Chapter 14
Web Mining: Extracting Knowledge from the World Wide Web

Zhongzhi Shi, Huifang Ma, and Qing He

Abstract This chapter addresses existing techniques for Web mining, which is moving the World Wide Web toward a more useful environment in which users can quickly and easily find the information they need. In particular, this chapter introduces the reader to methods of data mining on the Web developed by our laboratory, including uncovering patterns in Web content (semantic processing, classification, clustering), structure (retrieval, classical link analysis method), and event (preprocessing of Web event mining, news dynamic trace, multi-document summarization analysis). This chapter would be an excellent resource for students and researchers who are familiar with the basic principles of data mining and want to learn more about the application of data mining to their problems in Web mining.

14.1 Overview of Web Mining Techniques

The amount of information on the World Wide Web and other information sources such as digital libraries is quickly increasing. These information cover a wide variety of aspects. The huge information space spurs the development of data mining and information retrieval techniques. Web mining, which is moving the World Wide Web toward a more useful environment in which users can quickly and easily find information, can be regarded as the integration of techniques gathered by means of traditional data mining methodologies and its unique techniques.

As many believe, it is Oren Etzioni that first proposed the term of Web mining. He claimed that Web mining is the use of data mining techniques to automatically discover and extract information from World Wide Web documents and services [5]. Web mining is a research area that tries to identify the relevant pieces of infor-
Web mining uses techniques from data mining and machine learning to analyze Web data and documents. In general, Web mining uses document content, hyperlink structure, and event organization to assist users in meeting their needs for information. Madria et al. claimed the Web involves three types of data [21]: data on the Web, Web log data, and Web structure data. Cooley classified the data type as content data, structure data, usage data, and user profile data [18]. M. Spiliopoulou categorized the Web mining into Web usage mining, Web text mining, and user modeling mining [15]. Raymond systematically surveyed Web mining, pointing out some confusions regarding the usage of term Web Mining and suggested three Web mining categories [17]. When looked upon in data mining terms, Web mining can be considered to have three operations of interests: clustering (finding natural groupings of information for users), association (which URLs tend to be more important), and event analysis (organization of information).

Web content mining is the process to discover useful information from the content of a Web page. Since Web data are mainly semi-structured or even unstructured, Web content mining therefore combines available applications of data mining and its own unique approaches. In the following section, we would like to introduce some research results in the field of Web content mining we conclude these years: including semantic text analysis by means of conceptual semantic space; a new way of classification: multi-hierarchy text classification and clustering analysis that is clustering algorithm based on Swarm Intelligence and k-Means.

Web structure mining exploits the graph structure of the World Wide Web. It takes advantage of the hyperlink structure of the Web as an (additional) information source. The Web is viewed as a directed graph whose nodes are the Web pages and the edges are the hyperlinks between them. The primary aim of Web structure mining is to discover the link structure of the hyperlinks at the inter-document level. In the following section, we will analyze Web structure mining through information retrieval’s point of view and compare two famous link analysis methods: PageRank vs. HITS.

Web event mining discovers and delivers information and knowledge in a real-time stream of events on the Web. A typical Web event (news in particular) is composed of news title, major reference time, news resource, report time, condition time, portrait, and location. We can use a knowledge management model to organize the event. In the following section, these problems will be addressed: preprocessing for Web event, mining news dynamic trace and multi-document summarization.

The remainder of this chapter is organized as follows. Web content mining techniques are explained in Sect. 14.2. Sect. 14.3 deals with Web structure mining. Web event mining is discussed in Sect. 14.4, conclusions and future works are mentioned in the last section.
14.2 Web Content Mining

Web content mining describes the automatic search of information resource available online [21], and involves particularly mining Web content data. It is a combination of novel methods from a wide range of fields including data mining, machine learning, natural language processing, statistics, databases, information retrieval and so on.

