Chapter 2

Related Work

Recommendation on the Web is a general term representing a specific type of information filtering technique that attempts to present information items (queries, movies, images, books, Web pages, etc.) that are likely of interest to the users.

2.1 Collaborative Filtering

There are two types of collaborative filtering technique that is being studied. They are :

- Neighborhood based
- Model based.

The neighborhood-based approaches are the most popular prediction methods and are widely adopted in commercial collaborative filtering systems. The most common methods of neighborhood-based collaborative filtering are user-based approach and item-based approach. User-based approaches predict the ratings of active users based on the ratings of their similar users, and item-based approaches predict the ratings of active users based on the computed information of items similar to those chosen by the active user. In the model-based approaches, training data sets are used to train a predefined model. Examples of model-based approaches include the clustering model and the aspect model. This approach focuses on fitting the user-item rating matrix using low-rank approximations, and use it to make further predictions. Here only a small number of factors influences preferences, and that a users preference vector is determined by how each factor applies to that user.

One of the main disadvantages of using collaborative filtering is that most of these methods require the user-item rating matrix. However, on the Web, in most of the cases, rating data are always unavailable since information on the Web is less structured and more diverse. Hence, collaborative filtering algorithms cannot be directly applied to most of the recommendation tasks on the Web, like query suggestion and image recommendation.

2.2 Query Suggestion

Query suggestion is used for recommending relevant queries to Web users. This has been employed by many prominent commercial search engines like Google, Yahoo!, Ask etc.

The goal of query suggestion is similar to that of query expansion, query substitution and query refinement , which all focus on under- standing users search intentions and improving the queries submitted by users. Query suggestion is closely related to query expansion or query substitution, which extends the original query with new search terms to narrow down the scope of the search. But different from query expansion, query suggestion aims to suggest full queries that have been formulated by previous users so that query integrity and coherence are preserved in the suggested queries. Query refinement aims at interactively recommending new queries related to a particular query.

2.3 Clickthrough Data Analysis

Clickthrough data record the activities of Web users, which reflect their interests and the latent semantic relationships between users and queries as well as queries and clicked Web documents. each line of clickthrough data contains the following information: a user ID (u), a query (q) issued by the user, a URL (l) on which the user clicked, the rank (r) of that URL, and the time (t) at which the query was submitted for search. Thus, the clickthrough data can be represented by a set of quintuples (u , q , l , r , t). In the field of clickthrough data analysis, the most common usage is for optimizing Web search results or rankings. Web search logs are utilized to effectively organize the clusters of search results by

- learning interesting aspects of a topic and
- generating more meaningful cluster labels.

Clickthrough data can also be used to provide personalized search results for users, which Is done with the help of a ranking function which is learned from the implicit feedback extracted from search engine clickthrough data. Besides ranking, clickthrough data is also used in query clustering. Query clustering is a process used to discover frequently asked questions or most popular topics on a search engine. This process is crucial for search engines based on question- answering.

2.4 Image Recommendation

Besides query suggestion, another interesting recommendation application on the Web is image recommendation. Image recommendation system focuses on recommending interesting images to Web users based on users preference, example: Photoree. Normally, these systems first ask users to rate some images as they like or dislike, and then recommend images to the users based on the tastes of the users.

Chapter 3

Diffusion on Graphs

In this section, we first introduce a novel graph diffusion model based on heat diffusion. This model can be applied to both undirected graphs and directed graphs.

3.1 Heat Diffusion

Heat diffusion is a physical phenomenon. In a medium, heat always flows from a position with high temperature to a position with low temperature. Recently, heat diffusion-based approaches have been successfully applied in various domains such as classification and dimensionality reduction problems.

In this paper, we use heat diffusion to model the similarity information propagation on Web graphs. In Physics, the heat diffusion is always performed on a geometric manifold with initial conditions. However, it is very difficult to represent the Web as a regular geometry with a known dimension. This motivates us to investigate the heat flow on a graph. The graph is considered as an approximation to the underlying manifold, and so the heat flow on the graph is considered as an approximation to the heat flow on the manifold.

3.2 Diffusion on Undirected Graphs

Consider an undirected graph G = (V , E), where V is the vertex set, and V =v_1, v_2, v_3 v_n. E=(v_i1 ,v_j)—there is an edge between v_i to v_j is the set of all edges. The edge (vi , ,vj) is considered as a pipe that connects nodes v_i and v_j. The value f_i(t) describes the heat at node vi at time t, beginning from an initial distribution of heat given by $f_i(0)$ at time zero. f(t) denotes the vector consisting of f_i(t).

