

Towards Cohesive Anomaly Mining*

Yun Xiong

Research Center for
Dataology and DataScience
School of Computer Science
Fudan University
yunx@fudan.edu.cn

Yangyong Zhu

Research Center for
Dataology and DataScience
School of Computer Science
Fudan University
yyzhu@fudan.edu.cn

Philip S. Yu

Department of Computer
Science
University of Illinois at
Chicago
psyu@uic.edu

Jian Pei

School of Computing
Science
Simon Fraser University
jpei@cs.sfu.ca

Abstract

In some applications, such as bioinformatics, social network analysis, and computational criminology, it is desirable to find compact clusters formed by a (very) small portion of objects in a large data set. Since such clusters are comprised of a small number of objects, they are extraordinary and anomalous with respect to the entire data set. This specific type of clustering task cannot be solved well by the conventional clustering methods since generally those methods try to assign most of the data objects into clusters. In this paper, we model this novel and application-inspired task as the problem of mining cohesive anomalies. We propose a general framework and a principled approach to tackle the problem. The experimental results on both synthetic and real data sets verify the effectiveness and efficiency of our approach.

Introduction

Clustering, an essential data mining task, has been studied across various disciplines (Han, Kamber, and Pei 2011; Jain 2010). Conventionally, clustering methods assign most data objects to clusters. However, in some applications, a user may want to find compact clusters formed by a (very) small portion of objects in a large data set. Although in general it is still a clustering problem, it cannot be served well by the conventional clustering methods.

For example, downstream genes are regulated by human TFs (Transcription Factors). Zheng *et al.* (2008) showed that many TFs only regulate several or even one downstream genes. For instance, the TF *Adenosine deaminase domain-containing protein 2 (ADAD2)* only regulates gene *MUC5AC*. The *Actin filament-associated protein 1-like 1 (AFAPIL1)* only regulates gene *CAVI*. It is interesting to discover a small group of genes that share high similarity in expression, that is, likely to be regulated by a TF.

The task of finding small but compact clusters from bio-sequences is encountered frequently in bioinformatics, such

as analyzing genes and protein sequences. It is well recognized that, more often than not, only a very small number of sequences in a large data set may be similar to each other (Hastie *et al.* 2000; Dettling and Buhlmann 2002). Conventional clustering methods always suffer from a large number of false positives, since they assign most sequences to clusters.

As another example, consider detecting price manipulation groups in stock market (Xiong and Zhu 2009). In general, individuals' stock trading behaviors are assumed largely independent. However, a violating broker may abuse multiple accounts to manipulate market prices to gain illegal profit. To detect such faults, the accounts exhibiting common behaviors on a considerable number of days may be identified as suspects of manipulating market prices.

As the third example, Leskovec *et al.* (2008) and Leskovec *et al.* (2010) found that, in real life social networks, there are small communities with low conductance. However, as the size increases, the communities start to "blend in with the rest of the network and become less community like" (Leskovec *et al.* 2008). It is important to identify those small communities, which involve only a very small number of nodes in large networks.

Can we use the conventional clustering methods to tackle the problem of finding compact clusters formed by a small portion of objects in a large data set? Some traditional clustering algorithms, such as K-Means (Jain 2010), assign every object to a cluster (Deodhar *et al.* 2008), and thus cannot solve our problem. Moreover, those methods are often sensitive to outliers (Han, Kamber, and Pei 2011; Jain and Dubes 1988; Bohm *et al.* 2010). Although the density-based methods, such as DBSCAN (Ester *et al.* 1996) and DENCLUE (Hinneburg and Keim 1998), can cluster a subset of data into multiple dense regions, those methods are often sensitive to many parameters – using different parameter values often give dramatically different clusterings (Gupta and Ghosh 2008). Finding suitable parameter values is not trivial at all (Bohm *et al.* 2010). The hierarchical clustering methods, such as Single-Link Agglomerative clustering (Sibson 1973), may use a threshold to terminate the clustering early in order to limit the number of objects in clusters. However, linkage metrics-based hierarchical clustering suffers from high time complexity because of similarity computation (Dettling and Buhlmann 2002). Under reasonable assumptions, it may have complexity $O(N^2)$ (Berkhin

*This work was supported in part by the National Natural Science Foundation of China Grants 61170096, an NSERC Discovery Grant project, US NSF Grants IIS-0905215, CNS-1115234, IIS-0914934, DBI-0960443, and OISE-1129076, and Huawei Grant. Copyright © 2013, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

2002) or higher.

To the best of our knowledge, Bregman Bubble Clustering (BBC) (Gupta and Ghosh 2008) is the only unsupervised method designed with similar motivation. BBC identifies k dense clusters of a small subset of objects while ignoring the rest objects. BBC is shown to substantially outperform the conventional clustering algorithms, such as DBSCAN and Single-Link Agglomerative clustering, for finding clusters formed by a small portion of objects.

