Computing Useful Recommendations
... still requires knowledge

Francesco Ricci
Information and Database Systems Engineering
Free University of Bozen-Bolzano
fricci@unibz.it
Content

- Preferences and choices
- Recommender systems – the *classical view*
- How can we compute useful recommendations?
- Better guessing the user utility function vs optimizing predictive precision
- A novel approach: from choices to utility
  - Groups of users with **similar behaviours may better reveal the hidden utility of choices**
- *Inaccurate* recommendation for a user that deviates from the predicted behaviour may be a *good* recommendation.
What we like may not be what we choose

... and a Recommender System should suggest even better choices!
Preferences and Choices

- **Preferences** (utility) are hard to model and acquire
  - RSs use mostly ratings (utility), very few asks users to rank or pairwise compare
- **Choices** are often easier to observe but they are noisier: good and bad ones are mixed (clicks, purchases, replies, bookmarks)
- Observed choices should be used to predict preferences but in RSs are mostly used to predict other choices

- A useful recommendation is for a novel item that the user will:
  1. Accept to choose (persuasion – expected utility)
  2. Like to consume (experienced utility)
Knowing your goals / preferences

- "what do I want?" – depends on how a choice will make us feel

- **Future**: what you expect an experience will make you feel is called **expected utility**

- **Present**: The way an item (movie, travel, car) makes you feel in the moment is called **experienced utility**

- **Past**: Once you had an experience (e.g. a movie), future choice will be based on what you remember about that: **remembered utility**.
Classical Recommendation Model

Three types of entities: **Users, Items and Contexts**

1. A **background knowledge**:
   - A set of **ratings – preferences**
     - $r: \text{Users} \times \text{Items} \times \text{Contexts} \rightarrow \{1, 2, 3, 4, 5\}$
     - A set of “features” of the Users, Items and Contexts

2. A method for **predicting** the function $r$ where it is unknown:
   - $r^*(u, i, c) = \text{Average ratings } r(u', i, c'): \text{users } u' \text{ are similar to } u \text{ and context } c' \text{ is similar to } c$

3. A method for **selecting** the items to recommend (choice):
   - In context $c$ recommend to $u$ the item $i^*$ **with the largest predicted rating** $r^*(u, i, c)$
Context Aware RS Algorithms

- Reduction-based Approach, 2005
- Exact and Generalized Pre Filtering, 2009
- Item Splitting, 2009
- Tensor Factorization, 2010
- User Splitting, 2011
- Context-aware Matrix Factorization, 2011
- Factorization Machines, 2011
- Differential Context Relaxation, 2012
- Differential Context Weighting, 2013
- UI splitting, 2014
- Similarity-Based Context Modelling, 2015
- Convolutional Matrix factorization, 2016
- Contextual bandit, 2018

The research has focussed mostly on context-dependent preference prediction rather than choice modelling and support.
We should understand how users make choices in context in order to understand how a RS can better support them in identifying items that they will happily choose and not regret.
Travel Choice Criteria

- When is cost effective
- When is liked by people that likes what we like
- When is good for the full traveling group
- When we did not yet think about that
- When is not what we did last year
- When it has the features that we usually like
- When it has some impressive features
- When it is much better than other options
- When it is similar to what we did previous years
- When the weather will be great.
Music Choice Criteria

- When it is liked by people that likes what we like
- When it has the features that we usually like
- When it has some impressive features
- When we did not yet think to play it again
- When it is not what we played a few minutes ago
- When it is good for the full party group
- When it is played by the musicians that we like
- When it has the right mood
- When it is good for running
- When it is played with care and cleverness.
Do you still believe that by simply mining a data set of users’ ratings (or choices) one can generate useful travel/music recommendations?

We need to **structure** the **knowledge** that can be derived from the data!

