Scalable Machine Learning

8. Recommender Systems

Alex Smola
Yahoo! Research and ANU

http://alex.smola.org/teaching/berkeley2012
Stat 260 SP 12

Significant content courtesy of Yehuda Koren
8. Recommender Systems

Thousands of movies and TV episodes including these:

New Arrivals in TV
- BEINGHUMAN
- TARA
- WHITE COLLAR
- AFV
- Shameless
- The WALKING DEAD
- SUPERNATURAL

TV Drama
- CRIMINAL MINDS
- FLASHPOINT
- WORKAHOLICS
- SAVING GRACE
- MAD MEN
- DAMAGES
- WEEDS

Much content courtesy of (Mr Netflix) Yehuda Koren
• Neighborhood methods
  • User / movie similarity
  • Iteration on graph
• Matrix Factorization
  • Singular value decomposition
  • Convex reformulation
• Ranking and Session Modeling
  • Ordinal regression
  • Session models
• Features
  • Latent dense (Bayesian Probabilistic Matrix Factorization)
  • Latent sparse (Dirichlet process factorization)
  • Coldstart problem (inferring features)
• Hashing
Why
Thousands of movies and TV episodes including these:

New Arrivals in TV

TV Drama

TV Comedy

Children & Family
Thousands of movies and TV episodes including these:

New Arrivals in TV
- BEINGHUMAN
- TARA
- WHITECOLLAR
- AFV
- Shameless
- WALKING DEAD
- SUPERNATURAL

TV Drama
- CRIMINAL MINDS
- FLASHPOINT
- WORKAHOLICS
- SAVING GRACE
- MAD MEN
- DAMAGES
- WEEDS

TV Comedy
- RAISING HOPE
- HOT CLEVELAND
- TV LAND
- myboys
- PSYCH
- ARCHER
- MONK

Children & Family
- JANE LYNCH
- PIGSOEWS
- 6teen
- THE INCREDIBLE HULK
- TURTLE: THE INCREDIBLE MONKEY
- CURIOUS GEORGE
- LEAP FROG
- THE LITTLE ENGINE THAT COULD
Three Kings

George Clooney, Mark Wahlberg, Ice Cube conspire to steal a huge cache of gold hidden near their desert base.

Starring: George Clooney, Mark Wahlberg
Directed by: David O. Russell
Runtime: 1 hour 56 minutes
Release year: 1999
Studio: Warner Bros.

Also available in HD with Amazon Instant Video on Your TV

Your Amazon Prime membership now includes unlimited, commercial-free, instant streaming of thousands of movies and TV shows at no additional cost.

Customers Who Bought This Item Also Bought:

- **Tower Heist** Amazon Instant Video ~ Eddie Murphy
  - **42**
  - $3.99

- **Syriana** Amazon Instant Video ~ George Clooney
  - **355**
  - $2.99

- **Five Minutes of Heaven** Amazon Instant Video ~ Liam Neeson
  - **70**
  - $2.99

- **Foolproof** Amazon Instant Video ~ Ryan Reynolds
  - **81**
  - $2.99

- **The Recruit** Amazon Instant Video ~ Al Pacino
  - **161**
  - $1.99

Have a promotion code? Redeem a gift card or promotion code

View Balance

[Send us Feedback]
Recommended for You

These recommendations are based on items you own and more.

view: All | New Releases | Coming Soon

1. **Convex Optimization**
   by Stephen Boyd (March 8, 2004)
   Average Customer Review: ★★★★★
   In Stock
   - List Price: $84.00
   - Price: $68.13
   44 used & new from $61.32
   ![Add to Cart](#) | ![Add to Wish List](#)

   Recommended because you purchased Nonlinear Programming and more (Fix this)

   by Nir Friedman (July 31, 2009)
   Average Customer Review: ★★★★★
   In Stock
   - List Price: $96.00
   - Price: $93.55
   47 used & new from $91.33
   ![Add to Cart](#) | ![Add to Wish List](#)

   Recommended because you purchased Nonlinear Programming and more (Fix this)

3. **Doing Bayesian Data Analysis: A Tutorial with R and BUGS**
   by John K. Kruschke (November 10, 2010)
   Average Customer Review: ★★★★★
   In Stock
   - List Price: $89.96
   - Price: $77.98
   44 used & new from $68.40
   ![Add to Cart](#) | ![Add to Wish List](#)

   Recommended because you purchased Bayesian Nonparametrics and more (Fix this)

4. **Parallel and Distributed Computation: Numerical Methods (Optimization and Neural Computation)**
   by Dimitri P. Bertsekas (January 1, 1997)
   Average Customer Review: ★★★★★
   In Stock
   - Price: $49.50
   15 used & new from $45.49
   ![Add to Cart](#) | ![Add to Wish List](#)

   Recommended because you purchased Nonlinear Programming (Fix this)
Personalized Content

adapt to general popularity
pick based on user preferences
### Spam Filtering

Something went wrong!
A more formal view

- User (requests content)
- Objects (that can be displayed)
- Context (device, location, time)
- Interface (mobile browser, tablet, viewport)

interface

recommend relevant objects
Examples

• Movie recommendation (Netflix)
• Related product recommendation (Amazon)
• Web page ranking (Google)
• Social recommendation (Facebook)
• News content recommendation (Yahoo)
• Priority inbox & spam filtering (Google)
• Online dating (OK Cupid)
• Computational Advertising (Yahoo)
### Running Example

**Netflix Movie Recommendation**

#### Training data

<table>
<thead>
<tr>
<th>user</th>
<th>movie</th>
<th>date</th>
<th>score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>21</td>
<td>5/7/02</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>213</td>
<td>8/2/04</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>345</td>
<td>3/6/01</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>123</td>
<td>5/1/05</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>768</td>
<td>7/15/02</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>76</td>
<td>1/22/01</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>45</td>
<td>8/3/00</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>568</td>
<td>9/10/05</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>342</td>
<td>3/5/03</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>234</td>
<td>12/28/00</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>76</td>
<td>8/11/02</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>56</td>
<td>6/15/03</td>
<td>4</td>
</tr>
</tbody>
</table>

