Feature Selection in Intrusion Detection: a simple and partial proposal

Feature Selection in Intrusion Detection: a simple and partial proposal

Adrian Pino Angulo　Kilho Shin

兵庫県立大学応用情報科学研究科
Graduate School of Applied Informatics, University of Hyogo

Abstract: It was very recent that a formal definition of multiple alignments is given to general data structures that include not only strings but also trees and graphs. Also, it has been shown that the center star method, which is to compute approximately optimal multiple alignments for strings, is effective for the generalized multiple alignments. In this paper, we report the results of our experiments to prove effectiveness of the generalized multiple alignments and the extended center star method by taking trees as an example.

1 Introduction

Since the emergence of Internet in the early 1980s, we all have experienced an incremental access to a wide range of information resources and services. Nowadays, with billions of interconnected electronic devices, network services are extensible to e-commerce, bank account management, data transmission for military communication and others. Unfortunately, science has been also used to attack and destroy network systems in order to have some benefits. Follows from the emergence of the first malware at the end of 80s, firewalls were created to protect all the valuable data generated by network services. Firewalls are able to confront external intrusions, but they are susceptible to be cheated from internal attacks.

In contrast, Intruder Detection Systems (IDSs) constantly monitor the state of a network and trigger an alarm if a suspicious event take place, regardless its origin. There are two basic classifications of IDSs: host-based and network-based. The former aims to protect an individual computer system by monitoring its inputted and outputted data while the latter collect and analyze raw network packets to look for suspicious events in the whole computer network. Both approaches can use different methods to inspect the behavior of the network: misuse detection and anomaly detection. The misuse detection method consists on monitoring packets in the network and compare them with stored attack patterns known as signatures. Therefore, this method can reliably identify attacks that have been found previously, with a very low false positive rate. However, intruders are heterogeneous and generally evolve faster than IDSs. Consequently, IDSs that relies on misuse detection method are often unsecure.

Anomaly-based systems adaptively improve their performance by discovering intrusion patterns and knowledge from the observed data. In this approach the system builds classification models for normal behavior in the network and when an event is not matched by the model then is classified as an attack. Because of the dynamic nature of the network, this approach often suffer from high false positive rates.

A wide range of machine learning algorithms have been proposed to improve the performance of this kind of systems. In particular, supervised classification plays an essential role to efficiently improve the performance of IDSs. Let $x = [x_1, \ldots, x_n]$ be a connection characterized by m properties collected during its life time in the network. Let c be the classification of the connection x given by experts after analyzing the behavior of x (e.g. normal, abnormal). Provided a set of pairs $D = \langle x, c \rangle$, stored from the past experience, the task of supervised learning is to infer a function $\ell_D : x \to c$ built from D such that future

connections can be accurately predicted.

In the literature a vast number of supervised classifiers have been proposed for intruder detection, just to name a few: Decision rule-based, Random Forest and Support Vector Machine. However, when the collected data is high-dimensional and comes with irrelevant and redundant information, the performance of the classifiers can be affected in three directions: high complexity in the inferred functions in terms of number of bytes to represent it, large running time to discover ℓ and detriment in the accuracy when classifying potential intruders. Feature selection aims to turn aside all these three problems by detecting and removing irrelevant and redundant features.

Being $F = \{f_1, \ldots, f_n\}$ the set of features collected for each connection with $x_i \in f_i$, and denoting $D_{\tilde{F}}$ a new dataset resulted from projecting $\tilde{F}$ over D, the task of feature selection is to select a feature set $\tilde{F} \subset F$ such that $\ell_{D_{\tilde{F}}}$ is at least as accurate as ℓ_D. Feature selection algorithms fall into three basic groups: filter, wrapper and hybrid. On the one hand, wrappers use a classifier as a black box to evaluate the goodness of a candidate set. Since the classifier must be trained and tested every time the evaluation of a subset is required, this approach is high-time consuming. On the other hand, filter algorithms are faster because they rely on the structure and distribution of data to discriminate between predictive and non predictive features. However, speaking about accuracy we can expect better results when a wrapper is used instead. Hybrid approaches can be fast and accurate because they combine both of wrapper and filter approaches.

