A new approach for efficacious pattern frame in the various categories of data in Mining

Kalyan Savalapurapu[1] D.T.V. Dharmajee Rao[2]
[1](Final Year M.Tech Student,Dept. of CSE,Aditya Institute of Technology and Management(AITAM), Tekkali,Srikakulam,Andhra Pradesh ,kalyan826@gmail.com)
[2](Professor, Dept. of CSE,Aditya Institute of Technology and Management(AITAM), Tekkali,Srikakulam,Andhra Pradesh,dharmajee@hotmail.com)

Abstract:

The discoveries on mining and to achieve the perfect and effectual pattern are still open challenge.Here in this, we proposed how to get large and best patterns from huge data sets. For that we take a new approach after cleaning the documents called PNR and DYNPRO (prefix with indexing). In DYNPRO the documents are need to be compare Asynchronously with each other for extract the best patterns with less complexity. Then we have to generate invert matrix for retrieved document. So these two algorithms results consequently will provide the best feasible data patterns/ prefix wise. The Experiment results show best and significant improvement in searching performance over different documents over the datasets. Lastly, we evaluate the impact of different interference and models on the various categories of data mining.

Key Terms*: Text classification, Invert Matrix, Pattern extraction, Document categorization.*

Introduction:

Mining is the process of discovering and retrieve optimal and meaningful knowledge in a data set. It has been successfully applied to many real-life problems, for instance, web personalization, network intrusion detection, and customized marketing. Recent advances in computational sciences have led to the application of data mining to various domains. As an area combining ideas from database systems, machine learning, and statistical learning, data mining has been successfully applied to many application domains.

In previous days, a number of data mining techniques have been proposed in order to perform different knowledge tasks. These techniques include association rule mining, frequent item set mining, sequential pattern mining, maximum pattern mining, and closed pattern mining. Most of them are proposed for the purpose of developing efficient mining algorithms to find different Patterns within a reasonable and acceptable time frame. With a large number of patterns generated by using data mining approaches, how to effectively use and update these patterns is still an open research issue.

Text mining is the discovery of interesting knowledge in text documents. It is a challenging issue to find accurate knowledge in text documents to help user's to find what they want. For that, in this paper, we propose Efficacious pattern mining which was taken as challenge to the previous techniques to retrieve best and optimal patterns as results. The information on Documents needs to be categorized as patterns in datasets. The user required documents are extracted in the best way. For

every user given query the documents are need to be filtered and associated to give best patterned results according to the requirements. These results show knowledge and understandable patterns.

Related work:

There are positive and negative document forming and this is very tedious, because once the all the documents scanned and framed these types, the combinations clustering will be done on two deferent types like positive and negative. So the categories will be repeated in the initial process itself. This issue grows and shows much impact for further pattern discovery and double repetitive things will occur.

The documents are getting compared with the regular way (inner/outer loop) which is of normal looping, which leads to self/repetitive comparisons. Which is time consuming? If the documents size is 6 i.e. the comparisons would be 6^2, which is 36.

Patterns will be discovered using normal approach with clustering and non indexing methodology, which is very time consuming process for search engines. For the current sequences, extensions would be more tedious and extra complications for further discoveries.

Effectual pattern frame:

Our proposed algorithm of DYNPRO. In this approach we will not compare then after cleaning the documents each document is compared with other document without any repetition to get the possible combinations in PNR (possible non repetitive things).

The Documents which are already existed in datasets are needed to be filtered according to Document which is already compared with source and destination.

For instance, as shown in figure.1, the Documents are compared with each other. Here the documents are categorized as positive documents and Negative Documents. The motivation will be prepared as until they have to be clean. If ***D1*** is compared with ***D2***, again ***D2*** will not be compared with ***D1***for next iterations. And also whenever the documents are getting compared source and destinations will be marked for the small and big documents. This approach simplifies for next process of combining the patterns for the fine prefix patterns. If ***D1*** is compared with ***D6*** and if founds that ***D6*** is bigger than ***D1*** the combination in the matrix will be marked as **Pn (← +)**. This is the part of PNR approach.

DYNPRO is proposed algorithm for text mining to be in order of prefixes. The threshold will be maintained though out the document before clustering to achieve the best possible sequence.

