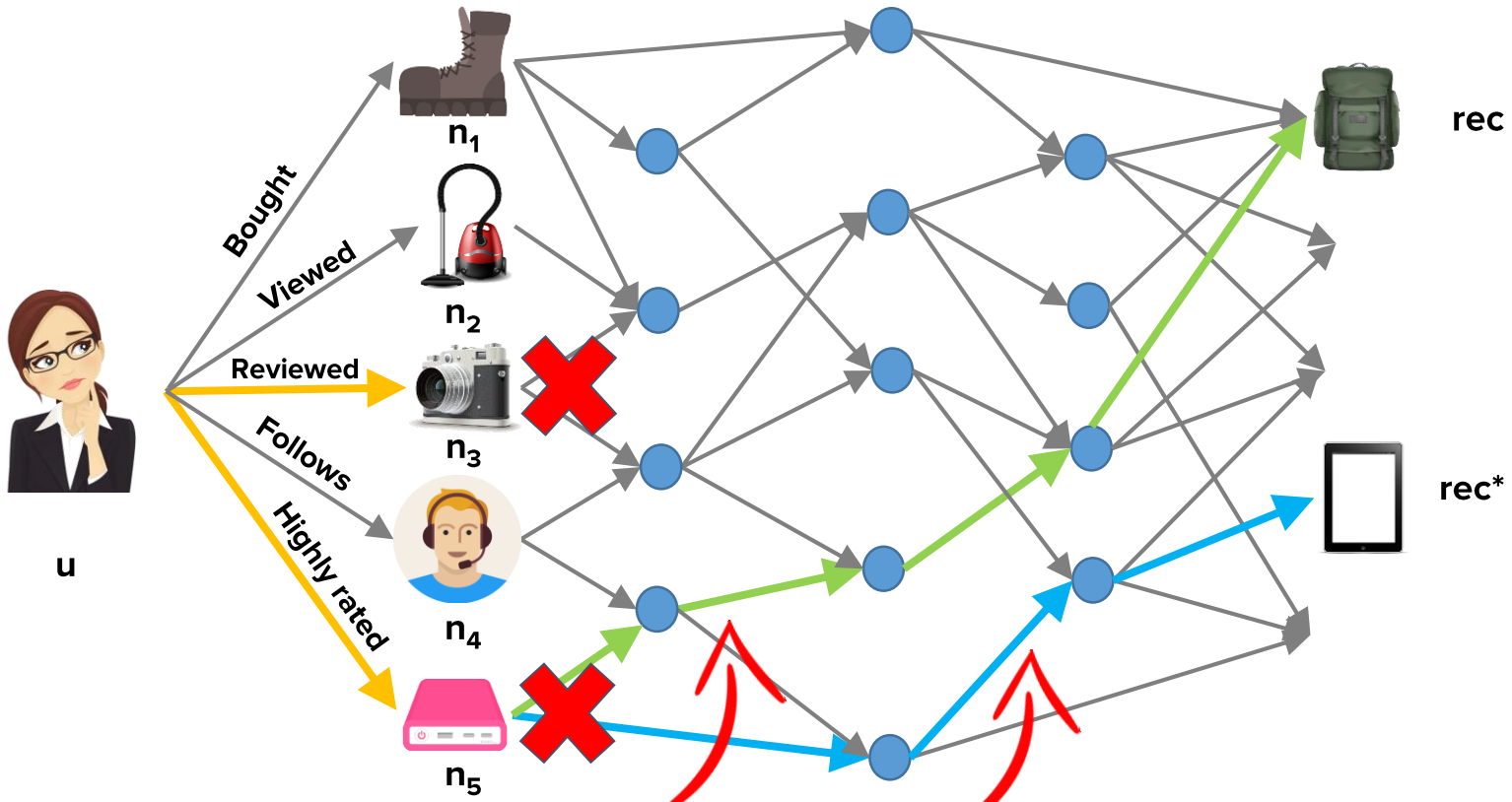
Difference in contributions: $W(u, n_i) \cdot PPR(n_i, rec) - PPR(n_i, rec^*)$

