Research on Sequential Optimized Method of Network Security Alarms Based on Users Feedback

Peihua Su

Xi'an International University, 710077, Xi'an Shaanxi China

Abstract

As for the doubts about the most serious network security alarms presented by users, a method of modifying evaluation parameters based on user’s feedback about network security alarms has been proposed. In case a preference is set in system-initialization and top-k technology is used to acquire k alarms that users think are the most serious among network security alarms. As for user’ similar doubts, this paper provides a method of modifying evaluation parameters based on user’s feedback. This method obtains candidate weights by sampling from subspace of weight to redefine a new query. At first, it defines the assessment model function which can be used to measure changes of initialized top-k query. Then based on this, a further fine-grained depiction is carried out for sampling the space of the sampling to make sampling space more precise. At last, in given candidate weight, the new optimal query can be gained from processing the incremental top-k algorithm. In this process, the assessment model function is used to make a further optimization for the terminal conditions of the incremental top-k algorithm. It is needed to terminate top-k query which can not be used to acquire the optimal solution as early as possible, which greatly improves the efficiency to execute the algorithm. The results of this experiment shows that the algorithm proposed in this paper has a better execution efficiency

Keywords: Sequential optimized method; Network security; Alarm system; user feedback mechanism

1. INTRODUCTION

In the perception of the network security, in order to facilitate users to use limited resources to deal with the most serious alarms, this article returns k as an uncertain number among the most threatening alarms to users from network security alarms by using the method of top-k (Porras, 2009). But for the k network security alarms returning from the system, the users may ask the question “why”. One such question was raised, “I think the alarm m should be more serious than alarm p, but why did alarm p appear in the most threatening alarms and alarm m didn’t?” (Bass, 1999). This question indicates that the query parameters set in the top-k query (the parameter k and user preferences) may have a gap in the expectation of users (Fan and Zhang, 2001).

In response to this question and based on users’ feedback, how to answer the question of “why” raised by users must be studied (Luker, 2003). In view of the above question, this article proposes a query modification method based on users’ feedback. Generally, the top-k query depends on two parameters: the parameter k and the parameter $\overline{w}$,
the parameter $k$ control returned results, while the parameter $\vec{w}$ shows user preferences in various attributes of data.

So in order to explain user’s questions for the sequential results, we need to adjust the parameters of query based on users’ feedback. To solve this problem, this article defines an appraisal model to measure the differences between new query and the results of initialized query. Based on the appraisal model, this chapter tries to return a new query to users, and this new query can either efficiently answer users’ questions or minimize the variety of initialized results.

2. TOP-K QUERYING QUESTIONS

2.1 Problem description

To define the top-$k$ Query, in case a positive integer $k$ and a weight vector $\vec{w}$ are given, the result of top-$k$ query $Q(k,\vec{w})$ is an ordered object column, and this object column satisfies $Q(k,\vec{w}) \in D$, $|Q(k,\vec{w})|=k$ and $\forall p_1, p_2: p_1 \in Q(k,\vec{w}), p_2 \in D - Q(k,\vec{w})$ and all of them conform to $\text{score}(p_1,\vec{w}) > \text{score}(p_2,\vec{w})$.

<table>
<thead>
<tr>
<th>Warning signs</th>
<th>Importance of assets</th>
<th>Threat degree</th>
<th>The frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.6</td>
<td>0.4</td>
<td>0.1</td>
</tr>
<tr>
<td>2</td>
<td>0.1</td>
<td>0.3</td>
<td>0.8</td>
</tr>
<tr>
<td>3</td>
<td>0.6</td>
<td>0.5</td>
<td>0.6</td>
</tr>
<tr>
<td>4</td>
<td>0.3</td>
<td>0.6</td>
<td>0.5</td>
</tr>
<tr>
<td>5</td>
<td>0.9</td>
<td>0.5</td>
<td>0.8</td>
</tr>
</tbody>
</table>

