ISSN:0975-9646

N-list based Friend Recommendation System Using Pre-rule Checking

Anil K.R
Research Scholar School of Computer Science
Mahatma Gandhi University
Kottayam, Kerala, India

Gladston Raj S
Head, Department of Computer Science
Government College, Nedumangad
Thiruvananthapurm, Kerala, India

I. ABSTRACT.

The web friend recommendation system is the emerging trend in the social network and e-commerce. It plays an important role in the web based applications. The frequent pattern mining is one of the techniques used in mining applications and N-list is the efficient data structure representation used with this. The data pruning strategy is always been a challenge on improving the efficiency of N-list based system. In this paper we present a pre-rule based N-list which explore the power of association rule mining with pre-rule checked N-list. This technique enhances the speed and efficiency of the friend recommendation system.

Keywords: N-lists,Pre Post Code Tree

2. INTRODUCTION

Data Analyticalsystems are playing pivotal roles in fine-tuning goals such as improving customer retention, product recommendations and community specific profle recommendations. In most cases, these insights are driven by analyses of historic data. The recomendations are commonly conducted based on topic modelling, which strictly relies with semantic meaning and statistical frequencies. The concept of recommendations has improved much with the introduction of frequent pattern mining for large databases of social circle data. The frequent pattern based operataions works on transaction records, is of great interest in data mining and knowledge discovery since its inception in 1993, by Agrawal et al. The present paper focus on a novel algorithm, NList based Prerule Checking, which is an extension of frequent pattern tree algorithm. The algorithm is experimentally proved with the implementation of a Web based friend recommendation system.

3. N-LIST

N-list is data structure proposed to represent frequent patterns. The ability to mine patterns along the N-list nodes are made possible by maintaining a subset concept. N-list have a property called single path property which helps to mine data in a faster way.

Recently proposed Pre-Post Code Tree algorithm for mining FP based N-list structure and children parent equivalence pruning[1]is another high-performance technique for mining frequentitemsets. It employs N-list to represent itemsets and directly discovers frequent itemsets using a set enumeration search tree.

4. DRAWBACKS OF N-LIST

Implementation of N-list may be in- appropriate in large data sets.

The efficiency of Nlist based algorithms are challenged when the consumption of memory is high and the processing speed is low. Also low support features are always been generated in mined results, which caused unfavourable results.

5. FREQUENT CLOSED PATTERNS[1]

Table 1: Shows the transaction set containing different number of items.

Transaction	Items
1	A,C,T,W
2	C,D,W
3	A,C,T,W
4	A,C,D,W
5	A,C,D,T,W,S
6	C,D,T,E

In the above transaction set each transaction contains a number of items. The support of the pattern X, denoted by σ(X),where x € I and I is the set of all items in DB, is the number of transactions containing all the items in X. A pattern with K items is called a k-pattern an $_{Ii}$ is the set of I-patterns sorted in order of descending frequency.

Consider the minSup (minimum support threshold) be a given threshold. A pattern X is called a frequent pattern if σ(X) ≥ minSup x n.A frequent pattern is called an FCP if none of its subset has the same support.

6.PPC-TREE

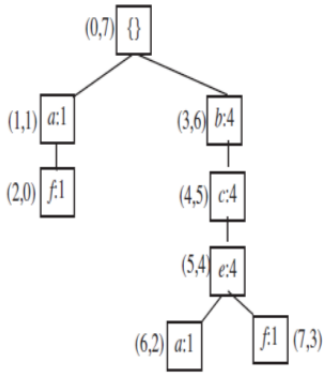
The Pre – Post Code Tree is a tree structure like FP tree which is used to represents the frequent pattern[1]. The Node of the PPC hold five values they are $n(N_i)$,$f(N_i)$,$childs(N_i)$,$pre(N_i)$,$post(N_i)$ which are the frequent I-patterns in I_i the frequency of this node,the set of child nodes associated with this node, the order number of nodes when traversing the tree.

In this section, we discussed the details of N-list pre post[3]. In a given transaction database .DB and threshold. The succinct version of DB is a database generated by deleting infrequent item sets. Therefore we can directly mine frequent items sets from the succinct version of DB instead of the DB.

Let us consider an example

Table 2 Illustrates the order of items from item collection with in a transcationdatabase

ID	ITEM	Order of Items
1	a,f,g	a,f
2	a,b,c,e	b,c,e,n
3	b,c,e,i	b,c,e
4	b,c,e,h	b,c,e
5	b,c,d,e,f	b,c,e,f



The frequent I-item sets set F1={a,b,c,e,f}. Note that the last column is the succinct version. The node with (3,6) means that its pre-order is and post-order is 6, and the item name b, and count 4.

