International Journal of Soft Computing 13 (2): 61-68, 2018
ISSN: 1816-9503

Pattern Deploying Methods for Text Mining

Ravindra Changala and D. Rajeswara Rao
Department of CSE, KL University, Andhra Pradesh, India

Abstract: Effective use of discovered patterns in text documents is an open issue. Many data mining techniques have been proposed and gave solutions with respective difficulties. Still, improving the effectiveness of patterns use became a research issue. This study presents pattern deploying for text mining patterns by considering pattern as atom. The significance of pattern can be measured with weighing functions; it is time consuming in the computational aspect. Hence, we need an efficient and effective pattern evolution technique for effective use of discovered patterns. The objective of this study is to explore pattern deploying methods after pattern is discovered in text documents based on supports. Finally, explains interference and significance of a specific pattern.

Key words: Pruning, closed sequential pattern, PDS, PDM, SCPM, consuming

INTRODUCTION

Text mining is one of the methods to get required information from volumes of databases. Many approaches have been proposed to achieve this goal (e.g., key-word based approach) and other pattern based techniques also used. Even Information Retrieval (IR) does not satisfy the user's need where irrelevant data is coming out. The Pattern Taxonomy Model (PTM) worked well but failed due to its low capability of handling patterns. Hence, we felt that the need of pattern deploying approaches for text mining to increase the performance and effective usage of discovered patterns of German credit data set (Zhong *et al*., 2012; Pipanmaekaporn and Li, 2012; Wu *et al*., 2006).

Text mining: "The nontrivial extraction of implicit, previously unknown and potentially useful information from (textual) data" (strict) or "the science of extracting useful information from large (textual) data sets" (loose) (Ref: Introduction to Text Mining, MandarMitra CVPR Unit, Indian Statistical Institute, Kolkata). Text mining is about looking for patterns in text. Nowadays data gathering from various sources have been increasing rapidly, hence, getting useful information is an issue. Traditional information retrieval techniques become inadequate for the increasingly vast amounts of text data. Without knowing what could be in thedocuments, it is difficult to formulate effective queries for analyzing and extracting useful information from the data. Users need tools to compare different documents, rank the importance and relevance of the documents or find patterns and trends across multiple documents. Thus, text mining has become an increasingly popular and essential the mein data mining. It is known that precision and recall are the basic measure for the text mining and document selection and document ranking are the methods of text mining (Wu *et al*., 2006; Pipanmaekaporn and Li, 2012).

Literature review: According to Lewis (1992), Huang and Lin (2003) used phrase based techniques for text categorization got poor results than word based as well as Scott and Matwin (Li and Liu, 2003). Many Information Retrieval (IR) techniques (term frequency (tf)), inverse document frequency (idf), weighing schemes, dimensionality reduction, feature selection techniques the details of these selection functions are stated by Li and Zhong (2006). Improved nominal performance of effective usage of patterns.

MATERIALS AND METHODS

Deploying method: Support and confidence are the basic properties of a pattern which are not suitable while discovering useful pattern in text mining. Hence, the properties of patterns by deploying their correlations to pattern taxonomies can be done by using Pattern Deploying Method (PDM). Firstly, it performs pattern deploying then pattern deploying based on supports (PDS) (Zhong *et al*., 2012; Pipanmaekaporn and Li, 2012):

$$W(p) = |\{da|da \in D+, p \text{ in } da\}| / |\{db|db \in D, p \text{ in } db\}|$$

Where:
da and db = Documents
D+ = Positive documents in D such that D, D+⊆D

Corresponding Author: Ravindra Changala, Department of CSE, KL University, Andhra Pradesh, India

However, the problem of this method was the low pattern frequency due to the fact that it is difficult to match patterns in documents, especially, when the length of the pattern is long.

Pattern deploying method: By using this method, we can differentiate low and high frequencies of patterns in the documents (e.g., compare sequential pattern mining with pattern mining, first one giving specific meaning). Hence, we emphasize on significance and interference of a patterns. The basic nature of data mining techniques is to generate large amount of small patterns leads to low performance which can be resolved by using pruning algorithms (Zhong *et al*., 2012; Wu *et al*., 2006).

To represent the overlaps among patterns, we deploy the set of patterns for the document dk on T, the set of terms, to obtain the following vector:

$$dk = <(tki, nki), (tk2, nk2), ..., (tkm, nkm)> \quad (1)$$

where, ti in pair (ti, ni) denotes a single term and ni is its support in dk which is the number of patterns that contain ti.

The detailed deploying process is presented in algorithm. Note that the SPMining algorithm is used in line 3 for generating frequent sequential patterns. The main process of pattern deploying is between line 5 and line 7 inclusively. The normalization in line 8 is normally used to assume the contribution of each document is equal. The result of this algorithm is a set of documents, which can be expressed as follows:

$$\eta = \{d1, d2, ..., dn\} \quad (2)$$

We also need to determine the weight for each term in T when we use the discovered knowledge in η. The weighting scheme for a given term t is denoted as the following function:

$$Weight(ti) = \sum = (nki/\sum_{(t,w)\in dk} w)\ d_k \in n \quad (3)$$

where, 'w' is the support of term t indicating the number of patterns that contain t in a document d. In order to deploy discovered patterns and acquire deployed support of each term in these patterns, we use a pattern composition operation to join two patterns.

