A Spectral Framework for Detecting Inconsistency across Multi-Source Object Relationships

Jing Gao[†], Wei Fan[‡], Deepak Turaga[‡], Srinivasan Parthasarathy[‡], and Jiawei Han[†]
[†]University of Illinois at Urbana-Champaign, IL USA
[‡]IBM T. J. Watson Research Center, Hawthorn, NY USA

***Abstract*—In this paper, we propose to conduct anomaly detection across multiple sources to identify objects that have inconsistent behavior across these sources. We assume that a set of objects can be described from various perspectives (multiple information sources). The underlying clustering structure of normal objects is usually shared by multiple sources. However, anomalous objects belong to different clusters when considering different aspects. For example, there exist movies that are expected to be liked by kids by genre, but are liked by grown-ups based on user viewing history. To identify such objects, we propose to compute the distance between different eigen decomposition results of the same object with respect to different sources as its anomalous score. We also give interpretations from the perspectives of constrained spectral clustering and random walks over graph. Experimental results on several UCI as well as DBLP and MovieLens datasets demonstrate the effectiveness of the proposed approach.**

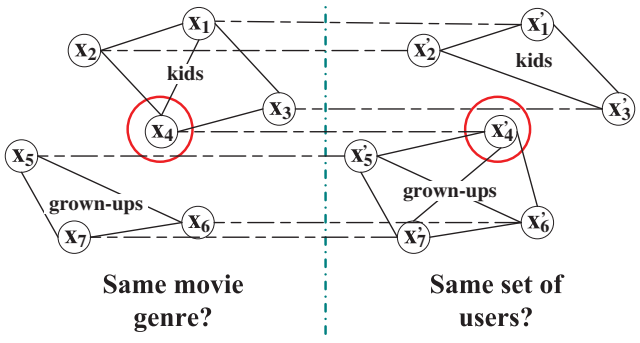
***Keywords*-anomaly detection; multiple information sources; spectral methods**

I. Introduction

Nowadays, there are usually several sources of information that describe different properties or characteristics of individual objects. For example, we can learn about a movie from its basic information including genre, cast, plots, etc., or the tags users give to the movie, or the viewing histories of the users who watched the movie. On each of the information source, a relationship graph can be derived to characterize the pairwise similarities between objects where the edge weight indicates the degree of similarity. As an example, Figure 1 shows the similarity relationships among a set of movies derived from two information sources: movie genres and users. The genre information may indicate that two movies that are both "animations" are more similar than two other movies where one is an "animation" and the other is a "romance" movie. Similarly, movies watched by the same set of users are likely to be more similar than movies that are watched by completely different sets of users.

Clearly, objects form a variety of clusters or communities based on individual similarity relationship. For example, two clusters can be found from both of the similarity graphs in Figure 1. One cluster represents the movies that are animations, which are loved by kids; while the other cluster represents romance movies, which are liked by grown-ups. Most of the movies belong to the same cluster even though different information sources are used. However, there are some objects that fall into different clusters with respect to different sources. In this example, the animated movie "Wall-E" by genre is expected to be liked by kids, but it is liked by grown-ups based on user viewing history. Finding such "inconsistent" movies can help film distributors better understand the expected audiences of different movies and make smarter marketing plans. In this paper, we propose to detect objects that have "inconsistent behavior" among multiple information sources, which we refer to as **horizontal anomaly detection**. Some other example scenarios of horizontal anomaly detection include detecting people who fall into different social communities with respect to different online social networks and detecting inconsistency across multiple module interaction graphs derived from different versions of a software project. Furthermore, identifying horizontal anomalies can find applications in many fields including smarter planet, internet of things, intelligent transportation systems, marketing, banking, etc.

