一种基于分类的关联规则研究*)

王 勇 张 伟
(重庆教育学院计算机与现代教育技术系 重庆 400067)

摘 要 传统的 Apriori 关联法则算法必须经过大量反复的数据库扫描才能产生候选项集,效率较低。提出一个改进的 CBA(Classification Based Apriori)算法。此算法仅需扫描数据库一次,将数据库经过预处理后,再将事务数据库进行分类并保存分类结果,比较时可以不与所有事务记录进行比较,从而减少扫描数据库的次数与比较时间,且又能确保挖掘结果的完整性与正确性。

关键词 数据挖掘,关联法则,分类

Study of Association Rules Based on Classification Technology

WANG Yong ZHANG Wei
(Department of Computer and Modern Education Technology, Chongqing Education College, Chongqing 400067, China)

Abstract Typical association rules algorithm contais weaknesses such as often requiring a large number of repeated passes over the database to generate the candidate item sets. In this paper, we present an improved method named CBA (Classification Based Apriori). The CBA method scans the database once. After preprocessing the database, we imploy the classification technology to classify the database and save the results. Thus, the large itemsets are generated by contrasts with the partial classified transaction records. This not only prunes considerable amounts of data reducing the time needed to perform data scans and requiring less contrast, but also ensures the correctness of the mined results.

Keywords Data mining, Association rule, Classification

1 引言

经典的 Apriori 关联法则算法在大量数据的挖掘过程中,必须经过逐层的重复连接与运算步骤,才能找出所有的频繁项集。它在每一层中都会先产生大量的候选项集,而每一个候选项集又都必须与数据库中的每一笔事务记录做比较,不断地扫描数据库以找出所有符合最小支持度限制的频繁项集,直到找出所有频繁项集或无法再继续产生新的候选项集,而后再利用这些频繁项集探讨事务之间的关系,推导出所有的关联法则。因为反复与数据库中的事务记录比较,要耗费大量的时间与内存空间,所以 Apriori 效率较低。

因此,本研究提出一个改进的算法,称为 CBA(Classification Based Apriori)算法。本算法着重于效率的提高,可以在不扫描整个数据库的状态下就提早找出频繁项集,以节省大量的比较时间,且又能确保结果的完整性与正确性。实验证明,本算法在支持度变化时,执行效率仍可以维持在稳定的状态,而 Apriori 算法的执行效率会随着支持度变小而差距越来越大。

2 CBA 关联法则

CBA 关联法则算法执行步骤分为 5 步,其流程图如图 1。

(1)生成单一频繁项集

扫描数据库一次,利用使用者自订的最小支持度产生单一频繁项集;此时必须要注意最小支持度的选择,因为若 Level-1 的最小支持度定得太高,会删剪掉过多的资料;反之,若以太低的最小支持度,则可能会留下过多的数据而耗费大量的内存。

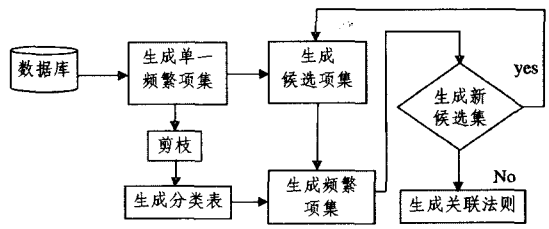

图 1 CBA 算法流程图

(2)剪枝

在事务库中,有许多的事务项目或整笔事务记录是没有用处的,因此在扫描过数据库后,同时利用步骤一所产生的单一频繁项集进行数据库的修剪,将事务记录中不属于单一频繁项集的事务项目或事务记录删除,只保留包含于单一频繁项集的事务记录。经过修剪后的数据库,可缩短事务记录的长度,使事务记录的分布更加的密集,大幅减少事务数据中所含的事务项目及事务记录,且能减少事务记录存入分类表时所需的内存。