We construct our 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 f_j(t)- f_i(t). Moreover, the heat flows from node j to node i through the pipe that connects nodes i and j. Based on this consideration, we assume that

M (i , j , t , Δt) = α (f_j(t) - f_i(t)) Δt , where α is the thermal conductivity the heat diffusion coefficient. As a result, the heat difference at node i between time t + Δ t and time t will be equal to the sum of the heat that it receives from all its neighbors. This is formulated as

$$\frac{f_i(t+\triangle t)-f_i(t)}{\triangle t} = \alpha\Sigma(f_j(t) - f_i(t))$$

Where E is set of edges. To find a closed form solution, we express it in a matrix form

$$\frac{f_i(t+\triangle t)-f_i(t)}{\triangle t} = \alpha\text{(H-D)f(t)}$$

Where

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

And

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

Where d(v_i) is the degree of node v_i . From the definition matrix D is a diagonal matrix. We modify the above equations, by normalizing all the entries in matrix H and D. Hence the above equations change to

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

And

$$\text{D}_{ij} = \begin{cases} 1, & \text{i} = j, \\ 0, & otherwise. \end{cases}$$

In the limit $\Delta t \rightarrow 0$, this becomes

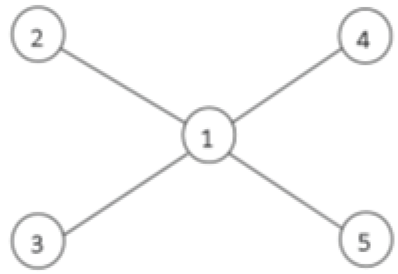
$$\frac{d}{dt}(f(t)) = \alpha(H - D)f(t)$$

solving this equation we get ,

$$\text{f(1) = e}^{\alpha(H-D)} \text{ f(0)}$$

where d(v) denotes the degree of node v ,and $e^{\alpha(H-D)}$ could be extended as $e^{\alpha(H-D)}$ =1 + (H - D) + $\alpha^2/2!(H-D)^2 + \alpha^3/3!(H-D)^3 + ..$

The matrix $e^{\alpha(H-D)}$ is called the diffusion kernel in the sense that the heat diffusion process continues infinitely many times from the initial heat diffusion. In order to interpret and the heat diffusion process more intuitively, we construct a small undirected graph with only five nodes as showed in Figure 1.

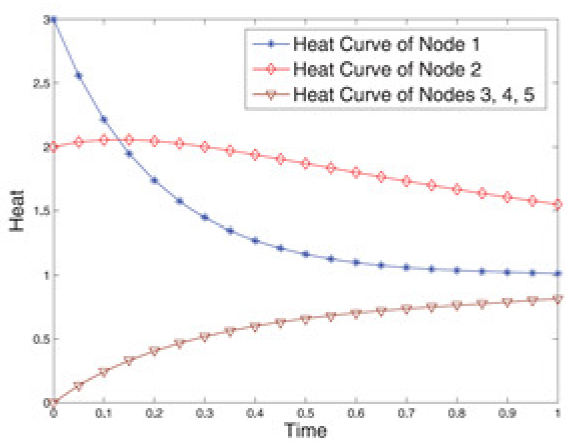


Heat diffusion example

Initially, at time zero, suppose node 1 is given 3 units of heat, and node 2 is given 2 units of heat; then the vector f(0) equals [3 , 2 , 0 , 0 , 0].The entries in matrix H - D are

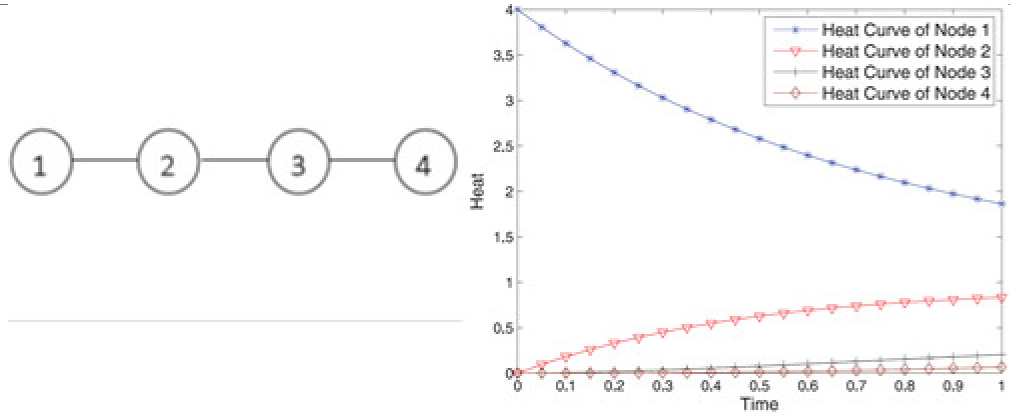
$$H - D = \begin{pmatrix} -1 & 1 & 1 & 1 & 1 \\ 1/4 & -1 & 0 & 0 & 0 \\ 1/4 & 0 & -1 & 0 & 0 \\ 1/4 & 0 & 0 & -1 & 0 \\ 1/4 & 0 & 0 & 0 & -1 \end{pmatrix}$$