7.WEB BASED FRIEND RECOMMENDATION SYSTEM

The recommendation systems are gaining wide spread acceptance in social networking and e-commerce applications. User habits,search patterns, subscription information etc acts as an input of recommendation systems. There needs many approaches to tackling the information overload on web portals, when building friend recommendation models. There are stringent relations within the transaction and between transactions used with frequent data. Basic stages in making a web based friend recommendation system are stages (i) Formation of user or item neighbourhood stage (ii) Generation of top N-list with algorithms that construct a list.

The Degree of sparsity however depends on the application type[2].

Stage (iii) Application of quality assessment using pre post method of the top N-list.

Factors affecting the Web Based Friend Recommendation Model[2]

8.SPARSITY

The Sparsity is the most[2] important factor indata mining applications, which is caused by the missing data in datasets. In some applications, users rate sparcity as a negligible factor.But on large applications in Social media context, the data set has to become free from sparsity. The problem of sparsity is a complicated issue. Major studies are conducted in this issue for handling the missing values, to get dataset free from sparsity.

9.SCALABILITY

Scalability is also an important issue in the case of web based friend recommendation system because the real world application like Social Media needs correct and complete information to generate better results. Scalability allows the researcher to work with large item datasets quickly and efficiently.

10.TRAIN DATA SET

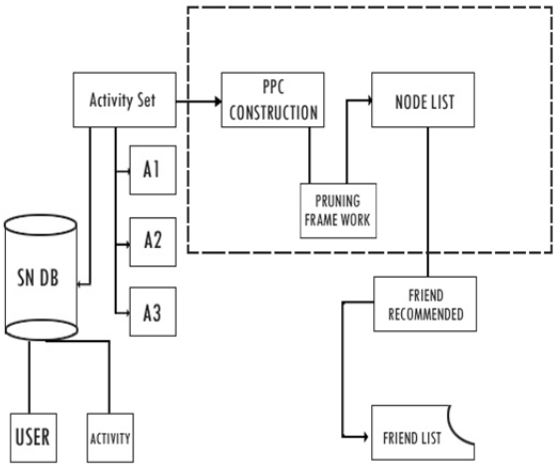
There is a clear dependence between training set size and accuracy of the algorithms [2]. So the size of the training is an important factor that determining the accuracy and efficiency of the algorithm. However the effect of over fitting is less significant compared to general classification problem [2]. On the basis of the experiments conducted, 75% of the training data is selected as the adequate choice of input dataset.With the selected training set,we could establish the concepts easily and clearly.

11.NEIGHBOURHOOD SIZE

The formation of neighbourhood is another important measure to determine the accuracy of the algorithm. If the number of nearest neighbours is very small the accuracy is low [2]. If enough neighbours are present the prediction is correct.

12.PROPOSED MODEL

The proposed model consist of a frame work that may identify all frequent item sets from a ppc tree, then construct an N-list and apply a pruning strategy on data related to user activities. The user activities are mined to find frequent patterns and it generates friend recommendations.The proposedPrerule based N-listworks on a rule confidence and fitness function that decides the quality of recommendation.



***Algorithm 1*: Pre rule checking based N-list** Generation[1]

Input: Transaction database DB containing socialnetwork activity,minimum support S, minimum confidence C, fitness threshold.

Step 1: Scan DB to obtain frequent one item set and build PPC Tree
Step 2: Scan PPC Tree to generate N-list of one item set (activity)
Step 3: Continue step 1 until item set i=1 to n, n is the last activity
Step 4: For each frequent item sets join the N-list using N-intersection (N-list, i=0....N)
Step5: Build Pattern tree (node lost),candidate item, activity confidence)
Step 6: Return frequent item set containing SN activity.

