Correspondence Clustering: An Approach to Cluster Multiple Related Spatial Datasets

Vadeerat Rinsurongkawong and Christoph F. Eick

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

Abstract. Domain experts are frequently interested to analyze multiple related spatial datasets. This capability is important for change analysis and contrast mining. In this paper, a novel clustering approach called correspondence clustering is introduced that clusters two or more spatial datasets by maximizing cluster interestingness and correspondence between clusters derived from different datasets. A representative-based correspondence clustering framework and clustering algorithms are introduced. In addition, the paper proposes a novel cluster similarity assessment measure that relies on re-clustering techniques and co-occurrence matrices. We conducted experiments in which two earthquake datasets had to be clustered by maximizing cluster interestingness and agreement between the spatial clusters obtained. The results show that correspondence clustering can reduce the variance inherent to representative-based clustering algorithms, which is important for reducing the likelihood of false positives in change analysis. Moreover, high agreements could be obtained by only slightly lowering cluster quality.

Keywords: Spatial Data Mining, Mining Related Spatial Datasets, Variance in Clustering, Representative-based Clustering Algorithms, Change Analysis.

1 Introduction

Domain experts are frequently interested to compare clustering results of two or more related datasets. For example, meteorologists may want to understand the change in this year's sea water temperature patterns with respect to those observed in previous years. Zoologists may attempt to relate animals' habitats and their source of foods. We can use traditional clustering algorithms to cluster each dataset separately and compare the results, but this approach frequently will not lead to good results due to the following reasons:

1. The clustering algorithms do not take into consideration the correspondences between the datasets.
2. The randomness inherent in most clustering algorithms further complicates the correspondence analysis of clusterings. For example, K-means is sensitive to initialization, noise, and outliers.

M.J. Zaki et al. (Eds.): PAKDD 2010, Part I, LNAI 6118, pp. 216–227, 2010.

In this paper, we introduce a novel spatial clustering approach called *correspondence clustering*. Correspondence clustering clusters two or more spatial datasets by taking the correspondence between the different clustering into consideration. Therefore, the obtained clusterings relate to one another; that is, the clustering of one dataset depends on the clusterings of the other datasets. Consequently, variances in clusterings produced by traditional clustering algorithms are reduced. Moreover, the hidden relationships between related clusterings can be discovered.

Applications for correspondence clustering include:

1. Change analysis [7] where changes between two datasets are compared; correspondence clustering reduces the likelihood of identifying false change patterns by enhancing agreement between the clustering results for different datasets.
2. Regional co-location mining [2] that seeks for regions in which two types of events are co-located; for example, correspondence clustering can find regions where deep and severe earthquakes co-locate.
3. Correspondence clustering can help dealing with missing values. For example, when identifying clusters with high ozone concentration, failures of ozone measurement equipments result in missing values. Correspondence clustering can use past clusterings to guide clustering when missing values are present.

Challenges to develop a good correspondence clustering framework include:

1. Techniques have to be developed to deal with the variance inherent to most clustering algorithms. If it is not dealt properly, false changes, and false novelties will be detected.
2. Methods have to be developed that measure the correspondence between two clusterings which is a non-trivial problem if object identity is not known.
3. Clustering algorithms have to be extended to cluster multiple datasets jointly considering cluster agreement or other relationships between the clustered datasets.
4. Measuring cluster correspondence is usually quite expensive, and efficient techniques have to be developed to keep its overhead in check.

The main contributions of the presented paper include:

1. A unique framework for correspondence clustering is introduced.
2. Representative-based correspondence clustering algorithms that allow for plug-in fitness functions are introduced.
3. Novel techniques that measure the agreement between two clusterings are proposed.

2 Correspondence Analysis Framework

In this section, we propose a correspondence analysis framework. Basically, our framework is designed for spatial data, especially for geo-referenced datasets. The challenges of discovering interesting patterns in spatial data include the complexity of spatial data types, the presence of hidden spatial relationships, and spatial autocorrelation. Moreover, spatial space is continuous and contains many patterns at different levels of granularities.

Let us assume that a set of related spatial datasets $O=\{O_1,...,O_n\}$ are given. We are interested in finding interesting relationship among these datasets. In general, our framework seeks for clustering results that maximize two objectives: (1) the interestingness in each clustering, (2) the correspondence between the set of obtained clusterings. Correspondence clustering is defined as follows.