Without loss of generality, we set the thermal conductivity $\alpha = 1$, and vary time t from 0 to 1 with a step of 0.05. The curve for the amount of heat at each node with time is shown in Figure 2. We can see that, as time passes, the heat sources nodes 1 and 2 will diffuse their heat to nodes 3, 4, and 5. The heat of nodes 3, 4, and 5 will increase respectively, and the trends of their heat curves are the same since these three nodes are symmetric in this graph.



Curve of heat change with time

Another example is shown in Figure 3. Initially, at time zero, suppose node 1 is given 4 units of heat ,then the vector f(0) equals [4 , 0 , 0 , 0].The related heat curve is shown in figure 4



Example of heat diffusion and Curve of heat change with time

We can see that the node 2, the closest node to the heat source, gains more heat than other nodes. This also indicates that if a node has more paths connected to the heat source, it will potentially obtain more heat. This is a perfect property for recommending relevant nodes on a graph.

3.3 Diffusion on Directed Graphs

The model discussed earlier was designed for undirected graph, 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. Hence this forms a directed graph scenario. 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. These weights must be taken into consideration while designing a framework.

Based on this consideration, we modify the heat diffusion model for the directed graphs. Consider a directed graph G = V , E, W , where V is the vertex set, and V =v_1, v_2, $v_3 v_n$. W= w_{ij}— where w_{ij} is the probability that edge (v_i , v_j) exists or the weight that is associated with this edge. E= (v_i , v_j) — there is an edge from v_i to v_j and $w_{ij} \prod 0$ is the set of all edges. On a directed graph G (V , E) in the pipe (v_i , v_j) heat flows only from v_i to v_j. Suppose at time t, each node v_i receives RH = RH (i , j ,t , $\triangle$t) amount of heat from v_j during a period of $\triangle$t.

We make 3 assumptions :

- RH should be proportional to the time period $\triangle$t.
- RH should be proportional to the heat at node v_j.
- RH is zero if there is no link from v_j to v_i.

As a result v_i will receive j: (v_j,v_i) $\in$ E j ft (t) $\triangle$t amount of heat from all its neighbors that point to it.

At the same time, node vi diffuses DH (i ,t ,$\triangle$t) amount of heat to its subsequent nodes. We assume that

- The heat DH (i ,t , $\triangle$t) should be proportional to the time period $\triangle$t.
- The heat DH(i ,t , $\triangle$t) should be proportional to the heat at node v_i.
- Each node has the same ability to diffuse heat.
- The heat DH (i ,t , $\triangle$t) should be proportional to the weight assigned between node v$_i$ and its subsequent nodes.

Therefore, σj = α $w_{ji}/\Sigma_{k:(j,k)\epsilon E} w_{jk}$. In the case that the outdegree of node v_i equals zero, we assume that this node will not diffuse heat to others. To sum up, the heat difference at node v_i between time t +Δ t and t will be equal to the sum of the heat that it receives, deducted by what it diffuses. Therefore the equation becomes,

$$\frac{f_i(t+\triangle t)-f_i(t)}{\triangle t} = \alpha\left[\tau_i f_i(t) + \Sigma \frac{w_{ij}}{\Sigma_{k:(j,k)\epsilon E} w_{jk}} f_i(t)\right]$$

(3.1)

Where τ_i is a flag to identify whether node v_i has any outlinks. Solving it we obtain

$$\text{f}(\ 1\ ) = e^{\alpha(H-D)}\text{f}\ (0)$$

where,

$$\text{H}_{ij} = \begin{cases} W_{ij}/\Sigma_{k:j,k\in E} W_{jk}, & (v_i, v_j) \in E \\ 0, & \text{i} = j, \\ 0, & otherwise. \end{cases}$$

And

$$\text{H}_{ij} = \begin{cases} \tau_i, & \text{i} = j, \\ 0, & otherwise. \end{cases}$$

3.4 Complexity Analysis

Supposing a graph is connected by M edges (relationships between nodes), the complexity of executing the heat diffusion process is O(PM), where P is a constant number which gives the number of iterations, multiplied by the number of edges M in a graph. In most cases, P = 10 is enough for approximating the heat diffusion equation. The complexity O(PM) shows that our heat diffusion algorithm enjoys very good performance in scalability since it is linear with respect to the number of edges in the graph.

