Semantic Based Clustering of Web Documents

Tsau Young ('T. Y.') Lin[1]* and I-Jen Chiang[2]

[1] Department of Computer Science
San Jose State University
One Washington Square
San Jose, CA 95192-0249
tylin@cs.sjsu.edu
[2] Graduate Institute of Medical Informatics
Taipei Medical University
205 Wu-Hsien Street
Taipei, Taiwan 110
ijchiang@tmu.edu.tw

Abstract. A new methodology that structures the semantics of a collection of documents into the geometry of a simplicial complex is developed. A simplicial complex is topologically equivalent to a polyhedron in Euclidean space. The semantics of documents are structured by the geometry: A primitive concept is represented by a *simplex.* and a concept is represented by a *connected component.* Based on these structures, documents can be clustered into some meaningful classes. Experiments with three different data sets from web pages and medical literature have shown that our approach performs significantly better than traditional clustering algorithms, such as *k-means*, *AutoClass* and *Hierarchical Clustering* (HAC).

keyword clustering, association(rule)s, topology, simplicial complex, polyhedron

1 Introduction

Clustering a given collection of documents has been a contemporary challenge. There are many proposals; see Section 2. Almost all the known techniques are based on some metrics in Euclidean geometry; these metrics are rather ad hoc, and in general, do not reflect the semantic of the documents.

In this paper, we are proposing a novel approach, in which the semantics of a collection of documents can be structured into a simplicial complex, and hence, a polyhedron in Euclidean space without even using any metrics. It is well known in functional analysis that a given finite dimensional Euclidean space has many metrics, but only one unique topology. Briefly, though traditional approaches are based on the various metric spaces, they have the same underlying topological

* T. Y. Lin

spaces as our approach [42]. Our approach has utilize this unique topology in a semantically meaningful way.

This paper is a continuation of previous work [29], in which we have considered a simpler problem, namely, clustering the documents by the maximal PIMITIVE COMCEPTs. Here we explore the full notion. The results are very encouraging; so the proposal seems promising.

1.1 Capture Semantics via Simplexes

We will illustrate the idea by examples [29]. Let "wall" and "street" be two keywords that appear in a set of documents. These two keywords together often denote some financial notions that have nothing to do with the two keywords individually. A parallel situation also occurs in geometry. Let us regard the two keywords as two abstract vertices, say v_0 and v_1. In combinatorial topology, such two vertices determine an open segment that consists of linear points between the two vertices exclusively. This open segment, called a 1-simplex, represents one dimensional geometric object that does not include the two points, v_0 and v_1.

By generalize such analogy to high dimensional spaces, we have the following correspondence:

1. A set of vertices determine a new object, called a simplex; see Section **??**
2. An association of keywords (e.g., "Wall Street") corresponds to a new notion that represents the semantics (financial concept) of two keywords "Wall" and "Street."

We would like to assert a delightful observation that the arpiori condition is exactly the same as that of simplicial complex. This observation implies that the keyword-association (frequent itemset) is an abstract simplicial complex in combinatorial topology.

– So the semantics of simplicial complex of keywords impose a geometric structure into abstract space, called Latent Semantic Space (LSS), of human thoughts that are hidden in the given documents.

1.2 An Overview of the Idea

What is a document? It is an ordered list of character strings that characterizes a human thought. The list will be referred to as a **linear text**, and the character string a **term** or a **token**. For a computer system, however, a document is merely a linear text; it has no idea as to what a human thought is.

As we have pointed out that to handle such a large set of linear texts has been a contemporary challenge, and many models have been proposed. However, most of the solutions are based on some ad hoc Euclidean metrics that do not reflect the semantics of the data. For example, the well known Latent Semantic Index has no obvious connections with the semantics of the documents. The index actually is related more to the "distributions" of keywords.

So in this paper, we will introduce, based on the semantics of keywords, the Euclidean topology to the semantics space without using any metrics. Here is few important points of the idea

1. Each document in the given collection is associated automatically to a tuple of keywords via the notion of TFIDF.
2. The totality of these keywords will be called the universal attributes in a very precise sense of relational theory.
3. Based on HIGH TFIDF values a set of keyword-associations(roughly frequent itemsets of length q) can be selected. The totality of such selected keyword-associations forms an abstract simplicial complex, which is topologically equivalent to a triangulation (linear simplicial complex) of a polyhedron in Euclidean space.
4. The polyhedron is topologically equivalent to the human thoughts that are constraints by those keyword-associations in the documents. We will call these human thoughts the Latent Semantic Space (LSS) of the collection. So we have introduced the Euclidean topology into the semantic space of the documents without using a metric.

Based on the topology of LSS, we propose the following conceptual structure:

1. A PRIMITIVE CONCEPT is represented by a *simplex*;
2. A maximal PRIMITIVE CONCEPT is represented by a *maximal dimension simplex*; in [29], we called this one primitive concept.
3. A CONCEPT is represented by a *connected component*.
4. An IDEA is the whole polyhedron.
5. Based on these structures, documents can be clustered into some meaningful classes:
   (a) Clustering the documents by maximal A PRIMITIVE CONCEPTs.
   (b) Clustering the documents by CONCEPTs.
   (c) Clustering the documents hierarchically(an forest) by the set of CONCEPTs. 0-simplex is in the highest level, and 1-simplex is a sub-concept, and so forth. For example, mathematics is a sub-concept of science if the PRIMITIVE CONCEPT (represented by a simplex) of science is a face of the PRIMITIVE CONCEPT of mathematics (represented by another simplex).

In this paper, we are interested in those short documents, so the collection of the returned web pagers by a Google search engine is the best example fro our applicatons.

In what follows, we start by reviewing some related work on document clustering in section 2. Section 3 introduces the mathematics of simplicial complex. Section 4 constructs conceptually the abstract simplicial complex of keywords And the corresponding topological space, called *Latent Semantic Space*. Section 5, we how the semantics are captured in the polyhedron. Section 6 shows some experimental results from different data sets, followed by the conclusion.

2 Related Work

Document classification/clustering has been considered as one of the most crucial techniques for dealing with the diverse and large amount of information present on the World Wide Web. In particular, clustering is used to discover latent concepts in a collection of Web documents, which is inherently useful in organizing, summarizing, disambiguating, and searching through large document collections [25].

Numerous document clustering methods have been proposed based on probabilistic models, distance and similarity measures, or other techniques, such as SOM. A document is often represented as a feature vector, which can be viewed as a point in the multi-dimensional space. Many methods, including *k-means*, support vector machines, hierarchical clustering and nearest-neighbor clustering, etc., select a set of key terms or phrases to organize the feature vectors corresponding to different documents. Suffix-tree clustering [46], a phrase-based approach, formed document clusters depending on the similarity between documents.

