11-2012

Mining coherent anomaly collections on web data

Hanbo DAI
Singapore Management University

Feida ZHU
Singapore Management University, fdzhu@smu.edu.sg

Ee-peng LIM
Singapore Management University, eplim@smu.edu.sg

Hwee Hwa PANG
Singapore Management University, hhpang@smu.edu.sg

DOI: https://doi.org/10.1145/2396761.2398472

Mining Coherent Anomaly Collections On Web Data

Hanbo Dai, Feida Zhu, Ee-Peng Lim, and HweeHwa Pang
School of Information Systems, Singapore Management University
Singapore
hanbo.dai.2008@smu.edu.sg, fdzhu@smu.edu.sg, eplim@smu.edu.sg,
hhpang@smu.edu.sg

ABSTRACT

The recent boom of weblogs and social media has attached increasing importance to the identification of suspicious users with unusual behavior, such as spammers or fraudulent reviewers. A typical spamming strategy is to employ multiple dummy accounts to collectively promote a target, be it a URL or a product. Consequently, these suspicious accounts exhibit certain coherent anomalous behavior identifiable as a collection. In this paper, we propose the concept of *Coherent Anomaly Collection (CAC)* to capture this kind of collections, and put forward an efficient algorithm to simultaneously find the top-K disjoint CACs together with their anomalous behavior patterns. Compared with existing approaches, our new algorithm can find disjoint anomaly collections with coherent extreme behavior without having to specify either their number or sizes. Results on real Twitter data show that our approach discovers meaningful and informative hashtag spammer groups of various sizes which are hard to detect by clustering-based methods.

Categories and Subject Descriptors

H.2.8 [**Database applications**]: Data mining

General Terms

Algorithms, Design, Experimentation

Keywords

Anomaly/Outlier Detection, Anomaly Collection/Cluster

1. INTRODUCTION

The recent boom of weblogs and social media has provided an unprecedented degree of freedom for ordinary users to generate content online. At the same time, the openness of the platforms leaves them highly susceptible to user abuse, and, even worse, the sheer volume of the generated data makes it infeasible to manually inspect their veracity. To find trustworthy information in these data, it is increasingly important to automatically identify suspicious users with unusual behavior, which are in many cases spammers or fraudulent reviewers. In real life, in order to draw attention amid this information swamp, spammers rarely operate with just a single account. Instead, they typically employ multiple dummy accounts to collectively promote certain targets, such as a URL or a product. For example, in Twitter, a group of users may collaboratively spam on popular hashtags to promote their websites or businesses. Their strategy is to post a large number of tweets containing both their advertisement content and the popular hashtags, so that other users querying any of these hashtags would see their spamming tweets. These activities are classified as spamming according to Twitter's rules[1]. Figure 1 shows three collections of real spammers in Twitter detected by our approach.

These observations show that the key to detecting suspicious collaborative accounts is to identify their *shared* anomalous behavior patterns. We call such a user group a *Coherent Anomaly Collection(CAC)*, and propose an information theory based definition to characterize it.

Few existing studies have focused on collective anomaly detection [2], [4], [7], [5], and [8]. Furthermore, their frameworks are not appropriate for collective anomalies of extreme behaviors. The concept of an anomaly collection with extreme behavior was introduced in [3], which proposed algorithms to find top-K anomaly collections no greater than a user-specified size. However, in that work the anomaly collections are not optimized for the coherence in their unusual behavior, resulting in multiple spammer groups clustered in the same collection. Furthermore, the top-K anomaly collections heavily overlap one another and are size-bounded by user-specified constraints, offering limited information on the true anomaly collections in a data set. In this paper, we propose to simultaneously find top-K disjoint coherent anomaly collections together with their anomalous behavior patterns, without having to specify either the number or sizes of the target collections.

Our contributions are summarized as follows:

- We propose for the first time the concept of coherent anomaly collection and the problem of detecting top-K disjoint coherent anomaly collections.
- We introduce a heuristic algorithm based on the properties of p-values to efficiently sample candidate collections with potentially high anomaly scores, which

[1]http://support.twitter.com/articles/18311-the-twitter-rules

CIKM'12, October 29–November 2, 2012, Maui, HI, USA.

