2006

Concept Based Document Clustering using a Simplicial Complex, a Hypergraph

Kevin Lind
San Jose State University

Concept Based Document Clustering using a Simplicial Complex, a Hypergraph

A Writing Project

Presented to

The Faculty of the Department of Computer Science

San José State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Kevin Lind

December 2006

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

Dr. Tsau Young Lin

Dr. Chris Pollett

Dr. Teng Moh

APPROVED FOR THE UNIVERSITY

Abstract

CONCEPT BASED DOCUMENT CLUSTERING USING A

SIMPLICIAL COMPLEX, A HYPERGRAPH

by

Kevin Lind

This thesis evaluates the effectiveness of using a combinatorial topology structure (a simplicial complex) for document clustering. It is believed that a simplicial complex better identifies the latent concept space defined by a collection of documents than the use of hypergraphs or human categorization. The complex is constructed using groups of co-occurring words (term associations) identified using traditional data mining methods. Disjoint subsections of the complex (connect components) represent general concepts within the documents' concept space. Documents clustered to these connect components will produce meaningful groupings. Instead, the most specific concepts (maximal simplices) are used as representative connect components to demonstrate this technique's effectiveness. Each document in a cluster is compared against its human assigned category to determine the cluster's precision. It is shown that this technique is better able to cluster documents than human classifiers.

Table of Contents

1. Introduction

A word alone can seldom identify the concept of a document. A group of words begins to describe a document, however that description may be clouded if the relationship between the words is unknown. In a collection of documents, the associations between the words represent the concepts the documents represent.

Association discovery is a cornerstone in the field of data mining. Association rules data mining was originally designed for market basket analysis [1] but is recently being used to discover term associations in document collections. Term associations, in turn, provide a natural structure for document clustering [8], and hence, document retrieval. It is believed that term associations of documents represent a latent semantic structure which can improve document retrieval results where polysemic words are involved. Polysemy is the issue where a word may have multiple meanings [5]. For example, “wall” and “street” are common building and city structures; however “Wall Street” carries a very different connotation.

Document clustering is the process of grouping documents together in a meaningful way. This paper considers document clustering by using a geometric structure formed by terms and their associations. The term associations of the document corpus are used to form a simplicial complex which is a topological concept space where the terms of the document corpus are vertices and their associations define unique concept spaces. Reducing the

simplicial complex produces disjoint sections, which form term clusters. Documents are then matched to these term clusters. A similar technique is the use of hypergraphs to represent the term associations [13]. A simplicial complex is a stronger notion, however, because it correlates directly to term associations and their *a priori* property.

The next section of the paper introduces the technologies of term association data mining, term weighting by term frequency and inverse document frequency and term stemming. Follows is a description of the techniques used to prepare the data for this study and the process used to generate the term associations of the document corpus. The paper ends with a description of hypergraphs and simplicial complexes, and experimental results and conclusions.

2. Background / Related Work

2.1. Mining Association Rules

The process of mining association rules was first introduced by Rakesh Agrawal in [1] and later refined in [2]. Association rule data mining builds rules which identify items in a database which frequency occur together. Association rule data mining on text documents is the process of discovering sets of terms which occur together in a document corpus.

A term association is a set of terms which appear together more frequently than if the terms were completely independent. An association rule is an implication of the form A→B where A and B are term associations in the document corpus and the intersection of A and B is the empty set.

Term association mining begins with a set of terms *T* and a document corpus *D* where all the terms in *T* appear in at least one document in *D*. Each document $d_i \in D$ is a set of terms $\{t_1,t_2,\ldots,t_i\} \in T$. At this point, it's easy to view the document collection as a database table where each document is a transaction and the terms are the items of the transaction. A term association is a set of terms $\{t_1,t_2,\ldots,t_k\} \in T$ and is called an *itemset*. A *k*-itemset is an itemset with *k* terms. The terms in the documents and itemsets are kept in sorted order to improve the efficiency of the data mining algorithms.

2.1.1. Frequent Set Generation

Term association mining involves making multiple passes over the database of documents. Each pass starts with a *candidate set* of *k*-itemsets. Each itemset in the candidate set is compared for inclusion in each document in the database. The *support* of an itemset is the percentage of documents which contain the itemset. Support is the frequency of the itemset in the database. If an itemset meets a specified minimum support threshold, the itemset is considered *frequent* and is added to the *frequent set* of *k*-itemsets. Itemsets which are not frequent are discarded. At the end of the k^{th} pass, the

frequent set contains all the frequent *k*-itemsets (itemsets which meet the minimum support). These itemsets are used to generate the candidate set of (*k+1*)-itemsets for pass *k+1* over the database. Passes over the database are continued until no frequent itemsets are discovered. In the rest of this paper, the term 'term association' specifically refers to an itemset of terms which is frequent.

Definition (Support) *Let T be the set of all terms* $\{t_1,t_2,\ldots,t_i\}$ *in the document set D. An itemset I is a set of terms such that* $I \subseteq T$. *The* ***support*** *of I is the percentage of documents* $d \in D$ *where* $I \subseteq d$.[1]

2.1.2. Candidate Set Generation

At the end of each pass the candidate set is generated based on the method used in the Apriori algorithm [2]. This method produces candidate itemsets of size *k+1* from the frequent itemsets of size *k* discovered in the previous pass. First, every pair of itemsets in the frequent set which match by the first *k-1* items is joined to produce an itemset of size *k+1*. For example, the 3-itemsets {a b c} and {a b d} are joined to produce the 4-itemset {a b c d}. Next, the candidate set is pruned by removing all *k*-subset itemsets which are not frequent. It is known that if a *k*-itemset is frequent, then all (*k-1*)-subsets must also be frequent. Therefore, if a candidate itemset contains a subset which is not frequent, then the itemset will also not be frequent and is removed from the candidate set. The pruning step takes advantage of what is known as the *Apriori Property* of frequent associations.

If an itemset is found to be frequent then, by a priori knowledge, it is known that every subset of items within the frequent itemset must also be frequent.

Definition (a priori) *Let I be an itemset with k items. If I is a frequent itemset, then every (k-1)-subset of I must also be frequent.*[3]

As an example, consider the frequent itemsets found from the 3rd pass F_3 = {{a b c}, {a b d}, {a c e}, {a c d}, {b c d}}. After joining, the candidate set produced is C_4 = {{a b c d}, {a c d e}}. Itemset {a c d e} is pruned because its subset {c d e} is not in F_3. Therefore, C_4 only contains the itemset {a b c d}.

