2018.5.16.

Report on Personalized User Profiling for Recommendation (May 2018)

王梓睿 515021910551

ABSTRACT The report is mainly about the graph-based learning method for the user app behavior profiling. It is shown in the passage about its idea and realization. Also, the passage makes a brief introduction on the background of the user profiling and some former studies related.

INDEX TERMS User profiling, personalized recommendation, graph-based algorithm, deep neural network

I. BACKGROUND

Personalized services can be based on user needs, habits, preferences and other characteristics to form targeted presentation information retrieval results and recommendations, thus effectively improving the user experience and service satisfaction. Internet service providers are gradually providing personalized services as the core competitiveness of enterprises. Identifying and understanding user characteristics is the basis for developing differentiated services, and creating User Profile is the key factor of personalized service. With the rapid development of cooperative tag system, social network and mobile Internet, the source of user feature information is becoming more and more rich. How to create User Profile from various information data with different structure, different potential semantic and different scale according to application requirements, thus providing high quality personalized service is becoming the academic community and The joint research goal of the industry. Among different methods, the graph-based deep learning for profiling the user behaviors is a efficient one.

The User Profiling:

A user profile is a visual display of personal data associated with a specific user. A user profile can also be considered as the computer representation of a user model. A profile can be used to store the description of the characteristics of person. Profiling is the process that refers to construction of a profile via the extraction from a set of data. User profiles can be found on recommendation systems.

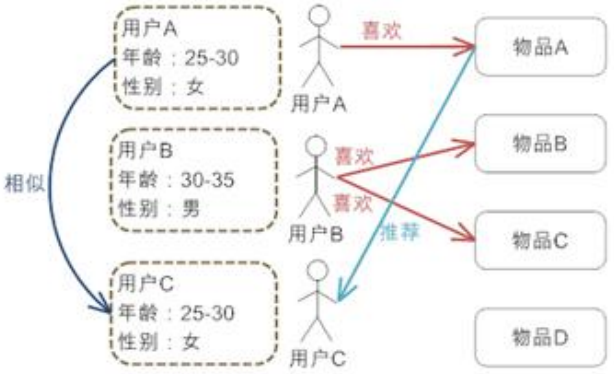
Factor of profiling:

Content based:User demographic information, social relations, preference habits and consumer behavior, etc.
Target analysis:Label and Weight.
Type of data:Static or Dynamic(for touch point).
Essential factors of the model:Character, Time, Place, Key word, etc.

II. FORMER STUDIES

Demography based representation

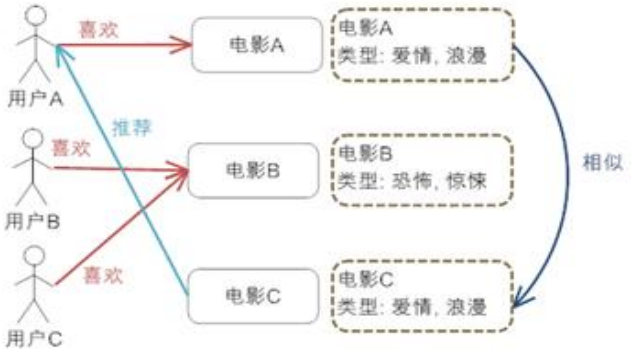
The core is based on user data modeling.



This is the simplest recommendation algorithm, which simply finds the user's relevance based on the basic information of the system user, and then recommends other items that similar users love to the current user.
The system will first model according to the user's attributes, such as user's age, gender, interest and so on. The similarity between users is calculated according to these features. For example, the system calculates that users are similar to A and C. A's favorite items will be recommended to C.
Advantage:
1. it does not need historical data, there is no cold start problem
2. it does not depend on the attributes of items, so the problems in other fields can be seamlessly connected.
Insufficient:
The algorithm is coarse, and the result is very difficult to satisfy. It is only suitable for simple recommendation.

Content based recommendation

The core is based on content data modeling.



This is similar to the above method, but this time the center turns to the item itself. Use the similarity of items rather than user similarity.

The system first models the attributes of an object (an example of a movie in a picture), using a type as an attribute. In practical applications, only the type is obviously too rough, and more information about actors, directors and so on is needed. Through similarity calculation, it is found that the similarity between A and C is high, because they belong to love category. The system also finds that user A likes A, and concludes that user A is probably interested in C. So the movie C is recommended to A.

