Available online at www.sciencedirect.com
ScienceDirect
Expert Systems with Applications 33 (2007) 256–262
Expert Systems with Applications
www.elsevier.com/locate/eswa

Discover the semantic topology in high-dimensional data

I-Jen Chiang *

Graduate Institute of Medical Informatics, Taipei Medical University, Taipei 110, Taiwan, ROC
Graduate Institute of Biomedical Engineering, National Taiwan University, Taipei 100, Taiwan, ROC

Abstract

Discovering the homogeneous concept groups in the high-dimensional data sets and clustering them accordingly are contemporary challenge. Conventional clustering techniques often based on Euclidean metric. However, the metric is ad hoc not intrinsic to the semantic of the documents. In this paper, we are proposing a novel approach, in which the semantic space of high-dimensional data is structured as a simplicial complex of Euclidean space (a hypergraph but with different focus). Such a simplicial structure intrinsically captures the semantic of the data; for example, the coherent topics of documents will appear in the same connected component. Finally, we cluster the data by the structure of concepts, which is organized by such a geometry.

Keywords: Document clustering; Association rules; Hierarchical clustering; Simplicial complex

1. Introduction

In various domains applications often lead to very high-dimensional data; the dimension of the data being in the hundreds, thousands or more, for example in text/web mining for browsing related documents that matched users query and bioinformatics for finding out genes and proteins that have similar functionality. In addition to the high dimensionality, these data sets are also often sparse. Heterogeneous concept groups are combined in the data sets. Clustering such high-dimensional data sets to discover homogeneous concept groups is a contemporary challenge.

Clustering analysis divides data into meaningful groups (clusters). The resulting clusters should capture concepts within the data. Taxonomies and conceptual hierarchies are crucial way making use of declarative concepts about the data it intents to deal with. Conventional clustering techniques become computationally expensive when the data to be clustering is getting large when there are millions of instances, many thousands of features, and many thousands of clusters. Some schemes on the measures of similarity or distance among the data fail to produce meaningful clusters if the number of attributes is large. For text/web mining, it is hard to say that two features are relevant if both of them occur frequently in the collection of documents but are far away each other located in a document, because multiple concepts can be simultaneously defined in a single Web page.

Information retrieval is normally making use of cluster analysis to organize documents into a collection of topic-coherent groups (Rijsbergen, 1979). In addition to aim users better understand the retrieval documents to focus their search, clustering has been used as alternate organization of documents (Hearst & Pedersen, 1996). The purpose that we seek is to investigate the implicit structure discovered by using a clustering method. Such a structure would help user to resolve their information needs more efficiently and more effectively.

This paper introduces a novel algorithm for clustering to discover the semantic structure based on combinatorial topology that is efficient when an application domain is large. In a real world application given a high-dimensional data set, it often mixes up multiple heterogeneous concepts. These concepts are considered to organize a

* Tel.: +886 2 3393 1666.
E-mail address: ijchiang@tmu.edu.tw

0957-4174/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2006.05.033

high-dimensional semantic space in which could contain several separated concepts. Along with those separate concepts, a data set can be clustered into meaningful groups. Each primitive concept can be considered to be the co-occurrences or associations (high frequent itemsets) of features in the data set, which are obviously as simplices in combinatorial topology. Therefore, the semantic space is represented as simplicial complex composed of a collection of connected simplices.

It is naturally to cluster the data set groups semantically in accordance with the simplicial complex. For example, the association that consists of "wall" and "street" denotes some financial notions that have meaning beyond the two nodes, "wall" and "street". This is similar to the notion of open segment (v_0, v_1), in which two end points represent one-dimensional geometric object that have meaning beyond the two 0-dimensional end points. In general, an r-association represents some semantic generated by a set of r keywords, may have more semantics or even have nothing to do with the individual features. The *Apriori* property of such associations is reflected exactly in the mathematical structure of simplicial complex in combinatorial topology. We could regard such a structure as a triangulation (partition, granulation) of the space of latent semantics of Web pages.