Let termset (P) = {t|(t, f)∈P} be the termset of pattern P where f denotes term frequency in pattern P. The composition of two patterns P1 and P2 can be processed by using the following expression:

$$P1 \oplus P2 = \{(t, f1+f2) | (t, f1) \in P1 (t, f2) \in P2\} \cup \{(t, f) | t \in (termset(P1) \cup termset(P2)) - (termset(P1) \cap termset(P2)), (t, f) \in P1 \cup P2$$

For example, Pa = ht1, t2i and Pb = ht2, t3i are two frequent patterns. The termsets of Pa and Pb are {(t1, 1) (t2, 1)} and {(t2, 1)(t3, 1)}, respectively. The composition of these two patterns Pa⊕Pb can be denoted as P0 where termset (P0) = {(t1, 1)(t2, 2)(t3, 1)}.

Pattern deploying algorithm: To have effective use of discovered patterns by summarizing them as d-patterns to evaluate accuracy In order to use the semantic information in the pattern taxonomy to improve the performance of closed patterns in text mining, we need to interpret discovered patterns by summarizing them as d-patterns in order to accurately evaluate term weights (supports). The rational behind this motivation is that d-patterns include more semantic meaning than terms that are selected based on a term-based technique (e.g., tf*idf). As a result, a term with a higher tf*idf value could be meaningless if it has not cited by some d-patterns (some important parts in documents). The evaluation of term weights (supports) is different to the normal term-based approaches. In the term-based approaches, the evaluation of term weights are based on the distribution of terms in documents. In this research, terms are weighted according to their appearances in discovered closed patterns.

Pattern deploying algorithm 1; (Zhong *et al*., 2012; Pipanmaekaporn and Li, 2012; Joachims, 1996):

```
PDM (D+, min_sup)
    Input: Documents, Minimum Support
    Ouput: d-patterns in D+ and supports of terms
1.  For each document d of D+ then start
2.  Let PS(d) be the paragraphs set in d,
3.  SP = SPmining (PS(d), min_sup)
4.  d = ø
5.  for each pattern p, ∈ SP start
6.  p = {t,1|t∈p}
7.  d = d⊕p                                  end
8.  DP = DP U {d}                            end
9.  T = {t|(t,f) ∈p, p ∈ DP}
10. For each term t ∈ T do
11. Support (t) = 0                          end
12. For each d-pattern p ∈ DP do
13. For each (t,w) ∈ β(p) do
14.  Support(t) = support(t) + w;            end
15. end
```

This algorithm 1 gives the deploying of patterns with minimum support and confidence to get discovered patterns. The sequential pattern mining gives complete significance of the pattern through d-pattern algorithm.

Closed patterns: It is complicated to derive a method to apply discovered patterns in text documents for