To the best of our knowledge, this is the first work on identifying horizontal anomalies by exploring the inconsistencies among multiple sources. Traditional anomaly detection [4] approaches focus on identifying objects that are dissimilar to most of the other objects from a single source [9], [3], [12], [8]. On the other hand, most of the existing work on mining multiple information sources concerns merging and synthesizing models, rules, patterns obtained from multiple sources by reconciling their differences, such as multi-view learning [2], emerging or contrast patterns [5], multi-view clustering [1], [18] and consensus clustering [15], [6]. As for multi-source anomaly detection, the studies focus on how to identify anomalies within a specific context where the pre-defined contextual attributes include spatial attributes [13], neighborhoods in graphs [16], social communities [7], and contextual attributes [17], [14]. Although these studies take two types of attributes (behavioral and contextual [4]) into consideration, they cannot be easily generalized to horizontal anomaly detection spanning multiple sources. The reason is that they simply detect anomalies from the behavioral attributes while the contextual attributes only provide the context in which the anomalies are detected. In some sense, these contextual anomalies are still extracted from one source, whereas the proposed method can identify



X1-The Lion King; X2-Toy Story; X3-Kungfu Panda; X4-Wall-E; X5-Casablanca; X6-Titanic; X7-The Notebook

Figure 1: A Horizontal Anomaly Detection Example

objects with inconsistent behavior across multiple sources.

In this paper, we propose a systematic approach to identify horizontal anomalies from multiple information sources. We assume that each individual information source captures some similarity relationships between objects that may be represented in the form of a similarity matrix (whose entries represent the pairwise quantitative similarity between objects). We combine the input matrices into one large similarity matrix and adopt spectral techniques to identify the key eigenvectors of the graph Laplacian of the combined matrix. Horizontal anomalies are identified by computing cosine distance between the components of these eigenvectors. The method can be regarded as conducting spectral clustering on multiple information sources simultaneously with a joint constraint that the underlying clustering structures are similar, and objects that are clustered differently are categorized as horizontal anomalies. The horizontal anomalies can also be regarded as those having long commute time in the random walk defined over the graph. We validate the proposed algorithm on both synthetic and real data sets, and the results demonstrate the advantages of the proposed approach in finding horizontal anomalies.

II. Methodology

Suppose we have a set of N objects $X = \{x_1, x_2, \ldots, x_N\}$ and there are P information sources that describe different aspects of these objects. The objective is to assign an anomalous score s_i to each object x_i, which represents how likely the object is anomalous when its behavior differs among the P different information sources. In this section, we present a **HO**rizontal **A**nomaly **D**etection (**HOAD**) algorithm to solve the proposed problem. An object can be regarded as a horizontal anomaly if it is assigned to different clusters when using various information sources, and thus we calculate the anomalous degree of an object based on how much its clustering solutions differ from each other. To simplify the notations, we start with the cases having two distinct information sources (Section II-A), give spectral clustering and random walk interpretations in Section II-B, and explain how it is generalized to multiple information sources in Section II-C.

A. HOAD Algorithm

Suppose we have two $N \times N$ similarity matrices on the N objects: A and W, where a_{ij} and w_{ij} define the similarity between x_i and x_j from different aspects. The algorithm consists of two major steps: First, we conduct soft clustering on A and W together with the constraint that an object should be assigned to the same cluster; Second, we quantify the difference between the two clustering solutions to derive anomalous scores. The details are as follows. We construct a combined graph by connecting the nodes which correspond to the same object in the two similarity graphs with an edge weighted m. m, a large positive number, is a penalty parameter. An example of such a graph is illustrated in Figure 1. The set of nodes in the combined graph consists of two copies of the objects: $\{x_1, \ldots, x_N, x'_1, \ldots, x'_N\}$ ($2N$ nodes in total). Let M be an $N \times N$ diagonal matrix with m on the diagonal: $M = m \cdot I$ where I is an $N \times N$ identify matrix. Let Z be the adjacency matrix of the combined graph, which is a $2N \times 2N$ matrix:

$$Z = \begin{bmatrix} A & M \\ M & W \end{bmatrix}. \tag{1}$$

First, we compute the graph Laplacian L as:

