Information Retrieval with Effective Web Document Clustering Using FP growth Algorithm

O. Prudhi, S. Muni Kumar

1Student, MCA, KMM Institute of Post Graduate Studies
2Associate Professor, MCA, KMM Institute of Post Graduate Studies

Abstract—The size of web has increased exponentially over the past few years with thousands of documents related to a subject accessible to the user. With this much quantity of data accessible, it's unfeasible to take the total advantage of the planet wide net without having a correct framework to search through the accessible data. This requisite organization can be done in many ways. During this paper we tend to introduce a combine approach to cluster the online pages that first finds the frequent sets so clusters the documents. These frequent sets area unit generated by using Frequent Pattern growth technique. We tend to found clusters having documents that are highly connected and have similar features. Experimental results show that our approach is more efficient.

Index Terms—Clusters, Frequency pattern growth algorithm, web.

I. INTRODUCTION

Data mining techniques is used to extracting information from that data. Many researchers developed a lot of algorithm and techniques for finding useful information in the database. Frequent item-set mining is a core data mining operation and has been extensively studied over last decade. It plays an essential role in many important data mining tasks. Algorithms for frequent item-set mining form the basis for algorithms for a number of other mining problems, including association rule mining, correlations mining, and mining sequential and emerging patterns. Association rule mining technique is very effective data mining technique to finding the useful hidden information in the data, its aim to extract correlation, frequent pattern, association by transaction database or other data repository. This rules generated by datasets and it derives by measurement of support and confidence of every rule, which define the frequency of that role. Association rule is a rule which is depends on the association relationship of objects and items. For example data item interrelation ship which is occurs simultaneous with other data items. This rule is calculated by data and association calculated with the help of probability. Share market and recommended system etc. are its practical applications. To concept of document based clustering, association rule mining is very effective algorithm and useful approach. It utilizes that FP-Growth approach discovers frequent patterns and modified it to frequent sub graph discovery. Originally this algorithm is design to frequent item-set mine in market basket analysis. In this paper, analyze FP-Growth approach to document clustering. For graph mining, change in FP-Growth algorithm which discover frequent connected graph and perform better for thick connected graph.

A beeline of research is in the area of application of machine learning and information technology on web platforms. The data from the web are rather more of complex nature to handle as they are generated asynchnronously. The web data is more of dynamic, unstructured with complex attributes. To deal with web data of complex attributes we need a finer representation of these data. Bag of words representation commonly used cannot in these cases work intelligently due to synonym problem and polyevery problem. For instance having a word "semantics" in one and "meaning" in another in the normal syntactic search parlance is considered as two different words with no semantic relation, adding up to two different frequency words. In the vice versa if a word "bear" appears as a noun and as a verb the existing system fails in identifying and counts the frequency to two .In this proposed system first measure taken to confront this issue of synonymy is by using concept vectors i.e a single concept vector mapping "semantic" and "meaning" to a common word could help resolve the issue.
II. PROPOSED ALGORITHM

FP-growth Algorithm:-
Input: A transaction database D and a minimum support threshold ξ.
Output: FP-tree, the frequent-pattern tree of D.
Method: The FP-tree is constructed as follows.
1. Collect the set of frequent items (Fitems) and their support counts after scanning the transaction database (D) once. Sort Fitems according to descending support count as Lfreq, the list of frequent items.
2. Create the root of an FP-tree, and label it as "null". For each transaction Itrans in D do the following, Select and sort the frequent items in Itrans according to the order of Lfreq. Let the sorted frequent list in Itrans be [e | Elist], where e is the first element and Elist is the remaining list. Call insert_tree ([e | Elist], T), which is performed as follows.

Procedure insert_tree ([e | Elist], T)
if T has a child N such that N.item-name = e.itemname, then increment N’s count by 1; else create a new node N, and let its count be 1, its parent link be linked to T, and its node-link to the nodes with the same item-name via the node-link structure. If Elist is nonempty, call insert tree (Elist, N) recursively.

Algorithm 2: (FP-growth: Mining frequent patterns with FP-tree by pattern fragment growth)
Input: A database D, represented by a FP-tree constructed according to Algorithm 1, and a minimum support threshold ξ.
Output: The complete set of frequent patterns.
Method: call FP-growth (FP-tree, null).
Procedure FP-growth (Tree, α)
{
(1) If Tree contains a single prefix path
(2) Then {
(3) Let P be the single prefix-path part of Tree;
(4) Let Q be the multipath part with the top branching node replaced by a null root;
(5) For each combination (denoted as β) of the nodes in the path P do
(6) Generate pattern β | α with support = minimum support of nodes in β;
(7) Let freq_pattern_set(P) be the set of patterns so generated;
(8) Else let Q be Tree;
(9) For each item ai in Q do {
(10) Generate pattern β = ai \cup α with support = ai. support;
(11) Construct β’’s conditional pattern-base and then β’’s conditional FP-tree Treeβ’’;
(12) If Treeβ’’ ≠ Φ (13) then call FP-growth (Treeβ, β);
(14) Let freq_pattern_set (Q) be the set of patterns so generated;
(15) Return (freq_pattern_set (P) \cup freq_pattern_set (Q) \cup (freq_pattern_set(P) \times freq_pattern_set(Q)))
}
}

III. EXPERIMENTS AND RESULTS

This section, we briefly review the K-means and cosine similarity techniques for comparison with our approach. The discussions on performance evaluation presented at the end.

