Recommendaions in Context

Francesco Ricci
Free University of Bolzano/Bozen
fricci@unibz.it

What movie should I see?

<table>
<thead>
<tr>
<th>Rank</th>
<th>Title</th>
<th>Votes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The Godfather (1972)</td>
<td>888,130</td>
</tr>
<tr>
<td>2</td>
<td>The Godfather: Part II (1974)</td>
<td>440,096</td>
</tr>
<tr>
<td>3</td>
<td>The Godfather, Part III (1990)</td>
<td>94,772</td>
</tr>
<tr>
<td>4</td>
<td>The Godfather, Part IV (1997)</td>
<td>99,375</td>
</tr>
<tr>
<td>5</td>
<td>The Godfather (1972)</td>
<td>62,225</td>
</tr>
<tr>
<td>6</td>
<td>The Godfather Part II (1974)</td>
<td>62,000</td>
</tr>
<tr>
<td>7</td>
<td>Schindler's List (1993)</td>
<td>54,201</td>
</tr>
<tr>
<td>8</td>
<td>Star Wars, Episode IV: A New Hope (1977)</td>
<td>142,568</td>
</tr>
<tr>
<td>9</td>
<td>Shawshank Redemption (1994)</td>
<td>63,406</td>
</tr>
<tr>
<td>10</td>
<td>Pulp Fiction (1994)</td>
<td>171,174</td>
</tr>
<tr>
<td>11</td>
<td>The Shawshank Redemption (1994)</td>
<td>186,312</td>
</tr>
<tr>
<td>12</td>
<td>Goodfellas (1990)</td>
<td>88,475</td>
</tr>
<tr>
<td>13</td>
<td>Pulp Fiction (1994)</td>
<td>82,265</td>
</tr>
<tr>
<td>14</td>
<td>The Godfather (1972)</td>
<td>62,265</td>
</tr>
<tr>
<td>16</td>
<td>The Godfather, Part III (1990)</td>
<td>123,214</td>
</tr>
<tr>
<td>17</td>
<td>Pulp Fiction (1994)</td>
<td>114,397</td>
</tr>
<tr>
<td>18</td>
<td>Schindler’s List (1993)</td>
<td>30,546</td>
</tr>
<tr>
<td>19</td>
<td>Schindler’s List (1993)</td>
<td>80,720</td>
</tr>
<tr>
<td>20</td>
<td>The Godfather, Part IV (1997)</td>
<td>77,367</td>
</tr>
<tr>
<td>21</td>
<td>The Godfather, Part IV (1997)</td>
<td>164,360</td>
</tr>
<tr>
<td>22</td>
<td>Schindler’s List (1993)</td>
<td>73,689</td>
</tr>
<tr>
<td>23</td>
<td>The Godfather Part II (1974)</td>
<td>86,742</td>
</tr>
<tr>
<td>24</td>
<td>Goodfellas (1990)</td>
<td>88,321</td>
</tr>
</tbody>
</table>
What book should I buy?

<table>
<thead>
<tr>
<th>FICTION</th>
<th>NONFICTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fiction &amp; Literature</td>
<td></td>
</tr>
<tr>
<td>Graphic Novels</td>
<td></td>
</tr>
<tr>
<td>Horror</td>
<td></td>
</tr>
<tr>
<td>Mystery</td>
<td></td>
</tr>
<tr>
<td>Mystery &amp; Crime</td>
<td></td>
</tr>
<tr>
<td>Poetry</td>
<td></td>
</tr>
<tr>
<td>Romance</td>
<td></td>
</tr>
<tr>
<td>Science Fiction &amp; Fantasy</td>
<td></td>
</tr>
<tr>
<td>Science Fiction &amp; Fantasy</td>
<td></td>
</tr>
<tr>
<td>Science Fiction &amp; Fantasy</td>
<td></td>
</tr>
<tr>
<td>Self-Help</td>
<td></td>
</tr>
<tr>
<td>Self-Help</td>
<td></td>
</tr>
<tr>
<td>Sports</td>
<td></td>
</tr>
<tr>
<td>Sports</td>
<td></td>
</tr>
<tr>
<td>Travel</td>
<td></td>
</tr>
<tr>
<td>Travel</td>
<td></td>
</tr>
<tr>
<td>Women's Fiction</td>
<td></td>
</tr>
<tr>
<td>Women's Fiction</td>
<td></td>
</tr>
</tbody>
</table>

What news should I read?

