Applying Biclustering to Text Mining: An Immune-Inspired Approach

Pablo A.D. de Castro, Fabrício O. de França, Hamilton M. Ferreira, and Fernando J. Von Zuben

Laboratory of Bioinformatics and Bio-Inspired Computing - LBIC
School of Electrical and Computer Engineer – FEEC
University of Campinas – UNICAMP
Campinas-SP, Brazil
{pablo,olivetti,hmf,vonzuben}@dca.fee.unicamp.br

Abstract. With the rapid development of information technology, computers are proving to be a fundamental tool for the organization and classification of electronic texts, given the huge amount of available information. The existent methodologies for text mining apply standard clustering algorithms to group similar texts. However, these algorithms generally take into account only the global similarities between the texts and assign each one to only one cluster, limiting the amount of information that can be extracted from the texts. An alternative proposal capable of solving these drawbacks is the biclustering technique. The biclustering is able to perform clustering of rows and columns simultaneously, allowing a more comprehensive analysis of the texts. The main contribution of this paper is the development of an immune-inspired biclustering algorithm to carry out text mining, denoted BIC-aiNet. BIC-aiNet interprets the biclustering problem as several two-way bipartition problems, instead of considering a single two-way permutation framework. The experimental results indicate that our proposal is able to group similar texts efficiently and extract implicit useful information from groups of texts.

Keywords: Artificial Immune System, Biclustering, Two-way Bipartition, Text mining.

1 Introduction

With the popularization of the web and the collaboration of users to produce digital contents, there was an expressive increase in the amount of documents in electronic format. Therefore, several text mining tools have been proposed to organize and classify these documents automatically, since a personal manipulation is becoming more and more prohibitive.

The main difficulty associated with automated analysis of documents is that textual information is highly subjective. Although there are many efficient data mining tools available in the literature, converting such information into a rather objective and computer-interpretable codification is far from being straightforward. Invariably, this scenario imposes several limitations to the performance of the analytical tools. On the other hand, while this conversion step is still not entirely satisfactory, there must be

L.N. de Castro, F.J. Von Zuben, and H. Knidel (Eds.): ICARIS 2007, LNCS 4628, pp. 83–94, 2007.

an effort to enhance the clustering and classification techniques in order to handle the noisy and incomplete information generated by methods for semantic interpretation.

Most of the text mining tools reported in the literature represent the documents by a vector of attributes [6]. After submitting the text to a filter capable of extracting irrelevant words and particles of composed words (filters may differ a lot and may present very distinct degrees of sophistication), each position of the vector is related to a word or textual expression of the preprocessed text, and each entry of the vector contains a number which says how many times the corresponding word or expression appears in the text. Therefore, the text mining tools performs several types of data analysis based on statistical properties presented by the attributes extracted from preprocessed texts.

In the context of text clustering, although standard clustering algorithms such as *k*-means, Self Organized Maps, and Hierarchical Clustering have been successfully applied to text mining, they present a well- known limitation when dealing with large and heterogeneous datasets: since they group texts based on global similarities in their attributes, partial matching cannot be detected. For instance, if two or more texts share only a subset of similar attributes, the standard clustering algorithms fail to identify this specificity in a proper manner. Besides, they assign a text to only one category, even when the texts are involved in more than one category.

A recent proposal to avoid these drawbacks is the so-called biclustering technique, which performs clustering of rows and columns simultaneously, allowing the extraction of additional information from the dataset [2]. Biclustering may be implemented considering a two-way permutation problem or performing several two-way bipartitions of the whole dataset, as will be clarified in Section 2.

The objective of this paper is twofold. The first one is to apply a biclustering technique to the text mining problem, centered on the two-way bipartition framework. To do that, a very flexible methodology is considered to implement multiple two-way bipartitions, characterized by the possibility of discarding an arbitrary number of rows (texts) and columns (attributes of the corresponding texts) of the original matrix. This methodology is in contrast with the ones that are capable of finding all biclusters in a matrix [12]. So, if we have a large matrix, the computational cost to generate the biclusters using the latter approach becomes prohibitive.

