Contextual Recommendations

Kalle Miller, Anne Maunu, Marko Pellinen & Jukka Ala-Fossi
The Articles


  An overview of Context-aware recommender systems and the different approaches for using contextual information in the recommendation process.


  A comparison of two approaches.
Why Contextual Recommendations?

- Users’ preferences and needs are different in different contexts
- Non-context-aware recommender systems may end up recommending items not appropriate for the current context
- Context-aware recommender systems (CARS) are designed to fix this problem
- The concept of context can be hard to define
Context in Recommender Systems

• Contextual factors
  ○ The context in which recommendations are provided.
  ○ e.g. time, location, purchasing purpose, shopping companion

• Structure of contextual factors
  ○ e.g. time: seconds, minutes, hours, days, months, years
What a recommender system knows about contextual factors

- **Fully observable**
  - Relevant contextual factors, their structure and values are known explicitly.

- **Partially observable**
  - Only some information is known explicitly.

- **Unobservable**
  - No information about contextual factors is explicitly available.
  - Recommendations are based on the latent knowledge of context.
How contextual factors change over time

- **Static**
  - The relevant contextual factors and their structure remain the same over time.

- **Dynamic**
  - Contextual factors may change.
  - For example some factors may be dropped if they are no longer relevant.
  - The structure of a contextual factor may also change.
  - Passive observation or user feedback can lead to changes.
Two extremes: everything is known – nothing is known

The representational view is when everything is known about context

- Domain-specific systems
- Relevant contextual factors may be difficult to determine a priori

Unobservable or dynamic context may also be problematic if there is no explicit model or representation of context
Representing and modeling context

- Traditional recommender systems try to estimate a rating function
  \[ R : \text{Users} \times \text{Items} \rightarrow \text{Ratings} \]

- In case of context-aware recommender systems, the rating function is multidimensional:
  \[ R : \text{Users} \times \text{Items} \times \text{Contexts} \rightarrow \text{Ratings} \]

- Contextual information may be obtained
  - explicitly: provided by user or sensors that measure specific physical or environmental information
  - implicitly: derived or inferred from other observed data
Paradigms for Using Contextual Information

In context aware recommender systems the context can be used in various stages of the process.

- **Contextual prefiltering**
  - The context is used to filter the relevant data, and then traditional recommendation systems are used to rank the data.

- **Contextual postfiltering**
  - Traditional ranking is done first, but after this the contextual information is used to adjust the recommendations.

- **Contextual modeling**
  - Contextual information is used directly in the rating estimation.
**Contextual Prefiltering**

In contextual prefiltering, the contextual information is used to filter only the most relevant 2D data. This could be for example (users x items). This relevant data can then be used for generating the actual recommendations.

The benefit is that only data that is in the right context gets recommended.
Contextual Postfiltering

In contextual postfiltering the contextual data is first ignored. The data is first ranked normally, and after that it is adjusted for each customer based on the context.

A major advantage is that this allows using any traditional recommendation technique, since the context is only adjusting the data afterwards.
**Contextual Modeling**

In contextual modeling the contextual information is used directly in the recommendation.

Due to the direct usage of the contextual mode, it is possible to create multidimensional recommendation functions.

- Multidimensional functions base the result on more than data than 2D functions. Example: (Users x Items x Contexts) $\rightarrow$ Ratings
Comparison of Pre- and Post-Filtering

- Panniello et al. did an experimental comparison of the pre- and post-filtering methods.
- They compared exact pre-filtering (EPF) with weight post-filtering and filter post-filtering methods.

<table>
<thead>
<tr>
<th>EPF</th>
<th>Weight</th>
<th>Filter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selects all the transactions referred to the exactly specified context</td>
<td>Reorders the recommended items by weighting the predicted rating with the probability of relevance in the specific context</td>
<td>Filters out recommended items that have small probability of relevance in the specific context</td>
</tr>
</tbody>
</table>
Comparison Results

- Filter dominates EPF, which dominates Weight
- In other words, either pre- or post-filtering methods are not clearly better
- Instead, which is better depends on the post-filtering method used
A More Practical Approach

In order to avoid an expensive search for a good post-filtering method, Panniello et al. suggest the following:

1. Compare pre-filtering with the un-contextual method
   a. un-contextual method outperforms: use post-filtering
   b. pre-filtering outperforms:

2. Choose a or b
   a. use pre-filtering to achieve “reasonable” results
   b. use post-filtering, but go for the best post-filtering method