
UNIT 12 TEXT AND WEB MINING

Structure

- 12.0 Introduction
- 12.1 Objectives
- 12.2 Text Mining and its Applications
- 12.3 Text Preprocessing
- 12.4 BoW and TF-IDF For Creating Features from Text
 - 12.4.1 Bag of Words
 - 12.4.2 Vector Space Modeling for Representing Text Documents
 - 12.4.3 Term Frequency-Inverse Document Frequency
- 12.5 Dimensionality Reduction
 - 12.5.1 Techniques for Dimensionality Reduction
 - 12.5.1.1 Feature Selection Techniques
 - 12.5.1.2 Feature Extraction Techniques
- 12.6 Web Mining
 - 12.6.1 Features of Web Mining
 - 12.6.2 Web Mining Tasks
 - 12.6.3 Applications of Web Mining
- 12.7 Types of Web Mining
 - 12.7.1 Web Content Mining
 - 12.7.2 Web Structure Mining
 - 12.7.3 Web Usage Mining
- 12.8 Mining Multimedia Data on the Web
- 12.9 Automatic Classification of Web Documents
- 12.10 Summary
- 12.11 Solutions/Answers
- 12.12 Further Readings

12.0 INTRODUCTION

In the earlier unit, we had studied about the Clustering. In this unit let us focus on the text and web mining aspects. This unit covers the introduction to text mining, text data analysis and information retrieval, text mining approaches and topics related to web mining.

12.1 OBJECTIVES

After going through this unit, you should be able to:

- understand the significance of Text Mining;
- describe the dimensionality reduction of text;
- narrate text mining approaches;
- discuss the purpose of web mining and web structure mining; and
- describe mining the multimedia data on the web and web usage mining.

12.2 TEXT MINING AND ITS APPLICATIONS

Text mining, also known as text data mining, is the process of transforming unstructured text into a structured format to identify meaningful patterns and new insights. By applying advanced analytical techniques, such as Naïve Bayes, Support Vector Machines (SVM), and other deep learning algorithms, companies are able to explore and discover hidden relationships within their unstructured data.

Text is a one of the most common data types within databases. Depending on the database, this data can be organized as:

- *Structured data:* This data is standardized into a tabular format with numerous rows and columns, making it easier to store and process for analysis and machine learning algorithms. Structured data can include inputs such as names, addresses, and phone numbers.
- *Unstructured data:* This data does not have a predefined data format. It can include text from sources, like social media or product reviews, or rich media formats like, video and audio files.
- *Semi-structured data:* As the name suggests, this data is a blend between structured and unstructured data formats. While it has some organization, it doesn't have enough structure to meet the requirements of a relational database. Examples of semi-structured data include XML, JSON and HTML files.

Since 80% of data in the world resides in an unstructured format, text mining is an extremely valuable practice within organizations. Text mining tools and Natural Language Processing (NLP) techniques, like information extraction, allow us to transform unstructured documents into a structured format to enable analysis and the generation of high-quality insights. This, in turn, improves the decision-making of organizations, leading to better business outcomes.

For example, the tweets or messages on WhatsApp, Facebook, Instagram or through text messages and the majority of this data exists in the textual form which is highly unstructured in nature now in order to produce significant and actionable insights from the text data it is important to get acquainted with the techniques of text analysis.

Text analysis or text mining is the process of deriving meaningful information from natural language. It usually involves the process of structuring the input text deriving patterns within the structured data and finally evaluating the interpreted output compared with the kind of data stored in database text is unstructured amorphous and difficult to deal with algorithmically. Nevertheless in the modern culture text is the most common vehicle for the formal exchange of information now as text mining refers to the process of arriving high-quality information from text the overall goal here is to turn the text into data for analysis.

Text mining has various areas to explore as shown below:

Information Extraction is the techniques of taking out the information from the unstructured text data or semi-structured data contains in the electronic documents.

The processes identify the entities, then classify them and store in the databases from the unstructured text documents.

Natural Language Processing (NLP): The human language which can be found in WhatsApp chats, blogs, social media reviews or any reviews which are written in any offline documents. This is done by the application of NLP or natural language processing. NLP refers to the artificial intelligence method of communicating with an intelligent system using natural language by utilizing NLP and its components one can organize the massive chunks of textual data perform numerous or automated tasks and solve a wide range of problems such as automatic summarization, machine translation, speech recognition and topic segmentation.

Data Mining: Data mining refers to the extraction of useful data, hidden patterns from large data sets. Data mining tools can predict behaviors and future trends that allow businesses to make a better data-driven decision. Data mining tools can be used to resolve many business problems that have traditionally been too time-consuming.

Information Retrieval: Information retrieval deals with retrieving useful data from data that is stored in our systems. Alternately, as an analogy, we can view search engines that happen on websites such as e-commerce sites or any other sites as part of information retrieval.

