Collaborative Filtering

Tamara Berg

Advanced Multimedia
Reminders

• Final Project Presentations next week – presentation slot draw today.

• Final Project Write-ups due May 20 – 8 page document including abstract, introduction (motivation), method, results, & figures.

• Next class – Kinect day!
Collaborative filtering

- Collaborative filtering (CF) is the process of filtering for information or patterns using techniques involving collaboration among multiple agents, viewpoints, data sources.
Uses

- Collaborative filtering methods have been applied to many different kinds of data including sensing and monitoring data - such as in mineral exploration, environmental sensing over large areas or multiple sensors; and financial data

- Also used in electronic commerce and web 2.0 applications where the focus is on user data, etc
More simply

- The method of making automatic predictions (filtering) about the interests of a user by collecting taste information from many users (collaborating).
- The underlying assumption of CF is that those who agreed in the past tend to agree again in the future.
An example

- A collaborative filtering or recommendation system for music tastes could make predictions about which music a user should like given a partial list of that user's tastes (likes or dislikes).
- Note that these predictions are specific to the user, but use information gleaned from many users.
- This differs from the simpler approach of giving an average (non-user specific) score for each item of interest, for example based on its number of votes.
Everyday example
Goodnight Moon (Hardcover)
by Margaret Wise Brown (Author), Clement Hurd (Illustrator) "The cow jumping over the moon..." (more)

List Price: $17.99
Price: $12.23 & eligible for FREE Super Saver Shipping on orders over $25. Details
You Save: $5.76 (32%)

In Stock.
Ships from and sold by Amazon.com. Gift-wrap available.

Want it delivered Thursday, April 30? Order it in the next 4 hours and 27 minutes, and choose One-Day Shipping at checkout. Details

37 new from $8.52  24 used from $4.55  3 collectible from $17.99

Also Available in:
Paperback  $6.99  $4.05  77 used & new from $0.01
Audio Downloaded (Audible.com)  $4.95  $0.74  9 used & new from $0.00
Library Binding  $17.00  $13.97  33 used & new from $8.92
Audio Cassette (Abridged, Audiobook)  $4.95  $4.95  9 used & new from $4.94

Show more editions and formats
Everyday example

Frequently Bought Together

- Goodnight Moon by Margaret Wise Brown
- The Very Hungry Caterpillar by Eric Carle

Price For All Three: $28.64

Add all three to Cart

Customers Who Bought This Item Also Bought

- Brown Bear, Brown Bear, What Do You See? by Bill Martin Jr. 342 $7.95
- Guess How Much I Love You by Sam McBratney 312 $7.99
- Pat the Bunny (Touch and Feel Book) by Dorothy Kunhardt 166 $9.99
- Where the Wild Things Are by Maurice Sendak 394 $10.77
- The Going-To-Bed Book by Sandra Boynton 188 $5.99
Everyday Example
Everyday example
Inlinks are “good” (recommendations)

Inlinks from a “good” site are better than inlinks from a “bad” site

but inlinks from sites with many outlinks are not as “good”...

“Good” and “bad” are relative.
Google’s PageRank

Imagine a “pagehopper” that always either

• follows a random link, or
• jumps to random page

Slide Credit: William W. Cohen
Google’s PageRank

Imagine a “pagehopper” that always either
- follows a random link, or
- jumps to random page

PageRank ranks pages by the amount of time the pagehopper spends on a page:
- or, if there were many pagehoppers, PageRank is the expected “crowd size”

Slide Credit: William W. Cohen
Everyday Examples of Collaborative Filtering...

- Bestseller lists
- Top 40 music lists
- The “recent returns” shelf at the library
- Unmarked but well-used paths thru the woods
- Many weblogs
- “Read any good books lately?”
- ....

- **Common insight:** personal tastes are *correlated*:
  - If Alice and Bob both like X and Alice likes Y then Bob is more likely to like Y
  - especially (perhaps) if Bob knows Alice

Slide Credit: William W. Cohen
Methodology 1

• Collaborative filtering systems usually take two steps:

1. Look for users who share the same rating patterns with the active user (the user whom the prediction is for).

2. Use the ratings from those like-minded users found in step 1 to calculate a prediction for the active user.
Methodology 1

• Collaborative filtering systems usually take two steps:

1. Look for users who share the same rating patterns with the active user (the user whom the prediction is for).
2. Use the ratings from those like-minded users found in step 1 to calculate a prediction for the active user.

