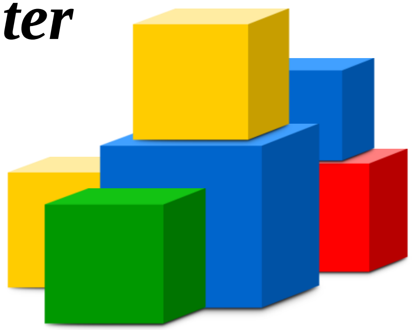
A Query Weighted-based Method for User Modeling

Hu Juan， Bai Yu, Cai Dongfeng

Knowledge Engineering Research Center
Shenyang Aerospace University



Outlines

- Background
- Query Weighted-based User Modeling
- Experiments and Results

Outlines

- Background
- Query Weighted-based User Modeling
- Experiments and Results

Background

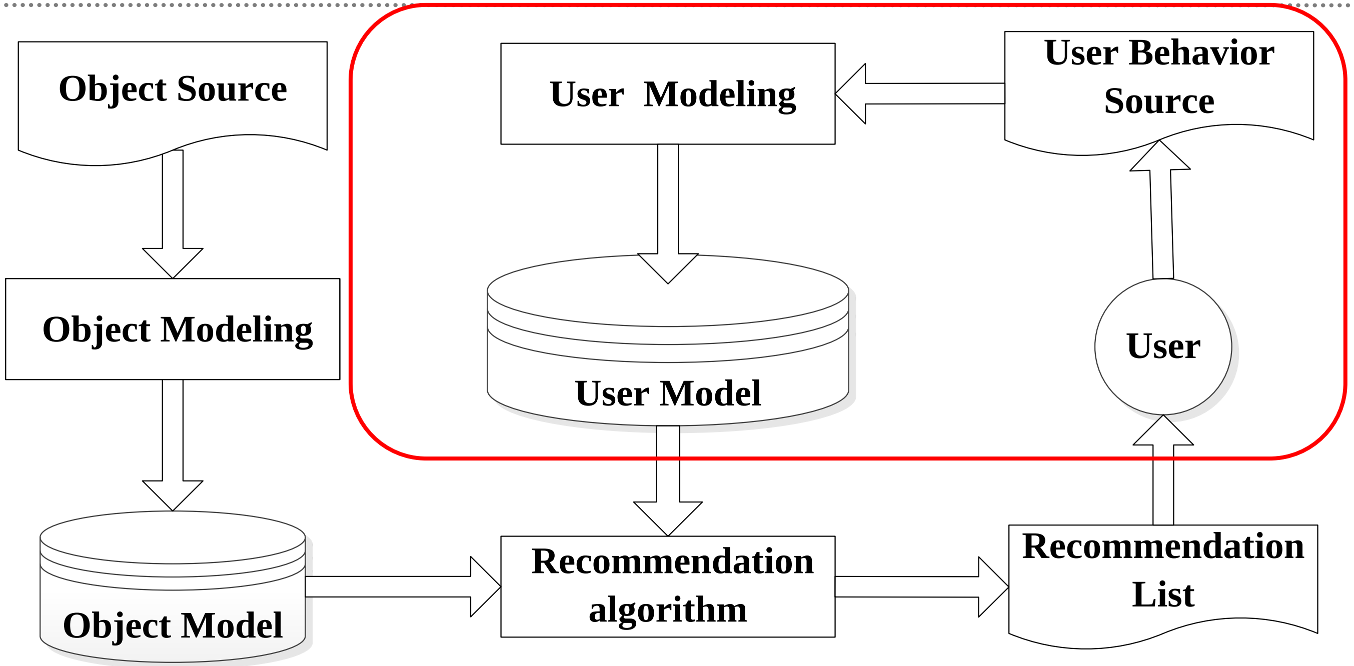
- With the rapid growth of Internet-scale，information overload is an increasing problem for web users.
- Recommendation system is one of the most promising approaches to solve the problem of information overload.
- A personalized recommendation system can divide into three parts:

  User interest modeling

  Recommendation object modeling

  Recommendation algorithm

Background



The recommendation system

- Object source by the object modeling methods obtain the object model
- User behavior by the user modeling methods generate the user model
- Combining the user model and object model to obtain recommender list, and then return to user

Background

- User Modeling：

is a process of obtaining and maintaining the user interest, needs and habits, and generates user model that can reflect the users' specific interest.

