Recommender Systems as IDSS

Chad Hogg

2006-11-13
Outline

1. Recommendations
   - Problems
   - Content-Based
   - Collaborative
   - Other
   - Hybrids

2. Compromise Driven Retrieval
   - Constraint Satisfaction
   - Completeness

3. Conclusions
   - Summary
Outline

1. Recommendations
   - Problems
     - Content-Based
     - Collaborative
     - Other
     - Hybrids

2. Compromise Driven Retrieval
   - Constraint Satisfaction
   - Completeness

3. Conclusions
   - Summary
There is too much stuff for anyone to read / watch / buy / experience all of it.

We would like to spend our time and money wisely, on things of interest.

How do we know what we won’t like without trying it?
There is too much stuff for anyone to read / watch / buy / experience all of it.

We would like to spend our time and money wisely, on things of interest.

How do we know what we won’t like without trying it?
There is too much stuff for anyone to read / watch / buy / experience all of it.

We would like to spend our time and money wisely, on things of interest.

How do we know what we won’t like without trying it?
Making Recommendations

- Recommender systems make predictions of what people will enjoy.
- Typically, input is ratings of some items by a user.
- Output is a list of unrated items that may be of interest to the user.
Making Recommendations

- Recommender systems make predictions of what people will enjoy.
- Typically, input is ratings of some items by a user.
- Output is a list of unrated items that may be of interest to the user.
Making Recommendations

- Recommender systems make predictions of what people will enjoy.
- Typically, input is ratings of some items by a user.
- Output is a list of unrated items that may be of interest to the user.
Outline

1. Recommendations
   - Problems
   - Content-Based
     - Collaborative
     - Other
     - Hybrids

2. Compromise Driven Retrieval
   - Constraint Satisfaction
   - Completeness

3. Conclusions
   - Summary
Early recommender systems used information about rated items.

- Inter-item similarity may be computed based on features.
  - Each feature may be of a different type and have a local similarity metric.
- Items similar to those ranked highly by user will be recommended.
Early recommender systems used information about rated items.

Inter-item similarity may be computed based on features.
- Each feature may be of a different type and have a local similarity metric.
- Items similar to those ranked highly by user will be recommended.
Early recommender systems used information about rated items.

Inter-item similarity may be computed based on features.
- Each feature may be of a different type and have a local similarity metric.

Items similar to those ranked highly by user will be recommended.
Early recommender systems used information about rated items.

- Inter-item similarity may be computed based on features.
  - Each feature may be of a different type and have a local similarity metric.

- Items similar to those ranked highly by user will be recommended.
### Example Content Data

<table>
<thead>
<tr>
<th>Title</th>
<th>Instructor</th>
<th>Level</th>
<th>Bldg</th>
<th>Days</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sys. Software</td>
<td>Kay</td>
<td>100</td>
<td>PL</td>
<td>MWF</td>
<td>8:00</td>
</tr>
<tr>
<td>Databases</td>
<td>Korth</td>
<td>200</td>
<td>PL</td>
<td>MWF</td>
<td>14:00</td>
</tr>
<tr>
<td>Graphics</td>
<td>Huang</td>
<td>300</td>
<td>MG</td>
<td>MWF</td>
<td>9:00</td>
</tr>
<tr>
<td>Automata</td>
<td>Munoz-Avila</td>
<td>300</td>
<td>MG</td>
<td>MWF</td>
<td>14:00</td>
</tr>
<tr>
<td>Pattern Rec.</td>
<td>Baird</td>
<td>300</td>
<td>MG</td>
<td>TR</td>
<td>11:00</td>
</tr>
<tr>
<td>IDSS</td>
<td>Munoz-Avila</td>
<td>300</td>
<td>LL</td>
<td>MWF</td>
<td>9:00</td>
</tr>
</tbody>
</table>
Example Recommendation

- (Presume that courses are always taught by the same professor, in the same room and at the same time.)
- Suppose Bob has previously taken Pattern Recognition, which he hated, and Automata, which he loved.
- IDSS would probably be a good choice for Bob, because it has the same instructor, level, and days as another course he liked.
Example Recommendation

- (Presume that courses are always taught by the same professor, in the same room and at the same time.)

- Suppose Bob has previously taken Pattern Recognition, which he hated, and Automata, which he loved.

- IDSS would probably be a good choice for Bob, because it has the same instructor, level, and days as another course he liked.
(Presume that courses are always taught by the same professor, in the same room and at the same time.)

Suppose Bob has previously taken Pattern Recognition, which he hated, and Automata, which he loved.

