Distributed Outlier detection

Submitted for Blind Review

Abstract

Detecting outliers from distributed data has been a challenge since straightforward distributed implementation of popular centralized outlier detection algorithms suffer either from low accuracy or high communication cost resulting from the fact that the problem is not decomposable in nature, i.e., the global outliers may be different from the local outliers. This paper offers a generic outlier detection technique that is appropriate for large asynchronous peer-to-peer (P2P) networks. It first presents a brief overview of a P2P distributed data mining system called PADMini that can be used for P2P classifier learning among others. The proposed distributed outlier detection technique is then presented in the context of this system and the related applications. The paper offers a set of analytical results regarding the properties of local and global outliers. The paper also presents a detailed description of the distributed algorithm and extensive experimental results.

Keywords Outlier, LOCI, Eigen Decomposition, Entropy, P2P

1 Introduction

Distributed outlier detection from data stored at different locations is an important problem with many applications. Large peer-to-peer networks are creating many such applications where centralized data collection and subsequent outlier detection are extremely difficult, if not impossible. There are a bunch of outlier detection techniques that are popular and widely used for many applications. These include statistical, distance / density based techniques. Distributed versions of outlier detection algorithms are therefore of high interest. However, just combining the local outliers in order to obtain the global set of outliers may not work always and hence the straightforward distributed extension of the centralized counterpart may suffer from high communication cost, since in the worst case all the data may be needed to be exchanged between the sites.This causes high communication overhead and synchronous algorithms that do not scale very well in large distributed environments like the P2P networks.

This paper motivates the current work by introducing the readers with the PADMini system that offers distributed outlier detection among other capabilities over P2P networks. The paper poses the distributed outlier detection problem in that context. This paper first explores a class of outlier detection algorithms introduced elsewhere [4], [6], [8] and then proposes a generalized technique to provide an approximate solution of the distributed outlier detection problem, by investigating analytically the decomposable properties of the global outlier detection techniques, and adapts those to design a distributed asynchronous outlier detection algorithm for P2P networks, avoiding too much communication overhead.

The paper is organized as follows. Section 2 describes the PADMini system in order to motivate the current work. Section 3 reviews the existing literature. Section 4 defines the distributed outlier detection problem. Section 5 explores various analytical aspects of the outlierness property as well as different approaches. Section 6 offers the experimental results. Finally, section 7 concludes this paper.

2 Motivation

Before going to further details, let's first try to find the answer to a very basic question: why it is that important (if at all) to detect outliers in data? What if we leave them undetected? The answer to this question can probably be best answered when it is related to the domain of data. In general, there can be primarily couple of motivations for detecting outliers in data. First one is that the outliers themselves can represent interesting pattern in data that may lead us to answer some very important cause-effect relations in real world (e.g., sudden increase in traffic data at a point of time may signify an accident, while sudden increase in bank transaction data may be a candidate of a fraud) and in data mining our primary interest is to find hidden nontrivial useful patterns inside data (in a resource and computation efficient manner). Secondly, many of the existing data mining algorithms see outliers as noise points in data and the performance of them degrade due to presence of these outliers, thus require removal of outliers in preprocessing steps (e.g., in case of k-means clustering, if one of the outlier points is selected as centroid of a cluster). Hence, it's important to separate outlier points from the non-outlier points in the data set.

Distributed outlier detection from data stored at different locations is an important problem with many applications. Large peer-to-peer networks are creating many such applications where centralized data collection and subsequent outlier detection are extremely difficult, if not impossible. Also, in order to obtain the global set of outliers, the straightforward approach of just finding the set of local outliers and then combining them may not work always. Hence, the distributed extension is non-trivial. In this paper we propose a generalized framework for the distributed extension for this problem.

3 Related Work

Outlier detection is one of the major problems in data mining and has attracted attention since a long time. There can be many definitions of the term outlier. In one way it can be defined as 'extreme data points' (noise), but in some other way it can be defined as (usually a small) set of data points behaving anomalously w.r.t. the other data points. Off course, one needs to quantify the term 'anomalous behavior' and it is usually defined by introducing notions of metrics: distance, density etc. and accordingly we have distance / density based methods for outlier detection. Also, we need to consider the neighborhood in which the point can be considered to be an outlier, interestingly the same point may not be considered as an abnormally behaving point if present in some other neighborhood. That brings the notion of local and global outliers. That brings the famous definition of distance-based outlier: if $p\%$ of the points are greater than d distance away from a given point, then that particular point can be treated as an outlier, with values of the given parameters p and d [6]. There are statistical techniques that assume some distribution (e.g., Gaussian) and then use some distance metrics like Mahalanobis to find the data points falling outside the 3σ limit from the mean. But it is not always the case that the extreme (or noise) points are the only candidates for being selected as outlier data points, since the outlier points may be scattered throughout the entire data space or they can together form a small set of outlier clusters as well. The outlier detection problem was also considered as neighborhood correlation problem [8] and in terms of local correlation integral (LOCI) . Since the basic goal is to divide the entire dataset into two parts, one being outliers, while the other being non-outlier data tuples, the problem can be thought in terms of flow-CUT problem in graph theory as well [7] [4]. There are already some existing literatures on the distributed extension of this problem. In [5] the distributed version is solved by using eigen analysis via gossip, where in [2] the distributed problem is addressed using majority voting that may demand potential communication overhead.

