Feature selection for intrusion detection

Slobodan Petrović
NISlab, Gjøvik University College
Contents

• The feature selection problem
• Intrusion detection
• Traffic features relevant for IDS
  – The CFS measure
  – The mRMR measure
  – Solving the optimization problems
The feature selection problem

• The curse of dimensionality
  – 2 close points in a 2-dimensional space are probably distant in a 100-dimensional space

• Any machine learning algorithm
  – Makes a prediction of unseen data points by a hypothesis constructed from a limited number of training instances

• In high-dimensional space - difficult
The feature selection problem

• Hypothesis (in this setting)
  – A pattern or function that predicts classes based on given data

• Hypothesis space
  – Contains all the hypotheses that can be learned from data
The feature selection problem

- A linear increase in the number of features (i.e. the dimension of the feature space) leads to the exponential increase of the hypothesis space
  - Example: 2 classes, $N$ binary features
    - The cardinality of the hypothesis space: $2^N$
The feature selection problem

• Feature selection
  – Removes irrelevant features
  – Removes redundant features

• Consequence
  – Efficient reduction of the hypothesis space
  – Easier to find the correct hypothesis
  – Reduced number of required training instances (the reduction is exponential)
The feature selection problem

• Removing irrelevant features
  – Does not affect learning performance

• Removing redundant features
  – Redundant features – a type of irrelevant features
  – The difference: a redundant feature requires co-presence of another feature
  – Each individual feature is relevant, but removal of one of them will not affect learning performance
The feature selection problem

• 2 types of feature selection methods
  – Feature ranking
    • Rank features according to some criterion and select the top $k$ features
    • A threshold is needed in advance to select the top $k$ features
  – Feature subset selection
    • Selects the minimum subset of features that does not deteriorate learning performance
    • No threshold necessary
The feature selection problem

• Models of feature selection
  – The filter model
    • Considers statistical properties of a data set directly
    • No learning algorithm involved
    • Efficient
  – The wrapper model
    • Performance of a given learning algorithm is used to determine the quality of selected features
Intrusion detection

• Intrusion
  – Activities aimed at violating security (i.e. confidentiality, integrity and availability of computer and network resources)

• Intrusion detection
  – Process of detection and identification of attacks

• Intrusion prevention
  – Process of attack detection and defense management
Intrusion detection

• Intrusion detection system - IDS
  – A system that automatically detects attacks against hosts and networks

• Intrusion prevention system - IPS
  – A system, whose ambition is to detect attacks and manage defence activities
  – An IPS contains an IDS
    • IPS combine IDS with other preventive measures (firewall, anti-virus, vulnerability scanning, etc.)
Intrusion detection

• IDS classification
  – According to the protected object
    • Host-based IDS
    • Network-based IDS
  – According to the detection model
    • Misuse detection IDS
    • Anomaly detection IDS
Intrusion detection

• Host-based IDS
  – Collect data from internal sources, usually at the operating system level (various logs)
  – Monitor user activities
  – Monitor execution of system programs
Intrusion detection

• Network-based IDS
  – Collect packets, usually by means of network interfaces in so-called "promiscuous" mode (such a device collects all the packets that reach the interface, not only those addressed to the host)
  – Perform analysis of the collected packets
  – Monitor network activity
Intrusion detection

• Misuse detection systems
  – Collect information about attack indicators and then determine whether those indicators are present in incoming data

Attack indicators (signatures) → Analysis (e.g. pattern matching) → Attack

Activities
Intrusion detection

- Anomaly detection systems
  - Define profiles of normal behaviour of users or networks, compare actual behaviour with those profiles and generate alerts if the discrepancy from the profiles is too high

Profiles of normal behaviour → Analysis → Attack

Activities
Intrusion detection

Data pre-processor

Activity data

Detection algorithm

Alerts

Alert filter

Decision criteria

Action/report

Incoming traffic/logs
Intrusion detection

• What properties should the data pre-processor possess?
• Which detection model is optimal?
• What is the best detection algorithm?
• What are the optimal decision criteria?
• What alert filter gives the best results?
Intrusion detection

