Collaborative Recommending Based on Core-Concept Lattice

Kai Li, YaJun Du, and Dan Xiang

School of Mathematical and Computers Science, Xihua University, Chengdu, Sichuan, 610039, China
likai0148@hotmail.com,duyajun@xhu.edu.cn,jintianxd@163.com

Abstract. In this paper, two new notions called core-concept and core-concept lattice are proposed and applied to collaborative recommendation system. The core-concept lattice is constructed based on the core-concept, which is extracted from rating matrix between users and items in collaborative recommendation systems. Compared with traditional FCA, it is obviously that the extraction of core-concept very easy and fast. We present the improved nearest neighbors algorithm, it use core-concept lattice as an index to the recommendation's ratings matrix. The improved nearest neighbors algorithm could remarkably accelerate finding the nearest neighbors. Therefore, it could evidently improve efficiency of recommendation.

1 Introduction

As the continuously developing of internet technology, internet has become an important tool which be used to retrieve information. But the rapidly expanded World Wide Web also causes the overloading problem of information. So, how to help users find the information efficiently and conveniently has become the concerned research.

Recommendation system is a special class of personalized systems that aim at predicting a user's interest on available products and services by relying on previously rated items or item attributes [1]. Current common approaches for personalized recommendation systems are the content-based filtering (CBF) and collaborative filtering (CF) [2, 3, 4, 5].

Content-based filtering makes predictions upon the assumption that a user's previous preferences or interests are reliable indicators for his/her future behavior. CBF requires that items are described by attributes, and is typically applied upon text-based documents, or in domains with structured data [6, 7].

On the other hand, Collaborative filtering operates upon the assumption that if a user A and B rates some items similarly, they share similar tastes and hence will rate other items similarly. Collaborative filtering is applicable to any type of content [6], while it can also capture concepts that are hard to represent, such as quality and taste [8]. Additionally, collaborative filtering does not restrict the spectrum of recommendations to items similar to the ones that the user has previously evaluated. Collaborative filtering has been acknowledged as the most successful and most widely implemented recommendation technique to date [9, 10]. For these reasons, we will focus on the collaborative filtering strategy in this paper.

O. Castillo et al. (Eds.): Theor. Adv. and Appl. of Fuzzy Logic, ASC 42, pp. 583–592, 2007.
springerlink.com

Collaborative filtering approaches can be distinguished into two major classes: model-based and memory-based [3, 11]. Model-based methods develop a model, which is applied upon the target user's ratings to make predictions for unobserved items.

In contrast to model-based, memory-based methods operate upon the entire database of users to find the closest neighbors of the target user and weight their recommendation according to their similarities. The fundamental algorithm of the memory-based class is the nearest neighbors (denoted as NN, hereafter); it can be described as a process divided in three steps as follows, for more details, sees [1, 12]:

1. Measurement of similarities between the target and the remaining users. A typical measure of similarity is the Pearson correlation coefficient.
2. Selection of the neighbors who will serve as recommenders.
3. Prediction based on the weighted average of the neighbors' ratings, weighted by their similarity to the target user.

The efficiency of computing similarity between all users in huge data must very low. For solving low efficiency of finding the nearest neighbors, the method of Formal Concept Analysis (FCA) was put forward in [13]. It regard rating matrix as formal context, based context, concept lattice was established as index, therefore accelerate finding the nearest neighbors. Experiment in [13] has proved feasibility of this method, but extract formal concept in formal context and establish concept lattice are also time-consuming.

Based above questions, aiming at collaborative recommending systems, we present core-concept and core-concept lattice in this paper. It obviously that extraction of core-concept is easier and faster than formal concept. Then we can do the work of collaborative recommendation on the basis of this method, and solve the low efficiency of finding the nearest neighbors ultimately.

We introduce relevant knowledge of FCA and present the approach of constructing core-concept lattice in section 2. Then, in section 3 we present how to apply core-concept lattice to collaborative recommending systems and propose the improved nearest neighbors algorithm. We conclude in section 4 with a look into the future.

2 Formal Concept Analysis

Formal Concept Analysis (FCA) is a mathematical method for analyzing binary relations, it's a power tool which used to analyze data and extract knowledge from formal context by concept lattice. 1982, concept lattice was first introduced by Wille [14], it established on the basis of FCA in theory. In FCA, data are structured into formal concepts, which form a concept lattice, ordered by a subconcept–superconcept relation. At present, FCA has been extensively applied in several areas such as knowledge discovery [15], software engineering [16] and case-based reasoning [17].