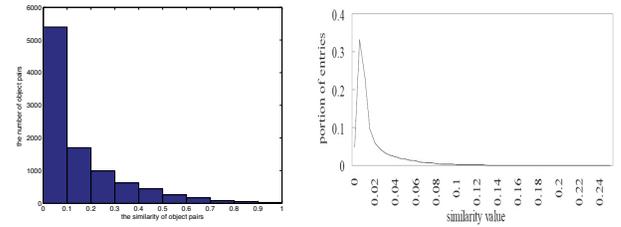
BBC takes a size threshold τ as the input and starts with k random centers. It first assigns each point to the nearest cluster representative, and then picks points closest to their representatives until τ points are picked. These steps are repeated until the assignment stabilizes. One drawback of BBC is that the number of clusters must be given a priori. The results of BBC may depend on the initial selection of cluster representatives. Because it is difficult for users to specify these representatives, one may have to run the algorithm multiple times with different initial cluster representatives to obtain good results in practice. Moreover, BBC tries to constrain both the number of points τ and the number of clusters k , so it falls into the case where all τ points must be clustered into k clusters. However, it is difficult for users to specify the appropriate k for a given τ .

In this paper, we formulate and tackle the problem of cohesive anomaly mining (CAM), which is novel and strongly inspired by important applications. Given a data set containing a large number of objects, we want to find cohesive clusters formed by a small number of objects that are more likely to form meaningful clusters than the rest. Since such cohesive clusters are comprised of a small number of objects, they are anomalous with respect to the entire data set. Accordingly, these clusters are regarded as cohesive anomalous (also called abnormal groups).

Contributions. We motivate and model the problem of mining cohesive anomalies, which is a non-trivial variation of clustering. Although Gupta and Ghosh (2008) motivated and justified the significance of the problem, as discussed above, the problem remains largely open for effective solutions. We propose a novel principled approach to tackle the problem. Our approach only needs a parameter, τ , the number of objects that those abnormal groups should cover. The parameter is easy to set. We develop an efficient abnormal groups mining algorithm. Since τ is typically small, our algorithm prunes a significant number of objects in mining, and thus results in substantial saving in computation. Unlike conventional clustering methods, our algorithm can mine abnormal groups flexibly without a predefined similarity threshold. We present experimental results on both synthetic and real data sets to verify the efficiency and effectiveness of the proposed method. Using real data, we also illustrate the applications of abnormal groups mining in biological analysis.

Problem Definition

We denote by \mathcal{F}^d a d -dimensional feature space. Let $\mathcal{O} = \{O_1, \dots, O_n\}$ be a set of objects in \mathcal{F}^d . We assume a similarity function $\mathcal{S}(\cdot, \cdot)$ that returns a value in $[0, 1]$ for two objects. The larger the similarity value is, the more similar two objects are. Given a similarity threshold δ , two object



(a) The distribution of similar- (b) The portions of similarity of
ity of object pairs on a real mi- object pairs on DBLP data set
croRNA data set

Figure 1: The distribution of similarity values.

O_i and O_j are said to be *similar* if $\mathcal{S}(O_i, O_j) \geq \delta$. As mentioned in (Han, Kamber, and Pei 2011), “measures of similarity $\mathcal{S}(\cdot, \cdot)$ can often be expressed as a function of measures of dissimilarity $dis(\cdot, \cdot)$, for example, $\mathcal{S} = \frac{1}{1+dis}$ ”. In this paper, we use Euclidean distance as the measure of dissimilarity between two objects that is by far the most popular choice in literature (Han, Kamber, and Pei 2011; Berkhin 2002). Our method can be extended to adopt other similarity measures.

In real applications, often an object is very similar to a small number of other objects. For example, the distribution of similarity of object pairs on a microRNA data set (Lu et al. 2005) is shown in Figure 1(a). Clearly, most object pairs are dissimilar. Only a very small number of pairs are similar. As another example, Yin, Han, and Yu (2006) showed the distribution of pairwise similarity values between 4,170 authors in DBLP data set (Figure 1(b)) and pointed out “majority of similarity entries have very small values which lie within a small range (0.005-0.015). While only a small portion of similarity entries have significant values (1.4% of similarity entries are greater than 0.1)”.

On a large data set where most object pairs are not similar, finding large clusters is not meaningful, since such large clusters are not compact. Instead, finding small but compact clusters formed by highly similar objects provides valuable insights. This is the problem we tackle in this paper.

Definition 1 (Critical Score of Objects, Critical Object)

The critical score ω of an object O_i is the largest similarity value between O_i and the other objects, that is,

$$\omega(O_i) = \max_{1 \leq j \leq n, j \neq i} \mathcal{S}(O_i, O_j)$$

Given a critical score threshold $\delta > 0$, an object O is a critical object if $\omega(O) \geq \delta$. We denote by \mathbb{O} the set of critical objects in \mathcal{O} .

Due to the symmetry of the similarity function, for each critical object $O \in \mathbb{O}$, there exists at least one object $O' \in \mathbb{O}$ such that O and O' are similar. Hence, each object in $\mathcal{O} \setminus \mathbb{O}$ is not similar to any other objects, no matter whether they are in \mathbb{O} or not. Since the critical objects are minority in the population, groups of critical objects are regarded as abnormal in the whole data set. We can use a critical score threshold to identify critical objects that may form compact small clusters.

Definition 2 (Abnormal Group) A set \mathbb{G} of critical objects is a potential abnormal group if $|\mathbb{G}| \geq 2$ and every two objects in \mathbb{G} are similar, that is, $\mathcal{S}(O_i, O_j) \geq \delta$ for $O_i, O_j \in \mathbb{G}$. \mathbb{G} is an abnormal group if there does not exist a proper superset of \mathbb{G} that is a potential abnormal group.

