



Recommender Systems & Collective Intelligence

COMP47580

Dr. Michael O'Mahony

michael.omahony@ucd.ie



Overview

- Robustness & Collaborative Recommender Systems
- Malicious Attacks
- Attack Types & Models
- Evaluating System Robustness
- Detecting Attacks



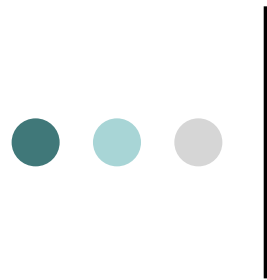
Robust Collaborative Recommendation

○ Collaborative Recommender Systems

- User-based, item-based and matrix factorisation collaborative filtering (CF).
- Rely on user-supplied ratings to make predictions and recommendations.

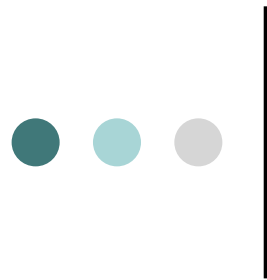
○ Robustness

- Not all users are trustworthy – potential for malicious users to “game” the recommender system.
- How easy is it for such users to bias system output – e.g. increase the probability that a certain item might be recommended, and reduce the probability that another item might be recommended?
- What techniques can be used to protect the integrity of recommender systems that are subject to such attacks?
- Robustness – a new performance criterion, as important as relevance, coverage, diversity, serendipity...



Malicious Attacks

- Motivation
 - For “fun”, sabotage or profit.
- Attack Types
 - **Product Push**: objective is to promote target items by manipulating the system to generate high predicted ratings for these items – irrespective of whether users actually like the items or not
 - **Product Nuke**: objective is to demote target items by generating low predicted ratings for them.
 - **Spoiling Attack**: attack on system integrity, objective is to degrade overall recommendation performance.



Malicious Attacks

○ Attack Methodology

- Assume system algorithm and database are secure
- Assume attackers can only interact with the system through the normal user interface
- Strategy involves the creation of multiple user profiles – *attack profiles* (APs)
- Each AP attempts to bias recommendations made for a particular target item

○ Attack Cost

- Domain knowledge: obtaining information about users, items, ratings distribution, algorithms
- Effort: ease of interacting with RS (creating attack profiles)
- Attack size: the number of APs needed

An Example...

- Suppose we want to predict the rating of item 7 for user h :
 - Users a and f have similar tastes to user h
 - Since both neighbours like item 7, the system predicts user h **likes** item 7

	Items						
	1	2	3	4	5	6	7
a	+	-		+	+		+
b	-	+		-	-	+	-
c		-	+		-	-	-
d	-	+	+	-			
e	-		-	-	-		-
f	+	-	+	+			+
g		-	+		-	-	+
h	+	-	+	+	+		?

Authentic profiles

Target profile

An Example...

- Now a malicious user inserts attack profiles (i through m)...
 - Does the system still predict that user h likes item 7??

		Items						
		1	2	3	4	5	6	7
Users	a	+	-		+	+		+
	b	-	+		-	-	+	-
	c		-	+		-	-	-
	d	-	+	+	-			
	e	-		-	-	-		-
	f	+	-	+	+			+
	g		-	+		-	-	+
	h	+	-	+	+	+		?
	i	+	-	+		-	-	-
	j	-	+	+	-			-
	k	-		-	-	-		-
	l	+	-	+	+	+		-
	m		-	+	+	-	-	-

Authentic profiles

Target profile

Attack profiles

Real-world Examples...



[Home](#) [Reviews](#) **[News](#)** [Downloads](#) [Video](#)

[Latest News](#) [CNET River](#) [Webware](#) [Crave](#) [Business Tech](#) [Green Tech](#) [Wireless](#) [Security](#) [Blogs](#) [More](#)

[Home](#) > [News](#) > [News - Digital Media](#)

December 6, 2002 5:38 PM PST

Amazon blushes over sex link gaffe

By [Stefanie Olsen](#)
Staff Writer, CNET News

[Post a comment](#)

In a incident that highlights the pitfalls of online recommendation systems, Amazon.com on Friday removed a link to a sex manual that appeared next to a listing for a spiritual guide by well-known Christian televangelist Pat Robertson.

The two titles were temporarily linked as a result of technology that tracks and displays lists of merchandise perused and purchased by Amazon visitors. Such promotions appear below the main description for products under the title, "Customers who shopped for this item also shopped for these items."

Amazon's [automated results](#) for Robertson's "Six Steps to Spiritual Revival" included a second title by Robertson as well as a book about anal sex for men.


