Collaborative Filtering

Nearest Neighbor Approach
Bad news

Netflix Prize data no longer available to public.

- Just after contest ended in July 2009:
  - Plans for Netflix Prize 2 contest were announced
  - Contest data was made available for further public research at UC Irvine repository

- But a few months later:
  - Netflix was being sued for supposed privacy breaches connected with contest data
  - FTC was investigating privacy concerns

- By March 2010:
  - Netflix had settled the lawsuit privately
  - Withdrawn the contest data from public use
  - Cancelled Netflix Prize 2
Good news

An older movie rating dataset from GroupLens is still available, and perfectly suitable for the CSS 490 / 590 project.

- Consists of data collected through the MovieLens movie rating website.
- Comes in 3 sizes:
  - MovieLens 100k
  - MovieLens 1M
  - MovieLens 10M

http://www.grouplens.org/node/12
http://movielens.umn.edu/login
MovieLens 100k dataset properties

- 943 users
- 1682 movies
- 100,000 ratings
- 1 - 5 rating scale
- Rating matrix is 6.3% occupied
- Ratings per user
  - min = 20
  - mean = 106
  - max = 737
- Ratings per movie
  - min = 1
  - mean = 59
  - max = 583
Recommender system definition

**DOMAIN**: some field of activity where *users* buy, view, consume, or otherwise experience *items*

**PROCESS**:  
1. *users* provide *ratings* on *items* they have experienced  
2. Take all *< user, item, rating >* data and build a predictive model  
3. For a *user* who hasn’t experienced a particular *item*, use model to *predict* how well they will like it (i.e. *predict rating*)
Types of recommender systems

Predictions can be based on either:

- **content-based** approach
  - *explicit* characteristics of users and items

- **collaborative filtering** approach
  - *implicit* characteristics based on similarity of users’ preferences to those of other users
Collaborative filtering algorithms

- Common types:
  - Global effects
  - Nearest neighbor
  - Matrix factorization
  - Restricted Boltzmann machine
  - Clustering
  - Etc.
Nearest neighbor in action

<table>
<thead>
<tr>
<th>User</th>
<th>Movie 1</th>
<th>Movie 2</th>
<th>Movie 3</th>
<th>Movie 4</th>
<th>Movie 5</th>
<th>Movie 6</th>
<th>Movie 7</th>
<th>Movie 8</th>
<th>Movie 9</th>
<th>...</th>
<th>Movie 17770</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>User 2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>?</td>
</tr>
<tr>
<td>User 3</td>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User 4</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>User 5</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>User 6</td>
<td></td>
<td>2</td>
<td></td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>User 7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>User 8</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>User 9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User 10</td>
<td>1</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User 480189</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Identical preferences – strong weight

Similar preferences – moderate weight
Nearest neighbor classifiers

Requires three inputs:
1. The set of stored samples
2. Distance metric to compute distance between samples
3. The value of $k$, the number of nearest neighbors to retrieve
Nearest neighbor classifiers

To classify unknown record:

1. Compute distance to other training records
2. Identify $k$ nearest neighbors
3. Use class labels of nearest neighbors to determine the class label of unknown record (e.g., by taking majority vote)
Nearest neighbor classification

- Compute distance between two points
  - Example: Euclidean distance
    \[ d(x, y) = \sqrt{\sum_i (x_i - y_i)^2} \]

- Options for determining the class from nearest neighbor list
  - Take majority vote of class labels among the \( k \)-nearest neighbors
  - Weight the votes according to distance
    - example: weight factor \( w = 1 / d^2 \)
Nearest neighbor in collaborative filtering

- For our implementation in Project 2:
  - Actually a regression, not a classification.
    - Prediction is a weighted combination of neighbor’s ratings (real number).
  - We consider all neighbors, not the k-nearest subset of neighbors.
    - Since we’re not ranking neighbors by distance, distance no longer relevant.
  - Instead of distance, we calculate similarities that determine weightings of each neighbor’s rating.
Nearest neighbor in action

- For this example:
  - Find every user that has rated movie 10
  - Compute similarity between user 2 and each of those users
  - Weight those users’ ratings according to their similarities
  - Predicted rating for user 2 is sum of other users’ weighted ratings on movie 10

<table>
<thead>
<tr>
<th>User</th>
<th>Movie 1</th>
<th>Movie 2</th>
<th>Movie 3</th>
<th>Movie 4</th>
<th>Movie 5</th>
<th>Movie 6</th>
<th>Movie 7</th>
<th>Movie 8</th>
<th>Movie 9</th>
<th>Movie 10</th>
<th>...</th>
<th>Movie 1777</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User 2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User 3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User 4</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User 5</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User 6</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User 7</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User 8</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User 9</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Identical preferences – strong weight
Similar preferences – moderate weight
Measuring similarity of users

- For Project 2 we will use *Pearson’s correlation coefficient* (PCC) as a measure of similarity between users.

- Pearson’s correlation coefficient is covariance normalized by the standard deviations of the two variables:

$$\text{corr}(x, y) = \frac{\text{cov}(x, y)}{\sigma_x \sigma_y}$$

  - Always lies in range -1 to 1
Measuring similarity of users

- PCC similarity for two users $a$ and $b$:

$$PCC(a, b) = \frac{\sum_{j=1}^{n} (r_{a,j} - \bar{r}_a)(r_{b,j} - \bar{r}_b)}{\sqrt{\sum_{j=1}^{n} (r_{a,j} - \bar{r}_a)^2} \sqrt{\sum_{j=1}^{n} (r_{b,j} - \bar{r}_b)^2}}$$

- Both sums are taken over only those movies rated by both $a$ and $b$ (indexed by $j$)
- $r_{a,j} = \text{rating by user } a \text{ on movie } j$
- $\bar{r}_a = \text{average rating on all movies rated by user } a$
- $n = \text{number of movies rated by both } a \text{ and } b$
Measuring similarity of users

- Calculating PCC on sparse matrix
  - Calculate user average rating using only those cells where a rating exists.
  - Subtract user average rating only from those cells where rating exists.
  - Calculate and sum user-user cross-products and user deviations from average only for those movies where a rating exists for both users.