User similarity based!
Methodology 2

• Alternatively, item-based collaborative filtering popularized by Amazon.com (users who bought x also bought y) proceeds in an item-centric manner:

1. Build an item-item matrix determining relationships between pairs of items
2. Using the matrix, and the data on the current user, infer his taste
Methodology 2

• Alternatively, item-based collaborative filtering popularized by Amazon.com (users who bought x also bought y) proceeds in an item-centric manner:

1. Build an item-item matrix determining relationships between pairs of items
2. Using the matrix, and the data on the current user, infer his taste

Item similarity based!
Ratings vs Observations

• Collaborative filtering can be based on explicit user ratings of items or implicit observations of normal user behavior (asking the user to explicitly rate items might impose artificial behavior).

• In implicit systems you observe what a user has done together with what all users have done (what music they have listened to, what items they have bought) and use that data to predict the user's behavior in the future.
Why use filtering?

Such techniques can prove very useful as the number of items in a particular category (such as music, movies, books, news, web pages) have become so large that a single person cannot possibly view them all in order to select relevant ones.

Relying on a scoring or rating system which is averaged across all users ignores specific demands of a user, and is particularly poor in tasks where there is large variation in interest, for example in the recommendation of music.
Information Tapestry Project

• The first system to use collaborative filtering was the Information Tapestry project at Xerox PARC (1992).

• This system allowed users to find documents based on previous comments by other users.
Information Tapestry Project

• Built a collaborative email system that allowed users to record their reactions to incoming mail (interesting, uninteresting etc) – collaborative annotation - and then let other users access these annotations through search.

• There were many problems with this system as it only worked for small groups of people and had to be accessed through word specific queries which largely defeated the purpose of collaborative filtering.
Firefly

• One of the largest early collaborative filtering services for music recommendations widely available on the World Wide Web was Firefly, which evolved from MIT Media Lab project (founded in 1995, sold to Microsoft in 1998).

• Firefly's website changed many times. Initially it created a community for users to navigate and discover new artists and albums. Later it was changed to allow users to discover movies, websites, and communities as well. It was well known at the time for its sense of community.

• Firefly technology was used by quite a number of well known businesses, including the recommendation engine for barnesandnoble.com, ZDnet, launch.com, and MyYahoo.
BellCore’s MovieRecommender

- *Recommending And Evaluating Choices In A Virtual Community Of Use.* Will Hill, Larry Stead, Mark Rosenstein and George Furnas, Bellcore; CHI 1995

By virtual community we mean "a group of people who share characteristics and interact in essence or effect only". In other words, people in a Virtual Community influence each other *as though* they interacted but they *do not interact*. Thus we ask: "Is it possible to arrange for people to share some of the personalized informational benefits of community involvement without the associated communications costs?"

Slide Credit: William W. Cohen
MovieRecommender Goals

Recommendations should:

• simultaneously **ease and encourage rather than replace** social processes....should make it easy to participate while **leaving in hooks for people to pursue more personal relationships** if they wish.

• be for **sets of people** not just individuals...multi-person recommending is often important, for example, when two or more people want to choose a video to watch together.

• be **from people not a black box** machine or so-called "agent".

• tell **how much confidence to place in them**, in other words they should include indications of how accurate they are.
BellCore’s MovieRecommender

• Participants sent email to videos@bellcore.com
• System replied with a list of 500 movies to rate on a 1-10 scale (250 random, 250 popular)
  – Only subset need to be rated
• New participant P sends in rated movies via email
• System compares ratings for P to ratings of (a random sample of) previous users
• Most similar users are used to predict scores for unrated movies
• System returns recommendations in an email message.