$$L = D - Z \tag{2}$$

using degree matrix D (a $2N \times 2N$ diagonal matrix):

$$D = \text{diag}\big(\{\sum_{j=1}^{2N} z_{ij}\}_{i=1}^{2N}\big). \tag{3}$$

Secondly, compute the k smallest eigenvectors of L (with smallest eigenvalues) and let $H \in \mathbb{R}^{2N \times k}$ be the matrix containing these eigenvectors as columns. We divide H into two submatrices U and V each with size $N \times k$ so that $H = [U \quad V]^T$. Therefore, the i-th and $(i+N)$-th rows of H are represented as:

$$\vec{u}_i = \vec{h}_i, \quad \vec{v}_i = \vec{h}_{i+N}, \tag{4}$$

which correspond to two "soft clustering" representations of x_i with respect to A and W respectively. As can be seen, with the help of the edge between the copies of the same object, we try to cluster the objects in the same way across different sources. In Section II-B, we give a theoretical justification of this claim. Finally, compute the anomalous score for object x_i using cosine distance between the two vectors:

$$s_i = 1 - \frac{\vec{u}_i \cdot \vec{v}_i}{||\vec{u}_i|| \cdot ||\vec{v}_i||}. \tag{5}$$

The algorithm flow is summarized in Algorithm 1.

Algorithm 1 HOAD algorithm

Input: similarity matrices A and W, number of eigenvectors k, penalty parameter m;
Output: anomalous score vector $\vec{s}$;
Algorithm:

1. Compute matrix Z according to Eq. (1)
2. Compute graph Laplacian L as in Eq. (2)
3. Conduct eigen-decomposition of L and Let H be the k smallest eigenvectors with smallest eigenvalues
4. Compute anomalous score of each object s_i based on Eq. (4) and Eq. (5) for $i = 1, \ldots, N$

return $\vec{s}$

B. Interpretations

In this part, we explain the algorithm from the perspectives of spectral clustering and random walk.

Clustering on Combined Graph. As can be seen, we first perform spectral clustering on the combined graph in Algorithm 1. The basic idea of spectral clustering is to project the objects into a low-dimensional space (defined by the k smallest eigenvectors of the graph Laplacian matrix) so that the objects in the new space can be easily separated. We call the projections as *spectral embeddings* of the objects. It has been shown that the matrix formed by the k eigen vectors (H) of L is the solution to the following optimization problem [11]:

$$min_{H \in \mathbb{R}^{N \times k}} \quad \text{Tr}(H'LH) \quad \text{s.t.} \quad H'H = I \tag{6}$$

H is a $2N \times k$ matrix, which equals to $[U \quad V]^T$. The graph Laplacian L is defined as $D - Z$ (Eq. (2)), and Z is defined in Eq. (1). Moreover, we suppose the degree matrices for A and W are D^a and D^w respectively:

$$D^a = \text{diag}\big(\{\sum_{j=1}^{N} a_{ij}\}_{i=1}^{N}\big), \quad D^w = \text{diag}\big(\{\sum_{j=1}^{N} w_{ij}\}_{i=1}^{N}\big).$$

Then we can derive an equivalent formulation for the problem in Eq. (6):

$$\begin{aligned} &min_{U,V} \quad \text{Tr}(U'(D^a - A)U) + \text{Tr}(V'(D^w - W)V) \\ &-2m \sum_{i=1}^{n} \sum_{j=1}^{k} u_{ij} v_{ij} \quad \text{s.t.} \quad U'U + V'V = I \end{aligned} \tag{7}$$

The proof is omitted due to space limit. Clearly, each of the first two terms in Eq. (7) corresponds to the spectral clustering problem using A or W alone. The third term acts as the constraint that the two clustering solutions should be similar (cosine similarity). Therefore, the first three lines in Algorithm 1 can be interpreted as conducting spectral clustering on the two input similarity graphs simultaneously with a joint constraint.

