Semi-supervised Clustering of Medical Text

Pracheta Sahoo
Dept. of Computer Science
University of Texas at Dallas
pracheta.sahoo@utdallas.edu

Asif Ekbal and **Sriparna Saha**
Dept. of Computer Science
and Engineering
IIT Patna, Bihta
asif,sriparna@iitp.ac.in

Diego Mollá
Dept. of Computing
Macquarie University
Sydney, Australia
diego.molla-aliod@mq.edu.au

Kaushik Nandan
Dept. of Computer Science
and Engineering
IIT Patna, Bihta
kaushalta@gmail.com

Abstract

Semi-supervised clustering is an attractive alternative for traditional (unsupervised) clustering in targeted applications. By using the information of a small annotated dataset, semi-supervised clustering can produce clusters that are customized to the application domain. In this paper, we present a semi-supervised clustering technique based on a multi-objective evolutionary algorithm (NSGA-II-clus). We apply this technique to the task of clustering medical publications for Evidence Based Medicine (EBM) and observe an improvement of the results against unsupervised and other semi-supervised clustering techniques.

1 Introduction

Clustering is an unsupervised machine learning method that attempts to find groups (clusters) in a collection of documents (Jain et al., 1999). Clustering is useful for applications where the goal is to find structure in a collection of documents, and can be applied in a wide range of tasks, such as finding groups among patients with breast cancer, or identifying groups of shoppers with similar browsing and purchase histories. A common problem with clustering, however, is that the structure that is found might not reflect the desired structure that is relevant for a particular application. For example, one might wish to cluster words in the hope of learning their parts-of-speech, but instead the clusters group words according to their meanings. In supervised learning, we have labeled information, but the annotation can be costly to produce (Zhu and Goldberg, 2009). So a trade-off is needed, and a semi-supervised framework provides this trade-off. In semi-supervised clustering (Zhu and Goldberg, 2009), part of the documents to cluster are annotated with information about how they cluster, and the task consists of clustering the entire set of documents. By incorporating the information of the known clusters of a part of the documents, the final clusters have a better chance to match the desired clusters of the application domain. In this paper we focus on clustering the documents that are relevant to a clinical query for the practice of Evidence Based Medicine (EBM) (Shash and Molla, 2013). Here, each cluster is expected to group the documents that describe a particular aspect of the answer to the clinical question. Let us take an example of the disease Asperger's syndrome. There are five policies for the treatment of this disease, namely 'special education', 'behavior modification', 'speech', 'physical and occupational therapy and medication', and 'social skill therapies and medications'. Now the documents which are assigned to each of these possible treatment policies represent a cluster. Table 1 shows an example of such clustering. Moreover, Table 1 shows that some documents may be associated with multiple treatments, and therefore the clustering task is non-overlapping.

Most of the clustering techniques in existing literature focus on optimizing only one validity index (Jain et al., 1999; Maulik and Bandyopadhyay, 2002), which measures the goodness of an obtained par-

Proceedings of the Clinical Natural Language Processing Workshop,
pages 23–31, Osaka, Japan, December 11-17 2016.

Table 1: Asperger's syndrome and its possible treatments. Each item enclosed in [] indicates a text document ID.

Which treatments work best for Asperger's syndrome?

Name of the treatment	IDs of documents assigned
Special education	[1080], [1178]
Behavior modification	[8545], [4123], [5523]
Speech, physical or occupational therapy	[1080], [8545]
Social skills therapies	[5523], [3321], [6434]
Medications	[8545], [3321], [6434], [6755]

titioning. However, in order to determine a proper partitioning, optimizing a single cluster validity index is not always sufficient, especially in the situation when we deal with text documents having clusters of different shapes and sizes. The concept of multi-objective optimization (MOO) can be brought into consideration where we need to optimize several objective functions at the same time. The advantage of MOO is that we can generate clusters by optimizing several cluster validity indices. Inspired by this, Ekbal et al. (2013) proposed a MOO-based approach for clustering medical documents for EBM by using the search capability of a simulated annealing based approach, AMOSA (Archived MultiObjective Simulated Annealing based technique) (Bandyopadhyay et al., 2008). However, it has been shown that for some benchmark datasets, AMOSA performs slowly compared to a popular genetic algorithm based MOO technique, NSGA-II (Non-dominated Sorting Genetic Algorithm-II) (Bandyopadhyay et al., 2008). Therefore, an alternative MOO-based approach is needed in order to verify whether we can improve the run-time complexity of AMOSA. Moreover, Ekbal et al. (2013) have used some labeled information to select a single solution from the final set of trade-off solutions. In general semi-supervised methods perform well compared to unsupervised clustering techniques.

