Case Based Knowledge Management and Case Mining in Optimization of GSM Network

Wu Jing
Department of Electronic Engineering, Beijing University of Post and Telecom
Mailbox: 117 Zip Code: 100876 Beijing, China
Tel: 86-10-62282747, 86-13910614659
Fax: 86-10-62282747 Email: wuj@ipinhand.com
Yang Lu
Department of Management Engineering, Beijing University of Post and Telecom
Song Jun De
Department of Electronic Engineering, Beijing University of Post and Telecom

Abstract: In GSM (Global System for Mobile communication) network optimization, the engineers analyze data in OMC (Operation and Maintenance Center) based on their knowledge and experience till they have found what caused the low guidelines. CBR is used to simulate this procedure. However, vast data contains much information even the experienced engineers do not know yet, and the knowledge for optimization changes rapidly along with the network updating. In this paper, we describe the case based knowledge management in an intelligent system and use KDD to obtain cases. The relation between KM and case mining is discussed. The steps of KDD are defined. Fuzzy Logic and statistics are used to handle the uncertainty of data. The algorithms to find cases are described in detail. A new clustering algorithm to make the result simpler is especially proposed. Finally, the result is given.
Key Words: GSM, Data Mining, KDD, CBR, Fuzzy Logic

1. Introduction

1.1 Why Use Case in the optimization of GSM network

GSM (Global System for Mobile communication) is a standard of mobile communication used in many countries [1]. Along with the increasing of networks and users, optimization became more and more important for both operators and vendors. There are some experienced engineers but are not enough. So we decide to develop a new system to work like the real experts. The kernel of this system is Expert System [10].

Traditional Expert System is based on rules [10]. So in the first, we designed our knowledge base as rule based. But while we communicate with human experts, they feel very difficult to arrange their experience into rules. When they are optimizing the GSM network, they use the knowledge about GSM as well as their experience. They consider the problems and reasons simultaneously other than step by step. So CBR (Case Based Reasoning) technology can be used to model it [1], which is based on a memory-centered cognitive

Case 1: (simple representation)
{
problem: Call Drop Rate High
reason1: Qual_Up
reason2: Interf_Up
reason3: Qual_Down
}

model [2].

In optimization, “case” means a set of reasons causing the abnormal guidelines. We call the abnormal guideline “problem”. Each optimization task is for one cell or several cells. For example, when the Call Drop Rate is high, it is possible that there are simultaneously three reasons: 1. The quality of Up-Link [1] is low. 2.The interference in the Up-Link is high. 3.The quality of Down-Link [1] is high. The simplest case is just the list of all the problems and reasons.

In the simple representation, all reasons affect the problem in the same grade, so it is impossible to identify individually significance of each reason. As a result, we use “level” to denote the grade. In addition, most guidelines are continuous and can’t be simply classified into “wrong” or “correct”. Fuzzy logic [3] is introduced to judge them. We use “degree” to represent in what degree the value makes the reason “abnormal”. “degree” is between 0.0 and 1.0. Therefore, both the original value and its degree must be recorded in the case. However, above records are still not enough to represent the case properly. For instance, the Qual_Up is abnormal when the value is too “low”, though the Interf_Down is abnormal when the value is too “high”. And in some other situations the value is abnormal when it is either too large or too small. So it is necessary to introduce the flag of each reason. The flag is defined to 1, 2, or 3 along with the above order.

Case 1: (intact)

{

problem: CallDropRateHigh

num	*reason*	*flag*	*level*	*value*	*degree*
1	Qual_Up	1	1	71.4%	0.8
2	Interf_Up	2	1	85.3%	0.9
3	Qual_Down	2	1	90.2%	1

}

1.2 Fuzzy Logic

Recently, many KDD researchers noticed the uncertainty of data [4]. However, during the first period of KDD, data are handled only with Boolean Logic that involves several disadvantages [4]:

1. The boundaries of subsets are precise, which means one data value belongs to either some subset or another.
2. The data values in the same subset are treated equally.

Fuzzy Logic was proposed to deal with these disadvantages. It is a superset of traditional Boolean logic. It extended to handle the concept of partial truth – the true values between "completely true" and "completely false" [3].

