Recommender Systems: Latent Factor Models

CS246: Mining Massive Datasets
Jure Leskovec, Stanford University
http://cs246.stanford.edu
The Netflix Prize

Training data
- 100 million ratings, 480,000 users, 17,770 movies
- 6 years of data: 2000-2005

Test data
- Last few ratings of each user (2.8 million)
- Evaluation criterion: Root Mean Square Error (RMSE)
- Netflix Cinematch RMSE: 0.9514

Competition
- 2700+ teams
- $1 million prize for 10% improvement on Cinematch
The Netflix Utility Matrix

17,700 movies

480,000 users
### Utility Matrix: Evaluation

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>?</td>
<td>?</td>
<td>?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>?</td>
<td>?</td>
<td>?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Test Data Set

**SSE** = $\sum_{(i,u) \in R} (r_{ui} - \hat{r}_{ui})^2$
BellKor Recommender System

- Basically the winner of the Netflix Challenge
- Multi-scale modeling of the data: Combine top level, regional modeling of the data, with a refined, local view:
  - **Global:**
    - Overall deviations of users/movies
  - **Factorization:**
    - Addressing regional effects
  - **CF (k-NN):**
    - Extract local patterns
Global:

- Mean movie rating: 3.7 stars
- The Sixth Sense is 0.5 stars above avg.
- Joe rates 0.2 stars below avg.

⇒ Baseline estimation:

 Joe will rate *The Sixth Sense* 4 stars

Local neighborhood (CF/NN):

- Joe didn’t like related movie *Signs*

⇒ Final estimate:

 Joe will rate *The Sixth Sense* 3.8 stars
Recap: Collaborative Filtering (CF)

- Earliest and most popular collaborative filtering method
- Derive unknown ratings from those of “similar” movies (item-item variant)
- Define similarity measure \( s_{ij} \) of items \( i \) and \( j \)
- Select \( k \)-nearest neighbors, compute the rating
  - \( N(i; u) \): items most similar to \( i \) that were rated by \( u \)

\[
\hat{r}_{ui} = \frac{\sum_{j \in N(i; u)} S_{ij} r_{uj}}{\sum_{j \in N(i; u)} S_{ij}}
\]

\( s_{ij} \)… similarity of items \( i \) and \( j \)
\( r_{uj} \)... rating of user \( u \) on item \( j \)
\( N(i; u) \)... set of similar items
In practice we get better estimates if we model deviations:

\[
\hat{r}_{ui} = b_{ui} + \frac{\sum_{j \in N(i;u)} S_{ij} (r_{uj} - b_{uj})}{\sum_{j \in N(i;u)} S_{ij}}
\]

Baseline estimate for \( r_{ui} \)

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

- \( \mu \) = overall mean rating
- \( b_u \) = rating deviation of user \( u \)
- \( = \text{avg. rating of user } u - \mu \)
- \( b_i \) = avg. rating of movie \( i \) – \( \mu \)

Problems:
1) Similarity measures are arbitrary
2) Pairwise similarities neglect interdependencies among neighbors
3) Taking a weighted average is restricting
Idea: Interpolation Weights

- Use a **weighted sum** rather than weighted avg.:
  \[
  \hat{r}_{ui} = b_{ui} + \sum_{j \in N(i;u)} w_{ij} (r_{uj} - b_{uj})
  \]

- **How to set** \( w_{ij} \)?
  - Remember, error metric is **SSE**: \( \sum_{(i,u) \in R} (r_{ui} - \hat{r}_{ui})^2 \)
  - Find \( w_{ij} \) that minimize **SSE** on training data!
    \[
    \min_w \sum_v (r_{vi} - [b_{vi} + \sum_{j \in N(i;v)} w_{ij} (r_{vj} - b_{vj})])^2
    \]
    - Models relationships between item \( i \) and its neighbors \( j \)
    - \( w_{ij} \) can be learnt through **gradient decent** based on \( u \) and all other users \( v \) that rated \( i \)
Interpolation Weights

- Find $w_{ij}$ that minimize $\text{SSE}$ on training data!

$$
\min_w \sum_v \left( r_{vi} - \left[ b_{vi} + \sum_{j \in N(i;v)} w_{ij} (r_{vj} - b_{vj}) \right] \right)^2
$$

- Gradient decent
  - Iterate until convergence: $w \leftarrow w - \eta \cdot \nabla w$
  - Where:
    $$
    \nabla w_{ij} = 2 \sum_v \left( r_{vi} - \left[ b_{vi} + \sum_{k \in N(i;v)} w_{ik} (r_{vk} - b_{vk}) \right] \right) (r_{vj} - b_{vj})
    $$
    for $j \in N(i;v)$ (else $\nabla w_{ij} = 0$)
Interpolation Weights

