Clustering Wireless Network Subscribers

Lucas A. Andrianos

Foundation for Research and Technology, Hellas
Orthodox Academy of Crete
Kolympari, 73006, Chania, Greece

Abstract

This paper reports the cluster computing of billing records from the Telus Mobility Cellular Digital Packet Data (CDPD) network using AutoClass clustering tool. Analytical description of AutoClass and a comparison with the clustering results from K-means algorithm are given. AutoClass proved to be an efficient tool for a thorough and automatic clustering of complex database, such as user billing records database from 2096 customers with 12 distinct attributes. Clustering results from AutoClass revealed that CDPD subscribers might be classified into 26 classes of subscribers. Cluster computing of billing records from CDPD network is undertaken for the first time in this study. It provided useful information about the usage of an operational wireless network and the possibility of valorization of a network billing records to study subscribers' behavior. Such information may be valuable for the management of commercial wireless network.

Keywords: Cluster computing, traffic analysis, mobile networks, billing records, CDPD networks, wireless networks.

1. Introduction

A number of recent research reports have been devoted to collection and analysis of traces from operational wireless networks. Analysis of these traces provided useful information about the behavior of network users and network cells [Andriantiatsaholiniaina et. Al. (2002)], [Tang et. Al. (1999)], [Tang et. Al. (2000)]. Collected traces also enabled traffic-based performance evaluations of wireless networks [Hutchins et. Al. (2001)], [Hutchins et. Al. (2002)].

This paper reports the analysis of a billing record of twenty one days obtained from Telus Mobility CDPD network. One of the objectives was to classify network subscribers. We performed data extraction in a variety of ways. Java programs were used to parse the billing records and to write billing data as a series of Structured Query Language (SQL) statements that load the billing data into a MySQL database.

The main difficulty in analyzing billing records has been dealing with its sheer volume. Hence, we used data mining and a machine learning technique called cluster computing. Cluster computing is used for discovering hidden patterns and trends in a given data set. It groups data into categories with similar behavior. In our analysis, we

choose to use AutoClass classification tool [Hanson et. Al (1991)], [Cheeseman et. Al. (1995)] and K-means clustering algorithm from the S-PLUS statistical package [Jiawei et. Al (2001)], [Russell et. Al. (1995)].

We were able to provide a classification of network subscribers. These results may help in improving wireless network management and in creating workload models for commercial wireless networks.

The remainder of this paper is organized as follows. In Section 2, we present an overview of CDPD protocol and the CDPD billing data. In Section 3, we describe AutoClass clustering tool and the preparation of CDPD billing records for input to AutoClass. Clustering results with AutoClass are presented in Section 4. In Section 5, we give a comparison of AutoClass clustering results with K-means algorithm clustering results. We conclude with Section 6.

2. Background

In this section, we describe briefly the CDPD protocol and the CDPD billing data.

2.1 CDPD Protocol

CDPD is a standard protocol developed for commercial public mobile data communication networks [Agosta et. Al. (1996)], [Sreetharan et. Al. (1996)]. The CDPD communication architecture is based on the Open Systems Interconnection (OSI) Reference Model [Walrand et. Al. (2000)]. It deals only with the lower three OSI layers. CDPD network's function is to enable data transmission between Mobile End Systems (M-ESs) and Fixed End Systems (F-ESs). M-ESs are connected to the backbone network through the Mobile Data Base Station (MDBS), the Mobile Data Intermediate System (MD-IS also called a mobile router), and an Intermediate system. The network also includes several Fixed End Systems (F-ESs) connected to the wired backbone network [Jiang et. Al. (2001)].

Although the CDPD network supports multiple protocols at the network layer, all CDPD network support services are based on International Standards Organization (ISO)/OSI protocol suites [Walrand et. Al (2000)]. The two categories of CDPD support services are CDPD network support and CDPD network application support.

We are in particular interested in CDPD network application support services, which include network management, message handling, and accounting. Accounting services provide information on how and by whom the CDPD network resources are used. The CDPD accounting services collect network usage data for every subscriber to enable compilation for billing purposes. The collected data may include packet count, packet size, source and destination addresses, cell ID used by M-ESs, and an approximate time of transmission.

