Evaluation of Clustering Algorithms: a methodology and a case study

Mountaz Hascoët, Guillaume Artignan

Evaluation of Clustering Algorithms: a methodology and a case study

Abstract— Clustering is often cited as one of the most efficient ways to face the challenging scaling problem. Thousands of different approaches for clustering have been proposed over the past decades. Hence, the problem of designing appropriate clustering algorithm has been slowly replaced by the problem of choosing one implementation of one given algorithm amongst a large number of choices. However, because of the complexity of the field, choosing the appropriate implementation can rapidly turn into a dilemma. This paper introduces a methodology for the evaluation of clustering algorithms based on (1) theoretical complementary quality measures proposed in a unified notation system, (2) empirical studies on original datasets and (3) new technological instruments useful to both run experiments and visually analyze the results. Such a methodology is important not only to facilitate the choice of a clustering algorithm but also to consolidate the validity of the results by enabling reproducibility and comparison of experiments. By proposing a methodology with a case study, our aim is to bring to the scene new insights on the evaluation and comparison of clustering approaches that hopefully help clarify the field.

Index Terms—Clustering evaluation, Evaluation Methodology, Parallel Coordinate Diagrams.

1 Introduction

It is often considered that using clustering is one way of managing and controlling large and complex networks at a higher level of abstraction. Therefore clustering is often used in information visualization as a pre-processing or interactively. However, anyone eager to perform clustering has to make a potentially critical choice amongst thousands of algorithms. This choice can rapidly become a real dilemma. First, clustering literature is very dense, diverse, and sometimes complex. Reviews or meta-analysis are numerous but only partial. The evaluation of the quality of clustering algorithms is still difficult. Jain in his recent review on clustering [17], agreed that, if one consider all potential criteria for quality, "there is no best clustering algorithm". Kleinberg [21] has shown that it is impossible for a unique clustering algorithms to satisfy the following set of basic properties : (1) scale-invariance, (2) a richness requirement and (3) a consistency condition.

Our objective is to facilitate the understanding and choice of appropriate clustering algorithm that might be used prior visualization or while interacting. Our contribution is threefold: (1) a methodology based on the combination of formal, empirical, and technological backgrounds, (2) a case study using this methodology to evaluate a selected set of 17 clustering algorithms published in the literature and (3) a system designed and developped to support this methodology and favor repetition and reproduction of experiments.

We first introduce the methodology. Second, we present the case study by starting with transformations on datasets, followed by the selection and presentation of clustering algorithms. We further present a unified notation system to integrate a set of theoretical quality measures found in a broad and heterogeneous literature. Then, we present the visual exploration of the results of the case study. We further present MUSCA the system designed and developed to support the methodology and conduct the case study. We finally conclude with lessons learned and future work.

2 Methodology

The choice of the appropriate algorithm is often a matter of trade-offs for which the analysis of the quality of a clustering over varying datasets and tasks is useful.
If we consider T: a set of tasks, D : a set of datasets, Q: a set of quality measures and F : a set of clustering functions, then an evaluation problem can be considered as a point P in the evaluation space E such that:

$$E = T \otimes D \otimes Q \otimes F \text{ and}$$

$$P(t,d,q,f) \in E \text{ and } t \in T,\ d \in D,\ q \in Q,\ f \in F.$$

It is a truism to say that E is large and that many methodologies have been used to explore it. Benchmarking based methodologies can be considered as a mature way to address empirically a subset of E by limiting variations over T, D and possibly Q to better study F. A radically different methodological approach, probably even more mature and more theoretically oriented, consists in focusing on F possibly ignoring or making strong hypothesis about the nature of T, D, and Q interactions with F. Many approaches in this direction focus on comparing objective functions maximized in a clustering method when they exist or focus on comparing other important aspects of a clustering method characterizing F. Leading to axiomatic approaches that can be seen as a generalization in this direction [41][21][2].

Our methodology can be seen as complementary to previous approaches. It differs from them by making the hypothesis that interactions between T, D, Q and F are important and possibly chaotic.

3 Datasets and Tasks

The task studied in this paper is a basic task of exploring large datasets based on multi-level visual exploration techniques. These techniques make the hypothesis that the dataset is clustered automatically and that the resulting clusters provide different levels of abstraction such that at each level a cluster can be considered as an abstraction of a set of similar elements and different clusters discriminates the elements they contain.

The fixed chosen datasets are based on representative user data sets, e.g. ad-hoc datasets of various natures. The first dataset used in the case study is extracted from "Jeux de Mots", one of the largest lexical network of the French language [14]."Jeux de Mots" is built cooperatively by users playing a coordination game. For example, a player is asked to provide as many terms as possible given a specification and a target term. For example, "find related terms to

"bateau" (e.g boat). The answers of two players are then compared. The two players earn points based on how many common terms they spontaneously proposed. These terms and the relations between the terms are then appended to the lexical network following a cumulative weighting system [22]. "Jeux de Mots" now contains more than 200,000 lexical terms and 1,200,000 lexical relations including more than 20 different types of lexical relations.

The second dataset comes from a collection of research papers gathering ten years of the SIGIR conference papers. The similarities between each pair of documents is computed using the TF-IDF measure [32] and a Pearson's correlation. A complete large graph network is then obtained, where nodes are documents and similarities are weighted links.

