Tailoring Recommendations for a Multi-Domain Environment

Emanuel Lacić
Dominik Kowald
Elisabeth Lex

© Know-Center GmbH, www.know-center.at
Traditionally ...

- Recommender Systems focus on **a single domain**
  - Movies
  - Music
  - ...

- Algorithms are usually **adapted** to support the available **domain-specific** data models

- The focus is completely on a particular market where the recommender is being utilized
  - Not actually a limitation, but suppose we …
Suppose we

... got a task to build a really great movie recommender

... did an exhaustive domain analysis, prepared the domain-specific data model, implemented the needed algorithms and made them available in a real system
Supose we

… got a task to build a really great movie recommender

… did an exhaustive domain analysis, prepared the domain-specific data model, implemented the needed algorithms and made them available in a real system

But then we

… got another task to build a really great music recommender
Supose we

… got a task to build a really great movie recommender

… did an exhaustive domain analysis, prepared the domain-specific data model, implemented the needed algorithms and made them available in a real system

But then we

… got another task to build a really great music recommender

Repeat the 2\textsuperscript{nd} step?
Supose we

… got a task to build a really great movie recommender

… did an exhaustive domain analysis, prepared the domain-specific data model, implemented the needed algorithms and made them available in a real system

But then we

… got another task to build a really great music recommender

Ok, maybe only this one time. Why not?
Suppose we … got a task to build a really great movie recommender.

… did an exhaustive domain analysis, prepared the domain-specific data model, implemented the needed algorithms and made them available in a real system

But then we … got another task to build a really great music recommender

Ok, maybe only this one time. Why not?

… got another task to build a really great venue recommender
Our goal

- Support recommendations in a **multi-domain environment**

- Address the **design decisions** which need to be considered when building a multi-domain recommender system

- Extend the **scope** and **notation** of multi-domain recommender systems
What does the literature say?

- First attempt to define the **concept of a domain** in the context of recommender systems by [Cantador et al., 2015]
  - Make a distinction between four different domain notations
  - Focus to enhance recommendations in the context of cross-domain recommender systems

- Others implicitly share this definition and focus also on cross-domain recommendations

Domain notations

- **Attribute level**
  - Items are of the **same type**, but have **different values** for certain **attributes**
    (e.g., comedy and romance)

- **Type level**
  - Items are of **similar types** and **share** some **attributes**
    (e.g., movies and TV shows)
Domain notations

- **Item level**
  - Items are not of the same type and differ in most or all attributes
    (e.g., music and venues)

- **System level**
  - Items and users belong to different systems
    (e.g., LastFM and MovieLens)
Multi-domain definition

- From [Cantador et al., 2015]:

  *Multi-domain recommendation*: recommend items in both the source and target domains, i.e., recommend items in $\mathcal{I}_S \cup \mathcal{I}_T$ to users in $\mathcal{U}_S$ (or, equivalently, in $\mathcal{U}_T$ or $\mathcal{U}_S \cup \mathcal{U}_T$).

- Mainly focus on the provision of cross-system recommendations
Multi-domain definition

- Our focus is on supporting multiple (possibly unrelated) domains within the same environment

- We extend the multi-domain notation by addressing the topics of:
  - Service Isolation
  - Data Heterogenity
  - Recommender Customization
  - Fault Tolerance
Show us some examples!
What was the approach?
Service Isolation

- Different **domains** have **different requirements** with respect to the request **load**

- Improve **hardware** utilization rate by **sharing** it across multiple domains

- **Performance** still needs to be guaranteed
  - A high request load of performance-intensive operations in one domain should **not impact** the performance of others
  - For example:
    - News recommender systems have challenging load peaks during morning hours and the lunch break at working days
Service Isolation

System level isolation:

- **Dynamically scale** the system to handle performance intensive load peaks

- For example, by adopting a Microservices architecture design pattern to build the recommender
Data Heterogenity

- The amount of data is doubled approximately every 40 months [McAfee and Brynjolfsson, 2012]
- Most systems have migrated to distributed solutions that can scale more easily handle streams of heterogeneous data

Data Heterogenity

- The amount of data is doubled approximately every 40 months [McAfee and Brynjolfsson, 2012]

- Most systems have migrated to distributed solutions that can scale more easily handle streams of heterogeneous data

- A multi-domain recommender system needs to handle a diverse set of data structures

- An easy integration of new data structures is important when new domains need to be supported

One possibility would be to use popular search engines like Apache Solr or Elasticsearch.

For example, such schema-less mode can be leveraged to dynamically add the data structure of new domains.

```json
{
    "id": "e12c4fb–ba85–46d5–896d–af65d1f3b48c",
    "item": "5a2bc423–15dc–47d3–8a8c–f543dc267a7c",
    "users_listened": [3907, 57017]
    "users_listened_count": 2
    "domain": "LastFM"
},
{
    "id": "8fe89bab–9631–4e87–81a0–a8dd3fba774c",
    "user": "23861",
    "item": 4105,
    "rating": 1.0
    "domain": "MovieLens"
}
```
Data Heterogenity

- Item level support?
- How does the recommender know about the domain?
Recommender Customization

- There are multiple ways of configuring the same algorithm
  - What combination of parameters does my domain need?