2.2 Description of Billing Data

CDPD analysis of network records often focuses on the overall network behavior (network elements, network events, and network activity characterization) and on traffic characteristics from a user point of view (when, for how long, and in which manner customers use the network). Records could be session data, transport layer data, application layer data, and movement data [Andriantiatsaholiniaina et. Al. (2002)], [Tang et. Al. (1999)], [Tang et. Al. (2000)], [Hutchins et. Al. (2001)], [Hutchins et. Al. (2002)].

The billing records from a CDPD network that we analyzed summarize the network activity. The service provider uses them to bill its customers for network usage. The structure of the billing record is a set of directories and files. Each directory is named after a sequence number (e.g., 528/, 529/) and contains up to 100 data files also named sequentially (e.g., 00074-52800, 00074-52801, 00074-52802). Each file contains a record of approximately fifteen minutes of network activities. A file consists of a header that contains timestamp and sequencing information, and a series of Traffic Matrix Segments (TMS) rows. The smallest number of recorded TMS rows in a file is 300 and the largest is 2,694. Each row represents a single event of either registration, deregistration, or IP data type. Table 1 shows the list of all the attributes that may contain a TMS row.

Table 1. *List of TMS row attributes*

Attribute index	Attribute description	Range	Values
1	Traffic Type	3	Discrete
2	Cell ID	60	Discrete
3	SPI	1	Real
4	WASI	2	Real
5	Serving SPI	1	Real
6	Serving BRI	1	Real
7	Actual PIC	1	Real
8	Serving Description	1	Real
9	Version	2	Real
10	Length	2	Real
11	Primary SPI	7	Real
12	Primary HAD	6	Real
13	Routing SPI	4	Real
14	Routing HAD	4	Real
15	Subscriber PIC	1	Real
16	Customer Class	3	Real
17	Home BRI	37	Real

18	Home Location S-D ID	1	Discrete
19	Mobile NEI IP address	2096	Discrete
20	Equipment ID	2122	Discrete
21	Authentication Sequence No.	16974	Real
22	Home MD-IS-NEI-CLNP address	9	Discrete
23	Deregistration Time	69255	Timestamp
24	Deregistration reason	3	Discrete
25	Group Mobile ID	19	Discrete
26	Registration attempt time	518903	Timestamp
27	Registration attempt result	4	Discrete
28	Sender Receiver NEI	23883	Discrete
29	Mobile source/destination	3	Discrete
30	Data packet count	2323	Real
31	Data octet count	40528	Real
32	Control packet count	200	Real
33	Control octet count	663	Real
34	Discarded packet count	291	Real

3. What is AutoClass?

AutoClass is an unsupervised Bayesian classification system that seeks a maximum posterior probability classification [Hanson et. Al. (1991)], [Cheeseman et. Al. (1995)]. AutoClass' principal key features are the automatic determination of the number of classes, the possibility to use mixed discrete and real valued data, the ability to handle missing values, the optimized time processing which is roughly linear in the amount of data, the availability of probabilistic class membership for each case and the correlation between attributes within a class.

3.1 Description of AutoClass

AutoClass uses only vector valued data, in which each instance to be classified is represented by a vector of values, each value characterizing some attribute of instance. Values can be either real numbers, normally representing a measurement of the attribute, or discrete, a set of characteristics of the attribute.

AutoClass models the data as mixture of conditionally independent classes. Each class is defined in terms of a probability distribution over the meta-space defined by the attributes, so that an object can have partial membership in different classes, and the class definitions can overlap. AutoClass uses Gaussian distribution over the real valued attributes and Bernoulli distributions over the discrete attributes.

AutoClass provides four models of attributes:

- Single_multinomial – discrete attribute multinomial model with missing values;
- Single_normal – real valued attribute model with no missing values;
- Single_normal_missing – real valued attribute model with missing values;
- Multi_normal – real valued covariant normal model with no missing values.

AutoClass finds the set of classes that is maximally probable with respect to the data and the model. The output is a set of class descriptions and partial membership of the instances in the classes.

