CS6120: Intelligent Media Systems

Dr. Derek Bridge
School of Computer Science & Information Technology
UCC

Informal Comparison

Content-based
- New item:
  - can recommend new items even if they have not yet been rated (provided they have a description)
- New user:
  - needs a profile of items plus their descriptions plus their ratings
- Serendipity:
  - unlikely since recommendations are similar to profile items

User-Based Collaborative
- New item:
  - cannot recommend new items until they have been rated by at least one, preferably several users
- New user:
  - needs a profile of items plus their ratings
- Serendipity:
  - possible since one user's tastes may be extended by her neighbours' tastes

Evaluating Recommender Systems

- Many algorithms; many variations; many parameters
- We must make comparisons
  - Practitioner
    - deciding what approach to use
  - Researcher
    - deciding when a new approach is better than existing approaches
- Evaluate by running experiments
  - involving two or more systems
  - recording and comparing metrics that attempt to measure desirable properties

Good Experimental Design

- Form a hypothesis:
  - e.g. system A will have higher accuracy than system B
  - i.e. not a “fishing expedition”
- Control all other factors:
  - experimental conditions for systems A and B should vary only in what is being tested
  - i.e. make the comparison as fair as possible
- Be clear how to generalise the findings:
  - based on, e.g., how many users, how many datasets,…
  - use confidence tests

Types of Experiment

- Offline experiments
  - use datasets
  - measure against some ’ground truth’
- User studies
  - recruit a set of users
  - measure their performance in a controlled environment on a set of tasks
- Online evaluation
  - use real users who are oblivious to the experiment
  - measure their performance when using variants of the deployed system

OFFLINE EXPERIMENTS

Offline Experiments

- Pro:
  - can compare many systems at low cost
- Con:
  - narrow set of metrics, based on whatever 'ground truth' you have in the dataset
- Issues:
  - ensure dataset has no distribution bias
  - if you must simulate user behaviour, avoid oversimplifying the simulation
  - ensure it's an allowable use of the data; ensure privacy is protected;...
- Good for identifying promising variants/winnowing out the rest

Datasets

- Offline experiments use pre-collected datasets
- Practitioners
  - collect & understand your own dataset
- Researchers
  - use a publicly available dataset
- Datasets from www.grouplens.org/datasets/movielens/:
  - MovieLens 100K, 1M and 10M
  - Delicious
  - Last.FM
  - MovieLens extended with IMDb/Rotten Tomatoes
  - WikiLens
  - BookCrossing
  - Jester
- No longer available
  - EachMovie
  - Netflix

MovieLens 100k Dataset

- Collected from September 1997 to April 1998
- 100,000 ratings (1-5) from 943 users for 1682 movies
- For each movie, some identification and descriptive data including a set of genres
- For each user, some demographic data (age, sex, occupation, zip)
- Excludes users who had rated fewer than 20 movies and users who had incomplete demographic data

MovieLens 100k Dataset

- `u.data`: userid, movieid, rating, timestamp

196	242	3	881250949
186	302	3	891717742

- `u.item`: movieid, movie title, release date, video release date, IMDb URL, Unknown, Action, Adventure, Animation, Children's, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, Musical, Mystery, Romance, Sci-Fi, Thriller, War, Western

186	The Blues Brothers	1980	-	http://...	0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0

- `u.user`: userid, age, gender, occupation, zip code

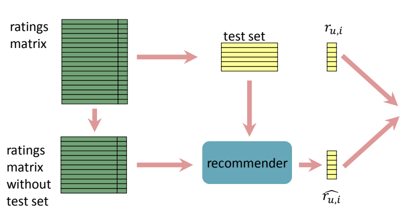
302	42	M	educator	77904

Ground Truth

- How do you know whether the recommender's prediction is right?
- You need the correct answer
  - the *ground truth*

- The key idea: split the dataset
  - use some ratings to build the recommender (*training set*)
  - ask the recommender to predict what you withhold (*test set*)

Holdout



Holdout

randomly split dataset into *Train* and *Test*
for each $r_{u,i}$ in *Test*
make prediction, $\widehat{r_{u,i}}$, using ratings in *Train*
compare $\widehat{r_{u,i}}$ with ground truth, $r_{u,i}$ in *Test*

- This methodology is called *holdout*
  - because the true ratings are withheld from the system
  - typically the split is 70%/30% or 80%/20% training/test

Repeated Holdout

- But suppose we get a lucky/unlucky training set/test set
- To avoid this, *repeat* the process and average the results
  - i.e. run the experiment 5 or 10 times with different random partitions