Text mining often includes the following techniques:

- **Information Extraction** is a technique for extracting domain specific information from texts. Text fragments are mapped to field or template lots that have a definite semantic technique.
- **Text Summarization** involves identifying, summarizing and organizing related text so that users can efficiently deal with information in large documents.
- **Text Categorization** involves organizes documents into a taxonomy, thus allowing for more efficient searches. It involves the assignment of subject descriptors or classification codes or abstract concepts to complete texts.
- **Text Clustering** involves automatically clustering documents into groups where documents within each group share common features.

12.2.1 Applications of Text Mining

Following are some of the applications of Text Mining:

- **Customer service**: There are various ways in which we invite customer feedback from our users. When combined with text analytics tools, feedback systems such as chatbots, customer surveys, Net-Promoter Scores, online reviews, support tickets, and social media profiles, enable companies to improve their customer experience with speed. Text mining and sentiment analysis can provide a mechanism for companies to prioritize key pain points for their customers, allowing businesses to respond to urgent issues in real-time and increase customer satisfaction.
- **Risk management**: Text mining also has applications in risk management. It can provide insights around industry trends and financial markets by monitoring shifts in sentiment and by extracting information from analyst reports and whitepapers. This is particularly valuable to banking institutions as this data provides more confidence when considering business investments across various sectors.

- **Maintenance:** Text mining provides a rich and complete picture of the operation and functionality of products and machinery. Over time, text mining automates decision making by revealing patterns that correlate with problems and preventive and reactive maintenance procedures. Text analytics helps maintenance professionals unearth the root cause of challenges and failures faster.
- **Healthcare:** Text mining techniques have been increasingly valuable to researchers in the biomedical field, particularly for clustering information. Manual investigation of medical research can be costly and time-consuming; text mining provides an automation method for extracting valuable information from medical literature.
- **Spam filtering:** Spam frequently serves as an entry point for hackers to infect computer systems with malware. Text mining can provide a method to filter and exclude these e-mails from inboxes, improving the overall user experience and minimizing the risk of cyber-attacks to end users.

12.2.2 Text Analytics

Text mining emphasizes more on the process, whereas text analytics emphasizes more on the result. Text mining and analytics implies to turn text data into high quality information or actionable knowledge.

Text analytics is a sub-set of Natural Language Processing (NLP) that aims to automate extraction and classification of actionable insights from unstructured text disguised as emails, tweets, chats, tickets, reviews, and survey responses scattered all over the internet.

Text analytics or text mining is multi-faceted and anchors NLP to gather and process text and other language data to deliver meaningful insights.

12.2.3 Need for Text Analytics

Need for Text Analytics is to:

Maintain Consistency: Manual tasks are repetitive and tiring. Humans tend to make errors while performing such tasks – and, on top of everything else, performing such tasks is time-consuming. Cognitive biasing is another factor that hinders consistency in data analysis. Leveraging advanced algorithms like text analytics techniques enable performing quick and collective analysis rationally and provide reliable and consistent data.

Scalability: With text analytics techniques, enormous data across social media, emails, chats, websites, and documents can be structured and processed without difficulty, helping businesses improve efficiency with more information.

Real-time Analysis: Real-time data in today’s world is a game-changer. Evaluating this information with text analytics allows businesses to detect and attend to urgent matters without delay. Applications of Text analytics enable monitoring and automated flagging of tweets, shares, likes, and spotting expressions and sentiments that convey urgency or negativity.

The simplest traditional process of text mining is Text preprocessing, Text Transformation (attribute generation) , Feature Selection (attribute selection), Data Mining and Evaluation. In the next sections we will study them one by one.

Check Your Progress 1:

1) Define structured, un-structured and semi-structured data with some examples for each.

.....

2) Differentiate between Text Mining and Text Analytics.

.....

12.3 TEXT PREPROCESSING

Text preprocessing is an approach for cleaning and preparing text data for use in a specific context. Developers use it in almost all natural language processing (NLP) pipelines, including voice recognition software, search engine lookup, and machine learning model training. It is an essential step because text data can vary. From its format (website, text message, voice recognition) to the people who create the text (language, dialect), there are plenty of things that can introduce noise into your data.

The ultimate goal of cleaning and preparing text data is to reduce the text to only the words that you need for your NLP goals.

Noise Removal: Text cleaning is a technique that developers use in a variety of domains. Depending on the goal of your project and where you get your data from, you may want to remove unwanted information, such as:

- Punctuation and accents
- Special characters
- Numeric digits
- Leading, ending, and vertical whitespace
- HTML formatting

The type of noise that you need to remove from text usually depends on its source.