3.2 Running AutoClass

We need to prepare four "AutoClass files" for running AutoClass: data file (.db2 file), header file (.hd2 file), model file (.model file), and search parameter file (.s-params file). AutoClass starts to search for the best classification of the data by the command:
% autoclass –search <.db2 file path> <.hd2 file path> <.model file path> <.s-params file path>
To restart a previous search, specify that "force_new_search_p" has the value "false" in the search parameter file, since its default is "true". Specifying "false" tells "AutoClass –search" to try to find a previous compatible search *(<…>.results[-bin] & <…>.search*), to continue from, and will restart using it if found. To force a new search instead of restarting an old one, give the parameter "force_new_search_p" the value of "true", or use the default. If there is an existing search *(<…>.results[-bin] & <…>.search*), one will be asked to confirm continuation since continuation will discard the existing search. If a previous search is continued, the message "restarting search" will be given instead of the usual "beginning search". It is generally better to continue a previous search than to start a new one, unless the user is trying a significantly different method, in which case statistics from the previous search may mislead the current one.
A running commentary on the search will be printed to the screen and to the log file, unless "log_file_p" is "false".
After each try, a very short report of only a few characters long is given. After each new best classification, a longer report is given, but no more than "min_report_period" (default is 30 seconds).

3.3 Generating Reports with AutoClass

AutoClass provides three standard reports that can be generated from "results" files by the command:
% autoclass –reports <.results[-bin] file path> <.search file path> <.r-params file path>
The report parameters definitions are contained as comments in a report parameter file, file type ".r-params". These standard reports are the attribute influence values, the cross-reference by case (datum) number and the cross-reference by class number.

The reports are output to files whose names and pathnames are taken from the ".r-params" file pathname. The default is to generate reports only for the best classification in the "results" file.

3.4 Input preparations for AutoClass

To perform cluster computing with AutoClass, four files need to be prepared: header file, data file, model file, and search parameter file. The header files (*Autoclass cdpd.hd2 file*) describe the specific data format and attribute definitions. Data file contains the dataset to be analyzed (*AutoClass cdpd_data.db2 file)*. The model file specifies the form of the probability distribution function for classes in that database (*AutoClass cdpd_model file*). The search parameter file contains search parameter definitions as comments (*AutoClass cdpd_.s-params file*).
The type of all attributes is real scalar and we used the single multinomial type of model with no missing values. We choose for clustering variables all real attributes, which present a range greater than ten in the billing records. All discrete attributes and real attributes whose range are less than ten were ignored during the classification process because each of them has little contribution to the classification and only complicate the analysis. Therefore, 13 attributes (12 attributes as variables and 1 as attribute index) are used to perform network subscribers clustering analysis.

4. Clustering results with AutoClass

AutoClass is looking for the best possible classification of the data. A classification is composed of a set of classes, a set of class weights and a probabilistic assignment of cases in the data to these classes.
AutoClass has found 26 populated classes of users after 480 tries over 30 minutes 2 seconds. Due to lack of space, all entries of the Tables of model terms for all the 26 classes of users found by AutoClass could not be shown. The properties of each class in term of data packets transferred ("Mean – jk" for Log DP) and cells visited ("Mean – jk" for Log CV) are given in the Table 2.

Table 2. *"Mean – jk" values of "Log DP" and "Log CV" for all subscribers' classes*

Class number	"Mean – jk" for Log DP (Data Packets)	"Mean – jk" for Log CV (Cells Visited)	Normalized class weight
0	8.72e+00	2.80e+00	0.106
1	7.82e+00	2.50e+00	0.099
2	1.04e+01	2.65e+00	0.078
3	4.13e-03	2.12e-01	0.076
4	9.04e+00	2.18e+00	0.070
5	4.62e+00	1.25e+00	0.062
6	7.97e+00	2.55e+00	0.061

7	4.13e-03	1.00e+00	0.055
8	5.37e+00	1.21e+00	0.049
9	3.4e+00	2.66e-01	0.039
10	7.52e+00	1.98e+00	0.038
11	3.29e+00	1.61e-09	0.033
12	3.90e+00	7.80e-01	0.031
13	7.55e+00	1.20e+00	0.026
14	8.02e+00	1.44e+00	0.023
15	4.13e-03	8.10e-02	0.021
16	6.09e+00	1.60e+00	0.020
17	9.71e+00	2.88e+00	0.019
18	5.54e+00	1.91e+00	0.018
19	1.02e+01	2.10e+00	0.016
20	4.13e-03	1.16e+00	0.015
21	5.76e+00	1.71e+00	0.015
22	3.22e+00	4.53e-09	0.011
23	7.74e+00	3.56e+00	0.010
24	4.13e-03	5.47e-01	0.008
25	1.79e-01	2.27e-08	0.002

Class 0 includes 10.6% of total customers. It has the highest number of population and, therefore, it might be considered as the class of the majority of users. Users of the class 0 have a level of network usage valued by the "Mean – jk" for Log DP = 8.72e+00 in term of data packets transferred. We notice also that the majority of users move a lot; "Mean – jk" for Log CV of the *class 0* ("Mean – jk" for Log CV = 2.80e+00) is among the highest values (Mean – jk for Log CV = 3.56e+00).

