

Hierarchical Web-page Clustering via In-page and Cross-page Link Structures

[†]Cindy Xide Lin, [†]Yintao Yu, [†]Jiawei Han, and [‡]Bing Liu

[†]University of Illinois at Urbana-Champaign, [‡]University of Illinois at Chicago
[†]{xidelin2, yintao}@uiuc.edu, [†]hanj@cs.uiuc.edu, [‡]liub@cs.uic.edu

Abstract. Despite of the wide diversity of web-pages, web-pages residing in a particular organization, in most cases, are organized with semantically hierarchic structures. For example, the website of a computer science department contains pages about its people, courses and research, among which pages of people are categorized into faculty, staff and students, and pages of research diversify into different areas. Uncovering such hierarchic structures could supply users a convenient way of comprehensive navigation and accelerate other web mining tasks. In this study, we extract a similarity matrix among pages via in-page and cross-page link structures, based on which a density-based clustering algorithm is developed, which hierarchically groups densely linked webpages into semantic clusters. Our experiments show that this method is efficient and effective, and sheds light on mining and exploring web structures.

1 Introduction

Web page clustering has been studied extensively in the literature as a means to group pages into homogeneous topic clusters. However, much of the existing study [1] [7] [18] [9] is based on any arbitrary set of pages, *e.g.*, pages from multiple websites. Limited work has been done on clustering pages from a specific website of an organization. Despite of the wide diversity of webpages, webpages residing in a particular organization, in most cases, have some semantically hierarchic structures. For example, the website of a computer science department may contain a large set of pages about its people, courses, news and research, among which pages of people can be categorized into the ones of faculty, staff and students, and pages of research may diversify into different areas. Uncovering such hierarchic structures could supply users a convenient way of comprehensive navigation, accelerate other searching and mining tasks, and enables us to provide value-added services.

This is, however, a challenging task due to the semantic and structural heterogeneity of the webpages. Nevertheless, one can observe that the information in a site is usually organized according to certain logical relationships, *e.g.*, related items are often organized *together* in a way that is easier for users to find relevant information items. For example, in a university department, there is usually a page about its faculty, a page about its courses, *etc.* Exploring such

site organizational information, *i.e.*, *information item togetherness*, will help us cluster the items of the same type.

From an implementation point of view, such togetherness is typically manifested in HTML code through two means: *in-page structures* and *cross-page hyper-links*. Information items of the same type are usually coded as sibling nodes of the same parent in the HTML tag tree (*i.e.*, *DOM tree*), and links that represent similar items often reside together in a page as siblings, forming *parallel links* of a page. Such page structure and parallel links provide an important clue in the design of similarity functions for meaningful clustering.

Based on this idea, we develop a novel method, HSClus, for hierarchical site clustering of webpages in order to discover the inherent semantic structure of an organization’s website. Our major contributions include:

1. Deriving from *DOM* trees, a novel concept called *parallel links* is proposed, based on which a new similarity function between pages is developed.
2. A new clustering algorithm called HSClus is designed to group densely linked webpages into semantic clusters and identify their hierarchical relationships.

Our experiments show that HSClus is efficient and effective at uncovering webpage structures at some organization’s website, which sets a foundation for further mining and exploring web semantic structures.

2 Related Work

Spectral partitioning [5] is a group of one-level network clustering algorithms, which targets to cutting a graph into a set of sub-graphs, *i.e.*, clusters, with an object function that minimizes the number of cross-cluster edges and maximizes the number of in-cluster edges. Because its solution relies on the calculation of eigen values, the time complexity is square to the number of edges. Agglomerative hierarchical clustering [11] [3] treats each data point as a singleton cluster, and then successively merges clusters until all points have been merged into a single remaining cluster. However, these methods are sensitive to outliers. *DOM tree structures* have been widely used for webpage segmentation and partitioning. As the correspondence of in-page parallel links in this paper, [14] enhances web page classification by utilizing labels and contents information from sibling pages. *Web patterns* [10] are formalized descriptions of common features of objects on web pages. Each page is presented by a vector of pattern weights, which record the extent of importance of the pattern for the web page. Based on pattern vectors, similarity between pages is defined. To automatically extract main classes of pages offered by a website, [4] compares structures of *DOM* trees. In order to improve search results via text contents [1] uses the path length and [7] uses weighted path between two pages to adjust clusters. [19] combines out-links, in-links and terms in page contents to improve the clustering quality on web search results. [2] finds dense units by density-based algorithms, and then merges units by agglomerative hierarchical clustering.