#### Test data

<table>
<thead>
<tr>
<th>user</th>
<th>movie</th>
<th>date</th>
<th>score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>62</td>
<td>1/6/05</td>
<td>?</td>
</tr>
<tr>
<td>1</td>
<td>96</td>
<td>9/13/04</td>
<td>?</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>8/18/05</td>
<td>?</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>11/22/05</td>
<td>?</td>
</tr>
<tr>
<td>3</td>
<td>47</td>
<td>6/13/02</td>
<td>?</td>
</tr>
<tr>
<td>3</td>
<td>15</td>
<td>8/12/01</td>
<td>?</td>
</tr>
<tr>
<td>4</td>
<td>41</td>
<td>9/1/00</td>
<td>?</td>
</tr>
<tr>
<td>4</td>
<td>28</td>
<td>8/27/05</td>
<td>?</td>
</tr>
<tr>
<td>5</td>
<td>93</td>
<td>4/4/05</td>
<td>?</td>
</tr>
<tr>
<td>5</td>
<td>74</td>
<td>7/16/03</td>
<td>?</td>
</tr>
<tr>
<td>6</td>
<td>69</td>
<td>2/14/04</td>
<td>?</td>
</tr>
<tr>
<td>6</td>
<td>83</td>
<td>10/3/03</td>
<td>?</td>
</tr>
</tbody>
</table>

---

**Note:** The score column for test data entries is left blank, indicating these are the records to be predicted.
Challenges

- Scalability
  - Millions of objects
  - 100s of millions of users
- Cold start
  - Changing user base
  - Changing inventory (movies, stories, goods)
- Attributes
- Imbalanced dataset
User activity / item reviews are power law distributed

Netflix competition yardstick

- Least mean squares prediction error
- Easy to define
  \[ \text{rmse}(S) = \sqrt{|S|^{-1}} \sum_{(i,u) \in S} (\hat{r}_{ui} - r_{ui})^2 \]
- Wrong measure for composing sessions!
- Consistent (in large sample size limit this will converge to minimizer)
1 Neighborhood Methods
Basic Idea

Part I: Basic neighborhood methods

Joe
Basic Idea

- (user,user) similarity to recommend items
  - good if item base is smaller than user base
  - good if item base changes rapidly
  - traverse bipartite similarity graph
- (item,item) similarity to recommend new items that were also liked by the same users
  - good if the user base is small is small
- Oldest known CF method
Neighborhood based CF

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>3</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>
</tbody>
</table>

- unknown rating
- rating between 1 to 5
Neighborhood based CF

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>?</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>3</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>
</tbody>
</table>

- unknown rating
- rating between 1 to 5
### Neighborhood based CF

#### Rating Matrix

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>?</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>3</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>
</tbody>
</table>

- **- unknown rating**
- **- rating between 1 to 5**

### Notes
- The matrix represents user-item ratings in a neighborhood-based collaborative filtering context.
- Users are listed along the top, and items are listed along the left side.
- The unknown ratings are indicated by a question mark (?).
Neighborhood based CF

<table>
<thead>
<tr>
<th>users</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>?</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td></td>
<td>2</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>3</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>
</tbody>
</table>

- unknown rating
- rating between 1 to 5

similarity

\[ s_{13} = 0.2 \]
\[ s_{16} = 0.3 \]
Neighborhood based CF

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
<td>2.6</td>
<td>5</td>
<td></td>
<td>5</td>
<td></td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>5</td>
<td>4</td>
<td></td>
<td>4</td>
<td></td>
<td>2</td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td></td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td></td>
<td>4</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td></td>
<td>2</td>
<td></td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- unknown rating
- rating between 1 to 5

similarity
\[ s_{13} = 0.2 \]
\[ s_{16} = 0.3 \]

weighted average
\[
\frac{0.2 \cdot 2 + 0.3 \cdot 3}{0.2 + 0.3} = 2.6
\]
• Intuitive
• No (substantial) training
• Handles new users / items
• Easy to explain to user

Properties

• Accuracy & scalability questionable
Normalization / Bias

• Problem
  • Some items are significantly higher rated
  • Some users rate substantially lower
  • Ratings change over time
• Bias correction is crucial for nearest neighborhood recommender algorithm
  • Offset per user
  • Offset per movie
  • Time effects
  • Global bias

\[ b_{ui} = \mu + b_u + b_i \]
Baseline estimation

- Mean rating is 3.7
- Troll Hunter is 0.7 above mean
- User rates 0.2 below mean
- Baseline is 4.2 stars
- Least mean squares problem

\[
\text{minimize } \sum_{(u,i)} (r_{ui} - \mu - b_u - b_i)^2 + \lambda \left[ \sum_u b_u^2 + \sum_i b_i^2 \right]
\]

- Jointly convex. Alternatively remove mean & iterate

\[
b_i = \frac{\sum_{u \in R(i)} (r_{ui} - \mu - b_u)}{\lambda + |R(i)|} \quad \text{and} \quad b_u = \frac{\sum_{i \in R(u)} (r_{ui} - \mu - b_i)}{\lambda + |R(u)|}
\]
Parzen Windows style CF

• Similarity measure $s_{ij}$ between items
• Find set $s_k(i,u)$ of $k$-nearest neighbors to $i$ that were rated by user $u$
• Weighted average over the set

$$\hat{r}_{ui} = b_{ui} + \frac{\sum_{j \in s_k(i,u)} s_{ij} (r_{uj} - b_{uj})}{\sum_{j \in s_k(i,u)} s_{ij}} \text{ where } b_{ui} = \mu + b_{u} + b_{i}$$

• How to compute $s_{ij}$?
(item,item) similarity measures

- **Pearson correlation coefficient**
  - **nonuniform support**
  - compute only over shared support
  - shrinkage towards 0 to address problem of small support (typically few items in common)
Empirical Pearson correlation coefficient

\[ \hat{\rho}_{ij} = \frac{\sum_{u \in U(i, j)} (r_{ui} - b_{ui})(r_{uj} - b_{uj})}{\sqrt{\sum_{u \in U(i, j)} (r_{ui} - b_{ui})^2 \sum_{u \in U(i, j)} (r_{uj} - b_{uj})^2}} \]

Smoothing towards 0 for small support

\[ s_{ij} = \frac{|U(i, j)| - 1}{|U(i, j)| - 1 + \lambda \hat{\rho}_{ij}} \]

Make neighborhood more peaked \( s_{ij} \to s^2_{ij} \)