2.1.3. Rule Generation

Association rule generation is the final step in association rule data mining, though it may be performed immediately after a frequent set is generated. As previously stated, an association rule is an implication of the form A→B where A and B are itemsets and the intersection of A and B is the empty set. Every frequent *k*-itemset will produce $\sum_{i=1}^{k-1}\binom{k}{i}$ rules. The 3-itemset {a b c} generates the 6 rules ab→c, ac→b, bc→a, a→bc, b→ac, and c→ab. The *confidence* of a rule is the ratio of the itemset support to the rule's antecedent support and indicates the strength of the rule. If a rule meets a given minimum confidence, then the rule is significant.

Definition *Let I be a frequent itemset and X,Y ⊆ I and X ∩ Y = ∅. The **confidence** of the rule X→Y is the ratio support(I) / support(X).* [1]

For association rule discovery of a document corpus, only the undirected term associations are important. Therefore, the association rules are not generated. However, calculating the confidence of a term association is used in some techniques to apply a weight to the association. Specifically, the average confidence of an association is used when building a hypergraph of the term associations.

2.2. Term weighting

Term weighting is a way to determine the importance of a term within a document. The most basic weighting system is a binary representation where a 1 is applied to a term which appears in the document and 0 is applied to a term which does not. It has been shown that a greater degree of granularity produces better results [6]. One of the most popular retrieval models is the TFIDF vector model. TFIDF assigns a term weight based on the term's frequency within the document and its inverse document frequency across the entire document corpus. The frequency of a term t_i in document d_j (the *tf* factor) is a measure of how well that term identifies the contents of the document. The inverse document frequency of term t_i (the *idf* factor) is a measure of how well that term can identify a relevant document. The theory behind inverse document frequency is if a term appears in a large number of documents, then it is less unique to a subset of the document

corpus and is less likely to retrieve relevant documents. Therefore, a term is considered more important if it appears frequently in a small number of documents.

Definition *Let D be the number of documents in a collection and d_i be the document in which the term t_i appears. Let $N(t_i,d_j)$ be the frequency of term t_i in document d_j. Then the weight of term t_i in document d_j is:*

$$tfidf(t_i, d_j) = \frac{N(t_i, d_j)}{\max(N(t_l, d_j))} \log \frac{D}{d_j}$$

Equation 1: tfidf [4]

where the term frequency is normalized by the most frequent term in document d_j

$$tf(t_i, d_j) = \frac{N(t_i, d_j)}{\max(N(t_l, d_j))}$$

Equation 2: term frequency [4]

Normalization of the term frequency is necessary to even out the differences between large documents and small documents. If the term frequency is not normalized, larger documents would carry more weight simply because they contain more words.

2.3. Stemming

Typically it is desirable to consider variations of the same word as a single concept. Stemming is a technique where a word is reduced to its root concept to eliminate variations in word choice. The removal of a word's affixes (its prefixes and suffixes) produces the word's *stem*. For example, the word variants *contained*, *containing*, *container*, and *contains* are reduced to the stem *contain*.

Various stemming strategies exist, but one of the simplest and most popular is affix removal [4]. Affix removal algorithms generally focus on suffix removal since most word variants are formed by adding suffixes. Suffix removal uses a series of rules to progressively reduce a word to its stem. For instance, the rule *s* $\rightarrow \varnothing$ strips the final *s* of a word. Application of multiple rules like *s* $\rightarrow \varnothing$ and *ing* $\rightarrow \varnothing$ reduces the word *endings* to its stem *end*.

3. Data Preparation

3.1. Dataset

The Reuters-21578, Distribution 1.0, [3] dataset is used to evaluate the proposed clustering process. Reuters-21578 is a text corpus of 21,578 English news articles from 1987. The choice for this dataset comes from the fact that the documents have already

been categorized by humans which provides a benchmark to test the accuracy of the document clustering results.

Each of the Reuters-21578 documents is categorized in five category sets where each set contains a number of categories. Each document may be categorized in as many categories as the researchers felt were appropriate. Table 1 shows the distribution of categories to category sets.

Category Set	Number of Categories	Number of Categories w/ 1+ Occurrences	Number of Categories w/ 20+ Occurrences
Exchanges	39	32	7
Organizations	56	32	9
People	267	114	15
Places	175	147	60
Topics	135	120	57

Table 1 Reuters-21578 Categories [3]

To prepare the dataset for use, the Reuters-21578 dataset was first reduced by only selecting the documents contained in the Modified Lewis Split training set. Reducing the Reuters-21578 dataset by the Modified Lewis Split reduces the dimensionality of the data fed into the data mining algorithms, thereby reducing computation time, while still preserving data distribution. The resulting 13,625 documents were filtered by removing documents which were not categorized by humans (topics=’bypass’) and which had unusual structuring (type=’unproc’). The remaining 13,542 documents were stripped of

their SGML tag structure and categorization data, leaving only the documents' title, author, date, and body.

3.2. Data Cleansing

First, the documents are parsed using a simple lexical parser which produces single word terms. Words which contain hyphens, commas or slashes are split into two single word terms. All punctuation is removed from each term which has the effect of normalizing the representation of abbreviations and numbers. A dollar sign ($) which appears before a number, however, is preserved. Each term is also converted to lower case.

Next, the documents are filtered to remove terms which are poor discriminators of the documents' topics. Terms which appear too frequently in a document collection are useless in the clustering process and are removed. Filtering has the bonus effect of reducing the size of the dataset which reduces the computation time of the data mining algorithms. The terms are filtered in one of three ways to produce three distinct datasets.

The first dataset, "Lewis-stem", filters the terms using a stopword list and stemming techniques. A stopword is a commonly occurring word which is of little or no use in discovering the concepts of a document. Stopwords are typically articles, prepositions and conjunctions [4]. The stopword list used on the Lewis-stem dataset contains 571 words and was originally generated by Chris Buckley and Gerard Salton. Stemming is

performed by using the Porter algorithm [9][10] which is a popular suffix stripping algorithm because it is simple yet provides very good results [4].

The second and third datasets, “Lewis-tfidf (0.5)” and “Lewis-tfidf (1.0)”, filters the terms by weighing each term and removing the terms which do not meet a minimum threshold. A tfidf weight is calculated for each term in each document using Equation 1. If a term’s weight is below a given minimum weight, the term is removed from the document. Two minimum weights (1.0 and 0.5) are used to produce two datasets. As Table 2 shows, a minimum term weight of 0.5 reduces the dataset by 35% while a minimum term weight of 1.0 reduces the dataset by 58%.