- **So far:** \( \hat{r}_{ui} = b_{ui} + \sum_{j \in N(i,u)} w_{ij} (r_{uj} - b_{uj}) \)
  - Weights \( w_{ij} \) derived based on their role; no use of an arbitrary similarity measure \( (w_{ij} \neq s_{ij}) \)
  - Explicitly account for interrelationships among the neighboring movies
- **Next:** Latent factor model
  - Extract “regional” correlations
Latent Factor Models (e.g., SVD)

- Geared towards females
  - The Color Purple
  - Sense and Sensibility
  - The Princess Diaries

- Geared towards males
  - Amadeus
  - Braveheart
  - Lethal Weapon

- Funny
  - Ocean's 11
  - Independence Day
  - Dumb and Dumber

- Serious
  - The Lion King

2/6/2012
“SVD” on Netflix data: \( R \approx Q \cdot P^T \)

For now let’s assume we can approximate the rating matrix \( R \) as a product of “thin” \( Q \cdot P^T \). There are important differences between “SVD” and the real SVD. We will get to them later.
Ratings as Products of Factors

- How to estimate the missing rating?

\[ \hat{r}_{iu} = q_i \cdot p_u^T \]
Ratings as Products of Factors

- How to estimate the missing rating?

\[ \hat{r}_{iu} = q_i \cdot p_u^T \]
Ratings as Products of Factors

- How to estimate the missing rating?

\[ \hat{r}_{iu} = q_i \cdot p_u^T \]
Latent Factor Models

The Color Purple
Sense and Sensibility
The Princess Diaries
The Lion King

Geared towards females

Serious

Braveheart
Lethal Weapon
Ocean’s 11

Geared towards males

Factor 1

Funny

Independence Day
Dumb and Dumber

Factor 2
Latent Factor Models

The Color Purple
Amadeus
Sense and Sensibility
Ocean’s 11
The Princess Diaries
The Lion King
Independence Day
Dumb and Dumber

Factor 1
Factor 2

serious
funny

Geared towards females
Geared towards males

serious
funny
Recap: SVD

- **Remember SVD:**
  - \( A \): Input data matrix
  - \( U \): Left singular vecs
  - \( V \): Right singular vecs
  - \( \Sigma \): Singular values
  - **SVD gives minimum reconstruction error (MSE!)**

\[
\min_{U,V,\Sigma} \sum_{ij} (A_{ij} - [U\Sigma V^T]_{ij})^2
\]

- The sum goes over all entries.
- Our \( A/R \) has missing entries!

- So in our case, “SVD” on Netflix data: \( R \approx Q \cdot P^T \)
  - \( A = R, \ Q = U, \ P^T = \Sigma V^T \)
  - \( \hat{r}_{iu} = q_i \cdot p_u^T \)

- But, we are not done yet! \( R \) has missing entries!
SVD isn’t defined when entries are missing

Use specialized methods to find $P$, $Q$

$$\min_{P,Q} \sum_{(i,u) \in R} (r_{iu} - q_i \cdot p_u^T)^2$$

$$\hat{r}_{iu} = q_i \cdot p_u^T$$

Don’t require cols of $P$, $Q$ to be orthogonal/unit length

$P$, $Q$ map users/movies to a latent space

The most popular model among Netflix contestants

---

Latent Factor Models

SVD isn’t defined when entries are missing

Use specialized methods to find $P$, $Q$

$$\min_{P,Q} \sum_{(i,u) \in R} (r_{iu} - q_i \cdot p_u^T)^2$$

$$\hat{r}_{iu} = q_i \cdot p_u^T$$

Don’t require cols of $P$, $Q$ to be orthogonal/unit length

$P$, $Q$ map users/movies to a latent space

The most popular model among Netflix contestants
Want to minimize SSE for test data

Idea: Minimize SSE on Training data

- Want large $f$ (# of factors) to capture all the signals
- But, test SSE begins to rise for $f > 2$

Regularization is needed

- Allow rich model where there are sufficient data
- Shrink aggressively where data are scarce

$$\min_{P,Q} \sum_{\text{training}} (r_{ui} - q_i p_u^T)^2 + \lambda \left[ \sum_u \|p_u\|^2 + \sum_i \|q_i\|^2 \right]$$

$\lambda$... regularization parameter

“error”

“length”

2/6/2012  Jure Leskovec, Stanford C246: Mining Massive Datasets
The Effect of Regularization