Shrink towards baseline for small neighborhood

\[ \hat{r}_{ui} = b_{ui} + \frac{\sum_{j \in s_k(i, u)} s_{ij}(r_{uj} - b_{uj})}{\lambda + \sum_{j \in s_k(i, u)} s_{ij}} \]
Similarity for binary data

- Pearson correlation meaningless
- Views
- Purchase behavior
- Clicks
- Jaccard similarity (intersection vs. joint)

\[ s_{ij} = \frac{m_{ij}}{\alpha + m_i + m_j - m_{ij}} \]

- Observed/expected ratio
  Improve by counting per user (many users better than heavy users)

\[ s_{ij} = \frac{\text{observed}}{\text{expected}} \approx \frac{m_{ij}}{\alpha + m_i m_j / m} \]
2 Matrix Factorization
Basics
Basic Idea

Matrix factorization techniques

\[ M \approx U \cdot V \]
Latent variable view

The Color Purple

Sense and Sensibility

The Princess Diaries

Amadeus

Ocean’s 11

Lethal Weapon

Braveheart

Dumb and Dumber

The Lion King

Independence Day

Geared towards females

escapist

serious

Geared towards males

Latent factor models
Basic matrix factorization model

A rank-3 SVD approximation
Estimate unknown ratings as inner products of latent factors.

A rank-3 SVD approximation
Estimate unknown ratings as inner products of latent factors.

A rank-3 SVD approximation
Estimate unknown ratings as inner products of latent factors.

A rank-3 SVD approximation.

<table>
<thead>
<tr>
<th>users</th>
<th>1.1</th>
<th>-.2</th>
<th>.3</th>
<th>.5</th>
<th>-.2</th>
<th>-.5</th>
<th>.8</th>
<th>-.4</th>
<th>.3</th>
<th>1.4</th>
<th>2.4</th>
<th>-.9</th>
</tr>
</thead>
<tbody>
<tr>
<td>-.8</td>
<td>.7</td>
<td>.5</td>
<td>1.4</td>
<td>3</td>
<td>-1</td>
<td>1.4</td>
<td>2.9</td>
<td>-.7</td>
<td>1.2</td>
<td>-.1</td>
<td>1.3</td>
<td></td>
</tr>
<tr>
<td>2.1</td>
<td>-.4</td>
<td>.6</td>
<td>1.7</td>
<td>2.4</td>
<td>.9</td>
<td>-.3</td>
<td>.4</td>
<td>.8</td>
<td>.7</td>
<td>-.6</td>
<td>.1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>items</th>
<th>.1</th>
<th>-.4</th>
<th>.2</th>
<th></th>
<th>-.5</th>
<th>.6</th>
<th>.5</th>
<th></th>
<th>-2</th>
<th>.3</th>
<th>.5</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>-.2</td>
<td>.3</td>
<td>.5</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>1.1</td>
<td>2.1</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>
<tr>
<td>-.7</td>
<td>2.1</td>
<td>-2</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>-1</td>
<td>.7</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>
Properties

- SVD is undefined for missing entries
- Stochastic gradient descent (faster)
- Alternating optimization
- Overfitting without regularization particularly if fewer reviews than dimensions
- Very popular on Netflix

\[ \begin{pmatrix}
1 & 3 & 5 & 5 & 4 \\
5 & 4 & 4 & 2 & 1 & 3 \\
2 & 4 & 1 & 2 & 3 & 4 & 5 \\
2 & 4 & 5 & 4 & 2 & 5 \\
1 & 3 & 3 & 2 & 4 \\
\end{pmatrix} \sim \begin{pmatrix}
.1 & -.4 & .2 \\
-.5 & .6 & .5 \\
-.2 & .3 & .5 \\
1.1 & 2.1 & .3 \\
-.7 & 2.1 & -2 \\
-1 & .7 & .3 \\
\end{pmatrix} \begin{pmatrix}
1.1 & -2 & .3 & .5 & -2 & -5 & .8 & -.4 & .3 & 1.4 & 2.4 & -.9 \\
-.8 & .7 & .5 & 1.4 & .3 & -1 & 1.4 & 2.9 & -.7 & 1.2 & -.1 & 1.3 \\
2.1 & -.4 & .6 & 1.7 & 2.4 & .9 & -.3 & .4 & .8 & .7 & -.6 & .1 \\
\end{pmatrix} \]
Risk Minimization View

- **Objective Function**

\[
\min_{p,q} \sum_{(u,i) \in S} (r_{ui} - \langle p_u, q_i \rangle)^2 + \lambda \left[ \|p\|_{\text{Frob}}^2 + \|q\|_{\text{Frob}}^2 \right]
\]

- **Alternating least squares**

\[
p_u \leftarrow \left[ \lambda \mathbf{1} + \sum_{i|(u,i) \in S} q_i q_i^\top \right]^{-1} \sum_i q_i r_{ui}
\]

\[
q_i \leftarrow \left[ \lambda \mathbf{1} + \sum_{u|(u,i) \in S} p_u p_u^\top \right]^{-1} \sum_i p_u r_{ui}
\]

good for MapReduce
Risk Minimization View

- **Objective Function**

\[
\text{minimize}_{p,q} \sum_{(u, i) \in S} (r_{ui} - \langle p_u, q_i \rangle)^2 + \lambda \left[ \|p\|_{\text{Frob}}^2 + \|q\|_{\text{Frob}}^2 \right]
\]

- **Stochastic gradient descent**

\[
p_u \leftarrow (1 - \lambda \eta_t)p_u - \eta_t q_i (r_{ui} - \langle p_u, q_i \rangle)
\]

\[
q_i \leftarrow (1 - \lambda \eta_t)q_i - \eta_t p_u (r_{ui} - \langle p_u, q_i \rangle)
\]

- **No need for locking**

- **Multicore updates asynchronously**

  (Recht, Re, Wright, 2012 - Hogwild)
Theoretical Motivation
deFinetti Theorem

- Independent random variables
  \[ p(X) = \prod_{i=1}^{m} p(x_i) \]

- Exchangeable random variables
  \[ p(X) = p(x_1, \ldots, x_m) = p(x_{\pi(1)}, \ldots, x_{\pi(m)}) \]

- There exists a conditionally independent representation of exchangeable r.v.
  \[ p(X) = \int dp(\theta) \prod_{i=1}^{m} p(x_i | \theta) \]

