Finding Regional Co-location Patterns for Sets of Continuous Variables

Christoph F. Eick[†], Rachana Parmar[†], Wei Ding[†], Tomasz F. Stepinski[‡] and Jean-Philippe Nicot[*]

Abstract

This paper proposes a novel framework for mining regional co-location patterns with respect to sets of continuous variables in spatial datasets. The goal is to identify regions in which multiple continuous variables with values from the wings of their statistical distribution are co-located. One particular challenge of regional co-location mining is that the employed algorithms need to search for both interesting places and interesting patterns at the same time. A co-location mining framework is introduced that operates in the continuous domain without the need for discretization and which views regional co-location mining as a clustering problem in which an externally given fitness function has to be maximized. Interestingness of co-location patterns is assessed using products of z-scores of the relevant continuous variables. A novel, prototype-based region discovery algorithm named CLEVER is introduced that uses randomized hill climbing, a variable number of clusters, and which searches larger neighborhood sizes. The proposed framework is evaluated in a case study that analyzes chemical concentrations in Texas water wells centering on co-location patterns involving Arsenic. Our approach was able to identify known and unknown regional co-location sets. Different sets of algorithm parameters lead to the characterization of arsenic distribution at different scales. Moreover, inconsistent co-location sets were found for regions in South Texas and West Texas that can be clearly attributed to geological differences in the two regions, emphasizing the need for regional co-location mining techniques.

Keywords

spatial data mining, regional co-location mining, regional data mining, clustering, finding associations between continuous variables.

1 Introduction

The goal of spatial data mining [16] is to automate the extraction of interesting and useful patterns that are not explicitly represented in spatial datasets. Discovery of co-location patterns, a co-occurrence of different types of features at approximately the same locations, is an important example of data mining task with many practical applications. Most existing research has concentrated on discovering collocation patterns with respect to categorical features, which identify sets of classes whose instances co-occur in geographical proximity with high frequency. A classic example [17] of such a relationship is the collocation of two types of animals, the Nile crocodile and the Egyptian plover, which is traced by domain scientists to their symbiotic relationship.

However, not all real-life problems are susceptive to the categorical formulation. In a broad range of problems the spatial dataset is given in the form of continuous variables. Formulating such a problem in terms of categorical, discrete features is not natural. In this paper, we are interested to identify places in which multiple continuous variables with values from the wings of their statistical distribution are co-located. In other words, we are interested to identify regions where extreme values of different continuous variables are present in geographical proximity.

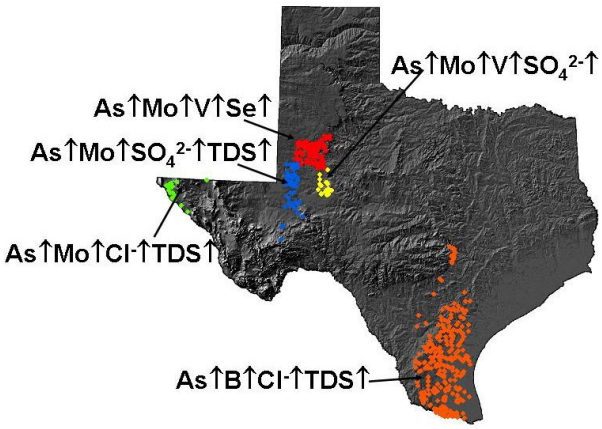

Figure 1: Regional Co-location Patterns Involving Chemical Concentrations in Texas Wells

[†] Department of Computer Science, University of Houston, Houston, TX, 77204-3010, {ceick, rparmar, wding}@uh.edu

[‡] Lunar and Planetary Institute, Houston, TX 77058, tstepinski@lpi.usra.edu

[*] Bureau of Economic Geology, Jackson School of Geosciences, University of Texas at Austin, jp.nicot@beg.utexas.edu

Figure 1 illustrates what we are trying to accomplish for a data set that contains concentrations of chemicals in different wells in Texas. The goal is to find regions (sets of wells that cover a contiguous area) in which concentrations of multiple chemicals take extreme values. The figure shows the result of running a regional co-location mining algorithms on such a dataset; it identifies five regions with five different patterns of chemical concentrations. For example, two interesting regions were identified in the Western half of Texas: the first region, colored in red, contains high concentrations of Arsenic (As↑) Molybdenum (Mo↑), Vanadium (V↑) and Selenium (Se↑), whereas the second region, colored in blue, is characterized by high concentrations of Arsenic (As↑), Molybdenum (Mo↑), Sulfate (SO_4^{2-}↑) and Total Dissolved Solids (TDS↑).

In this paper, we describe and evaluate a novel framework for discovering co-location regions and their associated patterns in a highly automated fashion in continuous datasets without the need for discretization. The proposed framework treats region discovery as a clustering problem in which clusters have to be obtained that maximize an externally given fitness function. The fitness function combines contributions of interestingness from an individual cluster and can be customized by a domain expert. The framework allows the actual clustering task to be performed by a variety of different algorithms. A highly desirable feature of our approach is its search-engine-like capabilities, returning a set of regions ranked by interestingness thus providing a domain expert with pertinent information.

Related Work. An early version of the region discovery framework that was restricted to categorical datasets has been described in [7]. This framework has later been successfully used for discovering regional association rules [6]. This paper extends our work to continuous datasets, supports arbitrary fitness functions, and uses it for a new task: regional co-location mining. Shekhar et al. [17] discuss several interesting approaches to mine co-location patterns with respect to a given set of events. Huang et al. [9] center on co-location mining involving rare events and introduce a novel measure of interestingness for this purpose. However, it should be stressed that all mentioned approaches are restricted to categorical datasets and center on finding global co-location patterns, whose scope is the whole dataset. Our approach, on the other hand, as we will explain later in more detail, centers on discovering regional co-location patterns.

