The Research on Analyzing Risk Factors of Type 2 Diabetes Mellitus Based on Improved Frequent Pattern Tree Algorithm

Wei Zhe[1,2, a], Ye Guangjian[1, b]

[1] School of Electrical Engineering and Information Engineering, Lanzhou University of Technology, Gansu, Lanzhou 730050,China

[2]Lanzhou General Hospital, Lanzhou Military Area Command, Gansu, Lanzhou 730050, China

[a]weizheu@163.com, [b]11445697@qq.com

Keywords: data mining; Apriori Algorithm; Association rules; FP-tree Algorithm

Abstract. *Purpose:* We do it to improve the low efficiency in analyzing risk factors of type 2 Diabetes Mellitus by Apriori Algorithm. **Method:** We use the patients' data from the information department of one tertiary referral hospital in Lanzhou which include course note of disease and their health record form January 2009 to March 2014.We find out that the improved FP-tree Algorithm analyzes risk factors of type 2 diabetes better. And we analyze the efficiency by programming improved FP-tree and Apriori Algorithm with C# **.Result:** We can analyze the chart of time and number of records, time and support degree, main risk factors. **Conclusion:** The improved FP-tree Algorithm can be used to analyze the risk factors of Diabetes Mellitus and holds a higher efficiency.

Introduction

Diabetes Mellitus is considered to be caused by the secretion of insulin and the role of defects caused by chronic high blood sugar with carbohydrates, metabolic disabled of fat and protein chronic disease characterized. Type 2 Diabetes Mellitus, which is called non-insulin-dependent Diabetes Mellitus as well, dues to insulin resistance with relatively lack of insulin secretion, and Type 2 Diabetes Mellitus, which has the characteristic of adult lesion, slow process, light degree, is not together with lesion of β cells and holds most of all the numbers of Diabetes Mellitus patients[1]. It is counted that the number of global patients with Diabetes Mellitus was only 30 million in 1985 which increased to 135 million in 10 years, and it reached 171 million in 2000. Even it is forecasted to overwhelming 300 million before 2025. The so large number and quicker increasing speed shows the importance of research on Diabetes Mellitus.

We find defects of Apriori Algorithm in researching on mining association rules of Type 2 Diabetes Mellitus risk factors. First, Apriori Algorithm has to used to scan the database once when generate a frequent item set each time. And second, when generating k candidate item sets from (k-1) frequent item sets, it will product many candidate item sets which is unnecessary later and have a long time in data mining of risk factors and a low work efficiency. We propose a modified Frequent Pattern Tree Algorithm to analyze the risk factors of Type 2 Diabetes Mellitus with the characteristic of large data and variable[2].

Structuring the Mining Rules

Frequent Pattern Growth Algorithm

Frequent Pattern Tree Algorithm is a kind of basic method without candidate item sets. The improved process and developed tree form is called Frequent Pattern Growth Algorithm. Frequent Pattern Growth Algorithm bases on Divide and Conquer: we first compress the original data of database into one Frequent Pattern Tree, and keep the association information. Then we divide the database by conditions, and each frequent item is connected with one condition[3].

The Frequent Pattern Growth Algorithm can be divided into two parts: structuring the tree form based on original database and recurrently mining in the tree. The first step equals the one which

produces candidate item set L_1 in Apriori Algorithm. This step of Frequent Pattern Growth structures the basic tree form and is the main step which is called the first scanning of database. Next we scan the database again using the frequent item of database to structure the tree form to find out nodal points and add them into the tree form orderly. In the second step, we find some short patterns met the conditions by recurrently searching, and get long frequent patterns by connecting the short patterns with suffixes. In classic Apriori Algorithm, it is defined of connecting that: The connection step: we connect the L_{k-1} item set with the candidate item set C_k to find L_k. We assume l_1 and l_2 are the item sets of item set L_k , so we can define l_i [j] as the jth item, then we do the connection. The connection demands l_1 and l_2 from L_k can be connected, and if (l_1[1]= l_2[1])^ (l_1[1]= l_2[2])^ ……^(l_1[k-2]= l_2[k-2])^ (l_1[k-1]< l_2[k-1]), when l_1[1], l_1[2] ……l_1[k-1], l_1[k-2] are the connection result item sets. Frequent Pattern Growth Algorithm has the same method that we connect the short patterns with the suffixes to find out the frequent item sets under a certain condition. However, differently from the classic algorithm, we search the conditional frequent item sets instead of scaning the database each time based on the Frequent Pattern Tree structured in first step. Usually the conditional frequent item sets are much smaller than database, and in this way we can save much searching memory and improve the efficiency of algorithm[4,5,6].

