

# Collaborative Filtering

Mathieu Lagrange



slides from Lester Mackey and Aleksandr Simma (used with permission)



March 23, 2021

# Outline

- ① Problem Formulation
  - Centering
  - Shrinkage
- ② Preliminaries
  - Naive Bayes
  - KNN
- ③ Classification/Regression
  - SVD
- ④ Low Dimensional Matrix Factorization
  - References

# What is Collaborative Filtering?

## Group of users



## Group of items



# What is Collaborative Filtering?

## Group of users



## Group of items



- ⌘ Observe some user-item preferences
- ⌘ Predict new preferences:

# Does Bob like strawberries???

# Collaborative Filtering in the Wild...

**Amazon.com** recommends products based on purchase history

Amazon.com: Recommended for You

Hello, LESTER We have recommendations for you. (Not LESTER?)

LESTER's Amazon.com Today's Deals Gifts & Wish Lists Gift Cards

Shop All Departments Search All Departments

LESTER's Amazon.com Your Browsing History Recommended For You Rate These Items Improve Your Recommendations Your Profile Your Communities Learn More

LESTER, Welcome to Your Amazon.com (If you're not LESTER, W. MACKEY JR., click here.)

Today's Recommendations For You

Here's a daily sample of items recommended for you. Click here to [see all recommendations](#).

Page 1 of 22

<p><b>Asymptotic Theory of Statistics...</b> (Hardcover) by Arindam DasGupta ★★★★☆ (2) \$71.96 <a href="#">Fix this recommendation</a></p>	<p><b>Computer Networking: A Top-Down...</b> (Hardcover) by James F. Kurose ★★★★☆ (67) \$80.90 <a href="#">Fix this recommendation</a></p>	<p><b>Second Foundation (Foundation...</b> (Mass Market Paperback) by Isaac Asimov ★★★★☆ (72) \$7.99 <a href="#">Fix this recommendation</a></p>	<p><b>Learning About Language...</b> (Paperback) by Kathleen Bailey \$22.77 <a href="#">Fix this recommendation</a></p>	<p><b>Introduction to Probability...</b> (Hardcover) by Sheldon M. Ross ★★★★☆ (24) \$79.96 <a href="#">Fix this recommendation</a></p>	<p><b>Hyperion (Mass Market Paperback)</b> by Dan Simmons ★★★★☆ (495) \$7.99 <a href="#">Fix this recommendation</a></p>
--	--	--	---	--	--

Any Category Action & Adventure Adventure Artificial Intelligence Calculus Card, Orson Scott Computer Mathematics Contemporary Disciplines Epic Fantasy Fiction L'Engle, Madeleine Linear Linear Programming Linguistics Literature & Fiction Mathematical Analysis Mathematical Physics Methods Operating Systems Professional & Technical Programming Languages Science Fiction, Fantasy, & Magic Software Development Stochastic Modeling

Linder et al., 2003



# Collaborative Filtering in the Wild...

Web Images Videos Maps News Shopping Gmail more ▼

Google news  Search News Search the Web [Advanced news search](#) [Preferences](#)

Personalized ▼ Recommended for @gmail.com

Top Stories

**Recommended**

U.S.  
World  
SciTech  
Business  
Sports  
Entertainment  
Spotlight  
Health  
Most Popular

▼ All news  
[Headlines](#)  
[Images](#)

**Obama Nobel Peace Prize: Obama wins, and partisan fighting continues**  
Chicago Tribune - Mark Z. Barabak, Geraldine Baum - 45 minutes ago  
President Barack Obama's winning of the Nobel Peace Prize brought nothing of the sort at home, as political combatants were quick to assume their usual battles: Democrats largely hailed the ...  
Video: Did Obama Deserve Nobel Prize? CBS  
If Obama can get one, you can, too Detroit Free Press  
New York Times - Philadelphia Inquirer - Fort Worth Star Telegram - Wikipedia: 2009 Nobel Peace Prize  
[all 10,171 news articles](#) [Email this story](#)

**US Senate panel votes to extend security law**  
Reuters - Thomas Ferraro, Anthony Roadie - Oct 8, 2009  
WASHINGTON (Reuters) - A Senate Judiciary Committee, drawing criticism from both liberals and conservatives, voted on Thursday to extend expiring provisions of a post-September 11 law designed to protect the United States from another attack.  
AP Interview: White House expands climate campaign The Associated Press  
US Senate Panel Unlikely To Debate CO2 Bill Before Nov Wall Street Journal  
New York Times - Houston Chronicle - Politico - Red, Green and Blue  
[all 265 news articles](#) [Email this story](#)

