Biclustering in Data Mining using a Memetic Multi-Objective Evolutionary Algorithm

Cristian A. Gallo[†] **Ana G. Maguitman**[‡] **Jessica A. Carballido**[†] **Ignacio Ponzoni**[†↕]

[†]Laboratorio de Investigación y Desarrollo en Computación Científica (LIDeCC)
[‡]Laboratorio de Investigación y Desarrollo en Inteligencia Artificial (LIDIA)
Universidad Nacional del Sur, Av. Alem 1253, Bahía Blanca, 8000, Argentina
[↕]Planta Piloto de Ingeniería Química (PLAPIQUI) – UNS – CONICET Complejo CRIBAB,
Co. La Carrindanga km. 7, CC 717, Bahía Blanca, Argentina
{cag, agm, jac, ip}@cs.uns.edu.ar

Abstract

In this paper, a new memetic strategy that integrates a multi-objective evolutionary algorithm (the SPEA2) with a local search technique for data mining is presented. The algorithm explores a Term Frequency-Inverse Document Frequency (TF-IDF) data matrix in order to find biclusters that fulfill several objectives. The case of study was a dataset corresponding to the *Reuters-21578 corpus*. Our algorithm performed satisfactorily, finding biclusters that have large size and coherent values, yielding to undeniably promising outcomes. Nonetheless, more experiments with data from other corpus are necessary, thus leading to more concluding results.

Keywords: data mining, biclustering, evolutionary algorithms

1 INTRODUCTION

Clustering is a division of data into groups of similar objects. Each group, called cluster, consists of objects that are similar between themselves and dissimilar to objects of other groups. Representing the data by fewer clusters necessarily loses certain fine details, but achieves simplification. In other words, the data are modeled by the corresponding clusters. Data modeling puts clustering in a historical perspective rooted in mathematics, statistics, and numerical analysis. From a machine learning perspective, clusters correspond to hidden patterns. In particular, the search for clusters constitutes an especial case of unsupervised learning, and the resulting system represents a data concept. Therefore, clustering is an unsupervised learning approach to discover a hidden data concept. From a practical perspective clustering plays an outstanding role in data mining applications such as scientific data exploration, information retrieval and text mining, spatial database applications, Web analysis, CRM, marketing, medical diagnostics, computational biology, and many others. Data mining deals with large databases that impose additional severe computational requirements on clustering analysis.

A more interesting general idea is to produce attribute groups in conjunction with clustering of points themselves. This approach leads to the concept of biclustering, which constitutes the simultaneous clustering of both points and their attributes. This approach reverses the struggle: to improve clustering of points based on their attributes, it tries to cluster attributes based on the points. The idea of applying biclustering on data points and attributes has several antecedents [1,2] and is known under the names co-clustering, simultaneous clustering, bi-dimensional clustering, block clustering, conjugate clustering, distributional clustering, and information bottleneck method. In this work, the points represent documents and the attributes represent terms. The level of expression of terms under a set of documents can be arranged into a matrix, namely a TF-IDF data matrix (M_{DT}), where rows and columns correspond to documents and terms respectively. Each matrix entry e_{dt} is a real value that indicates the level of occurrence of a term in a particular document. In this context, a satisfactory *bicluster* consists in a group of rows and columns of the M_{DT} matrix that satisfies some similarity score in conjunction with some other criteria.

As we will discuss in the following sections, the elevated complexity of this task has motivated the development of several approximation methods to generate near optimal solutions. In particular, in this paper, a memetic multiobjective evolutionary algorithm called BicMinig is proposed. The new algorithm hybridizes the SPEA2 [3] with a new variant of the local search algorithm proposed by Cheng & Church [4]. For each potential bicluster, BicMinig evaluates the following objectives: the mean squared residue proposed by Cheng and Church [4] as a similarity measure for biclusters; the mean; and the dimension (number of rows and columns). Also it considers maximizing the coverage of documents on the resulting biclusters. In order to assess the performance of our approach, a comparative study between BicMinig and a human categorized data set is presented, using as case studies the *Reuters-21578 corpus* [5].

The paper is organized as follows: in the Section 2 some basic definitions about biclustering are introduced; next the core algorithms used to build the memetic approach are described; later the main features of BicMinig are presented; and finally, in Sections 5 and 6, the experimental results and conclusions are discussed.

