Tutorial on Robustness of Recommender Systems

Neil Hurley

Complex Adaptive System Laboratory
Computer Science and Informatics
University College Dublin

Clique Strategic Research Cluster
clique.ucd.ie

October 2011

Outline

Outline

Outline

Outline

Outline

Outline

Outline

Outline

Defining the Problem

- Recommender Systems use personal information about end-users to make useful personalised recommendations.
- When ratings are provided explicitly, recommender algorithms have been designed on the assumption that the provided information is correct.
- However . . .

> *"One can have, some claim, as many electronic personas as one has time and energy to create"*
>
> –Judith Donath (1998) as quoted in Douceur (2002)

- How does the system perform if multiple identities are used to try to deliberately bias the recommender output?

Defining the Problem

- In 2002, John Douceur of Microsoft Research coined the term **Sybil Attack** to refer to an attack against identity on peer-to-peer systems in which an individual entity masquerades as multiple separate entities

> *"If the local entity has no direct physical knowledge of remote entities, it perceives them only as informational abstractions that we call identities. The system must ensure that distinct identities refer to distinct entities; otherwise, when the local entity selects a subset of identities to redundantly perform a remote operation, it can be duped into selecting a single remote entity multiple times, thereby defeating the redundancy"*

- In the same year, the first paper (O'Mahony et al. 2002) appeared on the vulnerability of Recommender Systems to malicious strategies for "*recommendation promotion*" – later dubbed **profile injection attacks**.

Defining the Problem

- **Robustness** refers to the ability of a system to operate under stressful conditions.
- While there are many possible stresses that can be applied to Recommender Systems, research on RS robustness has focused on performance when the dataset is stressed specifically when
  - the dataset is full of noisy, erroneous data;
  - typically, imagined to have been corrupted through a concerted **sybil attack**, with an aim of biasing the recommender output.

Robust RS Research

- The goal of robust recommendation is to prevent attackers from manipulating an RS through large-scale insertion of false user profiles: **a profile injection attack**

Robust RS Research

- The goal of robust recommendation is to prevent attackers from manipulating an RS through large-scale insertion of false user profiles: **a profile injection attack**
- An attack is a concerted effort to bias the results of a recommender system by the insertion of a large number of profiles using false identities or sybils.

Robust RS Research

- The goal of robust recommendation is to prevent attackers from manipulating an RS through large-scale insertion of false user profiles: **a profile injection attack**
- An attack is a concerted effort to bias the results of a recommender system by the insertion of a large number of profiles using false identities or sybils.
- Each identity is referred to as an attack profile.

Robust RS Research

- The goal of robust recommendation is to prevent attackers from manipulating an RS through large-scale insertion of false user profiles: **a profile injection attack**
- An attack is a concerted effort to bias the results of a recommender system by the insertion of a large number of profiles using false identities or sybils.
- Each identity is referred to as an attack profile.
- Research has concentrated on attacks designed to achieve a particular recommendation outcome

Robust RS Research

- The goal of robust recommendation is to prevent attackers from manipulating an RS through large-scale insertion of false user profiles: **a profile injection attack**
- An attack is a concerted effort to bias the results of a recommender system by the insertion of a large number of profiles using false identities or sybils.
- Each identity is referred to as an attack profile.
- Research has concentrated on attacks designed to achieve a particular recommendation outcome
  - A **Product Push** attack: attempt to secure positive recommendations for an item or items;

Robust RS Research

- The goal of robust recommendation is to prevent attackers from manipulating an RS through large-scale insertion of false user profiles: **a profile injection attack**
- An attack is a concerted effort to bias the results of a recommender system by the insertion of a large number of profiles using false identities or sybils.
- Each identity is referred to as an attack profile.
- Research has concentrated on attacks designed to achieve a particular recommendation outcome
  - A **Product Push** attack: attempt to secure positive recommendations for an item or items;
  - A **Product Nuke** attack: attempt to secure negative recommendations for an item or items.

Robust RS Research

- The goal of robust recommendation is to prevent attackers from manipulating an RS through large-scale insertion of false user profiles: **a profile injection attack**
- An attack is a concerted effort to bias the results of a recommender system by the insertion of a large number of profiles using false identities or sybils.
- Each identity is referred to as an attack profile.
- Research has concentrated on attacks designed to achieve a particular recommendation outcome
  - A **Product Push** attack: attempt to secure positive recommendations for an item or items;
  - A **Product Nuke** attack: attempt to secure negative recommendations for an item or items.
- We can also think of attacks that aim to simply destroy the accuracy of the system.

Robust RS Research

- We assume that the attacker has no direct access to the ratings database – manipulation achieved via the creation of false profiles only.

Robust RS Research

- We assume that the attacker has no direct access to the ratings database – manipulation achieved via the creation of false profiles only.
- We ignore system-level methods (e.g. Captchas) for preventing the generation of false identities or ratings

Robust RS Research

- We assume that the attacker has no direct access to the ratings database – manipulation achieved via the creation of false profiles only.
- We ignore system-level methods (e.g. Captchas) for preventing the generation of false identities or ratings
  - Focus is on the recommendation algorithm's ability to resist manipulation either by

Robust RS Research

- We assume that the attacker has no direct access to the ratings database – manipulation achieved via the creation of false profiles only.
- We ignore system-level methods (e.g. Captchas) for preventing the generation of false identities or ratings
  - Focus is on the recommendation algorithm's ability to resist manipulation either by
    - Identifying false profiles from their statistical properties and ignoring or lessening their impact on the generation of recommendations;

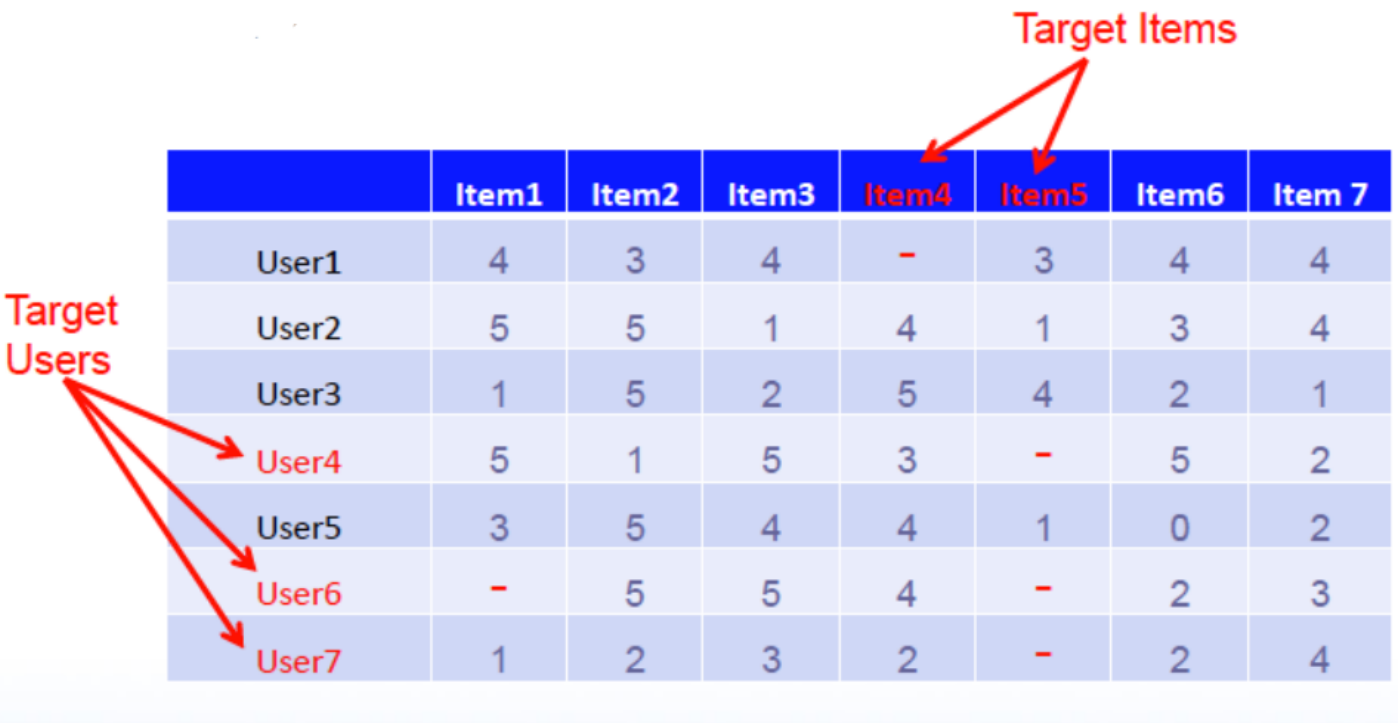
Robust RS Research

- We assume that the attacker has no direct access to the ratings database – manipulation achieved via the creation of false profiles only.
- We ignore system-level methods (e.g. Captchas) for preventing the generation of false identities or ratings
  - Focus is on the recommendation algorithm's ability to resist manipulation either by
    - Identifying false profiles from their statistical properties and ignoring or lessening their impact on the generation of recommendations; or
    - Generating recommendations in a manner that is inherently insensitive to manipulation.

Example

	Item1	Item2	Item3	Item4	Item5	Item6	Item 7
User1	4	3	4	-	3	4	4
User2	5	5	1	4	1	3	4
User3	1	5	2	5	4	2	1
User4	5	1	5	3	-	5	2
User5	3	5	4	4	1	0	2
User6	-	5	5	4	-	2	3
User7	1	2	3	2	-	2	4

Example



	Item1	Item2	Item3	Item4	Item5	Item6	Item 7
User1	4	3	4	-	3	4	4
User2	5	5	1	4	1	3	4
User3	1	5	2	5	4	2	1
User4	5	1	5	3	-	5	2
User5	3	5	4	4	1	0	2
User6	-	5	5	4	-	2	3
User7	1	2	3	2	-	2	4

Example

	Item1	Item2	Item3	Item4	Item5	Item6	Item 7
User1	4	3	4	-	3	4	4
User2	5	5	1	4	1	3	4
User3	1	5	2	5	4	2	1
User4	5	1	5	3	-	5	2
User5	3	5	4	4	1	0	2
User6	-	5	5	4	-	2	3
User7	1	2	3	2	-	2	4
Attacker1	3	4	3	4	5	3	3
Attacker2	2	5	3	4	5	4	3

Filler Items

Threats to Reputation Systems I

It is interesting to compare the scenario studied in RS research with the threats identified for reputation systems in the 2007 ENISA report (Carrara and Hogben 2007):

1. **Whitewashing attack**: reseting a poor reputation by rejoining the system with a new identity.
2. **Sybil attack** or **pseudospoofing**: the attacker uses multiple identities (sybils) in order to manipulate a reputation score.
3. **Impersonation and reputation theft**: acquiring the identity of another and stealing her reputation.
4. **Bootstrap issues and related threats**: the initial reputation of a newcomer may be particularly vulnerable to attack.

Threats to Reputation Systems II

5. **Extortion**: co-ordinated campaigns aimed at blackmail by damaging an individual's reputation for malicious motives.
6. **Denial-of-reputation**: attack designed to damage reputation and create an opportunity for blackmail in order to have the reputation cleaned.
7. **Ballot stuffing and bad mouthing**: reporting of a false reputation score; the attackers collude to give positive/negative feedback, to increase or lower a reputation.
8. **Collusion**: multiple users conspire to influence a given reputation.
9. **Repudiation of data and transaction**: denial that a transaction occurred, or denial of the existence of data for which individual is responsible.

Threats to Reputation Systems III

10. **Recommender dishonesty**: dishonest reputation scoring.
11. **Privacy threats for voters and reputation owners**: for example, anonymity improves the accuracy of votes.
12. **Social threats**: Discriminatory behaviour, herd behaviour, penalisation of innovative, controversial opinions, vocal minority effect etc.
13. **Threats to the underlying networks**: e.g. denial of service attack.
14. **Trust topology threats**: e.g.targeting most highly influential nodes.
15. **Threats to ratings**: exploiting features of metrics used by the system to calculate the aggregate reputation rating

The Recommendation Attack Game

- An attack has an associated context-dependent *cost*

The Recommendation Attack Game

- An attack has an associated context-dependent *cost*
  - real dollar cost if the insertion of a sybil rating requires the purchase of the corresponding item;

The Recommendation Attack Game

- An attack has an associated context-dependent *cost*
  - real dollar cost if the insertion of a sybil rating requires the purchase of the corresponding item;
  - time/effort cost;

The Recommendation Attack Game

- An attack has an associated context-dependent *cost*
  - real dollar cost if the insertion of a sybil rating requires the purchase of the corresponding item;
  - time/effort cost;
  - risk cost, associated with the likelihood of being detected;

The Recommendation Attack Game

- An attack has an associated context-dependent *cost*
  - real dollar cost if the insertion of a sybil rating requires the purchase of the corresponding item;
  - time/effort cost;
  - risk cost, associated with the likelihood of being detected;
- In any case, we may model the cost as proportional to

The Recommendation Attack Game

- An attack has an associated context-dependent *cost*
  - real dollar cost if the insertion of a sybil rating requires the purchase of the corresponding item;
  - time/effort cost;
  - risk cost, associated with the likelihood of being detected;
- In any case, we may model the cost as proportional to
  - The number of sybil profiles created ;

The Recommendation Attack Game

- An attack has an associated context-dependent *cost*
  - real dollar cost if the insertion of a sybil rating requires the purchase of the corresponding item;
  - time/effort cost;
  - risk cost, associated with the likelihood of being detected;
- In any case, we may model the cost as proportional to
  - The number of sybil profiles created ;
  - the total number of constituent ratings within the sybil profiles.

Impact Curve

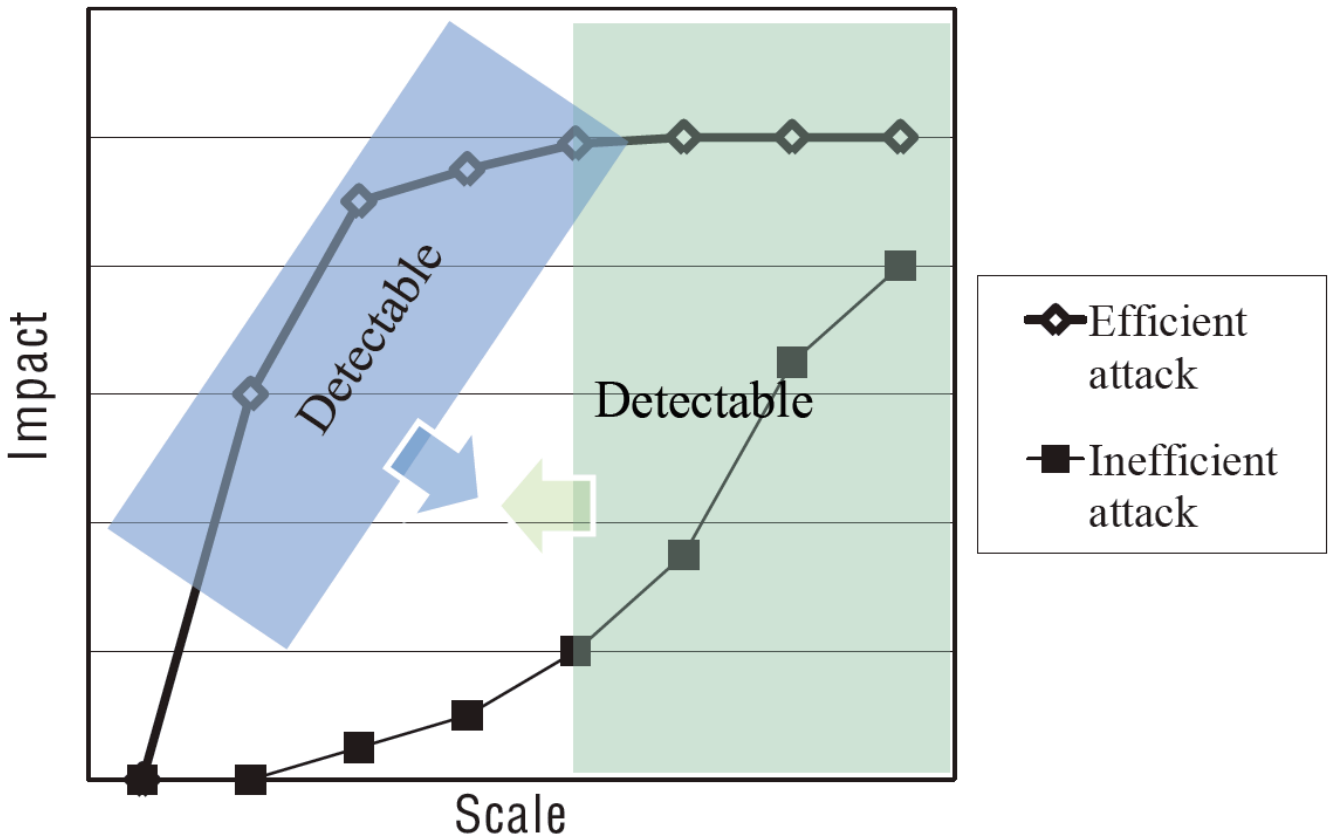


Figure: Impact curve from Burke et al. (2011)

Notation

- Consider R to be an $n \times m$ database of ratings provided by a set of n genuine users for m items in a system catalogue.

Notation

- Let $\mathcal{G}$ be the set of n genuine users, and let $\mathbf{r}_a = (r_{a,1}, \ldots, r_{a,m})^T$ represent the set of ratings provided by user a for each item.
  - $r_{a,i} \in \mathcal{L}$, some quality label, typically a numerical value over some discrete range. Write $r_{a,i} = \emptyset$ if user a has not rated item i.

Notation

- Let $\mathcal{A}$ be the set of attack profiles of size n_A = number of profiles and m_A = number of ratings. Let $\mathbf{a}_i$ denote a single attack profile $1 \leq i \leq n_A$.

Notation

- Let R' be the $(n + n_A) \times m$ database of genuine and attack profiles available to the recommendation algorithm post-attack.

Notation

- Let $c_A = c_A(n_A, m_A)$ denote the cost of mounting an attack.

Notation

- Let $c_A = c_A(n_A, m_A)$ denote the cost of mounting an attack.
- The recommendation algorithm may be represented as a function ϕ, that uses the rating database R and a user's history of previous ratings, $\mathbf{r}_u$, to make a set of *predictions* $p(u, i) = \phi(i, R, \mathbf{r}_u) \in \mathcal{L}$ on the set of items.

Notation

- Let $c_A = c_A(n_A, m_A)$ denote the cost of mounting an attack.
- The recommendation algorithm may be represented as a function ϕ, that uses the rating database R and a user's history of previous ratings, $\mathbf{r}_u$, to make a set of *predictions* $p(u, i) = \phi(i, R, \mathbf{r}_u) \in \mathcal{L}$ on the set of items.
  - More generally, $\phi(.)$ results in a *probability mass function* over the label space $\mathcal{L}$ for each predicted item.

Notation

- Let $c_A = c_A(n_A, m_A)$ denote the cost of mounting an attack.
- The recommendation algorithm may be represented as a function ϕ, that uses the rating database R and a user's history of previous ratings, $\mathbf{r}_u$, to make a set of *predictions* $p(u, i) = \phi(i, R, \mathbf{r}_u) \in \mathcal{L}$ on the set of items.
  - More generally, $\phi(.)$ results in a *probability mass function* over the label space $\mathcal{L}$ for each predicted item.
- Let $l(\phi(., R', .), \phi(., R, .))$ be a loss function representing the quality loss of the recommendation process due to an attack.

Notation

- Let $c_A = c_A(n_A, m_A)$ denote the cost of mounting an attack.
- The recommendation algorithm may be represented as a function ϕ, that uses the rating database R and a user's history of previous ratings, $\mathbf{r}_u$, to make a set of *predictions* $p(u,i) = \phi(i, R, \mathbf{r}_u) \in \mathcal{L}$ on the set of items.
  - More generally, $\phi(.)$ results in a *probability mass function* over the label space $\mathcal{L}$ for each predicted item.
- Let $l(\phi(., R', .), \phi(., R, .))$ be a loss function representing the quality loss of the recommendation process due to an attack.
  - This function may depend on the goal of the attack e.g. the overall loss in accuracy, if the attack seeks to distort general recommendation performance,

Notation

- Let $c_A = c_A(n_A, m_A)$ denote the cost of mounting an attack.
- The recommendation algorithm may be represented as a function ϕ, that uses the rating database R and a user's history of previous ratings, $\mathbf{r}_u$, to make a set of *predictions* $p(u,i) = \phi(i, R, \mathbf{r}_u) \in \mathcal{L}$ on the set of items.
  - More generally, $\phi(.)$ results in a *probability mass function* over the label space $\mathcal{L}$ for each predicted item.
- Let $l(\phi(., R', .), \phi(., R, .))$ be a loss function representing the quality loss of the recommendation process due to an attack.
  - This function may depend on the goal of the attack e.g. the overall loss in accuracy, if the attack seeks to distort general recommendation performance,
  - or the shift in ratings for some targeted set of items over some targeted set of users, in a focused attack.