(3)根据事务记录长度分类并存入分类表

为了能快速找出所有的频繁项集,进而推导出所有的关联法则,在采用单一频繁项集进行修剪数据库之后,即将修剪后的数据依事务记录长度分类并存入分类表中,其做法为将包含 k 个项目的事务记录归为第 k 群并存入第 k 个分类表 $table(k)$($1\leqslant k\leqslant M$,M 为数据库中最长的事务记录长度)中。

(4)生成候选项集

*)基金项目:中国博士后科学基金一等资助项目(No. 20060390175),重庆市教委资助项目(No. kj071502, No. kj051501, No. kj061501),重庆市科委自然科学基金资助项目(No. CSTC, 2005BB2286, 2006BB2254)。王 勇 讲师,硕士,主要研究方向为群体智能与数据挖掘;张 伟 教授,博士后,主要研究方向为信息安全、计算智能与数据挖掘。

执行程序必须由第一层的单一项集开始，逐层扩展，利用第 $k-1$ 层所得到的频繁项集进行连接，产生第 k 层的候选项集，每个候选项集再与分类表中的事务记录比较，若其支持度大于或等于使用者所订定的最小支持度，则成为第 k 层的频繁项集，再使用连接操作，将所有第 k 层的频繁项集连接成第 $k+1$ 层候选项集；其中，根据一个项集若是频繁项集，则其所有的子集合也都为频繁项集的特性，若结合产生的候选项集中有任一子集合不属于第 k 层频繁项集，此候选项集就必须被删除，如此重复地做连接与剪枝的动作，直到无法再产生候选项集为止。产生候选项集后，通过比较分类表中的事务记录来计算支持度以产生第 $k+1$ 层的频繁项集，若无法再产生新的频繁项集或无法产生新的频繁项集，表示所有的频繁项集都已找出。

```
算法: CBA (事务数据库D, 最小支持度min_sup)
    for (each transaction t∈D)  {
        if  (item∈t)  then  W_item ++ ;
        for  (item = 1; item≤.ItemNo; item ++) {
            if  (W_item≥.min_sup)  then  put  item  into L1;
        }
    }
    Classification_Table (length)= Classification_Table_gen (t)
    for  (k=2; L_{k-1} ≠ Ø ; k++) {
        Ck =Candidate_gen (L_{k-1});
        Lk =Large_itemset_gen (Ck);
    }
    Return L = ∪k Lk ;
Procedure Classification_Table_gen (t)
    set max_length=0;
    for (each transaction t∈D) {
        if  (item of t∉.L1)  then  delete the item from t;
        length = the item number of t ;
        if length > max_length then
            max_length = length ;
            put t into Classification_Table(length);
    }
    Return Classification_Table(length);
```

图 2　CBA 关联法则算法(1)

```
Procedure Candidate_gen(L_{k-1})
    For each itemset l1 ∈ L_{k-1}
        For each itemset l2 ∈ L_{k-1}
            if (l1[1]=l2[1]∧l1[2]=l2[2]∧...∧l1[k-2]=l2[k-2]∧
                l1[k-1]<l2[k-1] ) then {
                c= l1 link l2; //join step:generate candidates
            }
    For all itemsets c∈. Ck  {
        For all (k-1)-subsets s of c {
            if  (s∉. L_{k-1})  thendelete c from Ck ;
            else add c to Ck ;
        }
    }
    return Ck ;
Procedure Large_itemset_gen(Ck)
    for (level=k; level≤max_length; level++) {
        while (Ck ≠ Ø ) {
            pick c from Ck;
            temp=the count of c in the Cluster_Table(l);
            count(c)=count(c) + temp ;
            if count(c) ≥. min_sup then
                c≥Lk ;
                break ;
        }
    }
    return Lk;
```

图 3　CBA 关联法则算法(2)

