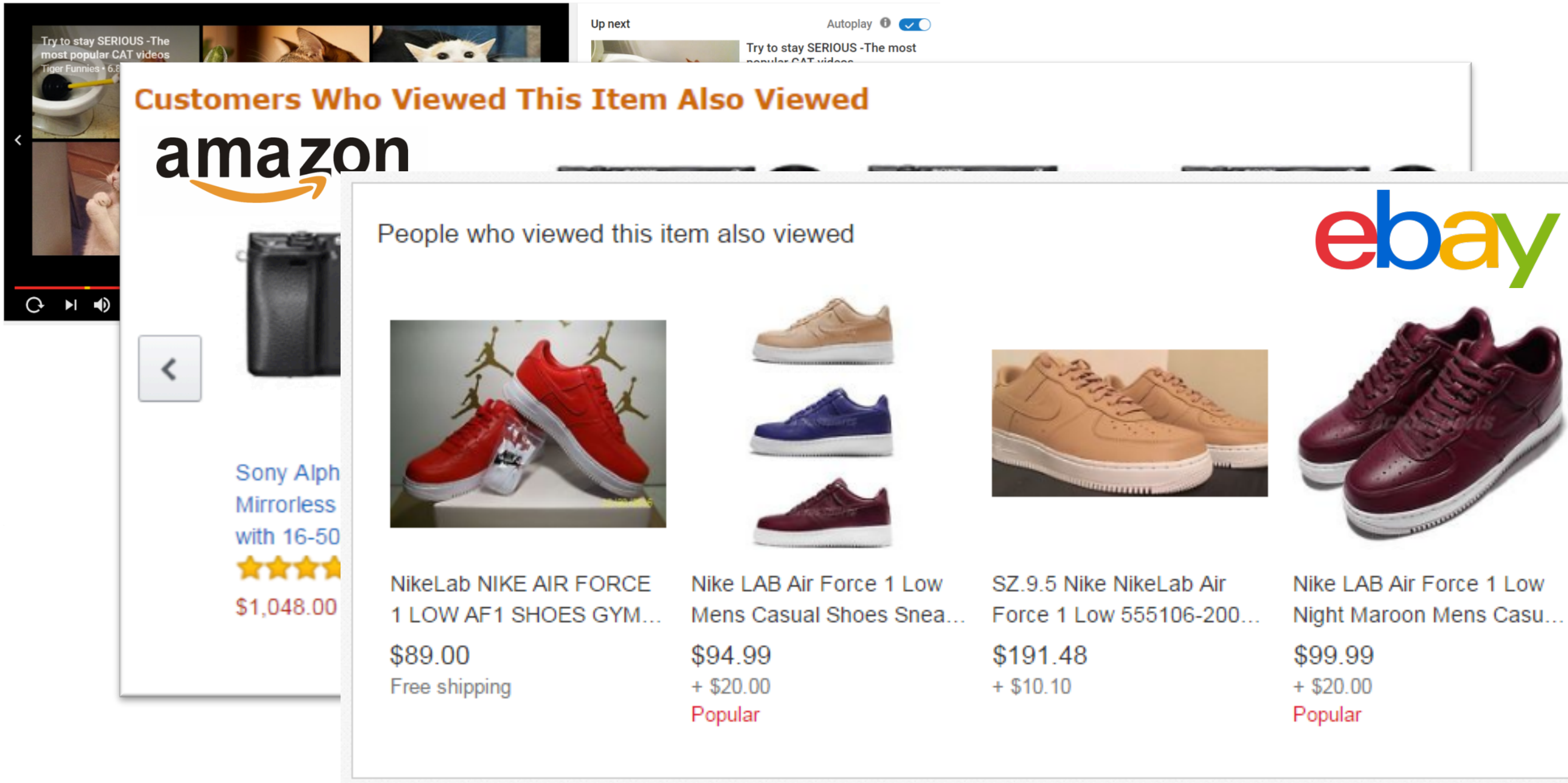
Fake Co-visitation Injection Attacks to Recommender Systems

Guolei Yang, Neil Zhenqiang Gong, Ying Cai

IOWA STATE UNIVERSITY

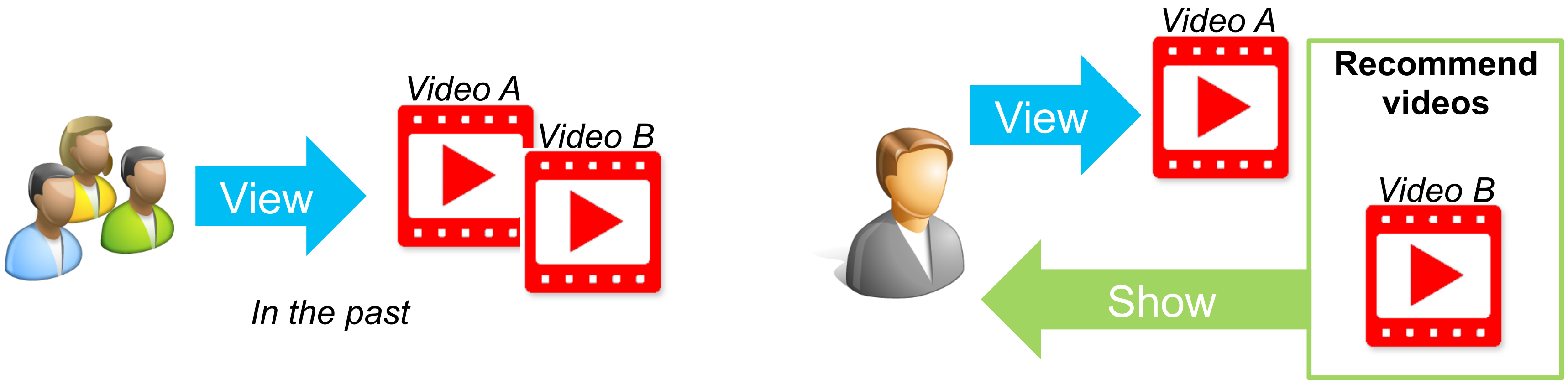
Co-visitation Recommender System is Popular



We show co-visitation recommender systems can be spoofed to recommend items as an attacker desires

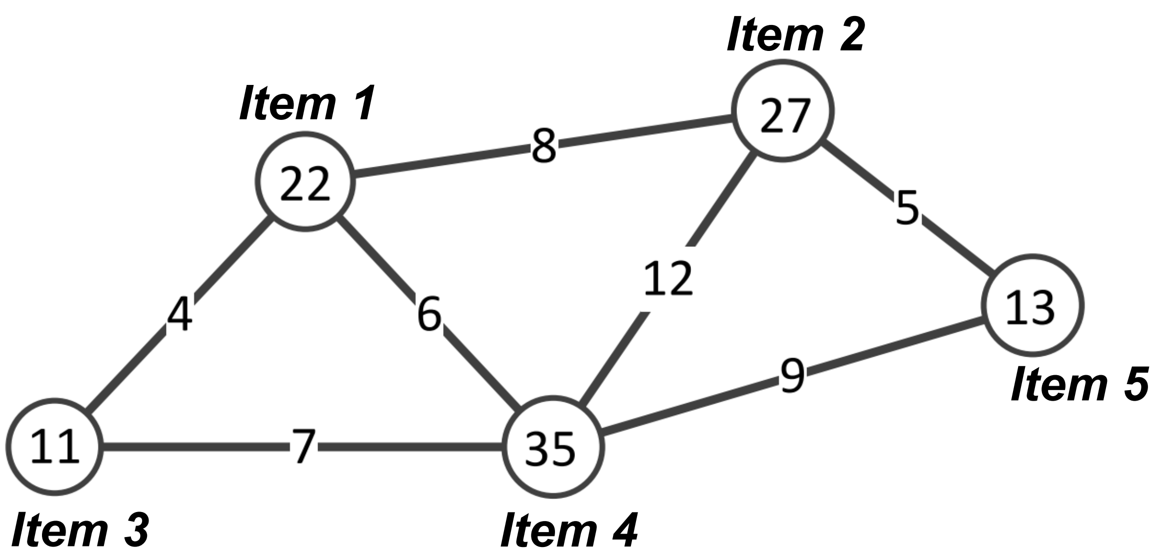
Brief Intro to Co-visitation Recommender System

- Key idea: *Items that are frequently visited together in the past are likely to be visited together in the future*