The Recommender Attack Game

- Then the **manipulation game**, from the attacker's point of view is to choose the attack $\mathcal{A}^*$ that maximises the loss for a given cost bound $c_{\max}$.

Attack Goal

$$\mathcal{A}^* = \arg\max_{\{\mathcal{A}|c_A \leq c_{\max}\}} \min_{\phi} l(\phi(R'), \phi(R))$$

The Recommender Attack Game

- Then the **manipulation game**, from the attacker's point of view is to choose the attack $\mathcal{A}^*$ that maximises the loss for a given cost bound $c_{\max}$.

Attack Goal

$$\mathcal{A}^* = \arg\max_{\{\mathcal{A}|c_A \leq c_{\max}\}} \min_{\phi} l(\phi(R'), \phi(R))$$

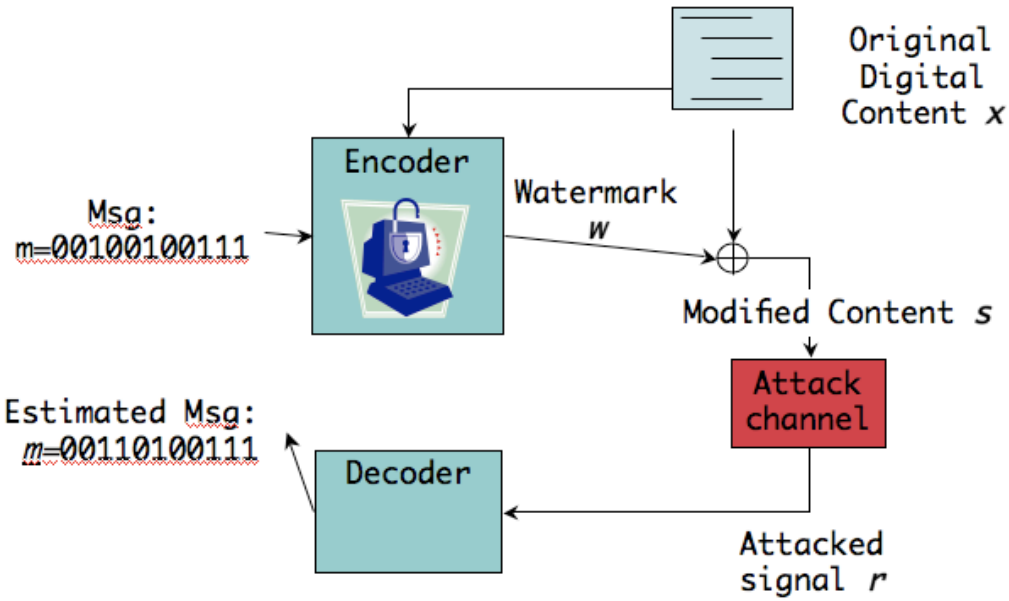
- While the system designer strives to find a recommendation algorithm that militates against attack

Defense Goal

$$\phi^* = \arg\min_{\phi} \max_{\{\mathcal{A}|c_A \leq c_{\max}\}} l(\phi(R'), \phi(R))$$

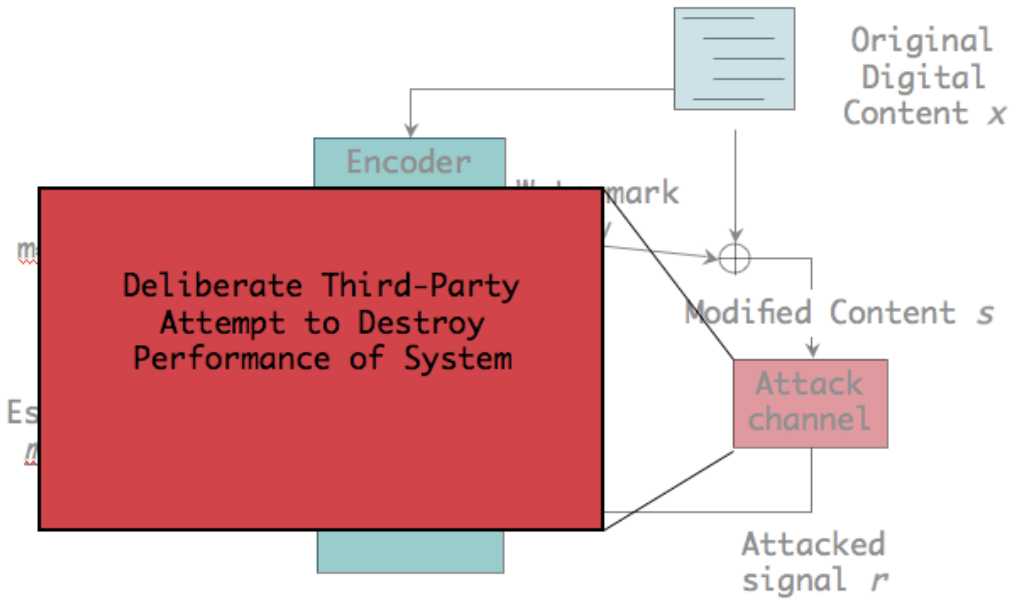
Outline

Our Original Motivation



Inspired by Work in Digital Watermarking ...

Our Original Motivation



Inspired by Work in Digital Watermarking ...

How Realistic Is this Scenario?

news.bbc.co.uk/2/hi/4741259.stm **(2001)** " *... ads for films including Hollow Man and A Knight's Tale quoted praise from a reviewer called David Manning, who was exposed as being invented ...*"

How Realistic Is this Scenario?

http://tinyurl.com/3d6d969 **(Sept. 2003)** "AuctionBytes conducted a reader survey to find out how serious these problems were. According to the survey, 39% of respondents felt that feedback retaliation was a very big problem on eBay. Nineteen percent of respondents had received retaliatory feedback within the previous 6 months, and 16% had been a victim of feedback extortion within the previous 6 months."

How Realistic Is this Scenario?

http://tinyurl.com/n3d5lp **(June 2009)** "Elsevier officials said Monday that it was a mistake for the publishing giant's marketing division to offer $25 Amazon gift cards to anyone who would give a new textbook five stars in a review posted on Amazon or Barnes & Noble. While those popular Web sites' customer reviews have long been known to be something less than scientific, and prone to manipulation if an author has friends write on behalf of a new work, the idea that a major academic publisher would attempt to pay for good reviews angered some professors who received the e-mail pitch.."

How Realistic Is this Scenario?

http://tinyurl.com/cfmqce **(2009)** Paul Lamere cites an example of "precision hacking" of a Time Poll.

How Realistic Is this Scenario?

http://tinyurl.com/mupy7d **(2009)** Hotel review manipulation

How Realistic Is this Scenario?

Lang et al. (2010) Social Manipulation in Buzznet *"...Hey, I know you don't know me, but could you do me a huge fav and vote for me in this contest? All you have to do is buzz me..."*

How Realistic Is this Scenario?

- All examples of types of manipulation attacks on recommender systems.
- No hard evidence of *automated* shilling attacks by sybil insertion bots.

Outline

Simple Measures of Attack Impact

Average Prediction Shift

The change in an item's predicted rating before and after attack, averaged over all predictions or over predictions that are targetted by the attack.

$$p_{\text{shift}}(u, i) = \phi(i, \mathrm{R}', \mathbf{r}_u) - \phi(i, \mathrm{R}, \mathbf{r}_u)$$

Simple Measures of Attack Impact

Average Prediction Shift

The change in an item's predicted rating before and after attack, averaged over all predictions or over predictions that are targetted by the attack.

$$p_{\text{shift}}(u,i) = \phi(i, \mathrm{R}', \mathbf{r}_u) - \phi(i, \mathrm{R}, \mathbf{r}_u)$$

Average Hit Ratio

The average likelihood over tested users that a top-N recommendation will recommend an item that is the target of an attack. For each such item i and each tested user, u, in a test set of t users, let $h(u,i) = 1$ if $i \in R_u$, the recommended set. Then, $H(i) = \frac{1}{t}\sum_{u \in U} h(u,i)$

Other Measures of Attack Impact

- Considering $\phi(i, \mathrm{R}, \mathbf{r}_u)$ as a pmf over $\mathcal{L}$, attack impact can be measured in terms of the change in this distribution as a result of the attack.
- For instance (Yan and Roy 2009), measure impact in terms of the average *Kullback-Liebler* distance between $\phi(i, \mathrm{R}, \mathbf{r}_u)$ and $\phi(i, \mathrm{R}', \mathbf{r}_u)$ over the set of inspected items.
- Resnick and Sami (2007) propose *loss functions* $L(l, q)$ where $l \in \mathcal{L}$, $q = \phi(i, \mathrm{R}, \mathbf{r}_u)$ where the true label of item i is l

Other Measures of Attack Impact

- Considering $\phi(i, \mathrm{R}, \mathbf{r}_u)$ as a pmf over $\mathcal{L}$, attack impact can be measured in terms of the change in this distribution as a result of the attack.
- For instance (Yan and Roy 2009), measure impact in terms of the average *Kullback-Liebler* distance between $\phi(i, \mathrm{R}, \mathbf{r}_u)$ and $\phi(i, \mathrm{R}', \mathbf{r}_u)$ over the set of inspected items.
- Resnick and Sami (2007) propose *loss functions* $L(l, q)$ where $l \in \mathcal{L}$, $q = \phi(i, \mathrm{R}, \mathbf{r}_u)$ where the true label of item i is l
  - e.g. the *quadratic loss* over a two rating scale $[HI, LO]$, with q the probability of HI is given by

$$L(HI, q) = (1 - q)^2; \qquad L(LO, q) = q^2$$

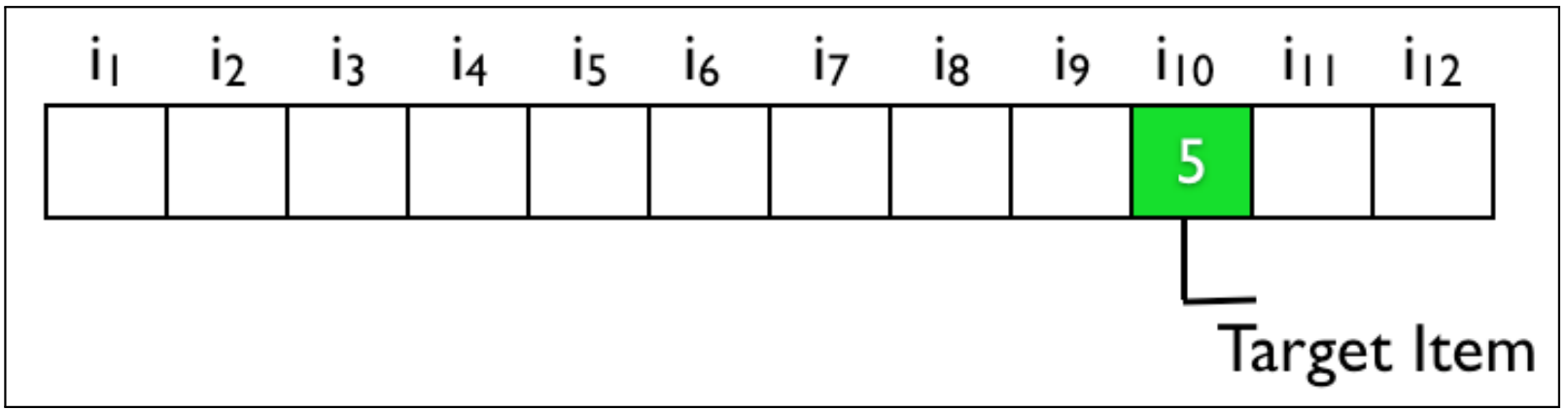
Outline

Forming Attack Profiles

- (Mobasher et al. 2007) introduce the following notation for the components of an attack profile:

Forming Attack Profiles

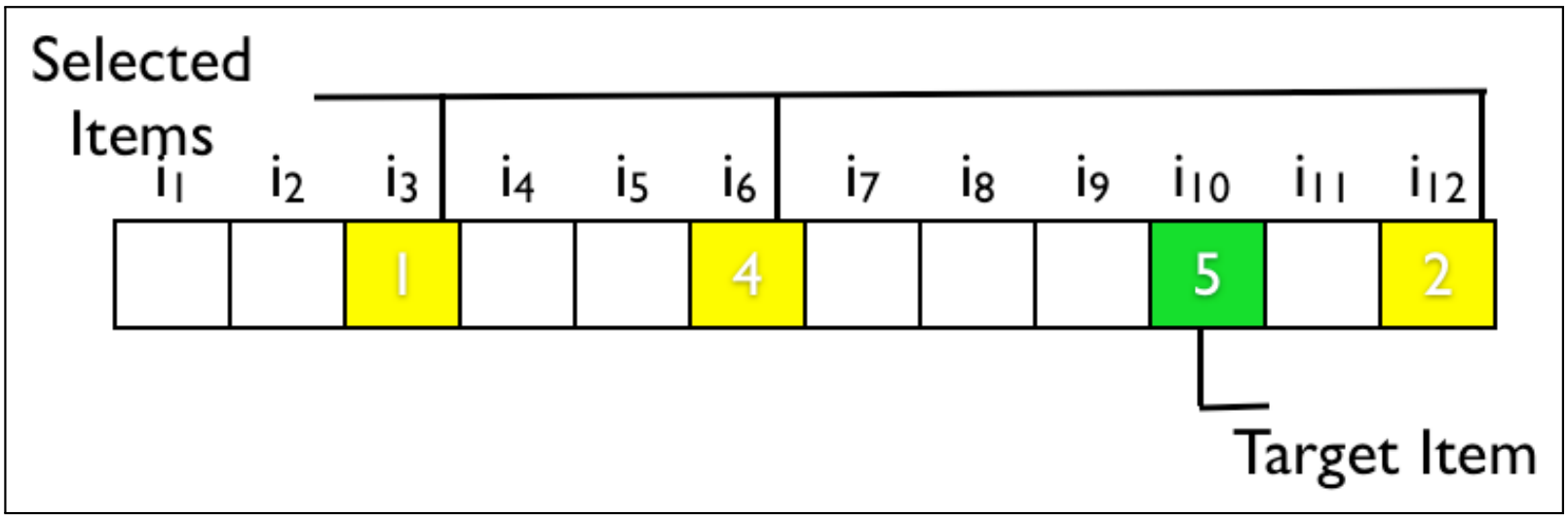
- (Mobasher et al. 2007) introduce the following notation for the components of an attack profile:



- I_T, the target item(s) receive typically the maximum (resp. minimum) rating for a push (resp. nuke) attack.

Forming Attack Profiles

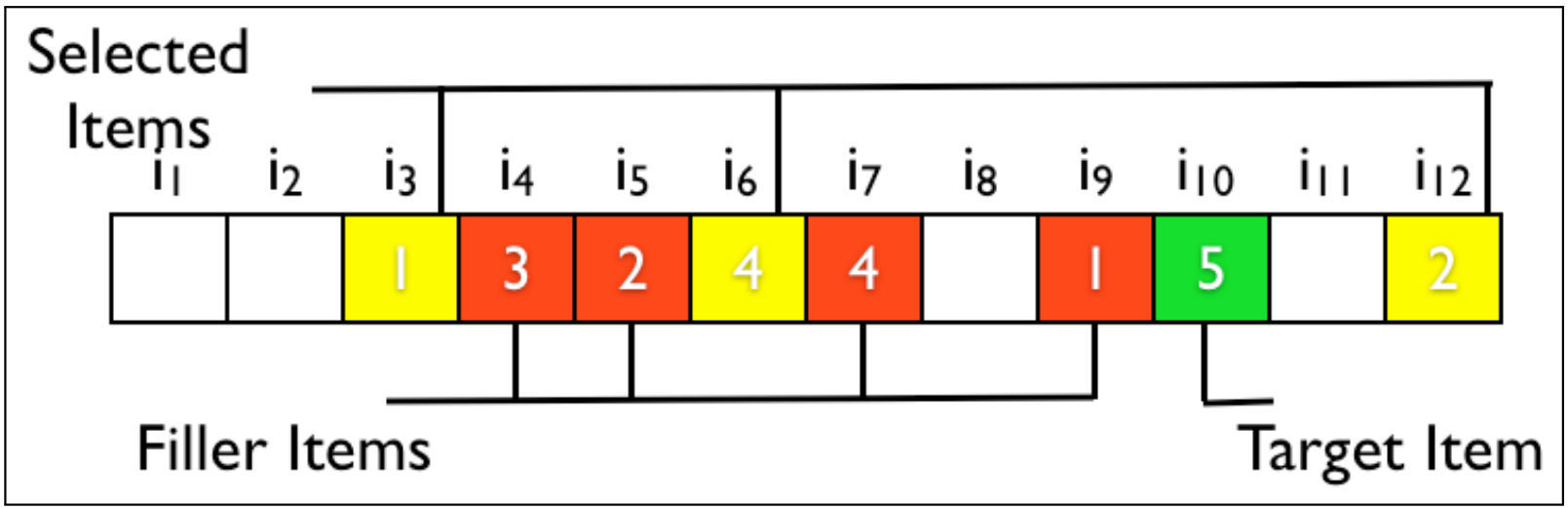
- (Mobasher et al. 2007) introduce the following notation for the components of an attack profile:



- I_S, the selected item(s) are chosen and rated in a manner to support the attack.

Forming Attack Profiles

- (Mobasher et al. 2007) introduce the following notation for the components of an attack profile:



- I_F, the filler items(s) fill out the remainder of the ratings in the attack profile.

Forming Attack Profiles

- The goal of attack profile creation must be to
  - Effectively support the purpose of the attack – selection of target rating and I_S;

Forming Attack Profiles

- The goal of attack profile creation must be to
  - Effectively support the purpose of the attack – selection of target rating and I_S;
  - But, remain unobtrusive so as to avoid detection and filtering – selection of the filler items I_F.

Forming Attack Profiles

- The goal of attack profile creation must be to
  - Effectively support the purpose of the attack – selection of target rating and I_S;
  - But, remain unobtrusive so as to avoid detection and filtering – selection of the filler items I_F.
- Must also consider what information is available to the attacker

Forming Attack Profiles

- The goal of attack profile creation must be to
  - Effectively support the purpose of the attack – selection of target rating and I_S;
  - But, remain unobtrusive so as to avoid detection and filtering – selection of the filler items I_F.
- Must also consider what information is available to the attacker
  - Typically assume some knowledge of the statistics of the rating database is available;

Forming Attack Profiles

- The goal of attack profile creation must be to
  - Effectively support the purpose of the attack – selection of target rating and I_S;
  - But, remain unobtrusive so as to avoid detection and filtering – selection of the filler items I_F.
- Must also consider what information is available to the attacker
  - Typically assume some knowledge of the statistics of the rating database is available;
  - May also have knowledge of the recommendation algorithm – an *informed* attack.

Forming Attack Profiles

- The goal of attack profile creation must be to
  - Effectively support the purpose of the attack – selection of target rating and I_S;
  - But, remain unobtrusive so as to avoid detection and filtering – selection of the filler items I_F.
- Must also consider what information is available to the attacker
  - Typically assume some knowledge of the statistics of the rating database is available;
  - May also have knowledge of the recommendation algorithm – an *informed* attack.
- Note **Kerkchoff's principal** – avoid "security through obscurity".

Outline

User-based kNN Attack

- The first CF profile insertion attack (O'Mahony et al. 2002), was an informed attack that exploited a particular weakness in the basic version of Resnick's user-based kNN algorithm.

