Text Mining with Hybrid Biclustering Algorithms

Patryk Orzechowski[1(✉)] and Krzysztof Boryczko[2]

[1] Department of Automatics and Bioengineering,
AGH University of Science and Technology,
Mickiewicza Av. 30, 30-059 Cracow, Poland
patrick@agh.edu.pl

[2] Department of Computer Science,
AGH University of Science and Technology,
Mickiewicza Av. 30, 30-059 Cracow, Poland
boryczko@agh.edu.pl

Abstract. Text data mining is the process of extracting valuable information from a dataset consisting of text documents. Popular clustering algorithms do not allow detection of the same words appearing in multiple documents. Instead, they discover general similarity of such documents. This article presents the application of a hybrid biclustering algorithm for text mining documents collected from Twitter and symbolic analysis of knowledge spreadsheets. The proposed method automatically reveals words appearing together in multiple texts. The proposed approach is compared to some of the most recognized clustering algorithms and shows the advantage of biclustering over clustering in text mining. Finally, the method is confronted with other biclustering methods in the task of classification.

1 Introduction

Finding similarity between documents enables grouping documents in collections, possibly contributing to significant reduction of database query time [32]. Common clustering techniques such as hierarchical clustering, k-means, shared nearest neighbours or deep learning networks are capable of detecting similarities within texts based on the comparison of the words used [12,18,19,27,30]. Nonetheless, the majority of the studies aim at classification, wherein the documents are assigned into fully separated clusters. Whilst the usefulness of such an approach is unquestionable, the content of many documents may cover multiple subjects, thus the issues raised within each document should not be restricted to merely a single cluster. This justifies the formation of other types of algorithms for text mining, with analogical representation in the field of clustering: soft or fuzzy clustering [9,29].

Extracting keywords from particular documents and comparing them with similar ones provides means for aggregation of knowledge regarding the content. One of the popular data mining techniques intended to detect local similarities

L. Rutkowski et al. (Eds.): ICAISC 2016, Part II, LNAI 9693, pp. 102–113, 2016.
DOI: 10.1007/978-3-319-39384-1_9

within the dataset is biclustering (also called co-clustering). Biclustering algorithms, as opposed to classic clustering approaches, take into account rows and columns of the input matrix simultaneously. Some biclustering methods have been successfully applied for text classification [4,10,17,23]. Various biclustering methods have also gained recognition in multiple domains, such as biology and biomedicine, genetics, marketing and text mining [3,13,20,22].

In this article we propose the application of a newly developed hybrid biclustering technique for mining text datasets. We present the Propagation-Based Biclustering Algorithm (PBBA) and the scope of its application towards different types of text datasets. By analysing the content of documents collected from a popular social network, Twitter (i.e. tweets), we compare the proposed approach with the most popular clustering methods popularly used for this task. We also present the application of the algorithm to knowledge spreadsheets in symbolic datasets. Finally, using the example of a popular 20 newsgroup dataset, the algorithm is compared to other state-of-the-art text mining methods in a scenario that lies beyond its design, namely a classification task. The low effectiveness of the proposed approach with regard to this task is discussed.

2 Methodology

Hybrid methods emerged from combining selected aspects of some other existing methods, which allow them to be applied for different types of data. Those methods make use of the techniques, structures or metrics used by other algorithms. The final result provided by a hybrid solution is determined as a combination or aggregation of its components obtained during the process of its execution. Another distinct feature of hybrid algorithms is their broader scope of application. For example, the approach presented herein has been already successfully applied not only to text data, but also to gene expression datasets [28], symbolic datasets, analysis of production data and many other uses.

2.1 Algorithm

The hybrid biclustering algorithm presented here, named the Propagation-Based Biclustering Algorithm (PBBA), has already been successfully applied to various biological datasets [26,28]. The PBBA mechanism inspired by neural networks and associative artificial intelligence [14,15] originates from two biclustering methods: Bimax [31] and xMotifs [24].