However, since the size of Web information is very large, the graph built upon the Web information can become extremely large. Then, the complexity O (PM) is also too high, and the algorithm becomes time consuming and inefficient to get a solution. To overcome this difficulty, we first extract a subgraph starting from the heat sources. Given the heat sources, the subgraph is constructed by using depth-first search in the original graph. The search stops when the number of nodes is larger than a predefined number. Then, the diffusion processes will be performed on this subgraph efficiently and effectively. Generally, it will not decrease the qualities of the heat diffusion processes since the nodes too far away from the heat sources are normally not related to the sources.

Chapter 4

Empirical Analysis

So far graph diffusion models for recommendations have been introduced. In this section,

- Method to convert different web data sources into correct graphs
- Several experiments on query suggestion and image recommendations are discussed.

4.1 Query suggestion

Query Suggestion is a technique widely employed by commercial search engines to provide related queries to users information need. In this section, we demonstrate how our method can benefit the query suggestion, and how to mine latent semantically similar queries based on the users information need.

4.1.1 Data collection

Clickthrough data record the activities of Web users, which reflect their interests and the latent semantic relationships between users and queries as well as queries and clicked Web documents. Each line of clickthrough data contains the following information: a user ID (u), a query (q) issued by the user, a URL (l) on which the user clicked, the rank (r) of that URL, and the time (t) at which the query was submitted for search. Thus, the clickthrough data can be represented by a set of quintuples (u ,q ,l ,r ,t). From a statistical point of view, the query word set corresponding to a number of Web pages contains human knowledge on how the pages are related to their issued queries. Thus, in this paper, we utilize the relationships of queries and Web pages for the construction of the bipartite graph containing two types of vertices (q ,l). The information regarding user ID, rank and calendar time is ignored. This data set is the raw data recorded by the search engine, and contains a lot of noise which will potentially affect the effectiveness of our query suggestion algorithm. Hence, the data is filtered by only keeping those frequent, well formatted, English queries (queries which only contain characters "a," "b," ..., "z," and space).

4.1.2 Graph construction

For the query-URL bipartite graph, consider an undirected bipartite graph Bql =(V_{ql}, E_{ql}) , where V_{ql} = Q ∪ L , Q =$q_1, q_2, .q_n$) and L = $l_1, l_2, .l_n$. Eql = $(q_i, l_j$) — there is an edge from q$_i$ to l$_j$ is the set of all edges. The edge (q_j, l_k) exists if and only if a user u_i clicked a URL l_k after issuing a query q_j. Example , in the figure values on the edges in specify how many times a query is clicked on a URL. The bipartite graph extracted from the clickthrough data into the diffusion processes since this bipartite graph is an undirected graph, and cannot accurately interpret the relationships between queries and URLs. . Hence, this bipartite graph must be converted into another form. In this converted graph, every undirected edge in the original bipartite graph is converted into two directed edges. The weight on a directed query-URL edge is normalized by the number of times that the query is issued, while the weight on a directed URL-query edge is normalized by the number of times that the URL is clicked.