Hierarchical clustering algorithms have been proposed in an early paper by Willett [45]. Cutting et al. introduced partition-based clustering algorithms for document clustering [11]. Buckshot and fractionation were developed in [27]. Greedy heuristic methods are used in the hierarchical frequent term-based clustering algorithm [4] to perform hierarchical document clustering by using frequent itemsets. We should note here that frequent itemsets are also referred to as associations(undirected association rules).

3 Background - Combinatorial Topology

This section is purely for reference purposes. Let us introduce and define some basic notions in combinatorial topology. The central notion is n-simplex.

Definition 1. *A n-simplex is a set of independent abstract vertices $[v_0, \ldots, v_{n+1}]$. A r-face of a n-simplex $[v_0, \ldots, v_{n+1}]$ is a r-simplex $[v_{j_0},\ldots,v_{j_{r+1}}]$ whose vertices are a subset of $\{ v_0, \ldots, v_{n+1} \}$ with cardinality $r + 1$.*

Geometrically 0-simplex is a vertex; 1-simplex is an open segment (v_0, v_1) that does not include its end points; 2-simplex is an open triangle (v_0, v_1, v_2) that does not include its edges and vertices; 3-simplex is an open tetrahedron (v_0, v_1, v_2, v_3) that does not includes all the boundaries. Formally,

Definition 2. *A simplicial complex C is a finite set of simplexes that satisfies the following two conditions:*

- *Any set consisting of one vertex is a simplex.*
- *Any face of a simplex from a complex is also in this complex.*

The vertices of the complex $v_0, v_1, \cdots, v_n$ is the union of all vertices of those simplexes ([42], pp. 108).

If the maximal dimension of the constituting simplexes is n then the complex is called n-complex.

Note that, any set of $n+1$ objects can be viewed as a set of abstract vertices, to stress this abstractness, some times we refer to such a simplex a combinatorial n-simplex. The corresponding notion of combinatorial n-complex can be defined by (combinatorial) r-simplexes.

A (n, r)-skeleton (denoted by S^n_r) of n-complex is a n-complex, in which all k-simplexes($k \leq r$) have been removed. Two simplexes in a complex are said to be *directly connected* if the intersection of them is a nonempty face. Two simplexes in a complex are said to be *connected* if there is a finite sequence of directly connected simplexes connecting them. For any non-empty two simplexes A, B are said to be r-*connected* if there exits a sequence of k-simplexes $A = S_0, S_1, \ldots, S_m = B$ such that S_j and S_{j+1} has an h-common face for $j = 0, 1, 2, \ldots, m-1$; where $r \leq h \leq k \leq n$.

The maximal r-connected subcomplex is called a r-*connected component.* Note that a r-connected component implies there does not exist any r-connected component that is the superset of it. A maximal r-connected sub-complexes of n-complex is called r-connected component. A maximal r-connected component of n-complex is called connected component, if $r = 0$.

Example 1 *In Figure 1, we have a simplicial complex that consist of twelve vertices that are organized in the forms of* 3*-complex, denoted by* S^3.

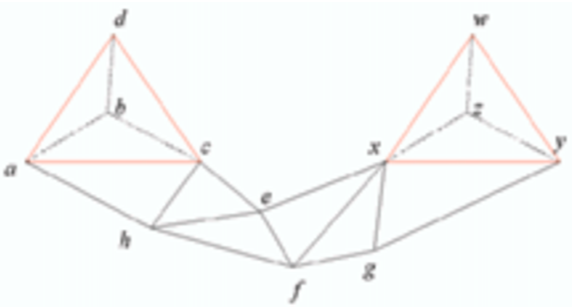

Fig. 1. A complex with twelve vertexes.

Let us enumerate every simplex of S^3*:*

1. *The maximal* 3*-simplex* $S(a, b, c, d)$*, and*
   (a) *Its four* 2*-simplex faces* $S(a, b, c)$*,* $S(a, b, d)$*,* $S(a, c, d)$*,* $S(b, c, d)$*, and*
   (b) *Its six* 1*-simplex faces* $S(a, b)$*,* $S(a, c)$*,* $S(a, d)$*,* $S(b, c)$*,* $S(b, d)$*,* $S(c, d)$*,*
2. *The maximal* 3*-simplex* $S(w, x, y, z)$
   (a) *Its four* 2*-simplex faces* $S(w, x, y)$*,* $S(w, x, z)$*,* $S(w, y, z)$*, and* $S(x, y, z)$*, and*
   (b) *Its six* 1*-simplex faces* $S(w, x)$*,* $S(w, y)$*,* $S(w, z)$*,* $S(x, y)$*,* $S(x, z)$*,* $S(y, z)$

3. The maximal 2-simplexes lying ”between’ two 3-simplexes: $S(a,c,h)$, $S(c,h,e)$, $S(e,h,f)$, $S(e,f,x)$, $S(f,g,x)$, $S(g,x,y)$ and
4. Some of their 1-simplex faces $S(a,h)$, $S(c,h)$, $S(c,e)$, $S(h,e)$, $S(e,f)$, $S(h,f)$, $S(e,x)$ $S(f,x)$, $S(f,g)$, $S(g,x)$, $S(g,y)$; non of them are maximal.
5. Their 0-simplex faces (certices) a, b, c, d, e, f, g, h, w, x, y, z

Let us consider the $(3,1)$-skeleton S^3_1, which consists of all 3-simplexex, 2-simplexes and 1-simplexes of S^3.

1. The maximal 3-simplex $S(a,b,c,d)$, and
   (a) Its four 2-simplex faces $S(a,b,c)$, $S(a,b,d)$, $S(a,c,d)$, $S(b,c,d)$, and
   (b) Its six 1-simplex faces $S(a,b)$, $S(a,c)$, $S(a,d)$, $S(b,c)$, $S(b,d)$, $S(c,d)$,
2. The maximal 3-simplex $S(w,x,y,z)$
   (a) Its four 2-simplex faces $S(w,x,y)$, $S(w,x,z)$, $S(w,y,z)$, and $S(x,y,z)$, and
   (b) Its six 1-simplex faces $S(w,x)$, $S(w,y)$, $S(w,z)$, $S(x,y)$, $S(x,z)$, $S(y,z)$
3. The maximal 2-simplexes lying ”between’ two 3-simplexes: $S(a,c,h)$, $S(c,h,e)$, $S(e,h,f)$, $S(e,f,x)$, $S(f,g,x)$, $S(g,x,y)$ and
4. Some of their 1-simplex faces $S(a,h)$, $S(c,h)$, $S(c,e)$, $S(h,e)$, $S(e,f)$, $S(h,f)$, $S(e,x)$ $S(f,x)$, $S(f,g)$, $S(g,x)$, $S(g,y)$; non of them are maximal.
5. No 0-simplex

Let us consider the $(3,2)$-skeleton S^3_2, which consists of all 3-simplexex and 2-simplexes of S^3.

