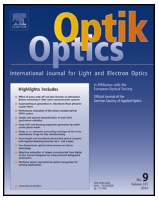
Contents lists available at ScienceDirect

Optik

journal homepage: www.elsevier.de/ijleo



Novel heuristic dual-ant clustering algorithm for network intrusion outliers detection

Tao Li[a,b,*], Nan-feng Xiao[b]

[a] Modern Education and Technology Center, South China Agricultural University, Guangzhou 510640, Guangdong, China
[b] School of Computer Science and Engineering, South China University of Technology, Guangzhou 510640, Guangdong, China

ARTICLE INFO

Article history:
Received 18 November 2013
Accepted 19 June 2014

Keywords:
Dual-ant
Anomaly detection
Clustering
Network intrusion
Heuristic algorithm

ABSTRACT

To solve the problem which unsupervised clustering algorithm is sensitive to parameter settings, the paper proposes a novel heuristic dual-ant clustering algorithm, some problems such as cluster dispersion and over many outliers which exists in traditional algorithm are resolved by adding a new kind of Maintenance Ants. The paper also propose novel heuristic functions to measures the instances similarity and to control the ant movement. Compared with other clustering algorithms, our algorithm do not need to know the number of clustering in advance, the dataset can be automatically clustered in the case of no prior knowledge, it is very suitable for intrusion anomaly detection based on unsupervised clustering. In experiments on network intrusion dataset, our algorithm is compared with the advanced cluster-based anomaly detection algorithm FindCBLOF, without knowing the original partition information of dataset, the experimental results is significantly better than FindCBLOF. It proved our algorithm has a good application value in network intrusion detection.

1. Introduction

Outlier detection is an important research content in data mining, the advantage of unsupervised clustering-based outlier detection algorithm is not need to label the training set and filtering the data set strictly, and it also has a better effect on unknown data detection and the efficiency of the algorithm is relatively high [1–3]; the drawback of clustering-based outlier detection algorithm is that we do not have suitable method to select the cluster radius, and the ratio of the number of normal class, so it still requires manual intervention [4,5]. The clustering radius values will directly affect the structure of the resulting cluster model, and outlier detection results are more sensitive to this parameter. Purpose of this paper's study is to find a solution to resolve this problem. In this paper, we presents a dual-ant clustering algorithm for outlier detection, analysis of the advantages and disadvantages of the algorithm processes, and put forward some measures to improve the clustering effect, apply it to outlier detection, and achieved good results [6,7].

2. Algorithm analysis

The ant clustering algorithm is a kind of bionic algorithm based on swarm intelligence [8]. In general, such as K-means clustering algorithm, is very sensitive to the number of initialization clusters, the number of initialization clusters directly affects the results of clustering. The ant clustering algorithm does not require a pre-specified number of clusters to initialize, so that it can overcome the high sensitivity defect of traditional clustering in initialization, thereby improving the effect of clustering [9,10].

The basic idea of the algorithm is: first, we distribute the multidimensional data points which to be classified in a two-dimensional plane randomly, and then introduce some artificial ants in the two-dimensional plane. These ants do not have a global vision, they cannot communicate with each other directly, and only has a very small temporary memory storage space. Ants randomly moving on the two-dimensional plane. According to the local area similarity of the data instances found by ants, we get the probability of the decision whether the ants "Pick up" or "Put down" the data instances. After a finite number of iterations, the data instances in the plane gather according to their similarity. In the basic model of this algorithm, the ants can only make random motion on a given range two-dimensional plane, follow four directions-up, down, left, and right [7,11].

Algorithm process is as follows:

Input: Initial number of ants – N, two-dimensional plane size – A, the number of cycles – *count*, memory space length – L, the loop count – $c = 1$.

[*] Corresponding author. Tel.: +86 13570292543.
E-mail address: cyclinster@gmail.com (T. Li).

http://dx.doi.org/10.1016/j.ijleo.2014.08.036
0030-4026/© 2014 Elsevier GmbH. All rights reserved.

Step 1: The data objects and ants are randomly placed in two-dimensional plane, in any one location of the two-dimensional plane can be placed only one data or one ant.

Step 2: Start clustering algorithm cycle, $c \leftarrow c + 1$, if $c \le count$, go to **Step 3**; otherwise go to **Step 1**.

Step 3: Choose an ant.

Step 4: The present ant moves on the two-dimensional plane randomly, if the ants encounters a data instance, we calculate the probability $P_{pick}(i)$ according to Eq. (1), it is the probability of the ant picking data instances.

$$p_{pick}(i) = \left(\frac{k^+}{k^+ + f(i)}\right)^2 \qquad (1)$$

Step 5: If the picking up probability $P_{pick}(i)$ is small, the ant does not pick up the data instance, the ants are still no-load state, and continues to move randomly, then go to **Step 4**; if the picking up probability $P_{pick}(i)$ is larger, the ant picks up the data instance, the status of the ant is changed to located, and continues to move randomly.

