A Framework for Multi-objective Clustering and its Application to Co-location Mining

Rachsuda Jiamthapthaksin, Christoph F. Eick and Ricardo Vilalta

Computer Science Department, University of Houston,
Houston, TX 77204-3010, USA
{rachsuda, ceick, vilalta}@cs.uh.edu

Abstract. The goal of multi-objective clustering (MOC) is to decompose a dataset into similar groups maximizing multiple objectives in parallel. In this paper, we provide a methodology, architecture and algorithms that, based on a large set of objectives, derive interesting clusters regarding two or more of those objectives. The proposed architecture relies on clustering algorithms that support plug-in fitness functions and on multi-run clustering in which clustering algorithms are run multiple times maximizing different subsets of objectives that are captured in compound fitness functions. MOC provides search engine type capabilities to users, enabling them to query a large set of clusters with respect to different objectives and thresholds. We evaluate the proposed MOC framework in a case study that centers on spatial co-location mining; the goal is to identify regions in which high levels of Arsenic concentrations are co-located with high concentrations of other chemicals in the Texas water supply.

Keywords: Multi-objective clustering, knowledge discovery, spatial data mining, co-location mining, clustering with plug-in fitness functions.

1 Introduction

The goal of clustering is to group similar objects into clusters while separating dissimilar objects. Although clustering is a very popular data mining technique which has been used for over 40 years, its objectives and how to evaluate different clustering results is still subject to a lot of controversy. For example, Saha et al. observe "*evidently one reason for the difficulty of clustering is that for many data sets no unambiguous partitioning of the data exists, or can be established, even by humans*" [1]. Moreover, matters are even worse for applications that seek of non-homogeneous groupings, because "*virtually all existing clustering algorithms assume a homogeneous clustering criterion over the entire feature space … because its intrinsic criterion may not fit well with the data distribution in the entire feature space*" [2]; furthermore, "*by focusing on just one aspect of cluster quality, most clustering algorithms … are not robust to variations in cluster shape, size, dimensionality and other characteristics*" [3]. Finally, in specific application domains, users seek for clusters with similar extrinsic characteristics; their definitions of "interestingness" of clusters are usually different from those used in typical clustering algorithms.

Consequently, clusters obtained by typical clustering algorithms frequently do not capture what users are really looking for.

The goal of this paper is to alleviate the problems that have been identified in the previous paragraph; in particular, a novel framework for multi-objective clustering is introduced and evaluated. The goal of multi-objective clustering is to decompose a dataset into similar groups maximizing multiple objectives in parallel. Multi-objective clustering can be viewed as a special case of multi-objective optimization which aims to simultaneously optimize trade-off among multiple objectives under certain constraints. In this paper, we provide a methodology, architecture and algorithms that, based on a large set of objectives, derive clusters that are interesting with respect to two or more of those objectives. The proposed architecture relies on clustering algorithms that support plug-in fitness functions and on multi-run clustering in which clustering algorithms are run multiple times maximizing different subsets of objectives that are captured in compound fitness functions. Using different combinations of single objective functions a cluster repository is created; final clusterings are created by querying the repository based on user preferences.

In the following, we discuss an example to better illustrate what we are trying to accomplish. Let us assume we like to assist a travel agent in finding different, interesting places (spatial clusters) with respect to a large set of objectives his customers are interested in, such as quality and price of hotels, quality of hiking, desirable weather patterns, acceptable crime rates, etc. This set of objectives Q will be captured in reward functions $reward_q$ each of which corresponds to a single objective q; moreover, for each objective $q \in Q$ a minimum satisfaction threshold θ_q is provided. Our proposed framework identifies the "best" places for our travel agent for which two or more objectives[1] in Q are satisfied. These places will be stored in a cluster repository. However, if places are overlapping certain dominance relations should be enforced and only non-dominated places will be stored in the repository. For example, if there is a region in South Texas that satisfies objectives $\{A,B\}$ and a slightly different region satisfying objectives $\{A,B,C\}$, then only the later region will be reported; similarly, if two overlapping regions satisfy the same objectives, but the rewards of one region are higher, only that region should be reported. As we will further discuss in section 2, dominance relations between clusters can be quite complicated, because the scope of a cluster has to be considered. For example, a cluster in South Texas is disjoint in scope from a cluster in West Texas; consequently, no dominance relation exists between these two regions. Finally, dealing with clusters with highly overlapping scope when creating the final clustering poses another challenge: for example, if we have two highly similar clusters, one satisfying objectives $\{A,B\}$ and the other satisfying objectives $\{B,C\}$: should we include both clusters in the final results; or—if we decided not to do so—how should we select the cluster to be included in the final clustering? In summary, the ultimate vision of this work is to develop a system that, given a large set of objectives, automatically identifies the "best" clusters that satisfy a large number of objectives; moreover, different clusters will usually serve quite different objectives.