Most of the approaches to mine association rules in continuous datasets use discretization. In [19], numerical attributes are discretized and then adjacent partitions are combined as necessary. This leads to information loss and can generate spurious rules. Aumann et al. [2] introduce numerical association rules that support statistical predicates for continuous attributes, such as variance, and algorithms that mine such rules. In [3], rank correlation is used to mine associations between numerical attributes. Basically, continuous attributes are transformed to ordinal attributes, and a method is proposed to find sets of numerical attributes with high attribute values.

In [1], an interesting method is presented for deriving equations describing clusters containing numerical data. The approach first uses a clustering algorithm to find correlation clusters, and then derive equations describing the linear space approximating each cluster's data points. Localized spatial statistics [8] also analyzes regional characteristics in spatial datasets. However, the proposed methodology is not suitable for large datasets and relies on extensive human interactions. Finally, Klösgen and May [11] propose a generic, multi-relational framework for subgroup discovery with a relational database system.

Contributions. First, a novel regional co-location mining framework is introduced that identifies places in which continuous variables taking values from the wings of their respective distributions co-occur. The proposed method directly operates in the continuous domain without any need for discretization. One particular challenge of this task is that the employed algorithms need to search for both interesting places and interesting patterns at the same time. Second, we apply our framework to the problem of identifying regional co-location patterns with respect to high and low arsenic concentrations in Texas water supply. A thorough analysis of this case study is presented including comparison of results obtained using different region discovery algorithms and an assessment of the found patterns by a domain expert. Third, as a by product, a novel prototype-based clustering algorithm named CLEVER is introduced and evaluated in the case study. CLEVER uses randomized hill climbing, allows for a variable number of clusters, and searches larger neighborhood sizes to battle premature convergence.

2 Methodology

Region Discovery Framework. The region discovery problem is defined as follows: given the dataset O, a set of k clusters (regions) $X = \{c_1,...,c_k\}$, $c_i \subseteq O$, is sought that maximizes a fitness function q. The regions are disjoint, contiguous, and elements of O can be outliers that do not belong to any region.

The fitness function q is defined as follows:

$$q(X) = \sum_{c \in X} reward(c) = \sum_{c \in X} i(c) * |c|^{\beta} \quad (2.1)$$

where i(c) is the interestingness measure of a region c—a quantity designed by a domain expert to reflect a degree to which regions are "newsworthy".

Within our present focus, *i*(c) must encapsulate a degree to which extreme values of variables are present together in region c. The region size is denoted by |c|, and the quantity $i(c)*|c|^{\beta}$ can be considered as a "reward" given to a region c; we seek X such that the sum of rewards over all of its constituent regions is maximized. The amount of premium put on the size of the region is controlled by the value of parameter β (β>1). A region reward is proportional to its interestingness, but a bigger region receives a higher reward than a smaller region having the same value of interestingness to reflect a preference given to larger regions.

Interestingness of Regional Co-Location Patterns. Our approach tries to discover regional co-location patterns involving sets of continuous variables having values on the wings of their distribution. The pattern A↑ denotes that attribute A has large values and the pattern A↓ indicates that attribute A has low values. For example, the pattern {A↑, B↓, D↑} describes that high values of A are co-located with low values of B and high values of attribute D. In the following a function *i* is introduced that measures the interestingness of co-location patterns for a region c:

Let

O be a dataset

c⊆O be a region

o∈ O be an object in the dataset O

F={A_1,…,A_q} be the set of continuous attributes in the dataset O

S={A_1↑, A_1↓,…, A_q↑, A_q↓} be the set of possible base co-location patterns

B⊆S be a set of co-location patterns

P(B) be a predicate over B that restricts the co-location sets considered[1]

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

$$(2.2)\quad z(A\uparrow,o)=\begin{cases} z\text{-}score(A,o) & if\ z\text{-}score(A,o)>0 \\ 0 & otherwise \end{cases}$$

$$(2.3)\quad z(A\downarrow,o)=\begin{cases} -z\text{-}score(A,o) & if\ z\text{-}score(A,o)<0 \\ 0 & otherwise \end{cases}$$

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

$$(2.4)\quad i(B,o)=\prod_{p\in B} z(p,o) * d^{|B|-2}$$

When using the above formula, the more extreme the z-values of the involved objects are the bigger the above product becomes—moreover, if the value of a continuous variable does not match its suggested pattern in B its z-value is 0 and the interestingness is therefore 0 as well. Although this approach compresses multiple z-values into a single value, the product of z-values still allows for meaningful statistical interpretation using the geometric mean; for example, if the geometric mean of the z-values of the patterns in set B is 1.5, this suggests that values of the involved variables are at an average 1.5 standard deviations off their mean value. Finally, the term $d^{|B|-2}$ (assuming $d \geq 1$) increases the interestingness of co-location sets by a factor of d, every time the cardinality of a set is increased by 1.

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 approach additionally considers purity when computing region interestingness, where *purity(B,c)* denotes the *percentage of objects o∈c for which i(B,o)>0.* In summary, the interestingness of a region c with respect to a co-location set B, denoted by *φ(B,c)*, is computed as follows[2]:

$$(2.5)\quad \varphi(B,c)=\frac{\left(\sum_{o\in c} i(B,o)\right)}{|c|} * purity(B,c)^{\theta}$$

The un-normalized, *raw interestingness* of a region c, denoted by $\kappa_S(c)$ is measured as the maximum interestingness *φ(B,c)* observed over all subsets B⊆S with cardinalities 2 and higher considered.

$$(2.6)\quad \kappa_S(c)=\max_{B\subset S\&|B|>1\&P(B)} \varphi(B,c)$$

Finally, the normalized[3] interestingness of a region c, *i*(c), is defined as follows:

[1] e.g. P(B)=|B|<5 ("only co-locations sets with cardinalities 2, 3 and 4 are considered") or P(B)=As↑∈B ("only look for patterns involving high arsenic")

[2] The parameter θ controls the importance attached to purity in interestingness computations.

