2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

Multi-View Clustering of Web Documents using Multi-Objective Genetic Algorithm

Abdul Wahid
Victoria University of Wellington
New Zealand
abdul.wahid@ecs.vuw.ac.nz

Xiaoying Gao
Victoria University of Wellington
New Zealand
xgao@ecs.vuw.ac.nz

Peter Andreae
Victoria University of Wellington
New Zealand
pondy@ecs.vuw.ac.nz

***Abstract*—Clustering ensembles are a common approach to clustering problem, which combine a collection of clustering into a superior solution. The key issues are *how to generate different candidate solutions* and *how to combine them*. Common approach for generating candidate clustering solutions ignores the multiple representations of the data (i.e., multiple views) and the standard approach of simply selecting the best solution from candidate clustering solutions ignores the fact that there may be a set of clusters from different candidate clustering solutions which can form a better clustering solution.**

This paper presents a new clustering method that exploits multiple views to generate different clustering solutions and then selects a combination of clusters to form a final clustering solution. Our method is based on Nondominated Sorting Genetic Algorithm (NSGA-II), which is a multi-objective optimization approach. Our new method is compared with five existing algorithms on three data sets that have increasing difficulty. The results show that our method significantly outperforms other methods.

I. Introduction

Clustering is an unsupervised learning technique for organizing similar objects into different groups. Since it is hard to define the similarity especially in high-dimensional data, thousands of clustering algorithms have been proposed in the last 50 years [1]. As no single clustering algorithm is suitable for all types of problems, researchers have been trying different techniques for combining different clustering algorithms (clustering ensembles) [2], [3], [4], [5], [6], [7].

The main goal of clustering ensembles is to solve the problem of producing superior clustering solution from given set of clustering solutions. This problem was previously approached by researchers from different angles and so far the best known approach for clustering ensembles is median partition based approach [8] in which a single candidate clustering solution that has the maximum similarity from all candidate clustering solutions is selected as the final clustering solution.

The clustering ensembles methods include two important steps: 1) generating a set of candidate clustering solutions and 2) combining the set of candidate clustering solutions to generate final clustering solution. In our evolutionary based clustering approach, step 1 corresponds to an initialization phase in which a set of initial candidate clustering solutions is generated, and step 2 is the evolutionary phase in which the final solution is evolved from the initial candidates.

A common practice for step 1 is to use different clustering algorithms with different initialization parameters. This is a good strategy if the data can only be represented from one perspective or one view. However, in case of textual data, a document can be represented by multiple views e.g., semantic view (i.e. topics, title, hypertext etc.) and a syntactic view (i.e. term frequencies). Considering multiple views at the same time has already proven to result in better clusters [9]. Based on this notion, we propose using multiple views for generating different candidate in a clustering ensemble method.

The second step in clustering ensembles method generally chooses the best clustering solution among a given set of clustering solutions. However, this might not be an optimal solution because each clustering solution generally consists of a mixture of high and low quality clusters[1]. In order to generate a better clustering solution, a selection of high quality clusters from different candidate clustering solutions should be combined instead of selecting one solution from the set of candidate clustering solutions.

Generally clustering ensemble methods focus on optimizing a single objective function (i.e., either maximizing the inter-cluster distances or minimizing the intra-cluster similarity). However, in recent years the trend has shifted toward formulating clustering ensembles as multi-objective optimization problems to gain better results [10], [3], [11].

This research work propose a Multi-view Multi-objective Evolutionary Algorithm (MMOEA) which uses multi-objective optimization approach for improving document clustering, focusing on the following key ideas:

1) Generate different candidate clustering solutions for step 1 from multiple views of the data.
2) Select the best combination of clusters from all candidate clustering solutions to form a final clustering solution instead of selecting a single clustering solution from the candidate clustering solutions.
   - Using a multi-objective optimization approach instead of single objective approach.
   - Allowing overlapping clusters so that one document can be in multiple clusters. This is often desirable for document clustering because many documents

[1]high quality clusters are the ones that have high intra-cluster similarity and are dissimilar from other clusters

978-1-4799-1488-3/14/$31.00 ©2014 IEEE

can be classified under multiple topics or categories.

The paper is organized as follows: section II discusses background and related methods for our approach; section III provides the details of the approach; section IV describes the experimental setup; section V discusses the comparison of the approach with other methods and provides a statistical analysis; lastly section VI concludes the paper and provides future directions.

II. Background and Related Work

This section provides a brief introduction of background knowledge of this paper along with related work.

A. Clustering as an Optimization Problem

Clustering can be defined as grouping a set of similar objects into different groups without any prior information [12]. The objects in the same cluster should be similar whereas objects in different clusters should be dissimilar. Formally, a clustering problem can be viewed as an optimization problem if the goal of the clustering is to maximize some criterion measuring the quality of the clusters given some similarity measure.

The algorithms proposed in the literature mainly differ in the criterion function and (dis)similarity measure. Generally clustering algorithms optimize only one criterion function. However, it is also possible to use multiple criterion functions through ensemble methods [10], [3] or using multi-objective optimization approach [11], [13].

B. Clustering Ensembles

Existing clustering ensemble methods include co-association matrix based methods [2], Bayesian approaches [14], hyper-graph partitioning [3], [15], mixture models [16], [4] and some use evolutionary approach [17], [18]. One of the limitation in many of these methods is that they can only select from candidate clustering solutions and are not able to construct new solutions out of the candidates. Furthermore, many of the methods follow single objective approach and implement conflicting criteria into single function, which is argued to be a bad practice in [11].

