

# Co-Preserving Patterns in Bipartite Partitioning for Topic Identification

Tianming Hu \*

Hui Xiong †

Sam Yuan Sung ‡

## Abstract

The claimed advantage of describing a document data set with a bipartite graph is that partitioning such a graph yields a co-clustering of words and documents. The topic of each cluster can then be represented by the top words and documents that have highest within-cluster degrees. However, such claims may fail if top words and documents are selected simply because they are very general and frequent. In addition, for those words and documents across several topics, it may not be proper to assign them to a single cluster. To that end, this paper introduces a new bipartite formulation that incorporates both word hypercliques and document hypercliques as super vertices. By co-preserving hyperclique patterns during the clustering process, our experiments on real-world data sets show that better clustering results can be obtained and the cluster topic can be more precisely identified. Also, we illustrate an application of the partitioned bipartite to search engines, returning clustered search results for keyword queries. We show that the topic of each cluster with respect to the current query can be identified more accurately with the words and documents from the patterns than with those top ones from the standard bipartite formulation.

## Keywords

Hyperclique Pattern, Pattern Preserving, Bipartite Partitioning, Co-Clustering, Topic Identification

## 1 Introduction

In text categorization, typically the data is arranged as a word-document co-occurrence matrix. Most clustering algorithms focus on one-way clustering, i.e., cluster one dimension of the table based on similarities along the second dimension. Such a duality between document and word clustering can be naturally formulated in a bipartite graph, with documents and words modeled as vertices on two sides respectively [1]. Finding an

optimal partitioning in such a bipartite gives a co-clustering of documents and words. It is expected that top documents and words in the same cluster can represent its topic, where top vertices usually refer to those with highest within-cluster degrees.

However, such claims may fail if the cluster is not pure enough or it includes words/documents across multiple topics. Some documents are top simply because they contain many general words with high degrees. Others may span several topics and it is improper to give them a hard classification. When it comes to words, it gets worse. Quite a few words come with multiple meanings, hence it is unreasonable to classify them to a single class. For instance, given a collection of documents with topics including business and health, it may not be appropriate to assign word ‘cell’ to a single class. In fact, it can appear in documents of any topic, with meaning ‘cell phone’ or ‘cancer cell’.

To perform natural clustering and to precisely capture the cluster topic, first we need to identify those micro-sets of words/documents that are very similar among themselves and, as whole, representative of their corresponding topics. Meanwhile, we need to ensure that they would not be separated into different clusters during the clustering process. Second, as for those documents and words across several topics, they should be allowed to go to more than one cluster.

In this paper, we exploit hyperclique patterns [8] to define such micro-sets. Hyperclique patterns truly possess such desirable property: the objects in a hyperclique pattern have a guaranteed level of global pairwise similarity to one another as measured by the cosine or Jaccard similarity measures [9]. We propose a new bipartite formulation for co-preserving patterns, where word hypercliques and document hypercliques are represented by super vertices on two sides of the bipartite respectively. Our approach, CO-preserving PATterns in bipartite PARTitioning (COPAP), is compared with the standard bipartite formulation on real-world document data sets from different domains. The experimental results show that we can make improvement on clustering results in terms of various external measures and the topic can be identified more precisely.

---

\*DongGuan University of Technology, tmhu05@gmail.com

†Rutgers University, hxiong@andromeda.rutgers.edu

‡South Texas College, sysung@southtexascollege.edu

Finally, due to the high affiliation within hyperclique patterns, the pattern preserving partitioned bipartite naturally lends itself to various applications in search engines. For instance, instead of a long ranked list for keyword queries, it is better to return clustered search results by topics. We demonstrate such an application of the COPAP method and show that the topic of each cluster with respect to the current query can be identified more accurately with the words and documents from the patterns than with those top ones from the standard bipartite formulation.

**Overview.** The rest of this paper is organized as follows. Section 2 describes background and related work. In Section 3, we introduce the details of the COPAP method. Section 4 describes an application of the COPAP method in search engines. Experimental results of co-clustering are reported in Section 5, together with a demonstration on returning clustered search results. Finally, we draw conclusions in Section 6.

## 2 Background and Related Work

In this section, we describe related work and introduce some background information including document clustering and hyperclique patterns.