• Until recently, the answers to those questions were heuristic – intrusion detection was a more technical discipline, without clear theoretical foundation

• After 2005, some theoretical models of IDS appeared
  – Models based on complexity theory
  – Information-theoretic models
Intrusion detection

• IDS model (1)
  – IDS is an 8-tuple \((\mathbb{D}, \Sigma, \mathcal{F}, \mathcal{K}, \mathcal{S}, \mathcal{R}, \mathcal{P}, \mathcal{C})\)
  – The first 4 components are data structures
    • Data source \(\mathbb{D}\)
    • The set of data states \(\Sigma\)
    • The set of data unit features \(\mathcal{F}\)
    • Knowledge base about data profiles \(\mathcal{K}\)
Intrusion detection

• IDS model (2)
  – The second 4 components are algorithms
    • Algorithm for feature selection $S$
    • Algorithm for reduction and representation $R$
    • Knowledge base generator $P$
    • Classification algorithm $C$
Intrusion detection

• Data source \( \mathbb{D} \)
  
  – A flow of consecutive data units (packets, data flow units, system calls)
  
  – \( \mathbb{D} = (D_1, D_2, \ldots) \), where \( D_i \) is the analyzed data unit, \( D_i \in \{d_1, d_2, \ldots\} \), \( d_j \) is a possible data unit
  
  – In network-based IDS, \( \mathbb{D} \) is a stream of packets \( P=(P_1, P_2, \ldots) \)
  
  – In host-based IDS, \( \mathbb{D} \) can be a stream of system calls \( C=(C_1, C_2, \ldots) \)
Intrusion detection

• The set of data states $\Sigma$
  – Contains normality indicators for each $D_i$
  – If $D_i$ is abnormal, it is possible that the corresponding indicator from $\Sigma$ also contains the type of the attack
  – In anomaly detection, $\Sigma=$\{normal, abnormal\} or $\Sigma=$\{N,A\} or $\Sigma=$\{0,1\}
  – In misuse detection, $\Sigma=$\{normal, attack type 1, attack type 2, ...\} or $\Sigma=$\{N,A_1,A_2,...\}
Intrusion detection

• The set of data unit features $F$
  – A vector of features that contains a finite number of attributes of a data unit, $F=(f_1, f_2, \ldots, f_n)$
  – Examples: protocol name, port number, etc.
  – Every feature has its domain $R$
    • A set of discrete or continuous values
Intrusion detection

• Knowledge base about data profiles $\mathcal{K}$
  – Contains profiles of normal and abnormal data units
  – Internal structure of the base $\mathcal{K}$ is different for each IDS (a tree, a Markov model, a Petri net, a set of rules, a base of attack signatures, etc.)
  – In misuse-based systems, $\mathcal{K}$ is a set of rules that describe attack profiles (i.e. attack signatures)
  – In anomaly detection systems, $\mathcal{K}$ is a profile of normal traffic
Intrusion detection

- An ideal data unit tester $\text{Oracle}_{\text{IDS}}$
  - Performs analysis of each data unit $D_i$
  - Gives the indicator value at the output
    - Normal
    - Abnormal
  - Always gives the correct value of the indicator
  - For each $D_i$, its state is $\text{Oracle}_{\text{IDS}}(D_i)$
Intrusion detection

• Algorithm for feature selection $\mathcal{S}$
  – Given $\mathcal{D}$ and the corresponding states from $\Sigma$, the algorithm $\mathcal{S}$ gives certain number of features that IDS will measure and decide on them
  – In general, $\mathcal{S}$ depends very much on the knowledge about the attack characteristics
  – The quality of the selected features mainly determines the effectiveness of the IDS
Intrusion detection

