Web Graphs Recommendation For Domain-Specific Search

[1]Sheeba.R,[2]Dr. E.R. Naganathan
[1,2]Department of Computer Science and Engineering, Hindustan University,

Abstract
The various contents generated on the Web, Recommendation techniques have become increasingly indispensable. Innumerable different kinds of recommendations are made on the Web every day, including movies, music, images, books recommendations, query suggestions, tags recommendations, etc. No matter what types of data sources are used for the recommendations, essentially these data sources can be modeled in the form of various types of graphs. In this paper, aiming at providing a general framework on mining Web graphs for recommendations, 1) first propose a novel diffusion method which propagates similarities between different nodes and generates recommendations; 2) then illustrate how to generalize different recommendation problems into graph diffusion framework. The proposed framework can be utilized in many recommendation tasks on the World Wide Web, including query suggestions, tag recommendations, expert finding, image recommendations, image annotations, etc. The experimental analysis on large data sets shows the promising future of the work.

Keywords: Recommendation, diffusion, query suggestion, image recommendation.

1.0 Introduction

With the diverse and explosive growth of Web information, how to organize and utilize the information effectively and efficiently has become more and more critical. This is especially important for Web 2.0 related applications since user-generated information is more freestyle and less structured, which increases the difficulties in mining useful information from these data sources.

In order to satisfy the information needs of Web users and improve the user experience in many Web applications, Recommender Systems, have been well studied in academia and widely deployed in industry. Typically, recommender systems are based on Collaborative Filtering [14], [22], which is a technique that automatically predicts the interest of an active user by collecting rating information from other similar users or items. The underlying assumption of collaborative filtering is that the active user will prefer those items which other similar users prefer. Based on this simple but effective intuition, collaborative filtering has been widely employed in some large, well-known commercial systems, including product recommendation at Amazon,1 movie recommendation at Netflix,2 etc. Typical collaborative filtering algorithms require a user-item rating matrix which contains user-specific rating preferences to infer users' characteristics. However, in most of the cases, rating data are always unavailable since information on the Web is less structured and more diverse. Fortunately, on the Web, no matter what types of data sources are used for recommendations, in most cases, these data sources can be modeled in the form of various types of graphs. If I design a general graph recommendation algorithm, I solve many recommendation problems on the Web. However, when designing such a framework for recommendations on the Web, I still face several challenges that need to be addressed.

2.0 Diffusion on Graphs

2.1 Diffusion on Undirected Graphs

Consider an undirected graph G=(V,E), where V is the vertex set, and V ={v1; v2; . . . ; vn}.. E ={(vi, vj)} there is an edge between vi to vj} is the set of all edges. The edge {vi; vj} is considered as a pipe that connects nodes vi and vj. The value fi(t) describes the

heat at node vi at time t, beginning from an initial distribution of heat given by fið0Þ at time zero. f(t) denotes the vector consisting of fi(t). I construct this model as follows: suppose, at time t, each node i receives an amount M(i, j, t,Δt) of heat from its neighbor j during a time period Δt. The heat M(i, j, t,Δt) should be proportional to the time period Δt and the heat difference fj(t) – fi(t). Moreover, the heat flows from node j to node i through the pipe that connects nodes i and j. Based on this consideration, I assume that M(i, j, t,Δt)=α(fj(t) – fi(t))Δt, where α is the thermal conductivity—the heat diffusion coefficient. As a result, the heat difference at node i between time t +Δt and time t will be equal to the sum of the heat that it receives from all its neighbors. This is formulated as

$$\frac{f_i(t+\Delta t)-f_i(t)}{\Delta t}=\alpha\sum_{j:(v_j,v_i)\in E}(f_j(t)-f_i(t) \qquad \text{.....(1)}$$

where E is the set of edges. To find a closed form solution to (1), I express it in a matrix form

$$\frac{\mathbf{f}(t+\Delta t)-\mathbf{f}(t)}{\Delta t}=\alpha(\mathbf{H}-\mathbf{D})\mathbf{f}(t), \qquad \text{.....(2)}$$

Where

$$H_{ij}=\begin{cases}1, & (v_i,v_j)\in E \text{ or } (v_j,v_i)\in E,\\ 0, & i=j,\\ 0, & \text{otherwise,}\end{cases} \qquad \text{.....(3)}$$

and

$$D_{ij}=\begin{cases}d(v_i), & i=j,\\ 0, & \text{otherwise,}\end{cases} \qquad \text{.....(4)}$$

where d(vi) is the degree of node vi. From the definition, the matrix D is a diagonal matrix.