[3] One assumption underlying our framework is that clusters that are not interesting for a domain expert receive a reward of 0. Therefore, fitness functions are usually normalized and scaled in collaboration with domain experts based on what the domain expert finds "newsworthy".

$$i(c) = \begin{cases} (\kappa_S(c) - th)^{\eta} \text{ if } \kappa_S(c) > th \\ 0 \quad \text{otherwise} \end{cases} \tag{2.7}$$

The threshold parameter $th \geq 0$ is introduced to weed out regions c with $\kappa_S(c)$ close to 0. Finally, η is a scaling factor that allows modifying raw interestingness super-linearly by choosing $\eta>1$, and sub-linearly by choosing $\eta<1$. Finally, as already introduced in beginning in formula (2.1), the reward of the region c is computed as follows:

$$reward(c) = i(c) * |c|^{\beta} \tag{2.8}$$

Example. Table 1 shows an example region c, containing four objects with the indicated values for attributes C and D, and intermediate values used in calculating *i*(B,o) for pattern B={C↑, D↓}. Column 3 and 4 display the z-values for C↑ and D↓ respectively that were calculated using formulas (2.2) and (2.3). Column 5 displays i(B,o) as per (2.4) assuming d=1. We can see that purity of pattern B is 0.5. Assuming θ=1, using formula (2.5) we obtain: φ(B,c)=0.06=((0.24+0.24)/4)*0.5,

Table 1: Example Data

Id	**C z-score**	**D z-score**	**C↑**	**D↓**	***i*(B, o)**
1	0.43	-0.56	0.43	0.56	0.24
2	0.42	-0.56	0.42	0.56	0.24
3	-0.06	0.13	0	0	0
4	-0.57	-0.22	0	0.22	0

When a dataset contains null values for continuous attributes, a region's average interestingness in formula (2.5) is calculated only considering those objects whose product for set B is not null. Purity, on the other hand, is still computed by determining the portion of objects in the region for which i(B,o) is greater than 0; that is, objects with missing values degrade purity of a region.

While evaluating $\varphi(B,c)$ (2.5) all the subsets $B \subseteq S$ with cardinality 2 to the maximum pattern length specified are considered. It is not possible to do pruning based on maximum valued co-location set of size m when computing co-location sets of size (m+1) because the interestingness function *i* is not anti-monotone. We explain this using a counter example. Let us assume that for a region c $B=\{A_1\uparrow,A_2\uparrow\}$ is the binary pattern with the highest interestingness, however, as we will see, the highest interestingness pattern of size three B' may not contain B as the subset. Let's assume, that all objects with positive z-scores for A_1 and A_2 in region c have zero z-scores for remaining attributes $A_3,\ldots,A_5$ and that there is at least one object in region c that has zero z-scores for A_1 and A_2 and positive z-scores for remaining attributes. All the patterns of size 3 having B as a subset will therefore have interestingness 0, but $\{A_3\uparrow,A_4\uparrow,A_5\uparrow\}$'s interestingness is above 0. Therefore, the maximum interestingness pattern of size 3 does not contain $\{A_1\uparrow,A_2\uparrow\}$ for region c.

The interestingness function *i* employs the following parameters: $d \geq 1$ (called *discount factor*), $\theta \geq 0$, and $\eta \geq 0$. Choosing higher discount factors raises the interestingness of co-location patterns involving larger sets; for example, if d=1.4 the raw interestingness of sets of cardinality of 3 is multiplied by 1.4, and of sets of cardinality 4 is multiplied by 1.4^2. η is used for reward scaling, using $\eta>1$ increases the spread of reward values, whereas using $\eta<1$ has the reverse effect. θ determines the weight purity carries in region interestingness computations; our approach balances the average interestingness of the objects belonging to a region with the purity of a region: when using θ=0 only the average product of z-values in a region is used, whereas selecting very large values for θ assigns a high weight to purity in region interestingness computations. These parameters are usually selected in close collaboration with domain experts in conjunction to considering statistical properties of the dataset analyzed. For example, if the domain expert is interested in regions of extreme products of z-values, even if their purity is not that high η=3 and θ=0 might be a good choice. The following default parameter settings are currently used in the proposed framework: d=1, θ=1, η=1 and th=0.

Moreover, using the average interestingness of regions for a particular co-location set B, a map of locally-defined interestingness can be produced to give domain scientists a visual indication of collocation patterns. Going beyond such visual information, region discovery algorithms will be introduced in the next section that calculate co-location regions automatically and are able to quantify their findings.

3 Region Discovery Algorithms

We have developed two prototype-based clustering algorithms, SPAM and CLEVER, for region discovery. Prototype-based clustering algorithms construct clusters by seeking for a set of representatives; clusters are then created by assigning objects in the dataset to the closest representative. Popular prototype-based clustering algorithms are K-Medoids/PAM [10] and K-means [12]. SPAM (Supervised PAM) is a variation of PAM. SPAM uses the fitness function q(X)—and not the mean square error of the distance of cluster objects to the cluster representative as PAM does—as its fitness function. SPAM starts its search with a randomly created set of *k* representatives, and then greedily replaces representatives with non-representatives as long as q(X) improves.

An empirical evaluation of SPAM on the task of region discovery reveals that—see also Table 5—that SPAM tends to terminate prematurely and running SPAM multiple times did not improve solution quality a lot. Moreover, the best number of regions k is very hard to know in advance especially for real-world problems. This motivated the development of CLEVER (CLustEring using representatiVEs and Randomized hill climbing) that seeks for the optimal value of k, uses larger neighborhood sizes to battle premature convergence, and uses randomized hill climbing and re-sampling to reduce algorithm complexity.

