6-2005

Automatically Discovering the Number of Clusters in Web Page Datasets

Zhongmei Yao
University of Dayton, zyao01@udayton.edu

Automatically Discovering the Number of Clusters in Web Page Datasets

Zhongmei Yao & Ben Choi

Computer Science, College of Engineering and Science
Louisiana Tech University, Ruston, LA 71272, USA
mayyao@cs.tamu.edu, pro@BenChoi.org

Abstract

Clustering is well suited for Web mining by automatically organizing Web pages into categories each of which contains Web pages having similar contents. However, one problem in clustering is the lack of general methods to automatically determine the number of categories or clusters. For the Web domain in particular, currently there is no such method suitable for Web page clustering. In an attempt to address this problem, we discover a constant factor that characterizes the Web domain, based on which we propose a new method for automatically determining the number of clusters in Web page datasets. We discover that the measure of average inter-cluster similarity reaches a constant of 1.7 when all our experiments produced the best results for clustering Web pages. We determines the number of clusters by using the constant as the stopping factor in our clustering process by arranging individual Web pages into clusters and then arranging the clusters into larger clusters and so on until the average inter-cluster similarity approaches the constant. Having the new method described in this paper together with our new Bidirectional Hierarchical Clustering algorithm reported elsewhere, we have developed a clustering system suitable for mining the Web.

Keywords: Web Mining, Clustering, Classification, Information Retrieval, Knowledge Discovery

1. Introduction

We are interested in cluster analysis that can be used to organize Web pages into clusters based on their contents or genres [1]. Clustering is an unsupervised discovery process for partitioning a set of data into clusters such that data in the same cluster is more similar to one another than data in other clusters [2-4]. Typical application areas for clustering include artificial intelligence, biology, data mining, information retrieval, image processing, marketing, pattern recognition, and statistics [2-4]. Compared to classification methods, cluster analysis has the advantage that it does not require any training data (i.e. the labeled data), but can achieve the same goal in that it can classify similar web pages into groups.

The major aspects of the clustering problem for organizing web pages are: to find the number of clusters, k, in a webpage dataset; and to assign web pages accurately to their clusters. Much work [5-21] has been done to improve the accuracy of assigning data to clusters in different domains, whereas no satisfactory method has been found to estimate k in a dataset [5,22] though many methods were proposed [22-33]. As a matter of fact, finding k in a dataset is still a challenge in cluster analysis [5]. Almost all work in this area assumes that k is known for clustering a dataset [5-20]. However in many applications, this is not true because there is little prior knowledge available for cluster analysis except the feature space or the similarity space of a dataset.

This paper addresses the problem of estimating k for Web page datasets. By testing many existing methods for estimating k for datasets, we find only the average inter-cluster similarity (*avgInter*) can be used as the criterion to discover k for a Web page datasets. Our experiments show that when the *avgInter* for a Web page dataset reaches a constant threshold, the clustering solutions for different datasets from the Yahoo! directory are measured to be the best. Compared to other criterions, e.g., the maximal or minimal inter-cluster similarity among clusters, *avgInter* implies a characteristic for Web page datasets.

The rest of this paper is organized as follows. Section 2 gives background and an overview of related methods for estimating the number of clusters for datasets. Section 3 describes the Web page datasets used in our experiments. Section 4 provides the experimental details for the discovery of a constant factor that characterized the Web domain. Section 5 shows how the constant factor is used for automatically discovering the number of clusters. And, Section 6 provides the conclusion and future research.

This research was funded in part by a grant from the Center for Entrepreneurship and Information Technology (CEnIT), Louisiana Tech University.

Proceedings of the 2005 International Conference on Data Mining, pp. 3-9, June 20-23, 2005

2. Background and Related Methods

In this section we first give the necessary background of cluster analysis and then briefly review existing methods for estimating the number of clusters in a dataset.

