

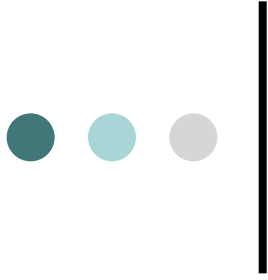


Recommender Systems & Collective Intelligence

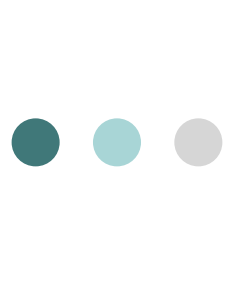
COMP47580

Dr. Michael O'Mahony

michael.omahony@ucd.ie



Trust in Recommender Systems

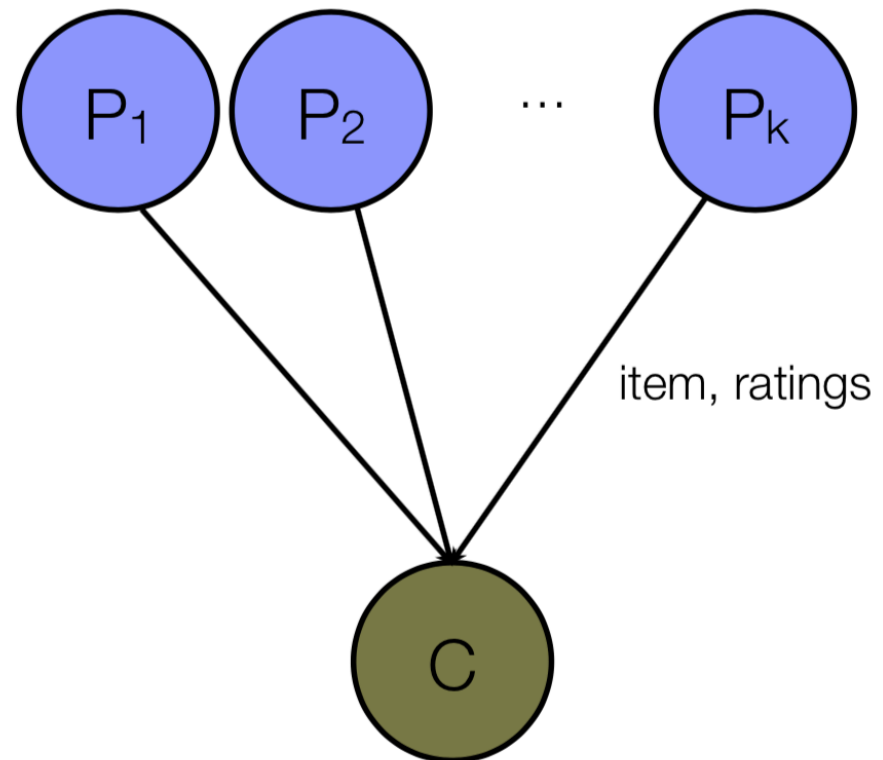


From Similarity to Trust

- Classical collaborative filtering approaches select recommendation partners (neighbours) on the basis of similarity.
- A wide variety of similarity metrics have been considered: Pearson correlation, cosine similarity, distance measures, etc...
- Is similarity enough? It makes sense for our recommendation partners to have similar tastes and preferences but ...
- ... intuitively it makes sense to also look for partners that are likely to be trustworthy.
- Can we incorporate a measure of trust into collaborative filtering style recommender systems?

Computational Models of Trust

- Consider two types of users in a recommendation session.
- Producer(s):
 - A profile that has been selected as a recommendation partner (neighbour) and that will participate in the recommendation session.
- Consumer:
 - The profile that is receiving the item rating from the producer profiles.





Calculating Binary Trust Scores

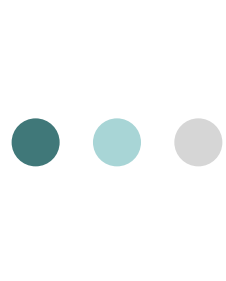
Objective is to calculate the trustworthiness of producers (neighbours) using a computational trust modeling approach

A prediction for item, i , by a (sole) producer p for a consumer c , is *correct* if the predicted rating, $p(i)$, is within ϵ of c 's actual rating $c(i)$

$$\textit{Correct}(i, p, c) \Leftrightarrow |p(i) - c(i)| < \epsilon$$

Calculate the trust score for a producer p when predicting item i for consumer c as a binary value, indicating whether or not the prediction of p for item i was correct for c .

$$T_p(i, c) = \textit{Correct}(i, p, c)$$



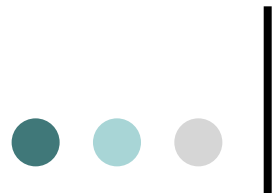
Trust Metrics Based on Ratings Data

RecSet is the set of recommendation sessions that a producer p has participated in ... that is the consumer-item pairs for which ratings have been predicted.