1. The maximal 3-simplex $S(a,b,c,d)$, and
   (a) Its four 2-simplex faces $S(a,b,c)$, $S(a,b,d)$, $S(a,c,d)$, $S(b,c,d)$, and
   (b) No 1-simplex
2. The maximal 3-simplex $S(w,x,y,z)$
   (a) Its four 2-simplex faces $S(w,x,y)$, $S(w,x,z)$, $S(w,y,z)$, and $S(x,y,z)$, and
   (b) No 1-simplex faces
3. The maximal 2-simplexes lying ”between’ two 3-simplexes: $S(a,c,h)$, $S(c,h,e)$, $S(e,h,f)$, $S(e,f,x)$, $S(f,g,x)$, $S(g,x,y)$ and
4. No 1-simplex faces.
5. No 0-simplex faces

Let us consider the $(2,1)$-skeleton S^2_1, which consists of all 2-simplexex and 1-simplexes of S^3.

1. No maximal 3-simplex
   (a) Four 2-simplex faces $S(a,b,c)$, $S(a,b,d)$, $S(a,c,d)$, $S(b,c,d)$, and
   (b) Six 1-simplex faces $S(a,b)$, $S(a,c)$, $S(a,d)$, $S(b,c)$, $S(b,d)$, $S(c,d)$,
2. No maximal 3-simplex
   (a) Four 2-simplex faces $S(w,x,y)$, $S(w,x,z)$, $S(w,y,z)$, and $S(x,y,z)$, and
   (b) Six 1-simplex faces $S(w,x)$, $S(w,y)$, $S(w,z)$, $S(x,y)$, $S(x,z)$, $S(y,z)$
3. The maximal 2-simplexes lying ”between’ two 3-simplexes: $S(a,c,h)$, $S(c,h,e)$, $S(e,h,f)$, $S(e,f,x)$, $S(f,g,x)$, $S(g,x,y)$ and
4. Some of their 1-simplex faces $S(a,h)$, $S(c,h)$, $S(c,e)$, $S(h,e)$, $S(e,f)$, $S(h,f)$, $S(e,x)$ $S(f,x)$, $S(f,g)$, $S(g,x)$, $S(g,y)$; non of them are maximal.
5. No 0-simplex faces

Let us consider the $(2,2)$-skeleton S^2_2, which consists of all 2-simplexex of S^3.

1. No maximal 3-simplex

(a) Four 2*-simplex faces* $S(a,b,c)$*,* $S(a,b,d)$*,* $S(a,c,d)$*,* $S(b,c,d)$*, and*
(b) No 1*-simplex*

2. *No maximal* 3*-simplex*
   (a) Four 2*-simplex faces* $S(w,x,y)$*,* $S(w,x,z)$*,* $S(w,y,z)$*, and* $S(x,y,z)$*, and*
   (b) No 1*-simplex*
3. *The maximal* 2*-simplexes lying "between' two* 3*-simplexes:* $S(a,c,h)$*,* $S(c,h,e)$*,* $S(e,h,f)$*,* $S(e,f,x)$*,* $S(f,g,x)$*,* $S(g,x,y)$ *and*
4. *No* 1*-simplex*
5. *No* 0*-simplex faces*

4 The Simplicial Geometry of Keywords

In this section, we will construct the abstract simplicial complex of keywords. For simplicity, we will use keywords to mean keywords, key phrases, and key terms; they include compound nouns. In [**?**], we have explained the idea in [43], how one can use the labeled sets of keywords (given by human experts) to learn the rules for document classifications. There, the keywords are selected by human and uses only in the syntactic levels. For example, given a set of documents (in English) and their Chinese translations, the strategy in there cannot identify the translation without human help. On the other hand, if use the strategy offered in this paper, the respective Chinese translations and their English original will correspond to homeomorphic polyhedra. So we may conclude the two sets have the same semantics(not implement here).

In this paper, we will automate the keywords selection (use TDITF) and explore (use data mining techniques) the deeper semantics hidden in the interactions among keywords.

4.1 Vertices, Keywords and TFIDF

We will use TFIDF [40] value as the weight of keywords in each document. A word will be selected as a keyword if the TFIDF value of a keyword is large. Roughly, TFIDF indexing is tf × idf indexing [40, 39], where tf denotes term frequency that appears in the document and idf denotes inverse document frequency where document frequency is the number of documents which contain the term. Moffat and Zobel [35] pointed out that tf × idf function demonstrates: (1) rare terms are no less important than frequent terms in according to their idf values; (2) multiple appearances of a term in a document are no less important than single appearances in according to their tf values. The tf × idf implies the significance of a term in a document, which can be defined as follows.

We observed that the direction of key terms (including compound words) is irrelevant information for the purpose of document clustering. So we ignore the *confidence* and consider only the *support*. In other words, we consider the structure of the *undirected* associations of key terms; we believe the set of key terms that co-occur reflects the essential information, the rule directions of the key terms are inessential, at least in the present stage of investigation. Let t_A and t_B be two terms. The *support* is defined for a collection of documents as follows.

Definition 3. *Let T_r denote a collection of documents. The significance of a term t_i in a document d_j in T_r is its TFIDF value calculated by the function* $\mathrm{tfidf(t_i, d_j)}$*, which is equivalent to the value* $\mathrm{tf(t_i, d_j)} \times \mathrm{idf(t_i, d_j)}$*. It can be calculated as*

$$\mathrm{tfidf(t_i, d_j) = tf(t_i, d_j) \log \frac{|T_r|}{|T_r(t_i)|}}$$

where $|T_r(t_i)|$ denotes the number of documents in T_r in which t_i occurs at least once, and

$$\mathrm{tf(t_i, d_j)} = \begin{cases} 1 + \log(N(t_i, d_j)) \text{ } \mathit{if} \text{ } N(t_i, d_j) > 0 \\ \\ 0 \qquad\qquad\qquad\qquad\quad \mathit{otherwise} \end{cases}$$

where $N(t_i, d_j)$ denotes the frequency of terms t_i occurs in document d_j by counting all its nonstop words.

To prevent the value of $|T_r(t_i)|$ to be zero, *Laplace Adjustment* is taken to add an observed count.

Definition 4. *The support of a keyword t_A in a collection is:*

$$\mathrm{support(t_A, T_r)} = \frac{1}{\mathrm{|T_r|}} \sum_{\mathrm{i=0}}^{\mathrm{|T_r|}} \mathrm{tfidf(t_A, d_j)}$$

where, $|T_r(t_A)|$ defines number of documents that contain term t_A, and $|T_r|$ denotes the number of documents in the collection.