(5)生成频繁项集

在计算长度为 k 的候选项集的支持度时，因其长度为 k，仅需由第 k 个分类表开始进行比较，在每扫描一个分类后，随即计算其支持度，而不一定要等到所有的数据扫描后才会进行计算，可立即判断其支持度是否符合所设定的最小支持度，若符合即成为频繁项集，则此候选项集就不需要再继续进行比较；反之继续到第 $k+1$ 个分类表进行比较，直到其支持度符合最小支持度，或是已比较至第 M 个(最后一个)分类表即结束比较。在搜寻的过程中，若所有的候选项集都已经判断完毕且无法再产生新的候选项集，即表示所有的频繁项集都已经找出，整个比较的流程就可以告一段落，并利用这些频繁项集进行关联法则的推导。

在给出 CBA 算法的伪代码之前，先介绍其中所使用的符号：D 是所有事务记录所形成的数据库，TID 是数据库中每笔事务记录的唯一代码，*length* 代表项集所包含的项目个数，亦即项集或事务记录的长度，max_*length* 是数据库中最长的事务记录长度，min_*sup* 是使用者设定的最小支持度，*item* 为项目编号，*k-itemset* 代表包含 k 个项目的项集，C_K 表示第 k 层的候选项集，它是由第 $k-1$ 层的频繁项集(L_{K-1})连接而成，L_K 表示第 k 层的频繁项集，亦即所有 C_K 中符合最小支持度的候选项集，cluster_Table(k)表示事务长度为 k 的事务记录所存于的第 k 个分类表，m_j 表示分类表 cluster_Table(j)中所包含的事务记录数目，W_{item} 代表 *item* 在数据库中出现的次数，*ItemNo* 表示所有的 *item* 总和，*level* 表示目前所处的层。

3　实验

实验用的测试数据为 Microsoft SQL Server 2000 中所附的 FoodMart 零售市场事务数据库，从中分别随机抽出 4000，8000，12000，16000，20000 与 24000 笔等六组事务记录表。

本实验的算法程序用 Vc＋＋编写，将分别仿真测试 CBA 算法与 Apriori 算法。

表 1　不同规模事务数据库的效率比较(min_*sup*=0.35)

	频繁项集的数量						时间(秒)	
事务数	L_1	L_2	L_3	L_4	L_5	L_6	CBA	Apriori
4000	348	776	114	18	0	0	654	4896
8000	299	1354	205	38	8	1	925	6065
12000	301	2356	285	40	4	1	1185	6686
16000	324	3607	408	61	9	0	1869	8653
20000	365	3897	453	66	11	1	2164	9212
24000	286	4163	561	69	13	0	2686	10857

表 2　不同规模事务数据库的效率比较(min_*sup*=0.39)

	频繁项集的数量						时间(秒)	
事务数	L_1	L_2	L_3	L_4	L_5	L_6	CBA	Apriori
4000	212	348	81	9	0	0	238	605
8000	150	323	69	13	2	0	267	737
12000	240	303	57	11	0	0	306	868
16000	298	388	63	15	0	0	326	927
20000	346	547	91	16	2	0	346	1008
24000	386	568	88	14	1	0	389	1281

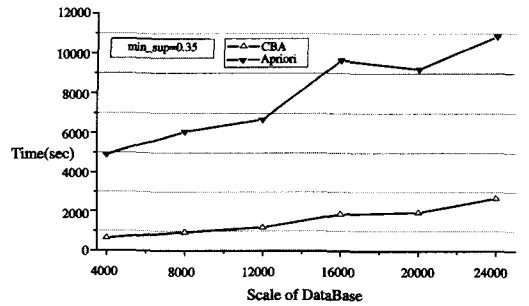

图 4　不同规模事务数据库的效率比较(min_*sup*=0.35)

结束语　由图 4，5 显示，用不同规模的事务数据库测试 CBA 算法与 Apriori 算法的效率，结果均显示 CBA 算法优于与

Apriori算法。因此，本文提出的CBA算法不仅有效减少数据库的重复扫描次数，还减少了事务的比较运算量，节省时间与存储空间，在挖掘大量的事务数据库时将发挥很大的优势。

