TWO COLLABORATIVE FILTERING PROJECTS FOR CSC2515

What is collaborative filtering (CF)

- The goal of collaborative filtering is to predict the preferences of a given user given a large collection of user preferences.

- For example:
  - Suppose you infer from the data that most of the users who like "Star Wars" also like "Lord of the Rings".

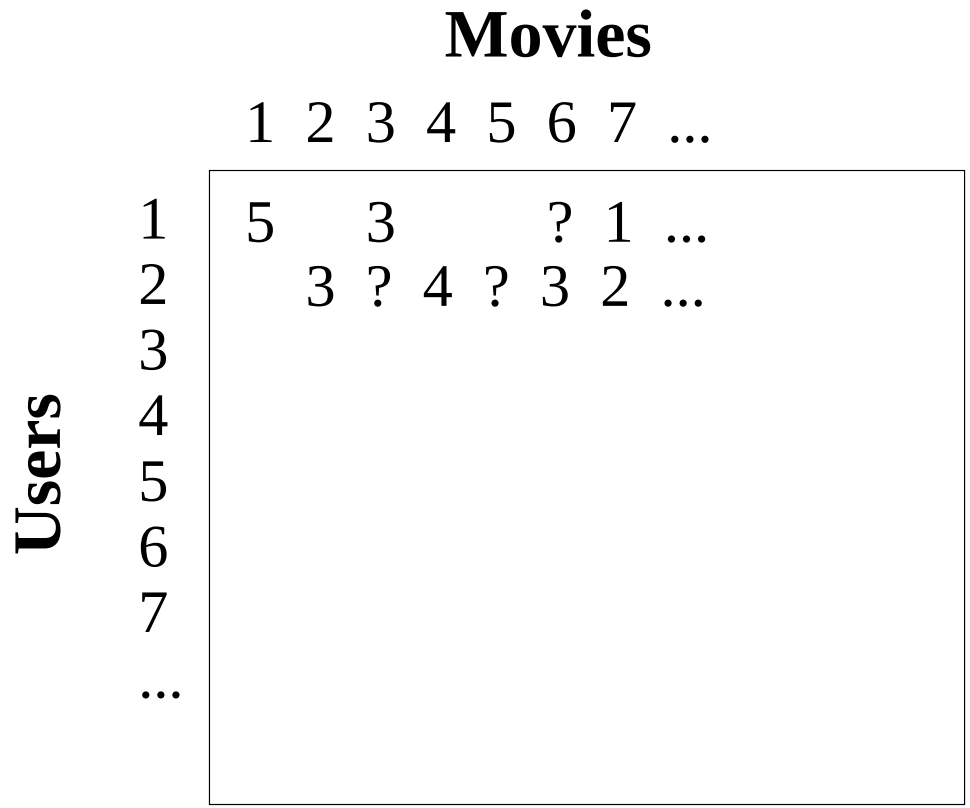
  - Then if a user watched and liked "Star Wars" you would recommend him/her to see "Lord of the Rings".

Netflix movie rating prediction competition

- A year ago, Netflix announced a movie rating predictions competition.
- Whoever improves Netflix's own baseline score by 10% will win the 1 million dollar prize.
- The training data set consists of 100,480,507 ratings from 480,189 randomly-chosen, anonymous users on 17,770 movie titles. The data is very sparse, most users rate only few movies.
- Also, Netflix provides a test set containing 2,817,131 user/movie pairs with the ratings withheld. The goal is to predict those ratings as accurately as possible.

Netflix movie-rating competition

- The goal is to correctly predict test ratings.



