Visual Analysis of Clustering Algorithms A Methodology and a Case Study

Guillaume Artignan, Mountaz Hascoët

Visual Analysis of Clustering Algorithms A methodology and a case study

Guillaume Artignan, Mountaz Hascoët

LIRMM, UMR 5506 du CNRS, Univ. Montpellier II
161, rue Ada 34392 MONTPELLIER Cedex, France
{artignan,mountaz}@lirmm.fr

Abstract. Clustering is probably one of the most frequently used approaches when facing a scaling problem in large collections of documents. In many situations, however, the choice of the most appropriate algorithm for clustering can turn into a real dilemma. Numerical criteria have been proposed to evaluate the quality of the results of clustering algorithms. However, so many different criteria have been proposed that the dilemma even worsens. Most criteria reveal different aspects of the quality of the results and hide others. The aim of this paper is to help with the understanding of clustering and to facilitate the comparison and the choice of clustering algorithm for a given purpose. Our proposal consists in studying both quality evaluation criteria and clustering algorithms. We start by discussing a selected set of representative criteria, and further conduct a case study on a large set of real data, measuring not only the quality of different representative clustering algorithms but also the impact of each criterion on the ranking of the algorithms. By providing empirical results on large scale corpus of either documents or lexical networks useful to digital library, we hope to clarify the field and facilitate designers' choices.

1 Introduction

In digital libraries, clustering can be used at various stages. Either as a post treatment for a search engine to cluster the results instead of displaying ranked lists, or as a tool to automatically extract thesauri from a set of documents. It can also be used to automatically organize a collection of documents into a catalog or a directory. In all cases, the designer eager to perform clustering has to choose amongst thousands of algorithms. As recalled by Jain in his recent review on that subject [10], "There is no best clustering algorithm". Indeed, the choice of the appropriate algorithm for one purpose is multi-factorial. Amongst all possible important factors that would impact the choice, the quality or nature of the results is probably the most important, and at the same time, the most obscure and difficult to evaluate. Surprisingly, very few efforts have been made to help non clustering experts to understand what is at stake and how to compare the quality of various clustering algorithms. Our aim is to report

the results of a case study conducted on large information collections relevant for most designers of information systems. Because our aim is to evaluate the quality of clustering, the data on which we have worked was chosen carefully. The first set of data was extracted from "Jeux de Mots", a lexical network of the French language. Jeux de Mots is one of the most accurate and complete publicly available lexical network for French [11]. The second set of data is extracted from papers published in the information retrieval field since 1980. Our approach to the multitude of models underlying clustering approaches is to use graph based representation as a common ground to simplify and unify notations. Most current data sets used in digital libraries can be conceptually represented by graphs and most current clustering algorithms can be simplified using graph based representations. Our methodology for analyzing the quality of various approaches is experimental and exploratory. For that purpose visual analysis was used to explore the results of our experiments. In this paper, we first describe the datasets used for the experiments. We further describe and discuss the quality criteria used to analytically evaluate the results. We then rapidly review the algorithms selected for the experiment. Finally, the two last sections report the results, and further discuss them.

2 Datasets

Four datasets are derived from "Jeux de Mots", *JdmAll* contains *Jdm2000* which contains *Jdm200* containing at his turn *Jdm20* and that they respectively contains 111701 nodes, 2000 nodes, 200 nodes and 20 nodes. We built two datasets from a corpus of 635 research papers in the field of information retrieval and digital libraries. We compute similarities between each pair of documents using the TF-IDF measure [15] and a Pearson's correlation. A complete graph is then obtained, where nodes are document and similarities are weighted links. We construct Sig1000 and Sig10000 by keeping respectively the 1000 and 10000 best similarity relations. For all these six datasets we studied the degree distribution. Jdm200, Jdm2000, JdmAll and Sig1000 have a degree distributions following a power-law tails. We note γ the exponent. The Tab. 2 describes the datasets by providing the name, the number of nodes N, the total number of edges E, the exponent γ, the graph diameter D, the averaged clustering coefficient C, an URL describing the different dataset and proposing a link for downloading them.

