A Survey on Recommender Systems (BR4CP 2012)

F. Koriche and J. Mengin

Merely based on the book of Jannach et al. [6]
Plan

1 Introduction

2 Collaborative Recommendation
   - User-Based Nearest Neighbor Recommendation
   - Item-Based Nearest Neighbor Recommendation
   - Limitations
   - Recent Approaches

3 Content-Based Recommendation
   - Linear Predictors
   - Linear Features
   - Limitations

4 Hybrid Recommendation
   - Voting Systems
   - Collaborative Features

5 Conclusions
Recommender System
Online decision maker that predicts which items should be shown to a specific person
Recommender System
Online decision maker that predicts which items should be shown to a specific person
Recommender System
Online decision maker that predicts which items should be shown to a specific person
Recommender System
Online decision maker that predicts which items should be shown to a specific person
Recommendation Problem (General)
Given
- a set $U = \{u_1, \cdots, u_n\}$ of users
- a set $P = \{p_1, \cdots, p_m\}$ of items
Find a ranking $\text{rank} : U \rightarrow \mathbb{S}(P)$, where $\mathbb{S}(P)$ is the symmetric group of all permutations over $P$

The problem can be generalized further using top $k$ permutations

Recommendation Problem (Standard)
Given
- a set $U = \{u_1, \cdots, u_n\}$ of users
- a set $P = \{p_1, \cdots, p_m\}$ of items
Find a utility function $\text{pred} : U \times P \rightarrow [0, 1]$
Recommendation Problem (General)

Given
- a set \( U = \{u_1, \cdots, u_n\} \) of users
- a set \( P = \{p_1, \cdots, p_m\} \) of items

Find a ranking \( \text{rank} : U \rightarrow \mathbb{S}(P) \), where \( \mathbb{S}(P) \) is the symmetric group of all permutations over \( P \)

The problem can be generalized further using top \( k \) permutations

Recommendation Problem (Standard)

Given
- a set \( U = \{u_1, \cdots, u_n\} \) of users
- a set \( P = \{p_1, \cdots, p_m\} \) of items

Find a utility function \( \text{pred} : U \times P \rightarrow [0, 1] \)
Plan

1 Introduction

2 Collaborative Recommendation
   - User-Based Nearest Neighbor Recommendation
   - Item-Based Nearest Neighbor Recommendation
   - Limitations
   - Recent Approaches

3 Content-Based Recommendation
   - Linear Predictors
   - Linear Features
   - Limitations

4 Hybrid Recommendation
   - Voting Systems
   - Collaborative Features

5 Conclusions
Collaborative Recommendation

Matrix representation of items rated by users

- Assumes a large community of users who rate most of the items
- Does not assume any information about users or items
Collaborative Recommendation

Matrix representation of items rated by users

- **Assumes** a large community of users who rate most of the items
- Does not assume any information about users or items
Collaborative Recommendation

Matrix representation of items rated by users

- **Assumes** a large community of users who rate most of the items
- **Does not assume** any information about users or items
### User-Based Nearest Neighbor Recommendation (early 1990s)

Given a rating matrix, a target user \( u \) and a target item \( p \)

- Identify the neighbors of \( u \)
- Predict the rating for \( p \) by \( u \) using the ratings for \( p \) made by the neighbors.

<table>
<thead>
<tr>
<th></th>
<th>Item 1</th>
<th>Item 2</th>
<th>Item 3</th>
<th>Item 4</th>
<th>Item 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>?</td>
</tr>
<tr>
<td>User 1</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>User 2</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>User 3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>User 4</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>
### User-Based Nearest Neighbor Recommendation (early 1990s)

Given a rating matrix, a target user $u$ and a target item $p$

- Identify the neighbors of $u$
- Predict the rating for $p$ by $u$ using the ratings for $p$ made by the neighbors.

**Rating Matrix**

<table>
<thead>
<tr>
<th></th>
<th>Item 1</th>
<th>Item 2</th>
<th>Item 3</th>
<th>Item 4</th>
<th>Item 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>?</td>
</tr>
<tr>
<td>User 1</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>User 2</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>User 3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>User 4</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>
## User-Based Nearest Neighbor Recommendation (early 1990s)

Given a rating matrix, a target user $u$ and a target item $p$

- Identify the **neighbors** of $u$
- Predict the rating for $p$ by $u$ using the ratings for $p$ made by the neighbors.