Delete neighbors greedily until replacement



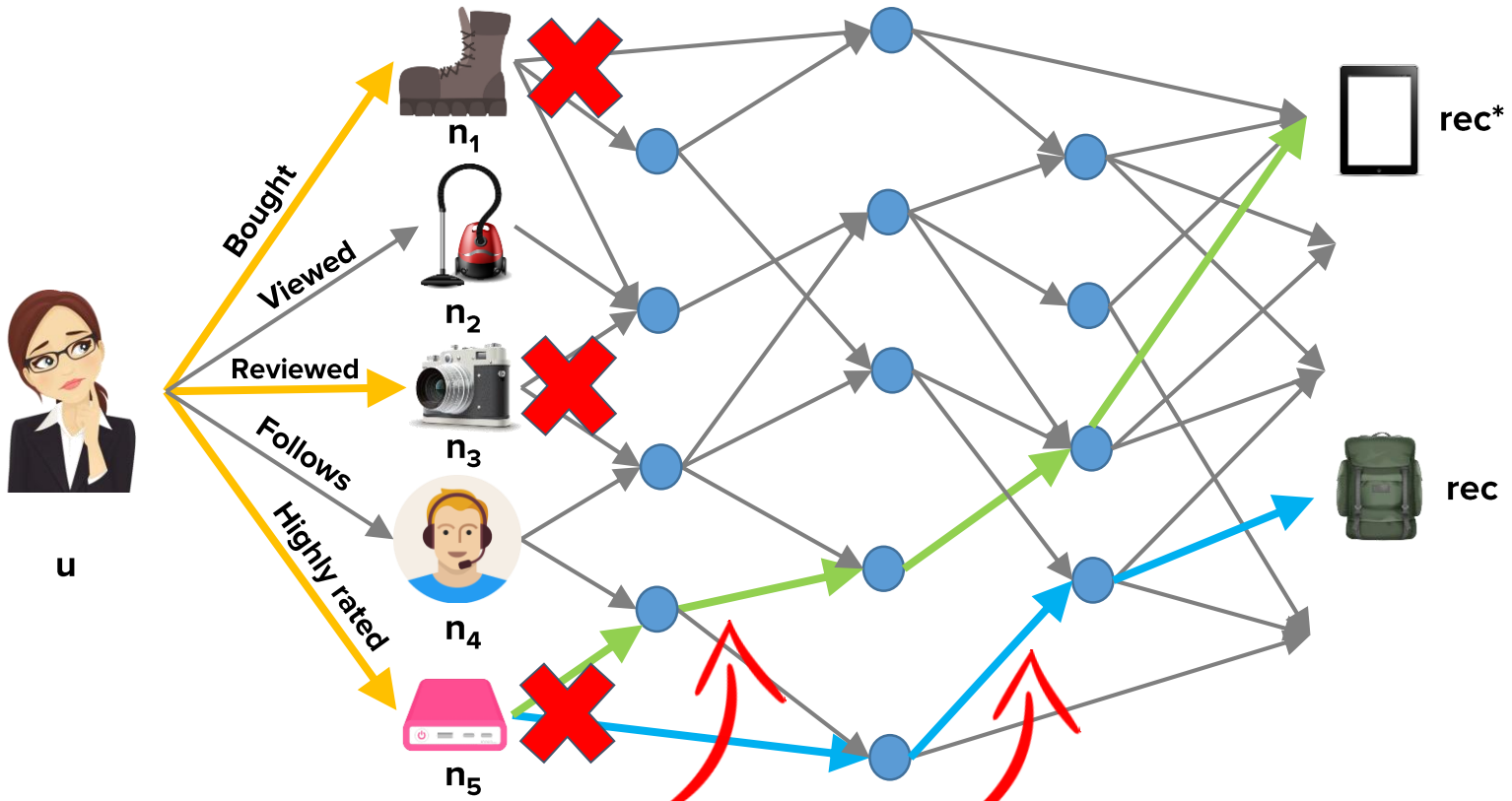
Difference in contributions: $W(u, n_i) \cdot PPR(n_i, rec) - PPR(n_i, rec^*)$

Delete neighbors greedily until replacement



Difference in contributions: $W(u, n_i) \cdot PPR(n_i, rec) - PPR(n_i, rec^*)$

Delete neighbors greedily until replacement



Difference in contributions: $W(u, n_i) \cdot PPR(n_i, rec) - PPR(n_i, rec^*)$

Why is this greedy strategy optimal? Theorems!