- The purpose of user modeling are:

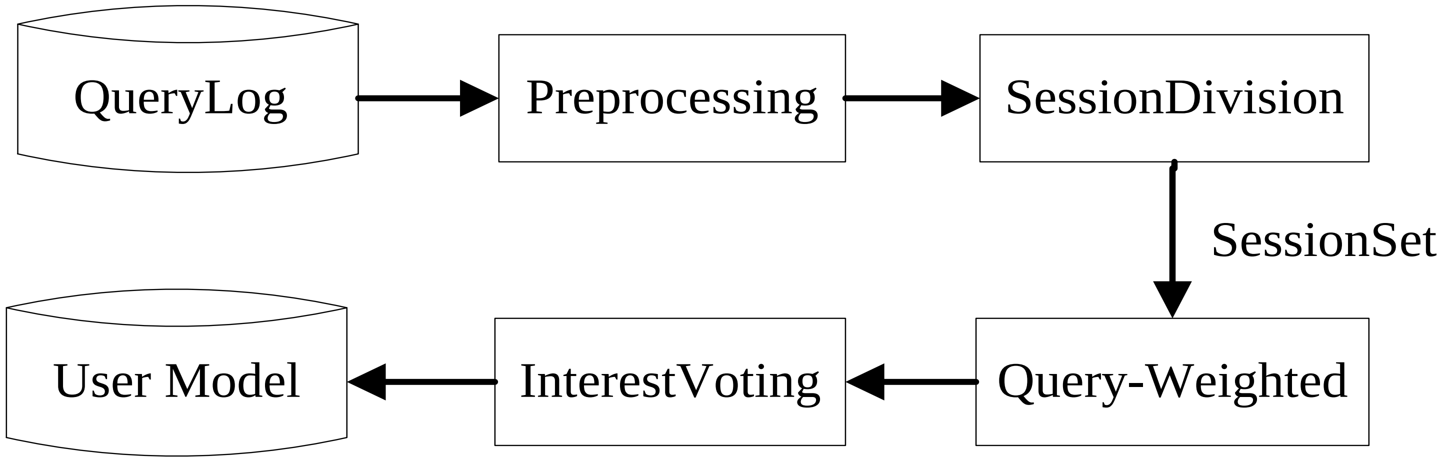
(1)Mining user interests →Query Weighted-based user Modeling

(2) Representing user model →Set of keywords

Outlines

- Background
- Query Weighted-based User Modeling
- Experiments and Results

Query Weighted-based User Modeling



The framework of user modeling

- We preprocess the query log
- The second step is session division, and we obtain session set for each user
- For each user, we use the query weighted method to get the weight of each query in a session
- The last step is interest voting, and then we get user model

