Lecture 4:
Collaborative Filtering I

Derek Bridge

Motivations

- Why do we want an alternative to content-based approaches?
  - Suppose items do not have readily available descriptions or description fail to capture the subjective experiences of consuming the item
    - e.g. music, video, art, photographs, jokes
  - Content-based approaches rarely make recommendations that extend our tastes
    - we would like *serendipitous* recommendations

Intuitions: sharing opinions

- Ann asks her friends whether she should see the latest Hollywood release
- Ben recommends it
  - but he seems to recommend everything
- Col doesn't think much of it
  - and he has a habit of recommending things Ann likes
- Deb hated it
  - but she hates all Hollywood movies
- …
- Over time, Ann learns whose opinions can be applied to help her determine the quality of items

Intuitions: sharing opinions

- In 'word of mouth' recommendations
  - we take into account how similar the other person's tastes are to our own
  - item descriptions are not needed
- Collaborative filtering (CF): evaluating items using the opinions of other people
  - automates word-of-mouth
  - but, through the Web, we can access the opinions of thousands of people
  - recommendations are based on the opinions of many similar users rather than a small group of friends

Ratings matrix

	Alien	Brazil	Crash	Dumbo	E.T.	Fargo
Ben		2	5	3	1	2
Col	5	5		3	4	
Deb					3	
Edd	5	4	2	4	3	3
Flo	5	5	4			

Ann	2			4	3	5

Ratings

- Scalar
  - often numeric, e.g. *1-5*
  - but always ordered, e.g. *strongly disagree, disagree, neutral, agree, strongly agree*
  - not too few values: why?
  - not too many values: why?

7 : BOOM! One of my FAVORITE few! Can't live without it.
6 : Solid. They are up there.
5 : Good Stuff.
4 : Doesn't turn me on, doesn't bother me.
3 : Eh. Not really my thing.
2 : Barely tolerable.
1 : Pass the earplugs.

- Binary
  - two values, e.g. *+/-*, *agree/disagree*, *good/bad*
- Unary
  - one value, e.g. to indicate that a link was clicked, a web page was visited, an item was purchased

Binary and unary ratings matrices

	Item1	Item2	Item3	Item4	Item5	Item6
User1		✖	✖	✔	✖	
User2	✔	✔		✔	✔	✔
User3					✔	
User4	✔	✔	✖	✖	✖	✔
User5	✔		✔			

	Item1	Item2	Item3	Item4	Item5	Item6
User1	✔			✔	✔	✔
User2					✔	
User3			✔		✔	
User4	✔	✔		✔		✔
User5			✔	✔	✔	

Ratings sparsity

- In all cases, a user may have no rating for an item (shown as blank or as $\perp$)
- Ratings density: proportion of entries in the matrix $\neq \perp$
- In most commercial scenarios,
  - very large number of items, e.g. thousands, perhaps millions
  - even the most active users likely to have rated < 1%
  - hence, very sparse
    - e.g. MovieLens test data: 93.7%
    - e.g. PTV data: 99.7%
- Makes it very hard to find similar users

Explicit ratings

- User is asked to provide the ratings directly
- Often thought to be more accurate than implicit ratings but
  - may be inadvertently inaccurate: do you know your own mind?
  - may be deliberately inaccurate
    - due to privacy/security concerns
    - due to attempts to bias the system or counteract perceived bias
    - due to 'posturing'

Explicit ratings

- But imposes a cost on users
  - requires user's willingness to provide the information
  - requires user's willingness to spend the time
- There were fears that users would not provide ratings without rewards, e.g.
  - a user only receives recommendations in exchange for ratings
  - a user receives other incentives for ratings (T-shirts, discounts, privileged content)
- On the other hand, some users enjoy providing and sharing feedback
  - prestige
  - social interaction
  - the system acts as an extension of their memory

Implicit ratings

- Ratings are inferred from user actions
  - clicks, read time, searches, purchases,…
- Might be recorded by the server or by a client-side module (that, at some point, passes them to the personalization engine, e.g., on the server)
  - client-side is likely to be more precise when measuring times
  - client-side is able to observe a wider range of actions (e.g. a page being bookmarked in browser, a file being downloaded & saved)
- There's no cost to the user
- But implicit ratings are often only unary
  - not easy to infer negative opinions
  - not easy to infer different degrees on a rating scale