Aiming at proposing a feasible solution, the second objective is to present a novel heuristic-based methodology to generate the multiple two-way bipartitions. Once the generation of biclusters can then be viewed as a multimodal combinatorial optimization process, we will explore the already attested ability of immune-inspired algorithms to deal with this challenging scenario.

We evaluate the proposed methodology by applying it to a dataset which contains 60 texts classified into 3 different categories. The results indicate that our algorithm is able to group similar texts correctly. Besides, the biclusters present the more relevant words to represent a certain category. This information is useful when composing an intelligent search engine for guiding a user on its search for related documents on different areas.

The paper is organized as follows. Section 2 presents a brief introduction to the biclustering technique and its applications. Section 3 describes in details the proposed approach to generate biclusters. In Section 4, the experimental results are presented and analyzed. Finally, in Section 5 we conclude the paper and provide directions for further steps in the research.

2 A Brief Overview of Biclustering

In data mining, biclustering is referred to the process of simultaneously find clusters on the rows and columns of a matrix [2]. This matrix may represent different kinds of numerical data, such as objects and its attributes (comprising the rows and columns of the matrix, respectively).

There are several approaches to deal with the biclustering problem [8][10][11][13][14]. The most traditional one is to interpret biclustering as a two-way permutation problem, so that the purpose is to simultaneously reorder rows and columns of the original matrix, in an interactive manner, toward the production of multiple clusters in different regions of the obtained matrix, as illustrated in Fig. 1.

$$\begin{bmatrix} 1 & 2 & 3 & 1 \\ 3 & 3 & 2 & 2 \\ 1 & 2 & 2 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 2 & 3 & 1 \\ 1 & 2 & 2 & 1 \\ 3 & 3 & 2 & 2 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 1 & 3 & 2 \\ 1 & 1 & 2 & 2 \\ 3 & 2 & 2 & 3 \end{bmatrix}$$

Fig. 1. The original matrix is reordered (rows and columns) to generate the biclusters

Another possibility, which will be explored in this paper, is to create several sub-matrices from the original matrix aiming at maximizing some index designed to measure similarity, or alternative clustering aspects, in these sub-matrices. As the construction of the sub-matrices involves defining which rows and columns of the original matrix will be included and which ones will be excluded, we may interpret the biclustering as multiple two-way partition problems, as illustrated in Fig. 2.

$$\begin{bmatrix} 1 & 2 & 3 & 1 \\ 3 & 3 & 2 & 2 \\ 1 & 2 & 2 & 1 \end{bmatrix} \Rightarrow \begin{matrix} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{matrix} rows = \{1,3\} \\ cols = \{1,4\} \end{matrix} \\ \begin{bmatrix} 2 & 2 \\ 2 & 1 \end{bmatrix} \begin{matrix} rows = \{2,3\} \\ cols = \{3,4\} \end{matrix} \end{matrix}$$

Fig. 2. Two biclusters extracted from the original matrix

Additional aspects may be considered to distinguish biclustering techniques. They can be classified by: (*i*) the type of biclusters they find; (*ii*) the structure of these biclusters; and (*iii*) the way the biclusters are discovered.

The type of the biclusters is related to the concept of similarity between the elements of the matrix. For instance, some algorithms search for constant value biclusters, while others search for coherent values of the elements or even for coherent evolution biclusters.

The structure of the biclusters can be of many types. There are single bicluster algorithms, which find only one bicluster in the center of the matrix; the exclusive columns and/or rows, in which the biclusters cannot overlap in either columns or rows of the matrix; arbitrary positioned, overlapping biclusters and overlapping biclusters with hierarchical structure.

The way the biclusters are discovered refers to the number of biclusters discovered per run. Some algorithms find only one bicluster, others simultaneously find several biclusters and some of them find small sets of biclusters at each run.

Besides, there are nondeterministic and deterministic algorithms. Nondeterministic algorithms are able to find different solutions for the same problem at each execution, while deterministic ones produce always the same solution. The algorithm used in this paper is nondeterministic.

