A Comparison of Fuzzy C-means Clustering and Rough Sets Based Classification in Network Data Analysis

LAAMANEN, VESA; LAMPINEN, TIMO; 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 compares two different classification methods, Fuzzy C-means clustering and rough sets. After introducing some basic concepts of both, the methods are tested with data collected from a test network. It becomes evident that the two methods are quite different from the beginning. They require different preprocessing of the data, have different features and give different results. Fuzzy C-means clustering and rough sets are also compared in the light of the results obtained. Each one has its advantages, and the choice of method should be made in accordance with the classification problem.

Keywords: Fuzzy C-means, Rough sets, Traffic analysis, Network analysis, Classification, Clustering

1 Introduction

A vast amount of numerical data is generated in today's computer networks. Extracting information from the data interests parties like operators, maintenance personnel and even users. It is often impossible to analyse the whole data, but one has to focus the analysis on an important portion of the data. This is where classification steps in: if the data can be classified using some criteria, only the classes of interest can be selected for analysis or processing while the rest is rejected.

It is crucial to choose a suitable method for the classification. The aim of this paper is to compare the features and results of two different methods, Fuzzy C-means clustering and classification based on rough sets. Each of the methods can be used for classification of network traffic data, and each one has its own advantages and disadvantages.

The topic of network data classification has been studied before in [1] and [11].

2 Theoretical background

2.1 Fuzzy C-means clustering

Fuzzy C-means (FCM) algorithm, also known as Fuzzy ISODATA, was introduced by Bezdek [3] as an extension to Dunn's [5] algorithm. FCM-based algorithms are the most widely used fuzzy clustering algorithms for practical purposes.

Let $\mathbf{X} = \{\mathbf{x}_1,\mathbf{x}_2,...,\mathbf{x}_N\}$, where $\mathbf{x}_i \in \Re^n$, present a given set of feature data. The objective of the FCM-algorithm is to minimize the Fuzzy C-means cost function formulated as

$$J(\mathbf{U}, \mathbf{V}) = \sum_{j=1}^{C} \sum_{i=1}^{N} (\mu_{ij})^m \left\| \mathbf{x}_i - \mathbf{v}_j \right\|^2 . \quad (1)$$

$\mathbf{V} = \{\mathbf{v}_1,\mathbf{v}_2,...,\mathbf{v}_C\}$, $\mathbf{v}_j \in \Re^n$, are the cluster centers. $\mathbf{U} = (\mu_{ij})_{N\times C}$ is a fuzzy partition matrix, in which each member indicates the degree of membership of data vector $\mathbf{x}_i$ in cluster j. The elements of matrix $\mathbf{U}$ should satisfy the following conditions:

$$\mu_{ij} \in [0,1] , \; i = 1,...,N, \; j = 1,...,C \quad (2)$$

$$\sum_{j=1}^{C} (\mu_{ij}) = 1 , \; i = 1,...,N \quad (3)$$

The exponent $m \in [1,\infty]$ is the weighting exponent determining the fuzziness of the clusters. The most commonly used distance norm is the Euclidean distance $d_{ij} = \|\mathbf{x}_i - \mathbf{v}_j\|^2$, although Babuska suggests that other distance norms could produce better results [2].

Minimization of the cost function $J(\mathbf{U},\mathbf{V})$ is a nonlinear optimization problem, which can be minimized with the following iterative algorithm:

Step 1. Initialize the membership matrix $\mathbf{U}$ with random values so that conditions (2) and (3) are sat-

isfied. Choose appropriate exponent m and termination criteria ε.

Step 2. Calculate cluster centers **V**:

$$\mathbf{v}_j = \frac{\sum_{i=1}^{N} (\mu_{ij})^m \mathbf{x}_i}{\sum_{i=1}^{N} (\mu_{ij})^m}, \; j = 1,...,C$$

Step 3. Calculate new distance norms

$$d_{ij} = \left\| \mathbf{x}_i - \mathbf{v}_j \right\|^2, \; i = 1,...,N, \; j = 1,...,C$$

Step 4. Update the fuzzy partition matrix **U**:
If $d_{ij} > 0$ (indicating $\mathbf{x}_i \neq \mathbf{v}_j$), then

$$\mu_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{d_{ij}}{d_{ik}} \right)^{\frac{2}{m-1}}}.$$

Else, $\mu_{ij} = 1$.

Step 5. If the termination criterion is met, stop. Else, go to Step 2.

A suitable termination criteria could be to evaluate the cost function (Eq. 1) and see whether it is below a certain tolerance value or if its improvement compared to the previous iteration is below a certain threshold [10]. Also the maximum number of iteration cycles can be used as a termination criterion.