This motivates latent variable models
Aldous Hoover Factorization

- Matrix-valued set of random variable
  
  Example - Erdos Renyi graph model

  \[ p(E) = \prod_{i,j} p(V_{ij}) \]

- Independently exchangeable on matrix

  \[ p(E) = p(E_{11}, E_{12}, \ldots, E_{mn}) = p(E_{\pi(1)\rho(1)}, E_{\pi(1)\rho(2)}, \ldots, E_{\pi(m)\rho(n)}) \]

- Aldous Hoover Theorem

  \[ p(E) = \int dp(\theta) \int \prod_{i=1}^{m} dp(u_i) \prod_{j=1}^{n} dp(v_j) \prod_{i,j} p(E_{ij} | u_i, v_j, \theta) \]
### Aldous Hoover Factorization

- Rating matrix is (row, column) exchangeable.
- Draw latent variables per row and column.
- Draw matrix entries independently given pairs.
- Absence / presence of rating is a signal.
- Can be extended to graphs with vertex attributes.

<table>
<thead>
<tr>
<th></th>
<th>u₁</th>
<th>u₂</th>
<th>u₃</th>
<th>u₄</th>
<th>u₅</th>
<th>u₆</th>
</tr>
</thead>
<tbody>
<tr>
<td>v₁</td>
<td></td>
<td>e₁₁</td>
<td>e₁₂</td>
<td></td>
<td>e₁₅</td>
<td>e₁₆</td>
</tr>
<tr>
<td>v₂</td>
<td></td>
<td></td>
<td></td>
<td>e₂₄</td>
<td></td>
<td></td>
</tr>
<tr>
<td>v₃</td>
<td>e₃₂</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>v₄</td>
<td></td>
<td>e₄₃</td>
<td></td>
<td>e₄₆</td>
<td></td>
<td></td>
</tr>
<tr>
<td>v₅</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>e₅₅</td>
</tr>
</tbody>
</table>
Aldous Hoover variants

• Jointly exchangeable matrix
  • Social network graphs
  • Draw vertex attributes first, then edges

• Cold start problem
  • New user appears
  • Attributes (age, location, browser)
  • Can estimate latent variables from that

• User and item factors in matrix factorization problem can be viewed as AH-factors
Improvements
Factor models: Error vs. #parameters

- NMF
- BiasSVD
- SVD++
- SVD v.2
- SVD v.3
- SVD v.4

Add biases
Bias

**Objective Function**

\[
\min_{p,q} \sum_{(u,i) \in S} (r_{ui} - (\mu + b_u + b_i + \langle p_u, q_i \rangle))^2 +
\lambda \left[ \|p\|_{\text{Frob}}^2 + \|q\|_{\text{Frob}}^2 + \|b_{\text{users}}\|^2 + \|b_{\text{items}}\|^2 \right]
\]

**Stochastic gradient descent**

\[
\begin{align*}
p_u &\leftarrow (1 - \lambda \eta_t)p_u - \eta_t q_i \rho_{ui} \\
q_i &\leftarrow (1 - \lambda \eta_t)q_i - \eta_t p_u \rho_{ui} \\
b_u &\leftarrow (1 - \lambda \eta_t)b_u - \eta_t \rho_{ui} \\
b_i &\leftarrow (1 - \lambda \eta_t)b_i - \eta_t \rho_{ui} \\
\mu &\leftarrow (1 - \lambda \eta_t)\mu - \eta_t \rho_{ui}
\end{align*}
\]

where \( \rho_{ui} = (r_{ui} - (\mu + b_i + b_u + \langle p_u, q_i \rangle)) \)
Factor models: Error vs. #parameters

RMSE vs. Millions of Parameters

- NMF
- BiasSVD
- SVD++
- SVD v.2
- SVD v.3
- SVD v.4

“who rated what”
Ratings are not given at random

• Characterize users by which movies they rated
Edge attributes (observed, rating)

• Adding features to recommender system

\[ r_{ui} = \mu + b_u + b_i + \langle p_u, q_i \rangle + \langle c_u, x_i \rangle \]
Alternative integration

- Key idea - use related ratings to average
- Salakhudtinov & Mnih, 2007
  \[ q_i \leftarrow q_i + \sum_u c_{ui} p_u \]
- Koren et al., 2008
  \[ q_i \leftarrow q_i + \sum_u c_{ui} x_j \]

Overparametrize items by q and x
Something Happened in Early 2004…

Netflix ratings by date

Netflix changed rating labels
Are movies getting better with time?
Sources of temporal change

- Items
  - Seasonal effects
    (Christmas, Valentine’s day, Holiday movies)
  - Public perception of movies (Oscar etc.)
- Users
  - Changed labeling of reviews
  - Anchoring (relative to previous movie)
  - Change of rater in household
  - Selection bias for time of viewing
Modeling temporal change

- Time-dependent bias
- Time-dependent user preferences
  \[ r_{ui}(t) = \mu + b_u(t) + b_i(t) + \langle q_i, p_u(t) \rangle \]
- Parameterize functions \( b \) and \( p \)
  - Slow changes for items
  - Fast sudden changes for users
- Good parametrization is key

Koren et al., KDD 2009 (CF with temporal dynamics)
Bias matters!

Sources of Variance in Netflix data

- Unexplained: 57%
- Biases: 33%
- Personalization: 10%

$0.732$ (unexplained) + $0.415$ (biases) + $0.129$ (personalization) = $1.276$ (total variance)
Factor models: Error vs. #parameters

\[ r_{ui} = q_i^T p_u \]

\[ r_{ui}(t) = \mu + b_u(t) + b_i(t) + q_i^T \left( p_u(t) + \sum_j b_{uj} x_j \right) \]

Netflix: 0.9514

Prize: 0.8563

RMSE

Millions of Parameters
More ideas

• Explain factorizations
• Cold start (new users)
• Different regularization for different parameter groups / different users
• Sharing of statistical strength between users
• Hierarchical matrix co-clustering / factorization (write a paper on that)
3 Session Modeling
Motivation
User interaction

- Explicit search query
  - Search engine
  - Genre selection on movie site
- Implicit search query
  - News site
  - Priority inbox
  - Comments on article
  - Viewing specific movie (see also ...)
  - Sponsored search (advertising)

Space, users’ time and attention are limited.
Sessions Modeling Studio
www.sessionsmodeling.com/
Sessions modeling studio is a licensed agency. We offer print modeling, promotions, fashion shows, movies, tv commercials, and more. Locates in Jacksonville ...
+ Show map of 12627 San Jose Blvd # 401, Jacksonville, FL 32223

Super Models
The Campbell Soup Company hired Sessions models for a ...