$$RecSet(p) = \{(c_1, i_1), \dots, (c_n, i_n)\}$$

The *CorrectSet* is the subset of these recommendation sessions for which correct predictions have been generated.

$$CorrectSet(p) = \{(c_k, i_k) \in RecSet(p) : Correct(i_k, p, c_k)\}$$



Profile Level Trust

- The simplest trust metric – *profile-level trust* – is calculated as the percentage of overall recommendation sessions in which the producer has correctly participated.

$$Trust^P(p) = \frac{|CorrectSet(p)|}{|RecSet(p)|}$$

- For example, if a producer has contributed to 100 recommendation sessions and their contributions have been correct in 40 of these sessions, then their profile-level trust will be 0.4.
- Coarse-grained trust metric \Rightarrow some users may be better at making predictions for certain items, but not for others, which will not be captured by this profile-level metric.



Item Level Trust

- Item-level trust offers a more fine-grained trust metric.
- It measures the percentage of p 's predictions for a particular item i that have proven to be correct.

$$Trust^I(p, i) = \frac{|\{(c_k, i_k) \in CorrectSet(p) : i_k = i\}|}{|\{(c_k, i_k) \in RecSet(p) : i_k = i\}|}$$

- For example, if user p has been involved in 20 predictions for item i and only 5 of these have been correct, then p 's item-level trust will be 0.25.



Adding Trust to Recommendation

- We have a way to calculate the trust-worthiness of individual users (overall and at the item level)... The next question is how we can incorporate this into the the recommendation process.
- One approach is to adapt *Resnick's* standard user-based CF prediction formula.
- Consider two adaptations – *trust-based weighting* and *trust-based filtering* – which can be used with either profile-level or item-level trust metrics.
- Each involves an adaptation to *Resnick's* prediction formula and both adaptations can be combined.

Trust-based Weighting

- The idea with trust-based weighting is to combine trust and similarity at prediction time.

$$c(i) = \bar{c} + \frac{\sum_{p \in P(i)} (p(i) - \bar{p})w(c, p, i)}{\sum_{p \in P(i)} |w(c, p, i)|}$$

Standard Resnick

*New weighting
formula*

$$w(c, p, i) = \frac{2(sim(c, p))(trust^I(p, i))}{sim(c, p) + trust^I(p, i)}$$

Harmonic mean of similarity & trust.

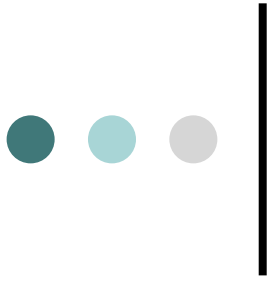


Trust-based Filtering

- The idea is to eliminate untrustworthy neighbours from consideration at prediction time.
- Filter (remove) neighbours with a trust score $< T$.

$$c(i) = \bar{c} + \frac{\sum_{p \in P^T(i)} (p(i) - \bar{p}) \text{sim}(c, p)}{\sum_{p \in P^T(i)} |\text{sim}(c, p)|}$$

$$P_i^T = \{p \in P(i) : \text{Trust}^I(p, i) > T\}$$



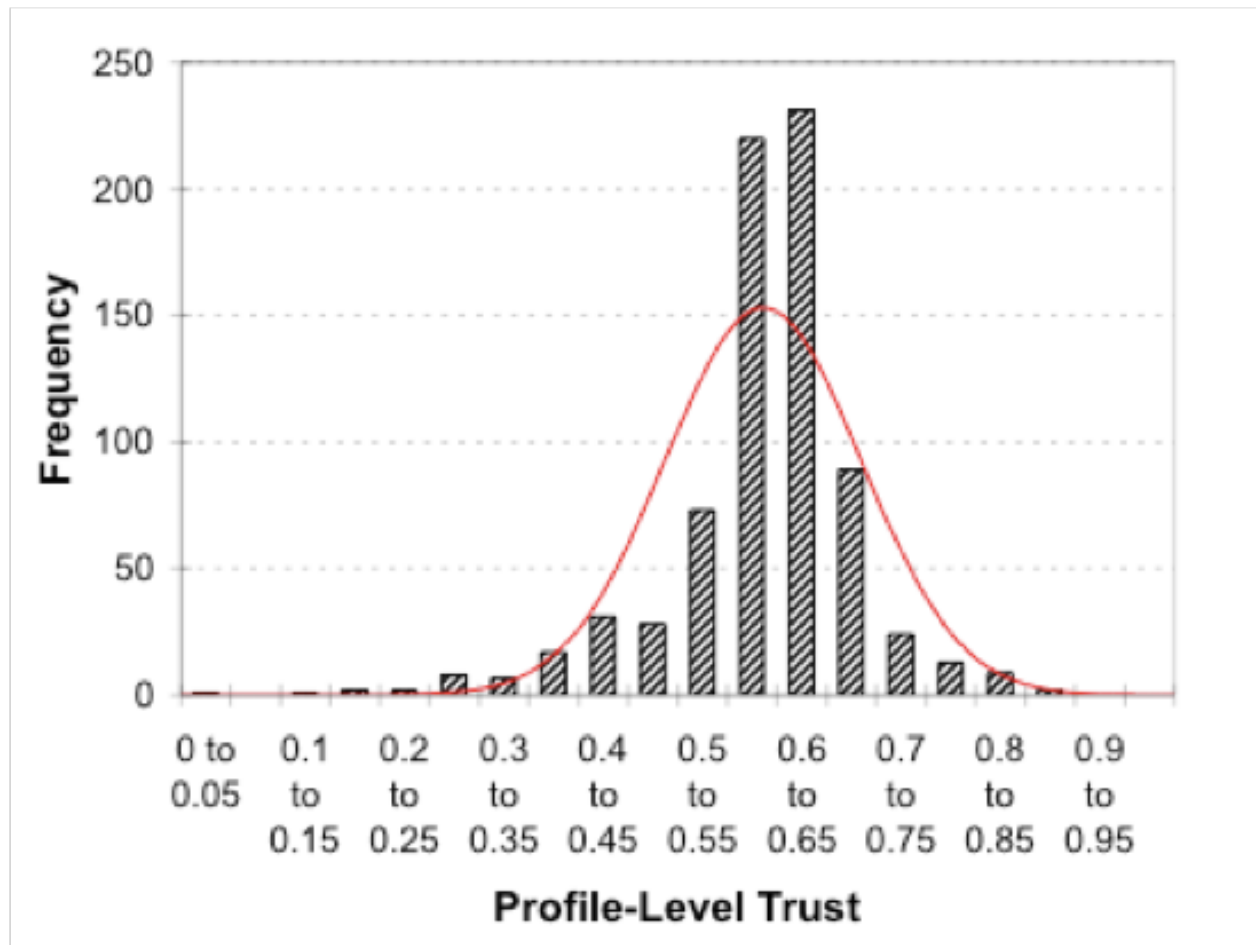
Does it work?