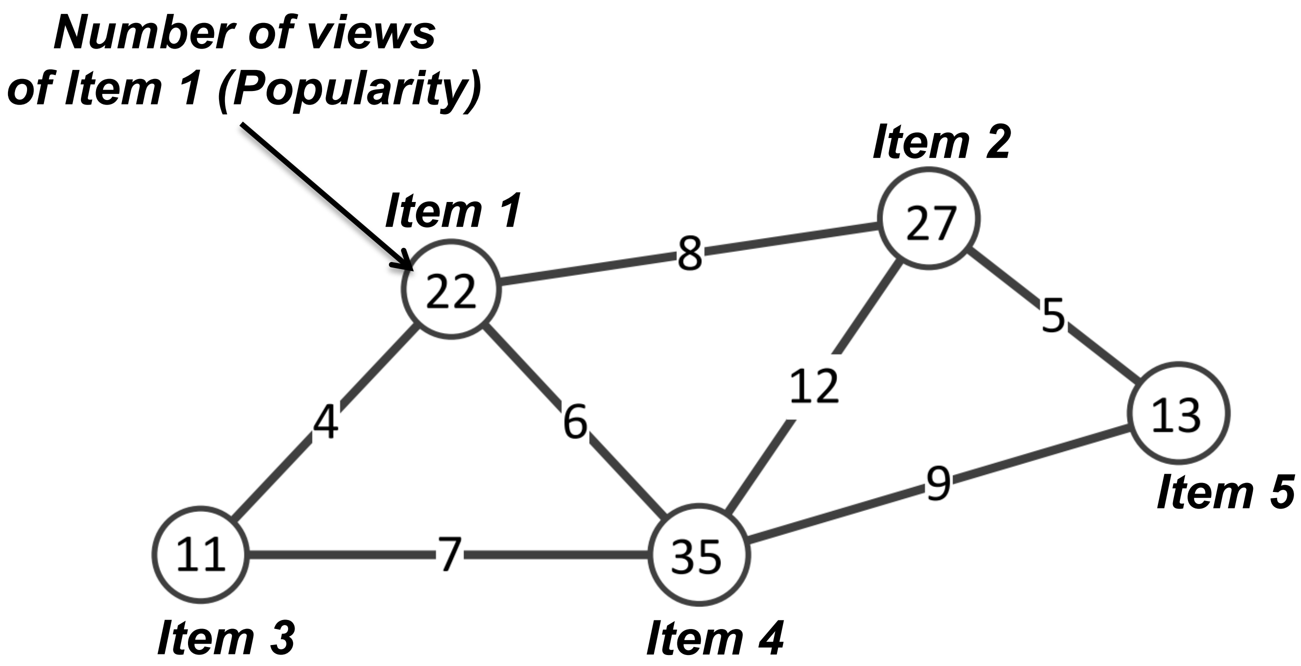


Key Data Structure: Co-visitation Graph

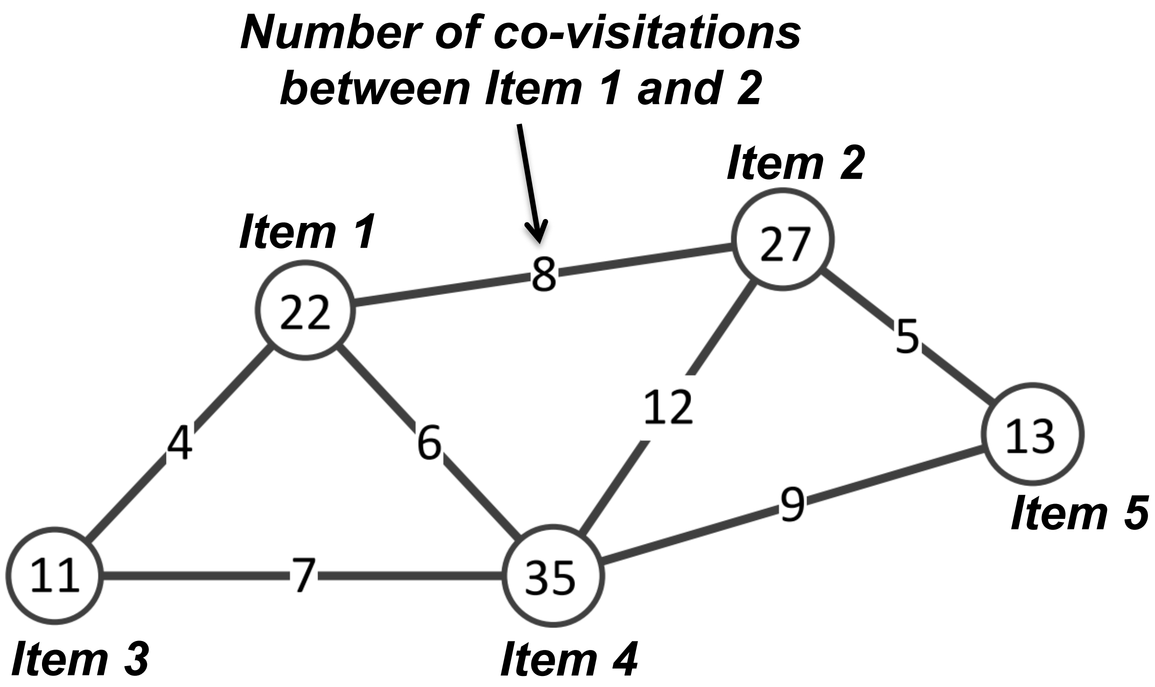
Each vertex represents an item



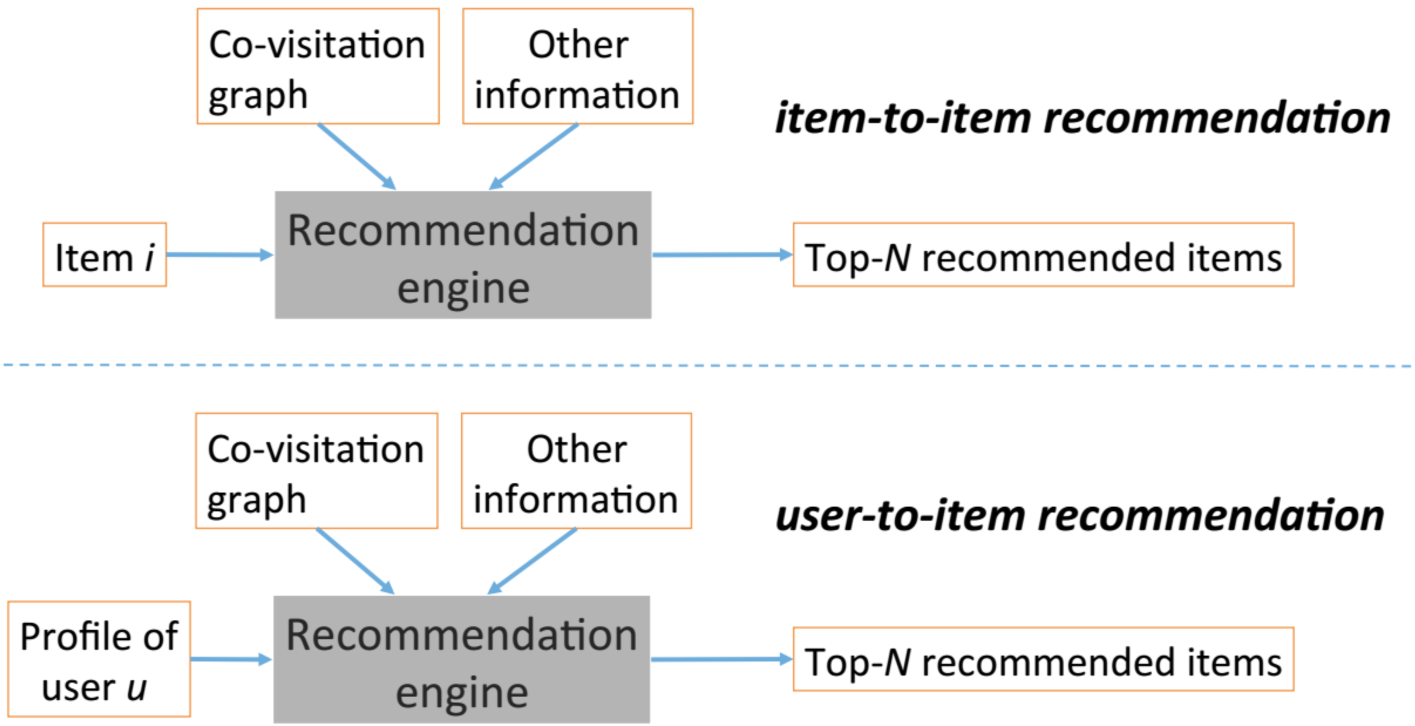
Key Data Structure: Co-visitation Graph



Key Data Structure: Co-visitation Graph

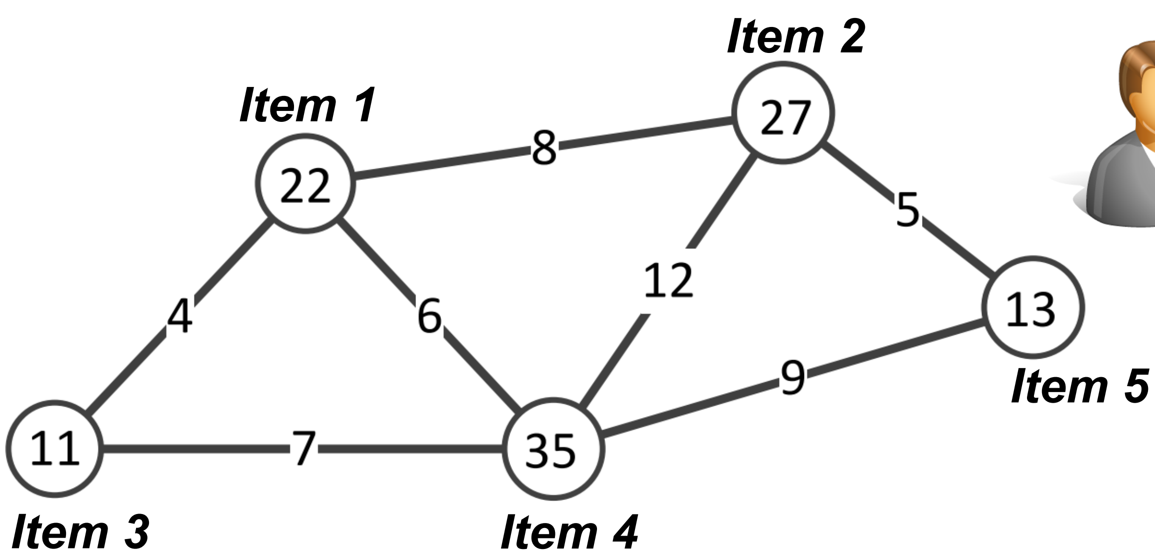


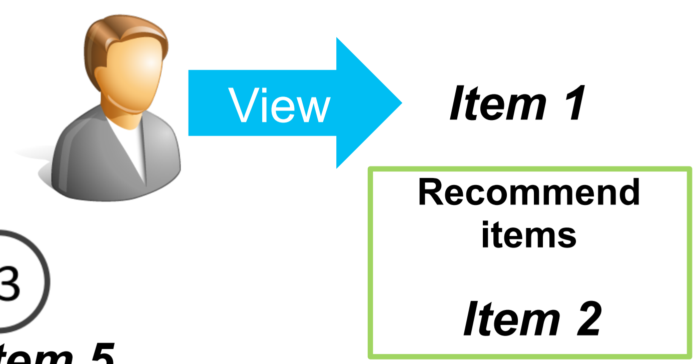
Two Recommendation Tasks



Item-to-Item Recommendation

Compute item-item similarity	Rank items by similarity	Generate recommendation list
$s_{ij} = \dfrac{w_{ij}}{f(w_i, w_j)}$ e.g., On YouTube $f(w_i, w_j) = w_i * w_j$	*For **Item 1:*** ***1. Item 2*** ***2. Item 4*** ***3. Item 3***	Include items 1) with high similarity 2) satisfy ***popularity threshold***





Related Work

- Xing et al. (USENIX Security'13) proposed *pollution attacks* to the user-to-item recommendation
  - *Relies on Cross-Site Request Forgery (CSRF)*
  - *Not applicable to item-to-item recommendation*

- *Profile injection* (*Shilling*) *attacks* to recommender systems via user-item rating matrices
  - Not applicable to co-visitation recommender systems which do not rely on user-item rating matrix.

- Relationship to adversarial machine learning
  - Our attack is data poisoning attack to recommender systems

Roadmap

- Threat model
- Proposed attacks
- Evaluations on synthetic data
- Evaluations on real-world recommender systems
- Countermeasures

Threat Model

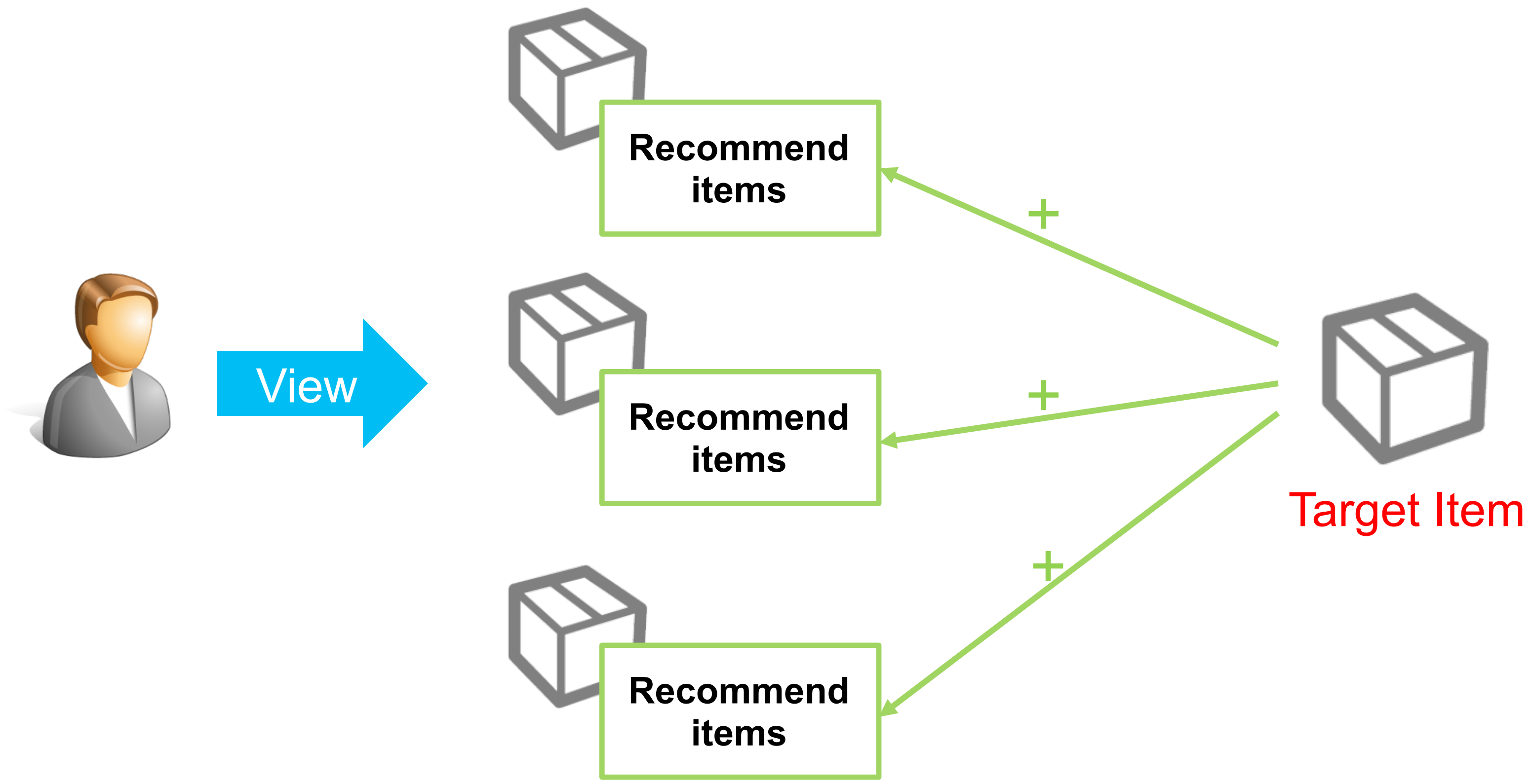
- Attacker's background knowledge

	High knowledge	Medium knowledge	Low knowledge
Knowledge	*Co-visitation Graph* *Popularity Threshold*	*Recommendation Lists* *Item Popularity*	*Recommendation Lists*
Scenario	*Insider*	*YouTube …*	*Amazon, eBay…*

- Attacker's goal
  - User Impression (**UI**) : The probability that a random visitor will see the item
  - Increase UI of a target item
  - Decrease UI of a target item