C. Multi-objective Optimization

Real life optimization problems are mostly multi-objective in nature and often have conflicting objectives. The goal of multi-objective optimization is to search for a set of solutions that optimize a number of functions along with satisfying some constraints. These solutions are often called *Pareto optimal* and their plotted form provides *Pareto front*. The solutions are called *nondominated* if they are on Pareto front. *Nondominated* solutions means there is no other feasible solutions which will provide better results on one objective without affecting another.

There are many techniques for multi-objective optimization problems, however only a few of them are not sensitive to the continuity and shape features of the Pareto front. Evolutionary Algorithms are considered to be very useful for multi-objective optimization problems because they are able to work with all types of Pareto fronts. Furthermore they can find multiple Pareto optimal solutions in single iteration of the algorithm and they can provide good approximation of the true Pareto front [19].

D. Multi-objective Evolutionary Algorithms

Evolutionary Algorithms are search and optimization techniques inspired by biological evolution [20]. An initial population of candidate solutions are used to start the search and at each iteration *selection*, *crossover* (or *recombination*) and *mutation* are performed to generate a new population (a set of solutions). The iteration is usually terminated by a fixed limit for maximum number of iterations.

Generally the set of candidate solutions (also called *individuals*) are randomly generated in the initial population. A *fitness function*, which is based on the objective function and constraints, evaluates each individual (candidate solution) to determine its *fitness* value.

The main aim of Multi-objective evolutionary algorithms (MOEA) is to approximate the true Pareto front as accurate as possible. Therefore, identifying and keeping the nondominated solutions in different iterations along with preserving diversity becomes an important factor for the algorithm. MOEAs generally include a *dominance based ranking* process in selection step along with an external archive (*elitism*) to keep the nondominated solutions that are found during the number of iterations.

Existing MOEA approaches that include elitism and dominance ranking are considered to be most successful MOEAs. In particular, NSGA-II (Nondominated Sorting Genetic Algorithm) [21], PESA-II (Pareto Envelop based Selection Algorithm) [22] and SPEA-II (Strength Pareto Evolutionary Algorithm) [23] are most prominent in literature. Comprehensive details about MOEA and their importance can be found in [19].

E. Related Work

Handl and Knowles [11], [24] proposed a state-of-the-art evolutionary approach for multi-objective clustering (Multi-objective clustering algorithm with K-determination; named Mock) based on PEAS-II. They used a fitness function based on two objectives connectedness and compactness of clusters. The placement of neighboring objects in the same cluster is called connectedness and is defined as:

$$Conn(C) = \sum_{i=1}^{N} \sum_{j=1}^{M} y_{i,x_{ij}} \tag{1}$$

where x_{ij} is the jth nearest neighbour of object o_i and M is the total number of neighbors that contribute to the measure and $y_{i,x_{ij}}$ is given by

$$y_{i,x_{ij}} = \left\{ \begin{array}{ll} \frac{1}{j} & \text{if } \nexists C_k : o_i \in C_k \wedge x_{ij} \in C_k \\ 0, & \text{otherwise} \end{array} \right\} \tag{2}$$

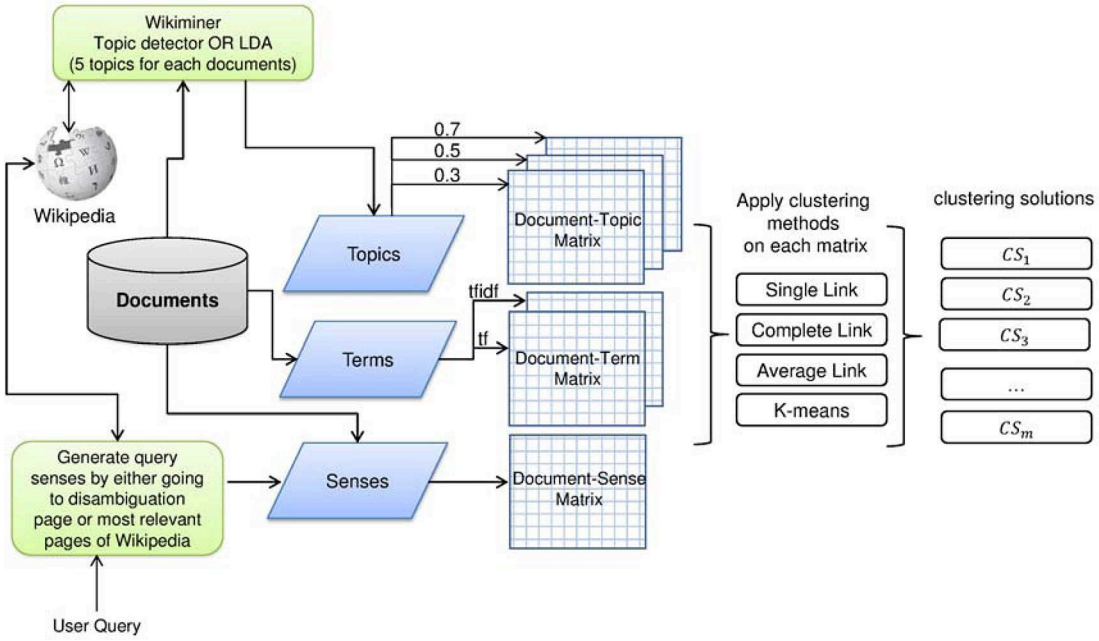

Fig. 1: Process of generating 24 different clustering solutions.