Bimax. The Bimax algorithm [31] is a fast divide-and-conquer approach developed for binary data biclustering (in other cases the data needs to be binarized with a given threshold). The algorithm locates all inclusion-maximal biclusters within a given input matrix by dividing rows and columns of the matrix into three smaller submatrices, one of which contains only zeros and may be disregarded in further iterations. Afterwards, the algorithm is applied recursively to the two remaining, possibly overlapping submatrices.

xMotifs. The motivation of the xMotifs algorithm [24] was to provide a representation for gene expression data. An xMotif is an acronym for a conserved gene expression motif. Taking text mining into consideration, this could be a distinct analogy to determining a set of words to characterize each document. Biclusters (or xMotifs) generated with this algorithm need to fulfil a minimum size (here: number of documents), conservation (analogous to the same number of occurrences of any given term) and maximality criteria (analogy of maximum number of terms common with a given xMotif that remains outside the bicluster).

Artificial Associative Intelligence. Artificial associative systems are designed to contextually recall and combine separate pieces of information to form knowledge in a process which resembles recalling information by a human [15]. The neural graphs on which the systems are based, consist of elements which react separately depending on the repeatability and frequency of their input signals triggered by sequences of objects (facts, rules, algorithms etc.).

Propagation-Based Biclustering Algorithm (PBBA). The PBBA algorithm iterates the consecutive rows and seeds non-zero values in each row [26,28]. The pattern created hereby serves to discover similar rows sharing at least one common value in column with the pattern. Subsets with seed row are detected by finding the nearest row with the exact value appearing in the particular column. A special mechanism is carried out to prevent the creation of non-maximum subsets. A general concept of the algorithm is presented in Fig. 1.

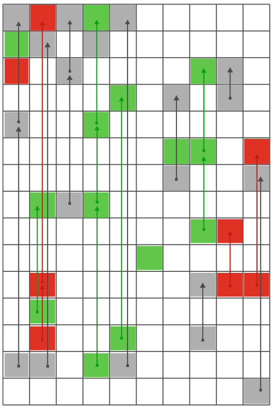


Fig. 1. The mechanism of PBBA - for each row, the algorithm seeks for the nearest row with the same value in a particular column.

Detailed information about the algorithm is presented hereafter. Similarly to Bimax, a minimum size of biclusters needs to be specified, otherwise PBBA (similarly to Bimax) may generate millions of overlapping biclusters. The modifications of the original version of PBBA involved ranking of the input matrix. The cardinality of the terms appearing in each document served as a basis for determining the ranks within the input matrix for the algorithm.

Algorithm 1. Propagation-based biclustering algorithm (PBBA)

```
procedure PBBA(matrix $A$)
    $M_{all} \leftarrow \emptyset$                                    ▷ set with all biclusters
    $R_{all} \leftarrow \emptyset$                                    ▷ set with restricted motifs
    for $i \leftarrow n \dots 1$ do                                   ▷ set each row as seed
        $M_i \leftarrow \emptyset$                                    ▷ store all biclusters common with i-th pattern
        $M_i \leftarrow \text{insert}(M_i, R_i,\ B(i, A_{i*})\ )$     ▷ add seed to retrieved biclusters
        $mask \leftarrow A_{i*}$                                      ▷ propagate the motif to further rows
        $\{lev, pat\} \leftarrow \text{next\_level}(mask)$
        while $pat \neq \emptyset$ do                                 ▷ proceed through all rows similar to $seed$
            $M_i \leftarrow \text{insert}(M_i, R_i,\ B(lev \cup i,\ pat)\ )$   ▷ intersect $lev$ with $M_i$
            $R_{lev} \leftarrow R_{lev} \cup \{pat\}$                 ▷ forbid addition of any subset of $pat$
            $mask(\{j : mask(j) = lev\}) \leftarrow v(mask(j))$       ▷ proceed to next row
            $\{lev, pat\} \leftarrow \text{next\_level}(mask)$
        end while
        $M_{all} \leftarrow M_{all} \cup M_i$;
    end for
    print($M_{all}$)
end procedure
```

3 Results

This section presents the results of algorithm application to various datasets. The PBBA algorithm has been implemented in C++. Three different scenarios have been carried out. First, the algorithm has been applied to the data obtained from Twitter[1] to measure its performance in discovering commonly appearing phrases within tweets. Two clustering methods have been used as references: k-means and hierarchical clustering. Secondly, the algorithm has been applied to a knowledge spreadsheet of US consumer complaints and finally to a 20 newsgroups dataset, which is popularly used for the task of classification.