Stages such as stemming, lemmatization, and text normalization make the vocabulary size more manageable and transform the text into a more standard form across a variety of documents acquired from different sources.

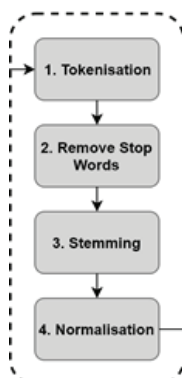


Figure 1: Text Preprocessing

Once you have a clear idea of the type of application you are developing and the source and nature of text data, you can decide on which preprocessing stages can be added to your NLP pipeline. Most of the NLP toolkits on the market include options for all of the preprocessing stages discussed above.

An NLP pipeline for document classification might include steps such as sentence segmentation, word tokenization, lowercasing, stemming or lemmatization, stop word removal, spelling correction and Normalization as shown in Fig 1. Some or all of these commonly used text preprocessing stages are used in typical NLP systems, although the order can vary depending on the application.

a) Segmentation

Segmentation involves breaking up text into corresponding sentences. While this may seem like a trivial task, it has a few challenges. For example, in the English language, a period normally indicates the end of a sentence, but many abbreviations, including “Inc.,” “Calif.,” “Mr.,” and “Ms.,” and all fractional numbers contain periods and introduce uncertainty unless the end-of-sentence rules accommodate those exceptions.

b) Tokenization

For many natural language processing tasks, we need access to each word in a string. To access each word, we first have to break the text into smaller components. The method for breaking text into smaller components is called tokenization and the individual components are called tokens as shown in Fig 2.

A few common operations that require tokenization include:

- Finding how many words or sentences appear in text
- Determining how many times a specific word or phrase exists
- Accounting for which terms are likely to co-occur

While tokens are usually individual words or terms, they can also be sentences or other size pieces of text.

Many NLP toolkits allow users to input multiple criteria based on which word boundaries are determined. For example, you can use a whitespace or punctuation to determine if one word has ended and the next one has started. Again, in some instances, these rules might fail. For example, don’t, it’s, etc. are words themselves that contain punctuation marks and have to be dealt with separately.

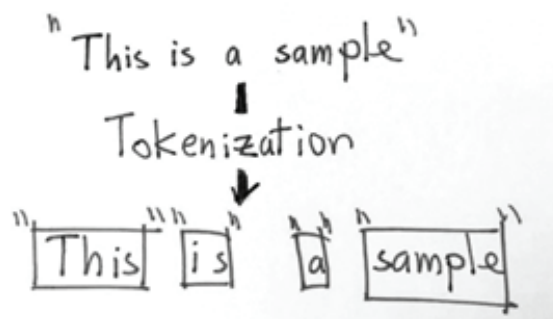


Figure 2; Tokenization

c) Normalization

Tokenization and noise removal are staples of almost all text pre-processing pipelines.

However, some data may require further processing through text normalization. Text normalization is a catch-all term for various text pre-processing tasks. In the next few exercises, we'll cover a few of them:

- Upper or lowercasing
- Stopword removal
- Stemming – bluntly removing prefixes and suffixes from a word
- Lemmatization – replacing a single-word token with its root

Change Case

Changing the case involves converting all text to lowercase or uppercase so that all word strings follow a consistent format. Lowercasing is the more frequent choice in NLP software.

Spell Correction

Many NLP applications include a step to correct the spelling of all words in the text.

Stop-Words Removal

“Stop words” are frequently occurring words used to construct sentences. In the English language, stop words include is, the, are, of, in, and and. For some NLP applications, such as document categorization, sentiment analysis, and spam filtering, these words are redundant, and so are removed at the preprocessing stage. See the Table 1 below given the sample text with stop words and without stop words.

Table 1: Sample Text with Stop Words and without Stop Words

Sample Text with Stop Words	Without Stop Words
TextMining – A technique of data mining for analysis of web data	TextMining, technique, datamining, analysis, web, data
The movie was awesome	Movie, awesome
The product quality is bad	Product, quality, bad

Stemming

The term word stem is borrowed from linguistics and used to refer to the base or root form of a word. For example, learn is a base word for its variants such as learn, learns, learning, and learned.

Stemming is the process of converting all words to their base form, or stem. Normally, a lookup table is used to find the word and its corresponding stem. Many search engines apply stemming for retrieving documents that match user queries. Stemming is also used at the preprocessing stage for applications such as emotion identification and text classification. An example is given in the Fig 3.



Figure 3: Example of Stemming

Lemmatization

Lemmatization is a more advanced form of stemming and involves converting all words to their corresponding root form, called “lemma.” While stemming reduces all words to their stem via a lookup table, it does not employ any knowledge of the parts of speech or the context of the word. This means stemming can’t distinguish which meaning of the word right is intended in the sentences “Please turn right at the next light” and “She is always right.”