The cluster compactness is the overall deviation of clustering solution. It improves when the number of clusters are high, whereas improving connectivity requires less number of clusters. These conflicting objectives make the algorithm to explore interesting areas of the search space.

Another multi-objective clustering algorithm was proposed in [13]. It uses two objectives the number of clusters and the intra-cluster variation (which is computed over all clusters). Both objectives are required to be minimized, although they are in conflict with each other. The algorithm uses Pareto dominance to discover a set of nondominated clustering solutions that are different from each other (diversity is preserved) by finding smallest intra-cluster variance for minimum number of clusters.

Bandyopadhyay et al. [25] introduced a multi-objective evolutionary algorithm that also performs fuzzy clustering. It uses two objective functions: J_m criterion [26] and the Xie-Beni index [27], which is a distance between two closest clusters.

Other variations of multi-objective clustering algorithms include [28], [29], [30]. Mostly these algorithm use two objectives and their main focus was on intra-cluster distance that needed to be minimized. The main limitations of these algorithms were considering only single view of the data and selecting the best solution from the set of candidate solutions.

III. Method

Our method is based on multi-view clustering in which different clustering solutions are derived from different views of the data and uses a modified version of NSGA-II [21] approach.

A. Initialization Method

Unlike previous clustering ensemble methods, we use different views for generating initial candidate clustering solutions. In our method, we considered terms in documents (bag of words), user query senses and topics in documents as three views. We assume the set of documents was generated using a query to a search engine.

The term view is generated by extracting the terms in the documents. The query senses view is generated by parsing Wikipedia disambiguation pages and extracting different senses of that query[1]. The topics for each documents are generated using the topic detection component of Wikiminer toolkit[2] [31] and Latent Dirichlet Allocation (LDA). Wikiminer generates the topics of a document by matching its terms with the titles of Wikipedia articles. Wikiminer is not able to generate multiple topics for documents that do not have sufficient terms corresponding to any Wikipedia article. Therefore, we use a simple implementation of LDA to tackle these (rare) cases and ensure that we will get at least five topics per document. The term view provides a syntactic representation of the document whereas the query sense and the topic view provide a semantic representation of the document.

We created two document-term matrices from the term view, one document-sense matrix from the query sense view and three document-topic matrices from the topic view of the documents.

In a document-term matrix, each row represents a document, each column represents a term, and a cell contains the

[1]We can skip the query sense view for datasets that do not provide queries
[2]for more information visit http://wikipedia-miner.cms.waikato.ac.nz/

weighted value of a term for a document. The first matrix is created using tfidf (a common weighting scheme) and the cell contains the tfidf score. The second matrix is created by simply computing the term frequencies (tf) for each cell of the matrix.

In a document-sense matrix, rows represent documents and columns represent different senses of the query. A cell contains 1 if the sense was present in the document and 0 if the sense is not present (exact string matching).

In a document-topic matrix rows represent documents and columns represent topics. A cell contains 1 if the similarity score between the topic and any topics of the document is above a threshold, otherwise the value is 0. We use Wu and Palmer similarity measure to match topics [32] and created three document-topic matrices based on three thresholds (0.3, 0.5 and 0.7).

We then apply four different clustering algorithms: single link, complete link, average link and k-means on these six matrices to generate 24 candidate clustering solutions.

Figure 1 depicts the initialization process. The total of 24 clustering solutions are generated from six matrices by applying four different clustering methods. The six matrices were generated from three views of the document.

B. Genetic Representation

Using MOEA for a clustering problem requires the representation of the clustering solutions, the objective functions and the operators (crossover and mutation).

The representation scheme used in our method is a matrix based binary encoding scheme [33]. In this scheme each clustering solution or individual is represented in form of a $k \times N$ matrix. In this matrix, columns represent documents and rows represent clusters. Figure 2 depicts a sample clustering solution and whose matrix based encoding is shown in Figure 3. The value 1 in cell (c,d) of the matrix means that the cluster c includes the document d.

The key advantage of this representation is that it can represent overlapping clusters. Note that this representation would allow a clustering in which some documents are not allocated to any cluster.

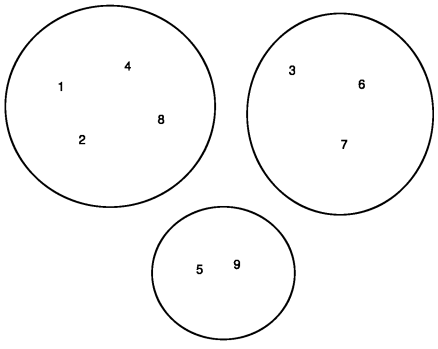

Fig. 2: Sample Clustering

	1	2	3	4	5	6	7	8	9
1	1	1	0	0	0	0	0	1	0
2	0	0	1	0	0	1	1	0	0
3	0	0	0	0	1	0	0	0	1

Fig. 3: A sample individual in a form of matrix-based binary encoding where columns represent the document ids and rows represent cluster numbers

C. Objective Functions

Our optimization is based on the standard three criteria: number of clusters, intra-cluster similarity and inter-cluster distances. The three criteria trade off against each other. Optimizing the first criterion leads to a few very large clusters, optimizing the second criterion leads to many very small possibly similar clusters, optimizing the third criterion leads to very distinct clusters.

