Network Traffic Classification Using Rough Sets

LAAMANEN, VESA; LAURIKKALA, MIKKO; KOIVISTO, HANNU
Institute of Automation and Control
Tampere University of Technology
P.O.Box 692, 33101 Tampere
FINLAND
firstname.lastname@tut.fi http://www.ad.tut.fi/aci/

Abstract: This paper studies the theory of rough sets in order to classify network data. First, some necessary basic concepts of rough sets are introduced. Methods needed for generating rules from a data base are reviewed. The theory is then utilized in a test case. The data used in the classification was recorded from a test network running different network applications. Preprocessing of the data, rule generation and validating classification are described. The target was to classify the traffic by applications, and in the test case 97,5 % of the data was classified correctly. Some useful applications of the results as well as future prospects are also discussed.

Keywords: Rough sets, traffic analysis, network analysis, classification, clustering, NetFlow

1 Introduction

During the recent years, Internet traffic volume has multiplied and numerous new applications are used through networks. This makes demands on security and on the whole serviceability of networks. Thereby it is important to know how the applications act, for example, which application generates most traffic in a certain network. Different types of data collectors recording network traffic have been developed. However, the amount of data is enormous and it is impossible to find regularities visually.

This paper introduces a way of classifying applications using rough sets. Applications have been classified earlier with other methods [1] but the data format has been different. In this paper, data collected from a router with NetFlow [2] is used. The data includes information about traffic going through the router, such as protocol, bytes and number of packets. Due to the format of data, that is, the data includes both quantitative and qualitative variables, rough sets conform to this problem well.

The next chapter introduces basic concepts of rough sets theory necessary for understanding the procedure. Chapter 3 details the data collection routine and also focuses the target of this research. Chapter 4 is an exposition of the work itself including tools, data preprocessing and rule generation. In chapter 5, main results are presented while chapter 6 discusses the importance of the results and also future prospects.

2 Theoretical background

Many clustering or classification methods use continuous values with meaningful measures between values. Consequently, these methods often have difficulties with qualitative values because no metric exists between the values.

The method of data base analysis presented here is based on the rough sets theory and was introduced by Pawlak [3] in the early 1980's. It deals with the classificatory analysis of data tables. All variables are expressed as qualitative values, which in turn requires discretisation of quantitative variables.

2.1 Information system

In rough sets theory, a data set is represented as a table called an *information system*. It is a pair $\mathcal{A} = (U,A)$, where U is a non-empty finite set of *objects*, called *universe*, and A is a non-empty finite set of *attributes*. For each object $x \in U$ and attribute $a \in A$, $a: U \rightarrow V_a$. The set V_a is called the *value set* of a.

One of the attributes is often called the *decision attribute*. It implies a known outcome of classification. Other elements of A are now called *condition attributes*. This kind of an information system is called a *decision system*.

Table 1 An example decision system.

	a_1	a_2	a_3	d
x_1	10-20	Yes	1-5	1
x_2	10-20	Yes	5-10	0
x_3	20-35	No	5-10	0
x_4	20-35	No	1-5	1
x_5	10-20	Yes	5-10	1
x_6	20-35	No	5-10	0

2.2 Indiscernibility

The starting point of rough sets theory is the indiscernibility relation. Indiscernibility relation is intended to express the inability to discern some objects from each other due to lack of knowledge.

Let $\mathcal{A} = (U,A)$ be an information system. Then any $B \subseteq A$ determines an equivalence relation [4] $IND_{\mathcal{A}}(B)$, which is called the *B-indiscernibility relation* and defined as follows:

$$IND_{\mathcal{A}}(B) = \{(x,x') \in U^2 \mid \forall a \in B\ a(x) = a(x')\}$$

If $(x,x') \in IND_{\mathcal{A}}(B)$, then objects x and x' are indiscernible from each other by attributes from B. The family of all equivalence classes of $IND_{\mathcal{A}}(B)$ are denoted $U/IND_{\mathcal{A}}(B)$, or simply U/B. For example in Table 1, objects x_1 and x_2 are indiscernible by attributes $\{a_1,a_2\}$, but after adding the attribute a_3 they are discernible from each other. The partition constructed by attributes of $B = \{a_1,a_2,a_3\}$ for the objects in Table 1 is