Name	$\|N\|$	$\|E\|$	γ	*Diam.*	C	URL
JDM 20	20	19	-1.0	6	0.0	`http://<Anonymized\forthe\reviewprocess>/icadl/jdm20`
JDM 200	200	265	-1.58	11	0.1140	`http://<Anonymized\forthe\reviewprocess>/icadl/jdm200`
JDM 2000	2000	3476	-1.8	13	0.1357	`http://<Anonymized\forthe\reviewprocess>/icadl/jdm2000`
JDM ALL	111701	441854	-1.9	13	0.1933	`http://<Anonymized\forthe\reviewprocess>/icadl/jdmall`
SIG 1000	378	903	-1.48	20	0.3928	`http://<Anonymized\forthe\reviewprocess>/icadl/sig1000`
SIG 10000	626	10000	-0.52	5	0.4002	`http://<Anonymized\forthe\reviewprocess>/icadl/sig10000`

Table 1. Datasets

3 Quality criteria

Criteria for the evaluation of the quality of clustering may vary widely in terms of notations and subtely in terms of concept. We propose notations that can express different criteria in a consistent notation way to help with their comparison. The notation is based on graph theory basic concepts. A graph G is composed of a set of nodes denoted by N and a set of edges denoted by E that represent links between nodes. Applying clustering to G usually results in k clusters denoted by $\{C_1...C_k\}$ as k subsets of N.

To describe the six different criteria selected from the litterature and used in the experiment, we furtherintroduce the following notations:

- n is the number of nodes in G,
- e is the number of edges in G,
- k is the number of clusters,
- n_i is the number of nodes in the cluster C_i,
- we_i is the number of edges *w*ithin the cluster C_i,
- oe_i is the number of edges *o*utgoing from the cluster C_i
- be_{ij} is the number of edges *b*etween the two clusters C_i and C_j
- pe_i is the number of possible edges between two different nodes in C_i. For an undirected graph : $pe_i = n_i(n_i-1)/2$. For a directed graph : $pe_i = n_i(n_i-1)$.
- me_i is the number of missing edges in C_i. Hence, $me_i = pe_i - we_i$.
- we, be,pe and me are the total number of respectively *w*ithin cluster edges, *b*etween cluster edges *p*ossible edges and *m*issing edges.
- Therefore, $we = \sum_{i=1}^{k} we_i$, $be = \sum_{i=1}^{k} be_i$, $me = \sum_{i=1}^{k} me_i$.

Cut [4] is computed as the number of between-edges (also called extra-edges) over the number of within-edges(also called intra-edges). Lowest values correspond to best clustering results.

$$Cut(G) = \frac{be}{we} \tag{1}$$

Perf [4] takes into accounts for the number of links between clusters, as well as the number of missing within edges, e.g nodes grouped in the same cluster without edges relating each other.

$$Perf(G) = 1 - \frac{be + me}{pe} \tag{2}$$

Cond [4] criterion is an average over the conductance of each pair of clusters. The conductance of a pair of cluster Ci and Cj is the proportion of edges between Ci to Cj divided by the minimum number of edges within Ci and Cj. Lowest values corresponds to best clustering results. We consider we_i or we_j equals to one in case of singleton clusters.

$$Cond(G) = \frac{\sum_{i<j}^{k} \frac{be_{ij}}{min(we_i, we_j)}}{k(k-1)/2} \tag{3}$$

Cov [4] is the ratio of number of within-edges to number total edges in the graph.

$$Cov(G) = \frac{we}{e} \tag{4}$$

MQ [4] MQ is a difference between the average within-cluster edge density and between-cluster edge density. Therefore it varies between -1 and +1, and highest values correspond to best clustering results. In case of singleton cluster, we_i and pe_i equal 0. In this case, we do not compute we_i/pe_i but use the value of 1 instead.

$$MQ(G) = \frac{\sum_{i=1}^{k}(we_i/pe_i)}{k} - \frac{\sum_{i<j}^{k}(be_{ij}/n_i n_j)}{(k(k-1))/2} \tag{5}$$

Mod [5] can be considered as a measure of Cov defined above corrected by the Cov computed for a random clustering of the same graph and that we note $rCov$. Therefore, the highest values for Mod correspond to best clustering results according to Cov and values below 0 correspond to clustering worse than random according to the Cov criteria. However, the computation of $rCov$ is still debatable and would deserve a discussion that would lead us beyond the scope of this paper.

$$Mod(G) = Cov - rCov \tag{6}$$

4 Clustering Algorithms

Clustering has a huge and multidisciplinary history since it has been used in many scientific fields including in information retrieval [19], data visualization [1], physics [5], etc. Several surveys have reviewed partially this litterature [20, 10, 16]. In order to choose the algorithms to be tested in our study we had three criteria in mind. First, source code for the proposed algorithm is provided by authors or the description of the algorithm is sufficiently clear, complete and precise to be implemented. Second, the algorithm is relevant to clustering data such as documents or keywords. Third the set of algorithms tested should be representative of different types of approaches. The Tab. 3 summarizes the choices made in terms of algorithms and indicate the URL of the implementation used in the experiment.