CLEVER, as SPAM, seeks to maximize the fitness function q(X). The algorithm (see Figure 3) starts with randomly created set of $\hat{k}$ representatives—$\hat{k}$ is a parameter of the algorithm. It samples p solutions in the neighborhood of the current solution; unlike CLARANS [13] which picks the first best neighbor as the next solution, CLEVER evaluates all the p neighbors and picks the best among them.

Neighborhood Size = 4 (4 operators will be applied on the current solution)

p(Insert) = 0.33, p(Delete) = 0.33, p(Replace) = 0.34

Current representative solution = {1, 2, 3}

Current set of non-representatives = {0, 4, 5, 6, 7, 8, 9}

{1, 2, 3} → (Insert) {1, 2, 3, **0**} → (Replace) {**6**, 2, 3, 0} → (Delete) {6, 3, 0} → (Insert)

{6, 3, 0, **5**}

A new neighboring solution = {6, 3, 0, 5}

Figure 2: Example describing neighboring solution generation

The function findNeighbors at step 7 in Figure 3 creates a set of neighboring solutions of the current solution using three operators: 'Insert' – inserts a new representative into the current solution, 'Delete' – deletes a representative from the current solution and 'Replace' – replaces a representative with a non-representative. Each operator has a certain selection probability. While generating a new neighboring solution, operators are selected as per their selection probabilities. The algorithm also allows for larger neighborhood sizes, i.e. most experiments in the paper were run for neighborhood size 3: in this case, solutions that are sampled are generated by applying three randomly selected operators to the current solution. Figure 2 gives an example of how a neighboring solution is generated.

CLEVER($\hat{k}$ *, neigbhborhoodDef, p, q*)

1. Randomly select initial set of $\hat{k}$ representatives. Set it to *currReps*.
2. *regions* = findRegions (*currReps*). {Find regions for current set of representatives}
3. *currentFitness* = findFitness (*regions*) {Find total fitness for the current regions}
4. Set *nonReps* to the data objects that are non-representatives.
5. *bestFitness = currentFitness*
6. **while** TRUE **do**
7. *neighbors* = findNeighbors (*currReps, nonReps, p, neighborhoodDef*) {Pick *p* neighbors from the neighborhood of current representatives as defined by the parameter '*neighborhoodDef*'}.
8. **for** each *neighbor* ∈ *neigbhors* **do**
9. *newRegions* = findRegions (*neighbor*). {Find regions using the neighboring solution}
10. *newFitness* = findFitness (*newRegions*) {Find total fitness for newly found regions}
11. **if** *newFitness* > *bestFitness*
12. *bestNeighbor = neighbor* {replacement}
13. *bestFitness = newFitness*
14. *bestRegions = newRegions*
15. **end**
16. **end for**
17. **if** fitness improved
18. *currReps = bestNeighbor*
19. *currentFitness = bestFitness*
20. *regions = bestRegions*
21. **if** resampling done
22. Revert to original p value.
23. **end**
24. Go back to step 7.
25. **else**
26. **if** resampling done twice and fitness does not improve after resampling
27. Terminate and return obtained regions, region representatives, no. of representatives (k), and fitness.
28. **else if** first resampling
29. p = p * 2 {resample with higher p value}
30. Go back to step 7.
31. **else** {Resample again}
32. p = p * (q – 3) (resample with much higher p value)
33. Go back to step 7.
34. **end**
35. **end**

Figure 3: Pseudo-code of Algorithm CLEVER

By adding and deleting representatives and by using neighborhood size of larger than one, CLEVER samples from much larger neighborhood of the current solution. This characteristic distinguishes CLEVER from other prototype-based clustering algorithms.

Out of the neighbors the one that improves q(x) the most becomes new current solution and the procedure is repeated. If none of the p sampled neighbors improves q(x), the number of samples is doubled (step 28-30). If after re-sampling the fitness does not improve, sample size is increased to p∗(q-3) (step 31-33). CLEVER terminates when q(x) does not improve after analyzing p*q (p + p*2 + p*(q-3)) samples. In general, the algorithm is self adaptive because it only samples a small number of neighbors p when moving upwards easily, but significantly increases the number of samples when the algorithm is in danger to converge prematurely.

4 Case Study: Finding Regional Co-location Patterns with respect to Arsenic in the Texas Water Supply

We evaluated our framework in a real world case study to discover regional co-location patterns involving Arsenic and other chemicals in the Texas water supply.

Dataset Description and Preprocessing. Datasets used in this case study are created using the Groundwater database (GWDB) maintained by the Texas Water Development Board (TWDB) [20]. Very high concentration of Arsenic is dangerous. Also, long term exposure to low level of Arsenic concentration can lead to increased risk of cancer [18]. It is important to understand factors that cause high concentration of Arsenic, i.e. when we have high concentration of Arsenic in a region what is the level of concentration of other chemicals.

Currently the GWDB has water quality data for 105,814 wells in Texas. The data have been collected regularly with a recurrence time of 3 to 5 years for an individual well over last 25 years. The database had to be cleaned of duplicate, missing and/or inconsistent values. In the dataset, each well has zero or more samples for a chemical. As we are particularly interested in Arsenic, we have considered only those wells where there is at least one sample for Arsenic concentration. Other chemical concentrations may have null values. When multiple samples exist for a well, we apply an aggregate function to the set of samples, such as max or avg. The particular dataset we used in the evaluation has 3 spatial attributes: longitude, latitude and aquifer and 10 non-spatial attributes: The 4 trace elements, Arsenic, Molybdenum, Vanadium, and Boron, the 2 minor elements, Fluoride and Silica, and the 2 major ions, Chloride and Sulfate to which Total Dissolved Solids (TDS) and Well Depth are added. The dataset has no records with null values, and has 1,653 records. Multiple samples were aggregated using average values. Here onwards we call this dataset Arsenic_10_avg. We also extracted several other datasets by using different aggregate function like maximum and adding concentrations of other chemicals. All of these datasets are available on the web [5].