### 2.1 Document Clustering

In general, clustering algorithms can be divided into two categories: hierarchical and partitional. Using highly affiliated subsets of documents as starting points, [7] proposed the HIERarchical Clustering with PAttern Preservation (HICAP) algorithm and showed that its clusters are more interpretable than other hierarchical methods. As for the partitional category, probably K-means is the most widely used method. As a modification, bisecting K-means has been proposed in hierarchical clustering of documents and produces competitive results [6]. Graph-theoretic techniques have also been considered for clustering. They model the word-document datasets by a graph whose vertices correspond to documents or words. The duality between document and word clustering can be naturally modeled using a bipartite, where documents and words are modeled as vertices on two sides respectively [1].

### 2.2 Hyperclique Patterns

In this paper, hyperclique patterns are what we preserve during clustering. They are based on the concepts on frequent itemsets. Let  $I = \{i_1, i_2, \dots, i_m\}$  be a set of distinct items. Each transaction  $T$  in database  $D$  is a subset of  $I$ . We call  $X \subseteq I$  an itemset. The support of  $X$ , denoted by  $supp(X)$ , is the fraction of transactions containing  $X$ . If  $supp(X)$  is no less than a user-specified threshold,  $X$  is called a frequent itemset. The

confidence of association rule  $X_1 \rightarrow X_2$  is defined as  $conf(X_1 \rightarrow X_2) = supp(X_1 \cup X_2) / supp(X_1)$ . To measure the overall affinity among items within an itemset, the h-confidence was proposed in [8]. Formally, the h-confidence of an itemset  $P = \{i_1, i_2, \dots, i_m\}$  is defined as  $hconf(P) = \min_k \{conf(i_k \rightarrow P - i_k)\}$ . Given a minimum threshold  $h_c$ , an itemset  $P \subseteq I$  is a hyperclique pattern if and only if  $hconf(P) \geq h_c$ . A hyperclique pattern  $P$  can be interpreted as that the presence of any item  $i \in P$  in a transaction implies the presence of all other items  $P - \{i\}$  in the same transaction with probability at least  $h_c$ . A hyperclique pattern is a maximal hyperclique pattern if no superset of this pattern is a hyperclique pattern.

### 2.3 Applications to Search Engines

Pattern preserving partitioned bipartites can also play a role in search engines. Instead of returning a long ranked list of documents for keyword queries, it is better to give the user a quick view of the whole results, say, by returning clustered search results by topic. [3] returns a set of topic sensitive lists by computing a set of PageRank vectors biased using a set of representative topics. Vivisimo(<http://vivisimo.com>) provides clustered search results based on distinct frequent words. Within the cluster, the documents are still shown according to their original ranks. However, a frequent word may not represent a topic and it may even be meaningless. Here for each topic(cluster), we can show only the documents in the patterns and use them for generating topical words.

## 3 COPAP: Co-Preserving Patterns in Bipartite Partitioning

Our approach COPAP is based on the bipartite graph partitioning with hyperclique patterns as super vertices. So the objects in the hyperclique pattern will not be separated during graph partitioning. Figure 1 gives the overview of the algorithm. Detailed description is given later in this section.

### 3.1 Mining Maximal Hyperclique Patterns

To apply clustering algorithms, a document data set is usually represented by a matrix by extracting significant words from documents. The matrix  $A$ 's non-zero entry  $A_{ij}$  indicates the presence of word  $w_i$  in document  $d_j$ , while a zero entry indicates an absence. Given  $A$ , if we treat words as transactions and documents as items, we can find maximal hyperclique patterns of documents. Next, we transpose  $A$ , where each row/transaction is for a document and each column/item for a word. In this case, we can identify maximal hyperclique patterns of words. For mining maximal hyperclique patterns, we

**Input:**

- $D$ : a data set represented by a word-document matrix.  
 $\alpha_w$ : a minimum support threshold for words.  
 $\theta_w$ : a minimum h-confidence threshold for words.  
 $\alpha_D$ : a minimum support threshold for documents.  
 $\theta_D$ : a minimum h-confidence threshold for documents.  
 $K$ : the desired number of clusters.

**Output:**  $C$ : the resulting result.

**Variables:**

- $MD$ : the set of maximal document hypercliques.  
 $LD$ : the set of documents not included in  $MD$ .  
 $MW$ : the set of maximal word hypercliques.  
 $LW$ : the set of words not included in  $MW$ .  
 $BG$ : a bipartite graph.