<table>
<thead>
<tr>
<th>Item 1</th>
<th>Item 2</th>
<th>Item 3</th>
<th>Item 4</th>
<th>Item 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>User 1</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>User 2</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>User 3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>User 4</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>2</td>
</tr>
</tbody>
</table>
Collaborative Recommendation
User-Based Nearest Neighbor Recommendation

<table>
<thead>
<tr>
<th></th>
<th>Item 1</th>
<th>Item 2</th>
<th>Item 3</th>
<th>Item 4</th>
<th>Item 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>?</td>
</tr>
<tr>
<td>User 1</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>User 2</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>User 3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>User 4</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

Notation

- Set of users: $U = \{u_1, \cdots, u_n\}$
- Set of items: $P = \{p_1, \cdots, p_m\}$
- Rating matrix: $R = [r_{ij}]$, where $i \in U$ and $j \in P$. 
Collaborative Recommendation

User-Based Nearest Neighbor Recommendation

<table>
<thead>
<tr>
<th></th>
<th>Item 1</th>
<th>Item 2</th>
<th>Item 3</th>
<th>Item 4</th>
<th>Item 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>+1.00</td>
<td>Alice</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>+0.85</td>
<td>User 1</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>+0.70</td>
<td>User 2</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>+0.00</td>
<td>User 3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>−0.79</td>
<td>User 4</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>2</td>
</tr>
</tbody>
</table>

User Similarity (Pearson’s correlation coefficient)

\[
sim(u, v) = \frac{\sum_{p \in P} (r_{u,p} - \bar{r}_u)(r_{v,p} - \bar{r}_v)}{\sqrt{\sum_{p \in P} (r_{u,p} - \bar{r}_u)^2} \sqrt{\sum_{p \in P} (r_{v,p} - \bar{r}_v)^2}}
\]

where \( \bar{r}_u \) is the average rating of user \( u \)
### Collaborative Recommendation

**User-Based Nearest Neighbor Recommendation**

<table>
<thead>
<tr>
<th></th>
<th>Item 1</th>
<th>Item 2</th>
<th>Item 3</th>
<th>Item 4</th>
<th>Item 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4.87</td>
</tr>
<tr>
<td>User 1</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>User 2</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>User 3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>User 4</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

**Predicted Rating**

\[
pred(u, p) = \bar{r}_u + \frac{\sum_{v \in N(u)} sim(u, v)(r_{v,p} - \bar{r}_v)}{\sum_{v \in N(u)} sim(u, v)}
\]

where \(N(u)\) is the set of \(u\)’s neighbors
Collaborative Recommendation

User-Based Nearest Neighbor Recommendation

<table>
<thead>
<tr>
<th></th>
<th>Item 1</th>
<th>Item 2</th>
<th>Item 3</th>
<th>Item 4</th>
<th>Item 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>+1.00</td>
<td>Alice</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>4.87</td>
</tr>
<tr>
<td>+0.85</td>
<td>User 1</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>+0.70</td>
<td>User 2</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>+0.00</td>
<td>User 3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>−0.79</td>
<td>User 4</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>2</td>
</tr>
</tbody>
</table>

Predicted Rating

\[
pred(u, p) = \bar{r}_u + \frac{\sum_{v \in N(u)} \text{sim}(u, v)(r_{v,p} - \bar{r}_v)}{\sum_{v \in N(u)} \text{sim}(u, v)}
\]

where \(N(u)\) is the set of \(u\)'s neighbors

Choosing the size of the neighborhood is a key problem
### Item-Based Nearest Neighbor Recommendation (early 2000s)

Given a rating matrix, a target user $u$ and a target item $p$

- Identify the neighbors of $p$
- Predict the rating for $p$ by $u$ using the ratings for the neighbors of $p$ made by $u$. 