We need to better understand the **current** user’s **relevant context and preference criteria**!
Marschall McLuhan

“The medium is the message”
Recommendation Lists

List 1
- Uffizi Galleries
- Michelangelo Square
- Duomo

List 2
- San Marco Museum
- S. Croce Square
- Santa Maria Novella

TripAdvisor images
Recommendation Lists

List 1
- Uffizi Galleries
- Michelangelo Square
- Duomo

List 2
- San Marco Museum
- S. Croce Square
- Santa Maria Novella

My selection of images
Points of Interest

- A number of features and contextual factors influence the **pre-visit** evaluation of a POI – expected utility
  
  - **Traveller’s knowledge of the place**
  - What she has already visited and when
  - Pictorial representation
  - Distinguished features
  - Travel party
  - Previous knowledge/usage of the app/recsys (the medium)
  - **Popularity, fashionableness, trendiness, fame, prominence, prestige, reputation, visibility, rank.**
Expected vs. Experienced Utility

- Should the system optimize expected or experienced utility?
- Hence, should it use choice data or ratings/judgements data?

**Expected Utility**
- It matches the **user goals** at decision time
  - It is subject to **bias** of user’s judgement
- It is derived from an **unbiased sample** of observations
  - It is derived from observations without **meaning**.

**Experienced Utility**
- It is an **explicit** user assessment
  - It is **incomplete** (data)
- It depends on the consequence of choice (**context** is used)
  - It is often derived from **remembered** utility.
Understanding and Influencing Users

- We do not know which criteria are influencing the expected utility estimation of the user and their choices.

- The user may make wrong estimations: *expected utility may be different from what they will actually experience*.
  - Novel items are hard to assess for their value.

- *We must identify good experiences (high experienced utility) to suggest by observing user previous behaviour*.

- ... and convince the user that this is a good choice (raise the expected utility).
Grouping People

- We have recently tried to learn/predict the (experienced) utility function of users with a technique that make use of groups and inverse reinforcement learning

- **Group** and **model travellers** with observable similar behaviour and optimize the recommendations for them – not purely individual recommendations.

Behaviour and Recommendation

- **Behaviour learning** and **recommendation** should be decoupled.

- An **exact behavioural model**, e.g., what points of interest a user is likely to visit may produce **not novel and uninteresting recommendations**.

- Recommendation should **optimize the criteria** that ultimately have determined the observed behaviour.
Behavioral Model Learning

- Learning user behaviour, but suggest to deviate from the usual behaviour

  - The user is predicted to take a coffee at 8:00 at Walter Bar
    - The system suggests to get coffee at Rosy Bar – it is cheaper and better

We must understand that the user likes good Italian and cheap coffee – not that he likes to go to Walter Bar at 8:00!
Grouping Travellers
Clustering Users’ Visit Trajectories

- One visit to Florence, *taking pictures of*:
  - Pitti Palace; Boboli Garden; Uffizi Museum

- Extract important keywords and combine them into a document visit

- Cluster visit documents

- Each cluster models a group of similar behaviours
5 Clusters in Florence

<table>
<thead>
<tr>
<th>Term</th>
<th>Cluster A</th>
<th>Cluster B</th>
<th>Cluster C</th>
<th>Cluster D</th>
<th>Cluster E</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>morning</td>
<td>hot</td>
<td>cloudy</td>
<td>warm</td>
<td>freezing</td>
</tr>
<tr>
<td>2</td>
<td>cold</td>
<td>afternoon</td>
<td>cold</td>
<td>cloudy</td>
<td>cloudy</td>
</tr>
<tr>
<td>3</td>
<td>square</td>
<td>16th century</td>
<td>church</td>
<td>14th century</td>
<td>afternoon</td>
</tr>
<tr>
<td>4</td>
<td>palace</td>
<td>palace</td>
<td>square</td>
<td>church</td>
<td>14th century</td>
</tr>
<tr>
<td>5</td>
<td>15th century</td>
<td>church</td>
<td>13th century</td>
<td>square</td>
<td>palace</td>
</tr>
<tr>
<td>6</td>
<td>13th century</td>
<td>square</td>
<td>palace</td>
<td>building</td>
<td>building</td>
</tr>
<tr>
<td>7</td>
<td>church</td>
<td>19th century</td>
<td>rain</td>
<td>palace</td>
<td>13th century</td>
</tr>
<tr>
<td>8</td>
<td>night</td>
<td>13th century</td>
<td>museum</td>
<td>bridge</td>
<td>church</td>
</tr>
<tr>
<td>9</td>
<td>Dante</td>
<td>museum</td>
<td>Brunelleschi</td>
<td>13th century</td>
<td>Foggini</td>
</tr>
<tr>
<td>10</td>
<td>10th century</td>
<td>Brunelleschi</td>
<td>Tadda</td>
<td>19th century</td>
<td>19th century</td>
</tr>
</tbody>
</table>