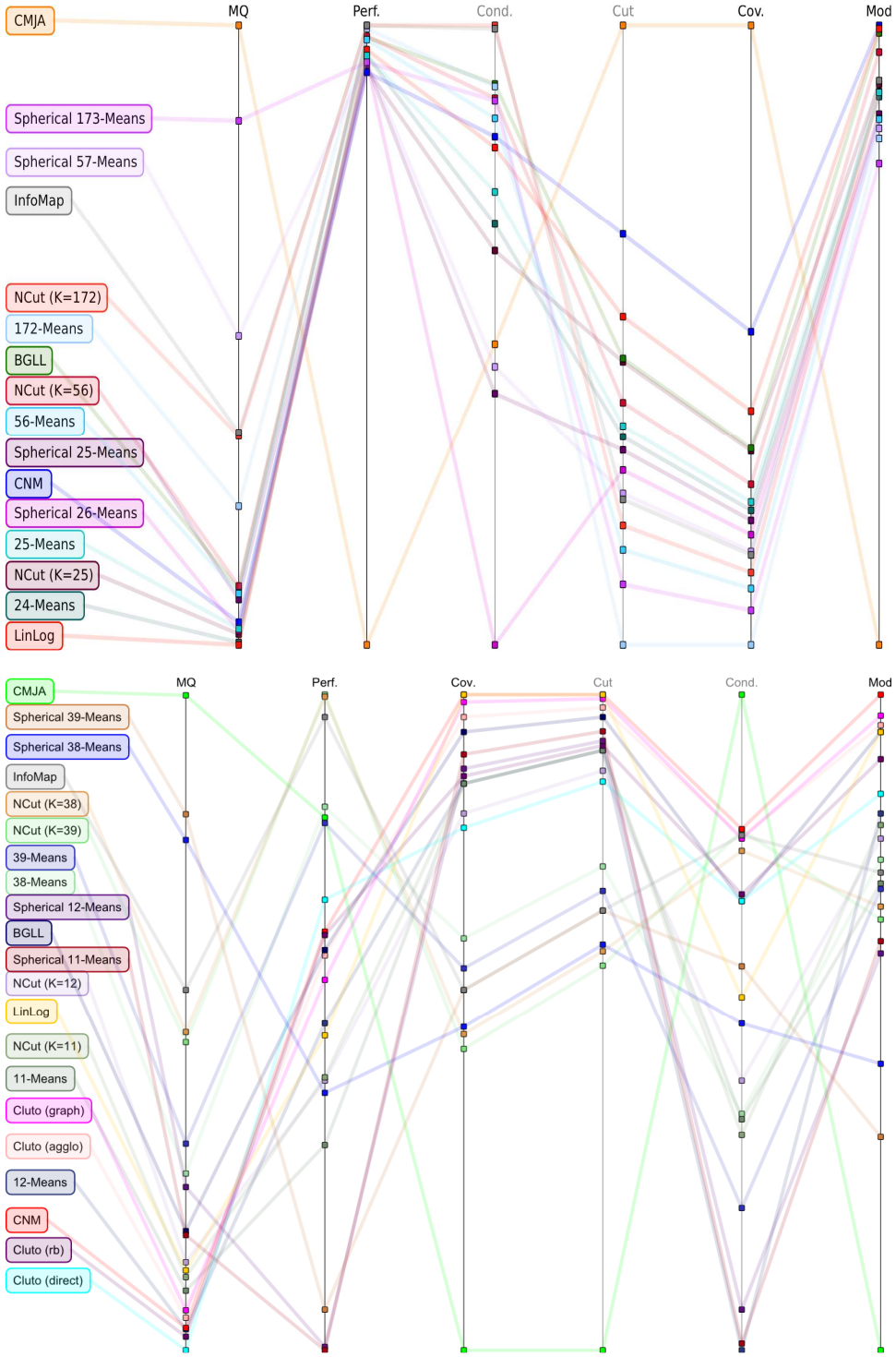

Figure 1: Parallel Coordinate Diagrams for Evaluating the Quality of Algorithms on JDM2000 (top), JDM200 (bottom)

The first step in our methodology consists in examining raw datasets at different scales without any clustering. Four datasets are built from the dataset "Jeux de Mots": (1) JdmAll containing 111701 lexical terms from "Jeux de Mots", (2) Jdm2000 containing 2000 lexical terms, (3) Jdm200 containing 200 terms from Jdm2000, and finally (4) Jdm20 containing 20 terms from Jdm 200.

In the case of the SIGIR conference papers, given the limited amount of nodes and potentially large number of edges, scaling dataset is based on controlling the number of edges. Therefore we consider Sig1000 and Sig10000 by considering respectively the subset of 1000 and 10000 edges from the computation of best similarity relations.

Table 1 summarizes the main characteristics of the datasets by providing name, number of nodes N, total number of edges E, the exponent γ of power law distribution when relevant, the graph diameter D, averaged clustering coefficient C, an URL describing the different datasets and proposing a link for downloading the datasets. As can be read from this table, the datasets are heterogeneous, jdm20 is a tree, Jdm200, Jdm2000, JdmAll and Sig1000 are power law networks and sig10000 is a dense network. With the exception of jdm20, such datasets characteristics are known to be challenging for clustering and at the same time, illustrative of real user data likely to be clustered.

Table 1. Datasets Characteristic Distributions

Name	*|N|*	*|E|*	*γ*	*D*	*C*	*URL*
JDM 20	20	19	∅	6	0.0	Anonymized for Review
JDM 200	200	265	-1,58	11	0.1140	Anonymized for Review
JDM 2000	2000	3476	-1.8	13	0.1357	Anonymized for Review
JDM ALL	111701	441854	-1.9	13	0.1933	Anonymized for Review
SIG 1000	378	903	-1.48	20	0.3928	Anonymized for Review
SIG 10000	626	10000	∅	5	0.4002	Anonymized for Review

4 QUALITY CRITERIA

Measuring the quality of the results of a clustering algorithm is challenging. Given the task and datasets, we focus on criteria defined in the literature to represent how similar the elements inside clusters are and how dissimilar the clusters are one from another. These criteria may vary widely in terms of notations and subtlety in terms of concept.

To make further analysis and discussion over many criteria, our first work consisted in setting up a notation capable of embracing several of the various criteria found in the litterature in various forms considered as target criteria. We have then rewritten these target criteria in this unifying notation. Our notation is designed to make the expression of criteria as simple as possible while maintaining a potential for expressiveness. As noted by Green designing a notation [13] is, in the general case, both important and challenging. This work is a preliminary effort in the long effort needed to come up with a notation oriented at the evaluation of clustering. Our notation is based on graph theory basic concepts since many datasets can be abstracted as graphs.

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 {C1...Ck} as k subsets of N.