**Ernesto makes landfall in North Carolina**

WILMINGTON, N.C. - Tropical Storm Ernesto made landfall in the northeastern Carolinas early today, bringing winds and heavy rains in its wake after a quick track across the U.S. east coast.

**MORE STORIES**

- [Hurricane trackers track Ernesto's path](link)
- [Photos: Ernesto's impact on the Carolinas](link)

**WATCH & SHARE**

- [Watch the latest weather updates](link)
- [Share the news on social media](link)
What paper should I read?

What travel should I do?

- I would like to escape from this ugly and tedious work life and relax for two weeks in a sunny place. I am fed up with these crowded and noisy places... just the sand and the sea... and some "adventure".

- I would like to bring my wife and my children on a holiday... it should not be too expensive. I prefer mountainous places... not to far from home. Children parks, easy paths and good cuisine are a must.

- I want to experience the contact with a completely different culture. I would like to be fascinated by the people and learn to look at my life in a totally different way.
What problems we’d like to be solved by recommender systems
What has been proposed – rating prediction
What does not work in this approach – just a bit!
Contextualization and personalization
Examples of contextualization
Learning to contextualize: process adaptation

Original Definition of RS

- In everyday life we rely on recommendations from other people either by word of mouth, recommendation letters, movie and book reviews printed in newspapers ...
- In a typical recommender system people provide recommendations as inputs, which the system then aggregates and directs to appropriate recipients
  - Aggregation of recommendations
  - Match the recommendations with those searching for recommendations

[Resnick and Varian, 1997]
Recommender Systems

- A **recommender system** helps to make choices without sufficient personal experience of the alternatives
  - To **suggest products** to their customers
  - To provide consumers with **information to help them decide** which products to purchase
- They are based on a number of **technologies**: information filtering, machine learning, adaptive and personalized system, user modeling, ...
- Not clear separation from IR – [Burke, 2002] claims that is the “individualized” and “interesting and useful” features that make the difference.

"Core" Recommendation Techniques

**U** is a set of users
**I** is a set of items/products

<table>
<thead>
<tr>
<th>Technique</th>
<th>Background</th>
<th>Input</th>
<th>Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collaborative</td>
<td>Ratings from <strong>U</strong> of items in <strong>I</strong>.</td>
<td>Ratings from <strong>u</strong> of items in <strong>I</strong>.</td>
<td>Identify users in <strong>U</strong> similar to <strong>u</strong>, and extrapolate from their ratings of <strong>I</strong>.</td>
</tr>
<tr>
<td>Content-based</td>
<td>Features of items in <strong>I</strong>.</td>
<td><strong>u</strong>’s ratings of items in <strong>I</strong>.</td>
<td>Generate a classifier that fits <strong>u</strong>’s rating behavior and use it on <strong>I</strong>.</td>
</tr>
<tr>
<td>Demographic</td>
<td>Demographic information about <strong>U</strong> and their ratings of items in <strong>I</strong>.</td>
<td>Demographic information about <strong>u</strong>.</td>
<td>Identify users that are demographically similar to <strong>u</strong>, and extrapolate from their ratings of <strong>I</strong>.</td>
</tr>
<tr>
<td>Utility-based</td>
<td>Features of items in <strong>I</strong>.</td>
<td>A utility function over items in <strong>I</strong> that describes <strong>u</strong>’s preferences.</td>
<td>Apply the function to the items and determine <strong>I</strong>’s rank.</td>
</tr>
<tr>
<td>Knowledge-based</td>
<td>Features of items in <strong>I</strong>. Knowledge of how these items meet a user’s needs.</td>
<td>A description of <strong>u</strong>’s needs or interests.</td>
<td>Infer a match between <strong>I</strong> and <strong>u</strong>’s need.</td>
</tr>
</tbody>
</table>

[Burke, 2002]
Collaborative-Based Filtering

So far you have rated 0 movies.
MovieLens needs at least 15 ratings from you to generate predictions for you.
Please rate as many movies as you can from the list below.