Dataset	Number of Documents	Number of Terms	Term Reduction	Average Document Length Reduction
Lewis-stem	13542	33443	24.27%	31.74%
Lewis-tfidf (0.5)	13542	44158	0%	35.27%
Lewis-tfidf (1.0)	13542	44158	0%	58.47%

Table 2 Dataset term reduction

In Table 2, the average document length reduction shows the actual reduction of the dataset after applying the various filters. Stemming and stopword removal reduced the set of terms by 24.27% which in turn reduced the dataset by an average of 31.74%. Term weighting is applied directly to the documents, which is why the set of terms are not reduced. The main goal of filtering the dataset is to remove frequently occurring terms which add no meaning to a document.

4. Term Association Discovery

The datasets are processed by a data mining algorithm to discover all the term associations. The association rule data mining process is the same as previously discussed, except the criteria used to determine support of an itemset is further constrained by the distance of the terms within the itemset.

4.1. Support by Distance

Instead of simply checking if an itemset appears in a document, a distance measure is used on the terms in the itemset. If all the terms in an itemset appear within a specified maximum distance of one another, then the itemset's support is increased. The theory is a set of terms which appear close together within a document are more related and therefore can better represent a concept within the document.

Distance is defined as a linear count of words. Consider a document as a list of terms where a term's position is a numeric number from 1…n, where n is the total number of words in the document. Two terms, a and b, are within a distance of x if $| a - b | < x$.

To calculate the distance between terms, an inverted index is created for each document. An inverted index is a word-to-occurrence mapping which speeds up the process of locating a term within a document. In an inverted index, each unique term in the

document is used as an index key which references a list of positions for that term in the document. Using an inverted index, the locations of a term in a document are found in unit time.

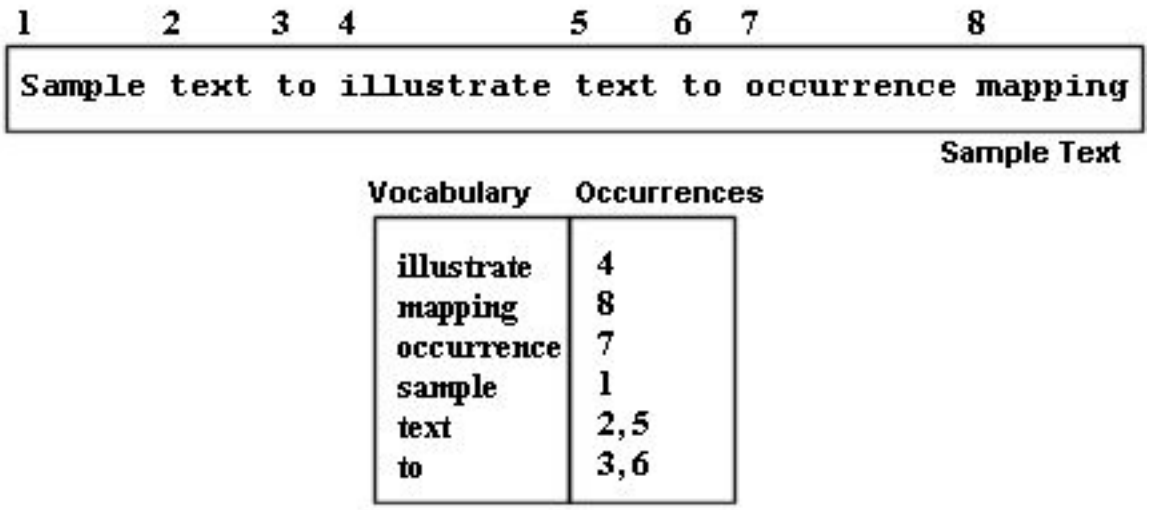

Figure 1 Inverse Index

When calculating the support using distance of an itemset, only the inverted indexes are needed. In determining the support of 1-itemsets, a distance measure is meaningless since there is only one term. Therefore, support is calculated in the usual way as the frequency of the term in the document corpus. In each subsequent check of *k*-itemsets, the positions of the k terms are checked against a maximum distance. If all of the *k* terms are within the maximum distance of each other, then the itemset is considered a subset of the document. If all the *k* terms appear in the document but are not within the maximum distance, the itemset is not considered as a subset of the document. To determine if an itemset is within the maximum distance, find the positions of all the terms of the itemset

then subtract the position of the lowest positioned term from the position of the highest positioned term. If the result is less than the maximum distance, the itemset is considered a subset of the document. This is further complicated by the fact that a term may appear in a document multiple times. Therefore, every combination of each term's positions must be checked. If the itemset meets the maximum distance at least once, then the itemset is a subset of the document. In the end, an itemset is added to the frequent set if the count of documents which are a superset of the itemset is above a minimum support threshold.

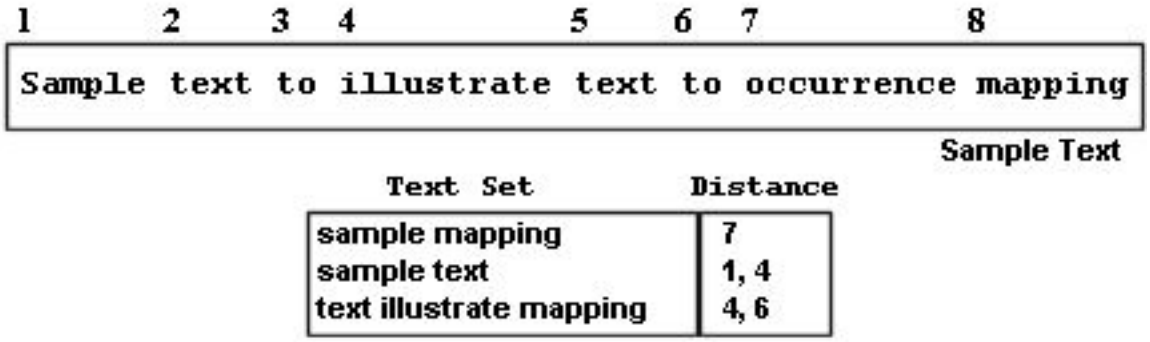

Figure 2 Itemset distance measurement

5. Document Clustering

5.1. Hypergraph Clustering

5.1.1. Hypergraph Defined

A hypergraph [11] is a graph where its edges, known as hyperedges, may connect more than two vertices. Formally, a hypergraph $G = (V, E)$ is a set of vertices $V = \{v_1, v_2, \ldots v_n\}$ and a set of hyperedges $E = \{e_1, e_2, \ldots, e_m\}$ where:

1) $e_i \neq \varnothing$ for $1 \leq i \leq m$

2) $$\bigcup_{i=1}^{m} e_i = V \qquad [11]$$

The order of a hypergraph is defined as $| V | = n$. Similarly, the order, or size, of a hyperedge is the number of vertices it connects, $| e_i |$. The *rank of hypergraph G* is defined as the hyperedge with the highest order. More formally, the rank of hypergraph G is defined as:

$$r(V) = \max_{i}^{m}(V \cap e_i)$$

Term associations discovered from association rule data mining form a hypergraph where the set of terms are the vertices and the term associations are the hyperedges. Weights are applied to each hyperedge for use in the clustering process. The weight of the hyperedge is the average confidence of the term association which forms the vertices of the hyperedge [13]. The confidence of a term association is the averaged confidences of

all possible rules of the term association. As previously stated, an association rule is an implication of the form A → B where its confidence is the support of A and B divided by the support of A. For example, if a hyperedge has five vertices, its weight would be the averaged confidences of the twenty association rules.