We implemented the criteria as three functions to be minimized. The optimization functions Φ_s, Φ_f and Φ_a corresponds to minimum clusters, minimum unshared features and minimum inter-cluster similarity respectively. The optimization functions are defined as follows:

$$\Phi_s(C) = \frac{|C|}{N} \tag{3}$$

where $|C|$ is the total number of clusters and N is total number of documents in clustering solution C.

$$\Phi_f(C) = 1 - \frac{\sum_{c \in C} \frac{|\bigcap_{d \in c} \mathcal{F}(d)|}{|F|}}{|C|} \tag{4}$$

where c is a cluster in clustering solution C. The d is a document in cluster c. The $\mathcal{F}(d)$ is the set of features in document d. The $|F|$ is the total number of features in clustering solution C.

$$\Phi_a(C) = \frac{\sum_{c,c' \in C \wedge c \neq c'} \delta(c, c')}{|C|(|C| - 1)} \tag{5}$$

where c and c' are the clusters in the clustering solution C, and δ is a similarity function that computes shared number of features between all the documents of cluster c and c'.

$$\delta(c, c') = \frac{|\bigcap_{d \in c \cup c'} \mathcal{F}(d)|}{|F|} \tag{6}$$

where d represents document in either cluster c or c' . The fitness function becomes a minimization problem and can be formulated as:

$$\mathcal{C} = \underset{C \in P}{\operatorname{argmin}} \{\Phi_s(C), \Phi_f(C), \Phi_a(C)\} \tag{7}$$

where C is a candidate clustering solution. P is a set of candidate clustering solutions.

D. Crossover

In the crossover step, child clustering are constructed from pairs of clustering from the current population. Pairs of parents are randomly selected from the population, then subset of clusters from each pair of parents is randomly selected to form the children. As each cluster is selected, it is checked against the feasibility criterion to ensure that it has less than 40% overlap with any previously selected cluster. Infeasible clusters (which have more than 40% of overlap) are rejected. Clusters continue to be selected for a child until all documents are in at least one cluster of the child. If parents are exhausted before all documents are covered, the algorithm will add additional parents.

E. Algorithm: MMOEA

The algorithm MMOEA developed in this research work solves the multiple objectives without combining them into single objective function. The algorithm ranks individuals at each generation based on Pareto ranking and has the following features:

- it is based on the elitist principle.
- it implements a mechanism for explicitly preserving the diversity of solutions.
- it finds optimal solutions in a multi-objective optimization problems and focuses on non-dominated solutions.

Following are the important definitions [34]:

Definition 1. (Pareto Dominance) Let $\mathcal{Z}$ be a multi-objective optimization problem of the form: $p^* = \text{argmin}_p \{f_1(p), ...f_n(p)\}$. Let p' and p'' be two candidate solutions of $\mathcal{Z}$. p' dominates $p''(p' \prec p'')$ if the value of p' is lower than that of p'' according to at least one objective function and is less than or equal to the remaining objective functions.

Definition 2. (Pareto non-dominated set) Let $\mathcal{Z}$ be a multi-objective optimization problem of the form: $p^* = \text{argmin}_p \{f_1(p), ...f_n(p)\}$. Let $\mathcal{X}$ be a population of individuals for $\mathcal{Z}$, i.e a set of candidate solutions of $\mathcal{Z}$. $\mathcal{X}^*_{\mathcal{Z}} \subseteq \mathcal{X}$ is a Pareto non-dominated solution set of $\mathcal{Z}$ w.r.t $\mathcal{X}$ if and only if $p \nprec p^*, \forall p \in \mathcal{X}, \forall p^* \in \mathcal{X}^*_{\mathcal{Z}}$

Definition 3. (Pareto ranking) Let $\mathcal{Z}$ be a multi-objective optimization problem of the form: $p^* = \text{argmin}_p \{f_1(p), ...f_n(p)\}$. Let $\mathcal{X}$ be a population of individuals for $\mathcal{Z}$. The Pareto ranking function $\sigma : \mathcal{X} \rightarrow \mathbb{N}^+$ for $\mathcal{Z}$ is defined iteratively as follows. Let $\mathcal{X}_1 = \mathcal{X}$. For any given set of individuals $\mathcal{X}_i$, the Pareto rank of any p belonging to the maximal Pareto non-dominated solution set $\mathcal{X}^*_{\mathcal{Z},i}$ of $\mathcal{Z}$ w.r.t. $\mathcal{X}_i$ defined to be i (i.e., $\sigma(p) = i, \forall p \in \mathcal{X}^*_{\mathcal{Z},i}$), and $\mathcal{X}_{i+1} = \mathcal{X}_i \backslash \mathcal{X}^*_{\mathcal{Z},i}$.

The Pareto ranking function formally described in Definition 3 provides a ranking (i.e., a score) of all solutions in a given population $\mathcal{X}$. The remaining solutions are ranked iteratively by considering the non-dominated solutions which do not have a rank. This means all non-dominated solutions in $\mathcal{X} \backslash \mathcal{X}^*_{\mathcal{Z},1}$ have rank 2 and all the non-dominated solutions in $(\mathcal{X} \backslash \mathcal{X}^*_{\mathcal{Z},1}) \backslash \mathcal{X}^*_{\mathcal{Z},2}$ have rank 3 and so on. The iterative process will continue ranking until a rank (or score) is assigned to all the solutions in $\mathcal{X}$.