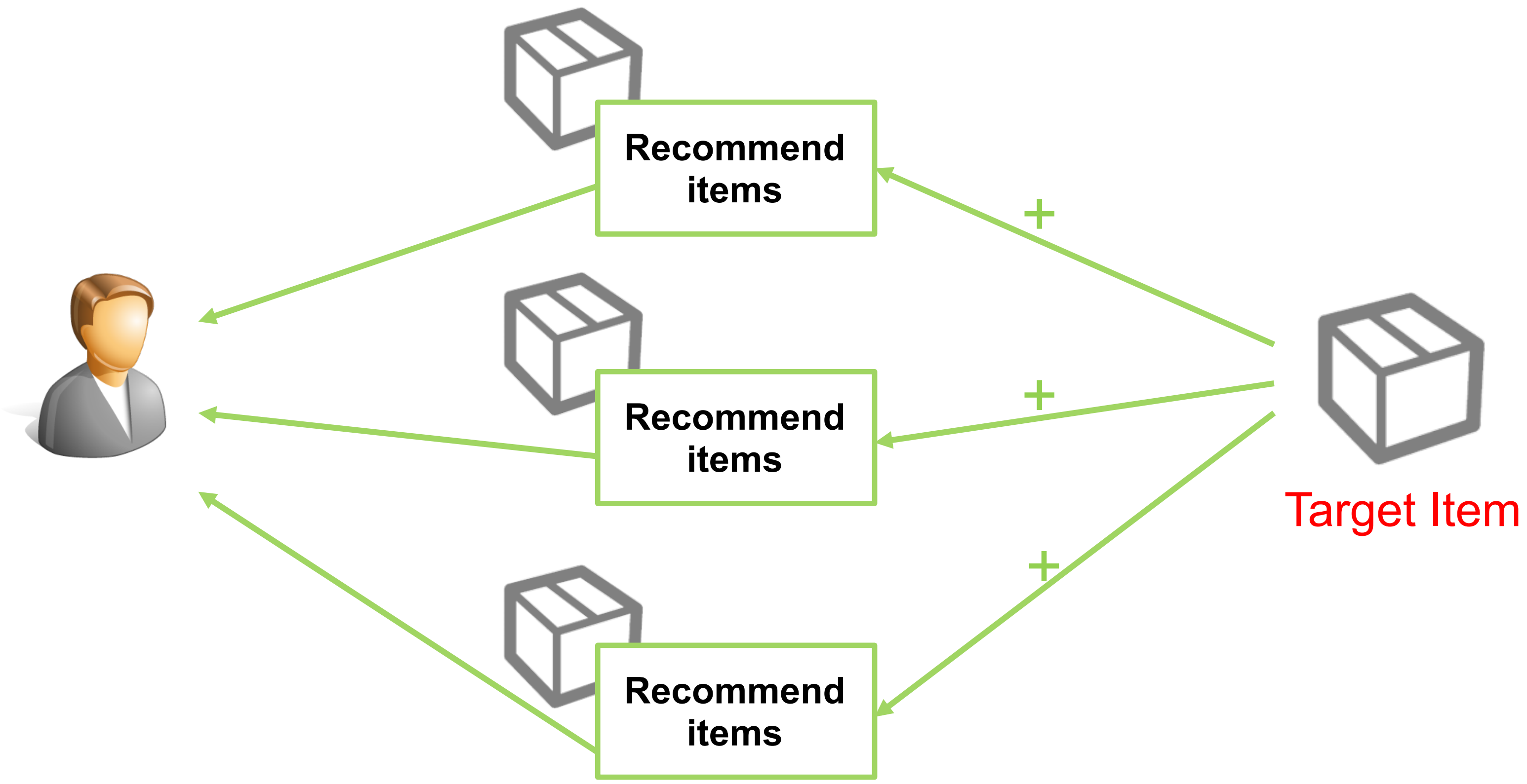
Proposed Attacks

- Promotion attack
  - Goal: Increase UI of a Target Item
  - Make the target Item appear in the recommendation lists of as many items as possible



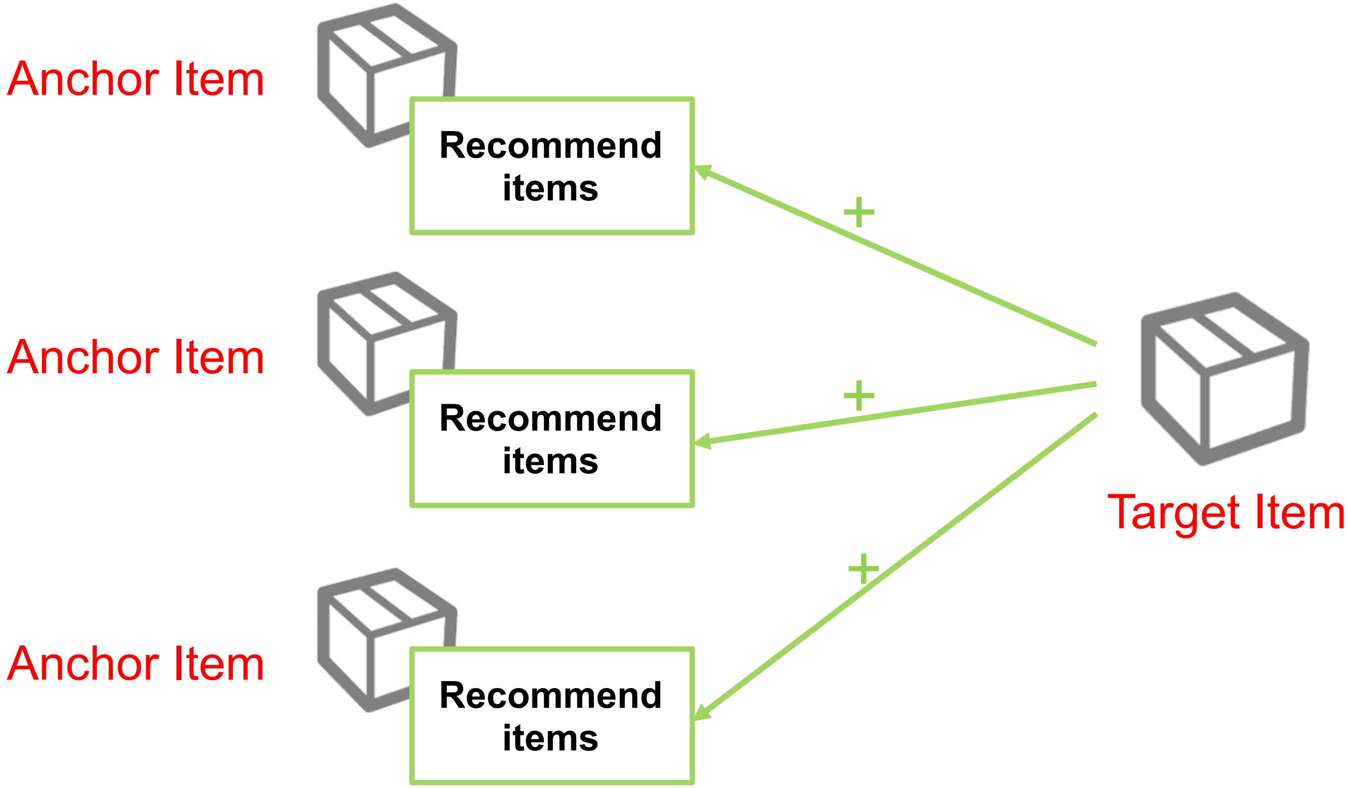
Proposed Attacks

- Promotion attack
  - Goal: Increase UI of a Target Item
  - Make the target Item appear in the recommendation lists of as many items as possible



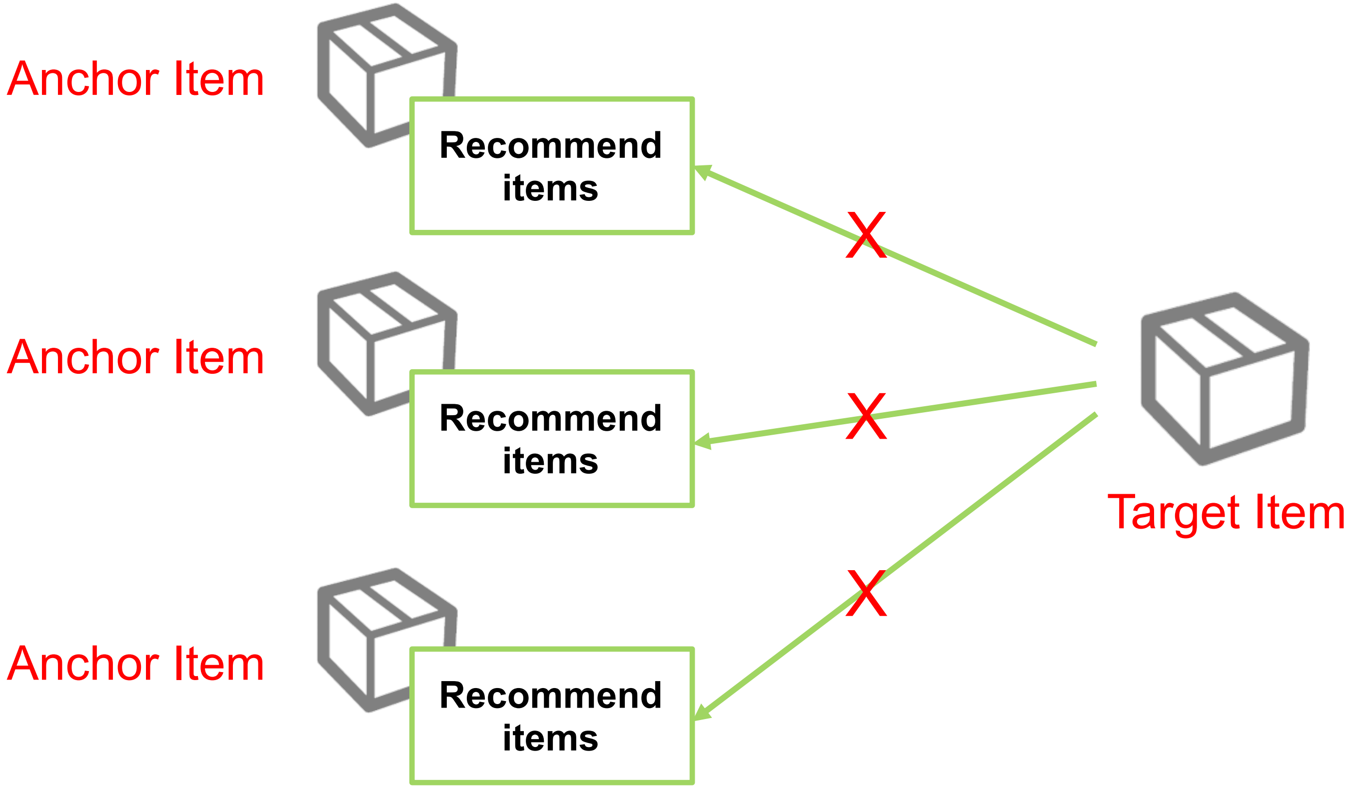
Proposed Attacks

- Promotion attack
  - Goal: Increase UI of a Target Item
  - Make the target Item appear in the recommendation lists of as many items as possible



Proposed Attacks

- Demotion attack
  - Goal: Decrease UI of a Target Item
  - Remove the target Item from the recommendation lists of as many items as possible



Key Challenge

- Given a target item and a limited number fake co-visitations
  - *How to select the anchor item(s) to attack?*
  - *How many fake co-visitations to insert for each anchor item?*

Key Challenge

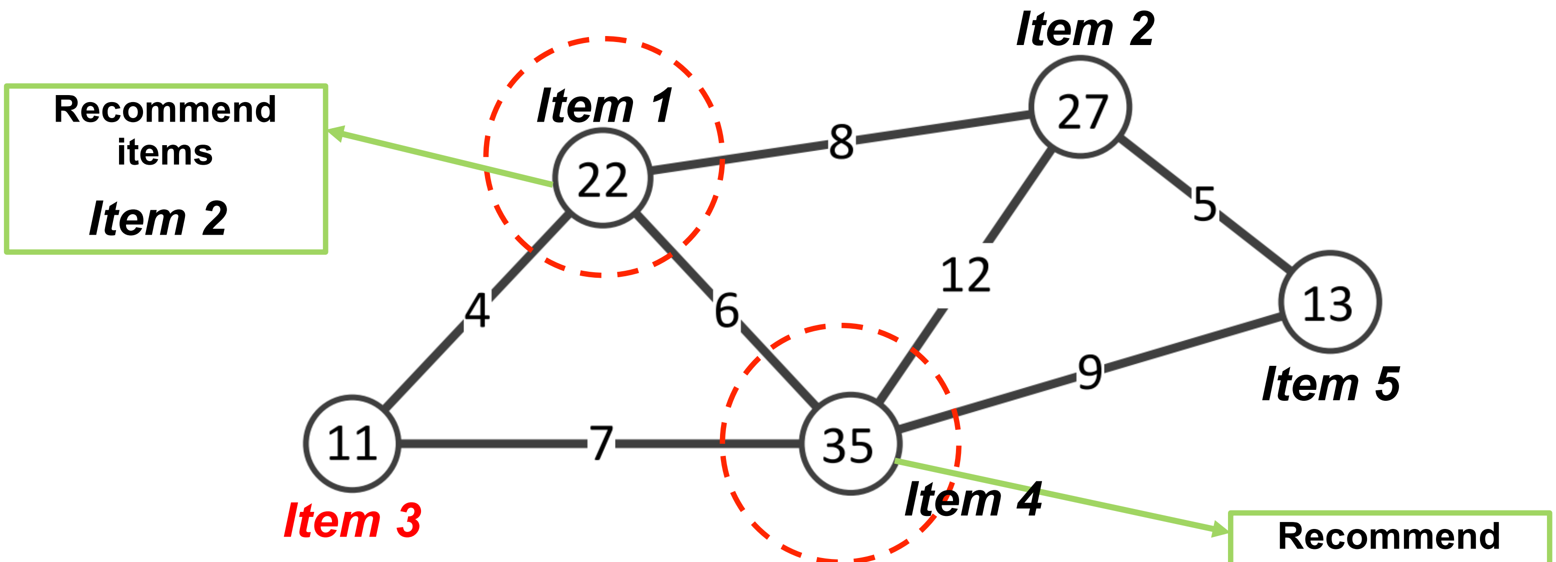
- Given a target item
  - *How to select the anchor item(s) to attack?*
  - *How many fake co-visitations to insert for each anchor item?*