User-based kNN Attack

- The first CF profile insertion attack (O'Mahony et al. 2002), was an informed attack that exploited a particular weakness in the basic version of Resnick's user-based kNN algorithm.
- User-based CF predicts a rating $p_{a,j}$ for item j, user a as follows:

User-based kNN Attack

- The first CF profile insertion attack (O'Mahony et al. 2002), was an informed attack that exploited a particular weakness in the basic version of Resnick's user-based kNN algorithm.

- User-based CF predicts a rating $p_{a,j}$ for item j, user a as follows:
  - Form a neighbourhood by picking the top-k most similar users to a
  - **Pearson Correlation**

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

  - Make a prediction by taking a weighted average of neighbours ratings using: $p_{a,j} = \bar{r}_a + \kappa \sum_{i=1}^{n} w_{a,i}(r_{i,j} - \bar{r}_i)$ where κ is a normalising factor

User-based kNN Attack

- Correlation calculated over the items which the target user and attack profile have in common. A small intersection set can lead to high correlations.

User-based kNN Attack

- Correlation calculated over the items which the target user and attack profile have in common. A small intersection set can lead to high correlations.
- Therefore a small profile of popular items will have a large correlation with users who have rated these items

User-based kNN Attack

- Correlation calculated over the items which the target user and attack profile have in common. A small intersection set can lead to high correlations.
- Therefore a small profile of popular items will have a large correlation with users who have rated these items
- Implies a low-cost, effective attack on a large proportion of the userbase.

User-based kNN Attack

- Correlation calculated over the items which the target user and attack profile have in common. A small intersection set can lead to high correlations.
- Therefore a small profile of popular items will have a large correlation with users who have rated these items
- Implies a low-cost, effective attack on a large proportion of the userbase.
- However, not at all unobtrusive.

Other Attacks Strategies

Many other attack variants proposed by (Lam and Riedl 2004) and (Mobasher et al. 2007). Assuming a push attack – a target item is given the maximum rating and the attack profile is filled out as follows:

Random Attack Randomly chosen filler items get randomly drawn rating values.

Other Attacks Strategies

Many other attack variants proposed by (Lam and Riedl 2004) and (Mobasher et al. 2007). Assuming a push attack – a target item is given the maximum rating and the attack profile is filled out as follows:

Average Attack Randomly chosen filler items. Ratings drawn from normal distribution with item means set to those of rating database.

Other Attacks Strategies

Many other attack variants proposed by (Lam and Riedl 2004) and (Mobasher et al. 2007). Assuming a push attack – a target item is given the maximum rating and the attack profile is filled out as follows:

Probe Attack Filler items filled by starting with a set of seed ratings, querying the recommender system to fill the ratings of the remaining items.

Other Attacks Strategies

Many other attack variants proposed by (Lam and Riedl 2004) and (Mobasher et al. 2007). Assuming a push attack – a target item is given the maximum rating and the attack profile is filled out as follows:

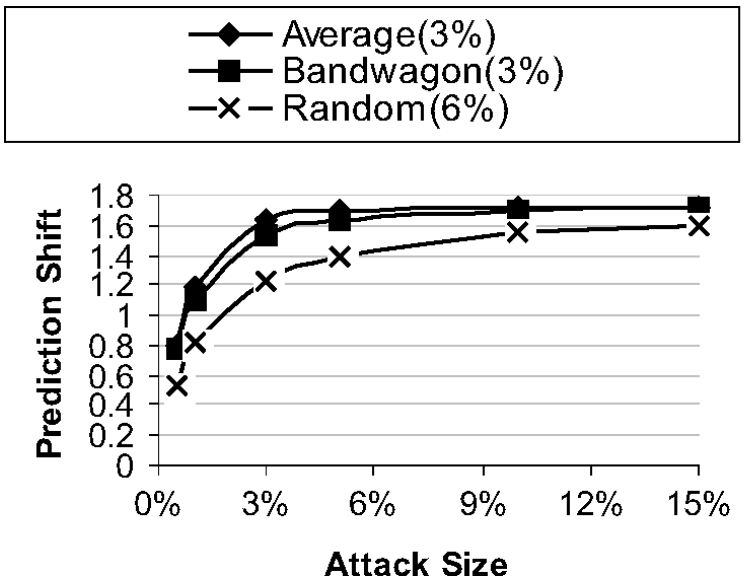
Segment Attack Identifying popular items in a particular user segment, these are given maximum rating and remaining filler items are given minimum rating.

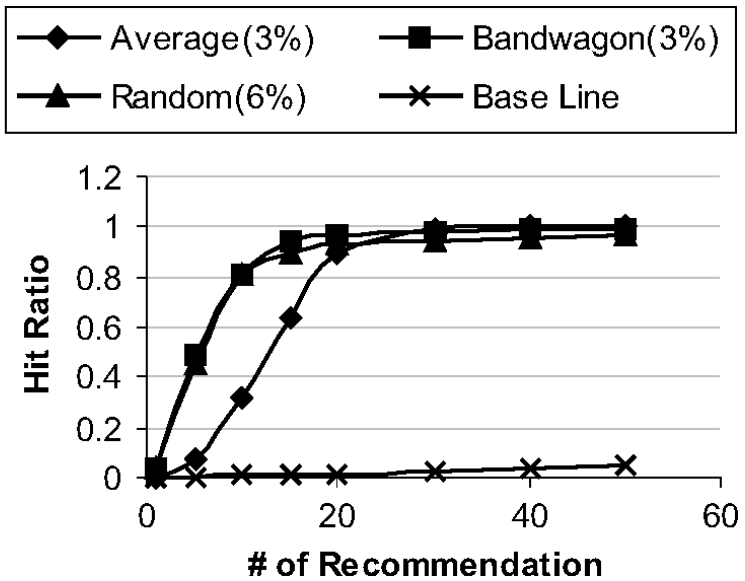
Other Attacks Strategies

Many other attack variants proposed by (Lam and Riedl 2004) and (Mobasher et al. 2007). Assuming a push attack – a target item is given the maximum rating and the attack profile is filled out as follows:

Bandwagon Attack A set of popular items is given the maximum rating, while remaining filler items are given random ratings.

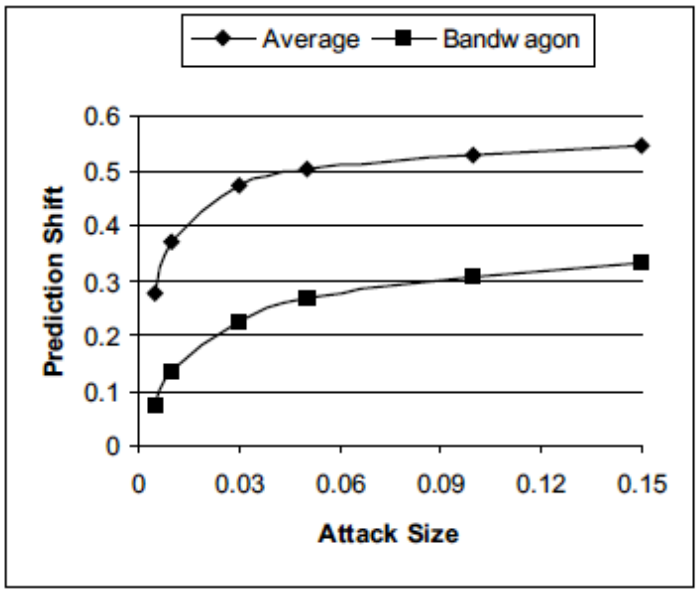
Evaluation User-based kNN

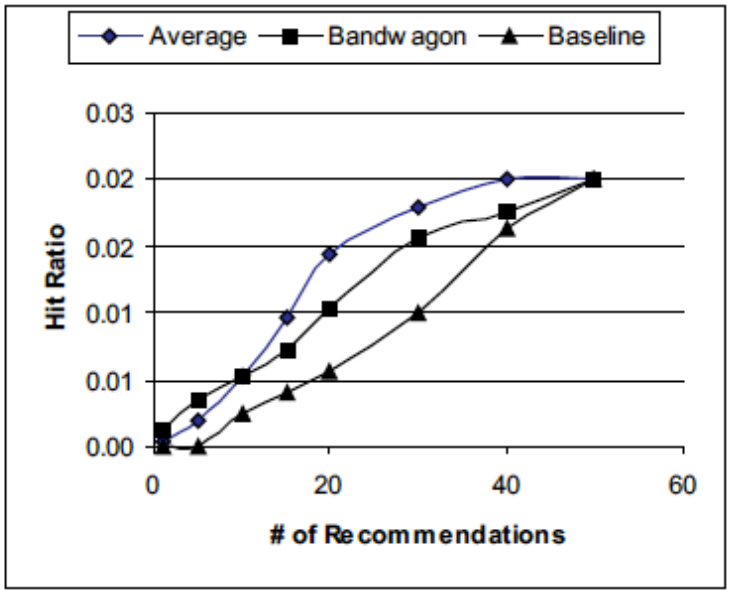




- Results from (Mobasher et al. 2007) on the Movielens 100K dataset. User-based kNN algorithm with $k = 20$. All users who have rated at least 20 items.
- **Attack Size** given as a percentage of the total number of users in the dataset on the x-axis. Results shown for different **filler sizes**, written as a percentage of the total number of items.
- Bandwagon attack uses a single popular item and 3% filler size.
- Baseline refers to hit-ratio pre-attack

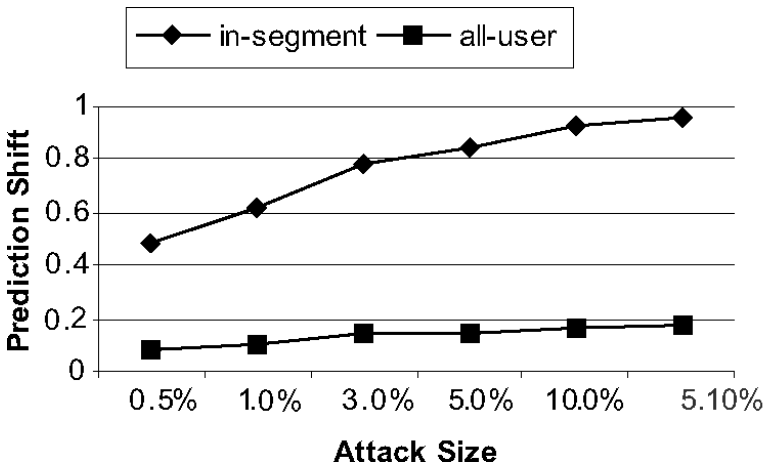
Evaluation Item-based kNN

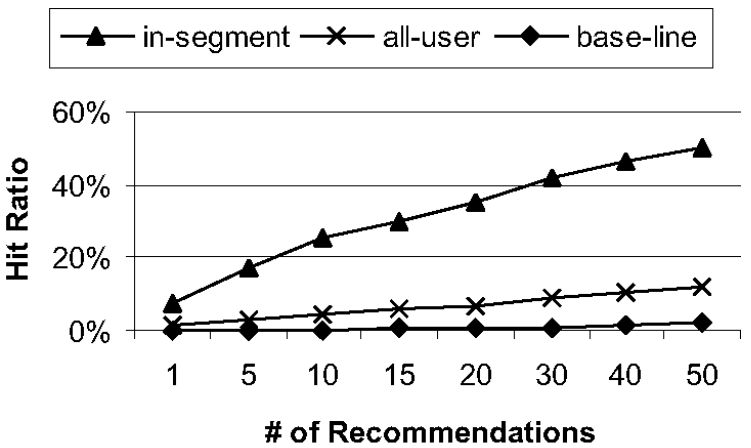




- Results from (Mobasher et al. 2007) on the Movielens 100K dataset. Item-based kNN algorithm with $k = 20$.

Evaluation Item-based kNN





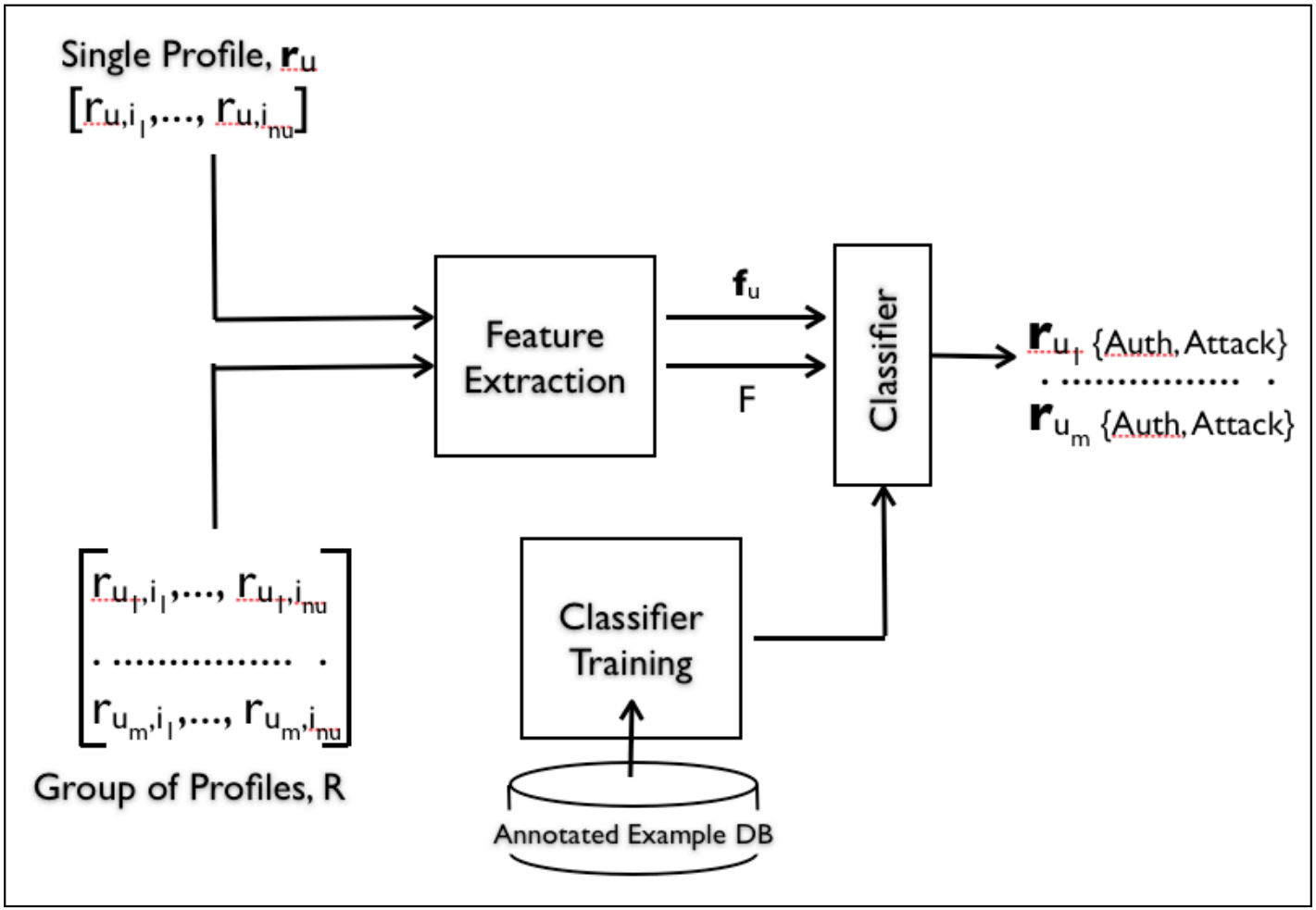
- Results from (Mobasher et al. 2007) on the Movielens 100K dataset. Item-based kNN algorithm with $k = 20$.
- Prediction shift and hit ratio results for the Horror Movie Segment.

Outline

Profile Injection Attacks

- Two well-studied attacks:
  - Construct spam profile with max (or min) rating for targeted item.
  - Choose a set of filler items at random
  - **Random Attack** – insert ratings in filler items according to $\mathcal{N}(\mu, \sigma)$
  - **Average Attack** – insert ratings in filler item i according to $\mathcal{N}(\mu_i, \sigma_i)$
- For evaluations, genuine profiles are drawn from 1,000,000 rating Movielens dataset.

Detection Flow



Outline

PCA Detector (Mehta et al. 2007)

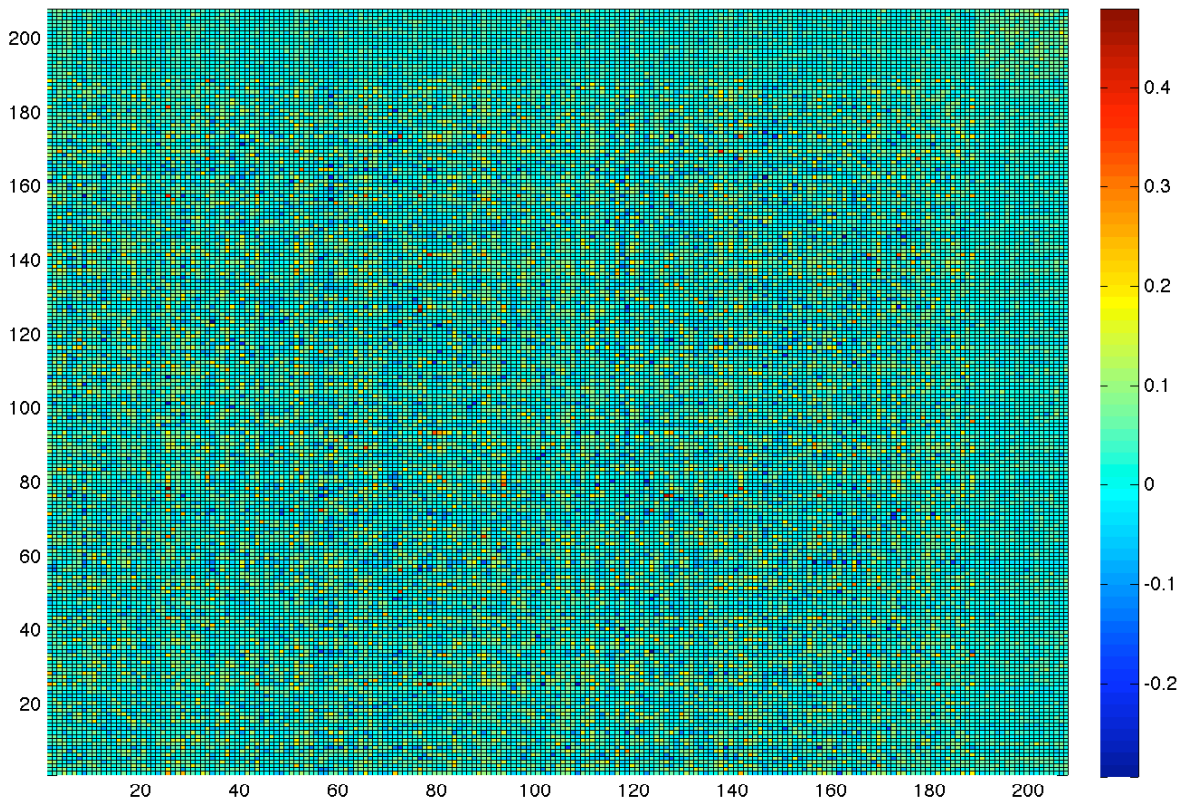
- A method of identifying and removing a cluster of highly-correlated attack profiles.
  - A cluster C defined by an indicator vector $\mathbf{x}$ such that $x_i = 1$ if user $u_i \in C$ and $x_i = 0$ otherwise.
  - S a profile similarity matrix, with eigenvectors/values $\mathbf{e}_i$, λ_i
  - Quadratic form

$$\mathbf{x}^T \mathrm{S} \mathbf{x} = \sum_{i \in C, j \in C} S(i,j) = \sum_{i=1}^{m} (\mathbf{x}.\mathbf{e_i})^2 \lambda_i \, .$$

- Find $\mathbf{x}$ that correlates most with first few eigenvectors of S.