Geared towards females

The Color Purple

Sense and Sensibility

serious

The Princess Diaries

The Lion King

 Braveheart

Lethal Weapon

Ocean’s 11

Geared towards males

Factor 1

Factor 2

Dumb and Dumber

funny

Independence Day

\[
\min_{p,q} \sum_{i \in \text{training}} (r_{ui} - q_i p_u^T)^2 + \lambda \left[ \sum_u \|p_u\|^2 + \sum_i \|q_i\|^2 \right]
\]

\min_{\text{factors}} \text{“error”} + \lambda \text{“length”}
**The Effect of Regularization**

The Color Purple

Sense and Sensibility

Geared towards females

The Princess Diaries

The Lion King

Factor 1

Ocean’s 11

Factor 2

Braveheart

Lethal Weapon

Dumb and Dumber

Geared towards males

serious

funny

min\(r_{ui} - q_i p_u^T\)\(^2 + \lambda \left[ \sum_u \|p_u\|^2 + \sum_i \|q_i\|^2 \right]

\min_{factors} \text{“error”} + \lambda \text{“length”}

2/6/2012

Jure Leskovec, Stanford C246: Mining Massive Datasets
The Effect of Regularization

\[ \min_{P,Q} \sum_{(i,j) \in \text{training}} (r_{ij} - q_i p_j^T)^2 + \lambda \left[ \sum_i \|p_i\|^2 + \sum_j \|q_j\|^2 \right] \]

\[ \min_{\text{factors}} \text{“error”} + \lambda \text{“length”} \]
The Effect of Regularization

The Color Purple

serious

Amadeus

Braveheart

Geared towards males

Sense and Sensibility

Ocean's 11

Geared towards females

Factor 1

Lethal Weapon

Factor 2

Dumb and Dumber

The Princess Diaries

The Lion King

Independence Day

funny

\[
\min_{p,q} \sum_{i \in \text{training}} (r_{ui} - q_i p_u^T)^2 + \lambda \left[ \sum_u \|p_u\|^2 + \sum_i \|q_i\|^2 \right]
\]

\[
\min_{\text{factors}} \text{“error”} + \lambda \text{“length”}
\]
Want to find matrices $P$ and $Q$:

\[ \min_{P,Q} \sum_{training} (r_{ui} - q_i p_u^T)^2 + \lambda \left[ \sum_u \|p_u\|^2 + \sum_i \|q_i\|^2 \right] \]

Online “stochastic” gradient decent:

- Initialize $P$ and $Q$ (random, using SVD)
- Then iterate over the ratings and update factors:
  - For each $r_{ui}$:
    - $\varepsilon_{ui} = r_{ui} - q_i \cdot p_u^T$ (derivative of the “error”)
    - $q_i \leftarrow q_i + \eta (\varepsilon_{ui} p_u - \lambda q_i)$ (update equation)
    - $p_u \leftarrow p_u + \eta (\varepsilon_{ui} q_i - \lambda p_u)$ (update equation)
  - $\eta$ … learning rate
Modeling Biases and Interactions

Baseline predictor
- Separates users and movies
- Benefits from insights into user’s behavior
- Among the main practical contributions of the competition

User-Movie interaction
- Characterizes the matching between users and movies
- Attracts most research in the field
- Benefits from algorithmic and mathematical innovations

- $\mu =$ overall mean rating
- $b_u =$ bias of user $u$
- $b_i =$ bias of movie $i$
Baseline Predictor

- We have expectations on the rating by user $u$ of movie $i$, even without estimating $u$’s attitude towards movies like $i$

  - Rating scale of user $u$
  - Values of other ratings user gave recently (day-specific mood, anchoring, multi-user accounts)
  - (Recent) popularity of movie $i$
  - Selection bias; related to number of ratings user gave on the same day ("frequency")
Putting It All Together

\[ r_{ui} = \mu + b_u + b_i + q_i \cdot p_u^T \]

- Overall mean rating
- Bias for user \( u \)
- Bias for movie \( i \)
- User-Movie interaction

Example:

- Mean rating: \( \mu = 3.7 \)
- You are a critical reviewer: your ratings are 1 star lower than the mean: \( b_u = -1 \)
- Star Wars gets a mean rating of 0.5 higher than average movie: \( b_i = +0.5 \)
- Predicted rating for you on Star Wars:
  \[ = 3.7 - 1 + 0.5 = 3.2 \]
Fitting the New Model

