Continuous Top-k Query for Graph Streams

Shirui Pan and Xingquan Zhu
Centre for Quantum Computation and Intelligent Systems
Faculty of Engineering and Information Technology
University of Technology, Sydney, NWS 2007, Australia
Shirui.Pan@student.uts.edu.au, Xingquan.Zhu@uts.edu.au

ABSTRACT

In this paper, we propose to query correlated graphs in a data stream scenario, where an algorithm is required to retrieve the top k graphs which are mostly correlated to a query graph q. Due to the dynamic changing nature of the stream data and the inherent complexity of the graph query process, treating graph streams as static datasets is computationally infeasible or ineffective. In the paper, we propose a novel algorithm, Hoe-PGPL, to identify top-k correlated graphs from data stream, by using a sliding window which covers a number of consecutive batches of stream data records. Our theme is to employ Hoeffding bound to discover some potential candidates and use two level candidate checking (one corresponding to the whole sliding window level and one corresponding to the local data batch level) to accurately estimate the correlation of the emerging candidate patterns, without rechecking the historical stream data. Experimental results demonstrate that the proposed algorithm not only achieves good performance in terms of query precision and recall, but also is several times, or even an order of magnitude, more efficient than the straightforward algorithm with respect to the time and the memory consumption. Our method represents the first research endeavor for data stream based top-k correlated graph query.

Categories and Subject Descriptors

H.2.8 [**Data Management**]: Database Applications—*Data Mining*

General Terms

Algorithm

Keywords

Correlated Graph Query, Graph Stream, Pearson's Correlation Coefficient

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
CIKM'12, October 29–November 2, 2012, Maui, HI, USA.
Copyright 2012 ACM 978-1-4503-1156-4/12/10 ...$15.00.

1. INTRODUCTION

Correlation mining is a crucial data mining task due to its uniqueness and advantage for revealing underlying dependency between objects. Recent years have witnessed extensive studies on correlation mining in various applications including market transaction databases [6] and quantitative databases [4]. This topic has been recently extended to graph databases, where data records or instances are linked through structural relationships [3, 5].

Correlation between two graphs measures the similarity of the pair's occurrence distributions. Given a graph database and a query graph q, correlated graph search (CGSearch) [3] intends to discover subgraph structures whose Pearson's correlation coefficients with q are larger than a given threshold θ. While discovering this type of correlated subgraphs patterns is crucial in many graph representation scenarios, it requires users to specify a threshold. In practice, it is difficult to decide a proper threshold because a low threshold value may result in a large number of correlated graphs and a high value may result in very few answers. Recently a top-k version of algorithm [5] is proposed to identify top-k correlated subgraphs from a database that have the highest correlation with q.

The above two works [3] [5] are limited to static graph databases. In practice, applications may involve data which constantly change or evolve over time. For instance, E-commerce web sites, such as Amazon, may have thousands of users every day. Each user's browsing history can be represented as a graph and the browsing patterns may evolve from time to time and vary from users to users. Correlated graph analysis from user browsing streams provides web site owners power to understand the users in a dynamic and instantaneous way. Unfortunately, to the best our knowledge, there is no existing research to query correlated graphs in data streams.

In this paper, we investigate top-k correlated graph query for data streams, with the following identified challenges:

- Each subgraph is a candidate of correlated graph to the query q and there are an exponential number of subgraphs in the graph databases. The challenge is further aggravated as computing the correlation for a candidate involves subgraph isomorphism which is NP-complete.
- The correlation between graphs is constantly changing over stream and recomputing all the correlation is time-consuming.
- The streaming scenario requires the algorithm to return answers in a timely fashion.

A straightforward approach to solve our problem is exhaustive search, which uses a sliding window to scan the stream and computes the correlation by employing CGSearch [3] or TopCor [5] algorithms to query the top-k correlated graphs in each sliding window. While this exhaustive approach can ensure complete and correct results, it is computationally inefficient because the reoccurring query process in each window requires a significant amount of computational costs. In this paper, we proposes a Hoeffding bound [2] based algorithm. More specifically, we utilize a Hoeffding bound to discover a set of top-k potential candidate graphs from each data batch. Then we design a *global-local check* scheme with two levels of lists to store candidate information, *i.e.*, potential global list (PG) and potential local lists (PLs). By maintaining and integrating the local lists and the global list carefully, we are able to estimate the true correlation of each candidate with small bias. Experimental results on chemical graph data stream demonstrate that our algorithm is several times more efficient than an exhaustive search method *w.r.t.* the system runtime and the memory consumption, and also achieves good performance in terms of query precision and recall.

