Lecture #16:
Recommender Systems

Prof. Dr. Sven Seuken
10.5.2012

Housekeeping

- Questions? Concerns?
- Last lecture (May 31st): review session
  → please prepare questions!
- Evaluation results
  → will discuss in last lecture

Today: Recommender Systems

- Pandora
- Last.fm
- Netflix…

Reputation vs. Recommender Systems

- Both aggregate user feedback and share it back
- Reputation systems:
  - One true type/quality
  - Objective
  - Usually: same ranking/recommendation for every user
  - Rarely “personalized”…
- Recommender Systems:
  - Different users have different tastes
  - Subjective preferences → not one objective underlying quality
  - Always: personalized recommendations

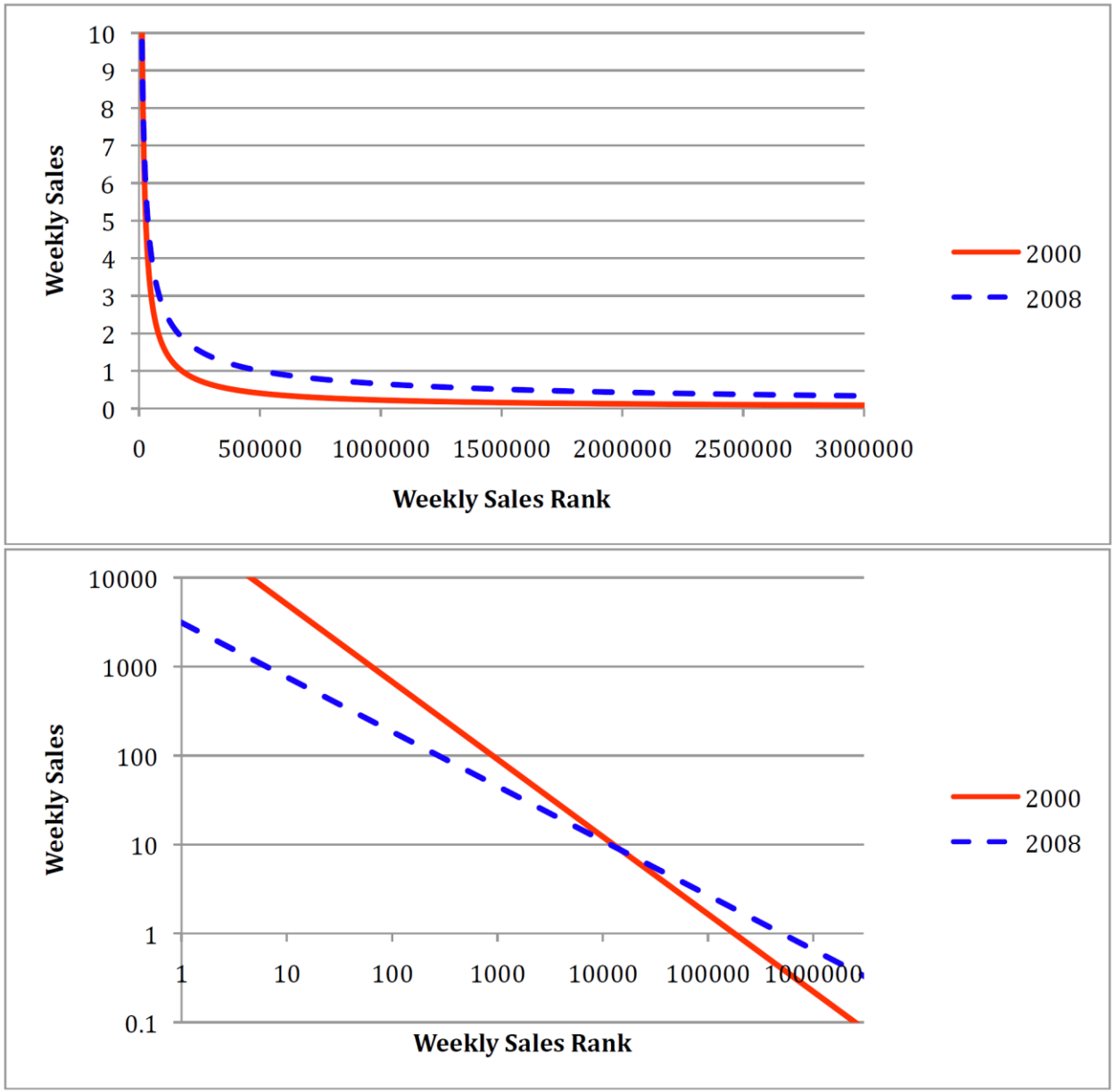
Information Use

- Explicit Ratings
- Implicit Ratings
- User Profiles
- …

Amazon

- "Long tail" of content (power law distribution)
  - probability of an object purchased in a week by x people falls off proportional to x^{-c} not 2^{-x}
- Recommender: sell a lot of niche products