Model Search
Contact Us. Sessions Modeling Studio 12627 San Jose Blvd ...

Child Sessions
Contact Us. Sessions Modeling Studio. 12627 San Jose Blvd ...

Teen Sessions
Contact Us. Sessions Modeling Studio. 12627 San Jose Blvd ...

More results from sessionsmodeling.com »

Sessions Modeling Studio - Local Business - Jacksonville, FL ...
www.facebook.com/pages/Sessions-Modeling-Studio/99577445805
Sessions Modeling Studio - Sessions Modeling Studio is a licensed agency. ... To interact with Sessions Modeling Studio you need to sign up for Facebook first.
+ Show map of 12627 San Jose Blvd # 401, Jacksonville, FL 32223

Rethinking Modeling Sessions
www.agilemodeling.com/essays/modelingSessions.htm
Recently reviewed, A modeling session is an activity where one or more people focus on the development of one or more models. Modeling sessions are an ...
Did the user SCROLL DOWN?
Bad ideas ...

• Show items based on relevance

• Yes, this user likes Die Hard.
• But he likes other movies, too

• Show items only for majority of users
  ‘apple’ vs. ‘Apple’
User response

Top Stories

Feds to investigate death of Florida teen
USA TODAY - 59 minutes ago

ORLANDO, Florida (AP) - Following a day of protests calling for the arrest of a Florida neighborhood watch captain who fatally shot an unarmed black teen, the US Justice Department announced late Monday it will investigate the case.

Feds to investigate fatal shooting of Fla. teen Boston.com
Black teen's slaying spur calls for man's arrest San Francisco Chronicle

Your preferred source: Federal agencies to open investigation into black teen's death Washington Post
From Florida: US Department of Justice, FBI and FDLE to probe Trayvon Martin killing Miami Herald.com
Opinion: Trayvon Martin and a vigilante's deadly zeal Pittsburgh Post Gazette
Wikipedia: Trayvon Martin

See all 1,241 sources »

euronews KSL-TV ABC News PolicyMic Washington... The Guard... Daily Mail CBS News

log it!

collapse

implicit user interest
Response is conditioned on available options

- User search for ‘chocolate’
  - What the user really would have wanted
    - User can only pick from available items
  - Preferences are often relative
Models
Independent click model

- Each object has click probability
- Object is viewed independently
  - Used in computational advertising (with some position correction)
  - Horribly wrong assumption
  - OK if probability is very small (OK in ads)

\[
p(x | s) = \prod_{i=1}^{n} \frac{1}{1 + e^{-x_i s_i}}
\]
Logistic click model

- User picks at most one object
- Exponential family model for click

\[ p(x|s) = \frac{e^{sx}}{e^{s_0} + \sum_{x'} e^{s_{x'}}} = \exp(s_x - g(s)) \]

- Ignores order of objects
- Assumes that the user looks at all before taking action
Sequential click model

- User traverses list
- At each position some probability of clicking
- When user reaches end of the list he aborts

\[ p(x = j|s) = \left[ \prod_{i=1}^{j-1} \frac{1}{1 + e^{s_i}} \right] \frac{1}{1 + e^{-sj}} \]

- This assumes that a patient user viewed all items
• User traverses list
• At each position some probability of clicking
• At each position the user may abandon the process
• This assumes that user traverses list sequentially
Context skip click model

- User traverses list
- At each position some probability of clicking which depends on previous content
- At each position the user may abandon the process
- User may click more than once
Context skip click model

\[ p(v, c | d) = \prod_{i=1}^{n} \left[ p(v_i | v_{i-1}, c_{i-1}) p(c_i | v_i, c_{i-1}, d^i) \right] \]
Context skip click model

• Viewing probability

\[ p(v_i = 1 | v_{i-1} = 0) = 0 \]
\[ p(v_i = 1 | v_{i-1} = 1, c_{i-1} = 0) = \frac{1}{1 + e^{-\alpha_i}} \]
\[ p(v_i = 1 | v_{i-1} = 1, c_{i-1} = 1) = \frac{1}{1 + e^{-\beta_i}} \]

• Click probability (only if viewed)

\[ p(c_i = 1 | v_i = 1, c_{i-1}, d^i) = \frac{1}{1 + e^{-f(|c_{i-1}|, d_i, d_{i-1})}} \]

\[ p(v, c | d) = \prod_{i=1}^{n} \left[ p(v_i | v_{i-1}, c_{i-1})p(c_i | v_i, c_{i-1}, d^i) \right] \]
Incremental gains score

\[
f(|c^{i-1}|, d_i, d^{i-1})
\]

\[
:= \rho(S, d^i|a, b) - \rho(S, d^{i-1}|a, b) + \gamma_{|c^{i-1}|} + \delta_i
\]

\[
:= \sum_{s \in S} \sum_j [s]_j \left( a_j \sum_{d \in d^i} [d]_j + b_j \left( \rho_j(d^i) - \rho_j(d^{i-1}) \right) \right)
\]

\[
+ \gamma_{|c^{i-1}|} + \delta_i
\]

- Submodular gain per additional document
- Relevance score per document
- Coverage over different aspects
- Position dependent score
- Score dependent on number of previous clicks
Optimization

- Latent variables

\[ p(v, c|d) = \prod_{i=1}^{n} \left[ p(v_i|v_{i-1}, c_{i-1})p(c_i|v_i, c_{i-1}^{i-1}, d^i) \right] \]

We don’t know \( v \) whether user viewed result

- Use variational inference to integrate out \( v \) (more next week in graphical models)

\[ -\log p(c) \leq -\log p(c) + D(q(v)||p(v|c)) \]
\[ = E_{v \sim q(v)} [-\log p(c) + \log q(v) - \log p(v|c)] \]
\[ = E_{v \sim q(v)} [-\log p(c, v)] - H(q(v)). \]
Optimization

• Compute latent viewing probability given clicks
  • Easy since we only have one transition from views to no views (no DP needed)
• Expected log-likelihood under viewing model
• Convex expected log-likelihood
• Stochastic gradient descent
• Parametrization uses personalization, too (user, position, viewport, browser)
Figure 1: Prec@1 (left) and Prec@FC (right) of various user models relative to the deployed model. SVCM is the Sequential View Click Model, TDN is the model in [9], SUM is the Session Utility Model of [7], CCM is the Cascade Click Model of [6]. All models were trained using SGD (See Figure 3).

