LITERATURE REVIEW FOR MINOR PROJECT

1.Personalized Recommender System Based on Web Log Mining and Weighted Bipartite Graph

Brief Abstract:
This paper says that current bipartite graph algorithms use unweighted edge graphs in their recommender systems. It stresses the importance, and improved accuracy of weighted edge graphs, where weight generally is a measure of the 'interest' of a user towards a product..

Principle: Weighted Bipartite Graph

Comments/Observations:
The paper explains how the log files are read, the data from this is used to understand the user's interest towards a product, and a weighted bipartite graph is constructed based on this. Now, user similarity is determined and the recommendation is given to each similar user.
In our project, we intend to use a similar framework to provide recommendation to similar users

2. Research on Personalized Recommender System for e-Commerce based on Web Log Mining and User Browsing Behaviors

Brief Abstract:
Explains a way to provide recommendations to users based on their browsing history and access frequency. It introduces a UserID-URL associated matrix, and uses that and a Distance matrix to cluster users into groups. Recommendations are provided to others in the group. Uses hamming distance to cluster users based on their activity.

Principle: Clustering Technology

Comments:
Not particularly useful for our project. Just a small introduction to web log mining, and user clustering.

3. Modeling User Behavior in Recommender Systems based on Maximum Entropy

Brief abstract:
Many online stores today use recommender systems to improve user-experience and gain better profits. But there is no mechanism in them to observe the user behaviour after the recommendations are made. Hence they significantly lack in evaluating the performance of the recommender system.

Principle/theory used : Maximum entropy principle

Some terms :
1. recommendation effect : degree of change in behaviour post the recommendations period

Observations/Comments :
The paper presents a probabilistic approach using Markov transitions to measure the change in user behaviour after the period of making recommendations to him. We can use this model in the Stage 4 of our project to measure the effectiveness of the recommender system that we build. Will be particularly useful for measures in the results section.

4. Mining web logs for a Personalized Recommender System.

Brief abstract:
The results of web searches are of the type: too high a recall and little or no value , making information extraction quite tedious and time consuming. So, arises the need for a personalised recommended system.Build a user profile based on weblog mining of the user

Principle/theory used:
Weblog analysis by:
a. Offline- classification of agent works to produce user meta profiles
b. Online- capture active user data to adapt to user's most recent preferences.

Observations/Comments :
Build user Meta profile based using:
a. monitoring agent
b. learning agent
c. recommendation agent

To improve the efficiency and effectiveness of the existing traditional information systems.

5. Recommender systems : An Introduction to approaches and algorithms, IBM developerWorks

An exhaustive article on the basics of recommender systems and their types. Explains in detail with valid examples.

Highlights:
Common algorithms used in recommender systems :-
1. Pearson correlation
   - linear dependence b/w two variables(users) as a function of their attributes
   - Disadvantage : should filter entire population into neighbourhoods based on some metric

2. Clustering algorithm
- many common clustering algos : k-means, Adaptive Resonance Theory(ART), Fuzzy C-means, Expectation-maximization
- need to explore each of these (if needed) to determine applicability of the algo in our project idea

3. Bayesian Belief Nets, Markov chains, Rocchio classification

Other observations/comments:
The article has some good points on discussion about the disadvantages and problems faced in recommender systems. Excellent content for beginning.

Note : There's a Part2 of the same article which discusses some common OpenSource recommender system engines

6. Efficient Recommender System based on Graph Data for Multimedia Application

Brief Abstract:
Scalability is the challenge when it comes to accessing the graph databases.Similarity between users and other users is most time consuming task.

Their idea is to build a b-tree and group similar users into one b-tree node in the leaf nodes.Through this the k-hop set in the graph is effectively reduced.After that it is a naive collaborative filtering on the users.

Principle: Collaborative filtering based-based on the usage pattern of connected users

Algorithm:
Input: Rating graph database G, rating threshold δr , rating frequency threshold δf .
Output: the candidate similar user vertices index I.
1. for each user vertex u $\in$G do
2. UR=UR$\cup$(all out rated edges from u and values of attributes of these rated edges V);
3. R = R(tail vertices from edges $\in$UR);
4. end do
5. for each item vertex r$\in$R in G do
6. ER= ER$\cup$(all rated edges head to r $\cup$values of attributes of these rated edges);
7. end do
8. for each rated edge ei $\in$ER do
9. if The value of subtracting vi UR rating value $\in$ from ei 's rating value v $\leq \delta r$
then Discard ei from ER ;
10. else FC = FC tail vertex from ei ;
11. end do

12. for each user vertex ci $\in$ FC
13. if the number of ci 's appearance frequency $\geq$ δf
then I = I$\cup$ci ;
14. end do
15. return I;