Projects

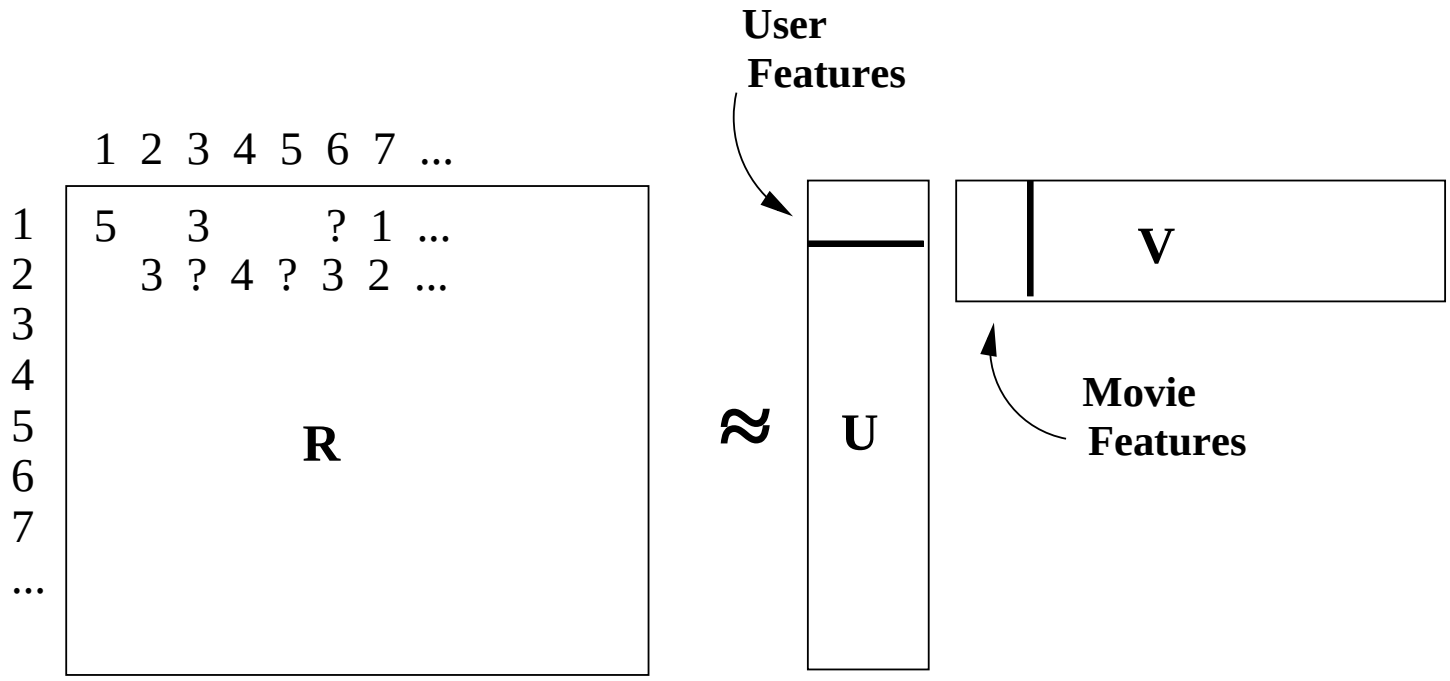
- We will provide you with a subset of the Netflix training data: a few thousand users + a few thousand movies, so that you can easily run your algorithms on CDF machines.

- We will also provide you with a validation set. You will report the achieved prediction accuracy on this validation set.

- There will be two projects based on the following two models:
  - Probabilistic Matrix Factorization (PMF)
  - Restricted Boltzmann Machines (RBM's)

- You can choose which model you would like to work on.

Probabilistic Matrix Factorization



- Let R_{ij} represent the rating of user i for movie j. The row and column vectors U_i and V_j represent user-specific and movie-specific latent feature vectors respectively.

- The model:

$$p(R_{ij}|U_i, V_j, \sigma^2) = \mathcal{N}(R_{ij}|U_i^T V_j, \sigma^2)$$

Probabilistic Matrix Factorization

- The model:

$$p(R_{ij}|U_i, V_j, \sigma^2) = \mathcal{N}(R_{ij}|U_i^T V_j, \sigma^2)$$

- To avoid severe overfitting, place a zero-mean spherical Gaussian prior on user and movie feature vectors:

$$p(U_i|\sigma_U^2) = \mathcal{N}(U_i|0, \sigma_U^2 \mathbf{I}), \quad p(V_j|\sigma_V^2) = \mathcal{N}(V_j|0, \sigma_V^2 \mathbf{I}).$$

- and do MAP estimation of U and V. See Bishop, Chapter 3.

Restricted Boltzmann Machines (RBM's)

- A different way to build a good recommendation system is to use RBM's.
- RBM's will be covered in the class on Oct 30th, but you can look up a Scholarpedia entry on "Boltzmann machine"
- Sample code for training RBM's on binary inputs is available at: http://www.cs.toronto.edu/~hinton/MatlabForSciencePaper.html
- Sample code for training RBM's on binary inputs is available at:

What we expect from these projects

- If you decide to work on this project, come talk to one of the TAs.
- You do not have to choose one of the suggested models. You can use your other favorite machine learning model (e.g. a neural net).
- We expect a clear write-up of your model (description of the model, choice of priors, hyper-priors, model training, etc.) as well as an analysis of the model's performance. For example, you can analyse how the model performs as a function of user-rating frequency.
- You model must perform better than a very simple model that predicts a combination of user and movie averages.
- If you are feeling ambitious, you can enter the Netflix competition and try to win the 1 million dollar Grand Prize.