- Solution: Formulate the attack as an optimization problem
  - *Select the best anchor items to attack*
  - *Determine how many fake co-visitation is needed to attack each anchor*

Promotion Attack – High Knowledge Attacker

Attacker's Goal: Promote *Item 3*

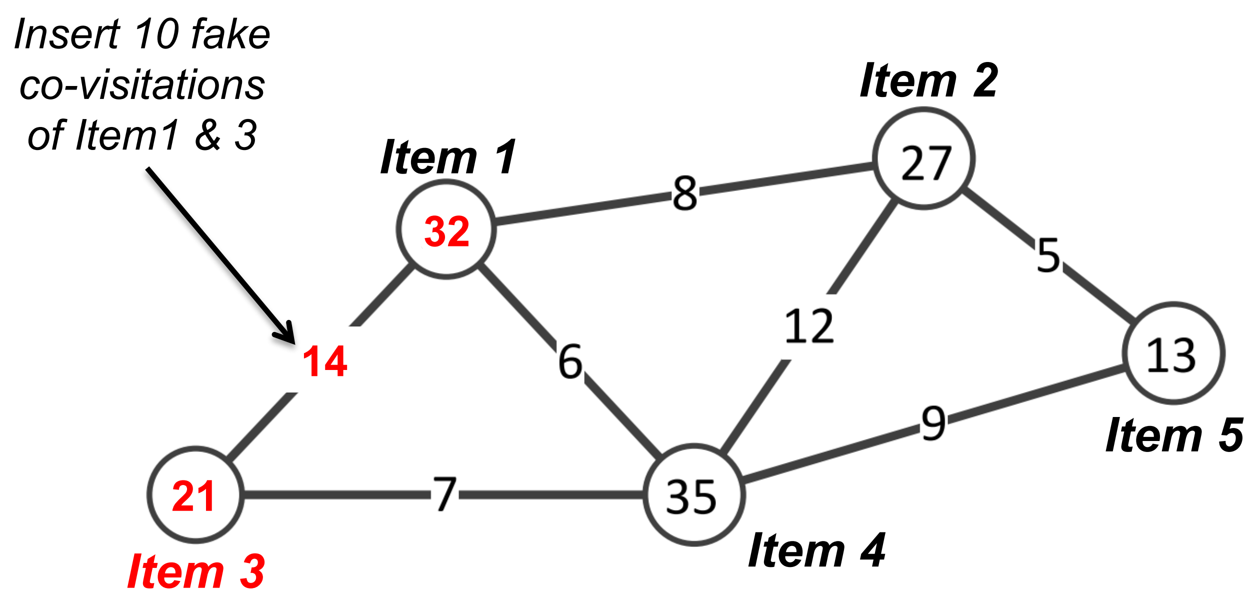
Select anchor items



Original Co-visitation graph

Promotion Attack – High Knowledge Attacker

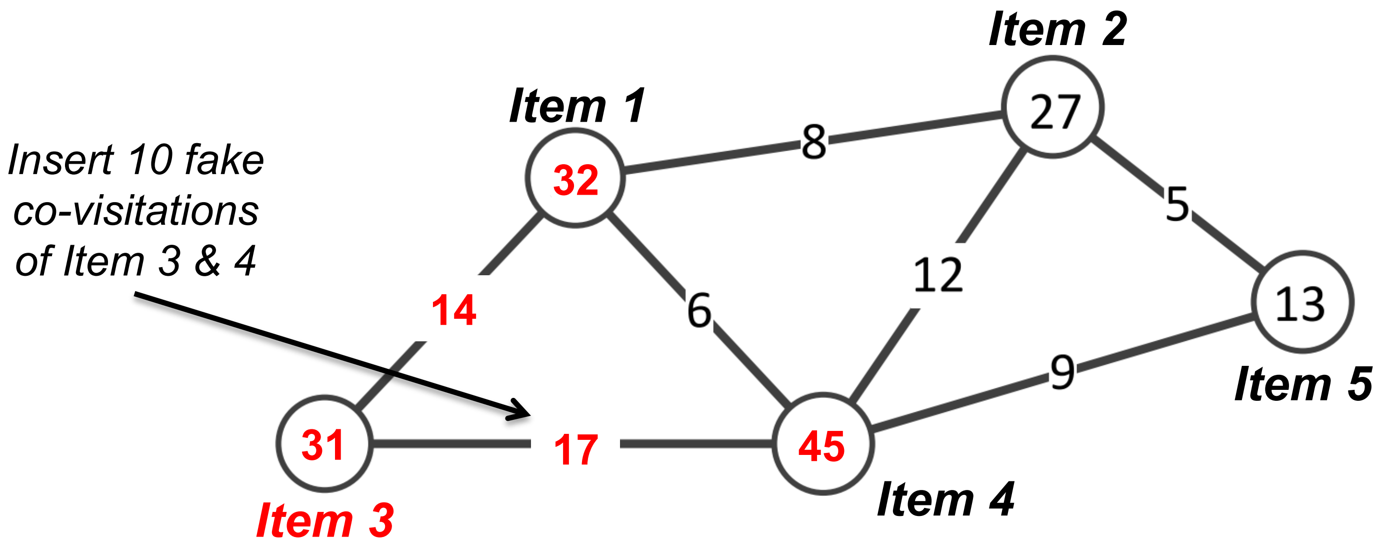
Attacker's Goal: Promote *Item 3*



Attacked Co-visitation graph

Promotion Attack – High Knowledge Attacker

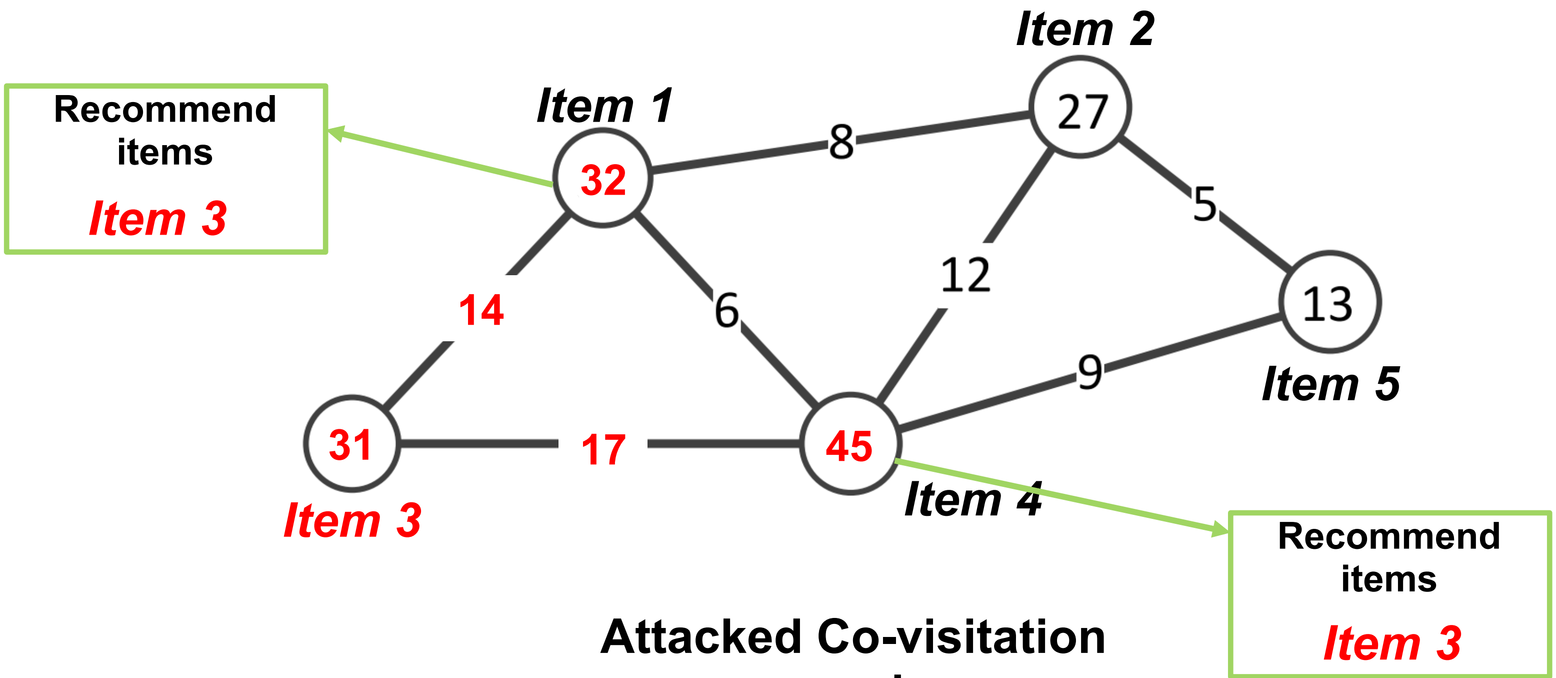
Attacker's Goal: Promote *Item 3*



Attacked Co-visitation graph

Promotion Attack – High Knowledge Attacker

Attacker's Goal: Promote *Item 3*



Attacked Co-visitation graph

Promotion Attack – High Knowledge Attacker

Attacker's Goal: Promote *Item 3*

- Medium knowledge attacker can be converted into high knowledge attacker by *estimating edge weight*
- Low knowledge attacker can be converted into medium knowledge attacker by *estimating vertex weight*

