Finding Significant Web Pages with Lower Ranks by Pseudo-Clique Search

Yoshiaki Okubo, Makoto Haraguchi, and Bin Shi

Division of Computer Science,
Graduate School of Information Science and Technology,
Hokkaido University,
N-14 W-9, Sapporo 060-0814, Japan
{yoshiaki, mh}@ist.hokudai.ac.jp

Abstract. In this paper, we discuss a method of finding useful clusters of web pages which are significant in the sense that their contents are similar or closely related to ones of higher-ranked pages. Since we are usually careless of pages with lower ranks, they are unconditionally discarded even if their contents are similar to some pages with high ranks. We try to extract such hidden pages together with significant higher-ranked pages as a cluster.

In order to obtain such clusters, we first extract semantic correlations among terms by applying *Singular Value Decomposition*(SVD) to the term-document matrix generated from a corpus w.r.t. a specific topic. Based on the correlations, we can evaluate potential similarities among web pages from which we try to obtain clusters. The set of web pages is represented as a weighted graph G based on the similarities and their ranks. Our clusters can be found as *pseudo-cliques* in G. We present an algorithm for finding Top-N weighted pseudo-cliques. Our experimental result shows that quite valuable clusters can be actually extracted according to our method.

1 Introduction

We often try to obtain useful information or knowledge from web pages on the Internet. *Information retrieval* (IR) *systems* are quite powerful and helpful tools for this task. For instance, *Google* is well known as a popular IR system with a useful search engine. Given some keywords we are interested in, such a system shows a list of web pages that are related to the keywords. These pages are usually ordered by some ranking mechanism adopted in the system. For example, the method of *PageRank* [1] adopted in Google is widely known to provide a good ranking.

In general, only some of the higher-ranked pages are actually browsed and the others are discarded as less important ones, since the list given by the system contains a large number of pages. However, such a system presents just one candidate of ranking from some viewpoint. Therefore, there might exist many pages which are unfortunately lower-ranked but are significant for us. More concretely

A. Hoffmann, H. Motoda, and T. Scheffer (Eds.): DS 2005, LNAI 3735, pp. 346–353, 2005.

speaking, the ranking by PageRank is determined based on the link structure of each web page. For example, pages without enough links from others tend to be lower-ranked even if they have significant contents similar to higher ranked pages. From this point of view, it would be worth investigating a framework in which such implicitly significant pages are listed together with higher-ranked pages. We discuss in this paper a method for finding useful clusters of web pages which are significant in the sense that their contents are similar or closely related to ones of higher-ranked pages.

1.1 Similarities Among Web Pages

In order to realize it, we first extract semantic correlations among terms by applying *Singular Value Decomposition*(SVD) [3] to the term-document matrix generated from a corpus gathered with respect to a specific topic. Given a set of ranked web pages for which we try to extract clusters, we can evaluate potential similarities among them based on the semantic correlations of terms. In previous approaches, similarities among web pages are often determined based on the link structure of web pages [2]. More concretely speaking, it has been considered that web pages with similar topical contents have dense links among them. Such a link structure might roughly reflect similarities among relatively *mature* pages. However, many interesting pages are newly released day by day and it is often difficult to expect a dense link structure of fresh pages. As the result, based on the link-based approach, we will fail in finding similarities among such new pages even if they have similar contents. On the other hand, we try to capture similarities among web pages independently of their link structure.

1.2 Extracting Clusters by Clique Search

The set of web pages is then represented as a weighted undirected graph based on the similarities and their ranks. If a pair of web pages has a similarity higher than a given threshold, they are connected by an edge. Moreover, each vertex (i.e. a web page) is assigned a weight so that higher-ranked pages have higher weights. Our clusters can be extracted by finding *pseudo-cliques* in the graph G. A pseudo-clique is defined as the union of several maximal cliques in G with a required degree of overlap. Simple theoretical properties of pseudo-cliques are presented. Based on the properties, we can obtain some pruning rules for pseudo-clique search. We design a depth-first algorithm for finding pseudo-cliques whose weights (evaluation values) are in the top N. Our preliminary experimental result shows that a quite valuable cluster can be actually extracted as a pseudo-clique in G.