**Steps**

1.  $MD = \text{MaximalHypercliquePattern}(\alpha_D, \theta_D, D)$
2.  $LD = \text{UncoveredObjects}(MD, D)$
3.  $MW = \text{MaximalHypercliquePattern}(\alpha_w, \theta_w, D^T)$
4.  $LW = \text{UncoveredObjects}(MW, D)$
5.  $BG = \text{BipartiteGraph}(MW, LW, MD, LD, D)$
6.  $C = \text{GraphPartition}(BG, K)$

Figure 1: Overview of the COPAP Algorithm

employ a hybrid approach proposed in [4], which exploited key advantages of both the depth first search strategy and the breadth first search strategy for efficient computation.

### 3.2 Generating the Bipartite

First some notations for general graph representation. A graph  $G = (V, E)$  is composed of a vertex set  $V = \{1, 2, \dots, |V|\}$  and an edge set  $\{(i, j)\}$  each with edge weight  $E_{ij}$ . The graph can be stored in an adjacency matrix  $M$ , with entry  $M_{ij} = E_{ij}$  if there is an edge  $(i, j)$ ,  $M_{ij} = 0$  otherwise.

Given the  $m \times n$  word-by-document matrix  $A$ , the standard bipartite graph  $G = (V, E)$  is constructed as follows. First we order the vertices such that the first  $m$  vertices index the words while the last  $n$  index the documents, so  $V = V_W \cup V_D$ , where  $V_W$  contains  $m$  vertices each for a word, and  $V_D$  contains  $n$  vertices each for a document. Edge set  $E$  only contains edges linking different kinds of vertices, so the adjacency matrix  $M$  may be written as  $\begin{pmatrix} 0, A \\ A^T, 0 \end{pmatrix}$ . In our case, with the word hyperclique set  $MW$  and the document hyperclique set  $MD$ , we first identify those remaining words  $LW$  that never appear in  $MW$  and

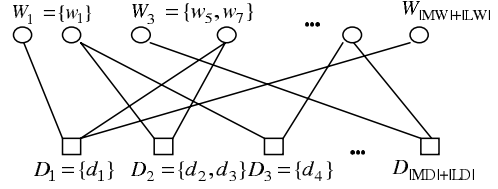


Figure 2: The bipartite with meta-words and meta-documents.

those remaining documents  $LD$  that never appear in  $MD$ . Then we construct vertex set  $V = V_W \cup V_D$  as follows.  $V_W$  contains  $|MW| + |LW|$  vertices each for a meta-word, i.e., either a word pattern in  $MW$  or a single word in  $LW$ .  $V_D$  contains  $|MD| + |LD|$  vertices each for a meta-document, i.e., either a pattern in  $MD$  or a document in  $LD$ . An example bipartite is shown in Figure 2, where there are pattern vertices on both sides. The new  $(|MW| + |LW|) \times (|MD| + |LD|)$  meta-word by meta-document matrix  $A'$  is defined as  $A'_{ij} = \sum_{w_k \in W_i, d_l \in D_j} A_{kl}$ . That is, the association between meta-word  $W_i$  and meta-document  $D_j$  is the sum of association between all words  $w_k$  in  $W_i$  and all documents  $d_l$  in  $D_j$ .

### 3.3 Graph Partitioning

Given a weighted graph  $G = \{V, E\}$  with adjacency matrix  $M$ , clustering the graph into  $K$  parts means partitioning  $V$  into  $K$  disjoint clusters of vertices  $V_1, V_2, \dots, V_K$ , by cutting the edges linking vertices in different parts. The general goal is to minimize the sum of the weights of those cut edges. To avoid trivial partitions, often the constraint is imposed that each part should be roughly balanced in terms of part weight  $wgt(V_k)$ , which is often defined as sum of its vertex weight. Here we employ Graclus [2], a fast kernel based multilevel algorithm, which involves coarsening, initial partitioning and refinement phases.

## 4 Applications to Clustered Search Results

The partitioned bipartite naturally lends itself to different clustering-related functions in search engines. In this section, we describe its applications to returning clustered search results.

As described in Figure 3, this job can also be done by the standard bipartite formulation. First we retrieve the set of documents  $D(q)$  that contains query  $q$  and then partition it into groups  $\{G\}$  according to the partitioned bipartite. The subsequent work is performed cluster by cluster. For representative documents, we directly select top documents from the cluster. When it comes to words, we give priority to

**Input:**

$D$ : a dataset represented by a word-document matrix.  
 $C$ : the partitioned bipartite containing the clustering results.  
 $q$ : a query word.  
 $k_1/k_2$ : the number of words/documents returned for each cluster.