2.1 Formal Context and Formal Concept

First, we recall some basic notions of FCA. The definitions and theorems in this subsection are quoted from [14, 18, 19, 20, 21].

Definition 1. A formal context is a triple K: = (G, M, I) where G and M are sets and I $\subseteq G \times M$ is a binary relation. The elements of G are called objects and the elements of M are called attributes. The inclusion (g, m) $\in$ I is read "object g has attribute m". For A $\subseteq$ G, we define

$$A' := \{m \in M \mid \forall g \in A: (g, m) \in I\};$$

and for B $\subseteq$ M, we define dually

$$B' := \{g \in G \mid \forall m \in B: (g, m) \in I\};$$

In this paper, we assume that all sets are finite, especially G and M.

Definition 2. A formal concept is a pair (*A, B*) with $A \subseteq G$, $B \subseteq M$, $A' = B$ and $B' = A$. (This is equivalent to $A \subseteq G$ and $B \subseteq M$ being maximal with $A \times B \subseteq I$.) *A* is called extent and *B* is called intent of the concept.

Definition 3. The set $\underline{\mathfrak{B}}(\mathbb{K})$ of all concepts of a formal context $\mathbb{K}$ together with the partial order $(A_1, B_1) \leq (A_2, B_2) :\Leftrightarrow A_1 \subseteq A_2$ (which is equivalent to $B_1 \supseteq B_2$) is called concept lattice of $\mathbb{K}$.

The fundamental theorem of FCA [14] shows that each concept lattice is a complete lattice, and that the set of its intents is a closure system [18].

Theorem 1. (Fundamental Theorem of FCA). Let $\mathbb{K}$: = (G, M, I) be a formal context. Then $\underline{\mathfrak{B}}(\mathbb{K})$ is a complete lattice in which infima and suprema can be described as follows:

$$\bigwedge_{j \in J}(A_j, B_j) = \left(\bigcap_{j \in J} A_j, \left(\bigcup_{j \in J} B_j\right)''\right), \quad \bigvee_{j \in J}(A_j, B_j) = \left(\left(\bigcup_{j \in J} A_j\right)'', \bigcap_{j \in J} B_j\right)$$

Table 1. A binary formal context

	a	b	c	d	e	f
1	×	×				×
2	×	×			×	
3	×	×	×			
4	×		×			×
5	×	×		×	×	×
6	×	×	×	×		×

Definition 4. if (A,B) and (C,D) are two concepts of $\mathbb{K}$, (A,B)≤(C,D) iff $A \subseteq C$ (or, equivalently, $D \subseteq B$). (A, B) is called sub-concept of (C, D), and (C, D) is called super-concept of (A, B).

Table 1 describes a binary formal context. *G*={1, 2, 3, 4, 5, 6}, *M*={a, b, c, d, e, f }, *I* depicts objects in *G* have attributes in *M*.

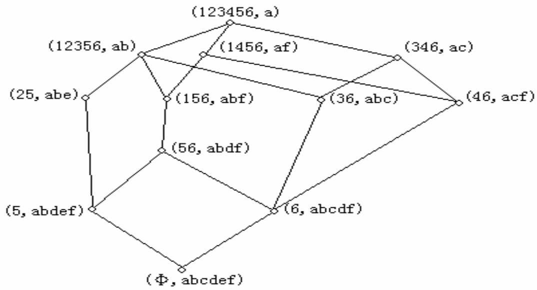

Fig. 1. The concept lattice that corresponds to the formal context in Table 1

2.2 Core-Concept and Core-Concept Lattice

In this subsection, we put forward definition of core-concept and core-concept lattice at first, then explain how to extract core-concepts and construct core-concept lattice.

Definition 5. A core-concept is a pair (A, B) with A$\subseteq$G, B$\subseteq$M. $\forall g \in G$, let E={g}, then $B = E'$ and $A = E''$. A is called extent and B is called intent of the core-concept.

Definition 6. The set $\Re(K)$ of all core-concepts of a formal context K together with the partial order (A1, B1) $\leq$ (A2, B2): $\Leftrightarrow$ A1$\subseteq$A2 (which is equivalent to B1$\supseteq$B2) is called core-concept lattice of K. Then $\Re(K)$ is a complete lattice after adding infima and suprema.