The stemmer would stem right to right in both sentences; the lemmatizer would treat right differently based upon its usage in the two phrases.

A lemmatizer also converts different word forms or inflections to a standard form. For example, it would convert less to little, wrote to write, slept to sleep, etc.

A lemmatizer works with more rules of the language and contextual information than does a stemmer. It also relies on a dictionary to look up matching words. Because of that, it requires more processing power and time than a stemmer to generate output. For these reasons, some NLP applications only use a stemmer and not a lemmatizer. In the below given Fig 4, difference between lemmatization and stemming is illustrated.



Figure 4: Illustration of Lemmatization and Stemming

Parts of Speech Tagging

One of the more advanced text preprocessing techniques is parts of speech (POS) tagging. This step augments the input text with additional information about the sentence’s grammatical structure. Each word is, therefore, inserted into one of the predefined categories such as a noun, verb, adjective, etc. This step is also sometimes referred to as grammatical tagging.

12.4 TEXT TRANSFORMATION USING BoW AND TF-IDF

We understand that sentence in a fraction of a second. But machines simply cannot process text data in raw form. They need us to break down the text into a numerical format that’s easily readable by the machine. This is where the concepts of Bag-of-Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF) come into play. Both BoW and TF-IDF are techniques that help us convert text sentences into numeric vectors.

For example, there are sample of reviews of a movie so, the reviews of the viewers can be:

- Review 1: This movie is very scary and long
- Review 2: This movie is not scary and is slow
- Review 3: This movie is spooky and good

You can easily observe three different opinions of three different viewers. You can

see thousands of reviews about a movie on the internet. All these users generated text can help us out to takeout some interpretation in gauging that how a movie has performed. The above three reviews mentioned above cannot be given to the machine learning engine to analyze positive or negative reviews. So, we apply some text filtering techniques like Bag of words.

12.4.1 Bag of words (BoW)

It is the kind of a model in which the text is written in the form of numbers. It can be represented as represent a sentence as a bag of words vector (a string of numbers).

The Bag of Words (BoW) model is the simplest form of text representation in numbers. Like the term itself, we can represent a sentence as a bag of words vector (a string of numbers).

Consider once again the 3 movie reviews:

- Review 1: This movie is very scary and long
- Review 2: This movie is not scary and is slow
- Review 3: This movie is spooky and good

We will first build a vocabulary from all the unique words in the above three reviews. The vocabulary consists of these 11 words: ‘This’, ‘movie’, ‘is’, ‘very’, ‘scary’, ‘and’, ‘long’, ‘not’, ‘slow’, ‘spooky’, ‘good’.

We can now take each of these words and mark their occurrence in the three movie reviews above with 1s and 0s. This will give us 3 vectors for 3 reviews as shown in the Table 2 below:

Table 2: Vector Representation for the Reviews

	1 This	2 movie	3 is	4 very	5 scary	6 and	7 long	8 not	9 slow	10 spooky	11 good	Length of the Review (in words)
Review 1	1	1	1	1	1	1	1	0	0	0	0	7
Review 2	1	1	2	0	1	1	0	1	1	0	0	8
Review 3	1	1	1	0	0	1	0	0	0	1	1	6

Vector of Review 1: [1 1 1 1 1 1 1 0 0 0 0]

Vector of Review 2: [1 1 2 0 0 1 1 0 1 0 0]

Vector of Review 3: [1 1 1 0 0 1 0 0 0 1 1]

And that’s the core idea behind a Bag of Words (BoW) model.

Drawbacks of using a BoW

In the above example, we can have vectors of length 11. However, we start facing issues when we come across new sentences:

- If the new sentences contain new words, then our vocabulary size would increase and thereby, the length of the vectors would increase too.
- Additionally, the vectors would also contain many 0s, thereby resulting in a sparse matrix (which is what we would like to avoid)

- We are retaining no information on the grammar of the sentences nor on the ordering of the words in the text.

12.4.2 Vector Space Modeling for Representing Text Documents

The fundamental idea of a vector space model for text is to treat each distinct term as its own dimension. So, let's say you have a document D , of length M words, so we say w_i is the i th word in D , where $i \in [1 \dots M]$. Furthermore, the set of words contained in w_i form a set called the vocabulary or, more evocatively, the term space, often denoted V .

Here's an example:

Let our actual document D be: "He is neither a friend nor is he a foe"

Then $M=10$, and $w_3="neither"$. Our term space consists of all distinct terms in D :
 $V=\{"He","is","neither","a","friend","nor","foe"\}$

Now, lets impose an (arbitrary) ordering on V , so that that we form a basis V of terms. In this basis, v_i refers to the i th term in the vocabulary (i.e. we convert the Python "set" V to a Python "sequence" V). Think $V = \text{list}(V)$

$V:=["He","is","neither","a","friend","nor","foe"]$

What we have done is define a basis for a vector space. In this example, we have defined a 7-dimensional vector space, where each term v_i represents an orthogonal axis in a coordinate system much like the traditional x,y,z axes.

