TEST Engineering & Management

July - August 2020
ISSN: 0193-4120 Page No. 5739 - 5744

Modern Logistics Route Optimization Considering Frequent Subgraph RFID Route Mining Algorithm

Jinjin Guo[1,*], Huiying Zhang[1] and Gui'e Sun[1]
[1]Chongqing Vocational College of Transportation, Chongqing, China, 402247

Article Info
Volume 83
Page Number: 5739 - 5744
Publication Issue:
July - August 2020

Article History
Article Received: *25 April 2020*
Revised: *29 May 2020*
Accepted: *20 June 2020*
Publication: *28 August 2020*

Abstract

The modern logistics route optimization technology that considers frequent subgraph RFID route mining algorithm effectively solves the problem of logistics route. The quantitative increase of the mining algorithm leads to the intrinsic problem of symmetrical composition. The problem is solved by the frequent subgraph FRID route mining algorithm Provide logistics path optimization. The successful development of modern logistics route optimization, taking into account the frequent sub-graph RFID route mining algorithm, can well show the direction of items and bring convenience to modern logistics.

Keywords: *Radio Frequency Identification, Logistics, Frequent Routes, Online Analysis and Processing, Frequent Subgraphs, Data Mining, Algorithms;*

1. Introduction

The frequent subgraph RFID route mining algorithm is a path tracing algorithm, which is a path tracing (path tracing is the organization of path tracing nodes like a tree, in fact an inverted tree) core algorithm ID3 improvement algorithm, So basically understand half of the path tracing construction method to construct it[1-3]. The path tracing construction method is actually to select a good feature and split point as the classification condition of the current node each time.

The frequent subgraph RFID route mining algorithm uses a compact data structure to store all the information needed to find frequent itemsets. Algorithm: Compress the databases that provide frequent itemsets together to retain itemset related information, and then divide the compressed database into a set of condition databases (a special type of projection database), and each condition database is associated with a frequent itemset . K-Means is one of the most classic and most widely used clustering methods. There are many improved models based on it. The idea of K-Means is very simple. For a clustering task (you need to specify how many classes are clustered, of course, according to natural thinking, you should not need to specify the number of classes. This problem is also a topic worth studying for current clustering tasks), First randomly select K cluster centers, and then repeatedly calculate the following process until all cluster centers do not change (the cluster set does not change): Step 1: For each object, calculate the similarity between each cluster center and Put it into the most similar cluster[4-6].

In recent years, logistics has expanded to every household. my country has also increased RFID investment to improve logistics efficiency. The concept of graphs is introduced through frequent subgraph mining algorithms, and logistics information is displayed in the form of graphs. Users can choose Getting the frequent subgraphs you want is not only an interesting expression, but also saves space for modern logistics paths.

2. Construct RFID logistics diagram

The RFID database is a collection of RFID data tuples in the form of ?EPC, (a1,...,an), (m1,...,mk), path?. Among them: EPC (electronic product code) represents the electronic product code, which is globally unique and provided by relevant international organizations; (a1,...,an), (m1,...,mk) have the same meaning as traditional databases, and the names are changed to Non-path dimension attribute value and non-path metric value; path represents path information. Compared with the traditional multi-dimensional database, the RFID database adds path data and has a globally unique identifier EPC.

Traditional data warehouse technology does not consider the relationship between different tuples, while RFID data tuples contain structural information such as paths. Aiming at these structural data, this paper applies the graph-based OLAP framework to RFID logistics management, and establishes a graph model for the transportation path of items.

2.1. Construct a logistics diagram

After the RFID reader senses the tag, it will generate stream data composed of tuples (EPC, location, time), where EPC represents the electronic product code of the tag, location represents the location of the reader, and time represents the time when the reading occurred. If you merge all these tuple records, you can get the path information of a particular item. After data cleaning, all paths will consist of segments (location, time_in, time_out). In the process of digging frequent paths to find the moving trend of items, you can ignore the specific time when the items arrive or leave a certain location. You only need to care about the stay time of the item at a certain location. The final path will be composed of locations (location, duration). Furthermore, duration not only stays at the level of 0/second, but also a higher level of abstraction can be used to express the residence time and clustering path.