Below is a figure depicting a hypergraph with vertices $V = \{v_1, v_2, v_3, v_4\}$ and edges $E = \{E_1, E_2, E_3\}$ where $E_1 = \{v_1, v_2\}$, $E_2 = \{v_1, v_3\}$ and $E_3 = \{v_2, v_3, v_4\}$. Hypergraph variant 1 on the left uses bubbles to depict the hyperedges which is a common method to draw hypergraphs. Hypergraph variant 2 on the right uses a more traditional line graph to display the same hypergraph. Note that hyperedge E_3 is not three separate line segments but one line connected to three vertices.

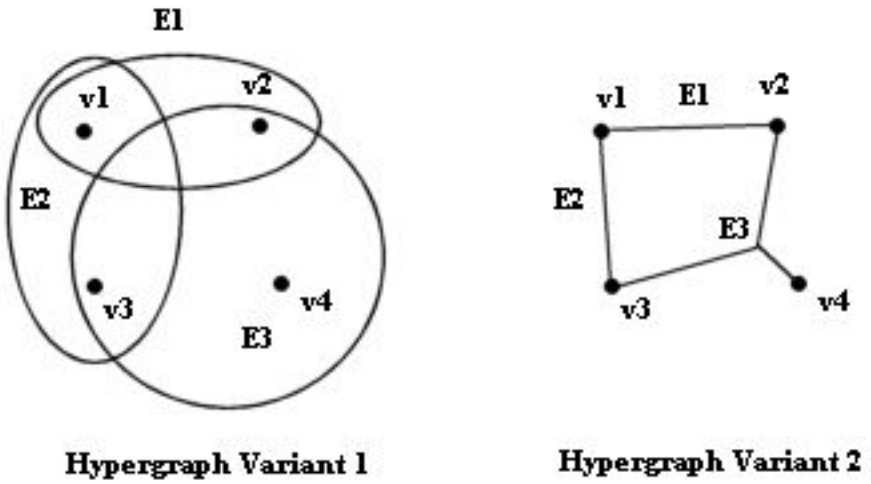

Figure 3 Two graphical representations of a Hypergraph

5.1.2. Document Clustering using Hypergraphs

In order to cluster the document corpus, the hypergraph is first partitioned into clusters of terms (vertices) then documents are matched to a cluster using a simple score. HMETIS is a popular hypergraph partitioning tool which focuses on minimizing the sum of the weights of cut hyperedges [13]. HMETIS uses a multilevel partitioning algorithm which recursively projects the hypergraph to a hypergraph of lower order, partitions the lower order hypergraph, and then recursively projects the hypergraph back to its original order while refining the partitions at each step [12]. Hyperedges which straddle multiple partitions are cut, where hyperedges with smaller weights are stronger candidates for cutting. HMETIS attempts to produce *k* partitions of equal number of vertices where *k* is user defined.

Once the terms are clustered, the documents are matched to a cluster with the maximum number of matching terms. The document-cluster score is the ratio $\frac{\left|D \bigcap C_i\right|}{\left|C_i\right|}$ where D is the document and C_i is the cluster of terms [13].

5.2. Simplicial Complex Clustering

5.2.1. Simplex Defined

In Euclidean space, the smallest convex hull which contains n+1 vertices $\{v_0, \ldots, v_n\}$ is called an *n-simplex* [14]. Also, the vertices of the n-simplex may not lie in a hyperplane

of less than n dimensions. For example, three vertices which form a 2-simplex may not align linearly on a 1-dimentional plane; they must form a 2-dimentional triangle. A *face* is a sub-simplex of the *n*-simplex with vertex set $\{v_0, \ldots, v_n\}$, whose vertices are any non-empty subset of $\{v_0, \ldots, v_n\}$ [14]. The faces of a simplex form the simplex's boundaries. Simply put, a simplex is bounded by the lower dimensional simplices which are completely contained within the higher dimensional simplex. The number of *i*-faces in an *n*-simplex is equal to the binomial coefficient, $\binom{n+1}{i+1}$ $(0 \leq i \leq n)$.

An *open simplex* is a simplex whose face boundaries are exclusive of the simplex. Conceptually, if a simplex contains a specific concept of a document space, then its faces contain more general concepts since they are lower dimensional simplices. An open simplex is bounded by, but does not contain, its faces just as a specific concept is bounded by its general concepts. In the rest of this report, open simplices are used exclusively and the term 'simplex' is assumed as an open simplex.

As examples, a 1-simplex is a line bounded by, but not including, its end-points (0-faces), a 2-simplex is the convex hull of a triangle bounded by, but not including, its edges (1-faces) and vertices (0-faces), and a 3-simplex is the convex hull of a tetrahedron bounded by, but not including, the triangles (2-faces), edges (1-faces) and vertices (0-faces) which border the tetrahedron. The intersection of two simplices is the common face between the two simplices.