**Barnes & Noble May Sell Its Own E-reader**  
PC World - Harry McCracken - Oct 9, 2009  
Is bookstore behemoth Barnes & Noble about to enter the e-book fray with its own Android-powered device? I like these rumors: The Wall Street Journal is reporting that bookstore behemoth Barnes & Noble will soon start selling its own e-reader device, ...  
Barnes & Noble's E-Reader Gets Real Wired News  
Barnes & Noble's Sales Down In Aug-Sep, New View Given Wall Street Journal  
CNET News - San Francisco Chronicle - Register - FQNews  
[all 208 news articles](#) [Email this story](#)

**Dow Ends Week at Highest Level in a Year**  
Washington Post - 3 hours ago  
US stocks gained last week, pushing the Dow Jones industrial average to its highest close in a year, as Alcoa unexpectedly reported a profit and economic data signaled the US recession is ending.  
Dun of IBM, Intel Propels Dow's Run Wall Street Journal  
Stocks Finish with Gains BusinessWeek  
Bloomberg - Reuters - The Associated Press  
[all 178 news articles](#) [Email this story](#)

⇒ Google News recommends new articles based on click and search history

⇒ Millions of users, millions of articles

Das et al., 2007

## Collaborative Filtering in the Wild...

**Netflix** predicts other “Movies You’ll ♥” based on past numeric ratings (1-5 stars)

The screenshot shows the Netflix homepage with a red header. Navigation tabs include 'Browse DVDs', 'Browse Instant', 'Your Queue', 'Movies You'll ♥', 'Friends & Community', and 'DVD Sale \$5.99'. A search bar contains the text 'Movies, actors, directors, genres' and a 'Search' button. Below the navigation bar, there are links for 'Home', 'Genres', 'New Releases', 'Netflix Top 100', 'Critics' Picks', and 'Award Winners'. The main content area features a section titled 'Because you enjoyed:' with links to 'Chinatown', 'Vertigo', and 'Dr. Strangelove'. Below this, it says 'We think you'll enjoy: The Last Laugh' with an 'Add' button. A movie poster for 'THE LAST LAUGH' is displayed with a 5-star rating and a 'Not Interested' link. To the right, the 'YOUR RECENT ACTIVITY' section shows a list of movies with status indicators like 'We shipped', 'We received', and 'We received'. Below that, the 'SUGGESTIONS FOR YOU' section states 'You have new suggestions in Movies You'll ♥'.

- ⌘ Recommendations drive 60% of Netflix’s DVD rentals
- ⌘ Mostly smaller, independent movies (Thompson 2008)

<http://www.netflix.com>

# Collaborative Filtering in the Wild...



⌘ **Netflix Prize:**  
Beat Netflix  
recommender system,  
using Netflix data →

Win \$1 million

⌘ **Data:**  
480,000 users  
18,000 movies  
100 million observed  
ratings = only 1.1% of  
ratings observed

“The Netflix Prize seeks to substantially improve the accuracy of predictions about how much someone is going to love a movie based on their movie preferences.”



# What is Collaborative Filtering?

**Insight:** Personal preferences are correlated

- ⌘ If Jack loves A and B, and Jill loves A, B, and C, then Jack is more likely to love C

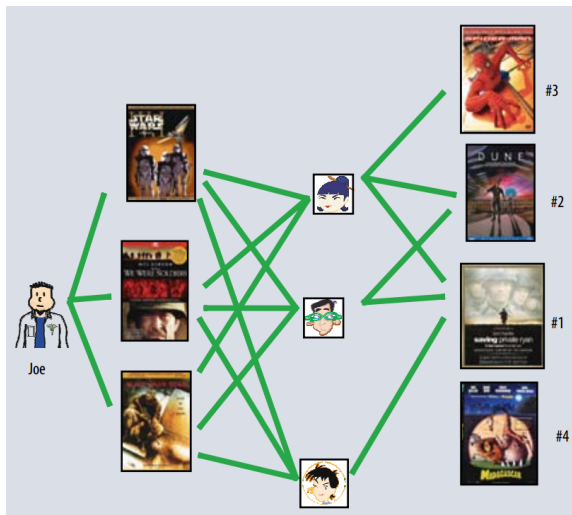
## Collaborative Filtering Task

- ⌘ Discover patterns in observed preference behavior (e.g. purchase history, item ratings, click counts) across community of users
- ⌘ Predict new preferences based on those patterns

Does not rely on item or user attributes (e.g. demographic info, author, genre)

- ⌘ Content-based filtering: complementary approach

# What is Collaborative Filtering?



# What is Collaborative Filtering?

## Given:

- ⊞ Users  $u \in \{1, \dots, U\}$
- ⊞ Items  $i \in \{1, \dots, M\}$
- ⊞ Training set  $\mathcal{T}$  with observed, real-valued preferences  $r_{ui}$  for some user-item pairs  $(u, i)$ 
  - ⊞  $r_{ui}$  = e.g. purchase indicator, item rating, click count ...