Figure 1: Three collections of real Twitter users ranked in descending order (left to right) by the usage frequency of four hashtags (labeled as f_1 to f_4). $S_1 = \{e_1, e_2, e_3\}$ = {blackberrypros, randomwireless, greenerblogs}, $S_2 = \{e_4, e_5, \ldots, e_{10}\}$= {LiveSEXsheLOve, LOVEsexyFREE, PornLOveCamFree, SexFullFreeCam, SEXsheylaPOrn, SHEYLLAsexPORN, LOVEsexCamFRee4} and $S_3 = \{e_{11}, e_{12}, \ldots, e_{15}\}$ = {SexFreeLive1, ChinaSexPOrnFRE, SEXYloveCAMFree, TokioSEXfreeLiv, FullSEXpornFREE}.

are then subject to our proposed coherence check using the idea of matrix encoding cost from information theory.

- We apply our algorithm on Twitter data to detect hashtag spammer collections. The results show that we are able to discover meaningful and informative spammer collections which are otherwise hard to find by existing approaches.

The rest of the paper is organized as follows. We formulate our problem in Section 2. The algorithm is presented in Section 3, and Section 4 reports on experiments. We conclude in Section 5.

2. COHERENT ANOMALY COLLECTION

To define a coherent anomaly collection, we show how to measure the anomalousness for a given collection in Section 2.1 and its coherence in Section 2.2, followed by the problem definition.

2.1 Measuring Anomalousness

As shown in [3], capturing a collection of anomalies requires a measure of the anomalousness of multiple entities as a collection. We adopt the definitions in [3] to measure the anomalousness of a collection in the following exposition. The advantage of this definition is that it is defined directly at the collection level, which is different from measuring individually on entity level followed by aggregating over the whole collection.

Denote the entity universe as E, and the feature universe as F. When E is ranked by a feature of F, **extremity index** r ($1 \leq r < |E|/2$) is defined to indicate the top r positions in the ranking. Given a set of entities $S \subset E$, a feature $f \in F$ and r, the **extreme subset** of S denoted as $E_f(S, r)$ is the set of entities in S which appear in top-r positions w.r.t. feature f. For a given r, the extremity of S is quantified by the cardinality of $E_f(S, r)$, denoted as i. It turns out that i is a random variable following the hypergeometric distribution. This is because if a set S is randomly picked from $|E|$ ranked entities, the number of entities in S that appear in top r positions follows the hypergeometric distribution. Thus the probability of observing i entities of S appearing in top r positions is $prob(i, |E|, r, |S|) = \frac{\binom{r}{i} \cdot \binom{|E|-r}{|S|-i}}{\binom{|E|}{|S|}}$.

The **p-value** of S w.r.t. extremity index r and feature f, denoted as $p_f(S, r)$, is the probability of observing at least i entities of a random collection S appearing in top r positions w.r.t. f. Thus, $p_f(S, r) = \sum_{j=i}^{min(r,|S|)} prob(i, |E|, r, |S|)$.

By definition, S has different p-values, each corresponding to a different r. For any given r and f, the smaller the p-value of S, the more anomalous or extremely ranked S is. Therefore, among all the choices of r, pick the one which gives the smallest p-value that S could possibly have as the **representative extremity index** of S, denoted as $\widehat{r}_f(S)$. Correspondingly, the **representative p-value** of S w.r.t. f is denoted as $\widehat{p}_f(S)$, i.e., $\widehat{p}_f(S) = p_f(S, \widehat{r}_f(S))$.

Formally, an anomaly collection is defined as follows.

DEFINITION 1. *Given an entity universe E and an entity set S, $S \subset E$, a set of independent features F and a threshold α, S is an **Anomaly Collection (AC)** w.r.t. F if (I) $\exists F^S \subseteq F$ such that $\forall f \in F^S, \widehat{p}_f(S) \leq \alpha$; (II) $1 < |S| < |E|/2$; (III) $|F^S| > 1$.*

The condition $1 < |S| < |E|/2$ is imposed, as an anomaly collection should contain more than one entity and yet remain the minority of the population. The condition $|F^S| > 1$ requires that S is significant in at least two statistical tests. F^S is called the **significant features** of S. The definition also requires a set of independent features F. The independency of any two features is defined by statistics including Kendall Tau rank correlation coefficient [6].