One might claim that a naive method would be sufficient for extracting clusters consisting of similar higher-ranked and lower-ranked pages. That is, for each web page with a higher rank, we can gather lower-ranked pages similar to the higher-ranked one. As well as this kind of clusters, our method can extract other various kinds of clusters simultaneously by changing the weighting of web pages in our graph construction process. Under some weighting, for example, a cluster

consisting of several pages which are moderately ranked might be obtained as in the top N. In this sense, our method includes such a naive method.

Our method for extracting clusters by clique search is a general framework. The literature [6,9] has investigated methods for finding appropriate *data abstractions* (groupings) of attribute values for classification problems, where each abstraction is extracted as a weighted exact clique. A gene expression data has been also processed in [7]. A cluster consisting of genes which behave similarly is extracted as an exact clique. The current pseudo-clique search can be viewed as an extension of these previous search methods for exact cliques [6,7,8,9].

Our clique search-based method has advantage over previous clustering methods in the following points. In the traditional *hierarchical* or *partitional* clustering, the whole set of data is divided into some clusters. Although the number of clusters is usually controlled by a user-defined parameter, it is well known that providing an adequate value for the parameter is not so easy. Under an inadequate parameter setting, we often obtain many useless clusters. From the computational point of view, the cost for producing such useless clusters will be quite waste. On the other hand, in our method, we can extract *only* nice clusters whose evaluation values are in the top-N, where N can be given arbitrarily. In this sense, we will never suffer from quite useless clusters. Furthermore, extracting only nice clusters has an advantage in the computation. We can enjoy a branch-and-bound search in order to extract them. In our search, we do not have to examine many branches concerning clusters not in the top N.

2 Semantic Similarity Among Web Pages

In order to find clusters of web pages, we have to measure similarities among web pages. For the task, we follow a technique in *Information Retrieval*(IR) [3].

2.1 Term-Document Matrix

Let $\mathcal{D}$ be a set of documents and $\mathcal{T}$ the set of terms appeared in $\mathcal{D}$ [1]. We first remove too frequent and too infrequent terms from $\mathcal{T}$. The set of remaining terms, called *feature terms*, is denoted by $\mathcal{T}^*$. Supposing $|\mathcal{T}^*| = n$, each document $d_i \in \mathcal{D}$ can be represented as an n-dimensional document vector $\boldsymbol{d_i} = (tf_{i1}, \ldots, tf_{in})^T$, where tf_{ij} is the frequency of the term $t_j \in \mathcal{T}^*$ in the document d_i. Thus, $\mathcal{D}$ can be translated into a *term-document matrix* $(\boldsymbol{d_1}, \ldots, \boldsymbol{d_{|\mathcal{D}|}})$.

2.2 Extracting Semantic Similarity with SVD

For the term-document matrix, we apply *Singular Value Decomposition*(SVD) in order to extract correlations among feature terms [3].

An $m \times n$ matrix A can be decomposed by applying SVD as $A = U\Sigma V^T$, where U and V are $m \times m$ and $n \times n$ orthogonal matrices, respectively. Each

[1] In order to obtain such terms from documents without spaces among words (like Japanese documents), we need to apply *Morphological Analysis* to $\mathcal{D}$.

column vector in U (V) is called a left (right) singular vector. Σ is an $m \times n$ matrix of the form

$$\Sigma = \left[\begin{array}{ccc|c} \sigma_1 & & O & \\ & \ddots & & O_{r\times(n-r)} \\ O & & \sigma_r & \\ \hline & O_{(m-r)\times r} & & O_{(m-r)\times(n-r)} \end{array}\right],$$

where $rank(A) = r$ ($r \leq \min\{m, n\}$) and σ_i is called a *singular value*. First r left singular vectors $\boldsymbol{u_1}, \ldots, \boldsymbol{u_r}$ correspond to a orthonormal basis and define a new subspace of the original one in which column vectors of A exist, where the $m \times r$ matrix $(\boldsymbol{u_1}, \ldots, \boldsymbol{u_r})$ is denoted by U_r.