2 BICLUSTERING

In the context of this work, a bicluster is defined as a pair (G, C) where $G \subseteq \{1,\ldots,|D|\}$ is a subset of documents (rows) and $C \subseteq \{1,\ldots,|T|\}$ is a subset of terms (columns) [4]. The main goal is to find the largest biclusters that have homogeneous values. It is also important to consider that the mean of the

bicluster should be relatively high, in order to capture terms exhibiting relevant values under some set of documents. The bicluster size is the number of rows *f*(G) and the number of columns *g(C).* The homogeneity *h(G,C)* is given by the mean squared residue score [4], while *k(G, C)* is the mean of all values in the bicluster. Therefore, our optimization problem can be defined as follows:

maximize

$$f(G) = |G| \tag{1}$$

$$g(C) = |C| \tag{2}$$

$$k(G,C) = e_{GC} \tag{3}$$

minimizing

$$h(G,C) \tag{4}$$

with $(G,C) \in X$, $X = 2^{\{1,\ldots,|D|\}} \times 2^{\{1,\ldots,|T|\}}$ being the set of all biclusters, where

$$h(G,C) = \frac{1}{|G| \cdot |C|} \sum_{g \in G, c \in C} \left(e_{gc} - e_{gC} - e_{Gc} + e_{GC}\right)^2 \tag{5}$$

is the mean squared residue score,

$$e_{gC} = \frac{1}{|C|} \sum_{c \in C} e_{gc}, \quad e_{Gc} = \frac{1}{|G|} \sum_{g \in G} e_{gc} \tag{6,7}$$

are the mean column and row expression values of (G,C) and

$$e_{GC} = \frac{1}{|G| \cdot |C|} \sum_{g \in G, c \in C} e_{gc} \tag{8}$$

is the mean expression value over all the cells that are contained in the bicluster (G,C). The residue quantifies the difference between the actual value of an element e_{gc} and its expected value as predicted for the corresponding row mean, column mean, and bicluster mean. If a bicluster has a lower mean square residue it means that the bicluster exhibits coherent values. When we try to find a set of biclusters on data mining, there is a meta-objective to consider: the coverage of the documents. That is to say, maximize the documents coverage over all the resulting biclusters. The problem of finding the largest bicluster with the lowest mean squared residue is NP-hard [4]. The high complexity of this problem has motivated the development of heuristic techniques to generate near optimal solutions. In particular, evolutionary algorithms are well suited for addressing this class of problems [6, 7, 8, 9].

3 ALGORITHM FRAMEWORK

The main idea is to use the evolutionary algorithm to explore the search space *X*. However, stand-alone evolutionary algorithms may not find good solutions satisfying the homogeneity constraint [9]. In such situations, local search strategies can be employed to improve the performance of the MOEA. In this section, we present the core algorithms used in our approach, given place to BicMining.

3.1 Strength Pareto Evolutionary Algorithm (SPEA2)

A brief report of the SPEA2 used to globally explore the search space *X* is introduced here. For a more in-depth study of this method we refer the reader to [3]. A definition of the concept of Pareto non-domination [10] is given next, trailed by the overall algorithm.

Definition 1. If there are M objective functions, a solution x^1 is said to weakly dominate another solution x^2, if both conditions (a) and (b) are true:
(a) The solution x^1 is no worse than x^2 in all the M objective functions.
(b) The solution x^1 is strictly better than x^2 in at least one of the M objective functions.

If x^1 does not weakly dominate x^2 and x^2 does not weakly dominate x^1 the two solutions are non-dominating to each other. When a solution i weakly dominates a solution j, we denote $i \preceq j$. If $\neg\exists\ i$: $i \preceq j$ then j is called a Pareto-optimal solution. Then the set of all Pareto-optimal solutions is called the Pareto-optimal front. The aim of a MOEA is to approximate the Pareto-optimal front.

Algorithm 1 (SPEA2 Main Loop)
Input: N *(population size),*
$\overline{N}$ *(archive size),*
T *(maximum number of generations)*
Output: A *(non-dominated set)*

Step 1: *Initialization: Generate an initial population* P_0 *and create the empty archive (external set)* $\overline{P}_0 = \phi$. *Set* $t = 0$.