PNR for 6 available documents/categorization:

	D1	D2	D3	D4	D5	D6
D1	$	P1(→ -)	P2(→ +)	P3(→ +)	P4(→ -)	P5(→ +)
D2	(P1← +)	$	(P6→ +)	(P7→ -)	(P8→ +)	(P9→ -)
D3	(P2← -)	(P6← -)	$	P10(→ -)	(P11→ +)	(P12→ +)
D4	(P3← -)	(P7← +)	P10(← +)	$	P13(→ -)	P14(→ -)
D5	(P4← +)	(P8← -)	P11← -)	P13(← +)	$	P15(→ -)
D6	(P5← -)	(P9← +)	P12← -)	P14(← +)	P15(← +)	$

Picture 1

DYNPRO Method:

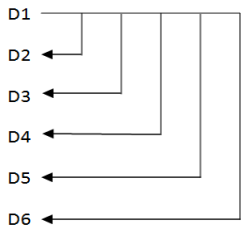
Once the documents scanned by new approach, documents will be categorized and will be cleaned in an effective way like stop words removal, marking for most frequent categories, and segmentations will be framed. These segments will be framed up with the set of documents/category wise.

Time complexity in DYNPRO: The main advantage of dynpro is , normally documents compared for seed input text to get the relative results. So while comparing with normal approach all documents will be compared in the loop with repetitive way as following.

For example we consider with ***{d1,d2,d3,d4,d5,d6}*** documents as seed documents. While comparing the documents comparision loop will start d1 which compares with itself and also compares with the rest of other documents too. When comes with the next loops the same repeats for d2 and so on which is large time complexity.

Where as coming to DYNPRO when comparision starts from document d1 , first of all self comparision is erased and also once the loop is compared the repetitive comparision flag sets , which results if in the next preceding document comparision with others found the flag rejects for the repetitive comparision. This reduces lot of time consumption for documents comparision.

Following is the practical approach for document comparision.



DYNPRO Algorithm Process:

1. We need to consider number of Datasets with number of different documents. Each document contains data with different patterns and also with different sizes.
2. We have to apply DynPro Algorithm for each data set.
3. The DynPro algorithm is used to compare the Documents to calculate time complexity for getting best document.

4. Compare Documents in every Dataset by taking the documents asynchronously.
5. Apply step4, 5, 6 to all datasets for which were we taken.
6. Then take time complexity from all datasets. And get best time complexity (low) from them.
7. Generate graph for Time complexity of all datasets.
8. Then get large, best and less time complexity Document based on the Graph.
9. Take that large document with less time complexity among datasets.

Terminology:

1) $\sum_0^{n-1} D \rightarrow$ Total documents
2) $\sum_0^{i} P_d \rightarrow$ Total positive documents
3) $\sum_0^{j} P_n \rightarrow$Total negative documents
4) $\phi \rightarrow$ { neutralization (nullify) / No operation
5) MI ← insert matrix
6) Size (d) ← size of the document
7) Mark(d) ← marking the document
8) $\sum_0^{n-1} M \leftarrow$ Marked documents
9) $\sum_0^{k} C_m \leftarrow$documents to be clustered
10) $F_{cd}\leftarrow$ final clustered document

Algorithm:

DYNPRO: Marking

Input: Documents of Datasets.

Output: patterns set

1. Initialization
2. Count ←0
3. $P_d\leftarrow 0$
4. $P_n\leftarrow 0$
5. for each d (document) in D
6. Loop start:
7. If size (d_i) < size (d_{i+1})
8. $P_n\leftarrow d_i$
9. Count++;
10. M ← Mark (d_i)
11. Else if size (d_i)> size (d_{i+1})
12. $P_d\leftarrow d_i$
13. Count ++
14. M← Mark (d_i)
15. End if
16. Count← 0 for each d in M
17. MI ← $P_n \Omega P_d$
18. Count++; // count increment
19. End for

Document Clustering:

After cleaning the documents by DynPro Algorithm, the compared documents which

are optimized as best results to show patterns are clustered. This technique is for close the labeled documents at a place for show results.

We simulate for the availability of 6 documents (assumption), the picture1 shows the combinations to compare for further combinational documents. The first step of this process (P) of PNR is to get the weight of the source document (***D1***) to destination document (***D3***). If source is having the higher weight than destination, then the result in matrix box is ***P2 (→+)***. So by getting this type of results, it is very easy for further clustering process/ patterns by using initial words like prefixes.

