A Weight-Based Personalized Recommendation using Idiocentric and Collaborative Filtering

1 Literature Survey

Recommendation systems have gained popularity in web based systems since the appearance of papers on collaborative filtering in the 1990s [7, 11, 5].

- [7] explains the concept of collaborative filters. They introduce GroupLens as a system for collaborative filtering of netnews, to help people find articles they will like in the huge stream of available articles. They hypothesize that the users who have agreed on a certain aspect in the past will probably agree again.
- [11] describes a technique for making personalized recommendations to a user based on the similarities between the interest profile of that user and those of other users. They also test and compare four different algorithms for making recommendations using social information filtering.
- [5] presents an approach for collaborative recommendation where in the history of other users is used in the automation of a social method for informing choices to the user. Their results show that the communal history-of-use data can serve as a powerful resource for use in interfaces.
- [6] determines the potential predictive utility for Usenet news. They develop a specially modified news browser that accepts ratings and displays predictions on a 1-5 scale. They compute the predictions using collaborative filtering techniques and compare the results with noncollaborative approaches.
- [8] describes several algorithms designed for collaborative filtering, including techniques based on correlation coefficients, vector-based similarity calculations, and statistical Bayesian methods. They compare the predictive accuracy of the various methods in a set of representative problem domains. Over the past decade, web systems have moved towards personalized recommendations for better user experience.
- [3] describes a tag-based system for personalized recommendation. They propose an approach which extends the basic similarity calculus with external factors such as tag popularity, tag representativeness and the affinity between user and tag.

- [1] investigates user modeling strategies for inferring personal interest profiles from social web interactions. They analyze individual micro-blogging activities on twitter and compare different strategies for creating user profiles based on the twitter messages.
- [10] presents a personalization algorithm for recommendation in folksonomies, which relies on hierarchical tag clusters.
- [9] explores and analyzes different item-based collaborative techniques. They look into different techiniques for computing item-item similarities and various techniques to obtain recommendations from them. Significant developments in learning using graph data has led to recent advances in recommendation techniques.
- [2] presents a recommendation algorithm that includes different types of contextual information. They model the browsing process of a user on a movie database by taking random walks over the contextual graph.
- [4] models personalized tag recommendation as a ”query and ranking” problem. They also propose a novel graph-based ranking algorithm for interrelated multi-type objects.

2 Detailed Design

Figures 1 and 2 represents the component diagram and activity diagram for the recommendation system respectively.

3 Implementation

3.1 Read item dataset

The input item dataset for the algorithm is a JSON file. The first line of the dataset gives the type information about all the attributes in the entire dataset. The JSON file is read into an in-memory object.

Eg:

```
{ ”rating”: ”float”,
”genres”: ”string”,
”filming_locations”: ”string”,
”rated”: ”string”,
”language”: ”string”,
”title”: ”string”,
”country”: ”string”,
”imdb_id”: ”string”,
”directors”: ”string”,
”rating_count”: ”integer”}
```