## Goal: Predict unobserved preferences

- ⊞ Test set  $Q$  with pairs  $(u, i)$  not in  $\mathcal{T}$

View as matrix completion problem

- ⊞ Fill in unknown entries of sparse preference matrix

$$\mathbf{R} = \underbrace{\begin{bmatrix} ? & ? & 1 & \dots & 4 \\ 3 & ? & ? & \dots & ? \\ ? & 5 & ? & \dots & 5 \end{bmatrix}}_{M \text{ items}} \Bigg\} U \text{ users}$$

# What is Collaborative Filtering?

## Measuring success

- Interested in error on unseen test set  $Q$ , not on training set
- For each  $(u, i)$  let  $r_{ui}$  = true preference,  $\hat{r}_{ui}$  = predicted preference
- Root Mean Square Error**

$$\text{RMSE} = \sqrt{\frac{1}{|Q|} \sum_{(u,i) \in Q} (r_{ui} - \hat{r}_{ui})^2}$$

- Mean Absolute Error

$$\text{MAE} = \frac{1}{|Q|} \sum_{(u,i) \in Q} |r_{ui} - \hat{r}_{ui}|$$

- Ranking-based objectives

- e.g. What fraction of true top-10 preferences are in predicted top 10?

# Centering Your Data

## ⌘ What?

- ⌘ Remove bias term from each rating before applying CF methods:  $\tilde{r}_{ui} = r_{ui} - b_{ui}$

## ⌘ Why?

- ⌘ Some users give systematically higher ratings
- ⌘ Some items receive systematically higher ratings
- ⌘ Many interesting patterns are in variation around these systematic biases
- ⌘ Some methods assume mean-centered data
  - ⌘ Recall PCA required mean centering to measure variance around the mean

# Centering Your Data

## ⌘ What?

- ⌘ Remove bias term from each rating before applying CF methods:  $\tilde{r}_{ui} = r_{ui} - b_{ui}$

## ⌘ How?

- ⌘ Global mean rating

- ⌘  $b_{ui} = \mu := \frac{1}{|\mathcal{T}|} \sum_{(u,i) \in \mathcal{T}} r_{ui}$

- ⌘ Item's mean rating

- ⌘  $b_{ui} = b_i := \frac{1}{|R(i)|} \sum_{u \in R(i)} r_{ui}$

- ⌘  $R(i)$  is the set of users who rated item  $i$

- ⌘ User's mean rating

- ⌘  $b_{ui} = b_u := \frac{1}{|R(u)|} \sum_{i \in R(u)} r_{ui}$

- ⌘  $R(u)$  is the set of items rated by user  $u$

- ⌘ Item's mean rating + user's mean deviation from item mean

- ⌘  $b_{ui} = b_i + \frac{1}{|R(u)|} \sum_{i \in R(u)} (r_{ui} - b_i)$

# Shrinkage

- ⌘ What?
  - ⌘ Interpolating between an estimate computed from data and a fixed, predetermined value
- ⌘ Why?
  - ⌘ Common task in CF: Compute estimate (e.g. a mean rating) for each user/item
  - ⌘ Not all estimates are equally reliable
  - ⌘ Some users have orders of magnitude more ratings than others
  - ⌘ Estimates based on fewer datapoints tend to be noisier

	A	B	C	D	E	F	User mean
<b>R</b> = Alice	2	5	5	4	3	5	4
Bob	2	?	?	?	?	?	2
Craig	3	3	4	3	?	4	3.4

- ⌘ Hard to trust mean based on one rating

# Shrinkage

## ⌘ What?

- ⌘ Interpolating between an estimate computed from data and a fixed, predetermined value

## ⌘ How?

- ⌘ e.g. Shrunk User Mean:

$$\tilde{b}_u = \frac{\alpha}{\alpha + |R(u)|} * \mu + \frac{|R(u)|}{\alpha + |R(u)|} * b_u$$

- ⌘  $\mu$  is the global mean,  $\alpha$  controls degree of shrinkage
- ⌘ When user has many ratings,  $\tilde{b}_u \approx$  user's mean rating
- ⌘ When user has few ratings,  $\tilde{b}_u \approx$  global mean rating

		A	B	C	D	E	F	User mean	Shrunk mean
<b>R =</b>	Alice	2	5	5	4	3	5	4	3.94
	Bob	2	?	?	?	?	?	2	2.79
	Craig	3	3	4	3	?	4	3.4	3.43

Global mean  $\mu = 3.58$ ,  $\alpha = 1$



# Classification/Regression for CF

**Interpretation:** CF is a set of  $M$  classification/regression problems, one for each item

- ⌘ Consider a fixed item  $i$
- ⌘ Treat each user as incomplete vector of user's ratings for all items except  $i$ :  $\vec{r}_u = (3, ?, ?, 4, ?, 5, ?, 1, 3)$
- ⌘ Class of each user w.r.t. item  $i$  is the user's rating for item  $i$  (e.g. 1, 2, 3, 4, or 5)
- ⌘ Predicting rating  $r_{ui} \equiv$  Classifying user vector  $\vec{r}_u$