Algorithm 1 MMOEA

Input: Number of clustering solutions P, population size s and maximum number of generations m.
Output: Final clustering solution $\mathcal{C}$.

1: $\mathcal{X} \leftarrow initializePopulation(P)$
2: **for** $i \leftarrow 1, m$ **do**
3: Compute Pareto ranking for $\mathcal{X}$ and sort $\mathcal{X}$
4: $\mathcal{X}' \leftarrow$ top half from $\mathcal{X}$
5: $\mathcal{X}_{child} \leftarrow$ generate child population from $\mathcal{X}$
6: $\mathcal{X} \leftarrow \mathcal{X}' \cup \mathcal{X}_{child}$
7: $\mathcal{X} \leftarrow \mathcal{X}' \cup \mathcal{X}_{child}$
8: **end for**
9: $\sigma \leftarrow$ Compute Pareto ranking for $\mathcal{X}$
10: $\mathcal{X}^* \leftarrow \{p' \in \mathcal{X} : \forall p' \in \mathcal{X}, \sigma(p') = 1\}$
11: Select $\mathcal{C}$ from $\mathcal{X}^*$

Algorithm 1 describes the high level operation of our algorithm MMOEA. It takes three arguments, the initial clustering solutions, the population size and the number of generations. The initial population is the set of candidate solutions $\mathcal{X}$ described in section III-A and using crossover to double the population size from 24 to 48 individuals. The loop specified on line 2 is repeated until the maximum number of iterations m is reached. In each iteration the Pareto ranking function σ is computed for current population $\mathcal{X}$ according to Definition 3 using the multi-objective function described in equation 7.

The ranking function σ is used to sort the candidate solutions in $\mathcal{X}$ and a top half X' is generated. The $\mathcal{X}$ goes under a crossover to generate a new generation $\mathcal{X}_{child}$. A new population is formed by combining set of candidate solutions of $\mathcal{X}'$ and $\mathcal{X}_{child}$.

The set of Pareto optimal solution $\mathcal{X}^*$ is computed from $\mathcal{X}$ once the loop is completed. Lastly, the final clustering solution $\mathcal{C}$ is randomly selected from rank 1 Pareto front.

IV. Evaluation Experiments

This research work was compared with five clustering methods by evaluating the quality of the clusters against gold standard provided by the datasets. We used F-measure scores [35] and Rand Index (RI) values [36] of three datasets for evaluation[1]. The total population size was 24 and maximum number of generations was 1000.

[1]Note that the evaluation measures are necessarily different from the objective functions, because the gold standard cannot be made available to the clustering system

