International Journal of Electronics Communication and Computer Engineering
Volume 6, Issue 3, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

Personalizing Recommender Systems Based on Neighborhood Collaborative Filtering

Tang Zhi-hang, Zhang Min-min, Ouyang Wen-min
School of Computer and Communication, Hunan Institute of Engineering Xiangtan 411104, China
Email: tang106261@126.com

***Abstract* – Recommender systems use historical data on user preferences and other available data on users and items to predict items a new user might like. Applications of these methods include recommending items for purchase and personalizing the browsing experience on a web-site. Collaborative filtering methods have focused on using just the history of user preferences to make the recommendations. These methods have been categorized as memory-based if they operate over the entire data to make predictions and as model-based if they use the data to build a model which is then used for predictions. Among various recommendation techniques, neighborhood- based Collaborative Filtering (CF) techniques have been one of the most widely used and best performing techniques in literature and industry. This paper proposes new approaches that can enhance the neighborhood-based CF techniques by identifying a few best neighbors (the most similar users to a target user) more accurately with more information about neighbors. To aid in the decision-making process, recommender systems use the available data on the items themselves. Personalized recommender systems subsequently use this input data, and convert it to an output in the form of ordered lists or scores of items in which a user might be interested. These lists or scores are the final result the user will be presented with, and their goal is to assist the user in the decision-making process. The application of recommender systems outlined was just a small introduction to the possibilities of the extension. Recommender systems became essential in an information- and decision-overloaded world. They changed the way users make decisions, and helped their creators to increase revenue at the same time.**

***Keywords* – Recommender Systems, Collaborative-Based Systems, Nearest Neighbour.**

I. INTRODUCTION

Recommender Systems(RecSys) are software tools and techniques providing suggestions for items to be of use to a user [1]. The suggestions relate to various decision-making processes, such as what items to buy, what music to listen to, or what online news to read.

As e-commerce Web sites began to develop, a pressing need emerged for providing recommendations derived from filtering the whole range of available alternatives. Users were finding it very difficult to arrive at the most appropriate choices from the immense variety of items (products and services) that these Web sites were offering. The explosive growth and variety of information available on the Web and the rapid introduction of new e-business services (buying products, product comparison, auction, etc.) frequently overwhelmed users, leading them to make poor decisions. The availability of choices, instead of producing a benefit, started to decrease users' well-being. It was understood that while choice is good, more choice is not always better. Indeed, choice, with its implications of freedom, autonomy, and self-determination can become excessive, creating a sense that freedom may come to be regarded as a kind of misery-inducing tyranny [2].

The study of recommender systems is relatively new compared to research into other classical information system tools and techniques. Recommender systems emerged as an independent research area in the mid-1990s [3]. In recent years, the interest in recommender systems has dramatically increased.

Recommender system is an active research area in the data mining and machine learning areas. Some conferences such as RecSys, SIGIR, KDD have it as a topic. Recommender systems are ubiquitous, and an average Internet user has almost certainly had experiences with them, intentionally or not. For example, the well-known Internet commerce, Amazon.com employs a recommender system that recommends products its users might be interested in, based on the shopping habits of other users. Social networking sites like Facebook or LinkedIn use recommender systems for recommending new friends to users based on their social network structure. The music website Last.fm uses a recommender system to recommend new music to a user, based on the listening habits of users with similar music taste. The Internet Movie Database (IMDb) recommends similar movies, based on the content and style of the movies user previously browsed. Streaming provider Netflix tries to predict new movies a user might be interested in based on his watching habits and movie ratings, compared to other users. These, and numerous other examples like Stumble Upon, Google AdSense, YouTube, etc., which differ in services provided, like audio, video, general item, social network, other Internet content, books, etc., demonstrate the importance of these systems.