$$U/B = \{\{x_1\},\{x_2,x_5\},\{x_3,x_6\},\{x_4,\}\}.$$

2.3 Set Approximation

Now a new partition of universe U can be found by the indiscernibility relation. Let $\mathcal{A} = (U,A)$ be an information system and let $B \subseteq A$ and $X \subseteq U$. X can be approximated using only the information contained in B by constructing the *B-lower* and *B-upper approximations* of X. These basic operations in rough sets theory are defined as follows:

$$\underline{B}X = \bigcup\{Y \in U/B \mid Y \subseteq X\}$$

$$\overline{B}X = \bigcup\{Y \in U/B \mid Y \cap X \neq \varnothing\}$$

$\underline{B}X$ is the set of all objects of U that can be certainly classified by set B as members of X and $\overline{B}X$ is a set of the objects that can be probably classified by B as members of X. The set

$$BN_B(X) = \overline{B}X - \underline{B}X$$

is referred to as the *B-boundary region* of X and thus consists of those objects that cannot surely be classified into X on the basis of knowledge in B. If the boundary region of X is the empty set, then X is *crisp* with respect to B and if it is not the empty set, then X is referred to as *rough* with respect to B. The set

$$U - \overline{B}X$$

is called the *B-outside region* of X and consists of the objects that can be certainly classified by set B as not belonging to X.

The decision attribute d induces a partition of the universe of objects U. The induced partition is therefore a collection of equivalence classes X_i, called *decision classes*. In most applications, decision classes are the sets to be approximated. For example, let $X_1 = \{x \mid d(x)=1\}$ in Table 1. The set approximations for X_1 are

$$\underline{B}X_1 = \{x_1,x_4\},$$

$$\overline{B}X_1 = \{x_1,x_2,x_4,x_5\},$$

$$BN_B(X_1) = \{x_2,x_5\} \text{ and}$$

$$U - \overline{B}X_1 = \{x_3,x_6\}.$$

As a measure of quality of a partition approximation by attribute set B, it is possible to compute the coefficient

$$\gamma(B, d) = \frac{\sum_{i=1}^{n} card(\underline{B}X_i)}{card(U)},$$

where *card* is a set cardinality. It expresses the ratio of elements that can be properly classified employing attributes in B to all elements of the universe. If $\gamma(B,d) = 1$, it is said that d depends totally on B; and if $\gamma(B,d) < 1$, it is said that d depends partially on B.

2.4 Relative Reduct

An information system may contain unnecessary attributes. For a decision system this means that all condition attributes are not needed to describe dependencies between condition and decision attributes. The simplification of dependencies is based on the concept of *relative reduct* of rough sets theory. [5]

The relative reduct of the attribute set B with respect to $\gamma(B,d)$ is defined as a subset $RED(B,d) \subseteq B$ such that

1. $\gamma(RED(B,d),d) = \gamma(B,d)$ and
2. for any $a \in RED(B,d)$, $\gamma(RED(B,d)-\{a\},d) < \gamma(B,d)$, that is the relative reduct is a minimal subset with respect to property 1.

An information system may have more than one reduct. Intersection of all reducts is called the *core*.

2.5 Decision Rules

The simplest way of rule generation is to interpret each row of a reduced decision system as a rule, i.e., the values of condition attributes imply a certain value of decision attribute. For example the first row in Table 1 can be read

if a_1 is 10-20 and a_2 is Yes and a_3 is 1-5
then d is 1

If the condition attributes always imply the same value of decision attribute, the decision rule is said to be consistent (certain), otherwise the decision rule is inconsistent (possible).

3 Problem formulation

3.1 Data collection

NetFlow is a switching technology developed by Cisco. In addition to switching, it enables collecting flow data from routers.[7] In NetFlow, a central concept is a *flow*. A flow is a uni-directional stream of packets with common source and destination, protocol, type of service and input interface [8]. A *session* in turn consists of one or more similar flows. Entries in the data collected by NetFlow are sessions and referred to as rows in this paper, describing the form of the data matrix. Each row has several attributes, for example source and destination ip-addresses, source and destination ports and timestamps.