- **Solve:**

\[
\min_{Q,P} \sum_{(u,i) \in R} \left( r_{ui} - (\mu + b_u + b_i + q_i p_u^T) \right)^2
\]

goodness of fit

\[
+ \lambda \left( \|q_i\|^2 + \|p_u\|^2 + \|b_u\|^2 + \|b_i\|^2 \right)
\]

\(\lambda\) is typically selected via grid-search on a validation set

- **Stochastic gradient decent to find parameters**

  - **Note:** Both biases \((b_u, b_i)\) as well as interactions \((q_i, p_u)\) are treated as parameters (we estimate them)
Performance of Various Methods

Global average: 1.1296
User average: 1.0651
Movie average: 1.0533
Netflix: 0.9514

Basic Collaborative filtering: 0.94
Collaborative filtering++: 0.91
Latent factors: 0.90
Latent factors+Biases: 0.89
Final BellKor: 0.869
Grand Prize: 0.8563
Performance of Various Methods

- CF (no time bias)
- Basic Latent Factors
- Latent Factors w/ Biases

RMSE vs. Millions of parameters graph.
Temporal Biases Of Users

- Sudden rise in the average movie rating (early 2004)
  - Improvements in Netflix
  - GUI improvements
  - Meaning of rating changed

- Movie age
  - Users prefer new movies without any reasons
  - Older movies are just inherently better than newer ones

Y. Koren, Collaborative filtering with temporal dynamics, KDD ’09
Temporal Biases & Factors

- **Original model:**
  \[ r_{ui} = \mu + b_u + b_i + q_i \cdot p_u^T \]

- **Add time dependence to biases:**
  \[ r_{ui} = \mu + b_u(t) + b_i(t) + q_i \cdot p_u^T \]
  - Make parameters \( b_u \) and \( b_i \) to depend on time
  - **(1)** Parameterize time-dependence by linear trends
  - **(2)** Each bin corresponds to 10 consecutive weeks
    \[ b_i(t) = b_i + b_{i,\text{Bin}(t)} \]

- **Add temporal dependence to factors**
  - \( p_u(t) \)... user preference vector on day \( t \)

---

Y. Koren, Collaborative filtering with temporal dynamics, KDD ’09

Jure Leskovec, Stanford C246: Mining Massive Datasets
Adding Temporal Effects

![Graph showing the improvement of adding temporal effects to CF models.](image)

- **RMSE** (Root Mean Square Error) is plotted against the number of millions of parameters.
- The graph compares different models:
  - **CF (no time bias)**
  - **Basic Latent Factors**
  - **CF (time bias)**
  - **Latent Factors w/ Biases**
  - **+ Linear time factors**
  - **+ Per-day user biases**
  - **+ CF**

The graph illustrates how adding temporal factors reduces RMSE as the number of parameters increases.
Many options for modeling

- Variants of the ideas we have seen so far
  - Different numbers of factors
  - Different ways to model time
  - Different ways to handle implicit information
- Other models (not described here)
  - Nearest-neighbor models
  - Restricted Boltzmann machines

Model averaging is useful....

- Linear model combining
The big picture

Solution of BellKor's Pragmatic Chaos

All developed CF models
- BRISMF
- SVD-Time
- Split RBM
- Movie KNN
- User KNN
- NSVDD
- RBM
- SBRAMF
- 1/2/3
- Movie KNN
- Baseline
- SVD++
- Integrated M.
- SVD-AUF
- KNN+time
- DRBM
- RBM
- ISVDD2
- MF2
- KNN
- GBM
- 3K1
- GTE
- 3K2
- 3K4
- 1/2/3
- Asym.

Latent User and Movie Features

Probe Blending

approx. 500 predictors

Probe Blending

Linear Blend 10.09 % improvement

200 blends

30 blends
Standing on June 26th 2009

June 26th submission triggers 30-day “last call”
The Last 30 Days

- **Ensemble team formed**
  - Group of other teams on leaderboard forms a new team
  - Relies on combining their models
  - Quickly also get a qualifying score over 10%

- **BellKor**
  - Continue to get small improvements in their scores
  - Realize that they are in direct competition with Ensemble

- **Strategy**
  - Both teams carefully monitoring the leaderboard
  - Only sure way to check for improvement is to submit a set of predictions
    - This alerts the other team of your latest score
Submissions limited to 1 a day
  - Only 1 final submission could be made in the last 24h

24 hours before deadline...
  - BellKor team member in Austria notices (by chance) that Ensemble posts a score that is slightly better than BellKor’s

Frantic last 24 hours for both teams
  - Much computer time on final optimization
  - run times carefully calibrated to end about an hour before deadline