Let us assume the matrix A is a term-document matrix generated from a set of documents. Intuitively speaking, by applying SVD to A, we can capture *potential but not presently evident* correlations among the terms. Highly semantically correlated terms give a base vector $\boldsymbol{u_i}$ and define a dimension corresponding to a compound term. Such new base vectors define a new subspace based on compound terms. For documents d_1 and d_2 not in A, therefore, if they are projected on the subspace, we can find similarity between them based on the semantic correlations among terms captured from the original documents in A.

In order to take such semantic similarities of web pages into account, we prepare a *corpus* of documents written about some specific topic. Then by applying SVD to the term-document matrix generated from the corpus, we obtain a subspace reflecting semantic correlations among terms in the corpus. Let U_r be the orthonormal basis defining the subspace [2].

Besides the corpus, with some keywords related to the corpus topic, we retrieve a set of web pages $\mathcal{P}$ from which we try to obtain clusters. Using the same feature terms for the corpus, each document $p_i \in \mathcal{P}$ is represented as a vector $\boldsymbol{p_i} = (tf_{i1}, \ldots, tf_{in})^T$, where tf_{ij} is the frequency of the feature term t_j in p_i. Then each web page $\boldsymbol{p_i}$ is projected on the subspace as $\boldsymbol{p_i^r} = U_r^T \boldsymbol{p_i}$.

A similarity between web pages p_i and p_j, denoted by $sim(p_i, p_j)$, is defined based on the standard *cosine measure*, that is, $sim(p_i, p_j) = \dfrac{\boldsymbol{p_i^r} \cdot \boldsymbol{p_j^r}}{\|\boldsymbol{p_i^r}\| \times \|\boldsymbol{p_j^r}\|}$.

3 Finding Clusters by Top-N Pseudo-Clique Search

3.1 Graph Representation of Web Pages

Let $\mathcal{P}$ be a set of web pages from which we try to extract clusters. In order to find our clusters, $\mathcal{P}$ is represented as an undirected weighted graph G.

Assume we computed the semantic similarities among pages in $\mathcal{P}$ according to the procedure just discussed above. Let δ be a similarity threshold. Each

[2] In IR, we do not always use r left singular vectors. A part of them, that is, $U_k = (\boldsymbol{u_1}, \ldots, \boldsymbol{u_k})$ ($k < r$) is usually used for *approximation*. Such an approximation with U_k is called *Latent Semantic Indexing* (LSI) [3].

page $p_i \in \mathcal{P}$ corresponds to a vertex in G. For any web pages $p_i, p_j \in \mathcal{P}$, if $sim(p_i, p_j) \geq \delta$, then they are connected by an edge. Furthermore, we assign a weight to each vertex (page) based on its rank, where a higher-ranked page is assigned a larger weight. The weight of a page p is referred to as $w(p)$.

3.2 Top-N Weighted Pseudo-Clique Problem

Our cluster of similar pages can be obtained as a weighted *pseudo-clique* in the graph G. In fact, we obtain only nice clusters by extracting maximal weighted pseudo-cliques whose evaluation values are in the top-N. Before giving the problem description, we first define *degree of overlap* for a class of maximal cliques.

Definition 1 (Degree of Overlap for Maximal Clique Class). Let $\mathcal{C} = \{C_1, \ldots, C_m\}$ be a class of maximal cliques. The *degree of overlap* for $\mathcal{C}$, denoted by $overlap(\mathcal{C})$, is defined as $overlap(\mathcal{C}) = \min_{C_i \in \mathcal{C}} \left\{ \left| \cap_{C_j \in \mathcal{C}} C_j \right| / |C_i| \right\}$. ■

Using the notion of overlap degree, our pseudo-cliques is defined as follows.