The data used here was collected from a test network with several types of servers and users. Fig.1 illustrates the system: The router running NetFlow is between a local area network (LAN) and the Internet. Traffic is recorded from the inside interface of the router, in other words the byte and packet counts of the data indicate traffic flowing out of the LAN. A suitable amount for the analysis was two days' traffic.

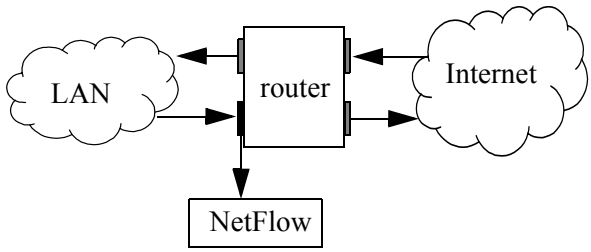

Fig.1 Illustration of the data collection system.

3.2 Target

The aim was to classify network traffic sessions (data rows) by applications. Rough sets were chosen to be the method, which means that the application should be used as the decision attribute and the decision classes induced by application should be approximated (see section 2.3). In practice, destination port can be used as the decision attribute because in this case destination port defines the application [9].

4 Problem solution

4.1 Preprocessing of data

There were 14 attributes altogether. Out of these 14, some were considered to be useless or even harmful to the classification. For example, ip-addresses were rejected because the goal was to recognize different applications from the characteristics of the sessions, not addresses. The selected condition attributes were time of the day, protocol, packet count, byte count, flow count, duration and total active time.

The data included a huge number of different applications using a huge number of ports. To focus the analysis to the relevant part of data, only the applications with a percentage of more than 0,1 % of the data were chosen. This elimination left 13 different applications in the data, rest of them were considered too rare to be classified.

The selections described above were done using MATLAB®, after which the data was moved to ROSETTA [10]. ROSETTA is a toolkit for analyzing tabular data within the framework of rough set theory. All operations introduced in Chapter 2 as well as several different algorithms for data processing and tools for classification are implemented.

After selecting the significant applications and attributes as described, there were 15832 rows of data left. The amount was split into a training set and a val-

idation set containing 75 % and 25 % of the data, respectively. Among the attributes, there were some of them with continuous values, so they had to be discretised.

Discretisation was performed using equal frequency binning. This involved fixing the number of intervals (8 for time of the day, 3 for others) and setting boundaries between the intervals so that approximately the same number of objects fell into each interval.[10]

4.2 Rule generation

As mentioned in section 4.1, the attributes were chosen using a priori knowledge. This knowledge turned out to be quite relevant: no reduction could be made using the technology described in section 2.4, which means that all attributes were necessary for the classification.

Rules were generated simply by interpreting each unique row of the training set as a rule. Inconsistent rules, i.e. rules with similar condition attributes but different decision attributes, were removed with a simple voting mechanism: each rule was attached to a counter telling how many data rows supported the rule, and the rule with the largest counter value was selected among the ambiguous ones. After this removal, the size of the rule base was 387 rules.

5 Results

To see how the generated rule set behaves with new data, the validation set was classified using the rules. Table 2 shows results from the validating classification.

The fourth column of the table shows a percentage of successfully classified data rows for each application. Most of the applications were classified with a percentage of more than 90. There are some applications showing a poor percentage. But taking into account the numbers from the third column, one can see that the poor performance is due to a small amount of data.

The classification of http-traffic (port 80) did not succeed as well as other main applications. It can be noted that 17 rows (7,2 %) of http were classified as ftp (port 20), which is quite natural since the two applications are similar in characteristics. This is the largest single error of the classification test. The total percentage on the bottom line of Table 2 is a weighted average of all applications.

In section 2.3, a quality measure γ for the set approximation was introduced. The value of γ for this validating classification is 62,4 %. Even though the value seems fairly low, classification was done remarkably well. Inconsistent rules deteriorate the γ value while they have no impact on the classification, thanks to the voting principle. In spite of measuring the roughness of the sets to be approximated, the γ coefficient is not a very good indicator of classifying performance when a voting method is used for inconsistent rules.