The path database is a collection of tuples of the form -d1,d2,,,dm:(l1,t1)...,(lk,tk)-. Among them: d1, d2,,, dm are non-path dimensions (the attribute value does not change with the change of the position of the item), which is used to describe the nature of the item, such as manufacturer, trademark, price, production date, etc.; (li,ti) means the item moves The time spent at position li during the process is ti.

Table 1 shows a path database, which has two non-path dimensions, product dimension (product) and brand dimension (brand). gc means factory, zx means logistics center, kc means truck transportation, ck means warehouse, sd means store, and sy means cash register. The unit of time is 0 per day.

Table 1. A sample route database.

Logo	Product	Trademark	Path
1	printer	HP	(gc,10)(zx,2)(kc,1)(sd,5)(sy,0)
2	printer	HP	(gc,5)(zx,2)(kc,1)(sd,10)(sy,0)
3	projector	Lenovo	(gc,10)(zx,2)(kc,1)(sd,10)(sy,0)
4	router	Cisco	(gc,10)(kc,2)(ck,10)
5	switch	HP	(gc,10)(zx,2)(kc,1)(sd,5)(sy,1)
6	switch	Cisco	(gc,5)(zx,2)(kc,2)(ck,10)
7	projector	HP	(gc,10)(zx,2)(kc,2)(sd,5)(sy,1)
8	projector	HP	(gc,10)(zx,2)(kc,2)(sd,10)(sy,0)
9	printer	Lenovo	(gc,5)(zx,2)(kc,2)(sd,10)
10	printer	Lenovo	(gc,5)(zx,2)(kc,1)(sd,10)(sy,1)

Different users have different understandings of the route. The manager of the store only cares about the details of the movement of the items in the store, and the person in charge of the transportation department also only cares about the details of the transportation, so different users will have different route views. . Figure 1 shows that the same path will have different path views. Among them, the path in the middle is the complete path; the path above is the path view used by the store manager, which merges the single location information during the transportation process; the path below is the path view used by the person in charge of the transportation department, which merges the store Single location information.

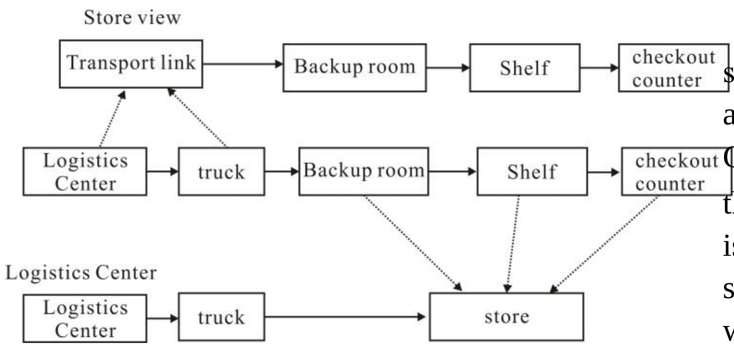

Figure 1. Path view: The same path has two levels of abstraction.

The abstraction level of /store 0 and /transportation link 0 in Figure 1 is higher than that of a single location. The concept of lattice can be used to express the abstraction level of a certain path. The path abstraction level can be expressed in the form of a tuple (v1, v2,..., vk, tl), where vi represents the node in the location abstraction level, and tl represents the abstraction level of time information, such as "hour", "day", " Zhou" and so on.

Definition 2: Marked map and marked gallery. A labeled graph can be represented by the four-tuple G=(V(G),E(G),L(V(G)),L(E(G))), where V(G)=(v1,v2, …, vn) is the set of vertices of graph G, E(G)={ek=(vi,vj)|vi,vj∈V(G)}V×V is the set of edges of graph G, L(V(G))={lb(vi)|vi∈V(G)} is the vertex mark set of graph G, such as the vertex mark set {a,b,c,d,e,...} in Figure 1. L(E(G))={lb(ek)|ek∈E(G)} is the edge label set of graph G, such as the edge label set {x,x,y,...} in Figure 1. The database GDB={G,G,G,...,G} composed of u marked graphs is a marked library.