We collect more than 30 thousand course notes of disease and health records of patients with Type 2 Diabetes Mellitus from the information department of one tertiary referral hospital in Lanzhou, and mine the data to find risk factors of Type 2 Diabetes Mellitus. We choose 15 risk factors: gender, age, education level, body mass index (BMI), waist hip ratio (WHR), personality, trauma history, drinking, tea, smoking, sleep, exercise, income level, occupation, meals on time. The data is operated in this way: we change the chosen 15 risk factors into 44 attribute values, and the more than 30 thousand data who have 44 attribute each one make the process large and complex if we use the classic Apriori Algorithm and it costs a lot I/O switches and time as well. So we try to use Frequent Pattern Growth Algorithm[7].

Realization of the Improved Frequent Pattern Tree Algorithm

We improved the Frequent Pattern Tree Algorithm based on the tree form from original database, then mine frequent patterns in Frequent Pattern Tree to improve into Frequent Pattern Growth Algorithm. Frequent Pattern Tree is a kind of constringent data structure includes 3 parts.

First, the tree includes a null, a prefix subtree item as a child of null, a frequent item header table. Second, each nodal point of the item prefix subtree structures by 3 parts: item name, count, node link. They orderly express the name of item of nodal point, the number of objects in the trajectory by the end of nodal point, nodal points which have the same name guiding the Frequent Pattern Tree. Third, each item is structured by two areas in frequent item header table: item name and nodal point head. Nodal point head guides to all the first nodal points which have the same item name.

Frequent Pattern Growth Algorithm can be realized in this way:

Input: Database *D*, the minimum support threshold defined as *min_sup* which is set as actually required.

Output: *Frequent Pattern Tree*.

1. Scan database once to get frequent item sets and the support degree of each frequent items. Sort all the frequent items by their support degree in descending order, then get frequent item table *L*.
2. Create null T, defined as *null.*
3. For

   Sort all the frequent items by *L,* express the frequent item table as *[p|P]*. *p* is the first element and *P* is the item table of frequent item table based on the other elements without *p.*
4. Call *insert_tree([p|P],T);*
5. End for.

Input: *Frequent Pattern Tree*, item *a*(initial value is null), minimum support defined as *min_sup*.

Output: Frequent item set *L* of database *D*.

1. Initial value of *L* is null.

2. if *Tree* only includes single path *P*, then
2.for each group in *P* defined as β.
3. Create item set $\beta \cup \alpha$, and the support degree equals to the *min_sup.*
4. Return $L=L \cup$item sets which support degree is bigger than *min_sup.*
4. else//more than one path.
5. for each frequent item α_j of *Tree*'s head table.
6. Creat a item set $\beta=\alpha_j \cup \alpha$, support degree less than α_j's.
7. Structure the conditional pattern *B* of β, and structure $Tree_B$ of *Frequent Pattern Tree* based on conditions of β.
8. if $Tree_B \neq \emptyset$ then
9. call *Frequent Pattern Growth*($Tree_B$, β)
10. end if
11. end for
12. end if
5. create a pattern $\beta=a_i \cup \alpha$, the support degree equals to $a_i support$.
6. structure conditional pattern base of β and conditional $Tree_B$ of *Frequent Pattern Tree* .
7. if $Tree_B \neq \emptyset$ then
8. call *Frequent Pattern Growth*($Tree_B$, β).