Figure 2: Prec@1 (left) and Prec@FC (right) of the combined (10) and submodular only coverage scores, relative to the deployed model.

Figure 3: Prec@1 (left) and Prec@FC (right) of stochastic gradient descent (SGD) vs exponentiated gradient (EG) relative to the deployed model.

Figure 4: Lift in Prec@FC achieved by SVCM over deployed system vs number of user visits (right) and vs the number of times a story was displayed (left).
Figure 1: Prec@1 (left) and Prec@FC (right) of various user models relative to the deployed model. SVCM is the Sequential View Click Model, TDN is the model in [9], SUM is the Session Utility Model of [7], CCM is the Cascade Click Model of [6]. All models were trained using SGD (See Figure 3).

Figure 2: Prec@1 (left) and Prec@FC (right) of the combined (10) and submodular only coverage scores, relative to the deployed model.

Figure 3: Prec@1 (left) and Prec@FC (right) of stochastic gradient descent (SGD) vs exponentiated gradient (EG) relative to the deployed model.

Figure 4: Lift in Prec@FC achieved by SVCM over deployed system vs number of user visits (right) and vs the number of times a story was displayed (left).
Figure 1: Prec@1 (left) and Prec@FC (right) of various user models relative to the deployed model. SVCM is the Sequential View Click Model, TDN is the model in [9], SUM is the Session Utility Model of [7], CCM is the Cascade Click Model of [6]. All models were trained using SGD (See Figure 3).

Figure 2: Prec@1 (left) and Prec@FC (right) of the combined (10) and submodular only coverage scores, relative to the deployed model.

Figure 3: Prec@1 (left) and Prec@FC (right) of stochastic gradient descent (SGD) vs exponentiated gradient (EG) relative to the deployed model.

Figure 4: Lift in Prec@FC achieved by SVCM over deployed system vs number of user visits (right) and vs the number of times a story was displayed (left).
Figure 1: Prec@1 (left) and Prec@FC (right) of various user models relative to the deployed model. SVCM is the Sequential View Click Model, TDN is the model in [9], SUM is the Session Utility Model of [7], CCM is the Cascade Click Model of [6]. All models were trained using SGD (See Figure 3).

Figure 2: Prec@1 (left) and Prec@FC (right) of the combined (10) and submodular only coverage scores, relative to the deployed model.

Figure 3: Prec@1 (left) and Prec@FC (right) of stochastic gradient descent (SGD) vs exponentiated gradient (EG) relative to the deployed model.

Figure 4: Lift in Prec@FC achieved by SVCM over deployed system vs number of user visits (right) and vs the number of times a story was displayed (left).
4 Feature Representation
Bayesian Probabilistic Matrix Factorization
Statistical Model

- Aldous-Hoover factorization
- normal distribution for user and item attributes
- rating given by inner product

Ratings

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

- Latent factors

\[ p(U|\sigma_U^2) = \prod_{i=1}^{N} \mathcal{N}(U_i|0, \sigma_U^2 I), \quad p(V|\sigma_V^2) = \prod_{j=1}^{M} \mathcal{N}(V_j|0, \sigma_V^2 I) \]

Salakhutdinov & Mnih, ICML 2008 BPMF
• Priors on all factors
• Wishart prior is conjugate to Gaussian, hence use it
• Allows us to adapt the variance automatically
• Inference (Gibbs sampler)
  • Sample user factors (parallel)
  • Sample movie factors (parallel)
  • Sample hyperparameters (parallel)
Making it fancier (constrained BPMF)

Let $W \in \mathbb{R}^{D \times M}$ be a latent similarity constraint matrix.

We define the feature vector for user $i$ as:

$$U_i = Y_i + \sum_{k=1}^{M} I_{ik} W_k \sum_{k=1}^{M} I_{ik}$$

$I$ is the observed indicator matrix, $I_{ij} = 1$ if user $i$ rated movie $j$ and 0 otherwise.

Performs considerably better on infrequent users.
Results (Mnih & Salakhutdinov)

• Left Panel: Performance of constrained PMF, PMF and movie average algorithm that always predicts the average rating of each movie.
• Right panel: Distribution of the number of ratings per user in the training dataset.

helps for infrequent users
Multiple Sources
Social Network Data

Data: users, connections, features
Goal: suggest connections
Social Network Data

Data: users, connections, features
Goal: suggest connections
Social Network Data

Data: users, connections, features
Goal: suggest connections
Social Network Data

Data: users, connections, features
Goal: model/suggest connections

\[ p(x, y, e) = \prod_{i \in \text{Users}} p(y_i)p(x_i|y_i) \prod_{i, j \in \text{Users}} p(e_{ij}|x_i, y_i, x_j, y_j) \]

Direct application of the Aldous-Hoover theorem. Edges are conditionally independent.
Applications
Applications
Applications

social network = friendship + interests
Applications

social network = friendship + interests

recommend users based on friendship & interests

recommend apps based on friendship & interests
Social Recommendation

recommend users based on friendship & interests

• boost traffic
• make the user graph more dense
• increase user population
• stickiness

recommend apps based on friendship & interests

• boost traffic
• increased revenue
• increased user participation
• make app graph more dense

... usually addressed by separate tools ...
Homophily

- Users with similar interests are more likely to connect
- Friends install similar applications

Highly correlated. Estimate both jointly
Model

(latent) app features

(latent) user features

app install
• Social interaction

\[ x_i \sim p(x|y_i) \]
\[ x_j \sim p(x|y_j) \]
\[ e_{ij} \sim p(e|x_i, y_i, x_j, y_j, \Phi) \]

• App install

\[ x_i \sim p(x|y_i) \]
\[ v_j \sim p(v|u_j) \]
\[ a_{ij} \sim p(a|x_i, y_i, u_j, v_j, \Phi) \]
Model

• Social interaction

\[ x_i \sim p(x \mid y_i) \]
\[ x_j \sim p(x \mid y_j) \]
\[ e_{ij} \sim p(e \mid x_i, y_i, x_j, y_j, \Phi) \]

\[ x_i = Ay_i + \epsilon_i \]
\[ v_j = Bu_j + \tilde{\epsilon}_j \]