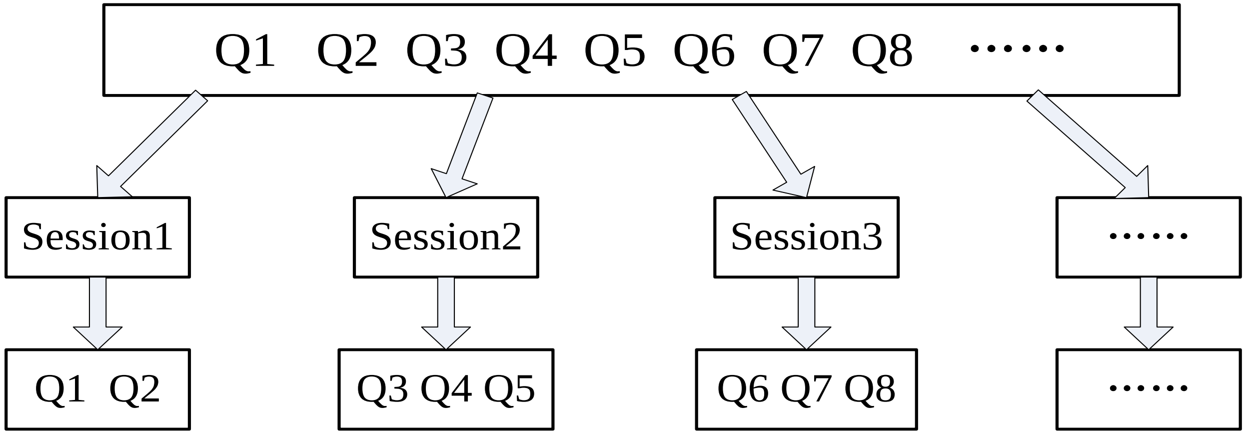
Preprocessing

The preprocessing of query log:

- Splitting the query log by user, put the query log of same user together;

- Filtering the users by the number of query log is more than the threshold.

Session Division



The framework of session division

The principles of session division:

(1)The time interval of a session <= session time threshold

(2)The time interval between adjacent queries in a session <= query time threshold

(3)The cosine similarity between adjacent queries >= query similarity threshold

Based on: Mining user web search activity with layered bayesian network or how to capture a click in its context. (2009)

Session Division

The session sample of an user:

Session	Query	QueryTime	Rank	ClickURL
Session1	midway online literary journal	2006-04-21 11:24:43	3	http://www.mndaily.com
	midway online literary journal	2006-04-21 11:24:44	9	http://www.smallspiralnotebook.com
	meridian literary magazine	2006-04-21 11:38:21	2	http://www.engl.virginia.edu
	meridian literary magazine	2006-04-21 11:38:25	6	http://www.fglaysher.com
Session2	mark twain middle school	2006-04-21 14:38:23	2	http://www.fcps.k12.va.us
	mark twain middle school	2006-04-21 14:38:27	1	http://www.fcps.k12.va.us
Session3	university of massachusettes mfa blog	2006-04-22 07:22:43	3	http://www.thepublishngspot.com
	university of massachusettes mfa blog	2006-04-22 07:22:48	5	http://www.pitt.edu
	university of massachusettes mfa blog	2006-04-22 07:22:49	10	http://snreview.wordpress.com
	university of massachusettes mfa blog	2006-04-22 07:29:42	15	http://www.myspace.com
	university of massachusettes mfa blog	2006-04-22 07:29:45	24	http://maudnewton.com
	babies are fireproof	2006-04-22 07:31:22	1	http://babiesarefireproof.blogspot.com

Query Weighted

We proposed three hypotheses :

The query weight is bigger when:

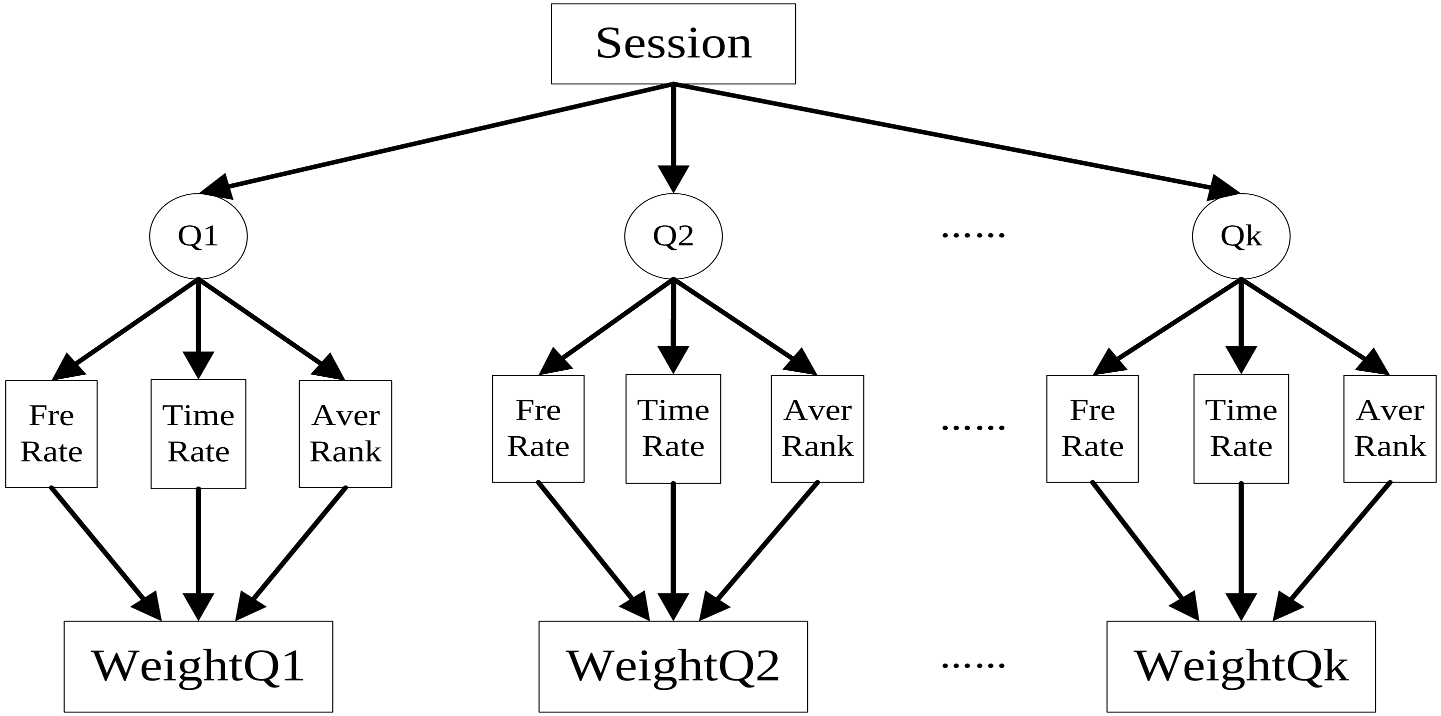
(1) the query occurs more times in a session;

(2) the query average duration in a session last longer;

(3) the url average rank of the query in a session is higher.

Query Weighted

The framework of query weighted



A session contains Q1Q2……Qk queries, for each query we calculate the *FreRate*, *TimeRate* and *AverRank*, and then, we get the weight of a query.

Query Weighted--*FreRate*

$FreRate_{Q_{kj}}$: the rate of query Q_{kj} occurrence times in session S_k

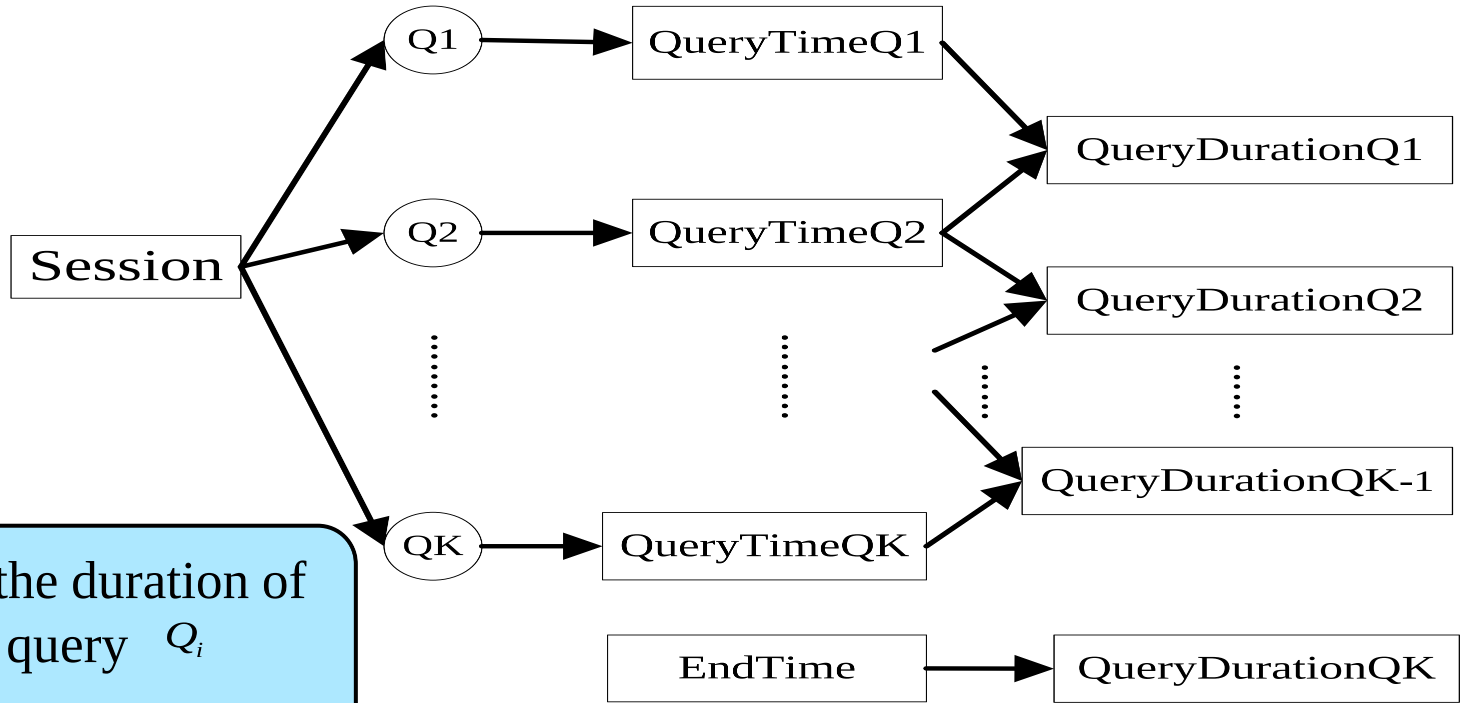
$$FreRate_{Q_{kj}} = \frac{Fre_{Q_{kj}}}{Q}$$