In order to generate a more generalized representation, I normalize all the entries in matrices H and D by the degree of each node. The matrices H and D can be modified to

$$H_{ij}=\begin{cases}1/d(v_i), & (v_i,v_j)\in E,\\ 0, & i=j,\\ 0, & \text{otherwise,}\end{cases} \qquad \text{.....(5}$$

and

$$D_{ij}=\begin{cases}d(v_i), & i=j,\\ 0, & \text{otherwise,}\end{cases} \qquad \text{.....(6)}$$

In the limit Δt →0, this becomes

$$\frac{d}{dt}\mathbf{f}(t)=\alpha t(\mathbf{H}-\mathbf{D})\mathbf{f}(t). \qquad \text{.....(7)}$$

Solving this differential equation, I have

$$\mathbf{f}(1)=e^{\alpha(\mathbf{H}-\mathbf{D})}\mathbf{f}(0), \qquad \text{.....(8)}$$

where d(v) denotes the degree of the node v, and e $^{\alpha(H-D)}$could be extended as

$$e^{\alpha(\mathbf{H}-\mathbf{D})}=\mathbf{I}+\alpha(\mathbf{H}-\mathbf{D})+\frac{\alpha^2}{2!}(\mathbf{H}-\mathbf{D})^2+\frac{\alpha^3}{3!}(\mathbf{H}-\mathbf{D})^3+\cdots. \qquad \text{.....(9}$$

The matrix e α(H-D) is called the diffusion kernel in the sense that the heat diffusion process continues infinitely many times from the initial heat diffusion.

2.2 Diffusion on Directed Graphs

The above heat diffusion model is designed for undirected graphs, but in many situations, the Web graphs are directed, especially in online recommender systems or knowledge sharing sites. Every user in knowledge sharing sites typically has a trust list. The users in the trust list can influence this user deeply. These relationships are directed since user a is in the trust list of user b, but user b might not be in the trust list of user a. At the same time, the extent of trust relations is different since user ui may trust user uj with trust score 1 while trust user uk only with trust score 0.2. Hence, there are different weights associated with the relations. Based on this consideration, I modify the heat diffusion model for the directed graphs as follows.

Consider a directed graph G = {V,E,W}, where V is the vertex set, and V = {v1, v2, . . . , vn}. W = {wij | where wij is the probability that edge (vi, vj) exist} or the weight that is associated with this edge. E = {(vi, vj)} | there is an edge from vi to vj and wij> 0} is the set of all edges. On a directed graph G(V ,E), in the pipe (vi,vj), heat flows only from vi to vj. Suppose at time t, each node vi receives RH =RH(I, j,t,Δt) amount of heat from vj during a period of Δt. I make three assumptions: 1) RH should be proportional to the time period Δt; 2) RH should be proportional to the heat at node vj; and 3) RH is zero if there is no link from vj to vi. As a result, vi will receive Σj:(vj;vi)εE$^{\sigma jfj(t)\Delta t}$ amount of heat from all its neighbors that point to it. At the same time, node vi diffuses

DH(i, t,Δt) amount of heat to its subsequent nodes. I assume that

1. The heat DH(i, t,Δt) should be proportional to the time period Δt.
2. The heat DH(i, t,Δt) should be proportional to the heat at node vi.
3. Each node has the same ability to diffuse heat.
4. The heat DH(i, t,Δt) should be proportional to the weight assigned between node vi and its subsequent nodes.

As a result, node vi will diffuse αwijfi(t)Δt/ $\Sigma_{k(I,k)\varepsilon E}$ wik amount of heat to each of its subsequent nodes vj, and each vj should receive αwijfi(t)Δt/ $\Sigma_{k(I,k)\varepsilon E}$ wikamount of heat from node vi. Therefore, σj =αwji/$\Sigma_{k:(j;k)\varepsilon E}$ wjk. In the case that the outdegree of node vi equals zero, I assume that this node will not diffuse heat to others. To sum up, the heat difference at node vi between time t + Δt and t will be equal to the sum of the heat that it receives, deducted by what it diffuses. This is formulated as

