CS 6035 Project II – Phase IV Malheur Summary

Zhaoran Yang
zyang625@gatech.edu

QUESTION 1

Precision, also known as positive predictive value, is the fraction of relevant instances among retrieved instances. [1] P reflects how precise the individual clusters agree with malware class and it's calculated as below.

$$P = \frac{1}{n} \sum_{c \in C} \#_c$$ [3]

where C represents a set of clusters and #c is the largest number of reports in the cluster C sharing the same class.[3]

QUESTION 2

Recall, also known as sensitivity, is the fraction of relevant instances that were retrieved.[1] R measures to which extent classes are scattered across clusters and it's calculated as below.

$$R = \frac{1}{n} \sum_{y \in Y} \#_y$$ [3]

where Y presents a set of malware classes and #y is the largest number of reports labeled y within one cluster.[3]

QUESTION 3

F-score, also known as F-measure, is a combined performance measurement of precision and recall. F-score is calculated as below.[1]

$$F = \frac{2 \cdot P \cdot R}{P + R}$$ [3]

where P is precision and R is recall. The value of F is between 0 and 1, which 0 represents the lowest value and 1 represents a perfect classification. Either a low precision or recall result in a low F-score.

QUESTION 4

In malware analysis, some malware binaries show similarity in family of variants that result in similar behavioral patterns. When machine learning techniques are applied to identify and classify behavior, the embedded reports form dense clouds in the vector space.[3] For each dense clouds, malheur uses **prototypes** to represent subgroups that contains reports being typical for a group of homogenous behavior.[3]

QUESTION 5

Prototype is introduced to bridge the gap between an increasing volume of malware and lack of proper learning methods to carry out clustering and classification analysis in time.

Malheur describes the clustering group in an incremental way. It first determines clusters of prototypes and then propagates to the original data.[3] The algorithm iterates all clusters and merge the pair of clusters if they are within the parameter **d**, which is a pre-determined value of the maximum distance between two clusters.

QUESTION 6

Assuming x is a report, malheur creates an embedding report based on instruction q-grams and S is the set of all possible q-grams for x. Then for each instruction q-grams, 1 or 0 is used to determine whether report x contains the instruction q-grams, where 1 represents a match and 0 otherwise. The corresponding embedding function is stored in $\varphi(x)$. However, in practice, we use a normalized embedding function below to mitigate bias due to various factors of input reports.

$$\hat{\varphi}(x) = \frac{\varphi(x)}{||\varphi(x)||}$$ [3]

Then, similarity of behavior is measured via distance **d**, an assessment of the geometric relations between embedded reports and is calculated as below.[3]

$$d(x,z) = ||\hat{\varphi}(x) - \hat{\varphi}(z)|| = \sqrt{\sum_{s\in\mathcal{S}} (\hat{\varphi}_s(x) - \hat{\varphi}_s(z))^2}$$ [3]

where x and z are embedded reports.[3]

QUESTION 7

Malheur uses API call names to identify and classify the corresponding malware and the order of the API call represents the signature or identity of the malware. If the order is lost, similarity of malware behaviors will be lost, such that it will not be classified to the same cluster as it should be, rather it will probably be treated as an unknown cluster and further classified into a separate cluster.

QUESTION 8

Cohesion, also known as compactness, measures how closely the objects are related within the same cluster while separation measures how distinct a cluster is from other clusters.[4] Ideally, intra-cluster cohesion is minimized, and inter-cluster separation is maximized.

Cohesion is commonly measured by sum of squares within the cluster:

$$\boldsymbol{WSS} = \sum_{i} \sum_{x \in C_i} (x - m_i)^2$$ [4]

Where Ci is the size of cluster i, m is the mean of each cluster i.[4]

QUESTION 9

For Phase 3 Goal 1, I have tuned below 2 parameters below to achieve at least 70% f-score in the testing phase.

prototype max_dist: this specifies the maximum distance to a prototype. During analysis, malheur will generate prototypes so that each report to its nearest prototype is smaller than this value. [5] Therefore, it determines the degree of similarity of malware behavior – the larger the value is, the fewer prototypes will be generated. The parameters ranges from 0 to sqrt(2).

cluster min_dist: this parameter defines the minimum distance between clusters. Clusters are successfully merged until the minimum distance between the closest pair of clusters is above this value.[5] The parameters ranges from 0 to sqrt(2).

I first ran malheur with both the default value set to 0, and from which I received an F-score of 33.7% as a result, with P = 96.4% and R = 20.4%. To achieve a higher F-score, according to the formula in Question 3, we will need the difference between P and R to be as small as possible, and theoretically, lowering the number of prototypes should work towards the direction.

Therefore, with all other parameters remain the same as default, I changed the prototype max_dist from 0 to 0.5, which I received an F-score of 62.6% as a result, with P = 94% and R = 68.3%. Since it's working towards the right direction, I changed the parameter again to 0.8. This time, I received an F-score of 75.9%, with P = 95.4% and R = 63.0%.

To explore whether there's any chance of improvement in F-score, I also increased the min_dist for cluster to 1 for experiment. I ended up getting 79.1% as my result in this task, with P = 94% and R = 68.3%.

REFERENCES

1. Wikipedia, W. (2022, February). Precision and recall. Retrieved February 27, 2022, from https://en.wikipedia.org/wiki/Precision_and_recall
2. Wikipedia, W. (2022, February). F-score. Retrieved February 27, 2022, from https://en.wikipedia.org/wiki/F-score
3. Automatic Analysis of Malware Behavior using Machine Learning. (2011) Konrad Rieck, Philipp Trinius, Carsten Willems, and Thorsten Holz. Journal of Computer Security (JCS), from http://www.iospress.nl.
4. Datanovia.com (2022 February). Cluster Validation statistics: Must know methods. Retrieved February 27, 2022, from https://www.datanovia.com/en/lessons/cluster-validation-statistics-must-know-methods.
5. Rieck, K. (2014). Malheur. Retrieved February 27, 2022, from http://www.mlsec.org/malheur/manual.html