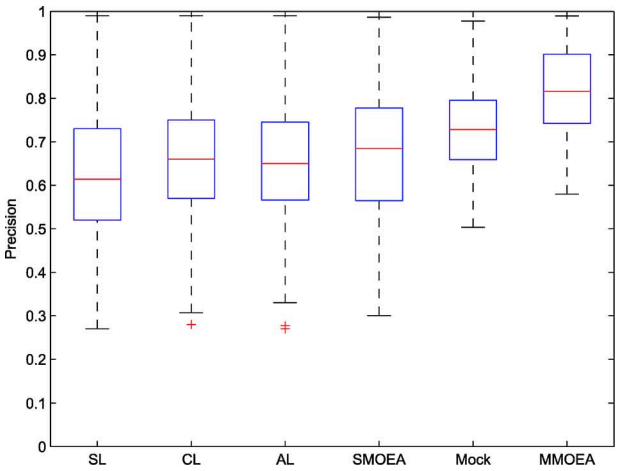

Fig. 4: Precision of the final solution on all datasets

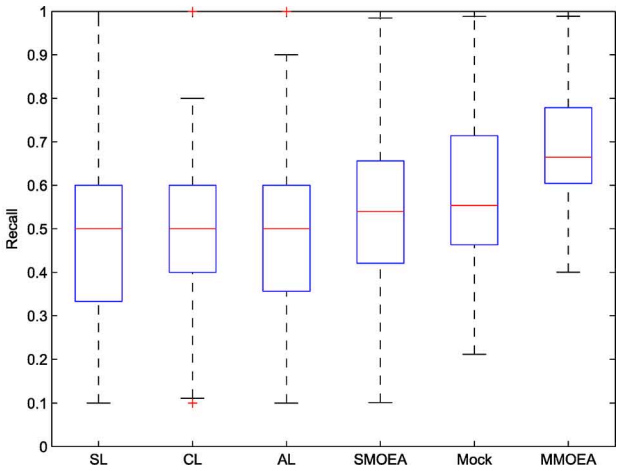

Fig. 5: Recall of the final solution on all datasets

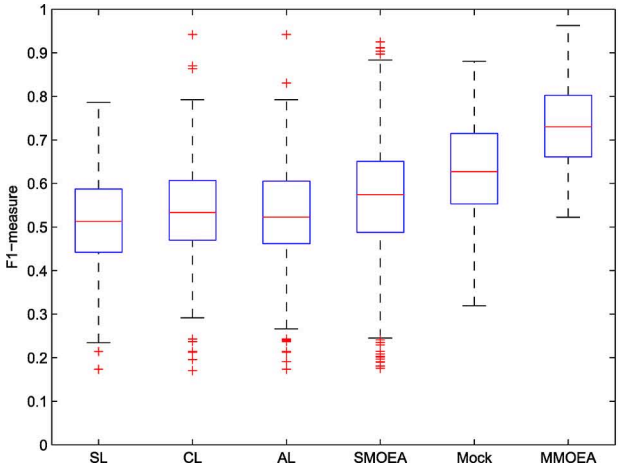

Fig. 6: F1-measure on all datasets

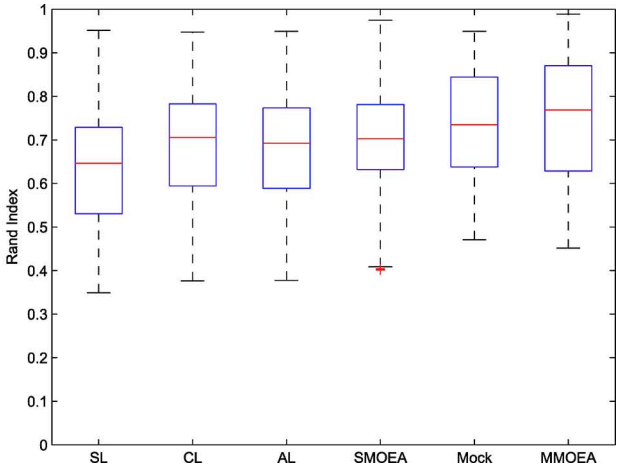

Fig. 7: Rand Index on all datasets

A. *Datasets*

The datasets used for evaluation are AMBIENT[1], MORESQUE[2] and ODP-239[3], which respectively increase the level of difficulty in finding the clusters. Following are the details of the datasets

1) AMBIENT [37] is generated from Wikipedia and has 100 documents for each of 44 ambiguous queries. The documents were gathered from the Yahoo search engine and are assigned different subtopics (ground truth). In this dataset most queries are only a single word.
2) MORESQUE [38] is an extension of AMBIENT dataset and has 100 documents for each of 114 ambiguous queries. It is also generated from Wikipedia. Unlike AMBIENT, MORESQUE has queries containing more than one word and is more difficult for finding clusters.
3) ODP-239 [39] has 100 web documents for each of 239 ambiguous queries and is generated from Open Directory Project[4]. This dataset has more ambiguous queries than AMBIENT and MORESQUE. Its subtopics are very hard to distinguish and they often have similar meanings, which makes it a hard dataset compared to AMBIENT and MORESQUE.

B. *Comparison*

Our method MMOEA was compared with five methods: SL, CL, AL, Single-view MOEA (SMOEA) and Mock. SL, CL and AL are three single objective clustering ensemble methods based on link-based pairwise similarity matrix known as Approximate SimRank-based similarity matrix (ASRS) [40] which is a modified version of SimRank-based similarity matrix (SRS) [41]. For SL, CL and AL Step 1, we generated candidate clustering by applying 10 different initializations with fixed number of clusters of k-means algorithm on one feature matrix (term view with tfidf scheme only). For Step 2 ASRS matrix was generated from the results of k-means algorithm and then to generate a final clustering solutions then we applied single-link hierarchical clustering for SL;

[1]downloaded from http://credo.fub.it/ambient/
[2]downloaded from http://lcl.uniroma1.it/moresque/
[3]downloaded from http://credo.fub.it/odp239/

[4]http://www.dmoz.org

complete-link hierarchical clustering for CL; and average-link hierarchical clustering for AL on ASRS matrix.

SMOEA is a single view version of our algorithm MMOEA and uses only the term view (tfidf view) with same GA parameters as MMOEA. Mock is the state-of-the-art multi-objective evolutionary algorithm [1] [11] and was used with standard parameters (code was provided by the author).

V. Results and Discussion

Figures 4, 5, 6 and 7 display boxplots of precision, recall, F1-measure and RI values computed for all the queries (397 queries in total where each query corresponds to 100 documents that are required to be clustered) in all three datasets using Single Link, Complete Link, Average Link, Mock and our methods (SMOEA and MMOEA). The y-axis represents the score from 0 to 1 and x-axis represents the clustering method. Mean is marked by a line in middle of the box and the box represents the quartiles (25% to 75%) of the values. The lower and upper dashed line on the box represents the deviation (5% to 95%) of the values and the rest of the values are marked as plus sign (generally considered as outliers).

Figures 4, 5, 6 and 7 shows that the mean value of MMOEA in terms of Precision, Recall, F1-measure and RI is significantly higher and the deviation is much lower than other methods. The SMOEA had better mean values than SL, CL and AL but worse than Mock and MMOEA in all experiments.

The 95% values (spread) of Precision, Recall and F1-measure shows that MMOEA is approached to be much better, However in terms of RI, MMOEA is approached to somewhat better than Mock while it is much better than other methods. SMOEA on both F1 and RI measure approached to be better than SL, CL and AL and worse than Mock, and MMOEA. The experiments on individual datasets (AMBIENT, MORESQUE and ODP239) also showed the similar results.

A. Statistical Analysis

Method	F1-Ranking	RI-Ranking
MMOEA	2.1511	2.1738
Mock	2.4169	2.7960
SMOEA	2.8249	3.1008
Complete Link	3.3249	3.1998
Average Link	3.4005	3.2821
Single Link	4.5466	3.6474

TABLE I: Average ranking of clustering methods on F1-measure and RI values

Statistical analysis was performed to draw a precise conclusion from the results of F1-measure and RI. Table I shows a ranking of clustering methods computed on F1 and RI values on all datasets based on Friedman's method [42]. It is evident that MMOEA is at the top with the ranking of 2.1511 and 2.1738 in terms of F1-ranking and RI-ranking respectively. Mock is at second place with ranking of 2.4169 and 2.7960. SMOEA secured third place with ranking of 2.8249 and 3.1008 in terms of F1-ranking and RI-ranking respectively. Friedman's measure, χ_F^2, for 4 degrees of freedom is 648.03 for F1 values and 198.1019 for RI values. These values signify that F1 and RI values are not random (observed by considering the critical values) and these results are statistically significant.