The biclustering approach covers a wide scope of different applications. The main motivation is to find data points that are correlated under only a subset of the attributes. Usual clustering methods cannot identify this type of local correlation. Some examples of biclustering applications are dimensionality reduction [1], information retrieval and text mining [5], electoral data analysis [9], and biological data analysis [1].

3 An Immune-Inspired Algorithm for Biclustering

Since the generation of biclusters can be viewed as a multimodal combinatorial optimization problem, we have considered an immune-inspired algorithm to generate them. The first attempt to synthesize an immune-inspired method for biclustering has taken copt-aiNet (Artificial Immune Network for Combinatorial Optimization) as the search engine [4]. Our approach is another extension of the aiNet algorithm, which was proposed by de Castro and Von Zuben [3], to deal with biclustering and is denoted BIC-aiNet, an artificial immune network for biclustering. Though copt-aiNet and BIC-aiNet are both derived from aiNet to deal with combinatorial optimization, BIC-aiNet considers the biclustering problem as multiple two-way bipartitions, and copt-aiNet has adopted the two-way permutation framework.

When compared with alternative approaches, immune-inspired algorithms present distinguishing characteristics that proved to be specially interesting for solving multimodal problems, such as: (*i*) the multi-population paradigm, where several populations are evolved at the same time (in contrast with usual population algorithms, where there is just one population that converges to only one solution); (*ii*) the dynamic control of the population size, which allows the automatic definition of the number of candidate solutions based on the characteristics of the problem; and (*iii*) the diversity maintenance property, which enhances the searching ability of the algorithm by allowing the coexistence of multiple alternative solutions in the same run.

Next, we present how the biclusters are represented in BIC-aiNet as well as its functioning.

3.1 Coding

Given a data matrix with n rows and m columns, the structure chosen to represent a bicluster (an individual in the population) on the algorithm for this data matrix is by using two ordered vectors. One vector represents the rows and the other one represents the columns, with length *n*’ < *n* and *m*’ < *m*, respectively. Each element in the vectors is an integer representing the index of row or column that is present at the bicluster. This representation tends to be more efficient than the binary representation (two vectors of length *n* and *m,* where the number 1 represents the presence of a given row/column and 0 the absence) once the biclusters are usually much smaller than the original data matrix.

Each individual in the population represents a single bicluster and may have a distinct value for *n*’ and *m*’. Figure 3 shows an example of coding

$$\begin{bmatrix} 1 & 2 & 3 & 1 \\ 2 & 3 & 2 & 2 \\ 1 & 2 & 2 & 1 \end{bmatrix}$$

(a)

row:	2	3	
col:	1	3	4

(b)

$$\begin{bmatrix} 2 & 2 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

(c)

Fig. 3. Example of coding: **(a)** the original data matrix; **(b)** an individual of the population; **(c)** the correspondent bicluster

3.2 Fitness Function

The fitness function used to evaluate a given bicluster is given as follows:

$$f(M,N)=\frac{R}{\lambda}+\frac{w_c\cdot\lambda}{|M|}+\frac{w_r\cdot\lambda}{|N|} \tag{1}$$

where *N* and *M* are the set of rows and columns, respectively, in the bicluster, *R* is called the residue of a bicluster and is calculated as in Eq. 2, λ is a residue threshold (the maximum desired value for residue), w_c is the importance of the number of columns, and w_r the importance of the number of rows. The operator |.| provides the number of elements on a given set. The residue of a bicluster assumes the form:

$$R=\sum_{i,j}\frac{\left(r_{ij}-r_{Ij}-r_{iJ}+r_{IJ}\right)^2}{|N|\cdot|M|} \tag{2}$$

where r_{ij} is the value of the element (*i,j*) on the original data matrix, r_{Ij} is the mean value of row i, r_{iJ} represents the mean value of column j, and r_{IJ} is the mean value considering all the elements in the bicluster.

With this fitness function we have a ratio between two conflicting objectives: minimizing the residue (variance of elements in the bicluster) and maximizing its volume. Notice that, for a bicluster to be meaningful, it should contain a reasonable number of elements so that some knowledge can be extracted. Also, it is important to maintain some cohesion between its elements.