In this KDD domain, it is necessary to introduce fuzzy logic to handle the continuous data. If we use these data directly, then the very close value will be handled in different way, and the results will be vast and duplicated. So we introduce fuzzy logic in order to use the reliability of data instead of the real value. Fuzzy logic is everywhere in our methods.

1.3 KDD in Optimization

At first, we derived cases from experts. But vast data contains much information even the experienced engineers do not know yet, and the knowledge for optimization changes rapidly along with the network updating. So we consider KDD to find cases directly from

data. The optimization of GSM network is based on the data in OMC (Operation and Maintenance Center), so these data are also regarded as the data source of KDD. Our research focuses on two aspects: processing steps and mining algorithms.

1.4 Relation between Knowledge Management and Case Mining

Apparently, in such a system, knowledge in the knowledge base is the gather of cases. KM is responsible for deciding the structure of case, for example, reason number, reason name, reliability, reliability function, and so on. The cases can come from expert, and can also be the results of KDD.

In our research, the structure of case is determined through the cooperation of knowledge engineers and domain experts. It is easy to store and use knowledge. The most important thing is to record all related information in a case.

Case Mining is based on this structure. The result should contain all the elements required in knowledge base. If KDD can't obtain some information such as the index of cases, then there are still work remained.

Except for the information to find, there is still some information to be used by case mining in the case structure. We only define the form of reliability function in the structure, and the parameter(s) for each item should be calculated during the data preparing of KDD. When looking for cases, the function form defined in advance and the obtained parameter(s) are combined.

2. Steps of KDD Designed for Optimization

KDD process has been divided into several steps by different ways. A clear and accessible way brought forward by Dr. George H. John in 1997 is adopted in this thesis [5].

Considering the characteristics of engineering field, especially the optimization of GSM network, we compress the phases into 4 steps: 1.Data Selection. 2.Data Preparation. 3.Data Mining. 4.Knowledge evaluation. We design the tasks and criterions for each step.

Mining algorithms will be described in next section. Knowledge evaluation is not different from other fields. We only discuss the other tasks in this section.

2.1 Purpose Analysis

KDD has been used in telecommunication network, for example to see [6]. In that system, it was used to find the relation among alarm sequences, and the association rule was the concerned model. But in CBR system, knowledge is represented as cases, so the target of KDD is case.

2.2 Data Selection.

In the definition of Dr. George H.John, data extracting is an important step aiming at collecting the concerned data into one table.

But it is not appropriate to take all data into one table in the domain of GSM optimization, because one table means too much redundancy in many situations. In addition, since the optimization always can be done cell by cell, we define the cell_id [1] (which is the identification of one cell) as the primary key of each row.

We define two table types:

1. Every field is defined only by the cell_id and the column name. For example, the

TRX_NUM (number of Transceiver and Receiver) of one cell is unrelated with other field.

2. Every field is defined not only by the cell_id and the column name, but also by another new key or several other new keys. For example, see table 1.

cell_id	trx_id	Erl_DL	Erl_UL
10692	1	4.7133	5.0111
10692	2	***3.6666***	3.5455
10692	3	4.6453	3.9987

Table1 example of the second table type

The bold and italic item means the traffic of up link in the TRX [1](Transceiver and Receiver) numbered 2. Because the number of TRX in different cell may be widely discrepancy, it will cause much redundancy if each TRX is represented by one item.

2.3 Data Preparation

Because of the close relation of Data Cleaning and Data engineering, we combine them into one step. And fuzzy logic is introduced to handle the uncertainty of data.

The purpose of this step is to take all the data into many items that are discrete with each other, and obtain the judge function of each item.

2.3.1 Distribution of Items

Because all data are arranged into different items, so each item obeys Gussian normal distribution, according to the famous Levy-Lindberg theorem [7], which points out that if variable X is decided by large numbers of individual little factors and each factor can only act on it slightly, then X obeys normal school approximately.

The number of samples is the number of all cells in the optimized network. For the convenience of validating, these data are divided into several sample sets $X_i(1 \le i \le m)$, in which m is the number of samples.

Normal school is decided by the mean $E(X)$ and the variance $D(X)$.