In what follows, we start by reviewing some related work on document clustering in Section 2. Section 3 introduces the concept hierarchy and its corresponding terminologies. The hierarchical clustering algorithm for partitioning a high-dimensional is presented in Section 4. Documents can then be clustered based on primitive concepts identified by this algorithm. Section 5 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 (Kosala & Blockeel, 2000).

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 (Zamir & Etzioni, 1998), 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 (1988). Cutting, Karger, Pedersen, and Tukey (1992) introduced partition-based clustering algorithms for document clustering. Buckshot and fractionation were developed in Lin and Chen (2002). Greedy heuristic methods are used in the hierarchical frequent term-based clustering algorithm (Beil, Ester, & Xu, 2002) 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. Geometric representations of feature-associations

The goal of this section is to model the internal concepts that are hidden in a data set. We observe that (1) feature–feature inter-relationships represent and carry the intrinsic semantics or concepts hidden in a data set, and (2) the co-occurred feature associations, will be called feature-associations, represent the feature–feature inter-relationships. So key to model the hidden semantics or concepts in a data set is lied in modeling the feature-associations. Somewhat a surprise, the mathematical structure of feature-associations is a known geometric/topological subject, called simplicial complex.

So a natural way to represent the latent semantic in a data set is to use geometric and topologic notions that capture the totality of thoughts expressed in this data.

3.1. Combinatorial topology

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 $v_0, v_1, \ldots, v_n$ is the union of all vertices of those simplexes (Spanier, 1966, 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. Now, by regarding the features, as defined by high support values, as abstract vertices, an association of $n+1$ features, called $n+1$-association, is a combinatorial n-simplex: A $(n+1)$-association is a combinatorial n-simplex of keywords that often carries some deep semantics that are well beyond the "union" of its vertices, or faces individually.

A (n,r)-skeleton (denoted by S_r^n) of n-complex is a n-complex, in which all k-simplexes ($k \leqslant r$) have been removed. Two simplexes in a complex are said to be *directly connected* if the intersection of them is a non-empty 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 $r \leqslant h \leqslant k \leqslant n$.

The maximal r-connected sub-complex 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 $r = 0$.

3.2. The geometry of feature-associations

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 complex of feature-associations) 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 feature-associations.

Example 1. In Fig. 1, we have an idea that consist of twelve features that 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 simplices of the highest dimension 3. Let us consider the $(3,2)$-skeleton S_2^3, 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 $S(b,c,d)$;

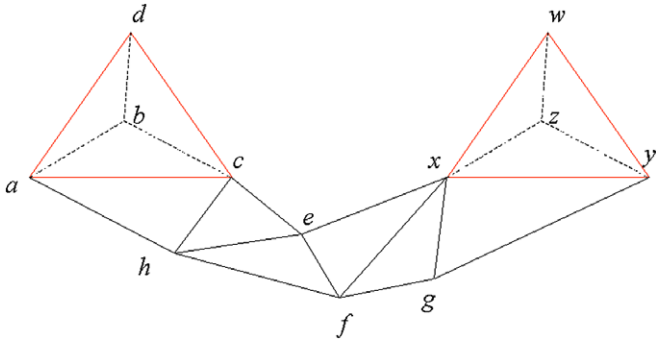

Fig. 1. A complex with twelve vertexes.

- 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 or PRIMITVE CONCEPT. If a maximal connected component of a skeleton contains only one simplex, this component is said to organize a primitive concept.

Definition 3. A set of maximal connected components is said to be *independent* if there are no common faces between any two maximal connected components.

4. Algorithm

Clustering the high-dimensional data can be based on the formed concepts from data. These concepts illustrate the latent semantic disciplines associated with the extracting features. Basically, the algorithm is divided into three main parts: first, to construct an undirected connected graph, i.e., a skeleton S_0^1 of the simplicial complex, from a data set; second, to generating the concepts from graph recursively; third, to cluster the data based on generated concepts.