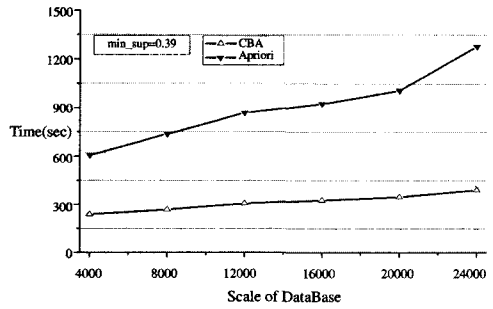

图5 不同规模事务数据库的效率比较(min_sup=0.39)

参 考 文 献

[1] Agrawal R, Imilienski T, Swami A. Mining Association Rules between Sets of Items in Large Databases// Proc. of the ACM SIGMOD Int'l Conf. on Management of Data. 1993:207-216

[2] Agrawal R, Srikant R. Fast Algorithm for Mining Association Rules in Large Databases// Proc. 1994 Int'l Conf. VLDB. Santiago, Chile, Sep. 1994:487-499

[3] Agrawal R, Srikant R. Mining Sequential Patterns// Proc. of the Int'l Conference on Data Engineering (ICDE). Taipei, Taiwan, March 1995. Expanded version available as IBM Research Report RJ9910, Oct. 1994

[4] Berzal F, Cubero J C, Marin N, et al. TBAR: An efficient method for associatin rule mining in relational databases. Elserier, Data & Knowledge, Engineering, 2001, 37:47-64

[5] Brin S, Motwani R, Ullman J D, et al. Dynamic Itemset Counting and Implication Rules for Market Basket Data// ACM SIGMOD Conference on Management of Data. 1997:255-264

[6] Brin S, Motwani R, Silverstein C. Beyond Market Baskets: Generalizing Association Rules to Correlations// 1997 ACM SIGMOD Conference on Management of Data. 1997:265-276

[7] Cabena P, Hadjinian P, Stadler R, et al. Discovering Data Mining From Concept to Implementation. Prentice-Hall Inc., 1997

[8] Carter C, Hamilton H, Cercone N. Share Based Measures for Itemsets. Principles of Data Mining and Knowledge Discovery// J. Komorowski and J. Zytkow, eds. 1997:14-24

[9] Chen M S, Han J, Yu P S. Data Mining: An Overview from a Database Perspective. IEEE Transactions on Knowledge and Data Engineering, 1996, 8(6)

[10] Chen M S, Park J S, Yu P S. Efficient Data Mining for Path Traversal Patterns. IEEE Transactions on Knowledge and Data Engineering, 1998, 10(2):209-221

(上接第136页)