Definition 1. A correspondence clustering algorithm clusters data in two or more datasets $O=\{O_1,...,O_n\}$ and generates clustering results $X=\{X_1,...,X_n\}$ such that for $1 \leq i \leq n$, X_i is created from O_i and the correspondence clustering algorithm seeks for X_i's such that each X_i maximizes interestingness $i(X_i)$ with respect to O_i as well as maximizes the correspondence $Corr(X_1,...,X_n)$ between itself and the other clusterings X_j for $1 \leq j \leq n, j \neq i$.

In summary, correspondence clustering can be viewed as a multi-objective optimization problem in which the interestingness of clustering and their correspondence are maximized. Moreover, different interestingness functions *i* and correspondence functions *Corr* can be used for different correspondence clustering tasks. In the next section, a representative-based correspondence clustering approach is introduced. The approach allows for plug-in fitness functions that are capable to capture different interestingness functions *i* and correspondence functions *Corr*.

3 Representative-Based Correspondence Clustering Algorithms

Since our representative-based correspondence clustering approach employs a region discovery framework, we first introduce the region discovery framework.

3.1 Region Discovery Framework

The region discovery framework [1] gears towards finding scientifically interesting places in spatial datasets. The framework adapts clustering algorithms for the task of region discovery by allowing plug-in fitness functions to support variety of region discovery applications corresponding to different domain interests. The goal of region discovery is to find a set of regions *X* that maximize an externally given fitness function $q(X)$; q is assumed to have the following structure:

$$q(X) = \sum_{c \in X} reward(c) = \sum_{c \in X} i(c) \times |c|^{\beta} \tag{1}$$

where $i(c)$ is the interestingness of a region c and $|c|$ is the number of objects belonging to a region c is denoted by $|c|$. The reward associated with region sizes is controlled by parameter β ($\beta>1$).

3.2 Representative-Based Correspondence Clustering Algorithms

In general, representative-based clustering algorithms seek for a subset of the objects in the dataset—called the "*representatives*"—and form clusters by assigning the remaining objects to the closest representative. In this section, representative-based correspondence clustering algorithms are introduced. The proposed algorithms are modifications of a region discovery algorithm named CLEVER [2]. CLEVER is a representative-based clustering algorithm that applies randomized hill climbing to maximize the fitness function q. Figure 1 gives the pseudo-code of CLEVER.

```
Inputs: Dataset $O$, $k'$, neighborhood-size, $p$, $p'$,$\beta$
Outputs: Clustering $X$, fitness $q$
Algorithm:
  1. Create a current solution by randomly selecting $k'$ representatives from $O$.
  2. Create $p$ neighbors of the current solution randomly using the given neighborhood
     definition.
  3. If the best neighbor improves the fitness $q$, it becomes the current solution. Go back
     to step 2.
  4. If the fitness does not improve, the solution neighborhood is re-sampled by
     generating $p'$ more neighbors. If re-sampling does not lead to a better solution,
     terminate returning the current solution; otherwise, go back to step 2 replacing the
     current solution by the best solution found by re-sampling.
```

Fig. 1. Pseudo-code of CLEVER

Given two datasets O_1 and O_2, the goal of correspondence clustering is to discover sets of clusterings X_1 and X_2 that maximize the compound fitness function $\tilde{q}(X_1,X_2)$. The compound fitness function $\tilde{q}(X_1,X_2)$ is defined as follows:

$$\tilde{q}(X_1,X_2) = \left(\alpha \times \left(q(X_1) + q(X_2)\right)\right) + \left((1-\alpha) \times Corr(X_1,X_2)\right) \tag{2}$$

where q is a fitness function that assess the quality of X_1 and X_2. The correspondence parameter α is a user-defined parameter. The correspondence function $Corr(X_1,X_2)$ measures the correspondence between X_1 and X_2.

CLEVER is modified to maximize the compound fitness function $\tilde{q}$ instead of the fitness function q. Two approaches that implement correspondence clustering are introduced in the following: (1) The Interleaved approach (C-CLEVER-I), and (2) The Concurrent approach (C-CLEVER-C). The algorithms of C-CLEVER-I and C-CLEVER-C are given in Figure 2 and 3, respectively.