Our goal is to detect horizontal anomalies that have inconsistent behavior across sources, and thus the final step is to compute anomalous scores. Note that in Algorithm 1, the i-th row vector in U (the first N rows of H) and V (the last N rows of H) contain the projections of object x_i. Due to the principle of spectral clustering, if the spectral embeddings $\vec{u}_i$ and $\vec{v}_i$ are close to each other, the corresponding object x_i is more likely to be assigned to the same cluster with respect to two different sources. Therefore, the cosine similarity between the two vectors $\vec{u}_i$ and $\vec{v}_i$ quantifies how similar the clustering results of object x_i on the two sources are, and thus represents its "normal" degree. In turn, the cosine distance as defined in Eq. (5) gives the "anomalous" degree of x_i with respect to the two sources. The higher the score s_i is, the more likely x_i is a horizontal anomaly.

Random Walk. Suppose we define a random walk over the combined graph, where the transition probability from node x_i to node x_j is proportional to the edge weight in the graph. Let z_{ij} be the edge weight between two nodes x_i and x_j in the graph, and $\text{vol}(X) = \sum_{i=1}^{2N} \sum_{j=1}^{2N} z_{ij}$ be the sum of all the edge weights in the graph. Now we look at the commute distance between x_i and x_i', two copies of the same object in the combined graph. Commute distance is the expected time it takes for the random walk to travel from x_i to x_i' and back, and it can be computed using the eigenvectors of the graph Laplacian L as defined in Eq. (2). Suppose L has eigenvalues $\lambda_1, \ldots, \lambda_{2N}$, and U and V are two $N \times N$ matrices containing all the eigenvectors for the two copies of the objects respectively. Let $\vec{u}_i$ and $\vec{v}_i$ denote the i-th row of U and V. We define $\vec{\gamma}$ as a length-$2N$ vector with each entry γ_l equal to $(\lambda_l)^{-0.5}$ if $\lambda_l \neq 0$, and 0 otherwise. Now we divide $\vec{\gamma}$ into two length-N vectors: $\vec{\gamma} = [\vec{\gamma}_u \quad \vec{\gamma}_v]$. It can be derived that the commute distance c_i between x_i and x_i' is: $c_i = \text{vol}(X)||\vec{u}_i \cdot \vec{\gamma}_u - \vec{v}_i \cdot \vec{\gamma}_v||^2$.

Recall that we compute the anomalous score of x_i as $1 - \frac{\vec{u}_i \cdot \vec{v}_i}{||\vec{u}_i|| \cdot ||\vec{v}_i||}$. We can see that both the anomalous score and the commute distance can be represented as *a distance function applied on the spectral embeddings of the two copies of the object*. The difference is that all the eigenvectors are used and they are scaled by $(\lambda_l)^{-0.5}$ in the commute distance computation. Also, Euclidean distance is used instead of cosine distance. Although the connection is loose, commute distance can be a helpful intuition to understand the anomalous scores. If it takes longer time to commute between the two copies of object x_i in the graph, x_i is more likely to be a horizontal anomaly.

C. Multiple Sources

We can adapt Algorithm 1 to handle more than two information sources as follows. Suppose we have similarity matrices $\{W^{(1)}, W^{(2)}, \ldots, W^{(P)}\}$ as the input. First, the combined graph is constructed in a similar fashion as discussed before: Duplicate the objects for P copies, in each copy retain the similarity information from each source, and connect each pair of the nodes corresponding to the same object with an edge weighted m. After that, we calculate its graph Laplacian and the k smallest eigenvectors following exactly the same procedure as in Algorithm 1. One concern

is that, when the number of information sources increases, the size of the matrix L grows quadratically. Note that the graph Laplacian of Z is a sparse matrix, and also, we only need the k smallest eigenvectors instead of the full eigenspace. In fact, efficient packages such as ARPACK [10], have been developed to compute a few eigenvectors of large-scale sparse matrix. Then we calculate the anomalous degree of an object x_i based on the following P vectors: $\{\vec{h}_i, \vec{h}_{i+N}, \vec{h}_{i+2N}, \ldots, \vec{h}_{i+(P-1)N}\}$. In the experiment, we use the average pairwise distance as the measure:

$$s_i = \frac{1}{P(P-1)} \sum_{a=0}^{P-1} \sum_{b=0}^{P-1} \mathbb{1}_{a \neq b} \cdot \left[1 - \frac{\vec{h}_{i+aN} \cdot \vec{h}_{i+bN}}{||\vec{h}_{i+aN}|| \cdot ||\vec{h}_{i+bN}||} \right]$$