Recommender systems facilitate making choices, improve user experience, and increase revenue, therefore should be easily accessible for deployment to interested parties. This led us to write a paper on recommender systems in a clearly understood and easily applied way through RapidMiner. We believe that RapidMiner's workflow approach entices systematic research and facilitates its implementation in combination with Rapid Analytics. The combination of research and production environment renders itself as an excellent environment for understanding recommender systems through practice. Throughout this paper, we will learn the basics of theory related to recommender systems, with a strong emphasis on practical implementation. This practical work will introduce you to all the necessary knowledge for rapid prototyping of recommender systems, thus enabling you to

master them through application of your data. The implementation of recommender systems in RapidMiner has been additionally simplified through the Recommender Extension.

II. Basic Conception

A) First Start

If we are starting RapidMiner for the first time, we will be welcomed by the so-called Welcome Perspective. The lower section shows current news about RapidMiner, if we have an Internet connection. The list in the center shows the analysis processes recently worked on. This is useful if we wish to continue working on or execute one of these processes. We can open a process from this list to work on or execute it simply by double clicking. The upper section shows typical actions which we as an analyst will perform frequently after starting RapidMiner. Here are the details of these:

1. New Process: Starts a new analysis process. This will be the most often used selection for you in the future. After selecting this, RapidMiner will automatically switch to the Design perspective .
2. Open Recent Process: Opens the process which is selected in the list below the actions.

Alternatively, we can open this process by double-clicking inside the list. Either way, RapidMiner will then automatically switch to the Design Perspective.

3. Open Process: Opens the repository browser and allows we to select a process to be opened within the process Design Perspective.
4. Open Template: Shows a selection of different pre-defined analysis processes, which can be configured in a few clicks.
5. Online Tutorial: Starts a tutorial which can be used directly within RapidMiner and gives an introduction to some data mining concepts using a selection of analysis pro- cesses. Recommended if we have a basic knowledge of data mining and are already familiar with the fundamental operation of RapidMiner.

B) Design Perspective

We will find an icon for each (pre-defined) perspective within the right-hand area of the toolbar:

Fig.1. Toolbar icons for perspectives.

The icons shown here take we to the following perspectives:

1. Design Perspective: This is the central RapidMiner perspective where all analysis processes are created and managed.
2. Result Perspective: If a process supplies results in the form of data, models, or the like, then RapidMiner takes you to this Result Perspective, where you can look at several results at the same time as normal thanks to the views.
3. Welcome Perspective: The Welcome Perspective already described above, in which RapidMiner welcomes you with after starting the program.

We can switch to the desired perspective by clicking inside the toolbar or alternatively via the menu entry "View" – "Perspectives" followed by the selection of the target per- spective. RapidMiner will eventually also ask you automatically if switching to another perspective seems a good idea, e.g., to the Result Perspective on completing an analysis process.

Now switch to the Design Perspective by clicking in the toolbar. This is the major working place for us while using RapidMiner. Since the Design Perspective is the central working environment of RapidMiner, we will discuss all parts of the Design Perspective separately in the following and discuss the fundamental functionalities of the associated views.

All work steps or building blocks for different data transformation or analysis tasks are called operators in RapidMiner. Those operators are presented in groups in the Operator View on the left side.

We can navigate within the groups in a simple manner and browse in the operators provided to your heart's desire. If RapidMiner has been extended with one of the available extensions, then the additional operators can also be found here. Without extensions we will find at least the following groups of operators in the tree structure:

- Process Control: Operators such as loops or conditional branches which can control the process flow.
- Utility: Auxiliary operators which, alongside the operator "Subprocess" for grouping subprocesses, also contain the important macro-operators as well as the operators for logging.
- Repository Access: Contains the two operators for read and write access in repositories.
- Import: Contains a large number of operators in order to read data and objects from external formats such as files, databases, etc.
- Export: Contains a large number of operators for writing data and objects into external formats such as files, databases, etc.Data Transformation: Probably the most important group in the analysis in terms of size and relevance. All operators are located here for transforming both data and meta data.

Modeling: Contains the actual data mining process, such as classification methods, regression methods, clustering, weightings, methods for association rules, correlation and similarity analyses as well as operators, in order to apply the generated models to new datasets.

Evaluation: Operators using which one can compute the quality of a modeling and thus for new data, e.g., cross-validations, bootstrapping, etc.