Our notation is summarized by:

n	number of nodes in G
e	number of edges in G
k	number of clusters after clustering
n_i	number of nodes in the cluster Ci
we_i	number of edges within the cluster Ci
oe_i	number of edges outgoing from the cluster Ci
be_{ij}	number of edges between two clusters Ci and Cj
pe_i	number of possible edges in Ci. For an undirected graph: pei= ni(ni-1)/2. For a directed graph:pei=ni(ni-1).
me_i	number of *missing* (e.g. non-represented) edges in Ci. mei=pei - wei.
we, *be*, *pe* and *me*	total number of within-cluster edges, between-cluster edges, possible edges and missing edges.

Ratio of between-edges over within-edges is used in the

$$be = \sum_{i=1}^{k} be_i \, me = \sum_{i=1}^{k} me_i \, we = \sum_{i=1}^{k} we_i$$

definition of four (e.g. Cut, Cov, Cond and Mod) out of the six criteria presented below. However these ratio are not exactly computed the same way and small differences in their definition may have important impact on the results. With these notations, the six criteria selected for the evaluation can be written as follows:

Cut [6] is computed as the number of between-edges (also called extra-edges) over the number of within-edges (also called intra-edges). One frequent expectation is that this criteria be minimize in best clustering results.

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

Perf [7] takes into account the number of undesirable edges that can be considered as errors compared to an ideal clustering. These undesirable links are considered as edges between clusters, as well as missing edges within clusters. The number of missing edges is equivalent to the number of couples of nodes grouped in the same cluster without any edge relating them. Perf finally measures the ratio of undesirable edges over the number of possible edges and compares it to 1. Best values for Perf correspond to highest values.

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

Cond [7] criterion equals an average over the conductance of each pair of clusters. The conductance of a pair of clusters 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 are expected to characterise best clustering results. However, some particular cases of clustering results can't be measured with this definition of Cond. For example, when singleton clusters are part of a clustering result, the number of within-edge for singleton is arbitrarily replaced by a value of one. Other cases where stable can be found as clusters cannot be measured with a computation of Cond. However, these cases are not expected to be frequent considering the aims of algorithms studied in this work.

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

Cov [7] is the ratio of the number of within-edges to the total number of edges in the graph. Cov can be considered as the inverse of a normalized version of cut.

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

Mod [8] can be considered as a measure of Cov defined above corrected by the value of a Cov for a random clustering of the same graph denoted as rCov. Therefore, the highest values for Mod correspond to best clustering results according to Cov and values below 0 correspond to clustering that can be considered worse than a random clustering according to the Cov criteria.

$$Mod(G) = Cov - rCov$$

MQ [25] 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 the case of a singleton cluster, w*e*i and p*e*i equal 0. In this case, we do not compute w*e*i / p*e*i but use the value of 1 instead.

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

5 Clustering Algorithms Selection

Clustering has a huge and multidisciplinary history since it has been used in many scientific fields including information retrieval [38][36], data visualization [1], physics [8], etc. Several surveys have partially reviewed this literature [39][17][33]. In order to choose the algorithms to be tested in our study we had three criteria in mind. First, authors either provide source codes for the proposed algorithm 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 complex networks. Third, the set of algorithms tested should be representative of the variety of approaches of clustering found in the literature. Table 2 summarizes the choices made in terms of algorithms and indicates the URL of the implementation used in the experiment.

The CNM algorithm [8] has a bottom-up approach. Communities are made for each node and further merged iteratively with others to increase the Mod criteria. CNM results can be represented by a hierarchical clustered graph or a simple clustered graph depending on how merging is handled.

The BGLL algorithm [5] 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 [4] has a different approach from the two previous ones. 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 [31] 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 [15], (3) searches a clustering minimizing the average number of bits useful to describe it.

The MCL Algorithm [37] 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 [34].

K-Means Algorithm [24] is one of the most frequently used algorithms for clustering and many slightly different versions have been proposed. The main principle is to start with an arbitrary partition of the dataset and try to move each element to a better cluster as long as possible to improve the overall within cluster cohesion. It is one very efficient and very simple algorithm to implement. However, it’s based on centroid computation. Therefore it requires that as a prerequisite over other algorithms that meaningful centroids can be computed for the datasets.

LinLog Algorithm [27] 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 [35] comes from the image segmentation domain but can be adapted to graphs. Its principle is to optimize a criterion named “Normalized Cut”, using a spectral technique.

The Cluto Toolkit [42] 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.

Table 2. Algorithms and implementations used in the case studies

Algorithm Name	***Article***	***Implementation***
CNM	[8]	http://www.cs.unm.edu/~aaron/research/fastmodularity.htm
SPK-MEANS	[9]	http://www.cs.utexas.edu/users/dml/datamining/spkmeans.html
Cluto	[42]	http://glaros.dtc.umn.edu/gkhome/views/cluto
LinLog	[27]	http://www.informatik.tu-cottbus.de/~an/GD/
InfoMap	[31]	http://www.tp.umu.se/~rosvall/code.html
CMJA	[4]	our implementation (link removed for blind reviews)
BGLL	[5]	http://sites.google.com/site/findcommunities/
Simple K-Means	[24]	our implementation (link removed for blind reviews)
NCut Algorithm	[35]	http://www.cis.upenn.edu/~jshi/software/
MCL	[37]	http://www.arbylon.net/projects/
GraClus	[10]	http://www.cs.utexas.edu/users/dml/Software/graclus.html
WalkTrap	[28]	http://igraph.sourceforge.net/download.html
GN	[12]	http://igraph.sourceforge.net/download.html
MeTis	[18][19]	http://glaros.dtc.umn.edu/gkhome/views/metis
LPA	[29]	http://igraph.sourceforge.net/download.html
LEA	[26]	http://igraph.sourceforge.net/download.html
SpinGlass	[30]	http://igraph.sourceforge.net/download.html

The Spherical K-Means algorithm [9] is an extension of the well-known Euclidian K-Means algorithm. This algorithm partitions the dimension using great hyper-circles.