Step 2: *Fitness assignment: Calculate fitness values of individuals in* P_t *and* $\overline{P}_t$ *(cf. Section 3.1).*

Step 3: *Environmental selection: Copy all non-dominated individuals in* P_t *and* $\overline{P}_t$ *to* $\overline{P}_{t+1}$. *If size of* $\overline{P}_{t+1}$ *exceeds* $\overline{N}$ *then reduce* $\overline{P}_{t+1}$ *by means of the truncation operator, otherwise if size of* $\overline{P}_{t+1}$ *is less than* $\overline{N}$ *then fill* $\overline{P}_{t+1}$ *with dominated individuals in* P_t *and* $\overline{P}_t$ *(cf. Section 3.2).*

Step 4: *Stopping: If* $t \geq T$ *or another stopping criterion is satisfied then set* A *to the set of decision vectors represented by the non-dominated individuals in* $\overline{P}_{t+1}$. *Stop.*

Step 5: *Mating selection: Perform binary tournament selection with replacement on* $\overline{P}_{t+1}$ *in order to fill the mating pool.*

Step 6: *Variation: Apply recombination and mutation to the mating pool and set* $\overline{P}_{t+1}$ *to the resulting population. Increment generation counter and go to Step 2.*

3.1.1 Fitness Assignment

Each individual i in the archive $\overline{P}_t$ and the population P_t is assigned a strength value $S(i)$, representing the number of solutions it dominates:

$$S(i) = \left| \left\{ j \mid j \in P_t + \overline{P}_t \wedge i \succ j \right\} \right| \tag{9}$$

Using values of equation 9, the raw fitness $R(i)$ of an individual i is calculated:

$$R(i) = \sum_{j \in P_t + \overline{P}_t,\, j \succ i} S(j) \tag{10}$$

The raw fitness is determined by the strengths of its dominators in the archive and in the population. Note that the fitness must be minimized. In order to be more accurate, additional density information is incorporated to discriminate between individuals having identical raw fitness values. The density estimation technique is based on the following idea: for each individual i, the distances (in the objective space) to all individuals j in the archive and in the population are calculated and stored in a list. After sorting the list in increasing order, the k^{th} element gives the distance sought, denoted as σ_i^k. We chose k=1 because it is often sufficient and leads to a more efficient implementation [11]. Then, the density $D(i)$ corresponding to i is defined by:

$$D(i) = \frac{1}{\sigma_i^k + 2} \tag{11}$$

Finally, the fitness function *F(i)=R(i)+D(i)* can be obtained from equations 10 and 11.

3.1.2 Environmental Selection

During environmental selection, the first step is to copy all the non-dominated individuals from the archive and from the population to the archive of the next generation ($\overline{P}_{t+1}$). If the non-dominated front exactly fits into the archive, the environmental selection step in completed. If the archive is too small, the best dominated individuals in the previous archive and population are copied to the new archive. Otherwise (when the archive is too large) an archive truncation procedure is invoked which iteratively removes individuals from $\overline{P}_{t+1}$ to decrease its size to $\overline{N}$. At each repetition, an individual *i* is chosen for removal, for which $i \leq_d j$ for all $j \in \overline{P}_{t+1}$ with:

$$i \leq_d j \quad \Leftrightarrow \quad \forall \ \ 0 < k < \left|\overline{P}_{t+1}\right| : \sigma_i^k = \sigma_j^k \vee \\ \exists \ \ 0 < k < \left|\overline{P}_{t+1}\right| : \left[\left(\forall 0 < l < k : \sigma_i^l = \sigma_j^l \right) \wedge \sigma_i^k < \sigma_j^k \right] \tag{12}$$

where σ_i^k denotes the distance of *i* to its k-th nearest neighbor in $\overline{P}_{t+1}$.

3.2 Local Search Procedure

This subsection describes the local search procedure that hybridizes the SPEA2 giving place to BicMining. This greedy approach is based on [4], with some modifications introduced in order to consider the overall efficiency of the proposal. The algorithm starts from a given bicluster (G, C). The documents or terms having mean squared residue above (or below) a certain threshold are selectively eliminated (or added) according to the following algorithm:

Algorithm 2 (Local Search)
Input: (G, C) *(a bicluster)*
Output: (G, C)' *(an improved bicluster)*

Step 1: *Compute* e_{gC}, e_{Gc}, e_{GC} *and h(G, C) by equations 5-8.*

Step 2: *Remove all documents* $g \in G$ *satisfying:* $\frac{1}{C}\sum_{c \in C}\left(e_{gc} - e_{gC} - e_{Gc} + e_{GC}\right)^2 > \alpha \cdot h(G,C)$

Recalculate all means and perform the same operation on terms. The equation for terms is analogous.