• App install

\[ x_i \sim p(x \mid y_i) \]
\[ v_j \sim p(v \mid u_j) \]
\[ a_{ij} \sim p(a \mid x_i, y_i, u_j, v_j, \Phi) \]

\[ e_{ij} \sim p(e \mid x_i^\top x_j + y_i^\top W y_j) \]
\[ a_{ij} \sim p(a \mid x_i^\top v_j + y_i^\top M u_j) \]
minimize \[ \lambda_e \sum_{(i,j)} l(e_{ij}, x_i^T x_j + y_i^T W y_j) + \]
Optimization Problem

minimize \( \lambda_e \sum_{(i,j)} l(e_{ij}, x_i^\top x_j + y_i^\top W y_j) \)
minimize \( \lambda_e \sum_{(i,j)} l(e_{ij}, x_i^T x_j + y_i^T Wy_j) + \) social

\( \lambda_a \sum_{(i,j)} l(a_{ij}, x_i^T y_j + y_i^T Mu_j) + \) app
minimize \[ \lambda_e \sum_{i,j} l(e_{ij}, x_i^T x_j + y_i^T Wy_j) + \]

\[ \lambda_a \sum_{i,j} l(a_{ij}, x_i^T v_j + y_i^T Mu_j) + \]

\[ \lambda_x \sum \gamma(x_i|y_i) + \lambda_v \sum \gamma(v_i|u_i) + \]

social

app

reconstruction
Optimization Problem

\[
\text{minimize} \quad \lambda_e \sum_{(i,j)} l(e_{ij}, x_i^T x_j + y_i^T W y_j) + \\
\lambda_a \sum_{(i,j)} l(a_{ij}, x_i^T v_j + y_i^T M u_j) + \\
\lambda_x \sum_i \gamma(x_i | y_i) + \lambda_v \sum_i \gamma(v_i | u_i) + \\
\lambda_W \|W\|^2 + \lambda_M \|M\|^2 + \lambda_A \|A\|^2 + \lambda_B \|B\|^2
\]
Loss Function

The graph shows different loss functions as a function of $y_f$. The $l_2$ loss is represented by the black solid line, while the log loss is shown by the red dotted line. The Huber loss is depicted by the blue dashed line, and the Psi loss is indicated by the green dashed line. Each curve illustrates how the loss changes with respect to $y_f$. At $y_f = 0$, the $l_2$ loss is highest, and it decreases as $y_f$ moves away from zero, while all other losses decrease more gradually.
Loss

- Much more evidence of application non-install (i.e. many more negative examples)
- Few links between vertices in friendship network (even within short graph distance)
- Generate ranking problems (link, non-link) with non-links drawn from background set
Loss

application recommendation

social recommendation
• Nonconvex optimization problem
• Large set of variables
• Stochastic gradient descent on $x, v, \varepsilon$ for speed
• Use hashing to reduce memory load, i.e.

\begin{align*}
x_i &= Ay_i + \epsilon_i \\
v_j &= Bu_j + \tilde{\epsilon}_j \\
e_{ij} &\sim p(e|x_i^T x_j + y_i^T W y_j) \\
a_{ij} &\sim p(a|x_i^T v_j + y_i^T M u_j) \\
x_{ij} &= \sigma(i, j) X[h(i, j)]
\end{align*}
Y! Pulse

Share what's important to you

... with the people you care about

Connect to your favorite sites
Y! Pulse Data

1.2M users, 386 items
6.1M friend connections
29M interest indications
## App Recommendation

<table>
<thead>
<tr>
<th>Models</th>
<th>loss</th>
<th>$\Omega[\cdot]$</th>
<th>MAP@5</th>
<th>MAR@5</th>
<th>nDCG@5</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIM</td>
<td></td>
<td>0.630</td>
<td>0.186</td>
<td>0.698</td>
<td></td>
</tr>
<tr>
<td>RLFM</td>
<td></td>
<td>0.729</td>
<td>0.211</td>
<td>0.737</td>
<td></td>
</tr>
<tr>
<td>NLFM</td>
<td></td>
<td>0.748</td>
<td>0.222</td>
<td>0.761</td>
<td></td>
</tr>
<tr>
<td>FIP</td>
<td>$l_2$</td>
<td>$l_2$</td>
<td>0.768</td>
<td>0.228</td>
<td>0.774</td>
</tr>
<tr>
<td>FIP</td>
<td>lazy $l_2$</td>
<td>$l_2$</td>
<td>0.781</td>
<td>0.232</td>
<td>0.790</td>
</tr>
<tr>
<td>FIP</td>
<td>logistic $l_2$</td>
<td>$l_2$</td>
<td>0.781</td>
<td>0.232</td>
<td>0.793</td>
</tr>
<tr>
<td>FIP</td>
<td>Huber  $l_2$</td>
<td>$l_2$</td>
<td>0.781</td>
<td>0.232</td>
<td>0.794</td>
</tr>
<tr>
<td>FIP</td>
<td>$\Psi$</td>
<td>$l_2$</td>
<td>0.777</td>
<td>0.231</td>
<td>0.771</td>
</tr>
<tr>
<td>FIP</td>
<td>$l_2$</td>
<td>$l_1$</td>
<td>0.778</td>
<td>0.231</td>
<td>0.787</td>
</tr>
<tr>
<td>FIP</td>
<td>lazy $l_2$</td>
<td>$l_1$</td>
<td>0.780</td>
<td>0.231</td>
<td>0.791</td>
</tr>
<tr>
<td>FIP</td>
<td>logistic $l_1$</td>
<td>$l_1$</td>
<td>0.779</td>
<td>0.231</td>
<td>0.792</td>
</tr>
<tr>
<td>FIP</td>
<td>Huber  $l_1$</td>
<td>$l_1$</td>
<td><strong>0.786</strong></td>
<td><strong>0.233</strong></td>
<td><strong>0.797</strong></td>
</tr>
<tr>
<td>FIP</td>
<td>$\Psi$</td>
<td>$l_1$</td>
<td>0.765</td>
<td>0.215</td>
<td>0.772</td>
</tr>
</tbody>
</table>

SIM: similarity based model;
RLFM: regression based latent factor model (Chen&Agarwal);
NLFM: SIM&RLFM
Social recommendation