• Feature selection

\[ \mathcal{D} \quad D_3 \quad D_2 \quad D_1 \]

\[ \text{Algorithm for feature selection } \mathcal{S} \]

\[ \mathcal{F} \quad f_1 \quad f_2 \quad \ldots \quad f_n \]

\[ \text{Oracle}_{\text{IDS}}(\mathcal{D}) \quad \ldots \quad A \quad N \quad N \]
Intrusion detection

• Algorithm for reduction and representation $\mathcal{R}$
  – During data processing, IDS first performs data reduction, i.e. extraction of characteristics that are the results of the execution of the algorithm $\mathcal{S}$, and then their representation in the form of a vector with coordinates in $\mathbb{F}$
  – Thus, $\mathcal{R} : \mathbb{D} \rightarrow \mathbb{F}$
Intrusion detection

- Knowledge base generator $\mathcal{P}$
  - To generate the knowledge base, we need an algorithm that, based on the vectorial data representations and their states, generates the knowledge base $\mathcal{K}$
Intrusion detection

• Knowledge base generation

\[ \mathcal{D}' \]

\[ \text{... } D_3 | D_2 | D_1 \]

Representation algorithm \( \mathcal{R} \)

Feature vectors

\[ \{ f_1', f_2', \ldots \} \]

\[ \{ f_1'', f_2'', \ldots \} \]

\[ \ldots \]

Oracle_{IDS}(\( \mathcal{D}' \))

\[ \text{... } A | N | N \]

Knowledge base generator \( \mathcal{P} \)

Knowledge base \( \mathcal{K} \)
Intrusion detection

• Classification algorithm $\mathcal{C}$
  
  – That is a function that maps the representation of the given data unit into some state, based on the knowledge base $\mathcal{K}$

  – Formally, $\mathcal{C} : \mathcal{F} \rightarrow \Sigma$
Intrusion detection

- Detection procedure (classification)
Intrusion detection

• Phases of operation of an IDS (1)
  – Feature selection
    • In general, this phase is executed only once, during the development of the IDS
  – Knowledge base generation
    • Sometimes called the training procedure
    • The algorithm $\mathcal{P}$ (with the help of the algorithm $\mathcal{R}$) is executed over a large quantity of training data
    • In general executed once, but the base $\mathcal{K}$ may occasionally be updated
Intrusion detection

• Phases of operation of an IDS (2)
  – Detection procedure
    • IDS is applied over real data in order to detect attacks
    • The most important and most often used phase
Traffic features relevant for IDS

• The goal of the feature selection algorithm in an IDS
  – To determine the most relevant features of the incoming traffic, whose monitoring ensures reliable detection of abnormal behavior

• Effectiveness of the classification heavily depends on the number of features
  – It is necessary to minimize that number, without dropping indicators of abnormal behavior
Traffic features relevant for IDS

• In the contemporary IDS
  – The most of work on feature selection is still done manually
  – The feature selection depends too much on expert knowledge – unreliable
  – Better algorithms for automatic feature selection in IDS are needed
Traffic features relevant for IDS

• For IDS
  – Due to high-dimensional data, the filter model is more appropriate for automatic feature selection
  – To eliminate redundant features, the feature-subset-evaluating method seems to be better than the feature ranking method

• A generic feature selection measure is defined first and then the methods to maximize it are found
Traffic features relevant for IDS

• The generic feature selection measure (*)

\[ GeFS(X) = \frac{a_0 + \sum_{i=1}^{n} A_i(X)x_i}{b_0 + \sum_{i=1}^{n} B_i(X)x_i}, \quad X = (x_1, \ldots, x_n) \in \{0,1\}^n \]

\( x_i = 1 \) indicates appearance of the feature \( f_i \)

\( a_0 \) and \( b_0 \) are constants

\( A_i(X) \) and \( B_i(X) \) are linear functions

• The feature selection problem

  – Find \( X \in \{0,1\}^n \) that maximizes \( GeFS(X) \)
Traffic features relevant for IDS