Item 3

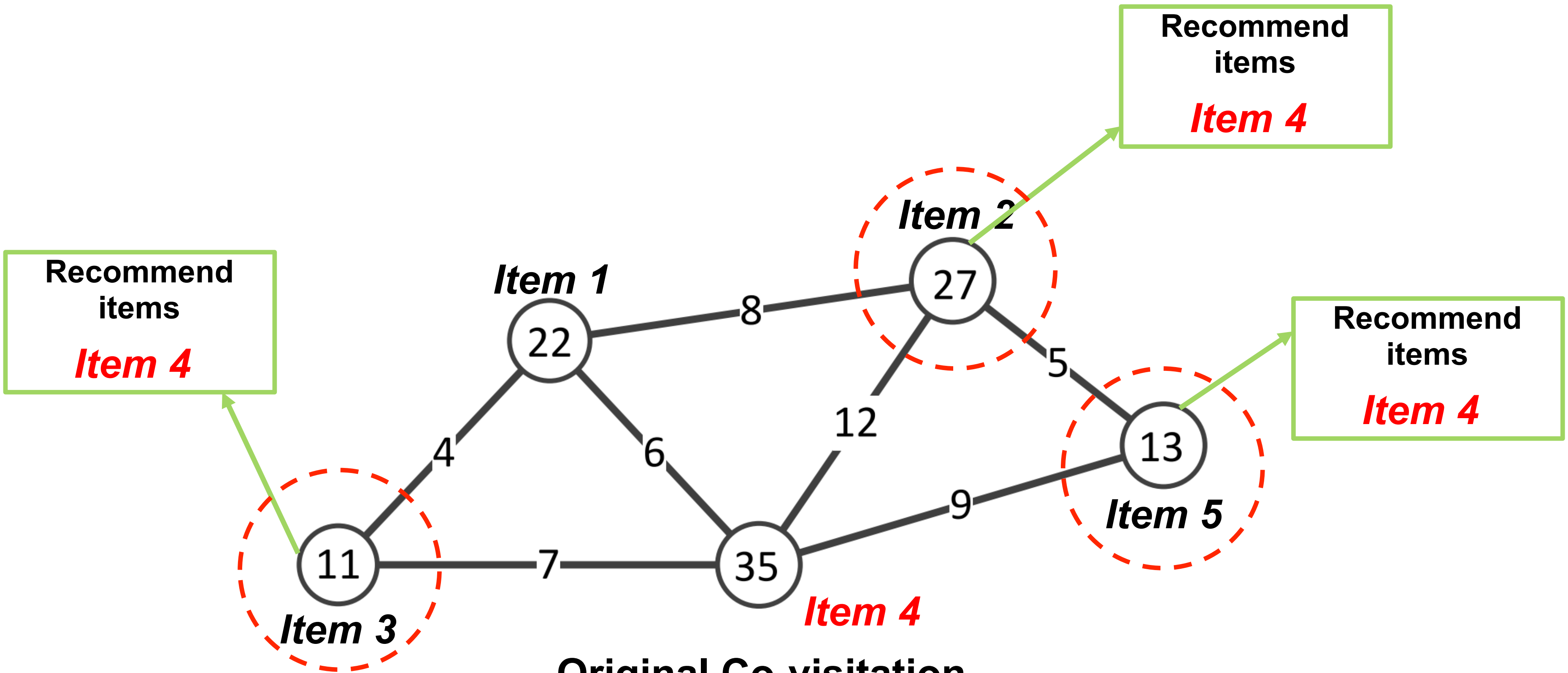
Recommend items

Item 3

Attacked Co-visitation graph

Demotion Attack – High Knowledge Attacker

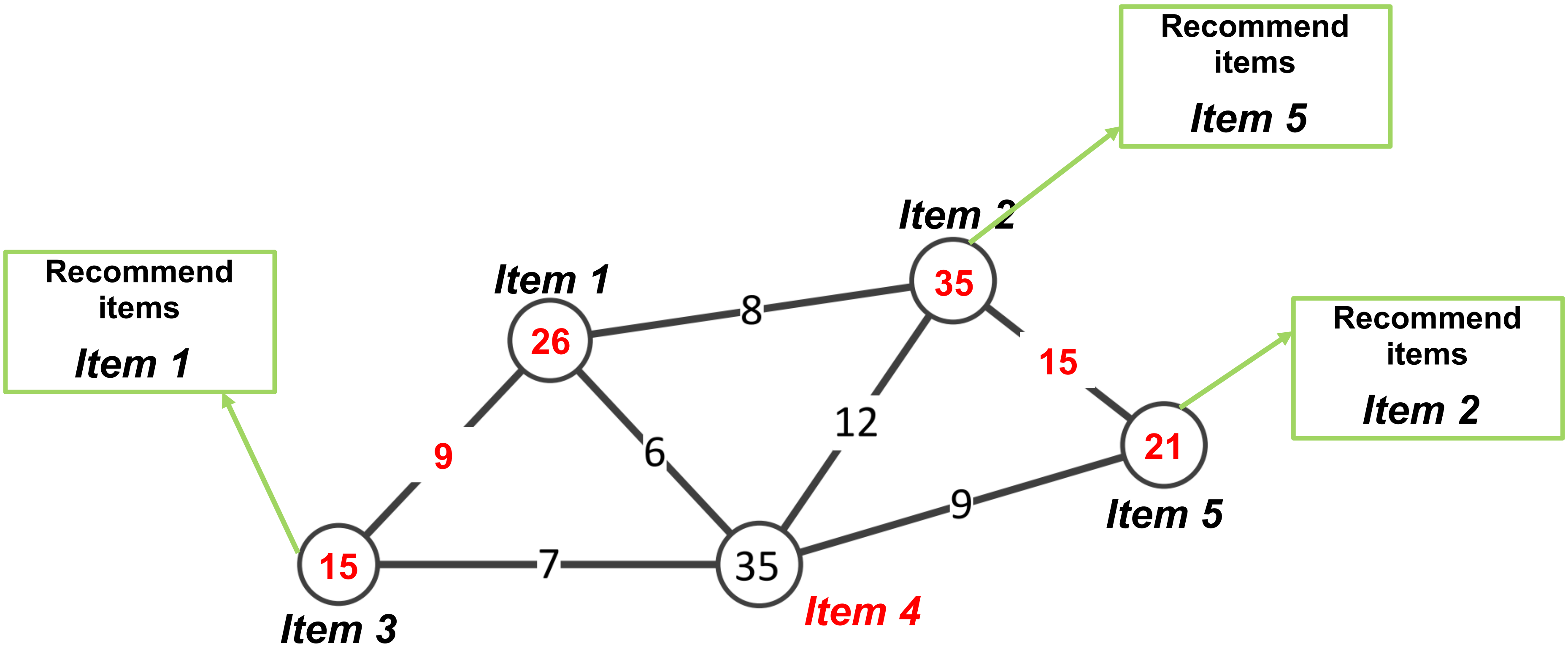
Attacker's Goal: Demote *Item 4*



Original Co-visitation graph

Demotion Attack – High Knowledge Attacker

Attacker's Goal: Demote *Item 4*

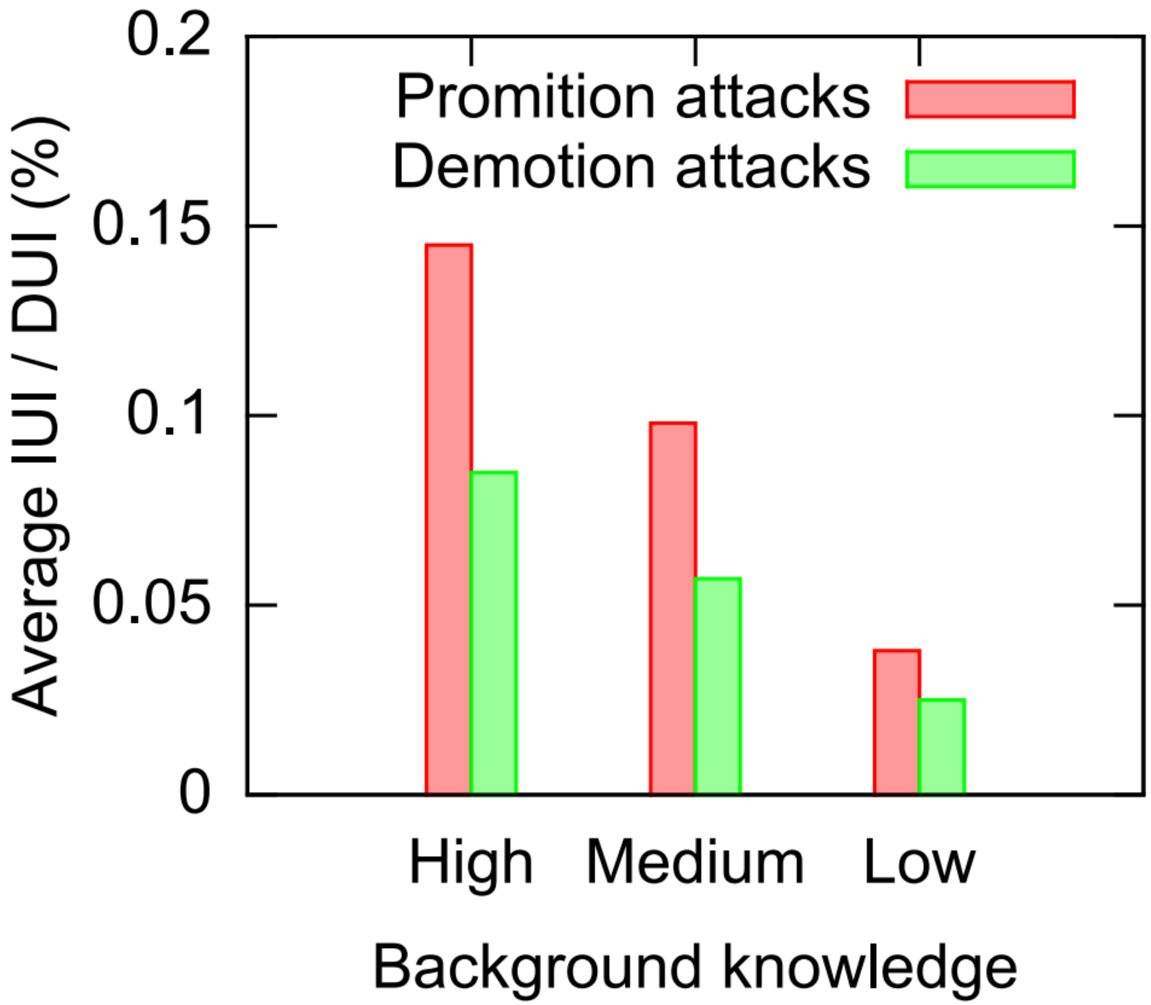


Attacked Co-visitation graph

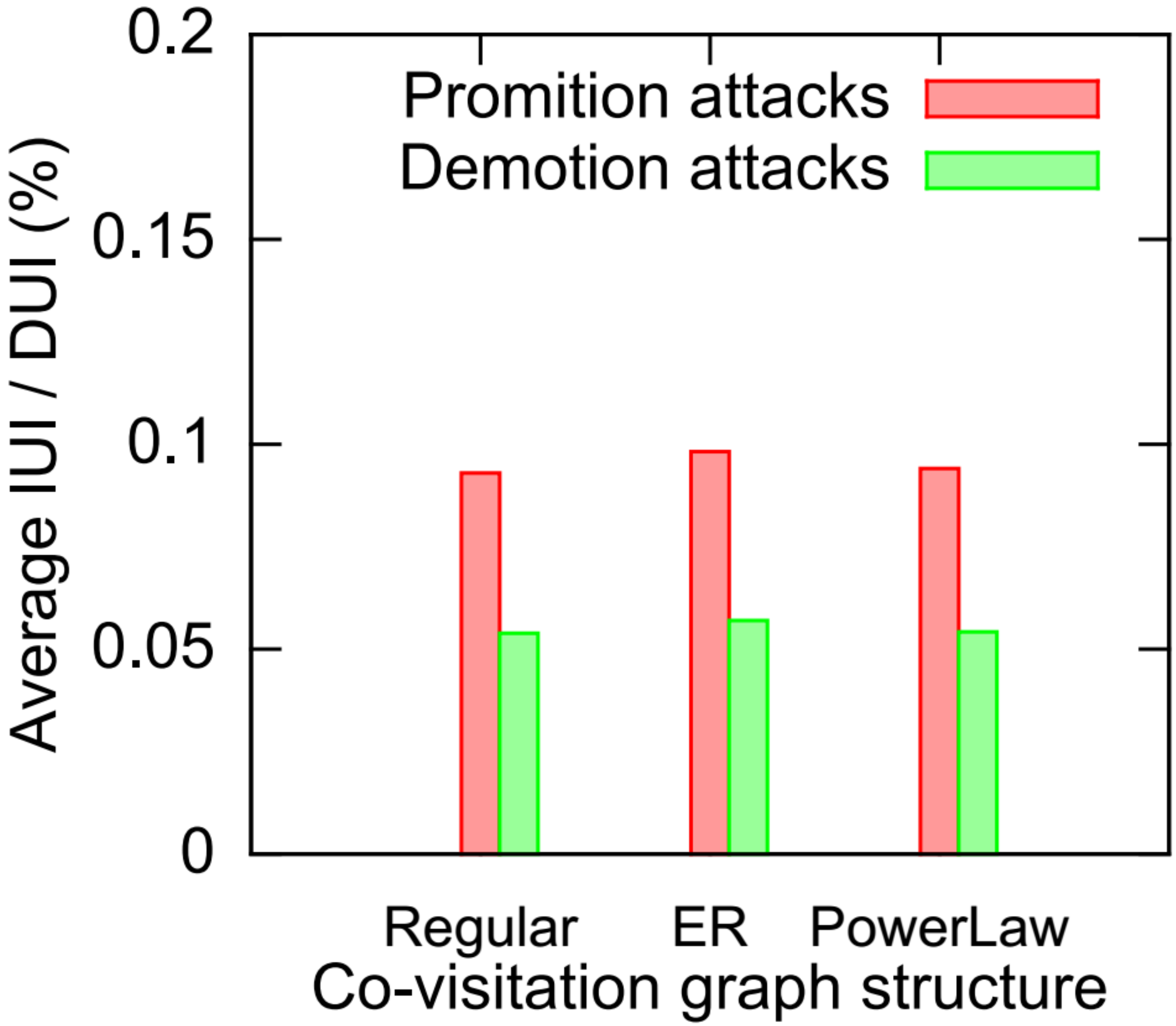
Evaluation on Synthetic Data

- Question we aim to answer
  - How does attacker's background knowledge impact our attacks
  - How does the co-visitation graph structure impact our attacks?
  - How does the number of inserted fake co-visitations impact our attacks?