Definition 3: Isomorphism of labeled graphs. There are labeled graphs G and G'. If there is a mapping function f: V(G)→V(G'), it satisfies: ①vi∈V(G), lb(v)=lb(f(v));②(v,v)∈E(G),(f(v),f(v))∈E(G'),lb(v,v)=lb(f(v),f(vj)). Then it is said that G and G' are isomorphic, denoted as G≌G'.

Definition 4: subgraphs of the labeled graph. There are labeled graphs G and H. If there is a subgraph H' of H that is isomorphic to G, then G is called the subgraph of H, and it is denoted as G⊆H.

Definition 5: Graph support and frequent subgraphs. There is a library GDB={G,G,G,…,G} and a graph G', the support of graph G' in the library GDB sup(G')=|{H|H∈GDB and G'⊆H}| , That is, the number of graphs G that have subgraph isomorphism with G' in GDB. Given a minimum support min_sup, in the library GDB, subgraphs with support greater than min_sup are called frequent subgraphs. Let G' be a frequent subgraph, if there is no frequent subgraph G, so that G' is a subgraph of G, then subgraph G' is called the maximum frequent subgraph.

2.2. Graph-based online analysis and processing

On traditional relational databases, users can obtain interesting views through related OLAP operations from different perspectives, without the need to output a large amount of irrelevant data. Similar OLAP operations can also be used on the gallery mentioned in this article.

(1) Roll up On a certain attribute dimension of a snapshot, multiple snapshots with the same dimension value are overlapped through an aggregate function to achieve a higher level view. The above example is the logistics map established for each commodity in the warehouse each year. Since the logistics map has path information, when it is gathered, a set of qualified snapshots is first selected, and the sub-graph isomorphism test is performed on the corresponding map, namely Find the subgraphs that contain the same path, then merge the nodes with the same label (location) on the subgraphs, and the weights of the edges are added accordingly. For example, the transportation volume and times of running shoes on the f-d-t path from 2001 to 2004 can be obtained, without the need to show the results of each year to the user.

(2) Drill down to expand the high-level view into a low-level collection. For example, if you expand a general transportation map of all running shoes by year, you can get specific transportation information for each year.

(3) Slice/block select a subset of snapshots whose attribute dimension meets the threshold in the given

condition. If the user is only interested in the shipment of all products in 2001, he can select those snapshots whose time dimension is 2001 through slices.

3. Frequent subgraph mining algorithm on RFID logistics graph

When the number of generated logistics diagrams is large and the user is only interested in some of them, you can first find the atlas that meets the needs of the user from the atlas, and then dig on the target atlas. For example, in the above example, if the user wants to find the frequent route map of running shoes between cities, they only need to select the logistics map of running shoes in the original library, and then use the frequent map mining algorithm on the reduced atlas.

The main idea of the RFSM algorithm in this paper is to reduce the target atlas according to the needs of users, and then use the depth first search method (Depth First Search, DFS) to generate frequent subgraphs. Establish DFS dictionary order for each picture, and achieve the purpose of uniquely marking each picture with the smallest DFS code. Call extended graph patterns recursively, find their frequent descendants, until no new candidate subgraphs or frequency subgraphs are generated. The main steps of the algorithm are as follows:

(1) Determine whether the current DFS code is the smallest code. A graph may have several DFS codes, and only the rightmost extension is required for the smallest DFS code, because this extension will ensure the completeness of the mining results, thereby excluding non-minimal codes.

(2) According to the conditions given by the user, traverse the snapshot set to find the graph that meets the requirements, that is, the user is only interested in a certain type of product or the commodity flow information of a certain year, and only needs to dig in the relevant atlas.

(3) List all the rightmost extensions in the result of (2) and insert the search tree.

(4) Calculate the support degree of the extended mode.

The RFSM form of the frequent graph mining algorithm based on RFID logistics graph is described as follows:

Algorithm 1 is based on the RFID logistics graph mining algorithm RFSM. Input: DFS code s; original atlas D; target atlas PD; minimum support threshold min_sup; attribute value Q=(I1,I2,...,Ik).