In our present work we propose to develop a semi-supervised clustering technique and apply that for EBM. The proposed approach uses only 10% labeled information which is easy to obtain. The proposed technique is novel in a way that it uses the labeled information during the internal steps of the proposed clustering process. More specifically we can say that the internal steps of NSGA-II based clustering are modified to take care of this labeled information. The labeled information was used by Ekbal et al. (2013) to select a single solution from the final Pareto optimal front after the execution of AMOSA based clustering technique. Moreover, as mentioned by Bandyopadhyay et al. (2008), the complexity of AMOSA is higher than that of NSGA-II. Thus, the use of NSGA-II as the underlying optimization technique makes the system less complex and time consuming. In this paper, we propose the use of NSGA-II (Deb et al., 2013) for semi-supervised clustering of documents. We propose two different versions of the NSGA-II based semi-supervised clustering technique. In the first approach the available supervised information in the form of must-link and cannot-link constraints can be used during the selection phase of clustering. These constraints are taken into account while calculating crowding distance which is further used to assign ranks to different solutions of the combined population. Thus, the available supervised information is used in each generation of the proposed technique. In the second approach, we use a semi-supervised approach to select a single solution from the set of final solutions produced by the MOO-based approach. In this case, supervised information is used only at the final stage rather than during the clustering phase. In recent years, several semi-supervised clustering techniques (Xing et al., 2002; Basu et al., 2004) have been proposed in the literature which are applicable for general data sets. In this paper we also extend those techniques to solve the problem of EBM. Some of the promising methods include the ones based on K-means with a distance metric (Xing et al., 2002) and K-means with a probabilistic framework (Basu et al., 2004). We, thereafter, present a thorough comparative analysis with our proposed methods and other existing semi-supervised clustering techniques.

2 Background

In this section we describe some concepts related to multi-objective optimization (MOO).

2.1 MultiObjective Optimization

Simultaneously optimizing several objective functions is known as multi-Objective optimization (MOO) (Deb, 2001). In general the objective functions used in MOO are conflicting in nature. A real-life example could be buying a car where the objectives are : i) minimizing cost and ii) maximizing comfort. In mathematical terms, a MOO problem can be formally stated as: Finding the vectors of decision variables $x = [x_1, x_2, x_3, ..., x_n]^T$ which will satisfy m inequality constraints: $g_i(x) \geq 0, i = 1, 2, ...m$ and p equality constraints $h_j(x) = 0, j = 1, 2, ..., p$ and simultaneously optimize M objective functions $f_1(x), f_2(x), ..., f_M(x)$.

2.2 Domination

A solution $x^i = \{f_1(x^i), f_2(x^i), \ldots, f_M(x^i)\}$ is said to dominate a solution $x^j = \{f_1(x^j), f_2(x^j), \ldots, f_M(x^j)\}$ denoted as $x^i \prec x^j$ iff $f_m(x^i) < f_m(x^j),\ \exists m \in \{1, 2, \ldots, M\}$, and $f_m(x^i) \leq f_m(x^j),\ \forall m \in \{1, 2, \ldots, M\}$

Two solutions x^i and x^j are said to be non-dominated with each other if and only if neither $x^i \prec x^j$ nor $x^j \prec x^i$.

A solution $x \in P$ is called *Pareto Optimal* with respect to P if there is no solution $x' \in P$ such that x is dominated by x'. The set of Pareto Optimal solutions is known as *Pareto set*.

Non Dominated Sorting is to divide the population $\mathbb{P}$ in $K(1 \leq K \leq N)$ fronts. Let $\mathfrak{F} = \{F_1, F_2, \ldots, F_K\}$ be the set of these K fronts in decreasing order of their dominance. The division of the solutions is such that i) Each solution in a front is non-dominated with each other, and ii) each solution in a front F_k is dominated by at least one solution in its preceding front $F_{k'}, k' < k \bigwedge 1 \leq k, k' \leq K$.

