International Journal of Latest Research in Engineering and Technology (IJLRET)
ISSN: 2454-5031
www.ijlret.com || Volume 02 - Issue 05 || May 2016 || PP. 100-107

Preference Elicitation by Solving the Cold Start Problem

Jaswanth Kannan, R1, Karthiga, S2, Manivel, S3
[1,3]*Student,* [2]*AssistantProfessor*
Department of Information Technology,
Thiagarajar College of Engineering, Madura

Abstract— There is an extensive class of Web applications that involve predicting user responses to options. Such a facility is called a recommendation system. A specific example of recommendation systems is to offer customers a non-line retailer suggestions about what they might like to buy, based on past history of purchases and/ or product searches. Cold start is a potential problem in computer-based information systems which involve a degree of automated data modeling. Specifically, it concerns the issue that the system cannot draw any inferences for users or items about which it has not yet gathered sufficient information. The cold start problem is most prevalent in recommender systems. Typically, are commander system compares the user's profile to some reference characteristics. These characteristics may be from the information item (the content-based approach) or the user's social environment (the collaborative filtering approach). Preference elicitation is a decision support system capable of generating recommendations to a user, thus assisting in decision making. It gives preferences accurately, finds hidden preferences and avoids redundancy.
Index Terms — Recommender System, Big Data, Neo4j tool, Graph.

I. INTRODUCTION

Big data analysis is one of the upcoming disciplines in data mining where the large unstructured data that is very difficult to store and retrieve in an efficient manner. Big data doesn‟t refer not only to exa bytes or peta bytes of data. When the amount of data that is needed to be processed is greater than the capacity of the system, then it refers to Big data. The three perspectives of big data are volume, velocity and variety [1]. Volume refers to the amount of data that is being processed. It has moved to Zetta bytes and Peta bytes as of 2014 and expected to increase in future. Velocity refers to the speed at which the data can be processed with minimal error rate. Variety refers to all types of data starting from un structure draw data to semi-structured and structured data which can be easily analyzed and used for the process of decision making and predictive analysis. In recommender systems research, most models work with rating data sets, such as Netflix data set and Movie lens dataset. The rating information is very important for obtaining good prediction accuracy, because it precisely indicates user's preferences and the degree of the interest on certain items. However, the rating information is not always available [2]. Some websites do not have a rating mechanism and thus their users cannot leave any rating feedback on the products. This situation requires evaluating implicit information which results in a lower prediction accuracy of the recommender systems. The information provided includes user ID, product ID and the clicking history of users with corresponding date.

II. LITERATUREREVIEW

A. Social Network

Social network [8] is a collection of social actors and the relationship, including nodes (socialactors), the edge between the nodes (theactors' association) and the weights of the edges (the impact between the actors). Each node, not independent individuals, is interdependent by sides. Sides, the channels of resource flowing, provide guidance for individual actions, greater weight bring stronger guidance. Sub-community whose points have strong relationship with each other means nodes' impact to others is larger than those outside. When multiple edges between the nodes, it can be changed to the matrix for data analysis and refining side weight in order to simplify the network.

B. Ontology Modeling.

According to Studer's definition putting forward in 1998, the ontology [9] is a shared conceptual model explicit formal specification. The goal of ontology is to capture the knowledge of related fields, to provide a common understanding of the domain knowledge to determine the terms of mutual recognition in the field, and give a clear definition of the mutual relations between these terms and terminology from the different levels of formalization model.

C. User profile learning techniques

User Modeling researchers have been investigating finding ways to elicit user preferences on various domains for years [16, 20]. For example, researchers examined if it would be a better idea to unobtrusively learn user-profiles from the natural interactions of users with the system. One way to categorize the methods proposed

thus far is by grouping them into explicit and implicit methods [13]. Implicit preference collection works by observing the behavior of the user and inferring facts about the user from the observed behavior [13]. In the recommender systems domain, implicit techniques may be more suitable if the item scan be consumed directly within the system. Also, implicit preference elicitations may be the only option where members can only provide implicit feedback or evaluation, such as by listening to or skipping a song, by browsing a webpage, by downloading some content, and so on. Explicit techniques, on the other hand, garner knowledge that is obtained when an individual provides specific facts to the user model [13]. Examples include users providing explicit feedback or evaluations on some rating scale. A comparative analysis between the explicit and implicit techniques can be found in [18]. Another way to classify the techniques of building user profile scan be based on the interaction process between the users and the system, particularly by looking at who is in control of the interaction process. [2] calls the possible interaction techniques human controlled, system controlled, and mixed initiative [11].