Advantage:

The user interest can be well modeled, and a better recommendation accuracy can be obtained by increasing the dimension of item attributes.

Insufficient:

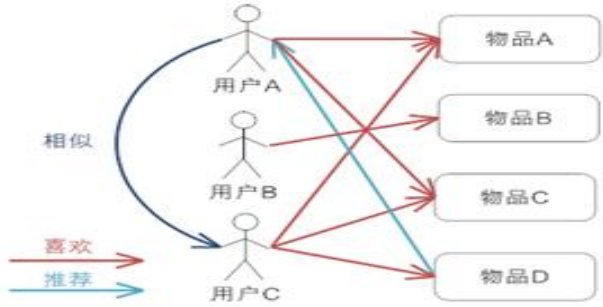
1. the attributes of goods are limited, so it is difficult to get more data effectively.
2. the measurement of the similarity of items only takes into account the goods themselves.
3., we need the historical data of users' articles, and have the problem of cold start.

Collaborative filtering:

The core is user interactive data modeling. Collaborative filtering based recommendations can be divided into three sub-classes: User-based Recommendation, project-based recommendation and model-based recommendation.

User based collaborative filtering recommendation:

Its basic assumption is that users who like similar items may have the same or similar tastes and preferences. According to the preference of all users on items or information, the "neighbor" user group, similar to the current user tastes and preferences, is used in the general application to calculate the "K- neighbor" algorithm; then, based on the historical preference information of the K neighbors, the current user is recommended.



Suppose user A likes items A and C, user B likes item B, user C likes item A, C and D;

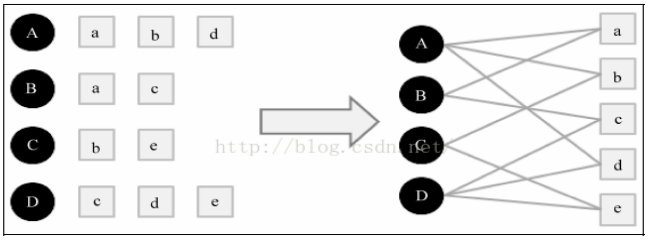
From the user's historical preference information, we can find that the tastes and preferences of user A and user C are similar, while user C also likes item D, so we can infer that user A may also like D, so you can recommend the item D to the user A.

The user based collaborative filtering recommendation mechanism and the demographics based recommendation mechanism are both calculated users' similarity, and are based on "neighbor" user groups, but they differ in how to calculate the user's similarity, and the mechanism based on demography only takes into account the user's own characteristics, and based on the user's Association. The same filtering mechanism calculates user similarity on user's historical preference data.

III. GRAPH-BASEDALGORITHM

Graph-based model is an important part of recommendation system. In fact, many researchers call the neighborhood based model as a graph based model, because a neighborhood based model can be seen as a simple form of a graph based model. Before the graph based model is studied, the user's behavior data is first needed to be represented as a graph. The following user behavior data is made up of two element arrays, in which each two-tuples (u, i) represents the user u 's over behavior of the item i, which is easily represented by a bipartite graph.

Let G (V, E) represent the user product bipartite graph, which is composed of user vertex set and item vertex set. For every two-tuples (u, i) in the data set, there is a set of corresponding edges in the graph. The following figure is a simple bipartite graph model of a user's item, in which a circular node represents a user, a square node represents an item, and the edges between a circular node and a square node represent the behavior of the user to the object. For example, the user node A in the graph is linked to the a, b and d of the item node, which shows that user A has generated behavior on items a, b and d.



After the user's behavior data is expressed as a two sub graph, the next is based on the two sub graph for the user to recommend, then the recommended items for the user u can be converted to measure the correlation of the vertex of the user's vertex Vu and the vertex that the Vu is not directly connected. The higher the correlation is, the higher the weight of the item on the recommended list, the recommendation is that the higher the weight of the recommended list, the higher the recommendation of the item on the list. The better the position is.