Fre_{Q_j} : the occurrence times of query Q_{kj} in session S_k

Q : the total number of query in session S_k

Query Weighted--*TimeRate*

the query stream, sorted by time: $\{Q_1, Q_2, \ldots, Q_i, \ldots, Q_K\}$



the duration of query Q_i

$$QueryDuration_{Qi} = \begin{cases} QueryTime_{Qi+1} - QueryTime_{Qi} & 1 \le i < K \\ EndTime & i = K \end{cases}$$

The time of query Q_i

$$EndTime = \begin{cases} 10s & (Q_K \text{ didn't click url}) \\ 60s & (Q_K \text{ clicked url}) \end{cases}$$

Query Weighted--*TimeRate*

$TimeRate_{Q_{kj}}$: the rate of the average duration of query Q_{kj} in session S_k

$$TimeRate_{Q_{kj}} = \frac{\overline{QueryDuration_{Q_{kj}}}}{SessionTime_{S_k}}$$

$$\overline{QueryDuration_{Q_{kj}}} = \frac{\sum^{Fre_{Q_{kj}}} QueryDuration_{Q_{kj}}}{Fre_{Q_{kj}}}$$

$\overline{QueryDuration_{Q_{kj}}}$: the average duration of query Q_{kj} in session S_k

$SessionTime_{S_k}$: the total duration of session S_k

Query Weighted--*AverRank*

$AverRank_{Q_{kj}}$:the reciprocal of the average clicked URL rank of query Q_{kj} in session S_k

$$AverRank_{Q_{kj}} = \frac{Fre_{Q_{kj}}}{\sum^{Fre_{Q_j}} Rank_{Q_{kj}}}$$

$Rank_{Q_{kj}}$: each clicked URL rank of query Q_{kj}

Query Weighted

$W_{Q_{kj}}$: the weight of query Q_{kj} in session S_k

$$W_{Q_{kj}} = \alpha * FreRate_{Q_{kj}} + \beta * TimeRate_{Q_{kj}} + \gamma * AverRank_{Q_{kj}}$$

$$\alpha + \beta + \gamma = 1$$

$$0 \le \alpha \le 1 \quad 0 \le \beta \le 1 \quad 0 \le \gamma \le 1$$

Query Weighted—Interest Voting

Calculating the weight of each word in each user's query log.

