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  - Withdrawn the contest data from public use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 581 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 <u>*users*</u> buy, view, consume, or otherwise experience <u>*items*</u>

PROCESS:

1. *users* provide <u>*ratings*</u> 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 <u>*predict*</u> how well they will like it (i.e. *predict rating*)

Types of recommender systems

Predictions can be based on either:

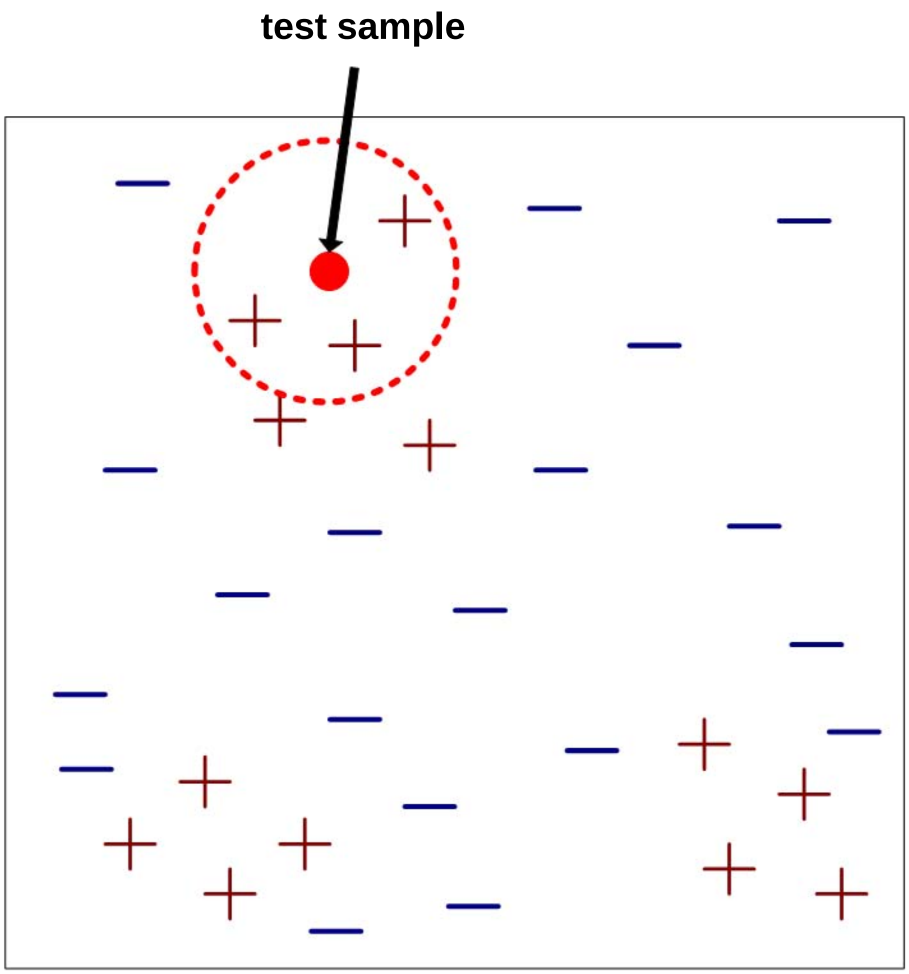
- content-based approach
  - *explicit* characteristics of users and items
- collaborative filtering approach
  - *implicit* characteristics based on similarity of users' preferences to those of other users

Collaborative filtering algorithms

- Common types:
  - Global effects
  - Nearest neighbor
  - Matrix factorization
  - Restricted Boltzmann machine
  - Clustering
  - etc.

Project 2 will explore types highlighted in red.