Table 2 Results from classification of the validation data set. [9]

Application	Port	Number of data rows	Successfully classified
icmp	0	602	97,7 %
ftp/default data	20	355	99,7 %
ftp/control	21	7	0,0 %
dns	53	2177	99,5 %
http	80	235	86,8 %
ntp	123	76	93,4 %
netbios-ns	137	66	97,0 %
mobileip-agent	434	150	99,3 %
unknown	1321	13	7,7 %
napster	6699	12	33,3 %
napster	6700	5	0,0 %
unknown	8888	146	99,3 %
unknown	31779	114	99,1 %
Total			97,5 %

6 Conclusion

The experiment reported here shows some important advantages of rough sets theory over many other methods. Selecting the variable to be classified as the decision attribute provides an inherent way of getting a set of qualitative classes. This is desirable when no meaningful metric exists between the classes.

The data used was not perfectly suitable for rough sets because it included several continuous-valued variables. Nevertheless, the percentage of all successfully classified data rows was 97,5 %, which can be considered a fairly good result.

In addition to classifying network applications, an interesting use for the rules could be eliminating known applications from network data. A supervisory system observing a network could regard the fraction of data conforming to the rule set as safe and concen-

trate only on the fraction that can not be recognised by the rules.

Vast amount of data is an increasing problem in many industrial applications. The results of this paper can also be seen as information compression: essential features of 16 000 rows of data were reduced into a few hundred rules.

6.1 Needs for further research

These results were achieved using offline data from a period of two days. To be used in network monitoring, the algorithm should be working online. This requires no major changes to the procedure itself, but a possibility to receive online traffic data is naturally necessary.

The generated rules revealed a common problem of most artificial intelligence systems: a large rule base. Even though classification of the validation set with 387 rules was computationally not very tedious, reducing the size of the rule base is a relevant subject of further research.

To get a better picture of the power of rough sets, a comparison with other classification methods using the same data set should be made. Other methods studied by the authors include self-organising maps and fuzzy rule generation.

References:

[1] Ali, A.A.; Tervo, R., Traffic Identification using Bayes' Classifier, *Canadian Conference on Electrical and Computer Engineering*, Halifax, Canada, 7-10 March 2000.

[2] http://www.cisco.com/warp/public/732/Tech/netflow/: Cisco IOS NetFlow, 14 Nov 2001 12:30.

[3] Pawlak, Z: *Rough Sets – Theoretical Aspects of Reasoning about Data*, Kluwer Academic Publishers, The Netherlands, 1991.

[4] Pawlak, Z.: Granularity of Knowledge, Indiscernibility and Rough Sets, *The 1998 IEEE International Conference on Fuzzy Systems Proceedings. IEEE World Congress on Computational Intelligence*, Anchorage, USA, 4-9 May 1998.

[5] Ziarko, W.; Shan, N.: Discovering Attribute Relationships, Dependencies and Rules by Using Rough Sets, *Eighth Hawaii International Conference on System Sciences*, Wailea, USA, 3-6 Jan 1995.

[6] Pal, S.K.; Skowron, A.: *Rough Fuzzy Hybridization – a New Trend in Decision-Making*, Springer-Verlag, Singapore, 1999.

[7] Della Maggiora, P.L.; Elliott, C.E.; Pavone, R.L.; Phelps, K.J.; Thompson, J.M.: *Performance and Fault Management*, Cisco Press, Indianapolis, USA, 2000.

[8] http://www.caida.org/tools/measurement/cflowd/newfaq.xml: cflowd – Frequently asked questions, 14 Nov 2001 12:30.

[9] http://www.iana.org/assignments/port-numbers: Internet Assigned Numbers Authority, Port Numbers, 14 Nov 2001 12:30.

[10] Øhrn, A.: *ROSETTA Technical Reference Manual*, Department of Computer and Information Science, Norwegian University of Science and Technology (NTNU), Trondheim, Norway, 2000.