Unfortunately, much of the data is unstructured and semi-structured. The Web document usually contains different types of data, such as text, image, audio, video, metadata and hyperlinks. Providing a relational interface to all such databases may be complicated. This unstructured characteristic of Web data forces the Web content mining towards a more complicated approach.

Our lab has implemented a semantic indexing system based on concept space: GHUNT [16]. Some new technologies are integrated in GHUNT. GHUNT can be regarded as an all-sided solution for information retrieval on Internet [34]. In the following, some key technologies concerning Web mining are demonstrated: the way of constructing conceptual semantic space, multi-hierarchy text classification and clustering algorithm based on Swarm Intelligence and k-Means.

14.2.1 Classification: Multi-hierarchy Text Classification

The goal of text classification is to assign one or several proper classes to a document. At present, there are a lot of machine learning approaches and statistics methods used in text classification, including Support Vector Machines (SVM) [26], K-Nearest Neighbor Classification (KNN) [31], Linear Least Square Fit (LLSF) developed by Yang [32], decision trees with boosting by Apte [3], Neural network and Naïve Bayes [24] and so on.

Most of these approaches adopt the classical vector space model (VSM). In this model, the content of a document is formalized as a dot of the multi-dimension space and represented by a vector. The frequently used document representation in VSM is the so-called TF.IDF-vector representation. Lu introduced an improved approach named TF.IDF.IG by combining the information gain from information theory [20].

Our lab has proposed an approach of multi-hierarchy text classification based on VSM [23]. In this approach, all classes are organized as a tree according to some given hierarchical relations, and all the training documents in a class are combined into a class-document [30].

The basic insight supporting our approach is that classes that are attached to the same node have a lot more in common with each other than other classes. Based on this intuition, our approach divides the classification task into a set of smaller classification problems corresponding to the splits in the classification hierarchy. Each of these subtasks is significantly simpler than the original task, since the classifier at a node of the tree needs only to distinguish between a small number of classes. And this part of classes have a lot more in common with each other, so the models
of these classes will be based on a small set of features.

We first construct class models by feature selection after training the documents classified by hand corresponding to the classification hierarchy. In the selection of feature terms, we synthesize two factors, term frequency and term concentration. In the algorithm, all the training documents in one class will be combined into a class-document to perform feature selection. The algorithm CCM (construct class models) is listed as follows:

Input: A tree according to some given hierarchical relations (each node, except the root node, corresponds to a class and all the documents are classified into subclasses corresponding to the leaf nodes in advance)
Output: All the class models, saved as text files

Begin
Judge all the nodes from the bottom layer to the top layer using bottom-up method:
1. If the node $V_0$ is a leaf node, then analyze the corresponding class-document, including the term frequencies, the number of terms and the sum of all term frequencies.
2. If $V_0$ is not a leaf node (assume it has $t$ node children from $V_1$, $V_2$ to $V_t$, and there are $s$ terms from $T_1$, $T_2$, to $T_s$ in the corresponding class-document), then
   a. Calculate the probability of the class-document $d_i$ corresponding to $V_i$
   b. Calculate $H(D)$ and $H(D/T_k)$, then get $IG_k$, where $k = 1, 2, ..., s$, $H(D)$ is the entropy of the document collection $D$, $H(D/T_k)$ is the conditional entropy of term $T_k$
   c. Construct the class model $C_i$ corresponding to $V_i$, where $i = 1, 2, ..., t$
      i. Initialize $C_i$ to null
      ii. Calculate term frequency $W_{ik}$ of $T_k$, where $k = 1, 2, ..., s$
      iii. Resort all the terms to a new permutation $T_1, T_2, ..., T_s$ according to the descending weights
      iv. Judge the terms from $T_1$ to $T_s$ individually:
         If the number of feature terms in $C_i$ exceeds a certain threshold value $NUM_T$
            Then the construction of $C_i$ ends up
         Else
            If $W_{ik}$ exceeds a certain threshold value $\alpha$, the term frequency of $T_k$ exceeds a certain threshold value $\beta$, the term concentration of $T_k$ exceeds a certain threshold value $\gamma$ and $T_k$ is not in the stop-list given in advance, then $T_k$ will be a feature term and should be added to the class model $C_i$ with its weight.
         End

The calculation of term weight in this part considers two factors: term frequency and term position [23]. Then one top-down matching process is hierarchically performed from the root node of the tree until the proper subclass is found corresponding to a leaf node. For more details of the algorithm, you can refer to [23].