Algorithm Name	Article	Implementation	Author's Impl.
CNM	[5]	`http://www.cs.unm.edu/~aaron/research/fastmodularity.htm`	Yes
SPK-MEANS	[6]	`http://www.cs.utexas.edu/users/dml/datamining/spkmeans.html`	Yes
Cluto	[21]	`http://glaros.dtc.umn.edu/gkhome/views/cluto`	Yes
LinLog	[13]	`http://www.informatik.tu-cottbus.de/~an/GD/`	Yes
InfoMap	[14]	`http://www.tp.umu.se/~rosvall/code.html`	Yes
CMJA	[2]	our implementation (link after blind reviews)	No
BGLL	[3]	`http://sites.google.com/site/findcommunities/`	Yes
Simple K-Means	[12]	our implementation (link after blind reviews)	No
NCut Algorithm	[18]	`http://www.cis.upenn.edu/~jshi/software/`	Yes
MCL	[7]	`http://www.arbylon.net/projects/`	No

Table 2. Algorithm's and Implementations used in this evaluation

The CNM algorithm [5] has a bottom-up approach. Communities are made for each node and further merged iteratively merged with others to increase the criteria of modularity measure defined in equation Equ. 6. CNM results can be represented by a hierarchical clustered graph or a simple clustered graph depending on how merging is handled.

The BGLL algorithm [3] approach is very similar to CNM, but the definition of modularity differs and it makes the hierarchical clustered graph explicit as well as the level at which the clusters are extracted from the hierarchical clustered graphs.

The CMJA algorithm [2] has a different approach from the two previous. CMJA is proposed for detecting communities in small world networks by identifying weak edges. The algorithm operates in two steps. Firstly, it processes a score on each edge, this score is proportional with the number of 4-cycles and 3-cycles containing the edge. Secondly it removes the k edges with the lowest scores. Clusters are the resulting connected components.

The InfoMap approach [14] treats the problem of finding community structures in networks as an information coding problem. The approach has three steps: (1) Infomap processes a random walk on the graph and generates the random path, (2) assigns a codeword to each node in the random pass using Huffman coding [8], (3) searches a clustering minimizing the average number of bits useful to describe it.

The MCL Algorithm [7] detects communities using a Markov Matrix. The algorithm computes random walks by flow simulation. An operator named *expansion* computes n multiplications of the matrix with itself. An operator named *inflation* computes the Hadamard matrix [17].

K-Means Algorithm [12] is one of the most frequently used algorithm for clustering and many slightly different versions have been proposed. The main principle is to start with an arbitraty partition of the dataset and try to move each element to a better cluster as long as possible to improve the overall within clusters cohesion. It is one very efficient and very simple algorithm to implement. However, its based on centroid computation. This implies that its prerequisite is that computing centroids makes sense for the dataset to be clustered.

LinLog Algorithm [13] is a layout algorithm based on an energy model that aims at geometrically exhibiting clusters. Its principle is to optimize the layout accounting mainly for attraction and repulsion forces between nodes.

The NCut Algorithm [18] comes from image segmentation domain but can be adapted to graphs. Its principle is to optimize a criteria named normalized

cut, using a spectral technique.

The Cluto Toolkit[21] is a toolkit made of several clustering algorithms. Four approaches are tested in this paper: (1) The rb-based clustering approach proposed clustering computed by K-1 bisections, (2) the direct-based clustering approach, (3) an agglomerative approach, (4) the graph-based approach based on a similarity graph and a min-cut criterion.

The Spherical K-Means algorithm [6] is an extension of the well-known Euclidian K-Means algorithm. This algorithm partitions the dimension using great hypercircle.