4.1. Construct a simplicial complex

An undirected graph is constructed from features that have been extracted from a high-dimensional data. Perhaps thousands or millions of features have been reserved. Instead of converting data to be a matrix in which each instance is a vector, all its distinct attribute-value pairs are

transferred to be individual nodes (0-simplex). If two features are co-occurred in some instances, an edge (1-simplex) connected these two features are generated. Each edge is associated with a support value to denote how significant the association of the two nodes connected by the edge is. That is, a undirected graph $G = (V, E, W)$, where V is the set of nodes, E is the set of edges, and W denotes the set of significant supports of G, that is, supports can be defined on nodes or edges. The support defined on a node (0-simplex) is said to the *elementary support* that it means the significant value of the corresponding feature of the node.

The support determines how significant a concept is. As we have seen, a concept can be a n-simplex with n 0-simplices (nodes) or $\binom{n}{i+1}$ i-simplices. Let Support$(\cdot)$ be the support function of a simplex. Given a simplex S that is composite of simplices $S_1, S_2, \ldots, S_n$, the support of that S is defined to be

$$\text{Support}(S) = \text{Support}\left(\bigcup_{i=1}^{n} S_i\right)$$

where $\bigcup_{i=1}^{n} S_i$ denotes the set of data contained all the simplices. It obviously satisfies the following *Apriori* property Agrawal and Srikant (1994): If the supports of simplices $S_1, S_2, \ldots$ are less than a fixed value, all the supports of the supersets that contain any these simplices are less than that value. According to the *Apriori* property if the support of a simplex is less than a minimum support value, the simplex will be ignored to generate sup-simplices that contained it.

4.2. Concept formulation

In order to discover meaning of clusters in a data set, the algorithm always starts from a simplex with a maximum degree, i.e., the number of simplices made up of it is the maximum. Following a divide and conquer method, the algorithm recursively separates the simplicial complex into two parts: one contains that simplex and the other does not contain that simplex.

In Algorithm (Fig. 2), we define a simplicial difference between two simplices is as follows.

Definition 4. 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 the simplex S_1 but erases the simplex S_2 and all its faces.

Initially, the algorithm starts from a 0-simplex with the maximum degree to divide the whole simplicial complex into two independent complexes: one takes the 0-simplex as its common face and the other excludes the 0-simplex. Recursively, each simplicial complex is then hierarchical partitioned into several simplicial complexes.

Considering Example 1, let us start from 0-simplex $S(a)$. Let $S_C(n, 0)$ denote the *layered skeleton* of 0-simplices with a common face C where $n \geqslant 0$. Therefore, $S_\phi(n, 0) = \{S(a), S(b), S(c), S(d), S(e), S(f), S(g), S(w), S(x), S(y), S(z)\}$. Let $S_{S(a)}(n, 1) = S(a) \cup \{S(b), S(c), S(d), S(h)\}$ be the layered skeleton of 1-simplices contained a common face $S(a)$. So are $S_{S(b)}(n, 1), \ldots$, and $S_{S(z)}(n, 1)$.

Let $S_{CH}(n, m)$ be the m-layered skeleton with the common face C except the face H. If $S(a)$ is picked up as the common face for generating the skeleton of 2-simplices, then

$$S_{S(a)\cup S(b)}(n, 1) = \{S(a, b)\}$$
$$S_{S(a)\setminus S(b)}(n, 1) = \{S(a, c), S(a, d), S(a, h)\}$$

where both of previous two simplices belong to the skeleton S_2^3.

- Algorithm CONCEPT_DISCOVERY(C, S)
  - If C or S is empty, then return.
  - Find out a simplex H connected to C with 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_DISCOVERY(K, X).
  - Let $U = S - X$ and call CONCEPT_DISCOVERY(C, X).
  - The skeleton $S_m^n \leftarrow S_m^n \cup K$ where $m = |H \cup C|$ and $n \geq m$.