The task of clustering can be expressed as follows [2-4]. Let n be the number of objects, data points, or samples in a dataset, m the number of features for each data point d_i with $i \in \{1,...,n\}$, and k be the desired number of clusters to be recovered. Let $l \in \{1,...,k\}$ denote the unknown cluster label and C_l be the set of all data points in the l cluster. Given an m-dimensional data point, the goal is to estimate the number of clusters k and to estimate its cluster label l such that similar data points have the same label. Hard clustering assigns a label to each data point while soft clustering assigns the probabilities of being a member of each cluster to each data point. In the next following subsections we present an overview of several common methods for estimating k for a dataset.

Calinski and Harabasz [23] defined an index, *CH(k)*, to be

$$CH(k) = \frac{trB(k)/(k-1)}{trW(k)/(n-k)} \quad (1)$$

Where *tr* represents the trace of a matrix, *B(k)* is the between cluster sum of squares with k clusters and *W(k)* is the within cluster sum of squares with k clusters [24]. $\arg\max_{k\geq 2} CH(k)$ is determined to be the number of clusters for a dataset.

Krzanowski and Lai [25] defined the following indices for estimating k for a dataset:

$$diff(k) = (k-1)^{2/m} trW_{k-1} - k^{2/m} trW_k \quad (2)$$

$$KL(k) = \frac{|diff(k)|}{|diff(k+1)|} \quad (3)$$

where m is number of features for each data point. The number of clusters for a dataset is estimated to be $\arg\max_{k\geq 2} KL(k)$.

The Silhouette width is defined in [26] to be a criterion for estimating k in a dataset as follows.

$$sil(i) = \frac{b(i)-a(i)}{\max(a(i),b(i))} \quad (4)$$

where *sil(i)* means the silhouette width of data point i, *a(i)* denotes the average distance between i and all other data in the cluster which i belongs to, and *b(i)* represents the *smallest* average distance between i to all data points in a cluster. The data with large *sil(i)* is well clustered. The overall average silhouette width is defined by $\overline{sil} = \sum_i sil_i / n$ (where n is the number of data in a dataset). Each k ($k\geq 2$) is associated with a $\overline{sil_k}$ and the k is selected to the right number of clusters for a dataset which has the largest $\overline{sil}$ (i.e. $k = \arg\max_{k\geq 2} \overline{sil_k}$).

Similarly Strehl [5] defined the following indices:

$$avgInter(k) = \sum_{i=1}^{k} \frac{n_i}{n-n_i} \sum_{j\in\{1,...,i-1,i+1,...,k\}} n_j \cdot Inter(C_i, C_j) \quad (5)$$

$$avgIntra(k) = \sum_{i=1}^{k} n_i Intra(C_i) \quad (6)$$

$$\phi(k) = 1 - \frac{avgInter(k)}{avgIntra(k)} \quad (7)$$

where *avgInter(k)* denotes the weighted average inter-cluster similarity, *avgIntra(k)* denotes the weighted average intra-cluster similarity, *Inter*(C_i,C_j) means the inter-cluster similarity between cluster C_i with n_i data points and cluster C_j with n_j data points, *Intra*(C_i) means the intra-cluster similarity within cluster C_i, and $\phi(k)$ is the criterion designed to measure the quality of clustering solution. The *Inter*(C_i,C_j) and *Intra*(C_i) are given by [5]

$$Inter(C_i, C_j) = \frac{1}{n_i n_j} \sum_{d_a\in C_i, d_b\in C_j} sim(d_a, d_b) \quad (8)$$

$$Intra(C_i) = \frac{2}{(n_i-1)n_i} \sum_{d_a, d_b\in C_i} sim(d_a, d_b) \quad (9)$$

where d_a and d_b represent data points. To obtain high quality with small number of clusters, Strehl also designed a penalized quality $\phi^T(k)$ which is defined as

$$\phi^T(k) = (1-\frac{2k}{n})\phi(k). \quad (10)$$

The number of clusters in a dataset is estimated to be $\arg\max_{k\geq 2} \varphi^T(k)$.