Your Rating Movie Information

★★★★ 3.0 stars Austin Powers: International Man of Mystery (1997) Action, Adventure, Comedy
★★★★★ 4.0 stars Contact (1997) Drama, Sci-Fi
★★★ Not seen Eraser (1996) Action, Drama, Thriller
★★★ Not seen Moulin Rouge (1992) Action, Crime, Western
★★★★★ 4.5 stars Plato's Retreat, The (1993) Drama, Romance
★★★★★ 5.0 stars Asia (1998) Drama, Romance

To get a new set of movies click the next button.
Collaborative-Based Filtering

- The collaborative based filtering recommendation techniques proceed in these steps:

  1. For a target/active user (the user to whom a recommendation has to be produced) the set of his ratings is identified
  2. The users more similar to the target/active user (according to a similarity function) are identified (neighbor formation)
  3. The products bought by these similar users are identified
  4. For each one of these products a prediction - of the rating that would be given by the target user to the product - is generated
  5. Based on this predicted rating a set of top N products are recommended.

Content-Based Filtering: Syskill & Webert

The user indicated interest in

System Prediction

The user indicated no interest in

The user indicated interest in
Content-Based Recommender

- It is mainly used for recommending **text-based products** (web pages, usenet news messages,)
- The items to recommend are “described” by their associated **features** (e.g. keywords)
- The **User Model** can be structured in a “similar” way as the content: for instance the features/keywords more likely to occur in the preferred documents (lazy approach)
  - Then, text documents can be recommended based on a comparison between their content (words appearing in the text) and a user model (a set of preferred words)
- The user model can also be a **classifier** based on whatever technique (Neural Networks, Naïve Bayes, C4.5,)

Demographic-based personalization
Utility-based

Utility methods

- A utility function is a map from a state onto a real number, which describes the associated degree of happiness.
- Can build a long term utility function but more often the systems using such an approach try to acquire a short term utility function.
- They must acquire the user utility function, or the parameters defining such a function.
Knowledge-Based Recommender System

- Entree is a case-based restaurant recommender system – it finds restaurants:
  1. in a new city similar to restaurants the user knows and likes
  2. or those matching some user goals (case features).

Partial Match

- In general, only a subset of the preferences will be matched in the recommended restaurant.
A Simplified Model of Recommendation

1. Two types of entities: Users and Items
2. A background knowledge:
   - A set of ratings: a map $R: \text{Users} \times \text{Items} \to [0,1] \cup \{?\}
   - A set of "features" of the Users and/or Items
3. A method for eliminating all or part of the '?' values for some (user, item) pairs – substituting '?' with the true values
4. A method for selecting the items to recommend
   - Recommend to $u$ the item $i^*$ such that:
     - $i^* = \arg \max_{i \in \text{Items}} \{R(u,i)\}$

[Adomavicius et al., 2005]

A Bidimensional Model

user

<table>
<thead>
<tr>
<th>item</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
</tr>
</tbody>
</table>

ratings

<table>
<thead>
<tr>
<th>User features</th>
</tr>
</thead>
</table>

Product features
Collaborative Filtering

User

Item

Ratings

Ex: 4 out of 5

Content Based Filtering (Classical)

User

Item

Ratings

Uses only the ratings of the target (active) user

Product features
Knowledge-Based

Rich user and product profiles and complex relationships between the two models

What these techniques forget

- What the user is doing when asking for a recommendation
- What the user really wants (e.g., improve his knowledge or really buy a product)
- Is the user alone or with other fellows?
- Are there many products to choose or only few?
- Is the word economy growing or falling?
- ...
Contextual Computing

- **Contextual computing** refers to the enhancement of a user’s interactions by understanding the user, the context, and the applications and information being used, typically across a wide set of user goals.