To explain these in the context of the recommender systems, a preference elicitation technique would be a) human controlled, if it is the user herself who selects (by typing the titles, for example) the items to evaluate, b) system controlled, if the system makes the list of items for the user to evaluate, and c) mixed initiative, if there are provisions for both user and system controlled interactions. The user controlled scheme may cause more work on behalf of the users; however the users may feel good being in charge [16]. One potential limitation of the user controlled scheme is that the users may not be able to identify the items to evaluate that express their preferences well. Further, they may only remember what they liked the most, not the opposite.

III. METHODOLOGY

There are several solutions that have been proposed to tackle the cold start problem. One of the effective solutions is to apply Active learning (machine learning) techniques, i.e., selectively choosing and obtaining more data, that can most improve the performance of there commander system. This is done by analyzing the available data and estimating the usefulness of the data points (e.g., ratings). [3]

In Collaborative filtering recommender systems, these techniques are so called rating elicitation Strategies. [4] In scenarios involving interface agents, the cold start problem may be overcome by introducing an element of collaboration amongst agents assisting various users.

This way, novel situations may be handled by requesting other agents to share what they have already learnt from their respective users.

[2] In recommender systems, the cold start problem is often reduced by adopting a hybrid approach between content-based matching and collaborative filtering. New items (which have not yet received any ratings from the community) would be assigned a rating automatically, based on the ratings assigned by the community to other similar items. Item similarity would be determined according to the items' content-based characteristics. [1]

The construction of the user's profile may be automated by integrating information from other user activities, such as browsing histories. If, for example, a user has been reading information about a particular music artist from a media portal, then the associated recommender system would automatically propose that artist's releases when the user visits the music store. [5]

It is also possible to create initial profile of a user based on the Personality characteristics of the user and use such profile to generate personalized recommendation. [6] [7] Personality characteristics of the user can be identified using a personality model such as Five Factor Model (FFM).

A. Graphs

Graphs are data structures that describe both data and their relationships. The most commonly recognized forms of a graph are networks and hierarchies. Graphs have nodes and relationships between those nodes. Both the nodes and the relationships can have properties. We can also apply labels to nodes.

B. Networks

A network comprises a set of nodes, with relationships between them.

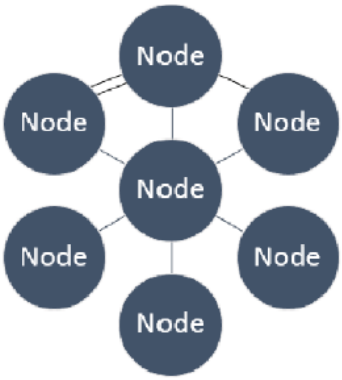
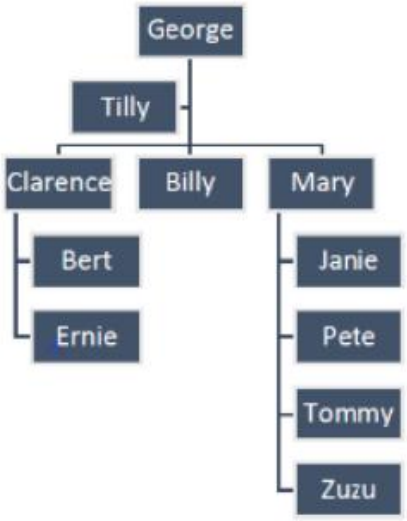


Fig 3.1Network

You might think of a network of friends, servers, databases or customers. In a graph database, the nodes in a graph don't have to be all the same type of thing–and neither do the relationships. Instead of defining an entity that imposes a standard on all instances, we build nodes and relationships that can each have their own set of properties. The nodes in Figure1- Network might be customers, orders, products and promotions. Each of these could have different types of properties. Not just by type of node, but also by instance of each node. For the customer node, we might have given name and family name, birth date and acquisition date as properties. But we might not have all of those properties for all customers. In a relational database, we would define common properties but set some of them to NULL if they don't have values for that instance. The relationships we have between nodes gets the same power: we can store properties about each relationship without imposing those properties on all of them. That's right, we can store properties about the relationships as well. This is a key difference from relational databases, where we would need to convert a relationship to at able to store metadata about that relationships.