14.2.2 Clustering Analysis: Clustering Algorithm Based on Swarm Intelligence and k-Means

Although it is hard to organize the whole Web, it is feasible to organize Web search results of a given query. The standard method for information organization is
concept hierarchy and categorization. The popular technique for hierarchy construction is text clustering. Generally, major clustering methods can be classified into five categories: partitioning methods, hierarchical methods, density-based methods, grid-based methods and model-based methods. Many clustering algorithms have been proposed, such as CLARANS [10], DBSCAN [14], STING [27] and so on.

Our lab has proposed a document clustering algorithm based on Swarm Intelligence and K-Means: CSIM [28], which combines Swarm Intelligence with k-Means clustering technique. Firstly, an initial set of clusters is formed by swarm intelligence based clustering method which is derived from a basic model interpreting ant colony organization of cemeteries. Secondly, an iterative partitioning phase is employed to further optimize the results. Self-organizing clusters are formed by this method. The number of clusters is also adaptively acquired. Moreover, it is insensitive to the outliers and the order of input. Actually, the swarm intelligence based clustering method can be applied independently. But by second phase, the outliers which are single points on the ant-work plane are converged on the nearest neighbor clusters and the clusters which are piled too closely to collect correctly on the plane by chance are also split. K-means clustering phase softens the casualness of the swarm intelligence based method which is originated from a probabilistic model. The algorithm can be described as follows:

Input: document vectors to be clustered
Output: documents labeled by clustering number

1. Initialize Swarm similarity coefficient $\alpha$, ant number maximum iterative times $n$, slope $k$, and other parameters;
2. Project the data objects on a plane at random, i.e. randomly give a pair of coordinate $(x, y)$ to each data object;
3. Give each ant initial objects and initial state of each ant is unloaded;
4. for $i=1,2...$ // while not satisfying stop criteria
   a. for $j = 1,2...,ant\_number$;
      i. Compute the Swarm similarity of the data object within a local region with radius $r$;
      ii. If the ant is unloaded, compute picking-up probability $P_p$. Compare $P_p$ with a random probability $Pr$, if $P_p < Pr$, the ant does not pick up this object, another data object is randomly given the ant, else the ant pick up this object, the state of the ant is changed to loaded, a new random pair of coordinate is given to the ant;
      iii. If the ant is loaded, compute dropping probability $P_d$. Compare with a random probability $Pr$, if $P_d > Pr$, the ant drops the object, the pair of coordinate of the ant is given to the object. the state of the ant is changed to unloaded, another data object is randomly given the ant, else the ant continue moving loaded with the object, a new random pair of coordinate is given to the ant.
5. for $i = 1,2...,pattern\_num$; //for all patterns
   a. if this pattern is an outlier, label it as an outlier;
   b. else label this pattern a cluster serial number; recursively label the same serial number to those patterns whose distance to this pattern is smaller than a short distance $dist$. i.e. collect the patterns belong to a same cluster on the ant-work plane; Serial number $serial\_num +=$.
6. Compute the cluster means of the $serial\_num$ clusters as the initial cluster centers;
7. repeat
a. (re)assign each pattern to the cluster to which the pattern is the most similar, based on the mean value of the patterns in the cluster;
b. update the cluster means, i.e. calculate the mean value of the patterns for each cluster;
8. until not change.

If you want to know more about the algorithm, see [28].