4 Definition

Consider the dataset $X_{m\times n}$ horizontally partitioned to N nodes, s.t., $X = \bigcup_{i=1}^{N} X_i$, with datset X_i $(m_i \times n)$ at node i, $\forall i \in 1 \ldots N$, s.t., $\sum_{i=1}^{N} m_i = m$.

The distributed (global) outlier detection will consist of the following steps:

- Define outlierness property by means of some matrix M (e.g., covariance or information matrix) and find outliers by eigen decomposition of the matrix.
- We do not want to compute the matrix M and global eigenvectors set V globally (since it will need too much communication). Instead we do the following steps.
- $\forall i = 1 \ldots N$, compute matrix M_i locally at each node i.
- Perform eigen decomposition locally, e.g., $M_i = V_i \wedge_i V_i^T$ at each node i.
- Combine the local set of eigenvectors $V_i(n \times n)$ to obtain an approximation for the global set of eigenvectors $V_{n\times n}$ as $\hat{V}_{n\times n}$ in any of the following ways:
  - $\hat{V} = \sum_{i=1}^{N} w_i.V_i$, where $w_i \in \Re$.
    How to find the weights (w_i)? since the centralized version does the eigen decomposition $M = V \wedge V^T$, we shall be interested in the vector w^* that minimizes $||V - \hat{V}||$, i.e., $w^* = \underset{w}{argmin}||V - \hat{V}||$.
  - $V_c = [V_1 V_2 \ldots V_N]$, where $V_c(Nn \times n)$ is the combined matrix. Next, do SVD on $V_c = \hat{V}\Sigma\hat{U}^T$, to find the approximate global eigenvectors $\hat{V}$.
- Find top k approximate global eigenvectors set $U_{n\times k}$ as $U = \hat{V}(:, 1 : k)$.
- Project each of the local datasets accross U as $\hat{X}_l = X_l.U.U^T$, $\forall l \in 1 \ldots N$.
- Find the outliers scores matrix locally for each node l as

$$dist(X_l, \hat{X}_l) = \left[\sum_j (x_{ij}^l - \hat{x}_{ij}^l)\right].$$

5 The Outlierness Property

The outlierness property is to be defined by the matrix M (e.g., covariance or information matrix) in the following manner.

5.1 Eigen Analysis of The Matrix M

It is easy to see that M can be defined as covariance matrix C. Given $m \times n$ dataset X, the outlier detection problem can be solved by the solving following (PCA) optimization problem:

$$y = \underset{ww^T=I}{argmax}\ var(w^T X)$$

Alternatively, M might be defined by the notion of chaosness (uncertainty) in the data that can be measured by entropy. Given $m \times n$ dataset X, the outlier detection problem can be solved by solving the following (ECA) optimization problem:

$$y = \underset{ww^T=I}{argmax}\ H(w^T X)$$

which physically means that we want to apply a linear transformation on the (data) vector space to obtain another new vector space where the entropy is maximized along the basis vectors.
We compute the $n \times n$ mutual information matrix I (symmetric and positive semidefinite) defined by

$$I_{ij} = I(X_i; X_j) = H(X_i) + H(X_j) - H(X_i, X_j)$$

where X_i and X_j denotes any two attributes of the dat set.
We do the eigen decomposition of the matrix I and find the most dominant eigenvector corresponding to the largest eignevalue, which ensures highest entropy, followed by projection along the highest entropy eigen vector in the new space denotes the most chaotic direction in the data.

5.2 Proofs

Let's assume $M = f(M_1, M_2, \ldots M_N)$, where $f : \boldsymbol{M}^N \rightarrow \boldsymbol{M}$ is a function that combines the local matrices into a global matrix. (when $M = C$, i.e., for the covariance matrix, f is linear, because of additive decomposibility of the covariance matrix, since $C = \sum_{i=1}^{N} w_i.C_i$, with $w_i = \frac{m_i}{\sum_{j=1}^{N} m_j}$. Using eigen decomposition, we get the following:

$$\begin{aligned} M &= f(M_1, M_2, \ldots M_N) \\ \Rightarrow V \wedge V^T &= f(V_1 \wedge_1 V_1^T, \ldots, V_N \wedge_N V_N^T) \\ \Rightarrow \sum_{i=1}^{n} \lambda_i \bar{v}_i \bar{v}_i^T &= f\left(\sum_{i=1}^{n} \lambda_i^1 \bar{v_i^1} \bar{v_i^1}^T, \ldots, \sum_{i=1}^{n} \lambda_i^N \bar{v_i^N} \bar{v_i^N}^T\right) \end{aligned}$$