So how do you evaluate the relevance of the two vertexes? In general, it depends on three factors:

1: the number of paths between two vertexes

2: the length of the path between the two vertexes

3: the vertex passing through the path between the two vertexes

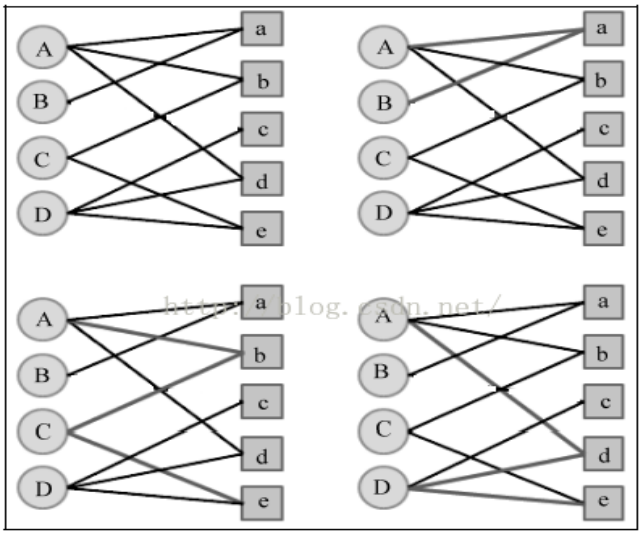
A pair of vertexes with high correlation generally has the following characteristics:

1: there are many paths connected between the two vertexes

2: the path between two vertexes is shorter

3: the path between two vertexes will not go through a larger vertex

For example, as a simple example, as shown in figure, the user A is connected to the item c and e, but the user A and the item c are connected by 1 paths of 3 length, and the user A and the item e are connected by a path of 2 lengths of 3. Then, the correlation between the vertex A and the e is higher than the vertex A and the c, so the item e should be placed before the item c in the recommendation list of the user A, because there are two paths between the vertex A and E. Among them, (A, b, C, e) path passes through the vertex of the degree of (3, 2, 2, 2), and (A, d, D, e) path through the vertex of the degree of (3, 2, 3, 2). Therefore, (A, d, D, e) passes through a relatively large vertex D, so the contribution of (A, d, D, e) to the correlation between the vertex A and e is less than that.



IV. REALIZATON

For collaborative filtering, the graph distance between user or item is a two-hop, and the relationship between further distances can not be taken into account.Then how to realize it?

The graph algorithm can break this limitation and regard the relationship between user and deal as a bipartite graph, and the relationship between them can be spread on the graph.

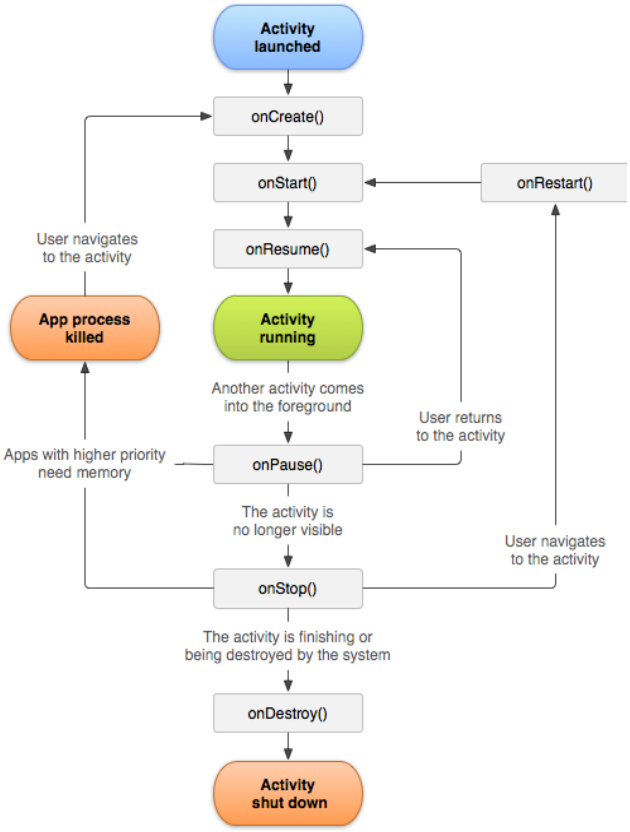
There is a design to generate the profile representation for each app. We use a function call graph to capture each app's comprehensive behavior information, Then an unsupervised deep learning is utilized to automatically extract representations from function call graph and we carefully design a graph encoding method to transfer a function call graph to a vector while preserving the function call relationship and function properties. That is, to follow three significant points to generate the representation:

Function call graph extraction:

We use Soot and Flow-Droid to extract function call graph from apps.