For each non-spatial attribute, we calculate z-scores and then calculate $z(A\uparrow,o)$ and $z(A\downarrow,o)$ using formulas (2.2) and (2.3).

Experimental Results. We have tested our regional co-location mining framework by applying the algorithm CLEVER described in Section 3 using the fitness function described in Section 2 to the dataset described above. In this subsection we discuss co-location patterns discovered by CLEVER algorithm and in the next subsection we briefly compare CLEVER with SPAM. We have conducted experiments using Arsenic_10_avg and other datasets described above but in this paper only experiments using Arsenic_10_avg are discussed in detail. Complete results involving all the datasets can be found in [14]. Arsenic_10_avg has 10 non-spatial attributes, and therefore 20 base patterns exist to construct co-location sets. Moreover, maximum co-location set size is limited to 4 in the experiments that are discussed in the paper. Because we are interested in discovering co-location patterns with respect to arsenic, only co-location sets that contain $As\uparrow$ or contain $As\downarrow$ are considered.

Table 2: Parameters Used in Experiments

Common Parameter Settings	d=1.0, η=1, th=0.0, $\hat{k}$=50 (initial no. of clusters), p=50, q=50, Neighborhood Size=3, p(Insert)=0.2, p(Delete)=0.2, p(Replace)= 0.6	
$P(B) = \{As\uparrow \in B \text{ or } As\downarrow \in B\}$		
Exp. 1	|B|<5	β = 1.3, θ=1.0
Exp. 2	|B|<5	β = 1.5, θ=1.0
Exp. 3	|B|<5	β = 2.0, θ=1.0
Exp. 4	|B|<5	β = 1.5, θ=5.0

Table 2 summarizes the parameters used in the experiments. As the value of parameter β affects size of the regions found, we have conducted experiments using three different values for this parameter (Experiment 1-3); maximum co-location set sizes are restricted to four in these experiments. The parameter θ determines importance of purity when evaluating regions. Experiment 4 uses a very high θ value but shares all other parameters with experiment 2.

Table 3: Top 5 Regions Ranked by Interestingness (as per formula 2.7)

Exp. No.	Top 5 Regions	Region Size	Region Interest-ingness	Maximum Valued Pattern in the Region	Purity	Average Product for maximum valued pattern
Exp. 1	1	23	174.3191	As↑Mo↑V↑F^-↑	0.83	211.0179
	2	40	104.8576	As↑Mo↑V↑	0.65	161.3194
	3	11	92.9385	As↑Mo↑V↑SO_4^{2-}↑	0.55	170.3873
	4	36	89.4068	As↑B↑Cl^-↑TDS↑	0.58	153.2687
	5	7	30.5775	As↑Mo↑Cl^-↑TDS↑	0.57	53.5107
Exp. 2	1	80	33.5978	As↑B↑Cl^-↑TDS↑	0.48	70.7322
	2	181	25.3314	As↑Mo↑V↑F^-↑	0.49	52.1020
	3	17	6.4819	As↑Mo↑Cl^-↑TDS↑	0.29	22.0383
	4	23	6.4819	As↓Cl^-↑SO_4^{2-}↑TDS↑	0.78	8.1287
	5	10	3.4645	As↓B↑Cl^-↑TDS↑	0.4	8.6612
Exp. 3	1	238	5.3234	As↑B↑Cl^-↑TDS↑	0.22	23.9052
	2	833	1.8118	As↑Mo↑V↑F^-↑	0.16	11.4334
	3	152	0.3201	As↓SiO_2↑WD↑	0.53	0.6006
	4	432	0.1969	As↓TDS↓	0.93	0.2122
	5	N/A				
Exp. 4	1	7	630.1098	As↑B↑Cl^-↑TDS↑	1.0	630.1097
	2	2	541.4630	As↑Mo↑V↑B↑	1.0	541.4630
	3	1	466.8389	As↑B↑ SO_4^{2-}↑TDS↑	1.0	466.8389
	4	4	275.4066	As↑V↑ SO_4^{2-}↑TDS↑	1.0	275.4066
	5	3	234.7918	As↑Mo↑B↑SO_4^{2-}↑	1.0	234.7918

We further evaluate the search-engine like capability of our framework using Table 3, 4, 6, 7. Table 3 gives details of top 5 regions ranked by interestingness, and Table 6 visualizes these regions on the map of Texas. Table4 describes the top 5 co-location regions ranked by reward in more detail, and Table 7 depicts these regions on the map of Texas.

The parameter β affects the size of the co-location regions discovered. As displayed in Table 3, as β increases, CLEVER finds fewer, larger regions. For example, for β=2.0, CLEVER finds only 4 quite large regions capturing almost global patterns. Moreover, as β increases, the general trend is that purity of regions decreases as shown in purity column of Table 3. The algorithm was able to determine known areas of high arsenic concentrations as well as interesting unknown features. High arsenic is a well-known problem in the Southern Ogallala Aquifer in the Texas Panhandle and in the Southern Gulf Coast Aquifer north of the Mexican border. Figure 4 (Exp. 1) did recognize the higher arsenic concentration areas in the Panhandle (ranks 1, 2, and 3) associated with high molybdenum and vanadium but was also able to discriminate among companion elements such as fluoride (rank 1 area) or sulfate (rank 3 area). The Gulf Coast area (rank 4 area) is characterized by a boron marker, not present in the Panhandle, maybe suggesting different arsenic mobilization mechanisms. When the clusters are not as tightly defined (Exp. 2, Figure 5, larger β), they display the usually recognized extend of arsenic contamination in Texas: Ogallala Aquifer, Southern Gulf Coast, and West Texas basins. Areas delimited by clusters of ranks 4 and 5 are characterized by low arsenic but general chemistry similar to the high arsenic cluster (rank 1). A further loosening of cluster definition (Exp. 3, Figure 6) results in a display of the known, often described as sharp, boundaries between high and low arsenic areas in the Ogallala (ranks 2 and 4) and the Gulf Coast (rank 1 and 3) aquifers. In general, for β=1.3 and β=1.5 the discovered regions tend to lie inside Texas aquifers, which