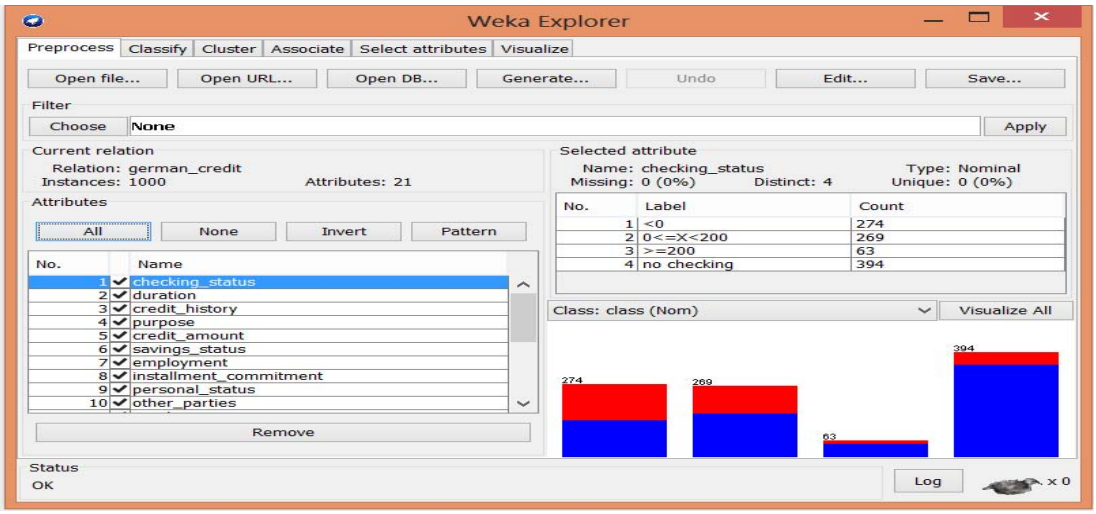


Fig. 1: Lodaind data set

Table 1: Cross validation

	Clusters										
Attributes	0 (0.2)	1 (0.04)	2 (0.04)	3 (0.02)	4 (0.64)	5 (0.07)	6 (0.07)	7 (0.01)	8 (0)	9 (0.04)	10 (0.04)
Checking_status											
<0	7.1983	15.0481	26.9500	5.6595	176.1100	25.3276	3.6958	1.7020	3.8300	14.7918	4.6871
0≤X<200	8.4032	6.4855	8.5102	2.0925	206.8923	23.8309	5.8328	4.8644	1.6091	5.8389	5.6402
≥200	1.0089	4.1493	5.0408	2.0590	38.4406	6.8688	6.4817	1.1328	1.0591	5.3936	2.3655
No checking	6.4788	18.0595	7.6304	9.8150	224.9076	17.2324	62.7430	7.3801	1.8652	15.1104	33.7775
Total	23.0893	43.7423	48.1314	19.6261	646.3503	73.2596	78.7533	15.0793	8.3633	41.1347	46.4703
Duration											
Mean	33.4917	12.8535	26.7523	14.1666	22.0829	13.5510	20.7793	32.6929	16.8475	10.3298	20.1251
SD	11.1499	6.6890	13.9575	6.0028	12.3844	5.7051	7.3448	17.4800	8.7136	4.8228	10.0900
Credit_history											
No credits/All paid	3.0382	1.1324	5.9137	1.1799	32.2701	1.8009	1.4400	1.0184	1.0128	1.1782	1.0154
All paid	2.9651	6.1092	6.4204	1.0986	32.1085	4.4068	1.1619	1.5039	1.0128	1.1272	2.0856
Existing paid	9.7447	31.9435	17.2597	10.4699	366.1930	58.1821	26.1441	9.4818	1.0230	1.4954	9.0628
Delayed priviously	5.9131	2.4033	6.6328	1.1722	60.2539	2.6016	7.1672	1.7766	1.1180	4.2224	5.7389
Critical/other existing credit	2.4282	3.1539	12.9047	6.7055	156.5250	7.2683	43.8401	2.2906	5.1968	34.1115	29.5676
Total	24.0893	44.7423	49.1314	20.6261	647.3503	74.2596	79.7533	16.0793	9.3633	42.1347	47.4703

information filtering systems. To simplify this process, we first review the composition operation ⊕ defined by Lang (1995). Let p1 and p2 be sets of term-number pairs. p1⊕p2 is called the composition of p1 and p2 which satisfies:

$$p1 \oplus p2 = \{(t, x1+x2) | (t, x1) \in p1 (t, x2) \in p2\}$$
$$\cup \{(t, x) | (t, x) \in p1 \cup p2 \text{ not } (t,\_) \in p1 \cap p2)\}$$

where - is the wild card that matches any number. For the special case we have p⊕0 = p and the operands $d_i^{\wedge}$ of the composition operation are interchangeable. The result of the composition is still a set of term-number pairs. For example:

$$\{(t1, 1), (t2, 2)(t3, 3)\} \oplus \{(t2, 4)\} = \{(t1, 1)(t2, 6)$$
$$(t3, 3)\} \{(t1, 2\%), (t2, 5\%), (t3, 9\%)\} \oplus \{ (t1, 1\%)$$
$$(t2, 3\%)\} = \{(t1, 3\%)(t2, 8\%)(t3, 9\%)\}$$

Formally, for all positive documents $d_i \in D^+$, we first deploy its closed patterns on a common set of terms T inorder to obtain the following d-patterns (deployed patterns, non sequential weighted patterns) (Fig. 1):

$$\hat{d}_i = (til, ni1)(ti2, ni2), ..., (t_{im}, n_{im})$$

where t_{ij} in pair (t_{ij}, n_{ij}) denotes a single term and n_{ij} is its support in d_i which is the total absolute supports given by closed patterns that contain t_{ij} or n_{ij} (simply in this study) is the total number of closed patterns that contain t_{ij}. For example using Table 1 and 2, we have:

$$sup_a = (< t_3, t_4, t_6 >) = 3$$
$$sup_a = (< t_1, t_2 >) = 3$$
$$sup_a = (< t_6 >) = 2$$
$$\hat{d} = (t_1,3), (t_2,3), (t_3,3), (t_4,3), (t_6,8)$$