The C-CLEVER-C uses the compound fitness function (equation 2) to cluster two data sets concurrently. For the C-CLEVER-I, dataset O_1 and O_2 are clustered one at a time—not concurrently— therefore, the compound fitness function (equation 2) simplifies to (3) and (4) when clustering the first and second dataset, respectively.

$$\tilde{q}_1(X_1) = \left(\alpha \times q(X_1)\right) + \left((1-\alpha) \times Corr(X_1,X_2)\right) \tag{3}$$

$$\tilde{q}_2(X_2) = \left(\alpha \times q(X_2)\right) + \left((1-\alpha) \times Corr(X_1,X_2)\right) \tag{4}$$

In general, there are many possible choices in selecting initial representatives of C-CLEVER-I and C-CLEVER-C. Our current implementation supports three options: The first option is to randomly select a subset of size k' from O as in CLEVER. The second option uses the final set of representative R from the previous iteration as the initial set of representatives. The third option uses the set of representatives *R'* of its counterpart clustering *X'* to compute a set of "nearby" representatives *R* taken from the dataset *O* as follows:

1. For each *r'* in *R'* determine its (1-)nearest neighbor in *O* obtaining a set *R*
2. Remove duplicates from *R*.

There are many choices for termination condition (*TCond*). The possible choices are: (1) fix the number of iterations; (2) terminate the program if the compound fitness function in the present iteration does not improve from the previous iteration.

```
Input: $O_1$ and $O_2$, $TCond$, $k'$, neighborhood-size, $p$, $p'$, $\alpha$, $\beta$
Output: $X_1$, $X_2$, $q(X_1)$, $q(X_2)$, $\tilde{q}(X_1,X_2)$, $Corr(X_1,X_2)$
Algorithm:
 1. Run CLEVER on dataset $O_1$ with fitness function $q$ and get clustering result $X_1$ and
    a set of representative $R_1$:
        $(X_1,R_1)$ :=Run CLEVER($O_1$, $q$);
 2. Repeat until the Termination Condition $TCond$ is met.
    a. Run CLEVER on dataset $O_2$ with compound fitness function $\tilde{q}_2$ that uses
       the representatives $R_1$ to calculate $Corr(X_1,X_2)$:
       $(X_2,R_2)$ :=Run CLEVER($O_2$,$R_1$, $\tilde{q}_2$)
    b. Run CLEVER on dataset $O_1$ with compound fitness function $\tilde{q}_1$ that uses
       the representatives $R_2$ to calculate $Corr(X_1,X_2)$:
       $(X_1,R_1)$ :=Run CLEVER($O_1$,$R_2$, $\tilde{q}_1$)
```

Fig. 2. Pseudo-code of C-CLEVER-I

```
Input: $O_1$ and $O_2$, $TCond$, $k'$, neighborhood-size, $p$, $p'$, $\alpha$, $\beta$
Output: $X_1$, $X_2$, $q(X_1)$, $q(X_2)$, $\tilde{q}(X_1,X_2)$, $Corr(X_1,X_2)$
Algorithm:
 1. Run CLEVER on dataset $O_1$ with fitness function $q$ and get clustering result $X_1$ and
    a set of representative $R_1$:
        $(X_1,R_1)$ :=Run CLEVER($O_1$, $q$);
 2. Run CLEVER on dataset $O_2$ with fitness function $q$ and get clustering result $X_2$ and
    a set of representative $R_2$:
        $(X_2,R_2)$ :=Run CLEVER($O_2$, $q$);
 3. Repeat until the Termination Condition $TCond$ is met.
    a. Run CLEVER on datasets $O_1$ and $O_2$ concurrently maximizing the
       compound fitness function $\tilde{q}$:
       $(X_1,R_1,X_2,R_2)$:=Run CLEVER($O_1$,$R_1$,$O_2$,$R_2$, $\tilde{q}$)
```

Fig. 3. Pseudo-code of C-CLEVER-C

4 Agreement Assessment by Forward and Backward Re-clustering Techniques and Co-occurrence Matrices

Using agreement between two clusterings X_1 and X_2 is a popular choice for a correspondence function. In applications such as change analysis [7] or co-location mining [2], domain experts want to discover clusterings that are good and agree at least to some extent. In such case, *Agreement*(X_1,X_2) would be used as the correspondence function. In addition, domain experts might be interested to discover regions with disagreement between the two datasets in anti-co-location or novelty detection. In the later case, *Corr*(X_1,X_2) can be defined as *(1-Agreement*(X_1,X_2)*)*. For the remaining of the paper, *Agreement*(X_1,X_2) will be used as correspondence function *Corr*(X_1,X_2); in the section we will introduce a measure to assess agreement.