It can be noticed that the above methods can not be used for estimating k=1 for a dataset. Some other methods, e.g. Clest [22], Hartigan [27], and gap [28] were also found in literature. In summary most existing methods make use of the distance (or similarity) of inter-cluster and (or) intra-cluster of a dataset. The problem is that none of them is satisfactory for all kinds of cluster analysis [5, 22]. The reason is that so far people still have problems in how a cluster is well defined [28]. Different opinions exist about the granularity of clusters and there may be several right answers to k with respect to different desired granularity. Unlike partitional (flat) clustering algorithms, hierarchical clustering algorithms may have different k's by cutting the dendrogram at different levels.

In the next section we will report the testing results of estimating k for Web page datasets, which consists of pretty well-separated clusters. Throughout this paper, we use term "documents" or "Web pages" to denote Web pages, use term "true class" to mean a class of web pages which contains web pages labeled

with the same class label, and use "cluster" to denote a group of Web pages in which Web pages may have different class labels.

3. Web Page Datasets for Experiments

We conducted experiments with different methods of estimating *k* on web page datasets. For our experiments, we generated four Web page datasets (see Table 1) taking from Yahoo.com. The first dataset, *DS1*, contains 766 web pages which are randomly selected from two true classes: *agriculture* and *astronomy*. This dataset is designed to show our method of estimating *k* for a dataset which consists of clusters of widely different sizes: the number of web pages from the *astronomy* true class is about ten times the number of web pages from the *agriculture* true class. The second dataset, *DS2*, contains 664 web pages from 4 true classes. The third dataset, *DS3*, includes 1215 web pages from 12 true classes. In order to show the performance on a more diverse dataset, we produce the forth dataset, *DS4*, which consists of 2524 web pages from 24 true classes. After we remove stop words and conduct reduction of dimensionality [21], the final dimension for each dataset is listed in Table 1.

Table 1. Compositions of four Web page datasets

DS1: true classes = 2, the number of web pages= 766, dimension= 1327

true class (the number of web pages):
agriculture(73) astronomy(693)

DS2: true classes = 4, the number of web pages=664, dimension=1362

astronomy(169) biology(234) alternative(119) mathematics(142)

DS3: true classes = 12, the number of web pages = 1215, dimension= 1543

agriculture(108) astronomy(92) evolution(74) genetics(108) health(127) music(103) taxes(80) religion(113) sociology(110) jewelry(108) network (101) sports(91)

DS4: true classes = 24, the number of web pages = 2524, dimension= 2699

agriculture(87) astronomy(96) anatomy(85) evolution(76) plants(124) genetics(106) mathematics(106) health(128) hardware(127) forestry(68) radio(115) music(104) automotive(109) taxes(82) government(147) religion(114) education(124) art(101) sociology(108) archaeology(105) jewelry(106) banking(72) network (88) sports(146)

4. Discovery of a Constant Factor

We apply our Bidirectional Hierarchical Clustering (BHC) algorithm [21, 42] to cluster the Web page datasets. It consists of the following major steps:

1) Generating an initial sparse graph;
2) Bottom-up merging clusters; and
3) Top-down refining clusters.

First, it generates an initial sparse graph, G_0, where a node (or vertex) represents a cluster, and is connected only to its k-nearest neighbors by similarity-weighted edges. It then creates a hierarchical structure of clusters for a dataset in the two directional phases, the bottom-up cluster-merging phase and the following top-down refinement phase. During the bottom-up cluster-merging phase, it transfers the initial graph G_0 into a sequence of smaller graphs by grouping nodes into a new node in the next smaller graph. This grouping process requires a stopping factor that will be described. After the bottom-up cluster-merging phase is completed, the top-down refinement phase then eliminates the early errors that may occur in the greedy bottom-up cluster-merging phase by minimizing the