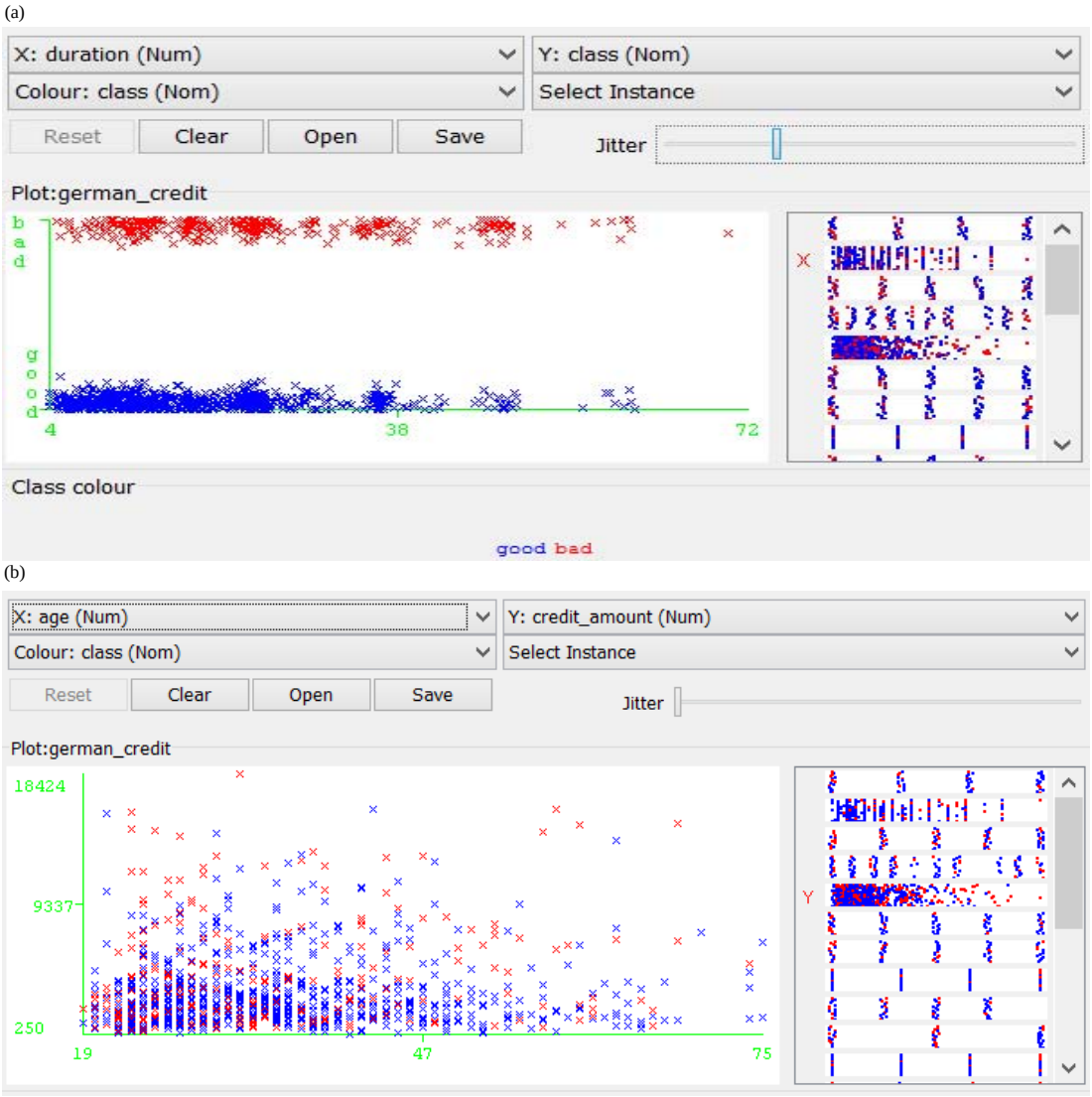


Fig. 2: a, b) Plot matrix of age and credit amount in visualization

Table 2: Results for k = 1000-3000

Dataset/	Execution time (sec)			Maximum memory usage (MB)		
Algorithm	k = 1000	k = 2000	k = 3000	k = 1000	k = 2000	k = 3000
Leviathan						
TKS	10	23	38	302	424	569
TSP	103	191	569	663	856	974
Bible						
TKS	16	43	65	321	531	658
TSP	88	580	227	601	792	957
Sign						
TKS	0.5	0.8	1.2	46	92	134
TSP	4.8	736	9.1	353	368	383
Snake						
TKS	1.1	1.63	1.8	19	38	44
TSP	19	33	55	446	595	747
FIFA						
TKS	15	34	95	436	663	796
TSP	182	O.O.M	O.O.M	979	O.O.M	O.O.M

The process of calculating d-patterns can be easily described by using the operation in algorithm 1 (PTM) shown in Fig. 2, that will be described in the next section where a term's support is the total number of closed patterns that contain the term. Table 3 illustrates a real example of pattern taxonomy fora set of positive documents. We also can obtain the d-patterns of the five sample documents in Table 3 which are expressed as follows.

D-pattern mining algorithm: To improve the efficiency of the pattern taxonomy mining, an algorithm, SPMining was proposed by Agrawal and Srikant (1994) to find all closed sequential patterns which used the well-known Apriori property in order to reduce the searching space.