3.1 Scenario 1: Application to Social Network Data

Tweets are short, up to 140 characters, text messages that are sent by the users of one of the most popular social networks, Twitter. Twitter was queried for 50000 entries containing the "#data" tag published since 2000-01-01. All the data was collected on 2015-04-21. As a result, a total of 44394 documents has been collected. The most common terms (appearing in at least 600 tweets) with *wordcloud* package [8], as presented in Fig. 2.

The data was obtained with *twitteR* package [11], an R-based Twitter client, and processed with text mining (*tm*) package [6,7]. The preprocessing involved removing links, usernames, "RT" (used as an abbreviation of "retweet") and symbols ("&amp;" etc.). All sentences were transformed to lower case. All English stop words were removed ("he's", "we'd" etc.), punctuation and white spaces

[1] twitter.com.

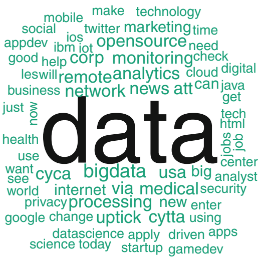
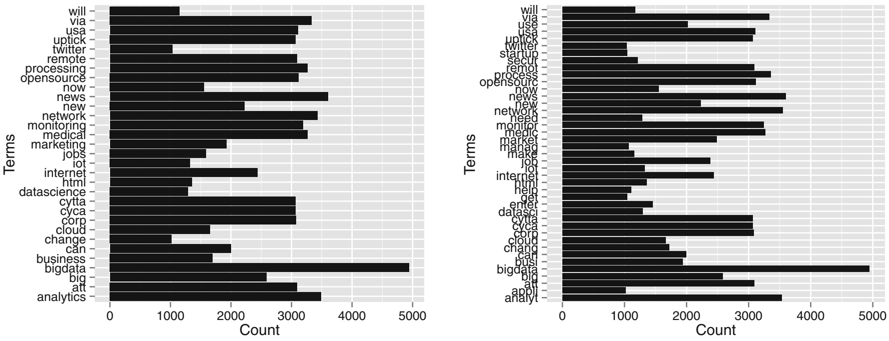


Fig. 2. The wordcloud of the most common terms appearing in Tweets together with the #data tag.

were ignored. Limits on the size of a tweet may cause tweets to be cut at the end of the message. Therefore, the additional removal considered the case of beginning of an unfinished link ("htt") at the end of the tweet. Finally, all the words of at least 3 letters were used to form a document-term matrix. *SnowballC* package [1] was used for stemming words. The statistics for words that appeared at least 1000 times in the database are presented in Fig. 3.



(a) Most common terms.

(b) ¡ost common stemmed terms.

Fig. 3. Most common terms appearing in tweets with the #data tag.

As the majority of words appeared only in a few tweets, sparse terms were removed from the matrix. Notice that the term-document matrix has not been sparsified as in [35] as that produced incorrect results. Terms were accepted with at least 0.1 % non-zero values across all documents. As a result, the original matrix shrank to 1290 terms and contained 301659 out of 406716 values. The stemmed matrix featured 1238 terms and 323410 non-zero values.

The PBBA was set to detect terms that appeared in at least 10 different tweets. At least two terms needed to appear in the same tweet. The input matrix for the PBBA contained words and their occurrences in the text, forming a document-term matrix with zeroes indicating lack of term in a document. The effect of stemming the words has been also presented.

Task 1: All Tweets. The PBBA algorithm detected 11036 biclusters in total for the non-stemmed dataset and 14576 biclusters for the stemmed dataset. The statistics of most common terms appearing in the same tweet with the #data tag for the stemmed and not stemmed datasets are presented in Table 1.

Table 1. Text biclusters detected by PBBA for the Twitter dataset - the most common terms used in the same tweet for (a) not stemmed and (b) stemmed dataset. Difference in occurrences of terms is the result of merging words with the same root.

Tweets	Terms in the same tweet
4828	bigdata data
3575	data news
3345	data network
3260	data processing
3257	data medical

(a) Not stemmed dataset

Tweets	Terms in the same tweet
4828	bigdata data
3575	data news
3460	data network
3352	data process
3261	data medic

(b) Stemmed dataset

For both datasets, twenty of the most popular terms used in the tweets had the same roots. This indicates that the most popular terms appear in the vast majority of the tweets in their exact form. A slight difference may be noticed within the "processing-process" term, which is considered the same phrase in the stemmed database.