Since our interest was to compare MMOEA with other methods, we took F1 values and RI values of MMOEA as control group separately and performed the Bonferroni-Dunn test for $\alpha = 0.05$. The $p\text{-}values$ were significantly lower than 0.0001 which indicated that MMOEA is significantly better than others.

It is interesting to know how using multiple views can affect the clustering results. We did more experiments by using the multiple views in single objective clustering ensembles (SL, CL and AL) and observed that results were almost as good as the original version of SL, CL and AL but worse compared to Mock, SMOEA and MMOEA. Using multiple views will result in diverse clusters in different candidate clustering solutions. However directly applying standard algorithms is not enough to achieve better performance. Generating diverse clusters and choosing the high quality clusters from candidate clustering solutions, both in conjunction seems to result in better clustering approach.

VI. Conclusion

This paper has presented a multi-objective approach for clustering ensembles that uses multiple views to generate a set of candidate solutions and selects high-quality overlapping clusters from the candidate solutions to form a superior clustering solution. This paper has three contributions to improve clustering results. The first contribution is to use multiple views to generate an initial set of candidate clustering solutions. This results in diverse candidate clustering solutions having mixtures of very high and low quality clusters. The second contribution is to make a final clustering solution by combining the individual clusters from different candidate clustering solutions. The third contribution is to use the multi-objective ranking system from NSGA-II to guide the optimization because we our objective criteria are in conflict with each other.

The experiments have shown that MMOEA outperformed other methods in terms of F1 and RI index. SMOEA, the single view version of MMOEA, was better than the single objective methods but as it did not have a diverse set of clusters in step 1 of the clustering algorithm, it was not able to produce a better result than Mock or MMOEA. Therefore we conclude that having a diverse set of clusters from multiple meaningful views plays an important factor in our approach.

The presented approach is limited to domains that can have overlapping clusters and whose data can be represented by multiple views. Since documents can naturally be categorized under different topics and can be represented from different views, this work can be applied to various collections of documents (corpus). The only modification required is in the method which generates the matrix. For example, in the case

[1] http://personalpages.manchester.ac.uk/mbs/julia.handl/mock.html

of the Reuters data set for which there are no queries, we would just disable the user query sense and use only two views (topics and terms).

One of the potential future directions for this research work would be to identify the multiple views automatically without any domain knowledge. Other directions are to extend this work for domains that require partitions and to improve the efficiency by reducing the number of objectives while maintaining the same quality of clusters. Our next step is to extend this approach by improving crossover method and by adding different types of mutation operators which adds, deletes, replace, splits and merge clusters.

REFERENCES

[1] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010.

[2] A. L. Fred and A. K. Jain, "Data clustering using evidence accumulation," in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, vol. 4. IEEE, 2002, pp. 276–280.

[3] A. Strehl and J. Ghosh, "Cluster ensembles—a knowledge reuse framework for combining multiple partitions," *The Journal of Machine Learning Research*, vol. 3, pp. 583–617, 2003.

[4] A. Topchy, A. K. Jain, and W. Punch, "A mixture model of clustering ensembles," in *Proc. SIAM Intl. Conf. on Data Mining*. Citeseer, 2004.

[5] A. Goder and V. Filkov, "Consensus clustering algorithms: Comparison and refinement." in *ALENEX*, vol. 8, 2008, pp. 109–117.

[6] X. Wang, C. Yang, and J. Zhou, "Clustering aggregation by probability accumulation," *Pattern Recognition*, vol. 42, no. 5, pp. 668–675, 2009.

[7] S. Vega-Pons, J. Correa-Morris, and J. Ruiz-Shulcloper, "Weighted partition consensus via kernels," *Pattern Recognition*, vol. 43, no. 8, pp. 2712–2724, 2010.

[8] J.-P. Barthélemy and B. Leclerc, "The median procedure for partitions," *DIMACS series in discrete mathematics and theoretical computer science*, vol. 19, pp. 3–34, 1995.

[9] S. Bickel and T. Scheffer, "Multi-view clustering." in *ICDM*, vol. 4, 2004, pp. 19–26.

[10] M. H. Law, A. P. Topchy, and A. K. Jain, "Multiobjective data clustering," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2. IEEE, 2004, pp. II–424.

[11] J. Handl and J. Knowles, "An evolutionary approach to multiobjective clustering," *Evolutionary Computation, IEEE Transactions on*, vol. 11, no. 1, pp. 56–76, 2007.

[12] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM computing surveys (CSUR)*, vol. 31, no. 3, pp. 264–323, 1999.

[13] E. E. Korkmaz, J. Du, R. Alhajj, and K. Barker, "Combining advantages of new chromosome representation scheme and multi-objective genetic algorithms for better clustering," *Intelligent Data Analysis*, vol. 10, no. 2, pp. 163–182, 2006.