Algorithm 1 (PTM) shown in Fig. 2 describes the training process of finding the set of d-patterns. For every positive document, the SPMining algorithm is first called in step 4 giving rise to a set of closed sequential patterns SP. The main focus of this study is the deploying process which consists of the d-pattern discovery and term support evaluation. In algorithm 1 (Fig. 2), all discovered patterns in a positive document are composed into a d-pattern giving rise to a set of d-patterns DP in steps 6-9.

Thereafter, from steps 12-19, term supports are calculated based on the normal forms for all terms in d-patterns.

Algorithm 2:

Input: Positive documents D^+, minimum support min_sup
Output: d-pattern DP, support terms
1. DP = ϕ
2. For each document dϵSP do
3. P = {(t,1)|tϵp_i}
4. $\hat{d}$ = {$\hat{d}$}ϕp
end
5. DP = DP U {$\hat{d}$}
end
6. T = {t|t, fϵp, pϵDP}
7. Foreach term tϵT do
8. Support (t) = 0
end
9. For each d-pattern dϵDP do
10. Foreach (t, w)$\epsilon\beta$(p) do
11. Support t = support (t)+w
end
end

Let m = |T| be the number of terms in T, n = $|D^+|$ be the number of positive documents in a training set, K be the average number of discovered patterns in a positive document and k be the average number of terms in a discovered pattern. We also assume that the basic operation is a comparison between two terms.

The time complexity of the d-pattern discovery (from steps 6-9) is $O(Kk^2n)$. Step 10 takes O(mn). Step 12 also gets all terms from d-patterns and takes $O(m^2n^2)$. Steps 13-15 initialize support function and take O(m) and the steps 16-20 take O(mn). Therefore, the time complexity of pattern deploying is:

$$O(Kk^2n+mn+m^2n^2+m+mn) = O(Kk^2n+m^2n^2)$$

After the supports of terms have been computed from the training set, the following weight will be assigned to allincoming documents d for deciding its relevance:

$$\text{Weight}(d) = \sum_{t\in T} \text{support}(t)T(t, d)$$

where, support(t) is defined in algorithm 1 (Fig. 2) and T(t, d) = 1 if tϵd otherwise T(t, d) = 0

RESULTS AND DISCUSSION

In this study, we used the German credit data text collection to evaluate get the new approach. Data preprocessing techniques (Data reduction, data selection, data cleaning, etc.) applied on text. We used WEKA (Waikato environment for knowledge analysis) and given the different results of patterns with data mining functionalities.

The data preprocessing makes the data set ready. The following data gives amount of data classified. Which is out 1000 instances the correctly classified data is 700 (70%) and incorrectly classified data is 300 (30%). Apart from the above details it tells about Kappa statistic (0), mean absolute error (0.4202), root mean squared error (0.4583), relative absolute error (100%) and root relative squared error (100%) (Algorithm 3).

Algorithm 3:

Classifier model (full training set)
=== Summary ===
Correctly Classified Instances 700: 70%
Incorrectly Classified Instances 300:30%
Mean absolute error 0.4202
Root mean squared error 0.4583
Relative absolute error 100%
Root relative squared error 100%
Total Number of Instances 1000

From Fig. 1-3, the blue color keys represents maxmimum accuracy of the out put where as red colors gives inconsistent and incomplete classification of instances. The relation can be made with the given data for having plot matrixes.

Comparative result: Many methods have been proposed to perform this task. After conducting sequence of experiments it is proved that our methods has given an effective results for discovering patterns and usage of patterns. We compared with few of the studies such as (Pr: the keyword-based method which uses Eq. 11 for term weighting. PTM: PTM model where min sup = 0.2 with pattern pruning. Rocchio: the keyword-based model using Eq. 10 to represent the concept of documents. PDR: proposed method deploying method with relevance function) and the results were given in Table 1-3 (Fig. 5-8). Number of clusters selected by cross validation: 11.

Experimental dataset: The business of banks is making loans. Assessing the credit worthiness of an applicant is of crucial importance. You have to develop a system to help a loan officer decide whether the credit of a customer is good or bad. A bank's business rules regarding loans