2. PRELIMINARIES AND PROBLEM DEFINITION

2.1 Preliminaries

We consider undirected, labeled connected graphs in this paper. A graph is denoted as $g = (V, E, \ell)$, where V is the vertices set, E is the edge set, and ℓ is a labeling function to assign labels to a node or an edge. Given two graphs $g_1 = (V_1, E_1, \ell_1)$ and $g_2 = (V_2, E_2, \ell_2)$, a subgraph isomorphism from g_1 to g_2 is an injective function f: $V_1 \to V_2$, such that $\forall (u, v) \in E_1$, we have $(f(u), f(v)) \in E_2$, $\ell_1(u) = \ell_2(f(u))$, $\ell_1(v) = \ell_2(f(v))$, $\ell_1(u, v) = \ell_2(f(u), f(v))$. It is known that subgraph isomorphism is an NP-complete problem [1]. A graph g_1 is a subgraph of another graph g_2 ($g_1 \subseteq g_2$) if there is a subgraph isomorphism from g_1 to g_2.

Given a batch of graphs G_i, the projected dataset with respect to a graph g is denoted by $S_g^{G_i} = \bigcup\{g' | g \subseteq g', g' \in G_i\}$, whose frequency N_g is denoted by $N_g = |S_g^{G_i}|$, where $|\bullet|$ denotes the cardinality of the set $\bullet$. The joint frequency between two graphs g and q in G_i is the number of graphs in G_i which contains both graph g and q, denoted by $N_{gq} = |S_g^{G_i} \bigcap S_q^{G_i}|$.

2.2 Problem Definition

Given a query graph q and a continuous graph stream G, we emphasize on discovering top-k correlated graphs between q and G. Because G represents a dynamic graph stream which is constantly changing, we assume that graph data arrive batch by batch, and use a sliding window $D = \{G_{i-w+1}, G_{i-w+2}, \cdots, G_i\}$ to denote a consecutive region of the graph stream, where $G_j, i - w + 1 \leq j \leq i$ represents a batch of graphs and G_i is the most recent batch of the graph stream. Then our problem is to continuously monitor and report the top-k correlated graphs with the highest Pearson's correlation with q in a sliding window (the most recent w batches). A typical top-k correlated graph query in a data stream with window size w=3 is shown in Fig. 1.

Definition 1. **Pearson's Correlation Coefficient:** Pearson's Correlation Coefficient [6] between two graphs g and q over a number of N graphs is defined in Eq.(1).

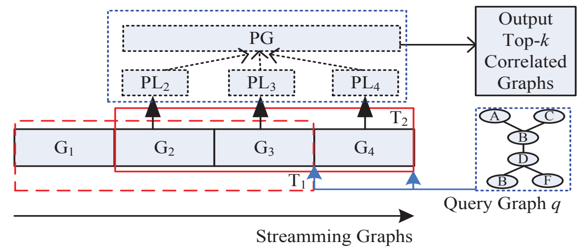

Figure 1: A sliding window based top-k correlated graph query for data streams. At time point T_1, the sliding window (dashed red rectangle) covers batches G_1, G_2, and G_3. A new batch G_4 arrives at time point T_2 (G_4 is the most recent batch), the sliding window updates to cover G_2, G_3, and G_4 (solid red rectangle). We use two levels of candidate lists to store the potential candidates, *i.e.*, PLs for potential local lists and PG for potential global lists, and output the answers from PG in each sliding window.

$$\phi(g, q) = \frac{NN_{gq} - N_g N_q}{\sqrt{N_g(N - N_g)N_q(N - N_q)}} \quad (1)$$

In Eq.(1), N_g, N_q, and N_{gq} denote, for the N graphs, the number of graphs containing g, q, and g and q, respectively. For clarity of presentation, when these variables are collected from the whole window D, the correlation between g and q is denoted as $\phi_D(g, q)$; when they are computed over a local batch of graphs (*e.g.*, G_j), the correlation is represented as $\phi_L(g, q)$.