Traditionally, TFIDF values are often organized into the following matrix form: Let a document d_j in T_r be represented as a vector $V_j = <$ $\mathrm{tfidf(t_1, d_j)}$, $\mathrm{tfidf(t_2, d_j)}$, $\cdots$, $\mathrm{tfidf(t_n, d_j)}$ $>$ and therefore T_r be represented as a matrix $M_r = < V_1, V_2, \cdots, V_I, \cdots >^T$. Most previous works [12, 13, 15] proposed to finding the association rules or partitioning the association rules into clusters [6] from M_r.

However, these index values have no obvious connections with the semantics of documents. So the clustering of the document based on these partitioning or association rules are not very meaningful. In fact, these values are more like distributions of keywords. So, we choose

Definition 5. *Those terms t_i in document d_j as the keywords or the* 0*-simplexes, if the TFIDF* $\mathrm{tfidf(t_i, d_j)}$ *is large.*

4.2 Simplexes, co-Occurrences of Keywords and High TFIDF

In the last subsection, we have focuses on the 0 dimension, here We will explain the higher dimension cases. Let us start with dimension one.

Definition 6. *The significance of keyword-associations in a collection will be defined in terms of 1-dimensional TFIDF of* t_A *and term* t_B*:*

$$\mathrm{significance(t_A, t_B, T_r) = \frac{1}{|T_r|}\sum_{i=0}^{|T_r|} TFIDF_1(t_A, t_B, d_i)}$$

where

$$\mathrm{TFIDF_1(t_A, t_B, d_i) = tf(t_A, t_B, d_i) \log \frac{|T_r|}{|T_r(t_A, t_B)|}}$$

, $|T_r(t_A, t_B)|$ *defines number of documents contained both term* t_A *and term* t_B*, and* $|T_r|$ *denotes the number of documents in a collection.*

The term frequency $\mathrm{tf(t_A, t_B, d_i)}$ of both term t_A and t_B can be calculated as follows.

Definition 7.

$$\mathrm{tf(t_A, t_B, d_j)} = \begin{cases} 1 + \log(\min\{N(t_A, d_j), N(t_B, d_j)\}) \\ \textit{if } N(t_A, d_j) > 0 \textit{ and } N(t_B, d_j) > 0 \\ \\ 0 \\ \textit{otherwise.} \end{cases}$$

A minimal threshold θ is imposed to filter out the terms that their significance values are small. It helps us to eliminate the most common terms in a collection and the nonspecific terms in a document.

Next, we will define the support of co-occurrences keywords (keyword-association) in a document collection. Let t_A and t_B be two terms. The *support* defined in the document collection is as follows.

Definition 8. Support *denotes to be the significance of associations of term* t_A *and term* t_B *in a collection, that is,*

$$\mathrm{Support(t_A, t_B) = significance(t_A, t_B, T_r)}$$

Now, we will explain the q dimension cases.

Definition 9. *The significance of keyword-associations in a collection will be defined in terms of q-dimensional TFIDF of* $t_{A_1}\ \ldots\ t_{A_q}$*:*

$$\mathrm{significance(t_{A_1}, \ldots t_{A_q}, T_r) = \frac{1}{|T_r|}\sum_{i=0}^{|T_r|} TFIDF_q(t_{A_1}, \ldots t_{A_q}, d_i)}$$

where

$$\mathrm{TFIDF_q(t_{A_1}, \ldots t_{A_q}, d_i) = tf(t_{A_1}, \ldots t_{A_q}, d_i) \log \frac{|T_r|}{|T_r(t_{A_1}, \ldots t_{A_q})|}}$$

, $|T_r(t_{A_1}, \ldots t_{A_q})|$ *defines number of documents contained all terms* $t_{A_1}, \ldots t_{A_q}$ *and* $|T_r|$ *denotes the number of documents in a collection.*

The term frequency $\mathrm{tf}(\mathrm{t_{A_1}}, \ldots \mathrm{t_{A_q}}, \mathrm{d_i})$ can be calculated as follows.

Definition 10.

$$\mathrm{tf}(\mathrm{t_{A_1}}, \ldots \mathrm{t_{A_q}}, \mathrm{d_j}) = \begin{cases} 1 + \log(\min\{N(t_{A_1}, d_j), \ldots, N(t_{A_q}, d_j)\}) \\ \textit{if } N(t_{A_1}, d_j) > 0 \ \ldots N(t_{A_q}, d_j) > 0 \\ \\ 0 \\ \textit{otherwise.} \end{cases}$$

Definition 11. Support *denotes the significance of keyword-associations of terms, $t_{A_1}, \ldots t_{A_q}$, in a collection, that is,*

$$\mathrm{Support}(\mathrm{t_{A_1}}, \ldots \mathrm{t_{A_q}}) = \mathrm{significance}(\mathrm{t_{A_1}}, \ldots \mathrm{t_{A_q}}, \mathrm{T_r})$$

4.3 Simplicial Complex of Keywords

It is obvious that the support evaluated by $\mathit{tfidf}_q, q = 1, 2 \ldots$ satisfies the *Apriori* condition. So we have the following

Proposition 1

$$\mathrm{Support}(\mathrm{t_{A_1}}, \ldots \mathrm{t_{A_q}}) \Longrightarrow \mathrm{Support}(\mathrm{t_{A_1}}, \ldots \mathrm{t_{A_j}}, \mathrm{t_{A_{(j-1)}}}, \ldots, \mathrm{t_{A_q}}), \mathrm{j} = 1, \ldots \mathrm{q}$$

This proposition is equivalent to the conditions of simplicial complex; see *Definition* 2. So we have the following

Proposition 2 *The set of all keywords-associations that meet the support conditions forms an abstract simplicial complex of keywords.*

Here is our belief and our hypothesis:

- An IDEA (in the forms of a simplicial complex) may consist of many CONCEPTs (in the form of connected components) that are constructed by PRIMITIVE CONCEPTs (in the form of maximal simplexes).
- A simplex is said to be a maximal if no other simplex in the complex is a superset of it. The geometric dimension represents the degree of preciseness or depth of the semantics that are represented by keyword-associations.

Example 2 *In Figure 1, we have an IDEA that consist of twelve attributes that are organized in the forms of* 3*-complex, denoted by S^3. $S(a, b, c, d)$ and $S(w, x, y, z)$ are two maximal simplexes of the highest dimension* 3*. Let us consider the* $(3, 2)$*-skeleton S^3_2, by removing all* 0*-simplexes and* 1*-simplexes from S^3:*

- CONCEPT$_1$ *composite of $S(a, b, c, d)$ and its four faces (*2*-simplices): $S(a, b, c)$, $S(a, b, d)$, $S(a, c, d)$, and $S(b, c, d)$;*
- CONCEPT$_2$ *composite of $S(a, c, h)$*
- CONCEPT$_3$ *composite of $S(c, h, e)$*
- CONCEPT$_4$ *composite of $S(e, h, f)$*
- CONCEPT$_5$ *composite of $S(e, f, x)$*
- CONCEPT$_6$ *composite of $S(f, g, x)$*

- CONCEPT$_7$ *composite of* $S(g,x,y)$
- CONCEPT$_8$ *composite of* $S(w,x,y,z)$ *and its four faces (2-simplices):* $S(w,x,y)$, $S(w,x,z)$, $S(w,y,z)$, *and* $S(x,y,z)$.