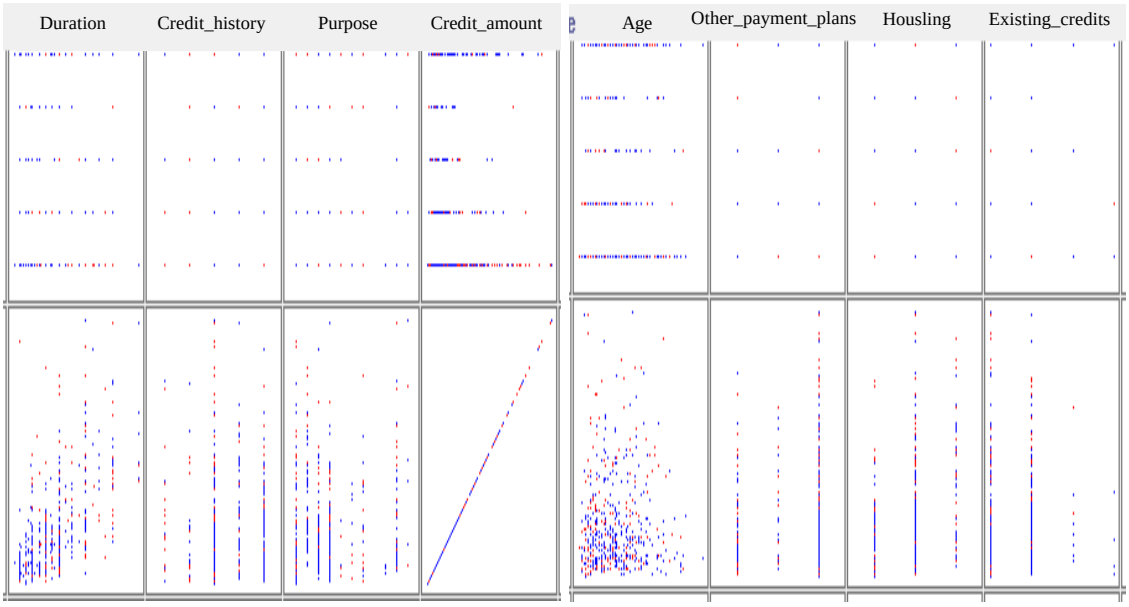


Fig. 3: Relation among all the instances in plot matrix

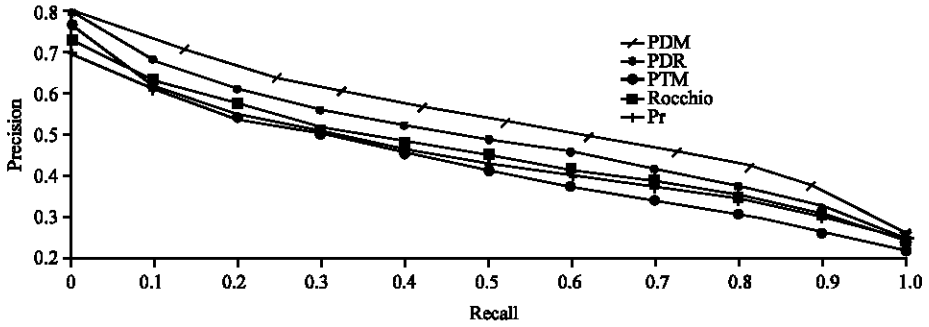


Fig. 4: Averaged 11 points plot on all German credit data set topics for all methods

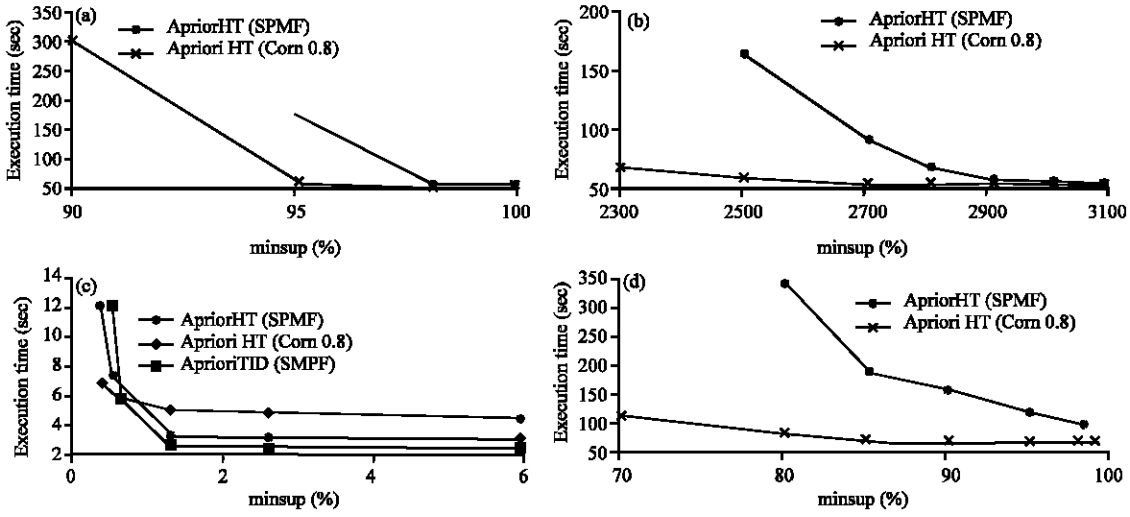
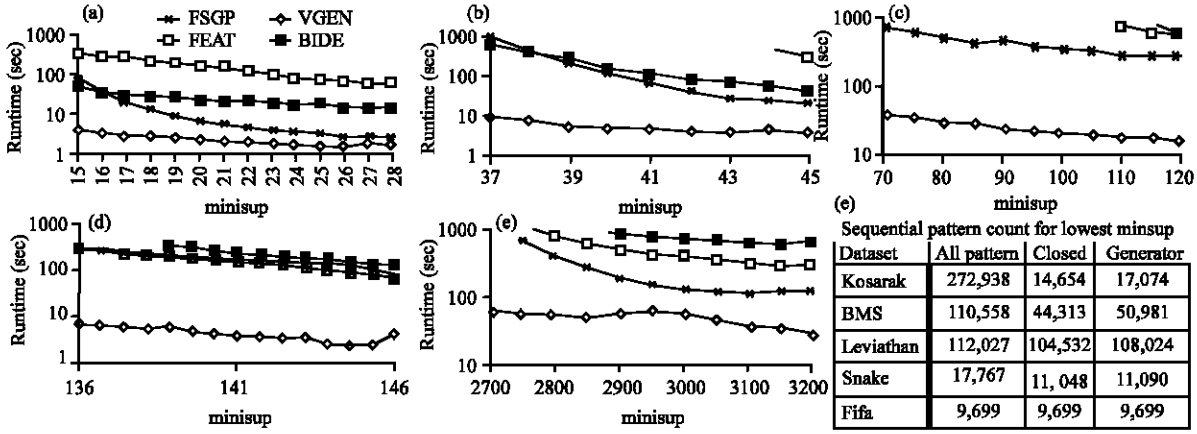


