# A comparative study on community detection and clustering algorithms for text categorisation

Michelangelo Misuraca[1], Germana Scepi[2], Maria Spano[2]

[1] DiScAG, Università della Calabria – michelangelo.misuraca@unical.it

[2] DiSES, Università di Napoli Federico II – germana.scepi@unina.it, maria.spano@unina.it

## Abstract

One of the main tasks of Text Mining is organising a large number of unlabelled documents into a smaller set of meaningful and coherent clusters, similar with respect to their content. Clustering algorithms are usually carried on *documents* × *terms* matrices, algebraically representing each document as a vector. Nevertheless, a collection of documents can also be encoded differently, e.g. by considering a *documents* × *documents* representation. This peculiar data structure can be seen as an adjacency matrix and graphically displayed as a graph. In the framework of Network Analysis, community detection is performed on such graphs to find groups of nodes sharing common characteristics, and play similar roles. This paper aims at evaluating the use of different data structures and different grouping criteria, showing the effectiveness of the different alternatives in a text categorisation strategy. We performed a comparative study involving both classical text clustering approaches and community detection approaches, testing and discussing their performances.

**Keywords:** text clustering, community detection, data representation, weighting schemes, similarity measures

## 1. Introduction

One of the primary tasks of Text Mining is to organise a collection of documents written in natural language. Texts can express a broad and varied range of information, but this information is often difficult to process automatically due to its particular encoding. The family of techniques that allows grouping the different documents without any other knowledge than the textual content itself, having a data-driven standpoint, is known as *text clustering*. Clustering algorithms are usually carried on *documents* × *terms* matrices, algebraically representing the different documents belonging to the analysed collection as vectors. This representation – known as *vector space model* (Salton *et al.*, 1975) – is the most common way to transform documents in structured data, disregarding the grammatical and syntactical roles of the different terms within the texts. The strategies adopted to group the documents are almost similar to those used for classical data and usually based on the optimisation of a criterion function. The two main approaches include hierarchical algorithms and partitive algorithms. Nevertheless, data can be organised differently, e.g. by considering a *documents* × *documents* matrix. This data structure can be seen as an adjacency matrix and diagrammatically depicted as a graph. Each document is then seen as a node and linked to the others when there is a similar textual content, i.e. documents share common terms. Different similarity measures can be used to express the level of proximity between linked documents. In the framework of Network Analysis, the so-called *community detection* allows highlighting if there are groups of nodes sharing common characteristics, and/or playing similar roles within the graph. The rationale of these techniques is then the same as the one at the base of clustering analysis.

In this paper, we want to evaluate the use of different data representations and grouping strategies, aiming at showing the effectiveness of the alternative strategies in a text categorisation context. We performed both classical clustering approaches and community detection approaches, considering the effect of different weighting schemes to express the importance of the terms used by each document and of different similarity measures to quantify the proximities between the different documents of the collection.

## 2. Theoretical background

There are several ways to model documents for quantitative analyses, but most of the clustering algorithms are commonly carried on a collection encoded via a *bag of words* (BoW) scheme. In this scheme, a document is seen as an unordered set of terms, disregarding grammatical and even syntactic roles. Let consider a *corpus* of $n$ documents $\mathbf{d}_i$ ($i = 1, \ldots, n$). According to the vector space model, each $\mathbf{d}_i$ can be represented as a vector in the space spanned by the $p$ terms belonging to the vocabulary of the *corpus*:

$$\mathbf{d}_i = (t_{i1}, \ldots, t_{im}, \ldots, t_{ip}) \tag{1}$$

where $t_{im}$ is the importance of the *m-th* term in $\mathbf{d}_i$. This importance is usually measured by the *term frequency* – i.e. the number of occurrences of the terms into the document – but other different weighting schemes can also be considered (for a wider discussion on the choice of $t_{im}$ see Balbi and Misuraca, 2005).

By juxtaposing the different document-vectors, it is possible to build a *documents* × *terms* matrix **T**. It is possible to derive from this data structure both a *terms* × *terms* matrix and a *documents* × *documents* matrix. Generally, in a *terms* × *terms* matrix, each element is the number of times two terms co-occur in the document collection. In a *documents* × *documents* matrix, each element is the number of terms which occur together in two different documents. In both cases, it is possible to consider more complex weights to express the relationship between couples of terms (Cheng *et al.*, 2013) or couples of documents (Huang, 2008).