According to Definition 5, we could fast process the objects that correspond to the formal context in Table 1 one by one. The core-concept extracted from Table 1's formal context and the corresponding core-concept lattice is shown as follows:

(156, abf); (25, abe); (36, abc); (46, acf); (5, abdef); (6, abcdf).

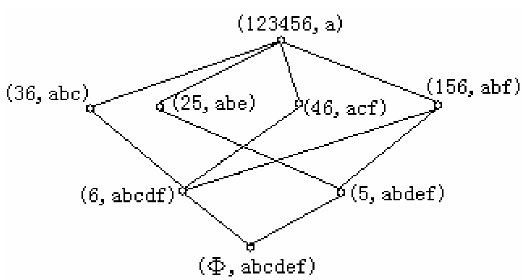

Fig. 2. The core-concept lattice that corresponds to the context in Table 1

The concise format [13, 21] of Fig. 2's core-concept lattice is shown in Fig. 3. In this lattice, the node labeled object denote a core-concept. Its extent is read from the descendants, and intent is read from the ancestors.

The number of core-concept is not better than the number of objects. It obviously that core-concept is formal concept all the same, but all core-concepts just a subset of whole formal concept.

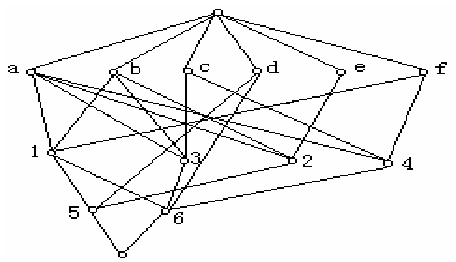

Fig. 3. The concise format core-concept lattice

3 Applying Core-Concept Lattice to Collaborative Recommendation

We apply core-concept lattice to collaborative recommendation for accelerating search for the nearest neighbors of target user. Aiming at collaborative recommendation, we propose the concept of core-concept and core-concept lattice. The core-concept lattice can then act as an index [13] to the ratings matrix to speed up the search for neighbors. Compared with traditional method of FCA, the all core-concepts just a subset of whole formal concept, however, based on Definition 5 and 6, each object must have a corresponding node in the concise format core-concept lattice, so it can completely predict and recommend for all users.

3.1 Obtaining Core-Concept and Core-Concept Lattice in Collaborative Recommendation

Core-concept requires a formal context, i.e., a binary relation between objects and attributes. An example of a ratings matrix for music is shown as Table 2. We take users to be objects and items to be attributes, the rating matrix shown in Table 2 correspond to a multi-value formal context.

Literature [13] states a Hypothesis: Users who rate the same items tend to rate items the same. For simplifying the computation and basing on the Hypothesis [13], we produce a formal context from the ratings matrix: a cell contains × iff $r_{u,i} \neq$^. The formal context that corresponds to Table 2 ratings matrix is shown as a cross-table in Table 3.

Table 2. A rating matrix for music

	Angel	Vincent	Incomplete	Ghetto
John	5	^	^	2
David	^	^	3	4
Nick	2	5	^	2
Anna	2	5	4	2
Christina	^	4	^	2
Billy	4	3	4	^
Jordan	4	5	^	3

Table 3. The context that corresponds to the rating matrix in Table 2

	Angel	Vincent	Incomplete	Ghetto
John	×			×
David			×	×
Nick	×	×		×
Anna	×	×	×	×
Christina		×		×
Billy	×	×	×	
Jordan	×	×		×

According to the section 2.2, the core-concept extracted from Table 3's formal context and the concise format core-concept lattice is shown as follows:

{Anna, (Incomplete, Ghetto, Angel, Vincent)};
{(Billy, Anna), (Incomplete, Angel, Vincent)};
{(Nick, Jordan, Anna), (Ghetto, Angel, Vincent)};
{(David, Anna), (Incomplete, Ghetto)};
{(John, Nick, Jordan, Anna), (Ghetto, Angel)};
{(Christina, Nick, Jordan, Anna), (Ghetto, Vincent)}.