2.3.2 Data Cleaning

The mean of one item may be affected heavily by the very big or very small value especially when the abnormal values are seldom. Take the item “CONG_CHANNEL” as an example, which expresses the channel congestion rate. In most situations, its value is 0; however, there is averagely several very large values such as “33.33333”. If we calculate the mean directly, the result will be much higher than 0, and the value “0” will possibly judged as abnormal (for the judge method, to see 2.3.3 and 2.3.4). Obviously it is unreasonable, because the abnormal value affects the parameter especially when n is not large enough.

Two methods are used to reduce the effects of abnormal values. First is that several largest and smallest values are dropped. Second is that we use the “median of sample” instead of $E(X)$. The median of sample is expressed by $\tilde{X}_n$, which is defined as follows:

$$\tilde{X}_n = F_n^{-1}\left(\frac{1}{2}\right) = \inf\left\{x : F_n(x) \ge \frac{1}{2}\right\},$$

in which F is the approximate distributing function of X derived from the sample.

Then in the above example, the final $\tilde{X}_n$ and $D(X)$ will both be 0, and any value larger than 0 will be judged abnormal.

There are still some useless values in some items. For example, in item "MAX_POWER" value "0" means there is no report. So the data equal to 0 in this item should be dropped in the step 2 of KDD.

2.3.3 Verifying

After $\tilde{X}_n$ and $D(X)$ was obtained, we use the veracity, validity, and consistency [8] to judge whether the parameters are correct.

Let μ be the genuine mean of X and δ be the genuine variance of X.

Veracity is defined as $E(E(X_i)-\mu)=0$ which means there is agonic between estimated mean and real mean. However, the criterion "0" is too strict. We define ε as a very small number, which value is decided according to the requirement of verifying. Then above criterion is revised as $E(E(X_i)-\mu)\prec\varepsilon$.

Validity is defined as $E\left[(E(X_i)-\mu)^2\right]=\delta$, which means the variance is small. Similarly, this criterion is revised as:

$E\left[(E(X_i)-\mu)^2\right]-\delta\prec\varepsilon$.

Consistency is defined as $\lim_{n\to\infty}P(|E(X_i)-\mu|\geq\varepsilon)=0$ which means that when the number of samples increases, the estimating parameter is also true.

We use $E(X)$ and $D(X)$ of all the data as genuine value to do this test.

2.3.4 Boundary Decision

The borderlines are decided according to the characteristic of normal school. We suppose that the value happening at probability less than 5% is abnormal with degree 0.9. The value equal to the mean is abnormal with degree 0.

	X	Y
Boundary1	$\mu+\delta$	0.6
Boundary2	$\mu+\delta*2$	0.95

Table2 Boundaries when flag=1

Let $\mu = E(X)$, $\delta = D(X)$. According to normal school, the probability of that value is larger than $\mu + \delta$ is 15.87%, and the probability of that value is larger than $\mu + \delta * 2$ is 2.275%.

Then when flag is equal to 1, the boundaries are shown as table2.

	X	Y
Boundery1	$\mu - \delta * 2$	0.95
Boundery1	$\mu - \delta * 2$	0.95
Boundery3	$\mu + \delta$	0.6
Boundery4	$\mu + \delta * 2$	0.95

Table3 Boundaries when flag=3

2.3.5 Determination of Flag

When flag is equal to 1, the function is shown in fig1 (a). When flag is equal to 2, the function is shown in fig1 (b).

At first, the flag of every item is decided artificially. But it is bored and unfaithful. So we design an algorithm to find out it.

The key idea is that at the beginning, we set the flags of all items to 3, and then the boundaries are changed as table3.

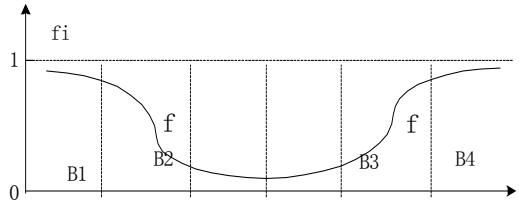
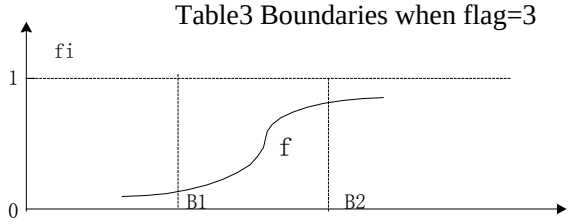


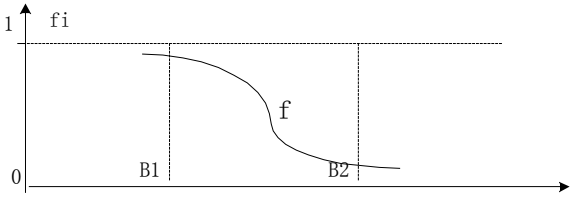
fig2 fi when direction type=3

The function is shown in fig2.