[1] A cluster x is potentially interesting with respect to objective q, if its reward value is larger than q's satisfaction threshold: $reward_q(x) > \theta_q$.

An idea employed by our approach is to run a clustering algorithm with a plug-in compound fitness function multiple times to generate clusters that maximize sets of objectives captured in the compound fitness function. Our clustering approach to cope with multi-objective problem is similar to the ensemble clustering approach as it is based on the key hypothesis that better clustering results can be obtained by combining clusters that originate from multiple runs of clustering algorithms [4]. However, our approach is an incremental approach that collects and refines clusters on the fly, and the search for alternative clusters takes into consideration what clusters already have been generated, rewarding novelty.

The rest of our paper is organized as follows: Section 2 proposes an architecture and algorithms for multi-objective clustering. Section 3 demonstrates the proposed system for a co-location mining case study. Section 4 discusses related work and Section 5 summarizes our findings.

2 An Architecture and Algorithms for Multi-run Clustering

A main goal of multi-objective clustering (MOC) is to find individual clusters that are good with respect to multiple objectives; due to the nature of MOC only clusters that are good with respect to at least two objectives are reported. In the remainder of this section we focus on a specific architecture and algorithms for MOC. The following features are of major importance for the proposed architecture: the use of clustering algorithms that support plug-in fitness/reward functions, the capability to create compound fitness functions to instruct clustering algorithms to seek for clusters that are good with respect to multiple objectives, the use of a repository M that stores clusters that are potentially interesting, the use of a multi-objective dominance relation to determine what clusters should be stored in M, and the availability of a cluster summarization tool that creates final clusterings based on user preferences from the clusters in the repository M.

2.1 Building Blocks for Multi-objective Clustering

A. Clustering algorithms that support plug-in fitness and reward functions.
In this paper, we assume that $Q=\{q_1, q_2, \ldots, q_z\}$ is the set of objectives that multi-objective clustering maximizes. For each objective $q \in Q$ a reward function $Reward_q$ has to be provided that measures to which extent the objective q is satisfied—higher rewards mean better clusters with respect to objective q. Moreover, reward thresholds $\theta_{q1},\ldots,\theta_{qz}$ are associated with each reward function. If $Reward_q(x)>\theta_q$ holds, we say that "*cluster x satisfies objective q"*. In general, the goal of MOC is to seek for clusters that satisfy a large number of objectives in Q, but rarely all objectives in Q; different clusters usually serve different objectives.

Our approach employs clustering algorithms that support plug-in fitness function; given a dataset $O=\{o_1,\ldots,o_n\}$, the algorithm seeks for a clustering $X=\{x_1,\ldots,x_k\}$ that maximizes a plug-in fitness function q:

$$q(X) = \sum_{i=1}^{k} Reward_q(x_i)\,. \quad (1)$$

subject to: $x_i \cap x_j = \emptyset$ for $i \neq j$, $x_i \subseteq O$ for $i = 1, \ldots, k$ and $\bigcup_{i=1}^{k} x_i \subseteq O$.

A family of clustering algorithms that support such fitness functions (CLEVER [5], SCMRG [6], and MOSAIC [7]) has been designed and implemented in our past research. Our approach measures the quality of a clustering $X=\{x_1,..,x_k\}$ as the sum of rewards obtained for each cluster x_i (i=1,…,k) using the reward function $Reward_q$. Additionally, reward functions are used in our multi-objective clustering approach to determine dominance and when creating the final clustering. Reward functions typically correspond to popular cluster evaluation measures, such as entropy or compactness.