The GraClus [10] clustering algorithm is a multilevel algorithm. This algorithm operates in three steps: (1) the coarsening phase, (2) the initial clustering phase and (3) the refinement phase. The coarsening phase takes the initial graph and reduces it into a smaller graph. When the graph is sufficiently coarsened a spectral approach is used for clustering [40]. The refinement phase rebuilds the initial graph. The WalkTrap algorithm [28] computes a distance measure between each pair of adjacent nodes. At each step, the algorithm: (1) chooses two adjacent communities according to the similarity measures (2) merges these two communities and (3) updates the distances between communities. The algorithm terminates when only one cluster remains.

The GN [12] algorithm is a generic algorithm computing communities in two steps: (1) the computation of a score for each edge, (2) the removal of the edge with the best score. These two steps are repeated until a number of X edges is removed. In [12] the authors propose three measures: the shortest path measure, the network resistor measure and the random walk measure. The used implementation processes the shortest path measure. This measure is inspired from the vertex betweeness measure [11] and is adapted for edges. In the used implementations all the edges are removed in order to build a dendrogram of communities to merge. In adaptation version of the algorithm we made, we used a parameter indicating the number of wished clusters instead.

The MeTis Clustering algorithm [18][19] is also a multilevel algorithm and operates also in three steps. In the step of coarsening, MeTis uses a method named HEM (Heavy Edge Matching). Four algorithms are presented for the partitioning of the coarsened graph: a spectral bisection algorithm, a KL algorithm [20] a graph growing partitioning (GGP) or a greedy graph growing partitioning (GGGP). The refinement is then done using a edge-cut measure.

The Label Propagation Algorithm (LPA) [29], was introduced for discovering communities in web pages. Web pages are represented by nodes, hyperlinks are represented by edges. In the extraction of the initial graph, the authors construct a graph from an initial set of documents. The algorithm sets a weight on each node computed from both a non-negative authority-weight and a non-negative hub-weight. For a node the authority-weight is updated by summing all hub-weight of the neighbors referring the node. Similarly the hub-weight of a node is updated by summing the authority weight of all referenced nodes. The Leading Eigenvector Algorithm (LEA) [26] computes a graph clustering using modularity measure. This modularity measure is expressed in term of eigen values and eigen vectors of matrix call modularity matrix.

The SpinGlass algorithm [30] is an algorithm based on a SpinGlass model and simulated annealing. The authors demonstrate also the equivalence between their Hamiltonian measure and the modularity measure introduced by Newman and Girvan [12].

6 VISUALLY EXPLORING RESULTS

Parallel coordinate diagrams also called Inselberg's diagrams [16] are automatically created to display results. A set of parallel coordinate diagrams makes possible the exploration of variations along three of the four dimensions of the evaluation space considering one dimension is kept invariant. With such an approach, each diagram corresponds to variations along one dimension, each axe on a diagram corresponds to variations along the second dimension and each line corresponds to variations along the third dimension. For example, Figure 1 corresponds to different datasets (e.g. variations along D), where axes represent different criteria (e.g. variations along Q) and lines represent different clustering algorithms (e.g variations along F).

A Pareto optimal solution is a solution where any improvement in one criterion can only occur through the worsening of at least one other criterion. A Pareto set is composed of all Pareto optimal solutions and is usually considered as important in multi-criteria decision. Using parallel coordinate makes Pareto sets easily visible. For example, a polyline that always appear below another polyline can be considered non Pareto optimal. Reciprocally, a polyline A with no polyline always above it in the diagram can be considered as Pareto optimal.

From these diagrams, some specificities that sometimes introduce noise are also visually salient. For example, it is visually striking that CMJA algorithm exhibits extreme variations along Q. These extreme variations over Q are relatively invariant over D. Indeed, in most datasets, CMJA is best according to MQ, while at the same time being worse according to Mod and varying widely according to other criteria see Figure 1 for example. A closer examination at CMJA clustering results shows that it leads to clusters with disproportionate sizes, ranging from singletons to very large clusters which can explain this variability over Q.

A further visual analysis, consists in studying variations of F and Q while keeping D invariant. To provide an overview of groups of algorithms exhibiting satisfactory results over Q, we start by removing CMJA from the analysis, because its specificities not only do not fit the task, they also interfere with overall min/max values. We further focus on remaining algorithms on SIG10000 see Figure 2 (top). The intervals of values for each axe are automatically updated to reflect new min/max values after CMJA min/max values have been removed and axes are automatically scaled accordingly. Differences between remaining algorithms become easier to identify and algorithms with similar behaviors along Q can be visually grouped in six groups (cf Figure 2 - middle): (1) infomap and ncut (k=37), (2) Spherical 37 means and 37 means, (3) BGLL and Linlog, (4) 6 Means, NCut (k=12, 11), Spherical 6-means, (5) 11-12 Means, Spherical 11-12 Means and (6) CNM and nCut (k=6).

Table 3. Average Rankings

Algorithm	MQ	PERF	COV	CUT	COND	MOD	Average
CNM	4.50	4.33	1.66	4.50	3.16	2.33	**3.41**
BGLL	4.16	3.16	3.50	3.00	3.33	2.00	**3.19**
CMJA	1.00	4.50	2.83	3.50	3.16	6.00	**3.50**
InfoMap	2.16	1.33	5.00	1.83	4.33	4.00	**3.11**
LinLog	4.33	3.66	2.50	3.83	3.33	1.50	**3.19**
K-Means	3.83	2.83	3.66	2.50	2.16	4.16	**3.19**

The two groups (2) and (5) are not Pareto optimal since they are dominated respectively by group (1) and (3). However, the diagram also shows how similar the results of group (5) and group (3) correlate. Considering the important differences between their algorithmic approaches this result cannot possibly be found with methodologies only based on the comparison of F characteristics. Removing non Pareto optimal groups of algorithms results in Figure 2 bottom diagram that shows the Pareto set for the dataset sig10000 grouped in 4 categories of algorithms.