In fig2, the lowest point of f_i is corresponding to μ, because of the symmetry of the two functions.



(a) when low boundery<high boundery



(b) when low boundery>high boundery

fig1 fi using S function

Consider that only when there are some problems in the GSM network, the abnormal items will be concerned. So we scan all the cells that have problems and record the abnormal items as well as corresponding flags. Afterwards we check the flag of each item and determine its final flag. The algorithm is showed as algorithm1.

3 Algorithm for Mining Cases

3.1 Denotations

For the convenience of description, a set of denotation is defined.

In the optimization for GSM network, the problems concerned is finite, such as Call Drop Rate High, TCH (Traffic Channel) Congestion Rate High, and so on. Formally, let $P=\{p_1, p_2 \dots p_n\}$ be the problems. For each problem, there is a function to judge it. Let $F=\{f_1, f_{2\cdots} f_n\}$ be the functions and $f_i(1\le i\le n)$ is the judge function of p_i, where n is the number of concerned problems.

The result of a function is "correct" or "wrong", and represented by $R=\{C,W\}$, where C means correct, and W means wrong.

For each cell, the number of problems may be different. Let $S=\{S_{ij}\}$ be the status matrix coming from P. When problem $P_j(1\le i\le n)$ exists in $cell_i$, then $S_{ij}=1$, else $S_{ij}=0$.

When trying to solve problems, there are many reasons to be considered. Let that $R=\{r_1, r_2 \cdots r_m\}$ is the reasons, where m is the number of reasons. For each cell, all the reasons should be judged. Let $G=\{g_1, g_2 \cdots g_m\}$ be the functions and $g_i(1\le i\le m)$ is the judge function of r_i. Let $CS=\{CS_{ij}\}$ be the reasons of cells. When $reason(i)$ is correct in $cell(i)$, then $CS_{ij}=1$, else $CS_{ij}=0$.

Algorithm1:
Begin: Set direction type of all items to 3
Initialize a matrix $M(m*n)$
i =1
While ($i\le m$)
  if (there are problems in $cell_i$)
    j =1
    While ($j\le n$)
      if ($item_j$ of $cell_i$ is abnormal)
        $M(i,j)$ =direction of $item_j$
      else
        $M(i,j)$ =NULL
      j ++
  i ++
j =1
While ($j\le n$)
  if (direction types of all $item_j$ are 1)
    Set direction type of this item to1
  else if (direction types of all $item_j$ are 2)
    Set direction type of this item to2
  else
    Set direction type of this item to3

The purpose of KDD is case, which can be represented by $CASE=\{r_k, r_{k+1}, \cdots r_s\}\rightarrow\{p_l, p_{l+1}, \cdots p_t\}$, where t is the number of problem in this case, s

is the number of reason causing the problems, and k and l are both integer.

Because the number of problems is finite and will not be large, we focus on finding the reasons.

3.2 Finding problems of each cell

First of all, which cell has problems must be found. There can be one problem or several problems in a cell. The task to find problems of each cell can be described as follows:

$Input:\quad P, F \qquad\qquad Output:\quad S$

The algorithm is to go through P of every cell, and record the problems.

3.3 Finding reasons

Only the reasons in the cell with problems should be found. The task to find these is described as follows:

$Input:\quad P, F, S \qquad\qquad Output:\quad CS$

The algorithm is to go through all the cells with problem, and record the reasons in each cell.

3.4 Clustering to decrease the reason list

It often happens that once a reason exists, several other reasons also exist, because not all relations of the reasons have been known yet. So using clustering to find these relations can decrease the reason list and make the result simpler and more useful.

Traditional clustering method is to define a distance function, and for each item, find which items are clearest to it [9]. The key idea can be seen as “from little to more”. It will go through database for several times equal to the number of final subsets. Since going through database will take much time in most situations, this method is time expensive.