B. The role of the cluster repository *M*

Our approach runs clustering algorithms multiple times with the same or different reward/fitness functions and stores the potentially interesting clusters in a cluster list *M*. Each time a new clustering *X* is obtained, *M* is updated; some clusters in *X* might be inserted into *M*, and some clusters in *M* might have to be deleted due to the arrival of better clusters in *X*. Only non-dominated, multi-objective clusters are stored in *M*. We will define what clusters *M* can contain more formally next.

Definition 1. *x* is a multi-objective cluster with respect to a set of objectives *Q*:

$$MO\_Cluster(x,Q) \Leftrightarrow \exists q \in Q\ \exists q' \in Q(q \neq q' \wedge Reward_q(x) \geq \theta_q \wedge Reward_{q'}(x) \geq \theta_{q'})\,. \quad (2)$$

Definition 2. *x* dominates *y* with respect to *Q*:

$$Dominates(x,y,Q) \Leftrightarrow \forall q \in Q\ ((Reward_q(x) \geq Reward_q(y) \vee Reward_q(x) < \theta_q \wedge Reward_q(y) < \theta_q) \wedge Similarity(x,y) \geq \theta_{sim})\,. \quad (3)$$

Definition 2 introduces dominance relations between clusters. It is also important to observe that if *x* and *y* are both bad clusters with respect to a single objective *q*, the rewards associated with objective *q* are not used to determine dominance between *x* and *y*; in general, we only compare *x* and *y* based on those objectives that at least one of them satisfies. Moreover, the above definition assumes that clusters *x* and *y* have an agreement in their scope to make them comparable; a user-defined similarity threshold θ_{sim} has to be provided for this purpose. In our current work, similarity between two clusters *x* and *y* is assessed (|*c*| returns the cardinality of set *c*) as follows:

$$Similarity(x,y) = |x \cap y| / |x \cup y|\,. \quad (4)$$

It takes the ratio of the number of common objects between x and y over the total number of objects in x and y.

In the following, we use the symbol '$\rangle$' to express dominance relationships between clusters:

$$x \rangle y \Leftrightarrow Dominates(x,y,Q) . \quad (5)$$

In general, M should only store non-dominated clusters, and algorithms that update M should not violate this constraint; that is:

$$m \in M \Rightarrow \sim \exists m' \in M \; m' \rangle m . \quad (6)$$

2.2 The Proposed MOC Framework

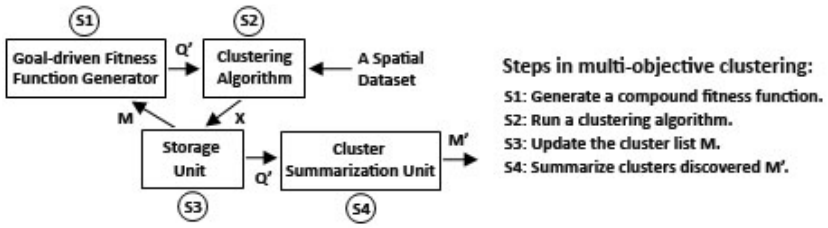


Fig. 1. An architecture of multi-objective clustering

The architecture of the MOC system that we propose is depicted in Fig. 1; it consists of 4 main components: a clustering algorithm, storage unit, goal-driven fitness function generator and cluster summarization unit. MOC is performed as follows: First, the goal-driven fitness function generator (*FG*) selects a new fitness function for the clustering algorithm (*CA*), which generates a new clustering X in the second step. Third, the storage unit (*SU*) updates its cluster list M using the clusters in X. The algorithm iterates over these three steps until a large number of clusters has been obtained. Later, in the fourth step, the cluster summarization unit (*CU*) produces final clusters based on user preferences which are subsets of the clusters in M. Details of algorithms proposed to serve the individual tasks are given in the following discussions.

Preprocessing step. The objective of this step is to obtain simple statistics for individual fitness functions for a given dataset. Results obtained from this step will be used to determine reward thresholds, $\theta_{q1},\ldots,\theta_{qz}$. Our current implementation uses percentiles to determine the satisfaction threshold for a particular reward function $Reward_q$. Alternatively, the thresholds could be acquired from a domain expert.