14.2.3 Semantic Text Analysis: Conceptual Semantic Space

An automatic indexing and concept classification approach to a multilingual (Chinese and English) bibliographic database is presented by H. Chen [8]. A concept space of related descriptors was then generated using a co-occurrence analysis technique. For concept classification and clustering, a variant of a Hopfield neural network was developed to cluster similar concept descriptors and to generate a small number of concept groups to represent (summarize) the subject matter of the database.

A simple way to generate concept semantic space is by using HowNet, which is an on-line common-sense knowledge base unveiling inter-conceptual relations and inter-attribute relations of concepts as connoting in lexicons of the Chinese and their English equivalents [29]. We develop a new way to establish concept semantic space by using clustering algorithm based on Swarm Intelligence and k-Means. Here, the bottom up way is taken. We first classify the Web pages from internet into some domains, and then the Web pages that belong to each domain are clustered. Such would evade subjective warp caused by the departure between document and level in the top down way. Also we can adjust the parameter to make the hierarchy flexible enough.

The concept space can be used to facilitate querying and information retrieval. One of the most important aspects is how to generate the link weights in concept space of specific domain automatically. Before generating the concept space, the concepts of a certain domain must be identified. In the scientific literature domain, the concepts are relatively stable, and there are existing thesauruses that can be adopted. However, in a certain domain, news domain in particular, the concepts are dynamic, so there is no existing thesaurus and it is unrealistic to generate a thesaurus manually. We need to extract the concepts from the document automatically. Using the following formulae we could compute the information gain of each term for classification, which sets a foundation for thesaurus construction.

\[
InfGain(F) = P(F) \sum_i P(\psi_i|F) \log \frac{P(\psi_i|F)}{P(\psi_i)} + P(\overline{F}) \sum_i P(\psi_i|\overline{F}) \log \frac{P(\psi_i|\overline{F})}{P(\psi_i)}
\]

(14.1)

Where \( F \) is a term, \( P(F) \) is the probability of that term \( F \) occurred, \( F \) means that term \( F \) doesn’t occur, \( P(\psi_i) \) is the probability of the \( i - \) th class value, \( P(\psi_i|F) \)
is the conditional probability of the \( i \)th class value given that word \( F \) occurred. If \( \text{InfGain}(F) > \omega \), we choose term \( F \) as the concept. Although in this way the thesaurus generated is not as thorough and precise as the way constructed manually in the field of scientific literature, it is acceptable.

After we have recognized the concept of a class, we could generate the concept space of that class automatically. Chen’s method that uses co-occurrence analysis and Hopfield net is adopted [9] [25]. By means of using co-occurrence analysis, we compute the term association weight between two terms, and then the asymmetric association between terms is computed; we could activate related terms in response to user’s input. This process is accomplished by a single-layered Hopfield network. Each term is treated as a neuron, and the association weight is assigned to the network as the synaptic weight between nodes. After the initialization phase, we repeat the iteration until convergence. For detailed description of establishment of concept semantic space, you can refer to [33].

In this section, some research results we conclude are introduced: including a new way of classification: multi-hierarchy text classification; clustering analysis that is clustering algorithm based on Swarm Intelligence and k-Means and semantic text analysis by means of conceptual semantic space.

### 14.3 Web Structure Mining: PageRank vs. HITS

Web structure mining is essentially about mining the links on the Web. Web pages are actually instances of semi-structured data, and thus mining their structure is critical to extracting information from them. The structure of a typical Web graph consists of Web pages as nodes and hyperlinks as edges connecting between two related pages. Web structure mining can be regarded as the process of discovering structure information from the Web. In the following, we would like to compare famous link analysis methods: PageRank vs. HITS.

Two most influential hyperlink based search algorithms PageRank and HITS were reported during 1997-1998. Both algorithms exploit the hyperlinks of the Web to rank pages according to their levels of "prestige" or "authority".

PageRank Algorithm is originally formulated by Sergey Brin and Larry Page, PhD students from Stanford University, at Seventh International World Wide Web Conference (WWW) in April, 1998 [22]. The algorithm is determined for each page individually according to their authoritativeness.