Output: frequent graph set S.

```
method:
(1)S←∅;
(2) Scan D once, for any Gi ∈ Ddo
(3) If(I1,i,I2,i,…,Ik,i)=Q then
(4) Insert Gi into D';
(5) PD=D';
Call RFSM(s,PD,min_sup,S);
    Procedure RFSM(s,PD,min_sup,S)
(6) ifs is not the smallest DFS encoding then return;
(7) Insert s into S;
(8) Set C to empty
(9) Scan the PD to find out the expansion map of all s,
(10) If the expansion graph s'does not exist in C then
(11) Insert it into C and calculate its frequency;
(12) Sort C according to DFS dictionary order;
(13) Pattern graph p, do not less than min_sup in for C
(14) PD=Dp,
(15) Call RMine;
(16) Return;
```

Example 2 takes the above-mentioned warehouse item flow path as an example to construct a graph model and mine frequent graphs.

(1) Construct a graph model for the original database. For example, build its RFID library for the original database in Table 1, as shown in Figure 1.

(2) Construct a search tree. Take y as the starting vertex, denote it as p0, insert its DFS code into the 0th layer of the search tree as a node, the original atlas D is the logistics diagram of all products from 2001 to 2008, if the user gives Q=("Running shoes",

2004), then traverse the gallery once, search out all the pictures whose product dimension is running shoes and the time dimension is 2004, and get the current target gallery D1 (G1, G2 are a subset).

(3) Calculate the frequency sup(y) of y in D1, and save Dp0 in the node attribute. If sup(y)≥min_sup, then traverse D1 and find all the edges that make y extend to the right. In G1, yz is one of the extensions, denoted as p1, insert the new node z into layer 0, and insert the extended graph DFS code into the first layer as a child node of p0, because p0 is a subgraph of p1, then Dp1 is Dp0 A subset of, traverse Dp0 to calculate the frequency of p1, and get Dp1, if sup(p1)<min_sup, then pruning, and stop searching for its expansion graph; otherwise, continue to expand along the branch.

4. Experimental results and analysis

In order to further compare the mining location sequence and the time sequence to mine closed paths, the MCP algorithm and the basic algorithm based on Apriori for directly mining closed paths are two mining strategies. This article mainly compares the mining time of the same database with different minimum support and the mining time of different size path databases. The experimental environment is based on Windows XP SP2, the memory is 2GB, the CPU is PIV218GHz, and the experimental links are written in Visual C++610. The experimental data used is based on artificial synthetic data in an RFID logistics management system.

Experiment 1: Compare the time of the two strategies MCP under the same path database with different support. The experimental results are shown in Figure 2. The basic algorithm in experiment 1 is based on Apriori's ideas, and some measures to check the closure are added. The route database of experiment 1 has a number of routes of 100,000, with a support degree of 015%~115%.

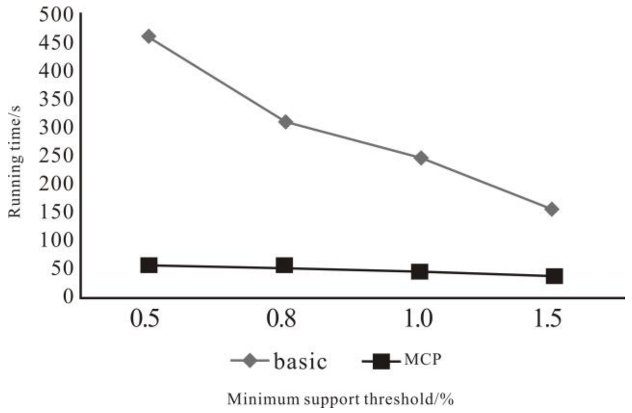

Figure 2. Comparison of running time for different minimum support thresholds.

As shown in Figure 2, when the support is small, the maximum length of the frequent path is correspondingly larger, and the advantage of the MCP algorithm is more obvious; when the support is large, the maximum length of the frequent path is smaller, and the corresponding basic algorithm scans the database times Reduce, the advantage of MCP algorithm declines. For MCP itself, because no matter how the maximum closed path length changes, the algorithm scans the database only twice, so the running time of the algorithm is small.

Experiment 2 compares the time of two MCP strategies for different sizes of path databases. The minSup=0110 used in this experiment, the size of the path database ranges from 100,000 paths to 500,000 paths, and the experimental results are shown in Figure 3. As the size of the database increases, the time to scan the database will increase accordingly, so the algorithm running time will also increase.