"It seemed to us that this is a rather curious juxtaposition of the two titles," said Amazon spokeswoman Patty Smith, explaining the company's decision to remove the link.

"Amazon conducted an investigation and determined these results were not that of hundreds of customers going to the same items while they were shopping on the site," Smith said.

Amazon removed the link to the sex manual earlier Friday after being notified of the listing. A section that shows direct suggestions by other customers still contained links to the book as of late Friday.

The linking casts a spotlight on potential pitfalls of technology that flags online shopping behavior for promotional purposes.

Ad Info



tv.com
uk
UK PODCAST
TUNE IN FOR THE LATEST TV NEWS,
REVIEWS AND PREVIEWS

Most Popular

- [Moving to IPv6: Now for the hard part \(FAQ\)](#)
- [Google wants to fight smartphone battle on Web](#)
- [Hotmail launches accounts you can throw away](#)
- [Netflix rises as studios' DVD money plunges](#)

 [Recently Viewed Products](#) [My Lists](#) [My software updates](#) [Follow CNET on Twitter](#) [CNET on Facebook](#) [Like](#) 132K

[log in](#) | [join CNET](#)



Real-world Examples...

[HOME PAGE](#) [TODAY'S PAPER](#) [VIDEO](#) [MOST POPULAR](#) [TIMES TOPICS](#) [MOST RECENT](#)

[Login](#) [Register Now](#) [Help](#)

The New York Times **Technology**

[COLLECTIONS](#) > [BOOK REVIEWS](#)

ADS BY GOOGLE


Amazon Glitch Unmasks War Of Reviewers

By AMY HARMON
Published: February 14, 2004

Close observers of Amazon.com noticed something peculiar this week: the company's Canadian site had suddenly revealed the identities of thousands of people who had anonymously posted book reviews on the United States site under signatures like "a reader from New York."

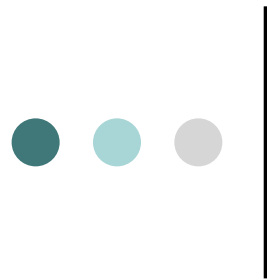
The weeklong glitch, which Amazon fixed after outraged reviewers complained, provided a rare glimpse at how writers and readers are wielding the online reviews as a tool to promote or pan a book -- when they think no one is watching.

John Rechy, author of the best-selling 1963 novel "City of Night" and winner of the PEN-USA West lifetime achievement award, is one of several prominent authors who have apparently pseudonymously written themselves five-star reviews, Amazon's highest rating. Mr. Rechy, who laughed about it when approached, sees it as a means to survival when online stars mean sales.

"That anybody is allowed to come in and anonymously trash a book to me is absurd," said Mr. Rechy, who, having been caught, freely admitted to praising his new book, "The Life and Adventures of Lyle Clemens," on Amazon under the signature "a reader from Chicago." "How to strike back? Just go in and rebut every single one of them."

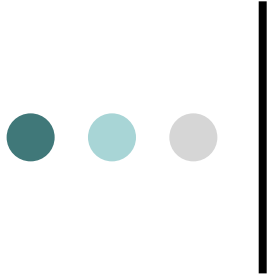
ADS BY GOOGLE

[SIGN IN TO E-MAIL](#)
[PRINT](#)
[SINGLE-PAGE](#)

Problem

- The challenge facing the attacker is to design attacks that can successfully bias the recommendations made for target items
- The challenge facing system managers is to develop techniques to provide robustness against attack
- “Arms race” between attackers and system managers
- Focus on a robustness analysis of the user-based CF algorithm



A review of user-based CF



User-based CF Prediction Algorithm

- Similarity Metric

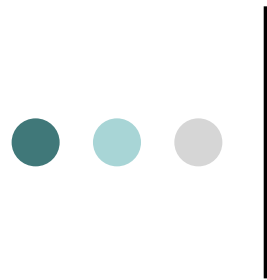
- Key to CF is the ability to compute the similarity between a pair of users based on their ratings profile.
- Examples are *Pearson correlation*, *Cosine*, *Mean Squared Difference*.

- Neighbourhood Policy

- We need a technique for constructing a neighbourhood of similar users (relative to the target user)
- Typically this is a set of neighbours based on a fixed-size (kNN) or similarity threshold, or both.

- Prediction Computation

- Given a set of neighbours, the final step is to aggregate their ratings in order to generate the predicted rating for the target user.
- Examples include the weighted average approach or *Resnick's algorithm* (deviation from mean approach).