There are no common faces between any two simplexes, so S_2^3 *has eight connected components. For* S_3^3*, it consists of two non-connected* 3*-simplexes that organized two CONCEPTs (*CONCEPT$_1$ *and* CONCEPT$_8$*), which are independent maximal connected components.*

A complex, connected component or simplex of a skeleton represent a more technically refined IDEA, CONCEPT, PRIMITVE CONCEPT.

4.4 Concept Formulation

Based on a divide and conquer method, the algorithm recursively partitions the generated simplicial complex into two parts: one contains a specific simplex and the other does not contain such a simplex.

```
– Algorithm CONCEPT_PARITION($C$, $S$)
    • If $C$ or $S$ is empty, then return.
    • Find out a simplex $H$ connected to $C$ that has the maximum degree in $S$ and
      Support(H ∪ C) is bigger than the given minimal support.[1]
    • Let $K \leftarrow C \cup H$.
    • If $X$ be the set of simplices that each simplex in $X$ has an common face in
      $C \cup H$; $X \cap (C \cup H) \neq \phi$ then call CONCEPT_PARITION($K$, $X$).
    • Let $U = S - X$ and call CONCEPT_PARITION($C$, $X$).
    • The skeleton $S_m^n \leftarrow S_m^n \cup K$ where $m = |H \cup C|$ and $n \geq m$.
```

Fig. 2. The algorithm to find connected components in a simplical complex.

In the algorithm 2, we define the *simplicial difference* between two simplices as follows.

Definition 12. *Let* S_1 *and* S_2 *be two simplices. The* simplicial difference *between two simplices* S_1 *and* S_2 *is a simplex* $S = S_1 - S_2$ *that contains simplex* S_1 *but erases simplex* S_2 *and all its faces.*

Hierarchical clustering performs on grouping the data based on the similar concepts among them. Unlike the conventional hierarchical clustering, the most latent semantics, i.e., those data have a concrete concept, is on the top of the hierarchy not at the bottom. Therefore, a hierarchical partition clustering is naturally from $(n,0)$-skeleton to (n,m)-skeleton ($m \geq 0$ and $m \leq n$). Each simplex in a skeleton represents an individual cluster at each skeleton. According to the connected components within each skeleton, some data are softly clustering into a lot of categories associated to their common faces. A common face identifies a common concept in a context.

A novel algorithm to formulate the hierarchical concepts from a set of high dimensional data is presented in this paper. Based on the generated concepts data can be hierarchically partitioned into distinct but overlapped clusters.

5 The Simplicial Geometry of Keywords

In this section, we will construct the abstract simplicial complex of keywords. For simplicity, we will use keywords to mean keywords, key phrases, and key terms; they include compound nouns. In [**?**], we have explained the idea in [**?**], how one can use the labeled sets of keywords (given by human experts) to learn the rules for document classifications. There, the keywords are selected by human and uses only in the syntactic levels. For example, given a set of documents (in English) and their Chinese translations, the strategy in there cannot identify the translation without human help. On the other hand, if use the strategy offered in this paper, the respective Chinese translations and their English original will correspond to homeomorphic polyhedra. So we may conclude the two sets have the same semantics(not implement here).

In this paper, we will automate the keywords selection (use TDITF) and explore (use data mining techniques) the deeper semantics hidden in the interactions among keywords.

5.1 Vertices, Keywords and TFIDF

We will use TFIDF [40] value as the weight of keywords in each document. A word will be selected as a keyword if the TFIDF value of a keyword is large. Roughly, TFIDF indexing is tf × idf indexing [40, 39], where tf denotes term frequency that appears in the document and idf denotes inverse document frequency where document frequency is the number of documents which contain the term. Moffat and Zobel [35] pointed out that tf × idf function demonstrates: (1) rare terms are no less important than frequent terms in according to their idf values; (2) multiple appearances of a term in a document are no less important than single appearances in according to their tf values. The tf × idf implies the significance of a term in a document, which can be defined as follows.

We observed that the direction of key terms (including compound words) is irrelevant information for the purpose of document clustering. So we ignore the *confidence* and consider only the *support*. In other words, we consider the structure of the *undirected* associations of key terms; we believe the set of key terms that co-occur reflects the essential information, the rule directions of the key terms are inessential, at least in the present stage of investigation. Let t_A and t_B be two terms. The *support* is defined for a collection of documents as follows.

Definition 13. *Let T_r denote a collection of documents. The significance of a term t_i in a document d_j in T_r is its TFIDF value calculated by the function*

$\mathrm{tfidf}(\mathrm{t_i}, \mathrm{d_j})$*, which is equivalent to the value* $\mathrm{tf}(\mathrm{t_i}, \mathrm{d_j}) \times \mathrm{idf}(\mathrm{t_i}, \mathrm{d_j})$*. It can be calculated as*

$$\mathrm{tfidf}(\mathrm{t_i}, \mathrm{d_j}) = \mathrm{tf}(\mathrm{t_i}, \mathrm{d_j}) \log \frac{|\mathrm{T_r}|}{|\mathrm{T_r}(\mathrm{t_i})|}$$

where $|T_r(t_i)|$ *denotes the number of documents in* T_r *in which* t_i *occurs at least once, and*

$$\mathrm{tf}(\mathrm{t_i}, \mathrm{d_j}) = \begin{cases} 1 + \log(N(t_i, d_j)) \; \textit{if} \; N(t_i, d_j) > 0 \\ \\ 0 \qquad\qquad\qquad\qquad\quad \textit{otherwise} \end{cases}$$

where $N(t_i, d_j)$ *denotes the frequency of terms* t_i *occurs in document* d_j *by counting all its nonstop words.*

To prevent the value of $|T_r(t_i)|$ to be zero, *Laplace Adjustment* is taken to add an observed count.

Definition 14. *The support of a keyword* t_A *in a collection is:*

$$\mathrm{support}(\mathrm{t_A}, \mathrm{T_r}) = \frac{1}{|\mathrm{T_r}|} \sum_{\mathrm{i}=0}^{|\mathrm{T_r}|} \mathrm{tfidf}(\mathrm{t_A}, \mathrm{d_j})$$

where, $|T_r(t_A)|$ *defines number of documents that contain term* t_A*, and* $|T_r|$ *denotes the number of documents in the collection.*

Traditionally, TFIDF values are often organized into the following matrix form: Let a document d_j in T_r be represented as a vector $V_j = <\mathrm{tfidf}(\mathrm{t_1}, \mathrm{d_j}), \mathrm{tfidf}(\mathrm{t_2}, \mathrm{d_j}), \cdots, \mathrm{tfidf}(\mathrm{t_n}, \mathrm{d_j})>$ and therefore T_r be represented as a matrix $M_r = < V_1,\ V_2,\ \cdots,\ V_I,\ \cdots >^T$. Most previous works [12, 13, 15] proposed to finding the association rules or partitioning the association rules into clusters [6] from M_r.