With $M = C$,

$$\begin{aligned} C &= \sum_{i=1}^{N} w_i.C_i \\ \Rightarrow V \wedge V^T &= \sum_{i=1}^{N} w_i.V_i \wedge_i V_i^T \\ \Rightarrow \sum_{i=1}^{n} \lambda_i \bar{v}_i \bar{v}_i^T &= \sum_{k=1}^{N} w_k. \left(\sum_{i=1}^{n} \lambda_i^k \bar{v_i^k} \bar{v_i^k}^T\right) \end{aligned}$$

5.3 Entropy Minimization Approach

By definition, outliers are the data points that behaves anomalously w.r.t. other data points and hence can be defined as the data tuples that introduce more chaos in the data than other data tuples.The chaos in the data is best measured by entropy of the dataset D, which is defined as:

$$H(X) = \sum_{x \in \chi} p(x) log\left(\frac{1}{p(x)}\right)$$

, where X is the associated random variable representing the data.
The contribution of outliers in the entropy of data is large, at least more than the other normally behaving data points. Being probability measure, $0 \leq p(x) \leq 1$ and log function (increasing in $(0, \infty)$) grows very rapidly in the interval $(0, 1]$, also note that when $p(x)$ becomes smaller and smaller, $log(\frac{1}{p(x)})$ becomes larger and larger and growth rate of log is much more rapid than the rate of decrease of $p(x)$. Hence, the tuple contributes a high value to the entropy (chaos) of the dataset, as it becomes more and more rare (becomes outlier). The following figure [1] explains it pictorially.
Consider the data set D and the goal is to find an outlier set O(containing top k outliers, where k is a user-chosen parameter) as a subset of the entire dataset D, then the problem of finding the (global) top-k-outliers by the following optimization problem as defined in [1]:

$$\begin{aligned} &\underset{O \subseteq D}{min} H(D - O) \\ &s.t.\ |O| = k. \end{aligned}$$

i.e., outlier set is the subset of the entire data, getting rid of which from the original dataset minimizes the entropy

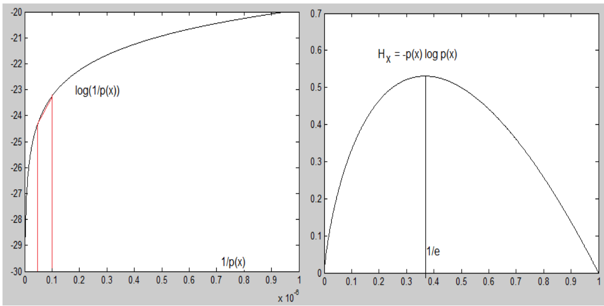
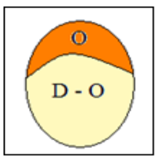


Figure 1. Contribution of a data point to the total entropy of the data set



of the original dataset. Obviously, this creates a partition in the dataset, $D \cap (D - O) = \phi$. If the variable X is multi-dimensional (e.g., n dimensional), then computation of joint entropy involves computation of joint distribution, that can be very expensive, especially when the number of dimensions are huge. If we assume that the dimensions are independent, we can use *chain rule of entropy* and easily prove the following:

$$\begin{aligned} H(X_1, X_2, \ldots, X_n) &= \sum_{i=1}^{n} H(X_i | X_{i-1} \ldots X_1) \\ &= \sum_{i=1}^{n} H(X_i) \end{aligned}$$

,
when X_is are i.i.d. random variables.

5.3.1 Distributed Extension

In order to extend the problem to its distributed version, we use the above same formulation for finding the local outliers at each and every node. In order to find global outliers from the local outliers stored in each node, we use gossiping [3] with epidemic/rumor spreading model.
Assume that the entire dataset D is horizontally partitioned into N data subsets and placed at N different nodes, i.e., $D = \bigcup_{i=1}^{N} D_i$ and the i^{th} site contains m_i data tuples ($m_i \times n$ data matrix) saved locally, so that the total number of data tuples m is given by $m = \sum_{i=1}^{N} m_i$. Then our optimization problem becomes:

$$\min_{O \subseteq \bigcup_i D_i} H\left(\bigcup_i D_i - O\right) \qquad s.t.\ |O| = k. \tag{1}$$

We approximate the above (centralized) global optimization problem by combination of the following N local optimization problems (solved parallelly and locally at N sites) followed by a global optimization problem of a much smaller size (solved by gossiping):

$$O_i^{min} \leftarrow \underset{O_i \subseteq D_i}{argmin}\ H\left(D_i - O_i\right) \qquad s.t.\ |O_i| = k, \forall i = 1 \ldots N \tag{2}$$

$$\min_{\left(O \subseteq \bigcup_i O_i^{min}\right)} H\left(\bigcup_i O_i^{min} - O\right) \qquad s.t.\ |O| = k. \tag{3}$$

Here we have made an important assumption: A global outlier must appear in at least one of the local outlier set and hence top-k global outliers must be present in the union of all top-k local outlier sets. If O is the global outlier set and O_i are the sets of local outliers, $\forall i = 1 \ldots k \Rightarrow O \subseteq \bigcup_{i=1}^{N} O_i$, where N is the number of nodes in the distributed version of the problem.