2.3 NSGA-II in the Light of MOO

Solving a problem consisting of multiple objectives in general produces more than one solution and these obtained solutions are termed as Pareto Optimal solutions. If no external condition is specified, it becomes really difficult to distinguish between these sets of solutions in terms of their performance. In the current state-of-art, we have always observed a tendency to convert the MOO problem into a single objective optimization (SOO) problem in order to produce single Pareto optimal solution at a time.

In this regard, a number of multiObjective-based evolutionary algorithms (MOEA) have been proposed (Deb, 2001; Fonseca and Fleming, 1993), where the algorithm deals with a number of competing objectives simultaneously. NSGA (Tanaka et al., 1995) is one of such members in the league of suggested EA (Evolutionary Algorithms) methods. NSGA-II (Deb et al., 2013) is an improvement over the existing NSGA algorithm, where a diverse set of solutions is found and it is observed that the algorithm tends to converge near the true Pareto optimal set. Among all such existing EAs, NSGA-II performs better than the rest and hence it is a promising algorithm for EA based MOO.

3 NSGA-II-Based Clustering Algorithm

In this section we describe the basic framework of NSGA-II-based clustering approach. The proposed clustering technique can detect the number of clusters automatically.

3.1 Problem Encoding

In this algorithm, cluster medoids are encoded in the form of a chromosome. We therefore assume that the medoid is the most representative point of a given cluster. The number of clusters is varied over a range, 2 to K_{max} where K_{max} is the maximum possible number of clusters. So for a given chromosome, first a random number is generated in the range of 2 to K_{max}. Let this be K_i. The K_i centers are encoded in that particular chromosome and those centers are randomly chosen from the set of all documents. Each document is assigned a positive integer value at the beginning. A point in the chromosome represents any such document. But all the values within the chromosome must be unique, that is there should not be any repetition while assigning values *i.e.*, document IDs in the chromosome should be unique. At first the population is initialized randomly. If the length of the population is p, we will generate p chromosomes at the beginning. For example, suppose we have 9 documents having IDs from 0 to 9 and the K_i value is 4. Let us assume that the initial selection of documents for the medoids have IDs 2, 6, 7 and 9. The chromosome becomes (**2 6 7 9**).

3.2 Assignment of Documents to Different Clusters

In our experiments we have used cosine and Euclidean distance as separate parameters to assign the documents in respective clusters. For each document we determine any of the available distance measures with respect to all the cluster medoids (encoded in a particular chromosome). Finally the document is assigned to that cluster medoid ($\overline{m}_i$) with respect to which it is having the minimum distance. Once the assignment has been done for all the documents, the new cluster medoids are calculated based on the new clusters formed. These new medoids replace the existing medoids represented in that particular chromosome.
$j = \arg\min_{j=1}^{K} d(\overline{x}, \overline{m}_j)$.
Here, $\overline{x}$ represents a document and $\overline{m}_j$ denotes the jth cluster-medoid. The function $d(\overline{x}, \overline{m}_j)$ represents any distance measure between document $\overline{x}$ and cluster medoid $\overline{m}_j$. The document $\overline{x}$ would be finally assigned to cluster j. Once the assignment has been done for all the documents, the new cluster medoids are calculated based on the new clusters formed. These new medoids replace the existing medoids represented in that particular chromosome.

3.3 Objective Functions Used

Several cluster validity indices exist in the literature like: Davies-Bouldin (DB) index (Davies and Bouldin, 1979), Dunns index (Dunn, 1973), Calinski Harabasz index (Caliński and Harabasz, 1974), Xie-Beni (XB) index (Xie and Beni, 1991), I-index (Maulik and Bandyopadhyay, 2002). These indices can measure the goodness of an obtained partitioning. It is established by Maulik and Bandyopadhyay (2002) that I-index performs better than the existing cluster validity indices in terms of finding the appropriate number of clusters. Hence, in order to measure the goodness of the partitioning represented in a particular chromosome, two cluster validity indices are used, I-index (Maulik and Bandyopadhyay, 2002) and XB index (Xie and Beni, 1991).