# Classification/Regression for CF

## Approach:

- ✧ Choose your favorite classifier/regression algorithm
- ✧ Train separate predictor for each item
- ✧ To predict  $r_{ui}$  for user  $u$  and item  $i$ , apply item  $i$ 's predictor to vector of user  $u$ 's incomplete ratings vector

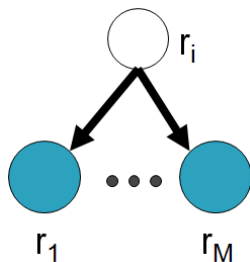
## Pros:

- ✧ Reduces CF to a well-known, well-studied problem
- ✧ Many good prediction algorithms available

## Cons:

- ✧ Predictor must handle missing data (unobserved ratings)
- ✧ Training  $M$  independent predictors can be expensive
- ✧ Approach may not take advantage of problem structure
  - ✧ Item-specific subproblems are often related

# Naive Bayes Classifier



- ⌘ Treat distinct rating values as classes
- ⌘ Consider classification for item  $i$
- ⌘ Main assumption
  - ⌘ For any items  $j \neq k \neq i$ ,  $r_j$  and  $r_k$  are conditionally independent given  $r_i$
  - ⌘ When we know rating  $r_{ui}$  all of a user's other ratings are independent
- ⌘ Parameters to estimate
  - ⌘ Prior class probabilities:  $P(r_i = v)$
  - ⌘ Likelihood:  $P(r_j = w | r_i = v)$

# Naive Bayes Classifier

Train classifier with all users who have rated item  $i$

⌘ Use counts to estimate prior and likelihood

$$P(r_i = v) = \frac{\sum_{u=1}^U \mathbf{1}(r_{ui} = v)}{\sum_{w=1}^V \sum_{i=1}^U \mathbf{1}(r_{ui} = w)}$$

$$P(r_j = w | r_i = v) = \frac{\sum_{u=1}^U \mathbf{1}(r_{ui} = v, r_{uj} = w)}{\sum_{z=1}^V \sum_{u=1}^U \mathbf{1}(r_{ui} = v, r_{uj} = z)}$$

⌘ Complexity

⌘  $O(\sum_{u=1}^U |R(u)|^2)$  time and  $O(M^2 V^2)$  space for all items

Predict rating for  $(u, i)$  using posterior

$$P(r_{ui} = v | r_{u1}, \dots, r_{uM}) = \frac{P(r_{ui} = v) \prod_{j \neq i} P(r_{uj} | r_{ui} = v)}{\sum_{w=1}^V P(r_{ui} = w) \prod_{j \neq i} P(r_{uj} | r_{ui} = w)}$$

# Naive Bayes Summary

## Pros:

- ⌘ Easy to implement
- ⌘ Off-the-shelf implementations readily available

## Cons:

- ⌘ Large space requirements when storing parameters for all  $M$  predictors
- ⌘ Makes strong independence assumptions
- ⌘ Parameter estimates will be noisy for items with few ratings

⌘ E.g.  $P(r_j = w | r_i = v) = 0$  if no user rated both  $i$  and  $j$

## Addressing cons:

- ⌘ Tie together parameter learning in each item's predictor
- ⌘ Shrinkage/smoothing is an example of this

# K Nearest Neighbor Methods

Most widely used class of CF methods

- ⌘ Flavors: **Item-based** and User-based
- ⌘ Represent each item as incomplete vector of user ratings:  
 $\vec{r}_i = (3, ?, ?, 4, ?, 5, ?, 1, 3)$
- ⌘ To predict new rating  $r_{ui}$  for query user  $u$  and item  $i$ :
  - 1 Compute similarity between  $i$  and every other item
  - 2 Find  $K$  items rated by  $u$  most similar to  $i$
  - 3 Predict weighted average of similar items' ratings
- ⌘ Intuition: Users rate similar items similarly.

# KNN: Computing Similarities

How to measure similarity between items?

⌘ Cosine similarity

$$S(\vec{r}_i, \vec{r}_j) = \frac{\langle \vec{r}_i, \vec{r}_j \rangle}{\|\vec{r}_i\| \|\vec{r}_j\|}$$

⌘ Pearson correlation coefficient

$$S(\vec{r}_i, \vec{r}_j) = \frac{\langle \vec{r}_i - \text{mean}(\vec{r}_i), \vec{r}_j - \text{mean}(\vec{r}_j) \rangle}{\|\vec{r}_i - \text{mean}(\vec{r}_i)\| \|\vec{r}_j - \text{mean}(\vec{r}_j)\|}$$

⌘ Inverse Euclidean distance

$$S(\vec{r}_i, \vec{r}_j) = \frac{1}{\|\vec{r}_i - \vec{r}_j\|}$$

**Problem:** These measures assume complete vectors

**Solution:** Compute over subset of users rated by both items

Complexity:  $O(\sum_u |R(u)|^2)$  time

# KNN: Choosing K neighbors

How to choose  $K$  nearest neighbors?