Task 2: Unique Tweets. The second scenario involved biclustering on unique tweets only. We filtered documents that have not been retweeted and those which had the same content. Thus, the size of the dataset was reduced to 17083 for non-stemmed and 17594 for stemmed terms. The procedure applied guaranteed that each tweet or retweet with the same content was counted once only.

All word associations presented in Table 2 are commonly used together. This makes PBBA an interesting choice for the analysis of the common data within different documents. On the other hand, the sensitivity of PBBA for variants of similar words could also be considered as the curse of the algorithm, as all tweets that differ by one word only would result in generation of other bicluster(s).

Comparison with K-means. To determining the optimum number of clusters, we adapted a technique described by Hothorn [16] explained hereafter. After normalizing the data, we tested different numbers of clusters (from 2 to 200) and plotted the within-group sum of squares for each partition. As k-means require setting a random seed, the results obtained in multiple runs of the algorithm

Table 2. Text biclusters detected by PBBA for unique tweets - the most common terms used in the same tweet for (a) not stemmed and (b) stemmed dataset. Notice that different terms occurred the most often in each of cases.

Tweets	Terms in the same tweet
1495	data via
1490	bigdata data
1382	analytics data
1097	data jobs
940	big data

(a) Not stemmed dataset

Tweets	Terms in the same tweet
1519	data via
1501	bigdata data
1423	analyt data
1337	data job
996	data market

(b) Stemmed dataset

were different. Nonetheless, in all scenarios taken into account the WCSS measure reached very high values. We concluded that no consensus on the optimum setting of the number of clusters for k-means may be reached. Thus, we decided to take into consideration 42 clusters for stemmed and non-stemmed data. The results of clustering with k-means are presented in Table 3. We selected a couple of the most repetitive terms, sorted in decreasing order, similarly to Zhao [35], to serve as representative for the tweets.

Table 3. Example clusters for Twitter obtained with k-means, for (a) non-stemmed and (b) stemmed dataset.

Tweets	Terms in the same tweet
28279	data via analytics new
4444	bigdata data analytics datascience
3070	cyca opensource remote uptick
1052	html gamedev appdev internet
739	apply now jobs looking

(a) Non-stemmed dataset

Tweets	Terms in the same tweet
24980	data via analyt new
4077	bigdata analyt data datasci
3070	cyca opensourc remot uptick
1625	use data via bigdata
1387	job now appli hire

(b) Stemmed dataset

Clustering in multiple cases revealed one large cluster (with over half of the tweets from the dataset), a couple of smaller clusters (with a couple of hundreds of tweets) and multiple small clusters. Clustering performed fine if the whole content of the tweet was exact. This was the issue with one of commonly retweeted contents. PBBA easily managed to detect commonly used phrases and built biclusters of different sizes around them.

Comparison with Hierarchical Clustering. We compared the results from PBBA and k-means with those obtained from clustering using Hierarchical Clustering. After removing terms that had over 98 % and 99 % of sparsity, the remaining 96 terms (117 in case of the stemmed dataset) were normalized and divided into 20 different clusters with original Ward's minimum variance criterion used to form spherical clusters [25]. Hierarchical clustering managed to find some connotations of the words between different tweets. It detected a couple of meaningful clusters, such as "jobs apply now" and "change enter", or "html internet", but also a meaningless "news network medical processing monitoring att opensource usa remote cytta cyca uptick". The majority of terms fell into multiple one-term-size clusters or into a single big cluster with non-related values. Different levels of threshold didn't change that. In all cases, "data" formed its own cluster.

3.2 Scenario 2: Symbolic Datasets

One of the very popular types of datasets operating on symbolic representations of structures instead of using numeric data only is called knowledge spreadsheet for symbolic computing [33]. In this particular domain, the semantics determine the symbolic relations between the structures.

An example of a symbolic dataset is the consumer complaint database[2] collected on 2015-07-22. It contains 409400 consumer complaints regarding financial products and the current status of the cases. It covers the product name, sub-product, company name, US state, information if the answer has been delivered on time and whether the customer was satisfied with it.