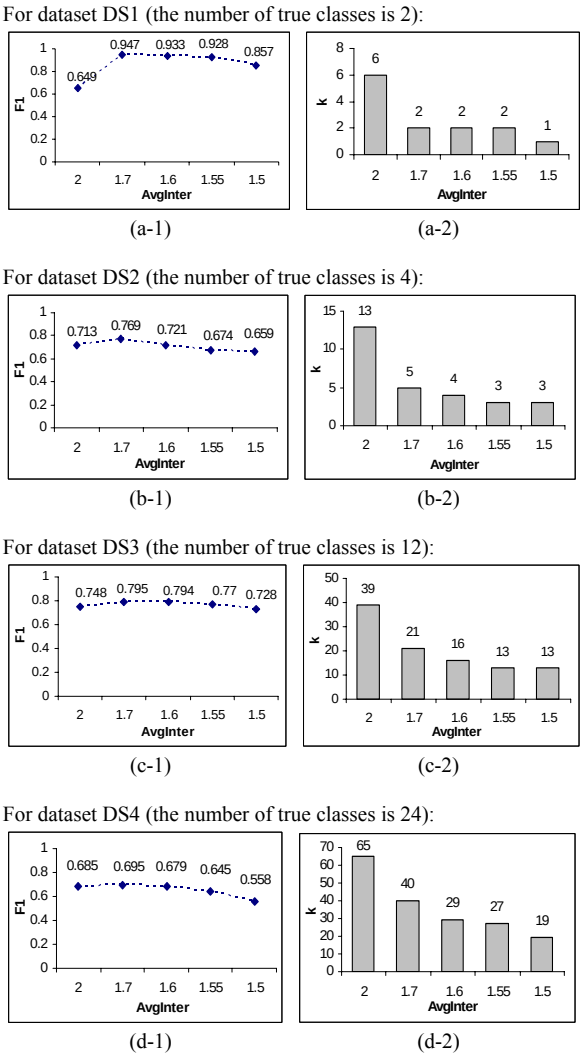

Figure 1. The impact of *avgInter* on the clustering performances for four Web page datasets.

inter-cluster similarities of clusters. The key features of our system are that it finds the hierarchical structure of clusters much faster than the existing hierarchical agglomerative clustering algorithms, and it improves the cluster solution by processing a refinement procedure [21, 42].

For all experiments, we use the metric, F_1 measure [6, 34], which makes use of the true class labels of web pages, to measure the quality of clusters in a Web page dataset. The F_1 measure indicates how well a cluster solution matches the true classes in the real world (e.g. the Yahoo! directory). In general, the greater F_1 score, the better clustering solution.

In our experiments we test the existing methods *CH(k)*, *KL(k)*, $\overline{sil_k}$, $\phi(k)$ and $\phi^T(k)$ (see Section 2) to discover k for Web page datasets. These five indices are computed for different k's for a Web page dataset. However, none of them work well. Our tests results showed that for any dataset in Table 1 their estimated k is more than 5 times different from the true number of classes in the Web page datasets and the corresponding cluster solutions have lower than 0.3 F_1 score.

After many trials, we find that *avgInter(k)* for any dataset in Table 1 reaches a common threshold of 1.7, when the F_1 measure of the cluster solution for a dataset is greatest. The relation between the thresholds of *avgInter(k)* and the F_1 scores of a cluster solution, and the relation between the thresholds of *avgInter(k)* and k's for the four Web page datasets are illustrated in Figure 1.

Table 2. The clustering solution for dataset *DS4*.
(F_1 scores are given only for 24 clusters because those clusters represent true classes in dataset *DS4*. The purity and the top three descriptive terms are given for each cluster.)