Behavior of users belonging to different classes might be evaluated with respect to those of the members of *class 1*. For example, users belonging to class 1 have relatively low network usage (Mean – jk for Log DP = 7.82e+00) and low mobility ("Mean – jk" for Log CV = 2.50e+00) compared to the majority (*class 0*) in term of data packets transferred and cells visited (Table 2). Similarly, users belonging to *class 2* have relatively high network usage and low mobility compared to the majority of users.

5. Comparison with "k-means" clustering results

K-means is among the most well known and commonly used portioning method for clustering analysis. Given a database of n objects and k, the number of clusters to form, a partitioning algorithm organizes the objects into k partitions ($k <= n$), where each partition represents a cluster. The K-means algorithm clustering process is based on the mean value of the objects in the cluster. K-means is an iterative algorithm,

where the number of clusters *k* is supplied in advance. The numbers of desired clusters *k* as well as the database containing *n* objects are chosen as inputs for K-means algorithm clustering analysis. The outputs are the mean values of the objects in each cluster, which can be viewed as the cluster's center of gravity.

5.1 Clustering of subscribers with K-means

We used K-means clustering algorithm to find possible classification of network subscribers. We used the same twelve variables that were used with AutoClass to classify network users. The total population is 2,096 subscribers. The chosen clustering results (Table 3) that best matched graphical analysis (Figure 2) of the variables are those providing four distinct classes of users. Figure 2 shows a matrix of graphs illustrating the inter-relationships between some of the variables. For lack of space, we do not show all twelve variables in the matrix graph. Data plots may give an insight for the possible number of classes to be chosen in K-means clustering algorithm. For example, one class counting for only two users is easily observable.

Table 3. *K-means clustering results for the network subscribers (center values)*

Cluster ID	Nbr. of subscribers	Cells visited	Data packets	IP-Data events	Total events	Network usage
1	2,028	9.3	4,541.0	356.6	689.6	Low
2	55	11.4	90,768.0	1,775.3	2,001.8	Medium
3	11	10.7	267,608.0	5,678.7	5,947.2	High
4	2	9.0	1,955,340.5	3,226.0	3,748.5	Very high

Class 1 (96.7% of users) consists of customers with relatively low network usage, *class 2* (2.6%) consists of customers having medium network usage, *class 3* (0.5%) consists of customers having relatively high network usage, and *class 4* (only users 59.207.209.164 and 59.207.209.151) consists customers having exceptional behavior.

5.2 Comparison of AutoClass with K-means clustering results

Using the same number of clustering variables associated with the same population objects (2096 cases of customers), K-means algorithm suggests 4 classes of subscribers as the best results, while AutoClass provided an automatic classification of network users in 26 classes. Given that the number of clusters in K-means algorithm is chosen in priory, its results might not represent the best unbiased and hidden solutions for the clustering analysis problem. However, AutoClass could provide an unbiased and thorough classification of network subscribers in despite of the large size and the complexity of the dataset.