Hence once any two nodes find their local top-k outliers, they can gossip and find their global top-k-outliers from their combined local top-k-outliers set. In order to find the top-k-outliers from the combined sets, we use the entropy minimization again even when using gossiping.

5.3.2 Analysis

The local top-k outlier finding algorithm (at i^{th} site) is $\theta((m_i - k).k) = \theta(n)$ for a small constant k. Also, rumor based gossip protocols converge in $\theta(log(N))$ steps [3], hence total of $Nlog(N)$ communication messages at the worst case.

5.4 The Statistical Approach

- The local Mahalanobis distance: computed from the local mean for each of the nodes, using the local co-

Algorithm 1 ComputeGlobalTopKOutliers: Distributed Outlier Detection via gossip/rumor spreading

Inputs

(1) $\aleph = \{n_1, n_2, \ldots, n_N\}$: set of N nodes, each node n_i with its dataset D_i ($m_i \times n$ matrix).
(2) K.
(3) $\lambda_{threshold}$: threshold to measure accuracy of computed outlier set.
(4) $stop_{threshold}$: probability with which an infected node stops spreading infection.

Output: top-k global outliers.

1: $\forall n_i \in \aleph$, $O_i \leftarrow ComputeLocalTopKOutliers(D_i, k)$.
2: $\forall n_i \in \aleph$, $state(n_i) \leftarrow susceptible$.
3: $state(n_1) \leftarrow infected$.
4: do steps $5 - 19$ parallelly at each node until convergence (no change in value at each node).
5: **if** $state(n_i) = infected$ **then**
6: $\quad j \leftarrow random(N)$.
7: $\quad$ send O_i to the node j {spread infection / rumor}.
8: $\quad$ receive combined outlier set O_{ij} from node j.
9: $\quad O_i \leftarrow O_{ij}$.
10: $\quad$ **if** $|O_{ij} - O_i| < \lambda_{threshold}$ **then** {check if it already knows the rumor}
11: $\quad\quad state(n_i) \leftarrow removed$ with probability $stop_{threshold}$. {no longer interested to spread the rumor}.
12: $\quad$ **end if**
13: **end if**
14: **if** node j receives O_i from node i **then**
15: $\quad$ compute combined outlier set O_{ij} by:
16: $\quad O_{ij} \leftarrow ComputeLocalTopKOutliers(O_i \cup O_j, k)$.
17: $\quad O_j \leftarrow O_{ij}$.
18: $\quad$ send O_{ij} as reply back to node i.
19: **end if**

Algorithm 2 ComputeLocalTopKOutliers: Local Top K Outlier Detection via entropy minimization

Inputs

(1) Local dataset D.
(2) k.

Output: top-k local outliers.

$m \leftarrow$ size of the dataset D (i.e., number of data tuples).
$O \leftarrow$ select any k tuples from dataset D (e.g., first k tuples).
for $i = 1$ to k **do**
$\quad$ **for** $j = 1$ to $m - k$ **do**
$\quad\quad$ swap i^{th} tuple from O and $m - k^{th}$ tuple from $D - O$.
$\quad\quad$ recompute the entropy of $D - O$.
$\quad\quad$ **if** entropy not reduced **then**
$\quad\quad\quad$ re-swap the pair.
$\quad\quad$ **else**
$\quad\quad\quad$ obtained a new set $D - O$ with lesser entropy.
$\quad\quad$ **end if**
$\quad$ **end for**
end for

variance matrix, i.e.,

$$
\begin{aligned}
(d_{ij})_{local} &= (x_{ij} - \mu_i)^T C_i (x_{ij} - \mu_i) \\
&= (x_{ij} - \mu_i)^T (\frac{1}{m_i} \bar{X_i}^T \bar{X_i})(x_{ij} - \mu_i) \\
(\bar{d_{ij}})_{local} &= \frac{(d_{ij})_{local}}{\max\limits_{j}(d_{ij})_{local}} \text{(Normalized)}
\end{aligned}
$$

Obviously, higher the local Mahalanobis distance, higher the point will behave anomalously w.r.t. the local data set in the partition X_i. If m_i is very huge and we want mean μ_i to be less affected by local outlier, we can randomly select (fairly large number of) samples from the partition X_i and use their mean to estimate μ_i. Rank the tuples w.r.t. ascending distance and choose top k data tuples, these are the local outlier points for node i.