With this space, we now have a convenient way of describing documents: Each document can be represented as a 7-dimensional vector (n_1, \dots, n_7) where n_i is the number of times term v_i occurs in D (also called the "term frequency"). In our example, we would represent D by projecting it onto our basis V , resulting in the following vector:

$$D||B = (2,2,1,2,1,1,1)$$

This representation forms the core of most text mining methods. For example, you can measure similarity between two documents as the cosine of the angle between their associated vectors. There are many more uses of this method for encoding documents (e.g., see TF-IDF as a refinement of the basic vector space model which is given below).

12.4.3 Term Frequency-Inverse Document Frequency (TF-IDF)

Term frequency-inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.

Term Frequency (TF)

Let's first understand Term Frequent (TF). It is a measure of how frequently a term, t , appears in a document, d :

$$tf_{t,d} = \frac{n_{t,d}}{\text{Number of terms in the document}}$$

Here, in the numerator, n is the number of times the term "t" appears in the document "d". Thus, each document and term would have its own TF value.

Consider the 3 reviews as shown below:

- Review 1: This movie is very scary and long
- Review 2: This movie is not scary and is slow
- Review 3: This movie is spooky and good

We will again use the same vocabulary we had built in the Bag-of-Words model to show how to calculate the TF for Review #2:

Review 2: This movie is not scary and is slow

Here,

- Vocabulary: ‘This’, ‘movie’, ‘is’, ‘very’, ‘scary’, ‘and’, ‘long’, ‘not’, ‘slow’, ‘spooky’, ‘good’
- Number of words in Review 2 = 8
- TF for the word ‘this’ = (number of times ‘this’ appears in review 2) / (number of terms in review 2) = 1/8

Similarly,

$$\text{TF}(\text{'movie'}) = 1/8$$

$$\text{TF}(\text{'is'}) = 2/8 = 1/4$$

$$\text{TF}(\text{'very'}) = 0/8 = 0$$

$$\text{TF}(\text{'scary'}) = 1/8$$

$$\text{TF}(\text{'and'}) = 1/8$$

$$\text{TF}(\text{'long'}) = 0/8 = 0$$

$$\text{TF}(\text{'not'}) = 1/8$$

$$\text{TF}(\text{'slow'}) = 1/8$$

$$\text{TF}(\text{'spooky'}) = 0/8 = 0$$

$$\text{TF}(\text{'good'}) = 0/8 = 0$$

We can calculate the term frequencies for all the terms and all the reviews in this manner:

Term	Review 1	Review 2	Review 3	TF (Review 1)	TF (Review 2)	TF (Review 3)
This	1	1	1	1/7	1/8	1/6
movie	1	1	1	1/7	1/8	1/6
is	1	2	1	1/7	1/4	1/6
very	1	0	0	1/7	0	0
scary	1	1	0	1/7	1/8	0
and	1	1	1	1/7	1/8	1/6
long	1	0	0	1/7	0	0
not	0	1	0	0	1/8	0
slow	0	1	0	0	1/8	0
spooky	0	0	1	0	0	1/6
good	0	0	1	0	0	1/6

Inverse Document Frequency (IDF)

IDF is a measure of how important a term is. We need the IDF value because computing just the TF alone is not sufficient to understand the importance of words:

We can calculate the IDF values for the all the words in Review 2:

$IDF('this') = \log(\text{number of documents}/\text{number of documents containing the word 'this'}) = \log(3/3) = \log(1) = 0$

Similarly,

$$IDF('movie',) = \log(3/3) = 0$$

$$IDF('is') = \log(3/3) = 0$$

$$IDF('not') = \log(3/1) = \log(3) = 0.48$$

$$IDF('scary') = \log(3/2) = 0.18$$

$$IDF('and') = \log(3/3) = 0$$

$$IDF('slow') = \log(3/1) = 0.48$$

We can calculate the IDF values for each word like this. Thus, the IDF values for the entire vocabulary would be:

Hence, we see that words like “is”, “this”, “and”, etc., are reduced to 0 and have little importance; while words like “scary”, “long”, “good”, etc. are words with more importance and thus have a higher value.