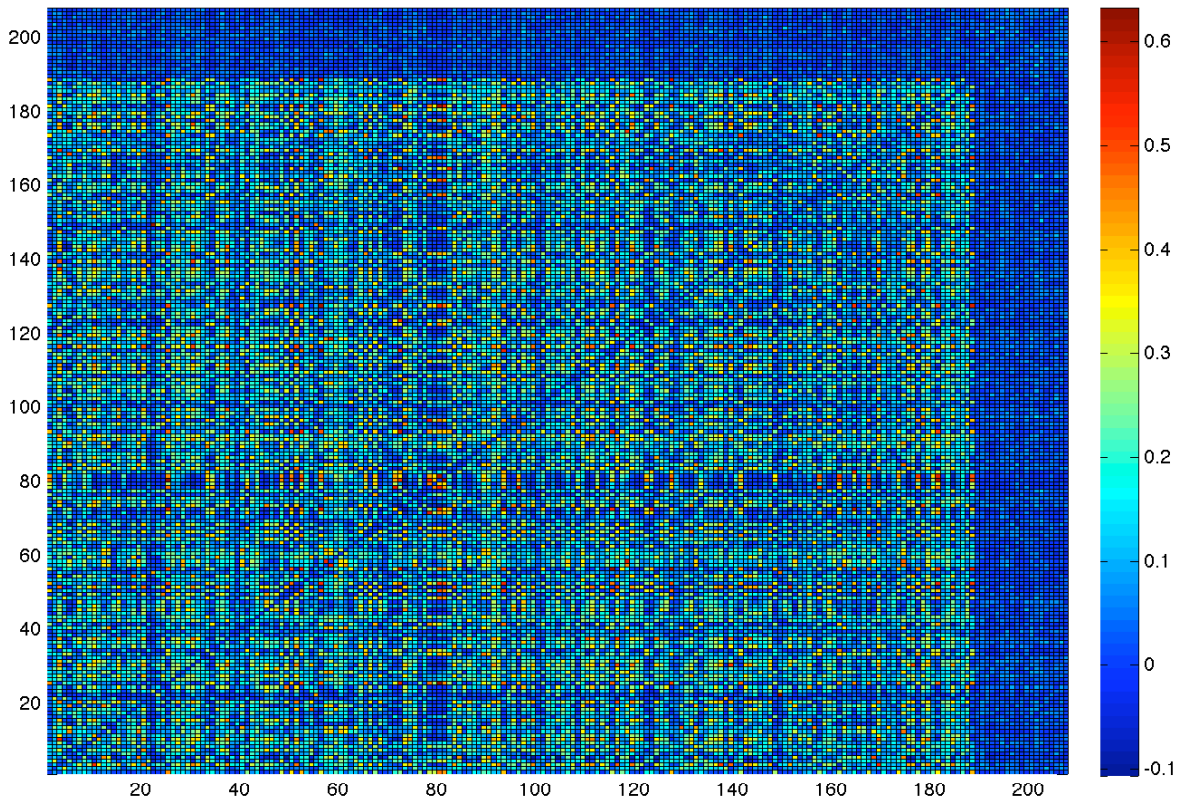
PCA Detector (Mehta et al. 2007)

- Success of PCA depends on how S is calculated.
- $\mathrm{S} = \mathrm{Z}_0$, covariance of profiles *ignoring* missing values.



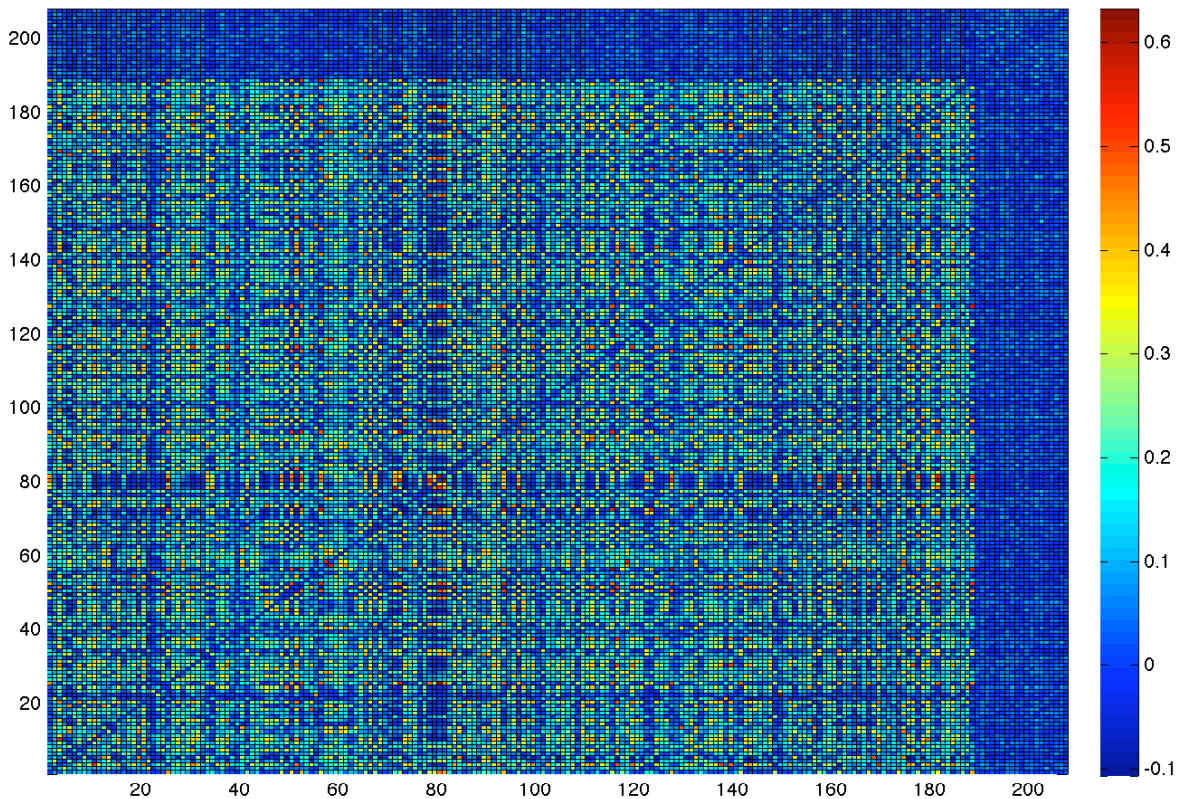
PCA Detector (Mehta et al. 2007)

- Success of PCA depends on how S is calculated.
- $\mathrm{S} = \mathrm{Z}_1$, covariance of profiles treating missing values as 0.



PCA Detector (Mehta et al. 2007)

- Success of PCA depends on how S is calculated.
- Small Overlap $\longrightarrow$ low genuine/attack covariance.



Outline

Neyman-Pearson Statistical Detection

- $\mathbf{y} \in \mathcal{Y}$ be an observation i.e. a user profile over all possible profiles $\mathcal{Y}$
- Two hypotheses
  - H_0 – that $\mathbf{y}$ is a genuine profile, associated pdf $f_{\mathbf{Y}|H_1}(\mathbf{y})$
  - H_1 – that $\mathbf{y}$ is an attack profile, associated pdf $f_{\mathbf{Y}|H_0}(\mathbf{y})$
- N-P criterion sets a bound α on *false alarm probability* p_f and maximises the *good detection probability* p_D

$$\psi^*(\mathbf{y}) = \begin{cases} H_1 & \text{if } l(\mathbf{y}) > \eta \\ H_0 & \text{if } l(\mathbf{y}) < \eta \end{cases} \quad \text{where } l(\mathbf{y}) = \frac{f_{\mathbf{Y}|H_1}(\mathbf{y})}{f_{\mathbf{Y}|H_0}(\mathbf{y})}$$

(likelihood ratio)

Modelling Attack Profiles

- As attacks follow well-defined construction, easy to construct model:

$$\prod_{i=1}^{m} \Pr[Y_i = y_i]$$

$$= \prod_{i=1}^{m} \Pr[Y_i = \phi]^{\theta_i} \left(\Pr[Y_i = y_i | Y_i \neq \phi] \Pr[Y_i \neq \phi]\right)^{1-\theta_i}$$

Which items are rated?

Modelling Attack Profiles

- As attacks follow well-defined construction, easy to construct model:

$$\prod_{i=1}^{m} \Pr[Y_i = y_i]$$

$$= \prod_{i=1}^{m} \Pr[Y_i = \phi]^{\theta_i} \left(\Pr[Y_i = y_i | Y_i \neq \phi] \Pr[Y_i \neq \phi]\right)^{1-\theta_i}$$

What ratings used?

Modelling Attack Profiles

- As attacks follow well-defined construction, easy to construct model:

$$\prod_{i=1}^{m} \Pr[Y_i = y_i]$$

$$= \prod_{i=1}^{m} \Pr[Y_i = \phi]^{\theta_i} \left(\mathcal{Q}\left(\frac{y_i - \frac{1}{2} - \mu_i}{\sigma_i} \right) - \mathcal{Q}\left(\frac{y_i + \frac{1}{2} - \mu_i}{\sigma_i} \right) \right)^{1-\theta}$$

$$\mathcal{Q}(.) = \text{ Gaussian } \mathcal{Q}\text{-function}$$

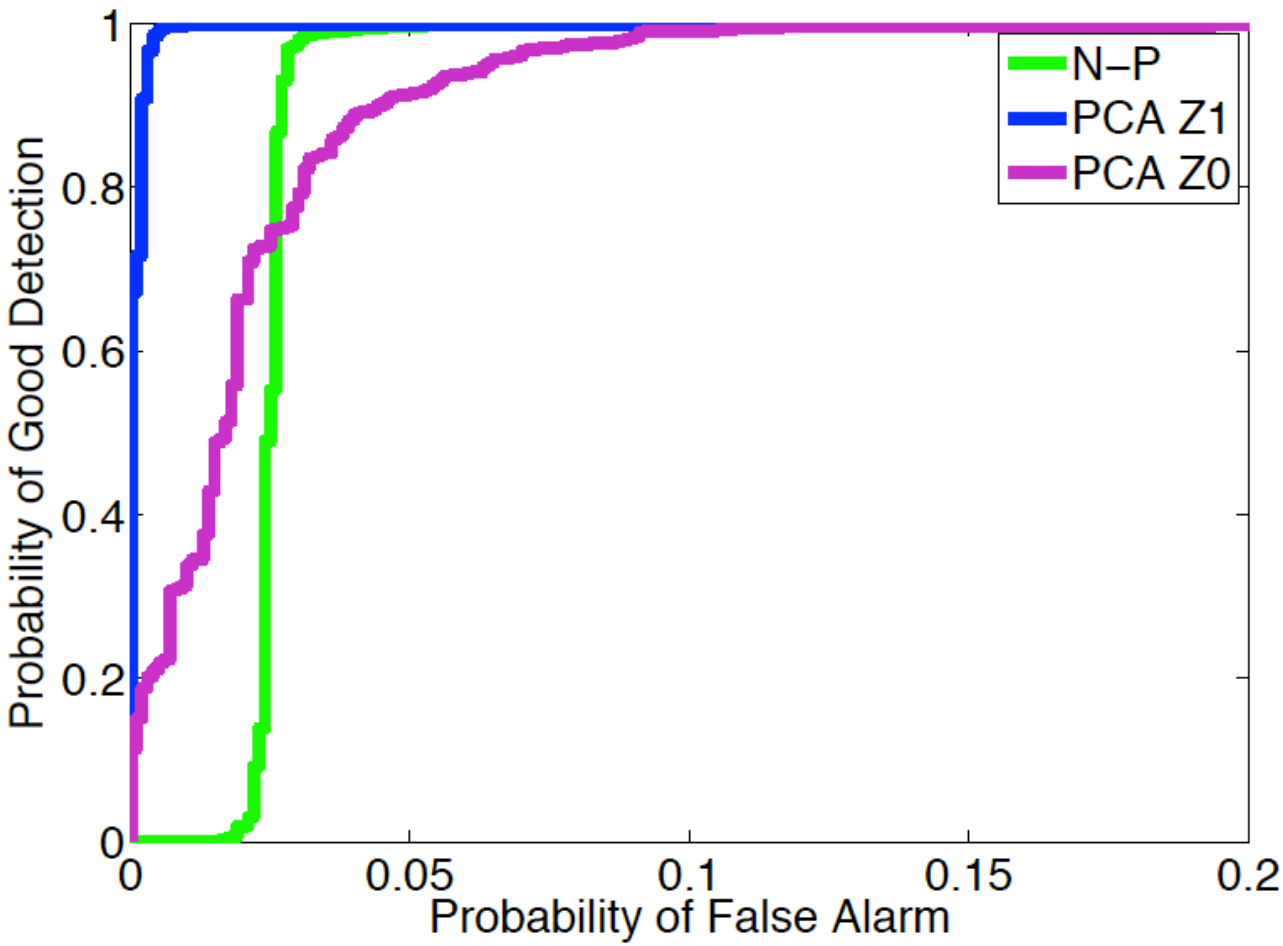
Modelling Genuine Profiles

- Simple model – identical to attack model except that probability of selecting a filler item is estimated as $\hat{p}_i \triangleq \Pr[Y_i = \phi]$ from a dataset of genuine profiles.

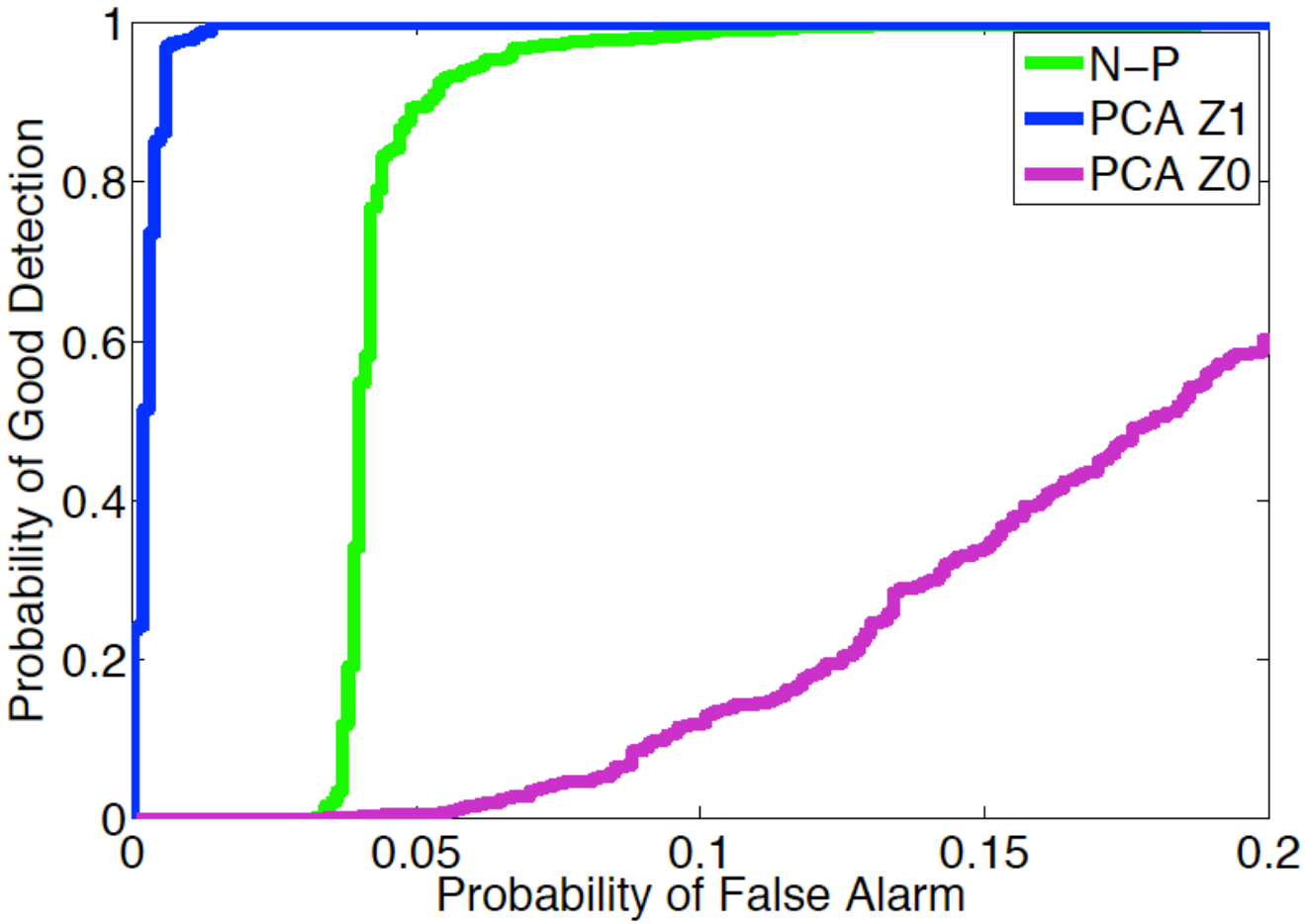
$$\begin{aligned} & \prod_{i=1}^{m} \Pr[Y_i = y_i] \\ = & \prod_{i=1}^{m} \hat{p}_i{}^{\theta_i} \left(\Pr[Y_i = y_i | Y_i \neq \phi](1 - \hat{p}_i)\right)^{1-\theta_i} \end{aligned}$$

- *Not* a realistic model of genuine ratings – assumes user's ratings are all *independent*
- But sufficient (almost) to distinguish from attack profiles.

Random Attack (Filler Size=5%)



Average Attack (Filler Size=5%)



Lesson Learned

- Filler item selection is key to the success of the standard attacks on k-NN user-based algorithm.
  - Low (attack profile / genuine profile) overlap (few common ratings) makes extreme correlations possible – hence attack profiles are unusually influential. (Basis of original attack proposed in O'Mahony et al. (2002).)
  - But also allows for successful detection – also highly perceptible.

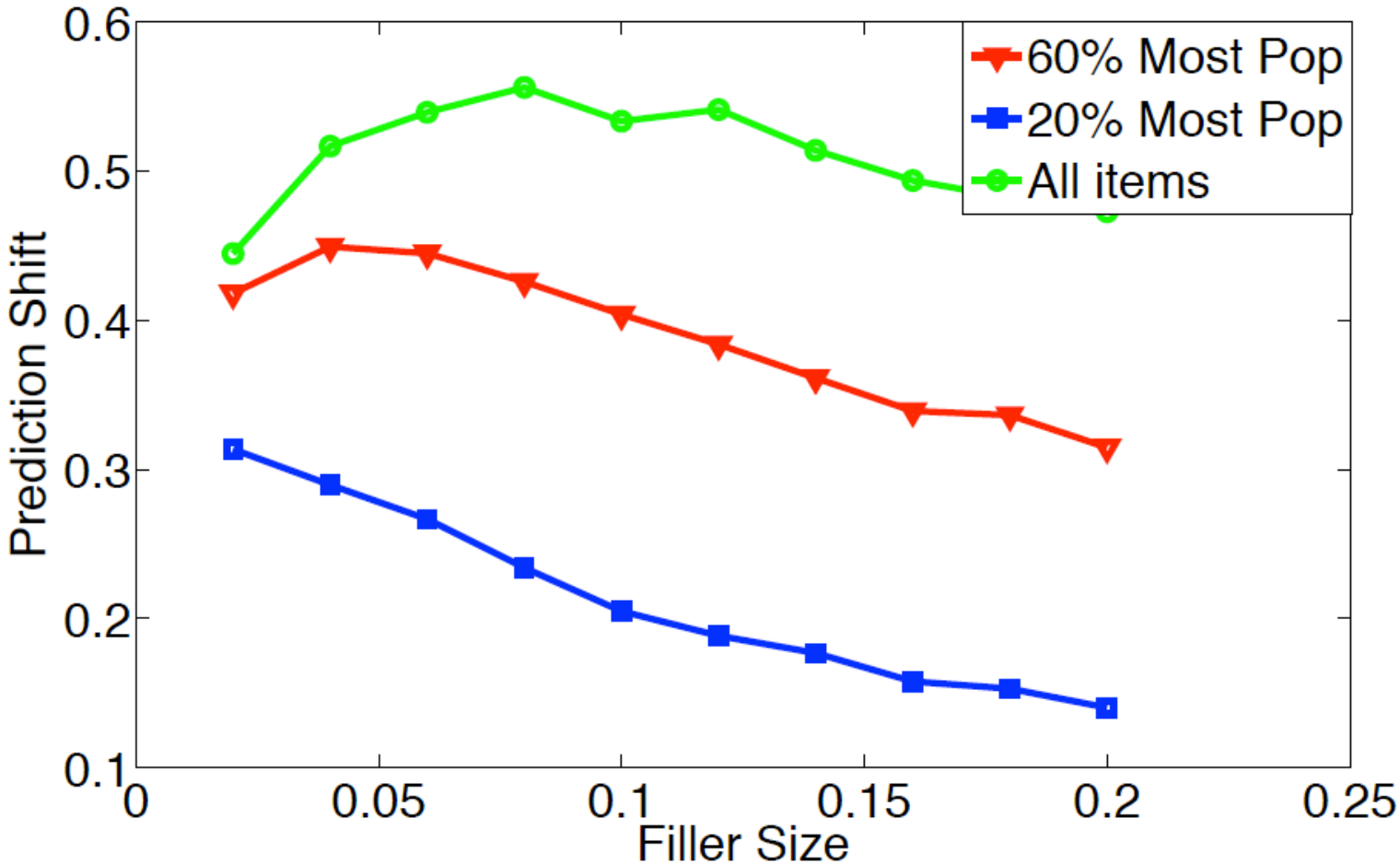
Attack Obfuscation

- Several obfuscation strategies evaluated previously (Williams et al. 2006).
- Effective obfuscation must try to reduce the statistical differences between genuine and attack profiles.