In this scenario, as each symbolic expression may be substituted by a number, PBBA allows us to discover inner logic behind the dataset. An exemplary advanced analysis difficult to perform on a pivot table is presented in Table 4. The analysis concerns the most popular response to a complaint and its punctuality for different companies. Thus, biclusters with at least 3 columns containing the product name, company name and the response timeliness have been considered.

Table 4. The most popular responses to complaints and their punctuality for individual companies

Bicluster size	Product	Company	Timely response
23493×3	Mortgage	Bank of America	No
14614×3	Mortgage	Wells fargo	No
11879×3	Mortgage	Ocwen	No
9769×3	Mortgage	JPMorgan chase	No
7738×3	Mortgage	Bank of America	Yes
6863×3	Mortgage	Nationstar mortgage	No

Analysis of the PBBA results offers an easy way to get a valuable insight into different structures within the data. For example, the majority of untimely responses that ended up with explanations concerned mortgages (84131 cases),

[2] catalog.data.gov/dataset/consumer-complaint-database.

debt collection (35881 cases) and credit reporting (26174 cases). Comparing the responses that were given on time, the majority of them concerned mortgages (28289 cases), debt collection (10137 cases) and bank accounts or services (6823 cases). The most popular state for complaints on mortgages that ended with explanations was California (CA), covering 20926 cases, out of which in 14773 cases the answer was outside and 5179 within time limitations (some of the cases are still under consideration). The next states were Florida (FL) with 10075 cases and New York (NY) with 4959 cases. Depending on the desired query, a different logic behind the dataset may be discovered.

3.3 Scenario 3: Classification of a 20 Newsgroups Dataset

The third scenario involves the application of the algorithm to the task of classification. A very popular 20 newsgroups dataset[3] has been used as a reference. This database contains set of almost 20000 articles collected from 20 different groups, some of which are very closely related to each other. For comparison, a database from Hussain et al. [17] has been taken, provided by Grimal[4]. The dataset is divided into 6 separate subsets: M2 (500 documents and 2 clusters), M5 (500 documents, 5 clusters), M10 (500 documents and 10 clusters), NG1 (400 documents and 2 clusters), NG2 (1000 documents and 5 clusters) and NG3 (1600 documents and 8 clusters). Each subset contains 10 folds with documents selected randomly using the k-medoids algorithm [17].

In concordance with other studies, micro-average precision was used for assessing the PBBA effectiveness [5]. The PBBA algorithm has been run on each fold of the dataset. In each case, PBBA has generated multiple overlapping biclusters. Including the smallest sizes of the biclusters caused PBBA to perform excessively long computations and generate as many as millions of small biclusters. On the other hand, disregarding the smallest biclusters meant that multiple documents have not been included in any of the biclusters. In this cas, we have decided to match them with an artificial empty class and treat as false positives. As the number of biclusters in PBBA is unknown beforehand, the bicluster matching a specific class the most has been considered as the right one.

For each dataset taken into considerations, the results obtained with PBBA have been unsatisfactory. Even for the theoretically easiest task (M2), the micro-average precision of PBBA was around 0.1, far below the results presented in literature. Modifying the PBBA parameters for the minimum bicluster size did not increase the score, as the level of misclassified documents went up.

We conclude that the PBBA algorithm is unsuitable for performing hard classification of text datasets Chi-sim [17] or other co-clustering methods are far more suitable for this task.

[3] http://www.qwone.com/~jason/20Newsgroups/.

[4] membres-lig.imag.fr/grimal/code/XSim.tar.gz.

4 Conclusions

Providing personalized content to users, such as recommendations, increases user involvement with a service. These days, personalized recommendation systems [21] or contextual advertising systems [2,34] match users with the service content. With a high dose of probability, this functionality is already implemented in Twitter, as users are recommended to follow certain people. We assume that recommendations utilizes context similarity.

This paper presents the concept of hybrid biclustering algorithms. Those are methods combining selected techniques from existing solutions. The major advantage of hybrid biclustering algorithms is the broader area of their application. Using the example of PBBA, we proved that the said algorithms may provide an insight into textual data. Analysis of common terms with biclustering detected much more realistic word connotations compared with clustering. This was particularly visible when considering unique tweets. Thus, the technique may be applied to dividing documents into multiple overlapping groups based on their content.