- The global Mahalanobis distance: computed from the global mean for each of the nodes, using the global (approx.) covariance matrix, i.e.,

$$
\begin{aligned}
&(d_{ij})_{global} = (x_{ij} - \mu)^T C (x_{ij} - \mu) \\
&= \left(x_{ij} - \frac{1}{N} \sum_{i=1}^{N} \mu_i \right)^T \left(\frac{1}{m} \sum_{i=1}^{N} \bar{X_i}^T \bar{X_i} \right) \left(x_{ij} - \frac{1}{N} \sum_{i=1}^{N} \mu_i \right) \\
&(\bar{d_{ij}})_{global} = \frac{(d_{ij})_{global}}{\max\limits_{j}(d_{ij})_{global}} \text{(Normalized)}
\end{aligned}
$$

The global distance reveals the tuples with global anomalous behaviors. We can select top k global outliers directly by choosing the points with the largest k distances.

5.5 The Spectral Graph Theory Approach

If the local Mahalanobis distance matrix is considered as the adjacency matrix for the underlying graph and $exp^{-\bar{d}(i,j)}$ is used to denote the similarity (proximity) of any data tuple with the local dataset (the tuple which is an outlier will have lesser similarity), and then compute the **Laplacian**, followed by the eigen vectors of the Laplacian, then use the second smallest eigen vector obtained to partition the graph, we shall obtain a clear demarcation between the outlier and the non-outlier points. Basically, the CUT value is to be minimized, given by, $cut(A,B) = \sum_{u \in A,\, v \in B} w(u,v)$. The optimal solution has a tendency to separate the isolation points, but we are trying to find anomalous points here, instead of a balanced cut, so will go with it. We try to minimize the Rayleigh coefficient to find the parition:

$$\underset{\substack{x \neq 0 \\ e^t x = 0}}{\text{minimize}} \frac{x^t L x}{x \cdot x} \text{ over } x \Rightarrow x = z_2, \frac{x^t L x}{x \cdot x} = \lambda_2$$

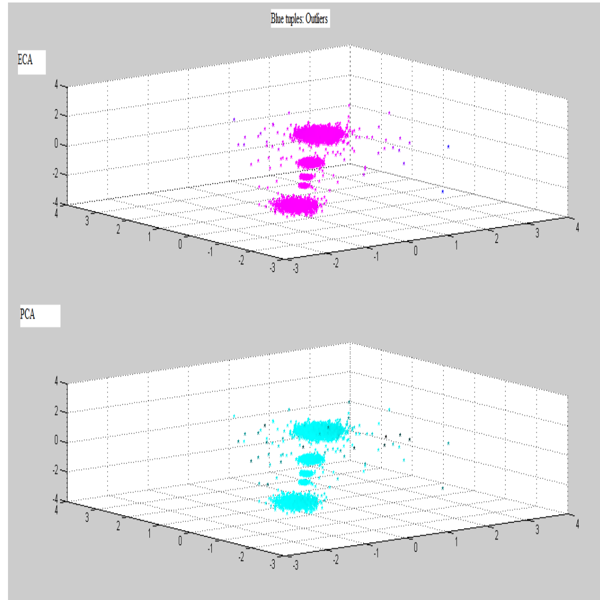


Figure 2. ECA detects more outliers than PCA for the multi-variate-synthetic dataset

6 Experimental Results

In the following results that are obtained using eigen analysis approach, color code mapping is used to show the outlier tuples obtained.

The following section describes the experimental results that were obtained using entropy minimization approach. The local outliers are computed and combined to form global outliers via gossip [3] using the same entropy minimization technique. Also the results between distributed and centralized versions are compared using metrics like precision and recall. The algorithm was implemented in Erlang and was tested with a 100 simulated nodes.

Here are some results (run in 10 nodes, each node containing 100 data tuples with 10 dimensions, we are interested to find top-k outliers):

Here HD denotes entropy of the data set and LO denotes the local outlier set, the nodes gossip and agree upon their local outlier sets and run the same entropy minimization algorithm to update each other's local outlier sets and converge to global outlier sets. As can be seen from the above result, in most of the cases the precision (measures what fraction of the outliers computed by the algorithm are actual top-k-outliers or not) and recall (measures what fraction of the