IDSS would probably be a good choice for Bob, because it has the same instructor, level, and days as another course he liked.
Disadvantages

- Requires lots of knowledge engineering to collect data about items.
- Hard to compare items of different types.
- Some properties are difficult to capture in objective features.
  - Difficulty of assignments
  - Quality of material
- What other information can we take advantage of?
Disadvantages

- Requires lots of knowledge engineering to collect data about items.
- Hard to compare items of different types.
- Some properties are difficult to capture in objective features.
  - Difficulty of assignments
  - Quality of material
- What other information can we take advantage of?
Disadvantages

- Requires lots of knowledge engineering to collect data about items.
- Hard to compare items of different types.
- Some properties are difficult to capture in objective features.
  - Difficulty of assignments
  - Quality of material
- What other information can we take advantage of?
Disadvantages

- Requires lots of knowledge engineering to collect data about items.
- Hard to compare items of different types.
- Some properties are difficult to capture in objective features.
  - Difficulty of assignments
  - Quality of material
- What other information can we take advantage of?
Disadvantages

- Requires lots of knowledge engineering to collect data about items.
- Hard to compare items of different types.
- Some properties are difficult to capture in objective features.
  - Difficulty of assignments
  - Quality of material
- What other information can we take advantage of?
Outline

1 Recommendations
   - Problems
   - Content-Based
   - Collaborative
   - Other
   - Hybrids

2 Compromise Driven Retrieval
   - Constraint Satisfaction
   - Completeness

3 Conclusions
   - Summary
Typically, recommender systems are used by large numbers of people.
We can store their preferences as a new piece of data.
Perhaps people will like things liked by people with similar tastes.
Typically, recommender systems are used by large numbers of people.

We can store their preferences as a new piece of data.

Perhaps people will like things liked by people with similar tastes.
Collaborative Filtering

- Typically, recommender systems are used by large numbers of people.
- We can store their preferences as a new piece of data.
- Perhaps people will like things liked by people with similar tastes.
### Example Collaborative Data

<table>
<thead>
<tr>
<th>User</th>
<th>Course</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>Pattern Rec.</td>
<td>Disliked</td>
</tr>
<tr>
<td>Alice</td>
<td>Automata</td>
<td>Liked</td>
</tr>
<tr>
<td>Alice</td>
<td>IDSS</td>
<td>Disliked</td>
</tr>
<tr>
<td>Alice</td>
<td>Databases</td>
<td>Liked</td>
</tr>
<tr>
<td>Bob</td>
<td>Pattern Rec.</td>
<td>Disliked</td>
</tr>
<tr>
<td>Bob</td>
<td>Automata</td>
<td>Liked</td>
</tr>
<tr>
<td>Chris</td>
<td>Pattern Rec.</td>
<td>Liked</td>
</tr>
<tr>
<td>Chris</td>
<td>Automata</td>
<td>Liked</td>
</tr>
<tr>
<td>Chris</td>
<td>IDSS</td>
<td>Liked</td>
</tr>
</tbody>
</table>
Example Recommendation

- Bob & Alice seem to have more similar tastes than Bob & Chris do.
- Since Alice enjoyed Databases, Bob probably will also.
- With many users, we could average recommendations of a group of similar users.
Example Recommendation

- Bob & Alice seem to have more similar tastes than Bob & Chris do.
- Since Alice enjoyed Databases, Bob probably will also.
- With many users, we could average recommendations of a group of similar users.
Example Recommendation

- Bob & Alice seem to have more similar tastes than Bob & Chris do.
- Since Alice enjoyed Databases, Bob probably will also.
- With many users, we could average recommendations of a group of similar users.
Advantages

- Does not require features of items.
- Works on arbitrary classes of items.
- Far surpasses accuracy of content-based systems in most trials.
Advantages

- Does not require features of items.
- Works on arbitrary classes of items.
- Far surpasses accuracy of content-based systems in most trials.
Advantages

- Does not require features of items.
- Works on arbitrary classes of items.
- Far surpasses accuracy of content-based systems in most trials.
The “slow start” problem – requires large amount of ratings.

- New items cannot be predicted – no one has taken Graphics with Huang to rate it.
- Computing similarities in large matrices is very time-consuming.
  - High-similarity neighborhoods may be pre-computed.
  - k-Nearest Neighbors
  - Clustering
Disadvantages

- The “slow start” problem – requires large amount of ratings.
- New items cannot be predicted – no one has taken Graphics with Huang to rate it.
- Computing similarities in large matrices is very time-consuming.
  - High-similarity neighborhoods may be pre-computed.
  - k-Nearest Neighbors
  - Clustering
Disadvantages

- The “slow start” problem – requires large amount of ratings.
- New items cannot be predicted – no one has taken Graphics with Huang to rate it.
- Computing similarities in large matrices is very time-consuming.
  - High-similarity neighborhoods may be pre-computed.
  - k-Nearest Neighbors
  - Clustering
The “slow start” problem – requires large amount of ratings.