3. PROPOSED METHOD

A straightforward method to solve our problem is to recompute all correlations for each sliding window, which is computationally inefficient for data streams. In this paper, we utilize Hoeffding bound [2] to discover a set of candidates in each batch, and employ a *global-local check* scheme with two levels of candidate lists to maintain these potential graphs.

3.1 Hoeffding Bound for Candidate Generation

Suppose there is a sequence of independently draw samples $\{\phi_1, \phi_2, \cdots, \phi_n\}$, each within in range $R = [a, b]$. Hoeffding bound [2] gives certain probabilistic guarantees on the estimation of statistics about the underlying data. More specifically, let r be the expectation of these points, and the estimated mean $\bar{r} = \frac{1}{n}\sum_{i=1}^{n} \phi_i$. Hoeffding bound states, for any ϵ,

$$\Pr\{|r - \bar{r}| \geq \epsilon\} \leq 2e^{-\frac{2n\epsilon^2}{R^2}} \quad (2)$$

Let the right side of Eq.(2) be $\delta = 2e^{-\frac{2n\epsilon^2}{R^2}}$, Hoeffding bound indicates that with probability $\leq \delta$, the estimated mean $\bar{r}$ is beyond $\pm\epsilon$ of r; or in other words, with confidence $1 - \delta$, the estimated mean $\bar{r}$ is within ϵ of the expectation. From above equation, we have

$$\epsilon = \sqrt{\frac{R^2 \ln(2/\delta)}{2n}} \quad (3)$$

Algorithm 1: Hoe-PGPL(G_i, PG, PLs, k, w, m)

1 $S_q^{G_i} = \bigcup\{g|q \subseteq g, g \in G_i\}, N_q = |S_q^{G_i}|$;
2 Get top-k correlated graphs in batch G_i from $S_q^{G_i}$;
3 Get the k^{th} correlation value ϕ_L and error ϵ_s;
4 $PL_i \leftarrow \bigcup\{g|\phi_L(g,q) \geq \phi_L - 2\epsilon_s, g \subseteq g', g' \in G_i\}$;
5 **foreach** $g \in PG$ **do**
6   Increasing N_g, N_{gq}, and N_q;
7   **if** *some batch is deleted* **then**
8   Decrease N_g, N_{gq}, and N_q;
9 $\boldsymbol{Merge(PL_i, PG, PLs)}$;
10 Re-compute the top-k correlated graphs based on PG, adding them to A_g;
11 **if** $|PG| > m * k$ **then**
12   Prune PG;
13 **return** A_g;

In our problem settings, we assume the correlation over data stream is a random variable, and stream data arrive batch by batch. In each batch we estimate $\bar{r} = \overline{\phi_L(g,q)}$, whose expected correlation value in a sliding window (denoted by $\phi_D(g,q)$) is at least $\overline{\phi_L(g,q)} - \epsilon$ with high probability $1-\delta$. Specifically, the $\overline{\phi_L(g,q)}$ is computed in a batch over a number of $|G_i|$ graphs, and we assume this computed result can be viewed as an average of $\phi_L(g,q)$.

Based on the Hoeffding bound, we mine a set of potential top-k correlated graphs in each batch over stream. For each candidate g, we have $\phi_L(g,q) > \phi_L - 2\epsilon_s$. Here, $\epsilon_s = \sqrt{\frac{R^2 \ln(2/\delta)}{2|G_i|}}$, and ϕ_L is the k^{th} smallest correlation value with q (all candidates are sorted in a descending order based on their correlation values), which can be discovered by using the method in [5] or [3].