Dynpro discover invert matrix for best available combinations for available prefixes. After invert matrix generation, by using a fixed threshold, the feasible/effective pattern/prefix will be generated. This approach is low cost solution for prefix mining after PNR. In Invert matrix the associations will be find out for optimal text patterns retrievals.

DYNPRO: Clustering

1. *Count← 0*
2. *For each k in M*
3. *Temp ← null*
4. *Temp ← cluster (temp,d_k)*
5. *Count ++*
6. *End loop*
7. *F_{cd}← temp;*

Invert Matrix Generation:

We simulate for the availability of 6 documents (assumption), the picture1 shows the combinations to compare for further combinational documents. The first step of this process (P) of PNR is to get the weight of the source document (***D1***) to destination document (***D3***). If source is having the higher weight than destination, then the result in matrix box is ***P2(→+)*** . So by getting this type of results, it is very easy for further clustering process/ patterns by using initial words like prefixes. Dynpro discover invert matrix for best available combinations for available prefixes. After invert matrix generation, by using a fixed threshold, the feasible/effective pattern/prefix will be generated. This approach is low cost solution for prefix mining after PNR.

In PNR, Invert matrix generation will be done by utilizing the documents which are cleaned by DYNPRO mechanism. The Invert matrix is used for evaluates various patterns of particular Document. The text data with non duplication patterns of each document are applied with this mechanism. The rows and columns will show with text data which was existed in documents. Column data will be indexed with numbers randomly. According to the text data by indexing we can find out the association patterns for required query which is given by the user for getting best search. The repeated terms in a document are discovering with the frequency that's what we get in invert matrix calculation. Based on results the invert matrix shows efficacious and optimal patterns.

There will not be any dataset formations. So the best possible matrices will be generated to achieve the best possible patterns among the documents. The DYNPRO approach is the best effective pattern for search engines.

After that the sequences will be clustered to get the final patters. So the result will be maintaining the offsets and also best possible and fine patterns.

Example:

"One four three five two three five four two one four two one three six four one three two five six three two" -- text for example(etext).

If the above text is example for invert matrix , mainly all unique items will be retrieved in the first step. After that all unique items will be identified in the sequence with randomly generated numbers with in unique items count range. If in the above example ***{one , two , three , four, five ,six}*** then each unique item will be identified with random number between 1 to 6(n) . In this case n is 6 which is total unique items count. This identification number for unique items is dynamic, the reason behind this is once dynpro approach generates the text the system cannot estimates the fixed count of unique items. This varies for each search. So the system automatically generates the identification for unique/distinct items.

	One	two	three	four	five	six
One – 1	5,1	5,3	2,3	2,4	$	$
Two – 4	2,1	1,1	1,2	6,3	$	$
Three -2	6,1	6,2	3,1	4,4	4,5	$
Four -5	2,1	4,2	4,3	1,3	$	$
Five -6	4,1	5,4	3,2	$	$	$
Six -3	5,2	2,2	$	$	$	$

Figure.2

Algorithm for Invert Matrix:

Dp= paragraph in document.

Dp(t)←terms in document

Step1:

Take the input as a document consisting of the stop words(the document which was positive).

Initialization:

Dp(t)←0

I←0(index)

T←0(terms)

Step2:

loop starts //take the next term of the term in document

i←0

Next (ti)←dp(t)

Ai←next (ti)

i++;

End loop

step3:

loop starts //take the index values as randomly

j←0 to n

x←random(j);

j++;

End loop

step4:

loop starts // for frequency value generation

Count←0

f←0

step5:

if // condition starts

Index (ti) ==index (ti)

Count++

f←count

Return f;

End loop

step6:

if //condition starts

(ti,ai)∩dp(t)=Ǿ;

Then

[(ti,x),ti]→$

End if

Step7:

loop starts // generate inverse matrix for rows

i←0

i<=size (t),

Increment i upto the size will compete.