3.3 The Algorithm

The main algorithm begins by generating a random population of biclusters consisting of just one row and one column, that is, just one element of the data matrix will be used as a “seed” to promote the growth of a local bicluster.

After the initialization, the algorithm enters its main loop. Firstly, the population is cloned and mutated as it will be described later, then if the best clone has a better fitness than its progenitor, it will replace it in the population.

Every **sup_it** iterations the algorithm performs a suppression of very similar biclusters and then inserts new cells. This procedure causes a fluctuation in population size and helps the algorithm to maintain diversity and to work with just the most meaningful biclusters.

The main algorithm of BIC-aiNet is given by the following pseudo-code:

```
Cells = initial_population();
For it=0..max_it do
     For each cell do

          C = clone(cell);
          C = mutate(C);
          If best(C) better than cell
               cell = C;
          End If
     End For
     If it mod sup_it
          Suppress (cell);
          Insert_new_cells();
     End If
End For
```

3.4 Mutation

The mutation operation consists of simple random insertion/removal procedures applied to the bicluster. Given a bicluster, the algorithm first chooses between insertion or removal operation with equal probability. After that, another choice is made, also with the same probabilities: it will perform the operation on a row or a column. On the insertion case, an element that does not belong to the bicluster is chosen and inserted on the row/column list in ascending order. If the removal case is chosen, it generates a random number that represents the position of the element to be removed from the vectorial representation of the bicluster.

The mutation operator is given as follows:

```
If rand1 < 0.5
       If rand2 < 0.5
              Insert_new_row();
          Else
              Insert_new_column();
       End If
    Else
       If rand2 < 0.5
              Remove_row();
         Else
              Remove_column();
       End If
   End If
```

3.5 Suppression

The suppression step is given by:

```
For each pair i,j of biclusters do
     If |celli ∩ cellj| > ε*max(size(celli, cellj)/100
          destroy worst cell
     End If
End For
```

The suppression operation is a straightforward procedure. For each pair of biclusters on the population, it generates the intersection of both sets and counts the number of elements on it, that is, the number of common elements. If this number is greater than a certain threshold ε, it removes the bicluster presenting the worse fitness. After every suppression, the algorithm inserts new cells trying to create biclusters with elements (rows or columns) that still does not belong to any existing bicluster.

4 Experimental Results

This section describes the experiments carried out to evaluate the proposed methodology and to show that interesting information can be extracted from the generated biclusters.

We have applied the methodology to a dataset containing 60 texts and the objective is to group similar documents into one or more categories. Instead of considering the frequency of the attributes, we will represent only presence or absence of a given attribute in the text under analysis. Also, though textual expressions could be considered as a single attribute, we will be restricted to words in their radical form.

Moreover, analyzing the generated biclusters, we will be able to extract the relevant words for each category, extending the usefulness of the methodology. By extracting these keywords, conventional search engines in the internet can be used to find related texts in additional databases.

Considering these two objectives, we show the advantages of biclustering techniques over standard clustering algorithms for text mining.

4.1 Dataset Description and Its Vectorial Representation

The dataset used during the experiments is composed of 60 documents divided into three main categories of 20 texts, and each of these contains two subcategories of 10 texts. Notice that the labels of the texts will not be provided to the biclustering tool, but will be used for further analysis, after concluding the biclustering. The documents were taken from the Brazilian newspaper *Folha de São Paulo* and the categories chosen correspond to sections of the newspaper: Money, Sport and Informatics. Sport news are labeled S and contain two subclasses: Car Racing (S1) and Soccer (S2). Money reports are labeled M and its subcategories are Oil (M1) and International Commerce (M2). The last class, Informatics, is labeled I and is divided into the subcategories Internet (I1) and Technology (I2).

In order to build this dataset, each document had its words extracted and only the radical of each word was taken. After that, a matrix was generated in which each line represents a document and each column presents binary values representing the absence (0) or presence (1) of the related word in the document. Finally, the words that appear in only one document are eliminated from the matrix. After filtered, a total of 1007 words were used as attributes of the 60 documents.