5 Results

In this paper, we combine different approaches to compare the results of clustering. First, we unified the analytical criteria extracted from the literature to facilitate the comparison of different criteria and discuss their interpretation. Second, we empirically measured these criteria on datasets clustered by the different clustering algorithms presented in previous sections. " To analyze the results we started by computing the ranking of each algorithm according to each criteria and for each dataset. Table 3 reports average ranking over different datasets of a subset of algorithms. It also reports the aggregated ranking over all criteria and computed as the average of all criteria.

We further computed the Spearman correlation for each pair of criteria on each dataset and computed an average of all Spearman measure over all datasets. The results are reported in the diagram of Fig. 1 where average Spearman values between two criteria is written next to the edge connecting the two criteria on the diagram. For example, the average value found for Spearman correlation between cov and mod is 0.43. Overall, with the exception of 1/cut and cov, most criteria are not strictly correlated.

The variability of the results was an incentive to use a visual analysis approach to better understand the cause of variability. Therefore we used parallel coordinate diagrams [9] to interpret the results of the experiments. Figure 2 and 3 show two such diagrams corresponding to two different datasets. In these diagrams several vertical axes are used to embody different dimensions of the data explored. In our case, the vertical axes were used to embody the different

Algo	MQ	PERF	COV	CUT	COND	MOD	Total Average
CNM	4,5	4,333333333	1,666666667	4,5	3,166666667	2,333333333	3,416666667
BGLL	4,166666667	3,166666667	3,5	3	3,333333333	2	3,194444444
CMJA	1	4,5	2,833333333	3,5	3,166666667	6	3,5
InfoMap	2,166666667	1,333333333	5	1,833333333	4,333333333	4	3,111111111
LinLog	4,333333333	3,666666667	2,5	3,833333333	3,333333333	1,5	3,194444444
K-Means	3,833333333	2,833333333	3,666666667	2,5	2,166666667	4,166666667	3,194444444

Table 3. Average of rankings

quality criteria. Each item, in our case each algorithm is further represented by polyline that joins the values corresponding to that same item for all axes. These diagrams offer well-known benefits: (1) covariance/contravariance of ranking between two adjacent criteria is visually obvious, (2) the distribution of the values for each criteria is also visually obvious for all criteria and (3) it is very easy to select an algorithm that performs best for one criteria and to see how it compares to other algorithms on the other criteria.

6 Discussion

General trends : important variations of ranking over criteria and datasets. Table 3 shows the average ranking for a subset of datasets and algorithms used in the experiment. It illustrates (1) important variations over different criteria and (2) very average ranking for all algorithms when all ranking are combined. These results tend to corroborate the position of both Jain and Buxton. Indeed W. Buxton used to say: "everything that is best for something is worse for something else". And Jain, recently wrote: "While numerous clustering algorithms have been published and new ones continue to appear, there is no single clustering algorithm that has been shown to dominate other algorithms across all application domains [...] with the emergence of new applications, it has become increasingly clear that the task of seeking the best clustering principle might indeed be futile". However, that said, it is important to better understand the variations of the quality of algorithms measured by different criteria over varying datasets and how the three interact.

What are the dependencies between Cut, Cov, Cond and Mod ? Somehow all criteria used in the experiment try to capture how similar the elements inside clusters are and how dissimilar the clusters are one from another. Ratio of between-edges over within-edges are used in the definition of four out of the six criteria used in the experiment. However these ratio are not exactly computed the same way and small differences in their definition sometimes has huge impact on the results.

Cut and 1/cov are strictly covariant. In fact, Cov can be considered as the normalized version of cut. This is confirmed by the empirical evidence as Spearman correlation between 1/cut and cov is 1, and parallel coordinates

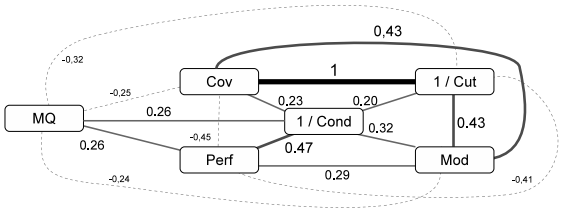

Fig. 1. Spearman Rank Correlation on Quality Measures

Both empirical and analytical results further suggests that mod can be considered as a measure of cov corrected by random. Therefore cov and mod are partially correlated. This is corroborated by both Spearman and parallel coordinate diagrams that clearly show that the two criteria are related but not strictly correlated. More work is needed to characterize the relationship more accurately.