We should preprocess the query as follow:

- Splitting words by white space
- Removing the stop words and the noise words
- Stemming by Porter

$$W_{T_i} = Vote\,(T_i) = \sum_{k}^{K_i} \sum_{j}^{N_{ki}} \left(W_{Q_{kj}} * F_{ij} \right)$$

F_{ij} : the occurrence times of keyword in query

Keyword T_i occurred in K_i sessions , and occurred in N_{ki} queries in session S_k

And we can represent the user model:

$$UserInterest = \{(T_1, W_{T_1})(T_2, W_{T_2})\ldots\ldots(T_{T_M}, W_{T_M})\}$$

Outlines

- Background
- Query Weighted-based User Modeling
- Experiments and Results

Dataset

Dataset: AOL query log (http://www.datatang.com/data/42724)

- Time: 2006-03-01~2006-05-31
- Form: {UserID, Query, QueryTime, Rank, ClickURL}
- Users: 657,426
- Records: 10,154,742
- We used: 376 users(query number is bigger than 20 both in training set and test set)
- Training set: 2006-03-01~2006-05-15
- Test set: 2006-05-16~2006-05-31

Evaluation

Evaluation Design:

- The preprocessing work in test set
- Representing the test set as a keyword set to be considered as the real interest word set of user
- Comparing the real interest word set and the user model by the evaluation metrics

Evaluation Metrics:

- **MeanP: the mean of user prediction precision**

The value of MeanP is higher, the prediction precision of user model is higher

- **MAP: the mean of the average of each user precision**

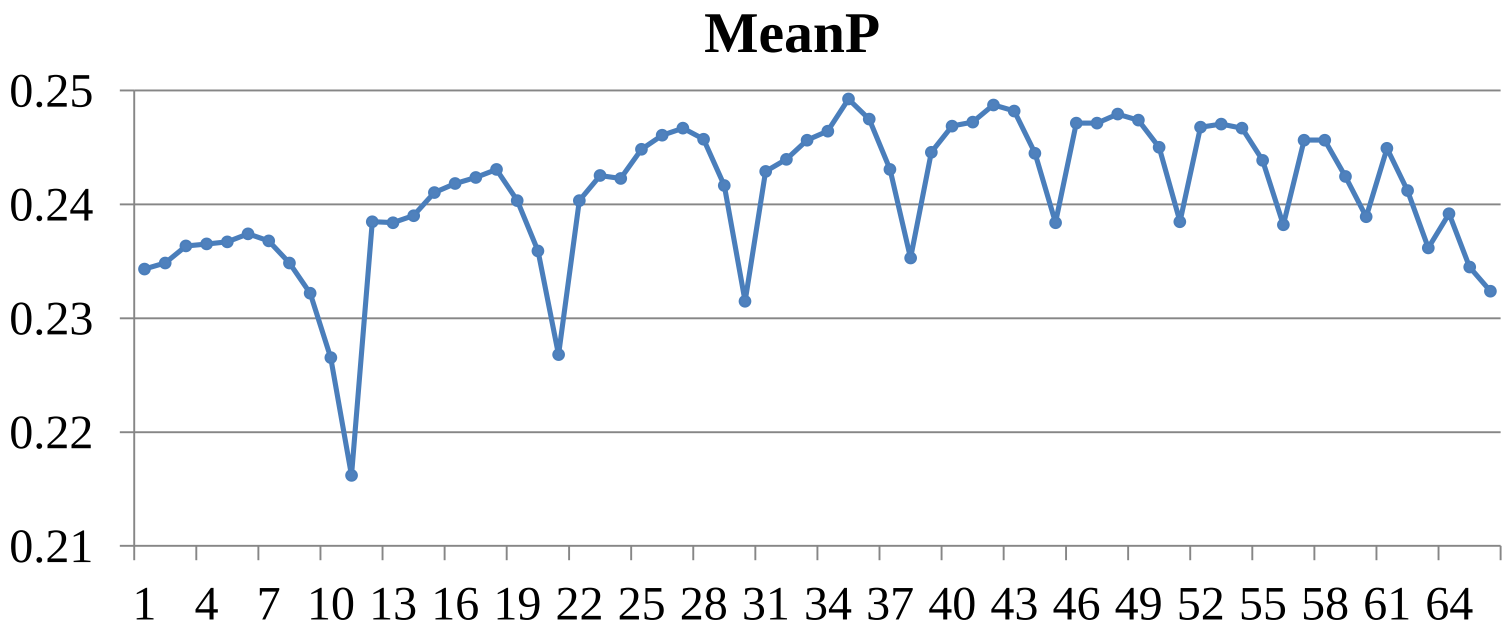
The value of MAP is higher, the higher ranks of the successful prediction interests