Supervised and unsupervised categorisation methods aim at grouping similar documents in distinct subsets, taking into account their content. In the first case, some prior knowledge is available, usually related to the topics embodied in the collection. The information concerning the number of groups, their peculiarities and their composition, is used in the categorisation process. An unsupervised approach, instead, aims at grouping a collection of unlabelled documents into a smaller number of meaningful categories similar in content, without any additional knowledge. In this latter case, the process is based solely on the available data. The strategies usually adopted for unsupervised text categorisation are similar to those used for classical data, both considering hierarchical and partitive algorithms. Hierarchical algorithms allow visualising the association structure among documents at different levels of granularity. One of the main consequences is that the number of clusters is not an input parameter of the algorithm, and the different solutions are sequentially nested and displayed in a tree structure. Although, hierarchical algorithms are not very scalable in the case of large collections of documents (Steinbach *et al.*, 2000). Partitive algorithms – also known as centre-based – create a one-level solution instead, given the number $k$ of desired clusters as input parameter. The documents initially selected as centroids of the $k$ clusters are usually chosen randomly, but other options can be considered (see Larsen and Aone, 1999). After that, the proximity (by a suitable measure) of each other document to the $k$ centroids is computed, and the assignment

to a cluster of a document is done looking at the highest proximity. The clustering process is repeatedly refined in order to optimise the chosen criterion function.

The *terms* × *terms* matrix and the *documents* × *documents* matrix can also be seen as a graph *G=(V,E)*, where *V* is a finite set of nodes (or vertices) and *E* is a finite set of edges (or lines). Edges indicate the relationships between the nodes. If the nodes are the terms, the edges express the co-occurrence between linked terms. If the nodes are the documents, the edges represent the strength of the similarity between linked documents. In a Network Analysis framework, the task of grouping nodes sharing common characteristics, and/or playing similar roles within the graph, is usually performed by referring to community detection methods. Communities are usually thought as subgraphs densely inter-connected and sparsely connected to other parts of the network (Wasserman and Faust, 1994). From a theoretical viewpoint, communities are then not very different from clusters (Fortunato and Hric, 2016). As a consequence, there is an overlap among the scientific contributions developed in these two research areas. Clustering algorithms have been successfully used for graph data (e.g. Flake *et al.*, 2003; Rattigan *et al.*, 2007). Many recent works (e.g. Lim *et al.*, 2017; Misuraca *et al.*, 2018; Jia *et al.*, 2018) proposed community detection methods on *terms* × *terms* networks for identifying combinations of terms – i.e. concepts or topics − occurring in the collection of documents. Nevertheless, the possibility of using a community detection approach on the *documents* × *documents* network for a text clustering has not been sufficiently explored. To the best of our knowledge, the papers of Mikhina and Trifalenkov (2018) and Cadot *et al.* (2018) are the only ones evaluating a clustering strategy based on the idea that the collection of documents can be represented as a weighted graph.

## 3. A comparative study

As claimed above, clustering and community detection share the same logic so that they can achieve equivalent results in a text categorisation framework. In the following, in order to test the effectiveness of the two approaches, we performed a comparative study.

### *3.1. Data description and pre-processing*

We downloaded from Kaggle a dataset[1] containing 2225 complete news articles published during 2004-2005 on the BBC website, and reporting stories categorised in five different topical areas (Greene and Cunningham, 2005). The type-token ratio of the collection is 0.031, substantiating the use of clustering techniques (Bolasco, 2013). Table 1 shows the distribution of documents over the five topics and some descriptive statistics.

| ***Topic*** | ***# of doc*** | ***% of doc*** | ***Avg terms per doc*** | ***Avg sentences per doc*** |
|---|---|---|---|---|
| *Business* | 510 | 22.92 | 374.25 | 15.67 |
| *Entertainment* | 386 | 17.35 | 375.88 | 16.30 |
| *Politics* | 417 | 18.74 | 511.80 | 20.88 |
| *Sports* | 511 | 22.97 | 374.30 | 16.90 |
| *Tech* | 401 | 18.02 | 563.37 | 24.04 |

*Table 1 – Characteristics of BBC news collection.*

[1] https://www.kaggle.com/pariza/bbc-news-summary

Before carrying on the analyses, we stripped numbers, dates, punctuation, and URLs from the original documents. We also normalised the documents by removing special characters and any separators than blanks. On the cleaned collection, we then performed lemmatisation and removed English stopwords. Moreover, we deleted terms occurring less than two times and kept only the terms which occurred at least in two documents.