To decrease the spent time, a new algorithm is proposed. Generally, the reasons and the problems of one cell are recorded in one row, that is to say, one row corresponds all records of one cell. Following description will use cell and line by the same meaning. The algorithm’s key idea is to suppose all the items in the first cell belong to the same subset, and use the later lines to validate it. It can go through the database only once.

In 3.3, the idea of “suppose and prove” is used. The result of this algorithm is a non-crossing subsets list named “S_List”, in which each subset include one or more reasons, and each reason only belongs to one subset. The subset in S_List is expressed by “$SubSet$” plus its serial number. “$tmpSubSet$” and “$NewSubSet$” both represent temporary subset which maybe dropped or added into S_List. First of all, suppose that all the reason1 in the row1 exists altogether and put them into $SubSet_1$. Then go to following line and test all the former subsets. When some reasons of one subset are in the new row and others are

not in, divide this subset into two subsets.

$Algorithm2$:

i =1, j =1

$While(j \le m)$

DO (put R_j into $SubSet_1$)

DO (put $SubSet_1$ into S_List)

j ++

While ($i \le$ number of cells with problems)

i ++;

j =1;

While ($j \le$ number of problems in $cell_i$)

DO (put p_j into *NewSubSet*)

j ++;

k =1;

While ($k \le$ Subsets number in S_List)

j =1;

While ($j \le$ problem number in *NewSubSet*)

if ($SubSet_{kj}$ exists in $cell_i$)

DO (Remove the problem from *NewSubSet*)

else

DO (Remove the problem from $SubSet_{kj}$)

DO (Add the problem into *tmpSubSet*)

if (problem number in $SubSet_k$ = 0)

DO (Remove $SubSet_k$)

if (problem number in *tmpSubSet* > 0)

DO (Add *tmpSubSet* into *S _ List*)

if (problem number in *NewSubSet* > 0)

DO (Add *NewSubSet* into *S _ List*)

3.5 An example

For example, look at follow table:

cell	r1	r2	r3	r4	r5	p1	p2	p3
1	1	1	1	0	0	1	0	0
2		1	1	0	0	1	0	0
3	0	0	0	0	0	0	0	0
4	1	0	0	1	1	1	1	0
5	0	1	1	1	0	1	0	1

$P = \{p_1, p_2, p_3\}$. $F = \{0,1\}$.

After clause 3.1,

$S = \{\{p_1\},\{p_1\},\{\ \},\{p_1, p_2\},\{p_1, p_3\}\}$ is obtained.

After clause 3.2,

$CS = \{\{r_1, r_2 r_3\},\{r_2, r_3\},\{\ \},\{r_1, r_4, r_5\},\{r_2, r_3, r_4\}\}$ is obtained.

Now algorighm2 is used. The number in the first place of each line means that the reasons of the corresponding cell has been considered.

1. (r_1, r_2, r_3) /*Initialization*/
2. $(r_1),(r_2, r_3)$ /*Because r_2 and r_3 exist simultaneously in the second row but r_1 does not exist*/
3. No change happens because the $cell_3$ has no problem
4. $(r_1),(r_2, r_3),(r_4, r_5)$
5. $(r_1),(r_2, r_3),(r_4),(r_5)$

CASE can be extracted directly from *CS* and *S* , and the reason list is simplified with clustering: $CASE1 = \{r_1, r_2\} \to \{p_1\}$, $CASE2 = \{r_2\} \to \{p_1\}$,

$CASE3 = \{r_1, r_4, r_5\} \to \{p_1, p_2\}$, $CASE4 = \{r_2, r_4\} \to \{p_1, p_3\}$

3.6 Redundancy Decreasing

There are two kinds of redundancies in the cases.

1.There may be a case C1 where the problem set is combination of problems of cases C2…Cn, and the reason set is combination of reasons of same cases. Then case C1 is just combination of cases C2…Cn. For example, see $CASE2$ and $CASE4$ in clause 3.5. The algorithm to erase this kind of redundancy is very simple and not listed in this paper.

2. Since the threshold for judging one item is empirical, there may be some reliability

too close to the threshold so that in some cases the item exist but in other cases the item does not exist. For example, see $CASE1$ and $CASE2$ in clause 3.5. We call $CASE1$ "SuperCase" of $CASE2$, and call $CASE2$ "SubCase" of $CASE1$. To remove this redundancy, algorithm3 is used.