We measure the compactness of an abnormal group.

Definition 3 (Compactness) The compactness \mathcal{C} of an abnormal group \mathbb{G} is the total critical score of all objects in the group, that is, $\mathcal{C} = \sum_{O_i \in \mathbb{G}} \omega(O_i)$.

Let $\mathcal{G} \subset \mathcal{O}$ be a set of abnormal groups. Then, $\mathcal{O} \setminus \mathcal{G}$ is the set of objects not belonging to any group. We define the compactness of \mathcal{G} as the sum of the compactness of abnormal groups in \mathcal{G} .

The critical score threshold is hard to set. To overcome this difficulty, a user can specify a threshold τ on the number of objects in the set of critical groups to be found. The relationship between the two thresholds is as follows. We can rank the objects in descending order of the critical score. Given a typically small threshold τ ($\tau \geq 2$), we keep the top τ objects of the highest critical score. Then, the critical score of the τ -th object is the corresponding critical score threshold δ . That is, these τ objects have the highest critical scores, and compactness of \mathcal{G} consisting of τ objects is maximized.

Now, we are ready to define our problem.

Definition 4 (Cohesive Anomaly Mining) The problem of cohesive anomaly mining is to discover all abnormal groups in \mathcal{O} , such that the compactness of the set \mathcal{G} consisting of these groups is maximum and $|\mathcal{G}| = \tau$, where $\tau \geq 2$ is a given threshold.

Framework of the AGM Algorithm

In our task, one main challenge is to derive an appropriate critical score threshold δ such that exactly τ critical objects are selected to form abnormal groups. In this section, we develop an efficient algorithm, called AGM (Abnormal Group Mining). This algorithm is decomposed into two phases (Figure 2 presents the overview of AGM): The main task of Phase 1 is to find the most similar pairs of objects in a given data set and prune the pairs that cannot contain any critical objects. Critical objects are then computed from those surviving candidate pairs and are assigned into abnormal groups in Phase 2.

Since we are interested in only τ objects, and typically τ is very small, our algorithm prunes a significant number of non-promising pairs, and thus results in substantial saving in computation. The details will be discussed as follows.

Phase 1: Generate Candidates and Identify τ Critical Objects

We use a top similar pairs query algorithm to find top k_p similar pairs, and the objects in these pairs are regarded as candidate objects.

Definition 5 (Top k_p similar pair query) Given a set of objects \mathcal{O} and a similarity function, a top k_p pair query returns k_p pairs of objects having the largest similarity scores.

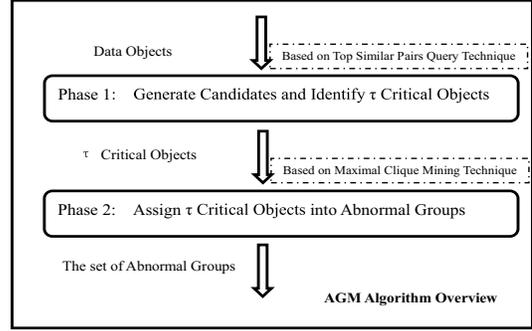


Figure 2: The framework of the AGM algorithm.

In order to identify τ critical objects from the candidate objects, we need to first compute an upper bound on the number of most similar pairs k_p to ensure that the τ critical objects are included.

A set of τ objects can form at most $\frac{\tau * (\tau - 1)}{2}$ pairs (the number of edges in a complete undirected graph with τ vertices). As a result, processing the $k_p = \frac{\tau * (\tau - 1)}{2}$ most similar pairs will yield at least τ objects. We have the following immediately.

Lemma 1 To obtain τ ($\tau \geq 2$) critical objects, we only need to consider no more than $k_p = \frac{\tau * (\tau - 1)}{2}$ most similar pairs.

As mentioned above, we use Euclidean distance as the dissimilarity measure, then the top k_p similar pairs can be regarded as the top k_p similar pairs with the closest distances. For the top k_p closest pairs query, one solution is to report all similar pairs in ascending order of distance incrementally (Hjalason and Samet 1998). Another method is to calculate all k_p most similar pairs in a shot to achieve the best pruning effect (Corral et al. 2004). We exploit the latter for two reasons. As Corral et al. (2004) showed that their algorithm outperformed the incremental ones by Hjalason and Samet (1998) through both the algorithmic and experimental comparisons. It exploits the relationship between bounding rectangles to prune irrelevant pairs. Moreover, we only need to get a small fraction of pairs. Note that, since there exist some techniques developed to answer top k_p similar pairs query under various similarity measures (Feng, Wang, and Li 2012; Xiao 2009), our method can be extended to other similarity measures.

So far, we can get the top k_p closest pairs. According to Lemma 1, τ critical objects are included in these pairs, hence all the objects in these pairs are regarded as candidate objects for identifying critical objects.