However, these index values have no obvious connections with the semantics of documents. So the clustering of the document based on these partitioning or association rules are not very meaningful. In fact, these values are more like distributions of keywords. So, we choose

Definition 15. *Those terms* t_i *in document* d_j *as the keywords or the* 0*-simplexes, if the TFIDF* $\mathrm{tfidf}(\mathrm{t_i}, \mathrm{d_j})$ *is large.*

5.2 Simplexes, co-Occurrences of Keywords and High TFIDF

In the last subsection, we have focuses on the 0 dimension, here We will explain the higher dimension cases. Let us start with dimension one.

Definition 16. *The significance of keyword-associations in a collection will be defined in terms of 1-dimensional TFIDF of* t_A *and term* t_B*:*

$$\mathrm{significance}(\mathrm{t_A}, \mathrm{t_B}, \mathrm{T_r}) = \frac{1}{|\mathrm{T_r}|} \sum_{\mathrm{i}=0}^{|\mathrm{T_r}|} \mathrm{TFIDF_1}(\mathrm{t_A}, \mathrm{t_B}, \mathrm{d_i})$$

where

$$\mathrm{TFIDF_1(t_A, t_B, d_i) = tf(t_A, t_B, d_i) \log \frac{|T_r|}{|T_r(t_A, t_B)|}}$$

, $|T_r(t_A, t_B)|$ defines number of documents contained both term t_A and term t_B, and $|T_r|$ denotes the number of documents in a collection.

The term frequency $\mathrm{tf(t_A, t_B, d_i)}$ of both term t_A and t_B can be calculated as follows.

Definition 17.

$$\mathrm{tf(t_A, t_B, d_j)} = \begin{cases} 1 + \log(\min\{N(t_A, d_j), N(t_B, d_j)\}) \\ \textit{if } N(t_A, d_j) > 0 \textit{ and } N(t_B, d_j) > 0 \\ \\ 0 \\ \textit{otherwise.} \end{cases}$$

A minimal threshold θ is imposed to filter out the terms that their significance values are small. It helps us to eliminate the most common terms in a collection and the nonspecific terms in a document.

Next, we will define the support of co-occurrences keywords (keyword-association) in a document collection. Let t_A and t_B be two terms. The *support* defined in the document collection is as follows.

Definition 18. Support *denotes to be the significance of associations of term t_A and term t_B in a collection, that is,*

$$\mathrm{Support(t_A, t_B) = significance(t_A, t_B, T_r)}$$

Now, we will explain the q dimension cases.

Definition 19. *The significance of keyword-associations in a collection will be defined in terms of q-dimensional TFIDF of $t_{A_1} \ \ldots \ t_{A_q}$:*

$$\mathrm{significance(t_{A_1}, \ldots t_{A_q}, T_r) = \frac{1}{|T_r|} \sum_{i=0}^{|T_r|} TFIDF_q(t_{A_1}, \ldots t_{A_q}, d_i)}$$

where

$$\mathrm{TFIDF_q(t_{A_1}, \ldots t_{A_q}, d_i) = tf(t_{A_1}, \ldots t_{A_q}, d_i) \log \frac{|T_r|}{|T_r(t_{A_1}, \ldots t_{A_q})|}}$$

, $|T_r(t_{A_1}, \ldots t_{A_q})|$ defines number of documents contained all terms $t_{A_1}, \ldots t_{A_q}$ and $|T_r|$ denotes the number of documents in a collection.

The term frequency $\mathrm{tf(t_{A_1}, \ldots t_{A_q}, d_i)}$ can be calculated as follows.

Definition 20.

$$\mathrm{tf(t_{A_1}, \ldots t_{A_q}, d_j)} = \begin{cases} 1 + \log(\min\{N(t_{A_1}, d_j), \ldots, N(t_{A_q}, d_j)\}) \\ \textit{if } N(t_{A_1}, d_j) > 0 \ \ldots N(t_{A_q}, d_j) > 0 \\ \\ 0 \\ \textit{otherwise.} \end{cases}$$

Definition 21. Support *denotes the significance of keyword-associations of terms,* $t_{A_1}, \ldots t_{A_q}$*, in a collection, that is,*

$$\mathrm{Support(t_{A_1}, \ldots t_{A_q}) = significance(t_{A_1}, \ldots t_{A_q}, T_r)}$$

5.3 Simplicial Complex of Keywords

It is obvious that the support evaluated by *tfidf* satisfies the *Apriori* condition.

Example 3 Support *denotes the significance of keyword-associations of terms,* $t_{A_1}, \ldots t_{A_q}$*, in a collection, that is,*

$$\mathrm{Support(t_{A_1}, \ldots t_{A_q}) = significance(t_{A_1}, \ldots t_{A_q}, T_r)}$$

In the last section, we have observed that a $n + 1$-association is an abstract n-simplex, in fact, the set of all associations has more structures. In this section, we will investigate the mathematical structures of feature-associations. A data set may carry a set of distinct concepts. Each concept, we believe, is carried by a connected component of the complex of feature-associations. Here is our belief and our hypothesis:

- An IDEA (in the forms of a simplicial complex) may consist many CONCEPTs (in the form of connected components) that are constructed by PRIMITIVE CONCEPTs (in the form of maximal simplexes).
- A simplex is said to be a maximal if no other simplex in the complex is a superset of it. The geometric dimension represents the degree of preciseness or depth of the semantics that are represented by keyword-associations.

Example 4 *In Figure 1, we have an IDEA that consist of twelve attributes that are organized in the forms of* 3*-complex, denoted by* S^3. $S(a, b, c, d)$ *and* $S(w, x, y, z)$ *are two maximal simplexes of the highest dimension* 3*. Let us consider the* $(3, 2)$*-skeleton* S^3_2*, by removing all* 0*-simplexes and* 1*-simplexes from* S^3*:*

- CONCEPT$_1$ *composite of* $S(a, b, c, d)$ *and its four faces (*2*-simplices):* $S(a, b, c)$, $S(a, b, d)$, $S(a, c, d)$*, and* $S(b, c, d)$*;*
- CONCEPT$_2$ *composite of* $S(a, c, h)$
- CONCEPT$_3$ *composite of* $S(c, h, e)$
- CONCEPT$_4$ *composite of* $S(e, h, f)$
- CONCEPT$_5$ *composite of* $S(e, f, x)$
- CONCEPT$_6$ *composite of* $S(f, g, x)$
- CONCEPT$_7$ *composite of* $S(g, x, y)$

- CONCEPT$_8$ *composite of* $S(w,x,y,z)$ *and its four faces (*2*-simplices):* $S(w,x,y)$*,* $S(w,x,z)$*,* $S(w,y,z)$*, and* $S(x,y,z)$*.*

There are no common faces between any two simplexes, so S^3_2 *has eight connected components. For* S^3_3*, it consists of two non-connected* 3*-simplexes that organized two CONCEPTs (*CONCEPT$_1$ *and* CONCEPT$_8$*), which are independent maximal connected components.*

A complex, connected component or simplex of a skeleton represent a more technically refined IDEA, CONCEPT, PRIMITVE CONCEPT.