• Several feature selection measures representable in the form (*)
  – The Correlation-Feature-Selection (CFS) measure
  – The minimal-Redundancy-Maximal-Relevance (mRMR) measure
  – Etc.
The CFS measure

• The merit function of a feature subset $S$ consisting of $k$ features

$$\text{Merit}_S(k) = \frac{k \overline{r_{fc}}}{\sqrt{k + k(k - 1) \overline{r_{ff}}}}$$

where $\overline{r_{fc}}$ is the average value of all feature-classification correlations and $\overline{r_{ff}}$ is the average value of all feature-feature correlations
The CFS measure

• The merit function reflects the following intuitive hypothesis about quality of a feature subset

  – Good feature subsets contain features highly correlated with the classification, yet uncorrelated to each other

• The merit function is maximized in the CFS measure

$$\max_s \left\{ \text{Merit}_S(k), 1 \leq k \leq n \right\}$$
The CFS measure

- It can be shown that the problem of maximization of the merit function can be presented as an instance of the GeFS measure ($GeFS_{CFS}$)

$$\max_x \left\{ \frac{\left( \sum_{i=1}^{n} a_i x_i \right)^2}{\sum_{i=1}^{n} x_i + \sum_{i \neq j}^{n} 2b_{ij} x_i x_j} \right\}$$
The mRMR measure

- Based on mutual information
- The relevance of features and the redundancy between features are considered simultaneously
The mRMR measure

• The relevance of a feature set $S$ for the class $c$

$$D(S,c) = \frac{1}{|S|} \sum_{f_i \in S} I(f_i, c)$$

• The redundancy between features in $S$

$$R(S) = \frac{1}{|S|^2} \sum_{f_i, f_j \in S} I(f_i, f_j)$$
The mRMR measure

• Combing the relevance and redundancy measures, we get the mRMR measure, which is to be maximized

$$\max_{S} \left\{ \frac{1}{|S|} \sum_{f_i \in S} I(f_i, c) - \frac{1}{|S|^2} \sum_{f_i, f_j \in S} I(f_i, f_j) \right\}$$
The mRMR measure

• It can be shown that the problem of maximization of the mRMR measure can also be presented as an instance of the GeFS measure \((\text{GeFS}_{m\text{RMR}})\)

\[
\max_{x} \left\{ \frac{\sum_{i=1}^{n} c_i x_i}{\sum_{i=1}^{n} x_i} - \frac{\sum_{i,j=1}^{n} a_{ij} x_i x_j}{\left(\sum_{i=1}^{n} x_i\right)^2} \right\}
\]
Solving the optimization problems

- The problems of maximizing $GeFS_{CFS}$ and $GeFS_{mRMR}$ can be solved if we analyze them as problems of fractional programming
  - In particular, these problems pertain to the category of Polynomial Mixed 0-1 Fractional Programming problems (PM01FP)
Solving the optimization problems

• The general form of \( PM01FP \)

\[
\min \sum_{i=1}^{m} \left( \frac{a_i + \sum_{j=1}^{n} a_{ij} \prod_{k \in J} x_k}{b_i + \sum_{j=1}^{n} b_{ij} \prod_{k \in J} x_k} \right)
\]

under the following constraints

\[
b_i + \sum_{j=1}^{n} b_{ij} \prod_{k \in J} x_k > 0, i = 1, \ldots, m
\]

\[
c_p + \sum_{j=1}^{n} c_{pj} \prod_{k \in J} x_k \leq 0, p = 1, \ldots, m
\]

\[
x_k \in \{0,1\}, k \in J
\]

\[
a_i, b_i, c_p, a_{ij}, b_{ij}, c_{pj} \in \mathbb{R}
\]
Solving the optimization problems

• By introducing appropriate substitutions, such a PM01FP can be transformed into a Mixed 0-1 Linear Programming Problem (M01LP)
  – M01LP can be solved by means of the branch-and-bound method
  – A globally optimal solution is obtained
  – The number of variables and constraints in the M01LP is linear in the number \( n \) of full-set features