We use a list *Objlist* to hold the results. We determine which candidates can be inserted into the list *Objlist* by selecting the top τ objects with the largest critical score. Since the smallest dissimilarity value (the closest distance) means the largest similarity value ($\mathcal{S} = \frac{1}{1 + dis}$), the top k_p closest pairs can be regarded as ranking in descending order of the similarity values. Then we compute the critical score of the objects included the top k_p pairs. According to Definition 1, the critical score ω of an object O is the largest similarity

between O and the other objects, hence we can insert the objects in the k_p pairs into the list *Objlist* one by one from top to down until the number of the objects in *Objlist* reaches τ . These selected top τ objects have the largest critical score and are regarded as critical objects.

Phase 2: Assign τ Critical Objects into Abnormal Groups

We assign the τ critical objects into groups according to the similarity measure $S(\cdot, \cdot)$ and then return the set \mathcal{G} of these groups. According to Definition 2, every two objects in a potential abnormal group \mathbb{G} must be similar, therefore, if O_i is similar to O_j , O_j is similar to O_k and O_i is similar to O_k , then we can say O_i , O_j , and O_k can be assigned into a group. Furthermore, there cannot exist a proper superset of \mathbb{G} that is a potential abnormal group. Accordingly, an abnormal group is a maximal clique of similar objects and we adopt the maximal clique mining method (Cheng et al. 2011) to assign all the critical objects into the abnormal groups.

Choosing an Appropriate Threshold of τ

It is difficult to give a recommendation of τ for all applications because as Gupta and Ghosh (2008) pointed out, “dense regions are indeed highly correlated with the cluster labels; the confirmation of this phenomena is tantalizing considering the fact that the clustering process was completely unsupervised”. In some applications, 20% means a small portion; however, in term of detecting price manipulation groups in stock market, 1% of the investors means a very large number. Fortunately, an important observation is that, for abnormal group mining applications, users often have an expectation on the number for how many abnormal objects they want to handle, such as how many customers a company expects to maintain, how many genes a biologist wants to examine, and how many suspicious insider investors to be further monitored. The threshold τ economically captures the budget constraint in anomaly detection.

Experimental Results

In this section, we compare the efficiency of our method against that of BBC (Gupta and Ghosh 2008), one state-of-the-art clustering algorithm SynC (Bohm et al. 2010), one representative density-based clustering algorithms DBSCAN and the baseline algorithm NavAllPairs which computes the score of each object and then selects the top τ ones with the largest score on similarity values. Nonetheless, the main focus of the effectiveness comparison is with BBC as it is the only algorithm designed with similar goal in mind.

All experiments are conducted on an Intel(R) Core(TM) 2.5GHz CPU, and 4GB of main memory with Windows 7 Professional. We implemented AGM and NavAllpairs in Java and obtained SynC source code (in Java) from the authors. DBSCAN was implemented in R, and BBC was implemented in Matlab.

Data Sets. We used the synthetic data set and publicly available real data sets to evaluate our algorithm. 1) We generated a synthetic data set containing 100,000 objects, denoted as PSD. 2) The first real data set is a MicroRNAs (miRNAs) expression data set. It consists of 217 mam-

malian miRNAs from 334 samples, including multiple human cancers. The data set is from Gene Expression Omnibus (GEO) (<http://www.ncbi.nlm.nih.gov/geo>) with the series accession number GSE264 (Lu et al. 2005). We delete those with missing expression values and without corresponding family names. Finally, we collect 140 mammalian miRNAs with 12 samples about lung cancer, called MID. The second real data set is a yeast gene expression data set from (Tavazoie et al. 1999). We also delete the genes with missing value and without annotations in Gene Ontology (GO) database (<http://www.geneontology.org>). Finally, we collect 380 genes and 17 samples, called YED.

Efficiency Analysis

We compare the running time of our algorithm against that of NavAllPairs, BBC, SynC and DBSCAN on the PSD data set. DBSCAN needs to adjust parameter values (*eps* and *MinPts*) to discover the clusters among τ objects. We increase the *eps* threshold at an equal rate repeatedly in order to find the τ objects (or approximate to τ) and their clusters. In each round, if the size of the result computed by the DBSCAN is less than τ , the threshold is increased and the above process is repeated. More specifically, at first glance, we may use a threshold of *eps* $1 + 0.1 \times (i - 1)$ for the i -th round (*MinPts* is set to 4, as recommended by (Ester et al. 1996)). But, as mentioned above, our problem cannot be served well by the conventional clustering methods. Since the number k_i of objects returned by each round may be either more than or less than τ , *eps* should be set to $1 \pm 0.1 \times (i - 1)$ for the i -th round and one more parameter is required on the difference σ between k_i and τ in case the repetition cannot be terminated. SynC detects clusters automatically.

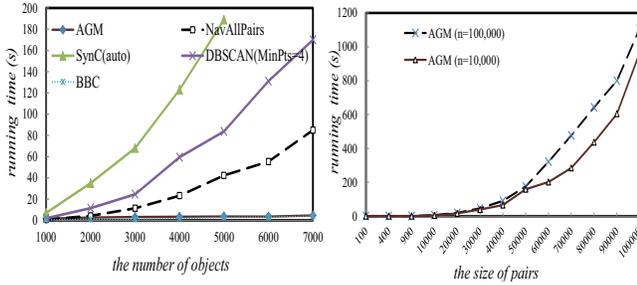
Figure 3(a) shows the running time for these algorithms as the input data size increases. For SynC, it runs 272 *sec* and 460 *sec* on 6,000 and 7,000 objects respectively, hence not showing on the Figure 3(a). It can be observed that, the running time of NavAllPairs, SynC and DBSCAN quickly exhibit sharp growth curve while AGM grows little. The reason is that DBSCAN needs repeated invocation and SynC automatically but iteratively adjusts the parameters in order to satisfy the MDL principle. BBC and AGM are not distinguishable, but AGM is more effective as mentioned in Section Effectiveness Analysis. Figure 3(b) shows the running time of AGM as the size of the most similar pairs increased up to 100,000 in data of size 100,000 and 10,000 respectively. It illustrates that our algorithm is suitable for a large data set. Since our goal is to mine abnormal groups, the AGM algorithm is efficient with the typically small τ .

