Information Storage and Retrieval

CSCE 670
Texas A&M University
Department of Computer Science & Engineering
Instructor: Prof. James Caverlee

Attacks on Recommenders
5 April 2018

Attacks on Recommenders

- Why?
    - Attract attention to particular items
    - Nuke attention on particular items
    - Joker strategy: “watch the world burn”
    - … others?
- Super important these days …
    - Manipulate opinion, news exposure, …
    - …

Strategy Version 0

- First, create many fake accounts
- Then, issue high or low ratings to the "target item"
- Done (?)
- Why not?

	Item1	Item2	Item3	Item4	…	Target	Pearson
Alice	5	3	4	1	…	?	
User1	3	1	2	5	…	5	-0.54
User2	4	3	3	3	…	2	0.68
User3	3	3	1	5	…	4	-0.72
User4	1	5	5	2	…	1	-0.02

	Item1	Item2	Item3	Item4	…	Target	Pearson
Alice	5	3	4	1	…	**?**	
User1	3	1	2	5	…	5	-0.54
User2	4	3	3	3	…	2	0.68
User3	3	3	1	5	…	4	-0.72
User4	1	5	5	2	…	1	-0.02
Attack	**5**	**3**	**4**	**3**	**…**	**5**	**0.87**

Push vs. Nuke Attack

Item1	…	ItemK	…	ItemL	…	ItemN	Target
r_1	…	r_k	…	r_l	…	r_n	X
selected items			**filler items**		unrated items		

Random Strategy

- Take random values for filler items
    - Typical distribution of ratings is known, e.g., for the movie domain
      (Average 3.6, standard deviation around 1.1)
- Idea:
    - Generate profiles with "typical" ratings so they are considered as neighbors to many other real profiles
    - High/low ratings for target items
- Limited effect compared with more advanced models

Average Strategy

- Use the individual item's rating average for the filler items
  - Intuitively, there should be more neighbors
- Additional cost involved: find out the average rating of an item
- More effective than Random Attack in user-based CF
  - But additional knowledge is required
- Quite easy to determine average rating values per item
  - Values explicitly provided when item is displayed

Algorithm	Intent	Attack	Bots	PredShift	ΔMAE
User-user	Push	Random	25	0.499	0.002
			50	0.671	0.004
			100	0.830	0.009
		Average	25	1.032	0.006
			50	1.189	0.011
			100	1.300	0.019
	Nuke	Random	25	0.422	0.002
			50	0.589	0.004
			100	0.759	0.010
		Average	25	0.656	0.007
			50	0.815	0.014
			100	0.956	0.023
			[illegible]	[illegible]	[illegible]

Algorithm	Intent	Attack	Bots	PredShift	ΔMAE
Item-item	Push	Random	25	0.030	0.002
			50	0.053	0.002
			100	0.069	0.004
		Average	25	0.363	0.002
			50	0.426	0.004
			100	0.471	0.010
	Nuke	Random	25	-0.046	0.002
			50	-0.069	0.002
			100	-0.092	0.004
		Average	25	0.332	0.003
			50	0.354	0.006
			100	0.361	0.014

Algorithm	Intent	Attack	Bots	POA	ExpTop40
User-user	Push	Random	25	0.900	711%
			50	0.865	1190%
			100	0.816	1649%
		Average	25	0.715	1286%
			50	0.609	1674%
			100	0.519	1918%
	Nuke	Random	25	0.943	-39%
			50	0.928	-33%
			100	0.908	-32%
		Average	25	0.963	-67%
			50	0.952	-70%
			100	0.943	-75%

Algorithm	Intent	Attack	Bots	POA	ExpTop40
Item-item	Push	Random	25	1.000	150%
			50	1.000	171%
			100	1.000	229%
		Average	25	0.999	158%
			50	0.999	154%
			100	0.999	117%
	Nuke	Random	25	0.954	146%
			50	0.954	204%
			100	0.954	333%
		Average	25	0.955	-33%
			50	0.955	-54%
			100	0.954	-71%

Bandwagon Strategy

- **Exploits additional information about the community ratings**
- **Simple idea:**
  - Add profiles that contain high ratings for "blockbusters" (in the selected items); use random values for the filler items
  - Will intuitively lead to more neighbors because
    - popular items will have many ratings and
    - rating values are similar to many other user-profiles
- **Example: Injecting a profile with high rating values for the *Harry Potter* series**
- **Low-cost attack**
  - Set of top-selling items/blockbusters can be easily determined
- **Does not require additional knowledge about mean item ratings**

Segment Strategy

- Find items that are similar to target item,
  - These items probably liked by the same group of people
  - Identify subset of user community that is interested in items similar to A
  - Inject profiles that have high ratings for fantasy novels and random or low ratings for other genres
- Thus, item will be pushed within the relevant community
- For example: Push the new Harry Potter book
  - Attacker will inject profile with positive ratings for other popular fantasy books
  - Harry Potter book will be recommended to typical fantasy book reader
- Additional knowledge (e.g. genre of a book) is required

Issues to Consider

- Cost
    - How costly is it to make an attack?
    - How many profiles have to be inserted?
    - Is knowledge about the ratings matrix required?
        - usually it is not public, but estimates can be made
- Algorithm dependability
    - Is the attack designed for a particular recommendation algorithm?
- Detectability
    - How easy is it to detect the attack

Countermeasures