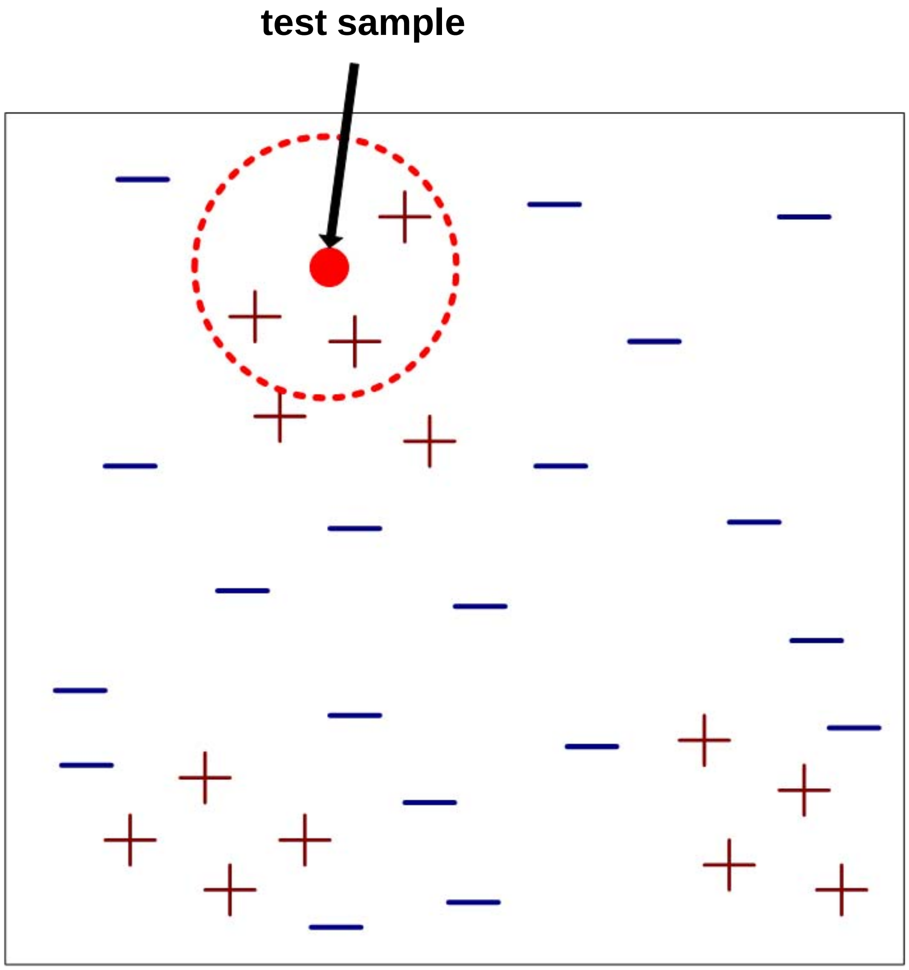
Nearest neighbor classifiers



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 test sample:

1. Compute distances to samples in training set
2. Identify *k* nearest neighbors
3. Use class labels of nearest neighbors to determine class label of test sample (e.g. by taking majority vote)

Nearest neighbor classification

- Compute distance between two points
  - Example: Euclidean distance

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_i (x_i - y_i)^2}$$

- 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 both all neighbors and various k-nearest subsets of neighbors.
  - Instead of distances, we calculate similarities that are used to:
    - rank neighbors to determine k nearest subset
    - compute weightings of each neighbor's rating

Nearest neighbor in action

- For this example:
  - Find <u>every</u> 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

	movie 1	movie 2	movie 3	movie 4	movie 5	movie 6	movie 7	movie 8	movie 9	movie 10	…	movie 17770
user 1			1		2							3
user 2		2		3	3			4		?		
user 3							5	3				
user 4	2				3			2				2
user 5		2		3		5		4		2		4
user 6			2									
user 7			2					4	2			
user 8	3	1			3	4		5		4		
user 9									3			

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:

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

  – 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*)
  - $r_{a,j}$ = rating by user *a* on movie *j*
  - $\bar{r}_a$ = average rating on all movies rated by user *a*
  - n = number of movies rated by both *a* and *b*

Measuring similarity of users

- PCC similarity for two users *a* and *b*, where ratings have first been mean-centered for each user:

$$PCC(a,b) = \frac{\sum_{j=1}^{n} m_{a,j} m_{b,j}}{\sqrt{\sum_{j=1}^{n} m_{a,j}{}^{2}} \sqrt{\sum_{j=1}^{n} m_{b,j}{}^{2}}}$$

  - Both sums are taken over only those movies rated by both *a* and *b* (indexed by *j*)
  - $m_{a,j}$ = mean-centered rating by user *a* on movie *j*
  - n = number of movies rated by both *a* and *b*

Mesauring similarity of users

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

	movie 1	movie 2	movie 3	movie 4	movie 5	movie 6	movie 7	movie 8	movie 9	movie 10	…	movie 17770
user 1			1		2							3
user 2		2		3	3			4		?		
user 3							5	3				
user 4	2				3			2				2
user 5		2		3		5		4		2		4
user 6			2									
user 7			2					4	2			
user 8	3	1			3	4		5		4		
user 9									3			

Identical preferences – strong weight

Similar preferences – moderate weight