First, we introduce re-clustering techniques that use the clustering model of one clustering to cluster the data in another dataset. In case of representative-based clustering, the cluster models are sets of representatives. Given two clusterings X_1 and X_2 of two datasets O_1 and O_2 and two sets of representatives of R_1 and R_2 of the two clusterings X_1 and X_2, forward and backward re-clusterings can be created using the representatives of one dataset to cluster the other dataset. More formally:

Definition 2. Let *O* be a dataset and *R* be an arbitrary set of representatives. Then $\chi_{REC}(O,R)$ denotes the result of re-clustering dataset *O* using the set of representatives *R*. The clusters of $\chi_{REC}(O,R)$ are created by assigning objects $o \in O$ to the closest representative $r \in R$ obtaining |*R*| clusters.

Definition 3. $\chi_{REC}(O_2,R_1)$ is called forward re-clustering and $\chi_{REC}(O_1,R_2)$ is called backward re-clustering.

To assess cluster similarity, the similarity between two representative-based clusterings X_1 and X_2 is computed by comparing *X1* with $\chi_{REC}(O_1,R_2)$ and *X2* with $\chi_{REC}(O_2,R_1)$.To assess the similarity of two clusterings, we construct a co-occurrence matrix M_X for each clustering *X* of *O* as follows:

1. If o_j and o_i belong to the same cluster in *X*, entries *(i,j)* and *(j,i)* of M_X are set to *1*.
2. If o_i is not an outlier in *X*, set *(i,i)* in M_X to *1*
3. The remaining entries of M_X are set to *0*

Let M_X and $M_{X'}$ be two co-occurrence matrices that have been constructed for two clusterings *X* and *X'* of the same dataset *O*; then the similarity between *X* and *X'* can be computed as follows:

$$Sim(X,X') := \frac{(Number\ of\ entries\ (i,j)\ with\ i \leq j\ in\ M_X\ and\ M_{X'}\ that\ both\ are\ 1\ in\ M_X\ and\ M_{X'})}{(Number\ of\ entries(i,j)\ with\ i \leq j\ that\ contain\ a\ 1\ in\ M_X\ or\ M_{X'}\ or\ a\ 1\ in\ both)} \quad (5)$$

Sim(X,X') in equation (5) is a generalization of the popular Rand Index [8] that additionally takes outliers into consideration. Finally, we define agreement between clustering X_1 and X_2, by comparing the clusterings of the two datasets with their respective forward and backward re-clusterings as follows:

$$Agreement(X1,X2) = \frac{Sim(X_1,\chi_{REC}(O_1,R_2)) + Sim(X_2,\chi_{REC}(O_2,R_1))}{2} \quad (6)$$

The advantage of the proposed agreement assessment method based on re-clustering techniques and co-occurrence matrices is that it can deal with: (1) datasets with unknown object identity, (2) different number of objects in two datasets, (3) different number of clusters in the two clusterings. Therefore, we claim that it is suitable for most types of spatial data.

5 Experiments

In the first experiment, we show that correspondence clustering provides comparable or better results than the traditional clustering. Moreover, the experimental results show that by enhancing agreement between two corresponding datasets, correspondence clustering produces clusterings that have lower variance than a traditional clustering algorithm. In the second experiment, we evaluate and compare different cluster initialization strategies for correspondence clustering.

5.1 Earthquake Dataset and Interestingness Function

We run our experiments on an earthquake dataset that is available on website of the U.S. Geological Survey Earthquake Hazards Program http://earthquake.usgs.gov/. The data includes the location (longitude, latitude), the time, the severity (Richter magnitude) and the depth (kilometers) of earthquakes. We uniformly sampled earthquake events from January 1986 to November 1991 as dataset O_1 and earthquake events between December 1991 and January 1996 as dataset O_2. Each dataset contains 4132 earthquakes.