Table 4: Top 5 Regions Ranked by Reward (as per formula 2.8)

Exp. No.	Top 5 Regi-ons	Region Size	Region Reward	Maximum Valued Pattern in the Region	Purity	Average Product for maximum valued pattern
Exp. 1	1	40	12684.6304	As↑Mo↑V↑	0.65	161.3194
	2	23	10270.49	As↑Mo↑V↑F^-↑	0.83	211.0179
	3	36	9431.1264	As↑B↑Cl^-↑TDS↑	0.58	153.2687
	4	11	2098.970187	As↑Mo↑V↑SO_4^{2-}↑	0.55	170.3873
	5	507	578.8116	As↓TDS↓	0.90	0.1968
Exp. 2	1	181	61684.5323	As↑Mo↑V↑F^-↑	0.49	52.1019
	2	80	24040.6315	As↑B↑Cl^-↑TDS↑	0.48	70.7322
	3	467	1884.8856	As↓TDS↓	0.91	0.2047
	4	23	701.7072	As↓Cl^-↑SO_4^{2-}↑TDS↑	0.78	8.1287
	5	189	587.9790	As↓F^-↓	0.78	0.2909
Exp. 3	1	833	1257170.945	As↑Mo↑V↑F^-↑	0.16	11.4334
	2	238	301539.908	As↑B↑Cl^-↑TDS↑	0.22	23.9052
	3	432	36754.1035	As↓TDS↓	0.93	0.2122
	4	152	7394.7640	As↓SiO_2↑WD↑	0.53	0.6006
	5	N/A				
Exp. 4	1	7	11669.7965	As↑B↑Cl^-↑TDS↑	1.0	630.1097
	2	117	10407.3250	As↑V↑F^-↑	0.91	12.8550
	3	4	2203.2526	As↑V↑ SO_4^{2-}↑TDS↑	1.0	275.4066
	4	2	1531.4887	As↑Mo↑V↑B↑	1.0	541.4630
	5	530	1426.9140	As↓TDS↓	0.90	0.1939

is expected, because wells inside the same aquifer are connected by water flow.

The algorithm also finds some inconsistent co-location sets. Inconsistency is observed in Table 3, Exp. 2 (Figure 5): the rank 3 region located in the area of the Hueco-Mesilla Bolson Aquifer is characterized by the co-location set {As↑Mo↑Cl^-↑TDS↑} and the rank 5 region in the Gulf Coast Aquifer has co-location set {As↓B↑Cl^-↑TDS↑}: As↑ is co-located with Cl^-↑ and TDS↑ in one region but As↓ is co-located with Cl^-↑ and TDS↑ in the other region. As displayed in Figure 5, the rank 3 region (in yellow color) is in West Texas, whereas the rank 4 region (in green color) is in South Texas. This discrepancy in regional co-location sets can be clearly attributed to geological differences in the two regions, emphasizing the need for regional co-location mining techniques.

Our approach was able to identify co-location sets of different sizes. Moreover, as we increase θ to 5, as expected, only co-location sets with purities above 90% are discovered. We also observe that the maximum reward region of Experiment 2 and the second ranked reward region of Experiment 4 occupy a similar spatial extent in North-West Texas. The first region is characterized by the co-location set {As↑Mo↑V↑F^-↑}, whereas the second region has the co-location set {As↑V↑F^-↑} associated with it and is slightly wider but significantly shorter than the first region. The dropping Mo↑ from the co-location set increases purity from 49% to 91%, but the average product drops from 52.1 to 12.8; this explains why the smaller co-location set is selected when θ is 5—but the larger set is better when θ is 1. When θ is decreased to 0, surprisingly, the complete dataset is returned as a single region with the co-location set of As↑Mo↑V↑F^-↑ with an average product of 5.95 and a purity of only 0.086.

Algorithm Comparison. In this subsection, we compare CLEVER with SPAM that was introduced in section 3. Table 5 lists the solution quality in terms of fitness value, and algorithm runtime for the algorithms CLEVER and

SPAM. These experiments were run using the same parameter settings which were listed in Table 2.

The algorithms CLEVER seeks the optimal number of regions, while SPAM discovers fixed number *k* of regions. In general, in our application it is hard to determine the optimal number of regions beforehand. SPAM obtains very low quality solution if k significantly deviates from the optimal number of regions. To do fair comparison between SPAM and CLEVER, we run SPAM 100 times using the no. of clusters discovered by CLEVER as the input. In the Table 5, reports the average fitness and average runtime over all the runs of SPAM.

Although SPAM was given a "good" k value as an input, its average fitness is significantly less than the fitness obtained by CLEVER. Only in one out of four experiments the maximum q(X) value obtained by SPAM is higher than the fitness achieved by CLEVER. In conclusion, CLEVER clearly outperforms SPAM with respect to quality of the regions discovered.