Step 3: *Recompute* e_{gC}, e_{Gc}, e_{GC} *and h(G, C).*

Step 4: *Add all terms* $c \notin C$ *satisfying:* $\frac{1}{G}\sum_{g \in G}\left(e_{gc} - e_{gC} - e_{Gc} + e_{GC}\right)^2 \leq h(G,C)$

Recalculate all means and perform the same operation on documents. The equation for documents is analogous.

The main difference with the implementation in [4] is that here, given the high dimensionality of the data matrix, we do not apply single node deletion for efficiency reasons. The parameter α determines how often multiple documents (or terms) deletion is used. A higher α leads to fewer multiple documents (or terms) deletions and thus, in general, reduces the coherence of the biclusters. On the other hand, if we chose a lower α, the coherence is increased at the cost of reducing the size.

4 BicMining: A HYBRID MOEA FOR BICLUSTERING

The aim of our study is to use the SPEA2 for approximating the Pareto front of biclusters from a given TF-IDF matrix, as this approach gives the best tradeoff between the objectives that we want to optimize. However, in view of the fact that the Pareto front also includes biclusters that are not homogeneous at all, we needed to guide the search to the area where this restriction is accomplished. In that context, we applied the aforementioned local search method to the initial and resulting population of the SPEA2, thus guiding the search of the MOEA by refining the chromosomes. This local search is applied in that way for efficiency reasons because this is the procedure that consumes a higher amount of CPU usage.

4.1 Representation

Each individual of the evolutionary algorithm represents one bicluster, which is encoded by a fixed size string built by appending a binary string for documents with a binary string for terms. The individual corresponds to a solution for the problem of optimal bicluster generation. If a bit is set to 1, it means that the relative row or column belongs to the encoded bicluster, otherwise it does not. Fig. 1 shows the encoding of documents and terms for a random individual.

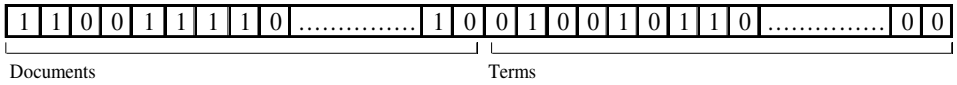

Figure 1: An encoded individual representing a bicluster

4.2 Crossover and Mutation Operators

It is important to give a brief description of the genetic operators used in this approach, since they have a key influence in how the search is performed.

Mutation. This operator is implemented in the following way: first it is determined if the individual needs to be mutated by means of the probability value assigned to the operator. In such case, a position of the binary string is selected at random, and then the corresponding bit is simply complemented. Otherwise the individual stays with no changes.

Recombination. We chose a two-point crossover with a little restriction: one random point is selected on the rows and the other random point is selected on the columns. In this way, we ensure that the recombination is performed over both the documents' and the terms' subspaces. Then, when both children are obtained by combining each one of the two parents' parts (i.e. the ends and the center), the individual that is selected to be the only descendant is the non-dominated one. If both are non-dominated, one of them is chosen at random.

4.3 Multi-Objective Fitness

As regards the objectives to be optimized, we observed that it was necessary to generate maximal sets of documents and terms while maintaining the "homogeneity" of the bicluster with a relatively high mean value. These bicluster features, conflicting to each other, are well suited for multiobjective modeling. In that context, we decided to optimize the objectives defined by equations 1-4: the quantity of documents, the quantity of terms, the mean, and the mean squared residue. The first three objectives are maximized while the last one is minimized.

4.4 Initial Population

Considering the meta-objective of maximizing the coverage of documents in the resulting population, the initial population is generated at random embracing the entirety of the documents (without

repetition) in the data matrix. In other words, each individual contains a set of documents where each set is disjoint among the others. Besides that, there is another question to take into account. The low density of relevant entries in the data matrix provokes an important quantity of biclusters with null values (zeros). This issue is solved by allowing that each generated bicluster contains at most a certain proportion of null values.