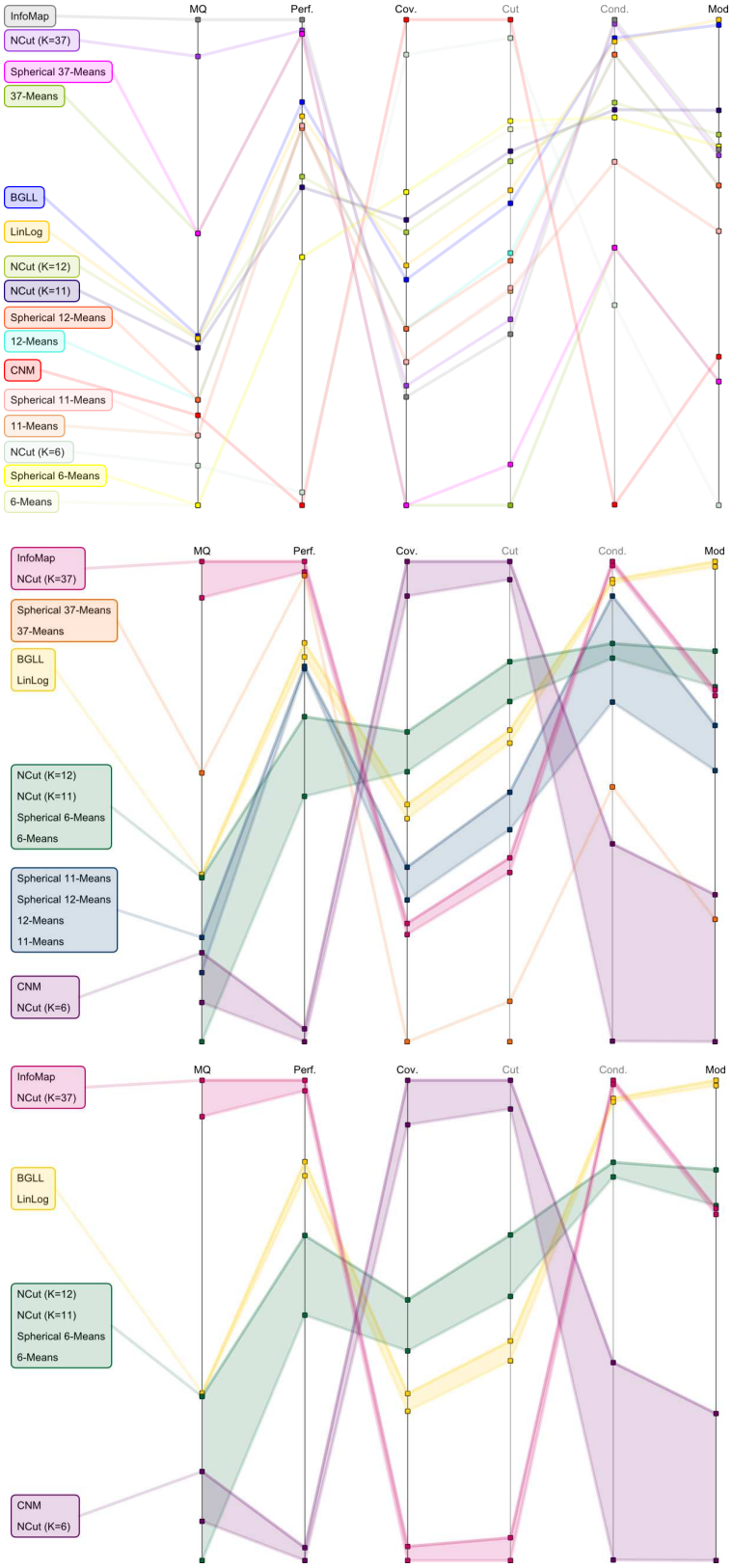

Figure 2:Parallel Coordinate Diagrams for SIG10000 without CMJA: detailed view of results (top), clustered view of results (middle), clustered view of Pareto set (bottom).

The first group is optimal for MQ, Perf and Cond. Second group optimal for Mod. The third group is not optimal for any criteria but is not dominated on all criteria by any other group. Interestingly, this is the only group where the grouping of the results is consistent with the similarity between the algorithmic approaches. Lastly, the fourth group is optimal in terms of Cov/Cut.

A second analysis aims at comparing quality criteria over datasets. Average ranking according to each criteria of algorithms tested on all datasets is summarized in Table3. Cut and 1/Cov are strictly covariant, because Cov can be considered as the normalized version of 1/Cut.

Analytical definitions of Cov and Mod further suggest that these two criteria are partially correlated. As mentioned earlier, Mod can be considered as a measure of Cov corrected by random. This can be confirmed by the empirical results. Spearman's correlation is computed for each pair of criteria over all datasets and reported as a graph where nodes are criteria and weighted edges correspond to the average Spearman's correlation for all datasets (see Figure 4). The partial correlation between Mod and Cov is also visible in Figure 6 where the two Mod and Cov axes are juxtaposed.

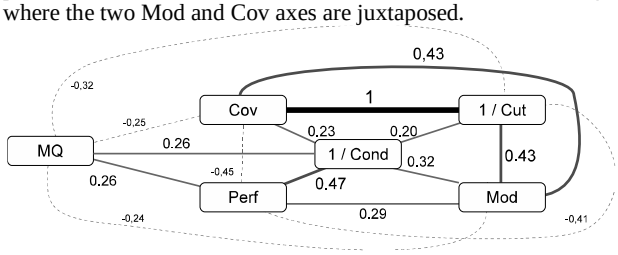

Figure 4. Spearman Rank Correlation on Quality Measures

Cut and Cond use ratio of between-edges over within-edges. Cut has a global computation of the ratio, whereas Cond not only computes the ratio at the cluster level but also considers only the minimum number of within-edges in each cluster. This difference between the two criteria has an important impact on the final results. Spearman's average correlation between Cond and Cut is 0.20. Most parallel coordinate diagrams show that there are crossings between Cond and Cut but not too many, confirming a partial relation between the criteria. Note that Cond and Cut are the only two criteria that have to be minimized and not maximized. Therefore, Spearman's correlations have been computed with 1/Cut and 1/Cond instead of Cut and Cond. It is also the reason why we have reversed their axis in the parallel coordinate diagrams so that for all criteria best values are on top, worst values at the bottom.