New items cannot be predicted – no one has taken Graphics with Huang to rate it.

Computing similarities in large matrices is very time-consuming.

- High-similarity neighborhoods may be pre-computed.
  - k-Nearest Neighbors
  - Clustering
Disadvantages

- The “slow start” problem – requires large amount of ratings.
- New items cannot be predicted – no one has taken Graphics with Huang to rate it.
- Computing similarities in large matrices is very time-consuming.
  - High-similarity neighborhoods may be pre-computed.
  - k-Nearest Neighbors
  - Clustering
Disadvantages

- The “slow start” problem – requires large amount of ratings.
- New items cannot be predicted – no one has taken Graphics with Huang to rate it.
- Computing similarities in large matrices is very time-consuming.
  - High-similarity neighborhoods may be pre-computed.
  - k-Nearest Neighbors
  - Clustering
Outline

1. Recommendations
   - Problems
   - Content-Based
   - Collaborative
   - Other
     - Hybrids

2. Compromise Driven Retrieval
   - Constraint Satisfaction
   - Completeness

3. Conclusions
   - Summary
Query-Based

Content and collaborative filtering based systems are good for general interest.

Sometimes users are looking for an item with specific properties.

This uses same data as content-based filtering, but input is a set of attributes.

Cases similar to problem may be retrieved as in our projects.
Query-Based

- Content and collaborative filtering based systems are good for general interest.
- Sometimes users are looking for an item with specific properties.
  - This uses same data as content-based filtering, but input is a set of attributes.
  - Cases similar to problem may be retrieved as in our projects.
Query-Based

- Content and collaborative filtering based systems are good for general interest.
- Sometimes users are looking for an item with specific properties.
- This uses same data as content-based filtering, but input is a set of attributes.
- Cases similar to problem may be retrieved as in our projects.
Query-Based

- Content and collaborative filtering based systems are good for general interest.
- Sometimes users are looking for an item with specific properties.
- This uses same data as content-based filtering, but input is a set of attributes.
- Cases similar to problem may be retrieved as in our projects.
The vast majority of recommender systems use one of the above systems.

Alternatives have been studied, including:
- Average ratings of trusted users
- Propagate ratings through a graph structure
The vast majority of recommender systems use one of the above systems.

Alternatives have been studied, including:

- Average ratings of trusted users
- Propagate ratings through a graph structure
The vast majority of recommender systems use one of the above systems.

Alternatives have been studied, including:
- Average ratings of trusted users
- Propagate ratings through a graph structure
The vast majority of recommender systems use one of the above systems.

Alternatives have been studied, including:

- Average ratings of trusted users
- Propagate ratings through a graph structure
Outline

1. Recommendations
   - Problems
   - Content-Based
   - Collaborative
   - Other
   - Hybrids

2. Compromise Driven Retrieval
   - Constraint Satisfaction
   - Completeness

3. Conclusions
   - Summary
Hybrid Approaches

- A combination of approaches may mitigate the disadvantages of each.
- Methods might be combined in the following ways:
  - A weighted combination returns a weighted sum of the results of submethods.
  - A switching system tries to use the method that is most effective in each situation.
  - A cascade uses different methods at different levels of granularity.
Hybrid Approaches

A combination of approaches may mitigate the disadvantages of each.

Methods might be combined in the following ways:

- A weighted combination returns a weighted sum of the results of submethods.
- A switching system tries to use the method that is most effective in each situation.
- A cascade uses different methods at different levels of granularity.
Hybrid Approaches

- A combination of approaches may mitigate the disadvantages of each.

- Methods might be combined in the following ways:
  - A weighted combination returns a weighted sum of the results of submethods.
  - A switching system tries to use the method that is most effective in each situation.
  - A cascade uses different methods at different levels of granularity.
A combination of approaches may mitigate the disadvantages of each.

Methods might be combined in the following ways:

- A weighted combination returns a weighted sum of the results of submethods.
- A switching system tries to use the method that is most effective in each situation.
- A cascade uses different methods at different levels of granularity.
A combination of approaches may mitigate the disadvantages of each.