4.5 Non-feasible Individuals

Although the used representation is perfect, that is to say, all the representable individuals belong to the domain of the problem and vice versa, the low density of relevant data in the matrix makes it possible to generate individuals with great quantity of null values. This question is addressed by adding two mechanisms. The first one corresponds to generated relevant individuals in the initial population as we have mentioned earlier. The other mechanism is used to correct the individuals that are generated by the application of the genetic operators, since there are many cases (mainly in the case of the crossover) where non-feasible individuals are generated. The following procedure is applied to each new individual:

Algorithm 3 (Bicluster Corrector)
Input: (G, C) (a bicluster)
δ (proportion of relevant values)
Output: (G, C)' (a relevant bicluster)

Step 1: *Remove all documents* $g \in G$ *that contain* $\lceil |G| \cdot (1-\delta) \rceil$ *null values in the bicluster.*

Step 2: *Remove all terms* $c \in C$ *that contain* $\lceil |C| \cdot (1-\delta) \rceil$ *null values in the bicluster.*

The δ value is set to 0 in the initial generation. Then it is increased by *1 / (max number of generations)* after each generation. This proposal gives the possibility, at the beginning of the evolutionary process, of using the genetic information of the non-feasible individuals to build useful individuals. As the evolutionary process progresses, this restriction is increased by guiding the search on those areas in the data matrix with non-zero values, obtaining at the end a population where the individuals have almost all relevant data.

4.6 Decision Maker

As regards the resulting population, it is necessary to establish a criterion on the kind of individuals we are more interested in. In view of the fact that our approach is a multi-objective method based on the concept of Pareto dominance, a trivial answer is to keep all of the non-dominated individuals. However, one of the disadvantages of applying evolutionary computation on biclustering is the overlap on the final population. In such a case, keeping the non-dominated individuals is not enough. We propose the following naïve procedure that selects individuals from the resulting non-dominated set and gives a good coverage of documents considering all of the selected biclusters:

Algorithm 4 (Decision Maker)
Input: *A* *(a set of non-dominated individuals)*
Output: *A'* *(a set of the best individuals)*

Step 1: *Let Mx be the set of all non-dominated individuals in A.*

Step 2: *Set Max, Min and Mn to* ϕ.

Step 3: *Let b be the bicluster with more quantity of documents on Mx. Remove b of Mx.*

Step 4: *If b has a 50% of the documents that are not found in the biclusters of Max then add b to Max. Otherwise add b to Mn.*

Step 5: *If $Mx \neq \phi$ go to Step 3.*

Step 6: *Let b be the bicluster with less quantity of documents on Mn. Remove b of Mn.*

Step 7: *If b has a 50% of documents that are not found in the biclusters of Min then add b to Min.*

Step 8: *If $Mn \neq \phi$ go to Step 6.*

Step 9: *Set A' to $Max \cup Min$.*

5 EXPERIMENTAL FRAMEWORK AND RESULTS

A comparison between BicMining and a text categorization test collection is presented here. For this analysis, we have used the Reuters-21578 *corpus* [5].

A test collection for text categorization contains, at minimum, a set of texts and, for each text, a specification of what categories that text belongs to. For the Reuters-21578 collection the documents are Reuters newswire stories, and the categories are five different sets of content related categories. For each document, a human indexer decided which categories from which sets that document belonged to. The category sets are as follows:

Table 1. Category sets for *Reuters-21578 corpus*

Category Set	Number of Categories	Number of Categories w/ 1+ Occurrences	Number of Categories w/ 20+ Occurrences
EXCHANGES	39	32	7
ORGS	56	32	9
PEOPLE	267	114	15
PLACES	175	147	60
TOPICS	135	120	57

5.1 Data Preparation

In Table 1 we note how many categories appear in at least 1 of the 21578 documents in the collection, and how many appear in no documents. The only categories that are of our interest are those that possess at least 2 documents, since no bicluster will have less than 2 documents because of the impossibility of computing the coherence measure. Therefore we have determined that the total number of categories with more than 2 documents is 341, and based on this, the comparison is carried out.

Concerning the construction of the TF-IDF matrix, we have processed all the terms found in the 21578 documents and systematically eliminated stopwords, applied the Porter's algorithm [12] and eliminated those terms that appear alone and those that appear in at least 80% of the documents. No terms that occur alone will be a part of a bicluster and those that appear in at least 80% of the documents are presumably noise. The number of terms obtained after applying the previous process was 17539, achieving a significant reduction if we compare them with the 42671 terms without stopwords of the original corpus.