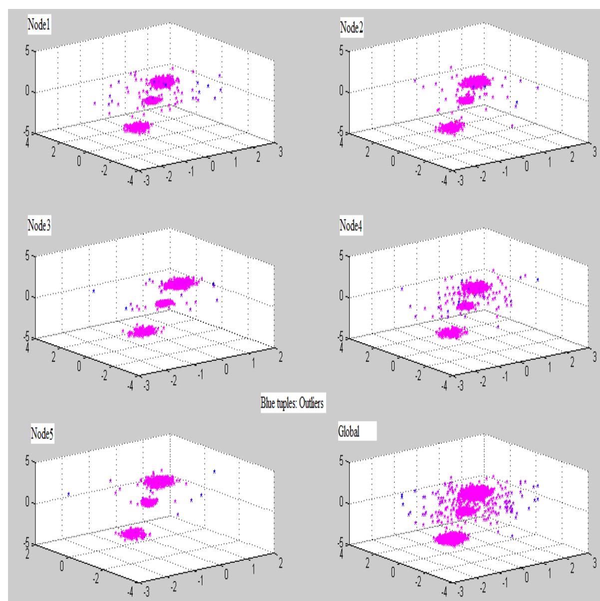


Figure 3. Distributed outlier detection using covariance matrix and weighted average of local eignevectors as global eigenvectors

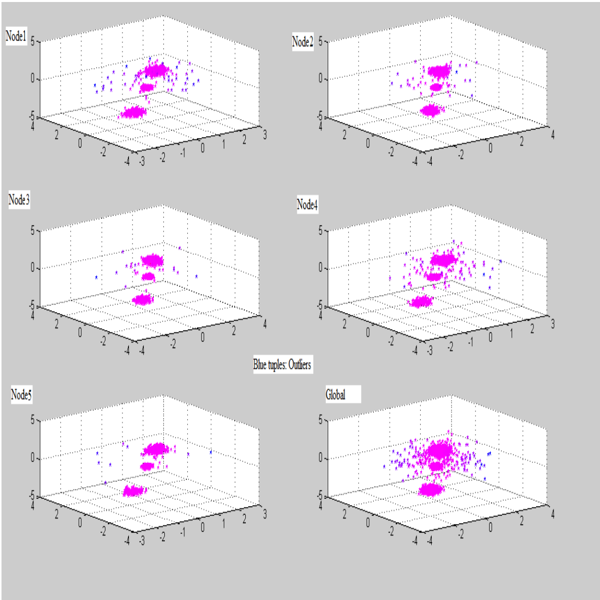


Figure 4. Distributed outlier detection using information matrix and weighted average of local eignevectors as global eigenvectors

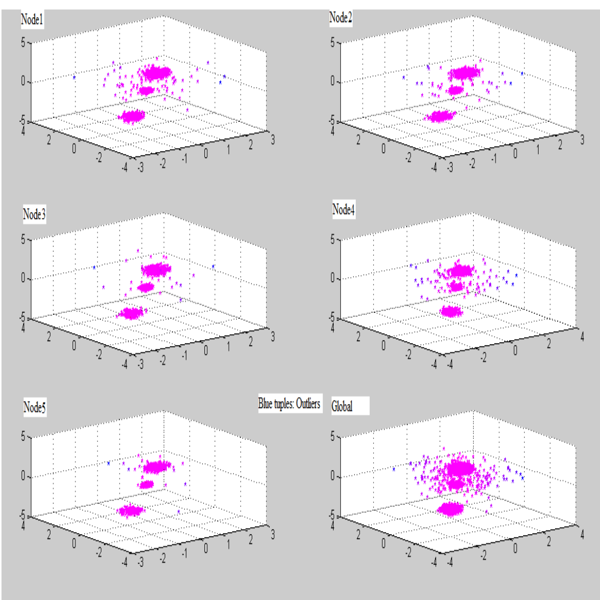


Figure 5. Distributed outlier detection using additive decomposibility of the covariance matrix

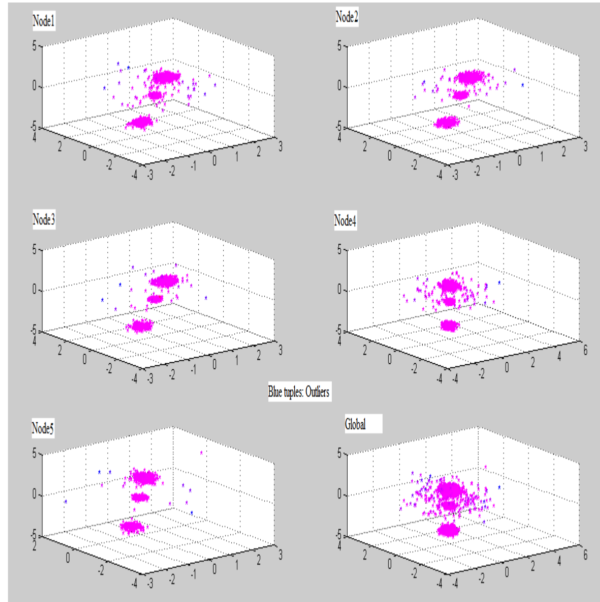


Figure 6. Distributed outlier detection assuming approximate additive decomposibility of the information matrix