<table>
<thead>
<tr>
<th></th>
<th>Item 1</th>
<th>Item 2</th>
<th>Item 3</th>
<th>Item 4</th>
<th>Item 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>?</td>
</tr>
<tr>
<td>User 1</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>User 2</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>User 3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>User 4</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>
## Item-Based Nearest Neighbor Recommendation (early 2000s)

Given a rating matrix, a target user $u$ and a target item $p$

- Identify the neighbors of $p$
- Predict the rating for $p$ by $u$ using the ratings for the neighbors of $p$ made by $u$. 

### Table

<table>
<thead>
<tr>
<th></th>
<th>Item 1</th>
<th>Item 2</th>
<th>Item 3</th>
<th>Item 4</th>
<th>Item 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>?</td>
</tr>
<tr>
<td>User 1</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>User 2</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>User 3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>User 4</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>
## Item-Based Nearest Neighbor Recommendation (early 2000s)

Given a rating matrix, a target user $u$ and a target item $p$

- Identify the neighbors of $p$
- Predict the rating for $p$ by $u$ using the ratings for the neighbors of $p$ made by $u$.

<table>
<thead>
<tr>
<th>Item 1</th>
<th>Item 2</th>
<th>Item 3</th>
<th>Item 4</th>
<th>Item 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>User 1</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>User 2</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>User 3</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>User 4</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>2</td>
</tr>
</tbody>
</table>
User Similarity (Cosine Similarity)

\[ \text{sim}(p, q) = \frac{\langle p, q \rangle}{\|p\| \|q\|} \]

where \( \| \cdot \| \) is the Euclidean norm. The cosine measure can be adjusted using average ratings.
Predicted Rating

\[
pred(u, p) = \frac{\sum_{q \in N(p)} sim(p, q) r_{u,q}}{\sum_{q \in N(p)} sim(p, q)}
\]

where \( N(p) \) is the set of \( p \)'s neighbors
### Sparsity and Cold-Start

In most applications the rating matrix is **sparse** (many missing values)

- Extend item ratings to paths of length $k$ [5].
- Use default values [2].
Sparsity and Cold-Start

In most applications the rating matrix is sparse (many missing values)

- Extend item ratings to paths of length $k$ [5].
- Use default values [2].
Sparsity and Cold-Start

In most applications the rating matrix is sparse (many missing values)

- Extend item ratings to paths of length $k$ [5].
- Use default values [2].

\[
P(Y \mid X) = \frac{\prod_{i=1}^{m} P(X_i \mid Y) \times P(Y)}{P(X)}
\]

where \( Y \) is the event "\( p \) is classified to \( r \), and \( X_i \) is the event "\( x_i \) is classified to \( r \)."

Slope One Predictors [7]

\[
dev(p, q) = \sum_{u \in S(p, q)} \frac{r_{u,p} - r_{u,q}}{|S(p, q)|}
\]

where \( S(p, q) \) is the set of rows containing entries for both \( p \) and \( q \)

\[
pred(u, p) = \sum_{q \in T(u, p)} \frac{dev(p, q) + r_{u,q}}{|T(u, p)|}
\]

where \( T(u, p) = \{ q \in P : r_{u,q} \neq *, |S(p, q)| > 0 \} \)

\[ P(Y \mid X) = \frac{\prod_{i=1}^{m} P(X_i \mid Y) \times P(Y)}{P(X)} \]

where \( Y \) is the event "\( p \) is classified to \( r \), and \( X_i \) is the event "\( x_i \) is classified to \( r \)."

Slope One Predictors [7]

\[ dev(p, q) = \sum_{u \in S(p, q)} \frac{r_{u,p} - r_{u,q}}{|S(p, q)|} \]

where \( S(p, q) \) is the set of rows containing entries for both \( p \) and \( q \)

\[ pred(u, p) = \sum_{q \in T(u, p)} \frac{dev(p, q) + r_{u,q}}{|T(u, p)|} \]

where \( T(u, p) = \{ q \in P : r_{u,q} \neq *, |S(p, q)| > 0 \} \)
Plan

1 Introduction

2 Collaborative Recommendation
   - User-Based Nearest Neighbor Recommendation
   - Item-Based Nearest Neighbor Recommendation
   - Limitations
   - Recent Approaches

3 Content-Based Recommendation
   - Linear Predictors
   - Linear Features
   - Limitations

4 Hybrid Recommendation
   - Voting Systems
   - Collaborative Features

5 Conclusions
Content-Based Recommendation

Items are rated by users using linear functions

- Assumes information (features) about items and users
- Does not assume a large number of users
Content-Based Recommendation