K-means Technique:
It consists the following steps:
1. Take the data space, which needs to be clusters.
2. Pre-determine the number of clusters ask.
3. Initialize k- means for the data, viz, m1, m2.
4. Using Euclidean distance, find the distance of data points from the mean.
5. Group the data points having minimum distance to the mean, to the corresponding mean.
6. Calculate the new mean of the each group formed in step 5.
7. Repeat steps 4-6 until the new mean formed is same as the previous mean.

Cosine similarity Technique:
1. Find out the initial clusters, Ci, using steps 1-3 of proposed approach.
2. Find out the centers of each Ci using steps 4 and 5 of proposed approach.
3. Check for the uncluster documents, UDk, from the document set D mentioned in proposed approach.
4. Find the similarities of UDk to every Ci’s centers using Eq. 1.
5. Assign the document to that cluster which has maximum similarity. In case similarities between two or more clusters are same for any document, then assign the document to anyone these clusters.
6. Repeat the steps 4 and 5 till all UDk assigned to their respective Ci.
Discussion on Performance Evaluation:
We have used the following metrics namely, Entropy and Purity for evaluating the performance of the proposed approach. The evaluation metrics are given below,

\[ \text{entropy} = \sum_{i=1}^{K} \frac{m_i}{m} E_i, \]

where \( K \) is the number of clusters and \( m \) is the total number of documents. \( m_j \) is the number of documents in cluster \( j \).

\[ E_i = -\sum_{j=1}^{C} \pi_{ij} \log_2 \pi_{ij}, \]

where \( C \) is the number of classes, \( \pi_{ij} = \frac{m_{ij}}{m_i} \), where \( m_{ij} \) is the number of documents of cluster \( i \) belongs to class \( j \).

\[ \text{purity} = \sum_{i=1}^{K} \frac{m_i}{m} \pi_i, \]

where \( \pi_i = \max_j \left( \pi_{ij} \right) \) and all terms having same meaning as above.

We have compared our results with both K-means and cosine similarity based clustering techniques, where both have been combine with FP-growth algorithm has been used with minimum support. The FP-growth algorithm used to give the number of frequent item sets which in turn gives the number of clusters to be form for K-means and each set center taken as the initial centroid for K-means. This concepts also used for cosine_similarity, where each document which is more similar to the center of a cluster will be added to that cluster by using Eq. 1. The details are listed in appendix. After comparison, we found the following results.

Final Clusters using FP-growth + Cosine_Similarity are
\{D1, D2, D3, D4, D6\}, \{D5, D7, D8, D9, D10\}

Final Clusters using FP-growth+ K-means are
\{D1, D2, D3, D4, D6\}, \{D5, D7, D8, D9, D10\}

### FP-growth+ Cosine_Similarity

<table>
<thead>
<tr>
<th>Clusters</th>
<th>Soci al N/w Class</th>
<th>Compu ter N/W Class</th>
<th>To tal</th>
<th>Entr opy</th>
<th>Puri ty</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster 1 {D1,D2,D3,D4,D6}</td>
<td>5</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Cluster 2 {D5,D7,D8,D9,D10}</td>
<td>1</td>
<td>4</td>
<td>5</td>
<td>.7</td>
<td>.8</td>
</tr>
<tr>
<td>Total</td>
<td>6</td>
<td>4</td>
<td>10</td>
<td>.35</td>
<td>.9</td>
</tr>
</tbody>
</table>

### Overall Comparison

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Entropy</th>
<th>Purity</th>
</tr>
</thead>
<tbody>
<tr>
<td>FP-growth+ Cosine_similarity</td>
<td>.35</td>
<td>.9</td>
</tr>
<tr>
<td>FP-growth+ K-means</td>
<td>.35</td>
<td>.9</td>
</tr>
</tbody>
</table>

Finding Initial Cluster Centers:-
For finding number of clusters and initial cluster centers, FP-growth algorithm for finding frequent item-sets has been used. The frequent sets generated are of frequency greater than the minimum support supplied by the user. In the generated frequent sets of documents, the terms are taken to be the transactions and the documents are the items of the transactions. In this way the frequent sets generated are the ones which have particular set of terms in common and hence are closely related. This helps by deciding the number of clusters and also the centers of these clusters which is simply the centroid of the respective frequent item-set.

### IV. CONCLUSION

The projected approach used FP-growth algorithmic rule for clustering the web documents. This approach keeps the connected documents within the same cluster in order that looking of documents becomes additional efficient. Experimental results show that our approach of FP-growth algorithmic rule provides higher results in terms of entropy and
purity compare to ancient K-means and cosine similarity techniques for clustering the net documents.

REFERENCES