C. Hierarchies & Trees

A hierarchy is a structure where nodes have other nodes above them and below them. A tree is a hierarchy with branches (multiples of nodes related to a parent node.) Of course, pure hierarchies rarely exist in the real world– more on that in the next section. We think of organizational reporting structures and supervisory assignments as examples of hierarchies. In a typical supervisory hierarchy, one employee reports to exactly one other employee. And that employee may have many other persons reporting to him. We might even be told these are the business rules around this reporting structure.



3.2 – Employee Hierarch

We can easily implement this in a relational database, with are cursive relationship on an EMPLOYEE table. A small hierarchy such as in Figure2 – Employee Hierarchy, maintaining those reporting relationships is

easy. As soon as we model a much larger set, though, maintenance gets more expensive. We often need to use workarounds such as special hierarchy data types or calculated columns that keep track of levels and pointers. Then what happens when a node gets a promotion? All those relationships must be reset and recalculated. If that node participates in many types of hierarchies, it's possible that many relationships must be reset and recalculated.

D. Data Modeling and Graph

In traditional data management, we prepare logical and physical data models. The logical data model describes business requirements for a data story and the physical data model specifies how data is to be persisted in a database. In a relational design, we apply a common structure to each instance of an entity. We have a CUSTOMER entity and all those entities share the same set of properties or attributes. This means we must discover and document all the properties we want to support prior to building the data base and importing data.
In a graph database, the logical mode list he physical model. You can even think about the graph model as a service model. We can do white board-level modeling of nodes and relationships, then add properties and labels. At that point we have completed all the data modeling we need to do to implement in a graph database. If we have traditional logical data models we can even pull in properties we already know about in to our graph model. Because the logical mode list he only model, going from data to database takes significantly less time and fewer resources (modelers, architects, DBAs and developers) than building relational master data solutions. We can also label each instances o that we can query our data based on a role they fill. For instance, we might want to query only organizational customers in certain queries.

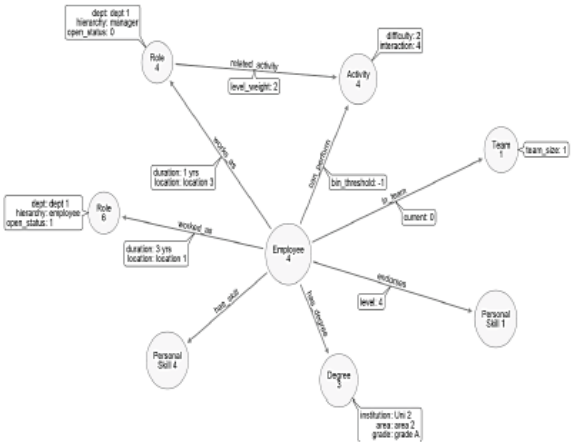


Fig 3.3 Employee Roles, Activities, Skills, Degrees

We can see that both nodes and relationships have varied properties.

IV. PROBLEM STATEMENTANDDESCRIPTION

A. Problem Statement

When a user first enters into a recommender system, the system knows nothing about her preferences. Consequently, the system is unable to present any personalized recommendations to her. This problem is sometimes referred to as the cold-start problem of recommender systems [5, 6, 15].

B. Problem Description

There are cold start problems for both new users and new items. In this paper, we investigate the cold-start problem for new users of recommender systems. We pose our research question as: how can we effectively learn preferences of new users so that they can begin receiving accurate personalized recommendations from the system? A related problem of recommender systems is the systemic bootstrapping problem recommender systems cannot serve anybody with personalized recommendations when the site has just started, devoid of any evaluations from anybody. We assume an existing recommender system with an established member base here.