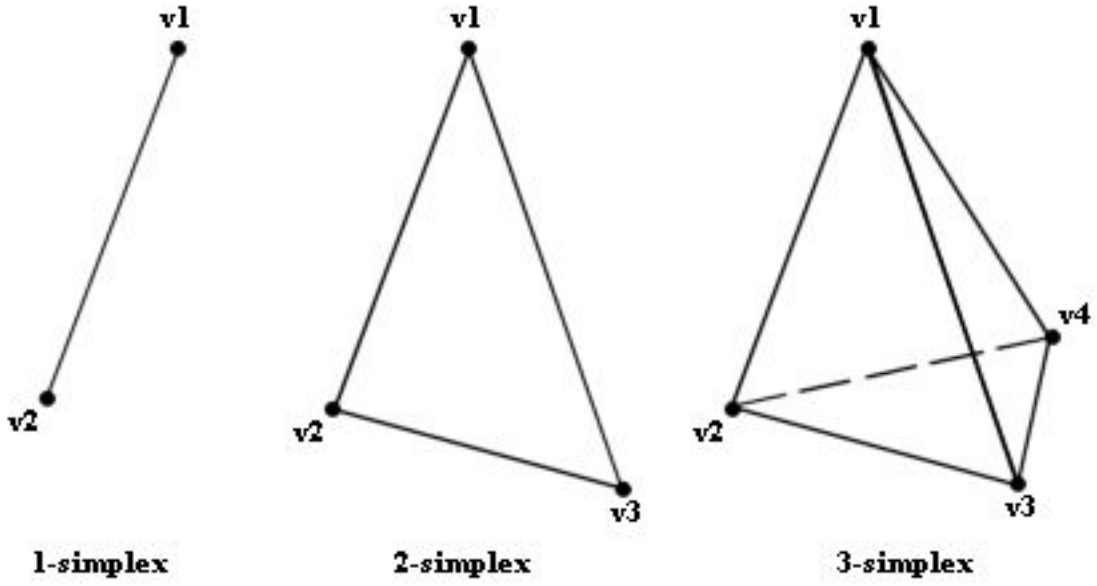

Figure 4 Three Simplices

A *simplicial complex* is a set of simplices where every simplex is uniquely identified by its vertices [14]. Every face of a simplex of simplicial complex *C* is also in *C*. Combinatorially, a simplicial complex with a set of vertices C_0 and a set of *n*-simplices C_n:

1) C_n are (n+1)-element subsets of the vertex set C_0
2) Each (k+1)-element subset of the vertices in C_n is a *k*-simplex in C_k ($k < n$)

[14]

Notice that the definition for a simplicial complex is the same as the Apriori property for term associations. Remember that the Apriori property states that if an *n*-association is frequent, then all its subset *i*-associations ($1 \leq i \leq n$) must also be frequent [1]. In the

same way, every $\binom{n+1}{i+1}$ *i*-face ($0 \leq i \leq n$) of an *n*-simplex in simplicial complex *C* must also be in *C*.

The relationship between simplices and term associations is exactly one-to-one. A (n+1) term association is viewed as an n-simplex where the terms of the association are vertices. Consider the term association {'wall', 'street'}. Its *1*-simplex represents the semantics of finance which is beyond the notion of its vertices 'wall' and 'street'. Thus, an *n*-simplex of a (n+1) term association carries a stronger semantic meaning than the union of its vertices.

Simplicial complexes appear both in algebraic topology and geometry. This report uses a simplicial complex in the algebraic topology sense where a complex is formed from a combination of simplices. It is more natural and easier, however, to view a simplicial complex structure in the geometric sense, as points in space. When discussing and graphing simplicial complexes, this report uses the algebraic topology meaning and does not assume any correlation to Euclidian space.

Unlike a hypergraph representation of term associations, a simplicial complex elegantly represents the Apriori property of term associations as stated above. Term associations have a stronger relation to simplicial complexes over hypergraphs in another way. Hypergraphs focus on the graph structure of term associations as hyperedges and have no

real structure to represent the semantic meaning of a term association. Simplicial complexes are in the realm of combinatorial topology and represent the space of the term associations. For a term association the physical space of its semantic meaning is its open simplex (a simplex with its faces removed). In this way, it is more natural to view a set of term associations as a simplicial complex than as a hypergraph.

5.2.2. Document Clustering using Simplicial Complex

The *connect components* of a simplicial complex are used to cluster a set of documents. Two simplices *a* and *b* are adjacent, or *directly connected*, if they share a common nonempty face. Two simplices *a* and *b* are *connected* if there is a finite chain of adjacent simplices between *a* and *b*. In connect component *P*, each simplex in *P* is connected to every simplex in *P* and only connected to simplices in *P*. More formally, simplicial complex *C* is the set of n-simplices *X* and has *n* connect components P_n where $P_i \subseteq X$ and $\bigcup_i^n P_i = X$ and $\bigcap_i^n P_i = \varnothing$.

Determining the connect components is an important step in finding the concepts of a set of documents. The semantics represented by the simplices in a connect component are all related in some way, and are thus considered as a concept. Within a connect component are various granularities of this singular concept. As an example, consider a complex which represents the concept of education. A sample of the 0-simplices contains the terms {college}, {high-school}, {math}, {literature}. Each of these

simplices represents a low level concept under the umbrella of ‘education’. The 1-simplices are {college, math}, {high-school, math}, and {high-school, literature} which have higher concepts, or deeper semantic meaning, than the 0-simplices. Each higher level of simplex represents finer-tuned concepts all under the general concept of the connect component.

The set of term associations discovered from the data mining process may produce large groups of connected simplices, and thusly a small number of connect components. It may be beneficial to increase the number of connect components of a simplicial complex creating more clusters with more specific concepts. A *skeleton* of a simplicial complex removes the lower order simplices, essentially disconnecting groups of simplices from each other. A *skeleton* S_r^n of simplicial complex C is a simplicial complex where all k-simplices in C are removed where $k \leq r < n$ [8]. Typically, n is equal to the dimension of the largest simplex so that the skeleton only has the lower dimensional simplices removed. Note that a skeleton of a complex removes lower dimensional simplices and not the actual vertices. Adjusting the r parameter produces simplicial complex skeletons with various number of connect components. Going back to the previous example, taking the skeleton S_0^1 would remove all 0-simplices, splitting the concept of ‘education’ into the three clusters ‘college math’, ‘high-school math’ and ‘high-school literature.’

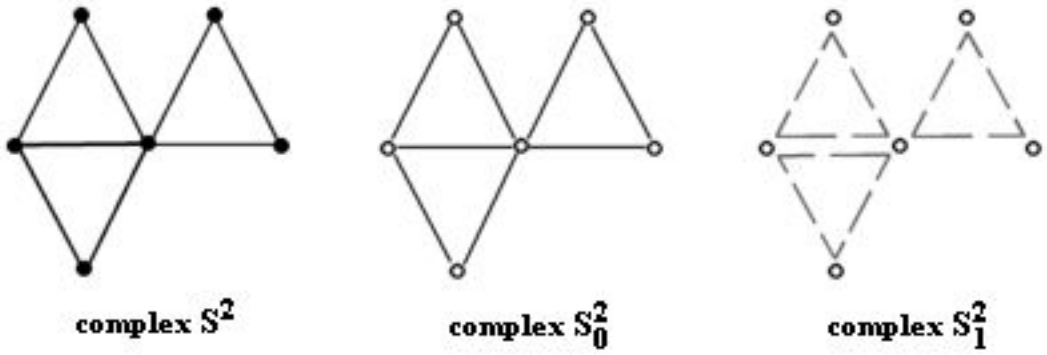

Figure 5 Complex with progressive skeleton reductions

The figure above shows various skeletons of a 2-complex (complex where its maximal simplex has 3 vertices). The first complex S^2 has one connect component, the entire complex. The second complex S_0^2 removes all 0-simplicies and has two connect components (the two inverted triangles on the left and the one triangle on the right). The last complex S_1^2 removes all 0-simplicies and 1-simplicies and has three connect components. Note that the concept space contained by the three 2-simplicies (represented by the dashed triangles in the figure) still remains.