Definition 2 (Pseudo-Clique). Let $\mathcal{C} = \{C_1, \ldots, C_m\}$ be a class of maximal cliques in a graph. $pseudo(\mathcal{C}) = \cup_{C_i \in \mathcal{C}} C_i$ is called a *pseudo-cliques* with the overlap degree $overlap(\mathcal{C})$. Its *size* and *weight* (evaluation value) are given by $|pseudo(\mathcal{C})|$ and $w(pseudo(\mathcal{C})) = \Sigma_{v \in pseudo(\mathcal{C})} w(v)$, respectively [3]. Moreover, the shared vertices, $\bigcap_{C_i \in \mathcal{C}} C_i$, is called the *core*. ■

We can now define the problem of finding Top-N weighted pseudo-cliques.

Definition 3 (Top-N Weighted Maximal τ Pseudo-Clique Problem). Let G be a graph and τ a threshold for overlap degree. The *Top-N Weighted Maximal τ Pseudo-Clique Problem* is to find any maximal pseudo-clique in G such that its overlap degree is greater than or equal to τ [4] and its weight is in the top N. ■

3.3 Algorithm for Finding Top-N Weighted Pseudo-Cliques

We present here an algorithm for finding Top-N weighted pseudo-cliques.

In our search, for a clique Q in G, we try to find a τ-valid pseudo-clique $\tilde{C}$ whose core is Q. In order to precisely discuss how it can be found, we introduce a notion of *extensible candidates* for a given clique.

Definition 4 (Extensible Candidates). Let G be a graph and Q a clique in G. A vertex $v \in V$ adjacent to any vertex in Q is called an *extensible candidate* for Q. The set of extensible candidates is denoted by $cand(Q)$. ■

From the definition, we can easily observe the followings.

[3] The weight of pseudo-clique is not restricted to the sum of vertex weights. Any monotone weight under the set inclusion can be accepted in the following discussion.

[4] Such a pseudo-clique is said to be *τ-valid*.

Observation 1. Let Q and Q' be cliques in G such that $Q \subseteq Q'$. Then, $cand(Q) \supseteq cand(Q')$ and $w(Q) + w(cand(Q)) \geq w(Q') + w(cand(Q'))$ hold, where $w(Q)$ is the weight of the clique Q. ■

Note here that the weight of a pseudo-clique with the core Q is *at most* $w(Q) + w(cand(Q))$. Therefore, a simple theoretical property can be derived.

Observation 2. Let Q be a clique. Assume we already have *tentative* Top-N weighted maximal pseudo-cliques and the minimum weight of them is w_{min}. If $w(Q) + w(cand(Q)) < w_{min}$ holds, then for any extension Q' of Q[5], there exists no pseudo-clique with the core Q' whose weight is in the top N. ■

Assume that a τ-valid pseudo-clique $\tilde{C}$ contains a clique Q as its core. $\tilde{C}$ can be obtained as the union of any maximal clique C such that $Q \subset C$ and $|Q|/|C| \geq \tau$. It should be noted here that for such a clique C, there exists a maximal clique D in $G(cand(Q))$ such that $Q \cup D = C$, where $G(cand(Q))$ is the subgraph induced by $cand(G)$. That is, finding any maximal clique D in $G(cand(Q))$ such that $|Q|/(|Q| + |D|) \geq \tau$ is sufficient to obtain the pseudo-clique $\tilde{C}$. Although one might claim that such a task is quite expensive from the computational point of view, we can enjoy a pruning in the maximal clique search based on the following observation.

Observation 3. For a clique Q in G, let us assume that we try to find a τ-valid pseudo-clique $\tilde{C}$ whose core is Q. For a clique D in $G(cand(Q))$, if $|D| > (\frac{1}{\tau} - 1) \cdot |Q|$, then any extension of D is useless for obtaining $\tilde{C}$. ■

Furthermore, in a certain case, we can immediately obtain a pseudo-clique without finding maximal cliques in $G(cand(Q))$.