cluster	The number of web pages	the majority's true class label	purity	F_1	top 3 descriptive terms
C_1	106	Astronomy	0.840	0.881	moon, mar, orbit
C_2	29	Agriculture	0.793	0.397	pest, weed, pesticid
C_3	24	Agriculture	0.917		crop, wheat, agronomi
C_4	64	Anatomy	0.906	0.779	anatomi, muscl, blood
C_5	64	Evolution	0.750	0.686	evolut, darwin, erectu
C_6	116	Plants	0.776	0.750	plant, flower, garden
C_7	161	Genetics	0.565	0.682	genom, genet, clone
C_8	101	Mathematics	0.782	0.763	mathemat, math, algebra
C_9	94	Health	0.649	0.550	mental, therapi, health
C_{10}	32	Health	0.875		grief, bereav, heal
C_{11}	115	Hardware	0.452	0.430	font, px, motherboard
C_{12}	21	Hardware	0.857		keyboard, pc, user
C_{13}	83	Forestry	0.675	0.742	forest, forestri, tree
C_{14}	86	Radio	0.709	0.607	radio, broadcast, fm
C_{15}	70	Music	0.800	0.644	guitar, music, instrum
C_{16}	13	Music	1.000		drum, rhythm, indian
C_{17}	86	Automotive	0.849	0.749	car, auto, automot
C_{18}	20	Automotive	0.800		motorcycl, bike, palm
C_{19}	120	Taxes	0.633	0.752	tax, incom, revenu
C_{20}	155	Government	0.806	0.828	congressman, hous, district
C_{21}	108	Religion	0.824	0.802	christian, bibl, church
C_{22}	92	Education	0.761	0.648	montessori, school, educ
C_{23}	43	Education	0.767		homeschool, home school, curriculum
C_{24}	60	Art	0.833	0.621	paint, canva, artist
C_{25}	89	Sociology	0.831	0.751	sociologi, social, sociolog
C_{26}	59	Archaeology	0.864	0.622	archaeologi, archaeolog, excav
C_{27}	18	Archaeology	0.722		egypt, egyptian, tomb
C_{28}	120	Jewelry	0.817	0.867	jewelri, bead, necklac
C_{29}	91	Banking	0.659	0.736	bank, banker, central bank
C_{30}	92	Network	0.565	0.578	network, dsl, storag
C_{31}	159	Sports	0.824	0.859	soccer, footbal, leagu
C_{32}	1	Religion	1.000		struggl, sex, topic
C_{33}	8	Religion	0.250		domain, registr, regist
C_{34}	10	Plants	0.300		florida, loui, ga, part, pioneer,
C_{35}	1	Archaeology	1.000		guestbook, summari, screen
C_{36}	3	Genetics	0.333		pub, patch, demo
C_{37}	3	Music	0.333		bell, slide, serial
C_{38}	1	Sociology	1.000		relief, portrait, davi
C_{39}	2	Music	0.500		ontario, predict, archaeolog
C_{40}	4	Music	0.250		unix, php, headlin
overall	2524		0.740	0.698	

In Figure 1 (a-1), (b-1), (c-1) and (d-1), the F_1 scores of cluster performances for the four datasets reach the maximal values when the threshold of *avgInter* is 1.7, and further increasing or reducing the threshold of *avgInter* would only worsen the F_1 scores for the datasets *DS1*, *DS2*, *DS3* and *DS4*. In other words, once the weighted average inter-cluster similarity (*avgInter*) reaches the common threshold, 1.7, the cluster solution is found to be best for a Web page dataset. This shows that, unlike other index such as *CH(k)*, *KL(k)*, $\overline{sil_k}$, or $\phi^T(k)$, *avgInter* implies a common characteristic in different Web page datasets.

Figure 1 (a-2), (b-2), (c-2) and (d-2) show the *k*'s for four Web page datasets produced by setting different thresholds for *avgInter*. In Figure 1 (a-2) it is shown that the *avgInter* method is able to find k=1 while many existing methods are unable to do so. As shown in the figure, when *avgInter* reaches 1.7, the best estimated values for *k* is found to be 2 for *DS1*, 5 for *DS2*, 21 for *DS3* and 40 for *DS4*.