5.2 Parameters Setup

As regards of the parameter settings, we have determined the setup values which yielded the best results in a few preliminary runs. Table 2 shows the parameters used for this study. Note that the sizes of the population and archives are 5% and 10% of the total number of documents, respectively. This information is useful when tests over other corpus are performed. Such kinds of sizes are needed to achieve the meta-objective of maximizing the coverage of documents. The local search parameter α is set to 1.5 as a tradeoff between size and coherence. And finally, the number of generations is small for efficiency reasons.

Table 2. Default parameter setting for this study.

SPEA2				Local Search
PopSize	Archive Size	N° of Gens.	Mutation Prob.	α
1000	2000	20	0.3	1,5

5.3 Qualitative Evaluation

In order to compare the artificial categories found by our algorithm with respect to the categories of the *Reuters-21578 corpus*, we propose the use of several ad-hoc metrics derived from [13]. Next, we will present the *Accuracy* and the *Coverage Metrics*.

Accuracy Metric: this measure can establish how accurate (on average) is a cluster in approximating a set of clusters A with respect to another cluster of a set of clusters B. It is computed as an average of the maximum match of documents of each element in A with respect to the elements on B. More formally, let $a \in A$ be a cluster and $b_a \in B$ the cluster that maximizes the documents shared with a. The precision of the cluster a can be established as follows:

$$p(a, b_a) = \left|a \cap b_a\right| / \left|a\right| \tag{13}$$

Then we define the *Accuracy* $\mathcal{A}(A, B)$ as an average of the equation 13 over all of the clusters in A:

$$\mathcal{A}(A, B) = \frac{1}{|A|} \sum_{a \in A, b_a \in B} p(a, b_a) \tag{14}$$

If $\mathcal{A}(A, B) = 1$, all the documents of each cluster in A belongs to a unique cluster in B. On the other hand, if $\mathcal{A}(A, B) = 0$, the clusters in A are disjoint with the clusters in B. Thus, to a greater value of $\mathcal{A}(A, B)$ a more accurate approximation is achieved by the clusters of A in modeling the clusters in B.

Coverage Metric: this metric analyzes the proportion (on average) of the documents of each cluster in B that is covered by each cluster in A. More formally, let $a \in A$ be a cluster and $b_a \in B$ the cluster that maximizes the documents shared with a. The coverage of the cluster a can be established as follows:

$$c(a, b_a) = \left|a \cap b_a\right| / \left|b_a\right| \tag{15}$$

Then we define the *Coverage* $\mathcal{C}(A, B)$ as an average of the equation 15 over all of the clusters in A:

$$\mathcal{C}(A, B) = \frac{1}{|A|} \sum_{a \in A, b_a \in B} c(a, b_a) \tag{16}$$

If $\mathcal{C}(A, B) = 1$, all the documents of each cluster in B are covered by a cluster in A. On the other hand, if $\mathcal{C}(A, B) = 0$, the clusters in A are disjoint with the clusters of B. Thus, to a greater value of $\mathcal{C}(A, B)$ more completeness is achieved by the clusters of A in modeling the clusters in B.

It is important to note that the previous two metrics assess the performance of a set of clusters A with respect to a set of clusters B locally, i.e., they only consider what happens in a cluster of A with respect to another cluster of B and then compute the average. This is insufficient since it does not show how precise (or complete) are on the whole all the clusters of A with respect to all of the clusters of B. To determine it we construct in the following section an approximation of a confusion matrix that shows (graphically) how precise and/or complete is the set of biclusters obtained by our approach.

5.4 Comparative Study

With respect to the experimental results, the *Reuters-21578 corpus* contains 341 clusters with at least 2 documents that we have selected to do the comparison. The biclusters that we chose for the evaluation

are those that are returned by the Decision Maker.

In table 3, the results of evaluating the metrics for the biclusters obtained by BicMining in 10 runs with respect to the clusters obtained from the *Reuters-21578 corpus* are presented. To establish the quality of the biclusters, the average on the size, the mean squared residue (MSR) and the mean on each execution are also shown. The table 4 shows several measures of interest for the 341 clusters of the *Reuters-21578 corpus* and the average over the 10 runs performed by BicMining.

Table 3. Values of the metrics for 10 runs of BicMining against the Reuters clusters.