III. Experiments

We evaluate the HOAD algorithm on synthetic data and real datasets including DBLP and MovieLens to validate its ability of identifying meaningful horizontal anomalies.

A. Synthetic Data

The concept of "horizontal anomaly" is new, and thus there are no benchmark datasets for it. Therefore, we propose a method to convert a classification problem into a horizontal anomaly detection problem, and then apply this procedure on several UCI machine learning data sets.

Data Generation. Suppose we have a training set from a classification problem where each object consists of feature values and a class label. We simulate inconsistency across multiple sources by swapping feature values of objects from different classes. Suppose there are N objects in the training set: $\{x_1, \ldots, x_N\}$, and the features X can be partitioned into two views. We assume that objects within the same class share similar feature values in each feature set. Therefore, for two objects x_i and x_j from different classes, if their feature values are swapped in one view but remain unchanged in the other, they have "inconsistent" behavior among these two views, and thus represent horizontal anomalies. We apply the above method on four data sets obtained from UCI machine learning repository[1]: Zoo, Iris, Letter and Waveform. On each data set, we repeat the transformation procedure 50 times and at each time, we generate a data set with around 10% anomalies. We evaluate the HOAD algorithm on the 50 data sets and report the average accuracy.

Evaluation Measure and Baseline Methods. For anomaly detection problems, one of the most widely used evaluation approaches is ROC analysis, which represents the trade-off between detection rate and false alarm rate. The area under ROC curve (AUC), which is in the range [0,1], is a good evaluation metric. The higher the AUC is, the better the algorithm performs. We show the performance of the proposed HOAD algorithm with various parameter settings. Note that the first step of the proposed algorithm is a constrained soft clustering on multiple information sources. Instead of conducting a joint clustering, the baseline method clusters multiple sources *separately* and calculates the anomalous scores based on the difference among different clustering solutions. Specifically, in two-source problems, we conduct eigen decomposition on the graph Laplacian matrices of the two similarity matrices A and W separately. Suppose the top k eigen representation of object x_i are u_i and v_i respectively, then we use Eq. (5) to compute the anomalous score of x_i for the baseline approach. Note that the major difference between the HOAD algorithm and the baseline method is on how to compute u_i and v_i.

Performance. The experimental results on the four data sets are shown in Figure 2 where we vary the values of the parameters m and k. m indicates the penalty we enforce when the two clustering solutions do not agree, and k represents the number of top eigenvectors. Neither m nor k is used in the baseline and its performance remains mostly stable except slight fluctuation due to random sampling in data generation. From the experimental results, we can see that HOAD algorithm generally outperforms the baseline, especially when k is small (e.g., $k = 3$). However, when the value of m is higher, the difference in AUC between the algorithms using different k is much smaller. Therefore, we focus on how to select the appropriate m in the following discussion. On UCI datasets, it is clear that when m increases, the proposed algorithm has a higher AUC. In the simulated study, the two feature sets are two disjoint subsets of the original features, and usually using all of the features leads to a better classification model. Hence the two views are correlated and using a large m captures this correlation well. However, this does not mean that we should assign a big number to m in all cases because this pattern may not always hold in real horizontal anomaly detection tasks. In the following experiments on DBLP data sets, we illustrate the relationship between m and the anomalous scores, and state some principles in setting m.