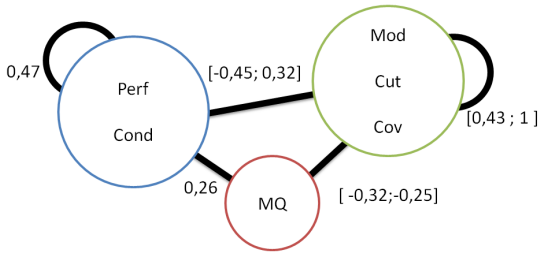

Figure 5: Groups of criteria

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 with very 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 first clustering results can be considered poor 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 captures different aspects of the quality of clustering. Using MQ in conjunction with Cov can be useful to balance other biases such as, for example, the bias coming from varying numbers of cluster.

Perf is probably the most debatable criteria amongst those reported in this paper. Perf captures the number of errors compared to a 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 algorithms.

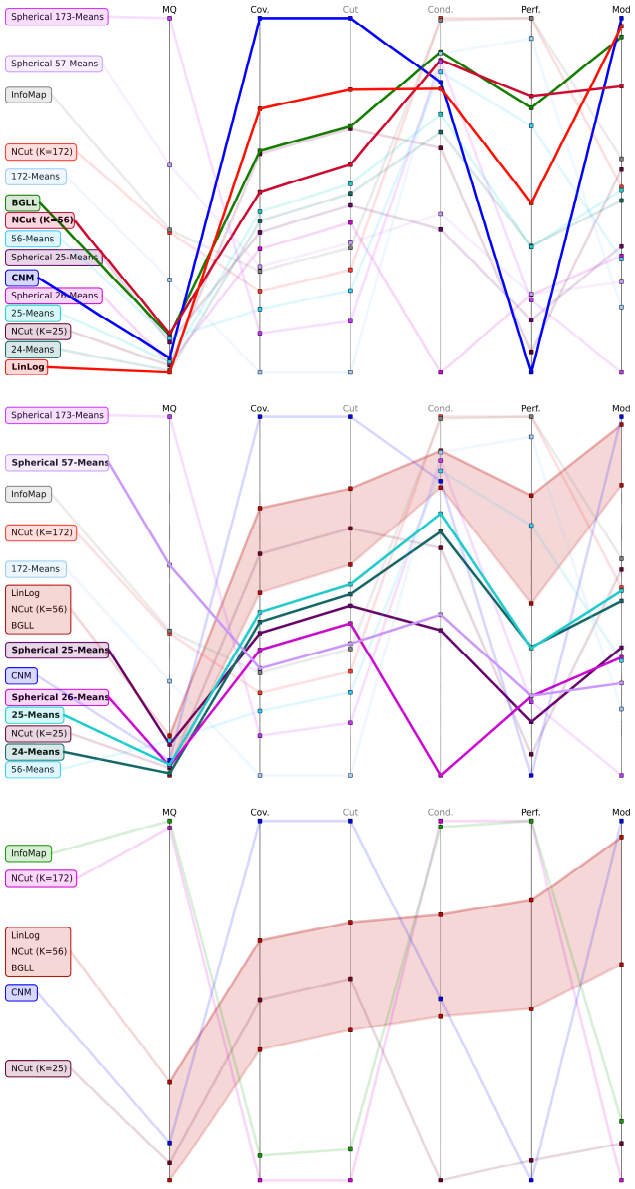

Figure 3: Parallel Coordinate Diagrams for Evaluating the Quality of Algorithms on JDM2000

From the analysis of both analytical definitions of criteria and ranking of the selected sets of algorithms over the selected data, we

can extract groups of criteria. Figure 5 depicts three groups and edges between groups is labeled with min and max of average Spearman's correlation on all datasets and for each pair of criteria with one criteria in each group. For example, Mod, Cut and Cov are considered to be part of the same group with between -0,32 and -0,25 correlation with MQ.

A third visual analysis aims at comparing clustering results for JDM2000 with specific quality criteria in mind. We focus on algorithms with best values for Mod (Figure 3, top) and find four clustering algorithms : NCUT(K=56), CNM, BGLL and Linlog.

Now, we consider that worse values for Perf are problematic so we remove CNM from our selection and further consider the group of three remaining algorithms NCUT(K=56), BGLL and Linlog as our reference (Figure 3, middle). We further remove all algorithms that have results below the performance of the reference group and we keep the remaining algorithms of Figure 3, bottom.

Two sets of algorithms are available (see Figure 3, bottom): the reference group (NCUT(K=56), BGLL and Linlog) and an alternative group with behaviors potentially useful in case of a slight change in the quality criteria selection: Infomap, NCut (K=251 and K=172) and CNM. Indeed, if we consider the previous discussion and the groups of criteria extracted from the previous study, it is obvious that the reference algorithms of Figure 3 correspond to the family of criteria Mod-Cut-Cov depicted in Figure 5. However that reference group is really poorly ranked compared to the alternative if we consider the criteria of the two other groups, e.g. MQ, perf and cond. Considering that these criteria are also of importance would imply that Infomap and nCut (K= 172) are good alternative choices.

7 Evaluation Tool

The results previously exposed are obtained thanks to our system named *MUSCA* (**Mu**lti-**Sc**ale **A**pplication for Graph Visualization). Three different types of functionalities are available in *MUSCA:* (1) transformation and clustering of heterogeneous data, (2) computation of metrics and charts for each dataset, (3) visual exploration and interaction with experimental results.

Architecture. *MUSCA* is a distributed system mainly implemented in Java. The application makes possible the upload by different users of datasets located on the Web. Datasets are referenced in a MySQL database and shared among experiments and users. Datasets and results from clustering algorithms are encoded in GraphML format. This format was chosen because it can encode both the representation of directed graphs, undirected graphs, multivariate graph (used to store original datasets) and clustered graphs, hierarchical graphs, compound graphs (used to store results of clustering algorithms applied on datasets). Necessary transformations of datasets from GraphML into the input format of the studied clustering algorithms as well as the interpretation of the algorithms output formats are all processed by *MUSCA*. *MUSCA* is extensible to enable the integration of additional layouts, clustering algorithms, graphical elements, etc.