⚡ Select  $K$  items with largest similarity score to query item  $i$

**Problem:** Not all items were rated by query user  $u$

**Solution:** Choose  $K$  most similar items rated by  $u$

Complexity:  $O(\min(KM, M \log M))$

Herlocker et al., 1999



## KNN: Forming Weighted Predictions

Predicted rating for query user  $u$  and item  $i$

- ⌘  $N(i; u)$  is the *neighborhood* of item  $i$  for user  $u$ 
  - ⌘ i.e. the  $K$  most similar items rated by  $u$
- ⌘  $\hat{r}_{ui} = b_{ui} + \sum_{N(i;u)} w_{ij}(r_{uj} - b_{uj})$

How to choose weights for each neighbor?

- ⌘ Equal weights:  $w_{ij} = \frac{1}{|N(i;u)|}$
- ⌘ Similarity weights:  $w_{ij} = \frac{S(i,j)}{\sum_{j \in N(i;u)} S(i,j)}$  (Herlocker et al., 1999)
- ⌘ Learn optimal weights for each user (Bell and Koren, 2007)
- ⌘ Learn optimal global weights (Koren, 2008)

Complexity:  $O(K)$

# KNN: User Optimized Weights

**Intuition:** For a given query user  $u$  and item  $i$ , choose weights that best predict other known ratings of item  $i$  using only  $N(i; u)$ :

$$\min_{\mathbf{w}_i} \sum_{s \in R(i), s \neq u} \left( r_{si} - \sum_{j \in N(i; u)} w_{ij} r_{sj} \right)^2$$

With no missing ratings, this is a linear regression problem:

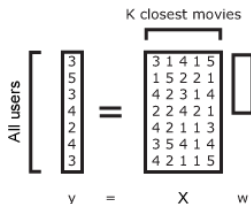
K closest movies

$$\begin{array}{c} \text{All users} \end{array} \begin{bmatrix} 3 \\ 5 \\ 3 \\ 4 \\ 2 \\ 4 \\ 3 \end{bmatrix} = \begin{array}{c} \begin{bmatrix} 3 & 1 & 4 & 1 & 5 \\ 1 & 5 & 2 & 2 & 1 \\ 4 & 2 & 3 & 1 & 4 \\ 2 & 2 & 4 & 2 & 1 \\ 4 & 2 & 1 & 1 & 3 \\ 3 & 5 & 4 & 1 & 4 \\ 4 & 2 & 1 & 1 & 5 \end{bmatrix} \end{array} \begin{bmatrix} \\ \\ \\ \\ \\ \\ \end{bmatrix}$$

$y = Xw$

# KNN: User Optimized Weights

- ⌘ Optimal solution:  $w = A^{-1}b$  for  $A = X^T X, b = X^T y$
- ⌘ **Problem:**  $X$  contains missing entries
  - ⌘ Not all items in  $N(i; u)$  were rated by all users
- ⌘ **Solution:** Approximate  $A$  and  $b$



Bell and Koren, 2007

$$\hat{A}_{jk} = \frac{\sum_{s \in R(j) \cap R(k)} r_{sj} r_{sk}}{|R(j) \cap R(k)|}$$

$$\hat{b}_k = \frac{\sum_{s \in R(i) \cap R(k)} r_{si} r_{sk}}{|R(i) \cap R(k)|}$$

$$\hat{w} = \hat{A}^{-1} \hat{b}$$

- ⌘ Estimates based on users who rated each pair of items

# KNN: User Optimized Weights

## Benefits

- ⌘ Weights optimized for the task of rating prediction
  - ⌘ Not just borrowed from the neighborhood selection phase
- ⌘ Weights not constrained to sum to 1
  - ⌘ Important if all nearest neighbors are dissimilar
- ⌘ Weights derived simultaneously
  - ⌘ Accounts for correlations among neighbors
- ⌘ Outperforms KNN with similarity or equal weights
- ⌘ Can compute entries of  $\hat{A}$  and  $\hat{b}$  offline in parallel

## Drawbacks

- ⌘ Must solve additional  $K \times K$  system of linear equations per query

## KNN: Globally Optimized Weights

Consider the following KNN prediction rule for query  $(u, i)$ :

$$\hat{r}_{ui} = b_{ui} + |N(i; u)|^{-\frac{1}{2}} \sum_{j \in N(i; u)} w_{ij} (r_{uj} - b_{uj})$$