Slide Credit: William W. Cohen
BellCore’s MovieRecommender

• Evaluation:
  – Withhold 10% of the ratings of each user to use as a test set
  – Measure correlation between predicted ratings and actual ratings for test-set movie/user pairs

Slide Credit: William W. Cohen
Figure 3  Two Scatterplots of Actual Ratings by Predicted Ratings. Plot on left shows movie critics as predictor ($r=0.22$). Plot on right shows virtual community as predictor ($r=0.62$) (all values are jittered for the purpose of visual presentation, 3269 predictions each for 291 users)
Another key observation: rated movies tend to have positive ratings:

Question: What method does this suggest?

Slide Credit: William W. Cohen
BellCore’s MovieRecommender

- Participants sent email to videos@bellcore.com
- System replied with a list of 500 movies to rate
  New participant $P$ sends in rated movies via email
- System compares ratings for $P$ to ratings of (a random sample of) previous users
- **Most similar users** are used to **predict scores** for unrated movies
  - *Empirical Analysis of Predictive Algorithms for Collaborative Filtering* Breese, Heckerman, Kadie, UAI98
- System returns recommendations in an email message.

User similarity based!
Memory based algorithms

• Record previous ratings.
• Estimate new ratings directly from previous ratings and similarity estimates.
• How might we do this?
Remember Nearest Neighbors?
Query document – which class should you label it with?
Classification by Nearest Neighbor

Classes in the vector space

Classify the test document as the class of the document “nearest” to the query document (use vector similarity to find most similar doc)

Should the document ★ be assigned to China, UK or Kenya?

Slide from Min-Yen Kan
Classification by kNN

Classify the test document as the majority class of the k documents “nearest” to the query document.
Nearest neighbors for Filtering

• Instead of documents with words we now have data items (movies, books, etc) with ratings.
• This gives us a vector for each user where each index is the users rating (if any) for that particular item.
• For a new active user we can use nearest neighbor similarity to find the nearest other users, then combine their ratings to predict ratings for the active user.
User-User Collaborative Filtering

Target Customer

Weighted Sum

Slide credit: John Riedl
Algorithms for Collaborative Filtering 1: Memory-Based Algorithms (Breese et al, UAI98)

- $v_{i,j}$ = vote of user $i$ on item $j$
- $I_i$ = items for which user $i$ has voted
- Mean vote for $i$ is
  \[
  \bar{v}_i = \frac{1}{|I_i|} \sum_{j \in I_i} v_{i,j}
  \]
- Predicted vote for “active user” $a$ is weighted sum
  \[
  p_{a,j} = \bar{v}_a + \kappa \sum_{i=1}^{n} w(a,i)(v_{i,j} - \bar{v}_i)
  \]

Slide Credit: William W. Cohen
Algorithms for Collaborative Filtering 1: Memory-Based Algorithms (Breese et al, UAI98)

- K-nearest neighbor
  \[ w(a, i) = \begin{cases} 
  1 & \text{if } i \in \text{neighbors}(a) \\
  0 & \text{else} 
  \end{cases} \]

- Pearson correlation coefficient (Resnick ’94, GroupLens):
  \[ w(a, i) = \frac{\sum_j (v_{a,j} - \bar{v}_a)(v_{i,j} - \bar{v}_i)}{\sqrt{\sum_j (v_{a,j} - \bar{v}_a)^2 \sum_j (v_{i,j} - \bar{v}_i)^2}} \]

- Cosine distance (from IR)
  \[ w(a, i) = \sum_j \frac{v_{a,j}}{\sqrt{\sum_{k \in I_a} v_{a,k}^2}} \frac{v_{i,j}}{\sqrt{\sum_{k \in I_i} v_{i,k}^2}} \]
Algorithms for Collaborative Filtering 1: Memory-Based Algorithms (Breese et al, UAI98)

• Evaluation:
  – split users into train/test sets
  – for each user $a$ in the test set:
    • split $a$’s votes into observed ($I$) and to-predict ($P$)
    • measure average absolute deviation between predicted and actual votes in $P$
    • predict votes in $P$, and form a ranked list
    • assume (a) utility of $k$-th item in list is $\max(v_{a,j} - d, 0)$, where $d$ is a “default vote” (b) probability of reaching rank $k$ drops exponentially in $k$. Score a list by its expected utility $R_a$
  – average $R_a$ over all test users
Item based memory techniques

• How might you do this?
• Instead of a vector for each user, have a vector for each item. Then measure similarity between items to predict whether a user might enjoy/buy related items.
• Amazon.com uses this technique to find items that customers tend to purchase together.