Table 1 Network security warning example

To define the object of No. $k$, in case a positive integer $k$ and a weight vector $\vec{w}$ is given, the object of No. $k$ Tuple($k,\vec{w}$) is an object $p$ in the database, and this object satisfies $p \in (k,\vec{w}) \cdot Q(k-1,\vec{w})$. In the beginning, the user gives a query of top-$k$ $Q_0(k_0,\vec{w})$. Based on the returned result from the query, users may raise the question: “The object $p$ appeared in the result, but why did the object $m$ not appear?” Before answering this question, it must first be assumed that the object $m$ actually exists in the database, and secondly that the object $p$ and the object $m$ can not be compared with each other; that is, neither the object $p$ can control the object $m$ nor can be controlled by $m$. Based on these assumptions, in order to answer the questions of “why” raised by users, the initialized query $Q_0(k_0,\vec{w})$ must be modified. After modification, the necessary and sufficient conditions of the new query answering the question of “why” are:

1. The object $m$ that users expect appears in the results of new query $Q(k,\vec{w})$.

\[
\text{score}(p_1,\vec{w}) > \text{score}(p_2,\vec{w}).
\]

(2) Obviously, there are many new queries which satisfy the two above conditions. So an appraisal model must be defined to measure the efficiency of the new queries. The appraisal model is defined as follows:

\[
\text{Eval}(Q(k,\vec{w})) = \lambda_\omega \Delta W + \lambda_\alpha \Delta C
\]
Where: \( \Delta W = \| \vec{w} - \vec{w}_0 \| \) is used to indicate the differences between two query vectors, \( \Delta C \) is used to measure the differences between the two times query results, \( \lambda_w, \lambda_c \) satisfy \( \lambda_w + \lambda_c = 1 \) (Fan and Zhang, 2001). The difference \( \Delta C \) between the query results is calculated by the following formula:

\[
\Delta C = \left| Q(\vec{k}_0, \vec{w}_0) \cup \overline{Q(\vec{k}_0, \vec{w}_0)} - \left[ Q(\vec{k}_0, \vec{w}_0) \cap \overline{Q(\vec{k}_0, \vec{w}_0)} \right] \right|
\]

(2)

Because \( \Delta C \) is much greater than \( \Delta W \), so the formula (1) must be further normalized. It can be seen from the appraisal model (1) that the new query possessing a smaller value has higher efficiency, because it made a slight change to the original query \( Q_0(\vec{k}_0, \vec{w}_0) \).

2.2 Problem analysis

According to the definitions and discussion about the above questions, a new query based on the modification of original query is needed. This new query must satisfy that the object \( m \) which users expect ranks ahead of its compared object \( p \); that is, \( \text{score}(\vec{m}, \vec{w}) > \text{score}(\vec{p}, \vec{w}) \). In the data space, if the object \( m \) controls the object \( p \), which means \( \text{score}(\vec{m}, \vec{w}) > \text{score}(\vec{p}, \vec{w}) \) always makes sense; if the object \( p \) controls the object \( m \), then \( \text{score}(\vec{m}, \vec{w}) > \text{score}(\vec{p}, \vec{w}) \) is reasonable. Hence it can be assumed that the object \( p \) and \( m \) in the questions of “why” raised by users cannot be compared with each other; that is, there is no controlling relationship between the object \( p \) and \( m \).

Based on the incomparable data objects \( p \) and \( m \), the one hyper-plane \( H \) of weight space \( \vec{H} : (\vec{m} - \vec{p}) \cdot \vec{w} = 0 \) divides the weight space into two parts \( W> \) and \( W< \). All the weight vectors in the area of \( W> \) make \( \text{score}(\vec{m}, \vec{w}) < \text{score}(\vec{p}, \vec{w}) \). Therefore to get a new query satisfying the conditions, the sub-space \( W> \) of weight space only need be considered. For every weight vector in the sub-space, a new optimal query can be obtained which satisfies the conditions [5]. On this basis, for the new queries of all weight vectors, the efficiency of optimal results under each weight vector can be obtained by the appraisal model (1). So, the new query possessing the highest efficiency can be considered as the final answer to users’ question of “why”. But in weight space there are infinite possible weight vectors \( \vec{w} \), so it is difficult to obtain the optimal result by listing all the weight vectors. Therefore, this paper proposes a new optimized algorithm to answer user’s question of “why”.