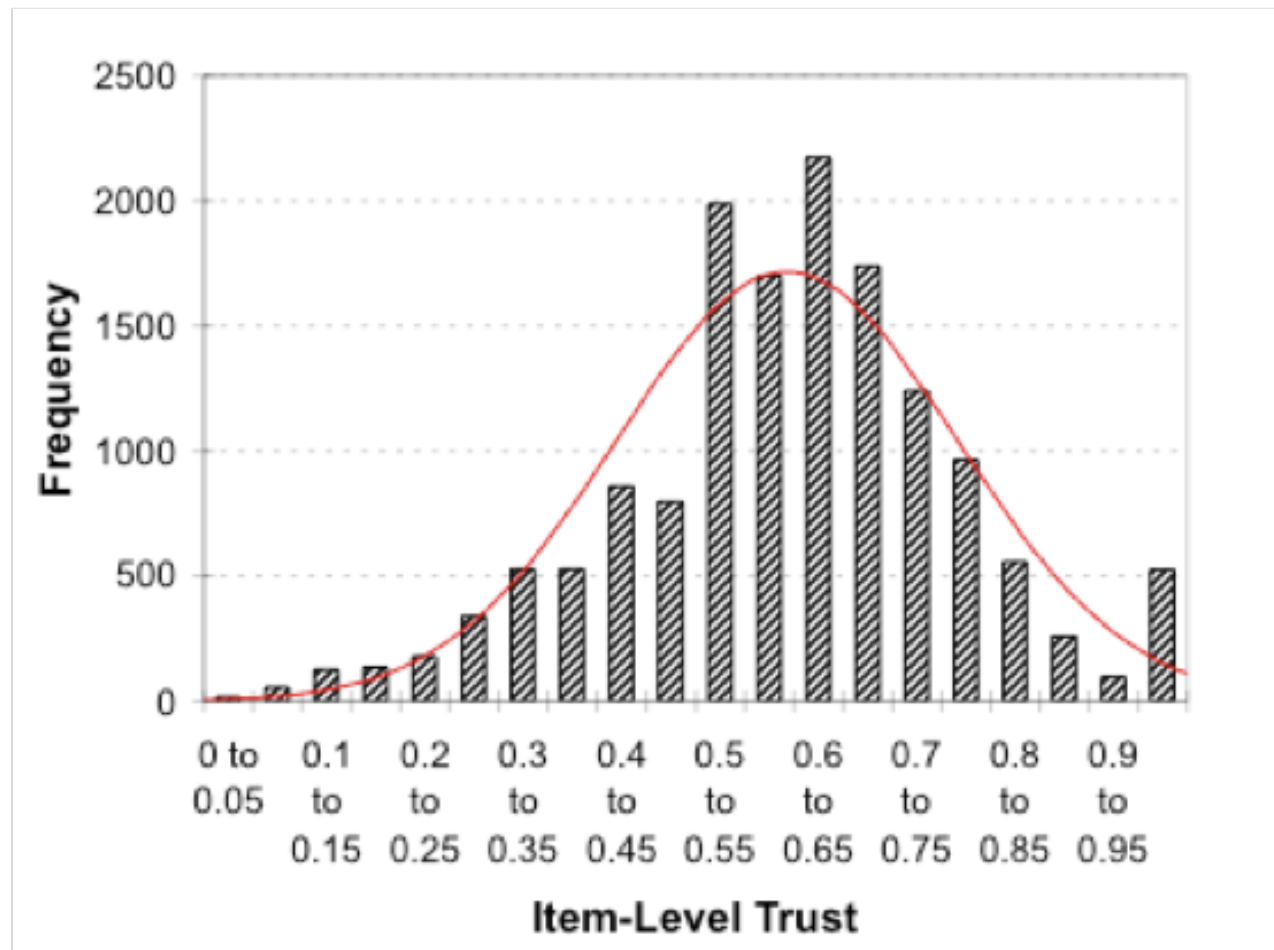
Evaluation Methodology

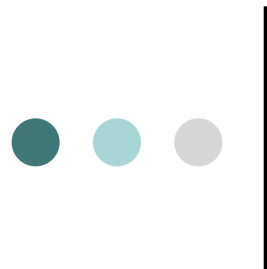
- MovieLens dataset (943 users, 1,683 movies, 100k ratings)
- 80% of profiles used as training set; 20% as test set.
- First need to calculate the trust scores based on training profiles:
 - Each profile acts as a consumer in turn
 - For a given consumer, find the set of neighbours
 - Each neighbour acts as a sole producer in turn and compute trust scores for each producer as described above
- First examine distributions of profile-level and item-level trust scores...

Distribution of Profile-Level Trust Scores



Distribution of Item-Level Trust Scores



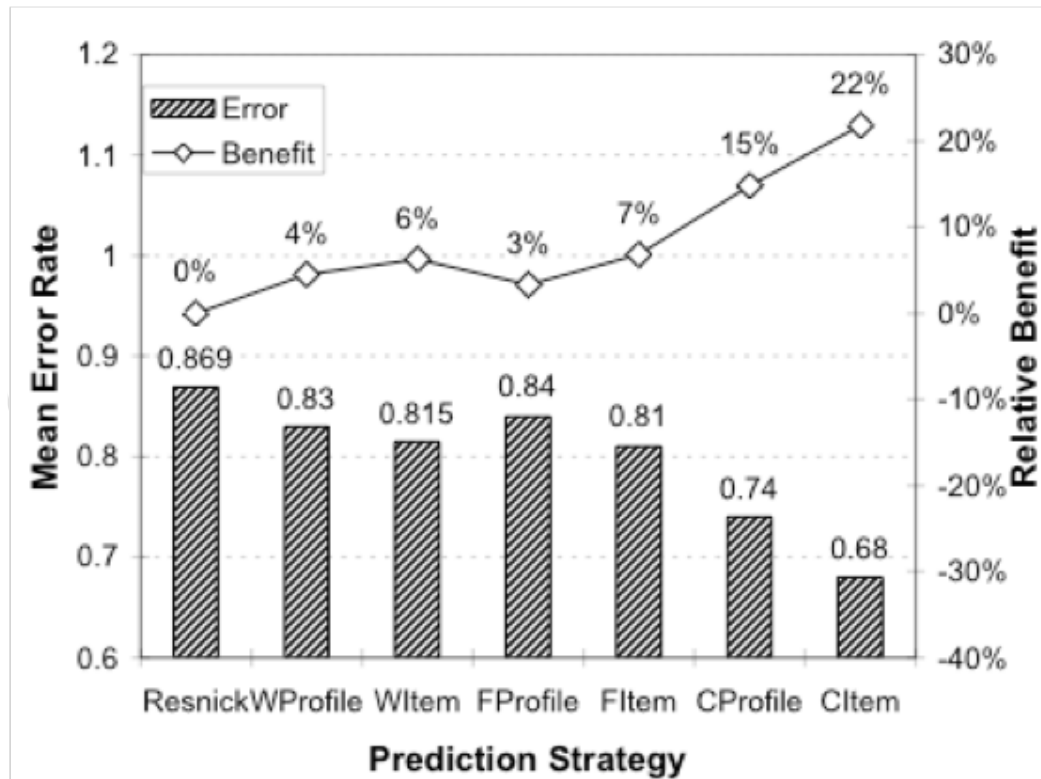


Recommendation Strategies

Use standard training/test approach to generate and evaluate predictions for test set profiles according to the following recommendation strategies...

- *Resnick* – Standard (no trust) user-based collaborative filtering.
- *WProfile* – Trust-based weighting using profile-level trust.
- *WItem* – Trust-based weighting using item-level trust.
- *FProfile* – Trust-based filtering using profile-level trust.
- *FItem* – Trust-based filtering using item-level trust.
- *CProfile* – Combination of trust-based weighting and filtering using profile-level trust.
- *CItem* – Combination of trust-based weighting and filtering using item-level trust.