We can select operators within the Operators View and add them in the desired place in the process by simply dragging them from the Operators View and dropping them into the large white area in the center, the so-called Process View. Every analysis in RapidMiner is a process, and every process consists of one or several steps which are the operators. Depending on your settings, those new operators might be directly connected with existing operators as suitably as possible on the basis of the available meta data information. If this is not happening or

the automatically inserted connection is not desired, we can delete the connection by selecting them and pressing the Delete key or by pressing the Alt key while clicking on any of the connection ports. Ports are the round bubbles on the sides of the operators and they are used to define the data flow through your analytical processes. We can insert new connections by either clicking on the source port and then clicking again on a target port or by dragging a line between those ports.Later on, when we have successfully defined your first RapidMiner processes, a typical result might look like in the image on the following page:We could now simply try and select a few operators from the Operator View and drag them into the Process View. Connect their ports, even if this is probably not leading to working processes, and get familiar with the user interface of RapidMiner. In order to edit parameters we must select an individual operator. We will recognize the operator currently selected by its orange frame as well as its shadow. If we wish to perform an action for several operators at the same time, for example moving or deleting, please select the relevant operators by dragging a frame around these. In order to add individual operators to the current selection or exclude individual operators from the current selection, please hold the CTRL key down while we click on the relevant operators or add further operators by dragging a frame. We can also move operators around by selecting them and dragging them in the Process View. We will notice that the parameters in the Parameter View on the right side of RapidMiner changes sometimes if you select different operators. As you can see, most operators provide a set of parameters which control the actual working mode of the respective operator.

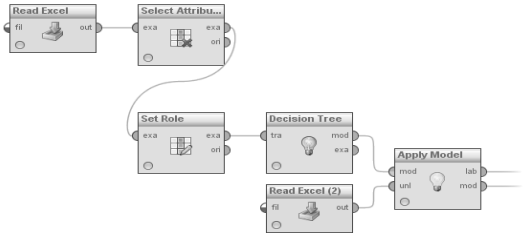

Fig.2. A typical process in RapidMiner consists of several operators.

C) Building a First Process

One of the first steps in a process for data analysis is usually to load some data into the system. RapidMiner supports multiple methods for accessing datasets. It supports more than 40 different file types and of course all major database systems. If the data is not originally stored in a relational database system, the best approach is to import the data first into the RapidMiner repository. Please follow the instructions from the RapidMiner manual for more information or just try to import a dataset, for example an Excel file, with "File" – "Import Data". Later we will also realize that there are dozens of operators for data import in the operator group "Import", which can also be used directly as part of the process.

1. Loading Dat

In the following we assume that we have managed to import the data into the Rapid- Miner repository and hence we will retrieve the data from there. If we are loading the data from a database or file, your following steps are at least similar to those described below.It is always recommended to use the repository whenever this is possible instead of files. This will allow RapidMiner to get access to the describing meta data and will ease process design a lot. We will now create the beginning of a data analysis process and will add a first data mining technique using this data. The very first operation in our process should be to load the data from the repository again in order to make it available for the next analysis steps:

1. Go to the Repositories view and open the repository Samples delivered with Rapid- Miner. Click on the small plus sign in front of this repository. We should now see two folders named data and processes. Open the data folder and we will find a collection of datasets coming together with RapidMiner. Click on the dataset named Iris and drag it onto the large white view named Process in the center of your frame. After releasing the dataset somewhere on the white area, it should be transformed into an operator named Retrieve with a bluish output port on the right side. RapidMiner automatically has transformed the dataset into an operator loading the dataset. If you click on the operator, we can see a parameter in the Parameters view pointing to the data location. The Retrieve operator in general, well, retrieves objects from a repository and makes them available in your process.