Fig. 5: Performance comparison of high utility itemset mining algorithms



(e) Sequential pattern count for lowest minsup

Dataset	All pattern	Closed	Generator
Kosarak	272,938	14,654	17,074
BMS	110,558	44,313	50,981
Leviathan	112,027	104,532	108,024
Snake	17,767	11, 048	11,090
Fifa	9,699	9,699	9,699

Fig. 6: Performance comparison of sequential generator pattern mining algorithms: a) Kosarak; b) BMS; c) Leviathan; d) Snake and e) Fifa

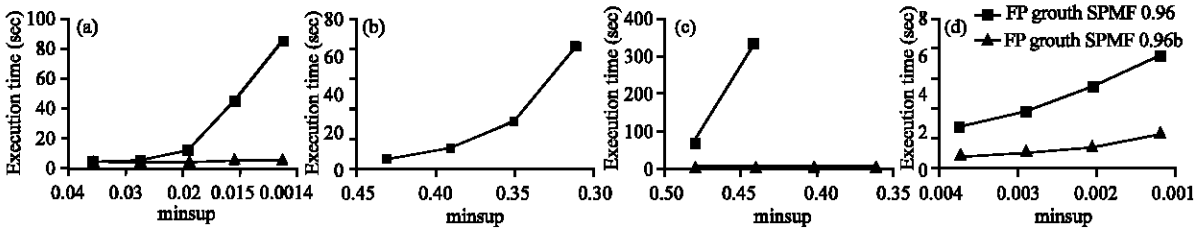


Fig. 7: Performance comparison of FP growth implementations: a) Kosarak dataset; b) Chess dataset; c) Accident dataset and d) Retail dataset

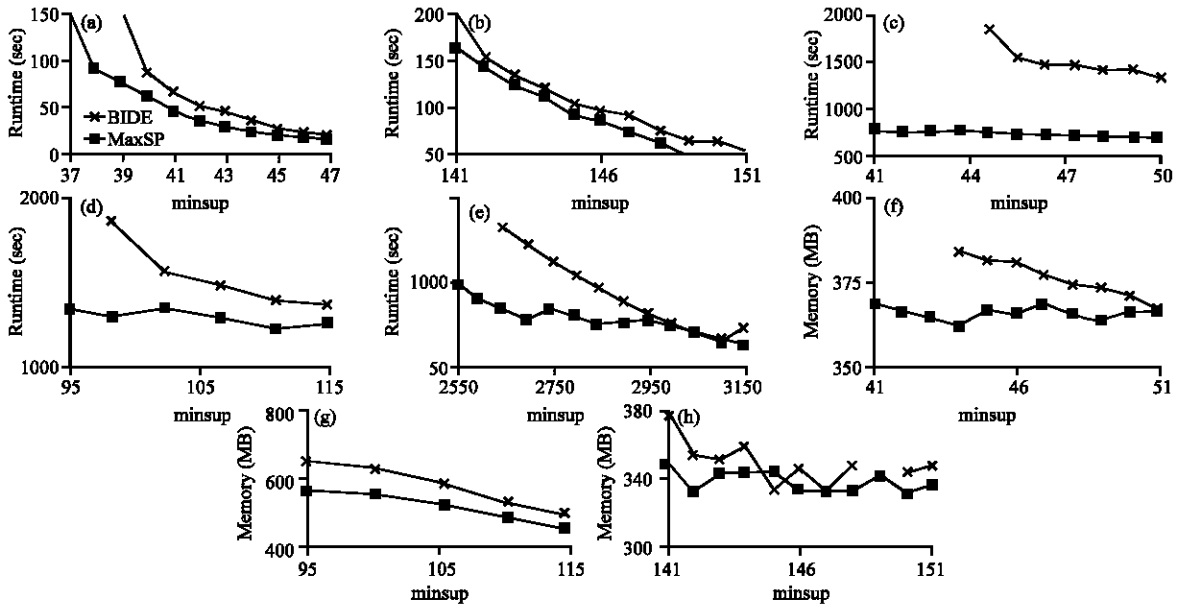


Fig. 8: a-e) Comparison of maximal sequential pattern mining algorithms: a) BMS; b) Snake; c) Sign and d) Leviathan and e) Fifa