Related acknowledge in Android:

Flow-Droid and Soot for model building



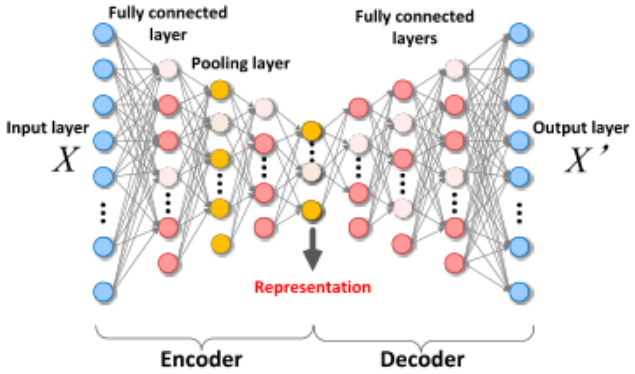
Graph encoding:

Before feeding function call graph into a neural network, we need to transfer graphs into vectors. We hold three principles to maintain complete relationships and function properties. First, every graph should have a unique encoded vector; Second, the similar graph should have similar vectors; Third, the mechanism should be robust to solve different circumstances.

Then it comes to function call graph, it provides a sufficient description of behaviors as defined graph G=(V,E,L), V is the vertex set for the nodes or functions, E is the call from one function to another, L represents the labels of every node. In this way, when the root node “DummyMain” is given, all user defined functions are the its children nodes without a fixed order though.
Then we need to decide the visiting order in the same layer. Be aware that system functions can be ordered by the lexicographical order of their function signatures, which is stable and consistent across apps, so the user defined functions can be considered as a set of system function and ordered based on its component system function. To be specific, it’s decided by the weight sum of the system functions’ priorities in it.
With the calculated order, we are ready to generate a vector Code(G) for the graph, when we traverse the graph from the root node completely, we get the corresponding vector Code(G).

Unsupervised representation learning:

The algorithm here refers to a deep auto encoder learning algorithm from unlabeled data. That is a feed forward neural network with an input layer, an output layer and several hidden layers. The nodes of the input layer is equal to those in output layer. We can divide the auto encoder into two parts or transitions: the encoder and the decoder. The training is conducted with back propagation algorithm to find the two transitions by minimizing the reconstruction error of the network. We choose deep fully-connected layers and the max-pooling layers to achieve the goal with a compact representation cost and help mitigate over-fitting. When the work is finished, we get the representation.
The following is a network with four fully-connected layers and two max-pooling layers, and we use the layer with the minim nodes as the representation.



V. CONCLUSION

Recommendation system is an effective tool to help users to solve the problem of information overload. It has been widely used in many fields including e-commerce. Collaborative filtering, content based recommendation, graph structure recommendation and mixed recommendation are the most common methods at present. This passage summarizes the related algorithms. And comparing to the representation method based on data, users or contents, that based on graph is improved in many aspects.
Though the recommendation technology has made such a great progress,the existing recommendation algorithms still face many difficulties and challenges, of which data sparsity, over-fitting, scalability and multimedia information feature extraction are still the main problems. The present technologies and methods can not yet solve these existing problems fundamentally. The continuous development of the domain, the recommendation system will also face new needs and problems. The development of the recommendation system is closely related to the problems and challenges it faces. The research on the recommended methods for the above problems is still a hot topic in the field of information retrieval, data mining and machine learning.

REFERENCES AND FOOTNOTES

[1]杜卿. 面向个性化服务的User Profile研究及应用[D].华南理工大学, 2014.
[2]AppDNA: App Behavior Profiling via Graph-based Deep Learning Shuangshuang Xue, Lan Zhang, Anran Li, Xiang-Yang Li,Chaoyi Ruan,Wenchao Huang University of Science and Technology of China
[3]Some other references from the Internet