Effectiveness Analysis

We report our results of effectiveness comparisons against that of BBC, DBSCAN on the real data sets¹.

We compare the result groups of our AGM algorithm with known functional classifications in order to detect the functional characteristics of the result groups. The P-value

¹For SynC, it cannot ensure the number (τ) of anomalies specified by the user. For the microRNA data set MID, its ARI is only 0.0794. Therefore, SynC is not included in the effectiveness analysis due to space limitation.



(a) Comparison of running time ($\tau=100$) (b) Running time of AGM

Figure 3: Efficiency analysis.

Table 1: P-value in AGM ($\tau=10\%$)

| GID | GO-ID | Description | p-value |
|-----|-------|---|-----------|
| 1 | 10741 | negative regulation of intracellular protein kinase cascade | 6.442e-04 |
| 2 | 6417 | regulation of translation | 9.517e-04 |
| 3 | 32543 | mitochondrial translation | 3.284e-04 |
| 4 | 32197 | transposition, RNA-mediated | 2.317e-04 |
| 5 | 18193 | peptidyl-amino acid modification | 2.999e-04 |
| 6 | 70150 | mitochondrial glycy1-tRNA aminoacylation | 4.832e-04 |
| 7 | 6364 | rRNA processing | 1.641e-03 |
| 8 | 31118 | rRNA pseudouridine synthesis | 1.288e-03 |
| 9 | 9208 | pyrimidine ribonucleoside triphosphate metabolic process | 9.663e-04 |
| 10 | 6406 | mRNA export from nucleus | 1.147e-04 |
| 11 | 462 | maturation of SSU-rRNA from tricistronic rRNA transcript | 2.220e-04 |

based on hypergeometric distribution is often used to estimate whether a group of genes/proteins is accumulated by chance. For each group, the P-value is calculated to measure the statistical significance for functional category enrichment (Jiang, Tang and Zhang 2004). For example, if the majority of genes in a group appear from one category, then it is unlikely that this happens by chance and the category's P-value would be close to 0. We expect a large fraction of the groups in the result to conform to the known classification.

For yeast expression data set YED, the P-value is calculated using BiNGO (Maere, Heymans, and Kuiper 2005) that maps over-represented functional themes in a given gene-set onto the GO hierarchy. It determines the predominant functional themes of a group of genes (the threshold of P-value is set to 0.05, accepting the default the BiNGO recommends). The number of objects τ is set to 10%. Table 1 shows the details of groups with enriched function in our result. It can be seen that every abnormal group identified has a low p-value, hence biological meaning. This is not true for the clusters obtained by BBC and DBSCAN as explained below.

Table 2 shows the result of BBC, where k is set to 7, 11 and 13 respectively. Note in the GID column, we use the notation $n-m$ to represent the m^{th} low P-value case with $k=n$. For both $k=7$ and $k=11$, BBC only reports one cluster with low P-value where the cluster is also identified in our result. Note that the other clusters cannot be reported as

Table 2: P-value in BBC ($\tau=10\%$, the number of clusters k is set to 7, 11 and 13 respectively)

| GID | GO-ID | Description | p-value |
|------|-------|-----------------------------|-----------|
| 7-1 | 32197 | transposition, RNA-mediated | 6.883e-04 |
| 11-1 | 32543 | mitochondrial translation | 1.924e-03 |
| 13-1 | 6417 | regulation of translation | 2.885e-05 |
| 13-2 | 32543 | mitochondrial translation | 3.284e-04 |
| 13-3 | 6364 | rRNA processing | 2.843e-04 |
| 13-4 | 15883 | FAD transport | 1.288e-03 |

Table 3: P-value in DBSCAN ($\tau=10\%$, $MinPts=4$)

| GID | GO-ID | Description | p-value |
|-----|-------|--|-----------|
| 1 | 6417 | regulation of translation | 8.924e-07 |
| 2 | 32435 | negative regulation of proteasomal ubiquitin-dependent protein catabolic process | 6.443e-04 |

their P-values are greater than 0.05, which means there is no predominant functional theme in these clusters. For $k=13$, BBC reports 4 clusters where three of them also appear in our result. Furthermore, these four clusters in BBC only cover 28.95% of the τ objects with a meaningful biological explanation. In fact, it is difficult to determine k . Table 3 shows the result of DBSCAN, where $MinPts$ is set to 4 as recommended by (Ester et al. 1996). It returns 2 meaningful clusters covering 21% of the τ objects with low P-value where one of them also appears in our result. For both BBC and DBSCAN, most of the top τ objects selected do not form meaningful clusters, i.e., with biological meaning. This is in contrast to AGM.