Results

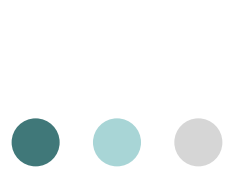


Item-level trust approaches outperform the profile-level approaches.

Expected as item-level trust provides a more fine-grained measure of the reliability of a profile as a predictor - individual profiles may be trustworthy when it comes to predicting the ratings of some items, but less so for others.

Combined strategies outperform the individual weighting and filtering strategies.

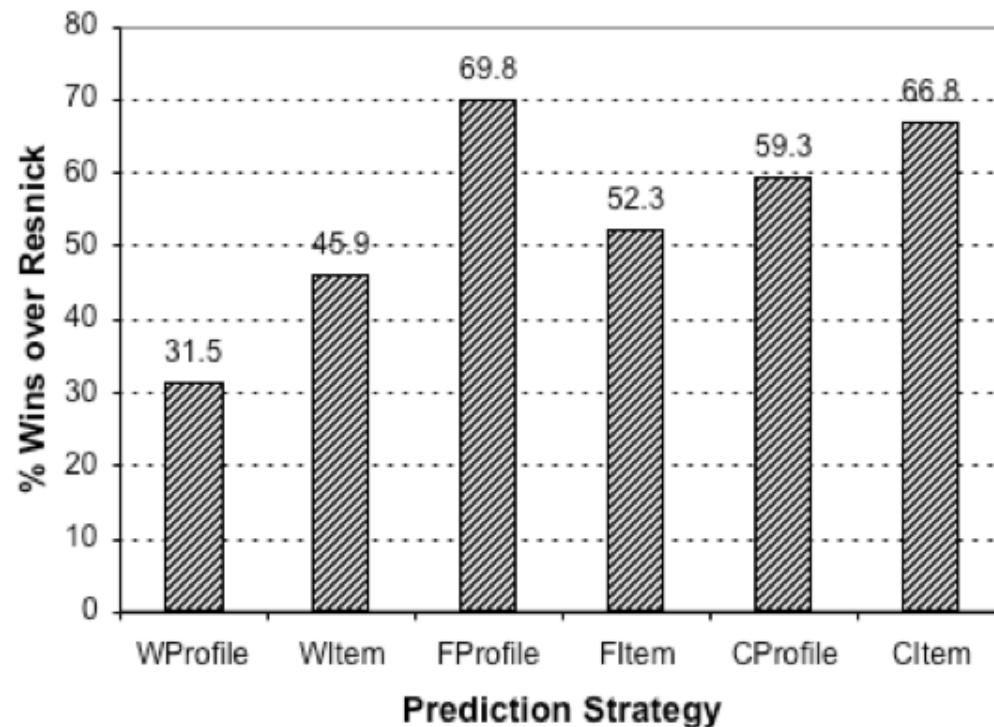
The item-level combination outperforms the profile-level combination.



Recommendation Strategies

- Trust-based predictions techniques achieve a lower overall error compared to standard collaborative filtering (*Resnick*).
- However, it is not clear whether these reduced errors arise out of a general improvement by the trust-based techniques compared to *Resnick*, over many individual predictions, or whether the reduction in error arises from a small number of very accurate predictions.
- Examine the percentage of predictions where each of the trust-based methods wins over *Resnick* – in the sense that they achieve lower error predictions on a prediction by prediction basis.

Wins over Resnick



WProfile and *WItem* have lower error compared than Resnick, but only win in ~32% and ~46% of prediction trials – i.e. *Resnick* provides better predictions in the majority of cases.

The filter-based (*FProfile* and *FItem*) and combination (*CProfile* and *CItem*) strategies offer much better performance, winning on the majority of trials; e.g. *FProfile* and *CItem* win in ~70% and ~67% of prediction trials.

FProfile provides best overall improvement in terms of wins (~70%), but offers only a 3% mean error reduction compared to *Resnick* – improvements are relatively minor.

CItem beats *Resnick* ~67% of the time, but with a much higher error reduction of 22%.

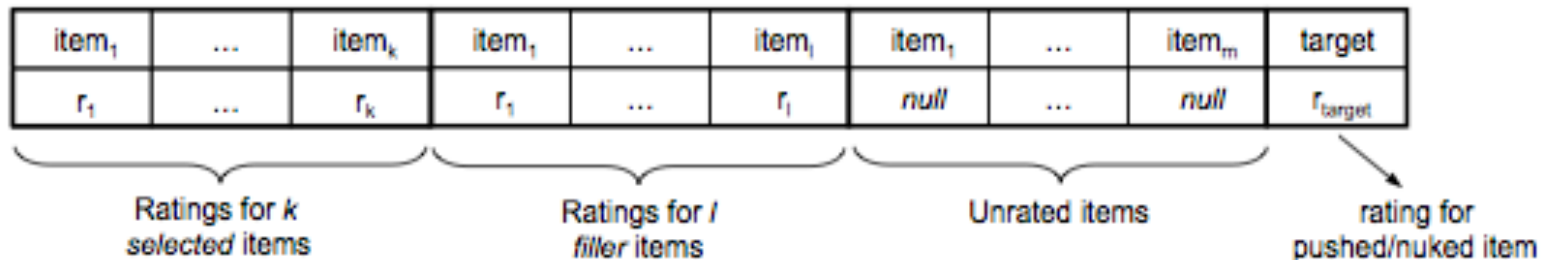


Discussion

- Beyond similarity in collaborative filtering.
- Introduced computational models of trust.
- Incorporated trust into the classical recommendation process.
- Leads to reduced prediction errors...
- ... but what about robustness against malicious attack?