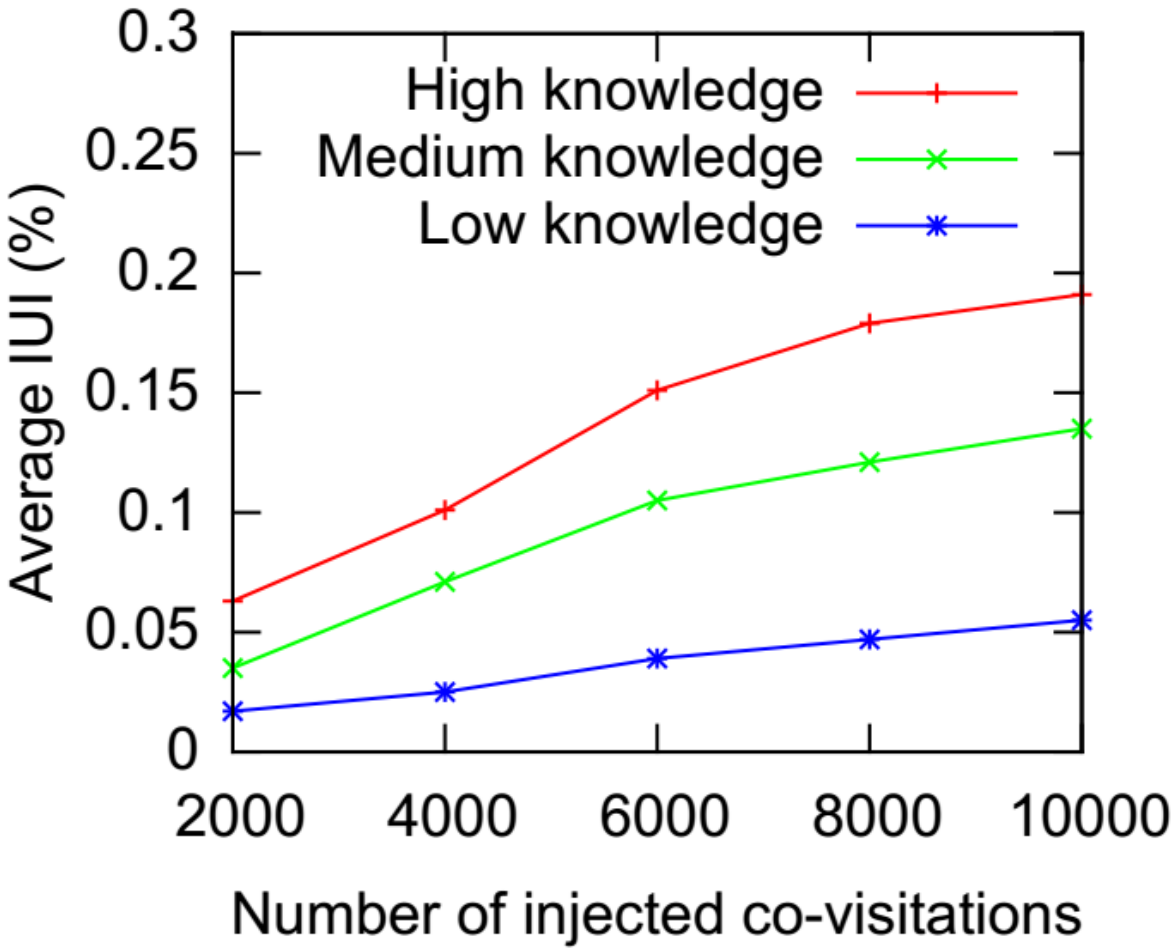
Impact of Attacker's Background Knowledge



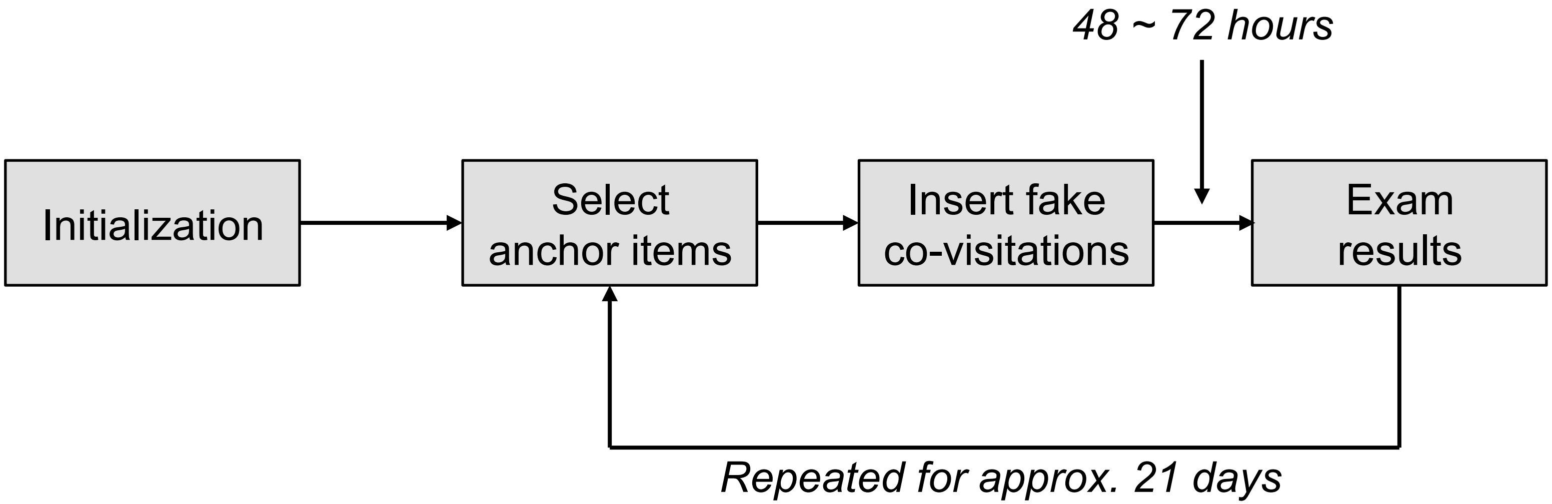
Impact of Co-visitation Graph Structure



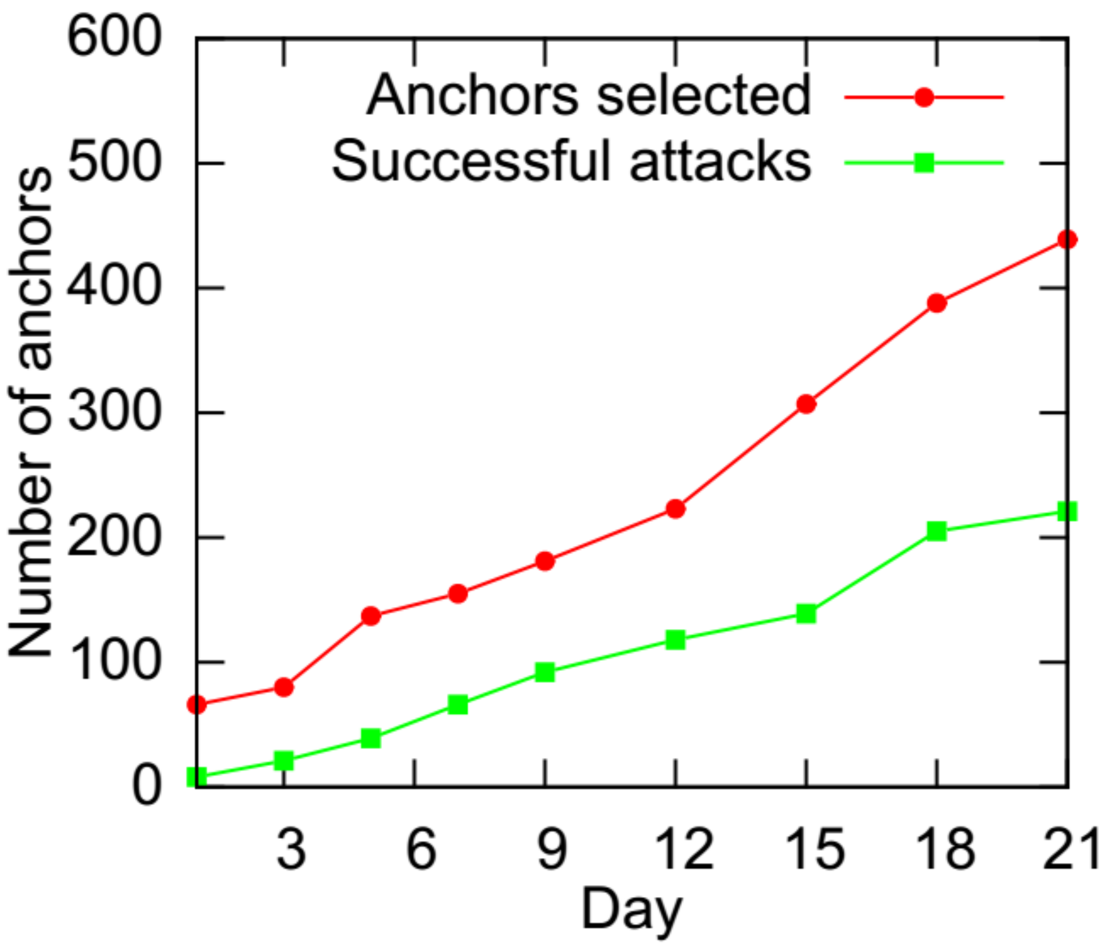
Impact of Number of Fake Co-visitations



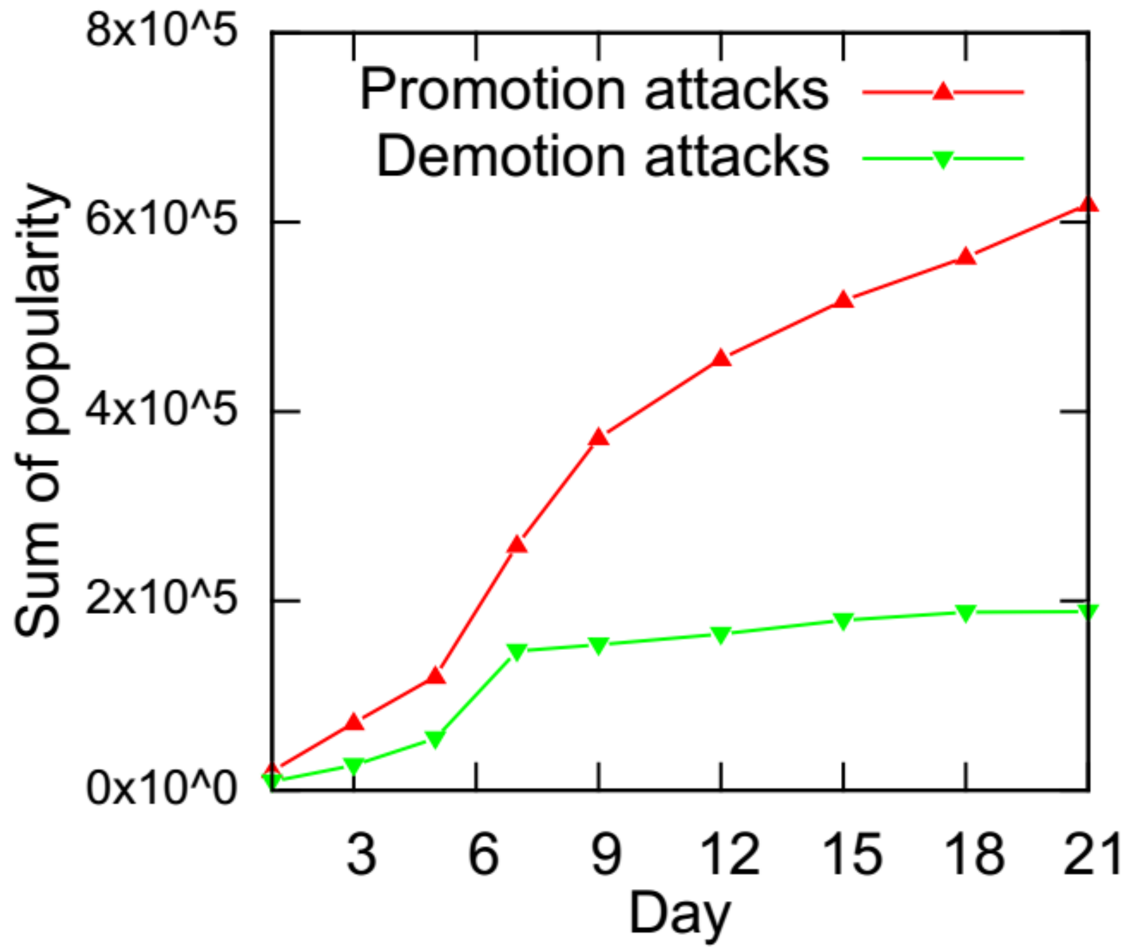
Evaluation on Real-World Recommender Systems



Results on YouTube



Results on YouTube



Countermeasures

- Limiting background knowledge
  - The website can *discretize item popularities*