Items are rated by users using linear functions

- Assumes information (features) about items and users
- Does not assume a large number of users
Content-Based Recommendation

Items are rated by users using linear functions

- Assumes information (features) about items and users
- Does not assume a large number of users
Linear Prediction

The utility function is a linear function (additive independence principle)

\[
pred(u, p) = \langle w, x \rangle = \sum_{i=1}^{m+n} w_i x_i
\]

where \( x = u + p \)
Content-Based Recommendation as Convex Optimization

Let \( \{(x_t, x'_t)\}_{t=1}^T \) be a set of pairs of instances such that \( x_t \) is preferred to \( x'_t \).

Let \( f : \mathbb{R}^{m+n} \to \mathbb{R}_+ \) be a convex regularization function.

\[
\begin{align*}
\text{Minimize} & \quad f(w) \\
\text{Subject to} & \quad \langle w, x_t - x'_t \rangle > \zeta_i \text{ for all } t \in [T]
\end{align*}
\]

Various Learning Algorithms [10]

- Ordinal SVMs
- Boosting Algorithms
- Perceptron-Like Algorithms

Most algorithms can be extended to nonlinear functions using the kernel trick
Content-Based Recommendation as Convex Optimization

Let \( \{(x_t, x'_t)\}_{t=1}^{T} \) be a set of pairs of instances such that \( x_t \) is preferred to \( x'_t \).

Let \( f : \mathbb{R}^{m+n} \rightarrow \mathbb{R}_+ \) be a convex regularization function.

Minimize \( f(w) \)
Subject to \( \langle w, x_t - x'_t \rangle > \zeta_i \) for all \( t \in [T] \)

Various Learning Algorithms [10]

- Ordinal SVMs
- Boosting Algorithms
- Perceptron-Like Algorithms

Most algorithms can be extended to nonlinear functions using the kernel trick
Text Features

**TF-IDF format** [9]: for a word \( i \) in a document \( j \),

\[
x_i = \text{TF}(i, j) \cdot \text{IDF}(i) = \frac{\text{occ}(i, j)}{\max_{i'} \text{occ}(i', j)} \quad \text{and} \quad \text{IDF}(i) = \log \frac{m}{m(i)}
\]

and \( m(i) \) is the number of documents \( j \) that include \( i \).

The feature description is improved using [1]:

- Stop words and stemming
- Top \( k \) most informative words
- Context filters
Text Features
TF-IDF format [9]: for a word $i$ in a document $j$,

$$x_i = \text{TF}(i, j) \times \text{IDF}(i)$$

where $\text{TF}(i, j) = \frac{\text{occ}(i, j)}{\max_i \text{occ}(i', j)}$ and $\text{IDF}(i) = \log \frac{m}{m(i)}$

and $m(i)$ is the number of documents $j$ that include $i$.

The feature description is improved using [1]:

- Stop words and stemming
- Top $k$ most informative words
- Context filters

Tag Features (Web 2.0)
In addition to their rating, items can be annotated using tags or keywords (folksonomies). Tags can be encoded using different schemes:

- Term: TF-IDF format.
- Opinion: the weight $x_i(u, r)$ of a tag $i$ is given by the frequency of $i$ assigned to items which user $u$ has rated with rating value $r$. 
Relevance
Many features in the instance vector are irrelevant.
- Model selection techniques
- $\ell_1$-regularization techniques

Cold-Start
By sample complexity, as the number of features increases, the number of rated instances must also increase in order to yield accurate recommendations.
Plan

1 Introduction

2 Collaborative Recommendation
   - User-Based Nearest Neighbor Recommendation
   - Item-Based Nearest Neighbor Recommendation
   - Limitations
   - Recent Approaches

3 Content-Based Recommendation
   - Linear Predictors
   - Linear Features
   - Limitations

4 Hybrid Recommendation
   - Voting Systems
   - Collaborative Features

5 Conclusions
Hybrid Recommendation
Making recommendation by combining content-based and collaborative approaches
Hybrid Recommendation