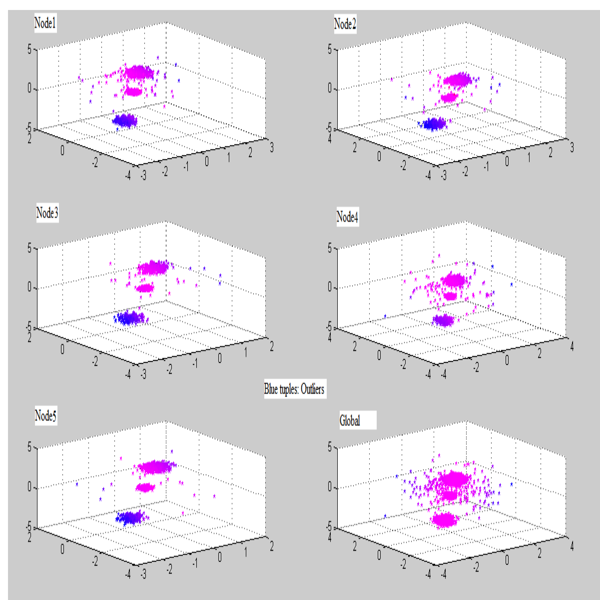


Figure 7. Distributed outlier detection using SVD of the local covariance eigenmatrices

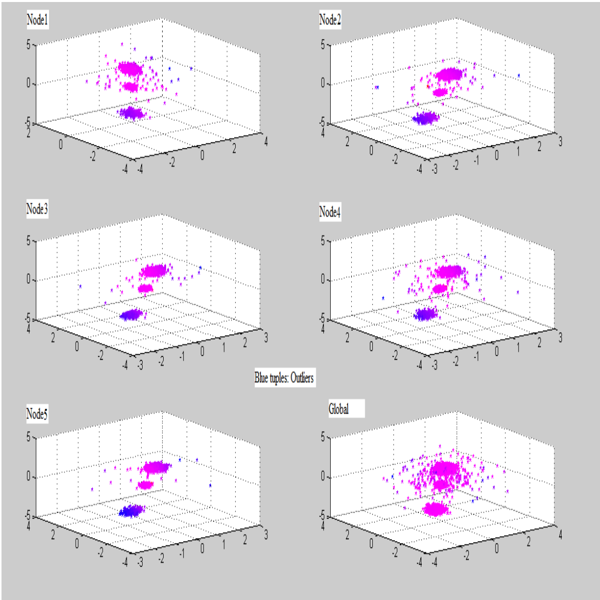


Figure 8. Distributed outlier detection using SVD of the local information eigenmatrices

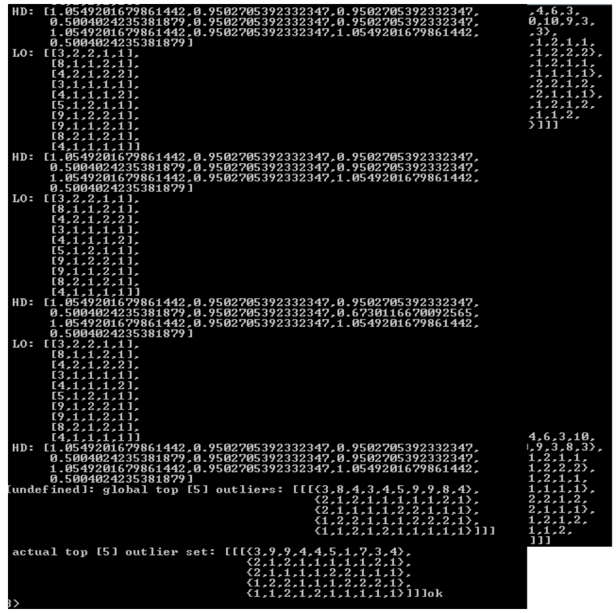


Figure 9. nodes gossiping to find top-k global outliers from their local-k outliers

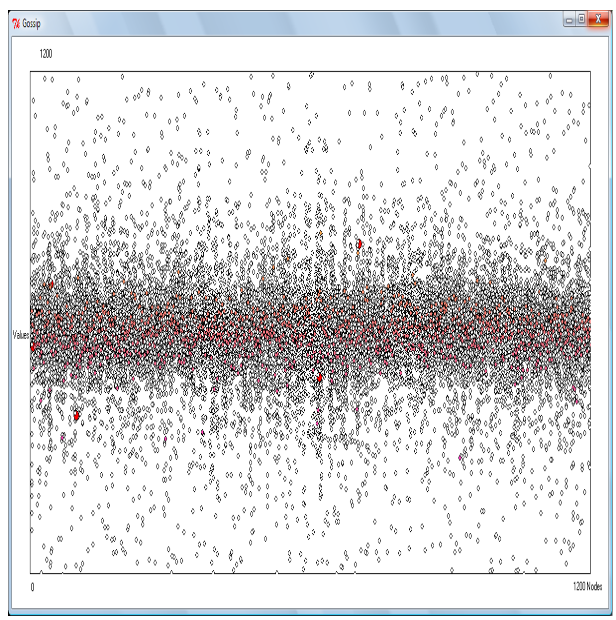


Figure 10. convergence of gossip protocol for distributed avergage computation

top-k outliers are actually output) are high (in the above example shown both are 80 percent though, a bit low, can be increased by increasing number of iterations).