As the representative p-value measures how anomalous an AC is for a single feature, the **anomaly score** of an AC S for F, denoted as $\Omega(S, F)$, is defined as the product of the representative p-values for significant features. As the resulting score is usually small, take the log form $\Omega(S, F) = -\sum_{f \in F^S} \log \widehat{p}_f(S)$. The more features on which S is significant and the more extremely ranked S is w.r.t. each of them, the larger anomaly score S has.

2.2 Measuring Coherence

By definition, the anomaly score of an entity collection is determined by the subset of its members which are most extremely ranked w.r.t. some features. It is possible that different subsets of members are extremely ranked w.r.t. different feature subsets. However, for many applications, we are most interested in ACs whose members are extremely ranked w.r.t the same set of features. For instance, in our Twitter example in Figure 1, we prefer to identify S_1, S_2 and S_3 as three different ACs instead of consider them as a single AC.

To capture this important notion of coherence in our problem definition, we formally define "coherence" by first representing an AC in a matrix form and using the matrix encoding cost from information theory in [1] to evaluate the coherence of the AC.

For a given AC S and its significant feature set F^S, we denote $E(S, F^S)$ as the members of S that appear in the positions indicated by the representative extremity index of any significant feature, i.e., $E(S, F^S) = \bigcup_{f \in F^S} E_f(S, \widehat{r}_f(S))$.

To tell how coherent an AC is, we represent it by a $|F^S|$

by $|E(S,F^S)|$ matrix. Specifically, given an AC S, its significant feature set F^S and its extreme subset $E(S,F^S)$, with f_a,$(a = 1,...,|F^S|)$ being the a-th feature in F^S and e_b,$(b = 1,...,|E(S,F^S)|)$ being the b-th entity in $E(S,F^S)$, the **extreme matrix** is $M(S) = [m_{ab}]$, where

$$m_{ab} = \begin{cases} 1, & \text{if } e_b \in E_{f_a}(S, \hat{r}_{f_a}(S)); \\ 0, & \text{otherwise.} \end{cases}$$

According to [1], any matrix can be encoded as one or multiple row and column clusters. The encoding cost is the sum of the code cost and description cost, where the first cost is for encoding each row and column cluster and the second cost is for describing the grouping information. If a matrix is highly homogeneous, e.g., containing all 1s or all 0s like $M(S_1)$, its encoding cost as one cluster is low. If a matrix is not homogeneous, e.g., containing multiple homogeneous clusters like $M(S_1 \cup S_2)$, we should be able to find a minimum cost to encode this matrix by first encoding each homogeneous cluster within and then describing the grouping information of these clusters. Moreover, this cost is expected to be much lower than the cost of encoding the original matrix as one single cluster.

DEFINITION 2. ***[CAC]*** *Given an entity universe E and an entity set S, $S \subset E$, a set of independent features F and a threshold α, S is a* ***Coherent Anomaly Collection (CAC)*** *if (I) S is an anomaly collection; (II) the cost of encoding $M(S)$ as one cluster is lower than the minimum cost of encoding $M(S)$ as multiple homogeneous clusters; (III) the number of 1s in its extreme matrix must be greater than half of the size of its extreme matrix.*

We formally define our problem of detecting top-K disjoint CACs as follows.