Once the connect components of a simplicial complex are discovered, the document corpus is clustered. Each document in the corpus is compared with each connect component to determine its cluster. Essentially, a document belongs to a connect component if its terms are vertices of the connect component. However, it is possible that all the terms of a connect component may not appear in any document or a document may contain terms which appear in multiple connect components. Therefore, documents

are clustered to a connect component if they contain a simplex of the connect component. A connect component contains simplices of various dimensions (number of vertices), so documents are strongly clustered where they contain a simplex of high dimension. A single document may contain multiple concepts and may naturally appear in multiple clusters.

6. Experimental Results

6.1. Data Mining Results

Using the association rules algorithm with maximum distance as a support criteria, the three datasets from Table 2 were run using a distance setting of 10. The maximal distance of 10 is used to represent a standard sentence from the Reuters document corpus.

The minimum support for each run was set to 0.1%. Since each dataset contained the same 13,542 documents, an association of terms is considered significant if it appears in at least 14 documents using a minimum support of 0.1%. A low minimum support is required to produce clusters of fine granularity.

Table 3 summarizes the data mining results which were run on a 3GHz Intel Xeon based machine with 1 GB main memory using Windows XP operating system.

Dataset	TFIDF	Distance	Total term associations	Longest term association	Run time (hours:min)
Lewis-stem	-	10	96,045	8	1:59
Lewis-split	0.5	10	71,368	10	1:16
Lewis-split	1.0	10	18,925	8	0:32

Table 3 Data Mining Results

6.2. Hypergraph Clusters

The program HMETIS (version 1.5) [13] is used to compute clusters based on a term association hypergraph. The first step is to create a weighted hypergraph using the term associations discovered during the data mining process. The hypergraph is composed of only the longest term associations (itemsets with the most terms). Using only the longest associations produces an appropriate sample size and gives results which are more comparable to the simplicial complex clustering results in the next section. Each hyperedge represents a single term association with its weight being the averaged confidence of the term association. The terms of the hypergraph are partitioned using HMETIS. The hypergraph partitioning vertex grouping scheme is used while partitioning which attempts to group terms by their hyperedges where hyperedges of larger weights are given preference. Also, during each recursive bisection step, a cut hypergraph is retained and copied to each partition.

Once the terms are partitioned, documents are clustered by matching the terms in the partitions to the document's terms. Documents are added to the cluster which has the highest number of matching terms. Documents which match less than half of the terms in its cluster are considered outliners and are not clustered.

Each document in the Reuters-21578 dataset was categorized by human interpreters. The categories applied to each cluster below, therefore, is the sum of the categories applied to each document in said cluster. Each cluster category is given a number which is the fraction of documents in the cluster which are assigned to the category. This ratio is the *precision* of the cluster. For readability, only the categories with high precisions are displayed in the result tables.

6.2.1. Lewis-stem Data Results

Consider the Lewis-stem dataset using a distance measure of 10 (dataset filtered using stopwords and stemming where all terms in each itemset are within 10 words of each other). The longest term association is 8 terms with 17 term associations of this length. HMETIS is set to produce four clusters from these 17 term associations.

Cluster	Terms
1	10, 15, 30, 31, april, ct, march, pai, prior, record, reuter
2	commiss, convert, cover, debentur, dlr, due, exchang, issu, mln, registr, statement, subordin
3	2012, agreement, arrang, billion, custom, govern, market, offer, repurchas, secur
4	1000, loss, net, oper, profit, rev, shr, year

Table 4 Lewis-stem dataset hypergraph clusters

Cluster	# Documents	Reuters Category	Precision
1	2634	uk usa earn	8.16% 69.17% 48.01%
2	878	uk usa acq	12.87% 66.74% 14.81%
3	285	japan usa uk west-germany acq interset money-fx trade	13.33% 53.33% 12.98% 7.37% 10.88% 18.59% 21.35% 7.37%
4	938	canada usa earn	11.94% 77.19% 95.20%

Table 5 Lewis-stem cluster categories (hypergraph)

6.3. Simplicial Complex Clusters

Clustering begins by viewing the term associations discovered in the data mining step as a simplicial complex. Each term association is a simplex of the complex uniquely identified by the association's terms, which are the vertices of the simplex. The space contained by each simplex is a unique concept in the document corpus. Joined simplices contain related concepts. Each connect component of the complex represents a general concept in the document corpus which is related to all the concepts contained in the simplices which construct the connect component.

As proof of concept, consider only the most specific concepts of the document corpus which are the maximal simplices where a *maximal simplex* is a simplex which is not a

subset of any other simplex in the complex [8]. These simplices are identified as containing the most vertices. The maximal simplices represent the most specific concepts in the document corpus and by the *a priori* property, all faces of the maximal simplices represent more general concepts. Considering only the maximal simplices and their faces will produce a smaller set of well defined clusters which are sufficient to demonstrate the results.

Skeletons of the simplicial complex are taken to produce the largest number of connect components (clusters) with the least amount of overlap of documents between clusters. Documents are matched to each cluster using the method described previously. Since each cluster is based on a simplicial complex formed from maximal simplices and not all simplices, the total number of documents matching all clusters is less than the total number of documents in the corpus. However, the sample size is sufficient to display the results of this technique.

As previously stated, each document in the Reuters-21578 dataset was categorized by human interpreters. The categories applied to each cluster below, therefore, is the sum of the categories applied to each document in said cluster. Each cluster category is given a number which is the fraction of documents in the cluster which are assigned to the category. This ratio is the *precision* of the cluster. Higher precisions mean all the documents in the cluster are talking about the same concept.

6.3.1. Lewis-stem Dataset Results

Consider the Lewis-stem dataset using a distance measure of 10 (dataset filtered using stopwords and stemming where all terms in each itemset are within 10 words of each other). The longest term association is 8 terms with 17 term associations of this length. The simplicial complex is reduced to only consider these 17 7-simplices and their containing faces. This 7-complex has one connect component. A S_0^7 skeleton removes all the 0-simplices, revealing two connect components. A S_1^7 skeleton removes all the 0-simplices and 1-simplices, revealing four connect components and the smallest number of documents which overlap multiple clusters. Each of the four connect components represent distinct concepts, but since the connect components share low dimensional simplices, the concepts are related in a general way.