3.4 Genetic Operators

We use classical mutation and crossover operators as proposed in NSGA-II (Deb et al., 2013) to bring diversity in our population. Suppose there is a chromosome (2 4 5 7 8 9) representing a parent chromosome. In a mutation two documents are selected and exchanged.
(2 **4** 5 7 8 **9**) => (2 **9** 5 7 8 **4**)
In the case of crossover operation the bits are exchanged between parent chromosomes to produce offsprings. Once a crossover point is selected, the permutation till this point is copied from the first parent, then the second parent is scanned and, if the number is not yet in the offspring, it is added. For example, suppose the parent chromosomes are represented by (1 2 3 4 5 6 7 8 9) and (4 5 3 6 8 9 7 2 1) and the crossover point is 5. The offspring becomes :
(**1 2 3 4 5** 6 7 8 9) + (4 5 3 6 8 9 7 2 1) => (1 2 3 4 5 6 8 9 7)
Thereafter the selection operation of NSGA-II is applied. As described in (Deb et al., 2013), first the old population and the new population obtained after the application of mutation and crossover are merged. Now the non-dominated sorting procedure of NSGA-II is applied to divide the merged population (if the population size is N, then the size of the merged population is 2 × N) into a set of non-dominated fronts. The selection operation is illustrated in Fig 2. Solutions belonging to the best non-dominated set F_1 are among the best solutions in the combined population. If the size of F_1 is smaller than N, all members of the set are selected for the new population. The remaining members of the population are chosen from subsequent non-dominated fronts in the order of their ranking. If for a particular front F_i, $\|F_i\| > (N - \sum_{j=1}^{i-1} \|F_j\|)$, then all the solutions of the F_i front cannot be accommodated in the new population. In that case, in order to select the required number of solutions, the concept of crowding distance is used. In order to ensure diversity, the solutions which are far away lying in some non-crowded region are given special attention. Those are given higher priority while being selected.

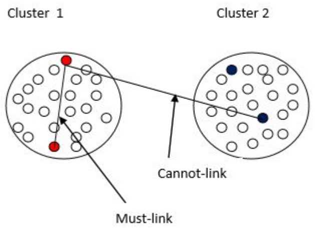

Figure 1: Concepts of must-link and cannot-link constriants

Until the limit is reached, iterations are performed and the steps of mutation, crossover and selection are repeated. Finally, a set of non-dominated solutions on the final Pareto front is obtained.

4 Application of Semi-supervision on NSGA-II Algorithm

As mentioned earlier we have used two different methods to induce the flavor of semi-supervision in NSGA-II algorithm. These methods are described below:

4.0.1 Internal-NSGA-II-clus

Here we perform some modifications in the selection step of NSGA-II to take care of the available supervised information in terms of must-link and cannot-link constraints. The computation of non-dominated fronts depends not only on the objective functions (XB and I indices) but on the available constraints (must-link, cannot-link) also. Here, the number of constraints violated by each solution also contributes in determining the rank of that solution during selection operation.

A must-link constraint ensures that two instances should remain in the same cluster as shown in Fig 1, whereas a cannot-link constraint ensures that two instances should be in two different clusters. From the initial labeled information the must-link and cannot-link constraints are chosen. It is assumed that the documents lying in the same cluster obey must-link and the documents lying in different clusters obey cannot-link.

Along with XB and I indices, 10% of the labeled information in the form of must-link and cannot link constraints is also used in crowded distance computation.

4.1 Computation of Crowding Distance

If n is the number of solutions in a given front F, $F(d_j)$ is the crowding distance of jth solution at a given front F, and $I(d_1)$ and $I(d_n)$ are boundary values for crowding distance in F, then the procedure through which the crowding distance is calculated is described in Algorithm 1, where $I(k).m$ is the m^{th} objective function of the k^{th} individual in I, f_m^{max} is the maximum value of m th objective function, and f_m^{min} is the minimum value of mth objective function.

Algorithm 1 Computation of crowding distance

```
for each front $F$ do
    $F(d_j) = 0$
    for each objective function $m$ do
        sort the individuals in $F$ based on $m$, such that
        $I = sort(F, m)$
        $F(d_1) = \infty$ and $F(d_n) = \infty$.
        for $k = 2$ to $(n-1)$ do
            $F(d_k) = F(d_k) + \frac{I(k+1).m - I(k-1).m}{f_m^{max} - f_m^{min}}$
        end for
    end for
end for
```

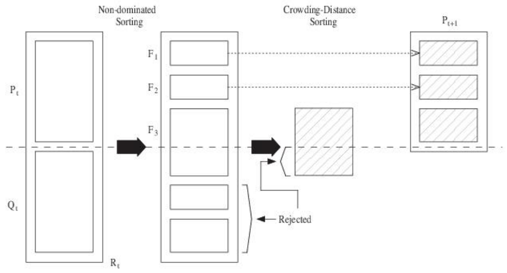

Figure 2: Selection operation of NSGA-II

4.2 Modified Crowding Distance

Along with the available objective functions, we also consider the available must-link and cannot-link constraints while computing the crowding distance. For a given solution (S), its must-score (w_m) and cannot-score (w_c) are calculated. If S obeys a must-link constraint then its must-score is increased by 1 whereas if it obeys a cannot-link constraint then its cannot-score is increased by 1.