4.2 Parameters Values

This subsection describes the parameters adopted during the experiments as well as their values. Table 1 summarizes these values.

Table 1. Parameters values

Parameter	values
# biclusters	300
# iterations	3000
Residue threshold (R)	1
Row weight (w_r)	5
Column weight (w_c)	19
Suppression Threshold (ε)	80

The algorithm was set to generate up to 300 biclusters during 3000 iterations. As we are dealing with binary data, the residue threshold chosen has a value of “1”. The row importance weight and the column importance weight were set empirically in order to achieve a balance between the high volume and the low residue scores. The column importance weight, in this particular case, controls how many words will be used on each bicluster. When the number of columns is high, the results tend to be closer to the ones produced by traditional clustering algorithm.

4.3 Analysis of the Obtained Biclusters

The generated biclusters have residue values in the range between 0.98 and 3.19, meaning that the grouped texts exhibit high coherence among each other. Every document on the dataset belongs to at least one bicluster (though this is not an imposition or necessary condition).

We have observed during the experiments that our algorithm is able to group similar documents efficiently. After the generation of the biclusters, we verified the labels of the texts of a same bicluster and most of them are of the same category. Next, we describe the most significative obtained biclusters and the interesting features that can be extracted from them.

The first bicluster, the one which has the smaller residue value, comprises six out of nine documents belonging to M1 category, indicating the capability of BIC-aiNet to group texts with the same subject. From this bicluster we may extract some dominant words, i.e., words that appear in every document of the bicluster, in order to categorize these texts. The words are: “*Brasil*” (Brazil), “*derivado*” (derived), “*energia*” (energy), “*exportação*” (exporting), “*hipoteses*” (hypothesis), “*Petrobras*”, "*petróleo*" (oil) and "*refinaria*" (refinery). With these keywords, popular search engines may be able to categorize on the fly other documents associated with this subject or suggest some "tag words" in order to refine a user search.

The next bicluster has a residue value of 1.19 and contains seven out of nine documents belonging to category M1 and 1 belonging to M2. The same dominant words found on the previous biclusters were found on this one with the addition of the word "*economia*" (economy). The reason for an M2 document being part of this bicluster was that this document was about "steel refinery" and "rationing of energy", closely related to most documents having "oil" as the main subject.

A bicluster referring to the topic I1 is formed by six out of eleven documents, and it has the following dominant words: "*cadastro*" (filling a form), "*digitar*" (to type), "*golpe*" (scam), "*instalar*" (to install), "*google*", "*orkut*", "*internet*", "*relacionamento*" (relationship), "*mail*", "*roubo*" (robbery), "*malicioso*" (malicious). It is interesting to notice that an intelligent system could present those words to a user, and the user would point out which subject he is looking for, leading to a more refined search.

Other analyzed bicluster refers to six documents belonging to subclass I1 and one document of M1. The "connection words", i.e., the words that connected the two subjects were: "*Brasil*", "programa" (program - can refer to a software or an economic planning), "*público*" (public - as a State service or a free software license), "*serviço*" (service -system service or State service). Here we must point out that the algorithm is sensitive to ambiguous words, which can also become useful as, for example, when a user performs a search on any of these words, an intelligent system may detect that they are ambiguous and ask which of the possible subjects the user is really searching for.

The topic S2 had nine out of its ten texts on a same bicluster connected with the words: "*atleta*" (athlet), "*Brasil*" (Brazil), "*domingo*" (Sunday, the soccer games are usually on Sundays), "*equipe*" (team), "*impulsionamento*" (impulse), "*jogo*" (game), "*segunda*" (second league).

Considering the I2 topic, we could extract significant words as: "*computador*" (computer), "*tecnologia*" (technology), "*software*", "*companhia*" (company), "*milhões*" (millions), "*versão*" (version), "*Brasil*" (Brazil), "*US*" and "*Americana*" (American). The four last dominant words are also connected to two documents on M2 category, as most economy related article refers to them.