3 Similarity Matrix

This section takes a set of pages $P = \{p_0, \dots, p_{n-1}\}$ at an organization’s website as *input objects*, and outputs a *similarity matrix* among pages in P for the clustering algorithm introduced in latter sections.

Web pages contain abundant information about link structures that can help discovering web clusters, which is mainly in two categories: *cross-page link-structures* and *in-page link-structures*. The former one refers to the link graph among webpages, while the latter one refers to the organization of links inside an individual page. If we regard cross-page link-structures as web structures at the macro-level, then in-page link-structures are the one at the micro-level. Combining macro- and micro-levels of web structures will gain great power for link-based web page clustering.

3.1 Cross-Page Link-Structures

Co-citation [15] and bibliography-coupling [8] are two popular measures in the analysis of link graph. Concretely, for pages p_i and p_j , their co-citation $C(i, j)$ and bibliography-coupling $B(i, j)$ are defined as the frequencies of common in-links and out-links, respectively, saying $C(i, j) = \sum_k E(i, k)E(j, k)$ and $B(i, j) = \sum_k E(k, i)E(k, j)$, where $E(i, j) = 1$ if there is a hyper-link in p_i pointing to p_j and otherwise $E(i, j) = 0$. We use *Cosine* function to calculate the similarity $Sim_{CB}(i, j)$ gained from $C(i, j)$ and $B(i, j)$ for pages p_i and p_j ¹:

$$Sim_{CB}(i, j) = \frac{C(i, j)}{\sqrt{C(i, i) \cdot C(j, j)}} + \frac{B(i, j)}{\sqrt{B(i, i) \cdot B(j, j)}} \quad (1)$$

3.2 In-Page Link-Structures

The DOM (Document Object Model) is a platform- and language-independent standard object model for representing HTML or XML documents. Building DOM trees from input web pages is a necessary step for many data extraction algorithms. Furthermore, nodes in DOM trees are written in the form of tags, indicating the structure in a web page and a way of hierarchically arranging text-based contents. Formally, we use $DOM(i)$ to denote the DOM tree extracted from the source code of a particular web page p_i with trivial HTML tags removed. For a tree node μ in $DOM(i)$, the sub-tree rooted at μ is denoted by $DOM_\mu(i)$.

In this sub-section, we will introduce *Parallel Link* as a novel concept derived from DOM trees. Note parallel links are independently extracted from each page of the targeting website, which reasonably assumes the homogeneity of the layout and the contents inside one particular page, e.g., the homepage of a laboratory may list hyper-links to its professors together and the link of each professor is followed by the professor’s name and then by the email. Here the consecutive

¹ Many other functions analyzed in [16] may also be good choices.

positions of these hyper-links and the homogeneous organization of each professor’s information are good examples of in-page link-structures, which give strong hints of the semantic meaning of these professors’ pages. It is necessary to understand that it does not make any assumptions about the homogeneity of pages among the whole website.

Concretely, for sibling sub-trees ² $DOM_{\mu_1}(i)$, $DOM_{\mu_2}(i)$, \dots , $DOM_{\mu_k}(i)$, they become a group of *Parallel Sub-Trees* if $DOM_{\mu_s}(i)$ and $DOM_{\mu_t}(i)$ for any $s, t \in 1..k$ are exactly the same (including the tree structures and the HTML tags). Tree nodes $\nu_1, \nu_2, \dots, \nu_k$ form a group of *Parallel Nodes* if they locate in the same position of $DOM_{\mu_1}(i)$, $DOM_{\mu_2}(i)$, \dots , $DOM_{\mu_k}(i)$, respectively, and furthermore become a group of *Parallel Links* if their HTML tags are ‘hyper-links’ (i.e., $\langle a \rangle$). Finally, we scan pages in P one by one, and extract all groups of parallel links with the size no less than 4. The similarity $Sim_P(i, j)$ of pages p_i and p_j gained from in-page link-structures equals to how many times p_i and p_j appear in a group of parallel links.

3.3 Consolidating with Content-based Similarities

The final similarity $Sim(i, j)$ for pages p_i and p_j is:

$$SIM(i, j) = SIM_{CB}(i, j) + w_2 \cdot SIM_P(i, j) + w_1 \cdot SIM_{content}(i, j) \quad (2)$$

Here $SIM_{content}(i, j)$ can be obtained by any kind of content-based similarity functions [6] [20]. w_1 and w_2 are parameters that tunes linear weights among the three parts. Different values of w_1 and w_2 express different emphasis to structure-based and content-based similarities. There could be more than one good answers for a page clustering task. It is not a competition between two runners (i.e., structure-based and content-based similarities) to see which one has the better performance, instead we are installing two engines for more effective similarity functions as well as clustering results.