In Figure 4, we show the running time of HOAD algorithm with respect to 1000 to 6000 objects represented in two, three or four information sources. We conduct the experiments on synthetic data sets where we randomly generate similarity matrices for 50 trials, and report the average running time. The eigenvectors are computed using Matlab eigs function, which is based on ARPACK package [10]. As can be seen, the HOAD algorithm can scale well to large data sets when the number of objects and number of sources both increase.

B. Real World Data

We discuss the issues of setting parameters on DBLP data and present illustrative results on MovieLens data.

DBLP. We define two horizontal anomaly detection tasks based on the DBLP[2] data where the objects are a set of

[1] http://archive.ics.uci.edu/ml

[2] http://www.informatik.uni-trier.de/~ley/db/

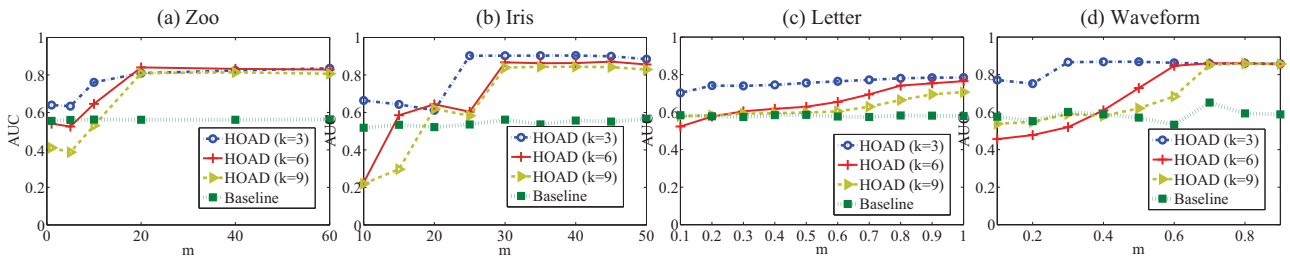


Figure 2: Anomaly Detection Performance Comparison on Simulated Detection Tasks based on UCI Data

conferences and authors respectively. 4220 conferences are represented in two views: the keywords in the conferences and the authors who published in the conferences. Specifically, each conference x_i has two vectors. In the first vector, each entry is the number of times each word appeared in the paper titles of x_i. In the second vector, each entry denotes the number of times an author published in x_i. The pairwise similarity between two conferences x_i and x_j is defined as the cosine similarity between the corresponding vectors. Therefore, the conferences that share lots of keywords, or share lots of authors are similar. Similarly, we select a set of 3116 authors from data mining related areas and extract two types of information from DBLP: the publications and the co-authorships. Each author x_i also has two vectors where in the first vector each entry denotes the occurrence of each word in the authors' publications, and each entry corresponds to the number of times two authors collaborate in the second one. Cosine similarity is used, and similar authors will share co-authors, or keywords in their publications.

We study the effect of m on the anomalous scores. For each m, we apply the HOAD algorithm to the data sets, and compute the mean and standard deviation of the objects' anomalous scores. The results on conferences and authors are shown in Figure 3 where the points on the line are the average anomalous scores and the error bar denotes the standard deviation. Obviously, the average anomalous score decreases as m increases. Recall that the anomalous scores indicate the degree of differences between the spectral embeddings derived from the two similarity matrices. When we give a heavy penalty on different embeddings by the two sources, we basically bias the two projections towards the ones that agree the most. Therefore, when m is larger, the spectral embeddings from the two sources are more likely to be the same, and thus the difference between them is smaller. Another observation is that the variance among the anomalous scores goes up first and then goes down as m increases. When m is quite large or quite small, the two projections of all the objects would be very similar or very different, and thus the objects receive similar anomalous scores. There exists a large variability among the anomalous scores only when m is in the middle of the spectrum. Although m can be drawn from $(0, \infty)$, the average anomalous scores are within a fixed range: [0,1]. Therefore, we can choose m which leads to an average