Words found as dominant of M2 group on its bicluster are: "*bilhões*" (billions), "*milhões*" (millions), "*importação*" (importing), "*mercadoria*" (products), "*recorde*" (record), "*vendas*" (sells), "*US*" and "*Americana*" (American).

So far, it can be seen from these examples that the BIC-aiNet successfully clusters documents strictly belonging to the same subject, documents that have some information in common, and extracts useful information that can be used by intelligent systems in order to refine a search, recommend other readings, finding ambiguities, relating topics on a newsgroup.

In order to visualize the quality of the generated biclusters, a bicluster grouping documents belonging to M1 category is shown on Fig. 4. Each line represents one document and each column a word, the black spots means that the word is present on a given document, the white spots means the absence of a word.

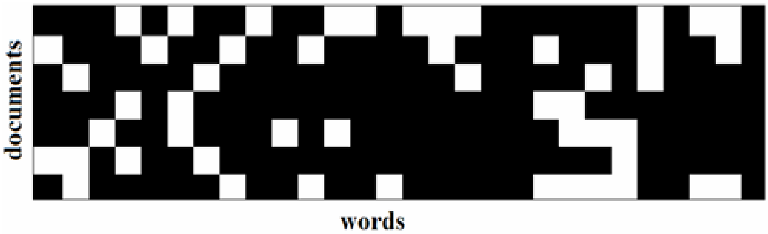

Fig. 4. Example of a bicluster of 7 documents belonging to M1 connected by 28 words

Figure 4 highlights an important aspect of a bicluster algorithm: data compression. In Fig. 5 we can see the original data matrix. As it can be seen, the original matrix is very sparse and of high dimension. Using a biclustering technique, only specific aspects of the whole dataset is taken into account, leading to a more functional clustering outcome.

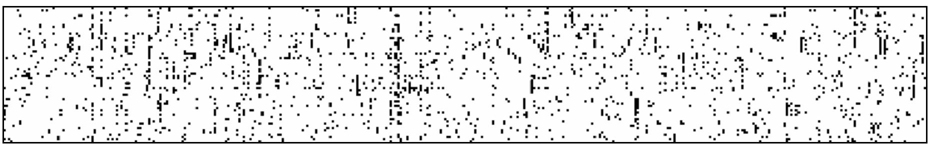

Fig. 5. Original dataset with 1007 words and 60 documents

4.4 Comparative Results

While more rigorous comparisons with other clustering algorithms are still in progress, we present in this subsection the results obtained from a preliminary comparison between BIC-aiNet and the classical K-means clustering algorithm when applied to the same text mining problem. Again, the labels of the texts were not provided to the algorithms. The amount of clusters generated by K-means varied along the experiments. We set K-means to generate 3, 6 and 10 clusters. For all cases, the clusters generated by K-means presented very poor results. By observing the labels of the texts grouped together, we can note that K-means did not separate them efficiently. In all tests, there was one cluster with more than 80% of the texts, while the remaining of the texts was divided into very small clusters.

This outcome suggests that, although the dataset is composed of only 60 documents, it is far from being simple; otherwise, a standard technique such as K-means would have obtained a considerable better performance.

5 Concluding Remarks

This paper introduced a new immune inspired algorithm for biclustering called BIC-aiNet. The proposed biclustering technique is not directly related to the conventional clustering or biclustering paradigm, characterized by the use of the whole dataset, mandatory correlation of all attributes, and the assumption that one item must belong to just one cluster. Instead, we interpret the biclustering problem as multiple bipartition problems. With this flexibility, it is possible to use just some attributes per cluster, making the whole process not only more efficient but also capable of generating information considering several alternative perspectives, possibly useful for classification or decision making in further analysis.

Most texts refer to more than one subject, and "off-topics" happen with some frequency in texts generated by forums or newsgroups. Also some additional information can be extracted from the texts, such as: words that commonly refer to a given subject, words that may refer to more than one subject, how to differentiate ambiguous words, how to classify texts in sub-topics, how to guide a search by using the biclusters to narrow the choices, and so on.