Could learn a single set of KNN weights  $w_{ij}$ , shared by all users, that minimize regularized MSE:

$$E = \frac{1}{|\mathcal{T}|} \sum_{(u, i) \in \mathcal{T}} \frac{1}{2} (\hat{r}_{ui} - r_{ui})^2 + \lambda \sum_{i=1}^M \sum_{j=1}^M \frac{1}{2} w_{ij}^2 = \frac{1}{|\mathcal{T}|} \sum_{(u, i) \in \mathcal{T}} E_{ui}$$

Optimize objective using **stochastic gradient descent**:

✚ For each example  $(u, i) \in \mathcal{T}$ , update  $w_{ij} \forall j \in N(i; u)$

$$\begin{aligned} w_{ij}^{t+1} &= w_{ij}^t - \gamma \frac{\partial}{\partial w_{ij}} E_{ui} \\ &= w_{ij}^t - \gamma (|N(i; u)|^{-\frac{1}{2}} (\hat{r}_{ui} - r_{ui}) (r_{uj} - b_{uj}) + \lambda w_{ij}^t) \end{aligned}$$

# KNN: Globally Optimized Weights

## Benefits

- ⌘ Weights optimized for the task of rating prediction
  - ⌘ Not just borrowed from the neighborhood selection phase
- ⌘ Weights not constrained to sum to 1
  - ⌘ Important if all nearest neighbors are dissimilar
- ⌘ Weights derived simultaneously
  - ⌘ Accounts for correlations among neighbors
- ⌘ Outperforms KNN with similarity or equal weights

## Drawbacks

- ⌘ Must solve global optimization problem at training time
- ⌘ Must store  $O(M^2)$  weights in memory

# KNN: Summary

## Pros

- ⌘ Intuitive interpretation
- ⌘ When weights not learned...
  - ⌘ Easy to implement
  - ⌘ Zero training time
- ⌘ Learning prediction weights can greatly improve accuracy for little overhead in space and time

## Cons

- ⌘ When weights not learned...
  - ⌘ Need to store all item (or user) vectors in memory
  - ⌘ May redundantly recompute similarity scores at test time
  - ⌘ Similarity/equal weights not always suitable for prediction
- ⌘ When weights learned...
  - ⌘ Need to store  $O(M^2)$  or  $O(U^2)$  parameters
  - ⌘ Must update stored parameters when new ratings occur

# Low Dimensional Matrix Factorization

## Matrix Completion

- ⌘ Filling in the unknown ratings in a sparse  $U \times M$  matrix  $R$

$$\mathbf{R} = \begin{bmatrix} ? & ? & 1 & \dots & 4 \\ 3 & ? & ? & \dots & ? \\ ? & 5 & ? & \dots & 5 \end{bmatrix}$$

## Low dimensional matrix factorization

- ⌘ Model  $R$  as a product of two lower dimensional matrices



- ⌘  $A$  is  $U \times K$  “user factor” matrix,  $K \ll U, M$
- ⌘  $B$  is  $M \times K$ , “item factor” matrix
- ⌘ Learning  $A$  and  $B$  allows us to reconstruct all of  $R$



# Low Dimensional Matrix Factorization



**Interpretation:** Rows of  $A$  and  $B$  are low dimensional feature vectors  $a_u$  and  $b_i$  for each user  $u$  and item  $i$

**Motivation:** Dimensionality reduction

- ⌘ Compact representation: only need to learn and store  $UK + MK$  parameters
- ⌘ Matrices can often be adequately represented by low rank factorizations