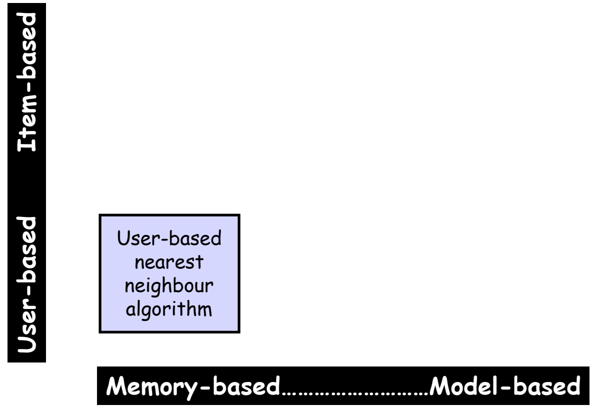
Implicit ratings

- Studies show "…no clear answer on whether implicitly created profiles are more or less accurate than explicitly created profiles." S. Gauch et al., 2007
- But inferences from user actions may not be sound
  - Consider read time
  - Suppose we infer that a user is interested in a topic because s/he spends a long time reading an article on that topic
  - But s/he may have taken a break (although we can infer this too to some extent)
  - S/he may have found the article confusing, rather than interesting
  - S/he may have read it and ultimately found it uninteresting
  - …
- Inferences are more likely to be sound when based on more data
  - E.g. reading several articles on the same topic
- For some users, privacy concerns are greater (esp. since the inferences may not be correct)
  - E.g. "If TiVo thinks you are gay, here's how to set it straight" (WSJ, 26/10/2002)

CF functionality

- *Predict* a rating for a given item
  - compute & show a predicted rating for the item
- *Recommend* items
  - compute & show a list of recommended items, probably ordered
  - one approach is to make predictions for all unrated items and recommend those with the highest predicted ratings
  - but it is possible to build systems that make good recommendations without making any predictions

Types of CF algorithm



The prediction algorithm

- To predict a rating for active user *a* and item *i*:
  - For every user *u* who has rated *i*,
    - Compute the similarity between *a* and *u*, *sim(a, u)*
  - Let *NN* be *a*'s nearest neighbours who have rated *i*, i.e. the set of size *k* for whom *sim(a, u)* is highest
  - Compute a predicted rating, *pred(a, i)*, from *NN*'s ratings for *i*
- Worked example
  - We'll predict Ann's rating for Brazil
  - Question: We must compute the similarity between Ann and…who? (Why?)

Similarity, *sim(a, u)*

- Many systems use *Pearson correlation*

$$sim(a,u) = \frac{\sum_{i \in CR_{a,u}} (r_{a,i} - \bar{r}_a)(r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i \in CR_{a,u}} (r_{a,i} - \bar{r}_a)^2} \sqrt{\sum_{i \in CR_{a,u}} (r_{u,i} - \bar{r}_u)^2}}$$

- It is computed over *a* and *u*'s co-rated items, $CR_{a,u}$
  - i.e. items rated by both *a* and *u*
- $r_{a,i}$ is *a*'s rating for *i*; similarly for $r_{u,i}$
- $\bar{r}_a$ is *a*'s average rating for the co-rated items; similarly for $\bar{r}_u$
- Pearson correlation will be 1.0 for users in perfect agreement and -1.0 for users in perfect disagreement

Example: *sim(Ann, Ben)*

$$sim(a,u) = \frac{\sum_{i \in CR_{a,u}} (r_{a,i} - \overline{r_a})(r_{u,i} - \overline{r_u})}{\sqrt{\sum_{i \in CR_{a,u}} (r_{a,i} - \overline{r_a})^2} \sqrt{\sum_{i \in CR_{a,u}} (r_{u,i} - \overline{r_u})^2}}$$

	Alien	Brazil	Crash	Dumbo	E.T.	Fargo
Ben		2	5	3	1	2
Ann	2			4	3	5

Exercise: *sim(Ann, Col)*

$$sim(a,u) = \frac{\sum_{i \in CR_{a,u}} (r_{a,i} - \overline{r_a})(r_{u,i} - \overline{r_u})}{\sqrt{\sum_{i \in CR_{a,u}} (r_{a,i} - \overline{r_a})^2} \sqrt{\sum_{i \in CR_{a,u}} (r_{u,i} - \overline{r_u})^2}}$$

	Alien	Brazil	Crash	Dumbo	E.T.	Fargo
Col	5	5		3	4	
Ann	2			4	3	5

Example

- The similarities
  - sim(Ann, Ben) =
  - sim(Ann, Col) =
  - sim(Ann, Edd) = -0.4
  - Sim(Ann, Flo) = 0.0
- *Suppose k=3*
- *Ann's 3 nearest neighbours are: Ben, Col, Flo*