Suppose that a domain expert interests in finding regions where deep and shallow earthquakes are in close proximity; that is, he/she is interested in regions with a high variance for the attribute earthquake depth. The following interestingness function captures the domain expert's notion of interestingness.

$$i(c) = \left\{ \begin{matrix} 0 & \frac{Var(c,z)}{Var(O,z)} \leq th \\ \left(\frac{Var(c,z)}{Var(O,z)} - th\right)^{\eta} & otherwise \end{matrix} \right\} \tag{7}$$

where $$Var(c,z) = \frac{1}{|c|-1} \sum_{o \in c} (z(o) - \mu_z)^2 \tag{8}$$

The attribute of interest $z(o)$ is depth of earthquake o; $|c|$ is the number of objects in region c and μ_z is the average value of z in region c; $th \geq 1$ is the reward threshold that captures what degree of earthquake depth variance the domain expert find news worthy; in general, regions with $i(c)=0$ are considered outliers. Finally, η with $\infty > \eta > 0$ is the reward function form parameter.

5.2 Experiment Investigating Variance Reduction

We run the interleaved approach of the representative based correspondence clustering, C-CLEVER-I, and the traditional clustering algorithm, CLEVER, and compare the results with respect to cluster quality and agreement.

First we run CLEVER on dataset O_1 and O_2 five times to generate five clusterings for each dataset. Then we run C-CLEVER-I for five iterations with $\alpha=1.0e\text{-}4$ and

α=*2.0e-6* for five times each. Figure 4 summarizes the experiments conducted. Each circle represents each clustering. The dashed lines between Clustering X_1 and X_2 in CLEVER show that fitness values ($q(X_1)+q(X_2)$), and $Agreement(X_1,X_2)$ of CLEVER are computed from all twenty five possible pairs of X_1 and X_2. When correspondence clustering is used, those values are only computed for the five pairs of clusterings obtained by C-CLEVER-I (one for each run; indicated by solid lines with two ways arrows). For each clustering of C-CLEVER-I, the representatives from the previous iteration of its own clustering are used as initial representatives of the present iteration. The parameter settings of CLEVER and C-CLEVER-I are shown in Table 1 and Table 2. All parameters for CLEVER and C-CLEVER-I are set to the same values except for the values of *p* and *p'*. Since C-CLEVER-I is run for five iterations for each pair of clustering X_1 and X_2, for a fair comparison, we set the *p* and *p'* of CLEVER to be five times higher than C-CLEVER-I. The experiment is evaluated by fitness function (equation (1)), agreement (equation (6)) and similarity (equation (5)). Table 3 shows average values of all the experimental results. The computation time measures the average wall clock time in milliseconds used by the algorithms to generate a pair of clusterings X_1 and X_2. We use similarity measure *Sim(X,X')* in equation (5) to assess variance between two clusterings generated using the same dataset. In general, the algorithm that produces higher *Sim(X,X)* creates clusterings that are more similar in different runs, thus, exhibiting lower variance.

Table 1. Parameter settings of CLEVER

β=2.0	*p*=100	*p'*=100	η=2.0	*th*=1.2
Neighborhood size = 3	Probabilities for add, delete, and replace representatives : 0.2, 0.2, 0.6			

Table 2. Parameter settings of C-CLEVER-I

TCond = 5 iterations	β=2.0	*p*=20	*p'*=20	η=2.0	*th*=1.2
Neighborhood size = 3	Probabilities for add, delete, and replace representatives : 0.2, 0.2, 0.6				

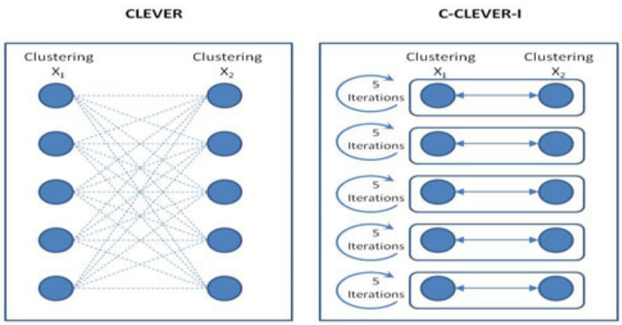

Fig. 4. Illustration of the experiment

Table 3. Comparison of average results of CLEVER and C-CLEVER-I

	CLEVER	C-CLEVER-I (α=*1.0e-5*)	C-CLEVER-I (α=*2.0e-6*)
Fitness $q(X_1)$	1896492	1896821	1870152
Fitness $q(X_2)$	1756189	1713519	1685332
$q(X_1) + q(X_2)$	3652681	3610341	3555485
$Agreement(X_1,X_2)$	0.332909	0.349231	0.776172
$Sim(X_1,X_1)$	0.665973	0.703770	0.663314
$Sim(X_2,X_2)$	0.344895	0.623614	0.619446
Computation Time	5.48E+06	2.18E+06	2.30E+06