***Algorithm 2* (building pattern tree)**

Step 1: *Inilizes node equivalent*
Item =0
Node Child node =0
Net Candidate Item =0
Step 2: For Each item in candidate set Do
Step 3: P1 = N0 Item set
P2 = i U (p1 – p1(1)
P= i U (Pi)
Step 4: P.N list =NL-Intersection of (p1,P2)
Step 5: if P.support> S
Step 6: Cj be the confidence of ith activity
Step 7: if Ci>c create a node Ndi
Repeat step 6 for (i = to n)
Step8: Calculate fitness function Φ = C(i) / support (i)
Step 9: Build a pattern tree (Ndi,Next Candidate item, S,C) if d>fitness threshold
Step10: return FP tree

Algorithm 3: NL-Intersection (NL1,NL2)

Input:NL1= ((x11,y11):Z11),((x12,y12):z12),.....((x1m, Y1m) : z1m) and NL2 = ((x21, y21) : z21), (x22 ,y22) : z22) , ... (x2n , y2n) : z2n) , which are the N-lists of P1= i U i1 i2 ..... i (k-2) and p2 = i V i1 i2.... i (k-2) i U > iv) respectively.

Step 1: i=1;
Step2: j=1
Step 3: while (i<=m && j<=n)
Step 4: if (1i<x2j) then
Step 5: if (y1i>y2j) then
Step 6: insert((x1i,y1i):z2j) into NL3:
Step 7: j++
Step 8 else i++
Step 9: else j++
Step10: ptr1= NL3First_element
Step11: ptr2=ptr1.next_element
Step 12: while ptr1 is not the last element of NL3 do
Step 13: if ptr1.pre-code = ptr2.pre-code and ptr1.posr and ptr1.post-code = ptr2.post-code
Step 14 ptr1.count=ptr1.count + ptr2.count
Step 15: delete ptr2 from NL3
Step 16: ptr2= ptr1.next_element
Step 17 :else
Step 18: ptr1=ptr2
Step 19: ptr2= ptr1.next_ekement
Step 20 return NL3

13.EVALUATION

We have conducted experiments to evaluate the performance of the proposed algorithm and also analysis time complexity and computation complexity of the new system.Present algorithm is evaluated against an available FPGrowthalgorithm and found to be correct and thecompletenew system was applied on a friendrecommendation scenario using a social network data set.

14.EXPERIMENTAL SETUP

In this section, we evaluate the performance of theproposed algorithm. Summarizing the studies,we can conclude that the algorithm Pre rule checking based N-list generationis the state-of-the-art algorithm employing the level-wise approach, FP Tree.

We compare it with the FP Tree pattern-growth approach and the data is implemented on social data set. All algorithms were written inC#.NETprogramming language. The configuration of the testing platform is as follows: Windows Operatingsystem, 2GB Memory, Intel(R) Core(TM) i3-2310 CPU @2.10 GHz;

There are four parameters used to generate thefriend recommendation results, including the confidence threshold, fitness threshold, time for processing and the number of outputs recommended.

We have used the following values as default throughempirical studies, i.e., the confidence threshold is set 20 to 60%, the fitness factor is set to 0.8, and the minimum number of attribute is to be recovered for recommendation is set to 5.

Experimental Inference

Table 3:Illustrates the Time inference for different algorithms -FP Growth and Prerule Based Algorithms under different confidence values

Confidence	FP Growth	Pre rule Based	Time Inference (Milli Second FP time)
20	42	57	210
30	39	51	232
40	33	39	235
50	27	37	239
60	21	29	267
90	12	15	311

A.Recommendation Precision Rp:
Precision is referred as the ratio of the number of inferred friends to the number of actual friends, when a target query is processed over a dataset of user activities..

$$R_p = \frac{\sum_i |F_i \cap G_i|/|F_i|}{1000}$$

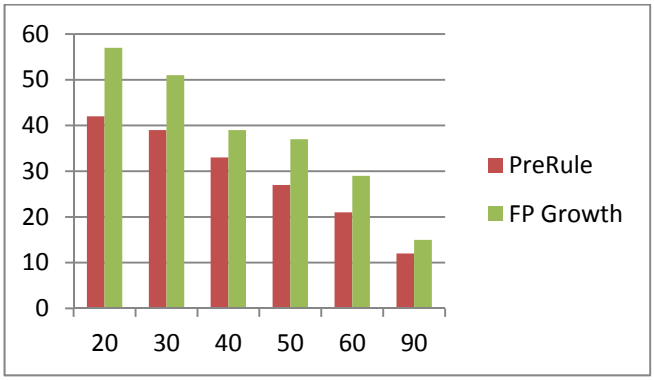
where j _ j denotes the number of elements in a set.The dominator is 1000 because Rpis the average of 1000 users in one experiment. The Precision factor is neared to 1, which is fine for the experiment.