[14] H. Wang, H. Shan, and A. Banerjee, "Bayesian cluster ensembles," *Statistical Analysis and Data Mining*, vol. 4, no. 1, pp. 54–70, 2011.

[15] J. Ghosh, A. Strehl, and S. Merugu, "A consensus framework for integrating distributed clusterings under limited knowledge sharing," in *Proc. NSF Workshop on Next Generation Data Mining*, 2002, pp. 99–108.

[16] A. Topchy, A. K. Jain, and W. Punch, "Clustering ensembles: Models of consensus and weak partitions," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 12, pp. 1866–1881, 2005.

[17] H.-S. Yoon, S.-Y. Ahn, S.-H. Lee, S.-B. Cho, and J. H. Kim, "Heterogeneous clustering ensemble method for combining different cluster results," in *Data Mining for Biomedical Applications*. Springer, 2006, pp. 82–92.

[18] L. Kuncheva, S. Hadjitodorov, and L. Todorova, "Experimental comparison of cluster ensemble methods," in *Information Fusion, 2006 9th International Conference on*. IEEE, 2006, pp. 1–7.

[19] C. A. C. Coello, G. B. Lamont, and D. A. Van Veldhuisen, *Evolutionary algorithms for solving multi-objective problems*. Springer, 2007.

[20] A. E. Eiben and J. E. Smith, *Introduction*. Springer, 2003.

[21] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 2, pp. 182–197, 2002.

[22] D. W. Corne, N. R. Jerram, J. D. Knowles, M. J. Oates *et al.*, "Pesa-ii: Region-based selection in evolutionary multiobjective optimization," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO2001*. Citeseer, 2001.

[23] E. Zitzler, M. Laumanns, L. Thiele, E. Zitzler, E. Zitzler, L. Thiele, and L. Thiele, "Spea2: Improving the strength pareto evolutionary algorithm," 2001.

[24] J. Handl and J. Knowles, "Evidence accumulation in multiobjective data clustering," in *Evolutionary Multi-Criterion Optimization*. Springer, 2013, pp. 543–557.

[25] S. Bandyopadhyay, U. Maulik, and A. Mukhopadhyay, "Multiobjective genetic clustering for pixel classification in remote sensing imagery," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 45, no. 5, pp. 1506–1511, 2007.

[26] L. I. Kuncheva and J. C. Bezdek, "Selection of cluster prototypes from data by a genetic algorithm," in *Proc. 5th European Congress on Intelligent Techniques and Soft Computing, Aachen, Alemanha*, 1997, pp. 1683–1688.

[27] X. L. Xie and G. Beni, "A validity measure for fuzzy clustering," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 13, no. 8, pp. 841–847, 1991.

[28] D. Dutta, P. Dutta, and J. Sil, "Simultaneous feature selection and clustering with mixed features by multi objective genetic algorithm," *International Journal of Hybrid Intelligent Systems*, vol. 11, no. 1, pp. 41–54, 2014.

[29] A. Mukhopadhyay, U. Maulik, and S. Bandyopadhyay, "Multiobjective genetic algorithm-based fuzzy clustering of categorical attributes," *Evolutionary Computation, IEEE Transactions on*, vol. 13, no. 5, pp. 991–1005, 2009.

[30] K. S. N. Ripon, C.-H. Tsang, S. Kwong, and M.-K. Ip, "Multi-objective evolutionary clustering using variable-length real jumping genes genetic algorithm," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 1. IEEE, 2006, pp. 1200–1203.

[31] D. Milne and I. H. Witten, "An open-source toolkit for mining wikipedia," *Artificial Intelligence*, 2012.

[32] Z. Wu and M. Palmer, "Verbs semantics and lexical selection," in *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 1994, pp. 133–138.

[33] E. R. Hruschka, R. J. G. B. Campello, A. A. Freitas, and A. P. L. F. De Carvalho, "A survey of evolutionary algorithms for clustering," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 39, no. 2, pp. 133–155, 2009.

[34] F. Gullo, C. Domeniconi, and A. Tagarelli, "Projective clustering ensembles," *Data Mining and Knowledge Discovery*, vol. 26, no. 3, pp. 452–511, 2013.

[35] D. Crabtree, X. Gao, and P. Andreae, "Improving web clustering by cluster selection," in *Web Intelligence, 2005. Proceedings. The 2005 IEEE/WIC/ACM International Conference on*. IEEE, 2005, pp. 172–178.

[36] L. Hubert and P. Arabie, "Comparing partitions," *Journal of classification*, vol. 2, no. 1, pp. 193–218, 1985.

[37] C. Carpineto and G. Romano, "Ambient dataset," 2008.

[38] R. Navigli and G. Crisafulli, "Inducing word senses to improve web search result clustering," in *Proceedings of the 2010 conference on empirical methods in natural language processing*. Association for Computational Linguistics, 2010, pp. 116–126.

[39] C. Carpineto and G. Romano, "Optimal meta search results clustering," in *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2010, pp. 170–177.

[40] N. Iam-on and S. Garrett, "Linkclue: A matlab package for link-based cluster ensembles," *Journal of Statistical Software*, vol. 36, no. i09, 2010.

[41] N. Iam-On, T. Boongoen, and S. Garrett, "Refining pairwise similarity matrix for cluster ensemble problem with cluster relations," in *Discovery Science*. Springer, 2008, pp. 222–233.

[42] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *Journal of the American Statistical Association*, vol. 32, no. 200, pp. 675–701, 1937.