Step 1: Generate a compound fitness function *Q'*. *FG* selects a subset Q'(⊂*Q*) of the objectives *Q* and creates a *compound fitness function* $q_{Q'}$ relying on a penalty function approach [8] that is defined as follows:

$$q_{Q'}(X) = \sum_{i=1}^{k} CmpReward(x_i)\ . \quad (7)$$

$$CmpReward(x) = \sum_{q \in Q'} (Reward_q(x) * Penalty(Q', x))\ . \quad (8)$$

Penalty(*Q'*,*x*) is a penalty function that returns 1 if *x* satisfies all objectives in *Q'*, but returns a smaller number in the case that some objectives in *Q'* are not satisfied. In general, $q_{Q'}$ sums the rewards for all objectives $q \in Q'$; however, it gives more reward to clusters x_i that satisfy all objectives in order to motivate the *CA* to seek for multi-objective clusters. Our current implementation uses 2-objective compound fitness functions $q_{Q'}$ with $Q'=\{q,q'\}$ in conjunction with the following penalty function:

$$Penalty(\{q, q'\}, x) = \begin{cases} 1 & Reward_q(x) \geq \theta_q \wedge Reward_{q'}(x) \geq \theta_{q'} \\ 0.5 & Otherwise\ . \end{cases} \quad (9)$$

Step 2: Run the *CA* with a compound fitness function $q_{Q'}$ to generate a clustering *X* maximizing *Q'*.

```
FOR ALL x∈X' DO
  IF ~MO_Cluster(x) ∨ ∃m∈M m⟩x THEN
    Discard x;
  ELSE
    Let D={m∈M | x⟩m};
    Insert x into M;
    Delete all clusters in D from M;
```

Fig. 2. Update_M_by_X algorithm

Step 3: Update the storage unit *M* using the obtained clustering *X*. To accomplish this task, we introduce the *Update_M_by_X* algorithm as specified in Fig. 2. The algorithm considers multi-objective dominance as discussed in Section 2.1. In a nutshell, the algorithm selectively inserts "good" clusters with respect to two or more objectives and makes sure that all clusters in *M* are non-dominated.

Step 4: Create a final clustering from *M*. The cluster summarization unit retrieves a subset of clusters *M'* from *M* based on user preferences. In this paper, we introduce an algorithm called *MO-Dominance-guided Cluster Reduction algorithm* (MO-DCR), whose pseudocode is given in Fig. 3. MO-DCR returns a subset of interesting clusters, based on the following two user-defined input parameters:

1. $\hat{Q} \subset Q$ and reward thresholds θ_q for each $q \in \hat{Q}$

2. A cluster removal similarly threshold θ_{rem}. (Basically, if two clusters have too much overlap, one is not included in the final clustering.)

The goal of the algorithm is to return a clustering that is good with respect to $\hat{Q}$ selected by a user, and to remove clusters that are highly overlapping. The algorithm iteratively performs two steps:

1. Identify multi-objective dominant clusters with respect to $\hat{Q}$, and
2. Remove dominated clusters which are in the θ_{rem}-neighborhood of a dominant cluster.

Let
DEDGE:={$(c1,c2)|c1\in M \wedge c2\in M \wedge sim(c1,c2)>\theta_{rem} \wedge better(c2,c1)$}
REMCAND:={$c|\exists d\ (c,d)\in$ DEDGE}
DOMINANT:={$c|\exists d\ (d,c)\in$ DEDGE $\wedge\ c\notin$ REMCAND}
REM:={$c|\exists d\ ((c,d)\in$ DEDGE $\wedge\ d\in$ DOMINANT)}
$Better(c1,c2)\leftrightarrow \forall q\in\hat{Q}, Reward_q(c1)>Reward_q(c2) \vee$
$(Reward_q(c1)=Reward_q(c2) \wedge$
$clusterNumber(c1)>clusterNumber(c2))$
Remark: Ties have to be broken so that DEDGE is always a DAG; no cycles in DEDGE are allowed to occur.