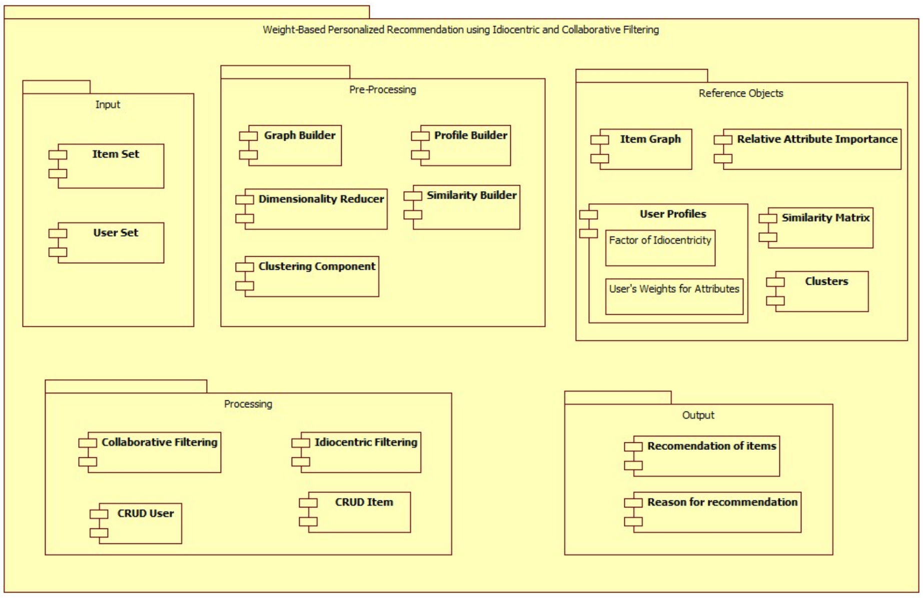


Figure 1: Component diagram for the Recommendation System

3.2 Build KeyValueNodes

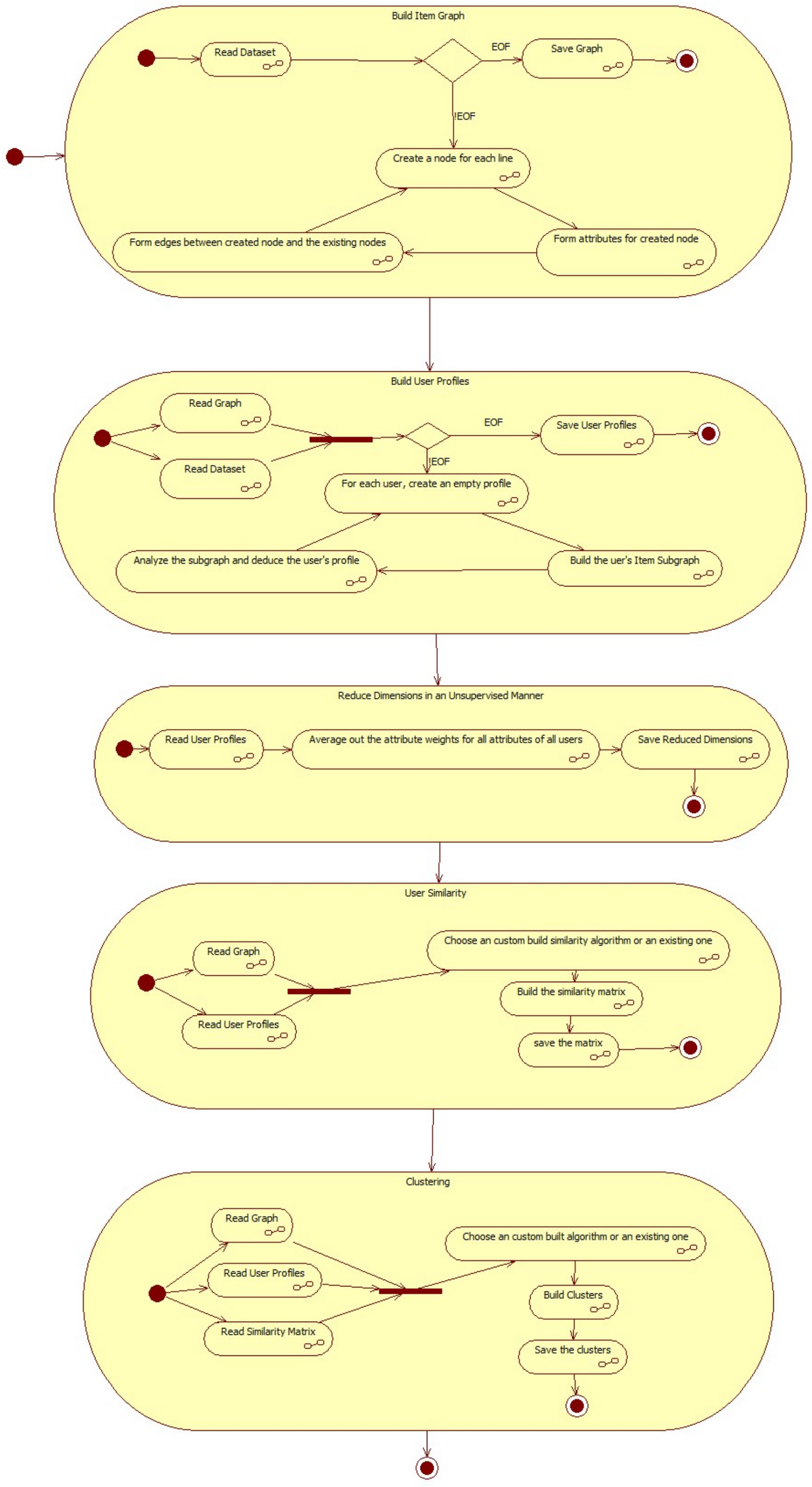
KeyValueNodes is a reverse-indexed associative array that we use to store details about items and the relationship among them.
Eg:

```
{
    ”directors”: {
                    ”Stuart Gillard”: [ ”3354” ],
                    ”Tobe Hooper”: [”1911”, ”2294”,”2377”,”2937”,”3210”],
                    ...
                },
    ...
}
```

3.2.1 Approach

This subsection explains the approach that we use to build KeyValueNodes. The item dataset consists of both categorical and non-categorical data. The categorical data will be either string or boolean type (Eg: Director and Like, respectively). Non-categorical data includes all the numerical attributes. (Eg: rating, year, timestamp). Algorithms 1 and 2 depict the approach that we use to build KeyValueNodes.