- Different domains have different domain-specific data features
  - What data structures should my algorithm use?

A multi-domain recommender approach needs to be aware of the underlying data structures and domain-specific parameters.
Recommender Customization

- Domain-specific algorithm setup should be outsourced

- For example, every algorithm could have a corresponding **profile** which provides the domain-specific information

  - Also useful for documenting approaches for the sake of reproducibility!

```plaintext
id: ktl_ub_cf_lastfm
# reference for a user-based-CF implementation
algorithm: GenericUBCF

# algorithm specific parameters
parameters:
  # item-level domain?
domain: lastFM
  # How to calculate user similarity?
similarity_function: OVERLAP # JACCARD, COSINE, etc.
neighbourhood_size: 40
user_action_fields: [ users_listened ]

id: ktl_hybrid_cs_lastfm
# reference for a hybrid implementation
algorithm: CrossSourceHybrid

# algorithm specific parameters
parameters:
  # item-level domain given by combining profiles
profile_ids: [ ktl_mp_lastfm, ktl_ub_cf_lastfm ]
recommender_weights: [0.1, 0.9]
```
Fault Tolerance

- Distributed deployment increases the probability of unexpected behavior (e.g., hardware shutdown, I/O problems, software bugs, etc.)

- Runtime **performance** should be guarantied for **all levels** in a multi-domain recommender system

- To cope with central node failures and increase the reliability of the system, orchestration services are usually utilized in combination with microservices
  - ZooKeeper
  - Eureka
  - Consul
  - …
Simple domain experiments

Data:

- LastFM which contains listening relationship between users and artists
- Foursquare which contains a 5-star scale for different venues
- MovieLens20M which contains a 5-star rating scale with a step size of 0.5

Task:

Find the best neighborhood of a UB-CF in each domain to be used in a hybrid recommender for cold-start users

Algorithms:

MP, UB-CF, Cross-Source Hybrid
Simple domain experiments

- Domains do not always share the same configuration parameters for a given algorithm.

- The parameter space needs to be explored for each domain independently.

<table>
<thead>
<tr>
<th></th>
<th>Approach</th>
<th>nDCG@10</th>
<th>UC</th>
</tr>
</thead>
<tbody>
<tr>
<td>LastFM CF</td>
<td>MP</td>
<td>.0180</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>N = 20</td>
<td>.1113</td>
<td></td>
</tr>
<tr>
<td></td>
<td>N = 30</td>
<td>.1129</td>
<td></td>
</tr>
<tr>
<td></td>
<td>N = 40</td>
<td>.1135</td>
<td>93.70%</td>
</tr>
<tr>
<td></td>
<td>N = 50</td>
<td>.1120</td>
<td></td>
</tr>
<tr>
<td></td>
<td>N = 60</td>
<td>.1112</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Hybrid</td>
<td>.1005</td>
<td>100%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Approach</th>
<th>nDCG@10</th>
<th>UC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Foursquare CF</td>
<td>MP</td>
<td>.0256</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>N = 20</td>
<td>.0364</td>
<td></td>
</tr>
<tr>
<td></td>
<td>N = 30</td>
<td>.0403</td>
<td></td>
</tr>
<tr>
<td></td>
<td>N = 40</td>
<td>.0426</td>
<td></td>
</tr>
<tr>
<td></td>
<td>N = 50</td>
<td>.0440</td>
<td></td>
</tr>
<tr>
<td></td>
<td>N = 60</td>
<td>.0452</td>
<td>49.58%</td>
</tr>
<tr>
<td></td>
<td>Hybrid</td>
<td>.0339</td>
<td>100%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Approach</th>
<th>nDCG@10</th>
<th>UC</th>
</tr>
</thead>
<tbody>
<tr>
<td>MovieLens CF</td>
<td>MP</td>
<td>.0658</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>N = 20</td>
<td>.0910</td>
<td></td>
</tr>
<tr>
<td></td>
<td>N = 30</td>
<td>.0945</td>
<td></td>
</tr>
<tr>
<td></td>
<td>N = 40</td>
<td>.0981</td>
<td></td>
</tr>
<tr>
<td></td>
<td>N = 50</td>
<td>.0965</td>
<td></td>
</tr>
<tr>
<td></td>
<td>N = 60</td>
<td>.0984</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>Hybrid</td>
<td>.0999</td>
<td>100%</td>
</tr>
</tbody>
</table>
Future work

- Study differences in the domain-specific configurations
  - Does a semantic relationship between domains impact the choice of domain-specific parameters?

- Explore the influence of textual content in different domains when trying to optimize diversity and serendipity
  - Do related domains have similar issues when trying to get out of the filter bubble?
THANK YOU FOR YOUR ATTENTION!

Questions?

Emanuel Lacić
elacic@know-center.at
Social Computing / Know-Center
Graz University of Technology