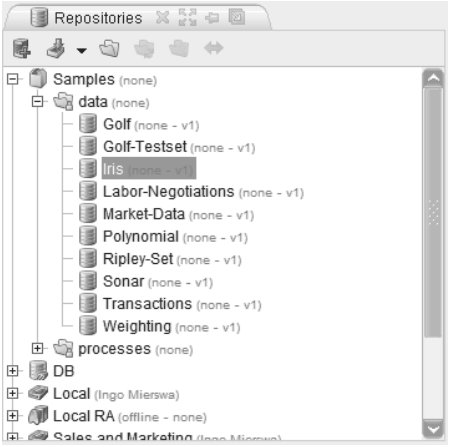

Fig.3. Drag the Iris dataset into the process view

2. Creating a Predictive Model

We have seen above how we create a new process just loading a dataset. The next step will be to create a predictive model using this dataset. This model predicts a categorical or nominal value, hence we could also say the model should describe rules which allow us to assign one of the three classes to new and unseen data describing new plants. We refer to this type of modeling as classification.
Adding a modeling technique to your process so that it calculated a predictive model is actually very easy. Just follow the following steps for creating such a model: Go to the Operators view and open the operator group Modeling, Classification, and Regression, and then Tree Induction. You should now see an operator named Decision Tree. Click on it and drag it to your process, somewhere to the right of your initial retrieve operator. We

International Journal of Electronics Communication and Computer Engineering
Volume 6, Issue 3, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

now only have to create the necessary connections. The dataset should be delivered to the modeling operator which is delivering a model then. However, we can also deliver the dataset itself to the user if we also connect the data port with one of the result ports. The complete process should look like the following figure.In the next section we will learn how to execute this process and how to inspect the created results.

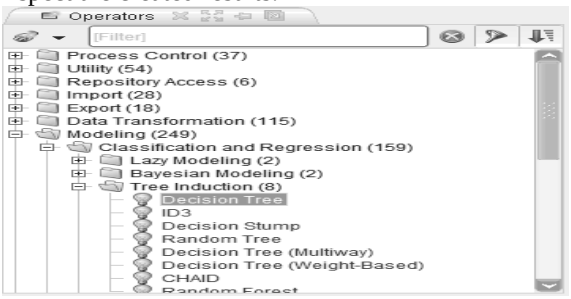

Fig.4. Drag the operator named "Decision Tree" into your process.

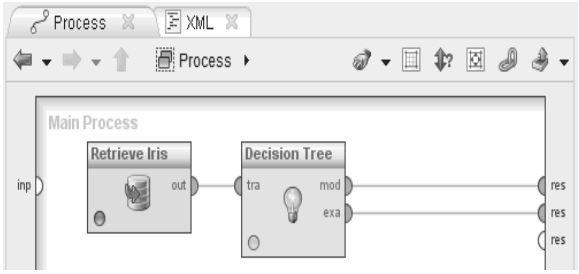

Fig.5. The complete process consisting of data loading and model creation.

D) Executing a Process

Now we are ready and want to execute the process we have just created for the first time. The status indicators of all used operators should now be yellow or green (the small traffic lights in each operator box) in the best case, and there should be no entries in the Problems View before you start executing a process. In such a case it should be possible to execute our process currently consisting of only one operator without any problems. However, the problems in the Problems view and also the red traffic lights only indicate that there might be a potential problem—we might be able to execute a process even if RapidMiner detects a potential problem. Just execute the process and check if it works despite the complaint as described below.

Looking at Results

After the process was terminated, RapidMiner should automatically have switched to the Result Perspective. If this was not the case, then we probably did not connect the output port of the last operator with one of the result ports of the process on the right-hand side. Check this and also check for other possible errors, taking the notes in the Problems View into consideration. Feel free to spend a little time with the results. The process above should have delivered a dataset and a decision tree used to predict the label of the dataset based on the attributes' values. We can inspect the data itself as well as the meta data of this dataset and try out some of the visualizations in the plot view. We can also inspect the decision tree and try to understand if this makes sense to you. If we wish to return to the Design Perspective, then we can do this at any time using the switching icons at the right of the toolbar.