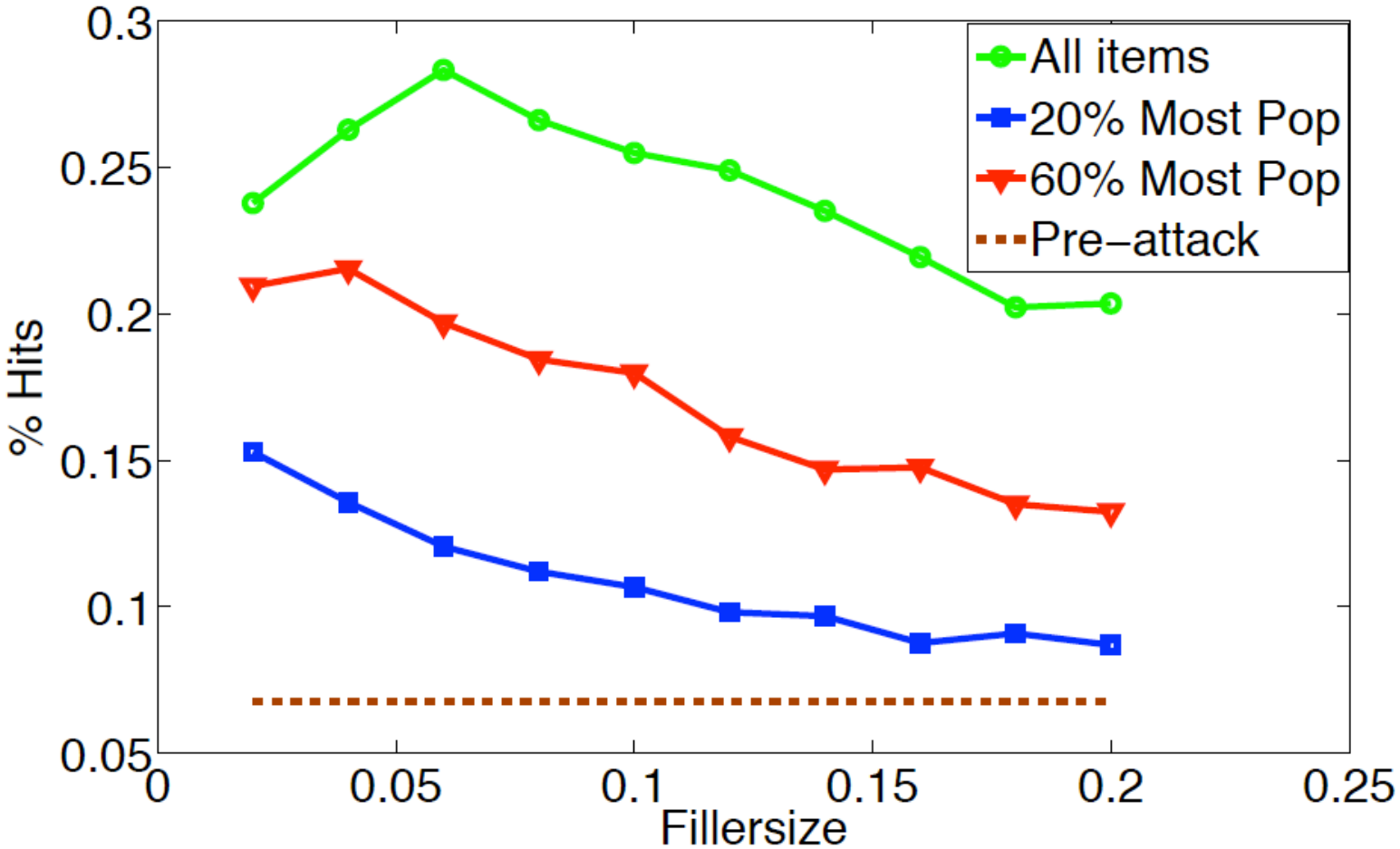
Average over Popular (AoP) attack

- It is clear that a major weakness of the average and random attacks is their unrealistic selection of filler items.
- To be imperceptible an attacker must choose items to rate in a similar fashion to genuine users.
- **Average Over Popular** Attack – identical to average attack, but filler items are chosen from x-% *most popular* items.
- AoP is a less perceptible but also less effective attack on kNN user-based algorithm.
- Nevertheless it does work . . .

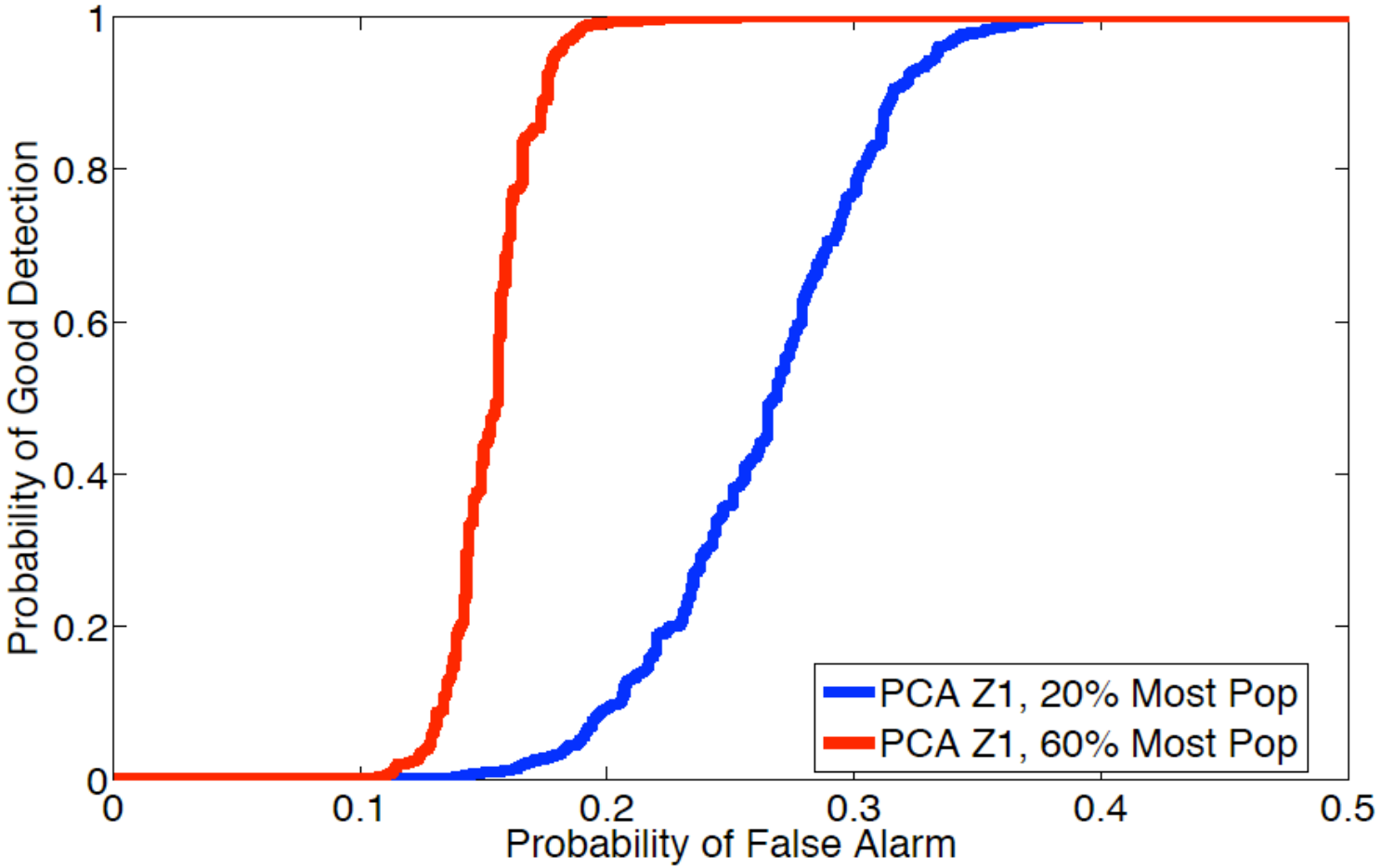
AoP Prediction Shift (Attack Size=3%)



AoP Hits (rating of $\geq$ 4) (Attack Size=3%)



AoP PCA Detection



Improved Genuine Profile Model

- Adopt model that takes account of correlations between ratings.
  - Assume ratings follow multivariate normal distribution – parameters:
    - $\boldsymbol{\mu}_0$, m-dimensional vector of mean-ratings for each item; and
    - Σ_0, $m \times m$ matrix of item correlations.
  - Assume attack profiles also multivariate gaussian with parameters $\boldsymbol{\mu}_1$, Σ_1
- Hence, attacked database can be modelled as a Gaussian Mixture Model.

Improved Genuine Profile Model

- Difficulty
  - Very high dimensional vectors
  - Missing values
- Factor Model: Assume that the rating matrix can be represented by the linear model

$$\mathrm{Y}^T = \mathrm{A}\mathrm{X}^T + \mathrm{N}\,,$$

  - X : $n \times k$ matrix, $x_{u,j}$ = extent user u likes category j
  - A : $m \times k$ matrix $a_{i,j}$ = extent item i belongs in category j
  - N : independent noise
  - $\mathrm{X}(i,:)$ assumed independent normal.
- Expectation maximisation to learn A from dataset ([Canny, 2002]).

Improved Genuine Profile Model

- N-P test on **multivariate normal** k-dimensional vector obtained by projection with A

$$\mathbf{w} = \mathrm{A}^T\mathrm{Y}^T = \mathrm{A}^T(\mathrm{A}\mathrm{X}^T + \mathrm{N})$$

  - X : $n \times k$ matrix, $x_{u,j}$ = extent user u likes category j
  - A : $m \times k$ matrix $a_{i,j}$ = extent item i belongs in category j
  - N : independent noise
  - $\mathrm{X}(i,:)$ assumed independent normal.

Projection by Clustering

- N-P test on **multivariate normal** k-dimensional vector obtained by projection with P

$$\mathbf{w} = \mathrm{P}^T \mathrm{Y}^T$$

  - Obtain a clustering of item-set into k clusters of similar items.
  - P : $n \times k$ projection matrix sums all ratings belonging to a cluster.

Supervised AoP Detection

- N-P test reduces to

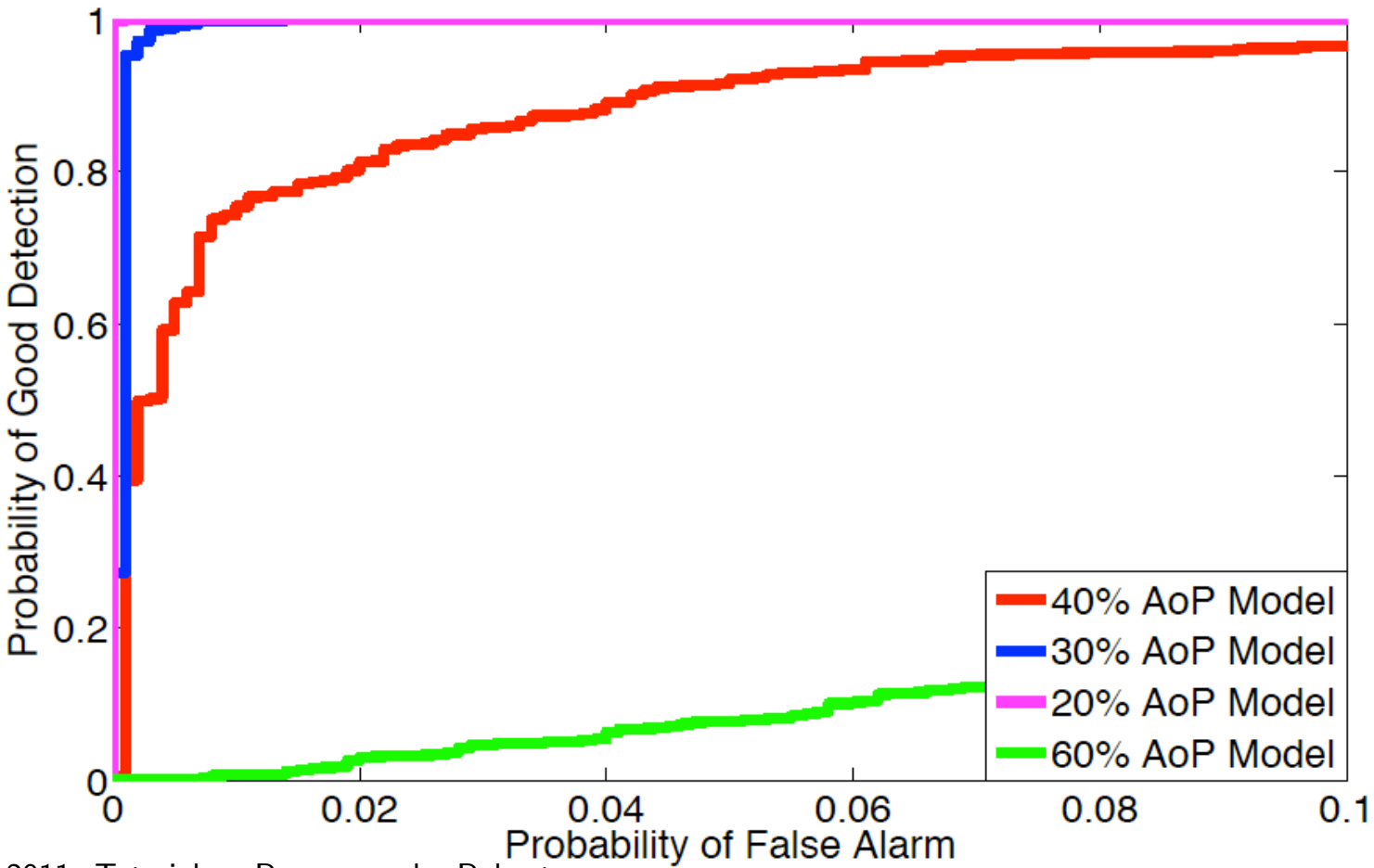
$$(\mathbf{w}-\boldsymbol{\mu}_{\boldsymbol{w}0})^T{\Sigma_{w0}}^{-1}(\mathbf{w}-\boldsymbol{\mu}_{\boldsymbol{w}0})-(\mathbf{w}-\boldsymbol{\mu}_{\boldsymbol{w}1})^T{\Sigma_{w1}}^{-1}(\mathbf{w}-\boldsymbol{\mu}_{\boldsymbol{w}1}) \lessgtr \eta$$

- Need
  - Database of genuine profiles *and*
  - Database of attack profiles

  to train the detector.

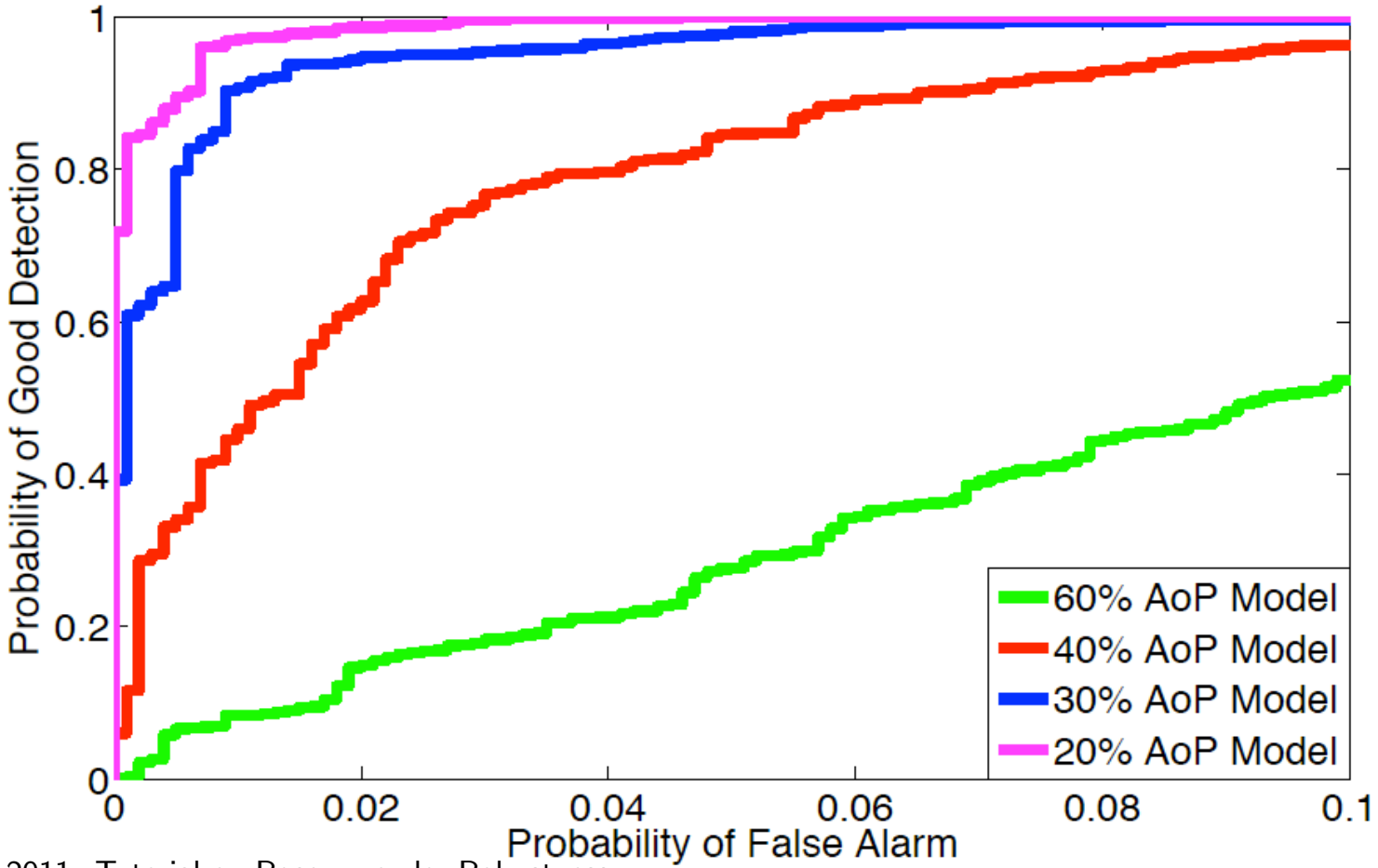
AoP Detection (Factor Analysis)

- *Actual attack*=20% AoP Attack. Plot shows training on different attack types.