Figure 1: Amazon's Long Tail in 2008 (our sample) vs. in 2000 (BHS sample)



(Brynjolfsson et al. 2010)

Recommender Systems and Problems

- Last.fm (Alexander, Balthasar, Robert)
- Amazon (Andrea)
- Stumbleupon (Basil)
- Ebay (Evgeny)
- Youtube, Booking.com (Jessica)
- Pandora (Nico)
- Spotify (Khoa)

New Recommender Systems?

- Lectures at UZH (Andrea, Balz, Evgeny, Khoa)
- International News (Mengia)
- Recipes (Andrin)
- Food shopping (Basil, Martin)
- Wines (Andras)
- Bars in Zurich (Jan)

A Formal Model

- $N = \{1, \ldots, n\}$ users
- $G = \{1, \ldots, m\}$ items
- $r_i(j) =$ user i's "true" rating for item j
- $\hat{r}_i(j) =$ the system's estimate for user i's rating for item j

Design Goals?

- Primary goal: recommend items that the user will like
- Secondary goal: produce good estimates $\hat{r}_i(j)$, i.e., as close to $\hat{r}_i(j)$ as possible
- Measure: RMSE (root-mean-squared-error) = RMSD (root-mean-squared-deviation)

$$\mathrm{RMSD}(\theta_1, \theta_2) = \sqrt{\mathrm{MSE}(\theta_1, \theta_2)} = \sqrt{\mathrm{E}((\theta_1 - \theta_2)^2)} = \sqrt{\frac{\sum_{i=1}^{n} (x_{1,i} - x_{2,i})^2}{n}}.$$

- This ignores:
  - Recomming very well-known items
  - Overspecialization
  - …

3 Approaches

- Content-based Methods
- Collaborative Filtering
  - User-based
  - Item-based
- Hybrid Approaches

Content-Based Methods

- $w_j = \left(w_{j1}, w_{j2}, \ldots, w_{jK}\right) \in [0,1]^K$ = content of item j (with K possible keywords)
- $w_i = (w_{i1}, w_{i2}, \ldots, w_{iK}) \in [0,1]^K$ = average rating of user i on keywords 1…K
- Use w_j and w_i to estimate $\hat{r}_i(j)$
- First idea: compute similarity between w_j and w_i!

Similarity Heuristic

$$\hat{r}_i(j) = cos(w_i, w_j) = \frac{w_i \cdot w_j}{||w_i||_2 ||w_j||_2} = \frac{\sum_{k=1}^{K} w_{ik} w_{jk}}{\sqrt{\sum_{k=1}^{K} w_{ik}^2} \sqrt{\sum_{k=1}^{K} w_{jk}^2}}$$

Problems of Content-Based Method

- Limited Distinction (granularity of the features/keywords) → difficult to fix
- Over-Specialization → add randomness?
- New-user problem

Collaborative Filtering

- Main ideas:
  - Do not rely on manually entered keywords → ratings without meaning (no content)
  - Use all users' ratings to compute $\hat{r}_i(j)$
  - If users have rated items similarly, those users probably have similar taste (user-based CF)
  - If items have rated similarly from users, those items are probably similar (item-based CF)

User-Based Collaborative Filtering

- Compute similarity between users i and i'
- $\overline{r_i}$ = average rating user i has provided so far
- G'_{ii} = set of items which both i and i' have rated
- *Pearson Correlation Coefficient:*

$$sim(i,i') = \frac{\sum_{j \in G_{ii'}} (r_i(j) - \overline{r}_i)(r_{i'}(j) - \overline{r}_{i'})}{\sqrt{\sum_{j \in G_{ii'}} (r_i(j) - \overline{r}_i)^2} \sqrt{\sum_{j \in G_{ii'}} (r_{i'}(j) - \overline{r}_{i'})^2}}$$

- Note: relation to cosine-similarity!