In addition, according to the evaluation function, feature selection algorithms can be divided into three categories: univariate, pairwise and multivariate. The univariate algorithms evaluate how good is a feature to predict whether or not a connection is a potential intruder by computing any of the correlation, distance, uncertainty or dependency $r(f; c)$ between the feature f and the class variable c. The most basic search is to select features $f \in F$ such that $r(f; c) > \delta$ where δ is a predefined value. This approach is quite simple and fast, but does not deal with redundancy between features, which can lead to a significant deterioration of the accuracy of classifiers.

The pairwise approach solve this problem by removing features high-correlated with others. Although, efficiently removing redundant features requires quadratic operation in terms of the number of features, this approach is still scalable for high-dimensional datasets

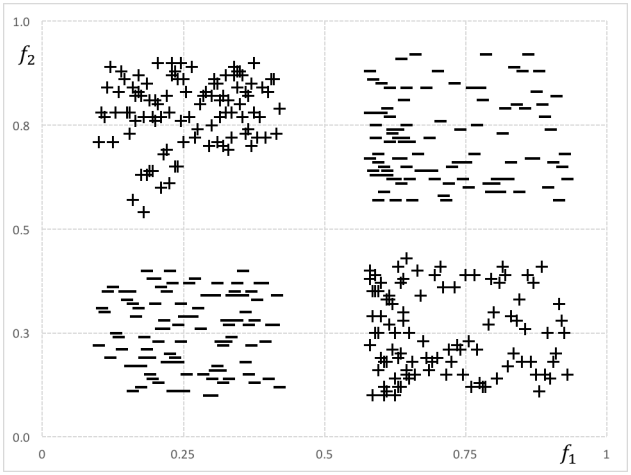

図 1: Non relevant features that interact with each other to accurately discriminate between + and - classes.[update label in the axes]

in IDSs. Since the evaluation of subsets lies on individual and pairwise evaluations, these approaches offer no guaranties to detect complex relation among features to accurately detect network intruders.

Consider the example depicted in Figure 2.2, where $F = \{f_1, \ldots, f_n\}$ is the IDS dataset and an intruder can be accurately discovered by the combination of ... (f_1) and ... (f_2) such that $c = (f_1 < a \wedge f_2 > b) \vee (f_1 > a \wedge f_2 < b)$. It is clear that neither of the approaches aforementioned select $\{f_1, f_2\}$ because both features are totally uncorrelated with the class variable and both are redundant.

The multivariate approach is characterized by evaluating a set of features as a whole. Therefore, they are able to detect complex relations among features.

The main motivation in this paper is to combine consistency-based measures with the wrapper approach to build a hybrid feature selection algorithm that increase the efficiency and accuracy of IDSs.

2 Proposal

Hybrid feature selection algorithms are often preferred because they are in between the efficiency of filters and the accuracy of wrapper algorithms. An efficient plan to remain close to both sides is to first apply a filter to remove useless features and then perform a wrapper search over the reduced set. Consistency-based algorithms are very accurate and to the best of

our knowledge SUPER-LCC and SUPER-CWC are the top algorithms within this group. In order to narrow down the search space to efficiently apply a wrapper, we have selected SUPER-LCC. In the remainder of this section we briefly describe the basic properties of consistency measures and SUPER-LCC algorithm. Finally, we describe how a classifier can improve the outputted set of SUPER-LCC by discovering the most suitable value of δ (to be defined).

2.1 Consistency measures and linear backward algorithms

Consistency-based algorithms can be a good choice to narrow the search space because they detect redundancy and interaction among features.