Conclusions

We use C# to program Apriori Algorithm and Frequent Pattern Growth Algorithm to test the efficiency and performance, and analyze the risk factors of Type 2 Diabetes Mellitus by data mining with the two models and the preprocessed data. The equipment of the experiment is: Intel i5 CPU, 4G RAM, Win5 system. We compare frequent item sets with time and number of records, time and support degree, and the 3 figures show the result.

The 1st figure shows the relation of operating time and the number of records. We can see that the improvement of Frequent Pattern Growth Algorithm to the efficiency is not very obvious when the number of records is small, and even has a lower efficiency than classic algorithm. However, the efficiency of Frequent Pattern Growth Algorithm increases into much higher level of efficiency as the number of records increasing. The reason is that the Frequent Pattern Growth Algorithm need scan the database twice and then rank the item orderly under the same condition and same minimum support degree. These two steps cost more time than the classic algorithm. In other word, when creating candidate item sets L_1 and L_2, and even the database only has frequent item sets L_1 and L_2, the classic algorithm gets the much higher efficiency. We can get the method that Frequent Pattern Growth Algorithm fits to analyze and operate much larger and more complex data. In the same way we can see in figure 2, by the increasing of support degree, the efficiency of Frequent Pattern Growth Algorithm is getting lower compared with the classic Apriori Algorithm. Because both two algorithm create less frequent item sets when the support degree gets bigger. At the same time, it costs more time when the Frequent Pattern Growth Algorithm structures the tree form than the classic algorithm. However, we define support degree in a low level actually which is about 0.2, then we can see from the figure Frequent Pattern Growth Algorithm has the much higher efficiency than the classic algorithm, besides the value of support degree shows the complex level of a database. So we keep the value low to find risk factors of Diabetes Mellitus as many as possible. At the same time, the so large and complex data increase the superiority very much.

We can get the information from figure 3 and figure 4: men with drinking is a very dangerous factor, and women with high BMI and WHR have a very high risk to get Diabetes Mellitus. Besides exercise with reduce the risk obviously. The result is similar with the one from classic Apriori Algorithm, and with a very high efficiency as well.

References

[1]Li Wucheng, Wang Guanquan, Jin Ke. Analysis on Risk Factors of Diabetes Mellitus with the Complication of Hypertension[J]. The Journal of Practical Medicine, 2010, 26(17): 3180-3181.

[2] Diego GC, Joel O.J.Q. Liver Cirrhosis and Diabetes: Risk Factors,Pathophysiology, Clinical Implications and Management[J]. World Journal of Gastroenterology, 2009, 3:141-143.

[3] Wild S, Roglic G,et al. Global prevalence of diabetes-Estimates for the year 2000 and projections for 2030[J]. Diabetes Care, 2004, 27(5): 1047-1053.

[4] Gruca. Improvement of FP-growth Algorithm for Mining Description-Oriented Rules[C]. 3rd International Conference on Man-Machine Interactions (ICMMI), 2014, 242: 183-192.

[5] Shao Fengjing, Yu Zhongqing. Principles and Algorithms of Data Mining[M]. Science Press , 2009, 8: 1-2.

[6] Totad SG, Geeta R.G. Batch Incremental Processing for FP-tree Construction Using FP-growth Algorithm[J]. Knowledge and Information Systems, 2012, 11, 33(2): 475-490.

[7] Hu Keyun, Tian Fengzhan. Methods and Applications of Data Mining[M]. Tsinghua university press, 2008, 4: 110-115.

[8]Jiang Shengyi, Li Xia. Principles and Applications of Data Mining[M]. Publishing House Of Electronics Industry, 2011, 8: 150-160.