User-based CF Recommendation Algorithm

- Key Difference
 - Instead of generating a rating prediction for a given item for the target user, the objective is to suggest a set of items to the target user.
- Get Candidate Items
 - Once the set of neighbours has been identified in the usual way, a candidate item list can then be generated by selecting those items that the neighbours have rated but that the target user has not.
- Top N Recommendation
 - The candidate items are ranked according to their likely relevance to the target user.
 - One way to do this is to generate a predicted rating for each candidate item for the target user, and then select the N items with the highest predicted rating.



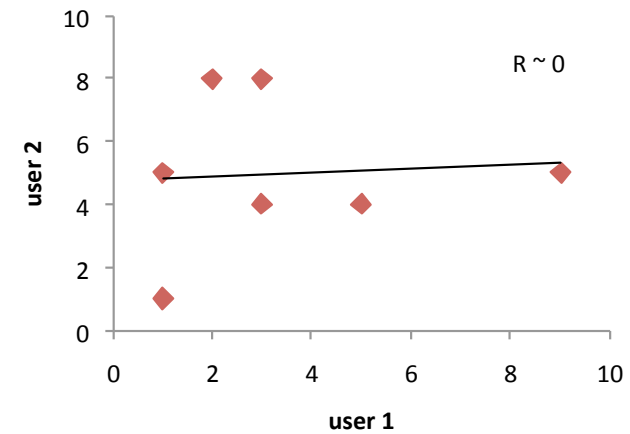
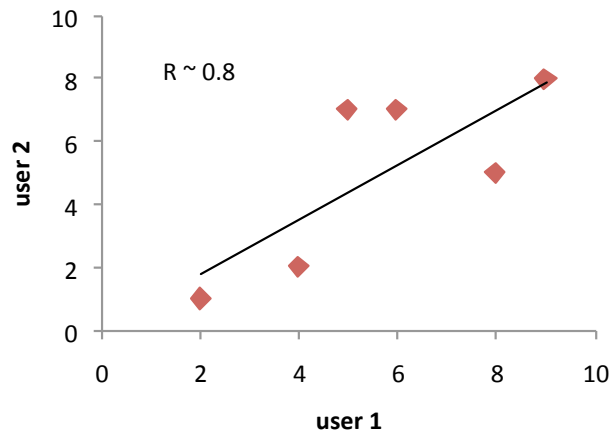
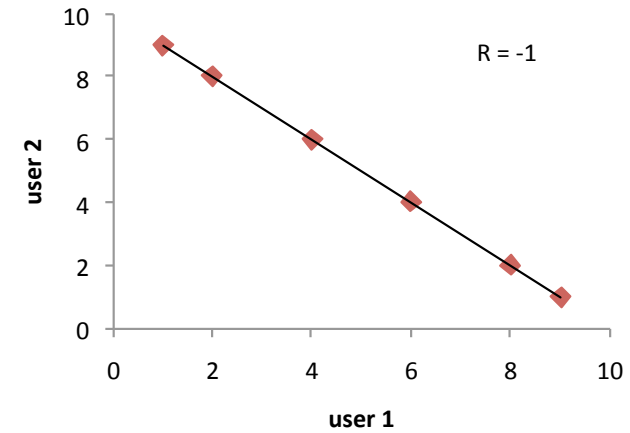
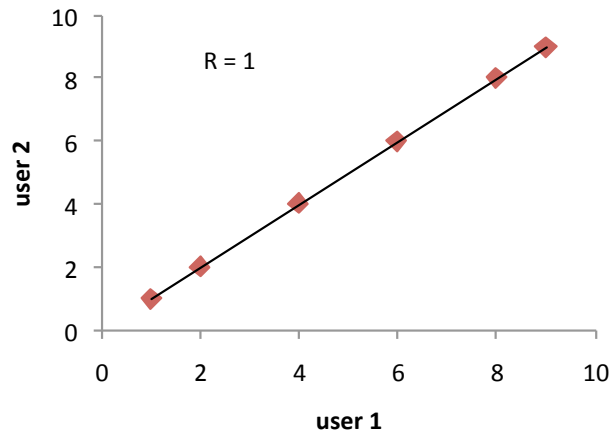
Similarity Metric