Observation 4. Let Q be a clique in G and τ a threshold for overlap degree. If the followings hold, then $Q \cup cand(Q)$ is a τ-valid maximal pseudo-clique with the core Q.

- $(\frac{1}{\tau} - 1) \cdot |Q| \geq k$ holds, where k is an upper bound of the maximum clique size in $G(cand(Q))$.
- For any $v \in cand(Q)$, its degree in $G(cand(Q))$ is less than $|cand(Q)| - 1$. ■

Upper bounds for the maximum clique size have been widely utilized in efficient depth-first branch-and-bound algorithms for finding maximum cliques [4,5,9]. The literature [5] has argued that the (*vertex*) *chromatic number* χ can provide the tightest upper bound. However, identifying χ is an NP-complete problem. Therefore, approximations of χ are usually computed [4,5,9].

Based on the above properties, Top-N τ-valid weighted pseudo-cliques can be extracted with a *depth-first hybrid search*. For each core candidate Q, its surroundings are explored by finding maximal cliques in $G(cand(Q))$. In the search for core candidates, we can enjoy a pruning based on Observation 2. In the surroundings search, a pruning based on Observation 3 can be applied. Furthermore, for some core candidates, our surroundings search can be skipped based on Observation 4. More precise description of our algorithm is found in [10].

[5] For a pair of cliques Q and Q', if $Q \subset Q'$, then Q' is called an *extension* of Q.

4 Experimental Result

In this section, we present a result of our experimentation conducted on a PC with Xeon-2.4 GHz and 512MB RAM.

We have manually prepared a (Japanese) corpus with 100 documents written about "Hokkaido" and have selected 211 feature terms from the corpus. Applying SVD to the 211×100 matrix, a 98-dimensional subspace has been obtained.

Besides the corpus, we have retrieved 829 web pages by Google with the keywords "Hokkaido" and "Sightseeing". The 211×829 term-document matrix for the pages has been projected on the subspace in order to capture semantic similarities among pages. Under the setting of $\delta = 0.95$, we have constructed a weighted graph G from the projected pages. The numbers of vertices and edges are 829 and 798, respectively. Each page d has been given a weight defined as $w(d) = 1/rank(d)^2$, where $rank(d)$ is the rank of d assigned by Google (PageRank). We have tried to extract Top-15 weighted 0.8-pseudo cliques in the graph.

Among the extracted 15 clusters (pseudo-cliques), the authors especially consider that the 11^{th} one is quite interesting. It consists of 6 pages with the ranks, 11^{th}, 381^{th}, 416^{th}, 797^{th}, 798^{th} and 826^{th}. The 11^{th} and 328^{th} pages are index pages for travel information and we can make reservations for many hotels via the pages. The 416^{th} page is an article in a private BBS site for travels. It reports on a private travel in Hokkaido and provides an actual information about hotels and enjoyable foods. The 797^{th} and 798^{th} personal pages give the names of two hotels serving smorgasbords in Hokkaido. The 826^{th} page lists hotels most frequently reserved in a famous travel site in 2004. Thus, their contents are very similar in the sense that all of them give some information about accommodations in Hokkaido, especially about hotels and foods. When we try to make travel plans for sightseeing in Hokkaido, we would often care about hotels and foods as important factors. In such a case, the cluster will be surely helpful for us.

Similar to the literature [8], we can find Top-N clusters of web pages by an *exact* clique search. In that case, however, our 11^{th} cluster can never be obtained. The cluster (that is, a pseudo-clique) consists of two exact maximal cliques: $\{11^{th}, 382^{nd}, 797^{th}, 798^{th}, 826^{th}\}$ and $\{382^{nd}, 416^{th}, 797^{th}, 798^{th}, 826^{th}\}$. In the exact case, the former is 11^{th} cluster, whereas the latter 343^{rd} one. It should be noted that the 416^{th} page will be invisible unless we specify a large N for Top-N (about 350). However, it would be impractical to specify such a large N because many clusters are undesirably extracted. Although 416^{th} page has valuable contents as mentioned above, we will lose a chance to browse it.