5.4 Concept Formulation

Based on a divide and conquer method, the algorithm recursively partitions the generated simplicial complex into two parts: one contains a specific simplex and the other does not contain such a simplex.

Hierarchical clustering performs on grouping the data based on the similar concepts among them. Unlike the conventional hierarchical clustering, the most latent semantics, i.e., those data have a concrete concept, is on the top of the hierarchy not at the bottom. Therefore, a hierarchical partition clustering is naturally from $(n,0)$-skeleton to (n,m)-skeleton ($m \geq 0$ and $m \leq n$). Each simplex in a skeleton represents an individual cluster at each skeleton. According to the connected components within each skeleton, some data are softly clustering into a lot of categories associated to their common faces. A common face identifies a common concept in a context.

A novel algorithm to formulate the hierarchical concepts from a set of high dimensional data is presented in this paper. Based on the generated concepts data can be hierarchically partitioned into distinct but overlapped clusters.

6 Experimental Results

Two data sets are involved in making the validation and evaluating the performance of our model and algorithm. Effectiveness is the important criterion for the validity of clustering.

The first dataset is Web pages collected from Boley et al.[6]. 98 Web pages in four broad categories: business and finance, electronic communication and networking, labor and manufacturing are selected for the experiments. Each category is also categorized into four subcategories. This data set has been used to compare our algorithm, LSS, with three traditional vector-based clustering methods, in which their similarity measures are distance-based, model-based, or association rules, separately.

The second dataset is the "Reuters-21578, Distribution 1" collection consisted of newswire articles. The articles are assigned into 135 so-called topics that are in use to affirm the clustering results.

In order to extract features from documents, Wordnet 2.0 and other ontology, such as MeSH, as our background knowledge are then chosen to select meaning corpus as features. All ingredients of terms within a short distance in a document are considered to be the co-occurred features and then use for generating a concept.

While considering relevant documents to a search query, if the TFIDF value of a term is large, then it will pull more weight than terms with lesser TFIDF values. The TFIDF value of features denotes the significance, i.e., the support, of the simplex [**?**]. If the TFIDF value of a simplex is lesser than a given minimum support, that simplex will be stopped continuing to generate its super-simplex.. The recursive generating simplices are in use for futher hierarchically data clustering.

The result of the algorithm, PDDP [6], is under consideration by all non-stop words, that is, the F1 database in their paper, with 16 clusters. The result of our algorithm, LSS, is under consideration by all non-stop words with the minimal support, 15%. Four hierarchical layers with 23 clusters have been produced. Removing the redundant, 19 separate clusters have extracted. According to some topics categorized into the same topic may mention different CONCEPTs, such as "computer manufacture" and "information manufacture", we thought they might belong to different clusters. However, in this experiment, we still follow the original defined class.

Table 1. The first dataset is compared with four algorithms, LSS, PDDP, k-means and AutoClass.

Method	LSS	PDDP	k_means	AutoClass	HCA
Precision	81.4%	65.6%	56.7%	34.2%	35%
Recall	76.2%	68.4%	34.9%	23.6%	22.5%
F_1 measure	0.787	0.67	0.432	0.279	0.274

There are 674 clusters in 8 hierarchical layers generated in the Reuter data set. Some terms indicated a generic category in Reuter classifications are not designated the same category, so that the number of clusters is larger than the number of Reuter's categories. Considering the *Oil* topic in the Reuter data set, it is a composite topic including 'Vegetable Oil', 'Crude Oil', and so on. There are about 1215 Reuter news clustered into the "Oil" group, which there are 1156 documents exactly in the "Oil" topic. 95% documents can be correctly clustered into "Oil." Some misclassified documents in "Oil" are related to "Gas," or "Fuel." Speaking strictly, those documents are able to say "*correctly*" classified. The other misclassified 19 documents that are assigned to the Reuter CPI (Consumer Price Index) topic describe the change of CPI is related to the change of oil prices. The subcategory "Crude Oil" of the cluster contains 520 (44%) documents, in which induces 88% precise rate by compared with the Reuter "Crude Oil" topic.

7 Conclusion

In order to perform clustering on high dimensional effectively and efficiently, we propose a topology-based method to naturally transfer the data into a latent semantic space. Several latent semantic patterns reveal connected components among the latent semantic space. According to highly association terms of each layered skeleton, the data can be hierarchically partitioned into several meaningful clusters.

Polysemy, *phrases* and *term dependency* are the limitations of search technology [22]. A single term is not able to identify a latent concept in a document, for instance, the term "Network" associated with the term "Computer", "Traffic", or "Neural" denotes different concepts. To discriminate term associations no doubt is concrete way to distinguish one category from the others. A group of solid term associations can clearly identify a concept. The term-associations (frequently co-occurring terms) of a given collection of Web pages, form a simplicial complex. The complex can be decomposed into connected components at various levels (in various levels of skeletons). We believe each such a connected component properly identify a concept in a collection of Web pages.

Some terms with similar meaning, for example, "anticipate," "believe," "estimate," "expect," "intend," "project", could be separated into several independent topics even with the other same sub-concepts. In our experiments, some data of a single concept have been specified into redundant clusters. That makes the number of clustering big. Thesauri and some other adaptive methods [**?**] are going to provide a solution for it. It will be further considered to solve in the future.

We can effectively discover such a simplicial complex and use them to cluster the collection of Web pages. Based on our web site and our experiments, we find that LSS is a very good way to organize the high dimensional data into several semantic topics. It illustrates that geometric complexes are effective models for automatic web pages clustering.

References

1. R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 International Conference on Management of Data (SIGMOD 93)*, pages 207–216, May 1993.
2. R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proceedings of the 20th VLDB Conference*, 1994.
3. T. W. Anderson. On estimation of parameters in latent structure analysis. *Psychometrika*, 19:1–10, 1954.
4. F. Beil, M. Ester, and X. Xu. Frequent term-based text clustering. In *Proceedings of 8th Int. Conf. on Knowledge Discovery and Data Mining (KDD 2002)*, Edmonton, Alberta, Canada, 2002.
5. M. W. Berry. Large scale sparse singular value computations. *International Journal of Supercomputer Applications*, 6(1):13–49, 1992.