Tip: After some time we will want to switch frequently between the Design Perspective and the Result Perspective. Instead of using the icon or the menu entries, we can also use keyboard commands F8 to switch to the Design Perspective and F9 to switch to the Result Perspective. What does that result mean to us? We now have managed to load a dataset from the RapidMiner repository and then we have built the first predictive model based on this data. Furthermore, we got a first feeling about how to build RapidMiner processes. We are now ready to learn more about the use cases for data mining and how to build corresponding processes with RapidMiner. Each of the following chapters will describe a use case together with the data which should be analyzed. At the same time each chapter will introduce new RapidMiner operators to you which are necessary to successfully solve the tasks at hand.

III. PERSONALIZING RECOMMENDER SYSTEMS

Collaborative recommender operators use the user-item matrix to build a recommendation model. This user-item matrix is presented as an example set of user-item pairs describing user consumption history. The recommendation model built with this matrix is used to recommend items to users from a query set. The query set is an example set containing identification numbers of users for which we want to make recommendations. For each user in the query set we recommend only the items not consumed by this user. Figure 6 depicts a basic collaborative recommender operator workflow.

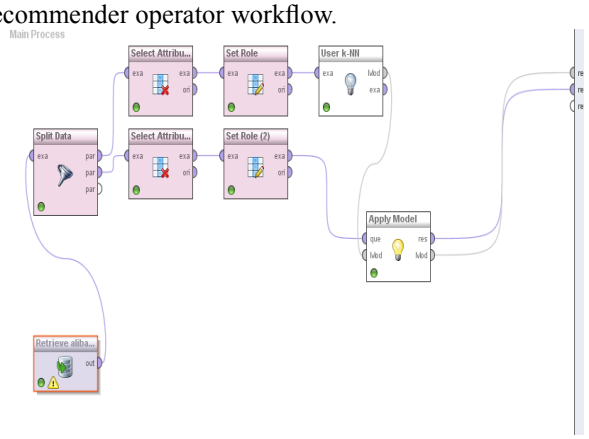

Fig.6. An example of an item recommendation work flow

The Recommended results shown in Figure 7.

In the item recommendation workflow, the first two operators read the train and the query example sets using the Read AML operators (1,4). Following, the appropriate roles are set to attributes using the Set Role operator (2). The user identification role was set to user id attribute and item identification role to item id attribute. Data attributes can have arbitrary names but roles for those attributes must be set. Next, we use the train data with the appropriately set roles to train an Item k-NN model (3).

International Journal of Electronics Communication and Computer Engineering
Volume 6, Issue 3, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209

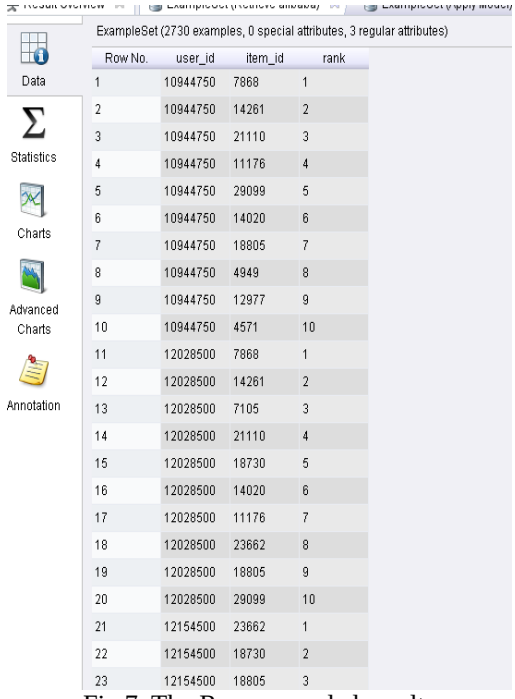

Fig.7. The Recommended results

At this point we can use our trained model to recommend new items to users in the query set using the Apply Model operator (6). Prior to model application, the user identification role was set for the query set (5). The Apply Model operator (6) returns an example set containing the first n ranked recommendations for every user in a query set. In Figure 6 we have seen how to make recommendations for particular users. In the following figure 8, we show how to measure performance of a recommendation model.