The overall must-score w_m of an individual solution (S) is calculated as follows: $w_m = \sum_{i=1}^{M} I_{f_M^i}$

$$I_{f_M^i} = 1 \quad , \quad \text{if } S \text{ satisfies the } i\text{th must-link} \tag{1}$$

$$= 0 \quad , \quad \text{otherwise} \tag{2}$$

Here M: total number of must-links, f_M^i represents ith must-link constraint. Similarly, cannot-score w_c of an individual solution is calculated as follows: $w_c = \sum_{i=1}^{C} I_{f_C^i}$

$$I_{f_C^i} = 1 \quad , \quad \text{if } S \text{ satisfies the } i\text{th cannot-link} \tag{3}$$

$$= 0 \quad , \quad \text{otherwise} \tag{4}$$

Here C: total number of cannot-links, f_C^i represents ith cannot-link constraint.

The modified crowding distance of kth solution is computed as follows.
$F(dnew_k) = F(d_k) + \frac{w_m}{M} + \frac{w_c}{C}$
where $F(d_k)$ is the original crowding distance of kth solution computed using the procedure mentioned in Section 4.1. And w_m and w_c are the total must-score and cannot-score of kth solution, respectively.

Selection After the computation of crowding distance the selection process is carried out using the crowded-comparison operator ($\prec_n$) (Deb et al., 2013). Let us assume that $F(dnew_j)$ corresponds to the new crowding distance for the j^{th} individual in non-dominated front F and p and q are the p^{th} and q^{th} individuals of a particular non dominated front F. After the application of the non-dominated sorting procedure, suppose solutions p and q are assigned ranks of p_{rank} and q_{rank}, respectively. Then the crowded-comparison operator is defined as follows in Algorithm 2.

Algorithm 2 Computation of crowded comparison operator

```
$p \prec_n q$ ($q$ dominates $p$) if
i) $p_{rank} < q_{rank}$ or ii) if $p_{rank} = q_{rank}$, i.e. $p$ and $q \in F$ then $F(dnew_p) > F(dnew_q)$ i.e., the crowding distance should be greater.
```

4.3 External-NSGA-II-clus

In this method, at first the unsupervised clustering technique NSGA-II-clus (as described in Section 3) is executed on the given set of documents to obtain different partitionings on the final Pareto front. Here no modification is done in the internal steps of NSGA-II-clus, the available supervised information is used to select a single best solution from the final non-dominated set of solutions. 10% information in the form of must-link and cannot-link constraints is used to rank each of the non-dominated set of solutions. Basically each solution on the final Pareto front represents a partitioning. Experiments are performed to check which partitioning obeys the available must-link and cannot-link information. The solution with the maximum match is selected as the final solution from the Pareto Optimal front. For each solution on the final Pareto front, we follow some scoring mechanism. For must-links, if two points present in the link lie in the same cluster present in that solution, then we increase the score of the solution by 1. Similarly, for cannot-links, if two points present in the link lie in two different clusters, we increase the score of the non-dominated solution by 1. Thus we calculate the scores of all non-dominated solutions. The solution with the highest score is selected as the final solution.

5 Datasets and Experimental Results

We use the dataset made available by Mollá and Santiago-Martinez (2011), from which we randomly extract 276 clinical questions. Each question is associated with an average of 5.89 documents, and can be seen as an independent clustering task. The proposed NSGA-II-clus (internal and external) and AMOSA-clus (Ekbal et al., 2013) clustering techniques are therefore applied on each question individually. The average entropy value of all the questions is then calculated. For both internal and external NSGA-II clus algorithm, we first select 10% of the must-link and cannot-link constraints. For Internal-NSGA-II-clus, this supervised information is used in providing ranking of all the solutions during selection phase of each generation. In case of External-NSGA-II-clus this available supervised information is used in assigning a score to each of the solutions on the final Pareto front. Based on the highest score we select a single solution and compute the entropy values accordingly. The parameters for the proposed NSGA-II-clus (internal and external) semi-supervised approach are as follows: population size = 20, number of generations = 20, mutation probability = 0.2 and crossover probability = 0.6. These values were determined after performing a thorough sensitivity study. The parameters of AMOSA-clus are kept similar to those reported by Ekbal et al. (2013).The proposed NSGA-II-clus (internal and external) and AMOSA-clus approaches along with two semi-supervised approaches, namely K-Means with Distance Metric (Xing et al., 2002) and K-Means with probabilistic framework (Basu et al., 2004) are applied on the same datasets.