$$\frac{f_i(t+\Delta t)-f_i(t)}{\Delta t} = \alpha\left(-\tau_i f_i(t)+\sum_{j:(v_j,v_i)\in E}\frac{w_{ji}}{\sum_{k:(j,k)\in E}w_{jk}}f_j(t)\right), \quad \text{.....(10)}$$

where Ti is a flag to identify whether node vi has any outlinks. Solving it, I obtain

$$\mathbf{f}(1) = e^{\alpha(\mathbf{H}-\mathbf{D})}\mathbf{f}(0), \quad \text{.....(11)}$$

where

$$H_{ij}=\begin{cases} w_{ji}\Big/\sum_{k:(j,k)\in E} w_{jk}, & (v_j,v_i)\in E,\\ 0, & i=j,\\ 0, & \text{otherwise,}\end{cases} \quad \text{.....(12)}$$

and

$$D_{ij}=\begin{cases}\tau_i, & i=j,\\ 0, & \text{otherwise.}\end{cases} \quad \text{.....(13)}$$

3.0 Query Suggestion

After the conversion of the graph, I easily design the query suggestion algorithm in Algorithm 1.

Algorithm 1. Query Suggestion Algorithm

1: A converted bipartite graph G = (V + U V *, E) consists of query set V + and URL set V *. The two directed edges are weighted using the method introduced in previous section.

2: Given a query q in V +, a subgraph is constructed by using depth-first search in G. The **search stops** when the number of queries is larger than a predefined number.

3: As analyzed above, set α = 1, and without loss of generality, set the initial heat value of query q fq(0) = 1 (the choice of initial heat value will not affect the suggestion results). Start the diffusion process using

$$\mathbf{f}(1) = e^{\alpha \mathbf{R}}\mathbf{f}(0).$$

4: Output the Top-K queries with the largest values in vector f (1) as the suggestions.

Impact of the Size of Sub graph As mentioned in Section 3.5, due to the reason that Web graphs are normally very huge, I will perform algorithm on a subgraph extracted from the original graph. Hence, it is necessary to evaluate how the size of this subgraph affects the recommendation accuracy. I observe that when the size of the graph is very small, like 500, the performance of the algorithm is not very good since this subgraph must ignore some very relevant nodes. When the size of subgraph is increasing, the performance also increases. I also notice that the performance on subgraph with size 5,000 is very close to the performance with size 100,000. This indicates that the nodes that are far.

away from the query node are normally not relevant with the query node.

4.0 Image Recommendation

Basically, the graph construction for image recommendation is similar to the one introduced. The only difference is that here the nodes in bipartite graph are images and tags, respectively. By using the similar algorithm which is introduced in Algorithm 1, I can also provide image recommendations. If I use the tags instead of the images as the diffusion sources, then this problem turns to be the problem of tag recommendations. Since the recommendation processes are the same, I do not discuss the results in this paper.

4.1 Personalized Image Recommendation

Personalization is becoming more and more important in many applications since it is the best way to understand different information needs from different

users. Actually, this method can be easily extended to the personalized image recommendations.

5.0 Existing System

The first challenge is that it is not easy to recommend latent semantically relevant results to users. Take Query Suggestion as an example, there are several outstanding issues that can potentially degrade the quality of the recommendations, which merit investigation. The first one is the ambiguity which commonly exists in the natural language. Queries containing ambiguous terms may confuse the algorithms which do not satisfy the information needs of users. Users tend to submit short queries consisting of only one or two terms under most circumstances, and short queries are more likely to be ambiguous. Through the analysis of a commercial search engine's query logs recorded over three months in 2006, I observe that 19.4 percent of Web queries are single term queries, and further 30.5 percent of Web queries contain only two terms. Third, in most cases, the reason why users perform a search is because they have little or even no knowledge about the topic they are searching for. In order to find satisfactory answers, users have to rephrase their queries constantly. The second challenge is how to take into account the personalization feature. Personalization is desirable for many scenarios where different users have different information needs. As an example, Amazon.com has been the early adopter of personalization technology to recommend products to shoppers on its site, based upon their previous purchases. Amazon makes an extensive use of collaborative filtering in its personalization technology. The adoption of personalization will not only filter out irrelevant information to a person, but also provide more specific information that is increasingly relevant to a person's interests.