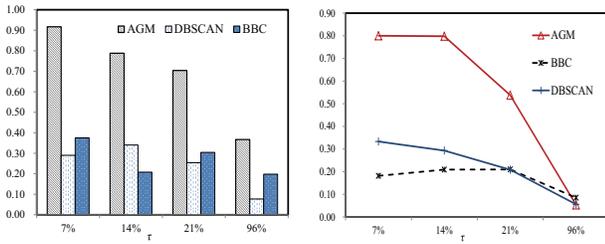
Next we consider the microRNA data set MID, we use the scoring scheme used in (Li et al. 2008) to determine how effectively a predicted cluster (or abnormal group) (\mathbb{G}) matches a known category (\mathbb{C}). The overlapping score $\mathcal{S}(\mathbb{G}, \mathbb{C})$ between a predicted cluster \mathbb{G} and a known category \mathbb{C} is calculated by the formula: $\mathcal{S}(\mathbb{G}, \mathbb{C}) = \frac{|V_{\cap}|}{|V_{\mathbb{G}}|}$ where $|V_{\cap}|$ is the size of the intersection set of the predicted cluster and the known category, and $|V_{\mathbb{G}}|$ is the size of the predicted cluster. A known category \mathbb{C} that has no genes in a predicted cluster \mathbb{G} has $\mathcal{S}(\mathbb{G}, \mathbb{C}) = 0$ and a known category \mathbb{C} that perfectly matches a predicted cluster \mathbb{G} has $\mathcal{S}(\mathbb{G}, \mathbb{C}) = 1$. Generally speaking, the more known categories are matched by algorithm, the stronger identification ability the algorithm has. The result is shown in Figure 4(a).

Furthermore, we exploit *Adjusted Rand Index* (ARI) (Hubert and Arabie 1985):

$$ARI = \frac{\sum_{i=1}^{|\mathbb{C}|} \sum_{j=1}^{|\mathbb{G}|} \binom{N_{ij}}{2} - [\sum_{i=1}^{|\mathbb{C}|} \binom{N_i}{2}] \sum_{j=1}^{|\mathbb{G}|} \binom{N_j}{2} / \binom{N}{2}}{\frac{1}{2} [\sum_{i=1}^{|\mathbb{C}|} \binom{N_i}{2} + \sum_{j=1}^{|\mathbb{G}|} \binom{N_j}{2}] - [\sum_{i=1}^{|\mathbb{C}|} \binom{N_i}{2}] \sum_{j=1}^{|\mathbb{G}|} \binom{N_j}{2} / \binom{N}{2}}$$

ARI returns 1 for a perfect agreement between clusters (or abnormal groups) and classifications and 0 when the result is as bad as random assignments (Gupta and Ghosh 2008). The result is shown in Figure 4(b).

Figure 4(a) and 4(b) show that the result abnormal groups of our AGM algorithm have higher overlapping score and better ARI than both DBSCAN and BBC with the same τ value. It indicates that the AGM algorithm has strong identification ability especially as the τ value is relatively small.



(a) Evaluation using the overlapping score

(b) Evaluation using ARI

Figure 4: Effectiveness Analysis.

Table 4: P-value for each group with different τ

| GId | $\tau=7\%$ | $\tau=14\%$ | $\tau=21\%$ | $\tau=28\%$ | $\tau=35\%$ |
|-----|------------|-------------|-------------|-------------|-------------|
| 1 | 4.55e-05 | 4.55e-05 | 4.55e-05 | 4.55e-05 | 1.36e-04 |
| 2 | 1.94e-03 | 1.94e-03 | 1.94e-03 | 1.94e-03 | 1.94e-03 |
| 3 | 4.55e-05 | 4.55e-05 | 4.55e-05 | 4.55e-05 | 4.55e-05 |
| 4 | 4.55e-05 | 4.55e-05 | 4.55e-05 | 4.55e-05 | 4.55e-05 |
| 5 | | 4.42e-04 | 4.42e-04 | 4.42e-04 | 4.42e-04 |
| 6 | | 3.78e-02 | 9.56e-04 | 1.64e-03 | 1.85e-04 |
| 7 | | 2.83e-02 | 5.60e-02 | 5.60e-02 | 5.60e-02 |
| 8 | | 1.90e-03 | 1.90e-03 | 1.90e-03 | 1.90e-03 |
| 9 | | | 3.74e-02 | 3.74e-02 | 2.84e-02 |
| 10 | | | 2.83e-02 | 1.31e-01 | 1.08e-01 |
| 11 | | | 7.28e-02 | 8.21e-02 | 8.98e-02 |
| 12 | | | | 8.98e-02 | 1.35e-04 |
| 13 | | | | 1.35e-04 | 5.54e-02 |
| 14 | | | | | 9.56e-03 |
| 15 | | | | | 4.78e-03 |
| avg | 3.42e-05 | 1.76e-05 | 1.37e-05 | 1.10e-05 | 0.68e-05 |

Furthermore, it can be observed that, the smaller the τ is, the larger the score and ARI are. When τ is set to 96%, both the score and ARI are very low. The reason is that the larger the τ is, the lower the threshold of similarity is. Thus more objects are required to be clustered. However, those individual objects cannot be clustered in real biological significance, so they become false positives when τ is large. In such cases, we are inclined to discover abnormal groups formed by top τ similar objects instead of clusters of most objects.