- Actively adapting the computational environment - for each and every user - **at each point of computation**

- Contextual computing approach focuses on understanding the **information consumption patterns** of each user.

- Contextual computing focuses on the **process** not only on the output of the search process.

  [Pitkow et al., 2002]

Contextualization and Individualization

- **Contextualization**: the interrelated conditions that occur within an activity
  - It includes factors like the nature of information available, the information currently being examined, the applications in use, when, and so on

- **Individualization**: the totality of characteristics that distinguishes an individual
  - It encompasses elements like the user’s goals, prior and tacit knowledge, past information-seeking behaviors, among others

- **Personalization must focus on the combination** of the user and the context within the application of search.

  [Pitkow et al., 2002]
Factors influencing Holiday Decision

Internal to the tourist
- Personality
- Disposable Income
- Health
- Family commitments
- Past experience
- Works commitments
- Hobbies and interests
- Knowledge of potential holidays
- Lifestyle
- Attitudes, opinions and perceptions

External to the tourist
- Availability of products
- Advice of travel agents
- Information obtained from tourism organization and media
- Word-of-mouth recommendations
- Political restrictions: visa, terrorism,
- Health problems
- Special promotion and offers
- Climate

Decision

Context

Preferences
Ranking is computed by considering more recommendable those products/services that were selected in other travel plans with similar contextual features.

When an Attribute-Based Search May Fail

- The user is seeking suggestions, hints, and inspiration rather than options that must optimize a collection of decision criteria.
- The user does not have knowledge of the tourism jargon that is typically used in the description of travel products and services.
- The user can be intimidated and even not able to use advanced search tools based on queries – conjunction of constraints.
- The preferences are not defined before the search process but are “constructed” while learning about available products [Bettman et al., 1998].

*All of these issues point to different contextual conditions.*
Seeking for Inspiration – Preference-based Feedback

[http://dietorecs.itc.it]

[Ricci et al., 2005b]

Seeking for Inspiration

Seeking for Inspiration

seed case

I-Like(c_i)

user

(c_1, c_2, c_3, c_4, c_5, c_6)

Retrieval

Selection

Case Base

Browsed Cases

Presentation

Explanation
www.visiteurope.com

- Major European Tourism Destination Portal of the European Travel Commission (ETC)
- 34 National Tourism Organizations
- Project started 2004
- Consortium: EC3, TIScover, ITC-irst, Siemens, Lixto
- On line since April 06
- 500.000 page views/month
- 100.000 visitors/month

Context-dependent travel planning

- There is no single best strategy for bundling – a strategy must be influenced by:

1. **The travel plan – the goal of the process:** it could be many “different” objects: package, flight&drive, itinerary, just an event

2. **Travelers – the user:** they have different motivations, goals, preferences, style of traveling, ...

3. **Information search and package bundling preferences:** user may need to consults parents, read reviews, compare offers, think about, ...
Multiple bundling strategies

- **Single iterative product selection from catalogues:** manual search of products/services with interactive query support (dealing with query failures)
- **Recommendation by proposal:** after products suggestion the user can critique these products and the critiques are incorporated in a new products' list
- **Selection of a complete package:** user search from a catalogue of fixed or simple customizable packages
- **Completion of a partial package:** a partial solution, e.g., product selected by the traveller (e.g. an event or an accommodation) is completed by the system either by fulfilling a still open goal or by simply proposing products types that can be found in similar travel plans.

Scenarios

- One strategy may better fit the current context than another
- Offer the user the possibility to choose one function and follow her preferred strategy
- Provide good metaphors to let her understand what she will get
For instance, this tries to dispatch users to strategies according to a user self evaluation and identification to a behavioral profile.