DEFINITION 3. ***[TOPK_CAC]*** *Given K, the entity universe E, a set F of independent features and the ranking of E on F, let $S^* = (S_1, S_2, \ldots, S_N)$ be the sequence of coherent anomaly collections ranked in descending order by their anomaly scores. The problem of* TOPK_CAC *is to find the length-K disjoint subsequence $\hat{S}$ of S^* where $\hat{S} = (S_{\hat{r}_1}, S_{\hat{r}_2}, \ldots, S_{\hat{r}_K})$ such that (I) $S_{\hat{r}_i} \bigcap S_{\hat{r}_j} = \emptyset$ and $1 \le \hat{r}_i < \hat{r}_j \le N$, for $1 \le i < j \le K$; and (II) for any other length-K disjoint subsequence S' of S^* where $S' = (S_{r'_1}, S_{r'_2}, \ldots, S_{r'_K})$ such that $S_{r'_i} \bigcap S_{r'_j} = \emptyset$ and $1 \le r'_i < r'_j \le N$, for $1 \le i < j \le K$, there exists an index $j, 1 \le j \le K$ such that $\hat{r}_i \le r'_i$, for all $1 \le i \le j$.*

3. ALGORITHM

We describe our algorithm for the TOPK_CAC problem in this section. Conceptually, we first find the top-1 CAC and then find the next most anomalous CACs that do not overlap with any of the previously detected CACs, and so on so forth. The algorithm would mine the most anomalous CAC with the constraint of being disjoint with a set of entities C, $C \subset E$. We call this C the constraint set.

The high complexity of these exact algorithms leads us to propose heuristics to solve the TOPK_CAC problem by sampling the candidate collections that are potentially more anomalous. We propose to sample candidates from small size to larger sizes. This is because, (I)anomalies are minorities and anomaly collections are in general of small sizes; (II)collections of larger sizes although may have larger anomaly scores but are less likely to be coherent. Before showing the first heuristic regarding sampling candidates of small size to large, we first define **first-maximal CAC with constraint** C as follows.

Given E and C, let $(S^2, \ldots, S^{|E|/2-1})$ be the sequence of top-1 CACs of size from 2 to $|E|/2-1$, s.t. $S^i \cap C = \emptyset$, $\forall 2 \le i < |E|/2-1$, the **first-maximal CAC with constraint** C is the S^i, $2 \le i < |E|/2-1$ such that (I) $\Omega(S^j, F) \le \Omega(S^{j+1}, F)$, for all $1 < j < i$; (II) $\Omega(S^i, F) > \Omega(S^{i+1}, F)$.

Intuitively, first-maximal CAC with constraint C is the CAC that does not overlap with C and are more anomalous than all collections that are of smaller sizes. With this, our first heuristic is as follows.

HEURISTIC 1. ***[first-maximal property of Top-K CACs]*** *Let $\hat{S} = (S_1, S_2, \ldots, S_K)$ be the top-K CACs. $\forall i, 1 \le i \le K$, S_i is the first-maximal CAC with constraint $\bigcup_{1 \le j < i} S_j$.*

Algorithm 1 CACD_H, *Heuristically detecting top-K disjoint CACs*

Input: E, F, K
Output: heuristic top-K CACs: $\hat{S}$

```
1: Ŝ ← ∅; C ← ∅
2: repeat
3:     n ← 2
4:     S ← topCAC_size(E, F, n, C)
5:     while S ≠ null AND n < |E|/2 do
6:         S' ← topCAC_size(E, F, n + 1, C)
7:         if S' ≠ null AND Ω(S, F) < Ω(S', F) then
8:             S ← S'
9:         else
10:            break
11:    if S ≠ null then
12:        add S to Ŝ
13:        add all entities in S to C
14: until S == null OR |Ŝ| > K
15: return Ŝ
```

In step 4 and step 6 of Algorithm 1, we need to find the top-1 CAC of a given size n. However, the number of CACs of size n is $|E|^n$ in the worst case. We therefore propose the second heuristic regarding *the importance of local extremity.*

HEURISTIC 2. ***[Importance of Local Extremity]*** *Given E and F, let S be the top-1 CAC w.r.t. F. There exists a feature $f \in F$ and a small integer threshold θ such that S is among the top-θ CACs w.r.t. f.*

3.1 Ordering of P-values

Recall that given an entity set S and a significant feature f, the anomaly score of S w.r.t. f depends on the representative p-value of S w.r.t. f. If we can order the representative p-values of all possible collections of size n, and then find the set of collections corresponding to each of these p-values, we will be able to derive the collections with larger anomaly score w.r.t. f. In other words, the ordering of collections by anomaly score can be derived from the ordering of p-values.