AoP Detection (Clustering)

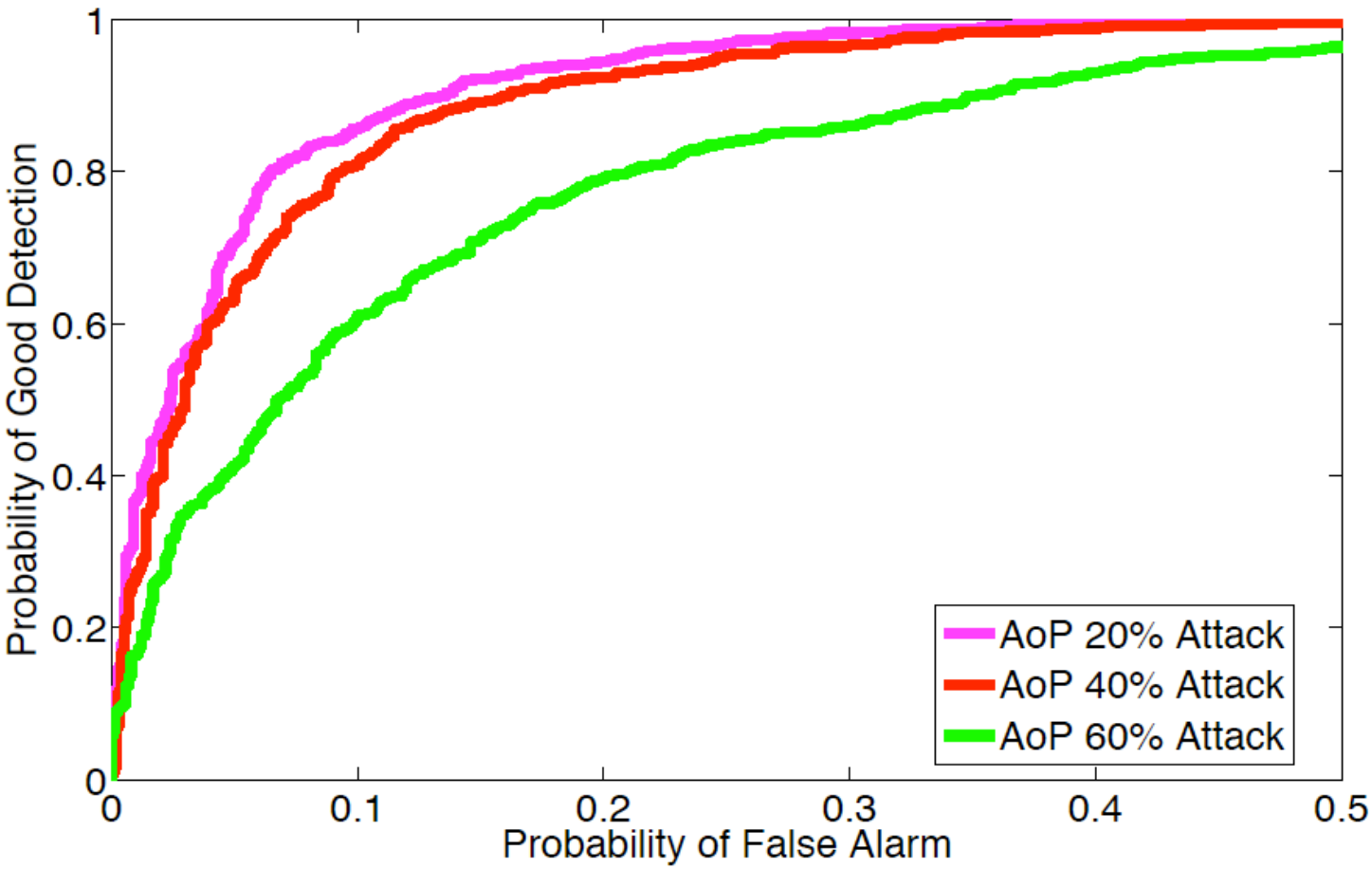
- *Actual attack*=20% AoP Attack. Plot shows training on different attack types.



Unsupervised AoP Detection

- Unsupervised Gaussian mixture model to learn model parameters
  - $\boldsymbol{\mu}_{\boldsymbol{w}0}$, Σ_{w0}, $\boldsymbol{\mu}_{\boldsymbol{w}1}$, Σ_{w1}, a_0 and a_1, where a_i is the probability that a profile belongs in cluster i.
- Implementation from William Wong, Purdue University, `http://web.ics.purdue.edu/~wong17/`.

AoP Detection



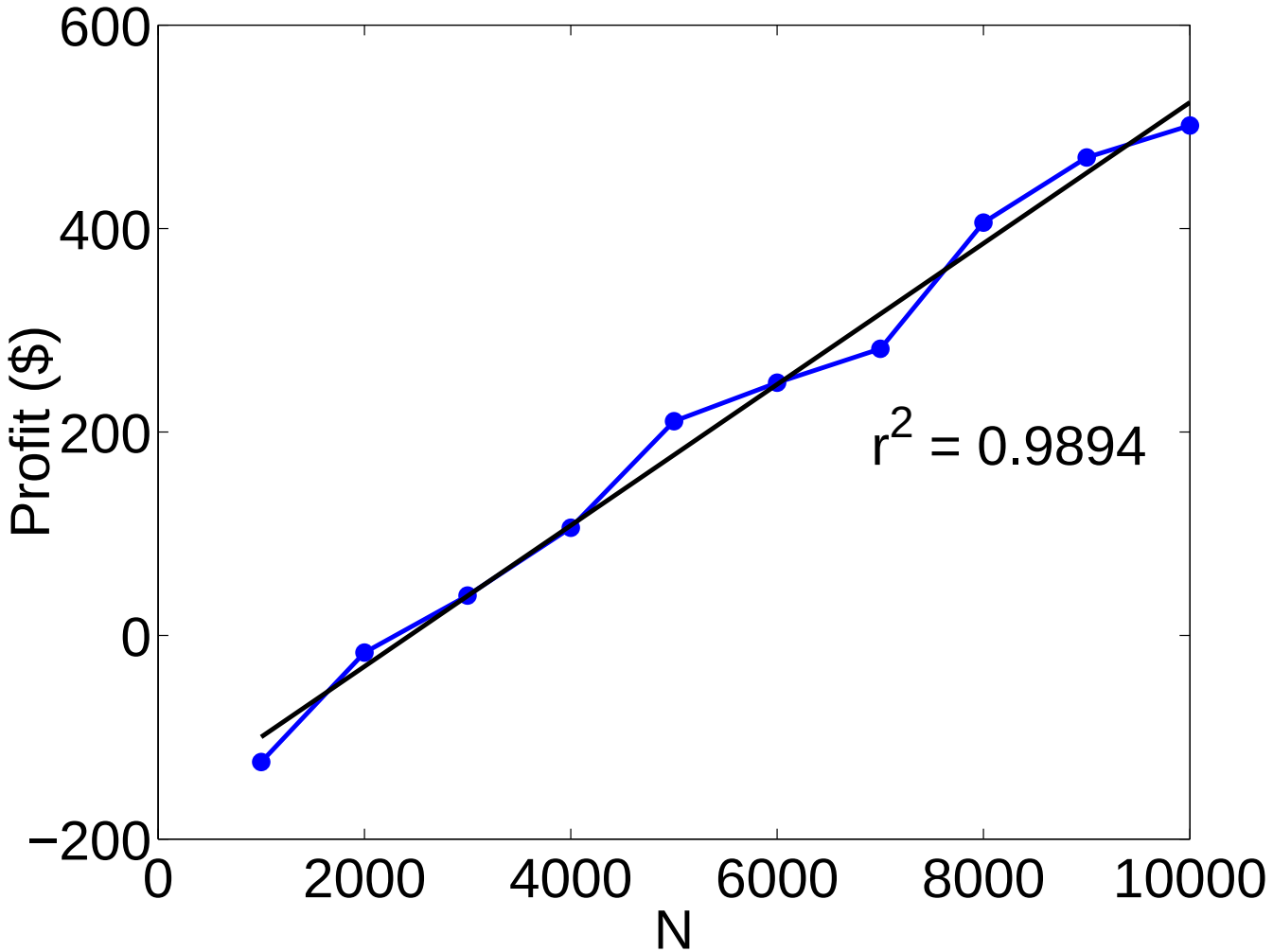
Outline

Cost-Benefit Analysis

- In O'Mahony et al. (2006), we examined a simple cost-benefit model
  - The ROI for an attack on item j given by:
    $$\text{ROI}_j = \frac{c_j N(n'_j - n_j) - \sum_{k \in I_j} c_k}{\sum_{k \in I_j} c_k}$$
  - Simplify by assuming $c_p = c_q, \ \forall \ p, q \in I$
    $$\text{ROI}_j = \frac{N(n'_j - n_j) - s_j}{s_j}$$
    - s_j – total number of ratings inserted in an attack on item j
  - Approximated the fractions n_j & n'_j using count of good predictions and browser-to-buyer conversion probability

Results

- $\text{Profit}_j = c\big(\alpha_j(\text{GP}'_j - \text{GP}_j) - s_j\big)$; $c = \$10$, $\alpha = 10\%$



Cost-benefit Analysis

- Vu et al. (2010) examines the *adversarial cost* to attacking a ranking system in terms of n_A = number of profiles and m_A=number of ratings.
- They assume a rating function $r(u, i) \in \{-1, 0, 1\}$ and a quality-popularity score for each item

$$f(i) = \sum_{u \in \mathcal{G}} r(u, i)$$

Cost-benefit Analysis

- Using a trust mechanism that can detect a malicious rating with probability γ, they show that a ranking function of the form

$$f(i) = \sum_{u \in \mathcal{G} \cup \mathcal{A}} r(u,i)t(u,i)$$

can be used to design a ranking system in which the minimum adversarial cost in expectation to boost the rank of an item from k to 1 includes the cost of posting $m_A{=}n_A$ ratings, with

$$n_A = (x_1 + x_k)\frac{1 - 2\epsilon + \epsilon\gamma}{1 - \gamma}$$

where x_i denotes the number of honest ratings for item i and ϵ is the probability of an erroneous rating.

Outline

Model-based Algorithms

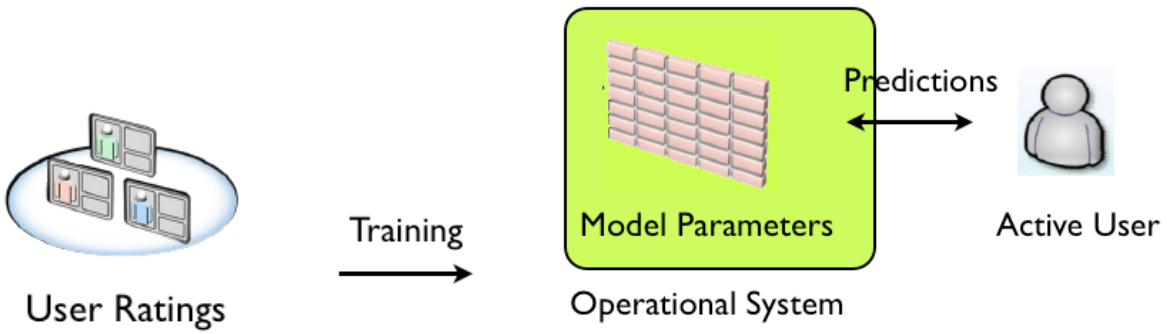
User Ratings

Model-based Algorithms

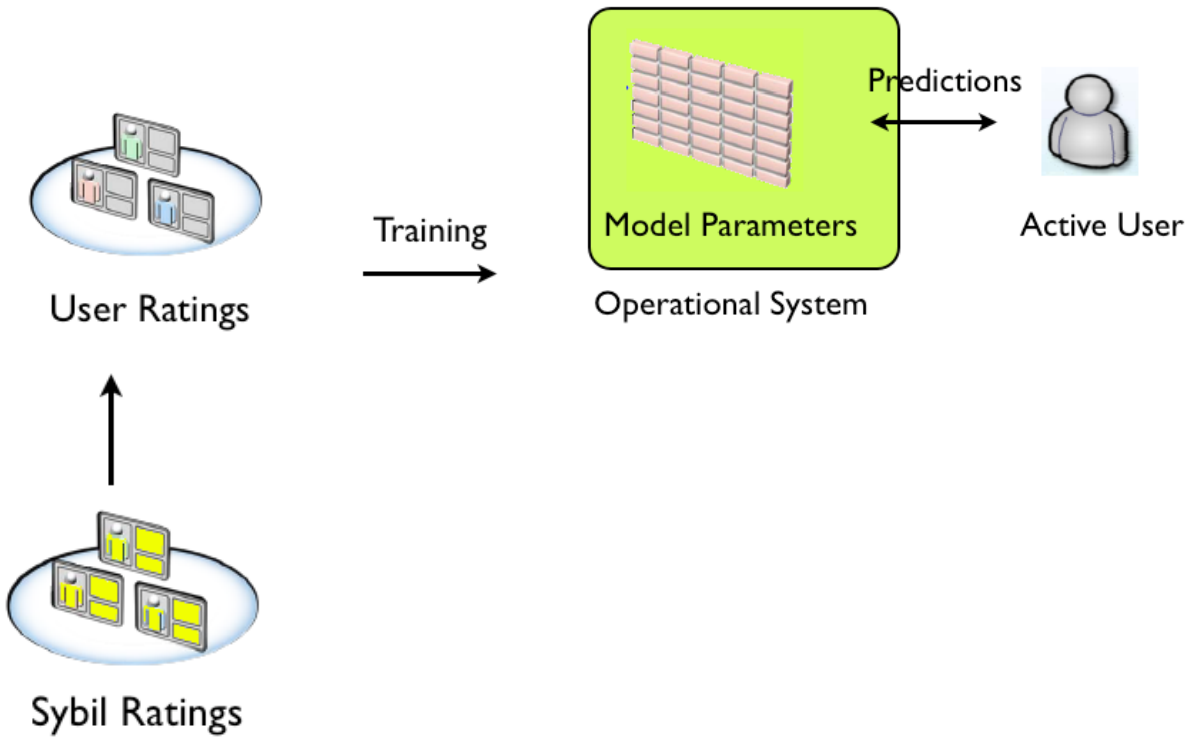
User Ratings

Training →

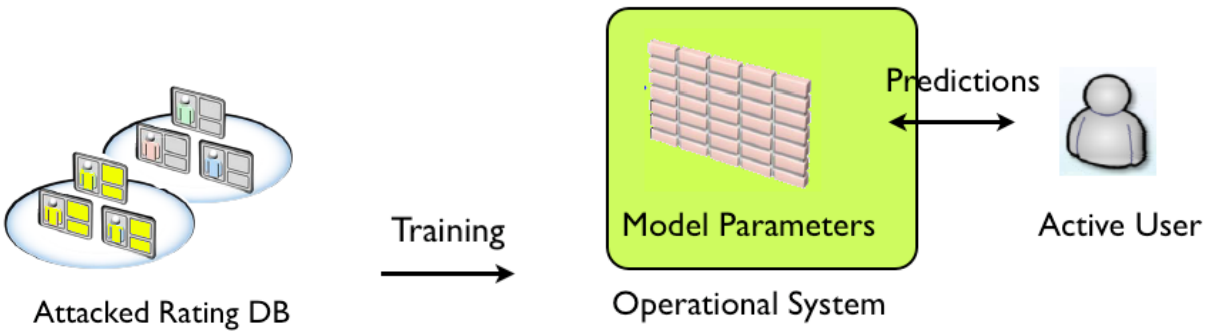
Model-based Algorithms



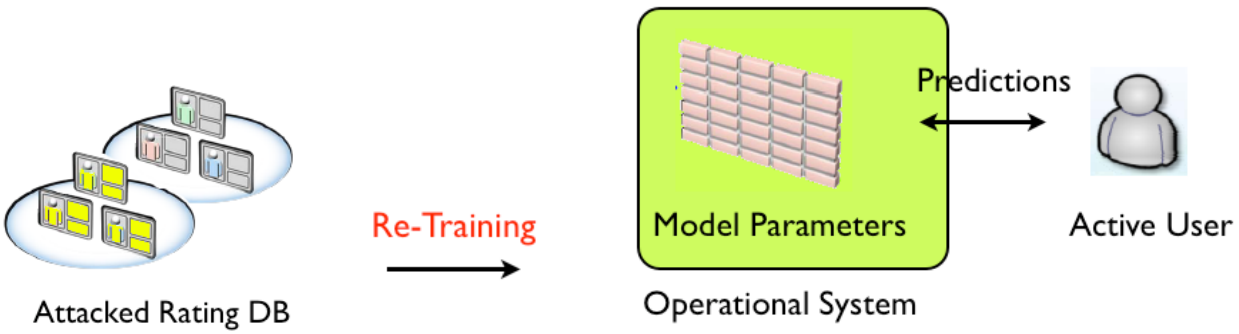
Model-based Algorithms



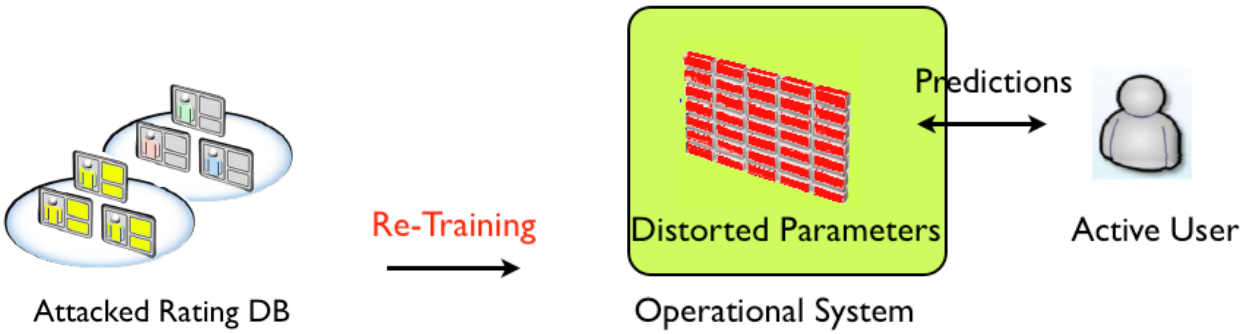
Model-based Algorithms



Model-based Algorithms



Model-based Algorithms



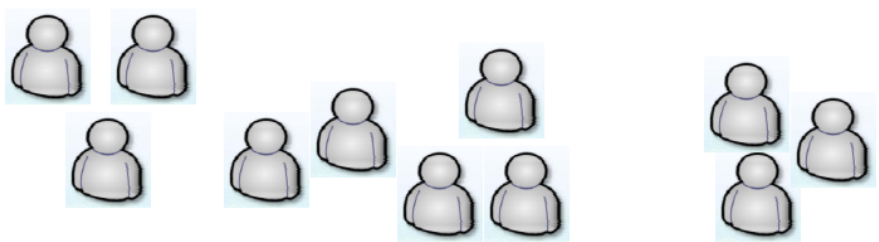
- Attack takes effect when model is re-trained, using corrupted ratings database.

Manipulation-resistance of Model-based Algorithms

- Initial work such as Mobasher et al. (2006) showed that model-based algorithms are more resistant to manipulation than memory-based algorithms.

Manipulation-resistance of Model-based Algorithms

- Initial work such as Mobasher et al. (2006) showed that model-based algorithms are more resistant to manipulation than memory-based algorithms.



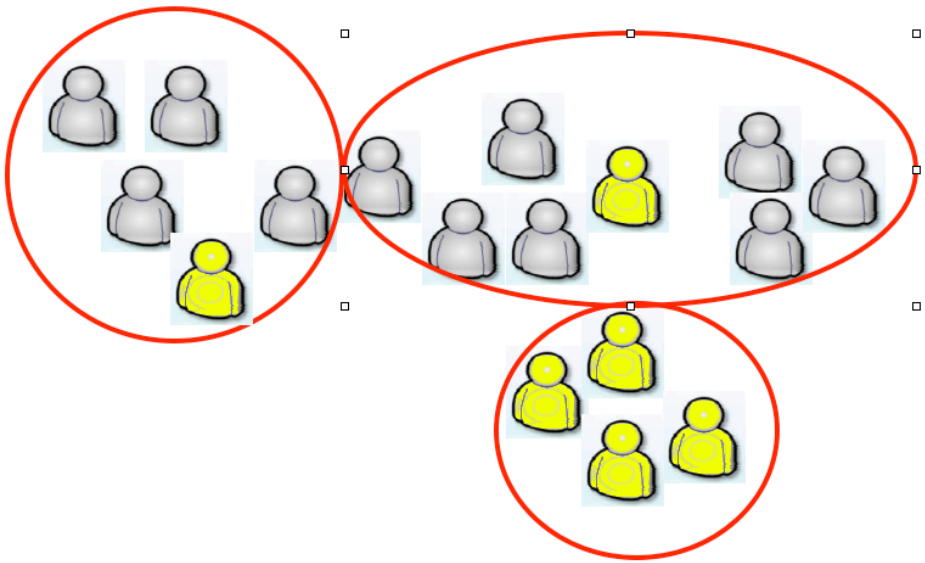
Manipulation-resistance of Model-based Algorithms

- Initial work such as Mobasher et al. (2006) showed that model-based algorithms are more resistant to manipulation than memory-based algorithms.

- The user-base is clustered into *segments* using k-means or PLSA clustering and users matched to closest segment profiles.