We noticed that term stemming may further increase the accuracy of the results obtained. For the stemmed dataset, PBBA managed to detect biclusters of larger size. This proves that the stemming provides better compression of data as terms originating from the same word are merged together. It should be noted, however, that multiple tweets in our dataset were not written in English. Therefore, stemming analysis becomes much more susceptible to errors.

Analysis of symbolic datasets in knowledge spreadsheets proved that a hybrid biclustering algorithm may successfully retrieve statistical information from a given dataset. This includes the information regarding the most commonly co-appearing values in a dataset. This application of the algorithm resembles aspect filtering or pivot tables.

A classification task revealed the algorithm weakness, involving generating multiple overlapping biclusters. PBBA failed to provide a correct division in each of the subsets of a 20 newsgroups dataset. Nonetheless, the criterion was very rigorous. Perhaps developing a mechanism of merging biclusters could increase the algorithm performance. Nonetheless, it is doubtful whether such a technique may outperform state of the art methods.

Acknowledgments. This research was funded by the Polish National Science Center (NCN), grant No. 2013/11/N/ST6/03204. This research was supported in part by PL-Grid Infrastructure.

References

1. Bouchet-Valat, M.: SnowballC: Snowball stemmers based on the C libstemmer UTF-8 library (2014). http://CRAN.R-project.org/package=SnowballC. r package version 0.5.1

2. Broder, A., Fontoura, M., Josifovski, V., Riedel, L.: A semantic approach to contextual advertising. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 559–566. ACM (2007)
3. Busygin, S., Prokopyev, O., Pardalos, P.M.: Biclustering in data mining. Comput. Oper. Res. **35**(9), 2964–2987 (2008)
4. de Castro, P.A.D., de França, F.O., Ferreira, H.M., Von Zuben, F.J.: Applying biclustering to text mining: an immune-inspired approach. In: de Castro, L.N., Von Zuben, F.J., Knidel, H. (eds.) ICARIS 2007. LNCS, vol. 4628, pp. 83–94. Springer, Heidelberg (2007). http://dl.acm.org/citation.cfm?id=1776274.1776284
5. Dhillon, I.S., Mallela, S., Modha, D.S.: Information-theoretic co-clustering. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 89–98. ACM (2003)
6. Feinerer, I., Hornik, K.: tm: Text Mining Package (2014). http://CRAN.R-project.org/package=tm. r package version 0.6
7. Feinerer, I., Hornik, K., Meyer, D.: Text mining infrastructure in r. J. Stat. Softw. **25**(5), 1–54 (2008). http://www.jstatsoft.org/v25/i05/
8. Fellows, I.: wordcloud: Word Clouds (2014). http://CRAN.R-project.org/package=wordcloud. r package version 2.5
9. Filippone, M., Masulli, F., Rovetta, S., Mitra, S., Banka, H.: Possibilistic approach to biclustering: an application to oligonucleotide microarray data analysis. In: Priami, C. (ed.) CMSB 2006. LNCS (LNBI), vol. 4210, pp. 312–322. Springer, Heidelberg (2006)
10. Franca, F.O.D.: Scalable Overlapping Co-clustering of Word-Document Data, pp. 464–467. IEEE (2012). http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6406666
11. Gentry, J.: twitteR: R Based Twitter Client (2015). http://CRAN.R-project.org/package=twitteR. r package version 1.1.8
12. Hartigan, J.A., Wong, M.A.: Algorithm as 136: a k-means clustering algorithm. Appl. Stat. **28**, 100–108 (1979)
13. Henriques, R., Madeira, S.: Biclustering with flexible plaid models to unravel interactions between biological processes. IEEE/ACM Trans. Comput. Biol. Bioinf. **PP**(99), 1–1 (2015)
14. Horzyk, A.: Information freedom and associative artificial intelligence. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2012, Part I. LNCS, vol. 7267, pp. 81–89. Springer, Heidelberg (2012). http://dx.doi.org/10.1007/978-3-642-29347-4_10
15. Horzyk, A.: How does human-like knowledge come into being in artificial associative systems?. In: Proceedings of the 8-th International Conference on Knowledge, Information and Creativity Support Systems, Krakow, Poland (2013)
16. Hothorn, T., Everitt, B.S.: A Handbook of Statistical Analyses using R, 3rd edn. Chapman and Hall/CRC, Boca Raton (2014)
17. Hussain, S.F., Bisson, G., Grimal, C.: An improved co-similarity measure for document clustering. In: Proceedings of the 2010 Ninth International Conference on Machine Learning and Applications, ICMLA 2010, pp. 190–197 (2010). http://dx.doi.org/10.1109/ICMLA.2010.35
18. Jain, A.K.: Data clustering: 50 years beyond k-means. Pattern Recogn. Lett. **31**(8), 651–666 (2010)
19. Jiang, Z., Li, L., Huang, D., Jin, L.: Training word embeddings for deep learning in biomedical text mining tasks. In: 2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 625–628. IEEE (2015)