Input: M, $\hat{Q}$, θ_q *for each* $q\in\hat{Q}$
Output: $M'\subseteq M$

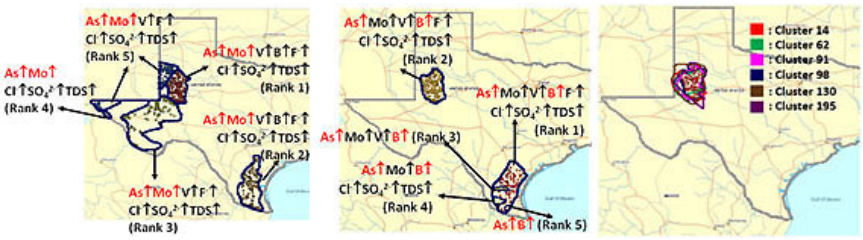
Remove $\{c\in M \mid \exists q\in\hat{Q}\ Reward_q(c)<\theta_q\}$
"Remove bad clusters with respect to $\hat{Q}$."
Compute DEDGE from M;
Compute REMCAND;
Compute DOMINANT;
WHILE true DO
{
Compute REM;
IF REM=$\emptyset$ THEN EXIT ELSE $M=M$/REM;
Update DEDGE by removing edges of deleted clusters in REM;
Update REMCAND based on DEDGE;
Update DOMINANT based on DEDGE and REMCAND;
}
RETURN(M);

Fig. 3. MO-Dominance-guided Cluster Reduction algorithm (MO-DCR)

3 Experimental Results

In this section, we demonstrate the benefits of the proposed multi-objective clustering framework in a real world water pollution case study; the goal of the study is to obtain a better understanding of what causes high arsenic concentrations to

occur. In this section, we report on experiments that use multi-objective clustering to identify regions in Texas where high level of Arsenic concentrations are in close proximity with high concentrations of other chemicals. For instance, the Rank3 region in Fig. 5b is found by the framework with an associated co-location pattern As↑Mo↑V↑B↑, indicating that high Arsenic concentrations are co-located with high Molybdenum, Vanadium and Boron concentrations in this region. TWDB has monitored water quality and collected the data for 105,814 wells in Texas over last 25 years. For this experiment we used a water well dataset called Arsenic_10_avg [9] which was created from a database provided by the Texas Water Development Board (TWDB) [10]. In this paper, we use a subset of this dataset containing 3 spatial attributes longitude, latitude and aquifer and the following 8 chemical concentrations for each water well: Arsenic (As), Molybdenum (Mo), Vanadium (V), Boron (B), Fluoride (F^-), Chloride (Cl^-), Sulfate (SO_4^{2-}) and Total Dissolved Solids (TDS).



(a) (b) (c)

Fig. 5. Visualization of experimental results: (a) and (b) are the top 5 regions ordered by rewards using user-defined query {As↑,Mo↑} and {As↑,B↑}, respectively, and (c) is overlay of similar regions in the storage unit located in the Southern Ogallala aquifer.

We used CLEVER (CLustEring using representatives and Randomized hill climbing), introduced in [5], as the clustering algorithm in the experiments. In a nutshell, CLEVER is a prototype-based clustering algorithm that seeks for a clustering X maximizing a plug-in fitness function $q(X)$. A single-objective fitness function for regional co-location mining has been introduced in [5]. In the following, we will reformulate this problem as a multi-objective clustering problem.

Let O be a dataset

$x \subseteq O$ be a cluster, called a region in the following

$o \in O$ be an object in the dataset O

$N=\{A_1,\ldots,A_m\}$ be the set of non-geo-referenced continuous attributes in the dataset O

$Q=\{A_1\uparrow, A_1\downarrow, \ldots, A_m\uparrow, A_m\downarrow\}$ be the set of possible base co-location patterns

$B \subseteq Q$ be a set of co-location patterns

z-score(A,o) be the z-score of object o's value of attribute A

$$z(A\uparrow, o) = \begin{cases} z\text{-}score(A,o) \text{ if } z\text{-}score(A,x) > 0 \\ 0 \qquad\qquad otherwise. \end{cases} \tag{10}$$

$$z(A\downarrow,o)=\begin{cases}-z\text{-}score(A,o)\ if\ z\text{-}score(A,x)<0\\ 0 \qquad\qquad otherwise.\end{cases} \quad (\mathbf{11})$$

z(p,o) is called the *z-value* of base pattern $p\in Q$ for object o in the following. The interestingness of an object o with respect to a co-location set $B\subseteq Q$ is measured as the product of the z-values of the patterns in the set B. It is defined as follows:

$$i(B,o)=\prod_{p\in B} z(p,o)\,. \quad (\mathbf{12})$$

In general, the interestingness of a region can be straightforwardly computed by taking the average interestingness of the objects belonging to a region; however, using this approach some very large products might dominate interestingness computations; consequently, our additionally considers purity when computing region interestingness, where *purity*(*B,x*) denotes the *percentage of objects* $o\in c$ *for which* $i(B,o)>0$. The interestingness $\varphi(B,x)$ of a region x with respect to a co-location set B is measured as follows:

$$\varphi(B,x)=\frac{\sum_{o\in c} i(B,o)}{|x|}\times purity(B,x)^{\tau}\,. \quad (\mathbf{13})$$

The parameter $\tau\in[0,\infty)$ controls the importance attached to purity in interestingness computations; $\tau=0$ implies that purity is ignored, and using larger value increases the importance of purity. Finally, the reward of the region x is computed as follows:

$$Reward_B(x)=\varphi(B,x)\times|x|^{\beta}. \quad (\mathbf{14})$$

where $|x|$ denotes the number of objects in the region and β is a parameter that controls how much premium is put on the number of objects in a region. Finally, $Reward_B$ is used to define a plug-in fitness function q_B as follows (see also section 2):

$$q_B(X)=q_B(\{x_1,\ldots,x_k\})=\sum_{i=1}^{k}\left(Reward_B(x_i)\right). \quad (\mathbf{15})$$

In the experiment, we use 7 different objective functions: $q_{\{As\uparrow,Mo\uparrow\}}$, $q_{\{As\uparrow,V\uparrow\}}$, $q_{\{As\uparrow,B\uparrow\}}$, $q_{\{As\uparrow,F^-\uparrow\}}$, $q_{\{As\uparrow,Cl^-\uparrow\}}$, $q_{\{As\uparrow,SO4^{2-}\uparrow\}}$, $q_{\{As\uparrow,TDS\uparrow\}}$. We are basically interested in finding regions for which at least two of those functions are high; in other words, regions in which high arsenic concentrations are co-located with high concentrations of two or more other chemicals. We claim that the MOC approach for regional co-location mining is more powerful than our original approach [5] that uses a single-objective function that relies on maximum-valued patterns: $max_B(\varphi(B,x))$ is used to assess interestingness for a region—which ignores alternative patterns: for example, if A is co-located with B,C in one region and with D,E in another region and the two regions overlap, the original approach will not be able to report both regions.

In the following, we report results of an experiments in which MOC was used with the following parameter setting for the fitness function we introduced earlier: $\tau=1$,

β=1.5. In the MOC preprocessing step, we obtain the reward thresholds at 40 percentile based on the fitness distribution of the individual fitness functions: $\theta_{q\{As\uparrow,Mo\uparrow\}}$=0.59, $\theta_{q\{As\uparrow,V\uparrow\}}$=2.91, $\theta_{q\{As\uparrow,B\uparrow\}}$=0.27, $\theta_{q\{As\uparrow,F^-\uparrow\}}$=2.87, $\theta_{q\{As\uparrow,Cl^-\uparrow\}}$=0.53, $\theta_{q\{As\uparrow,SO4^{2-}\uparrow\}}$=1.35 and $\theta_{q\{As\uparrow,TDS\uparrow\}}$=1.51. After finishing iterative process in MOC which exploring all pairs of the 7 fitness functions, out of original generated 1,093 clusters, only 227 clusters were selectively stored in the repository.