We can now compute the TF-IDF score for each word in the corpus. Words with a higher score are more important, and those with a lower score are less important:

We can now calculate the TF-IDF score for every word in Review 2:

$$TF-IDF('this', Review 2) = TF('this', Review 2) * IDF('this') = 1/8 * 0 = 0$$

Similarly,

$$TF-IDF('movie', Review 2) = 1/8 * 0 = 0$$

$$TF-IDF('is', Review 2) = 1/4 * 0 = 0$$

$$TF-IDF('not', Review 2) = 1/8 * 0.48 = 0.06$$

$$TF-IDF('scary', Review 2) = 1/8 * 0.18 = 0.023$$

$$TF-IDF('and', Review 2) = 1/8 * 0 = 0$$

$$TF-IDF('slow', Review 2) = 1/8 * 0.48 = 0.06$$

Similarly, we can calculate the TF-IDF scores for all the words with respect to all the reviews:

Term	Review 1	Review 2	Review 3	IDF	TF-IDF (Review 1)	TF-IDF (Review 2)	TF-IDF (Review 3)
This	1	1	1	0.00	0.000	0.000	0.000
movie	1	1	1	0.00	0.000	0.000	0.000
is	1	2	1	0.00	0.000	0.000	0.000
very	1	0	0	0.48	0.068	0.000	0.000
scary	1	1	0	0.18	0.025	0.022	0.000
and	1	1	1	0.00	0.000	0.000	0.000
long	1	0	0	0.48	0.068	0.000	0.000
not	0	1	0	0.48	0.000	0.060	0.000
slow	0	1	0	0.48	0.000	0.060	0.000
spooky	0	0	1	0.48	0.000	0.000	0.080
good	0	0	1	0.48	0.000	0.000	0.080

We have now obtained the TF-IDF scores for our vocabulary. TF-IDF also gives larger values for less frequent words and is high when both IDF and TF values are high i.e the word is rare in all the documents combined but frequent in a single document.

12.5 DIMENSIONALITY REDUCTION

The number of input features, variables, or columns present in a given dataset is known as dimensionality, and the process to reduce these features is called dimensionality reduction.

Dimensionality reduction is the process of reducing the number of random variables or attributes under consideration. High-dimensionality data reduction, as part of a data pre-processing-step, is extremely important in many real-world applications. High-dimensionality reduction has emerged as one of the significant tasks in data mining applications. For an example you may have a dataset with hundreds of features (columns in your database). Then dimensionality reduction is that you reduce those features of attributes of data by combining or merging them in such a way that it will not lose much of the significant characteristics of the original dataset. One of the major problems that occur with high dimensional data is widely known as the “Curse of Dimensionality”. This pushes us to reduce the dimensions of our data if we want to use them for analysis.

Curse of Dimensionality

Handling the high-dimensional data is very difficult in practice, commonly known as the curse of dimensionality. If the dimensionality of the input dataset increases, any machine learning algorithm and model becomes more complex. As the number of features increases, the number of samples also gets increased proportionally, and the chance of overfitting also increases. If the machine learning model is trained on high-dimensional data, it becomes overfitted and results in poor performance.

Hence, it is often required to reduce the number of features, which can be done with dimensionality reduction.

Some benefits of applying dimensionality reduction technique to the given dataset are given below:

- By reducing the dimensions of the features, the space required to store the dataset also gets reduced.

- Less Computation training time is required for reduced dimensions of features.
- Reduced dimensions of features of the dataset help in visualizing the data quickly.
- It removes the redundant features (if present) by taking care of multi-collinearity.

12.5.1 Techniques for Dimensionality Reduction

Dimensionality reduction is accomplished based on either *feature selection or feature extraction*.

Feature selection is based on omitting those features from the available measurements which do not contribute to class separability. In other words, redundant and irrelevant features are ignored.

Feature extraction, on the other hand, considers the whole information content and maps the useful information content into a lower dimensional feature space.

One can differentiate the techniques used for dimensionality reduction as linear techniques and non-linear techniques as well. But here those techniques will be described based on the feature selection and feature extraction standpoint.

As a stand-alone task, feature selection can be unsupervised (e.g. Variance Thresholds) or supervised (e.g. Genetic Algorithms). You can also combine multiple methods if needed.

12.5.1.1 Feature Selection Techniques

a) Variance Thresholds

This technique looks for the variance from one observation to another of a given feature and then if the variance is not different in each observation according to the given threshold, feature that is responsible for that observation is removed. Features that don't change much don't add much effective information. Using variance thresholds is an easy and relatively safe way to reduce dimensionality at the start of your modeling process. But this alone will not be sufficient if you want to reduce the dimensions as it's highly subjective and you need to tune the variance threshold manually. This kind of feature selection can be implemented using both Python and R.