The last challenge is that it is time consuming and inefficient to design different recommendation algorithms for different recommendation tasks. Actually, most of these recommendation problems have some common features, where a general framework is needed to unify the recommendation tasks on the Web. Moreover, most of existing methods are complicated and require to tune a large number of parameters.

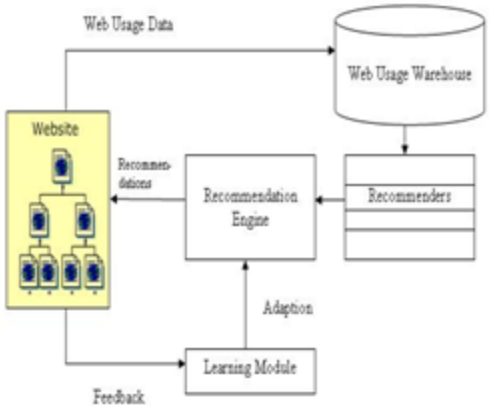
6.0 Proposed System

In this paper, aiming at solving the problems analyzed above, I propose a general framework for the recommendations on the Web. This framework is built upon the heat diffusion on both undirected graphs and directed graphs, and has several advantages.

1. It is a general method, which can be utilized to many recommendation tasks on the Web.
2. It can provide latent semantically relevant results to the original information need.
3. This model provides a natural treatment for personalized recommendations.
4. The designed recommendation algorithm is scalable to very large data sets.

The empirical analysis on several large scale data sets (AOL Click through data and Flickr image tags data) shows that the proposed framework is effective and efficient for generating high-quality recommendations.

7.0 Architecture



To achieve the above objectives, we use the recommendation system architecture shown in Fig. The main components of the system are:

The website – interacts with the web user, presents recommendations and gathers the feedback.

Web warehouse – stores information about the content of the website (e.g., products and product catalog), users, and the usage logs generated by the web server or the application server.

Set of recommender algorithms – the recommender

algorithms generate recommendations using data from the web warehouse. Recommendations can also be created by a human editor.

Recommendation rule database – stores the recommendations.

Learning module – refines the recommendation database based on the feedback obtained from the website.

As indicated in the figure we use two feedback loops for making recommendations. The first loop is periodically executed and involves calculating recommendation candidates by several recommendation algorithms utilizing more static information on the content as well as recent usageinformation from the web warehouse. The output of the algorithms is combined in one recommendation

database which is used to dynamically select recommendations. In the second feedback loop we continuously gather and evaluate user reactions on presented recommendations.

8.0 Conclusion

In this paper, I present a novel framework for recommendations on large scale Ib graphs using heat diffusion. This is a general framework which can basically be adapted to most of the Ib graphs for the recommendation tasks, such as query suggestions, image recommendations, personalized recommendations, etc. The generated suggestions are semantically related to the inputs. The experimental analysis on several large scale Ib data sources shows the promising future of this approach.

REFERENCES

[1] E. Agichtein, E. Brill, and S. Dumais, "Improving Ib Search Ranking by Incorporating User Behavior Information," SIGIR '07: Proc. 29th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval, pp. 19-26, 2006.

[2] E. Auchard, "Flickr to Map the World's Latest Photo Hotspots," Proc. Reuters, 2007.

[3] R. TiberiBaeza-Yates and A. Tiberi, "Extracting Semantic Relations from Query Logs," KDD '07: Proc. 13th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, pp. 76-85, 2007.

[4] R.A. Baeza-Yates, C.A. Hurtado, and M. Mendoza, "Query Recommendation Using Query Logs in Search Engines," Proc. Current Trends in Database Technology (EDBT) Workshops, pp. 588-596, 2004.

[5] D. Beeferman and A. Berger, "Agglomerative Clustering of a Search Engine Query Log," KDD '00: Proc. Sixth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, pp. 407-416, 2000.

[6] M. Belkin and P. Niyogi, "LaplacianEigenmaps for Dimensionality Reduction and Data Representation," Neural Computation, vol. 15, no. 6, pp. 1373-1396, 2003.

[7] J.S. Breese, D. Heckerman, and C. Kadie, "Empirical Analysis of Predictive Algorithms for Collaborative Filtering," Proc. 14th Conf. Uncertainty in Artificial Intelligence (UAI), 1998.

[8] S. Brin and L. Page, "The Anatomy of a Large-Scale HypertextualIb Search Engine," Computer Networks and ISDN Systems, vol. 30, nos. 1-7, pp. 107-117, 1998.