Table 5: Comparison of Algorithms (q(x)/ runtime in seconds)

Exp. No.	CLEVER	SPAM (average values over multiple runs)
Exp. 1	37809.19/41053.25	22333.13/108.13
Exp. 2	64700.63/4007.42	55094.24/45.99*
Exp. 3	1602859.72/7366.98	1344087/ 11.43
Exp. 4	31302.89/46335.39	4131.61/135.13

*: denotes that maximum fitness value achieved by SPAM over multiple runs is higher than that of CLEVER

We analyzed the run-time needed to conduct the experiments. Our algorithms have been developed using the open source, Java-based data mining and machine learning framework *Cougar^2*, which is developed by our research group [4]. All the experiments were conducted on the machine with 1.3 GHz of processor speed and 4 GB of memory. The machine runs RedHat Enterprise Linux 3 on an ia64 architecture. Our analysis shows that the CLEVER algorithm allocated more than 98% of its resources to the following two tasks: creating clusters for a given set of representatives and for fitness computations. With maximum pattern length set to 3, around 76% of time is allocated to computing q(X) and it takes around 1-2 hours for the algorithm to terminate. With maximum pattern length set to 4, 90% of the run time is allocated to fitness computations and in most cases the algorithm terminates in 6-15 hours. When additionally considering co-location sets of size 5, a program run usually takes between 18-36 hours. In summary, it is feasible to compute co-location sets up to size 5 or 6 on the described hardware with the CLEVER algorithm; for higher sizes a faster algorithms and/or faster hardware is necessary.

5 Discussion and Summary

This paper proposes a novel framework for mining co-locations patterns in spatial datasets. In contrast to past co-location mining research that centers on finding global co-location patterns in categorical datasets, co-location mining algorithms are introduced that operate in the continuous domain without need for discretization and discover regional patterns. The framework views regional co-location mining as a clustering problem in which an externally given reward-based fitness has to be maximized; in particular, fitness functions we employ in our approach, rely on products of z-scores of continuous variables to assess the interestingness of co-location patterns in the continuous space. A highly desirable feature of our approach is that it provides search-engine-like capabilities to scientists by returning regions ranked by the scientist's notion of interestingness that has been captured in a plug-in, reward-based fitness function.

Moreover, a novel, prototype-based region discovery algorithm named CLEVER has been introduced that uses randomized hill climbing and searches a variable number of clusters and larger neighborhood sizes to battle premature convergence.

The framework is evaluated in a case study involving chemical concentrations of Texas water wells centering on co-location patterns involving Arsenic. The tested region discovery algorithms were able to identify known and unknown regional co-location sets. Different sets of algorithm parameters lead to the characterization of arsenic distribution at different levels of granularity—stressing the need for parameterized, plug-in fitness functions that allow domain experts to express what patterns they are looking for at what level a granularity.

Arsenic water pollution is a serious problem for Texas and its causes are frequently difficult to explain, particularly for wells in the Ogallala aquifer [15]. A large number of possible explanations exist what causes high levels of arsenic concentrations to occur. Therefore, scientists face the problem to decide which hypotheses from a large set of hypotheses to investigate further. In general our regional co-location mining framework turned out to be valuable to domain experts in that it provided a data driven approach that suggests promising hypotheses for future research.