Methods might be combined in the following ways:

- A weighted combination returns a weighted sum of the results of submethods.
- A switching system tries to use the method that is most effective in each situation.
- A cascade uses different methods at different levels of granularity.
Recommendations

1. Recommendations
   - Problems
   - Content-Based
   - Collaborative
   - Other
   - Hybrids

2. Compromise Driven Retrieval
   - Constraint Satisfaction
   - Completeness

3. Conclusions
   - Summary
Consider the case of making a recommendation from a query.

Attribute-value pairs in the query are constraints on the cases to be recommended.

Ideally a case will be found that satisfies all constraints, but this is unlikely.

Some constraints may be more important than others – if you have a job on TH, you will only consider courses that meet MWF.
Consider the case of making a recommendation from a query.

Attribute-value pairs in the query are constraints on the cases to be recommended.

Ideally a case will be found that satisfies all constraints, but this is unlikely.

Some constraints may be more important than others – if you have a job on TH, you will only consider courses that meet MWF.
Consider the case of making a recommendation from a query.

Attribute-value pairs in the query are constraints on the cases to be recommended.

Ideally a case will be found that satisfies all constraints, but this is unlikely.

Some constraints may be more important than others – if you have a job on TH, you will only consider courses that meet MWF.
Constraints

- Consider the case of making a recommendation from a query.
- Attribute-value pairs in the query are constraints on the cases to be recommended.
- Ideally a case will be found that satisfies all constraints, but this is unlikely.
- Some constraints may be more important than others – if you have a job on TH, you will only consider courses that meet MWF.
Outline

1. Recommendations
   - Problems
   - Content-Based
   - Collaborative
   - Other
   - Hybrids

2. Compromise Driven Retrieval
   - Constraint Satisfaction
   - Completeness

3. Conclusions
   - Summary
Completeness

- The system does not know which constraints are more important.
- For completeness, try to return a set of cases that satisfy every possible maximal subset of constraints.
- k-NN does not solve this because chosen cases may be very similar to each other.
The system does not know which constraints are more important.

For completeness, try to return a set of cases that satisfy every possible maximal subset of constraints.

k-NN does not solve this because chosen cases may be very similar to each other.
Completeness

- The system does not know which constraints are more important.
- For completeness, try to return a set of cases that satisfy every possible maximal subset of constraints.
- k-NN does not solve this because chosen cases may be very similar to each other.
Compromise-Driven Retrieval

- Add most similar candidate M to list.
- Remove all cases that do not satisfy a constraint not satisfied by M from candidates.
- Repeat until no candidates remain.
- Like a cascade, uses similarity first and then constraint satisfaction.
- Provides a complete set.
Compromise-Driven Retrieval

- Add most similar candidate $M$ to list.
- Remove all cases that do not satisfy a constraint not satisfied by $M$ from candidates.
- Repeat until no candidates remain.
- Like a cascade, uses similarity first and then constraint satisfaction.
- Provides a complete set.
Compromise-Driven Retrieval

- Add most similar candidate M to list.
- Remove all cases that do not satisfy a constraint not satisfied by M from candidates.
- Repeat until no candidates remain.
- Like a cascade, uses similarity first and then constraint satisfaction.
- Provides a complete set.
Compromise-Driven Retrieval

- Add most similar candidate M to list.
- Remove all cases that do not satisfy a constraint not satisfied by M from candidates.
- Repeat until no candidates remain.
- Like a cascade, uses similarity first and then constraint satisfaction.
- Provides a complete set.
Compromise-Driven Retrieval

- Add most similar candidate M to list.
- Remove all cases that do not satisfy a constraint not satisfied by M from candidates.
- Repeat until no candidates remain.
- Like a cascade, uses similarity first and then constraint satisfaction.
- Provides a complete set.
Outline

1. Recommendations
   - Problems
   - Content-Based
   - Collaborative
   - Other
   - Hybrids

2. Compromise Driven Retrieval
   - Constraint Satisfaction
   - Completeness

3. Conclusions
   - Summary
Recommender systems support decision making by suggesting a few of many alternatives.

Recommender systems may use data about items, relations between users, and other information.

Although this technology is used by Amazon and many others, it remains an interesting research problem.
Recommender systems support decision making by suggesting a few of many alternatives.

Recommender systems may use data about items, relations between users, and other information.

Although this technology is used by Amazon and many others, it remains an interesting research problem.
Recommender systems support decision making by suggesting a few of many alternatives.

Recommender systems may use data about items, relations between users, and other information.

Although this technology is used by Amazon and many others, it remains an interesting research problem.
Thank You