Cut and cond use ratio of between-edges over within-edges. However cut has a global computation of the ratio, whereas cond not only computes the ratio at the cluster level but also consider only the minimum number of within-edges in each cluster. This observation is coherent with empirical evidence. Spearman average correlation between cond and cut is 0.20. Most parallel coordinate diagrams show that there are not too many crossings between cond and cut confirming a partial relation between the criteria. This implies that cond can discriminate among algorithms that provide clusters with highly variable numbers of within-edges than cut is expected to do. Note that cond and cut are the only two criteria that have to be minimized and not maximized. This is the reason why in the Spearman correlation we have computed 1/cut and 1/cond instead to facilitate their comparison with others. It is also the reason why we have reversed their axis in the parallel coordinate diagrams to make them visually coherent with the other where best values are on top, worst values at the bottom.

What makes perf and MQ different from Cov, Cut, Cond, Mod ? The particularity of MQ, is that it explicitly accounts for the number of clusters. The number of clusters clearly impacts the number of possible between-edges and therefore the overall values of other criteria. When comparing clustering

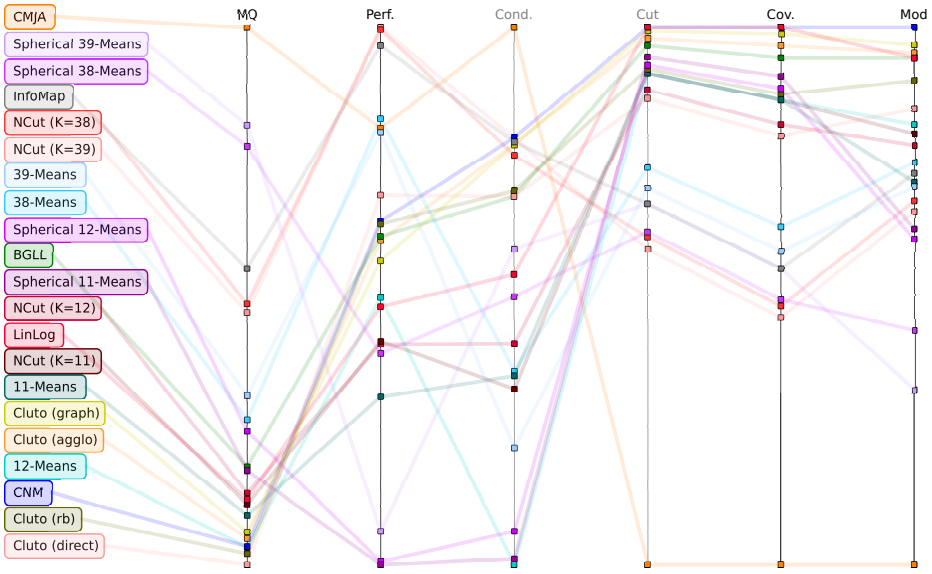

Fig. 2. Usage of parallel coordinates on quality Criteria for JDM 200

with different numbers of clusters, MQ is very useful. Other criteria can exhibit severe bias. For example, in the extreme case where a clustering results in a single cluster, and is compared to a much better clustering that provides 10 clusters, no between-edge will be found in the first clustering and most criteria will compute a high quality measure despite the fact that the clustering is really doing a poor job compared to the second. The fact that MQ accounts for the number of clusters prevents it from that bias. Also, experiments showed no correlation at all with criteria such as cov or mod and these results suggest that using MQ in conjunction with cov can be useful to balance the number of cluster bias.

Perf is probably the most debatable criteria amongst those reported in this paper. It surely tries to capture a different aspect of the quality of the results than the others. Somehow Perf captures the number of errors compared to an ideal clustering that would ideally lead to a disconnected set of cliques. However, the fact that the computation of Perf computes a ratio of the number of errors (between edges and missing within edges) over the total number of possible edges can lead to very misleading interpretations in many real situations. For example, previous experiments showed that random clustering can get better Perf ratings than other clustering.