$$MeanP = \frac{1}{|U|}\sum_{u}^{|U|}\frac{\Pr eNum_u}{M_u} \qquad MAP = \frac{1}{|U|}\sum_{u}^{|U|}\frac{1}{N_u}\sum_{m}^{N_u} Precision(R_{um})$$

Parameter Estimation

The purpose is to confirm the value of α, β, γ

- Set the step value is 0.1， obtain each corresponding values of MeanP and MAP



The figure is the 65 experiments of different of α, β, γ , and at the experiment 36, the value of *MeanP* is the highest

Parameter Estimation

In order to verify the effects of the three features:

- The 1,2,3 are the effects of only one feature, the 4,5,6 are the effects of two features, the 7 is the effect of all three features

Experiment	**Feature Selection**	***MeanP***	***MAP***
1	$FreRate\ (\alpha = 1.0, \beta = 0.0, \gamma = 0.0)$	0.23237	0.26883
2	$TimeRate\ (\alpha = 0.0, \beta = 1.0, \gamma = 0.0)$	0.21619	0.25081
3	$AveRank\ (\alpha = 0.0, \beta = 0.0, \gamma = 1.0)$	0.23432	0.27096
4	$TimeRate\ \&\ AveRank\ (\alpha = 0.0, \beta = 0.5, \gamma = 0.5)$	0.23742	0.27569
5	$FreRate\ \&\ AveRank\ (\alpha = 0.5, \beta = 0.0, \gamma = 0.5)$	0.24714	0.28657
6	$FreRate\ \&\ TimeRate\ (\alpha = 0.7, \beta = 0.3, \gamma = 0.0)$	0.23892	0.27581
7	$FreRate\ \&\ TimeRate\ \&\ AveRank\ (\alpha = 0.4, \beta = 0.3, \gamma = 0.3)$	**0.24873**	**0.28844**

In this table, we can get the result: when three features were used at the same time, the values of *MeanP* and *MAP* are both the highest;

So we considered the result of $\alpha = 0.4, \beta = 0.3, \gamma = 0.3$ **as user model**