b) Correlation Thresholds

Here the features are taken into account and checked whether those features are correlated to each other closely. If they are, the overall effect to the final output of both of the features would be similar even to the result we get when we used one of those features. Which one should you remove? Well, you'd first calculate all pair-wise correlations. Then, if the correlation between a pair of features is above a given threshold, you'd remove the one that has larger mean absolute correlation with other features. Like the previous technique, this is also based on intuition and hence the burden of tuning the thresholds in such a way that the useful information will not be neglected, will fall upon the user. Because of those reasons, algorithms with built-in feature selection or algorithms like PCA(Principal Component Analysis) are preferred over this one.

c) Genetic Algorithms

They are search algorithms that are inspired by evolutionary biology and natural selection, combining mutation and cross-over to efficiently traverse large solution spaces. Genetic Algorithms are used to find an optimal binary vector, where each bit is associated with a feature. If the bit of this vector equals 1, then the feature is allowed to participate in classification. If the bit is a 0, then the corresponding feature does not participate. In feature selection, “genes” represent individual features and the “organism” represents a candidate set of features. Each organism in the “population” is graded on a fitness score such as model performance on a hold-out set. The fittest organisms survive and reproduce, repeating until the population converges on a solution some generations later.

d) Stepwise Regression

In statistics, stepwise regression is a method of fitting regression models in which the choice of predictive variables is carried out by an automatic procedure. In each step, a variable is considered for addition to or subtraction from the set of explanatory variables based on some pre-specified criterion. Usually, this takes the form of a sequence of F-tests or T-Tests but other techniques are possible such as adjusted R², Akaike information criterion, Bayesian Information criterion etc..

This has two types: forward and backward. For forward stepwise search, you start without any features. Then, you’d train a 1-feature model using each of your candidate features and keep the version with the best performance. You’d continue adding features, one at a time, until your performance improvements stall. Backward stepwise search is the same process, just reversed: start with all features in your model and then remove one at a time until performance starts to drop substantially.

This is a greedy algorithm and commonly has a lower performance than the supervised methods such as regularizations etc.

12.5.1.2 Feature Extraction Techniques

Feature extraction is for creating a new, smaller set of features that still captures most of the useful information. This can come as supervised (e.g. LDA) and unsupervised (e.g. PCA) methods.

a) Linear Discriminant Analysis (LDA)

LDA uses the information from multiple features to create a new axis and projects the data on to the new axis in such a way as to minimize the variance and maximize the distance between the means of the classes. LDA is a supervised method that can only be used with labeled data. It consists of statistical properties of your data, calculated for each class. For a single input variable (x) this is the mean and the variance of the variable for each class. For multiple variables, this is the same properties calculated over the multivariate Gaussian, namely the means and the covariance matrix. The LDA transformation is also dependent on scale, so you should normalize your dataset first. LDA is a supervised, so it needs labeled data..

b) Principal Component Analysis (PCA)

PCA is a dimensionality reduction that identifies important relationships in our data, transforms the existing data based on these relationships, and then quantifies the importance of these relationships so we can keep the most important relationships. To remember this definition, we can break it down into four steps:

1. We identify the relationship among features through a Covariance Matrix.
2. Through the linear transformation or eigen-decomposition of the Covariance Matrix, we get eigenvectors and eigenvalues.
3. Then we transform our data using eigenvectors into principal components.
4. Lastly, we quantify the importance of these relationships using Eigenvalues and keep the important principal components.

The new features that are created by PCA are orthogonal, which means that they are uncorrelated. Furthermore, they are ranked in order of their “explained variance.” The first principal component (PC1) explains the most variance in your dataset, PC2 explains the second-most variance, and so on. you can reduce dimensionality by limiting the number of principal components to keep based on cumulative explained variance. The PCA transformation is also dependent on scale, so you should normalize your dataset first. PCA is a find linear correlations between the features given. This means that only if you have some of the variables in your dataset that are linearly correlated, this will be helpful.

c) **t-distributed Stochastic Neighbor Embedding (t-SNE)**

t-SNE is non-linear dimensionality reduction technique which is typically used to visualize high dimensional datasets. Some of the main applications of t-SNE are Natural Language Processing (NLP), speech processing, etc.

t-SNE works by minimizing the divergence between a distribution constituted by the pairwise probability similarities of the input features in the original high dimensional space and its equivalent in the reduced low dimensional space. t-SNE makes then use of the Kullback-Leiber (KL) divergence in order to measure the dissimilarity of the two different distributions. The KL divergence is then minimized using gradient descent.

Here the lower dimensional space is modeled using t distribution while the higher dimensional space is modeled using Gaussian distribution.

d) **Autoencoders**

Autoencoders are a family of Machine Learning algorithms which can be used as a dimensionality reduction technique. Autoencoders also use non-linear transformations to project data from a high dimension to a lower one. Autoencoders are neural networks that are trained to reconstruct their original inputs. Basically autoencoders consist with two parts.