Step8:

loop starts // for column terms of matrix

J←(ti,x)←0

j<=size (t)

Increment j upto the size will compete

Step9:

compare ([(ti,x),ti]) then

(ti,ai)→[Index (ai),f]

[(ti,x),ti]←[Index (ai),f];

Compare [(ti,x),ti+1]then

(ti,ai+1)→[Index (ai+1),f]

[(ti,x),ti+1]←[Index (ai+1),f]

……

……

……

Compare [(ti,x),tn]then

(ti,an)→[Index (an), f]

[(ti,x),tn]←[Index (an), f]

Step10:

endloop

Endloop

Experimental Results:

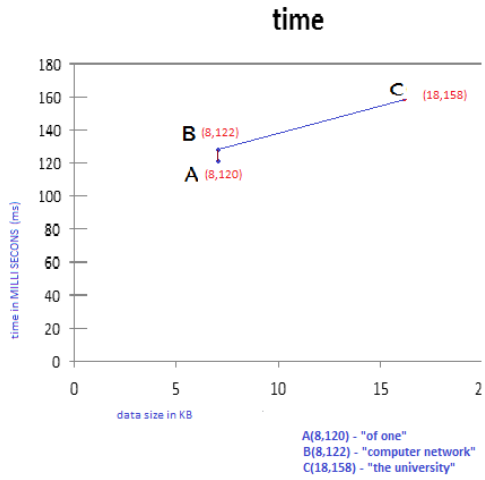
SEED DATA	SEARCH DATA STRING	SIZE
higher edu.txt , primary edu.txt , private edu.txt , secondary edu.txt	The university	18
data s1 , data s2 , data s3	of one	8
data s1 . data s2 . data s3	computer network	8

The above table contains 3 experimental results.

Data sets:

Ds1: {*higher edu.txt , primary edu.txt , private edu.txt , secondary edu.txt*}
Ds2: {*data s1 , data s2 , data s3*}
Ds3: {*data s1 , data s2 , data s3*}

For the above data sets we have the search engine data{ "The university" , "of one" "computer network"} sequentially with {18,8,8}kb of total data for processing and {158,120,122} milliseconds for processing.



Above is the result graph for total data processing time complexity. As the dynpro reduces repetitive comparisons and reduces the time of processing(document comparisons) the above graph is consolidated report for 3 sets of experimental results for various search options.

After the dynpro the invert matrix generation of experimental input is as follows:

"php java .net c java oracle php .net php c java .net oracle" , if the this is dynpro output and invert matrix output is in the following picture.

```
php java .net c java oracle php .net php c java .net oracle
[c, oracle, php, java, .net]
{c=1, php=3, java=4, .net=5, oracle=2}
c: ->    [4,1][4,2]
oracle: ->    $
php: ->    [4,1][5,2][1,3]
java: ->    [5,1][2,2][5,3]
.net: ->    [1,1][3,2][2,3]
```

Invert matrix approach can be processed for large dynpro output with less time complexity.

Conclusion:

The techniques included in association rule mining, frequent item set mining, sequential pattern mining, maximum pattern mining, and closed pattern mining are closely related with this proposed one. However, using these discovered knowledge (or patterns) in the field of text mining is difficult and ineffective. By this proposed algorithm in this paper we could reduce the complexity in extracting the data. In this research work, effective pattern discovery technique has been proposed to overcome the low-frequency and misinterpretation problems for text mining. The proposed technique uses two processes, pattern deploying and pattern evolving, to refine the discovered patterns in text documents with low complexity.

References:

1. Bi, Y., Wu, S., Wang, H., Guo, G.: Combination of evidence-based classifiers for text categorization. In: 2011 23rd IEEE International Conference on Tools with

Artificial Intelligence, pp. 422–429. IEEE (2011)

2. Jaillet, S., Laurent, A., Teisseire, M.: Sequential patterns for text categorization. Intelligent Data Analysis 10(3), 199–214 (2006)

3. Nanas, N., Vavalis, M.: A "Bag" or a "Window" of Words for Information Filtering? In: Darzentas, J., Vouros, G.A., Vosinakis, S., Arnellos, A. (eds.) SETN 2008. LNCS (LNAI), vol. 5138, pp. 182–193. Springer, Heidelberg (2008)
4. Shehata, S., Karray, F., Kamel, M.: A concept-based model for enhancing text categorization. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 629–637. ACM (2007)
5. Yan, X., Cheng, H., Han, J., Xin, D.: Summarizing item set patterns: a profile-based approach. In: 11th ACM SIGKDD International Conf. on Knowledge Discovery in Data Mining, pp. 314–323 (2005)