Attack Models – Review

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



- Attack profiles consist of the following items:
 - The *target item* (I_T) – assigned a rating of r_{max} or r_{min} (push/nuke attack).
 - Filler items* (I_F) – randomly chosen from those available.
 - Selected items* (I_S) – that have some association with the target item.
- Here – focus on the *average attack*:
 - Target item (I_T) – r_{max} (for push attack).
 - Filler items (I_F) – a randomly selected set of items with each item assigned a rating distributed around the mean of all genuine ratings assigned to that item.
 - Selected items (I_S) – none.

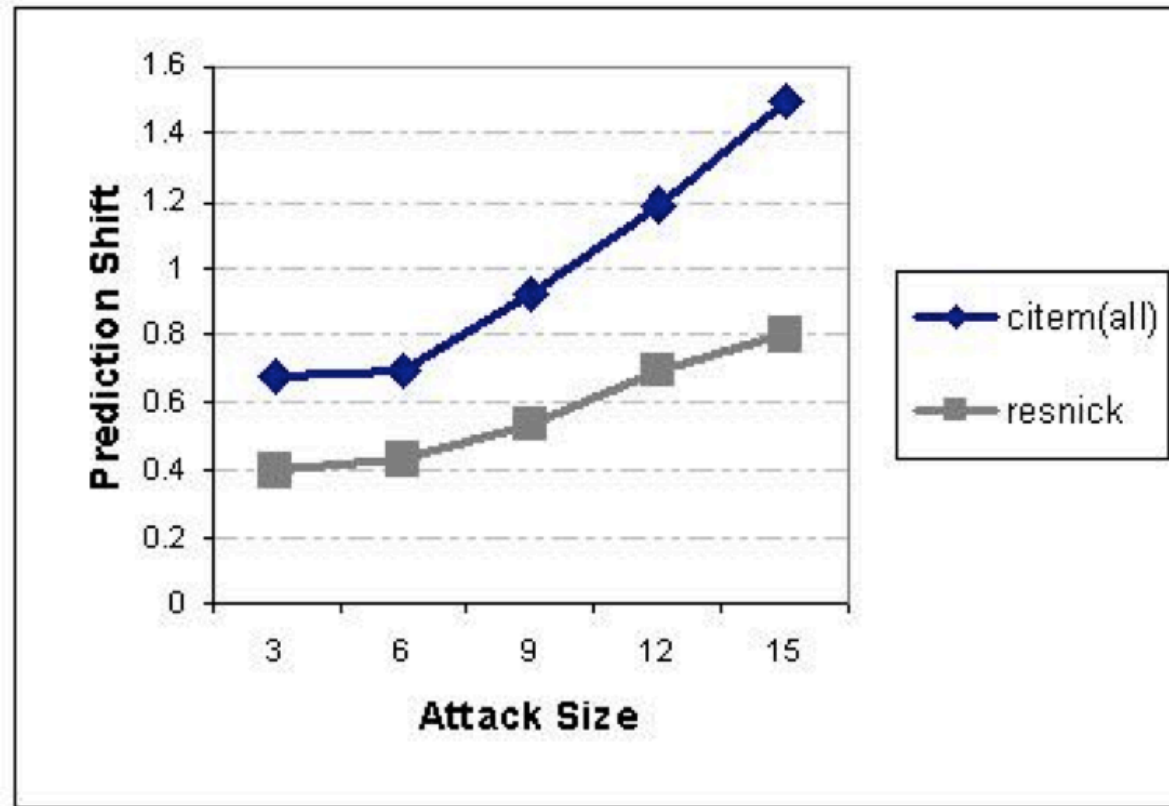


Average Attack – *Cltem*

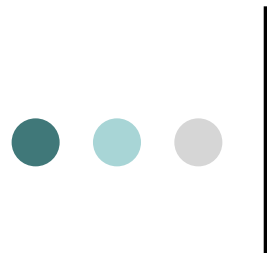
- Focus on *Cltem* (combination of trust-based weighting and filtering using item-level trust) since this approach provides good overall performance.
- Is *Cltem* robust to attacks?
- Evaluation:
 - Dataset: MovieLens (943 users, 1,683 movies, 100k ratings).
 - Attack size: number of attack profiles created as a percentage of the number of genuine profiles in the system (1% attack size ~ 10 profiles).
 - Target item: “Toy Story”
 - Metric: prediction shift – difference between pre- and post-attack predictions for target item:

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

Average Attack – *Cltem*

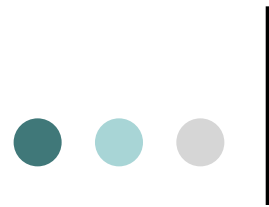


- *Citem* – higher prediction shift compared to standard user-based CF algorithm (*Resnick*)... But *Cltem* is a trust-based algorithm! Why??