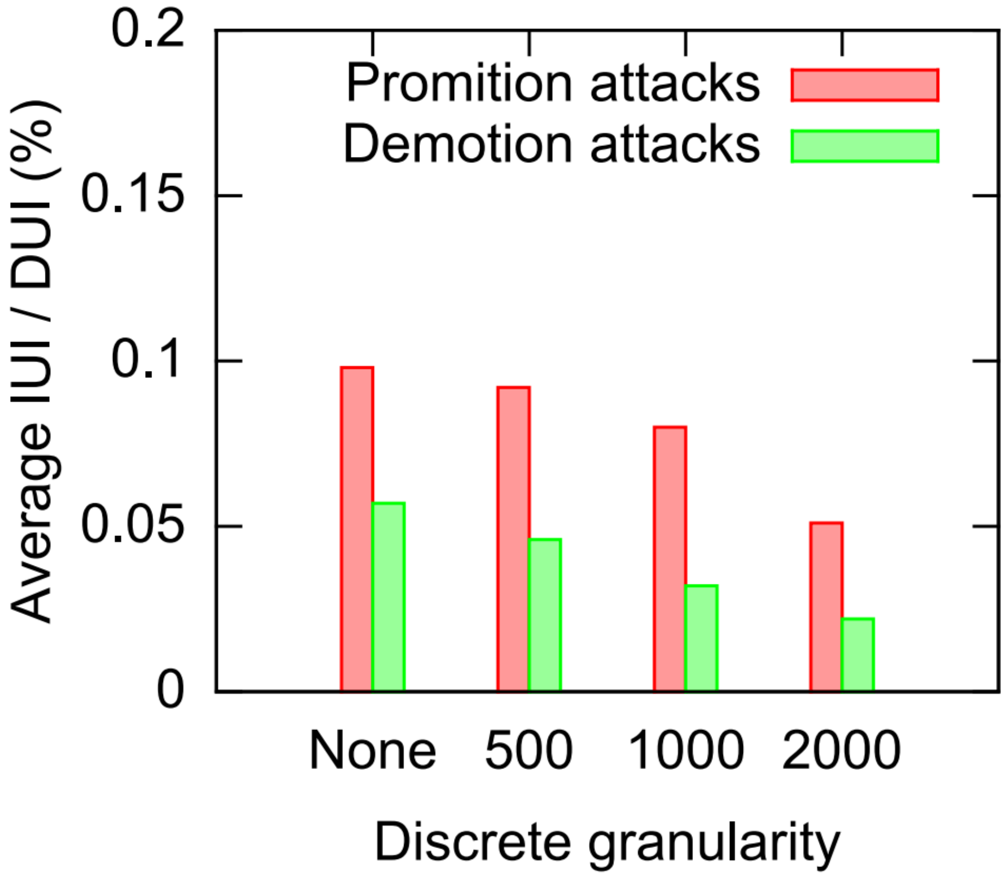
Shows exact popularity

Discretize Granularity = 500

Discretize Granularity = 2000

Countermeasures

- Limiting background knowledge
  - The website can *discretize item popularities*

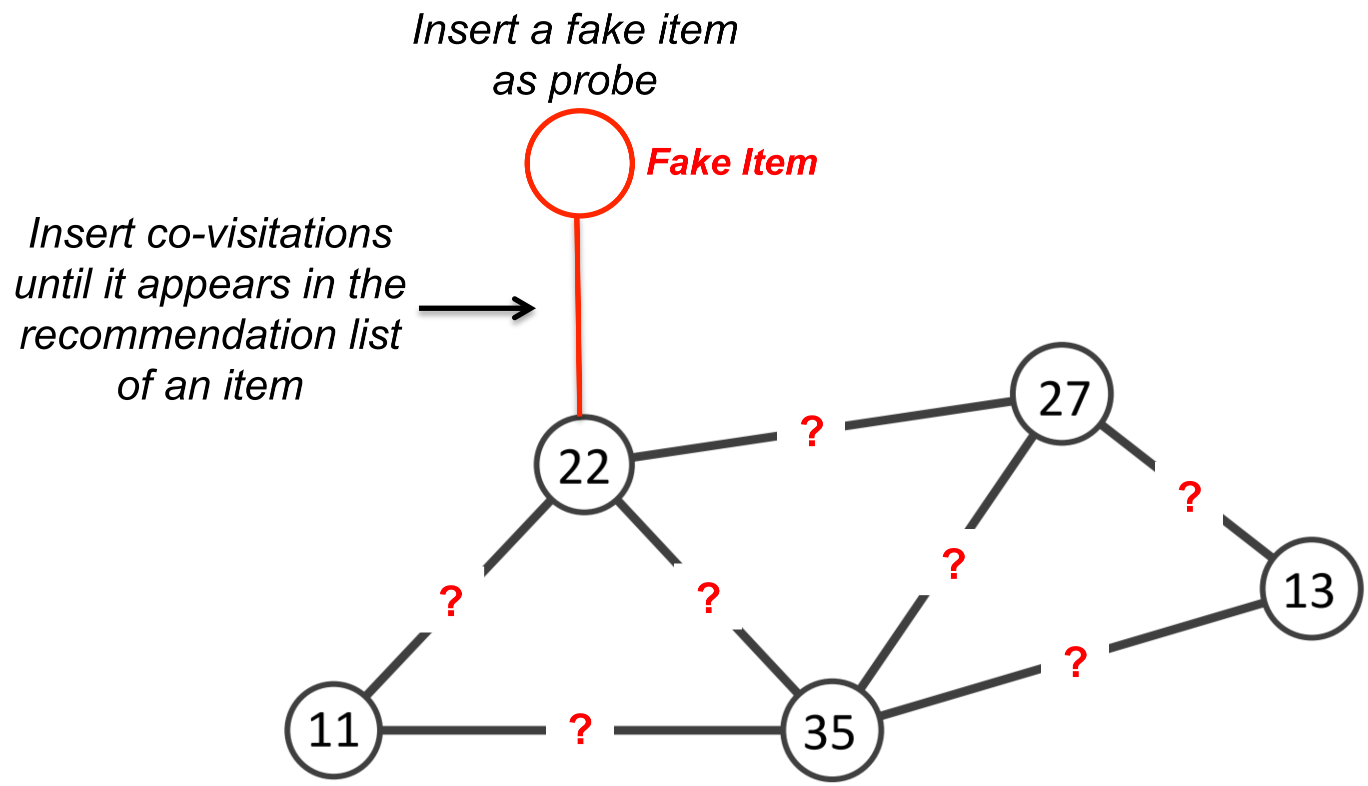


Conclusion

- Recommender systems are vulnerable to *Fake Co-visitation Injection Attacks*
- An attacker can use our attacks to spoof a recommender system to make recommendations as the attacker desires.

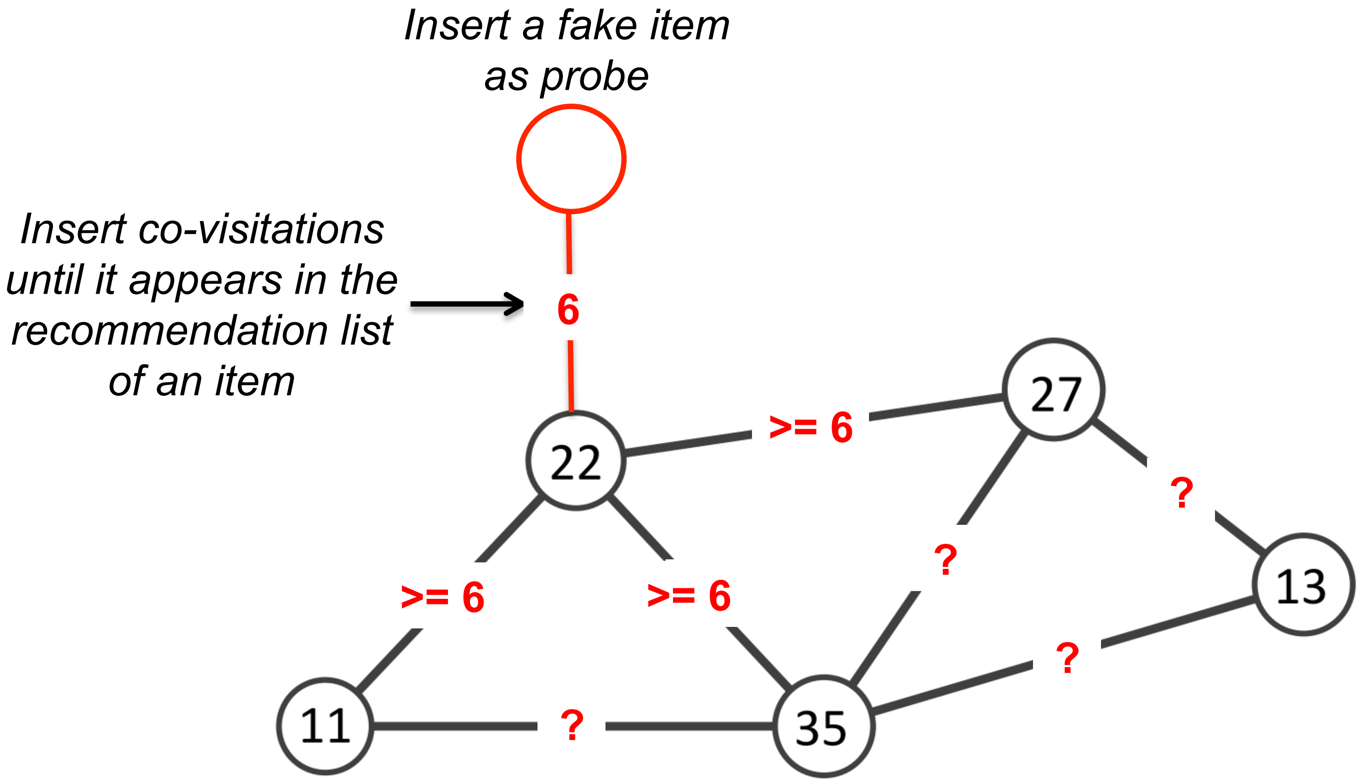
Parameter Estimation

- Convert *medium/low knowledge attackers* into *high knowledge attacker*
  - The missing knowledge is estimated based on publically available information



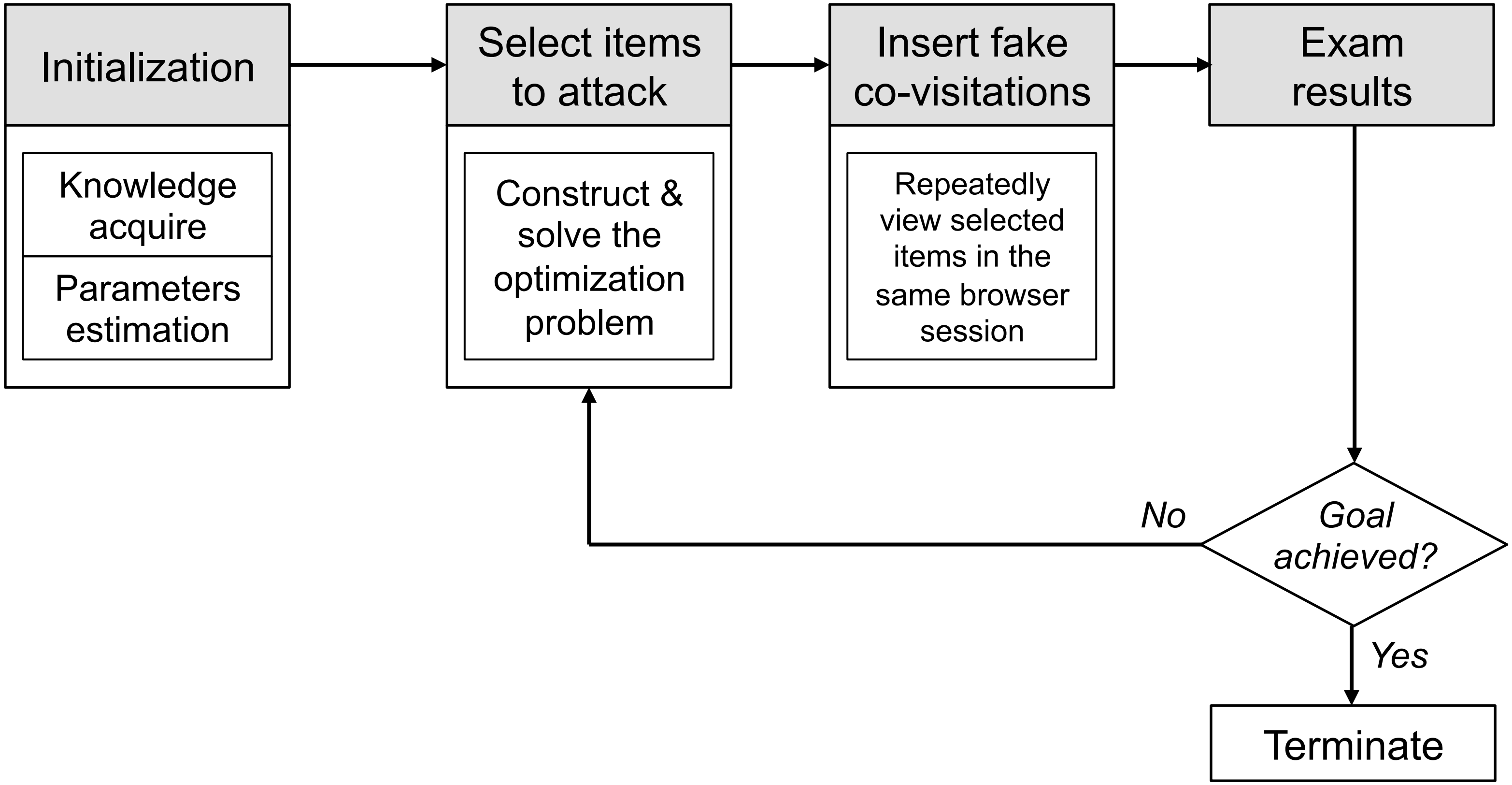
Parameter Estimation

- Convert *medium/low knowledge attackers* into *high knowledge attacker*
  - The missing knowledge is estimated based on publically available information