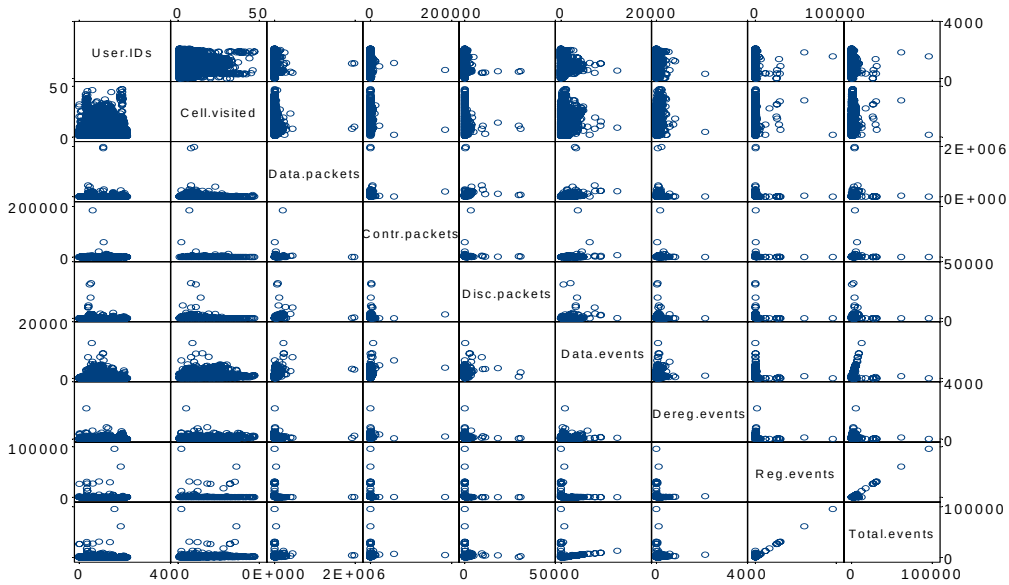

Figure 2. *Matrix of graphs showing the inter-relationship between clustering variables used in K-means algorithm*

6. Conclusions

Clustering analysis with AutoClass and K-means algorithm revealed that it was possible to analyze the behavior of the CDPD network subscribers through cluster computing of the billing record. Cluster computing of billing records from a CDPD wireless network is undertaken for the first time in this study. AutoClass clustering results revealed that subscribers might be automatically classified into 26 classes of users while K-means algorithm suggests 4 classes of subscribers as the best results.

AutoClass proved to be an efficient tool for a thorough clustering of complex database, such as user billing records database from 2096 subscribers with 12 distinct attributes. Moreover AutoClass provides automatic and unbiased clustering results that might be hidden and undiscovered by another clustering tool such as the K-means algorithm. Analysis of longer billing records and using more powerful clustering tools could provide additional insights.

This paper presents operational characteristics of the AutoClass clustering tool and provides useful information about the access to an operational wireless network. The results may be valuable for modeling network traffic, for creating commercial wireless workload models and for better management of wireless networks.

Acknowledgments

I would like to thank the "Foundation for Research and Technology – Hellas" for the financial support and the "Telus Mobility" for providing the billing records. Special credits are due to L. Trajković and P. Chan for initial contributions to this research.

References

Andriantiatsaholiniaina, L. A., Trajkovic, Lj. (2002), *Analysis of user behavior from billing records of a CDPD wireless network*, in Proc. LCN-2002: Local Computer Networks Conference, Tampa, FL, USA.

Tang, D., Baker, M. (1999), *Analysis of a metropolitan-area wireless network*, in Proc. MOBICOM-99: Mobile Communication International Conference, Seattle, WA, USA.

Tang, D., Baker, M. (2000), *Analysis of a local-area wireless network*, in Proc. MOBICOM-2000: Mobile Communication International Conference, Boston, MA, USA.

Hutchins, R., Zegura, E. W., Liashenko, A., Enslow, P. H. (2001), *Internet user access via dial-up networks – traffic characterization and statistics*, in Proc. CNP-2001: International Conference on Network Protocols, Riverside, CA, USA.

Hutchins, R., Zegura, E. W. (2002), *Measurements from a wireless campus network*, in Proc. ICC-2002: International Conference on Communications, New York City, NY, USA.

Hanson, R., Stutz, J., Cheeseman, P. (1991), *Bayesian classification theory*, Technical Report FIA-90-12-7-01, NASA Ames Research Center, Artificial Intelligence Branch, USA.

Cheeseman, P., Stutz, J. (1995), *Bayesian classification (AutoClass): theory and results,* Advances in Knowledge Discovery and Data Mining, U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, Eds. Menlo Park: The AAAI Press, USA.

Jiawei, H., Kamber, M. (2001), *Data mining: concepts and techniques*. San Francisco, CA: Morgan-Kaufmann, pp. 349-351.

Russell, S., Norvig, P. (1995), *Artificial Intelligence: A Modern Approach*. Upper Saddle River, NJ: Prentice-Hall.

Agosta, J., Russell, T. (1996), *CDPD: Cellular Digital Packet Data Standards and Technology*. Reading, MA: McGraw-Hill.

Sreetharan, M., Kumar, R. (1996), *Cellular Digital Packet Data*. Norwood, MA: Artech House.

Walrand, J., Varaiya, P. (2000), *High-Performance Communication Networks*, 2nd edition. San Francisco, CA: Morgan-Kaufmann, USA.

Jiang, M., Nikolic, M., Hardy, S., Trajković, Lj., *Impact of self-similarity on wireless data network performance*" in Proc. ICC-2001: International Conference on Communications, Helsinki, Finland.