must consider two opposing factors on the one hand a bank wants to make as many loans as possible. Interest on these loans is the banks profit source. On the other hand, a bank cannot afford to make too many bad loans

Table 3: Comparison of top-K sequential pattern mining algorithms

Datasets	No. of sequences	No. of distinct items	Avg. seq. length (items)
FIFA	20450	2990	34.74 (SD = 24.08)
Leviathan	5834	9025	33.81 (SD = 18.6)
Bible	36369	13905	21.64 (SD = 12.2)
Sign	730	267	51.99 (SD = 12.3)
Snake	163	20	60 (SD = 0.59)
BMS	59601	497	2.51 (SD = 4.85)
Kosarak 10 K	10000	10094	8.14 (SD = 22)

Too many bad loans could lead to the collapse of the bank. The bank's loan policy must involve a compromise: not too strict and not too lenient. Actual historical credit data is not always easy to come by because of confidentiality rules here is one such dataset consisting of 1000 actual cases collected in Germany credit dataset (original) Excel spread sheet version of the German credit data (Down load from web). In spite of the fact that the data is German you should probably make use of it for this assignment (Unless you really can consult a real loan officer).

CONCLUSION

The lack of significance of pattern is still an issue. Small length patterns may contains unnecessary information, hence which may causes to ineffectiveness. It is observed in our experiments text mining suffering from low frequency (from Fig. 2 out of 1000 instances correctly classified are only 700 (70%) and other are incorrectly classified, i.e., 300 (30%). Even we designed effective deploying methods the rate of useful discovered patterns are 70%. Still, there may be improvement is needed. Remaining 30% patters strictly not worthy. Even, 70% of discovered patterns are we got from which we identified few patterns not holding the useful information. Our pattern deploying refines maximum still we feel some improvements can leaving as future research and also information from non relevant documents.

REFERENCES

Agrawal, R. and R. Srikant, 1994. Fast algorithms for mining association rules. Proceedings 20th International Confernce on Very Large Data Bases (VLDB'94) Vol. 1215, September 12-15, 1994, Morgan Kaufmann Publisher, Santiago, Chile, pp: 487-499.

Huang, Y.F. and S.Y. Lin, 2003. Mining sequential patterns using graph search techniques. Proceedings of the 27th Annual International Conference on Computer Software and Applications (COMPSAC'03), November 3-6, 2003, IEEE, Dallas, Texas, USA., pp: 4-9.

Joachims, T., 1996. A probabilistic analysis of the rocchio algorithm with TFIDF for text categorization. Millersville University, Millersville, Pennsylvania.

Lang, K., 1995. News Weeder: Learning to Filter Netnews. In: Proceedings of the 12th International Conference on Machine Learning (ML'95), Prieditis, A. and S. Russell (Eds.). Morgan Kaufmann, San Mateo, California, pp: 331-339.

Lewis, D.D., 1992. An evaluation of phrasal and clustered representations on a text categorization task. Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, June 21-24, 1992, ACM, Copenhagen, Denmark, ISBN:0-89791-523-2, pp: 37-50.

Li, X. and B. Liu, 2003. Learning to classify texts using positive and unlabeled data. Proceedings of the 18th International Joint Conference on Artificial Intelligence Vol. 3, August 9-15, 2003, Acapulco Convention Center, Acapulco, Mexico, pp: 587-592.

Li, Y. and N. Zhong, 2006. Mining ontology for automatically acquiring web user information needs. IEEE. Trans. Knowl. Data Eng., 18: 554-568.

Pipanmaekaporn, L. and Y. Li, 2012. A Pattern Discovery Model for Effective Text Mining. In: Machine Learning and Data Mining in Pattern Recognition, Perner, P. (Ed.). Springer, Berlin, Germany, ISBN:978-3-642-31536-7, pp: 540-554.

Wu, S.T., Y. Li and Y. Xu, 2006. Deploying approaches for pattern refinement in text mining. Proceedings of the 6th International Conference on Data Mining (ICDM'06), December 18-22, 2006, IEEE, Hong Kong, China, pp: 1157-1161.

Zhong, N., Y. Li and S.T. Wu, 2012. Effective pattern discovery for text mining. IEEE. Trans. Knowl. Data Eng., 24: 30-44.