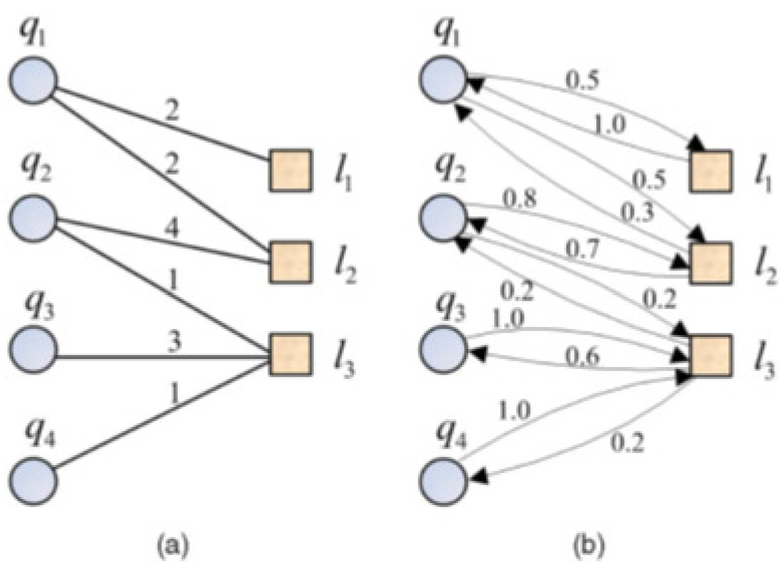
4.1.3 Query Suggestion Algorithm

After the conversion of the graph, the query suggestion algorithm can easily be designed.
Algorithm : 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 = 1, and without loss of generality, set the initial heat value of query q f_q (0) = 1 (the choice of initial heat value will not affect the suggestion results). Start the diffusion process using

$$f(1) = e^R f (0).$$

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



Graph construction for query suggestion. (a) Query-URL bipartite graph. (b) Converted query-URL bipartite graph.

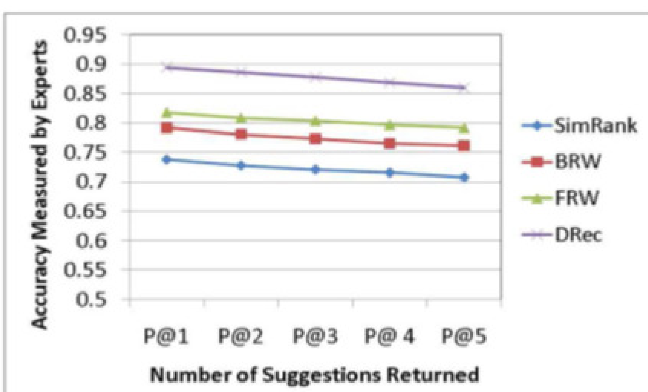
4.1.4 Query Suggestion Results

The suggestion results of our algorithm and those from Google, Yahoo!, Live Search, and AOL are displayed in the table. The algorithm discussed is called DRec which means Recommendations by Diffusion. The value of is set as 1 , and the size of subgraph is set as 5,000. From the resulting suggestions, its seen that the query suggestions generated by our method are generally as good as those from commercial search engines. For some queries, our suggestions are even better. In order to compare our method with other approaches, a set of 200 queries are set as the testing queries, covering a wide range of topics, such as Computers, Arts, Business, and others. Some of the results generated by our DRec algorithm are shown in the table below.

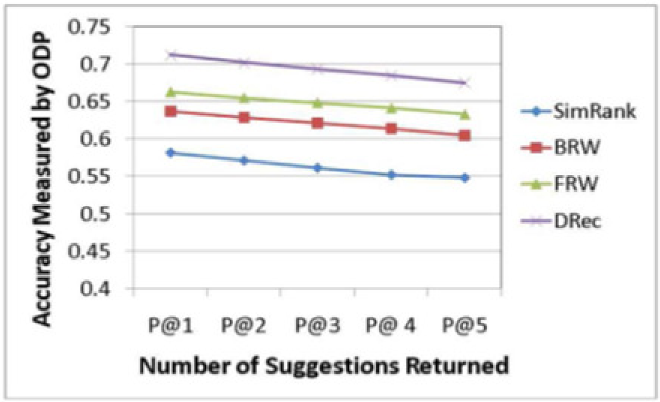
Testing Queries	Suggestions				
	Top 1	Top 2	Top 3	Top 4	Top 5
michael jordan	nba	nike	jordan xi	air jordans	michael jordan bio
java	sun java	java download	java updates	virtual machine	sun microsystems
apple	itunes	ipod	quicktime	apple ipod	apple stores
fitness	exercise	fitness magazine	muscle and fitness	mens fitness	weight loss
solar system	planets	jupiter	saturn	neptune	pluto
sunglasses	chanel sunglasses	oakley	maui jim sunglasses	designer sunglasses	oakley sunglasses
flower delivery	flowers	florist	gift baskets	cheap flowers	proflowers
wedding	wedding channel	wedding dresses	the knot	wedding plans	wedding poems
astronomy	apod	star charts	planets	solar system	skyandtelescope
real estate	remax	realtor	homes for sale	coldwell banker	houses for sale

Examples of DRec Query Suggestion Results (k = 50)

From the results, we observe that proposed recommendation algorithm not only suggests queries which are literally similar to the test queries, but also provides latent semantically relevant recommendations. For instance, if the test query is a technique, such as java, we recommend virtual machine and sun microsystems. The latter suggestion is the company who owns the Java Platform, and the former suggestion is a key feature of the Java programming language. They both have high latent semantic relations to the query java. If the test query is a human name, such as michael jordan, one of the most successful NBA basketball player, the latent semantic suggestions are nba, nike, jordan xi (a model of Air Jordan shoes), and air jordans, All of the results show that our latent semantic query suggestion algorithm has a promising future.