From Table 3, C-CLEVER-I with α=*1.0e-5* produces higher fitness values for clustering X_1 and but lower fitness values of X_2 than CLEVER. For $Agreement(X_1,X_2)$ and $Sim(X_1,X_1)$, C-CLEVER-I with α=*1.0e-5* produces slightly higher values than CLEVER but for $Sim(X_2,X_2)$, C-CLEVER-I produces significantly higher value than CLEVER. From this point of view, the higher values of $Sim(X_1,X_1)$ and $Sim(X_2,X_2)$ indicate than each run of C-CLEVER-I creates more similar clustering results for each clustering X_1 and X_2 which means that C-CLEVER-I produces lower variance than CLEVER. With α=*2.0e-6*, C-CLEVER-I is forced to emphasize agreement. Therefore, the fitness values of clustering X_1 and X_2 of C-CLEVER-I are slightly lower than CLEVER but $Agreement(X_1,X_2)$, and $Sim(X_2,X_2)$ of C-CLEVER-I are significantly higher than CLEVER. Moreover, C-CLEVER-I computes its results about half of the runtime CLEVER uses.

From the experimental results, we conclude that correspondence clustering can reduce the variance inherent to representative-based clustering algorithms. Since the two datasets are related to each other, using one dataset to supervise the clustering of the other dataset can lead to more reliable clusterings by reducing variance among clusterings that would have resulted from using traditional representative-based clustering algorithms, as they are more susceptible to initial representatives and outliers. Moreover, obtaining higher agreement could be accomplished with only a very slight decrease in cluster quality.

5.3 Experiment for Representative-Based Correspondence Clustering with Different Methods of Initial Representative Settings

We run experiments to compare results of three initialization strategies for C-CLEVER-I; the three tested strategies are as follows : (1) random representatives (C-CLEVER-I-R), (2) representatives from the nearest neighbor of representatives of the counterpart clustering (C-CLEVER-I-C), and (3) the final representatives from the previous iteration are used as the initial representatives for the next iteration (C-CLEVER-I-O). For each option of the initial representative setting techniques, five pairs of clustering X_1 and X_2 are generated, similar to the previous experiment. Table 4 shows parameter settings used in the experiments. The average values of the experimental results are shown in Table 5.

Table 4. Parameter settings of C-CLEVER-I

TCond = 5 iterations	α=2.0e-8	β=2.8	*p*=20	*p'*=20	η=2.0	*th*=1.2
Neighborhood size = 3	Probabilities for add, delete, and replace representatives : 0.2, 0.2, 0.6					

Table 5. Comparison of average results of C-CLEVER-I with different means of initial representative settings

	C-CLEVER-I-C	C-CLEVER-I-O	C-CLEVER-I-R
Compound Fitness $\tilde{q}(X_1,X_2)$	9.857655	10.10406	9.952686
Fitness $q(X_1)$	2.3E+08	2.66E+08	2.46E+08
Fitness $q(X_2)$	2.14E+08	2.17E+08	2.13E+08
$q(X_1) + q(X_2)$	4.44E+08	4.82E+08	4.59E+08
$Agreement(X_1,X_2)$	0.977259	0.459206	0.771505
Computation Time	3.23E+06	2.72E+06	7.10E+06

From Table 5, C-CLEVER-I-C produces clusterings with the highest agreement but the lowest compound fitness value. This is because C-CLEVER-I-C uses initial representatives that are closest to the representatives of its counterpart clustering. Then C-CLEVER-I-C generates clusterings X_1 and X_2 that are very similar which results in very high agreement. Though, the agreement is very high, the low fitness values lead to the low compound fitness value. For C-CLEVER-I-O, the initial representatives used are the final representatives from the previous iteration. In contrast to C-CLEVER-I-C, with α=*2.0e-8*, C-CLEVER-I-O favors increasing fitness values rather than increasing agreement between the two clusterings. This is indicated by the highest fitness values but the lowest agreement value. As for C-CLEVER-I-R, it produces comparable fitness values and intermediate agreement value but consumes the highest computation time. This is due to the fact that C-CLEVER-I-R randomizes its initial representatives, which allows the algorithm to explore the dataset more thoroughly than the others but in return, it needs more time to find the solution.