### *3.2. Experimental setup*

At the end of the pre-processing process, we obtained a *documents* × *terms* matrix **T** with 2225 rows and 14511 columns. This matrix, whose generic element $t_{ij}$ is the frequency of the *j-th* term in the *i-th* document, was transformed according to two other weighting schemes, obtaining a matrix $\mathbf{T_b}$ (Boolean scheme) and a matrix $\mathbf{T_{tf\text{-}idf}}$ (tf-idf scheme: Salton and Buckley, 1988). In a Boolean scheme, only the presence/absence of each term in each document is considered. In a tf-idf scheme, the occurrences of each term in a document are multiplied by the reciprocal of the fraction of documents containing the term on the total number of documents in the collection, jointly using as term importance a local weight and a global weight.

| | |
|---|---|
| ***weighting schemes*** | *Boolean* |
| | *Term Frequency* |
| | *Term Frequency – Inverse Document Frequency* |
| ***clustering algorithms*** | *UPGMA (Sokal and Michener, 1958)* |
| | *K-means (McQueen, 1967)* |
| | *Spherical K-means (Dhillon and Modha, 2001)* |
| ***community detection*** | *Louvain algorithm (Blondel et al., 2008)* |
| | *proximity measures:*<br>*jaccard similarity, cosine similarity, matching coefficient* |

*Table 2 – Comparative study setup.*

On each of these three matrices, we ran three clustering algorithms: *UPGMA* (Sokal and Michener, 1958), *K-means* (McQueen, 1967) and *Spherical K-means* (Dhillon and Modha, 2001). These algorithms are widely used in Text Mining for unsupervised text categorisation (Misuraca *et al.*, 2018). For UPGMA, which returns a set of nested partitions, we chose a 5-clusters solution according to the true numbers of categories in the BBC dataset. For centre-based algorithms (K-means and Spherical K-means), we instead set *a priori* the true numbers of categories in the BBC news collection as the number of clusters.

To execute the community detection, we considered the similarities among document-vectors. Starting from each lexical matrix, we computed a proximity matrix with three different measures: *jaccard similarity*, *cosine similarity* and *matching coefficient*. The jaccard similarity was used only for the Boolean weighting scheme, whereas the cosine similarity was calculated for the three different weighting schemes. The matching coefficient for each couple of document-vectors was calculated as their dot product. If the weighting scheme is Boolean, the similarity between two different documents is measured as the numbers of shared terms. Considering more complex schemes, the matching coefficient embodies the different occurrences of the shared terms in the documents or the occurrences of the terms in each document dampened by their discriminative power in the collection (i.e. the tf-idf scheme). On these matrices,

graphically viewed as networks of documents, we applied the well-known *Louvain algorithm* (Blondel et al., 2008). According to Yang *et al.* (2016), this algorithm outperforms the other modularity-based algorithms with shorter computing times both in the case of small and large networks. Since the Louvain algorithm does not have input parameters, the number of communities is automatically determined and may differ from the true number of categories. Table 2 synthetically reports the alternatives used in this study.