More specifically, a hyperlink from a page to another page is an implicit conveyance of authority to the target page. The more in-links that a page \( i \) receives, the more prestige the page \( i \) has. Let the Web as a directed graph \( G = (V, E) \) and let the total number of pages be \( n \). The PageRank score of the page \( i \) (denoted by \( P(i) \)) is defined by [2]:

\[
P(i) = \sum_{(i,j) \in E} \frac{P(j)}{O_j} \tag{14.2}
\]
$O_j$ is the number of out-link of $j$.

Unlike PageRank which is a “static” ranking algorithm, HITS is search-query-dependent. HITS was proposed by Jon Kleinberg (Cornell University), at Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, January 1998 [12]. This algorithm is initially developed for ranking documents based on the link information among a set of documents.

More specifically, for each vertex $v$ in a subgraph of interest: $a(v)$ shows the authority of $v$ while $h(v)$ demonstrates the hubness of $v$. A site is very authoritative if it receives many citations. Citation from important sites weight more than citations from less-important sites. Hubness shows the importance of a site. A good hub is a site that links to many authoritative sites. Authorities and hubs have a mutual reinforcement relationship.

In this section, we briefly introduced prestige link analysis: PageRank and HITS.

### 14.4 Web Event Mining

Web event mining is the application of data mining techniques to Web event repositories in order to produce results that can be used as the event’s cause and effect. Event mining is not a new concept, which has already been used in Petri nets, stochastic modeling, etc. However, there are new opportunities that come from the large amount of data that is stored in various databases.

An event can be defined as related topics in a continuous stream of newswire stories. Concept terms of an event are derived from statistical context analysis between sentences in the news story and stories in the concept database. Detection Methods also includes cluster representation. DeJong uses frame-based objects called “sketchy scripts” [7]. D. Luckham [4] provides a framework for thinking about complex events and for designing systems that use such events.

Our lab has implemented an intelligent event organization and retrieval system [11], which uses machine-learning techniques, and combines specialties of news to organize and retrieve Web news documents. The system consists of a pre-processing process for news documents to get related knowledge, event constructor component to collect the correlative news reports together, cause-effect learning process and event search engine.

There are many attributes for an event such as event ID ($ID_d$), name of the event ($Name_d$), time of the event ($Time_d$), model of the event ($Model_d$), the documents belonging to the event($DocS$). Besides, document knowledge ($Know_d$) is a set of pairs, term and weight. Namely, $Know_d = \{(term_1, weight_1), (term_2, weight_2), ..., (term_s, weight_s)\}$. model knowledge of event ($Know_m$), like $Know_m(j)$, is a set of pairs consists of term and weight. $Know_m(j) = \{(term_1, weight_1, j), ..., (term_n, weight_n, j)\}$. 
14.4.1 Preprocessing for Web Event Mining

The preprocessing process consists of two parsing processes: parse HTML files and segment text document. The term segmentation algorithm extracts Chinese terms based on the rule of “long term first” to resolve ambiguity. Meanwhile, the term segmentation program combines Name Entity Recognition Algorithm. That is, human names and place names can be extracted and marked automatically, while segmenting terms.

14.4.1.1 Event Template Learning

An event template represents participants in an event described by the keywords and relations among the participants. The model knowledge of event is the most important reference when constructing event template. We can learn it from a group of training documents that report a same event. The key problem of model knowledge is to compute the support of term to event, denoted as $weight_{t_i,st}$, in following expression:

$$weight_{t_i,st} = \sum_{D_j} weight_{t_i,D_j}$$

$$weight_{t_i,st} = \frac{weight_{t_i,st}}{\max\{weight_{t_i,D_j st}\}}$$

Here, $weight_{t_i,D_j}$ is the term support of $t_i$ to $D_j$, $D_j$ is a document belongs to the event. $weight_{t_i,st}$ is normalized to the range between 0 and 1.