20. Kaiser, S.: Biclustering: Methods, Software and Application. Ph.D. thesis, Ludwig-Maximilians-Universitt Mnchen (2011)
21. Liang, T.P., Lai, H.J., Ku, Y.C.: Personalized content recommendation and user satisfaction: theoretical synthesis and empirical findings. J. Manag. Inf. Syst. **23**(3), 45–70 (2006)
22. Madeira, S.C., Oliveira, A.L.: Biclustering algorithms for biological data analysis: a survey. IEEE/ACM Trans. Comput. Biol. Bioinf. **1**(1), 24–45 (2004)
23. Mimaroglu, S., Uehara, K.: Bit sequences and biclustering of text documents. In: icdmw, pp. 51–56. IEEE (2007)
24. Murali, T., Kasif, S.: Extracting conserved gene expression motifs from gene expression data. Proc. Pacific Symp. Biocomputing **3**, 77–88 (2003)
25. Murtagh, F., Legendre, P.: Wards hierarchical agglomerative clusteringmethod: which algorithms implement wards criterion? J. Classif. **31**(3), 274–295 (2014)
26. Orzechowski, P., Boryczko, K.: Propagation-based biclustering algorithm for extracting inclusion-maximal motifs. Computing and Informatics (2016), in print
27. Orzechowski, P., Boryczko, K.: Parallel approach for visual clustering of protein databases. Comput. Inform. **29**(6+), 1221–1231 (2010). http://www.cai.sk/ojs/index.php/cai/article/view/140
28. Orzechowski, P., Boryczko, K.: Hybrid biclustering algorithms for data mining. In: Squillero, G., Burelli, P. (eds.) EvoApplications 2016. LNCS, vol. 9597, pp. 156–168. Springer, Heidelberg (2016). doi:10.1007/978-3-319-31204-0_11
29. Peters, G., Crespo, F., Lingras, P., Weber, R.: Soft clustering fuzzy and rough approaches and their extensions and derivatives. Int. J. Approximate Reasoning **54**(2), 307–322 (2013). http://www.sciencedirect.com/science/article/pii/S0888613X12001739
30. Poikolainen, I., Neri, F., Caraffini, F.: Cluster-based population initialization for differential evolution frameworks. Inf. Sci. **297**, 216–235 (2015)
31. Prelić, A., Bleuler, S., Zimmermann, P., Wille, A., Bühlmann, P., Gruissem, W., Hennig, L., Thiele, L., Zitzler, E.: A systematic comparison and evaluation of biclustering methods for gene expression data. Bioinformatics **22**(9), 1122–1129 (2006)
32. Steinbach, M., Karypis, G., Kumar, V., et al.: A comparison of document clustering techniques. In: KDD Workshop on Text Mining, vol. 400, Boston, MA, pp. 525–526 (2000)
33. Travers, M., Paley, S.M., Shrager, J., Holland, T.A., Karp, P.D.: Groups: knowledge spreadsheets for symbolic biocomputing. Database 2013, bat061 (2013)
34. Zhang, K., Katona, Z.: Contextual advertising. Mark. Sci. **31**(6), 980–994 (2012)
35. Zhao, Y.: R and Data mining: examples and case studies. Elsevier Science (2012). http://books.google.com.au/books?id=FEOh08LBD9UC