Observations

- It is difficult for a user to understand what is the best “entrance”
- It is even more difficult to understand what is the system functions that might better suit her and how to better use them
- The user needs to explore the system and learn:
  - The precise behavior of system functions
  - The advantages of one function over the others
  - The possibility to combine and integrate functions to achieve a goal (e.g. bundle a set of services)
- Taks support is required – but how to learn to correctly support the user in the task?
Sequences of recommendation functions

<table>
<thead>
<tr>
<th>1</th>
<th>Incremental single item selection</th>
<th>Recommendation by proposal</th>
<th>Package customization</th>
<th>Incremental single item selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Package selection</td>
<td>Incremental single item selection</td>
<td>Package completion</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Dynamic bundling</td>
<td>Package customization</td>
<td>Time</td>
<td></td>
</tr>
</tbody>
</table>

- The possible sequences of recommendation functions can be too large to be explored by a user (if she has to learn a best system usage)
- Some "paths" could perform rather poorly in general
- Some "paths" may suit better a single user
- Can we rank paths?

... the word is one click away

<table>
<thead>
<tr>
<th>1</th>
<th>Incremental single item selection</th>
<th>Search Google</th>
<th>Package customization</th>
<th>BUY</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Package selection</td>
<td>Visit Competitor Site</td>
<td>NOT BUY</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Dynamic bundling</td>
<td>Package customization</td>
<td>ABORT SESSION</td>
<td>Time</td>
</tr>
</tbody>
</table>

- We should consider that the options are not limited to that included in the web site
- Some visit to external sites may be beneficial and keep the customer
- Some exits may be fatal
Learning Best Recommendation Strategy

- If we want to adapt the recommendation process to the context then the system must be able to decide on a number of open questions:
  - What packaging method/strategy should be suggested?
  - What type of information should be asked?
  - In which order the system must present all the information required?
  - When it is better to actively support the user with directions?
  - When the system should push a recommendation?
  - When the system would be better listening to the user needs?

- we need to LEARN the system behavior from the interaction logs rather than the products to recommend!
Augmenting a IS with a Recommender

- The information system provides information seeking and recommendation functions
- The recommender agent observes: the user, the interaction and IS
- The recommender agent interact with the user when believes there is a need for help
- The recommender agents suggests to the user actions to be performed on the IS

Multiple Output for a User Action

- The user is in $P_0$ and ask for a list of products ($f_1$)
- The system has now two options:
  - Show the full list (as before)
  - Recommend a product, i.e., bring the user in $P_3$

- $f_1$ asks for a lists of products
- $f_2$ asks to sort the list by an attribute
- $f_3$ asks a recommendation
- $f_4$ asks to buy a product
- $f_5$ asks to compare two products

How to learn to take this decision?

User Actions
**Agent/Environment**

- **Agent** = recommender agent
  - Performs actions and perceives reward and new state
- **Environment** = the information system and the user
  - Determine the state transition and returns to the user the next state information and the reward

---

**A Markov Decision Process Model**

- **State**: \( S \subseteq I \times R \times U \)
  - \( I \) is a model of the interaction (e.g. the current and last visited page, the number of page seen, etc.)
  - \( R \) is a model of the recommendation agent (e.g. the number of times it has pushed a suggestion, the type of the last action suggestion, etc.)
  - \( U \) is a model of the user (preferences, emotions, group composition, etc.)
- A set \( \mathcal{A} \) of actions available to the recommender (suggestion for actions that the user may or should do, e.g., "look at last minute offers!")
- A **transition model** \( T(s, a, s') \) that gives the probability to make a transition from state \( s \) to state \( s' \) when the recommendation agent makes the action \( a \).
- A **reward function** \( R(s, a) \) that assign a reward value, i.e., a real number, to the recommendation agent, for each state \( s \) and action \( a \) taken in state \( s \) of the interaction. The goal state (check out) has a positive reward, whereas each, intermediary state has a negative reward. Reward should be greater if the user "followed" the recommender suggestion.
- GOAL: compute a policy, i.e., a map from states to actions that when applied will maximize the total expected discounted reward

\[
R_T = \sum_{t=0}^{\infty} \gamma^t R_t
\]

where \( R_t \) is the reward at time \( t \) and \( 0 < \gamma < 1 \)
Example: Query Tightening

Currently when a query returns too many options (greater than 50) the system suggests tightening.

Is it a good strategy? Can we improve the strategy?