The estimated *k* is usually greater than the number of true classes in a Web page dataset because outliers are found and clustered into some small clusters, and a few true classes are distinguished into more than one cluster with finer granularity. This situation is exactly shown in Table 2, which shows the clustering solution for the most diverse dataset, *DS4*, obtained when the threshold of *avgInter* is 1.7. The naming for a newly formed cluster is by selecting the top three descriptive terms. The ranking of descriptive terms for a cluster is conducted by sorting the tf_{lj}'/df_j values of terms in the cluster (tf_{lj}' is defined to be the number of web pages containing term t_j in cluster C_l and df_j is the document frequency [35] of t_j). It can be noted that for most true classes, a true class has a dominant cluster in Table 2. For instance, the dominant clusters for true class *astronomy, anatomy and evolution* are cluster C_1, C_4 and C_5, respectively. We can see several true classes have been distinguished more precisely into more than one cluster; e.g. true class *automotive* has been separated into cluster C_{17} which is more related to *car* and *auto*, and cluster C_{18} more related *motorcycle* and *bike*, as indicated by their top descriptive terms. Similar situation happens to true class *agriculture*, *health*, *education* and *archaeology*, each of which has been distinguished into two clusters. As shown in Table 2 outliers can be found as cluster C_{32}, C_{33}, …, and C_{40}. These clusters have poor purity [36] scores.

5. Discovering the Number of Clusters

The constant factor described in the last section can be used to estimate the number of clusters in a clustering process. The number of clusters *k* for a Web page dataset is estimated to be

$$\arg\max_k(avgInter(k) \le 1.7) \text{ where } 1 \le k \le n. \quad (11)$$

The *avgInter(k)* is computed for different *k*'s. The *k* that results in *avgInter(k)* as close to (but less than) the threshold 1.7 is selected to be the final *k* for a Web page dataset.

For our Bidirectional Hierarchical Clustering system [21, 42], we determines the number of clusters by using the constant as the stopping factor in the clustering process. Our hierarchical clustering process starts by arranging individual Web pages into clusters and then arranging the clusters into larger clusters and so on until the average inter-cluster similarity *avgInter(k)* approaches the constant. As clusters are grouped to form larger clusters the value of *avgInter(k)* is reduced. This grouping process (bottom-up cluster-merging phase [21, 42]) is stopped when *avgInter(k)* approaches 1.7. The final number of clusters is automatically obtained as the result.

6. Conclusion and Future Research

Although many methods of finding the number of clusters for a dataset have been proposed, none of them is satisfactory for clustering Web page datasets. Finding the number of clusters for a dataset is often treated as an ill-defined question because it is still questionable how well a cluster should be defined. By recognizing this status, we preferred hierarchical clustering methods, which allow us to view clusters at different levels with coarser granularity at the higher level and finer granularity at the lower level. For Web mining in particular, our Bidirectional Hierarchical Clustering method [21, 42] is able to arrange Web pages into directory tree that allows users to browse the results in different levels of granularities.

In this paper we investigated the problem of estimating the number of clusters, *k*, for Web page datasets. After many trials, we discovered that the average inter-cluster similarity (*avgInter*) can be used as a criterion to estimate *k* for Web page datasets. Our experiments showed that when the *avgInter* for a Web page dataset reaches a threshold of 1.7, the clustering solutions achieve the best results. Compared to other criterions, *avgInter* implies a characteristic for Web page datasets. We then use the threshold as a stopping factor in our clustering process to automatically discovering the number of clusters in Web page datasets.

The future work includes investigating using our *avgInter* method on datasets from domains other than Web pages. Having the new method described in this paper together with our new Bidirectional Hierarchical Clustering algorithm reported in [21, 42], we have

developed a clustering system suitable for mining the Web. We plan to incorporate the new clustering system into our Information Classification and Search Engine [37-42].