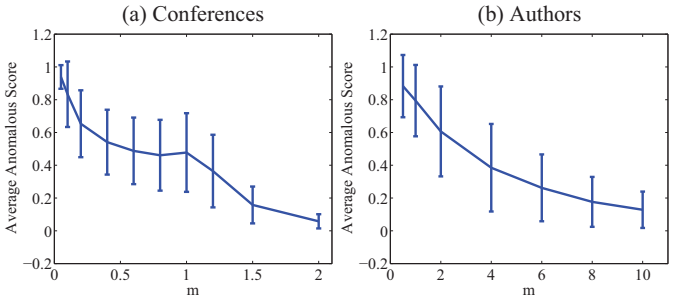


Figure 3: Analysis of Parameter m on DBLP Data

anomalous score around 0.5 because the variance of the anomalous scores usually reaches the highest point here and this helps us identify the horizontal anomalies.

MovieLens. We use the Movielens dataset[3] with movies as objects. There are three sources of information to capture their relationships: 1) Genre: Movies are classified as being of one or more of 18 genres, such as Comedy and Thriller, which can be treated as binary vectors. 2) User viewing history: Each movie has a list of users that watched the movie. This may also be represented as a vector (per movie) across all users. 3) User tagging history: Movies are tagged by different users. Looking across all users, we can determine a vector per movie. In all three cases, we compute the pairwise similarity using cosine similarity across the vectors. The data set contains 10 million ratings and 100,000 tags for 10681 movies by 71567 users. We choose a set of 7595 movies, each of which has both ratings and tags. We then have three similarity matrices, corresponding to these three different sources. To evaluate the performance of the HOAD algorithm on MovieLens dataset, we label some movies as "horizontal anomalies" based on the list of weirdest movies[4]. There are 572 movies listed as weirdest movies and among them 127 are found in the MovieLens dataset. These 127 movies are labeled as "anomalous" and the remaining 7468 movies are "normal". Based on these labels, we calculate the area under ROC curve (AUC) of both HOAD and the baseline method based on their computed anomalies scores for the 7595 movies. HOAD algorithm achieves a better AUC (0.4879) compared with that of the baseline method (0.4423). This demonstrates the ability

[3]http://www.grouplens.org/node/73

[4]http://366weirdmovies.com/the-weird-movie-list

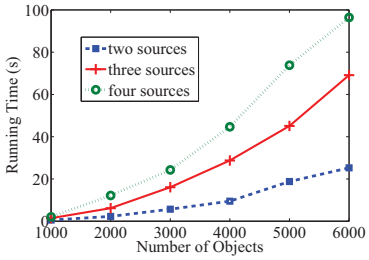


Figure 4: Running Time

Table I: Anomalous Scores of 20 Popular Movies from MovieLens

Movie	Score	Movie	Score
One Flew Over the Cuckoo's Nest	0.8079	Seven Samurai	0.6404
Pulp Fiction	0.7713	Fight Club	0.6364
Casablanca	0.7205	City of God	0.6278
The Shawshank Redemption	0.6949	The Lord of the Rings: The Return of the King	0.3512
The Godfather: Part II	0.6822	The Lord of the Rings: The Fellowship of the Ring	0.3478
The Godfather	0.6770	The Good, the Bad and the Ugly	0.3194
Goodfellas	0.6768	Raiders of the Lost Ark	0.3181
Schindler's List	0.6755	Rear Window	0.3095
12 Angry Men	0.6713	Star Wars	0.2982
The Dark Knight	0.6535	Star Wars: Episode V-The Empire Strikes Back	0.2562

of the proposed HOAD algorithm in detecting inconsistent movies across various information sources.

Moreover, we present the anomalous scores for the 20 most popular movies[5] as shown in Table I. As may be seen, the movies "One Flew Over the Cuckoo's Nest" and "Pulp Fiction" are identified as horizontal anomalies, as they tend to show strong disagreement between the genre classification, and the sets of users that watched and tagged them. Intuitively, this is expected as these two movies do not really fit into one classification or user category. Borrowing reviews from Wikipedia[6], "Pulp Fiction" is known for its rich, eclectic dialogue, ironic mix of humor and violence, and nonlinear storyline, which make it different and anomalous among movies. For "One Flew Over the Cuckoo's Nest", the review says "it is a comedy that can't quite support its tragic conclusion". These tell us the reasons why these two movies are detected as being inconsistent. On the other hand, "Star Wars" receives the lowest anomalous score as it attracts a particular set of audiences.