As stated in Section 2, given any collection S of size n, the p-value of S is determined by the extremity index r and i which is the number of entities in S that appear in top-r positions. Hence any p-value can be represented as $p(i, r, n)$.

For all collections of size n, we could derive their representative p-values by enumerating all possible (i, r, n) combinations with the constraints $1 \le r < |E|/2$, $1 \le i \le min(r, n)$. However, the total number of (i, r, n) combinations is large. Furthermore, we are more interested in those with the small

p-values, as they indicate more anomalous collections. We therefore make use of the intrinsic partial orderings among $p(i,r,n)$ values for deriving the next smallest p-value without enumerating all p-values. We define the **p-value frontier** of $p(i,r,n)$ as the set of p-values that are the immediate smaller p-values according to each partial order. The partial orders lead to the following Lemmas for deriving the next smallest $p(i,r,n)$ value.

LEMMA 1. *The anchor of column-n, i.e., $p(n,n,n)$, is the smallest p-value of all.*

LEMMA 2. *Given any $p(i,r,n)$ value, the next smallest p-value lies in the p-value frontier of $p(i,r,n)$ or the frontiers of the p-values that are no greater than $p(i,r,n)$.*

3.2 Collection Sampling

Now the question is how to derive the set of collections whose representative p-value is of a given $p(i,r,n)$. For a given $p(i,r,n)$, multiple collections may have $p(i,r,n)$ as their representative p-value. In fact, the number of collections having the same representative p-value w.r.t. any feature can be as many as $\binom{r}{i}\binom{|E|-r}{n-i}$. As a result, given a representative p-value, each feature has many corresponding collections. We therefore want to sample only a subset of these collections that have larger anomaly score not only w.r.t. a single feature, but also w.r.t. the whole feature set F.

Our idea is to select individual entities that are more anomalous w.r.t. F and construct the collections from them. Naturally, an individual entity e is more anomalous if its **singular anomaly score**, i.e., $\Omega(\{e\},F)$ is larger. We hence take the heuristic that those collections whose elements have larger sum of singular anomaly scores have larger anomaly scores. In other words, we approximate $\Omega(S,F)$ by $\Sigma_{e\in S}\Omega(\{e\},F)$.

Next, we need a way of pinpointing individual entities in the entity list corresponding to each feature. Intuitively, we need n pointers, each points to an individual entity. We denote the list of rankings indicated by the n pointers as π. For a given feature f, π uniquely indicates one size-n collection. We denote $E^f(\pi)$ as the set of entities associated with π for f. Next, we form a collection from the first entity on each candidate entity list. We then find the collection that has the next largest sum of singular anomaly score.

Putting the ideas together, we have Algorithm 2 to heuristically compute the top-1 CAC of a given size n. It needs two additional parameters. The first parameter θ^c is used for selecting collections for a given p-value. Specifically, for $p(i,r,n)$, we repeatedly pop a collection across multiple features and evaluate whether it is larger than the current top-1 CAC. If the top-1 CAC remains unchanged after θ^c number of times, we stop the collection selection process for $p(i,r,n)$. As the algorithm searches progressively larger p-values from the small to large, the second parameter θ^p served as a ceiling on the number of p-values that have contributed no collections as the current top-1 CAC. The intuition is that if θ^p number of p-values have not contribute any collection for the top-1 CAC, the unseen p-values which are even smaller are unlikely to be able to contribute collections. θ^c and θ^p can be decided empirically. Larger θ^c and θ^p settings imply going through more candidate collections, which necessitates a longer execution time. The setting of θ^c and θ^p will be studied in the experiments section.