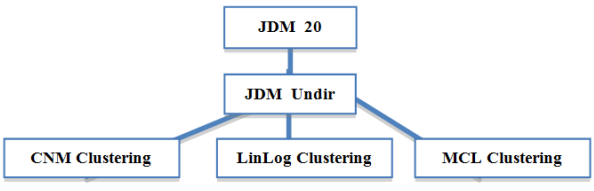

Figure 7: A Sample of Transformation Tree

Transformations. Datasets are organized following a tree of transformations, where each node is a dataset and each edge represents the transformation applied to the parent node to generate the child node. Figure 7 presents a sample sub-tree with a dataset named "JDM20" transformed to an undirected graph and further transformed into three clustered graphs using three different clustering algorithms.

Computation of metrics. For each dataset the computation of the metrics is done once for all. The system architecture implies that a dataset will be never modified. Indeed, all time the transformations build new datasets, keeping the previous one intact. In this case, the computation of the metrics of each dataset can be done once and can be stored in the database in order to make it available instantly for the other users.

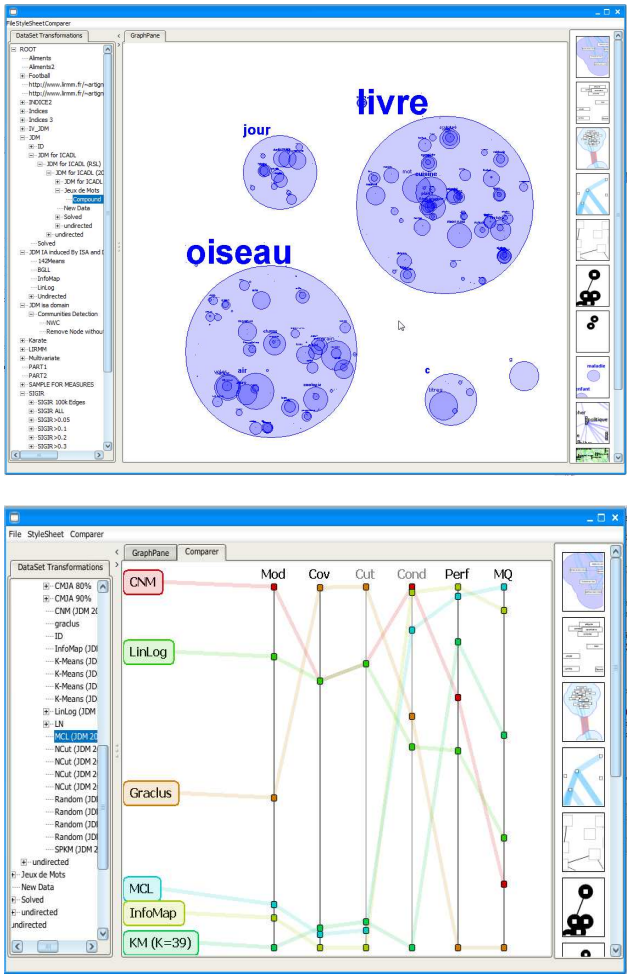

Figure 6: MUSCA environment

Visual Analysis. Visual analysis over both datasets and experimental results can be performed either by using Inselberg's diagrams or style-sheet based views. The resulting zoomable views are displayed in the central part of MUSCA. For example, a style-sheet based view of the results of a clustering algorithm on JDM2000 using a simple style-sheet is displayed in Figure 6 (top) while an Inselberg's view can alternatively be displayed to represent the same results using quality criteria and parallel coordinate instead. A set of style-sheets displayed in the right window of MUSCA (see Figure 6) can be applied to compatible datasets and make the graphical coding of the datasets completely configurable. Thanks to a style-sheet language described in [3], new style sheets can be created and added to MUSCA to provide new types of displays.

8 Lessons Learned and Perspectives

In order to compare clustering results, we have proposed a methodology. We make the hypothesis that the exploration space is potentially chaotic as soon as we consider real user needs and therefore consider interaction between the four dimensions T, D, Q, and F of the evaluation space.

Amongst the results that can be extracted from the case study, some are very specific to the parts of T, D, Q and F that have been

studied but other can probably be generalized. The first group of criteria mod-cut-cov (Figure 5) has analytical grounds so it can be expected to generalize to most cases even though the behaviour of mod and the rest cut-cov might be different on some datasets. Future work could be useful to see how the two other groups generalize to other datasets. Another issue is to investigate whether these groups of criteria correlate or not with some quality estimation made by experts. Such an approach can be performed either by reproducing similar experiments with benchmarks containing human cluster evaluation [23]. More generally, studying how results computed on benchmarks correlates with results computed on real ad-hoc datasets is also left to future work.

Another important result from our case study is in exhibiting groups of similar algorithms according to their behaviour with the studied datasets and criteria. How these groupings generalize to other datasets is left for future work. However, the fact that similar clustering approaches can exhibit important variations given our datasets and criteria, and that conversely, different clustering approaches can behave similarly according to the same datasets and criteria suggests that the methodology chosen in this paper is worthwhile.

Another preliminary finding worth considering for future work is that our results suggests that Cov and MQ used in conjunction could probably capture most of what the six selected criteria could capture altogether considering the dataset and criteria studied in this paper. Finding the minimal set of criteria given a set of criteria is probably an open issue in the general case. However, exhibiting redundancies amongst different criteria as well as important discriminating power of combinations of several criteria is probably both useful and generalizable to other datasets.

9 Acknowledgement

The authors wish to thank all the anonymous reviewers that have contributed to enhance the quality of this paper. The authors wish also to thank the anonymous reviewers that have contributed to delay its publication to the right moment.

10 References

[1] Abello, Van Ham, and Krishnan. 2006. ASK-GraphView: A Large Scale Graph Visualization System. IEEE Transactions on Visualization and Computer Graphics 12, 5 (September 2006), 669-676.

[2] Ackerman and Ben-David. Measures of Clustering Quality: A Working Set of Axioms for Clustering. *Neural Information Processing Systems Conference (NIPS 2008)*

[3] Anonymized for blind review.