3. THE TOP-K ALARM MODIFIED ALGORITHM BASED ON USERS’ FEEDBACK

3.1 Basic solution

This paper introduces the basic solution in detail, which is divided into the following four key steps:

(1) In initialized weight space \( \Gamma = \{ (\vec{m} - \vec{p}) \cdot \vec{w} > 0, \vec{w} \in [0,1], \sum w[i] = 1 \} \) s alternative weight vectors are selected from the space by sampling [7];

(2) Based on the assembly \( S \) of alternatives, for every weight vectors \( \vec{w} \in S \), each new query can be obtained by utilizing the incremental top-k computing method;
The appraisal model (1) can be utilized to appraise all new queries; The new query possessing the highest efficiency is chosen as the answer of users’ question of “why” and the query is returned to users.

3.2 Terminal condition of gradual model top-k algorithm

This subsection discusses how to get the terminal condition of gradual model top-k in the given weight vector \( \overline{w} \in S \). The gradual model top-k algorithm is an incremental calculation of top-k. Each time the algorithm is executed for a step, it will return the optimal object back to users. The terminal condition of the gradual model top-k algorithm is that all the \( k \) objects will be produced. Then, the parameter \( k \) is obtained according to \( \Delta C \).

According to the above mentioned discussion, in allusion to the given \( \overline{w} \), the new optimal query must be obtained to make the value of the appraisal model formula (1) minimum and make other new queries have the same \( \Delta W \). So to minimize the value of this formula (1), the value of \( \Delta C \) must be the minimum. According to formula (2) to solve \( \Delta C \), the \( \Delta C_{Q(k, \overline{w})} \) can be known, as well as the new query \( Q(k, \overline{w}) \). Because \( Q_0(k, \overline{w}) \) and \( \overline{w} \) have been completely confirmed, the size of \( \Delta C \) depends on the size of \( k \) (Blasch and Plano, 2002).

1 algorithm calculates the optimal parameter \( k \) in the given \( \overline{w} \). The suitability of this algorithm will be given below. To prove the correctness of algorithm 1, it is necessary to prove under the given \( \overline{w} \) that there is an upper limit to obtain the minimum value \( k \) by formula (1).

Theorem 1: in the given weight space \( \overline{w} \), the gradual model top-k algorithm can be used in \( k_m+2(k_0-a) \) to determine the minimum value \( k \) in formula (1), where \( k_m \) conforms to \( \text{Tuple}(k_m, \overline{w}) = m \) and \( a \) is the number of common objects in the sum \( Q_0(k, \overline{w}) \).

This theorem can be proved by the following two cases: (1) \( Q(k_m, \overline{w}) \) may not be the optimal value, so the gradual model top-k algorithm must be executed sequentially; (2) makes the minimum value \( k \) obtained from formula (1) with an upper limit of \( k_m+2(k_0-a) \).

The confirmation of case (1) can be demonstrated by making an example. If \( \text{Tuple}(k_m, \overline{w}) \in Q_0(k, \overline{w}) \), then the value of \( \Delta C \) decreases with an increase of \( k \), which means that in the appraisal model formula (1), \( Q_0(k_m+1, \overline{w}) \) is superior to \( Q(k_m, \overline{w}) \). This indicates that even though the object \( m \) which users expect appears in the result of top-k, to obtain the optimal solution, gradual model top-k algorithm still can not be terminated.