3.2 Framework of Hoe-PGPL Algorithm

After discovering the potential k-correlated graphs in each batch, we use two levels of candidate lists to maintain these candidates, *i.e.*, a potential global list (PG) and a set of potential local lists (PLs). PG and PLs are also shown in Fig. 1. The detailed procedures of the proposed Hoe-PGPL algorithm is shown in Algorithm 1. In addition to the PG, PLs, and G_i, Hoe-PGPL takes several parameters as inputs. Specifically, k indicates the number of answers (graphs) to be returned; w represents the size of the sliding window (in terms of number of batches); and m is a parameter controlling the maximum number of candidates in PG. The output of Hoe-PGPL is the top-k correlated graphs A_g in the current sliding window.

Hoe-PGPL in Algorithm 1 mainly consists of four parts: (1) Hoeffding bound based candidate generation (steps 1-4); (2) Increasing and decreasing frequency information in PG (steps 5-8); When new batches arrive or become outdated, for each $g \in PG$, we update the frequency information in the window D. (3) Inserting new candidates into PG (steps 9), which is a key step for estimating the correlations of candidates (the details are reported in Algorithm 2); (4) Pruning PG (steps 11-12). If $|PG|$ exceeds $m * k$, we delete some unpromising candidates. Let $\epsilon_w = \sqrt{\frac{R^2 \ln(2/\delta)}{2w|G_i|}}$ and ϕ_D be the k-th correlated graph in PG, we delete candidates whose correlation is smaller than $\phi_D - 2\epsilon_w$.

Algorithm 2: $Merge(PL_i, PG, PLs)$

1 $\tau_{min} = \arg\min_{g_0 \in PG} \phi_D(g_0, q)$;
2 **foreach** $\tau \in PL_i$ *and* $\tau \notin PG$ **do**
3   **foreach** PL_j, $j \in [i-w+1, i-1]$ **do**
4   **if** $\tau \in PL_j$ **then**
5             $g' = \tau$;
6   **else**
7             $g' = \arg\min_{g \in PL_j} \phi_L(g, q)$;
8   Add the statistics of g' in batch G_j into N_τ, $N_{\tau q}$, and N_q for the window;
9   **if** $\phi_D(\tau_{min}, q) < \phi_D(\tau, q)$ **then**
10   Further revise the statistics of τ;
11   Insert τ to PG;

3.3 Global-local Check Scheme

In graph stream mining, because we cannot afford to record all information about the stream data, the complications normally arise from the situations a pattern τ, which is insignificant in previous observations, may emerge and become significant. Because τ is insignificant in previous observations, its information in the historical stream data is not recorded and therefore lost. We regard such patterns as **emerging candidate patterns**. Because we are not allowed to check the historical graph data to find τ's frequency in the previous batches of the window, the correlation value of $\phi_D(\tau, q)$ will have to be estimated, which may, in turn, introduce errors. We use two level of candidate list to handle this problem.

- **Potential Global List (PG):** A global candidate list PG is used to store the global frequency information of each candidate (N_g, N_{gq}, and N_q) in the sliding window D, from which candidates are sorted according to their correlation values.

- **Potential Local Lists (PLs):** We also use a set of PLs to record the frequency information of candidates (N_g, N_{gq}, and N_q) in a local batch. Specifically, we use a list PL_j, $j \in [i-w+1, i]$ to store the candidates retrieved from batch G_j.

Basically, there are two rules to estimate each emerging candidate pattern τ's frequency. (1) From a global perspective, its estimated value $\hat{\phi}_D(\tau, q)$ should be less than the minimum correlation value ($\phi_D(\tau_{min}, q)$) in PG if it is not stored in PG. (2) From a local perspective, its estimated value $\hat{\phi}_L(\tau, q)$ in each batch should be less than the minimum correlation value ($\phi_L(\tau_{min}, q)$) in a local batch if it is not stored in a local list PL_j.