The BIC-aiNet algorithm produces several diverse and high quality solutions simultaneously. Diversity happens when there is little overlap among a set of biclusters, but this overlap is useful when there is several groups that have some features in common. High quality on a bicluster happens when there is little variance on its values (coherent groups).

A dataset of 60 documents extracted from a Brazilian newspaper was generated in order to perform some studies on the information generated by BIC-aiNet. As it was outlined, BIC-aiNet produced coherent clusters and extracted the relevant words belonging to a given subject. BIC-aiNet also correlated different subjects by indicating which words connected them.

As further steps in the research, the experiments will be performed on larger datasets and the information gathered from the experiments will be used to created an intelligent system that will automatically tag a given document and estimate the degree of membership to the subject it was related to, also creating a follow-up list in order to suggest further reading to a user. Also, some post-processing techniques, like variable selection, will be created in order to remove irrelevant words left on the bicluster and eventually include relevant documents left out, making the biclusters even more consistent.

Acknowledgements

The authors would like to thank Universo Online S.A. (UOL – http://www.uol.com.br), CNPq and CAPES for their financial support.

References

1. Agrawal, R., Gehrke, J., Gunopulus, D., Raghavan, P.: Automatic subspace clustering of high dimensional data for data mining applications. In: Proc. of the ACM/SIGMOD Int. Conference on Management of Data, pp. 94–105 (1998)
2. Cheng, Y., Church, G.M.: Biclustering of expression data. In: Proc. of the 8th Int. Conf. on Inteligentt Systems for Molecular Biology, pp. 93–103 (2000)
3. de Castro, L.N, Von Zuben, F.J.: aiNet: An Artificial Immune Network for Data Analysis. In: Data Mining: A Heuristic Approach, pp. 231–259 (2001)
4. de França, F.O., Bezerra, G., Von Zuben, F.J.: New Perspectives for the Biclustering Problem. IEEE Congress on Evolutionary Computation, 2768–2775 (2006)

5. Dhillon, I.S.: Co-clustering documents and words using bipartite spectral graph partitioning. In: Proc. of the 7th Int. Con. on Knowledge Discovery and Data Mining, pp. 269–274 (2001)
6. Feldman, R., Sanger, J.: The Text Mining Handbook. Cambridge University Press, Cambridge (2006)
7. Goldberg, D., Nichols, D., Brian, M., Terry, D.: Using collaborative filtering to weave an information tapestry. ACM Communications 35(12), 61–70 (1992)
8. Haixun, W., Wei, W., Jiong, Y., Yu., P.S.: Clustering by pattern similarity in large data sets. In: Proc. of the 2002 ACM SIGMOD Int. Conf. on Manag. Data, pp. 394–405 (2002)
9. Hartigan, J.A: Direct clustering of a data matrix. Journal of the American Statistical Association (JASA) 67(337), 123–129 (1972)
10. Madeira, S.C., Oliveira, A.L.: Biclustering algorithms for biological data analysis: a survey. IEEE/ACM Trans. on Computational Biology and Bioinformatics 1, 24–25 (2004)
11. Sheng, Q., Moreau, Y., De Moor, B.: Biclustering micrarray data by Gibbs sampling. Bioinformatics 19(suppl. 2), 196–205 (2003)
12. Symeonidis, P., Nanopoulos, A., Papadopoulos, A., Manolopoulos, Y.: Nearest-Biclusters Collaborative Filtering. In: Proc. of the WebKDD 2006 (2006)
13. Tang, C., Zhang, L., Zhang, I., Ramanathan, M.: Interrelated two-way clustering: an unsupervised approach for gene expression data analysis. In: Proc. of the 2nd IEEE Int. Symposium on Bioinformatics and Bioengineering, pp. 41–48 (2001)
14. Tanay, A., Sharan, R., Shamir, R.: Biclustering Algorithms: A Survey. In: Aluru, S. (ed.) Handbook of Computational Molecular Biology. Chapman & Hall/CRC Computer and Information Science Series (2005)