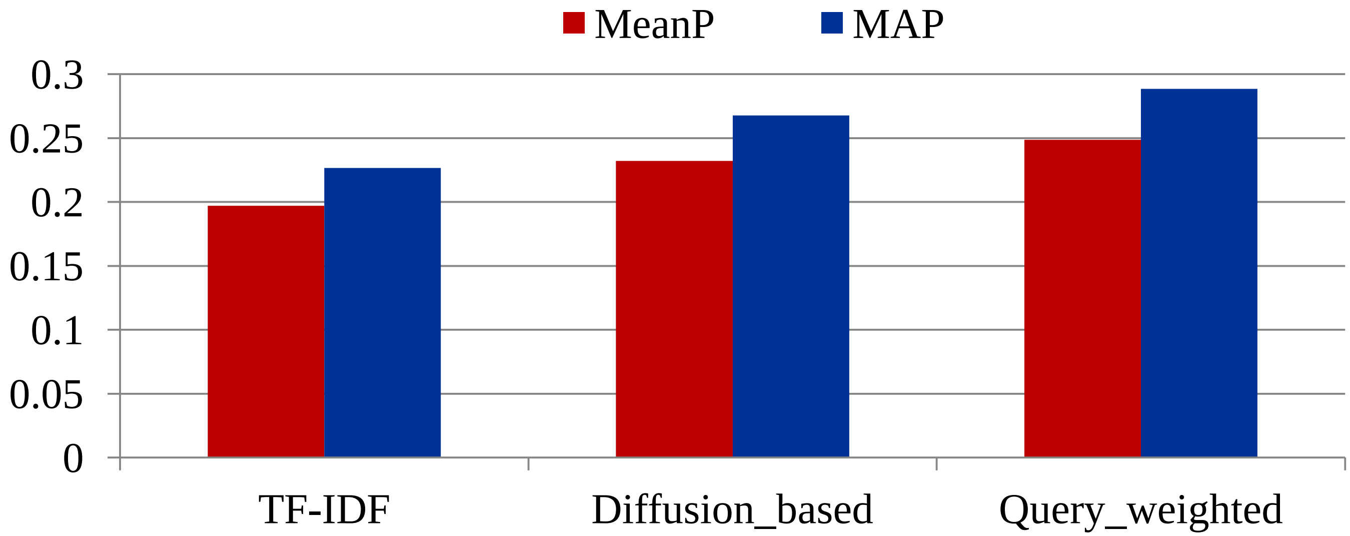
Experiments

Method 1: considering the user query log as documents, and calculating the TF-IDF value of each word .(**TF-IDF**)

Method 2: Weighted the bipartite graph, imported the diffusion theory, and then, recourses were allocated to realize the prediction of user behavior and generate user model.(**Diffusion-based**)

Method 3: the query weighted-based user modeling(**Query-weighted**)

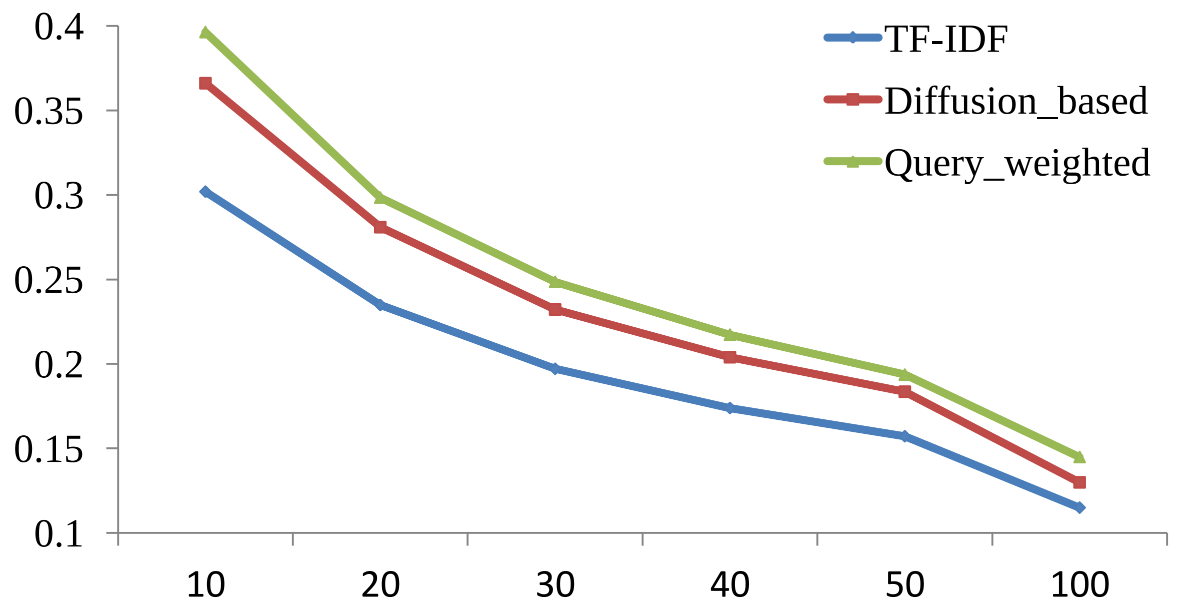
Experiments



The figure show the values of MeanP and MAP of each method,

It shows that our method is better than the TF-IDF Method and Diffusion Method

Experiments



This figure show that the different return number of interest of an user correspond with the value of *MeanP*, and the Query_weighted Method is the best all the time.

Conclusions

- We proposed a query weighted-based method for user modeling;
- The experiments show the effectiveness of the three hypotheses;
- The results show that user behavior reflected user interests, user modeling are not only the user contents modeling, but also the user behavior modeling;
- The method just considered the single user information, the information between the user and the user were not included;
- The future work is taking the information between the user and the user into account, and to obtain better prediction.

Thank you!