**Output:** the clustered search results.

**Steps**

1. According to  $D$ , retrieve the documents  $D(q)$  that contains  $q$ .
  2. According to the cluster label in  $C$ , partition  $D(q)$  into groups  $\{G\}$ , and keep those groups of size larger than  $k_2$ .
  3. **For** each group  $G$  **Do**
  4.     Return top  $k_2$  documents from  $G$ .
  5.     Compute  $W(G)$ , words shared by all documents in  $G$ .
  6.     **If**  $|W(G)| \geq k_1$  **Then**
  7.         Return top  $k_1$  words from  $W(G)$ .
  8.     **Else**
  9.         Return  $W(G)$ , and other top  $k_1 - |W(G)|$  words from  $G$ .
- End of for**

Figure 3: The standard bipartite algorithm for returning clustered search results.

those words shared by all documents in  $G$ (lines 5-8).

The counterpart in the bipartite with co-preserved patterns is more complicated, since we want to focus on those words/documents from patterns. The detailed procedure is shown in Figure 4. Within each group  $G$ , we first check if query  $q$  appears in (multiple)super vertex of word hypercliques. If yes, the words from the hypercliques receive priority of being selected(lines 5-6) and then we try to output any document that completely contains any single word hyperclique(lines 7-8). If not, we check if  $G$  contains(multiple)super vertex of document hypercliques. In this case, the document from the hypercliques are returned first(lines 11-12) and the words shared by such documents also get selected(lines 13-14). When the flow comes to line 16, it means that  $q$  appears in no word hypercliques and  $G$  contains no document hypercliques, then the word/document selection procedure is like the standard bipartite.

## 5 Experimental Evaluation

In this section, we present an experimental evaluation of COPAP. First we introduce the experimental datasets

Table 1: Characteristics of data sets.

data	RE0	RE1	K1	WAP	TR31	TR41
#doc	1504	1657	2340	1560	927	878
#word	2886	3758	4592	8460	4703	7454
#class	13	25	6	20	7	10
MinClass	11	13	60	5	2	9
MaxClass	608	371	1389	341	352	243
min/max	0.018	0.035	0.043	0.015	0.006	0.037
source	Reuters-21578		WebACE		TREC-6,7	

Table 2: Comparison on six datasets.

data	method	$ERR$	$F$	$NMI$	$CE$
RE0	COPAP	0.4109	0.3812	0.3288	2.385
	STD	0.4262	0.3341	0.2711	2.608
RE1	COPAP	0.4906	0.3983	0.3610	2.436
	STD	0.5214	0.3434	0.3434	2.800
K1	COPAP	0.1282	0.8734	0.6987	0.5676
	STD	0.1444	0.8097	0.6512	0.9000
WAP	COPAP	0.5147	0.4173	0.4615	0.9269
	STD	0.5551	0.3336	0.3677	1.125
TR31	COPAP	0.2808	0.5407	0.4411	1.600
	STD	0.3112	0.5177	0.3978	1.731
TR41	COPAP	0.2976	0.5129	0.4420	1.161
	STD	0.2654	0.6421	0.5657	1.499

and cluster evaluation criteria, then we evaluate the clustering performance of COPAP against the standard bipartite formulation. Finally we illustrate clustered search results.

### 5.1 Experimental Setup

In our experiments, we used six datasets from different sources, as shown in Table 1. For all data sets, we used a stoplist to remove common words, stemmed the remaining words using Porter’s suffix-stripping algorithm and removed those words with extreme low document frequencies.

Because the true class labels of documents are known, we can measure the quality of the clustering solutions using external criteria that measure the discrepancy between the structure defined by a clustering and what is defined by the true class labels. We use the following four measures: normalized mutual information( $NMI$ ), conditional entropy( $CE$ ), error rate( $ERR$ ) and F-measure [5].  $NMI$  and  $CE$  are entropy based measures. Error rate  $ERR(T|C)$  computes the fraction of misclassified data when all data in each cluster is classified as the majority class in that cluster. F-measure combines the precision and recall concepts from information retrieval.