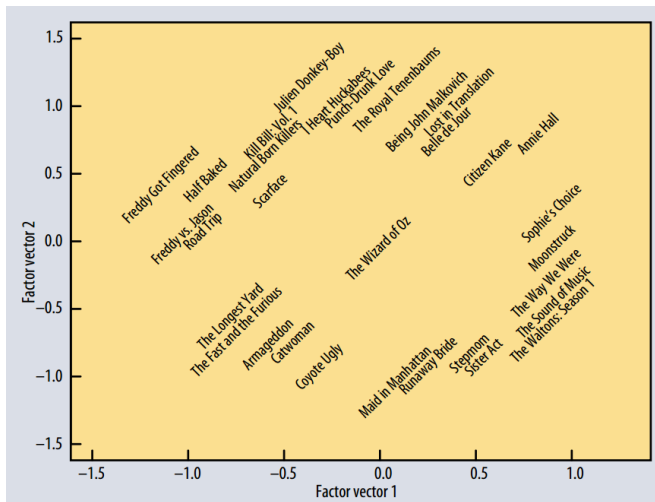
# Low Dimensional Matrix Factorization



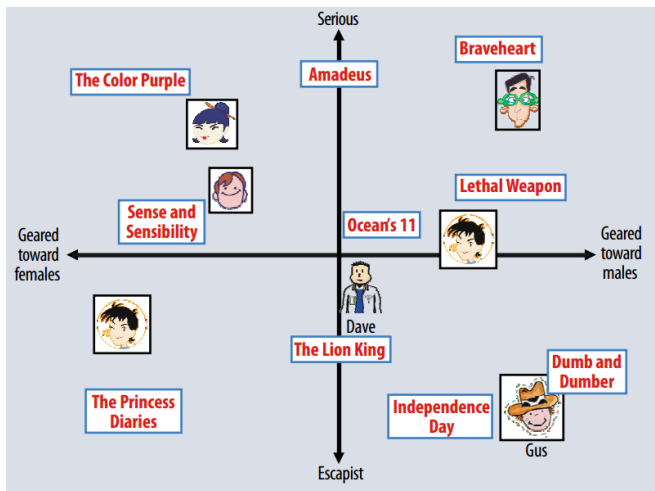
Very general framework that encapsulates many ML methods

- ⌘ Singular value decomposition
- ⌘ Clustering
  - ⌘  $A$  can represent cluster centers
  - ⌘  $B$  probabilities of belonging to each cluster
- ⌘ Factor Analysis/Probabilistic PCA

# Projecting movies



# Projecting users and movies



# Singular Value Decomposition

## Squared error objective for MF

$$\operatorname{argmin}_{A,B} \|R - AB^T\|_2^2 = \operatorname{argmin}_{A,B} \sum_{u=1}^U \sum_{i=1}^M (r_{ui} - \langle a_u, b_i \rangle)^2$$

⌘ Reasonable objective since RMSE is our error metric

**When all of  $R$  is observed**, this problem is solved by singular value decomposition (SVD)

⌘ **SVD:**  $R = H\Sigma V^T$

⌘  $H$  is  $U \times U$  with  $H^T H = I_{U \times U}$

⌘  $V$  is  $M \times M$  with  $V^T V = I_{M \times M}$

⌘  $\Sigma$  is  $U \times M$  and diagonal

⌘ **Solution:** Take first  $K$  pairs of singular vectors

⌘ Let  $A = H_{U \times K} \Sigma_{K \times K}$  and  $B = V_{M \times K}$

# SVD with Missing Values

## Weighted SE objective

$$\operatorname{argmin}_{A,B} \sum_{u=1}^U \sum_{i=1}^M W_{ui} (r_{ui} - \langle a_u, b_i \rangle)^2$$

## Binary weights

- ⊞  $W_{ui} = 1$  if  $r_{ui}$  observed,  $W_{ui} = 0$  otherwise
- ⊞ Only penalize errors on known ratings

## How to optimize?

- ⊞ Straightforward singular value decomposition no longer applies
- ⊞ Local minima exist  $\Rightarrow$  algorithm initialization is important

# SVD with Missing Values

## Insight: Chicken and egg problem

- ⌘ If we knew the missing values in  $R$ , could apply SVD
- ⌘ If we could apply SVD, we could find the missing values in  $R$
- ⌘ Idea: Fill in unknown entries with best guess; apply SVD; repeat

## Expectation-Maximization (EM) algorithm

- ⌘ Alternate until convergence:
  - 1 E step:  $X = W * R + (1 - W) * \hat{R}$   
(\* represents entrywise product)
  - 2 M step:  $[H, \Sigma, V] = SVD(X), \hat{R} = H_{U \times K} \Sigma_{K \times K} V_{M \times K}^T$

## Complexity: $O(UM)$ space and $O(UMK)$ time per EM iteration

- ⌘ What if  $UM$  or  $UMK$  is very large?
  - ⌘  $UM = 8.5$  billion for Netflix Prize dataset
- ⌘ Complete ratings matrix may not even fit into memory!

# SVD with Missing Values

## Regularized weighted SE objective

$$\operatorname{argmin}_{A,B} \sum_{u=1}^U \sum_{i=1}^M W_{ui} (r_{ui} - \langle a_u, b_i \rangle)^2 + \lambda \left( \sum_{u=1}^U \|a_u\|^2 + \sum_{i=1}^M \|b_i\|^2 \right)$$

## Equivalent form

$$\operatorname{argmin}_{A,B} \sum_{(u,i) \in \mathcal{T}} (r_{ui} - \langle a_u, b_i \rangle)^2 + \lambda \left( \sum_{u=1}^U \|a_u\|^2 + \sum_{i=1}^M \|b_i\|^2 \right)$$

## Motivation

- ⊖ Counters *overfitting* by implicitly restricting optimization space
  - ⊖ Shrinks entries of  $A$  and  $B$  toward 0
- ⊖ Can improve *generalization error*, performance on unseen test data