4 Hierarchical Clustering

Although there have been many clustering algorithms developed for web applications, we choose to further develop the density-based approach with the following reasoning.

1. Web clusters may have arbitrary shapes and the data points inside a cluster may be arbitrarily distributed. Density-based clustering is good at generating clusters of arbitrary shapes.
2. Web datasets are usually huge. Density-based clustering can be linear to the number of edges. Moreover, since the average number of hyper-links inside one page is regarded as a constant, the number of edges is approximately linear to the number of vertices.

² Two sub-trees are siblings if they have the same parent.

3. Web clusters may vary a lot in size, and web datasets contain noises. Spectral partitioning algorithms have constraints on cluster sizes, and agglomerative hierarchical clustering methods are sensitive to outliers.

SCAN [17] is a one-level density-based network clustering algorithm, of which one clear advantage is its linear time complexity that out-performs other methods. However, SCAN requires two parameters, and the optimal parameters that lead to the best clustering performance are given by human via visualization. In this section, we extend SCAN to a hierarchical clustering algorithm, called HSCLUS.

Algorithm Framework. It is natural to derive HSCLUS from SCAN by iteratively applying SCAN to each cluster obtained by SCAN in the previous step. However, since different sets of pages may have different optimal parameters, it is infeasible to select two fixed parameters as the input for each call of SCAN. To solve this problem, HSCLUS (i) tests SCAN with different pairs of parameters, (ii) uses a scoring function to evaluate the clustering results under different parameters, and (iii) finally clusters pages by the optimal parameters. For each resulting cluster, HSCLUS repeats the same procedure until termination conditions are met. Because of the space limitation, details are omitted.

Complexity Analysis. Usually, the number of levels L in a clustering hierarchy is small (e.g., no more than 10), and we select a constant number (say K) of parameters to test. Since SCAN is linear to the number of edges m , the time complexity of HSCLUS is also linear, which is $O(LKm)$.

5 Experiments

In this section, we evaluate the efficiency and the effectiveness of HSCLUS on both synthetic and real datasets. All algorithms are implemented in Java Eclipse and Microsoft Visual Studio 2008, conducted in a PC with 1.5GHz CPU and 3GB main memory. We compare HSCLUS with two algorithms: (i) *k-medoids* [13] and (ii) *FastModularity* [3].

5.1 Effectiveness

A real dataset *UIUC CS* is the complete set of pages in the domain of *cs.uiuc.edu* crawled down by Oct. 3, 2008. It has 12,452 web-pages and 122,866 hyper-links. The average degree of each page is 19.7, and 33,845 groups of parallel links are discovered.

To evaluate the usefulness of parallel links, *Figure. 1* and *2* show the clustering results with and without similarities gained from in-page link structure. As observed, the two figures are generally the same at high levels; in low levels, *Figure. 2* may mix pages that are in different kinds but have close semantic meanings, e.g., *Research/Faculty* and *Research/Area* alternate, and *Undergraduate/Transfer* is in the middle of *Undergraduate/Course*.