In case of pseudo-clique search, the 343^{rd} exact clique can be absorbed into the 11^{th} clique to form a pseudo-clique. That is, the 343^{rd} cluster can be drastically raised its rank. As the result, 416^{th} page can become visible by just specifying a reasonable N. Thus, our chance to get significant lower-ranked pages can be enhanced with the help of pseudo-cliques.

Our experimental result also shows that our pruning rules can be applied very frequently in our search. The number of cores actually examined was 69981 and our pruning were invoked at 40832 nodes of them. As the result, the total computation time was just 0.847 second.

5 Concluding Remarks

In this paper, we discussed a method of finding clusters of web pages which are significant in the sense that their contents are similar or closely related to ones of higher-ranked pages. Although we are usually careless of pages with lower ranks, they can be *explicitly* extracted together with significant higher-ranked pages. As the result, our clusters can provide new valuable information for users.

Obtained clusters are very sensitive to the assignment of vertex weights in our graph construction process. Although the reciprocal of the page rank squared currently adopted seems to be promising, we have to examine any other candidates. Furthermore, the required degree of overlap for pseudo-cliques also affects which clusters can be found. In order to obtain good heuristics for these settings, further experimentations should be conducted.

A meaningful cluster should have a clear explanation why the pages in the cluster are grouped together or what the common features in the cluster are. Our current method, unfortunately, does not have any mechanism to provide it clearly. If such a explanation mechanism is integrated, our clusters would be more convincing. An improvement on this point is currently ongoing.

References

1. L. Page, S. Brin, R. Motwani and T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web", http://dbpubs.stanford.edu/pub/1999-66, 1999.
2. A. Vakali, J. Pokorný and T. Dalamagas, "An Overview of Web Data Clustering Practices", Proceedings of the 9th International Conference on Extending Database Technology - EDBT'04, Springer-LNCS 3268, pp. 597 - 606, 2004.
3. K. Kita, K. Tsuda and M. Shishibori, "Information Retrieval Algorithms", Kyoritsu Shuppan, 2002 (in Japanese).
4. E. Tomita and T. Seki, "An Efficient Branch-and-Bound Algorithm for Finding a Maximum Clique", Proceedings of the 4th International Conference on Discrete Mathematics and Theoretical Computer Science - DMTCS'03, Springer-LNCS 2731, pp. 278 - 289, 2003.
5. T. Fahle, "Simple and Fast: Improving a Branch-and-Bound Algorithm for Maximum Clique", Proceedings of the 10th European Symposium on Algorithms - ESA'02, Springer-LNCS 2461, pp. 485 - 498, 2002.
6. K. Satoh, "A Method for Generating Data Abstraction Based on Optimal Clique Search", Master's Thesis, Graduate School of Eng., Hokkaido Univ., March, 2003. (in Japanese)
7. S. Masuda, "Analysis of Ascidian Gene Expression Data by Clique Search", Master's Thesis, Graduate School of Eng., Hokkaido Univ., March, 2005. (in Japanese)
8. B. Shi, "Top-N Clique Search of Web Pages", Master's Thesis, Graduate School of Eng., Hokkaido Univ., March, 2005. (in Japanese)
9. Y. Okubo and M. Haraguchi, "Creating Abstract Concepts for Classification by Finding Top-N Maximal Weighted Cliques", Proceedings of the 6th International Conference on Discovery Science - DS'03, Springer-LNAI 2843, pp. 418 - 425, 2003.
10. Y. Okubo and M. Haraguchi, "Finding Top-N Pseudo-Cliques in Simple Graph", Proceedings of the 9th World Multiconference on Systemics, Cybernetics and Informatics - WMSCI'05, Vol. III, pp. 215 - 220, 2005.