2.2 Rough sets theory

The method of data base analysis presented here is based on the rough sets theory and was introduced by Pawlak [13] in the early 1980's. It deals with the classificatory analysis of data tables.

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.

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.

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

Let $\mathcal{A} = (U,A)$ be an information system. Then any $B \subseteq A$ determines an equivalence relation [12] $IND_{\mathcal{A}}(B)$, which is called the B-indiscernibility relation and is 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

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

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

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

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.

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. [15]

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.

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).

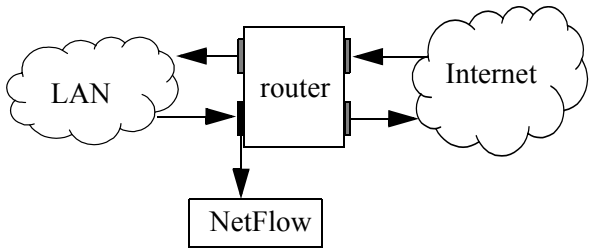

Fig.1 Illustration of the data collection system.

3 Analysis

3.1 Data collection

NetFlow is a switching technology developed by Cisco [8]. In addition to switching, it enables collecting flow data from routers [4]. 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 [7]. 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.

3.2 Preprocessing of data

All the preprocessing described below was done using MATLAB®. MATLAB's Fuzzy Logic Toolbox was used for Fuzzy C-means clustering, whereas rough set operations were done in ROSETTA [16]. ROSETTA

is a software toolkit capable of performing all the operations introduced in section 2.2 as well as several different algorithms for data processing and classification.

The data included a huge number of 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. Application as such was not included in the data, but it can be inferred from source and destination ports [9]. Because the data covered only unidirectional traffic, destination port was used in place of the application.

In rough sets classification, the number of classes is inherently induced by the decision attribute and was thus 13. In FCM on the other hand, the result is a set of clusters instead of classes. The number of clusters is the most important parameter set by the user. In this experiment, a priori knowledge was used to set the number of clusters to 25. A cluster number greater than the number of applications in the data is reasonable because of the ambiguous nature of some applications. For example, different parts of DNS (domain name server) may diverge to separate clusters. Several ways to determine the number of clusters in FCM can be found in [6] and [14].

There were 14 attributes available in the data. 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 for rough sets were time of the day, protocol, packet count, byte count, flow count, duration and total active time. For FCM, the number of attributes had to be kept smaller for computational reasons and thus the original attributes were processed to yield bytes per packet and bytes per second. In addition to these, packet count, flow count and duration were used. Qualitative attributes like protocol could not be used with FCM because there is no meaningful distance measure between qualitative attribute values.

The aim was to classify network traffic sessions (data rows) by applications. The two methods differ slightly in the nature of classification. Rough sets can be used to classify the data straightforwardly by using the application as the decision attribute and approximating the decision classes induced by application (see section 2.2). FCM in turn builds clusters of similar applications. A cluster can contain several applications and thus cannot be directly labelled with one single application.

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 validation set containing 75 % and 25 % of the data, respectively.

Among the attributes, there were some of them with continuous values. Additionally, discrete values with a large range (for example byte count) had to be considered continuous. These continuous-valued attributes were discretised for rough sets. The discretisation was performed with equal frequency binning [16].

In FCM, it is important to have similar ranges for different attributes to avoid dominating attributes in the cost function (Eq. 1). Some attributes had an exponential distribution, so a logarithmic scaling was applied to them. A linear scaling was sufficient for the rest of the attributes.

4 Results

4.1 Fuzzy C-means clustering

Several hundred rounds of the FCM algorithm presented in section 2.1 were run on the training data to achieve best results. The algorithm produces one rule per cluster, which means 25 rules in this case.

Subsequently, the validation data set was used to attach applications to the clusters obtained. This procedure can be thought of as turning clusters into classes: each cluster is given a name of an application. Because a cluster can contain more than one application, this naming procedure is not always feasible but was done here for comparison with rough sets classification results.

Each row of the test data was attached to the closest cluster center, which is called nearest prototype [3]. The rightmost column of Table 2 tells how many per cent of all data rows attached to the cluster represent the application mentioned in fourth column. It should be noted that the percentage is not a quality measure of the classification, it just tells how much the cluster contains data from the application that it is named after. Percentages in Tables 2 and 3 are thus not comparable.

Some observations can be made from the results in Table 2. Applications with features different from others, like icmp and dns, formed clusters almost alone. Most applications have several different subsets that were divided into separate clusters.