Case (2) can be proved by using reduction to absurdity, and it is necessary to prove that for \( \forall k > k_m+2(k_0-a) \), and \( \Delta C_{Q(k, \overline{w})} > \Delta C_{Q_0(k, \overline{w})} \). So when \( \forall k \) is greater than the given upper limit, any efficiency of a new query is below that of the existing new query \( Q(k_m, \overline{w}) \). The process of proof is as follows:


\[
\Delta \mathcal{C}^0 = |\mathcal{Q}(k_0, \overline{w}) \cup \mathcal{Q}(k_0, \overline{w})| - |\mathcal{Q}(k_0, \overline{w}) \cap \mathcal{Q}(k_0, \overline{w})|
\]

(3)

4 EXPERIMENTAL VERIFICATION

4.1 Acquisition of date sets

The method of network security evaluation based on index system can be used to make the value of every attribute field in network security quantified and uniform. As for the severity of network alarm, overall consideration are taken in multiple properties, including assets value of the targets of network attack, reliability of detection of the network attack and harmfulness of a network attacking itself (Boyd, 1996).

Because in its evaluation attribute field, the value of network security alarm in form is a real number from 0 to 1. This subsection adopts a classical method in which the date creates an algorithm to create simulative testing date set and test the effectiveness of the algorithm presented in this paper on a simulated date.

This simulated data set is generated from a classical data generation algorithm. This algorithm can generate three different data sets: uniformly distributed data set, related data set and data set of inverse correlation. In a uniformly distributed data set, the generation of the value in every attribute field is mutually independent and uniformly distributed; that is to say, the related data set means that if the value of one data record is dominant in a dimension, then it will be dominant in another dimension as well. The data set of inverse correlation is relative to related data set and it shows that one data record is dominant in a dimension but inferior in another dimension. Related data set in skyline and top-k are nearly the same, this section tests the properties of algorithm presented in this paper in uniformly distributed data set and data set of inverse correlation. When this algorithm is used to generate simulated data, it should normalize all values in every dimension to the range from 0 to 1.

To test the algorithm in different scales and different dimensions potential preferential efficiency in users’ mind must be activated. Table 2 shows the parameter generating simulated data. In subsequent experiments, the black body refers to the default parameter, and it is tested through changing each parameter’s different value.
4.2 Test and evaluation

This section is the experimental analysis about the effectiveness and execution efficiency of the algorithm presented in this paper. Because at present there is no research in this field about the query and modification problem presented in this paper, the performance evaluation of this section aims at the algorithm presented in this paper. To make an evaluation aiming at different users’ requirements, this subsection includes three kinds of different users:

Table 2 Test parameter Settings

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>The data size</td>
<td>1K,10K,50K,100K</td>
</tr>
<tr>
<td>Data dimension</td>
<td>2,3,4,5</td>
</tr>
<tr>
<td>Initialize the query</td>
<td>5,10,15,20</td>
</tr>
</tbody>
</table>

(1) TMW means that in terms of initial query result, users can handle more about the changes of weight \( \tilde{w} \).

(2) TMR means that in terms of changes of initial query weight, users can handle more about the changes of initial result of top-k.

(3) NM means that users can handle both weight and result of the initial query.

As for three kinds of users, it provides different parameters for the evaluation model formula 3 presented in this paper. As for TMW users, it respectively sets \( \lambda_c \) and \( \lambda_w \) to be 0.9 and 0.1; as for TMR users, it sets \( \lambda_c \) and \( \lambda_w \) to 0.1 and 0.9; as for NM users, it sets \( \lambda_c \) and \( \lambda_w \) to 0.5 and 0.5. All the codes of experiments are achieved by java and compiled in JDK. The procedure runs in desktop equipped with operating system of Ubuntu Linux, CPU is intel Core-2 dual-core processor and its memory capacity is 2GB.

The following experiment tests the performance of the algorithm presented in this paper in simulated data. An initial query \( Q(k_m, \tilde{w}) \) will be given and a comparable object \( p \) which the object ranking in the previous users’ question of ‘why’ will be set. The object \( m \) expected by users randomly generates from \( D-Q_0 \) but it must satisfy that \( m \) and \( p \) are not in relations of mutual control. In this premise, the effectiveness of the algorithm presented by this article is measured by the running time of test program.