We now calculate its P-value for each cluster to assess the effectiveness of the algorithm according to the formula (Jiang, Tang, and Zhang 2004): $P = 1 - \sum_{i=0}^{m-1} \frac{\binom{|C|}{i} \binom{|\mathcal{O}| - |C|}{|G| - i}}{\binom{|\mathcal{O}|}{|G|}}$, where $|\mathcal{O}|$ is the total number of genes/proteins in the data set, $|G|$ is the size of the predicted cluster, $|C|$ is the size of a family, and m is the number of genes/proteins of the family in the predicted cluster G . The functional classification of microRNA used can be collected from the microRNA database (<http://www.mirbase.org/>). (Note that, the above mentioned BiNGO is consistent with this formula in terms of P-value calculation, but it does not map functional themes for microRNA data set onto the GO hierarchy.)

The result in Table 4 shows that the result clusters have low P-value. Furthermore, the smaller the τ is, the lower the average P-value is. It indicates that AGM is more effective when the τ is relatively small.

Related Work

Clustering Conventional clustering is to partition most objects into clusters. Instead, our task is to partition a small portion of objects into abnormal groups. Though finding the top- k clusters (k is the number of the clusters) was presented in a few work (Macropo and Ambuj 2010; Jiang, Pei and Zhang 2005), an obvious difference is that k cannot determine the number of the objects in the clusters, i.e., the result may contain a large amount of objects. It is necessary to repeat the algorithms by adjusting the quality function of clusters (Jiang, Pei and Zhang 2005), but it is difficult and has redundant computation. A variation is to rank all the clusters and keep just a few (τ objects), but it is extremely inefficient. One recent paper has focused on finding the best clusters on large graphs (Macropo and Ambuj 2010) that uses a parameter of the number of the objects to control the memory usage and increase scalability.

Outlier Mining Most research on outlier mining focuses on the isolated objects (i.e., point outliers) (Chandola, Banerjee, and Kumar 2009). A common characteristic between point outliers detection and our task is to discover minority objects in a data set. However, for our problem, most objects are individual outliers while relatively few objects are similar and form abnormal groups; for outlier mining problem, minority objects are outliers while most objects are similar.

There exist a few works addressing the case where the outliers are clustered (micro-clusters or clustered anomalies) (Papadimitriou 2003; Liu, Ting, and Zhou 2010). They assign similar anomalies into micro-clusters, but also assume most data points fall into normal clusters. Here we assume most data points do not belong to any cluster and can be filtered out.

Collective Anomalies (Chandola, Banerjee, and Kumar 2009) suggested that a collection of related data is termed a collective anomaly if this collection is anomalous with respect to the entire data set. Three types of relations that have been exploited are sequential, spatial, and linkage relations. The reason why they are assigned together to form a collective anomaly (e.g., subsequence or subgraph) is that they have some relationships instead of similarity (Liu, Ting, and Zhou 2010). By contrast, the objects in an abnormal group are similar. Furthermore, we extend collective anomalies by utilizing the parameter τ that has not been mentioned before.

Rare Category Detection and Imbalanced Classification Imbalanced classification aims to construct a classifier in order to partition data objects into both the majority and the minority classes (Chawla 2009), which obviously differs from our problem. Rare category detection (He 2010) is to find at least one example from each minority class with the help of a labeling oracle. Rare category characterization and imbalanced classification require labeled training examples.

Conclusion

In this paper, we motivate and model a novel problem, mining cohesive anomalies (abnormal groups), inspired by important applications. It is an interesting and challenging task. The superiority of our proposed algorithm is demonstrated by extensive experiments on real and synthetic data sets.