Step 6: If the ant is on a loaded state when it is moved to an empty position, we calculate the probability $P_{drop}(i)$, it is the probability of the ant putting down the data instances, it is represented by Eq. (2):

$$p_{drop}(i) = \left(\frac{f(i)}{k^- + f(i)}\right)^2 \qquad (2)$$

Step 7: If the putting down probability $P_{drop}(i)$ is small, the ant do not put down the data instances, the ant is still on loaded state, and continues to move randomly, then go to **Step 6**; if the putting down probability $P_{drop}(i)$ is larger, the ant put down the ant, the ants change to no-load state, continue to move randomly until picking up a data instances.

Step 8: If all the ants in group moved over, then go to **Step 2**, otherwise go to **Step 3**.

Step 9: If the number of cycles achieves predefined parameters-count, output clustering results, the algorithm ends.

In the above Eqs. (1) and (2), the parameter i is the data instance which ant encountered, $P_{pick}(i)$ and $P_{drop}(i)$ are the probability conversion function, they convert the average of the similarity function $f(i)$ into the probability of ant picking up or putting down the data instance. The parameter k^+ and k^- are the constraint factor in the interval [0,1]. $f(i)$ indicates that the proportion of how many this categories of data instance in ants recently walked the few steps, for example, the length of ant's temporary memory space is m, if a total of n such categories of recorded data instances in size m memory space, then $f(i) = n/m$. Obviously, the value of $f(i)$ are in the interval [0,1].

In this paper, we consider the characteristics of unsupervised outlier detection, and improve the classical ant clustering algorithm, an improved dual-ant clustering algorithm is proposed, let it be better used in outlier detection. In the original ant clustering algorithm, only one kind of ant moves randomly and manipulates data instances, referred to herein as clustering ants. In this paper, on the base of the original algorithm has only one kind of clustering ants, we add another kind of Maintenance Ant, each Maintenance Ants belongs to one and only one cluster, so that we let Maintenance Ant manage each cluster, the algorithm can be from to control the clustering process on the cluster level. The main tasks of Maintenance Ant are: (1) expansion of clusters; (2) the integration of clusters; (3) recording the most dissimilar data instances. The primary role of Maintenance Ants is to maintain the generated cluster during the clustering process, take the maintenance for the size of the cluster, the members of cluster and the cluster center, dynamically updated the status of cluster. The purpose of the improved algorithm is to solve some problems exist in the original algorithm, including clustering dispersed, too much outliers, etc. In short, the improved algorithm can improve the clustering effect.

2.1. The expansion of clusters

The expansion of the cluster is carried out after the cluster is generated, Maintenance Ant search the coverage area around the center of cluster in a certain range, to identify those data instances in accordance with the corresponding cluster, such as the data instances with high similarity to the cluster center. This method adopted in this paper is that we firstly calculate the cluster center, and then calculate the peripheral distance from the center to around each instance, then the distance values are compared. In order to determine whether the instance belongs to the cluster and select the most dissimilar elements – outliers, in this paper we compared the distance between the data instance and the cluster center with the cluster radius.

The cluster center and the cluster radius are defined as follows:

For a given cluster C, the number of members in C is n, each member X_i is a m-dimensional vector:

$$X_i = \{x_{i1}, x_{i2}, \ldots x_{im}\}, \quad (1 \le i \le n)$$

The cluster center is defined as the m-dimensional vector $X_c = \{x_{c1}, x_{c2}, \ldots, x_{cm}\}$,wherein, $x_{ci} = (1/n)\sum_{j=i}^{n} x_{ji}$.

The cluster radius R_c is defined as Euclidean distance, the cluster radius is a distance $R_c = \max\{d_{ic}\}$, $(1 \le i \le n)$,wherein, $d_{ic} = \sqrt{\sum_{k=1}^{m}(x_{ik} - x_{ck})^2}$

The comparison process is defined as follows:

For any searched instances X_j surrounding the cluster and is not part of the cluster, we calculate the distance d_{jc} between the cluster center X_c and the instances X_j:

$$d_{jc} = \sqrt{\sum_{k=1}^{m}(x_{jk} - x_{ck})^2}$$

Then we compare d_{jc} and R_c. If $d_{jc} \le R_c$, then X_j belong to the cluster, and the instance is labeled as "processed", indicates that the instance has been classified into a cluster, and if the Clustering Ants encountered this instance, the ant directly leaves and do not deal with the instance; if $d_{jc} > R_c$, only record the distance, without further action. After all surrounding instances are finished searching, the instances that belong to the cluster has been finished adding into the cluster. For all of those instances who do not belong to the cluster, we identify the instance with the largest d_{jc} value and recorded it, it is looked on as the priority picking up operation instance by Clustering Ants. This instance has the greatest likelihood to be an outlier.

The search range is defined as following: let the number of members in the cluster C is n, each member X_i has the coordinates (x_i, y_i) on the two-dimensional plane, herein $(1 \le i \le n)$. The search center is C_{fake}, C_{fake} has the coordinates $p_{fake} = ((1/n)\sum_{i=1}^{n} x_i, \ (1/n)\sum_{i=1}^{n} y_i)$ on two-dimensional plane. The search range is a square area with the center C_{fake}.

2.2. The fusion of clusters

When Maintenance Ants search the surrounding area, Maintenance Ants will take different actions for the data instances they encounter according to different characteristics and states of the data instances.

Specific operational procedures are as follows:

Let the cluster where current Maintenance Ant belongs to is C_A, the cluster radius of C_A is R_A, the Maintenance Ants search and find a