The figure below shows the four clusters and the simplices they contain. Since a S_1^7 skeleton produced the clusters, a maximum overlap of two terms is allowed between the clusters. Note that stemming is used on this dataset so the terms are stems and not necessarily complete words.

cluster	10	15	20	30	31	april	ct	march	pai	prior	record	reuter	2012	commiss	convert	cover
1		1			1	1	1	1	1		1	1				
		1				1	1	1	1	1	1					
		1			1		1	1	1	1	1	1				
		1		1		1	1		1	1	1	1				
	1					1	1	1	1	1	1	1				
			1			1	1	1	1	1	1	1				
					1	1	1	1	1	1	1	1				
2														1		1
													1		1	
															1	1
															1	1
															1	1
													1		1	
3																
4							1									
							1									

cluster	2012	commiss	convert	cover	debentur	dlr	due	exchang	issu	mln	registr	statement	subordin	offer
2		1		1		1		1	1	1	1	1		
	1		1		1	1	1			1			1	1
			1	1	1	1	1		1	1			1	
			1	1	1	1			1	1		1	1	
			1	1		1			1	1	1	1	1	
	1		1		1	1	1		1	1			1	
3						1								
						1								
4										1				
						1				1				

cluster	agreement	arrang	billion	custom	govern	market	repurchas	secur	1000	loss	net	oper	profit	rev	shr	year
3	1	1	1	1		1	1	1								
		1	1	1	1	1	1	1								
4									1	1	1	1	1	1		
										1	1		1		1	1

Table 6 Lewis-stem itemsets grouped by clusters (simplicial complex)

Cluster	# Documents	Reuters Category	Precision
1	169	canada usa earn	1.18% 98.82% 98.82%
2	65	usa acq	100% 3.08%
3	20	usa interset money-fx	90% 85% 65%
4	197	canada usa earn	8.12% 90.86% 98.98%

Table 7 Lewis-stem dataset cluster categories (simplicial complex)

Reading a sample of the documents from each cluster gives deeper meaning to the numbers in table 5. Cluster 1's documents are company dividend earnings for the first quarter of the year reported in US markets. Cluster 2's documents discuss company statements of issued debentures. A couple of the documents state that the funds will be used for company acquisitions (hence the 'acq' category). Cluster 3's documents are concerned with Federal Reserve customer repurchase agreements. Cluster 4's documents are company dividend earnings for the fourth quarter plus yearly profit and loss reports. What's interesting is cluster 1 and 4 have the same categories (identified by humans) but the simplicial complex structure was able to identify a distinct difference between the two sets of documents. Specifically, cluster 1 is about first quarter earnings while cluster 4 is about yearly earnings.

6.3.2. Lewis-split (TFIDF 1.0) Dataset Results

Consider the Lewis-split dataset using a distance measure of 10 and a TFIDF 1.0 (dataset filtered using TFIDF value of 1.0 where all terms in each itemset are within 10 words of each other). The longest term association is 8 terms and 13 term associations are of this length. The simplicial complex of this dataset is reduced to only these 13 maximal 7-simplices and their containing faces. This simplicial complex has one connect component. In order to cut the complex and increase the number of connect components, a S_5^7 skeleton which removes all 5-simplices and lower is required. This means two itemsets are in different clusters if they match by 6 or less terms. Three connect components (clusters) result from the S_5^7 skeleton, however there is much overlap of the document between clusters. In theory, each connect component is a unique, yet related, concept within the document space. However, by reading a sample of the clustered documents, the specific concepts are unclear. The general concept shared by all three clusters is ‘stock dividend payouts.’

cluster	12	15	17	10	20	30	25	may	qtly	div	record	april	pay	cts	vs	prior
1		1						1			1	1	1	1	1	1
		1				1					1	1	1	1	1	1
2		1						1		1	1		1	1	1	1
						1				1	1	1	1	1	1	1
		1								1	1	1	1	1	1	1
				1						1	1	1	1	1	1	1
3				1					1	1		1	1	1	1	1
							1		1	1		1	1	1	1	1
					1				1	1		1	1	1	1	1
	1								1	1		1	1	1	1	1
		1							1	1		1	1	1	1	1
						1			1	1		1	1	1	1	1
			1						1	1		1	1	1	1	1

Table 8 Lewis-split (tfidf 1.0) itemsets grouped by clusters (simplicial complex)

Cluster	# Documents	Reuters Categories	Precision
1	32	usa earn	100% 100%
2	110	canada usa earn	1.81% 98.18% 98.18%
3	138	canada usa earn	1.45% 98.55% 98.55%

Table 9 Lewis-split (tfidf 1.0) dataset cluster categories (simplicial complex)

One explanation for the data is the large number of numeric terms in the itemsets. The numeric terms in the itemsets are the main difference between the itemsets. These numbers relate to monetary amounts and dates. However, the numbers possibly lost too much of their meaning when taken with the 10 closest words in the document (the distance support criteria for determining the term associations). For example, one itemset contains 15, april, may, and cts (shorthand for cents). It is unclear how '15' relates to the

other terms. Considering words with their ten closest neighbors is probably sufficient, however, numbers should be considered with words only two or three words distance.

6.3.3. Lewis-split (TFIDF 0.5) Dataset Results

Consider the Lewis-split dataset using a distance measure of 10 and a TFIDF 0.5 (dataset filtered using TFIDF value of 0.5 where all terms in each itemset are within 10 words of each other). The longest term association for this dataset is 10 terms. However, only 2 term associations were found with this length. These two associations share a common 8-face (9 out of 10 terms in common) which means the concepts they represent are closely related. Since there are few maximal simplices in this 9-complex, consider the 25 maximal simplices in the 8-complex subset. The 8-complex has two connect components which do not share any common face and therefore represent distinct concepts. Taking a S_6^8 skeleton cuts the larger connect component in to three, producing four connect components total. A S_6^8 skeleton separates term associations into different clusters if they match by seven or fewer terms. This produces clusters with closely related concepts.