Definition 1. *Given a consistency function $\mu : \langle \tilde{F}, C\rangle \to \mathbb{R}$ that estimates the error prediction of a feature set $\tilde{F}$ and provided that $\mu(\tilde{F}; C) = \delta$ and $\mu(\tilde{G}; C) = \delta'$, if $\mu(\tilde{F} \cup \tilde{G}; C) \lneq \min\{\delta, \delta'\}$ holds then both $\tilde{F}$ and $\tilde{G}$ interact with each other.*

Definition 1 suggests consistency-based measures are able to detect the relation existing between f_1 and f_2 in the example described in section 1, which can significantly improve the accuracy of classifiers. Speaking about efficiency, the most important properties of consistency-based measure is that they are monotonic and are characterized by being determinant.

Determinacy. $\mu(\tilde{F}; C) = 0$ holds if there exists any function $\ell_{D_{\tilde{F}}}$ that can accurately predict all connections in $D_{\tilde{F}}$.

Monotonicity. $\mu(\tilde{F}; C) \leq \mu(\tilde{G}; C)$ holds for $\tilde{G} \subset \tilde{F}$

The monotonicity property allows detecting the minimum set that optimize the function when a backward elimination search is performed in conjunction with, while the determinacy property avoid the algorithm being trapped by local optima by skipping blind alleys.

The algorithm of INTERACT was an important breakthrough in the consistency-based feature selection field[ref.]. INTERACT find very accurate sets by a two step procedure. First, all features are incrementally ranked according to their individual relevance. That is $\tilde{F} = \{f_1, \cdots, f_n\}$ such that $SU(f_i; C) < SU(f_j; C)$ for all $i < j$ and $j \leq n$. Second, a linear backward elimination search is performed over $\tilde{F}$ by removing features f_i starting from the less relevant, such that $\mu(\tilde{F} \setminus \{f_i\}; C) - \mu(\tilde{F}; C) < \delta$ holds. However, Shin and Xu in [ref.] demonstrate that INTERACT can eliminate too many features when the interaction among features in F is very low. When this happens the performance of INTERACT is very poor. The algorithm LCC solves this problem by removing features only when $\mu(\tilde{F} \setminus \{f_i\}; C) < \delta$ holds[ref.]. This function constrains the outputted set to have an inconsistency rate smaller than δ. The efficient of *Lcc* was improved by conducting binary search instead of linear search. This idea was materialized under the name of SUPER-LCC and works under the assumption that high-dimensional datasets are abundant in irrelevant features that can be removed in mass . Therefore, the set $\{f_{l_i}, \ldots, f_{l_{i-1}}\}$ can be removed in the $i-$th iteration with

$$l_i = \underset{j=l_{i-1}+1,\ldots,n}{\operatorname{argmax}} \{\mu(\tilde{F} \setminus \{f_{l_{i-1}+1}, \ldots, f_j\}) \leq \delta\}.$$

SUPER-LCC output the same set as LCC but on average has a computational complexity of $O(nm(\log n + \log m))$ where n is the number of features that describes the m connections in D.

2.2 Proposed algorithm

SUPER-LCC is a very fast algorithm that output high-quality sets. However, the parameter δ that acts as an inconsistency rate constrainer has never been further analyzed. The main motivation of this paper lies on the hypothesis that most of the time, for a given dataset, there is an optimal value of δ (different from the one fixed before running the algorithm) that yields a substantial increment in the accuracy of classifiers. The authors of SUPER-LCC state that the bigger the value of δ is, the smaller the size of the solution is. However, in this paper we analyze some other incidences of the variation of the value of δ to empirically demonstrate that the reducing the value of δ not necessary yields to high accuracy of classifiers.

Given the entire feature set $F = \{f_1, \ldots, f_n\}$ of a dataset D, we first define the ideal set as follows.