Proofs in paper!

★ PPR score for rec when **action subset A^* is removed from the graph**

can be computed as **product of two components,** and only one matters

$$PPR(u, rec \mid A^*) = \frac{1-\alpha}{\alpha} \cdot PPR(u, u \mid A^*) \cdot \boxed{\sum_{(u,n_i) \in A \setminus A^*} W(u, n_i \mid A^*) \cdot PPR(n_i, rec \mid A)}$$

★ The **condition for replacement** can be reformulated as

$$PPR(u, rec \mid A^*) < PPR(u, rec^* \mid A^*) \Leftrightarrow \sum_{(u,n_i) \in A \setminus A^*} W(u, n_i) \cdot [PPR(n_i, rec \mid A) - PPR(n_i, rec^* \mid A)] < 0$$

Putting it all together: The PRINCE Algorithm

Input: $G = (V, E, \theta), u, rec, I$

Output: A^*, rec^*

Counterfactual explanation

Initialize: $A^* \leftarrow A, rec^* \leftarrow rec$

for each $i \in I$

Iterate over each item i

$A^i = SwapOrder(G, u, rec, i)$

Core: Find actions to swap rec and i

if $|A^i| < |A^*|$

Minimality check

$A^* \leftarrow A^i, rec^* \leftarrow i$

return A^*, rec^*

Report minimal set

Experimental setup

- ★ **Datasets** Amazon, Goodreads
- ★ **Graph size** 2k users, 50k items, 58k actions (ratings + reviews),

  40 categories **(for Amazon)**
- ★ **Replacement item** From top-k recommendations
- ★ **Evaluations** Graph measurements, User study

Graph results

- ★ **Baselines:** HC (Highest Contributions), SP (Shortest Paths)
- ★ **Metric:** Explanation size
- ★ PRINCE is **difficult to approximate**
- ★ Explanations **shrink** with increasing *k*
- ★ PRINCE is **efficient**

Runtimes at $k = 5$:

1.3 milliseconds (Precomputed)

121.6 seconds (Online)

	Amazon			Goodreads		
k	**PRINCE**	**HC**	**SP**	**PRINCE**	**HC**	**SP**
3	**5.09***	6.87	7.57	**2.05***	2.86	5.38
5	**3.41***	4.62	5.01	**1.66***	2.19	4.37
10	**2.66***	3.66	4.15	**1.43**	1.45	3.28
15	**2.13***	3.00	3.68	**1.11**	1.12	2.90
20	**1.80***	2.39	3.28	**1.11**	1.12	2.90

Mechanical Turk study on usefulness

- ★ Explanations for 200 (user, rec) pairs from Amazon judged for **usefulness**
- ★ On a scale of **1 – 3** (most useful)
- ★ **Baseline:** CredPaths (Yang et al. ICDM 2018)
- ★ **Three Master** Workers per HIT
- ★ **Honeypots** in HITs
- ★ Controlled for **presentation, size and comparison** biases

Method	Mean	Std. Dev.	#Samples
PRINCE	**1.91***	0.66	200
CredPaths	**1.78***	0.63	200
PRINCE (Size = 1)	**1.87***	0.66	154
PRINCE (Size = 2)	**1.88***	0.70	28
PRINCE (Size = 2)	**2.21***	0.52	18

PRINCE is judged **more useful** at all explanation sizes!

Qualitative survey on usefulness in paper!

Contributions in PRINCE

- ★ **Counterfactual evidence** for discovering **causal explanations** in heterogeneous information networks
- ★ Explanations in terms of **users' own actions**
  - – **feasible** and **useful!!**
- ★ **Optimal algorithm** explores action space to find **minimal subsets**
- ★ Helps **provider** present **actionable, concise, and scrutable explanations!!**

Code github.com/azinmatin/prince/