6.1 Ranking the top k outliers

There can be several choices for selecting top k outliers for the entire data set X.

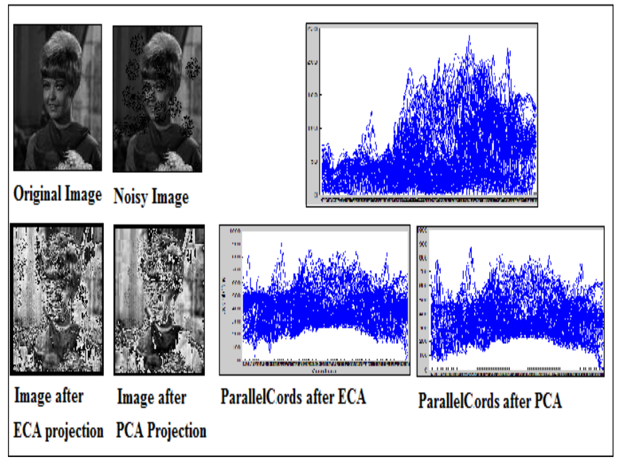
- We can select top k global outliers.
- We can select top k local outliers from each of the partitions (total Nk and assign some score (by normalization) to each of the outliers, then select top k out of them. In order to obtain the outlier scores we can use any formula like the following:
  $score = \alpha.\bar{d}_{local} + (1-\alpha).\bar{d}_{global},\ 0 \leq \alpha \leq 1,$
  or any other convex combination of local and global Mahalanobis distances. Notice that if both global and local distance are low, then the data point can never be an outlier, otherwise it is always susceptible to be one, it will almost surely be an outlier if both are high.
- we can use **Kemeny rule** for rank aggregation (ranks given / different scores assigned to the data tuples by different nodes locally and globally) and turn it into an optimization problem (IP, NP hard, LP relaxation).

We can use the existing eigen monitoring algorithm for many of the computations.

We may use l_∞ norm along with the Mahalanobis distance to rank.

6.2 Noise Removal from Images using Outlier Detection Techniques ECA and PCA

- Compute Mutual Information (or covariance) matrix and do eigen analysis on the matrix.
- Project the data on top k most dominant eigenvectors found and compute error from original data points.
- Outlier points are the one with maximum errors.
- Use the same idea on image data.
- Difficult to define attributes.
- The results shown below are not good because attributes were not just chosen properly, for an m n image the n columns were the attributes chosen, still the techniques are able to remove noise somewhat.
- For any 8 - bit gray-scale image each pixel can be thought of having 8 independent attributes, one belonging to a single bit plane.
- For colored 24 bit images, each pixel can be thought of having 3 idependent color components or HSVs.



7 Conclusion

References

[1] *An Optimization Model for Outlier Detection in Categorical Data*, 2005.

[2] Joel W. Branch, Boleslaw K. Szymanski, Chris Giannella, Ran Wolff, and Hillol Kargupta. In-network outlier detection in wireless sensor networks. In *ICDCS*, page 51, 2006.

[3] Alan J. Demers, Daniel H. Greene, Carl Hauser, Wes Irish, John Larson, Scott Shenker, Howard E. Sturgis, Daniel C. Swinehart, and Douglas B. Terry. Epidemic algorithms for replicated database maintenance. *Operating Systems Review*, 22(1):8–32, 1988.

[4] Inderjit S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *KDD*, pages 269–274, 2001.

[5] Haimonti Dutta, Chris Giannella, Kirk D. Borne, and Hillol Kargupta. Distributed top-k outlier detection from astronomy catalogs using the demac system. In *SDM*, 2007.

[6] Edwin M. Knorr, Raymond T. Ng, and V. Tucakov. Distance-based outliers: Algorithms and applications. *VLDB J.*, 8(3-4):237–253, 2000.

[7] Ying Liu and Alan P. Sprague. Outlier detection and evaluation by network flow. *IJCAT*, 33(2/3):237–246, 2008.

[8] Spiros Papadimitriou, Hiroyuki Kitagawa, Phillip B. Gibbons, and Christos Faloutsos. Loci: Fast outlier detection using the local correlation integral. In *ICDE*, pages 315–, 2003.