Accuracy

- We focus on accuracy of a user-based collaborative recommender
  - so the dataset can simply be a ratings matrix
- We look at
  - classification accuracy
  - rating accuracy
  - ranking accuracy

Classification Accuracy

- Suppose the recommender is a classifier
  - it predicts “like”/”dislike”
- During the experiment, build a confusion matrix

		Predicted	
		Like	Dislike
Actual	Like	True Positives	False Negatives
	Dislike	False Positives	True Negatives

$$Precision = \frac{True\ Positives}{Ture\ Positives + False\ Positives} \quad Recall = \frac{True\ Positives}{Ture\ Positives + False\ Negatives}$$

Example

		Predicted	
		Like	Dislike
Actual	Like	40	20
	Dislike	10	30

$$Precision = \frac{True\ Positives}{Ture\ Positives + False\ Positives} \quad Recall = \frac{True\ Positives}{Ture\ Positives + False\ Negatives}$$

- Calculate precision and recall

Ratings Accuracy

- Suppose the recommender does regression
  - predicts ratings
- Measure the magnitude of the *error* between $r_{u,i}$ and $\widehat{r_{u,i}}$
- To compute error take the difference $r_{u,i} - \widehat{r_{u,i}}$ but
  - either getting the absolute value: $abs(r_{u,i} - \widehat{r_{u,i}})$
  - or square it: $(r_{u,i} - \widehat{r_{u,i}})^2$
  - why must we use abs or square?
  - what is the thinking behind squaring?

MAE and RMSE

- Let *Test* be the set of ratings that you test on
- If you're using absolute difference, you compute the *Mean Absolute Error (MAE)*:

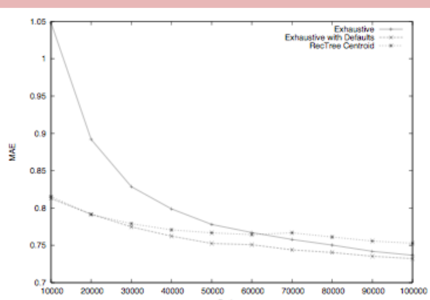
$$\frac{\sum_{r_{u,i} \in Test} abs(r_{u,i} - \widehat{r_{u,i}})}{|Test|}$$

- If you're squaring the difference, you compute the *Root Mean Squared Error (RMSE):*

$$\sqrt{\frac{\sum_{r_{u,i} \in Test} (r_{u,i} - \widehat{r_{u,i}})^2}{|Test|}}$$

- Lots of other possibilities too

MAE on MovieLens Dataset



Netflix Competition

- Netflix
  - their CF system, CineMatch, makes recommendations
- The Netflix Prize (www.netflixprize.com), 2006-2009:
  - 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 (as measured by RMSE) by 10%
  - winner was BellKor's Pragmatic Chaos Team, which improved accuracy by 10.06% using an ensemble approach
- Issues
  - finding identities (de-anonymization); lawsuit

Ranking Accuracy

- Most recommenders produce a ranked list
- Position in the list matters
  - we want recommendations that match the 'ground truth'
  - but we want these recommendations to come early in the list

- A simple approach
  - if a successful recommendation comes at position k, then score this success as ${}^{1}/_{k}$

Discussion: Averaging

- We've assumed you average over all predictions
  - but you might instead compute an average for each user (or item), and then average these
  - why?
- And we may want to measure accuracy for specific types of item or user
  - new or newish users, with no or few ratings (cold-start)
  - users who are "black sheep" or "grey sheep"
  - new or newish items, with no or few ratings
  - items which are in the long tail

Discussion: Be Aware of Assumptions

- I like this experimental method a lot:
  - divide ratings matrix into *Train* and *Test*, but *Test* contains only items users rated 5
  - for each $r_{u,i} \in$ *Test*
    - randomly select 1000 items not rated by u
    - predict rating for i and for the 1000 items
    - rank the 1001 items by their predicted ratings and take the top n (e.g. $n = 10$)
    - you have a true positive if i is in the top-n, otherwise a false negative
- But note an assumption this method shares with many others
  - that the 1000 items are not relevant to the user
  - it penalises you for predicting them ahead of i
- This assumption could be incorrect
  - hence we must recognise that many methods underestimate true accuracy

Discussion: Trade-offs

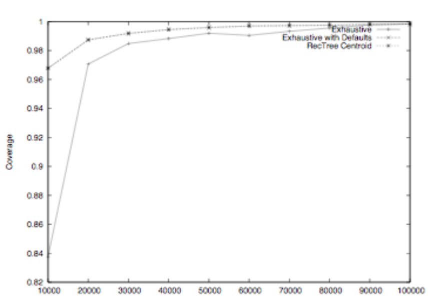
- Improving accuracy may worsen other properties
- Measure other properties of the system
  - but be warned again about fishing trips!