Voting Systems

User profile

Item features

Community data

Hybrid Recommender

Recommendation List

\[
pred(u, p) = \sum_{i=1}^{k} w_i \times pred_i(u, p)
\]
<table>
<thead>
<tr>
<th>User</th>
<th>$R_{u,p}$</th>
<th>$P_{Alice,u}$</th>
<th>$C_{Alice,u}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>?</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>User 1</td>
<td>4</td>
<td>0.8</td>
<td>0.15</td>
</tr>
<tr>
<td>User 2</td>
<td>2.2</td>
<td>0.7</td>
<td>0.66</td>
</tr>
</tbody>
</table>

**Collaborative Features [8]**

- $R_{u,p}$ is the rating of item $p$ by user $u$
- $P_{Alice,u}$ is the Pearson’s correlation coefficient between Alice and $u$
- $C_{Alice,u}$ is ratio of overlapping ratings between Alice and $u$

$$C_{Alice,u} = \frac{|\{q : (R_{Alice,q} \neq *) \land (R_{u,q} \neq *)\}|}{|\{q : R_{Alice,q} \neq *\}|}$$

$$pred(Alice, p) = \langle w, x \rangle + \sum_{u=1}^{n} w_u \times C_{Alice,u} P_{Alice,u} R_{u,p}$$
Collaborative Features [8]

- $R_{u,p}$ is the rating of item $p$ by user $u$
- $P_{Alice,u}$ is the Pearson’s correlation coefficient between Alice and $u$
- $C_{Alice,u}$ is ratio of overlapping ratings between Alice and $u$

$$C_{Alice,u} = \frac{|\{q : (R_{Alice,q} \neq *) \land (R_{u,q} \neq *)\}|}{|\{q : R_{Alice,q} \neq *\}|}$$

$$pred(Alice, p) = \langle w, x \rangle + \sum_{u=1}^{n} w_u \times C_{Alice,u} P_{Alice,u} R_{u,p}$$
Plan

1 Introduction

2 Collaborative Recommendation
   - User-Based Nearest Neighbor Recommendation
   - Item-Based Nearest Neighbor Recommendation
   - Limitations
   - Recent Approaches

3 Content-Based Recommendation
   - Linear Predictors
   - Linear Features
   - Limitations

4 Hybrid Recommendation
   - Voting Systems
   - Collaborative Features

5 Conclusions
Other Approaches

- Pairwise Preference Recommendation [4].
- Constraint-Based Recommendation [12].
- ...
Recommendation Systems vs Configuration Systems

Recommendation Systems

- Items are atomic objects \((m\) is part of the input dimension)\)
- Items are rated by many users, and users rate many items

Configuration Systems

- Items are combinatorial objects \((m\) is exponential in the number of components)\)
- Items are rated by very few users, and users rate very few items
Recommendation Systems vs Configuration Systems

Recommendation Systems

- Items are atomic objects ($m$ is part of the input dimension)
- Items are rated by many users, and users rate many items

Configuration Systems

- Items are combinatorial objects ($m$ is exponential in the number of components)
- Items are rated by very few users, and users rate very few items
M. Balabanovic and Y. Shoham. 
Content-based, collaborative recommendation. 

J. S. Breese, D. Heckerman, and C. M. Kadie. 
Empirical analysis of predictive algorithms for collaborative filtering. 

Combining content-based and collaborative filters in an online newspaper. 

Learning to order things. 

Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. 

*Recommender Systems: An Introduction*. 

D. Lemire and A. Maclachlan. 
Slope one predictors for online rating-based collaborative filtering. 
In *Proceedings of the 5th SIAM International Conference on Data Mining (SDM’05)*, 2005.

P. Melville, R. J. Mooney, and R. Nagarajan. 
Content-boosted collaborative filtering for improved recommendations. 

G. Salton, A. Wong, and C. S. Yang. 
A vector space model for information retrieval. 
A. J. Smola and P. J. Barlett (editors).  
*Advances in Large Margin Classifiers*.  

P. N. Tan, M. Steinbach, and V. Kumar.  
*Introduction to Data Mining*.  
Addison Wesley, 2006.

M. Zanker, M. Jessenitschnig, and W. Schmid.  
Preference reasoning with soft constraints in constraint-based recommender systems.  