6. D. Boley, M. Gini, R. Gross, E.-H. Han, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, and J. Moore. Document categorization and query generation on the world wide web using webace. *Artificial Intelligence Review*, 13(5-6):365–391, 1999.
7. S. Brin, R. Motwani, J. Ullman, and S. Tsur. Dynamic itemset counting and implication rules for market basket data. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, pages 255–264, 1997.
8. S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the Seventh International WWW Conference (WWW 98)*, Brisbane, Australia, 1998.
9. P. Cheeseman and J. Stutz. Baysian classification (autoclass): Theory and results. In U.M. Fayyad, G. Piatetsky-Shapiro, P. Smith, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 153–180. AAAI/MIT Press, 1996.
10. M. S. Chen, J. Han, and P. S. Yu. Data mining: An overview from a database perspective. *IEEE Transaction on Knowledge and Data Engineering*, 8(6):866–883, 1996.
11. D. R. Cutting, D. R. Karger, J. O. Pedersen, and J. W. Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proceedings of the Fifteenth Annual International ACM SIGIR Conference*, pages 318–329, 1992.
12. R. Feldman, Y. Aumann, A. Amir, W. Klósgen, and A. Zilberstien. Text mining at the term level. In *Proceedings of 3rd International Conference on Knowledge Discovery, KDD-97*, pages 167–172, Newport Beach, CA, 1998.
13. R. Feldman, I. Dagan, and W. Klósgen. Efficient algorithms for mining and manipulating associations in texts. In *Cybernetics and Systems, The 13th European Meeting on Cybernetics and Research*, volume II, Vienna, Austria, April 1996.
14. R. Feldman, M. Fresko, H. Hirsh, Y. Aumann, O. Liphstat, Y. Schler, and M. Rajman. Knowledge management: A text mining approach. In *Proceedings of 2nd International Conference on Practical Aspects of Knowledge Management*, pages 29–30, Basel, Switzerland, 1998.
15. R. Feldman and H. Hirsh. Mining associations in text in the presence of background knowledge. In *Proceedings of 3rd International Conference on Knowledge Discovery*, 1996.
16. W. B. Frakes and R. Baeza-Yates. *Information Retrieval Data Structures and Algorithms.* Prentice Hall, Englewood Cliffs, NJ, 1992.
17. N. Fuhr and C. Buckley. A probabilistic learning approach for document indexing. 9(3):223–248, 1991.
18. J. D. Holt and S. M. Chung. Efficient mining of association rules in text databases. In *Proceedings of CIKM*, Kansas City, MO, 1999.
19. A. Hyvarinen, J. Karhunen, and E. Oja. *Independent Component Analysis.* John Wiley & Son, 2001.
20. A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data.* Prentice Hall, 1988.
21. I. T. Jolliffe. *Principle Component Analysis.* Spring-Verlag, New York, 1986.
22. A. Joshi and Z. Jiang. Retriever: Improving web search engine results using clustering. In A. Gangopadhyay, editor, *Managing Business with Electronic Commerce: Issues and Trends*, chapter 4. World Scientific, 2001.
23. G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar. Multilevel hypergraph partition application in vlsi domain. *Proceedings ACM/IEEE Design Automation Conference*, 8:381–389, 1997.
24. T. Kohonen. *Self-Organization Maps.* Springer-Verlag, Berlin Heidelberg, 1995.

25. R. Kosala and H. Blockeel. Web mining research: A survey. *SIGKDD Explorations*, 2(1):1–15, 2000.
26. B. Lent, R. Agrawal, and R. Srikant. Discovering trends in text databases. In *Proceedings of 3rd International Conference on Knowledge Discovery, KDD-97*, pages 227–230, Newport Beach, CA, 1997.
27. K. I. Lin and H. Chen. Automatic information discovery from the invisible web. In *Proceedings of the The International Conference on Information Technology: Coding and Computing (ITCC'02), Special Session on Web and Hypermedia Systems*, 2002.
28. T. Y. Lin. Attribute (feature) completion - the theory of attributes from data mining prospect. In *Proceedings of 2002 IEEE International Conference on Data Mining (ICDM)*, pages 282–289, Maebashi, Japan, 2002.
29. T. Y. Lin and I-Jen Chiang "A simplicial complex, a hypergraph, 3 structure in the latent semantic space 4 of document clustering, International Journal of Approximate Reasoning, 2005
30. S.Y. Lu and K.S. Fu. A sentence-to-sentence clustering procedure for pattern analysis. *IEEE Transactions on Systems, Man and Cybernetics*, 8:381–389, 1978.
31. J. MacQueen. Isome methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297, Berkeley, CA, 1967. University of California Press.
32. M. Marchiori. The quest for correct information on the web: Hyper search engines. In *Proc. The Sixth International WWW Conference (WWW 97)*, Santa Clara, CA, 1997.
33. M. E. Maron and J. K. Kuhns. On relevance, probabilistic indexing, and information retrieval. *Journal of ACM*, 7:216–244, 1960.
34. D. Mladenic. Text-learning and related intelligent agents: A survey. *IEEE Intelligent Systems*, pages 44–54, 1999.
35. A. Moffat and J. Zobel. Compression and fast indexing for multi-gigabit text databases. *Australian Computing Journal*, 26(1):19, 1994.
36. J. S. Park, M. S. Chen, and P. S. Yu. Using a hash-based method with transaction trimming for mining association rules. *IEEE Transaction on Knowledge and Data Engineering*, 9(5):813–825, 1997.
37. B. Pinkerton. Finding what people want: Experiences with the webcrawler. In *Proc. The Second International WWW Conference*, Chicag, IL, 1994.
38. M. Rajman and R. Besanon. Text mining: Natural language techniques and text mining applications. In *Proceedings of seventh IFIP* 2.6 *Working Conference on Database Semantics (DS-7)*, Leysin, Switzerland, 1997.
39. G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1960.
40. G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill, 1983.
41. F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, pages 1–47, 2002.
42. E. Spanier. *Algebric Topology.* McGraw-Hill Book Company, New York, NY, 1966.
43. M. Viveros, *Extraction of Knowledge from Databases*, Thesis, California State University at Northridge , 1989.
44. R. Weiss, B. Velez, M. A. Sheldon, C. Manprempre, P. Szilagyi, A. Duda, and D. K. Gifford. Hypursuit: A hierarchical network search engine that exploits content-link hypertext clustering. In *Proceedings of the 7th ACM Conference on Hypertext*, New York, NY, 1996.

45. P. Willett. Extraction of knowledge from databases. *Information processing and management*, 24:577–597, 1988.
46. O. Zamir and O. Etzioni. Web document clustering: a feasibility demonstration. In *Proceedings of 19th international ACM SIGIR conference on research and development in information retrieval (SIGIR 98)*, pages 46–54, 1998.