User-Based CF (continued)

- Given a method to compute similarity $sim(i, i')$

  → compute neighborhood N_i, i.e., those users most similar to i

- Neighborhood size → fine-tuning
- In practice: 20-50…

User-Based CF (continued)

- Given neighborhood N_i:
  - Simple averaging: $\hat{r}_i(j) = \frac{1}{|N_i|} \sum_{i' \in N_i} r_{i'}(j)$
  - Weighted-sum aggregation

$$\hat{r}_i(j) = \frac{\sum_{i' \in N_i} sim(i, i') r_{i'}(j)}{\sum_{i' \in N_i} sim(i, i')}$$

  - Adjusted weighted sum aggregation

$$\hat{r}_i(j) = \overline{r}_i + \frac{\sum_{i' \in N_i} sim(i, i') \left(r_{i'}(j) - \overline{r}_{i'} \right)}{\sum_{i' \in N_i} sim(i, i')}$$

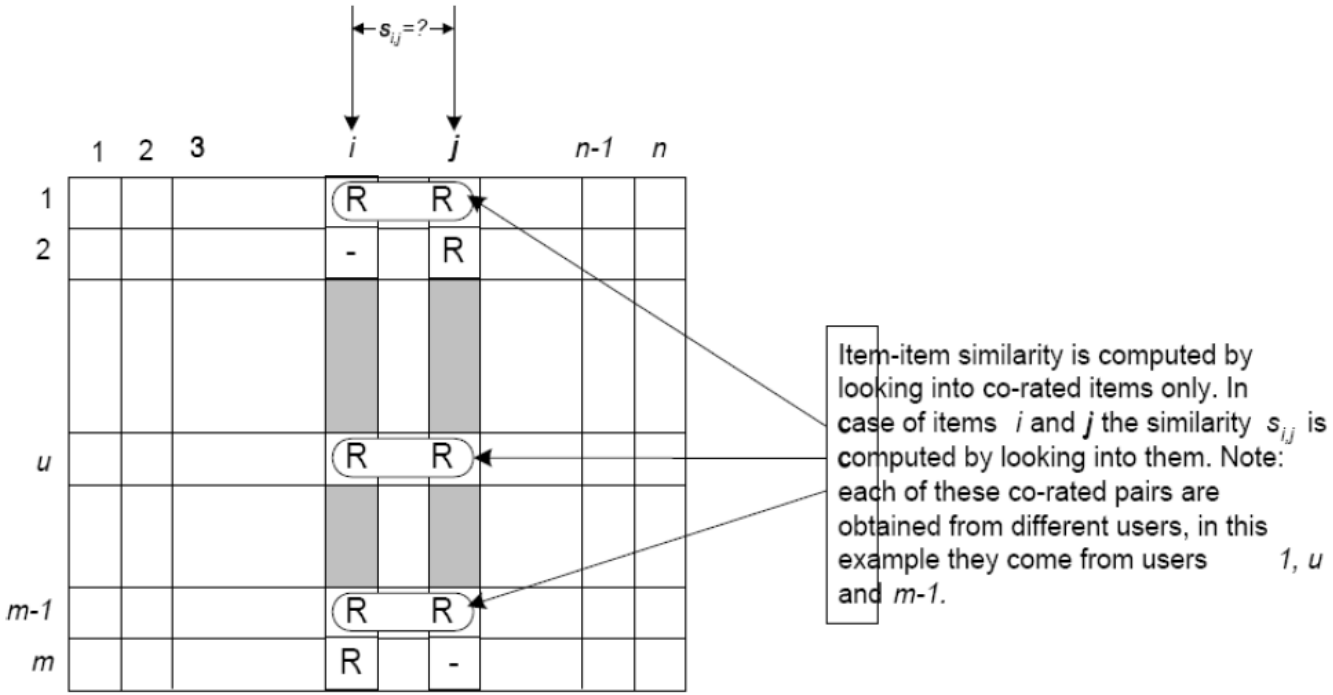
Drawbacks of User-Based CF

- Scalability

- Matrix Sparsity

Item-based CF

- Goal: estimate $\hat{r}_i(j)$
- Step 1: Find set of items similar to j
- Step 2: Use similar items to estimate $\hat{r}_i(j)$



Similarity Heuristics for Item-based CF

- Cosine-based

$$sim(j,j') = cos(x_j, x_{j'}) = \frac{x_j \cdot x_{j'}}{||x_j||_2 ||x_{j'}||_2} = \frac{\sum_{k \in N_{j,j'}} x_{jk} \cdot x_{j'k}}{\sqrt{\sum_{k \in N_{j,j'}} x_{jk}^2} \sqrt{\sum_{k \in N_{j,j'}} x_{j'k}^2}}$$