The K-Means+ Distance Metric (Xing et al., 2002) algorithm in its simplest sense is a variation of K-means. In the usual K-means, Euclidean or cosine distance is used as a measure of distance or separation between any two points in the space. Suppose an user wants certain points to be regarded as similar, according to some distance metric. Our task is to learn a distance metric automatically over a set of points which takes into account this relationship. In this algorithm, however instead of Euclidean or cosine distance, the concept of Distance Metric is used for our benefit.[1]

In the case of a probabilistic framework, a set of data points is randomly partitioned into a specific number of clusters which serve as the unsupervised partitioning initially. Here also supervision is provided in terms of two constraints *i.e.*, must-link and cannot-link (Basu et al., 2004). A modified version of Expectation-Maximization algorithm is used here to obtain the final partitioning which also obeys the available supervised information.

Table 2: Cluster entropies obtained by different approaches. Here KM_{DM} and KM_{Prob} denote, respectively, the *K*-means with distance-metric-based approach and *K*-means with probabilistic approach

Dist.	*AMOSA*		*Internal NSGA-II*		*External NSGA-II*		KM_{DM}	KM_{Prob}
	best	average	best	average	best	average		
Euclidean	0.177	0.235	0.025	0.092	0.063	0.117	0.534	0.274
Cosine	0.177	0.230	0.018	0.067	0.070	0.122	—	0.296

[1] To simplify our terminology, in this paper we use the term “cosine distance” to represent 1 - cosine similarity. The fact that neither the cosine distance nor the cosine similarity are true distance metrics does not affect the argumentation in this paper.

Two versions of the proposed NSGA-II-clus (internal and external) and AMOSA-clus algorithms are executed with the following distance measures: i) (version 1) Euclidean distance as the similarity measure for the assignment of documents to different clusters and also for the computation of objective functions; and ii) (version 2) with cosine similarity as the similarity measure for the assignment of documents to different clusters and also for the computation of objective functions. The average entropy values attained by these techniques are reported in Table 2. For the best-case computation we select those solutions from the final Pareto front obtained by internal-NSGA-II-clus and AMOSA-clus which possess the minimum entropy values. In the case of external-NSGA-II-clus, the best solution is selected using the steps as discussed in Section 4.3. The corresponding entropy values for those solutions are calculated. For the average case (unsupervised) computation we select all the solutions on the final Pareto Optimal front and calculate entropy for each of the solutions. Then we take the average entropy of all the solutions and report those values both for NSGA-II-clus (internal and external) and AMOSA-clus in Table 2.

Table 2 shows that, using 10% supervised information, the probabilistic framework approach outperforms the distance metric learning approach in case of Euclidean distance measure. Among all the algorithms, semi-supervised internal-NSGA-II-clus yields the highest performance in the best case as well as in the average case. This is also better than the AMOSA-based clustering algorithm, which was used for EBM by Ekbal et al. (2013). In order to show that our proposed NSGA-II-clus (internal and external) is also able to predict the correct number of clusters from different questions automatically, we have reported the error rate as below: $error = \frac{\sum_i (target_i - predicted_i)^2}{\#of questions}$
Here $target_i$ denotes the actual number of clusters for a particular question and $predicted_i$ denotes the predicted number of clusters by the proposed NSGA-II-clus (internal and external) technique for a particular question. Here as mentioned earlier in Section 2, for each question, we have varied the number of clusters in the range 2 to $\sqrt{n}$ where n is the number of documents per question. The average number of clusters identified by the proposed Internal-NSGA-II-clus optimizing XB-index and I-index as the objective functions for each question are 2.13 and 2.27, respectively, with cosine and Euclidean distance measurements. The average number of clusters identified by the proposed external-NSGA-II-clus optimizing XB-index and I-index as the objective functions for each question are 2.45 and 2.32, respectively, with cosine and Euclidean distance measurements. The average number of clusters in the actual annotated set is 2.38. Moreover we have also computed the error rates of different automatic clustering techniques. For AMOSA-clus the error rates are 1.90 with cosine similarity and 1.91 with Euclidean distance. For internal-NSGA-II-clus the error rates are 1.33 with cosine similarity and 1.49 with Euclidean distance. For External-NSGA-II-clus the error rates are 1.74 with cosine similarity and 1.69 with Euclidean distance. In Ref. (Ekbal et al., 2013) it has already been proved that AMOSA-clus provides the minimal error rate compared to different existing techniques and heuristics. But the proposed approach provides minimal error rate compared to AMOSA-clus. This again proves the efficacy of the proposed semi-supervised approach.