Our scheme for maintaining PG and PLs is illustrated in Algorithm 2. Firstly we go through PL_{i-w+1} to PL_{i-1} to collect the estimated frequency information of each emerging candidate τ in a local perspective (steps 3-8 in Algorithm 2), which assures that $\phi_L(\tau, q)$ in $PL_j, j \in [i-w+1, i-1]$ is close to its true value in batch G_j. Step 8 uses the frequency information of g to estimate candidate τ in batch G_j. After collecting the frequency information of τ from PLs, we check it in PG from a global perspective (steps 9-10). Let

τ_{min} be the graph with minimum correlation value in PG, if $\phi_D(\tau_{min}, q)$ is smaller than the updated correlation value $\phi_D(\tau, q)$, we further revise the frequency information of g as we want $\phi_D(\tau, q) < \phi_D(\tau_{min}, q)$ in a global perspective. In our implementation, if $\tau \notin PL_j, j \in [i-w+1, i-1]$, we replace the frequency information of τ in batch G_j with the frequency information of τ_{min} in that batch. By doing so, the newly emerging candidates can be inserted into the PG with a precise estimation of its genuine correlation with query graph q. As a result, the ranking of PG will be more accurate and the query results can be more precise.

4. EXPERIMENTAL RESULTS

To validate the performance of our method, we collect graph data stream from a real-world NCI graph database [1], which contains compound structures of cancer and AIDS data. The original dataset contains about 249,000 graphs. After preprocessing and removing some disconnected graphs, we have a data stream with about 233,000 graphs.

The query graphs in our experiments consists of 50 graphs, each of which is randomly selected with a support in the whole data stream is less than 0.05. In addition to the system runtime and memory consumption, Precision and Recall are used to measure the results of our algorithm. Suppose A_g and T_g denotes the returned answers and true answers respectively, then $Precision = |T_g \bigcap A_g|/|A_g|$ and $Recall = |T_g \bigcap A_g|/|T_g|$. An algorithm is perfectly accurate if both precision and recall are equal to 1.

We compare our algorithms with an exhaustive search method (denoted by **Fixwin**) in terms of system runtime and memory consumption. When implementing Fixwin, *i.e.*, whenever a new batch of graphs arrives, Fixwin restarts to mine the top-k correlated graphs in the sliding window from scratch. To justify our estimated scheme, we compare Hoe-PGPL with a **Hoe-PG** scheme, which only uses PG to estimate the emerging candidate pattern τ (PLs are not employed). Unless specified otherwise, we set the default values δ=0.05, k=30, w=10, m=5, and $|G_i|$=5000.

In Fig. 2, we report the memory consumptions *w.r.t.* different top-k values. The result shows that the exhaustive search method (Fixwin) consumes several times more memory comparing to the Hoe-PGPL, mainly because Fixwin needs to store the graph data in the entire window while Hoe-PGPL only saves the most up-to-date batch of data and some candidates in PG and PLs. Meanwhile, the runtime performance in Fig. 2 demonstrates that Hoeffding bound based methods are about 5-10 times faster than the Fixwin method. On the other hand, when increasing the top-k values, the results in Fig. 3 show that the precision and the recall values of Hoe-PG and Hoe-PGPL both decline, with Hoe-PGPL showing much better performance than Hoe-PG. This demonstrates that by combining the two levels of candidate lists and the dynamic estimation of missing values of emerging candidate patterns (τ) in the historical batches, our method is able to maintain high effectiveness and high accuracy for stream-based top-k correlated graph query.

5. CONCLUSION

In this paper, we studied top-k correlated graph query from data stream, which discovers a query graph q's top-k correlated graphs in a sliding window covering a portion of stream data records. We argued that, for data streams with dynamic increasing volumes, simple exhaustive search for top-k correlated graphs needs to store the graph data within the sliding window and repeatedly carry out the query process, which is computationally expensive and memory intensive. By employing Hoeffding bound to discover a set of candidates in each batch and utilizing a global-local check scheme that integrates both potential global candidates (PG) and local candidates (PLs), we can accurately estimate the correlation of each candidates in PG. Experimental results demonstrate that Hoe-PGPL is several times more efficient than the exhaustive search method with respect to time and memory consumptions, and achieves high performance in terms of query precision and recall.

[1] http://cactus.nci.nih.gov/ncidb2/download.html

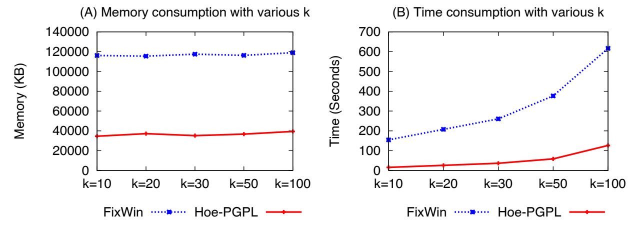

Figure 2: Memory and system runtime consumption VS. k values.