<table>
<thead>
<tr>
<th>Models</th>
<th>loss</th>
<th>$\Omega[\cdot]$</th>
<th>MAP@5</th>
<th>MAR@5</th>
<th>nDCG@5</th>
</tr>
</thead>
<tbody>
<tr>
<td>RLFM</td>
<td>$\ell_2$</td>
<td>$\ell_2$</td>
<td>0.164</td>
<td>0.202</td>
<td>0.174</td>
</tr>
<tr>
<td>FIP</td>
<td>$\ell_2$</td>
<td>$\ell_2$</td>
<td>0.359</td>
<td>0.284</td>
<td>0.244</td>
</tr>
<tr>
<td>FIP</td>
<td>lazy $\ell_2$</td>
<td>$\ell_2$</td>
<td>0.193</td>
<td>0.269</td>
<td>0.200</td>
</tr>
<tr>
<td>FIP</td>
<td>logistic</td>
<td>$\ell_2$</td>
<td>0.174</td>
<td>0.220</td>
<td>0.189</td>
</tr>
<tr>
<td>FIP</td>
<td>Huber</td>
<td>$\ell_2$</td>
<td>0.210</td>
<td>0.234</td>
<td>0.215</td>
</tr>
<tr>
<td>FIP</td>
<td>$\Psi$</td>
<td>$\ell_2$</td>
<td>0.187</td>
<td>0.255</td>
<td>0.185</td>
</tr>
<tr>
<td>FIP</td>
<td>$\ell_2$</td>
<td>$\ell_1$</td>
<td>0.186</td>
<td>0.230</td>
<td>0.214</td>
</tr>
<tr>
<td>FIP</td>
<td>lazy $\ell_2$</td>
<td>$\ell_1$</td>
<td>0.180</td>
<td>0.223</td>
<td>0.194</td>
</tr>
<tr>
<td>FIP</td>
<td>logistic</td>
<td>$\ell_1$</td>
<td>0.183</td>
<td>0.217</td>
<td>0.189</td>
</tr>
<tr>
<td>FIP</td>
<td>Huber</td>
<td>$\ell_1$</td>
<td>0.188</td>
<td>0.222</td>
<td>0.200</td>
</tr>
<tr>
<td>FIP</td>
<td>$\Psi$</td>
<td>$\ell_1$</td>
<td>0.178</td>
<td>0.208</td>
<td>0.179</td>
</tr>
</tbody>
</table>
app recommendation

L2 penalty
• **Multiple relations**

- (user, user)
- (user, app)
- (app, advertisement)
• Multiple relations

(user, user)
(user, app)
(app, advertisement)
• **Multiple relations**
  
  (user, user)
  
  (user, app)
  
  (app, advertisement)
Extensions

- Multiple relations
  - (user, user)
  - (user, app)
  - (app, advertisement)
- Users visiting several properties
  - news, mail, frontpage, social network, etc.
- Different statistical models
  - Latent Dirichlet Allocation for latent factors
  - Indian Buffet Process
Multiple factor LDA

• Discrete set of preferences
  (Porteous, Bart, Welling, 2008)
• User picks one to assess movie
• Movie represented by a discrete attribute
• Inference by Gibbs sampler
• Works fairly well

• Extension by Lester Mackey and coworkers to combine with BPMF model
More state representations

• Indian Buffet Process (Griffiths & Ghahramani, 2005)
• Attribute vector is binary string
• Models preferences naturally & very compact (Inference is costly)
• Hierarchical attribute representation and clustering over users ... TO DO
5 Hashing
Parameter Storage

- We have millions of users
- We have millions of products
- Storage - for 100 factors this requires
  \[10^6 \times 10^6 \times 8 = 8\text{TB}\]
- We want a model that can be kept in RAM (<16GB)
  - Instant response for each user
  - Disks have 20 IOP/s at best (SSDs much better)
- Privacy (what if parameter vector leaks)
Recall - Hash Kernels

instance:

Hey,
please mention subtly during your talk that people should use Yahoo mail more often. Thanks,

Someone

task/user (=barney):

Similar to count hash (Charikar, Chen, Farrach-Colton, 2003)
Collaborative Filtering

- Hashing compression

\[ u_i = \sum_{j,k: h(j,k) = i} \xi(j,k) U_{jk} \quad \text{and} \quad v_i = \sum_{j,k: h'(j,k) = i} \xi'(j,k) V_{jk}. \]

\[ X_{ij} := \sum_k \xi(k,i) \xi'(k,j) u_{h(k,i)} v_{h'(k,j)}. \]

- Approximation is $O(1/n)$
  - To show that estimate is unbiased take expectation over Rademacher hash.
• **Hashing compression**

\[
u_i = \sum_{j,k : h(k,j)=i} \xi(k,j)U_{kj} \quad \text{and} \quad v_i = \sum_{j,k : h'(k,j)=i} \xi'(k,j)V_{kj}.
\]

\[
X_{ij} := \sum_k \xi(k,i)\xi'(k,j)u_{h(k,i)}v_{h'(k,j)}.
\]

• **Expectation**

\[
X_{ij} := \sum_k \xi(k,i)\xi'(k,j) \sum_{l,k : h(k,l)=h(k,i)} \sum_{o,k : h'(k,o)=h'(k,j)} \xi(k,l)\xi'(k,o)U_{kl}V_{ko}
\]
Collaborative Hashing

- Combine with stochastic gradient descent
- Random access in memory is expensive (we now have to do k lookups per pair)
- Feistel networks can accelerate this

- Distributed optimization without locking
Examples

Eachmovie

MovieLens
Summary

- Neighborhood methods
  - User / movie similarity
  - Iteration on graph
- Matrix Factorization
  - Singular value decomposition
  - Convex reformulation
- Ranking and Session Modeling
  - Ordinal regression
  - Session models
- Features
  - Latent dense (Bayesian Probabilistic Matrix Factorization)
  - Latent sparse (Dirichlet process factorization)
  - Coldstart problem (inferring features)
- Hashing
• Collaborative Filtering with temporal dynamics
• Neighborhood factorization
• Matrix Factorization for recommender systems
• CoFi Rank (collaborative filtering & ranking)
  http://www.cofirank.org/
• Yehuda Koren’s papers
  http://research.yahoo.com/Yehuda_Koren