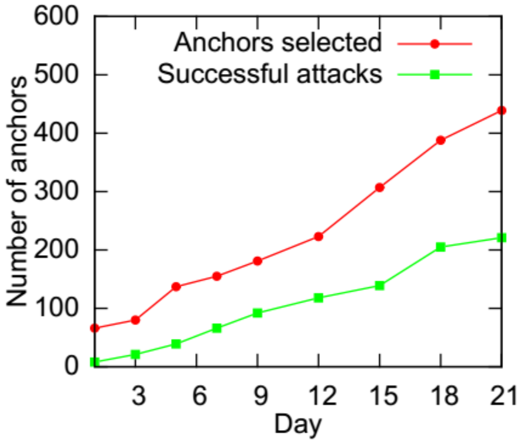
Proposed Attack Algorithm

- General steps

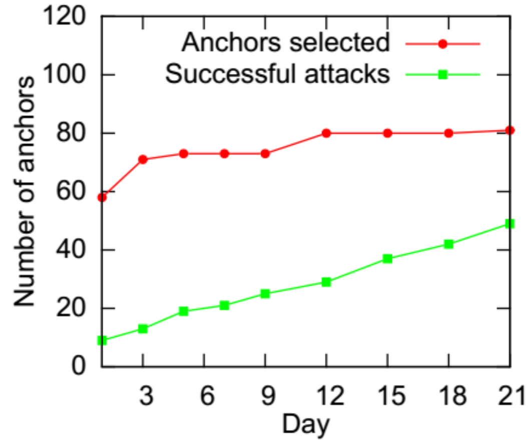


Experiments on Real-world Recommder Systems

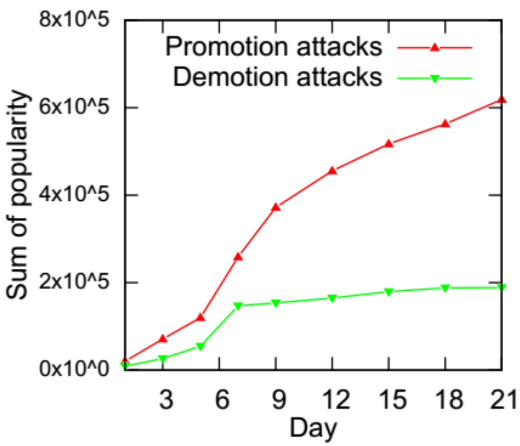
- Results on *YouTube*



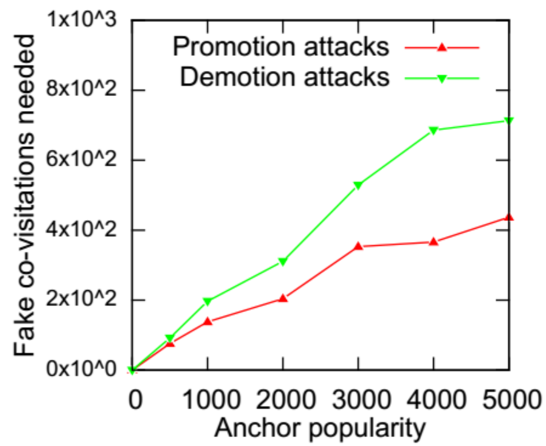
(a) Promotion attacks



(b) Demotion attacks



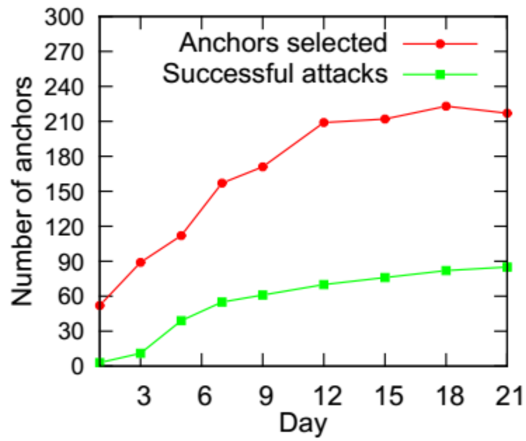
(c) Popularity of successfully attacked anchors



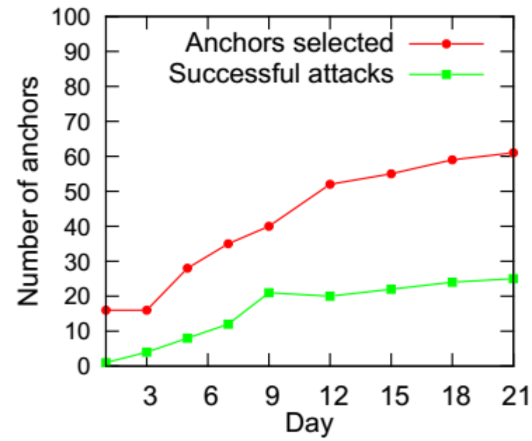
(d) Cost vs. anchor popularity

Experiments on Real-world Recommder Systems

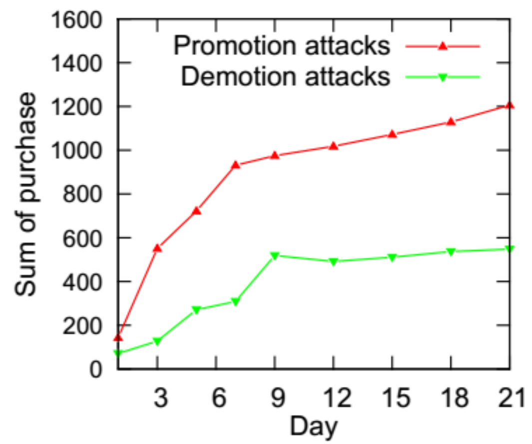
- Results on *eBay*



(a) Promotion attack



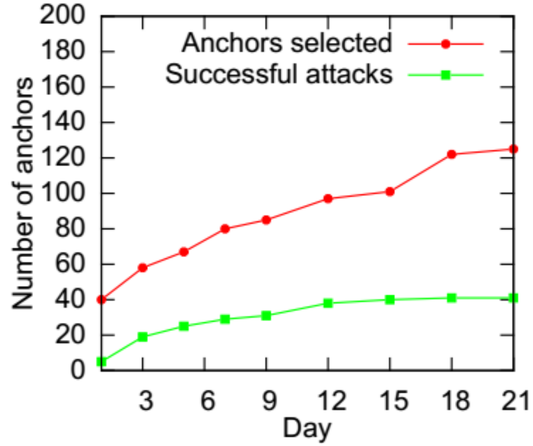
(b) Demotion attacks



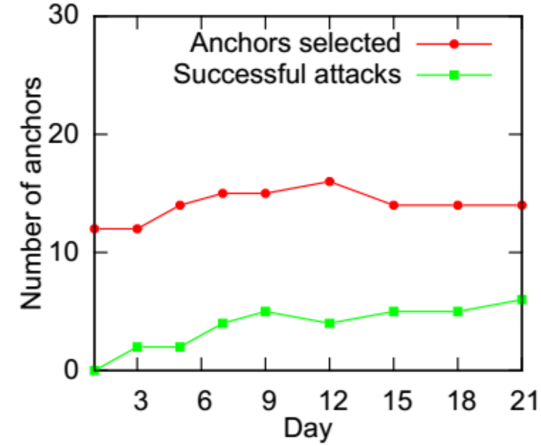
(c) Purchases of successfully attacked anchors

Experiments on Real-world Recommder Systems

- Results on *Amazon*

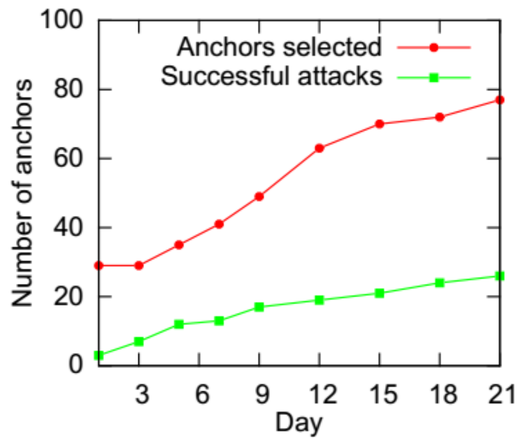


(a) Promotion attacks

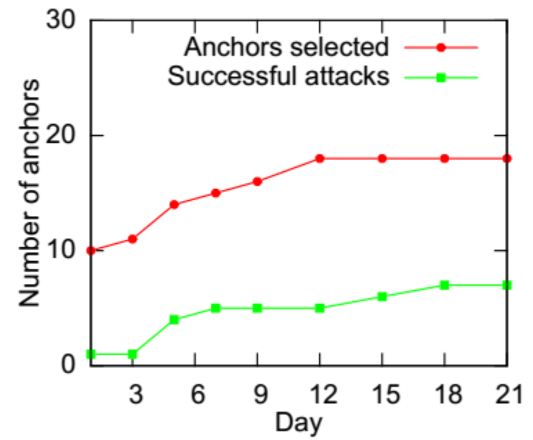


(b) Demotion attacks

- Results on *Yelp*



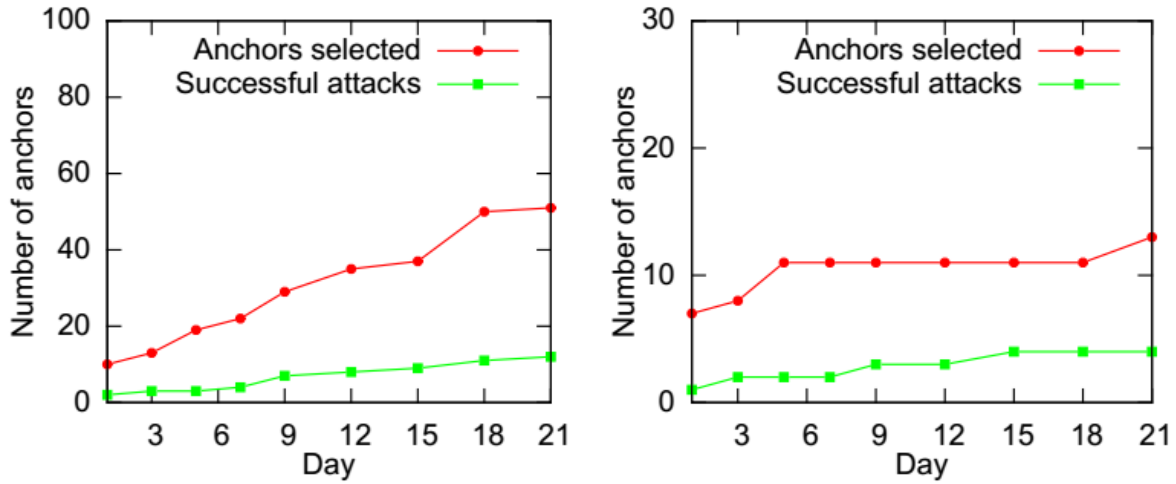
(a) Promotion attacks



(b) Demotion attacks

Experiments on Real-world Recommder Systems

- Results on *LinkedIn*

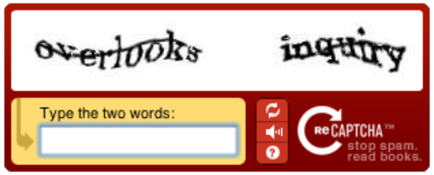


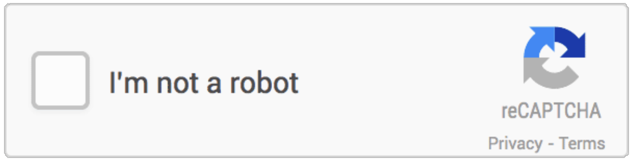
(a) Promotion attacks

(b) Demotion attacks

Countermeasures

- Limiting fake co-visitations
  - Use CAPTCHA





  - Fake co-visitation detection
  - Using co-visitations from registered users only