6 Conclusion

In this paper we have used semi-supervised clustering to find clusters of medical publications for the task of Evidence Based Medicine. We have proposed two different frameworks using the concepts of MultiObjective Optimization (MOO) for solving the problem of semi-supervised clustering. As the underlying optimization technique we have used a popular evolutionary strategy, NSGA-II. A comparative study between two MOO-based semi-supervised clustering approaches and two existing semi-supervised approaches is also provided. Our experiments show the efficacy of the MOO-based semi-supervised approach on medical publications. The improved results using several objective functions are encouraging. The comparative study on medical data proves the efficacy of our NSGA-II based approach. In future, we would like to compare the proposed technique with alternative supervised techniques. We wish to explore other similarity measures to determine the distance between two given documents. The proposed technique would also be evaluated on other data sets and non-overlapping clustering techniques.

References

Sanghamitra Bandyopadhyay, Sriparna Saha, Ujjwal Maulik, and Kalyanmoy Deb. 2008. A simulated annealing-based multiobjective optimization algorithm: Amosa. *IEEE Transactions on Evolutionary Computation*, 12(3):269–283, June.

Sugato Basu, Mikhail Bilenko, and Raymond J. Mooney. 2004. A probabilistic framework for semi-supervised clustering. *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining(KDD 2004*, pages 59–68, August.

Tadeusz Caliński and Jerzy Harabasz. 1974. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, (1):1–27.

David L. Davies and Donald W. Bouldin. 1979. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, pages 224–227.

Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. 2013. A fast and elitist multiobjective genetic algorithm: Nsga-II. *Artificial Intelligence in Medicine. Springer Berlin Heidelberg*, pages 305–309.

K. Deb. 2001. Multi-objective optimization using evolutionary algorithms. *John Wiley & Sons*.

Joseph C Dunn. 1973. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. pages 32–57.

Asif Ekbal, Sriparna Saha, Diego Molla, and K Ravikumar. 2013. Multi-objective optimization for clustering of medical publications. *Proceedings of the Australasian Language Technology Association Workshop 2013 (ALTA 2013)*, pages 53–61, December.

Carlos M. Fonseca and Peter J. Fleming. 1993. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. *ICGA*.

Anil K. Jain, M. Narasimha Murty, and Patrick J. Flynn. 1999. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323.

U. Maulik and S. Bandyopadhyay. 2002. Performance evaluation of some clustering algorithms and validity indices. *Pattern Analysis and Machine Intelligence, IEEE Transactions*, 24(12):1650–1654.

Diego Mollá and Maria Elena Santiago-Martinez. 2011. Development of a corpus for evidence based medicine summarisation. *Proceedings of the Australasian Language Technology Association Workshop*.

Sara Faisal Shash and Diego Molla. 2013. Clustering of medical publications for evidence based medicine summarisation. *Artificial Intelligence in Medicine. Springer Berlin Heidelberg*, pages 305–309.

Masahiro Tanaka, Hikaru Watanabe, Yasuyuki Furukawa, and Tetsuzo Tanino. 1995. Ga-based decision support system for multicriteria optimization. *Systems, Man and Cybernetics, 1995. Intelligent Systems for the 21st Century., IEEE International Conference*, pages 1556–1561.

Xuanli Lisa Xie and Gerardo Beni. 1991. A validity measure for fuzzy clustering. *IEEE Transactions on pattern analysis and machine intelligence*, 13(8):841–847.

Eric P. Xing, Andrew Y. Ng, Michael I. Jordan, and Stuart Russel. 2002. Distance metric learning, with application to clustering with side-information. *In Advances in neural information processing systems*, pages 505–512.

Xiaojin Zhu and Andrew B. Goldberg. 2009. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130.