		Average results					
run	Number of Biclusters	Number of documents	Number of terms	MSR	Mean	Accuracy	Coverage
1	102	264,49	19,3725	0,352051	1,87532	0,456642	0,31552
2	107	264,888	34,1682	0,224494	2,09618	0,483963	0,32068
3	125	208,304	21,744	0,31656	1,92236	0,43035	0,368234
4	106	259,906	20,1415	0,173474	2,05657	0,462544	0,282738
5	91	270,143	35,1538	0,206942	2,06897	0,454354	0,239066
6	121	154,727	29,4711	0,468243	2,07219	0,449041	0,260431
7	137	251,876	18,5401	0,315828	1,95197	0,447999	0,27883
8	96	261,531	18,0625	0,190044	2,10424	0,49555	0,318363
9	100	246,37	21,55	0,322	1,93951	0,445265	0,276199
10	105	297,819	27,4571	0,227147	1,99245	0,459906	0,327463
avr	109	248,0054	24,56608	0,2796783	2,007976	0,4585614	0,2987524

Table 4. Several measures between the average results of BicMining and Reuters Clusters.

	BicMining	Reuters 21578
Clusters	109	341
Coverage of documents of the corpus	44,91%	91,34%
Average of Documents	248,0054	95,6452
Max documents on a clusters	3923	11330
Overlapping	1,791765	1,59809

As we can observe, the memetic MOEA performs well finding homogeneous biclusters that are great in size with a relatively high mean value. This shows that the terms that are relevant belong to the biclusters and occur similarly among the documents of each one. Therefore we are in conditions of affirming that our approach finds artificial categorizations for the covered documents of the *Reuters-21578 corpus*. However, not all of the documents were covered by our algorithm. This can be solved running again the proposal over the documents that remain by creating the initial population on those documents.

Finally, we need to determine if the biclusters found by our method are in correspondence with the clusters of Reuters and vice versa. As we can see, the accuracy of an individual bicluster is acceptable (over the 45% of documents) with respect to the human categorization of the *Reuters-21578 corpus*, whereas the coverage is around the 30% of the documents. This is an expected result since it is impossible to determine which set of objectives in an algorithm gives a categorization like a human indexer. So as to illustrate these results, an ad-hoc approximation of a confusion matrix [13] is presented here. We compare the best result found over the 10 executions of BicMining and the clusters of Reuters. The confusion matrices are built as follows: for precision, the rows represent biclusters and the columns represent the clusters of Reuters; for recall, the rows represent the clusters of Reuters and the columns represent biclusters. Each entry a_{ij} in the matrices represents the documents proportion that the cluster in the column j shares with the cluster of the row i with regard to the total number of documents of that row. The figures 2 and 3 in gray scale represent the matrices for precision and recall respectively according to the previous definition. Each pixel is an entry in the matrix where the black color means that the maximum value corresponding to that row is

reached in that entry. The rest of the colors show the proportion of documents in relation to that maximum, being the pixel of white color when the proportion is 0. To assert a good performance the figures would have to be approximately a transverse line.

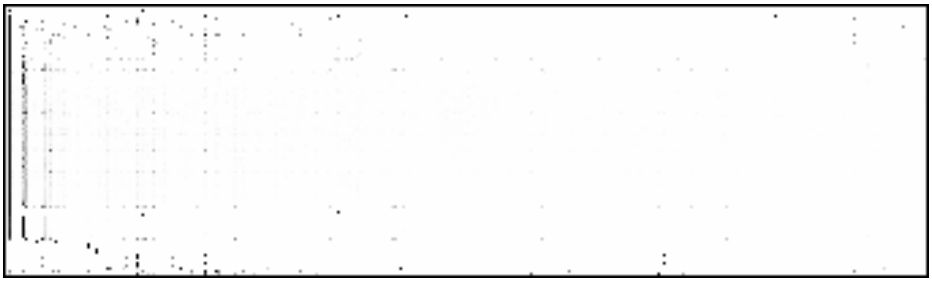

Figure 2: Approximation of a confusion matrix for precision, where the rows represent biclusters and the columns represent the Reuters clusters.