Item similarity based!
Item-Item Collaborative Filtering

Slide credit: John Riedl
Item-Item Collaborative Filtering

Slide credit: John Riedl
Probabilistic Model based algorithms

• Record previous ratings
• Build a probabilistic model based on previous ratings.
• Estimate new ratings based on model.
CF as density estimation
(Breese et al, UAI98)

- Estimate $Pr(R_{ij}=k)$ for each user $i$, movie $j$, and rating $k$
- Use all available data to build model for this estimator

<table>
<thead>
<tr>
<th>$R_{ij}$</th>
<th>Airplane</th>
<th>Matrix</th>
<th>Room with a View</th>
<th>...</th>
<th>Hidalgo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joe</td>
<td>9</td>
<td>7</td>
<td>2</td>
<td>...</td>
<td>7</td>
</tr>
<tr>
<td>Carol</td>
<td>8</td>
<td>?</td>
<td>9</td>
<td>...</td>
<td>?</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Kumar</td>
<td>9</td>
<td>3</td>
<td>?</td>
<td>...</td>
<td>6</td>
</tr>
</tbody>
</table>
CF as density estimation
(Breese et al, UAI98)

- Estimate $Pr(R_{ij}=k)$ for each user $i$, movie $j$, and rating $k$
- Use all available data to build *model* for this estimator
- A simple example:

\[
\forall \text{ movies } j, \ Pr(R_{ij} = k) = \frac{\#(\text{users } i : R_{ij} = k)}{\#(\text{users } i \text{ rating } j)}
\]

Leads to this expected value for unknown $R_{ij}$:

\[
E[R_{ij}] = \sum_k k \cdot Pr(R_{ij} = k) = \text{average rating of movie } j
\]
CF as density estimation
(Breese et al, UAI98)

• Estimate $Pr(R_{ij}=k)$ for each user $i$, movie $j$, and rating $k$
• Use all available data to build model for this estimator
• More complex example:
  • Group users into $M$ “clusters”: $c(1)$, ..., $c(M)$
  • For movie $j$,

\[
Pr(R_{ij} = k \mid i) = \sum_m Pr(R_{ij} = k \mid i \in c(m)) Pr(i \in c(m))
\]

\[
E[R_{ij}] = \sum_m Pr(i \in c(m)) \cdot (\text{average rating of } j \text{ in } c(m))
\]
Remember Clustering?
Clustering

– Clustering - the assignment of objects into groups (called clusters) so that objects from the same cluster are more similar to each other than objects from different clusters.

– Often similarity is assessed according to a distance measure.

– Clustering is a common technique for statistical data analysis, which is used in many fields, including machine learning, data mining, pattern recognition, image analysis and bioinformatics.
Clustering

- Clustering systems:
  - Unsupervised learning
  - Detect patterns in unlabeled data
    - E.g. group emails or search results
    - E.g. find categories of customers
    - E.g. detect anomalous program executions
  - Useful when don’t know what you’re looking for
  - Requires data, but no labels
  - Often get gibberish
Clustering

- Basic idea: group together similar instances
- Example: 2D point patterns

- What could “similar” mean?
Clustering

- Basic idea: group together similar instances
- Example: 2D point patterns

- What could “similar” mean?
  Any of the similarity metrics we talked about before (SSD, angle between vectors)
K-means

1. Ask user how many clusters they'd like. (e.g. $k=5$)
K-means

1. Ask user how many clusters they’d like. *(e.g. k=5)*

2. Randomly guess k cluster Center locations
K-means

1. Ask user how many clusters they’d like. *(e.g. k=5)*

2. Randomly guess k cluster Center locations

3. Each datapoint finds out which Center it’s closest to. (Thus each Center “owns” a set of datapoints)
K-means

1. Ask user how many clusters they’d like. \((e.g. k=5)\)

2. Randomly guess \(k\) cluster Center locations

3. Each datapoint finds out which Center it’s closest to.

4. Each Center finds the centroid of the points it owns
K-means

1. Ask user how many clusters they’d like. \((e.g. \ k=5)\)
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it’s closest to.
4. Each Center finds the centroid of the points it owns...
5. ...and jumps there
6. ...Repeat until terminated!
K-means continues ...
K-means continues

...
K-means continues...
K-means continues...
K-means continues...