Predicted rating, *pred(a, i)*

- We could just take an average: $pred(a,i) = \frac{\sum_{u \in NN} r_{u,i}}{k}$
- But we want to take into account that some neighbours are more similar to *a* than others
  - their ratings should contribute more to the prediction
- So instead take a weighted average: $pred(a,i) = \frac{\sum_{u \in NN} r_{u,i} \times sim(a,u)}{\sum_{u \in NN} sim(a,u)}$
- But some users are restrained; others effusive
  - A rating of 4 from the former means the same as a 5 from the latter
- So include an adjustment for users' average ratings: $pred(a,i) = \overline{r_a} + \frac{\sum_{u \in NN} (r_{u,i} - \overline{r_u}) \times sim(a,u)}{\sum_{u \in NN} sim(a,u)}$

Example: *pred(Ann, Brazil)*

	Alien	Brazil	Crash	Dumbo	E.T.	Fargo
Ben		2	5	3	1	2
Col	5	5		3	4	
Flo	5	5	4			
Ann	2			4	3	5

$$pred(a,i) = \bar{r}_a + \frac{\sum_{u \in NN} (r_{u,i} - \bar{r}_u) \times sim(a,u)}{\sum_{u \in NN} sim(a,u)}$$

Recommendation algorithms

- To recommend items to active user *a*:
  - Suppose there's a small set of items from which we must make a recommendation (e.g. movies at your multiplex this week)
    - Then compute a predicted rating for each item, as before
    - Recommend the one(s) with the highest predicted ratings
  - Suppose the recommendation is not constrained (e.g. can recommend any movie in the IMDb)
    - For every user *u* excluding *a*,
      - Compute the similarity between *a* and *u*, *sim(a, u)*
    - Let *NN* be *a's* nearest neighbours, i.e. the set of size *k* for whom *sim(a, u)* is highest
    - Let *Candidates* be items rated by at least one member of *NN* but not rated by *a*
    - How might you order *Candidates* to decide which to recommend?

Evaluation

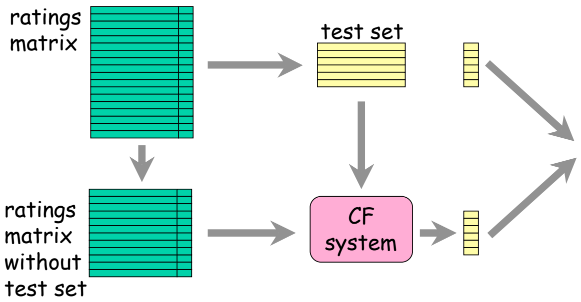
- Academics and practitioners want to measure how well a CF (or other) system meets its goals
- We need metrics
  - will depend on the goals of the system
- First we'll evaluate prediction
  - accuracy
  - coverage
  - time & space efficiency

Accuracy

- The magnitude of the error between a predicted rating and the 'true' rating
- To estimate accuracy:
  - take an item *i* whose rating by some user *a* is already known, $r_{a,i}$
  - get the CF system to predict the rating, *pred(a, i)*
  - compute the *absolute error*, *abs*($r_{a,i}$ - *pred(a, i)*)
  - do this again & again, for lots of different items and users
  - compute the mean (average) of the errors
  - this is called the *mean absolute error (MAE)*

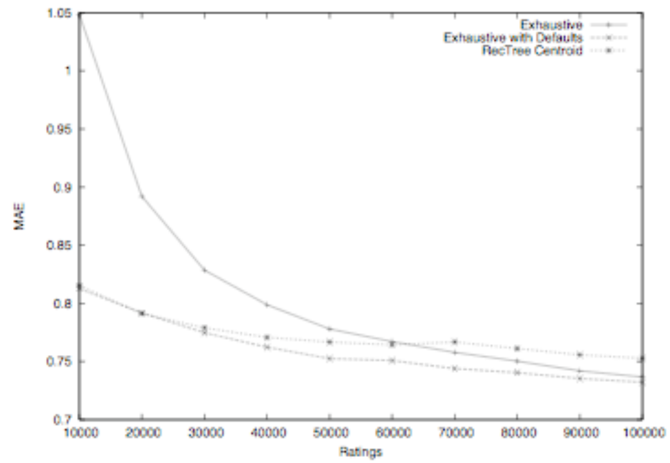
Accuracy

- Partition the ratings matrix:



- Repeat this process, e.g. with another ratings matrix

MAE on MovieLens dataset



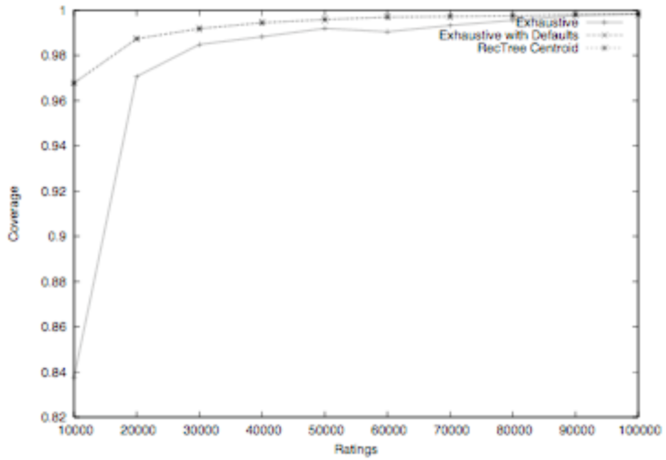
The Netflix competition

- Netflix, online DVD rental company: www.netflix.com
  - their CF system, CineMatch, makes recommendations
- The Netflix Prize:
  - ratings matrix of "more than 100 million ratings from over 480 thousand randomly-chosen, anonymous customers on nearly 18 thousand movie titles"
  - $1,000,000 Grand Prize for improving accuracy (measured as root mean squared error, RMSE) by 10%
  - possible $50,000 annual Progress Prizes
- How's it going? www.netflixprize.com/leaderboard
- Issues
  - finding identities (de-anonymization)
  - cheating

Coverage

- Sometimes the system cannot make a prediction
  - Why?
- Using the same evaluation methodology, compute coverage as the percentage of times the system was able to make a prediction

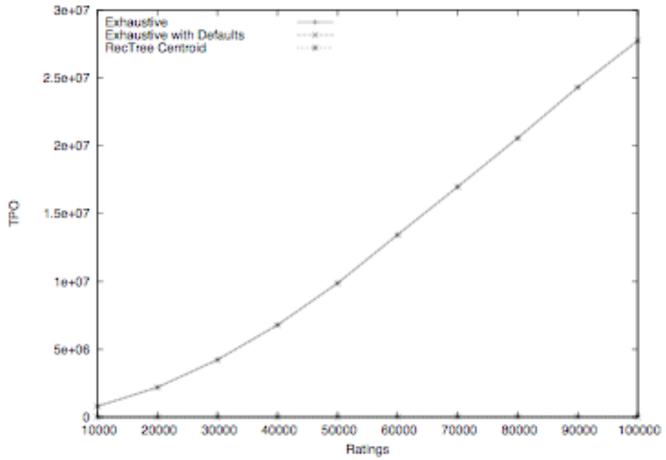
Coverage on MovieLens dataset



Time and space efficiency

- Using the same methodology, we can compute
  - the average time it takes to make a prediction
  - the average amount of memory used when making predictions
- Scalability is important too, e.g. compute the average time it takes to make a prediction
  - when the ratings matrix contains 10,000 ratings
  - when the ratings matrix contains 20,000 ratings
  - when the ratings matrix contains 30,000 ratings

Num of computations on MovieLens dataset



Evaluation of recommendations

- Evaluating recommendations is harder than evaluating predictions
- *Accuracy*
  - need to evaluate the accuracy of a ranked list
  - unreliable to just add up the MAE for each item
    - users perceive errors at the top of a ranked list of recommendations as much more serious than ones at the bottom
    - hence, perhaps bring in a weighting scheme
  - could use precision (percentage of items the user judges as relevant - see lecture 3)
    - but this ignores placement in the list
    - requires users to make judgements

Evaluation of recommendations

- Other factors
  - novelty: not recommending things the user has seen before
  - serendipity: extending the user's tastes
  - learning rate: how quickly it becomes useful for a user
  - …
- This discussion has been about evaluating the system in advance of use (experiments)
- We can also evaluate during use
  - user satisfaction questionnaires
  - but also usage statistics: clicks on recommended items, conversion rate (visit-to-buy ratio), repeat visits, etc.

Comparison with content-based filtering

- Assumptions
  - CF: people with similar tastes will rate items similarly
  - CB: items with similar item descriptions will be rated similarly
- Requirements
  - CF: requires ratings but not item descriptions
  - CB: requires item descriptions but not ratings
- Special advantages
  - CF: serendipitous recommendations
  - CB: more responsive to needs (if they can be articulated in terms of item descriptions)
- Conclusion: they are often complementary, hence hybrid systems