Final submissions
  - BellKor submits a little early (on purpose), 40 mins before deadline
  - Ensemble submits their final entry 20 mins later
  - ....and everyone waits....
## Leaderboard

Showing Test Score. Click here to show quiz score

Display top leaders.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Team Name</th>
<th>Best Test Score</th>
<th>% Improvement</th>
<th>Best Submit Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>BellKor's Pragmatic Chaos</td>
<td>0.8567</td>
<td>10.06</td>
<td>2009-07-26 18:18:28</td>
</tr>
<tr>
<td>2</td>
<td>The Ensemble</td>
<td>0.8567</td>
<td>10.06</td>
<td>2009-07-26 18:38:22</td>
</tr>
<tr>
<td>3</td>
<td>Grand Prize Team</td>
<td>0.8582</td>
<td>9.38</td>
<td>2009-01-10 21:24:46</td>
</tr>
<tr>
<td>4</td>
<td>Opera Solutions and Vandelay United</td>
<td>0.8588</td>
<td>9.84</td>
<td>2009-07-10 01:12:31</td>
</tr>
<tr>
<td>5</td>
<td>Vandelay Industries</td>
<td></td>
<td>0.8591</td>
<td>9.81</td>
</tr>
<tr>
<td>6</td>
<td>PragmaticTheory</td>
<td>0.8594</td>
<td>9.77</td>
<td>2009-06-24 12:06:56</td>
</tr>
<tr>
<td>7</td>
<td>BellKor in BigChaos</td>
<td>0.8601</td>
<td>9.70</td>
<td>2009-05-13 08:14:09</td>
</tr>
<tr>
<td>8</td>
<td>Dace</td>
<td>0.8612</td>
<td>9.59</td>
<td>2009-07-24 17:18:43</td>
</tr>
<tr>
<td>9</td>
<td>Feeds2</td>
<td>0.8622</td>
<td>9.48</td>
<td>2009-07-12 13:11:51</td>
</tr>
<tr>
<td>10</td>
<td>BigChaos</td>
<td>0.8623</td>
<td>9.47</td>
<td>2009-04-07 12:33:59</td>
</tr>
<tr>
<td>11</td>
<td>Opera Solutions</td>
<td>0.8623</td>
<td>9.47</td>
<td>2009-07-24 00:34:07</td>
</tr>
<tr>
<td>12</td>
<td>BellKor</td>
<td>0.8624</td>
<td>9.46</td>
<td>2009-07-26 17:19:11</td>
</tr>
</tbody>
</table>

### Progress Prize 2008 - RMSE = 0.8627 - Winning Team: BellKor in BigChaos

<table>
<thead>
<tr>
<th>Rank</th>
<th>Team Name</th>
<th>Best Test Score</th>
<th>% Improvement</th>
<th>Best Submit Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>xiangliang</td>
<td>0.8642</td>
<td>9.27</td>
<td>2009-07-15 14:53:22</td>
</tr>
<tr>
<td>14</td>
<td>Gravity</td>
<td>0.8643</td>
<td>9.26</td>
<td>2009-04-22 18:31:32</td>
</tr>
<tr>
<td>15</td>
<td>Ces</td>
<td>0.8651</td>
<td>9.18</td>
<td>2009-06-21 19:24:53</td>
</tr>
<tr>
<td>16</td>
<td>Invisible Ideas</td>
<td>0.8653</td>
<td>9.15</td>
<td>2009-07-15 15:53:04</td>
</tr>
<tr>
<td>17</td>
<td>Just a guy in a garage</td>
<td>0.8662</td>
<td>9.06</td>
<td>2009-05-24 10:02:54</td>
</tr>
<tr>
<td>18</td>
<td>J Dennis Su</td>
<td>0.8666</td>
<td>9.02</td>
<td>2009-03-07 17:16:17</td>
</tr>
<tr>
<td>19</td>
<td>Craig Carmichael</td>
<td>0.8666</td>
<td>9.02</td>
<td>2009-07-25 16:00:54</td>
</tr>
<tr>
<td>20</td>
<td>acmehill</td>
<td>0.8666</td>
<td>9.00</td>
<td>2009-03-21 16:20:50</td>
</tr>
</tbody>
</table>

### Progress Prize 2007

Jure Leskovec, Stanford C246: Mining Massive Datasets
Million $ Awarded Sept 21st 2009
Some slides and plots borrowed from Yehuda Koren, Robert Bell and Padhraic Smyth

Further reading:

- Y. Koren, Collaborative filtering with temporal dynamics, KDD ’09
  - http://www.the-ensemble.com/