# Low Dimensional MF: Summary

## Pros

- ⌘ Data reduction: only need to store  $UK + MK$  parameters at test time
  - ⌘  $MK + M^2$  needed for Factor Analysis
- ⌘ Gradient descent and ALS procedures are easy to implement and scale well to large datasets
- ⌘ Empirically yields high accuracy in CF tasks
- ⌘ Matrix factors could be used as inputs into other learning algorithms (e.g. classifiers)

## Cons

- ⌘ Missing data MF objectives plagued by many local minima
- ⌘ Initialization is important
- ⌘ EM approaches tend to be slow for large datasets

# References

- ✧ K. Ali and W. van Stam, "TiVo: Making Show Recommendations Using a Distributed Collaborative Filtering Architecture," Proc. 10th ACM SIGKDD Int. Conference on Knowledge Discovery and Data Mining, pp. 394-401, 2004.
- ✧ J. Basilico, T. Hofmann. 2004. Unifying collaborative and content-based filtering. In Proceedings of the ICML.
- ✧ R. Bell and Y. Koren, "Scalable Collaborative Filtering with Jointly Derived Neighborhood Interpolation Weights," IEEE International Conference on Data Mining (ICDM07), pp. 43-52, 2007.
- ✧ J. Bennet and S. Lanning, "The Netflix Prize," KDD Cup and Workshop, 2007. [www.netflixprize.com](http://www.netflixprize.com).
- ✧ L. Breiman, (1996). Stacked Regressions. Machine Learning, Vol. 24, pp. 49-64.
- ✧ J. Canny, "Collaborative Filtering with Privacy via Factor Analysis," Proc. 25th ACM SIGIR Conf.on Research and Development in Information Retrieval (SIGIR02), pp. 238-245, 2002.
- ✧ A. Das, M. Datar, A. Garg and S. Rajaram, "Google News Personalization: Scalable Online Collaborative Filtering," WWW07, pp. 535-544, 2007.

# References

- ⌘ S. Funk, “Netflix Update: Try This At Home,” <http://sifter.org/simon/journal/20061211.html>, 2006.
- ⌘ J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, “An Algorithmic Framework for Performing Collaborative Filtering,” in Proceedings of the Conference on Research and Development in Information Retrieval, 1999.
- ⌘ Y. Koren. Collaborative filtering with temporal dynamics KDD, pp. 447-456, ACM, 2009.
- ⌘ Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. Proc. 14th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD08), pp. 426-434, 2008.
- ⌘ N. Lawrence and R. Urtasun. Non-linear matrix factorization with Gaussian processes. ICML, ACM International Conference Proceeding Series, Vol. 382, p. 76, ACM, 2009.
- ⌘ G. Linden, B. Smith and J. York, “Amazon.com Recommendations: Item-to-item Collaborative Filtering,” IEEE Internet Computing 7 (2003), 76-80.

# References

- ⌘ B. Marlin, R. Zemel, S. Roweis, and M. Slaney, “Collaborative filtering and the Missing at Random Assumption,” Proc. 23rd Conference on Uncertainty in Artificial Intelligence, 2007.
- ⌘ A. Paterek, “Improving Regularized Singular Value Decomposition for Collaborative Filtering,” Proc. KDD Cup and Workshop, 2007.
- ⌘ M. Piotte and M. Chabbert, “Extending the toolbox,” Netflix Grand Prize technical presentation, <http://pragmatictheory.blogspot.com/>, 2009.
- ⌘ R. Salakhutdinov, A. Mnih and G. Hinton. Restricted Boltzmann Machines for collaborative filtering. Proc. 24th Annual International Conference on Machine Learning, pp. 791-798, 2007.
- ⌘ N. Srebro and T. Jaakkola. Weighted low-rank approximations. In 20th International Conference on Machine Learning, pages 720-727. AAAI Press, 2003.
- ⌘ Gabor Takacs, Istvan Pitaszy, Bottyan Nemeth, and Domonkos Tikk. Scalable collaborative filtering approaches for large recommender systems. Journal of Machine Learning Research, 10:623-656, 2009.
- ⌘ C. Thompson. If you liked this, you're sure to love that. The New York Times, Nov 21, 2008.

# References

- ⌘ J. Wu and T. Li. A Modified Fuzzy C-Means Algorithm For Collaborative Filtering. Proc. Netflix-KDD Workshop, 2008.
- ⌘ K. Yu, J. Lafferty, S. Zhu, and Y. Gong. Large-scale collaborative prediction using a nonparametric random effects model. In The 25th International Conference on Machine Learning (ICML), 2009.
- ⌘ Y. Zhou, D. Wilkinson, R. Schreiber, R. Pan. "Large-Scale Parallel Collaborative Filtering for the Netix Prize," AAIM 2008: 337-348.
- ⌘ Y. Koren, R. Bell, C. Volinsky "Matrix Factorization techniques for recommender systems" IEEE/ACM Computer Journal 30-37, 2009