[1] A simplex is said to be connected to the other simplex if all nodes in both of them are complete connected together, that is, these two simplices will be integrated into a single complex.

Fig. 2. The algorithm is to find connected components in a simplicial complex recursively.

$$S_{S(a)\cup S(b)}(n,2) = \{S(a,b,c), S(a,b,d)\}$$
$$S_{S(a)\setminus S(b)}(n,2) = \{S(a,c,d), S(a,c,h)\}$$
$$S_{S(a)\cup S(c)}(n,2) = \{S(a,c,b), S(a,c,d), S(a,c,h)\}$$
$$\ldots$$
$$S_{S(a)\cup S(h)}(n,2) = \{S(a,c,h)\}$$

No doubt that $S_{S(a)\cup S(b)}$, $S_{S(a)S(b)}(n,2)$, $S_{S(a)\cup S(c)}(n,2)$, and $S_{S(a)\cup S(h)}(n,2)$ belong to the skeleton S_3^3, so is the following layered skeleton, i.e., a maximal simplex:

$$S_{S(a,b)\cup S(c)}(n,3) = \{S(a,b,c,d)\}$$

or

$$S_{S(a,b,c,d)}(n,3) = \{S(a,b,c,d)\}$$

Considering the layered skeleton composite of the simplex $S(a)$ but without the simplex $S(b,c)$, the following set of simplices could be generated:

$$S_{S(a)\setminus S(b,c)}(n,1) = \{S(a,d), S(a,h)\}$$

and

$$S_{S(a)\setminus S(b,c)}(n,2) = \phi$$

Example 2. Given the data set (as seen in Table 1) and a minimum support 2/5 (at least two independent features are co-occurred in the five instances), the S_1^3 skeleton is as Fig. 3, S_2^3 is shown in Fig. 4, and S_3^3 is depicted in Fig. 5. Each figure demonstrates a latent semantics. Fig. 5 shows the most concrete concept.

Table 1
The simple dataset

Class	Sky	Air	Humidity	Wind	Water
+	Sunny	Warm	Normal	Strong	Warm
+	Sunny	Warm	High	Strong	Warm
−	Rainy	Cold	High	Strong	Warm
+	Sunny	Warm	High	Strong	Cold

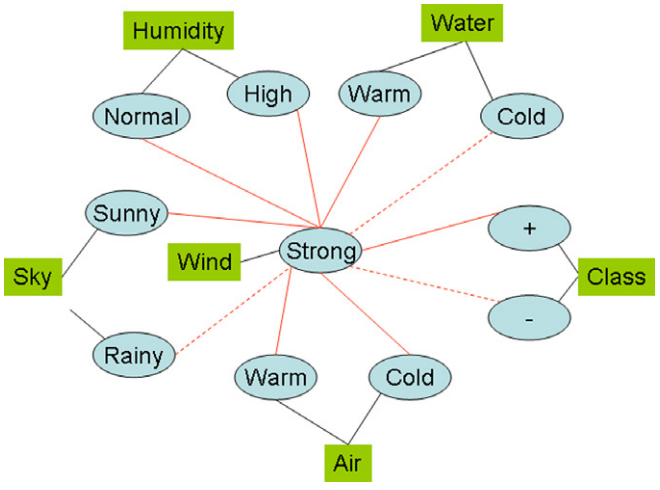

Fig. 3. The S_1^3 skeleton.