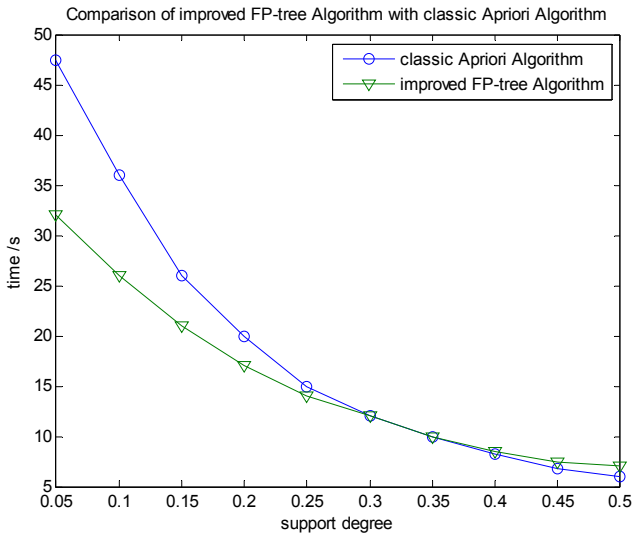

Fig.1 Comparison of time and number of records

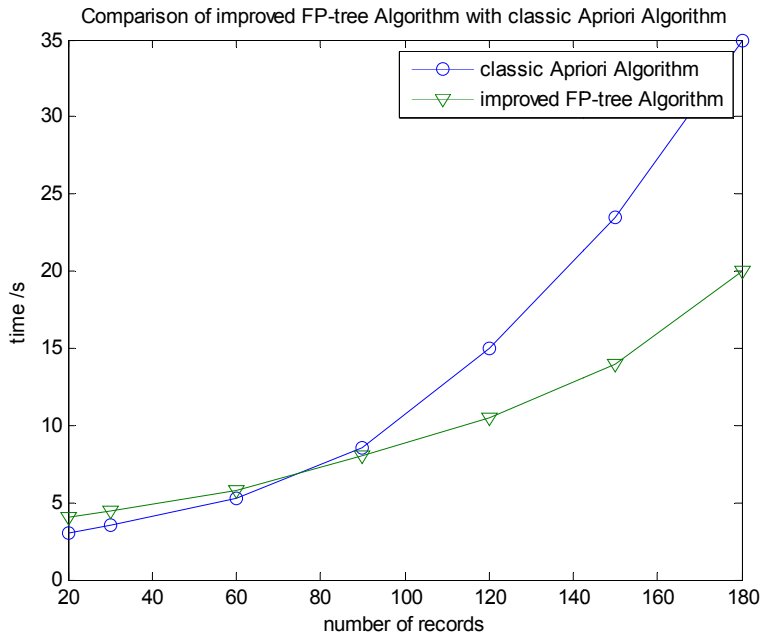

Fig.2 Comparison of time and support degree

DM	WHR BMI Personality Drinking	21.248	72.222
DM	Gender WHR	23.445	43.385
DM	WHR Education	20.956	44.154
DM	Drinking Gender	25.686	55.472
DM	Gender Drinking BMI	23.467	68.359
DM	WHR BMI Gender Drinking	21.248	72.222

Fig.3 Partial result of Data Mining

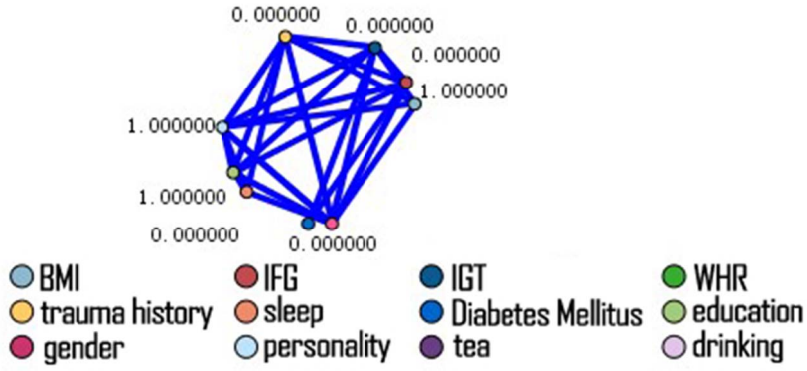

Fig.4 Partial result of Data Mining