### *3.3. Results evaluation*

The evaluation of the different approaches was performed, both considering the categorisation process accuracy and the running time. We calculated two external validation indices, *purity* (Manning *et al.*, 2008) and *normalised mutual information* (NMI: Strehl and Ghosh, 2002) to compare the obtained solutions with the so-called *gold standard*, i.e. the true classification of the BBC news collection. The advantage of using these metrics over other well-known external validation measures – such as *precision*, *recall* and *F-measure* (see Sokolova and Lapalme, 2009) – relies on the fact that they do not require a one-to-one correspondence between clusters and classes. Moreover, NMI provides a robust indication of the level of agreement between a given clustering solution and the gold standard. Noteworthily, the running times depend on the technical characteristics of the computer used to perform the analyses. The results presented in the following were obtained on a MacBook Pro 2.8 GHz Intel Core i7, 16 GB 1133 MHz DDR3 with macOS Sierra ver. 10.12.6.

| | *BOOLEAN* | | | *TERM FREQUENCY* | | | *TF-IDF* | | |
|---|---|---|---|---|---|---|---|---|---|
| | **Purity** | **NMI** | **TIME** *(sec.)* | **Purity** | **NMI** | **TIME** *(sec.)* | **Purity** | **NMI** | **TIME** *(sec.)* |
| ***UPGMA*** | *0.240* | *0.006* | *136.026* | *0.231* | *0.000* | *137.069* | *0.237* | *0.006* | *127.045* |
| ***Kmeans*** | ***0.746*** | ***0.595*** | *43.266* | ***0.643*** | ***0.419*** | *33.380* | *0.257* | *0.028* | *15.920* |
| ***Spherical Kmeans*** | ***0.948*** | ***0.842*** | *15.486* | ***0.888*** | ***0.721*** | *17.314* | ***0.949*** | ***0.846*** | *18.194* |
| ***Louvain algorithm*** | | | | | | | | | |
| *jaccard similarity* | *0.230* | *0.000* | *8.891* | - | - | - | - | - | - |
| *cosine similarity* | *0.082* | *0.169* | *3.011* | *0.082* | *0.169* | *3.015* | *0.134* | *0.047* | *4.005* |
| *matching coefficient* | ***0.790*** | ***0.662*** | *3.080* | ***0.740*** | ***0.559*** | *3.559* | ***0.962*** | ***0.880*** | *5.782* |

*Table 3 – Comparison of clustering results on BBC news collection.*

Table 3 reports the validation measures together with the running times for each algorithm. In bold font, we highlighted the best-performing algorithms according to the different weighting schemes. Louvain algorithm outperformed all the other clustering algorithms, considering as similarities among document vectors the matching coefficient and as the importance of each term the tf-idf. Louvain algorithm also provided good results with the *boolean* and *term frequency* weighting scheme. However, in latter this case, *spherical K-means* returned better partitions than the ones obtained with a community detection approach. These results were also confirmed by looking at Figure 1, where the confusion matrices of the best clustering solutions (highlighted in bold in Table 3) are shown. The colour scale ranges from red to green, looking at the frequency value in each cell. The closer the colour gets to the green, the better is the matching between the partition and the gold standard. Among the solutions obtained with the community detection approach, only in one case (term frequency - matching coefficient) Louvain algorithm returned a number of clusters (4) different from the gold standard.

Concerning the running time, we saw that a community detection approach with the Louvain algorithm was faster than the other alternative strategies.