Regarding the last step in MOC, we set up a user defined reward thresholds to create a final clustering to: $\theta_{q\{As\uparrow,Mo\uparrow\}}$=13, $\theta_{q\{As\uparrow,V\uparrow\}}$=15, $\theta_{q\{As\uparrow,B\uparrow\}}$=10, $\theta_{q\{As\uparrow,F^-\uparrow\}}$=25, $\theta_{q\{As\uparrow,Cl^-\uparrow\}}$=7, $\theta_{q\{As\uparrow,SO4^{2-}\uparrow\}}$=6, $\theta_{q\{As\uparrow,TDS\uparrow\}}$=8, and set the removal threshold θ_{rem}=0.1 to seek for nearly non-overlapping clusters. Examples of the top 5 regions and patterns with respect to two queries: query$_1$={As↑,Mo↑} and query$_2$={As↑,B↑} are shown in Fig. 5a and 5b, respectively. In general, query patterns are used to select and sort the obtained regions; that is, regions dissatisfying {As↑,Mo↑} would not be included in the result for query$_1$. The visualizations associate the chemicals whose co-location strengths are above the threshold with each visualized region. For instance, for the query {As↑,Mo↑}, all of the top 5 regions retrieved contain patterns whose length is 5 or more. It can be observed that the Rank1 region in Fig. 5a significantly overlaps with the Rank2 region in Fig. 5b and share the same co-location sets: different regions are reported to better serve the different interests expressed by the two queries. Moreover, as depicted in Fig. 5b MOC is also able to identify nested clusters (i.e. the regions ranked 3-5 are sub-regions of the Rank1 region), and particularly discriminate among companion elements, such as Vanadium (Rank3 region), or Chloride, Sulfate and Total Dissolved Solids (Rank4 region). In this particular case, the regions ranked 3-5 better serve specific objectives, whereas the Rank1 region satisfies a larger set of objectives; that is, there is no dominance relationship between the Rank1 region and the regions ranked 3-5. In general, in the experiment a large number of overlapping regions without any dominance relationships were identified in the Southern Ogallala aquifer, as depicted in Fig. 5c, which is a hotspot for arsenic pollution in Texas.

4 Related Work

Multi-objective clustering is considered a specific field of multi-objective optimization (MOO) whose goal is to simultaneously optimize trade-off between two or more objectives under certain constraints. According to our investigation, there are two approaches coping with multi-objective clustering: multi-objective evolutionary algorithms (MOEA) and dual clustering. MOEA have been widely used in MOO such as, NGSA-II [11] and PESA-II [12]. Their searching strategy, which automatically generates and preserves a set of diverse solutions, is desirable for this type of problem. In particular to multi-objective clustering, such MOEA are adapted to solve the multiple objectives clustering problem. The general idea is that by using clustering validation measures as the objective functions, the algorithm iteratively evolves clusters from one generation to another to improve quality as well as to explore diversity of cluster solutions. Handl and Knowles introduced VIENNA, an adaptive version of PESA-II EA incorporating specialized mutation and initialization procedures [3]. The algorithm employs two following internal measures to estimate

clustering quality: overall deviation and connectivity. Such clustering quality measures have also been used in many other MOEA, e.g. MOCK [13] and MOCLE [14]. Finally, work by Molina et al. [15] employ scatter tabu search for non-linear multi-objective optimization which can potentially be utilized for multi-objective clustering. Dual clustering is another approach for multi-objective clustering [16]. It makes use of both clustering and classification algorithms to generate and to refine a clustering iteratively serving multiple objectives.

Our approach differs from those two approaches in that it seeks for good individual clusters maximizing multiple objectives that are integrated into a single clustering by a user-driven post-processing step. The proposed post processing algorithm operates like a search engine that allows users to query a large set of clusters with respect to different objectives and thresholds, obtaining a final clustering from the viewpoint of a single or a small set of objectives that are of particular interest for a user.

5 Conclusion

The paper centers on multi-objective clustering; in particular, we are interested in supporting applications in which a large number of diverse, sometimes contradictory objectives are given and the goal is to find clusters that satisfy large subsets of those objectives. Applications that need such capabilities include recommender systems, constraints satisfaction problems that involve a lot of soft constraints, complex design problems, and association analysis. Although the deployment of such systems is highly desirable, they are still quite far away from becoming commercial reality, because of the lack of useful research in this area.

The main contribution of this paper is to provide building blocks for the development of such systems. In particular, we proposed novel dominance relation for the case when we have a lot of objectives, but it is impossible to accomplish many of them. A second building block are clustering algorithms that support plug-in fitness functions, and the capability to construct compound fitness functions when a small set of objectives has to be satisfied. The third building block is a system architecture in which a large repository of clusters will be generated initially based on a given set of objectives, relying on multi-run clustering, dominance-relations, and compound fitness functions. The repository can be viewed as a meta-clustering with respect to all objectives investigated. Specific clusterings are generated by querying the repository based on particular user preferences and objective satisfaction thresholds. The fourth building block is the domain-driven nature of our approach in which users can express clustering criteria based on specific domain needs, and not based on highly generic, domain-independent objective functions which is the approach of most traditional clustering algorithms. Finally, we provided evidence based on a case study that multi-objective clustering approach is particularly useful for regional co-location mining, for which it is very difficult to formalize the problem using a single objective due to the large number of co-location patterns.