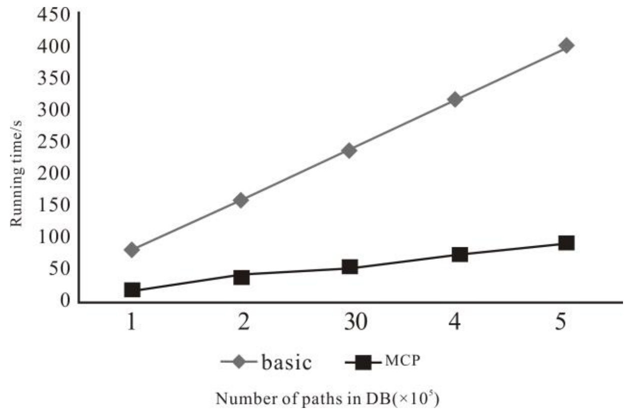

Figure 3. Running time under different path counts.

Experiment 3 tests the scalability of the MCP algorithm, and the experimental results are shown in Figure 4. The minSup used in the experiment is 0105 and 0110 respectively, and the number of paths contained in the path database is expanded from 500,000 to 2,000,000. It can be seen that the running time increases linearly with the increase of the number of routes in the route database.

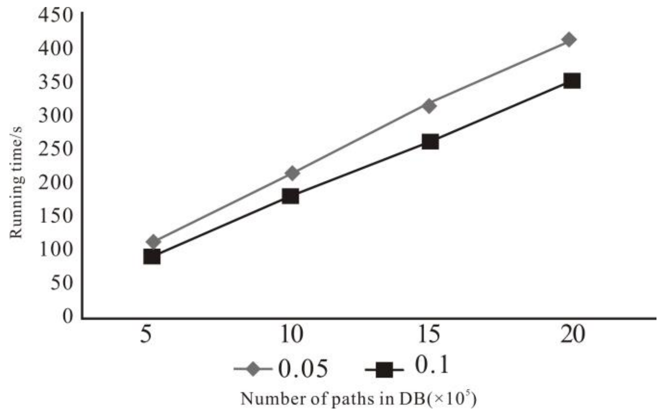

Figure 4. Running time of MCP under different path database sizes.

5. Conclusion

Aiming at the huge logistics database of RFID, most of the current researches are OLAP based on data cubes, using flow cubes to mine frequent routes, but they cannot be efficiently applied to the massive route data collection of RFID. This paper proposes to use the graph model to store the logistics path, perform graph-based OLAP operations, and provide users with multi-level views. After establishing a graph model for RFID logistics, the RFSM algorithm can be used to effectively mine RFID frequent graphs. The difference with the traditional frequent subgraph mining algorithm is that RFSM does not mine on the original atlas, but selects atlas that meets the given dimension value according to the needs of users, and executes the algorithm on the atlas, saving a lot of time and Space has high efficiency.

Acknowledgments

Chongqing vocational college of transportation study on layout optimization of energy replenishment facilities for electric vehicles (CJSK201916).

References

[1] Li, J., Zou, Z., & Gao, H. (2012). Mining frequent subgraphs over uncertain graph databases under probabilistic semantics. Vldb Journal, 21(6), 753-777.

[2] Acosta-Mendoza, N., Gago-Alonso, A., & Medina-Pagola, J. E. (2012). Frequent approximate subgraphs as features for graph-based image classification. Knowledge-Based Systems, 27, 381-392.

[3] Gago-Alonso, A., Puentes-Luberta, A., Carrasco-Ochoa, J. A., Medina-Pagola, J. E., & Martinez-Trinidad, J. F. (2010). A new algorithm for mining frequent connected subgraphs based on adjacency matrices. Intelligent Data Analysis, 14(3), 385-403.

[4] Kimelfeld, B., & Kolaitis, P. G. (2014). The complexity of mining maximal frequent subgraphs. Acm Transactions on Database Systems, 39(4), 13-24.

[5] Kobayashi, Y., & Yin, X. (2012). An algorithm for finding a maximum [formula omitted]-matching excluding complete partite subgraphs. Discrete Optimization, 9(2), 98-108.

[6] Thomas, L. T., Valluri, S. R., & Karlapalem, K. (2010). Margin: maximal frequent subgraph mining. Acm Transactions on Knowledge Discovery from Data, 4(3), 1-42