Currently cold start solutions are the following: 1) Statistical model-based approach [6]: the corresponding probability distribution statistics is made according to the user, project and initialize rates and high probability items are priority recommended; 2) Average approach [7]: the original rating matrix is filled

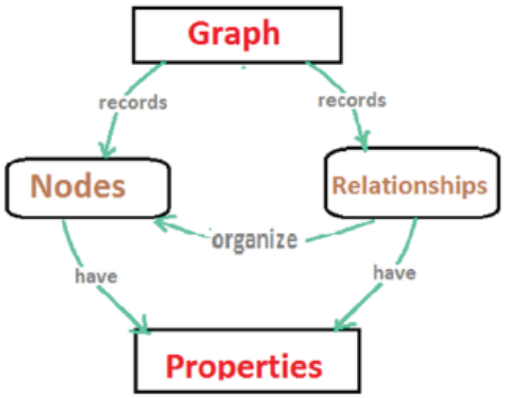
using the average of all ratings of the item before collaborative filtering. 3） Modeapproach: The predict results of the user is the score which occurred in his rating most often. However there is still the problem of low precision in recommendations in these methods.

V. PROPOSEDMETHOD

A. Neo4j tool

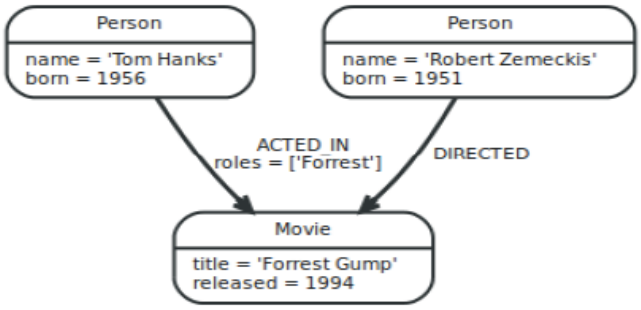
Neo4j unlike relational databases, graph databases are designed to store inter connected data that's not purely hierarchic, make it easier to make sense of that data by not forcing intermediate index in gate very turn, and also making it easier to evolve models of real world infrastructures, business services, social relationships, or business behaviors that are both fluid and multi-dimensional.

Neo4j is built "from the ground up" to support high to support high performance graph queries on large datasets for large enterprises with high-availability requirements. It includes its own graph query language, and uses native graph processing and a storage system natively optimized for graphs.



4.1 - Neo4

The graph contains nodes and relationships and each contains properties. The graph has a set of records
The persons are related through a movie. If the persons are related through a movie, they are all get connected by the relationship properties of acted in and directed.



4.2 – Relationship

VI. RESULTS AND DISCUSSIONS

Fig 5.1 Node

This shows the node for each persons and for every record node is created at the first instance

Fig 5.2 Node relationship

This shows the graph contains nodes and they are all related through the relationship of a movie dataset

Fig 5.3 Table Creation

For every record, the node contains properties. For a movie related, the properties are born date of actor, for a movie the properties are name and tagline and the released year is created.

Fig 5.4 Graph generated

The graph contains the related items. For a movie dataset, the relationship are acted_in, directed, produced.

VII. CONCLUSION AND FUTURE WORK

A. Conclusion

We have developed graph database for faster accessing through the graph. Implemented the graph database such that recommender system is built without relational table. The relational recommendation uses the table for the recommendation system. The relational recommendation used so far cannot solve the cold start problem. The recommendation system made of Graph database can solve the cold start problem. The graph database connects the nodes and makes relationships based on the particular property.

B. Future Work

The further extension to the project can be done by enhancing our method to implement the commercial website and restaurant management systems. It can be used in web-based application as an online mode and make a notification or alert to recommend to the user as well as customer. In addition, user privacy and security also need to be research to get a better user experience. The tool provide Master Data Management, Network and IT operations, Real Time recommendations, Fraud Detection, Social Network Identity and Access Management Graph Based Search.

REFERENCES

[1]. W.Chen, Z.Niu, X.Zhao, and Y.Li, "A Hybrid n Recommendation Algorithm Adapted in E-Learning Environments," World Wide Web, Sept. 2012, doi:10.1007/s11280-012-0187-z.

[2]. B. Mobasher, "Data Mining for Personalization," The Adaptive Web: Methods and Strategies of Web Personalization, P. Brusilovsky, A. Kobsa, and W. Nejdl, eds., pp.1-46, Springer, 2007.