Table 2 Results from classification of the validation data set based on Fuzzy C-means. Applications were attached to the nearest cluster, and the application with most hits is listed with the cluster.[9]

Cluster number	Number of data rows	1st application		
		Port	Name	Hit rate
1	257	0	icmp	100 %
2	34	0	icmp	100 %
3	35	0	icmp	100 %
4	55	0	icmp	100 %
5	148	0	icmp	100 %
6	109	0	icmp	100 %
7	51	20	ftp	100 %
8	55	20	ftp	83,6 %
9	187	20	ftp	95,5 %
10	69	20	ftp	99,1 %
11	15	53	dns	99,6 %
12	66	53	dns	100 %
13	463	53	dns	98,9 %
14	115	53	dns	98,9 %
15	136	53	dns	100 %
16	28	80	http	100 %
17	34	80	http	83,5 %
18	79	80	http	92,5 %
19	212	80	http	76 %
20	40	123	ntp	49,6 %
21	59	137	netbios-ns	84,8 %
22	25	434	mobileip-agent	100 %
23	1177	434	mobileip-agent	100 %
24	25	8888	unknown	100 %
25	301	31779	unknown	79,4 %

The table lists only the application with the highest hit rate. Also some of the second highest are interesting: For cluster 17, 16,5 % of the attached applications were dns. Hence, http and dns requests are obviously similar in characteristics because they were inseparable by FCM.

4.2 Rough sets

As mentioned in section 3.2, 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 2.2, which means that all attributes were necessary for the classification with rough sets.

Table 3 Results from rough sets based classification of the validation data set.[9]

Application	Port	Number of data rows	Successfully classified
icmp	0	602	89,2 %
ftp/default data	20	355	99,7 %
ftp/control	21	7	0,0 %
dns	53	2177	98,6 %
http	80	235	80,1 %
ntp	123	76	93,4 %
netbios-ns	137	66	95,5 %
mobileip-agent	434	150	99,3 %
unknown	1321	13	0,0 %
napster	6699	12	16,7 %
napster	6700	5	0,0 %
unknown	8888	146	97,9 %
unknown	31779	114	94,7 %
Total		3958	95,1 %

Rules were generated simply by interpreting each unique row of the training set as a rule. Some filtering was performed to reject the least relevant rules. 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 these operations, the size of the rule base was 193 rules.

To see how the generated rule set behaves with new data, the validation set was classified using the rules. Table 3 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 possibly similar in characteristics. This is the largest single error of the classification test. The total percentage on the bottom line of Table 3 is a weighted average of all applications.

5 Conclusion

A fair comparison of the results is difficult because of the differences in the classification methods used. Taking into account the experimental nature of this study, results from both methods can be considered fairly good.

The results show that Fuzzy C-means clustering is not a perfect method if a classification into predetermined classes is desired. FCM rather groups the data into clusters of similar data, regardless of which applications the data points in a cluster represent. While some applications were still classified remarkably well, the clusters are more useful in finding similar applications.

Rough sets theory showed its power in classifying data into sets inherently induced by a decision attribute. The price paid is the size of the rule base. Rough sets generated 193 rules, while FCM managed with only 25 rules.

The data used was not perfectly suitable for either of the methods. Rough sets theory could easily handle qualitative attributes but required discretisation of the quantitative ones, while FCM had problems with attributes with irregularly distributed values.

Further research needs on the topic include developing methods for preprocessing, reducing the rule base and perhaps applying classification online.

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] Babuska, R.: *Fuzzy Modelling for Control*, Kluwer Academic Publishers, The Netherlands, 1998.

[3] Bezdek, J.: *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, USA, 1981.

[4] 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.

[5] Dunn, J.C.: A Fuzzy Relative of the ISODATA Process and its Use in Detecting Compact, Well Separated Clusters, *Journal of Cybernetics*, Vol. 3, No. 3, 1974, pp. 32-57.

[6] Halkidi, M.; Batistakis, Y.; Vazirgiannis, M.: Clustering Algorithms and Validity Measures, *Thirteenth International Conference on Scientific and Statistical Database Management*, Fairfax, USA, 18-20 July 2001.

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

[8] http://www.cisco.com/warp/public/732/Tech/netflow/: Cisco IOS NetFlow, 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] Jang, J.-S.; Sun, C.-T.; Mizutani, E.: *Neuro-Fuzzy and Soft Computing*, Prentice-Hall, USA, 1997.

[11] Laamanen, V.; Laurikkala, M.; Koivisto, H.: Network Traffic Classification Using Rough Sets, *3rd WSEAS Symposium on Mathematical Methods and Computational Techniques in Electrical Engineering*, Athens, Greece, 29-31 Dec 2001 (accepted for publication).

[12] 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.

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

[14] Setnes, M.: Complexity Reduction Methods for Fuzzy Systems Design. In: Babuska, R. (editor); Verbruggen, H.B. (editor): *Fuzzy Logic Control: Advances in Applications*, World Scientific Publishing Company, Inc., The Netherlands, 1999, pp. 87-111.

[15] 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.

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