However, using the MOC approach creates new challenges for co-location mining: 1) a very large repository (containing more than 100,000 clusters) of highly overlapping, potentially interesting clusters has to be maintained and queried

efficiently, and 2) coming up with sophisticated summarization strategies that extract clusters from the repository based on user preferences and possibly other user inputs is very challenging task. Our current summarization algorithm is only one of many possible solutions to this problem.

Acknowledgments. This research was supported in part by a grant from the Environmental Institute of Houston (EIH).

References

1. Saha, S., Bandyopadhyay, S.: A New Multiobjective Simulated Annealing Based Clustering Technique Using Stability And Symmetry. In: 19th International Conference on Pattern Recognition. (2008)
2. Law, H.C.M., Topchy, A.P., Jain, A.K.: Multiobjective Data Clustering. In: IEEE Conputer Society Conference on Computer Vision and Pattern Recognition. (2004)
3. Handl, J., Knowles, J.: Evolutionary Multiobjective Clustering. In: Parallel Problem Solving from Nature-PPSN VIII. 3242. 1081--1091. Springer, Berlin/Heidelberg (2004)
4. Jiamthapthaksin, R., Eick, C.F., Rinsurongkawong, V.: An Architecture and Algorithms for Multi-Run Clustering. In IEEE Computational Intelligence Symposium on Computational Intelligence and Data Mining. (2009)
5. Eick, C.F., Parmar, R, Ding, W., Stepinki, T., Nicot, J.-P.: Finding Regional Co-location Patterns for Sets of Continuous Variables in Spatial Datasets. In: 16th ACM SIGSPATIAL International Conference on Advances in GIS. (2008)
6. Eick, C.F., Vaezian, B., Jiang, D., Wang, J.: Discovery of Interesting Regions in Spatial Datasets Using Supervised Clustering. In: 10th European Conference on Principles and Practice of Knowledge Discovery in Databases. (2006)
7. Choo, J., Jiamthapthaksin, R., Chen, C.-S., Celepcikay, O.C., Giusti, Eick, C.F.: MOSAIC: A Proximity Graph Approach to Agglomerative Clustering: In: 9th International Conference on Data Warehousing and Knowledge Discovery. (2007)
8. Baeck, T., Fogel, D.B., Michalewicz, Z.: Chapter 7 Penalty functions, Evolutionary computation 2, Advanced algorithms and operators. Institute of Physics Publishing, Philadelphia (2000)
9. Data Mining and Machine Learning Group website, University of Houston, Texas, http://www.tlc2.uh.edu/dmmlg/Datasets
10. Texas Water Development Board, http://www.twdb.state.tx.us/home/index.asp
11. Deb, K., Pratap, A., Agrawal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. J. Evolutionary Computation. 6, 182--197 (2002)
12. Corne, D.W., Jerram, N.R., Knowles, J.D., Oates, M.J.: PESA-II: Region-based Selection in Evolutionary Multiobjective Optimization. In: Genetic and Evolutionary Computation Conference. 283--290. (2001)
13. Handl, J., Knowles, J.: An Evolutionary Approach to Multiobjective Clustering. J. Evolutionary Computation. 11, 56--57 (2007)
14. Faceli, K., de Carvalho, A.C.P.L.F., de Souto, M.C.P.: Multi-Objective Clustering Ensemble. J. Hybrid Intelligent Systems. 4, 145--146 (2007)
15. Molina, J., Laguna, M., Martí, R., Caballero, R.: SSPMO: A Scatter Search Procedure for Non-Linear Multiobjective Optimization. INFORMS J. Computing. 19, 91--100 (2007)
16. Lin, C-R., Liu, K-H., Chen M-S.: Dual Clustering: Integrating Data Clustering over Optimization and Constraint Domains. J. Knowledge and Data Engineering. 17 (2005)