Definition 2. *If $F^* \in \{\operatorname{argmax} \overline{R}_r(\tilde{G}; C) \mid \mu(\tilde{G}; C) = \mu(F; C)$ and $\tilde{G}$ is minimal in size$\}$ holds, then F^* is said to be an ideal set in D.*

The ideal set F^* in D not only is as consistent as the entire feature set F, but also is the minimal set that maximize the collective relevance. Therefore, in many cases we can characterize F^* as a consistent set

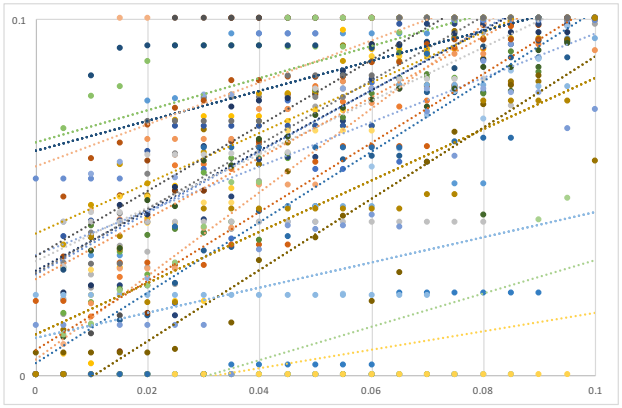

図 2: Relation between δ and $\overline{R}$ computed in x datasets.

composed by relevant features. Although the output of Super-Lcc is not exactly an ideal set, it is not far from being a good approximation.

Proposition 1. *Given an ordered set* $F = \{f_1, \ldots, f_n\}$, *with* $r(f_i; C) < r(f_j; C)$ *for all* $i < j$, *running a linear backward search over* F *by repeatedly removing* f_i *from* F *such that* $\mu(F \setminus \{f_i\}) > \mu(F; C)$; *exactly outputs*

$F' \in \{\text{argmax}\{\min r(f_i; C) \mid f_i \in \tilde{G}, \mu(\tilde{G}; C) = \mu(\tilde{G}; C)\}\}$

Proposition 1 suggests that Lcc/Super-Lcc outputs the consistent set with the less relevant feature closest to f_1. Therefore, we can expect that $\overline{R}(F'; C) \approx \overline{R}(F*; C)$ holds in many cases. However, we need a mechanism to find a set F'^* in between F' and F^* with an optimal balance between $\overline{R}$ and μ. Let $F = \{f_1, \ldots, f_n\}$ be the entire feature set ordered by some correlation measure r, and let $F^i = \{f_i, \ldots, f_1\}$ be a set in the same order as F, with features from f_i to f_1.

Proposition 2. *Provided that* $F_{\delta_1} = \{f_t, ...\}$ *is the minimal set found by* Super-Lcc *with an arbitrary value of* δ_1, *where* f_t *is the first feature selected by* Super-Lcc, *that is* $f_t \in \{\text{argmin}_{f \in F_\delta} r(f)\}$. $F_{\delta_2} \subset F^t$ *for* $\delta_2 > \delta_1$ *always holds.*

Proof. This can be easily derived from proposition 1 and the monotonicity property. □

By proposition1 we can expect that bigger the value of δ is, the outputted set of Super-Lcc F_δ is not only smaller, but the features are closer to f_1. Therefore, the outputted set of Super-Lcc with large value of δ are likely to be larger in terms of $\overline{R}$ than with small δ. Having as extreme examples $\delta = \mu(F; C)$ and $\delta = \mu(\{f_1\}; C)$. Under this assumption we can now evaluate F_δ in two dimensions: consistency rate μ and collective relevance $\overline{R}$. The task is now reduced to iteratively find δ^* that allows outputting F_{δ^*} that minimize the probability that a new connection is misclassified That is,

$$F_{\delta^*} \in \{\text{argmin}\, Pr[\ell_{D_{F_{\delta'}}}(x) \neq c]\},$$

were c is the real answer for the connection x and the probability of error can be computed through cross validating in $D_{F_{\delta'}}$ a given classifier. Now two critical issues must be faced. First, the $\delta-$ decrement problem and the stopping criterion decision. The first issue is referred to the problem of finding the most suitable value of δ_i in the $i-$th iteration that allows finding F_{δ^*} as soon as possible. A large value of δ_i can supply a set with high collective relevance, but with an inconsistency rate upper-bounded by δ_i, while with a very small value of δ_i the collective relevance may not be sufficiently large. Both cases can equally affect the performance of the classifier in terms of accuracy. By the moment we fix a parameter π to a small value and run Super-Lcc with $\delta_i = i\pi$ in the $i-$th iteration until $\delta_i > \delta$ holds.