Accuracy comparisons measured by experts.



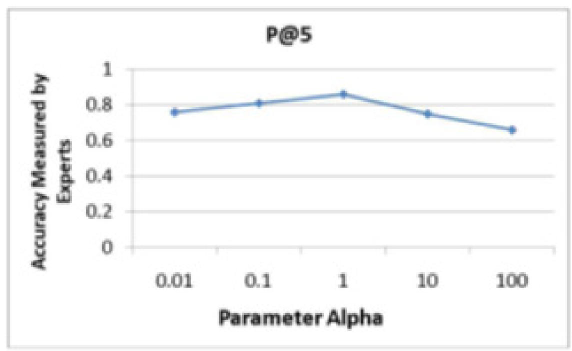
Accuracy comparisons measured by ODP.

For the automatic evaluation, the ODP database is used. ODP, also known as dmoz, is one of the largest, most comprehensive human-edited directories of the Web. . When a user types a query in ODP, besides site matches, categories matches can also be found in the form of paths between directories. Moreover, these categories are ordered by relevance. For instance, the query Java would provide the hierarchical category Computers : Programming : Languages : Java, where : is used to separate different categories. One of the results for Virtual Machine would be Computers : Programming : Languages : Java : Implementations. Hence, to measure how related two queries are, we can use a notion of similarity between the corresponding categories (as provided by ODP). Evaluating using ODP database , its observed that our proposed DRec algorithm increases the suggestion accuracy. This indicates again that the proposed query suggestion algorithm is very effective.

Query = aa					
DRec (with Heat Values)		Google	Yahoo!	Live Search	AOL
american airlines	0.0971	N/A	aa route planner	aa route finder	N/A
alcoholics anonymous	0.0294		aa route finder	aa route planner	
airlines	0.0193		aa autoroute	aa airlines	
airline tickets	0.0107		aa route map	aa route map	
american air	0.0069		aa roadwatch	aa meetings	
Query = travel					
DRec (with Heat Values)		Google	Yahoo!	Live Search	AOL
travelocity	0.0735	travel guide	yahoo travel	travelocity	aol travel
expedia	0.0310	travel magazine	travel agents	orbitz	travel channel
orbitz	0.0180	travelzoo	travel insurance	cheap tickets	liberty travel
airline tickets	0.0122	train travel	travel channel	expedia	american express travel
hotels	0.0036	last minute travel	travel inn	priceline	
Query = disease					
DRec (with Heat Values)		Google	Yahoo!	Live Search	AOL
bird flu	0.0118	N/A	lyme disease	list of diseases	lyme disease
hpv	0.0069		diseases	infectious diseases	parkinsons disease
cdc	0.0057		crohn's disease	rare diseases	celiac disease
scabies	0.0053		celiac disease	liver disease	crohns disease
center for disease control	0.0051		parkinson's disease	heart disease	
Query = pizza					
DRec (with Heat Values)		Google	Yahoo!	Live Search	AOL
pizza hut	0.0398	pizza recipe	pizza hut	pizza hut	pizza hut
dominos pizza	0.0283	pizza history	domino's pizza	pizza recipe	dominos pizza
california pizza kitchen	0.0134	making pizza	california pizza kitchen	pizza hut coupons	papa johns pizza
papa johns	0.0124	pizza games	round table pizza	pizza delivery	dominoes pizza
round table pizza	0.0082	homemade pizza	cici's pizza	pizza history	
Query = free music					
DRec (with Heat Values)		Google	Yahoo!	Live Search	AOL
free music downloads	0.0522	free online music	free music downloads	listen to free music	free music downloads
napster	0.0102	free ipod music	free music videos	free music downloads	free sheet music
limewire	0.0080	free music for myspace	free music lyrics	free radio	free music video downloads
music downloads	0.0062	free country music	legal free music downloads	limewire	
musicmatch	0.0039	free rap music	free music video codes	yahoo music	
Query = windows					
DRec (with Heat Values)		Google	Yahoo!	Live Search	AOL
microsoft	0.0282	house windows	windows media player	windows xp	windows update
windows update	0.0219	anderson windows	windows live messenger	house windows	windows media player
anderson windows	0.0202	pella windows	windows update	windows update	windows live
microsoft updates	0.0104	windows and doors	windows xp	windows live messenger	
internet explorer	0.0094	windows media player	windows vista	windows vista	
Query = job					
DRec (with Heat Values)		Google	Yahoo!	Live Search	AOL
monster	0.0266	job bible	job search	part time jobs	job search
americas job bank	0.0127	book of job	job centre	monster job	job listings
career builder	0.0119	teen job	job bank	job search	usa jobs
monster jobs	0.0077	story of job	job descriptions	government jobs	monster jobs
job search	0.0069	part time jobs	job corps	ups jobs	
Query = pets					
DRec (with Heat Values)		Google	Yahoo!	Live Search	AOL
petfinder	0.0165	neopets	pets at home	pets for sale	pet smart
yahoo pets	0.0144	types of pets	pets for sale	free pets	pet finder
petsmart	0.0122	pictures of pets	virtual pets	pet adoption	pet meds
neopets	0.0113	reptile pets	wonder pets	pets health	pet supplies
avma	0.0063	peta	sims 2 pets	pet suppliers	
Query = president					
DRec (with Heat Values)		Google	Yahoo!	Live Search	AOL
george washington	0.0110	N/A	president bush	list us presidents	barack obama
white house	0.0106		president obama	presidents in order	presidents day
abraham lincoln	0.0094		hillary clinton for president	president facts	us presidents
us presidents	0.0067		president bill clinton	president election	
thomas jefferson	0.0062		alan rosenberg president	presidents for kids	
Query = news					
DRec (with Heat Values)		Google	Yahoo!	Live Search	AOL
msnbc	0.0408	bbc news	false news	fox news	fox news
cnn	0.0129	sports	news yahoo	cnn news	new york daily news
bbc	0.0127	science news	fox news	abc news	cnn news
google news	0.0123	india news	bbc news	local news	buffalo news
fox news	0.0088	entertainment news	cnn news	yahoo news	