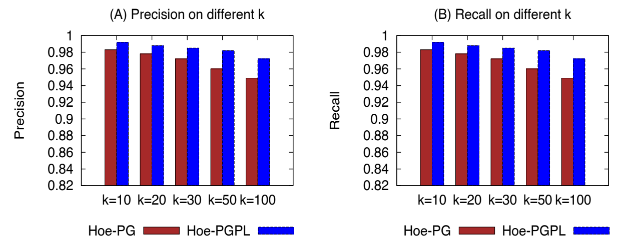

Figure 3: Precision and Recall VS. k values.

6. ACKNOWLEDGMENTS

This work was supported by the Australian Research Council (ARC) Future Fellowship under Grant No. FT100100971.

7. REFERENCES

[1] S. A. Cook. The complexity of theorem-proving procedures. In *ACM Symposium on Theory of Computing*, pages 151–158, 1971.

[2] W. Hoeffding. Probability Inequalities for Sums of Bounded Random Variables. *Journal of the American Statistical Association*, 58:13–30, 1963.

[3] Y. Ke, J. Cheng, and N. Wilfred. Correlation search in graph databases. In *KDD*, pages 390–399, 2007.

[4] Y. Ke, J. Cheng, and N. Wilfred. Correlated pattern mining in quantitative databases. *ACM Trans. on Database Systems*, 33:1–45, 2008.

[5] Y. Ke, J. Cheng, and J. X. Yu. Top-k Correlative Graph Mining. In *SIAM International Conf. on Data Mining*, pages 1038–1049, 2009.

[6] W. Zhou and H. Xiong. Volatile correlation computation: a checkpoint view. In *KDD*, pages 848–856, 2008.

CIKM12
The 21st ACM International Conference on Information and Knowledge Management
Oct. 29 - Nov. 2, 2012, Maui, Hawaii, USA

Reviews For Paper

Track	Posters
Paper ID	1780
Title	Continuous Top-k Correlated Graph Query for Data Streams

---

Masked Reviewer ID: Assigned_Reviewer_1
Review:

Question	
Overall Recommendation	Weak Reject: I don't really like the paper but won't object strongly against acceptance
Detailed Comments	In the paper, the authors propose Hoe-PGPL to identify top-k correlated graphs from data stream. Experiments have been conducted to verify the effectiveness and efficiency of the proposed approach. (1) Abstract. Our method represents the first research endeavor for data stream based top-k correlated graph query: downgrade the tune "the first research". (2) Section 3.2. Equation 2 is meaningless since the right hand size is always equal or larger 2. (3) I cannot see how equation 3 is derived from equation 2. It is like a magic - it is mostly a wrong claim. Equations 2 and 3 are quite important for the entire paper. I should be clarified.

Masked Reviewer ID: Assigned_Reviewer_2
Review:

Question	
Overall Recommendation	Weak Accept: I quite like the paper but won't object strongly against rejection
Detailed Comments	The paper studies an existing problem on a new setting; that is, correlation search over graph streams. Specifically, given a query graph and a continuous graph stream, it is to find the top-k correlated graphs between the query and the data stream over sliding windows. Strong points: 1) It investigates an existing problem in a streaming setting, which may have potential applications; 2) It proposes to use Hoeffding bound for candidate generation, and the potential global and local list data structure are introduced to support effective estimation. 3) Experimental result is good, compared with the baseline method. It is interesting to see the statistical bound has good performance on structure related problems. 4) With clear presentation, the paper is easily readable.

	Weak points: 1) Formatting problem: "stream-based top-k..." in Section 4. 2) Grammatical mistake: "Fixwin restart to mine..." in Section 4.