Due to the diversity of Internet documents, the number of terms in an event is large, and their term support is generally low. The feature selection process at the event level is more sophisticated than that at the document level. From analyzing these feature terms, we find the terms that have bigger weight can be represented as features of the event. For details please refer to [13].

14.4.1.2 Time Information Learning

Time is important factor for a news report, which can reveal time information of the event. According to time information, we can get event time, and organize cause-effect of event in time order. After analyzing many news reports on the Internet, we find there are different kinds of time. The formal format of time is defined as:

$$Time = year - month - day - hour - minute - second$$

Web document developers organize the time in different style according to their habits. According to the character of time, there are 3 sorts of form: Absolute
14.4.2 Multi-document Summarization: A Way to Demonstrate Event’s Cause and Effect

Cause-Effect of the event can represent the complete information coverage of an event, which is organized via listing the titles or abstracts of the documents belonging to the event in some order, such as time. The process of learning the Cause-Effect knowledge can follow the process: learning time, → learning summary, → sorting summary or title in time order → cause effect of event.

Another way of producing event’s cause and effect is Multi-document summarization, which presents a single summary for a set of related source documents. Multi-document summarizations include extractive and abstractive method. Extractive summarization is based on statistical techniques to identify similarities and differences across documents. It involves assigning salience scores to some units of the documents and then extracting the units with highest scores. While abstraction summarization usually needs information fusion, sentence compression and reformulation.

Researchers from Cornell University used a method of Latent Semantic Indexing to ascertain the topic words and generate summarization [19]. NeATS uses sentence position, term frequency, topic signature and term clustering to select important content [13]. The MEAD system is developed by Columbia University used a method called Maximal Marginal Relevance (MMR) to select sentences for summarization [1]. Newsblaster, a news-tracking tool developed by Columbia University generates summarizations of daily news [6].

In this section, these problems concerning Web event mining are addressed: preprocessing for Web event, mining news dynamic trace and multi-document summarization.

14.5 Conclusions and Future Works

In conclusion, this chapter addresses existing solutions for Web mining, which is moving the World Wide Web toward a more useful environment in which users can quickly and easily find the information they need. In particular, this chapter introduces the reader to methods of data mining on the Web, including uncovering patterns in Web content (semantic processing, classification, clustering), structure (retrieval, classical Link Analysis method), and event (preprocessing of Web event mining, news dynamic trace, multi-document summarization analysis). This chapter demonstrates our implementation of semantic indexing system based on concept
space: GHUNT as well as an intelligent Event Organization and Retrieval system.

The approaches described in the chapter represent initial attempts at mining content, structure and event of Web. However, to improve information retrieval and the quality of searches on the Web, a number of research issues still need to be addressed. Research can be focused on Semantic Web Mining, which aims at combining the two fast-developing research areas Semantic Web and Web Mining. The idea is to improve the results of Web Mining by exploiting the new semantic structures in the Web. Furthermore, Web Mining can help to build the Semantic Web. Various conferences are now including panels on Web mining. As Web technology and data mining technology mature, we can expect good tools to be developed to mine the large quantities of data on the Web.

Acknowledgements This work is supported by the National Natural Science Foundation of China (No. 60435010, 60675010), 863 National High-Tech Program (No.2006AA01Z128,2007AA01Z132) and National Basic Research Priorities Programme (No.2007CB311004).

References

16. Qing He, Ziyan Jia, Jiayou Li, Haijun Zhang, Qingyong Li, Zhongzhi Shi: GHUNT: A SEMANTIC INDEXING SYSTEM BASED ON CONCEPT SPACE. International Conference on Natural Language Processing and Knowledge Engineering (IEEEENLP&KE-2003), 2003: 716-721
29. www.keenage.com
33. Yuan Li, Qing He, Zhongzhi Shi: Association Retrieval based on concept semantic space. (in Chinese) Journal of University of Science and Technology Beijing, 2001, 23(6): 577-580