Manipulation-resistance of Model-based Algorithms

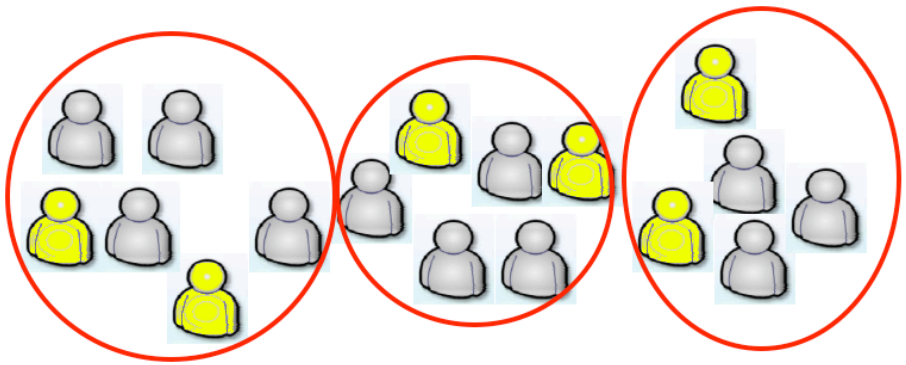
- Initial work such as Mobasher et al. (2006) showed that model-based algorithms are more resistant to manipulation than memory-based algorithms.



- Sybil profiles tend to be clustered into same segment, thereby reducing power of attack.

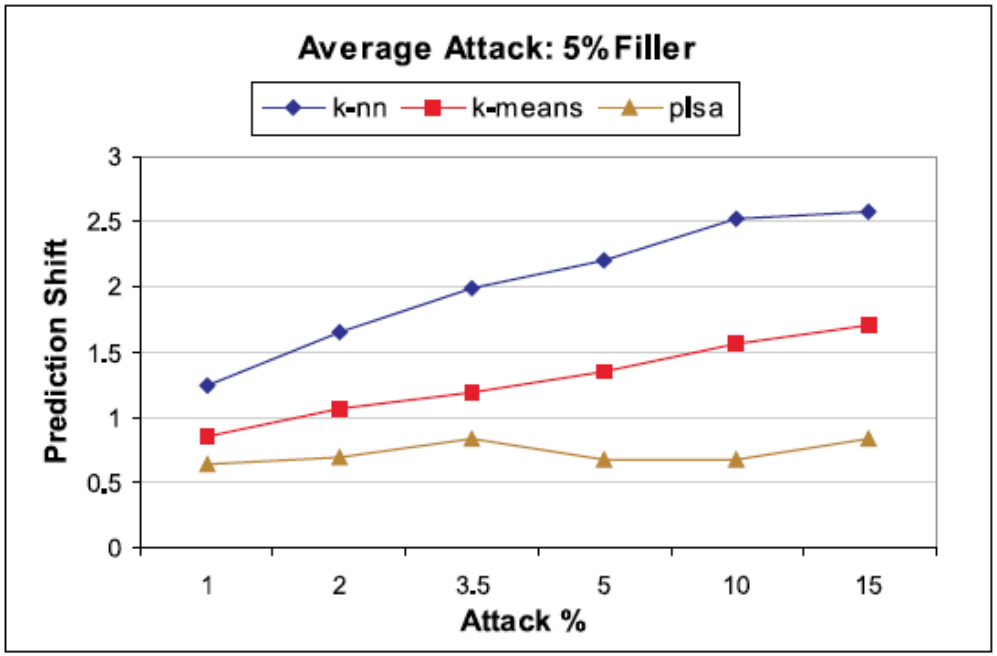
Manipulation-resistance of Model-based Algorithms

- Initial work such as Mobasher et al. (2006) showed that model-based algorithms are more resistant to manipulation than memory-based algorithms.



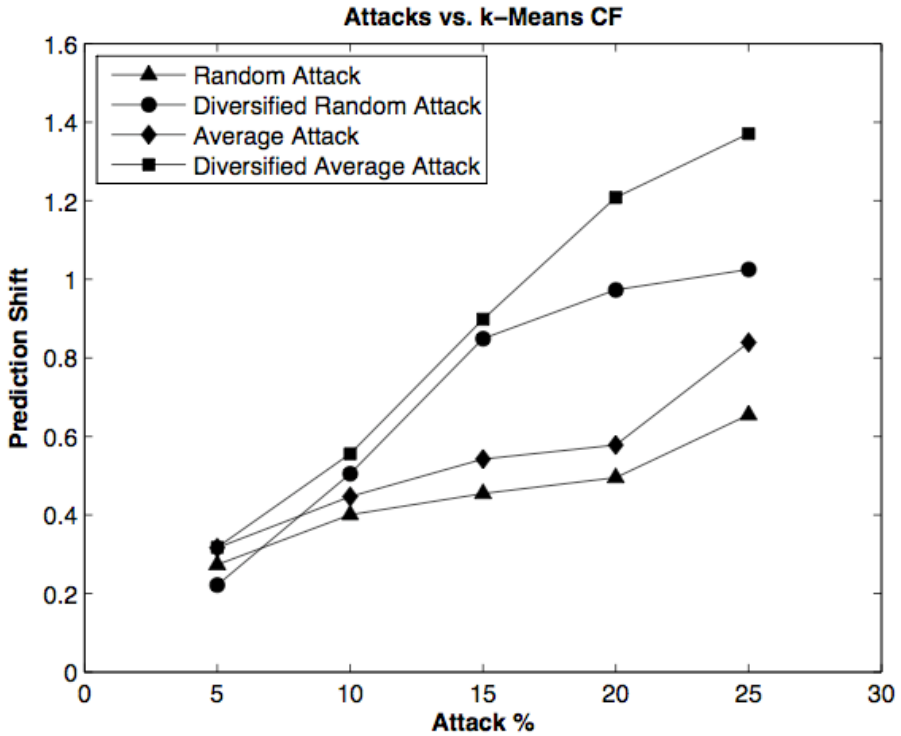
- However, as shown in Cheng and Hurley (2009a), a diversified attack can create less similar but yet still effective attack profiles.

Manipulation-resistance of Model-based Algorithms



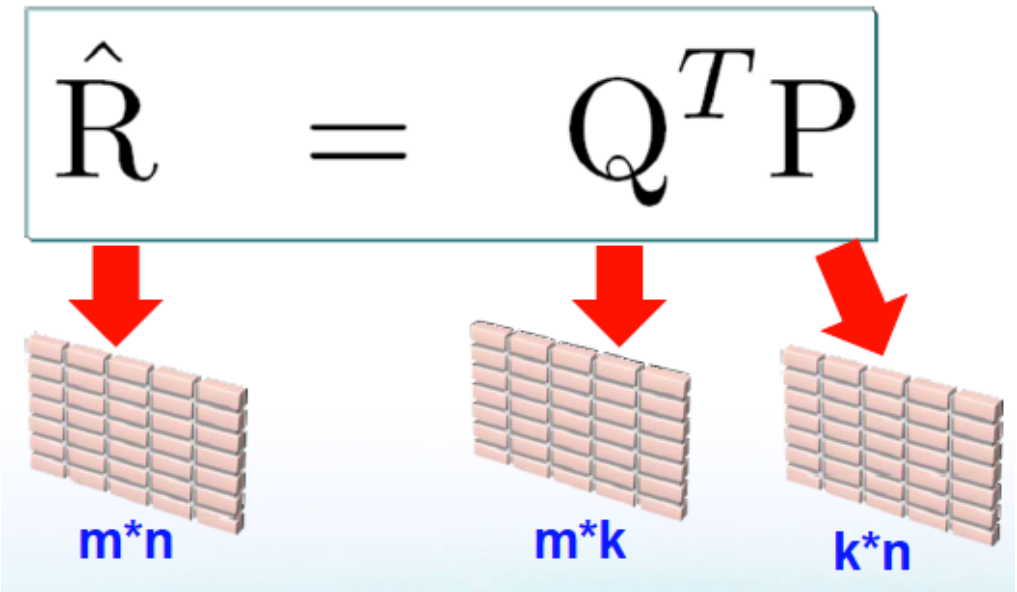
From Mobasher et al. (2006): Evaluation on MovieLens 100K dataset

Manipulation-resistance of Model-based Algorithms



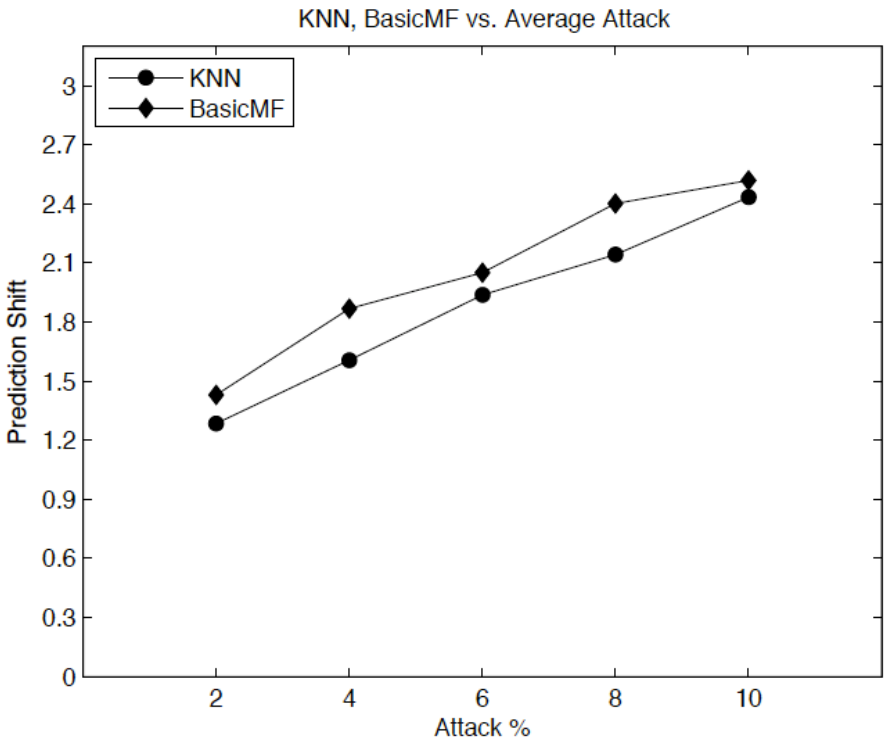
From Cheng and Hurley (2009a): Evaluation on MovieLens 1M dataset

Manipulation-resistance of Matrix Factorisation



- Modern matrix factorization algorithms use a least squares step to find the factors. This is known to be sensitive to outliers.

Manipulation-resistance of Matrix Factorisation



- Cheng and Hurley (2010) Prediction shift of basic Bellkor algorithm kNN for a single randomly chosen unpopular item.

Summary of Robustness Results on Standard Algorithms

- A number of general attack strategies have been proposed that work effectively in particular on memory-based algorithms.

Summary of Robustness Results on Standard Algorithms

- A number of general attack strategies have been proposed that work effectively in particular on memory-based algorithms.
- Standard attacks are generally detectable:

Summary of Robustness Results on Standard Algorithms

- A number of general attack strategies have been proposed that work effectively in particular on memory-based algorithms.
- Standard attacks are generally detectable:
  - Cost effectiveness implies that a sybil profile should be unusually influential.
- Special purpose (*informed*) attacks can be tailored towards particular CF algorithms
  - e.g. a high diversity attack proved effective against the k-means clustering algorithm.
- Sybil profiles can be obfuscated to make them less detectable
  - Selection of filler items is Achille's heal of standard average attack, but also explains its power.
  - Obfuscation reduces the effectiveness of attacks but memory-based algorithms are still vulnerable.

Outline

VarSelectSVD

- Some early work (O'Mahony et al. 2004) applied *neighbourhood filtering* to the standard user-based algorithm, to cluster and filter suspicious neighbours.

VarSelectSVD

- Some early work (O'Mahony et al. 2004) applied *neighbourhood filtering* to the standard user-based algorithm, to cluster and filter suspicious neighbours.
- Mehta and Nejdl (2008) introduces the the VarSelectSVD method – a matrix factorisation strategy taking robustness into account.

VarSelectSVD

- Some early work (O'Mahony et al. 2004) applied *neighbourhood filtering* to the standard user-based algorithm, to cluster and filter suspicious neighbours.
- Mehta and Nejdl (2008) introduces the the VarSelectSVD method – a matrix factorisation strategy taking robustness into account.
  - An SVD factorization of the rating matrix R requires the following to be solved:

$$\arg\min_{\mathrm{G,H}} \|\mathrm{R} - \mathrm{GH}\|$$

VarSelectSVD

- Some early work (O'Mahony et al. 2004) applied *neighbourhood filtering* to the standard user-based algorithm, to cluster and filter suspicious neighbours.
- Mehta and Nejdl (2008) introduces the the VarSelectSVD method – a matrix factorisation strategy taking robustness into account.
  - An SVD factorization of the rating matrix R requires the following to be solved:

$$\arg\min_{\mathrm{G,H}} \|\mathrm{R} - \mathrm{GH}\|$$

  - Users are flagged as suspicious using PCA clustering.

VarSelectSVD

- Some early work (O'Mahony et al. 2004) applied *neighbourhood filtering* to the standard user-based algorithm, to cluster and filter suspicious neighbours.
- Mehta and Nejdl (2008) introduces the the VarSelectSVD method – a matrix factorisation strategy taking robustness into account.
  - An SVD factorization of the rating matrix R requires the following to be solved:

$$\arg\min_{\mathrm{G,H}} \|\mathrm{R} - \mathrm{GH}\|$$

  - Users are flagged as suspicious using PCA clustering.
  - Flagged users do not contribute to the update of right eigenvector – they do not contribute to the model.

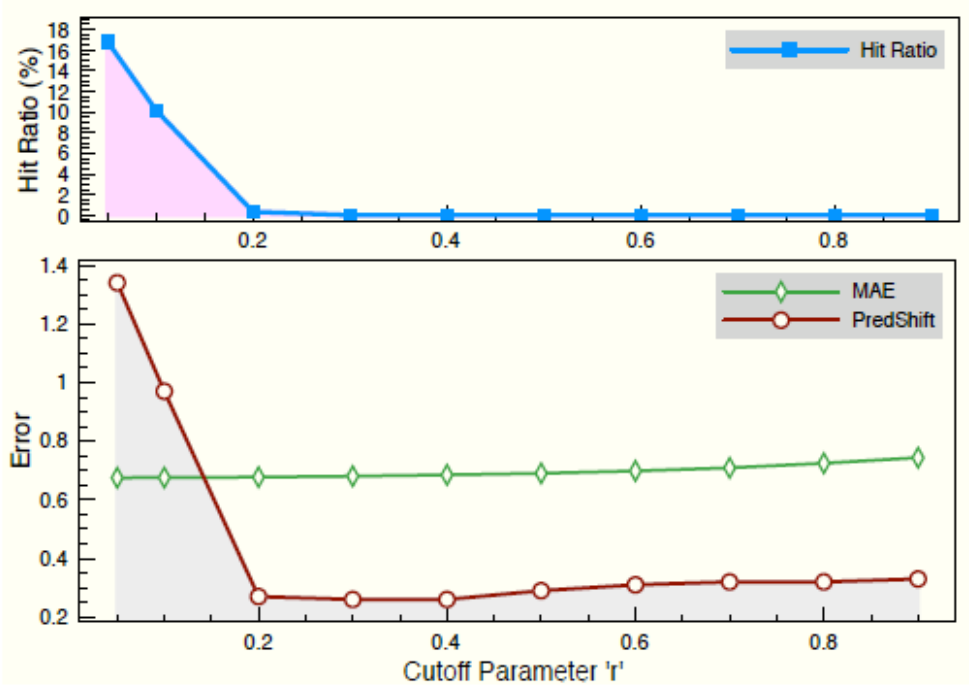
VarSelectSVD

Algorithm 1 VarSelectSVD ($\mathbf{D}$)

1: $\mathbf{D} \leftarrow$ z-scores($\mathbf{D}$) {$\mathbf{D}$ has N users and M items}
2: $\mathbf{U}\lambda\mathbf{V}^{\mathrm{T}}$= SVD($\mathbf{D}$,3) {Get 3 principal components $\mathbf{U}^T$}
3: $PCA_1 \leftarrow \mathbf{U}(:,1), PCA_2 \leftarrow \mathbf{U}(:,2), PCA_3 \leftarrow \mathbf{U}(:,3)$ {First 3 PC loadings }
4: **for all** columnid *user* in $\mathbf{D}$ **do**
5: $Score(user) \leftarrow (|PCA_1(user)| + |PCA_2(user)| + |PCA_3(user)|)/3$ {using LC ranking scheme}
6: **end for**
7: Normalize and Sort *Score* {*Score* now sum to 1.}
8: $r_1 \leftarrow$ number of users with *Score* below $\frac{1}{N}$
9: $r_2 \leftarrow N/5$ {Cutoff set to 20%.}
10: $r \leftarrow min(r_1, r_2)$
11: Flag top r users with smallest *Score* values
12: **for** Factor f_k with $k \leftarrow 1$ to d **do**
13: $\mathbf{D} = \mathbf{D} - \mathbf{G}_{k-1} \cdot \mathbf{H}_{k-1}^{\mathrm{T}}$
14: **repeat**
15: $res_{ij} = \mathbf{D}_{ij} - \hat{G}_i \cdot \hat{H}_j$ {set $\kappa = 0.01$}
16: $\triangle\hat{G}_i = \lambda(\hat{H}_j \cdot res_{ij} - \kappa \cdot \hat{G}_i)$
17: **if** u is not flagged *or* $v_{min} < D_{ij} < v_{max}$ **then**
18: $\triangle\hat{H}_j = \lambda(\hat{G}_i \cdot res_{ij} - \kappa \cdot \hat{H}_j)$
19: **end if**
20: **until** Convergence of $\hat{G}_i, \hat{H}_j$ for all i, j
21: **end for**

Output: Return Matrix factors $\mathbf{G}, \mathbf{H}$

VarSelectSVD Results



- MAE, Prediction Shift and Hit Ratio for 5% Average Attack, 7% Filler on Movielens 1M

Manipulation Resistance Through Robust Statistics

- Robust statistics describes an alternative approach to classical statistics where the motivation is to produce estimators that are not unduly affected by small departures from the model.
- Robust regression uses a bounded cost function which limits the effect of outliers.

Two Approaches

Manipulation Resistance Through Robust Statistics

- Robust statistics describes an alternative approach to classical statistics where the motivation is to produce estimators that are not unduly affected by small departures from the model.
- Robust regression uses a bounded cost function which limits the effect of outliers.

Two Approaches

1 **M-estimators** –

$$\arg\min_{\mathrm{G,H}} \sum_{r_{ij}\neq 0} \rho(r_{ij} - g_i h_j) \quad \text{where } \rho(r) = \begin{cases} \frac{1}{2\gamma} r^2 & |r| \leq \gamma \\ |r| - \frac{\gamma}{2} & |r| > \gamma \end{cases}$$

Manipulation Resistance Through Robust Statistics

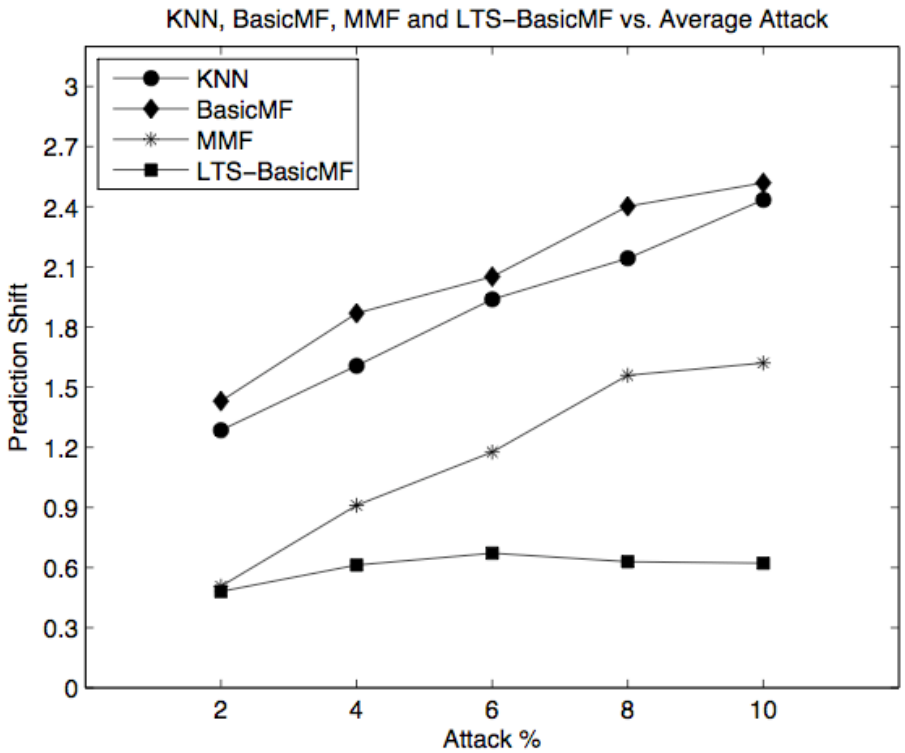
- Robust statistics describes an alternative approach to classical statistics where the motivation is to produce estimators that are not unduly affected by small departures from the model.
- Robust regression uses a bounded cost function which limits the effect of outliers.