$Algorithm3$

interval = some little value
while (there are cases becoming inexistent
and there still remains cases being SubCase of other case)
Threashold = old Threashold + interval
Calculate *CASE* again
If (some cases become inexistent)
These cases are redundancy

Notice that the interval selection is empirical. According to our experiment, the threshold should be set to 0.4~0.5 first, and the interval should be set to 0.005~0.015.

3.7 Level Calculating

The significance of reasons can be different in the same case. In fact, the level of reason can be calculated through the algorithm3 by recording the disappeared cases. This can be achieved by algorithm4.

$Algorithm4$

interval = some little value
while (there are cases becoming inexistent
and there still remains cases being SubCase of other case)
Threashold = old Threashold + interval
Calculate *CASE* again
If (some cases become inexistent)
These cases are redundancy
Increase the level of reason belonging to the redundant case in its SuperCase

3.8 Dimensionality Handling

In GSM network, the criterion to judge some item is not simple. The dimensionality often exists. For instance, when Handover Success Rate is concerned, there will be many values, since there are many adjacent cells of one cell.

If the dimensionality is handled only by mining algorithm, there will be much burden, since mining algorithm must try all kinds of relationships. Instead, we try our best to use the knowledge of expert to handle it during Data Preparation. That is to say, let experts decide which items represent dimensionality. The number of dimensionality items can be one or more than one. Then when the reliability is being calculated, there are two possible relation: "AND" or "OR". In "AND", the end reliability is the product of all reliabilities. In "OR", the end reliability is the max reliability.

4. Result Summarization

We use a set of data consisting 249 cells from real GSM network.

At first, we prepare the data, and decide the form of judge function. It is tested by the data and acknowledged by domain experts.

There are 106 cells with problems when the threshold is set to 0.4, though there are 81 cells with problems when threshold is set to 0.5. Finally, the threshold is set to 0.4 for more precise (to increase the criterion of optimization). That is reasonable, since the network we concerning is a very busy network and one cell affect many adjacent cells. Then we record the reasons in the cells with problems. Of course, there is at least one reason for each cell.

After clustering, we deduce the number of reason from 190 to 176. There are 9 clusters consisting more than one element.

If each cell with problems is regarded as a case, then there should be 106 cases. However, the number is deduced to 56 after redundancy decreasing.

In the former methods of case mining, only the simple lists of reasons and problems are recorded [11]. They didn't handle the uncertainty of data. Our result consists more information and is more useful.

Reference

[1] ETSI GSM Standard Series. http://www.etsi.org

[2] Li D.Xu: Case-Based Reasoning, IEEE potentials, the magazine for up-and-coming engineers, DECEMBER 1994, pp10-13

[3] Yager, R.R., and Zadeh, L. A.: "An Introduction to Fuzzy Logic, Applications in Intelligent Systems" Kluwer Academic Publishers, 1991.

[4] Michalis Vazirgiannis, Maria Halkidi: Uncertainty handling in the data mining process with fuzzy logic, Proceedings, The Ninth IEEE International Conference on Fuzzy System, pp393~398

[5] George H. John: Enhancements to the Data Mining Process, March 1997, pp8-18

[6] K.Hatonen,M.Klemettinen, H.Mannila, P.Ronkainen, H.Toivonen: Knowledge Discovery from Telecommunication Network Alarm Databases, The 12th International Conference on Data Engineering(ICDE'96), New Orleans, Louisiana, February/March 1996

[7] R. T. Larsen and M. L. Marz: An introduction to Mathematical Statistics, 1986

[8] Sun.RongHeng: Applied Statistics, Scientific Publishing Company, Chinese, 1998

[9] P.S.Bradley, Usarna M.Fayyad, O.L.Mangasarian: "Mathematical Programming for Data Mining: Formulations and Challenges", January 1998, pp4~5

[10] Clifford Grossner, P.Gokulchander, T.Radhakrishnan. "Revealing the structure of Rule-Based Systems". International Journal of Expert Systems, 1996, 9(2), p255~278

[11] Haihong Dai: "Discovery of Cases for Case-Based Reasoning in Engineering", Proceedings, Asia-Pacific Software Engineering Conference and International Computer Science Conference, 1997, pp89-96