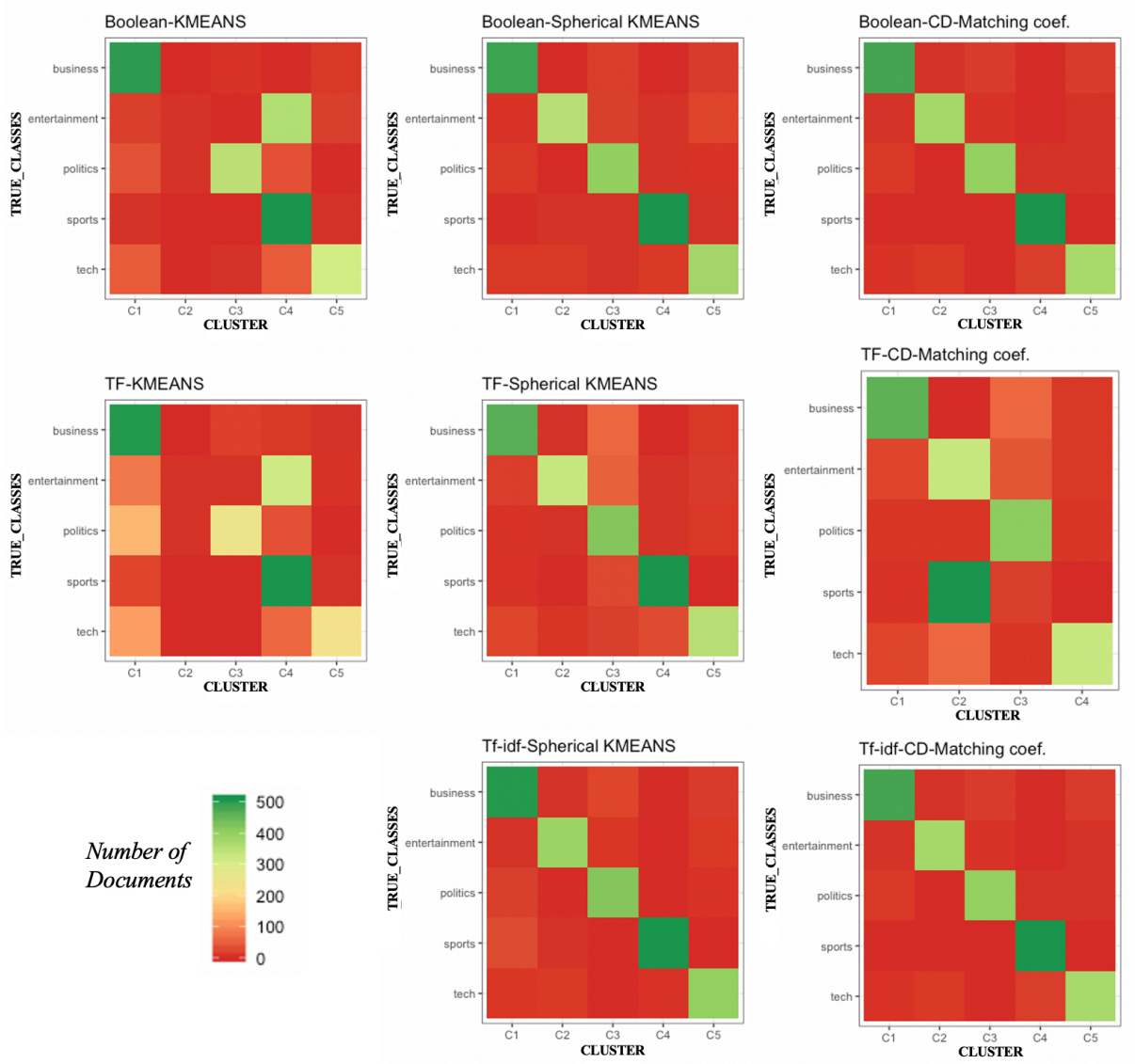


*Figure 1 – Confusion matrices of the clustering results.*

## 4. Remarks and future development

The study proposed in this paper was devoted to comparing the clustering algorithms widely used in the literature of Text Mining with the algorithms used in network analysis for discovering groups of nodes with similar characteristics, in the framework of the so-called community detection. Our preliminary results suggested that the choice of a different weighting scheme may affect the performances of the algorithms. We noted that a more complex weighting scheme such as tf-idf produced better results than those obtained with a binary or a term frequency scheme. In particular, using tf-idf and computing the proximities between documents with the matching coefficient, Louvain algorithm outperformed all the other alternative strategies both in terms of partition quality and running time.

In an unsupervised framework, it is essential to consider that the researcher does not have any prior information about the number and the composition of the clusters. Therefore, a community detection approach returns good results without any other input than the matrix of similarities among documents.

This paper can be considered as a preliminary study. Our findings have to be confirmed by analysing other collections of documents, including also *corpora* of short texts. Future developments will be lead to a broader and more detailed comparison, considering other weighting schemes for the *documents* × *terms* matrix, other proximity measures for the *documents* × *documents* matrix and other community detection methods.

## References

Balbi S. and Misuraca M. (2005). Visualization techniques in non-symmetrical relationships. In S. Sirmakessis (ed.), *Knowledge mining*. Springer-Verlag, pp. 23-29.

Bolasco S. (2013). *L'analisi automatica dei testi*. Carocci editore.