Input: set of items I



Build Item Graph
Read Dataset
EOF
Save Graph
!EOF
Create a node for each line
Form edges between created node and the existing nodes
Form attributes for created node
Build User Profiles
Read Graph
Read Dataset
EOF
Save User Profiles
!EOF
For each user, create an empty profile
Analyze the subgraph and deduce the user's profile
Build the uer's Item Subgraph
Reduce Dimensions in an Unsupervised Manner
Read User Profiles
Average out the attribute weights for all attributes of all users
Save Reduced Dimensions
User Similarity
Read Graph
Read User Profiles
Choose an custom build similarity algorithm or an existing one
Build the similarity matrix
save the matrix
Clustering
Read Graph
Read User Profiles
Read Similarity Matrix
Choose an custom built algorithm or an existing one
Build Clusters
Save the clusters

Figure 2: Activity diagram for the Recommendation System

Output: $KeyValueNodes$

```
1: Initialize $KeyValueNodes$ to an empty associative array
2: for all items $i$ in $I$ do
3:    for all categorical attributes $attribute$ in $i$ do
4:       for all values $value$ that $attribute$ can take do
5:          add $i$ to the $KeyValueNodes[attribute][value]$
6:       end for
7:    end for
8: end for
```

Algorithm 1: for categorical data

3.3 Generate user sequence

The user dataset originally is the form of :

$$user - id : item - id : rating : epoch$$

We convert this dataset to a JSON format in which we list all the users along with their item-id and corresponding rating in a chronological fashion. Eg:

```
{
"5988": [
            ["587", "1"],
            ["588", "3"],
            ["3006", "4"],
            ...
        ],
    ...
}
```

3.4 Build user profile

A user is an entity who consumes an item. In this section, we build a profile for each user. The user profile consists of characteristic parameters of the user that quantifies his behaviour. In Algorithm 3, we explain the methodology to build user profiles with frequency based relative attribute importance.

3.4.1 Frequency based relative attribute importance

3.4.2 Range based normalization

```
1: Initialize $D$ to an empty associative array
2: for all non-categorical attributes $attribute$ in the dataset do
3:   set D[$attribute$] to empty list
4: end for
5: for all item $i$ in I do
6:   for all non-categorical attributes $attribute$ of $i$ do
7:     D[attribute] $\leftarrow$ D[attribute] $\cup$ values taken by $attribute$ for $i$
8:   end for
9: end for
10: $clusters \leftarrow$ empty associative array
11: for all $attribute$ in D do
12:   $cutpoints \leftarrow []$
13:   sort D[$attribute$]
14:   $differences \leftarrow$ list of differences between consecutive values of D[$attribute$]
15:   $avrg \leftarrow$ average of $differences$
16:   populate $cutpoints$ with the indices for which the difference is greater than $avrg$
17:   populate $clusters[attribute]$ with list of values that fall between the consecutive
      indices
18: end for
19: for all item $i$ in I do
20:   for all non-categorical attribute $attribute$ of $i$ do
21:     clust $\leftarrow$ cluster that the value of $attribute$ belongs to
22:     add $i$ to the $KeyValueNodes[attribute][clust]$
23:   end for
24: end for
```

Algorithm 2: for non-categorical data

```
Input: $KeyValueNodes$
Output: $userProfiles$
1:  Initialize $userProfiles$ to an empty associative array
2:  for all users $u$ do
3:      $itemSequence \leftarrow$ items $u$ has consumed
4:      for all $attribute$ in $KeyValueNodes$ do
5:          Initialize $userProfiles[u][attribute]$ to an empty associative array
6:          $userProfiles[u][attribute]["RAI"] \leftarrow 0$
7:          for all $value$ in $KeyValueNodes[attribute]$ do
8:              $intersectionNodes \leftarrow itemSequence \cap KeyValueNodes$
9:              for all pairs of items $item_1$ and $item_2$ in $intersectionNodes$ do
10:                 increment $userProfiles[u][attribute]["RAI"]$ by the sum of ratings of
                    $item_1$ and $item_2$ by user $u$
11:                 if there is an increment then
                        $userProfiles[u][attribute][value] \leftarrow [increment,$ [$u$'s ratings of items
                        having $key$ and $value$]]
12:                 end if
13:             end for
14:         end for
15:         perform a sum-based normalization on all the values for
            $userProfiles[u][attribute]$
16:     end for
17:     perform a sum-based normalization on all the RAI's for all the attibutes of the
        user $u$.
18: end for
```