References

1. Choi, B. and Yao, Z. (2005). Book Chapter on "*Web Page Classification*". Recent Advances in Data Mining and Granular Computing (mathematical aspects of knowledge discovery), Springer-Verag (in print).
2. Everitt, B. S., Landua, S. and Leese, M. (2001). *Cluster Analysis*, Arnold, London Great Britain.
3. Jain, A. K. Murty, M. N. and Flynn, P. J. (1999). "*Data Clustering: A Review*". ACM Computing Surveys, Vol. 31, No. 3, pp.255-323.
4. Berkhin, P. (2002). "*Survey of Clustering Data Mining Techniques*". Technical Report, Accrue Software.
5. Strehl, A. (2002). "*Relationship-based Clustering and Cluster Ensembles for High-dimensional Data Mining*". Dissertation, The University of Texas as Austin.
6. Zhao, Y. and Karypis, G. (1999). "*Evaluation of Hierarchical Clustering Algorithm for Document Datasets*". Computing Surveys, 31(3), pp. 264-323.
7. Karypis, G., Han, E.-H., and Kumar, V. (1999). "*CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling*". IEEE Computer, 32(8), pp.68-75.
8. Dhillon, I., Fan S. J. and Guan, Y. (2001). "*Efficient Clustering of Very Large Document Collections*", Data Mining for Scientific and Engineering Applications, Kluwer Academic Publisher.
9. Ng, R. and Han, J. (1994). "*Efficient and Effective Clustering Methods for Spatial Data Mining*". 20th International Conference on Very Large Data Bases (VLDB-94), Santiago, pp. 144-155.
10. Karypis, G. and Kumar, V. (1999). "*A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs*". SIAM Journal of Scientific Computing, 20(1), pp. 359-392.
11. Ester, M., Kriegel, H. P., Sander, J. and Xu, X. (1996). "*A Density-based Algorithm for Discovering Clusters in Large Spatial Database with Noise*". International Conference on Knowledge Discovery in Databases and Data Mining (KDD-96), AAAI Press, Portland, Oregon, pp. 226-231.
12. Sander, J., Ester, M., Kriegel, H. P. and X. Xu, (1998). "*Density-based Clustering in Spatial Databases: The Algorithm GDBSCAN and its Applications*". Data Mining and Knowledge Discovery 2(2), pp. 169-194.
13. Agrawal, R., Gehrke, J., Gunopulos, D. and Raghavan, P. (1998). "*Automatic Subspace Clustering for High Dimensional Data for Data Mining Applications*". Proceedings of the 1998 ACM SIGMOD Conference on Management of Data, pp. 94-105.
14. Hinneburg, A. and Keim, D. A. (1999). "*An Optimal Grid-clustering: Towards Breaking the Curse of Dimensionality in High-dimensional Clustering*". Proceedings of 25th International Conference on Very Large Data Bases (VLDB-99), pp. 506-517.
15. Tantrum, J., Murua, A. and Stuetzle, W. (2002). "*Hierarchical Model-Based Clustering of Large Datasets through Fractionation and Refractionation*". The 8th ACM SIGKDD International Conference on Knowledge and Discovery and Data Mining Location (SIGKDD '02), Canada, pp. 183-190.
16. Zhang, T., Ramakrishnan, R. and Linvy, M. (1996). "*BIRCH: an Efficient Data Clustering Method for Very Large Databases*". Proceedings of the ACM SIGMOD Conference on Management of Data, Montreal, Canada, pp. 103-114.
17. Guha, S., Rastogi, R. and Shim, K. (1998). "*CURE: A Clustering Algorithm for Large Databases*". Proceedings of the 1998 ACM SIGMOD Conference on Management of Data, pp73-84.
18. Guha, S., Rastogi, R. and Shim, K. (1999). "*ROCK: A Robust Clustering Algorithm for Categorical Attributes*". Proceedings of the 15th International Conference on Data Engineering, pp. 512-521.
19. Yao Y. and Karypis, G. (2001). "*Criterion Functions for Document Clustering: Experiments and Analysis*". Technical Report TR#01-40, Department of Computer Science, University of Minnesota, Minneapolis.
20. Rajaraman, K. and Pan, H. (2000). "*Document Clustering using 3-tuples*". PRICAI'2000 International Workshop on Text and Web Mining, Melbourne, Australia, pp. 88-95.
21. Yao, Z. (2004). *Bidirectional Hierarchical Clustering for Web Browsing*. Master thesis, Louisiana Tech University.
22. Dudoit, S. and Fridlyand, J. (2002). "*A Prediction-Based Resampling Method to Estimate the Number of Clusters in a Dataset*". Genome Biology, 3(7), 0036.1- 0036.21.