[4] Auber, Chiricota, Jourdan, and Melançon. 2003. Multiscale visualization of small world networks. In Proceedings of the Ninth annual IEEE conference on Information visualization (INFOVIS'03), Tamara Munzner and Stephen North (Eds.). IEEE Computer Society, Washington, DC, USA, 75-81.

[5] Blondel, Guillaume, Lambiotte, and Lefebvre. Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment, 2008(10):P10008+, July 2008.

[6] Boutin, F.: Filtrage, partitionnement et visualisation multi-échelles de graphes d'interactions à partir d'un focus. Phd. Thesis (2005)

[7] Brandes, Gaertler, Wagner: Experiments on Graph Clustering Algorithms. ESA 2003: 568-579

[8] Clauset, M. E. J. Newman, and Cristopher Moore. Finding community structure in very large networks. August 2004.

[9] Dhillon and Modha. 2001. Concept decompositions for large sparse text data using clustering. Mach. Leam. 42, 1/2 (January 2001), 143-175.

[10] Dhillon, Guan, Kulis: A fast kernel-based multilevel algorithm for graph clustering. KDD 2005:629-634

[11] Freeman, A set of measures of centrality based on betweenness. Sociometry 40(1):35-41 (1977)

[12] Girvan and Newman, Community structure in social and biological networks, Proc. Natl. Acad. Sci. USA 99, 7821–7826 (2002).

[13] Green, T. R. G. (1989). Cognitive dimensions of notations. In *People and Computers V*, A Sutcliffe and L Macaulay (Ed.) Cambridge University Press: Cambridge., pp. 443-460.

[14] http://jeuxdemots.org.

[15] Huffman, D. A. "A Method for the Construction of Minimum-Redundancy Codes." Proc. Inst. Radio Eng. 40, 1098-1101, 1952.

[16] Inselberg, A., Dimsdale, B. 1990. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In Proceedings of the 1st conference on Visualization '90 (VIS '90), Arie Kaufman (Ed.). IEEE Computer Society Press, Los Alamitos, CA, USA, 361-378.

[17] Jain. 2010. Data clustering: 50 years beyond K-means. Pattem Recogn. Lett. 31, 8 (June 2010), 651-666.

[18] Karypis, Aggarwal, Kumar et Shekhar. Multilevel hypergraph partitioning : application in VLSI domain. In Proceedings of the annual Design Automation Conference, pages 526–529. ACM, 1997.

[19] Karypis, G. and Kumar, V. (1999). "A fast and high quality multilevel scheme for partitioning irregular graphs". SIAM Journal on Scientific Computing (ACM) 20 (1): 359.

[20] Kernighan and Lin, An Efficient Heuristic Procedure for Partitioning Graphs, Bell Sys. Tech. J., Vol. 49, 2, pp. 291-308, 1970.

[21] Kleinberg. An Impossibility Theorem for Clustering. Advances in Neural Information Processing Systems (NIPS) 15, 2002.

[22] Lafourcade, M.: Making People Play for Lexical Acquisition with the JeuxDeMots prototype. In: SNLP 2007: 7th International Symposium on Natural Language Processing, Pattaya, Thailand (2007).

[23] Lewis, Ackerman, and De Sa. Human Cluster Evaluation and Formal Quality Measures. Proc. 34th Annual Conference of the Cognitive Science Society, 2012.

[24] Macqueen. Some methods of classification and analysis of multivariate observations. In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, 1967.

[25] Mancoridis, Mitchell, Rorres, Chen, Gansner: Using Automatic Clustering to Produce High-Level System Organizations of Source Code. IWPC 1998.

[26] Newman. Finding community structure in networks using the eigenvectors of matrices. Physical Review E, vol. 74, no. 3, page 036104, 2006.

[27] Noack. A. 2004. An Energy Model for Visual Graph Clustering. In Proceedings of the 11th International Symposium on Graph Drawing (Perugia, Italy, Sep. 21--24), GD 2003, Springer-Verlag, Berlin, LNCS 2912, 425--436.

[28] Pons, Latapy: Computing Communities in Large Networks Using Random Walks. J. Graph Algorithms Appl. (JGAA) 10(2):191-218 (2006)

[29] Raghavan et al., Near Linear Time Algorithm to Detect Community Structures in Large-scale Networks, 2007

[30] Reichardt J, Bornholdt S (2006) Statistical mechanics of community detection. PhysRev E Stat Nonlin Soft Matter Phys 74:016110.

[31] Rosvall and Bergstrom. Maps of random walks on complex networks reveal community structure. Proceedings of the National Academy of Sciences, 105(4):1118-1123, January 2008.

[32] Salton and Buckley. 1987. Term Weighting Approaches in Automatic Text Retrieval. Technical Report. Cornell University, Ithaca, NY, USA.

[33] Schaeffer. Graph Clustering. Computer Science Review 1(1): 27-64, 2007.

[34] Seberry, Yamada, Hadamard matrices, sequences and block designs, Contemporary Design Theory:A Collection of Surveys , eds. J.Dinitz and D.Stinson, J.Wiley, New York, (1992), 431-560.

[35] Shi and Malik. 1997. Normalized Cuts and Image Segmentation. In Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97) (CVPR '97). IEEE Computer Society, Washington, DC, USA, 731-.

[36] Steinbach, Karypis, Kumaret al. A comparison of document clustering techniques. In KDD workshop on text mining, volume 400, pages 525–526. Citeseer, 2000.

[37] Stijn van Dongen, Graph Clustering by Flow Simulation. PhD thesis, University of Utrecht, May 2000.

[38] Wang, Zhang, and Li. 2007. Regularized clustering for documents. In Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '07). ACM, New York, NY, USA, 95-102.

[39] Xu and Wunsch, "Survey of clustering algorithms," IEEE Transactions on Neural Networks, vol. 16, no. 3, pp. 645-678, 2005.

[40] Yu and Shi: Multiclass Spectral Clustering. ICCV 2003:313-319

[41] Zadeh and Ben-David, A Uniqueness Theorem for Clustering, Uncertainty in Artificial Intelligence UAI, 2009.

[42] Zhao and Karypis. Criterion functions for document clustering: Experiments and analysis, 2001.