Blondel V.D., Guillaume J.-L., Lambiotte R. and Lefebvre E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008: retrieved from http://stacks.iop.org/ 1742-5468/2008/i=10/a=P10008.

Cadot M., Lelu A. and Zitt M. (2018). *Benchmarking seventeen clustering methods on a text dataset*. Research Report LORIA hal-01532894v6: retrieved from https://hal.archives-ouvertes.fr/hal-01532894v6.

Cheng X., Guo J., Liu S., Wang Y. and Yan X. (2013). Learning topics in short texts by non-negative matrix factorization on term correlation matrix. In: *Proceedings of the 13th SIAM International Conference on Data Mining,* pp. 749-757.

Dhillon I.S. and Modha D.S. (2001). Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42(1-2): 143-175.

Flake G., Tarjan R. and Tsioutsiouliklis M. (2003). Graph Clustering and Minimum Cut Trees. *Internet Mathematics*, 1(4): 385-408.

Fortunato S. and Hric D. (2016). Community detection in networks: A user guide. *Physics Reports*, 659: 1-44.

Greene D. and Cunningham P. (2005). *Producing accurate interpretable clusters from high-dimensional data*. Technical Report TCD-CS-2005-42. Department of Computer Science, Trinity College Dublin: retrieved from https://www.scss.tcd.ie/publications/tech-reports/reports.05/TCD-CS-2005-42.pdf.

Huang A. (2008). Similarity measures for text document clustering. In *Proceedings of the sixth New Zealand computer science research student conference (NZCSRSC2008),* pp. 49-56.

Jia C., Carson M.B., Wang X. and Yu J. (2018). Concept decompositions for short text clustering by identifying word communities. *Pattern Recognition*, 76(C): 691-703.

Larsen B. and Aone C. (1999). Fast and effective text mining using linear-time document clustering. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge Discovery and Data Mining*, pp. 16-22.

Lim K.H., Karunasekera S. and Harwood A. (2017). Clustop: A clustering-based topic modelling algorithm for twitter using word networks. In J.-Y. Nie, Z. Obradovic, T. Suzumura, R. Ghosh, R. Nambiar, C. Wang, H. Zang, R.A. Baeza-Yates, X. Hu, J. Kepner, A. Cuzzocrea, J. Tang, and M. Toyoda (Eds.), *Proceedings of the 2017 IEEE International Conference on Big Data*, pp. 2009-2018.

MacQueen J.B. (1967). Some Methods for classification and Analysis of Multivariate Observations. *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281-297.

Manning C.D., Raghavan P. and Schutze H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

Mikhina E.K. and Trifalenkov V.I. (2018). Text clustering as graph community detection. *Procedia computer science*, 123*:* 271-277.

Misuraca M., Scepi G. and Spano M. (2020). A network-based concept extraction for managing customer requests in a social media care context. *International Journal of Information Management*, 51: 101956.

Misuraca M., Spano M. and Balbi S. (2019). BMS: an improved Dunn index for Document Clustering validation. *Communications in Statistics. Theory and Methods*, 48(20): 5036-5049.

Rattigan M.J., Maier M. and Jensen D. (2007). Graph Clustering with Network Structure Indices. In *Proceedings of the 24th International Conference on Machine Learning*, pp. 783-790.

Salton G. and Buckley C. (1988). Weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5): 513-523.

Salton G., Wong A. and Yang C.S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18: 613-620.

Sokal R.R. and Michener C.D. (1958). A statistical method for evaluating systematic relationships. *University of Kansas Science Bulletin*, 38-2(22): 1409-1438.

Sokolova M. and Lapalme G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4): 427-437.

Steinbach M., Karypis G. and Kumar V. (2000). A comparison of document clustering techniques. In *KDD workshop on text mining*, pp. 525-526.

Strehl A. and Ghosh J. (2002). Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3(Dec): 583-617.

Yang Z., Algesheimer R. and Tessone C.J. (2016). A Comparative Analysis of Community Detection Algorithms on Artificial Networks. *Scientific Reports*, 6(1): 30750.

Wasserman S. and Faust K. (1994). *Social network analysis: Methods and applications*. Cambridge University Press.