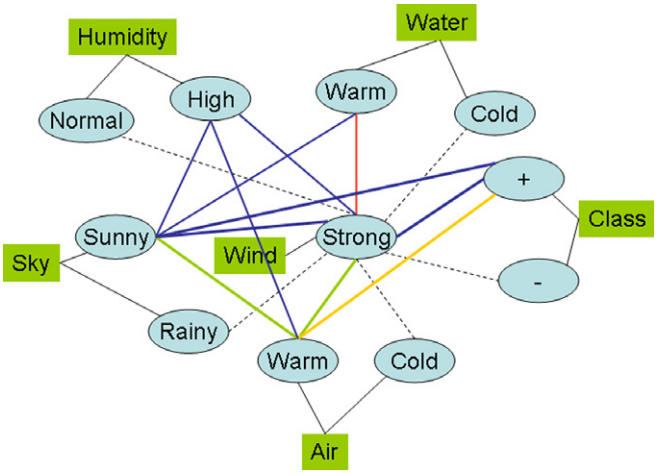

Fig. 4. The S_2^3 skeleton, in which the dish lines are not existed.

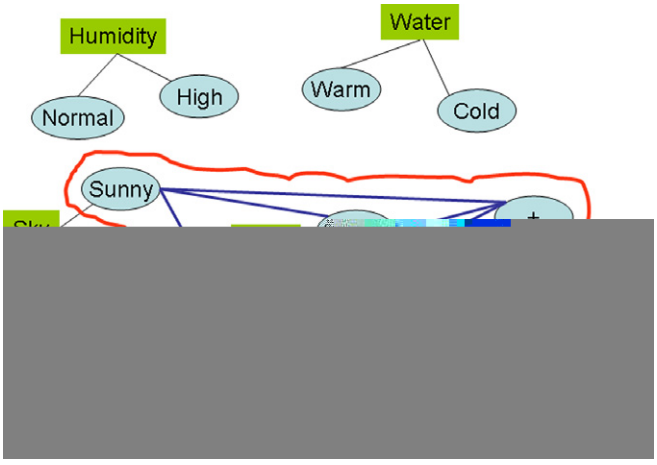

Fig. 5. The S_3^3 skeleton.

Since "Wind = Strong" is the most frequently occurred feature in this example, initially, it can be taken as the starting point for generating the (3,1)-skeleton after the simplicial complex being constructed. The data set is divided into two parts, one is the complex spawned by the feature "Wind = Strong", and the other are formed from other features except "Wind = Strong".

As shown in Fig. 4, there are six 2-simplices: S(Wind = Strong, Sky = Sunny, Air = Warm), S(Wind = Strong, Sky = Sunny, Humidity = High), S(Wind = Strong, Sky = Sunny, Water = Warm), S(Wind = Strong, Sky = Sunny, Class = +), S(Wind = Strong, Humidity = High, Air = Warm), and S(Wind = Strong, Air = Warm, Class = +) spawned from the feature "Wind = Strong".

The others created two simplices: S(Air = Warm, Sky = Sunny, Humidity = High), and S(Air = Warm, Sky = Sunny, Class = +). The union of all simplices in the skeleton becomes a connected component. It is easy to find that S(Wind = Strong, Sky = Sunny) is an 1-simplex with the highest degree of all simplices, a maximal simplex S(Wind = Strong, Sky=Sunny, Air = Warm, Class = +) can be generated from it as seen in Fig. 5.

According to this example, the skeleton S_m^n is obtained by take the union of all the m-layered skeleton of each possible m-simplex, in which $m \leqslant n$. Of course, if n is very big and m is very small, it becomes computational explosive. However, except the set of data are almost redundant, it is impossible to explore all layers. From the hierarchy of skeletons, one layered skeleton can obtain directly from its next lower level. Connected components in one layer can be the union of all layered skeletons.

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 close concept, is near the top of the hierarchy not the bottom. Therefore, a hierarchical partition clustering is naturally from $(n,0)$-skeleton to (n,m)-skeleton ($m \geqslant 0$ and $m \leqslant n$). 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. However, since some simplices can be further generated a same simplex at the next layered skeleton, a complicate redundancy is not often happened until the maximal independent simplices (PRIMITIVE CONCEPTS) have found.