- Main Page
- CIKM'12
- BooksOnline'12
- CDMW'12
- CloudDB'12
- CrowdSens'12
- DOLAP'12
- DTMBIO'12
- DUBMMSM'12
- ESAIR'12
- IKM4DR'12
- MIXHS'12
- PIKM'12
- PLEAD'12
- SHB'12
- Web-KR'12
- WIDM'12
- Compilation Author Index
- CIKM 2013 Call for Papers

CIKM 2012 Proceedings

Xuewen Chen, Guy Lebanon, Haixun Wang, Mohammed J. Zaki, & Wolfgang Gatterbauer (editors), Proceedings of the 21st ACM International Conference on Information and Knowledge Management, CIKM'12, October 29–November 2, 2012, Maui, HI, USA. ACM 2012, ISBN 978-1-4503-1156-4.

Table of Contents (Complete Table of Contents)

CIKM 2012 Online Program at a Glance

Keynote Addresses

(Return to Top)

Session 1 - KM Track: Recommender Systems

(Return to Top)

Session 2 - KM Track: Pattern Mining

(Return to Top)

Session 3 - IR Track: Evaluation Methodologies

(Return to Top)

Session 4 - IR Track: Social Media Search

(Return to Top)

Session 5 - KM Track: Link and Graph Mining

(Return to Top)

Session 6 - IR Track: Language Technologies

(Return to Top)

Session 7 - DB Track: Graph and Knowledge Base

(Return to Top)

Session 8 - DB Track: Temporal, Spatial and Multimedia Databases

(Return to Top)

Session 9 - KM Track: Matrix Methods and Anomaly Detection

(Return to Top)

Session 10 - KM Track: Social Networks

(Return to Top)

Session 11 - IR Track: Advertising

(Return to Top)

Session 12 - IR Track: System Architecture, Distributed IR, Scalability

(Return to Top)

Session 13 - KM Track: Advertisement and Products

(Return to Top)

Session 14 - KM Track: Clustering

(Return to Top)

Session 15 - IR Track: Recommendation Systems

(Return to Top)

Session 16 - IR Track: Digital Libraries and Citation Analysis

Session 17 - KM Track: Text Mining

Session 18 - IR Track: Formal Retrieval Models and Learning to Rank

Session 19 - DB Track: Probabilistic and Uncertain Data

(Return to Top)

Session 20 - DB Track: Top-k and Nearest Neighbor Queries

(Return to Top)

Session 21 - KM Track: Spatial and Temporal Methods

(Return to Top)

Session 22 - IR Track: Web Search

Session 23 - DB Track: Web Data Management

Session 24 - KM Track: Information Extraction

Session 25 - IR Track: Topic Modeling and Content and Sentiment Analysis

Session 26 - DB Track: Query Processing, Optimization and Performance

(Return to Top)

Session 27 - KM Track: Classification and Semantic Methods

(Return to Top)

Session 28 - IR Track: Multimedia and User Feedback

(Return to Top)

Session 29 - DB Track: Emerging and Advanced Topics

(Return to Top)

Session 30 - KM Track: Novel Applications

(Return to Top)

Session 31 - IR Track: Social Networks

(Return to Top)

Knowledge Management Short Paper Session

(Return to Top)

(Return to Top)

(Return to Top)

(Return to Top)

(Return to Top)

Information Retrieval Short Paper Session

(Return to Top)

(Return to Top)

(Return to Top)

(Return to Top)

(Return to Top)

(Return to Top)

Guido Zuccon *(Australian e-Health Research Centre)*
Joemon M. Jose *(University of Glasgow)*

Entity Centric Query Expansion for Enterprise Search (Page 1955)
Xitong Liu *(University of Delaware)*
Hui Fang *(University of Delaware)*
Fei Chen *(HP Labs)*
Min Wang *(HP Labs)*

Location-sensitive Resources Recommendation in Social Tagging Systems (Page 1960)
Chang Wan *(The University of Hong Kong)*
Ben Kao *(The University of Hong Kong)*
David W. Cheung *(The University of Hong Kong)*

Differences in Effectiveness across Sub-Collections (Page 1965)
Mark Sanderson *(RMIT University)*
Andrew Turpin *(University of Melbourne)*
Ying Zhang *(RMIT University)*
Falk Scholer *(RMIT University)*