6 Related Work

Correspondence clustering relates to coupled clustering, and co-clustering which both cluster more than one dataset at the same time. Coupled clustering [3] is introduced to discover relationships between two textual datasets by partitioning the datasets into corresponding clusters where each cluster in one dataset is matched with its counterpart in the other dataset. Consequently, the coupled clustering requires that the number of clusters in two datasets be equal. In general, the coupled clustering ignores intra-dataset similarity and concentrates solely on inter-dataset similarity. Our approach, on the other hand, provides no limitation on number of clusters. It considers both intra-dataset and inter-dataset similarities. The intra-dataset similarity is included through interestingness measures and the inter-dataset similarity is included through correspondences in sets of representatives.

Co-clustering has been successfully used for applications in text mining [4], market-basket data analysis, and bioinformatics [5]. In general, the co-clustering clusters two datasets with different schemas by rearranging the datasets. In brief, datasets are represented as a matrix with one dataset is organized into rows while the other dataset is organized into columns. Then, the co-clustering partitions rows and columns of the data matrix and creates clusters which are subsets of the original matrix. In case of spatial data mining, re-organizing the data into data matrix causes spatial relationships related to location of data points to be lost: clusters are no longer guaranteed to be contiguous. Accordingly, co-clustering is not applicable to spatial clustering.

Correspondence clustering is also related to evolutionary clustering [6] that is used for multi-objective clustering. Evolutionary clustering clusters streaming data based on two criteria: the clustering quality of present data and its conformity to historical data.

In conclusion, the three reviewed clustering techniques cluster data based on distances alone whereas the correspondence clustering approach allows to cluster multiple datasets based on a domain expert's definition of interestingness and correspondence. Consequently, correspondence clustering is more general and can serve a much larger group of applications. For example, none of the three approaches can be used for the earthquake clustering problem we used in the experimental evaluation; in the experiment, clusters are formed by maximizing the variance of a continuous variable and not by minimizing the distances between objects that belong to the same cluster.

7 Conclusion

In this paper, we introduce a novel clustering approach called correspondence clustering that clusters two or more related spatial datasets by maximizing cluster interestingness and correspondence between clusters for the different datasets. A representative-based correspondence clustering framework is introduced and two representative-based correspondence clustering algorithms are proposed. We conducted experiments in which two datasets had to be clustered by maximizing cluster interestingness and agreement between the spatial clusters obtained. The results show that correspondence clustering can reduce the variance inherent to representative-based clustering algorithms. Moreover, high agreements could be obtained by only slightly lowering clustering quality. In general, correspondence clustering is beneficial for many applications, such as change analysis, co-location mining and applications that are interested in analyzing particular, domain-specific relationships between two or more datasets.

In addition, the paper proposes a novel agreement assessment measure that relies on re-clustering techniques and co-occurrence matrices. The agreement assessment technique proposed is applicable for (1) datasets with unknown object identity, (2) different number of objects in two datasets, (3) different number of clusters in two clusterings. Therefore, it is suitable for most types of spatial data.

References

1. Ding, W., Jiamthapthaksin, R., Parmar, R., Jiang, D., Stepinski, T., Eick, C.F.: Towards Region Discovery in Spatial Datasets. In: Proceedings of 12th Pacific-Asia Conference on Knowledge Discovery and Data Mining (2008)
2. 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: Proceedings of 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (2008)

3. Marx, Z., Dagan, I., Buhmann, J.M., Shamir, E.: Coupled Clustering: A Method for Detecting Structural Correspondence. Journal of Machine Learning Research 3, 747–780 (2002)
4. Dhillon, I.S.: Co-clustering Documents and Words using Bipartite Spectral Graph Partitioning. In: Proceedings of 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2001)
5. Cheng, Y., Church, C.M.: Biclustering of Expression Data. In: Proceedings of 8th International Conference on Intelligent Systems for Molecular Biology (2000)
6. Chakrabarti, D., Kumar, R., Tomkins, A.: Evolutionary Clustering. In: Proceedings of 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2006)
7. Chen, C.S., Rinsurongkawong, V., Eick, C.F., Twa, M.D.: Change Analysis in Spatial Data by Combining Contouring Algorithms with Supervised Density Functions. In: Proceedings of 13th Pacific-Asia Conference on Knowledge Discovery and Data Mining (2009)
8. Rand, W.: Objective Criteria for the Evaluation of Clustering Methods. Journal of the American Statistical Association 66, 846–850 (1971)