Algorithm 2 *Compute $topCAC\_size(E,F,n,C)$ in algorithm 1 by sampling anomalous collections of size n*

Input: E, F, collection size n, constraint set C, θ^c and θ^p
Output: the top-1 CAC of size n: S

```
1: Let pt ← (n, n, n) { pt keeps the current p-value (i, r, n) tuple}
2: Let Ψ ← frontier_p(pt) { frontier_p() computes the frontier of the p(i, r, n) indicated by pt}
3: y ← 0
4: repeat
5:     x ← 0
6:     for each f ∈ F do
7:         initialize S(f) as the collection having the largest sum of singular anomaly score for feature f and disjoint with C.
8:         Γ(f) ← frontier_i(S(f), C) {Γ(f) keeps the pointer frontier for feature f; frontier_i(S(f), C) returns the pointer frontier of collection S(f) and the collections are disjoint with C}
9:     while (x < θ^c) AND ∃f ∈ F s.t. S(f) ≠ ∅ do
10:        f ← argmax_{f∈F} Ω(S(f), F)
11:        if Ω(S(f), F) > Ω(S, F) AND S(f) is coherent then
12:            S ← S(f)
13:        else
14:            x++
15:        S(f) ← pop(Γ(f)) {pop() pops out the collection with the largest anomaly score in Γ(f)}
16:        Γ(f) ← Γ(f) ∪ frontier_i(S(f), C)
17:    if S is never updated for this pt then
18:        y++
19:    pt ← pop(Ψ) {pop() returns the (i, r, n) tuple with the smallest p-value in the frontiers Ψ}
20:    Ψ ← Ψ ∪ frontier_p(pt)
21: until y > θ^p
22: return S
```

4. EXPERIMENTS

In this section, we present experimental studies of our approach on a Twitter dataset. We showcase the detected coherent spammer groups and compare our approach against an existing co-clustering algorithm.

4.1 Twitter Data.

Data Setting.
Our Twitter data[2] is composed of all the tweets published between September 8th 2011 and November 15th 2011, containing any of the 9 hashtags related to Singapore including #sg, #singapore and #sosingaporean. Altogether, there are 231,803 tweets from 21,666 users. With a total of 11,901 hashtags, a hashtag is used by 4.58 users on average. Ranked by the number of users, the top 5 percent of these hashtags are considered popular. We remove the rest of the hashtags along with the ones that we used to collect the data. We also remove all the retweets, as hashtags in retweets do not indicate that the retweeting user is spamming on the hashtags. In addition, we filter away users who use only a hashtag only once, since they are unlikely to be spamming on hashtags. After the preprocessing, we are left with 1899 users with 587 popular hashtags, which is considered as independent features. For each hashtag, we rank all the users in descending order by usage frequency. If a user never mentions a particular hashtag, the corresponding feature value is zero. For each feature, rankings of users with identical feature values are randomized.

Our Results.
In this experiment, we empirically set θ^c=100 and θ^p=20 for our algorithm CACD_H. Larger parameter values such as

[2]http://research.larc.smu.edu.sg/palanteer/index_tracker.php

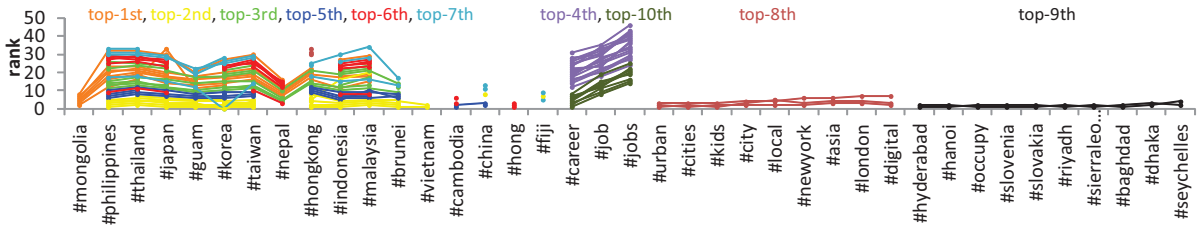

Figure 2: Top-10 CACs corresponding 63 users and 39 significant features (i.e., hashtags).

θ^c=500 and θ^p=50 have also been tried and give the same results with longer running time. We set K to a large value, so that CACD_H stops when it finds all disjoint CACs.

As a result, CACD_H produces 36 disjoint CACs. It is clear that members of a CAC collaborate in the same spamming campaign, as their tweets are often identical, with no real content other than a large number of hashtags appended with short URLs pointing to some website(s).