Reference

- Beckmann, N.; Kriegel, H. P.; Schneider, R.; *et al.* 1990. The R*-tree: an Efficient and Robust Access Method for Points and Rectangles. In *Proc. of SIGMOD*, 322-331.
- Berkhin, P. 2002. Survey of Clustering Data Mining Techniques. Technical report, Accrue Software.
- Bohm, C.; Plant, C.; Shao, J.; *et al.* 2010. Clustering by Synchronization. In *Proc. of KDD*, 583-592.
- Chandola, V.; Banerjee, A.; and Kumar, V. 2009. Anomaly Detection: A Survey. *ACM Computing Surveys*, 41(3): 1-58.
- Chawla, N. 2009. Mining When Classes are Imbalanced, Rare Events Matter More, and Errors Have Costs Attached. In *Proc. of SDM*.
- Cheng, J.; Ke, Y. P.; and Fu, A. W.; *et al.* 2011. Finding Maximal Cliques in Massive Networks. *ACM Transactions on Database Systems*, 36(4): 1-34.
- Corral, A.; Manolopoulos, Y.; and Theodoridis, Y.; *et al.* 2004. Algorithms for Processing K-closest-pair Queries in Spatial Databases. *Data & Knowledge Engineering*, 67-104.
- Deodhar, M.; Cho, H.; Gupta, G.; *et al.* 2008. Robust Overlapping Co-clustering. Technical report, The University of Texas at Austin.
- Dettling, M., and Buhlmann P. 2002. Supervised Clustering of Genes. *Genome Biology*. 3(12).
- Ester, M.; Kriegel, H. P.; and Sander J.; *et al.* 1996. A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proc. of KDD*, 226-231.
- Feng, J. H.; Wang J. N.; and Li, G. L. 2012. Trie-join: a Trie-based Method for Efficient String Similarity Joins. *VLDB Journal*, 21: 437-461.
- Gupta, G., and Ghosh, J. 2008. Bregman Bubble Clustering: A Robust Framework for Mining Dense Clusters. *ACM Transactions on Knowledge Discovery from Data*, 2(2): 1-49.
- Han, J. W.; Kamber, M.; and Pei, J. 2011. *Data Mining: Concepts and Techniques (3rd Edition)*. Morgan Kaufmann.
- Hastie, T.; Tibshirani, R.; and Eisen, M. B.; *et al.* 2000. Gene Shaving as a Method for Identifying Distinct Sets of Genes with Similar Expression Patterns. *Genome Biology*, 1, 1-21.
- He, J. R. 2010. *Rare Category Analysis*. Carnegie Mellon University.
- Hinneburg, A., and Keim, D. A. 1998. An Efficient Approach to Clustering in Large Multimedia Databases with Noise. In *Proc. of KDD*, 58-65.
- Hjalason, G. R., and Samet, H. 1998. Incremental Distance Join Algorithms for Spatial Databases. In *Proc. of SIGMOD*, 237-248.
- Hubert, L., and Arabie, P. 1985. Comparing Partitions. *Journal of classification*, 2(1): 193-218.
- Jain, A. K. 2010. Data clustering: 50 Years Beyond K-Means. *Pattern Recognition Letters*, 31(8): 651-666.
- Jain, A. K., and Dubes, R. C. 1988. *Algorithms for Clustering Data*. Prentice Hall.
- Jiang, D. X.; Pei, J.; and Zhang A. D. 2005. A General Approach to Mining Quality Pattern-Based Clusters from Microarray Data. In *Proc. of DASFAA*, 188-200.
- Jiang, D. X.; Tang, C.; and Zhang, A. D. 2004. Cluster Analysis for Gene Expression Data: A Survey. *IEEE Transaction on Knowledge and Data Engineering*, 16(11): 1370-1386.
- Leskovec, J.; Lang, K. J.; and Dasgupta, A.; *et al.* 2008. Statistical Properties of Community Structure in Large Social and Information Networks. In *Proc. of WWW*, 695-704.
- Leskovec, J.; Lang, K. J.; and Mahoney, M. W. 2010. Empirical Comparison of Algorithms for Network Community Detection. In *Proc. of WWW*, 631-640.
- Li, M.; Chen, J. E.; and Wang, J. X.; *et al.* 2008. Modifying the DPCLUS Algorithm for Identifying Protein Complexes based on New Topological Structures. *BMC Bioinformatics*, 9: 398.
- Liu, F. T.; Ting, K. M.; and Zhou, Z. H. 2010. On Detecting Clustered Anomalies using SCiForest. In *Proc. of ECML/PKDD*, 274-290.
- Lu, C. T.; DosSantos, R. F; and Liu, X.; *et al.* 2011. A Graph-based Approach to Detect Abnormal Spatial Points and Regions. *International Journal on Artificial Intelligence Tools*, 20(4): 721-751.
- Lu, J.; Getz, G.; Miska, E. A.; *et al.* 2005. MicroRNA Expression Profiles Classify Human Cancers. *Nature*, 435(9): 834-838.
- Macropo, K., and Ambuj, S. 2010. Scalable Discovery of Best Clusters on Large Graphs. In *Proc. of VLDB*, 693-702.
- Maere, S.; Heymans, K.; and Kuiper, M. 2005. BiNGO: a Cytoscape Plugin to Assess Overrepresentation of Gene Ontology Categories in Biological Networks. *Bioinformatics*, 21(16): 3448-3449.
- Papadimitriou, S.; Kitagawa, H.; and Gibbons, P. B.; *et al.* 2003. Loci: Fast Outlier Detection using the Local Correlation Integral. In *Proc. of ICDE*, 315-327.
- Sibson, R. 1973. SLINK: an Optimally Efficient Algorithm for the Single Link Cluster method. *Computer Journal*, 16: 30-34.
- Tavazoie, S.; Hughes, J. D.; and Campbell, M. J.; *et al.* 1999. Systematic Determination of Genetic Network Architecture. *Nature Genetics*, 22: 281-285.
- Xiao, C.; Wang, W.; and Lin, X.; *et al.* 2009. Top-k Set Similarity Joins. In *Proc. of ICDE*, 916-927.
- Xiong, Y., and Zhu, Y. Y. 2009. Mining Peculiarity Groups in Day-by-Day Behavioral Datasets. In *Proc. of ICDM*, 578-587.
- Yin, X. X.; Han, J. W.; and Yu, P. S. 2006. LinkClus: Efficient Clustering via Heterogeneous Semantic Links. In *Proc. of VLDB*, 427-438.
- Zheng, G. Y.; Tu, K.; and Yang, Q.; *et al.* 2008. ITFP: an Integrated Platform of Mammalian Transcription Factors. *Bioinformatics*, 24(20): 2416-2417.