Two Approaches

2 Least Trimmed Squares –

$$\arg\min_{\mathrm{G,H}} \sum_{i=1}^{h} e^2_{(i)} \quad \text{where } e^2_{(1)} \leq e^2_{(2)} \leq \cdots \leq e^2_{(n)}$$

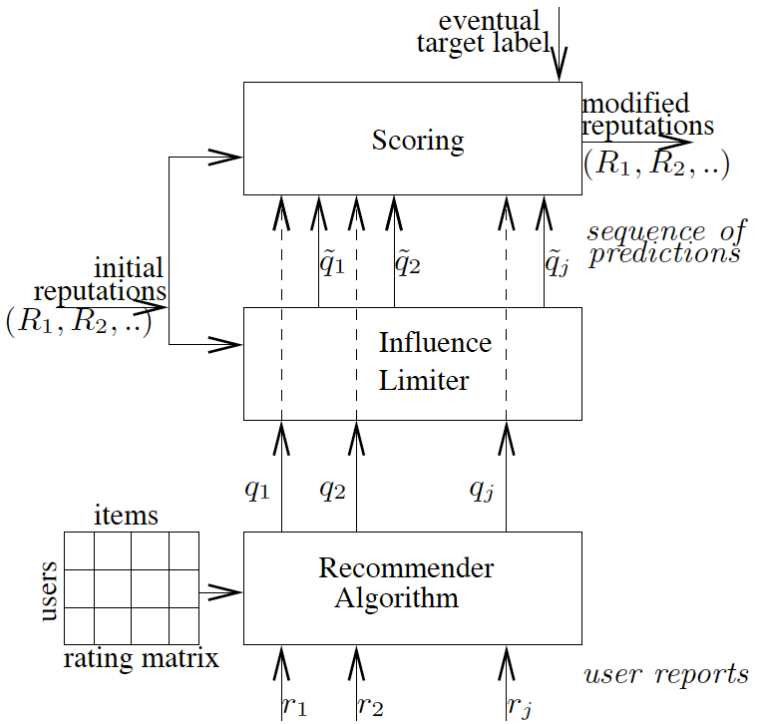
Robust Statistics Results on Bellkor Method



Outline

The Influence Limiter (Resnick and Sami 2007)

- Output of recommendation system q_j passes through an influence-limiting process to produce a modified output $\tilde{q}_j$.
- $\tilde{q}_j$ is a weighted average of $\tilde{q}_{j-1}$ and q_j
- The weighting depends on the reputation that user j has accumulated with respect to the target.



The Influence Limiter (Resnick and Sami 2007)

- A scoring function assigns reputation to j based on whether or not the target actually likes the item.
- Tuned so that honest users can reach full credibility after $O(\log n)$ steps.
- The main result states that the total impact of n sybils in terms of performance reduction is bounded by $-ne^{\lambda}$.

ComputeReputations(λ)

1. Initialize $R_j = e^{-\lambda}$ for all j.
2. For an item the target will eventually label do:
   a. $\tilde{q}_0 = p_0$
   b. Consider the ratings on the item in temporal order
   c. For each rater j:
   d. $\beta_j = \min(1, R_j)$
   e. $\tilde{q}_j = (1 - \beta_j)\tilde{q}_{j-1} + \beta_j q_j$
   f. After the target provides label l, $R_j = R_j + \beta_j[L(l, \tilde{q}_{j-1}) - L(l, q_j)]$

Yan and Roy (2009)'s Manipulation Robust Algorithm

- A class of CF algorithms – which the authors' call **linear CF algorithms** – is presented in which the manipulation distortion is bounded above by

$$\frac{1}{n}\frac{1}{1-r}.$$

where r is the fraction of the data that is generated by manipulators and n is the number of products that have already been rated by a user whose future ratings are to be predicted.

  - *"Suppose a CF system that accepts binary ratings predicts future ratings correctly 80% of the time in the absence of manipulation. If 10% of all ratings are provided by manipulators, according to our bound, the system can maintain a 75% rate of correct predictions by requiring each new user to rate at least 21 products before receiving recommendations."*

Yan and Roy (2009)'s Manipulation Robust Algorithm

- Let $r \in \mathcal{L}$ be a rating in the label space.
- Let $\boldsymbol{\nu}$ be a permutation of the items such that ν_n is the n^{th} item rated by an active user, a.
- Let $\mathbf{r}_a^{n-1}$ be the active user's profile after rating $n-1$ items.
- Then a CF algorithm may be described as

Yan and Roy (2009)'s Manipulation Robust Algorithm

- Let $r \in \mathcal{L}$ be a rating in the label space.
- Let $\boldsymbol{\nu}$ be a permutation of the items such that ν_n is the n^{th} item rated by an active user, a.
- Let $\mathbf{r}_a^{n-1}$ be the active user's profile after rating $n-1$ items.
- Then a CF algorithm may be described as

$$P(r_{a,\nu_n} = r | \mathbf{r}_a^{n-1}, \mathrm{R}, \boldsymbol{\nu})$$

Yan and Roy (2009)'s Manipulation Robust Algorithm

- Let $r \in \mathcal{L}$ be a rating in the label space.
- Let $\boldsymbol{\nu}$ be a permutation of the items such that ν_n is the n^{th} item rated by an active user, a.
- Let $\mathbf{r}_a^{n-1}$ be the active user's profile after rating $n-1$ items.
- Then a CF algorithm may be described as

$$P(r_{a,\nu_n} = r | \mathbf{r}_a^{n-1}, \mathrm{R}, \boldsymbol{\nu})$$

  - A PMF over the rating that the active user gives for the ν_n item.

Yan and Roy (2009)'s Manipulation Robust Algorithm

- Let $r \in \mathcal{L}$ be a rating in the label space.
- Let $\boldsymbol{\nu}$ be a permutation of the items such that ν_n is the n^{th} item rated by an active user, a.
- Let $\mathbf{r}_a^{n-1}$ be the active user's profile after rating $n-1$ items.
- Then a CF algorithm may be described as

$$P(r_{a,\nu_n} = r | \mathbf{r}_a^{n-1}, \mathrm{R}, \boldsymbol{\nu})$$

  - Depending on the active user's current profile

Yan and Roy (2009)'s Manipulation Robust Algorithm

- Let $r \in \mathcal{L}$ be a rating in the label space.
- Let $\boldsymbol{\nu}$ be a permutation of the items such that ν_n is the n^{th} item rated by an active user, a.
- Let $\mathbf{r}_a^{n-1}$ be the active user's profile after rating $n-1$ items.
- Then a CF algorithm may be described as

$$P(r_{a,\nu_n} = r | \mathbf{r}_a^{n-1}, \mathrm{R}, \boldsymbol{\nu})$$

  - the training set, R

Yan and Roy (2009)'s Manipulation Robust Algorithm

- Let $r \in \mathcal{L}$ be a rating in the label space.
- Let $\boldsymbol{\nu}$ be a permutation of the items such that ν_n is the n^{th} item rated by an active user, a.
- Let $\mathbf{r}_a^{n-1}$ be the active user's profile after rating $n-1$ items.
- Then a CF algorithm may be described as

$$P(r_{a,\nu_n} = r | \mathbf{r}_a^{n-1}, \mathrm{R}, \boldsymbol{\nu})$$

  - the order in which the user rates items.

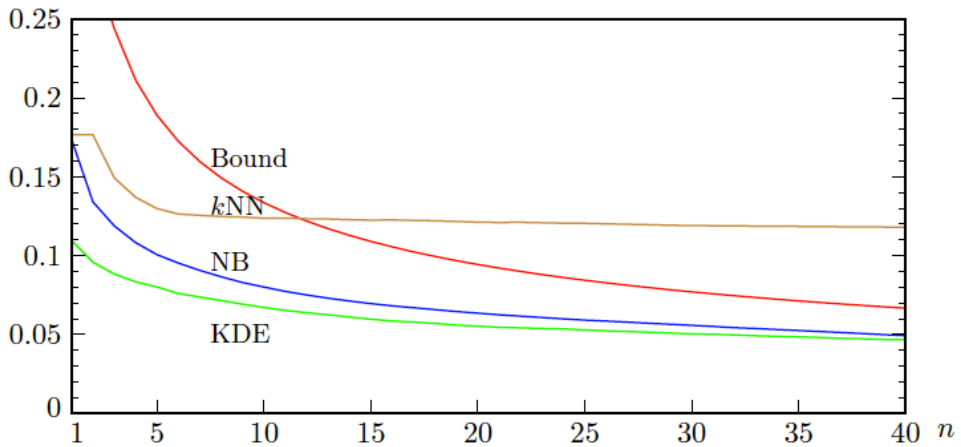
Yan and Roy (2009)'s Manipulation Robust Algorithm

- A *probabilistic* CF algorithm predicts independently of the order $\boldsymbol{\nu}$.
- Let $\mathrm{R}' = (\mathrm{R}, \mathrm{Y})$ be a database divided in proportion $(1 - r)$ to r between R and Y.
- Then a linear probabilistic CF algorithm is one in which

$$P(.|\mathbf{r}_a^{n-1}, (\mathrm{R}, \mathrm{Y})) = (1 - r)P(.|\mathbf{r}_a^{n-1}, \mathrm{R}) + rP(.|\mathbf{r}_a^{n-1}, \mathrm{Y})$$

- As the active user rates more and more products, his ratings will tend to be distinguished as sampled from $P(.|\mathbf{r}_a^{n-1}, \mathrm{R})$ and hence the influence of $P(.|\mathbf{r}_a^{n-1}, \mathrm{Y})$ diminishes as n grows.

Yan and Roy (2009)'s Manipulation Robust Algorithm



(b) $r = 0.3$

- The kNN algorithm is *not* linear and does not satisfy the bound
- The Naive Bayes (NB) algorithm is an *asymptotically* linear CF algorithm.
- Kernel Density Estimation (KDE) is a linear CF algorithm.

Outline

Robustness and Related Concepts

- **Stability** refers to a recommender system's ability to provide consistent recommendations

Robustness and Related Concepts

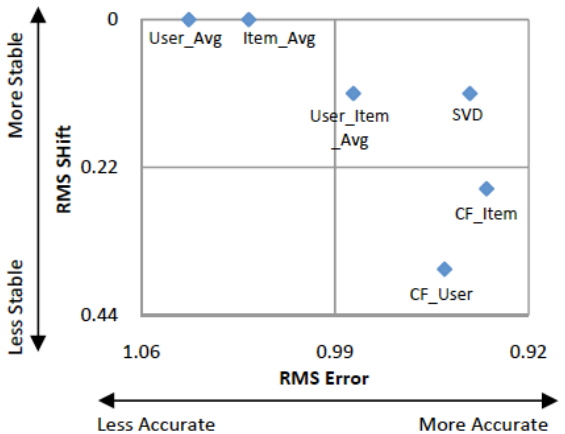
- **Stability** refers to a recommender system's ability to provide consistent recommendations
  - in the face of noise in the dataset.

Robustness and Related Concepts

- **Stability** refers to a recommender system's ability to provide consistent recommendations
  - in the face of noise in the dataset.
  - using different training sample;

Robustness and Related Concepts

- **Stability** refers to a recommender system's ability to provide consistent recommendations
  - in the face of noise in the dataset.
  - using different training sample;
  - over time, if new ratings 'agree' with past ratings
- Adomavicius and Zhang (2010), explores stability from the point of view of adding a CF algorithm's predictions to the dataset as new ratings, and measuring the prediction shift that is incurred.



Robustness and Related Concepts

- Amatriain et al. (2009) carries out a user study of *test re-test reliability* and shows that inconsistencies negatively impact the quality of the predictions.

Robustness and Related Concepts

- **Trust** has been widely explored in collaborative filtering

Robustness and Related Concepts

- **Trust** has been widely explored in collaborative filtering
  - . . . avoid manipulation by only using rating of trusted users or build trust based on user behaviour (c.f. the Influence Limiter)

Robustness and Related Concepts

- **Trust** has been widely explored in collaborative filtering
  - . . . avoid manipulation by only using rating of trusted users or build trust based on user behaviour (c.f. the Influence Limiter)
  - . . . nevertheless systems based on trust are still manipulatable – `tinyurl.com/6z6k6kr` two fictitious Facebook users successfully made friends with 95 users within two weeks.

Robustness and Related Concepts

- **Privacy** is an important security concern from the point-of-view of the end-user

Robustness and Related Concepts

- **Privacy** is an important security concern from the point-of-view of the end-user
  - In Cheng and Hurley (2009b), we demonstrate that a system architecture to support privacy can provide new opportunities for manipulation attacks.

Robustness and Related Concepts

- **Privacy** is an important security concern from the point-of-view of the end-user
  - In Cheng and Hurley (2009b), we demonstrate that a system architecture to support privacy can provide new opportunities for manipulation attacks.
  - *Differential privacy* has been studied in the context of recommender systems by a number of researchers. One approach to differential privacy is the use of robust statistics. The connection to manipulation resistance may be worth pursuing.

More complicated shilling scenarios

- We have focused in this tutorial on rating manipulation – the creation of sybil profiles and ratings that distort a recommendation system's output.
- However, in the real world, shilling can have a more complicated form.
  - e.g. The text of hotel reviews can be used to persuade users to select certain hotels.
  - Wu et al. (2010) has carried out work on identifying suspicious reviews in TripAdvisor.

Conclusion

- We've reviewed research that has been carried out in last number of years on robustness of RS.
- The conclusions are quite positive from system managers' points-of-view:
  - If desired, recommendation algorithms that are largely manipulation-resistant may be adopted.
  - Filtering strategies can effectively find unusual rating patterns.
  - Obfuscating attack profiles to avoid filtering generally results in less effective attacks.
- Good recommendation systems are personalised and hence, *should* be sensitive to the peculiarities of each user's rating behaviour.
- A system designer needs to find the right balance between sensitivity to the melting pot of human behaviour and avoiding easy manipulation.

Thank You

My research is sponsored by Science Foundation Ireland under grant 08/SRC/I1407: Clique: Graph and Network Analysis Cluster

References I

Adomavicius, G. and Zhang, J.: 2010, On the stability of recommendation algorithms, *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, ACM, New York, NY, USA, pp. 47–54.
URL: *http://doi.acm.org/10.1145/1864708.1864722*

Amatriain, X., Pujol, J. M. and Oliver, N.: 2009, I like it . . . I like it not: Evaluating user ratings noise in recommender systems, *Proceedings of the 17th International Conference on User Modeling, Adaptation, and Personalization: formerly UM and AH*, UMAP '09, Springer-Verlag, Berlin, Heidelberg, pp. 247–258.
URL: *http://dx.doi.org/10.1007/978-3-642-02247-0_24*

References II

Burke, R., O'Mahony, M. P. and Hurley, N. J.: 2011, Robust collaborative recommendation, *in* F. Ricci, L. Rokach, B. Shapira and P. B. Kantor (eds), *Recommender Systems Handbook*, Springer, pp. 805–835.

Carrara, E. and Hogben, G.: 2007, Enisa position paper no. 2,reputation-based systems: a security analysis, *Technical report*, ENISA.

Cheng, Z. and Hurley, N.: 2009a, Effective diverse and obfuscated attacks on model-based recommender systems, *in* L. D. Bergman, A. Tuzhilin, R. D. Burke, A. Felfernig and L. Schmidt-Thieme (eds), *RecSys*, ACM, pp. 141–148.

References III

Cheng, Z. and Hurley, N.: 2009b, Trading robustness for privacy in decentralized recommender systems, *in* K. Z. Haigh and N. Rychtyckyj (eds), *IAAI*, AAAI.

Cheng, Z. and Hurley, N.: 2010, Robust collaborative filtering by least trimmed squares matrix factorisation, *Proceedings of the International Conference on Tools in Artificial Intelligence*.

Donath, J.: 1998, *Communities in Cyberspace*, Routledge, chapter Identity and Deception in the Virtual Community.

Douceur, J.: 2002, The sybil attack, *Proceedings of the First International Workshop on Peer-to-Peer Systems*.

Lam, S. K. and Riedl, J.: 2004, Shilling recommender systems for fun and profit, *In Proceedings of the 13th International World Wide Web Conference* pp. 393–402.

References IV

Lang, J., Spear, M. and Wu, S. F.: 2010, Social manipulation of online recommender systems, *Proceedings of the Second international conference on Social informatics*, SocInfo'10, Springer-Verlag, Berlin, Heidelberg, pp. 125–139.
URL: *http://dl.acm.org/citation.cfm?id=1929326.1929336*

Mehta, B., Hofmann, T. and Fankhauser, P.: 2007, Lies and propaganda: Detecting spam users in collaborative filtering, *Proceedings of the 12th international conference on Intelligent user interfaces*, pp. 14–21.

Mehta, B. and Nejdl, W.: 2008, Attack resistant collaborative filtering, *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, ACM, New York, NY, USA, pp. 75–82.
URL: *http://doi.acm.org/10.1145/1390334.1390350*

References V

Mobasher, B., Burke, R., Bhaumik, R. and Williams, C.: 2007, Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness, *ACM Transactions on Internet Technology* **7**(4).

Mobasher, B., Burke, R. D. and Sandvig, J. J.: 2006, Model-based collaborative filtering as a defense against profile injection attacks, *AAAI*, AAAI Press.

O'Mahony, M. P., Hurley, N. J. and Silvestre, C. C. M.: 2004, An evaluation of neighbourhood formation on the performance of collaborative filtering, *Artificial Intelligence Review* **21**(1), 215–228.

References VI

O'Mahony, M. P., Hurley, N. J. and Silvestre, G. C. M.: 2002, Promoting recommendations: An attack on collaborative filtering, *in* A. Hameurlain, R. Cicchetti and R. Traunmüller (eds), *DEXA*, Vol. 2453 of *Lecture Notes in Computer Science*, Springer, pp. 494–503.

O'Mahony, M. P., Hurley, N. J. and Silvestre, G. C. M.: 2006, Attacking recommender systems: The cost of promotion, *Workshop on Recommender Systems at the 17th European Conference on Artificial Intelligence (ECAI'06), 28th–29th*, Riva del Garda, Italy.

References VII

Resnick, P. and Sami, R.: 2007, The influence limiter: provably manipulation-resistant recommender systems, *RecSys '07: Proceedings of the 2007 ACM conference on Recommender systems*, ACM, New York, NY, USA, pp. 25–32.

Vu, L.-H., Papaioannou, T. and Aberer, K.: 2010, Impact of trust management and information sharing to adversarial cost in ranking systems, *in* M. Nishigaki, A. Jsang, Y. Murayama and S. Marsh (eds), *Trust Management IV*, Vol. 321 of *IFIP Advances in Information and Communication Technology*, Springer Boston, pp. 108–124.

References VIII

Williams, C., Mobasher, B., Burke, R., Bhaumik, R. and Sandvig, J.: 2006, Detection of obfuscated attacks in collaborative recommender systems, *In Proceedings of the 17th European Conference on Artificial Intelligence (ECAI'06)* .

Wu, G., Greene, D. and Cunningham, P.: 2010, Merging multiple criteria to identify suspicious reviews, *in* X. Amatriain, M. Torrens, P. Resnick and M. Zanker (eds), *RecSys*, ACM, pp. 241–244.

Yan, X. and Roy, B. V.: 2009, Manipulation robustness of collaborative filtering systems, *CoRR* **abs/0903.0064**.