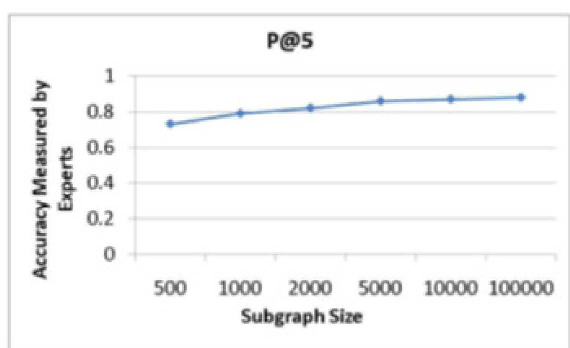
4.1.5 Impact of Parameter

The parameter plays an important role in the proposed model. It controls how fast heat will propagation on the graph. Hence, experiments are conducted on evaluating the impact of parameter .Results show that the best possible value is 1. If the value is relatively small the performance will drop since some relevant nodes cannot get enough heat. On the other hand, if value is relatively larger , the performance will also decrease. This is because if the heat transfers very fast, some irrelevant nodes will gain more heat, hence will hurt the performance.



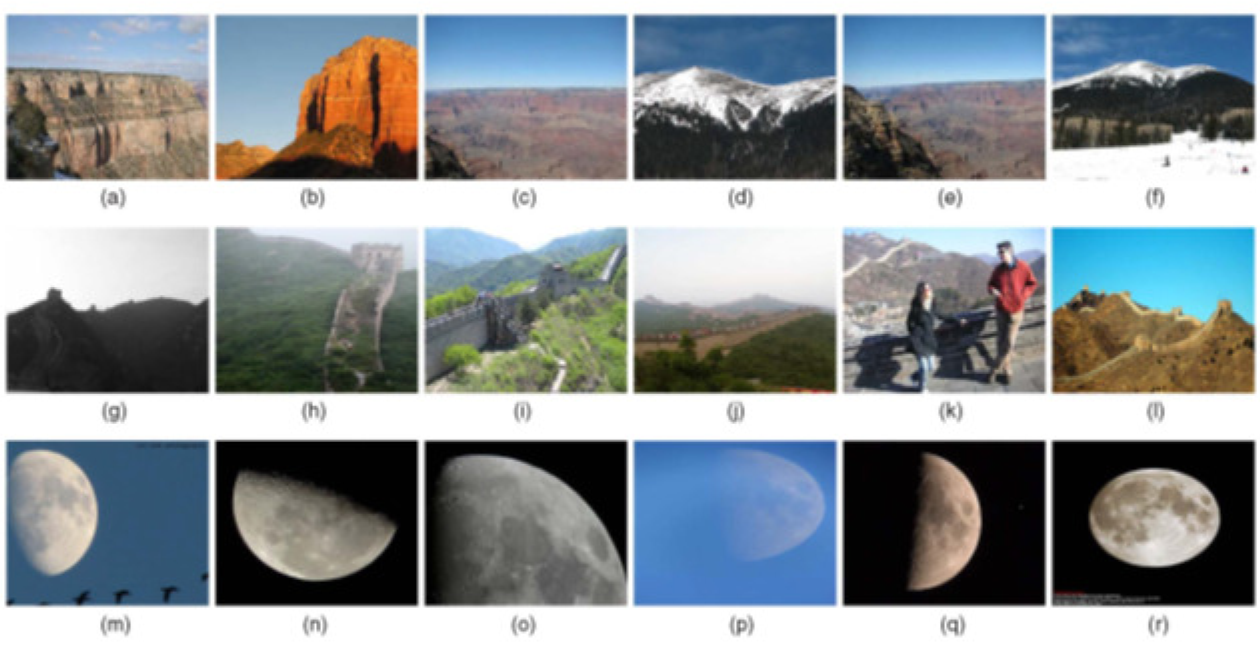
4.1.6 Impact of the Size of Subgraph

As mentioned above, when the size of the web graph is huge, the algorithm is performed on a subgraph extracted from the original graph. Its observed that when the size of the graph is very small, like 500, the performance of our 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



4.2 Image Recommendation

Image recommendation is used to search and retrieve images on the Web. In this model an image to image recommendation algorithm is discussed. Basically, the graph construction for image recommendation is similar to the one introduced before. The only difference is that here the nodes in bipartite graph are images and tags, respectively. By using the similar algorithm which is introduced before, image recommendations can also be provided.



4.2.1 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, the above method can be easily extended to the personalized image recommendations. In the query suggestions discussed above in and image suggestions performed in this section, we only employ one node (either a query or an image) as the heat source. In the personalized image recommendations, we can set all the images submitted by a specified user as the heat sources, and then start the diffusion process. This ensures that the suggested images are of interests of this user.

Chapter 5

Conclusion

5.1 Conclusion

In this paper, a novel framework for recommendations on large scale Web graphs using heat diffusion is presented. This is a general framework which can basically be adapted to most of the Web 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 Web data sources shows the promising future of this approach.