- Correlation-based

$$sim(j,j') = \frac{\sum_{i' \in N_{j,j'}} \left(r_{i'}(j) - \overline{r}(j)\right)\left(r_{i'}(j') - \overline{r}(j')\right)}{\sqrt{\sum_{i' \in N_{j,j'}} \left(r_{i'}(j) - \overline{r}(j)\right)^2} \sqrt{\sum_{i' \in N_{j,j'}} \left(r_{i'}(j') - \overline{r}(j')\right)^2}}$$

- Adjusted Cosine

$$sim(j,j') = \frac{\sum_{i' \in N_{j,j'}} \left(r_{i'}(j) - \overline{r}_{i'}\right)\left(r_{i'}(j') - \overline{r}_{i'}\right)}{\sqrt{\sum_{i' \in N_{j,j'}} \left(r_{i'}(j) - \overline{r}_{i'}\right)^2} \sqrt{\sum_{i' \in N_{j,j'}} \left(r_{i'}(j') - \overline{r}_{i'}\right)^2}}$$

Item-Based CF (continued)

- Weighted-sum prediction

$$\hat{r}_i(j) = \frac{\sum_{j' \in G_{ij}} sim(j, j') r_i(j')}{\sum_{j' \in G_{ij}} sim(j, j')}$$

Drawbacks of CF (in general)

- New user problem
- New item problem
- Matrix sparsity problem

Hybrids

- Combining recommenders
- Adding content-based features
- …

Netflix competition

june 26, 2009 around three years after launch of netflix prize. BellKor Pragmatic Chaos (merge of three teams) broke the 10% barrier in improving accuracy on movie prediction. NetFlix data set: 100 million + ratings, 1.4 million used as test set. 1-5 stars. RMSE on test set.

The NetFlix Competition

- October 2006. Released user ratings data spanning 6 years
  - <anon ID, date, title, year, #stars>
  - some noise added
  - 100million+ ratings, from 480,000+ users on 17,000+ movies
- Do better than NetFlix's algorithm in terms of root mean square error (RMSE) on **hold-out test data**; from 0.9525 RMSE to 0.8572 RMSE. $1M Grand Prize.

Bellkor Pragmatic Chaos

- Main ideas:
  - Baseline predictors to remove bias
  - Collaborative filtering
  - “Boosting” and “ensemble” techniques