The Reinforcement Problem...

- Consider a scenario where 10 push attack profiles (p_1, p_2, \dots, p_{10}) are created:
 - Each targets the same item, which is assigned the maximum rating in all attack profiles.
- When trust scores are calculated, each attack profile will “reinforce” the trust scores of the other attack profiles:
 - For example, p_1 will generate predictions (as the sole producer profile) for the nine other attack profiles p_2, \dots, p_{10} .
 - These predictions will have low errors since all the attack profiles have the same rating for the target item.
 - Hence attack profiles receive higher trust scores compared to genuine profiles!



Recall – Calculating Trust Scores...

- Partition dataset in training and test profiles.
- Calculate the trust scores based on training profiles:
 - Each profile acts as a consumer in turn
 - For a given consumer, find the set of neighbours
 - Each neighbour acts as a sole producer in turn and compute trust scores for each producer as described above



Consumer Selection Strategies

- The reinforcement problem in the trust modeling process can be solved if the profiles involved in this process are filtered:
 - For example, if attack profiles do not serve as consumers in the trust modeling process, then they cannot reinforce the trust scores of other attack profiles.
 - But the identity of attack profiles is not known...
- Adopt the following consumer selection strategies for trust modeling:
 - *CItem (all)* - the basic *CItem* approach described above in which all profiles in the training data are allowed to serve as consumer profiles.
 - *CItem (diverse)* - selects a diverse set (100) of profiles to serve as consumers.
 - *CItem (time)* - selects 100 older profiles to serve as consumers.
 - *CItem (random)* - randomly selects 100 profiles to serve as consumers.
 - *CItem (genuine)* - selects 100 known genuine profiles to serve as consumers; the ideal scenario in which none of the attack profiles serve as consumers (but attack profiles can serve as producers in the trust modeling process).

Results – Prediction Shift

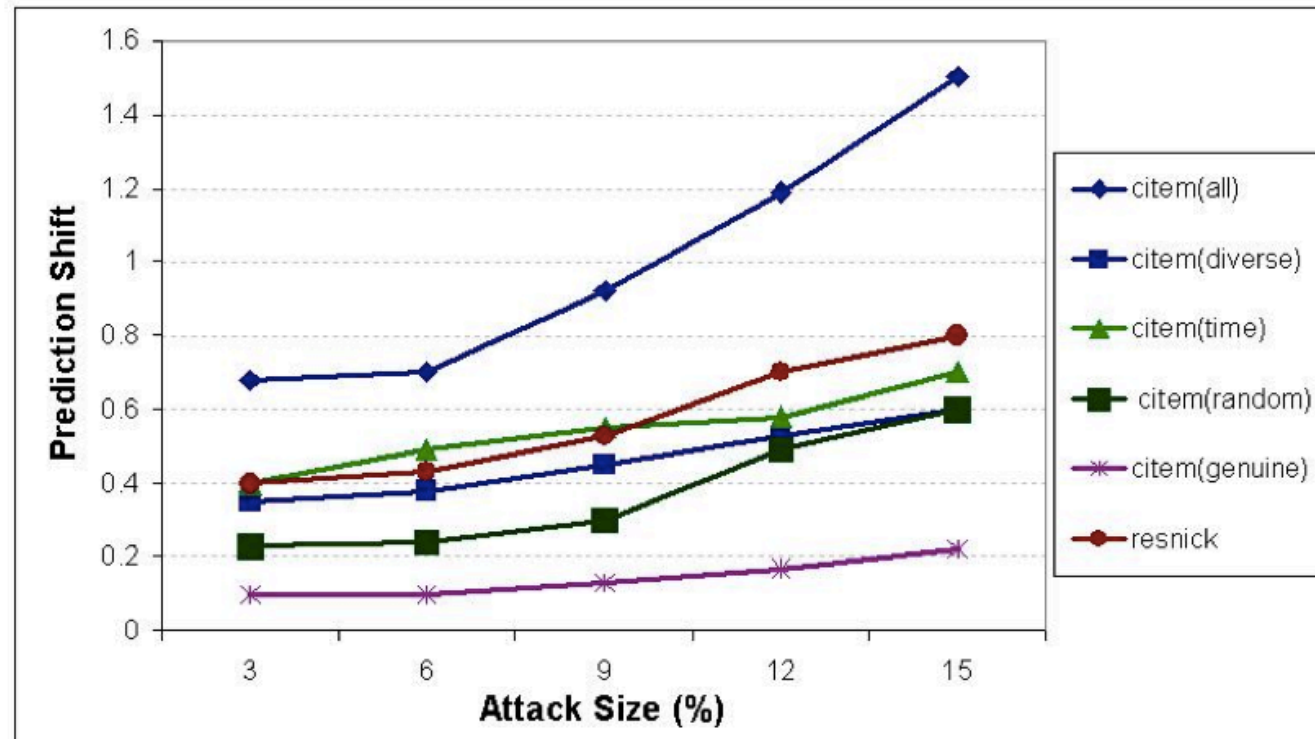


Figure 4: The average prediction shift for pushed item “toy story” (compared to Resnick) of each of the trust-based recommendation strategies, with varying attack sizes.