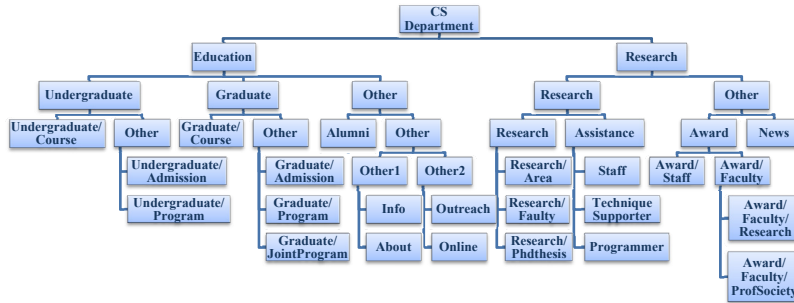


Fig. 1. Result of HSClus with the similarities gained from parallel links

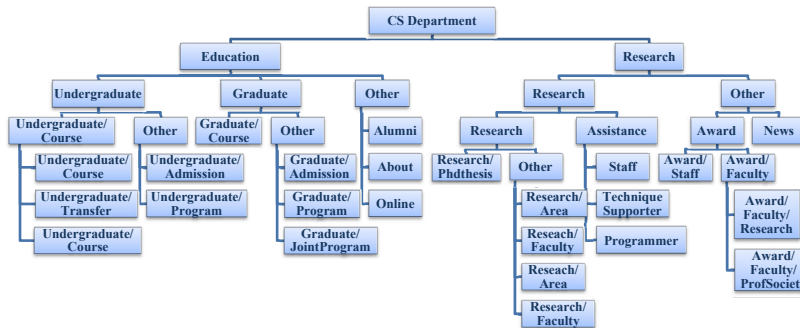


Fig. 2. Result of HSClus without the similarities gained from parallel links

Figure 3 and 4 are the results generated by FastModularity and k -medoids (some parts are omitted), respectively. We can observe that, the clustering quality is much lower than HSClus.

5.2 Efficiency

To verify that HSClus is as fast as linear against networks of different sizes, we generate 8 synthetic graphs, whose numbers of edges range from 2,414 to 61,713,102, to test corresponding running times. We can see in Figure 5 that the running time (in second) of HSClus is linear against the input size (number of edges); FastModularity increases more quickly than linear; and k -medoids rises dramatically.

6 Conclusion

This paper develops a novel method for hierarchical clustering of webpages in an organization in order to discover the inherent semantic structure of the website. Both cross-page link structure and in-page link organizations are explored to

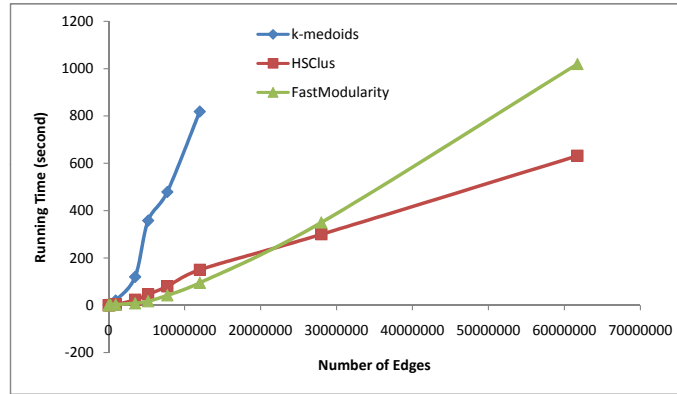


Fig. 5. Efficiency comparison

4. V. Crescenzi, P. Merialdo, and P. Missier. Clustering web pages based on their structure. *Data Knowledge Engineer*, 2005.
5. C. Ding. Spectral clustering tutorial. *ICML*, 2004.
6. C. Glassman, M. Manasse, and G. Zweig. Syntactic clustering of the web. *WWW*, 2005.
7. J. Hou and Y. Zhang. Utilizing hyperlink transitivity to improve web page clusterings. *ADC*, 2003.
8. M. Kessler. Bibliographic coupling between scientific papers. *American Documentation*, 1963.
9. J. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of ACM-SIAM*, 1998.
10. M. Kudelka, V. Snasel, O. Lehecka, E. El-Qawasmeh, and J. Pokorny. Web pages reordering and clustering based on web patterns. *SOFSEM*, 2008.
11. C. D. Manning and H. Schtzen. Foundations of statistical natural language processing. *MIT Press*, 1999.
12. G. Milligan and M. Cooper. A study of the comparability of external criteria for hierarchical cluster analysis. *Multivariate Behavioral Research*, 1986.
13. R. Ng and J. Han. Efficient and effective clustering methods for spatial data mining. *VLDB*, 1994.
14. X. Qi and B. Davison. Knowing a web page by the company it keeps. *CIKM*, 2006.
15. H. Small. Co-citation in the scientific literature: A new measure of the relationship between two documents. *American Social Info Science*, 1973.
16. T. Wu, Y. Chen, and J. Han. Association mining in large databases: A re-examination of its measures. *PKDD*, 2007.
17. X. Xu, N. Yuruk, Z. Feng, and T. A. J. Schweiger. Scan: a structural clustering algorithm for networks. *KDD*, 2007.
18. O. Yi. Ehm-based web pages fuzzy clustering algorithm. *MUE*, 2007.
19. M. K. Yitong Wang. Evaluating contents-link coupled web page clustering for web search results. *CIKM*, 2002.
20. O. Zamir and O. Etzioni. Web document clustering: a feasible demonstration. *SIGIR*, 1998.