- Focus on Pearson correlation – most widely-used metric

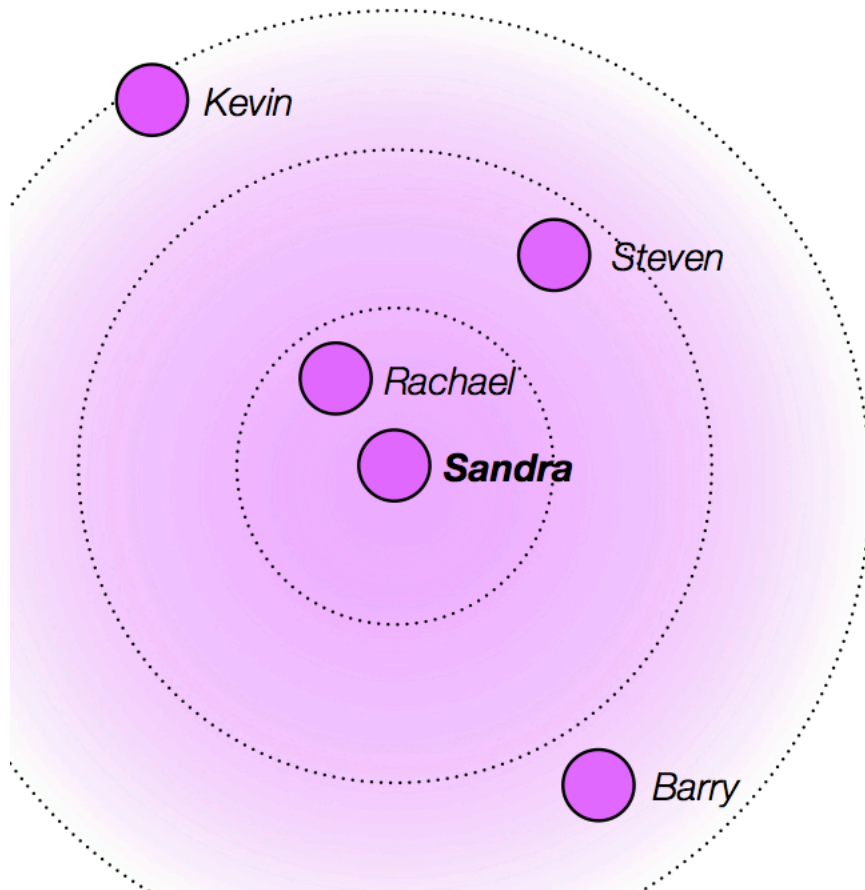
$$w_{a,i} = \frac{\sum_{j \in I_a \cap I_i} (r_{a,j} - \bar{r}_a)(r_{i,j} - \bar{r}_i)}{\sqrt{\sum_{j \in I_a \cap I_i} (r_{a,j} - \bar{r}_a)^2 \sum_{j \in I_a \cap I_i} (r_{i,j} - \bar{r}_i)^2}}$$

- Notation
 - I_a is the set of items rated by user a .
 - $r_{a,j}$ is the rating of user a for item j .
 - \bar{r}_a is the average rating of user a .
- Summations are over co-rated items only
 - Similarity is zero if there are no co-rated items.
- Results in a value of $[-1, +1]$
 - $+1$ indicates total agreement on co-rated items.
 - -1 indicates total disagreement on co-rated items.

Pearson Correlation – Examples



Neighbourhood Policy



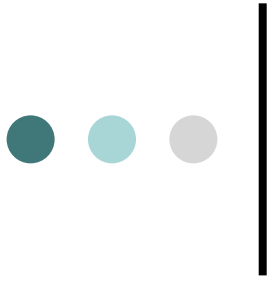
- Neighbourhood Selection Policy
 - Need to ensure that we include only high-quality (that is, very similar) neighbours from a large set of candidates.
- Fixed Size Neighbourhoods (kNN)
 - Select the top k most similar users.
- Threshold Neighbourhoods
 - Select only users exceeding a fixed similarity threshold.



Prediction Computation

- Some people are critical reviewers, rarely assigning the top rating to any item
 - Such users should receive maximum predictions on a relatively infrequent basis.
- In contrast, other users may generally give higher ratings to items
 - These users would expect to receive high or maximum predictions for liked items
- Key point: rating distributions of users are centered around different points
- **Resnick's Algorithm:**
 - Deviation from mean approach – target item is assigned a rating that is an adjusted form of the target user's average rating

$$p_{a,j} = \bar{r}_a + \frac{\sum_{i=1}^n w_{a,i} (r_{i,j} - \bar{r}_i)}{\sum_{i=1}^n |w_{a,i}|}$$



Implementing Attacks



Some Considerations

- Attackers wish to minimise *attack cost*, which is determined by:
 - Degree of knowledge required to implement the attack (user and item rating distributions, knowledge of frequently rated items, the algorithm used)
 - Attack size (the number of attack profiles created) – difficult to automate the creation of profiles (CAPTCHAs)
 - The number of items included in each profile – easy to automate
- Key point
 - To influence recommendations, attack profiles must have sufficiently high similarity to target users to be included in neighbourhoods
 - If attack profiles are not included in neighbourhoods, the attack can have no effect
- Attack detection
 - Some attacks may be highly effective but easy to detect
 - Obfuscation: reduce attack *signature* to defeat detection; may result in less effective attacks – arms race between attackers and detection strategies
 - Objective of system managers is to make the cost of successful attacks prohibitive