Map to Humans and Reduce Error - Crowdsourcing for Deduplication Applied to Digital Libraries (Page 1970)
Mihai Georgescu *(University of Hannover)*
Dang Duc Pham *(University of Hannover)*
Claudiu S. Firan *(University of Hannover)*
Wolfgang Nejdl *(University of Hannover)*
Julien Gaugaz *(University of Hannover)*

Full-Text Citation Analysis: Enhancing Bibliometric and Scientific Publication Ranking (Page 1975)
Xiaozhong Liu *(Indiana University, Bloomington)*
Jinsong Zhang *(Dalian Maritime University)*
Chun Guo *(Indiana University, Bloomington)*

Detecting Offensive Tweets via Topical Feature Discovery over a Large Scale Twitter Corpus (Page 1980)
(Return to Top)
Guang Xiang *(Carnegie Mellon University)*
Bin Fan *(Carnegie Mellon University)*
Ling Wang *(Carnegie Mellon University)*
Jason I. Hong *(Carnegie Mellon University)*
Carolyn P. Rose *(Carnegie Mellon University)*

Automatic Query Expansion Based on Tag Recommendation (Page 1985)
Vitor Oliveira *(Universidade Federal de Minas Gerais & Zunnit Technologies)*
Guilherme Gomes *(Universidade Federal de Minas Gerais)*
Fabiano Belém *(Universidade Federal de Minas Gerais)*
Wladmir Brandão *(Universidade Federal de Minas Gerais)*
Jussara Almeida *(Universidade Federal de Minas Gerais)*
Nivio Ziviani *(Universidade Federal de Minas Gerais & Zunnit Technologies)*
Marcos Gonçalves *(Universidade Federal de Minas Gerais)*

The Downside of Markup: Examining the Harmful Effects of CSS and Javascript on Indexing Today's Web (Page 1990)
Karl Gyllstrom *(Katholieke Universiteit Leuven)*
Carsten Eickhoff *(Delft University of Technology)*
Arjen P. de Vries *(Centrum Wiskunde & Informatica)*
Marie-Francine Moens *(Katholieke Universiteit Leuven)*

You Should Read This! Let Me Explain You Why: Explaining News Recommendations to Users (Page 1995)
Roi Blanco *(Yahoo! Research)*
Diego Ceccarelli *(ISTI-CNR & Università di Pisa)*
Claudio Lucchese *(ISTI-CNR & Università di Pisa)*
Raffaele Perego *(ISTI-CNR & Università di Pisa)*
Fabrizio Silvestri *(ISTI-CNR & Università di Pisa)*

Characterizing Web Search Queries That Match Very Few or No Results (Page 2000)
Ismail Sengor Altingovde *(University of Hannover)*
Roi Blanco *(Yahoo! Research)*
Berkant Barla Cambazoglu *(Yahoo! Research)*
Rifat Ozcan *(Bilkent University)*
Erdem Sarigil *(Bilkent University)*
Özgür Ulusoy *(Bilkent University)*

A Unified Optimization Framework for Auction and Guaranteed Delivery in Online Advertising (Page 2005)
Konstantin Salomatin *(Carnegie Mellon University)*
Tie-Yan Liu *(Microsoft Research Asia)*
Yiming Yang *(Carnegie Mellon University)*

Query Recommendation for Children (Page 2010)
(Return to Top)
Sergio Duarte Torres *(University of Twente)*
Djoerd Hiemstra *(University of Twente)*
Ingmar Weber *(Yahoo! Research)*
Pavel Serdyukov *(Yandex)*

Modeling Browsing Behavior for Click Analysis in Sponsored Search (Page 2015)

(Return to Top)

Databases Short Paper Session

(Return to Top)

(Return to Top)

(Return to Top)

(Return to Top)

Knowledge Management Poster Session

(Return to Top)

(Return to Top)

(Return to Top)

Information Retrieval Poster Session

(Return to Top)

(Return to Top)

(Return to Top)

(Return to Top)

(Return to Top)

(Return to Top)

Databases Poster Session

(Return to Top)

Knowledge Management Demonstration Session

Information Retrieval Demonstration Session

(Return to Top)

(Return to Top)

Databases Demonstration Session

(Return to Top)

(Return to Top)

Workshop Summaries

(Return to Top)

(Return to Top)

(Return to Top)