以上公式中，$L(i)$表示实际负荷，$\hat{L}(i)$表示负荷预测值，M表示一天内负荷数据由M个数据点组成，这里为24点。

表1 不同方法的预测误差比较

时段	DMPSO-SVM				BPNN	RM	时段	DMPSO-SVM				BPNN	RM
	σ	ε	C	RPE_D	RPE_B	RPE_R		σ	ε	C	RPE_D	RPE_B	RPE_R
1	2.8025	0.0835	279.6	1.25	2.67	3.60	14	9.4189	0.0493	548.2	1.52	1.97	2.93
2	9.9981	0.0711	961.1	1.53	2.70	3.67	15	4.2546	0.0586	248.5	1.25	1.31	1.91
3	7.1816	0.0529	639.8	0.74	1.83	4.08	16	9.9997	0.0299	1.2	0.72	1.77	2.95
4	9.0026	0.0619	495.8	0.67	3.70	3.35	17	10.000	0.0214	18.5	1.01	1.86	3.66
5	4.1808	0.0617	405.1	1.52	0.59	3.87	18	7.5609	0.0703	537.2	0.77	1.11	3.76
6	3.8819	0.0101	19.2	1.77	2.75	3.63	19	9.9086	0.0392	84.6	1.60	2.65	4.85
7	6.7168	0.0545	333.6	1.20	2.71	5.06	20	5.2874	0.1044	503.8	1.12	3.86	3.92
8	3.5551	0.0511	1000	0.90	1.74	4.22	21	3.3533	0.0972	679.8	0.70	3.59	1.33
9	5.1204	0.0281	799.1	1.46	4.25	3.82	22	1.2635	0.0618	237.7	1.66	2.66	3.78
10	4.4169	0.0272	547.3	1.21	4.49	2.98	23	4.4897	0.0531	949.3	2.33	3.83	2.84
11	5.7473	0.0505	344.2	1.60	2.48	3.66	24	1.2368	0.0647	951.8	1.42	3.69	3.98
12	10.000	0.0361	594.2	2.61	1.59	3.70	MAPE				1.37	2.61	3.57
13	4.7601	0.0686	556.9	2.26	3.00	4.31	RMSE				1.46	2.81	3.66

由表1可知，DMPSO-SVM方法的最大相对误差为2.61%，此结果明显好于BPNN方法和RM方法的4.49%和5.06%。DMPSO-SVM方法的平均相对误差和均方根相对误差也明显优于另外两种方法，这说明DMPSO-SVM方法在预测短期电力负荷时，预测的精度和预测的稳定性均强于另外两种方法。可见，将本文提出的方法应用于电力系统短期负荷预测中是有效可行的。

结束语 在对支持向量机的参数性能进行分析的基础上，本文提出了一种基于动态多种群粒子群支持向量机的电力系统短期负荷预测模型，其中利用动态多种群粒子群优化算法来优化支持向量机的参数，克服了SVM参数选择的盲目性；利用快速傅立叶变换得到隐含于负荷数据中的周期性特性，从而基于频谱特性确定预测模型的输入变量。电力系统短期负荷预测的实际算例表明，与人工神经网络及回归分析模型相比，本文所提出的方法具有更高的预测精度和鲁棒性。

参 考 文 献

[1] Hsu C C, Chen C Y. Regional load forecasting in Taiwan: applications of artificial neural networks[J]. Energy Convers. Manage., 2003, 44(12):1941-1949

[2] 唐亮贵，程代杰. 基于小波的支持向量机预测模型及应用[J]. 计算机科学，2006，33(3)：202-204

[3] 李元诚，方廷健，于尔铿. 短期负荷预测的支持向量机方法研究[J]. 中国电机工程学报，2003，23(6)：55-59

[4] Feng Pai Ping, Chang Hong Wei. Support vector machines with simulated annealing algorithms in electricity load forecasting[J]. Energy Conversion and Management, 2005, 46(3):2669-2688

[5] Kennedy J, Eberhart R. Particle swarm optimization[C]// Proceedings of IEEE International Conference on Neural Networks. Perth: IEEE Press, 1995:1942-1948

[6] Eberhart R C, Shi Y. Comparing inertia weights and constriction factors in particle swarm optimization[C]// Proceedings of the Congress on Evolutionary Computing. IEEE Service Center, California USA, 2000:84-88

[7] Ursem R K. Multinational evolutionary algorithms[C]// Proceedings of Congress of Evolutionary Computation. Washington, DC, USA, 1999(3):1633-1640

[8] 张浩然，韩正之，李昌刚. 支持向量机[J]. 计算机科学，2002，29(12)：135-137

[9] 李华，张简政. 基于模糊支持向量机的网络入侵检测研究[J]. 计算机科学，2005，32(11)：77-80

[10] Yang Jingfei, Stenzel J. Historical load curve correction for short-term load forecasting[C]// Proceedings of The 7th International Pwer Engineering Conference. Singapore, IEEE Press, 2005:1-6

[11] 邰能灵. 小波分析在电力系统中的应用及相关问题研究[D]. 上海：上海交通大学，2002