5.2 Future Scope

The heat values of suggested queries are listed. These values not only can be used in query suggestions, but also are very informative in the advertisement when customers bid for query terms. Actually, since the diffusions are between all the nodes in the graph (including the nodes representing queries and the nodes representing URLs), all the URLs also have heat values. Hence, it is easy to infer that, for a given query, after the diffusion process, the heat values of URLs represent the relatedness to the original query, which can also be employed as the ranking of these URLs. This ranking actually is the wisdom of the crowd since it is based on the query-URL click data, which reflects the intelligent judgements of the Web users. This method can also be used for social recommendations. Since the model is quite general, we can apply it to more complicated graphs and applications. Social recommendation, which produces recommendations by incorporating users social network information, is becoming to be an indispensable feature for the next generation of Web applications. This include two different data items, which are social network and user- item relation matrices. In the social network graph, there are trust scores between different users, while in the user-item relation matrix, binary relations connect users and items. We can convert these two graphs into a single and consistent one, using the method discussed before. With the constructed graph, for each user (heat source), we can start the diffusion process and then recommend the Top-N items to this user.

References

[1] Chuliang Weng, Member, IEEE, Minyi Guo, Yuan Luo, and Minglu Li: Hybrid CPU Management for Adapting to the Diversity of Virtual Machines: IEEE Transactions on Computers, Vol. 62, No. 7, JULY 2013

[2] Sriram Govindan, Jeonghwan Choi, Arjun R. Nath, Amitayu Das, Bhuvan Urgaonkar, and Anand Sivasubramaniam: Xen and Co.: Communication-Aware CPU Management in Consolidated Xen-Based Hosting Platforms: IEEE Transactions on Computers, Vol. 58, No. 8, AUGUST 2009

[3] Vijay K. Naik: Performance of NAS Parallel Application-Benchmarks on IBM SP1: IEEE 1994

[4] http://www.xen.org/: Xen Hypervisor