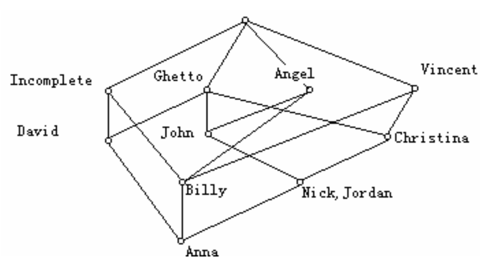

Fig. 4. The concise format core-concept lattice

3.2 Using Core-Concept Lattice to Find the Nearest Neighbors

Here we introduce how to use core-concept lattice to improve NN. Before computing the similarities between the target and the remaining users, we use core-concept lattice to reduce the number of remaining users. The improved nearest neighbors algorithm (denoted as INN, hereafter) shows as follows:

```
neighbor: The set of users who have not zero similarity with target
user.
nearest_neighbor: The set of users who will recommend to the target
user.
target_user.subset: The set of users who is sub-node in core-concept
lattice of the target user.
target_user.superset: The set of users who is super-node in core-concept
lattice of the target user.
target_user.accompanier: The set of users who share the same node in
core-concept lattice with the target user.
N: The number of nearest neighbors.
m: The number of all users.
Begin
  Input: target_user;
  Output: nearest_neighbor;
  neighbor={};
  nearest_neighbor={};
  for each user in target_user.subset do
  begin
    X←user;
    X. similarity ← similarity (target_user,X);
    neighbor←X;
  end;
  for i:1→m do
  begin
    if ( remain_user(i) in neighbor ) or (remain_user (i)
       in target_user.superset) or (remain_user (i) in
              target_user.accompanier) then
      continue;
    else
    if has_parent(remain_user (i), target_user) then
    begin
      X← remain_user (i);
      X. similarity ← similarity (target_user,X);
      neighbor←X;
    end;
  end;
nearest_neighbor=Select(neighbor, N);
end.
```

Given the target user, INN first walks core-concept lattice to find the neighbors who could recommend. It gets rid of the users who have no effect on the target user, but doesn't debase the accuracy and coverage.

There are conclusions about who could or not recommend to the target user. By using INN, it can visit the users' node of core-concept lattice in a most-likely order. Here we introduce how to find the candidate nearest neighbors, and give an example of recommending to Nick in the core-concept lattice depicted in Fig. 4:

a) If users shares one and the same node with target user, they have no additional ratings and so can't be the nearest neighbors. For Nick, according to Fig. 4, Jordan can't recommend new music to him.

b) If users' node is the super-node of target user's node, the items they has rated is a subset of the items which target user has rated, so they can't recommend to the target user. In Fig. 4, Christina and John can't be used to make recommendation for Nick.
c) If users haven't a common super-node with the target user, they have no co-rated items with target user, so they can't recommend to the target user.
d) The sub-nod should firstly considered, because may be they have the most similarity with target user. Thus Anna should take into account above all.
e) The node used to recommend should considered from the lowest level up, because the lower level the users locate, the more similarity the users and target user have.

Firstly, INN could remarkably reduce the numbers of the candidate user. Then it just need to compute similarities between the target user and a very few remaining users to find the nearest neighbors. The nearest neighbors' rating can then be used either to make predictions in the case where an objective item is also supplied, or to recommend items rated by the neighbors but not yet rated by the target user.

Based on the core-concept lattice, INN distinctly speeds up the search for the nearest neighbors. INN does considerably less work than NN, but both guarantee accuracy and coverage results equal to NN.

4 Conclusions

In this paper, aiming at collaborative recommendation system, we present two new concepts—core-concept and core-concept lattice, and have shown how core-concept lattice can be applied to collaborative recommendation system. It's time-consuming and troubled to extract formal concept in traditional FCA, however, compared with traditional FCA, our approach can obviously reduce the time of extracting concept, the extraction of core-concept is fast and easy. Though the number of all core-concepts less than whole formal concepts, it is enough for collaborative recommendation.

The method of traditional collaborative recommendation is first acquiring users' similarity matrix from users' rating matrix. By contrast, we build a core-concept lattice from a cross-table that is derived from the original ratings matrix when finding the neighbors. On the basis of core-concept lattice, we propose INN to predict and recommend, it doesn't need to compute the similarity between target user and all the other users, after finding the neighbors who have effect on the target user by using the core-concept lattice, it just needs to compute the similarity between the target and a very few candidate users. INN distinctly reduce the cost of finding the nearest neighbors, and also guarantee accuracy and coverage results equal to NN.

In the future works, we will apply INN to practice in collaborative recommendation system and prove its efficiency ulteriorly.