Attack Models

- Attack profiles – created according to a particular *attack model*

item ₁	...	item _k	item ₁	...	item _l	item ₁	...	item _m	target
r ₁	...	r _k	r ₁	...	r _l	null	...	null	r _{target}

Ratings for k selected items Ratings for l filler items Unrated items rating for pushed/nuked item

- Attack profiles consist of the following items
 - The *target item* (I_T) – assigned a rating of r_{max} (push attack) or r_{min} (nuke attack)
 - Filler items* (I_F) – randomly chosen from those available
 - Selected items* (I_S) – items that have some association with the target item
- Attack models are distinguished by the different ways in which the filler and selected items are chosen and rated
 - Ideally, attackers choose items to maximise the similarity between attack profiles and as many genuine users as possible, while at the same time making attack profiles difficult to detect.



Attack Models

- A number of attack models have been proposed for both product push and nuke attacks and for various CF algorithms
- Here focus on user-based CF and 3 product push attacks:
 - *Average Attack*
 - *Bandwagon Attack*
 - *Popular Attack*



Push Attack Models (1)

- *Average Attack*

- Selected items (I_S) – none
- Filler items (I_F) – a randomly selected set of items with each item assigned a rating distributed around the mean of all ratings assigned to that item (by genuine users)

* *High knowledge cost* – since mean ratings of filler items must be known

- *Bandwagon Attack*

- Selected items (I_S) – a small set of frequently rated items associated with the target item, each assigned a rating of r_{max}
- Filler items (I_F) – a randomly selected set of items with each item assigned a rating distributed around the overall system rating mean

* *Low knowledge cost* – since “blockbuster”, “best seller” items are typically easy to identify (IMDB , NYT best sellers list, etc.)

Push Attack Models (2)

○ Popular Attack

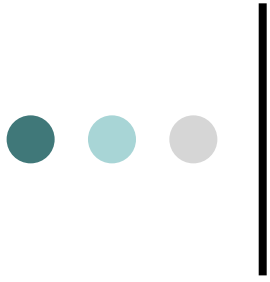
- Selected items (I_S) – a set of frequently-rated (i.e. popular) items associated with the target item, with the *more liked items* in the set assigned a rating of $r_{min} + 1$ and the *less liked items* assigned a rating of r_{min} (this approach is designed to achieve a high positive correlation between attack and genuine profiles)
- Filler items (I_F) – none

* *High knowledge cost* – since mean ratings of selected items must be known

○ Example of an *informed attack*, i.e. one that takes into account the characteristics of a particular CF algorithm:

- Assume Resnick's user-based algorithm
- Similarity metric: Pearson correlation
- Ratings assigned to selected items maximises the *contribution of attack profiles*.

$$p_{a,j} = \bar{r}_a + \frac{\sum_{i=1}^n w_{a,i} (r_{i,j} - \bar{r}_i)}{\sum_{i=1}^n |w_{a,i}|}$$



Some Results...



Methodology & Robustness Metrics

- Prediction problem: for each item j in the system
 - Create attack profiles which target item j
 - Evaluate attack over all genuine users who have rated item j by calculating the mean *prediction shift* over each of these users:

$$\Delta_{a,j} = p'_{a,j} - p_{a,j}$$

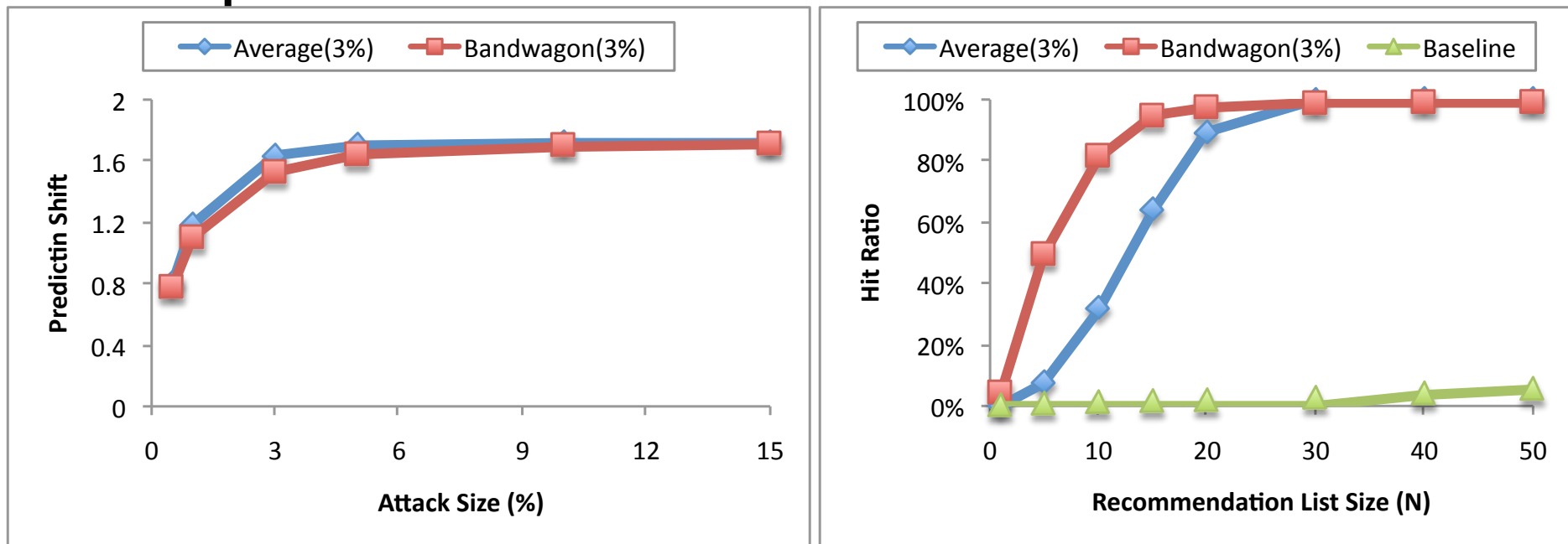
where $p_{a,j}$ and $p'_{a,j}$ are the pre-and post-attack predictions for user a , item j

- Finally, calculate mean prediction shift over all items in the system
- Top-N problem: for each item j in the system:
 - Create attack profiles which attack item j
 - Evaluate attack over all genuine users by calculating the mean *hit ratio* (HR) over each of these users:

$$\text{HR}(a, j) = 1 \text{ if } j \text{ is present in } R_N, \text{ and } 0 \text{ otherwise}$$

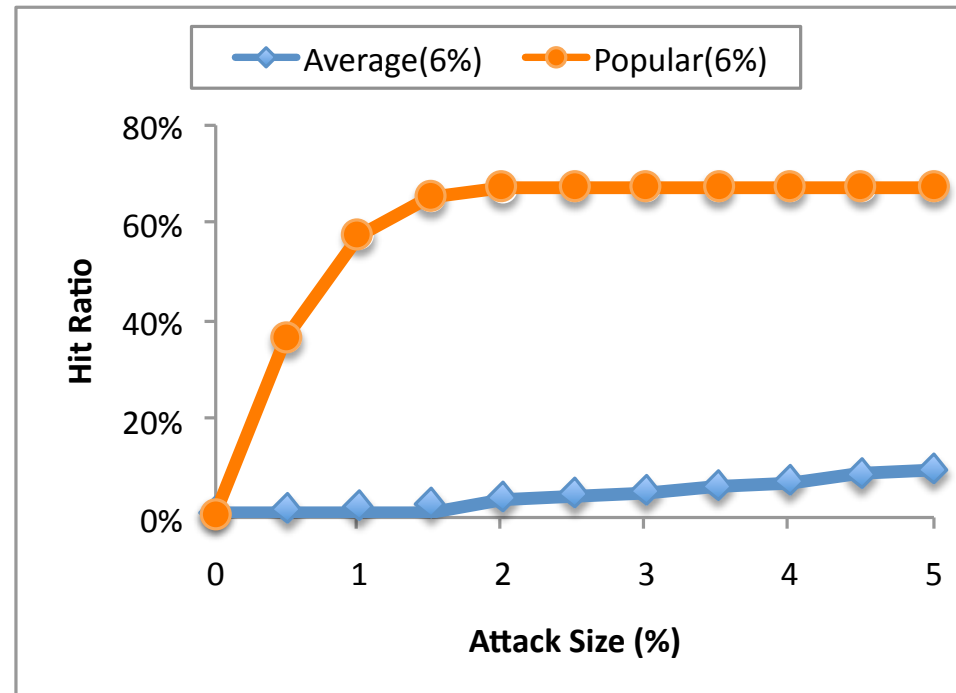
- Finally, calculate mean hit ratio over all items in the system

Push Attack Results (1)

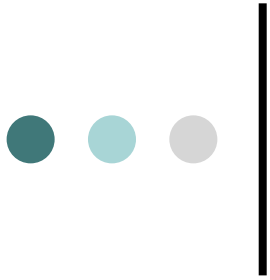


- MovieLens 100K dataset (943 users, 1683 movies, sparsity ~ 96%)
- Attack size: number of attack profiles created as a percentage of the number of genuine profiles in the system (1% attack size ~ 10 profiles).
- Average(3%): 3% filler size (~ 50 items)
- Bandwagon(3%): 1 selected item and 3% filler size (~ 51 items).
- Hit ratio results relate to a 10% attack size.