Results – Prediction Error (MAE) – No Attack

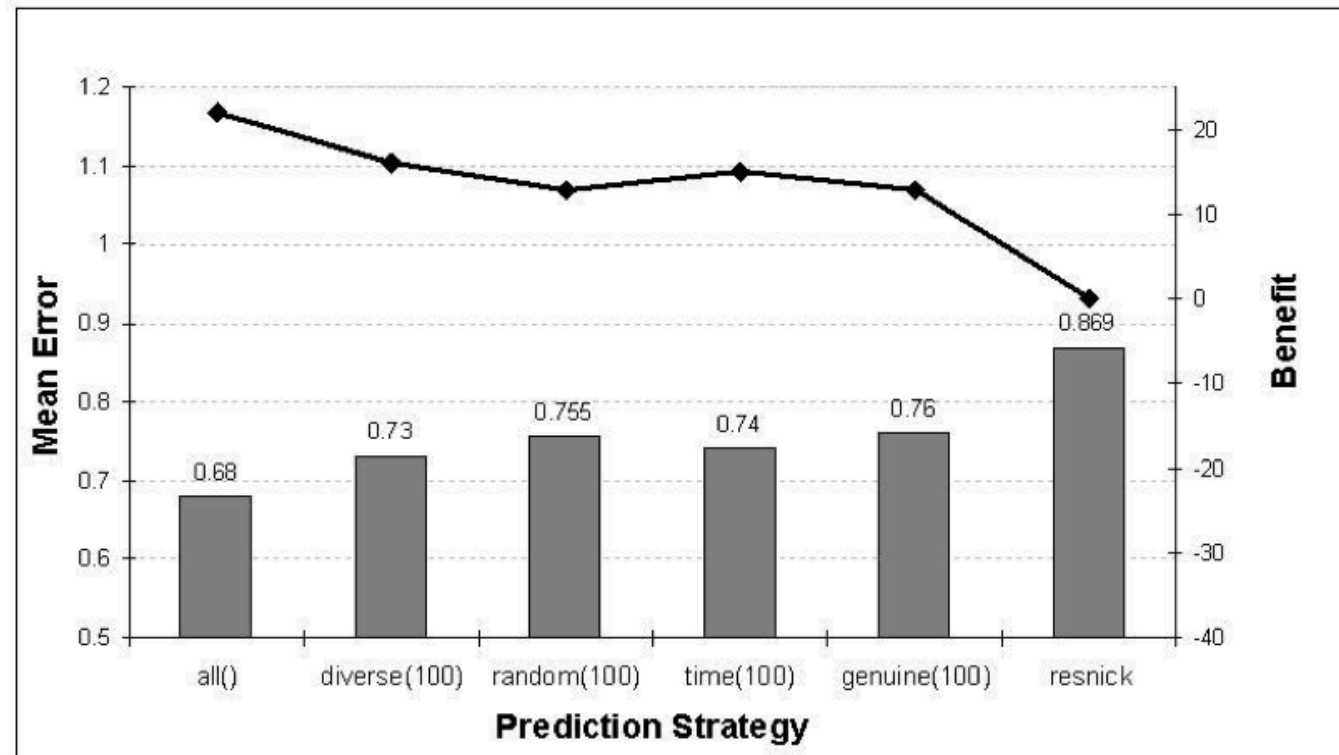
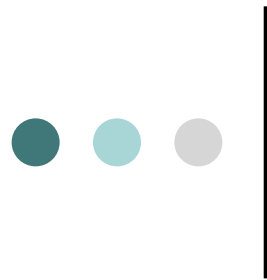


Figure 6: The average prediction error and relative benefit (compared to Resnick) of each of the *CItem* algorithm using different consumer selection strategies.



Conclusions

- Beyond similarity in collaborative filtering.
- Introduced computational models of trust.
- Incorporated trust into the classical recommendation process.
- Leads to reduced prediction errors.
- Provides robustness against malicious attack.
- Readings:
 - [John O'Donovan](#), Barry Smyth. “Trust in Recommender Systems”. Proceedings of the 10th International Conference on Intelligent User Interfaces, 2005.
 - [John O'Donovan](#), Barry Smyth. “Is Trust Robust? An Analysis of Trust-based Recommendation”. Proceedings of the 11th International Conference on Intelligent User Interfaces, 2006.