[3]. A. I. Schein, A. Popescul, and L.H. Ungar, "Methods and Metrics for Cold-Start Recommendations," Proc. 25th Ann. Int' lACM SIGIR Conf. Research and Development in Information Retrieval, pp.253-260, 2002.

[4]. S. McNee, J. Ried l, and J.A. Konstan, "Being Accurate Is Not Enough: How Accuracy Metrics Have Hurt Recommender Systems," Proc. ACM SIGCHI Extended Abstracts on Human Factors in Computing Systems (CHIEA '06), pp.1097-1101, 2006.

[5]. R. Burke, "Hybrid: Recommender Systems: Survey and Experiments," J. User Modeling and User-Adapted Interaction, vol.12, no.4, pp.331-370, 2002.

[6]. J. kay, "Lifelong Learner Modeling for Lifelong Personalized Pervasive Learning," IEEE Trans. Learning Technology, vol.1, no.4, pp.215-228, Oct. 2008.

[7]. A. Tzikopoulos, N. Manouselis, and R. Vuorikari, "An Overview of Learning Object Repositories, "Learning Objects for Instruction: Designand Evaluation, P. Northrup, ed., pp.29-55, Idea Group, 2007.

[8]. V. Kumar, J. Nesbit, and K. Han, "Rating Learning Object Quality with Distributed Bayesian Belief Networks: The Why and the How, "Proc. Fifth IEEE Int'l Conf. Advanced Learning Technologies (ICALT '05), pp.685-687, 2005.

[9]. N. Manouselis, H. Drachsler, R. Vuorikari, H. Hummel, and R. Koper, "Recommender Systems in Technology Enhanced Learning," Recommender Systems Handbook, P.B. Kantor, F. Ricci, L. Rokach, and B. Shapira, eds., pp.387-415, Springer, 2011.

[10]. Lops, M.de Gemmis, and G. Semeraro, "Content- Based Recommender Systems: State of the Art and Trends," Recommender Systems Handbook, pp. 73-105, Springer, 2011.

[11]. P.-C. Chang and C.-Y. Lai, "A Hybrid System Knowledge-Based Systems, Combining Self-Organizing Maps with Case-Based Reasoning in Wholesaler's New-Release Book Forecasting," Expert Systems with Applications, vol.29, no.1, pp.183- 192, 2005 .

[12]. Y. Blanco-Fernandezetal., "A Flexible Semantic Inference Methodology to Reason About User Preferences in Knowledge Based Recommender Systems," Knowledge-Based Systems, vol.21, no.4, pp.305-320, 2008.

[13]. G. Adomavicius, N. Manouselis, and Y. Kwon, "Multi-Criteria Recommender Systems," Recommender Systems Handbook, pp.769-803, Springer, 2011.

[14]. M.K. Khribi, M. Jemni, and O. Nasraoui, "Automatic Recommendations for E-Learning Personalization Based on Web Usage Mining Techniques and Information Retrieval," Educational Technology and Soc., vol.12, no.4, pp.30-42, 2009.

[15]. J. Bobadilla, F. Serradilla, and J. Bernal, "A New Collaborative Filtering Metric That Improves the Behavior of Recommender Systems," Knowledge Based System, vol.23, no.6, pp.520-528, 2010.

[16]. E. Garcı´a, C. Romero, S. Ventura, and C. Castro, "An Architecture for Making Recommendations to Course ware Authors Using Association Rule Mining and Collaborative Filtering," User Modeling and User- Adapted Interaction, vol.19, no.1, pp.99-132, 2009.

[17]. E. Garcı´a, C. Romero, S. Ventura, and C.D. Castroa, "A Collaborative Educational Association Rule Mining Tool," Internetand Higher Education, vol. 14, no.2, pp.77-88, 2011.

[18]. A. Walker, M. Recker, K. Lawless, and D. Wiley, "Collaborative Information Filtering: A Review and an Educationa lApplication," Int'l J. Artificial Intelligence and Education, vol.14, pp.1-26, 2004.

[19]. D. Lemire, H. Boley, S. Mc Grath, and M. Ball, "Collaborative Filtering and Inference Rules for Context-Aware Learning Object Recommendation," Int'l J.Interactive Technology and Smart Education, vol.2, no.3, pp.179-188, 2005.