Acknowledgements

This work is supported by The Application Foundation of Sichuan Province (Grant No.2006J13-056). This work forms part of projects supported by the cultivating foundation of the science and technology leader of Sichuan province and the science & technology foundation of Xihua University (Grant No.R2006-26).

References

1. George Lekakos , George M. Giaglis, Improving the prediction accuracy of recommendation algorithms: Approaches anchored on human factors, in: Interacting with Computers 18 (2006) pp.410–431.
2. J. L. Herlocker, Understanding and Improving Automated Collaborative Filtering Systems, in: Ph.D. thesis, University of Minnesota, 2000.
3. D.M. Pennock, E. Horvitz, Collaborative filtering by personality diagnosis: a hybrid memory- and model-based approach, in: IJCAI Workshop on Machine Learning for Information Filtering, Stockholm, Sweden, August 1999, International Joint Conference on Artificial Intelligence.
4. Goldberg, D., Nichols, D., Oki, B. M., & Terry, D. Using collaborative filtering to weave an information tapestry, in: Communications of the ACM, 35(12), 1992, pp.61–70.
5. Mooney, R., & Roy, L. Content-based book recommending using learning for text categorization, in: ACM Conference on Digital Libraries, 5, 2000, pp. 195–204
6. Balabanovic, M., Shoham, Y. Fab: content-based collaborative recommendation, in: Communications of the ACM 40 (3). 1997 , pp. 66–72.
7. Pazzani, M.. A framework for collaborative, content-based and demographic filtering, in: Artificial Intelligence Review 13 (5–6), 1999, pp. 393–408.
8. Herlocker, J., Konstan, J., Riedl, J.. An empirical analysis of design choices in neighborhood-base collaborative filtering algorithms. in: Information Retrieval 5, 2002, pp. 287–310.
9. Burke, R.. Hybrid recommender systems: survey and experiments, in: User Modeling and User Adapted Interaction 12, 2002, pp. 331–370.
10. Sarwar, B., Karypis, G., Konstan, J., Riedl, J. Analysis of recommendation algorithms for e-commerce, in: Proceedings of the ACM E-commerce Conference, 2000, pp. 158–167.
11. Breese, J.S., Heckerman, D., Kadie, C.. Empirical analysis of predictive algorithms for collaborative filtering, in: Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence, 1998, pp. 43–52.
12. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.. Grouplens: an open architecture for collaborative filtering of netnews, in: Proceedings of the ACM Conference on Computer Supported Cooperative Work, 1994, pp. 175–186.
13. Patrick du Boucher-Ryan, Derek Bridge, Collaborative Recommending using Formal Concept Analysis, in: Knowledge-Based System, Volume 19, Issue 5, September 2006, PP.309-315.
14. R. Wille, Restructuring lattice theory: an approach based on hierarchies of concepts, in: I. Rival (Ed.), Ordered Sets, Reidel, Dordrecht, Boston, 1982, pp. 445–470.
15. G. Stumme, R. Wille, U. Wille, Conceptual Knowledge Discovery in Databases using Formal Concept Analysis Methods, in: Proceedings of the 2nd European Symposium on Principles of Data Mining and Knowledge Discovery, 1998, pp. 450–458.
16. T. Tilley, R. Cole, P. Becker, P. Eklund, A Survey of Formal Concept Analysis Support for Software Engineering Activities, in: Proceedings of the First International Conference on Formal Concept Analysis, 2003.
17. B. Díaz-Agudo, P.A. González-Calero, Classification-Based Retrieval using Formal Concept Analysis, in: Proceedings of the 4th International Conference on Case-Based Reasoning, 2001, pp. 173–188.
18. B. Ganter, R. Wille, Formal Concept Analysis: Mathematical Foundations, Springer, Berlin, 1999.

19. S. Wrobel, K. Morik, T. Joachims, Maschinelles lernen und data mining, in: G. Grz, C.-R. Rollinger, J.Schneeberger (Eds.), Handbuch der Knstlichen Intelligenz, vol. 3, Auflage, Oldenbourg, Munchen, Wien, 2000, pp. 517–597.
20. Sergei O. Kuznetsov, Complexity of learning in concept lattices from positive and negative examples, in: Discrete Applied Mathematics 142, 2004, pp. 111–125.
21. Gerd Stumme, Rafik Taouil, Yves Bastide, Nicolas Pasquier, Lotfi Lakhal, Computing iceberg concept lattices with TITANIC, in: Data & Knowledge Engineering 42, 2002, pp. 189–2