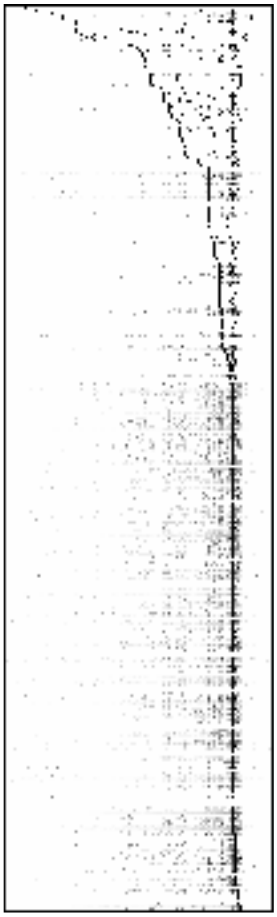

Figure 3: Approximation of a confusion matrix for recall, where the rows represent the Reuters clusters and the columns represent biclusters.

As we can see, the figure 2 shows that the biclusters found by our approach (considering the entire set) are far from corresponding with the Reuters clusters. However, a slightly better performance is shown by the figure 3. The completeness seems to be more a transverse line, although enough dispersion is also observed. A curious result is that when we consider the individual biclusters (see metrics for *Accuracy*

and *Coverage*), a better performance is reached for precision than for recall. On the other hand when we consider the entire set of biclusters, the results are the opposite.

All the testing has been made on an *Athlon X2* processor with 2 GB of RAM. The running time for one execution of our approach (on average) was of 2100s. We argue that running times of half an hour are well acceptable, especially in comparison with the effort, time and economical costs needed to perform the categorization manually.

6 CONCLUSIONS

In this paper we have introduced BicMining, a novel multi-objective framework for biclustering TF-IDF data hybridized with a local search procedure. For one of the most relevant datasets in the literature, the *Reuters-21578 corpus,* we have demonstrated that the quality of the artificial categories found by our approach (considering the objective proposed) are very good, showing high levels of coherence on relevant terms with considerable sizes.

However, these biclusters are far from being in correspondance with the human categorization of the *Reuters-21578 corpus*. In fact, this is an expected result since our method is an unsupervised leaning process that will perform well over any TF-IDF matrix. Nevertheless, this last asseveration needs to be confirmed with more experiments over data from other corpus.

REFERENCES

[1] Anderberg, M. Cluster Analysis and Applications. Academic Press, New York, 1973.

[2] Hartigan, J. Clustering Algorithms. John Wiley & Sons, New York, NY, 1975.

[3] Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In Giannakoglou, Tsahalis, Periaux, Papailiou, and Fogarty (eds), Evolutionary Methods for Design, Optimisations, and Control, pp.19-26, 2002.

[4] Cheng, Y., Church, G.M.: Biclustering of Expression Data. Proceedings of the 8th International Conf. on Intelligent Systems for Molecular Biology, La Jolla, USA, pp. 93-103, 2000.

[5] Lewis, D.: Reuters-21578 text Categorization test collection. Distribution 1.0. README file (version 1.2). Manuscript, September 1997. http://www.daviddlewis.com/resources/testcollections/reuters21578/readme.txt.

[6] Madeira, S., Oliveira, A.L.: Biclustering Algorithms for Biological Data Analysis: A Survey. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 1:24-45, 2004

[7] Mitra, S., Banka, H.: Multi-objective evolutionary biclustering of gene expression data. Pattern Recognition, 39:2464–2477, 2006.

[8] Divina, F., Aguilar-Ruiz, J.S.: Biclustering of Expression Data with Evolutionary Computation. IEEE Transactions on Knowledge and Data Engineering, 18(5):590- 602, 2006.

[9] Bleuler, S., Prelic, A., Zitzler, E.: An EA framework for biclustering of gene expression data, in: Proceeding of Congress on Evolutionary Computation, 1:166-173, 2004.

[10] Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons, Inc., New York, NY, 2001.

[11] Zitzler, E., Laumanns, M., Bleuler, S.: A Tutorial on Evolutionary Multiobjective Optimization. In X. Gandibleux and others, editors, Metaheuristics for Multiobjective Optimisation, Lecture Notes in Economics and Mathematical Systems, Springer, 2004.

[12] Porter, M.: An algorithm for suffix stripping. Readings in information retrieval, Morgan Kaufmann Publishers Inc., 313-316, 1997.

[13] Chakrabarti, S.: Mining the Web: Discovering Knowledge from Hypertext Data. Morgan Kauffman Publishers, Inc., 2002.