23. Davies, D and Bouldin D (1979). *A Cluster Separation Measure*. IEEE Trans. Pattern Analysis Machine Intelligence, I: 224-227.
24. Mardia, K.V., Kent, J. T. and Bibby, J. M. (1979). *Multivariate Analysis*. San Diego: Academic Press.
25. Krzanowski, W. and Lai, Y. (1985). *A Criterion for Determining the Number of Groups in a Dataset Using Sum of Squares Clustering*. Biometrics, 44:23-34.
26. Kaufman, L., Rousseeuw, P. J. (1990). *Finging Groups in Data: An Introduction to Cluster Analysis*. New York: Wiley.
27. Hartigan, J. A. (1985). "*Statistical Theory in Clustering*". Journal of Classification, 2, 63-76.
28. Tibshirani R., Walther, G. and Hastie, T. (2000). *Estimating the Number of Clusters in a Dataset via the Gap Statistic*. Technical Report, Department of Bio-statistics, Stanford University.
29. Fraley, C. and Raftery, A. (1998). H*ow Many Clusters? Which Clustering Method? – Answers via Model-based Cluster Analysis*. Technical Report 329, Department of Statistics, University of Washington.
30. Milligan, G. W. and Cooper, M. C. (1985). *An Examination of Procedures for Determining the Number of Clusters in a Data Set*. Psychometrika, 50:159-179.
31. Bock, H. H. (1985). *On some Significance Tests in Cluster Analysis*. Journal of Classification, 2:77-108.
32. Gordon, A. (1999). *Classification* (2nd edition). Chapman and Hall / CRC press, London.
33. Jain, A. K. and Dubes, R. C. (1988). *Algorithms for Clustering Data*. Englewood Cliffs, NJ: Prentice-Hall.
34. Larsen, B. and Aone, C. (1999). "*Fast and effective text mining using linear-time document clustering"*. Proceedings of the 5th ACM SIGKDD, Int'l Conference on Knowledge Discovery and Data Mining, pp. 16-22.
35. Yang, Y. and Pedersen, J. O. (1997). "*A Comparative Study on Feature Selection in Text Categorization*". Proceedings of the 14th International Conference on Machine Learning (ICML97) pp. 412-420.
36. Strehl, A. Ghosh, J. and Mooney, R. (2000). "*Impact of Similarity Measures on Web-page clustering*". AAAI-2000: Workshop of Artificial Intelligence for Web Search.
37. Choi, B. (2001). "*Making Sense of Search Results by Automatic Web-page Classifications*," WebNet 2001, pp.184-186.
38. Choi, B and Peng, Xiaogang (2004) "*Dynamic and Hierarchical Classification of Web Pages*," Online Information Review, Vol. 28, No. 2, pp. 139-147.
39. Choi, B. and Guo, Q. (2003) "*Applying Semantic Links for Classifying Web Pages*," Developments in Applied Artificial Intelligence, IEA/AIE 2003, Lecture Notes in Artificial Intelligence, Vol. 2718, pp. 148-153.
40. Choi, B and Dhawan, R (2004) "*Agent Space Architecture for Search Engines*". The 2004 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, September 2004.
41. Baberwal, S. and Choi, B. (2004) "*Speeding up Keyword Search for Search Engines*," The 3rd IASTED International Conference on Communications, Internet, and Information Technology, pp. 255-260.
42. Yao, Z and Choi, B. (2003). "*Bidirectional Hierarchical Clustering for Web Mining*," IEEE/WIC International Conference on Web Intelligence, pp. 620-624.