4.3 Experimental results

The first experiment is based on data dimension of 3 to test the running time in this paper in different data sizes.
As Figure 1 shows, this experiment consists of four different data scales and three different tolerance models. Fig. 1 shows that in this paper, running time shows a linear growth relation with the increase of data scale. The running time of the TMR model takes the minimal time because users are more able to tolerate modifications in results of the initial query compared with changes of initial query preference in the TMR model.

The second experiment is based on a data size of 10k to test the running time in this paper in different data sizes. As Figure 2 shows, in the same data size, running time of uniform data set nearly remains unchanged, as shown in Figure 2(a). However the running time of inverse correlation data set shows a trend of rapid growth according to the increase of data dimensions. The reason is that it takes too much time in executing the incremental top-k algorithm in the inverse correlation data set, and the incremental top-k algorithm shows rapid growth trend in scale of the candidate set of top-k with an increase in data dimension.
Figure 3. Performance of the program with different initial query parameters

The third experiment is based on a data size of 10k and data dimensions of 3 to test the running time of the algorithm presented in this paper in different initial queries and users’ desired objects. According to the experiment, there is no relation between running time, k0 and m in this paper. So the operational efficiency of the program is only set for an inverse data set.

Figure 4. Quality of the new query in a different sampling schedule

Figure 3(a) shows the running time of program in change k0, because the comparable object p is the last object returning from the initialized query, and the change to k0 is the same as changing comparable object p. In addition, the difference between the sorting position of object m that users expect in the initiale query and the sorting position of object p in the initiale query is equal to or greater than 10. In Figure 3(b), under the circumstance of k0=0, algorithm efficiency in this paper can be tested by changing users’ expected object m. Where distance represents the minimum distance between the initialized query sorting position of object p and sorting position of object m. It can be seen that the users’ expected object m has an effect on the running time in this paper, but this effect is limited to object m itself and has no relation with the distance of object p in initiale query.

The last experiment is based on a data size of 10k and data dimension of 3 to test sampling precision’s influence on the quality of answering the question “why”. Sampling precision is controlled by parameter α and parameter Pr. According to computational
formula 5 of sampling precision, it can be seen that the smaller the $\alpha$ and the larger the $Pr$ are, the higher accuracy of sampling precision will be. So when it is not necessary to test the change in parameter $\alpha$ and parameter $Pr$, the new quality of query can be obtained. Figure 4 shows the new qualities of query in different sampling precision, and it can be seen that the new query qualities become more accurate with the increase in sampling weights numbers. Meanwhile, in Figure 4(b), the new query quality does not increase with the increase of sampling numbers. The reason for this is that in the new query, the projective point of weight $\vec{w}_v$ is gained from $\vec{w}_r$. Therefore, no matter how much the sampling precision changes, the optimal quality of the new query remains unchanged.

5. CONCLUSION

As for the doubts about the most serious network security alarms which are presented by users, a method of modifying the evaluation parameters based on user’s feedback about network security alarms has been proposed. A preference is set in the system-initialization and top-k technology is used to acquire k alarms that users think are the most serious from network security alarms. However, based on the consequence returned from the system, users may have doubts such as "I think alarm m is more serious than alarm p and why does alarm p appear in the most dangerous k alarms without alarm m?" As for user’ similar doubts, this paper provides a method of modifying evaluation parameters based on user’s feedback. This method gains candidate weights by sampling from subspace of weight to redefine a new query. At first, it defines an assessment model function which can be used to measure changes of an initialized top-k query. Then based on this, a further fine-grained depiction is carried out for sampling space of the sampling method to make the sampling space more precise. Finally, in a given candidate weight, the new optimal query can be obtained from processing the incremental top-k algorithm. In this process, the assessment model function is used to make a further optimization for the terminal conditions of the incremental top-k algorithm. The top-k query must be terminated which can not be used to acquire the optimal solution as early as possible, which greatly improves the efficiency of executing the algorithm. The results of this experiment shows that the algorithm proposed in this paper has better execution efficiency

6. References