7 Conclusion

In order to compare clustering results, most existing numerical criteria found in the literature focus on evaluating the quality of the compromise between intra cluster cohesion and inter cluster differentiation. In this paper we report the

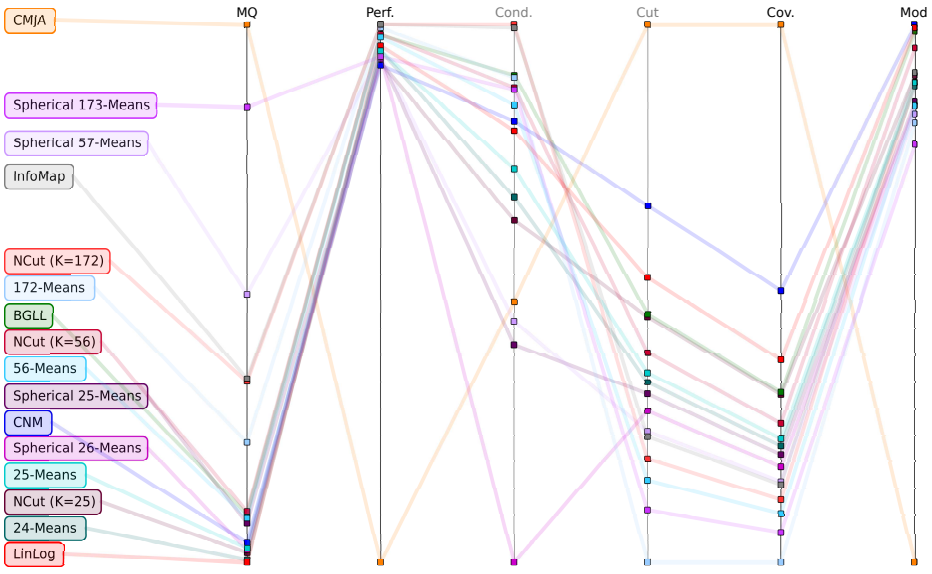

Fig. 3. Usage of parallel coordinates on quality Criteria for JDM 2000

results of several experiments with clustering algorithms over different datasets of keywords or documents. We have combined different criteria and analyzed the results using different approaches. The lessons learned are : (1) there is a lot of variation in the quality of the same clustering technique depending on the criteria / the datasets / the parameters used in the algorithm and (2) out of six different quality criteria found in the literature, cov and MQ used in conjunction can probably capture most of what the others can capture and (3) a lot of different aspects of the quality of the results cannot be captured at all with existing criteria. This experiment suggests that a lot of work is needed to better understand the quality and characteristics of automatic clustering results for keywords or documents datasets.

References

1. Abello, J., et al.: Ask-graphview: A large scale graph visualization system. TVCG (2006)
2. Auber, D., et al.: Multiscale visualization of small world networks (2003)
3. Blondel, V., et al.: Fast unfolding of communities in large networks. JSM (2008)
4. Boutin, F.: Filtrage, partitionnement et visualisation multi-échelles de graphes d'interactions à partir d'un focus. Phd. Thesis (2005)
5. Clauset, A., et al.: Finding community structure in very large networks. Phys. Rev. (2004)
6. Dhillon, I., et al.: Concept decompositions for large sparse text data using clustering. Machine learning 42(1), 143–175 (2001)
7. van Dongen, S.: Graph clustering by flow simulation (2000)
8. Huffman, D.: A method for the construction of minimum-redundancy codes. IRE (1952)
9. Inselberg, A., Dimsdale, B.: Parallel coordinates: a tool for visualizing multi-dimensional geometry. In: Proceedings of Visualization'90. pp. 361–378. VIS '90, IEEE (1990)
10. Jain, A.: Data clustering: 50 years beyond K-means. PRL (2010)
11. Lafourcade, M.: Making people play for lexical acquisition with the jeuxdemots prototype (2007)
12. MacQueen, J., et al.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. vol. 1, p. 14. California, USA (1967)
13. Noack, A.: An energy model for visual graph clustering. In: GD (2004)
14. Rosvall, M., et al.: Maps of random walks on complex networks reveal community structure. PNAS (2008)
15. Salton, G., et al.: Term-weighting approaches in automatic text retrieval* 1. IPM (1988)
16. Schaeffer, S.: Graph clustering. CSR (2007)
17. Seberry, J., et al.: Hadamard matrices, sequences, and block designs. CDT (1992)
18. Shi, J., et al.: Normalized cuts and image segmentation. TPAMI (2000)
19. Wang, F., et al.: Regularized clustering for documents. In: SIGIR (2007)
20. Xu, R., et al.: Survey of clustering algorithms. TNN (2005)
21. Zhao, Y., et al.: Criterion functions for document clustering: Experiments and analysis (2002)