The MP13 Approach to the KDD'99 Classifier Learning Contest

Miheev Vladimir
MP13
Russia, 103575, Moscow, 904-56
Miheev@mp13.msk.ru

Vopilov Alexei
IT Security Consultant
Russia, 103498, Moscow, Zelenograd
MIET, Technopark, off. 7307
AlexeiV@email.com

Shabalin Ivan
Senior Software Eng.,
DisCo Company,
Russia, Moscow
Shabalin@disco.ru

ABSTRACT

The MP13 method is best summarized as recognition based on voting decision trees using "pipes" in potential space.

Keywords

Voting; Decision Tree; Potential Space

1. INTRODUCTION

The approach employed by «MP13» team (http://mp13.msk.ru) works towards the idea of so-called 'Partner Systems'. It is aimed on effective data analisys and particular problem resolution based on intrinsic formalization of an expert knowledge that is intuitive by nature and hence cannot be extracted in compact form.

Concerning to particular contest task, the final decision on a connection belonging to one of the five classes is made according to the following steps:

- Verbal rules constructed by an expert proficient in network security technology and familiar with KDD methods
- First echelon of voting decision trees
- Second echelon of voting decision trees

Steps are applied sequentially. A branch to the next step occurs whenever the current one has failed to recognize particular connection. This sequencing was not assumed in advance; it became obvious during 'training' stage of the contest.

2. KDD'99 WORK DETAILS

In a preliminary stage, thirteen decision trees were generated based on a subset of the training data. The target was to identify "normal" connections and any attacks. The training dataset was randomly split into three subsamples: 25% for tree generation, 25% for tree tuning, and 50% for estimating model quality. The results of this experiment were quite encouraging.

We did not have enough hardware capacity to work with all the learning samples. Hence, we prepared our own learning data set as 10% of the given complete training database (about 400,000 entries). Since the task description pointed out the fact of different distributions in the training and testing datasets, we randomly removed some of the DOS and "normal" connections from the full training database. The resulting reduced dataset was used for all further manipulations.

Next, we proceeded with learning based on the "one against the rest"' principle. Classes were separated from each other using five sets of decision trees. Each connection was assigned a vector of proximity to the five classes. Each vector component was calculated as a sum of weighted votes from specific set of decision trees. This representation forms so-called "potential space" while a multidimensional interval on proximity vectors represents a "pipe."

Next, we converted the testing dataset into 'potential space' representation. We obtained two types of connections in this space: well recognized and not recognized. Some more work has been done to improve recognition. First, expert knowledge made it possible to construct some simple but quite stable verbal rules. Also, a second echelon (see Section 1) of decision trees was constructed to separate unrecognized pairs of classes.

3. TRAINING ALGORITHM

To construct our recognition model, we used a version of the 'Fragment' algorithm originally invented at the IITP (Russian Academy of Science), in the division 'Partner Systems'. The authors of "Fragment" are V. Miheev and V. Pereversev-Orlov.

For constructing a decision tree, training dataset is split into learning and testing samples. The learning sample is used to find the structure of a tree (with sufficient complexity) and to generate a hierarchy of models on this tree. The testing sample is used to select a sub-tree having optimal complexity.

Repeated application of the algorithm to various splits of the training data in the different subspaces of the initial data description allows us to generate a set of voting decision trees. Some heuristic approaches are used to make the trees independent.

4. HARDWARE RESOURCES

We used two computers: a Pentium Notebook 150 MHz with 32 MB RAM and about 200MB free disk space, and a Pentium Desktop 133 MHz with 40 MB RAM and about 200MB free disk space. It took about 30 minute to generate one set of trees. The total time for all calculations using the PERGAMENT software was about 6 hours. Additionally, we used DOS FoxPro(TM) package to perform various data transformations and interactive data analyses

5. ACKNOWLEDGMENTS

Special thanks to V. Pereversev-Orlov as forever guru in KDD science and original inventor of the approach used.

6. REFERENCES

[1] Pereverzev-Orlov V.S. Models and methods for automatic reading. Moscow, Nauka, 1976.

[2] Pereverzev-Orlov V.S. Professional advisor: experience in partner system development. Moscow, Nauka, 1990.

[3] Pereverzev-Orlov. A Partner System and the Concept of Recognition Learning. Pattern Recognition and Image Analysis. Vol. 2, No. 4, 1992, pp. 420-437.

[4] A.V.Genkin, V.A.Mikheev. The synthesis of voting collectives of regression trees. PRIA 1997, v.7, #2, pp.184-191.

About the authors:

Miheev Vladimir is author of 20 publications and a number of algorithms in the Data Mining area. He earned Ph.D. from Institute of Information Transmission Problems, Russian Academy of Science. Currently he is the head of laboratory in "Scientific Research Center", Zelenograd.

Vopilov Alexei is independent consultant specializing on custom projects development. His interests lie in IT security, especially Networking Protocols, PKI, Policy Management. He has gotten some contribution to Data Mining based on Syndrome Analysis.

Ivan R. Shabalin is senior software engineer, author of the programs such as Kaissa-PC, DISCo Commander http://www.disco.ru), Lexicon 4.0 for Windows, DISCo Pump and many others.