clusters	1	5	is	selling	pct	while	management	underwriting	combined	concession	pays	and	luxembourg	fees	listed	comprise
1	1			1	1					1		1	1	1	1	1
	1	1		1	1		1			1		1		1		1
	1	1		1	1		1	1	1	1		1				
2	1		1	1	1	1	1	1	1	1						
	1		1	1	1	1	1	1	1		1					
	1		1	1	1	1	1	1		1	1					
	1		1	1	1	1	1		1	1	1					
	1		1	1	1	1		1	1	1	1					
	1		1	1	1		1	1	1	1	1					
	1		1	1		1	1	1	1	1	1					
			1	1	1	1	1	1	1	1	1					
	1			1	1	1	1	1	1	1	1					
	1	1	1		1	1	1	1	1	1						
	1	1	1		1	1	1	1	1		1					
	1	1	1		1	1	1	1		1	1					
	1		1		1	1	1	1	1	1	1					
	1	1	1		1	1	1		1	1	1					
	1	1	1		1	1		1	1	1	1					
	1	1	1		1		1	1	1	1	1					
	1	1	1			1	1	1	1	1	1					
		1	1		1	1	1	1	1	1	1					
	1	1			1	1	1	1	1	1	1					
3		1	1		1		1	1	1		1					
		1			1	1	1	1	1		1					
4																

clusters	date	payment	than	basis	more	securities	non	points	treasury	comparable	yield
3	1	1									
	1	1									
4			1	1	1	1	1	1	1	1	1

Table 10 Lewis-stem (tfidf 0.5) itemsets grouped by clusters (simplicial complex)

Cluster	# Documents	Reuters Category	Precision
1	50	australia canada denmark finland france luxembourg sweden uk west-germany	2% 6% 2% 2% 2% 4% 4% 98% 2%
2	40	autralia canada france luxembourg uk usa	12.5% 5% 2.5% 2.5% 90% 2.5%
3	31	autralia france luxembourg uk usa	9.68% 3.23% 3.23% 90.32% 3.23%
4	20	autralia usa	5% 100%

Table 11 Lewis-split (tfidf 0.5) dataset cluster categories (simplicial complex)

When the documents are matched to the clusters, there is some overlap mainly across clusters 2 and 3. In fact, every document in cluster 3 is also in cluster 2. That is not unexpected since in the full simplicial complex (no skeleton taken) the simplices of cluster 1, 2, 3 are all connected. All the documents in the first three clusters deal with the listing of Eurobond issuances in various foreign cities. The documents in the fourth cluster are reports of companies selling notes to raise funds. Most documents in the fourth cluster have New York datelines and one involved a New Zealand company listed on the New York Stock Exchange.

As for the differences between the first three clusters, looking at the terms (words) in the clusters gives some clues. The words which are different between cluster 2 and 3 are {1, selling, concession} in cluster 2 and {payment, date} in cluster 3. Both clusters are about the listing and issuing of Eurobonds, however cluster 2 is more specifically about the selling concession rate and cluster 3 is more specifically about the payment date. Since all the documents in cluster 3 are also in cluster 2, all the documents in cluster 3 contain the concepts of 'selling concession' and 'payment date'.

7. Conclusion

Document clustering is an important technique in information retrieval for finding sets of relevant documents to queries. Clustering documents by associated terms has been shown as an effective clustering metric [7][8]. However, using hypergraphs to view the term associations (as in [7]) is not as effective or natural as using a simplicial complex. The *a priori* property of term associations states that every subset of a frequent term association is also a frequent term association [2]. This property directly correlates to a complex such that every sub-simplex of a simplicial complex is also a simplex in the simplicial complex. Therefore it is very natural to view the set of term associations of a document corpus as simplices of a simplicial complex.

The results show clustering using term associations as a simplicial complex produces a high level of precision as compared to the categories defined by humans in the Reuters-21578 dataset. Also, in certain cases the simplicial complex clusters better distinguished the concepts of the documents than the human defined categories.

The amount and type of data pre-processing has a large affect on the results. Using a stopword list and word stems produced better results than filtering words based on their term frequency inverse document frequencies. This is possibly due to the use of word stems which combine similar words together, increasing the frequency of a stem in a document, thereby increasing its chance of appearing in a term association. Also, a stopword list is a more targeted means of removing 'meaningless' words from a document. Stopword lists, however, are human generated and care must be taken when selecting stopwords.

Overall, using a simplicial complex constructed from a documents corpus' term associations is an effective method for document clustering. The simplicial complex keeps the term associations' *a priori* structure intact as well as the concepts the associations represent. More or less clusters are automatically produced by reducing the complex using skeletons of varying degrees. Also, the results of this technique are superior to that of topic categorization by humans.

8. References

[1] Agrawal R, Imielinski T, Swami A. Mining association rules between sets of items in large databases. In: Proc. of the ACM SIGMOD Conference on Management of Data; May 1993; Washington, D.C. p 207-216.

[2] Agrawal R, Srikant R. Fast algorithms for mining association rules. In: Proceedings of the 20th VLDB Conference; 1994.

[3] Lewis DD. Reuters-21578 text categorization test collection [Internet]. [cited 2005 May 5]. Available from: http://www.daviddlewis.com/resources/testcollections/reuters21578/.

[4] Baeza-Yates R, Ribeiro-Neto B. Modern Information Retrieval. New York: ACM Press; 1999. 513 p.

[5] Deerwester S, Dumais ST, Furnas GW, Landauer TK, Harshman R. Indexing by latent semantic analysis. Journal of the American Society of Information Science 1990; 41(6): 391-407.

[6] Salton G, Buckley C. Term-weighting approaches in automatic text retrieval. Information Processing and Management 1988; 24(5): 513-523.

[7] Clifton C, Cooley R, Rennie J. TopCat: Data mining for topic identification in a text corpus. In: Principles of Data Mining and Knowledge Discovery; 1999; p 174–183.

[8] Lin TY, Chiang I. A Simplicial Complex, a Hypergraph, Structure in the Latent Semantic Space of Document Clustering.

[9] Porter MF. An algorithm for suffix stripping. Program 1980; 14(3): 130-137.

[10] Porter MF. The Porter Stemming Algorithm [Internet]. [cited 2006 March 23]. Available from: http://tartarus.org/~martin/PorterStemmer/.

[11] Berge C. Graphs and Hypergraphs. New York: American Elsevier; 1976. p 389-396.

[12] Karypis G. Multilevel Hypergraph Partitioning. Computer Science and Engineering Dept., Univ. Minnesota, Minneapolis: 2002; Tech. Rep. 02-25.

[13] Han EH, Karypis G, Kumar V, Mobasher B. Clustering Based on Association Rule Hypergraphs.

[14] Hatcher A. Algebraic Topology. Cambridge: Cambridge University Press; 2002. p 102-107.

[15] Weisstein EW. Simplicial Complex – From MathWorld [Internet]. [cited 2006 May 8]. Available from: http://mathworld.wolfram.com/SimplicialComplex.html

[16] Munkres JR. Simplicial Complexes and Simplicial Maps. Elements of Algebraic Topology. Perseus Press; 1993. p 7-14.