A. Baseline predictors

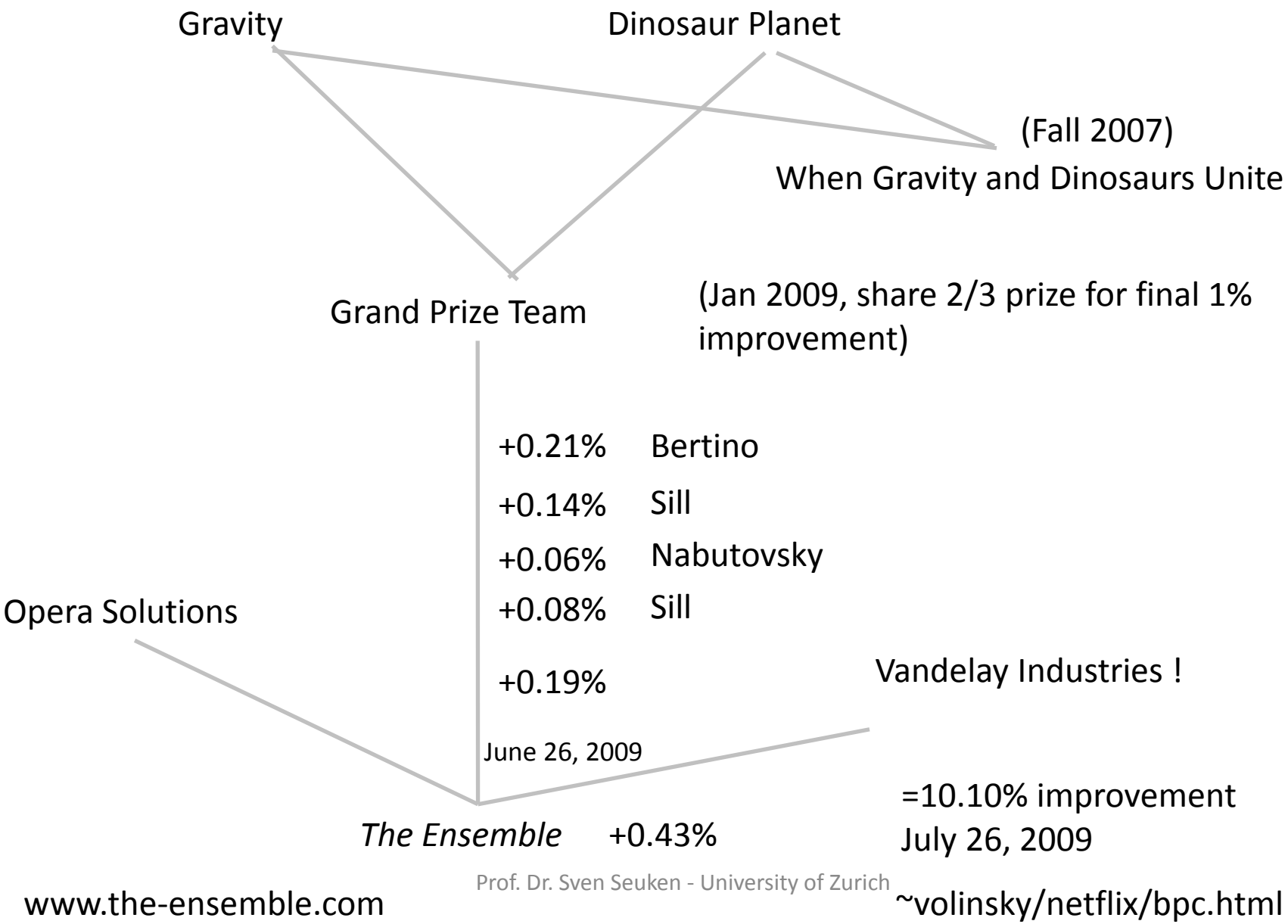
- For user i, item j. $B_{ij} = {}^{1} + b_i + b_j$
- E.g., $B_{ij} = 3.7 - 0.3 + 0.5$
  - Average over all movies is 3.7
  - Joe: $b_i = -0.3$
  - Titanic: $b_j = + 0.5$
- + trick with “same day effect” got to RMSE 0.9278, already 2.6% improvement

(Sill et al. ICML'10)

C. Blending Methods

- Combine 100+ individual predictors
- Use 20+ "meta-features" for this combination, such as # of user ratings, # movie ratings

$y(x) = g(y_1(x),\ldots, y_K(x))$
meta-features



Privacy and Netflix

- **Netflix Cancels Contest After Concerns Are Raised About Privacy**
- **By STEVE LOHR Published: March 12, 2010**
- Netflix's $1 million contest for the best method to improve its movie recommendations was such a research and business hit that, when the winners were declared last September, the company immediately announced plans for another one.
- But it turned out that letting very smart computer scientists and statisticians dig through the video rental site's data had one major, unforeseen drawback. A pair of researchers at the University of Texas showed that the supposedly anonymized data released for the contest, which included movie recommendations and choices made by hundreds of thousands of customers, could in fact be used to identify them.
- That brought the attention of the Federal Trade Commission, and also drew a lawsuit from KamberLaw L.L.C. On Friday, bowing to privacy concerns, Netflix said it was shelving its plans for a sequel to the first contest.

"No, all customer identifying information has been removed; all that remains are ratings and dates. This follows our privacy policy, which you can review here. Even if, for example, you knew all your own ratings and their dates you probably couldn't identify them reliably in the data because only a small sample was included (less than one-tenth of our complete dataset) and that data was subject to perturbation. Of course, since you know all your own ratings that really isn't a privacy problem is it?"

Algorithm idea

(Narayanan and Shmatikov)

- Given database D, and aux record, compute

  $$\text{score}(\text{aux},x') / \sum_{j \in \text{aux}} e^{(r_j - r_{j'})/ r0} + e^{(d_j - d_{j'})/ d0}$$

  for each record $x' \in D$. Used r0 = 1.5, d0= 30
- If high score significantly larger than second-highest, output record
- Found with 8 movies and dates within 14 days, could uniquely identify 99% of Netflix users
- Use IMDB with a few dozen users, uniquely identified two Netflix users
  - to outside 28 standard deviations, and 15 sd.s!

Privacy breach

- Problem: from public data from a user on some movies, get data from user on all movies watched and preferences