This paper presents a novel algorithm to formulate the hierarchical concepts from a set of high-dimensional data. Based on the generated concepts data can be hierarchically partitioned into distinct but overlapped clusters. All the generated concepts follow the *Apriori* property of association rules, so the time complexity of this algorithm is no more than finding association rules in a dataset.

5. 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. (1999). 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 sub-categories. 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 (Lin & Chiang, 2004). 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 further hierarchically data clustering.

The result of the algorithm, PDDP (Boley et al., 1999), 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 (see Tables 2 and 3).

The evaluation was conducted for the cluster numbers ranging from 2 to 10 on the Reuter data set. For each given cluster number k, the performance scores were obtained by averaging those k randomly chosen clusters from the Reuter corpus in an one-run test. 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. Table 3 indicates the evaluation results using the Reuter dataset. Each category is labeled by selecting the most occurred concept for all its documents. 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

Table 2
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

Table 3
The performance of Reuter dataset by HCD

HCD	$k=2$	$k=3$	$k=4$	$k=5$
Precision (%)	93	90.8	93.8	86.1
Recall (%)	68	63.5	77.9	76.2
F_1 measure	0.834	0.774	0.814	0.77

"*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 sub-category "Crude Oil" of the cluster contains 520 (44%) documents, in which induces 88% precise rate by compared with the Reuter "Crude Oil" topic.

6. 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 hierarchical semantic space. Several latent semantic patterns reveal connected components within the 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 (Joshi & Jiang, 2001, chap. 4). 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 (Cohen & Richman, 2002) 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

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules. In *Proceedings of the 20th VLDB conference*.

Beil, F., Ester, M., & Xu, X. (2002). Frequent term-based text clustering. In *Proceedings of eighth international conference on knowledge discovery and data mining (KDD 2002)*, Edmonton, Alberta, Canada.

Boley, D., Gini, M., Gross, R., Han, E.-H., Hastings, K., Karypis, G., et al. (1999). Document categorization and query generation on the world wide web using webace. *Artificial Intelligence Review, 13*(5–6), 365–391.

Cohen, W. W., & Richman, J. (2002). Learning to match and cluster large high-dimensional data sets for data integration. In *Proceedings of the eighth ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 475–480). Edmonton, Alberta, Canada.

Cutting, D. R., Karger, D. R., Pedersen, J. O., & Tukey, J. W. (1992). Scatter/gather: A cluster-based approach to browsing large document collections. In *Proceedings of the fifteenth annual international ACM SIGIR conference* (pp. 318–329).

Hearst, M. A., & Pedersen, J. O. (1996). Reexamining the cluster hypothesis: Scatter/gather on retrieval results. In *Proceedings of the 19th annual international ACM/SIGIR conference on research and development in information retrieval* (pp. 76–84). Zurich, Switzerland.

Joshi, A., & Jiang, Z. (2001). Retriever: Improving web search engine results using clustering. In A. Gangopadhyay (Ed.), *Managing business with electronic commerce: Issues and trends*. World Scientific.

Kosala, R., & Blockeel, H. (2000). Web mining research: A survey. *SIGKDD Explorations, 2*(1), 1–15.

Lin, K. I., & Chen, H. (2002). 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.

Lin, T. Y., & Chiang, I. J. (2004). Automatic document clustering of concept hypergraph decompositions. In *Proceedings of SPIE*. Vol. 5098. (pp. 168–177). Orlando, FL.

Rijsbergen, C. J. (1979). *Information retrieval*. Butterworths.

Spanier, E. (1966). *Algebric topology*. New York, NY: McGraw-Hill Book Company.

Willett, P. (1988). Extraction of knowledge from databases. *Information Processing and Management, 24*, 577–597.

Zamir, O., & Etzioni, O. (1998). Web document clustering: a feasibility demonstration. In *Proceedings of 19th international ACM SIGIR conference on research and development in information retrieval (SIGIR 98)* (pp. 46–54).