Push Attack Results (2)



- MovieLens 100K dataset.
- 1% attack size (~ 10 profiles).
- 6% filler size (~ 100 items).
- Popular (informed) attack much more effective...



Attack Detection



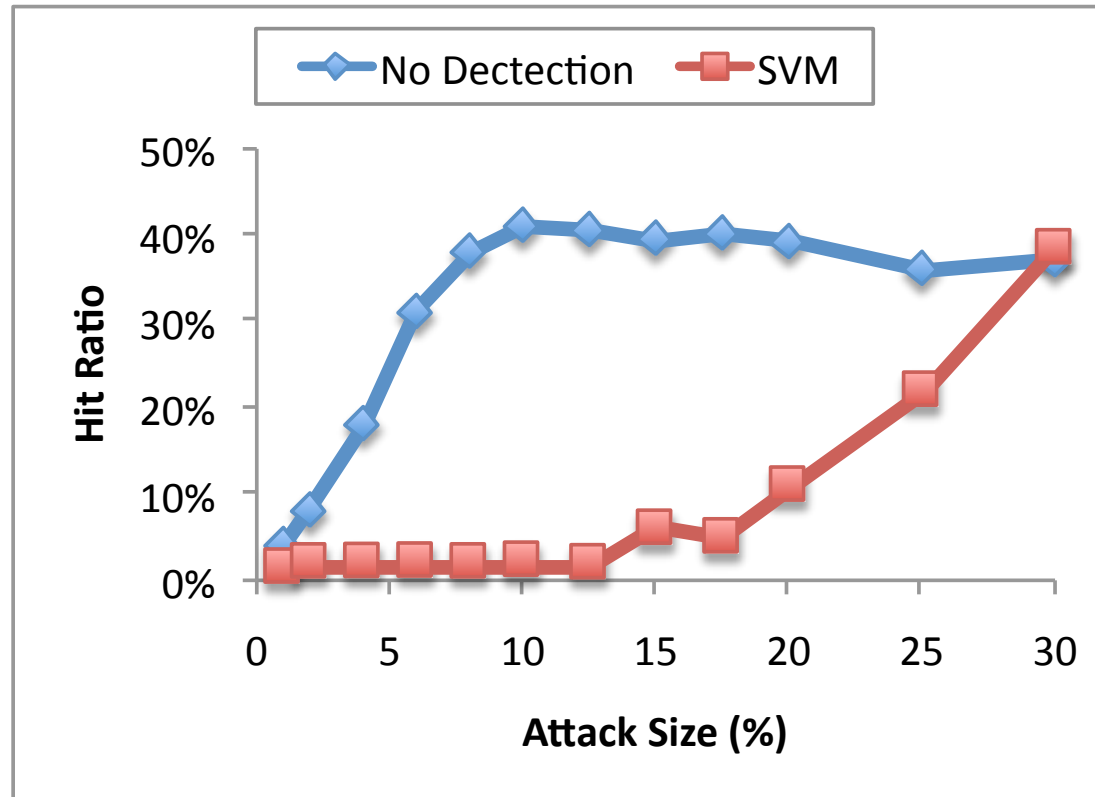
Protecting Against Attack

- We have seen that user-based CF is vulnerable to attack
- Users must have confidence that system recommendations are reliable
 - If users frequently receive recommendations for “promoted” items they are unlikely to continue to use the system...
- A number of techniques have been proposed to detect or to minimise the influence of attacks:
 - Supervised/unsupervised classification approaches
 - Trust-based approaches
 - Robust recommendation algorithms
 - Hybrid recommenders (e.g. CF and content-based)