Copyright © 2001, 2004, Andrew W. Moore
K-means terminates
CF as density estimation
(Breese et al, UAI98)

- Estimate $Pr(R_{ij} = k)$ for each user $i$, movie $j$, and rating $k$
- Use all available data to build model for this estimator
- More complex example:
  - Group users into $M$ "clusters": $c(1)$, ..., $c(M)$
  - For movie $j$,

\[
E[R_{ij}] = \sum_m \Pr(i \in c(m)) \cdot (\text{average rating of } j \text{ in } c(m))
\]
Some Commercial systems

Amazon
Barnes and Noble
Digg.com
eBay
iLike – music
Internet Movie Database – movies
iTunes – music
Last.fm – music
LibraryThing – books
Musicmatch
MyStrands - developer of social recommendation technologies
Netflix - In order to improve its algorithm Netflix has launched a competition, the Netflix Prize.
StumbleUpon – websites
Threadless - T-shirt
Non-Commercial systems

- AmphetaRate RSS articles
- Everyone's a Critic movies
- GiveALink.org websites
- Gnomoradio music (free)
- Musicmobs music
- Rate Your Music music
Netflix Prize

• An ongoing open competition for the best collaborative filtering algorithm that predicts user ratings for films, based on previous ratings.

• The competition is held by Netflix, an online DVD-rental service, and is opened for anyone (with some exceptions).

• The grand prize of $1,000,000 is reserved for the entry which bests Netflix's own algorithm for predicting ratings by 10%
Netflix Prize

- Netflix provided a training data set of over 100 million ratings that over 480,000 users gave to nearly 18,000 movies. Each training rating is a quadruplet <user, movie, date of grade, grade>. The user and movie fields are integer IDs, while grades are from 1 to 5 (integral) stars.[3]
Netflix Prize

• The qualifying data set contains over 2.8 million triplets <user, movie, date of grade>, with grades known only to the jury.

• A participating team's algorithm must predict grades on the entire qualifying set, but they are only informed of the score for half of the data, the quiz set.

• The other half is the test set, and performance on this is used by the jury to determine potential prize winners. Only the judges know which ratings are in the quiz set, and which are in the test set.
Progress

• The competition began on October 2, 2006. By October 8, a team called WXYZConsulting had already beaten Cinematch's results.

• By October 15, there were three teams who had beaten Cinematch, one of them by 1.06%, enough to qualify for the annual progress prize.

• By June, 2007, over 20,000 teams had registered for the competition from over 150 countries. 2,000 teams had submitted over 13,000 prediction sets.
Progress

On November 13, 2007, team BellKor got 8.43% improvement. The team consisted of three researchers from AT&T Labs, Yehuda Koren, Robert Bell, and Chris Volinsky.

In 2008 BellKor achieved a 9.44% improvement over Cinematch.

On Sept 21, 2009 the $1M grand prize was awarded to BellKor (Pragmatic Chaos).
BellKor

• How did they do it?
  – Factorized neighborhood model.
  – Account for temporal effects in the model.
    • Movie biases – movies go in and out of popularity over time.
    • User biases – users change their baseline ratings over time.
    • User preferences – users change their preferences over time, e.g. a fan of “psychological thrillers” may year later become a fan of “crime dramas”.
  – Blend multiple predictors for the final rating.
BellKor Observations

- Collaborative filtering methods address the sparse set of rating values. However, much accuracy is obtained by also looking at other features of the data. First is the information on which movies each user chose to rate, regardless of specific rating value (“the binary view”).

- Accounting for temporal effects and realizing that parameters describing the data are not static but dynamic functions played an important role in the accuracy of their system.

- You can achieve better accuracy is by blending multiple simple models. The first few predictors have a decisive contribution to improving accuracy, while the rest have a marginal contribution. A lesson here is that having lots of models is useful for the incremental results needed to win competitions, but practically, excellent systems can be built with just a few well-selected models.