- Properties you might measure in offline experiments
  - coverage
  - efficiency
  - diversity
  - novelty
  - serendipity
  - resilience to attack
  - …

Coverage

- Sometimes a user-based collaborative recommender cannot make a prediction
  - why?
- Using the same experimental methodology, compute coverage
  - e.g. as the percentage of times the system was able to make a prediction
- An easy way to improve coverage
  - resort to some non-personalised recommendation

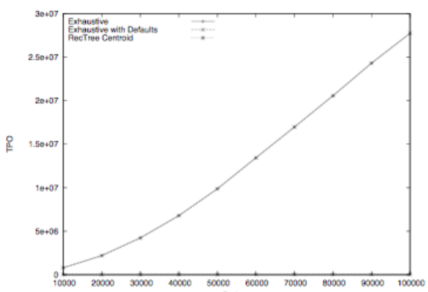
Coverage on MovieLens Dataset



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 a prediction
- Scalability is important too, e.g. compute the average time it takes to make a prediction
  - when the ratings matrix contains, say, 10,000 ratings
  - when the ratings matrix contains, say, 20,000 ratings
  - when the ratings matric contains, say, 30,000 ratings

Num of Computations on MovieLens Dataset



USER STUDIES

User Studies

- Recruit a set of users
  - e.g. a lecturer's students, people off the street, existing users
- Ask them to complete a set of tasks
  - measure performance quantitatively, e.g. time taken, number of clicks
  - survey them after, and even before & during, for qualitative judgments
- Pro:
  - can obtain wide range of quantitative and qualitative data
- Con:
  - limited in size and scope by expense (time, compensation)
- Issues:
  - need pilot studies to spot problems with the experiment
  - the bias of using volunteers (they are your more interested users)
  - the bias from being aware they are in an experiment

Within Subjects vs Between Subjects

Within subjects

- Each subject (user) tests all the candidate systems
- Advantages
  - can use fewer users in *within* studies
  - can ask users comparative questions in *within* studies
  - apparent superiority of one system could be due to bias in the user split in *between* studies

Between subjects

- Each subject tests only one candidate system (assigned to her at random)
- Advantages
  - users are more conscious of the experiment in *within* studies
  - order of testing needs to be controlled for in *within* studies
  - easier to test longer-term effects from repeated system use in *between* studies

User Studies

Quantitative metrics

- E.g. you might have ways of varying the diversity of recommendation lists
- Measure the effects of diversity on
  - time to complete a task
  - number of clicks to complete a task
  - position in recommended list of item the user clicks on
  - …

Qualitative metrics

- E.g. you might have different explanation facilities
- Survey the users to determine the effect the explanations have on, e.g.:
  - user's likelihood to purchase/consume
  - user's confidence/trust in the system
  - user satisfaction/enjoyment
  - …

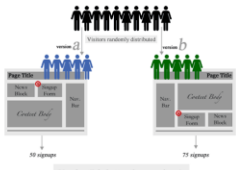
ONLINE EVALUATION

Online Evaluation

- Use real users who are oblivious to the experiment
- A/B testing (= between subjects)
  - randomly assign a small % of users to a variant of the real system and measure whether variant has, e.g., higher sales
- Measure real user behaviour, e.g.
  - logins, clicks, purchases, time spent...
- Pro:
  - measures what we really care about: behaviour change, long-term profit, user retention,...
- Con:
  - an unsuccessful variant may drive away the users who were assigned to it
- Issues:
  - need enough existing traffic

A/B Testing and the Web

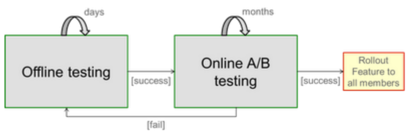
- Organizations use A/B testing to compare human-computer interfaces
  - e.g. different web site layouts



www.smashingmagazine.com/2010/06/24/the-ultimate-guide-to-a-b-testing/

Netflix

- Netflix is always running experiments
  - an approach they call Consumer Data Science
  - dozens of A/B experiments running in parallel
  - see techblog.netflix.com/2012/06/netflix-recommendations-beyond-5-stars.html



Bing

- Bing runs over 50 concurrent experiments
  - in a visit, you're in about 10 experiments
  - there is no single Bing
  - e.g. Ron Kohavi's talk: robotics.stanford.edu/~ronnyk/2012-09ACMRecSysNR.pdf
- The same is true of Google, Amazon,…