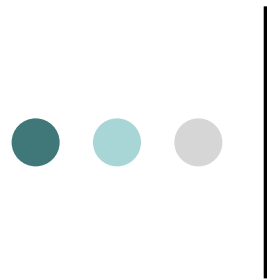
| Supervised Classification

- Assume we have a set of known genuine and attack profiles:
 - Attack profiles are created according to some attack model
- Extract features from genuine and attack profiles:
 - Idea is that the values of these features are different for genuine and attack profiles, thereby allowing for the correct detection of attack profiles
- Examples of features:
 - *Degree of similarity with top neighbours*: attack profiles have high similarity with their top neighbours since all are created according to a particular model
 - *Ratings variance*: defined as the variance of the ratings in the profile
 - *Ratings deviation*: compares, for each item in a profile, the difference between the assigned rating and the average rating (over all users) for that item
 - *Length variance*: is a measure of how much the length of a given profile varies from the average profile length in the database
- Compute feature values for all genuine and attack profiles and apply classification technique – filter profiles classified as attack profiles
- What about unknown attack models?

Supervised Classification – Results

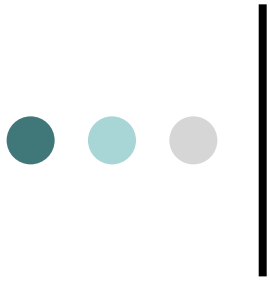


- Average push attack, 3% filler size (~ 50 items)
- Recommendation list size = 10
- 1% attack size ~ 10 profiles

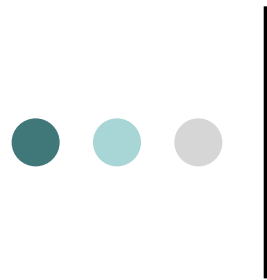


Obfuscation

- Attack models lead to profiles with distinctive features:
 - Make attack profiles more difficult to detect
 - Reduce similarity between attack profiles
 - Use a mix of different attack models
- Obfuscation strategies:
 - *Noise injection*: add Gaussian noise to subset of selected & filler items
 - *User shifting*: increment/decrement all ratings for subset of items
 - *Target shifting*: assign target item ratings from a distribution instead of r_{\max}/r_{\min}
 - Obfuscated attacks are less effective – but more difficult to detect... arms race...

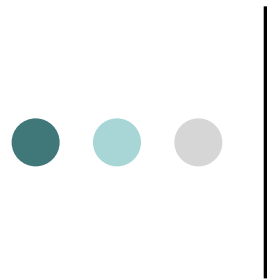


Conclusions



Conclusions

- CF relies on user supplied data to make recommendations:
 - Need users to contribute ratings
 - But not all users are trustworthy...
 - CF is vulnerable to attack...
- Attack types and models:
 - Product push and nuke attacks
 - Average attack, bandwagon attack, popular attack...
 - Attacks are characterised by the degree of knowledge (e.g. rating distributions) and effort required to implement them
 - Informed attacks (take advantage of properties of algorithms) can be particularly effective, but generally easier to detect...



Conclusions

- Detection:
 - Not possible to prevent malicious users from implementing attacks
 - Objective is to make the *cost* of successful attacks prohibitive
 - Detection techniques are effective
- Open issues:
 - Detecting attacks in *real-time*
 - Detecting *large scale, low impact* attacks.
 - Dealing with spam and malicious activity in other systems



References

- Burke, Robin and Mobasher, Bamshad and Williams, Chad and Bhaumik, Runa. *Classification features for attack detection in collaborative recommender systems*, Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '06), pp. 542-547, Philadelphia, PA, USA, 2006.
- Lam, Shyong K. and Riedl, John. *Shilling recommender systems for fun and profit*, Proceedings of the 13th International Conference on World Wide Web (WWW '04), pp. 393-402, New York, NY, USA, 2004.
- Mobasher, Bamshad and Burke, Robin and Bhaumik, Runa and Williams, Chad. *Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness*, ACM Trans. Internet Technol., 7(4), October, 2007.
- O'Donovan, John and Smyth, Barry. *Is trust robust?: An analysis of trust-based recommendation*, Proceedings of the 11th International Conference on Intelligent User Interfaces, (IUI '06), pp. 101-108, Sydney, Australia, 2006.
- O'Mahony, Michael P. and Hurley, Neil J. and Silvestre, Guénolé C. M. *Recommender Systems: Attack Types and Strategies*, Proceedings of the 20th National Conference on Artificial Intelligence (AAAI-05), pp. 334-339, Pittsburgh, PA, USA, 2005.
- Williams, Chad and Mobasher, Bamshad, and Burke, Robin and Sandvig, Jeff and Bhaumik, Runa. *Detection of obfuscated attacks in collaborative recommender systems*, Proceedings of the ECAI06 Workshop on Recommender Systems, Held at the 17th European Conference on Artificial Intelligence (ECAI'06), Riva del Garda, Italy, 2006.