Table 6: Top 5 Regions Ranked by Interestingness

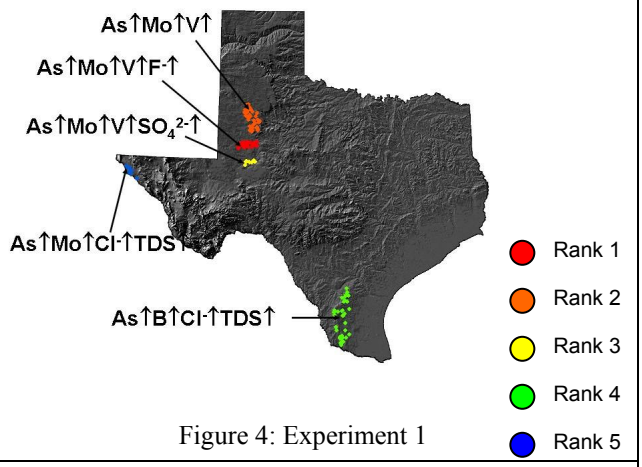

Figure 4: Experiment 1

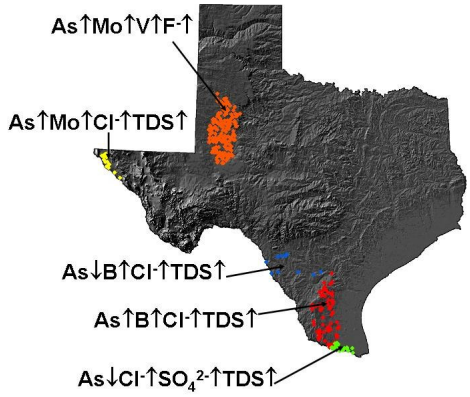

Figure 5: Experiment 2

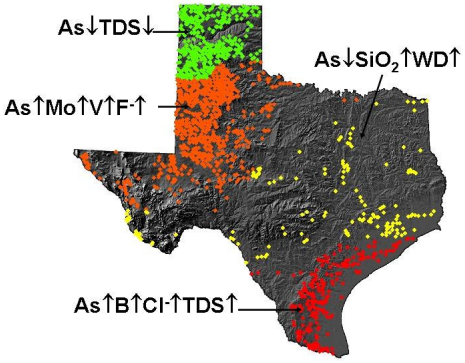

Figure 6: Experiment 3

Table 7: Top 5 Regions Ranked by Reward

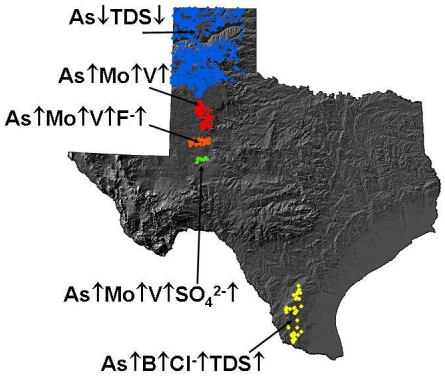

Figure 8: Experiment 1

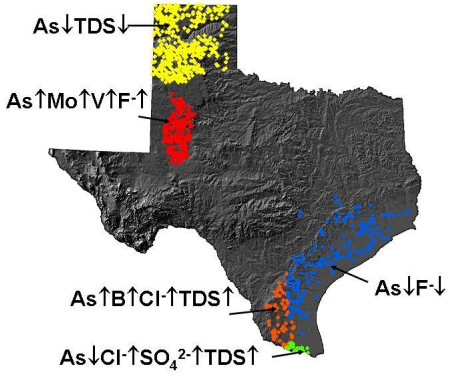

Figure 9: Experiment 2

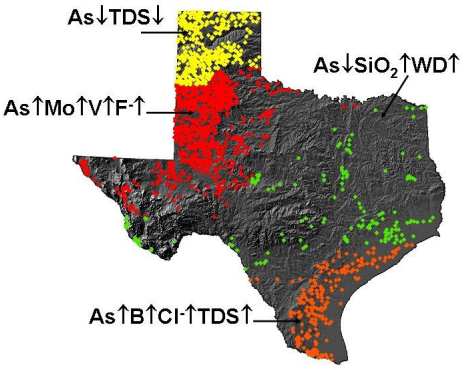

Figure 10: Experiment 3

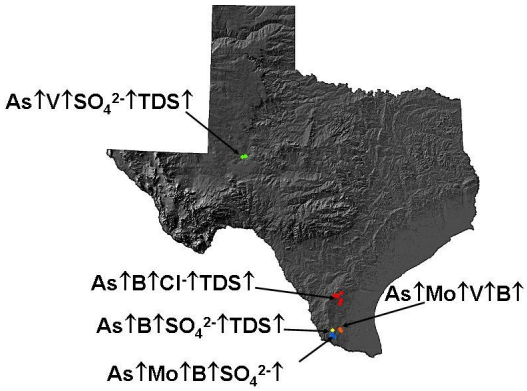

Figure 7: Experiment 4

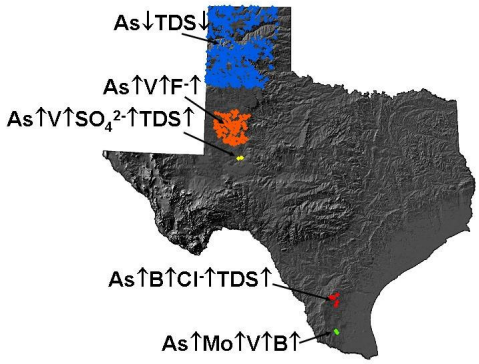

Figure 11: Experiment 4

References

[1] E. Achtert, C. Bőhm, H.P. Kriegel, P. Krőger, and A. Zimek, *Deriving quantitative models for correlation cluste,* In Proc. of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, Philadelphia, PA, 2006, pp. 4–13.

[2] Y. Aumann and Y. Lindell, *A Statistical Theory for Quantitative Association Rules,* In Proc. of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining San Diego, CA, 1999, pp. 261–270.

[3] T. Calders, B. Goethals, and S. Jaroszewicz, *Mining rank-correlated sets of numerical attributes,* In Proc. of the 12th ACM SIGKDD international Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, 2006, pp. 96–105.

[4] Cougar^2: Data Mining and Machine Learning Framework, https://cougarsquared.dev.java.net/.

[5] Data Mining and Machine Learning Group, University of Houston, http://www.tlc2.uh.edu/dmmlg.

[6] W. Ding, C. F. Eick, J. Wang, and X. Yuan, *A Framework for Regional Association Rule Mining in Spatial Datasets*, In Proc. of the Sixth international Conference on Data Mining, 2006, pp. 851–856.

[7] C.F. Eick, B. Vaezian, D. Jiang, and J. Wang, *Discovering of interesting regions in spatial data sets using supervised clustering,* In Proc. of the 10th European Conference on Principles of Data Mining and Knowledge Discovery, Berlin, Germany, 2006.

[8] A. Getis and J.K. Ord, *Local spatial statistics: an overview*, in Spatial analysis: modeling in a GIS environment, Cambridge, GeoInformation International, 1996, pp. 261–277.

[9] Y. Huang, J. Pei, and H. Xiong, *Mining co-location patterns with rare events from spatial data set,* Geoinformatica, 10, 2006, pp. 239–260.

[10] L. Kaufman and P. J. Rousseeuw, *Finding groups in data: An introduction to cluster analysis*, John Wiley and Sons, New Jersey, 2005.

[11] W. Klösgen and M. May, *Spatial Subgroup Mining Integrated in an Object-Relational Spatial Database*, In Proc. of the 6th European Conference on Principles of Data Mining and Knowledge Discovery, 2002, pp. 275–286.

[12] S.P. Lloyd, *Least squares quantization in PCM*, IEEE Trans. on Information Theory, 28, 1982, pp. 128–137.

[13] R. T. Ng and J. Han, *Efficient and Effective Clustering Methods for Spatial Data Mining,* In Proc. of the 20th International Conference on Very Large Databases, Santiago De Chile, Chile, 1994, pp. 144–155.

[14] R. Parmar, *Finding Regional Co-location Patterns using Representative-based Clustering Algorithms,* Master's Thesis, University of Houston, December 2007.

[15] B. R. Scanlon, J. P. Nicot et al., *Evaluation of arsenic contamination in Texas,* Final report prepared for TCEQ, under contract no. UT-08-5-70828, 2005.

[16] S. Shekhar and S. Chawla, *Spatial Databases: A Tour*, Prentice Hall, 2003.

[17] S. Shekhar and Y. Huang, *Discovering spatial co-location patterns: A summary of result,* Lecture Notes in Computer Science, 2121 (2001), pp. 236+.

[18] A. H. Smith et al, *Cancer risks from arsenic in drinking water*, Environmental Health Perspectives, 97, 1992, pp. 259–267.

[19] R. Srikant and R. Agrawal, *Mining Quantitative Association Rules in Large Relational Tables*, In Proc. of the ACM SIGMOD International Conference on Management of Data, Montreal, Quebec, Canada, 1996.

[20] Texas Water Development Board, http://www.twdb.state.tx.us/home/index.asp