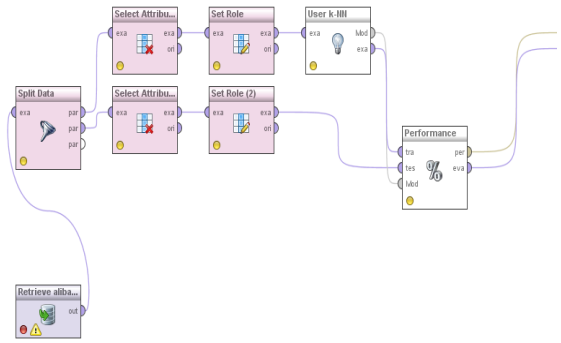

Fig.8. Measuring performance of a recommendation model.

The data management part of the workflow for measuring recommender model performance in Figure 8 is the same as in Figure 3. We use the Read AML operators (1,4) to load the data input, and the Set Role operators (2,5) to set the appropriate roles. In this workflow we use the test data (4) containing two attributes, the user id and the item id attribute and we set user identification and item identification roles to those at-tributes, respectively. The difference from the previous workflow is the need to calculate the performance of our built recommendation model (3). We use the Performance operator (6) to measure standard recommendation error measures we previously defined: AUC, Prec@k, NDCG, and MAP. The Performance operator (6) returns a performance vector and an example set containing performance measures. This enables a user to choose which format suits his or her needs. We can get Figure 9.

Fig.9. The performance of Recommender Systems

IV. Conclusion

In this paper we explain the k-NN classification algorithm and its operator in RapidMiner. The Use Case of this chapter applies the k-NN operator on the Teacher Evaluation dataset. The operators explained in this chapter are: Read URL, Rename, Numerical to Binominal, Numerical to Polynominal, Set Role, Split Validation, Apply Model, and Performance.

The k-Nearest Neighbor algorithm is based on learning by analogy, that is, by comparing a given test example with the training examples that are similar to it. The training examples are described by n attributes. Each example represents a point in an n-dimensional space. In this way, all of the training examples are stored in an n-dimensional pattern space. When given an unknown example, the k-nearest neighbor algorithm searches the pattern space forthe k training examples that are closest to the unknown example. These k training examples are the k "nearest neighbors" of the unknown example. The "Closeness" is defined in terms of a distance metric, such as the Euclidean distance.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (No. 61272295) and 2014 science and technology plan of Hunan province (No.2014GK3157).

References

[1] Mahmood, T., Ricci, F.: Improving recommender systems with adaptive conversational strategies. In: C. Cattuto, G. Ruffo, F. Menczer (eds.) Hypertext, pp. 73–82. ACM ,2009
[2] Schwartz, B.: The Paradox of Choice. ECCO, New York ,2004
[3] Anand, S.S., Mobasher, B.: Intelligent techniques for web personalization. In: Intelligent Techniques for Web Personalization, Springer ,2005, pp. 1–36.
[4] Tang Zhi-hang. Investigation and application of Personalizing Recommender Systems based on ALIDATA DISCOVERY. Int. J. Advanced Networking and Applications.Volume: 6 Issue: 2,2014 ,pp.2209-2213

AUTHOR'S PROFILE

Zhihang TANG

was born in Shaoyang, China, in 1974. He earned the M.S. degrees in control theory and control engineering from zhejiang University of techonlogy, in 2003 and Ph.D. from donghua University China in 2009. At the same time ,he is a teacher in department of computer and communication, Hunan Institute of Engineering (Xiangtan, China) from 2003.Chaired the 49th China Postdoctoral Science Foundation grant, presided over science and technology projects in Hunan Province in 2010, presided over the Education Department of Hunan Province in 2010 Outstanding Youth Project, as the first author more than 30 papers were published. His current research interests include intelligent decision and knowledge management.

Zhang Min-min

was born in gansu, China, in 1995,undergraduate of Hunan Institute of Engineering.

Ouyang Wen-min

was born in yueyang, China, in 1996,undergraduate of Hunan Institute of Engineering