IV. Conclusions

We propose to detect horizontal anomalies, or objects that have inconsistent behavior among multiple sources. Intuitively, they belong to different clusters when considering many aspects from multiple information sources. The proposed algorithm has two intrinsic steps. In the first step, we construct a combined similarity graph based on the similarity matrices and compute the k smallest eigenvectors of the graph Laplacian as spectral embeddings of the objects. After that, we calculate the anomalous score of each object as the cosine distance between different spectral embeddings. The physical meaning of the proposed algorithm is explained from both constrained spectral clustering and random walk point of view. Experimental results show that the proposed algorithm can find horizontal anomalies from real-world datasets, where other anomaly detection methods fail to identify these anomalies.

References

[1] S. Bickel and T. Scheffer. Multi-view clustering. In *Proc. of ICDM'04*, pages 19–26, 2004.

[2] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proc. of COLT'98*, pages 92–100, 1998.

[3] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. Lof: identifying density-based local outliers. In *Proc. of SIGMOD'00*, pages 93–104, 2000.

[4] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3), 2009.

[5] G. Dong and J. Li. Efficient Mining of Emerging Patterns: Discovering Trends and Differences. In *Proc. of KDD'99*, pages 43–52, 1999.

[6] X. Z. Fern and C. E. Brodley. Solving cluster ensemble problems by bipartite graph partitioning. In *Proc. of ICML '04*, pages 281–288, 2004.

[7] J. Gao, F. Liang, W. Fan, C. Wang, Y. Sun, and J. Han. On community outliers and their efficient detection in information networks. In *Proc. of KDD'10*, pages 813–822, 2010.

[8] N. Khoa and S. Chawla. Robust outlier detection using commute time and eigenspace embedding. In *Proc. of PAKDD'10*, pages 422–434, 2010.

[9] E. M. Knorr, R. T. Ng, and V. Tucakov. Distance-based outliers: algorithms and applications. *The VLDB Journal*, 8(3-4):237–253, 2000.

[10] R. Lehoucq, D. Sorensen, and C. Yang. ARPACK users' guide: Solution of large-scale eigenvalue problems with implicitly restarted arnoldi methods. *SIAM Publications*, 1998.

[11] U. Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[12] M. Markou and S. Singh. Novelty detection: a review-part 1: statistical approaches. *Signal Processing*, 83(12):2481–2497, 2003.

[13] S. Shekhar, C.-T. Lu, and P. Zhang. Detecting graph-based spatial outliers: algorithms and applications (a summary of results). In *Proc. of KDD'01*, pages 371–376, 2001.

[14] X. Song, M. Wu, C. Jermaine, and S. Ranka. Conditional anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, 19(5):631–645, 2007.

[15] A. Strehl and J. Ghosh. Cluster ensembles –a knowledge reuse framework for combining multiple partitions. In *Journal of Machine Learning Research*, 3: 583-617, 2003.

[16] J. Sun, H. Qu, D. Chakrabarti, and C. Faloutsos. Neighborhood formation and anomaly detection in bipartite graphs. In *Prof. of ICDM'05*, pages 418–425, 2005.

[17] X. Wang and I. Davidson. Discovering contexts and contextual outliers using random walks in graphs. In *Proc. of ICDM'09*, pages 1034–1039, 2009.

[18] D. Zhou and C. Burges. Spectral clustering and transductive learning with multiple views. In *Proc. of ICML'07*, pages 1159–1166, 2007.

[5] As listed on http://www.imdb.com/chart/top on November 2010.

[6] http://en.wikipedia.org