B.Recommendation Recall Rr:
The average of the ratio of the number of recommended friends in the set of true friends of the query user over the number ofthe set of true friends of the query user, which is 1000 in our experiments.

$$R_r = \frac{\sum_i |F_i \cap G_i|/|G_i|}{1000} = \frac{\sum_i |F_i \cap G_i|/100}{1000}$$

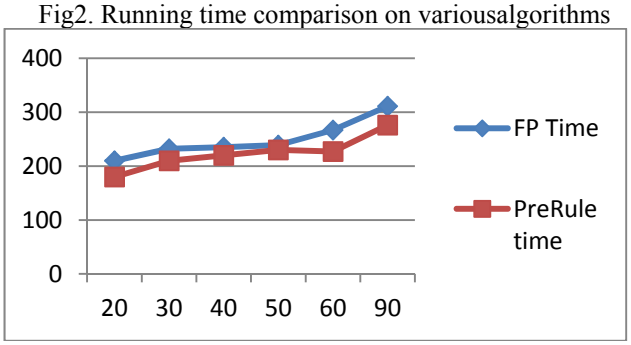
The Precision factor is neared to 1, which is fine for the experiment.

Fig.1 . No. of Users Generated based on various Conf.Level



C.Running Time Comparison
In this subsection, we compare new algorithm and FP-growth in terms of runtime. We have conducted substantial experiments spanning the datasets for various values of minimum confidence. Note that runtime here means the total execution time, which is the period between input and output. Figure 4 shows comparison of the running time the new algorithm based Friend recommendation system with FP growth based systems.

Fig2. Running time comparison on variousalgorithms



CONCLUSION

The analysis of the newly developed algorithm has proven to be fine with an existing algorithm. The area of recommendation systems for ecommerce applications like product recommendation, friend recommendation and service recommendations are still facing problems with accuracy in results and performance time. These issues may be resolved up to an extend by using the Prerule checked N-list based systems. The evaluation results were proved quantitately, with values arrived as the precision and accuracy of the recommendation systems. The values are generated in a range of 0.8 to 1. The efficiency of the new algorithmis also trailed under different values of fitness threshold. Thus the Pre-rule checking based algorithm has succeeded in establishing an improvement in the precision of recommendation systems.

The algorithm can be further enhanced to apply many features like Semantic analysis of the activity using any NLP Model, also the pattern equivalence features will help to improve the results.

REFERENCES

[1] Bay Vo, Tuong Le, FransCoenen&Tzung- PeiHong Mining Frequent ItemSets using N-list and subsume concept.
[2] Tuong Le, Bay Vo (2015) An N-list basedalgorithm for mining frequent closed patterns
[3] Zhi-Hong Deng, Sheng-Long Lv (2015) An efficient N-list based algorithm for miningfrequent item sets via children –parent Equivalence purning
[4] Aggarwal C.C,Li Y &wang (2009) Frequent Pattern Mining with uncertain data In: SIGKDD,pp.29-38
[5] Agarwal R, &Srikant R (1994) Fast algorithm for mining association rule In:VLDB,pp 487-499
[6] Agarwal R,Lmielinski T Swami AN (1993)Mining Association rule between sets of items inlarge databases In: Proceedings of the SIGMOD'93, pp 207-216
[7] Ayres J Gehrke JE Yui T Flannick j (2002)Sequential pattern mining using a bitmaprepresentation. In Proceedings of the SIGKDD'02. pp 429-435
[8] Bernecker. T Kriegal, H Renz, M. Verhein, F, &Zuefle, A (2009) Probabilistic frequent item setmining in uncertain database
[9] Baralis E, Cerquitelli T, Chiusano S (2010) Constrained Item set mining on a sequence of incoming data blocks In:Intell syst25(5):389-410
[10] Deng Z,Fang. G, Wang Z Xu X(2009) Mining Erasable Item sets In Proceedings of the ICMLC'09. Pp67-73
[11] Dong J.Han M (2007)An efficient mining Frequent item set algorithm Knowl BasedSyst20:329-335
[12] Han J, Pei J. Yin Y (2000) Mining frequent Patterns without candidate generation In:Proceedings of the SIGMODKDD'00.PP1-12
[13] Lucchese B, Orlando S, Perego R (2006) Fast and memory efficient mining of frequent item sets. IEEE Trans Knowl Data Eng.18(1):21-34
[14] GrahneC.&Zhu J (2005)Fast Algorithm frequent item set using NC-sets Expert systems with applications 39(4),4453-4463