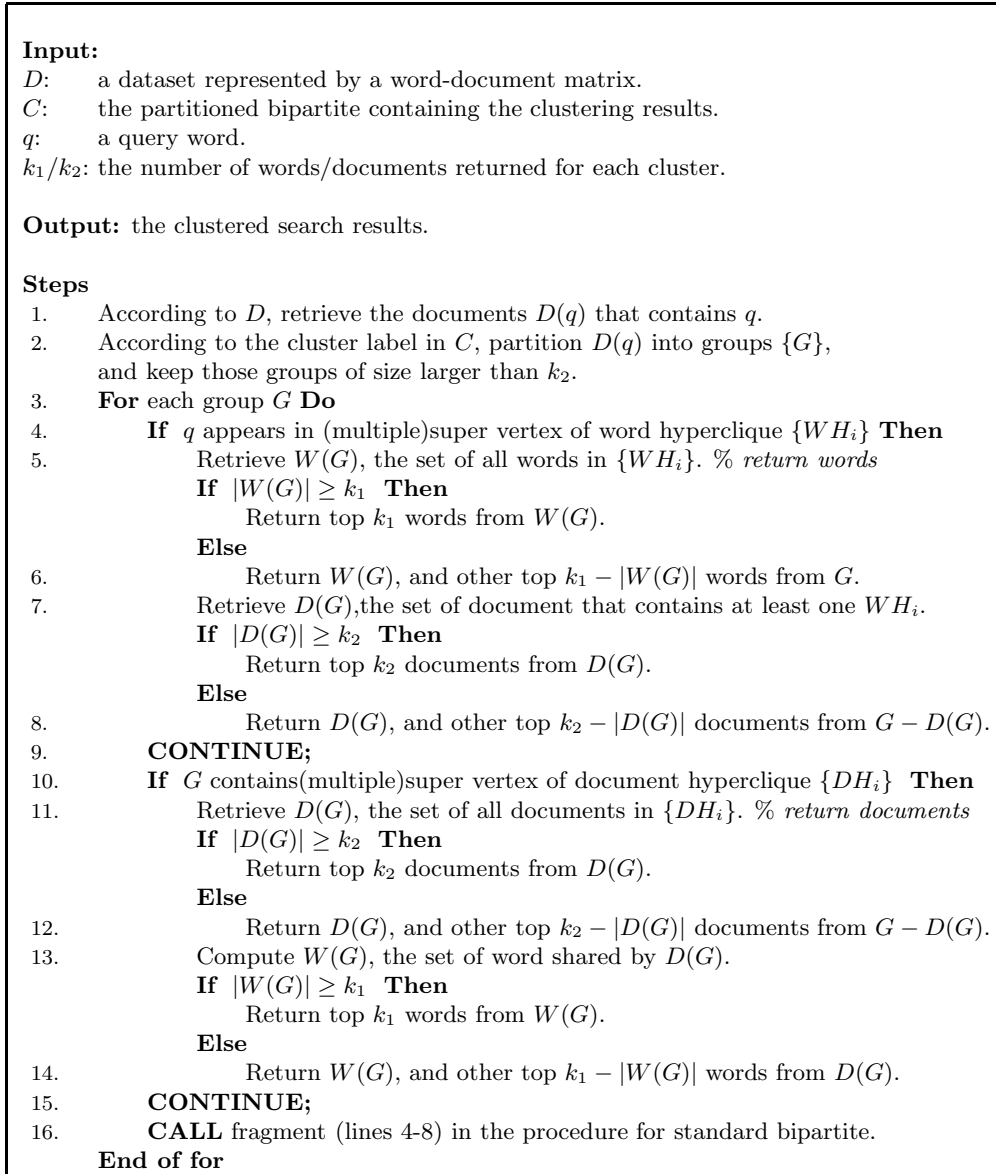


Figure 4: The co-preserving bipartite algorithm for returning clustered search results.

## 5.2 Clustering Results

Because our main purpose is to show the advantage of using hyperclique patterns as starting points, we just compare COPAP with the standard bipartite formulation on transactional data. By setting the number of clusters equal to the true number of classes, the clustering results are shown in Table 2, where STD denotes the standard bipartite formulation. *NMI* and *F* are preferred large while *ERR* and *CE* are preferred small. One can see that except for TR41, COPAP is able to achieve improvement on all datasets in terms of all four measures. The two parameters, support threshold and h-confidence threshold, were tuned separately for each dataset, but not for each criterion.