1663 geo-localized temporally ordered trajectories of users’ POI-visits (photos), recorded via GPS sensors in the historic centre of Florence (Italy)
Inverse Reinforcement Learning

- **Assumption:** the reward obtained by visiting a POI is determined by the POI’s **features** and the visit **context**

- Inverse Reinforcement Learning estimates the hidden reward function (experienced utility) that the users in a cluster apparently tried to maximise with the observed behaviour.

- The reward is a **function** of the selected features and context.

- The users choose visit actions with the **largest immediate and future reward** (Q function).

Deriving preferences from choices
Generating Recommendations

- Recommend to a user what is learned to be optimal for all the users in his cluster

<table>
<thead>
<tr>
<th></th>
<th>Q-BASE</th>
<th>SKNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reward@1</td>
<td>0.073</td>
<td>-0.007</td>
</tr>
<tr>
<td>Precision@1</td>
<td>0.043</td>
<td>0.109</td>
</tr>
<tr>
<td>Novelty@1</td>
<td>0.061</td>
<td>0.0</td>
</tr>
<tr>
<td>Reward@5</td>
<td>0.032</td>
<td>-0.010</td>
</tr>
<tr>
<td>Precision@5</td>
<td>0.045</td>
<td>0.068</td>
</tr>
<tr>
<td>Novelty@5</td>
<td>0.122</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Optimise preferences
Do not replicate the observed behaviour
Favour novel items

Have we correctly interpreted the user behaviour?
Alternative POI sample

- POIs were identified by action observation - *not corresponding to renowned ones*
- We repeated the test considering the subset of identified POIs present in TripAdvisor attractions – more popular

<table>
<thead>
<tr>
<th></th>
<th>Q-BASE</th>
<th>SKNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reward@1</td>
<td>0.369</td>
<td>0.097</td>
</tr>
<tr>
<td>Precision@1</td>
<td>0.101</td>
<td>0.108</td>
</tr>
<tr>
<td>Novelty@1</td>
<td>0.244</td>
<td>0.030</td>
</tr>
<tr>
<td>Reward@5</td>
<td>0.037</td>
<td>-0.061</td>
</tr>
<tr>
<td>Precision@5</td>
<td>0.056</td>
<td>0.062</td>
</tr>
<tr>
<td>Novelty@5</td>
<td>0.629</td>
<td>0.307</td>
</tr>
</tbody>
</table>

Now the precision is very similar
User study results

- **Model = Q-BASE**
- **Model = Q-POP PUSH**
- **Model = SKNN**
Why optimizing precision is bad

- If we optimize for precision the system will learn to recommend the items that the user found autonomously – not «useful» recommendations.
- When the precise recommendations are finished (already recommended) the system is unable to find novel recommendations.
- Measured precision is typically very low (10% in our data set) – so the user is mostly exposed to imprecise recommendations.

Advantages of the IRL method

- It is based on sequence mining but it can also generalise and suggest items never consumed before – depending on the chosen item features and context.

- By grouping users it can fix the “errors” of single users choosing apparently good items (wrong maximization of the experienced utility).

- Recommendations are not the predicted actions, they are the “optimal” ones (which could be further corrected if we better know the user utility function).
Lesson Learned

- Distinction between preferences and choices
- Use "knowledge" for modeling preferences and choices
- The distinction between expected and experienced utility and their role in recommendations
- Useful recommendations may be generated by deviating from the predicted behavior (imprecise)
- Individual recommendation may be generated by assuming that groups of similar user are driven by a hidden utility function.
Thanks

... in particular to my students and collaborators who contributed to develop these ideas:
- David Massimo
- Linas Baltrunas
- Laura Bledaite
- Marius Kaminskas
- Marko Gasparic
- Marko Tkalcic
- Matthias Braunhofer
- Mehdi Elahi
- Saikishore Kalloori
- Tural Gurbanov
- Thuy Ngoc Nguyen

I am hiring PhD students