- Goal: reduce the interaction length (reward is negative in each state unless the goal state is reached)
- How: have a dynamic strategy – the system decides (state by state) if it is better to suggest tightening or is better to show all the results

What are the relevant variables that should describe the state?

[Ricci and Mahmood, 2006]
Pages and user actions

- Pages
  - Start = s
  - Query form = QF
  - Tightening = T
  - Result set = R
  - End = G

- User actions
  - Start interaction = go
  - Modify query = modq
  - Execute query = execq
  - Accept tightening = acct
  - Reject Tightening show all = rejt
  - Add to cart = add

Query Form

In this page the user can only fill the form and execute the query (search)
Tightening

In this page the user can either
1. Accept tightening and modify the query with respect to one of the features suggested ("Category", "Car park", or "TV") - acct
2. Reject tightening and execute the original query ("Get all results") - rej
3. Modify the query on other features - modq

Result set

Here the user can
1. either add an item to the travel bag (cart) - add
2. Or modify the query an execute it - modq
Policy

- The transition probabilities - $T(s, a, s')$ - model the user stochastic reply to the system actions
- A policy is a function that assigns to each state an action
  $$\pi : S \rightarrow A$$
- If the system adopts that policy the value of a state is given by
  $$V^\pi(s) = R(s, \pi(s)) + \gamma \sum_{s' \in S} T(s, \pi(s), s') V^\pi(s'), \forall s \in S$$

Policy Learning

- If $T(s, a, s')$ are known then the optimal behavior of the recommendation agent would be a policy
  $$\pi : S \rightarrow A$$
- such that, for each initial state $s$, if the agent behaves accordingly to the policy then the expected total reward is maximum
  $$V^*(s) = \max_{\pi} E \left( \sum_{t=0}^{\infty} \gamma^t R_t \right)$$
Results

- When the cost of each interaction is **small** it is better to execute the query, do not propose tightening, and let the user to modify the query autonomously
- If the cost of interaction become **larger** than it is better to suggest the tightening.

<table>
<thead>
<tr>
<th>cost</th>
<th>(s, s)</th>
<th>(m, s)</th>
<th>(m, m)</th>
<th>(l, s)</th>
<th>(l, m)</th>
<th>(l, l)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Init Pol.</td>
<td>exec</td>
<td>exec</td>
<td>exec</td>
<td>sugg</td>
<td>sugg</td>
<td>sugg</td>
</tr>
<tr>
<td>−0.01</td>
<td>exec</td>
<td>exec</td>
<td>exec</td>
<td>exec</td>
<td>exec</td>
<td>exec</td>
</tr>
<tr>
<td>−0.02</td>
<td>exec</td>
<td>exec</td>
<td>exec</td>
<td>exec</td>
<td>exec</td>
<td>exec</td>
</tr>
<tr>
<td>−0.04</td>
<td>exec</td>
<td>exec</td>
<td>sugg</td>
<td>exec</td>
<td>sugg</td>
<td>sugg</td>
</tr>
<tr>
<td>−0.08</td>
<td>exec</td>
<td>sugg</td>
<td>sugg</td>
<td>sugg</td>
<td>sugg</td>
<td>sugg</td>
</tr>
<tr>
<td>−0.12</td>
<td>sugg</td>
<td>sugg</td>
<td>sugg</td>
<td>sugg</td>
<td>sugg</td>
<td>sugg</td>
</tr>
</tbody>
</table>

\( s: \text{small} \quad m: \text{medium} \quad l: \text{large} \quad \text{sugg: suggest} \quad \text{exec: execute} \)

Initial policy

(m,s) means current size m and expected new size (after tightening) small

It seems that they can be served with the same strategy and products

Different serving strategies and products for “lazy” and “eager” users
Conclusion

- Recommender systems have offered “complex” techniques to predict user ratings under “simple” contextual conditions.

- We need to devote more thoughts about what makes the recommendation process more useful for the user – considering that both the user and the system live in a larger context.

- We should explore learning technologies that adapts the recommendation process and not only the product to be recommended.

fricci@unibz.it