## 5.3 Applications to Search Engines: Clustered Search Results

In this subsection, we illustrate the application of the partitioned bipartite to showing clustered search results. The motivation is still the high affiliation within hypercliques. Due to lack of space, we only show two search results of the co-preserving bipartite in Table 3 for dataset K1. For each cluster, we show the number of documents in that cluster, top five words, and the sentence where the query word appears in the top document.

As shown in Figure 3, the standard bipartite formulation can do this job by first grouping all the documents containing the query according to the cluster

label, and then returning the top words and documents from each group of documents. In some cases, however, we find that its returned words are still too general, not closely enough related to the query. As for the bipartite with co-preserved patterns, this problem is relieved considerably. For instance, given query ‘cell’, the standard bipartite returned ‘risk, medic, diseas, find, drug’ from the cluster of health. Obviously they are related to health and medicine, but not closely related to cell. The reason is that for the current group of documents containing word ‘cell’, these words are still top, possessing the largest within-cluster degrees. In contrast, the bipartite with co-preserved patterns output ‘normal, gene, brain, professor, cancer’, because each word forms a two-word hyperclique with ‘cell’, according to the steps (lines 5-6) in Figure 4 which dictate the words from the hypercliques receive priority of being selected. Words like ‘medicine’ and ‘disease’ are too general to be able to form a hyperclique with ‘cell’, because  $conf(\text{cell} \rightarrow \text{medic})$  is high, but not vice versa.

Similar observations were also made when there are no word patterns and we select top words from top/hyperclique documents. Given query ‘model’, the standard bipartite only returned general words like ‘risk’ and ‘disease’ from the cluster of health. In contrast, the bipartite with co-preserved patterns output ‘protect, respons, risk, diseas, medicin’, because the first two words come from a document hyperclique talking about road safety for drivers. Therefore, according to steps (lines 13-14) in Figure 4, they are selected first.

## 6 Conclusions

In this paper, we presented a new approach, CO-preserving PATterns in bipartite PARTitioning (COPAP), for word-document co-clustering and cluster topic identification. Hyperclique patterns capture strong connections between groups of objects and should not be separated during clustering. Using them as starting points in the bipartite, our experiments showed that better clustering results could be obtained in terms of various external criteria and the cluster topic can be identified accurately. Besides, the co-preserved patterns in the partitioned bipartite enable those words and documents across several topics to appear in more than one cluster as needed. Due to the unique structure of the partitioned bipartite, it naturally lends itself to clustering related functions in search engines. Finally we illustrated such an application, returning clustered search results for keyword queries. Experiments indicated that compared to the standard bipartite formulation, selecting topical words from word/document patterns is able to identify the topic that is more closely related to the current query.

Table 3: Sample search results for K1.

cell
(6) finance stock analyst percent internet Online: AT&T launches pocketnet internet cell phone (127) normal gene brain professor cancer Health: The loss of key brain cells may be reversible
model
(10) protect respons risk diseas medicin Health: safety-oriented roadway design model in Europe (7) celebr fashion gala crash paris People: Five-hundred models appear in fashion show

## References

- [1] I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 269–274, 2001.
- [2] I. S. Dhillon, Y. Guan, and B. Kulis. A fast kernel-based multilevel algorithm for graph clustering. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 629–634, 2005.
- [3] T. H. Haveliwala. Topic-sensitive PageRank: A context-sensitive ranking algorithm for web search. *IEEE Transactions on Knowledge and Data Engineering*, 15(4):784–796, 2003.
- [4] Y. Huang, H. Xiong, W. Wu, and Z. Zhang. A hybrid approach for mining maximal hyperclique patterns. In *Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence*, pages 354–361, 2004.
- [5] B. Larsen and C. Aone. Fast and effective text mining using linear-time document clustering. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 16 – 22, 1999.
- [6] M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. In *KDD Workshop on Text Mining*, 2000.
- [7] H. Xiong, M. Steinbach, P.-N. Tan, and V. Kumar. HICAP: Hierarchical clustering with pattern preservation. In *Proceedings of the 4th SIAM International Conference on Data Mining*, pages 279–290, 2004.
- [8] H. Xiong, P.-N. Tan, and V. Kumar. Mining strong affinity association patterns in data sets with skewed support distribution. In *Proceedings of the 3rd IEEE International Conference on Data Mining*, pages 387–394, 2003.
- [9] H. Xiong, P.-N. Tan, and V. Kumar. Hyperclique pattern discovery. *Data Mining and Knowledge Discovery Journal*, 13(2):219–242, 2006.