The second issue consists on deciding the value of δ that defines how many times the algorithm will iterate. For simplicity we just fix a value $p \in \mathbb{N}$ that represents the allowable number of times the error of the classifier do not decrease when changing δ. However, in the future we will use a more sophisticated function based on simulated annealing. Finally, given a dataset D, a classifier ℓ, the decrement value π and p: the allowable number of of times the classifier do not decrease the error rate when changing δ, the proposed algorithm is as follows:

1. Rank all features in increasing order according to a measure $r(f_i; C)$, that is $\tilde{F} = \{f_1, \ldots, f_n\}$ with $r(f_i; C) < r(f_j; C)$ for $i < j$ and $j \leq n$.
2. Run Super-Lcc over $\tilde{F}$ with $\delta_{iter} = iter\pi$ and let $F_{\delta_{iter}} = \{f_{l_{iter}}, \ldots\}$ be the outputted set.
3. If $error(\ell_{F_{\delta_{iter}}}; C) \leq error(\ell_{F_{\delta_{iter-1}}}; C)$ then update $F_{\delta^*} = F_{\delta_{iter}}$ and $p = 0$. Otherwise, update p' = p'+1 and if p'=p holds then terminates and return F_{δ^*}.
4. Update $\tilde{F} = F_{\delta_{iter}}$ and repeat from step 2 until $|F_{\delta_{iter}}| < 1$ holds. We also have implemented the algorithm with an additional option named *accumulative*. If this option is activated then in step 3 F_{δ^*} is updated such that $F_{\delta^*} = F_{\delta^*} \cup F_{\delta_{iter}}$. In this way the inconsistency rate of the selected set *remains* equals

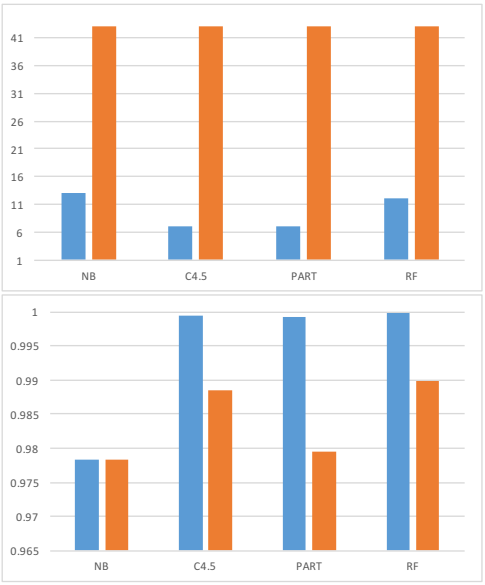

図 3: Comparison on the number of features and AUC values between the proposed algorithm and the entire data. Blue bars corresponds to the proposed algorithm and red bars to the entire feature set.

to $\mu(F;C)$ while its collective relevance $\overline{R}$ must increase in each iteration.

We have run experiments in the KDDCup99 dataset, which is commonly used for the Intruder detection problem in data mining. We run a 10-fold cross validation procedure by using the proposed algorithm and compared with the result of the classifier trained in the entire dataset. Naive Bayes, C4.5, Part and Random Forest where used in the experiments.

As can bee seen our proposal not only drastically reduce the dataset, but also the AUC value is improved in three of the classifiers.

3 Conclusion

In this paper, we have proposed a new algorithm based on consistency-measures that use a classifier to find the optimal value of the δ parameter. Experiments in the KDDCup99 dataset reveal our approach not only reduce more than 60 percentage of the features, but also the accuracy of the classifiers improve.