All members in the top-10 CACs are visualized in Figure 2 by parallel coordinates. Each member is represented by a line connecting the ranks of the user's usage of all the 39 significant features(i.e., hashtags). Members of the same CAC are given the same color. It is visually telling that all the 10 CACs are both extreme and coherent in their usage patterns of the hashtags. Moreover, our algorithm can identify subtle differences in the extreme behavior of CACs which seemingly belong to the same group. For example, the top-1st and top-2nd groups of "pornographic" spammers are in fact slightly different in their spamming patterns: (I) Besides the 8 hashtags in common, the top-1st group spams on #mongolia and #nepal while the top-2nd group spams on #brunei and #vietnam; (II) the top-2nd CAC, despite having fewer members, uses most of the hashtags more heavily than the top-1st CAC.

Comparison with Co-Clustering.
As our method can simultaneously detect anomaly collections and their corresponding significant features, one may suspect that similar results can be obtained by modeling the problem as a co-clustering task by clustering rows (features) and columns (users) of a matrix at the same time. We thus compare with the results of a co-clustering-based algorithm in [1]. This algorithm is chosen as it does not need the number of clusters as input and is denoted as Co-clustering.

To apply the co-clustering algorithm, we need to first derive the input matrix. The direct way of representing the input matrix on this Twitter data is to give a value of 1 to a cell if the corresponding user has used this hashtag, and give a value of 0 otherwise. Consequently, we have a 587 (number of features) by 1899 (number of users) matrix to feed into the co-clustering algorithm [1]. In the result, the matrix is co-clustered into 8 feature groups and 10 user groups, the largest user group being of size 467 and the smallest of size 2. Of all the 10 user groups, none of them are coherent and only 5 of them can be considered as anomaly collections. We manually go through the 5 anomaly collections and find that they are anomalous only because they contain subsets of members that are ranked at extreme positions on a small number of features. This is not surprising as co-clustering aims to group users using similar sets of hashtags, not necessarily those who *heavily* use these hashtags. While identified behavior are shared, they are not necessarily anomalous.

Even if we take only the union of users ranked in the top positions of each feature, the co-clustering algorithm would not output some extremely ranked collections as expected. We choose the top-31 positions of each feature so that the input matrix contains information of all users in our top-10 CACs. Yet, out of the 10 user collections identified by co-clustering, none of them are both anomalous and coherent. The most anomalous collection returned are of size 124, which contains some of the pornographic spammers, and many other users that are not even sharing the same significant features with the pornographic spammers. The poor performance of the co-clustering is due to its treating every feature the same when trying to simultaneously group users and features. In contrast, our approach is able to identify the significant features along with the anomalous users.

5. CONCLUSIONS

In this paper, we propose the problem of detecting top-K disjoint Coherent Anomaly Collections (CAC). We present an algorithm to identify CACs that does not need the collection number or collection size to be specified beforehand. Our algorithm is tested on a Twitter.com dataset to detect hashtag spammer collections. The experiment results demonstrate that our approach successfully finds suspicious spammer groups which are not easily identifiable with other approaches.

6. ACKNOWLEDGMENTS

This research is supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office.

References

[1] D. Chakrabarti, S. Papadimitriou, D. S. Modha, and C. Faloutsos. Fully automatic cross-associations. In *SIGKDD Conf.*, 2004.
[2] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3), 2009.
[3] H. Dai, F. Zhu, E.-P. Lim, and H. H. Pang. Detecting extreme rank anomalous collections. In *SDM Conf.*, 2012.
[4] K. Das, J. Schneider, and D. B. Neill. Anomaly pattern detection in categorical datasets. In *SIGKDD Conf.*, 2008.
[5] L. Duan, L. Xu, Y. Liu, and J. Lee. Cluster-based outlier detection. *Annals of Operations Research*, 168(1), 2009.
[6] M. Kendall. *Rank correlation methods.* Griffin, 1948.
[7] F. T. Liu, K. M. Ting, and Z.-H. Zhou. On detecting clustered anomalies using sciforest. In *ECML/PKDD Conf.*, 2010.
[8] A. Mukherjee, B. Liu, and N. Glance. Spotting fake reviewer groups in consumer reviews. In *WWW Conf.*, 2012.