Algorithm 3: Building user profile

```
Input: $userProfiles$
Output: Range based normalization of $userProfiles$
1:  for all $profile$ in $userProfiles$ do
2:      for all $attribute$ in $profile$ do
3:          $userProfiles[profile]["weights"][attribute] \leftarrow$
            $userProfiles[profile]["weights"][attribute] \times |KeyValueNodes[attribute]|$
4:      end for
5:  end for
6:  for all $profile$ in $userProfiles$ do
7:      sumOfWeights $\leftarrow$ sum(userProfiles[profile]["weights"])
8:      for all $attribute$ in $profile$ do
9:          userProfiles[profile]["weights"][attribute] $\leftarrow \frac{userProfiles[profile]["weights"][attribute]}{sumOfWeights}$
10:     end for
11: end for
```

Algorithm 4: Range based normalization

3.5 Dimensionality reduction

In this section, we present an empirical approach to perform dimensionality reduction on the dataset. We observe the users' pattern in consuming the items and deduce the importance of an attribute. The importance of an attribute in the item dataset increases as number of users who give higher relative importance to that attribute increases. Hence, the importance of an attribute a in the item set I can be quantified as the mean relative importance of a, taken over all the users $u \in U$. In the previous section, we created a profile for each user. The characteristic parameter w_{u_p} consists of relative weights of attributes for u_p. The relative importance of an attribute a in the item set I can be computed as follows:

$$weight_a = \frac{\sum_{u_p \in U} w_{u_p}[a]}{Number\ of\ Users} \tag{1}$$

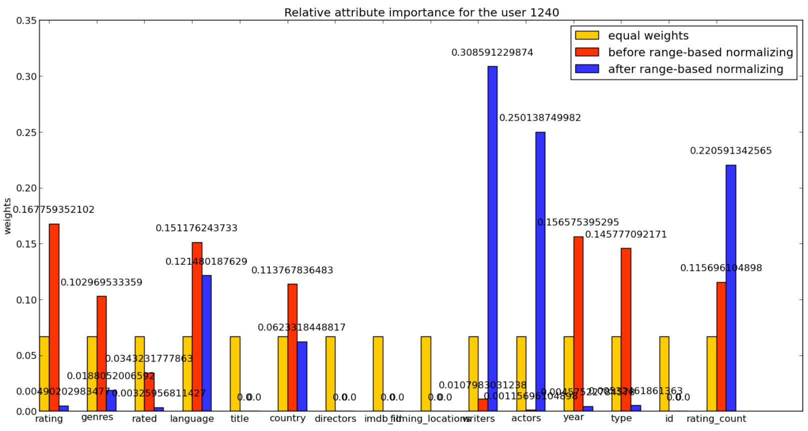


Figure 3: The figure indicates the relative attribute importance for the user 1240 before and after range based normalization.

References

[1] Fabian Abel, Qi Gao, Geert-Jan Houben, and Ke Tao. Analyzing temporal dynamics in twitter profiles for personalized recommendations in the social web. In *ACM WebSci'11*, pages 1–8, 2011.

[2] Toine Bogers. Movie recommendation using random walks over the contextual graph. In *CARS'10: Proceedings of the 2nd workshop on context-aware recommender systems*, 2010.

[3] Frederico Durao and Peter Dolog. Personalized tag-based recommendation in social web systems. In *CEUR Workshop Proceedings*, volume 485, pages 40–49, 2009.

[4] Ziyu Guan, Jiajun Bu, Qiaozhu Mei, Chun Chen, and Can Wang. Personalized tag recommendation using graph-based ranking on multi-type interrelated objects. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval. ACM*, 2009.

[5] W. Hill, Stead L., Rosenstein M., and Furnas G. Recommending and evaluating choices in a virtual community of use. In *Proceedings of CHI, '95*, 1995.

[6] Konstan J., Miller B., Maltz. D., Herlocker J., Gordon L., and Reidl J. Grouplens: Applying collaborative filtering to usenet news. Communications of the ACM, 40(3), pages 77–87, 1997.

[7] Resnick P., Lacovou N., Suchak M., Bergstrom P., and Reidl J. Grouplens: An open architecture for collaborative filtering of netnews. In *proceedings of CSCW '94, Chapel Hill, NC*, 1994.

[8] Breese J S, Heckerman D, , and Kadie C. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of 14th Conference on Uncertainty in Aritficial Intelligence*, pages 43–52, 1998.

[9] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *10th international conference on World Wide Web*, pages 285–295, 2001.

[10] Andriy Shepitsen, Jonathan Gemmell, Bamshad Mobasher, and Robin Burke. Personalized recommendation in social tagging systems using hierarchical clustering. In *ACM conference on Recommender systems*, pages 259–266, 2008.

[11] Shardanand U. and Maes P. Social information filtering: Algorithms for automating 'word of mouth'. In *Proceedings of CHI '95. Denver CO.*, 1995.