1. **Encoder:** takes the input data and compress it, so that to remove all the possible noise and unhelpful information. The output of the Encoder stage is usually called bottleneck or latent-space.
2. **Decoder:** takes as input the encoded latent space and tries to reproduce the original Autoencoder input using just it’s compressed form (the encoded latent space).

More on these techniques, you can read from MCS-224 Artificial Intelligence and Machine Learning course.

12.6 WEB MINING

Web mining as the name suggests that it involves the mining of web data. The extraction of information from websites uses data mining techniques. It is an application based on data mining techniques. The parameters generally to be mined in web pages are hyperlinks, text or content of web pages, linked user activity between web pages of the same website or among different websites. All user activities are stored in a web server log file. Web Mining can be referred as discovering interesting and useful information from Web content and usage.

12.6.1 Features of Web Mining

Following are some of the essential features of Web Mining:

- Web search, e.g. Google, Yahoo, MSN, Ask, Froogle (comparison shopping), job ads (Flipdog)
- The web mining is not like relation, it has text content and linkage structure.
- On the www the user generated data is increasing rapidly. So, Google's usage logs are very huge in size. Data generated per day on google can be compared with the largest data warehouse unit.
- Web mining can react in real-time with dynamic patterns generated on the web. In this no direct human interaction is involved.
- Web Server: It maintains the entry of web log pages in the log file. This web log entries helps to identify the loyal or potential customers from ecommerce website or companies.
- Web page is considered as a graph like structure, where pages are considered as nodes, hyperlinks as edges.
 - o Pages = nodes, hyperlinks = edges
 - o Ignore content
 - o Directed graph
- High linkage
 - o 8-10 links/page on average
 - o Power-law degree distribution

12.6.2 Web Mining Tasks

Web Mining performs various tasks such as:

- 1) Generating patterns existing in some websites, like customer buying behavior or navigation of web sites.
- 2) The web mining helps to retrieve faster results of the queries or the search text posted on the search engines like Google, Yahoo etc.
- 3) The ability to classify web documents according to the search performed on the ecommerce websites helps to increase businesses and transactions.

12.6.3 Applications of Web Mining

Some of the Applications of Web Mining are as follows:

- Personalized customer experience in Business to Consumer (B2C)
- Web Search
- Web-wide tracking (tracking an individual across all sites he visits, is an intriguing and controversial technology)
- Understanding Web Communities
- Understanding Auction Behaviour
- Personalized portal for the web.
- Recommendations: e.g. Netflix, Amazon
- improving conversion rate: next best product to offer
- Advertising, e.g. Google AdSense
- Fraud detection
- Improving Web site design and performance

12.7 TYPES OF WEB MINING

There are three types of web mining as shown in the following Fig 5.

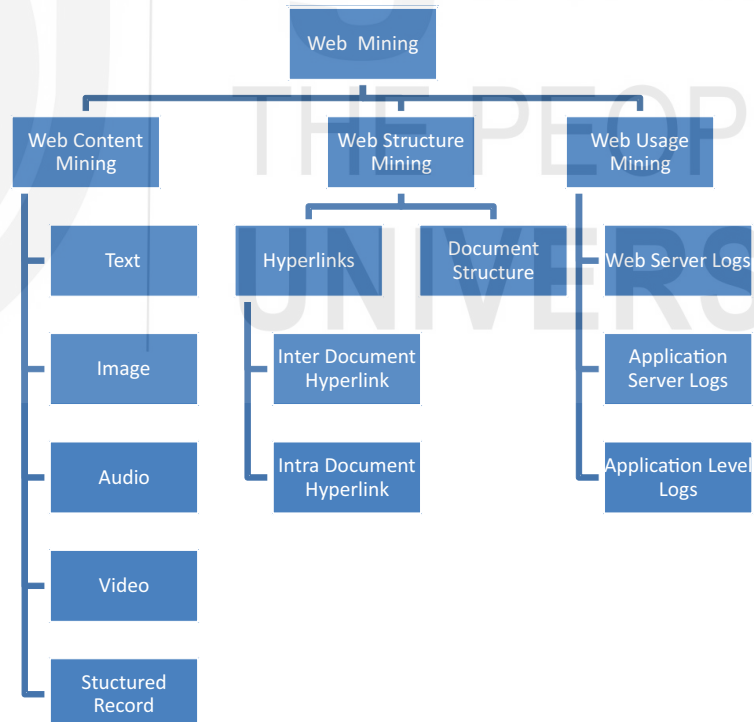


Figure 5: Three types of Web Mining

12.7.1 Web Content Mining

Web content mining is the process of extracting useful information from the contents of web documents. Content data is the collection of facts a web page is designed to contain. It may consist of text, images, audio, video, or structured records such as lists and tables. Application of text mining to web content has been

the most widely researched. Issues addressed in text mining include topic discovery and tracking, extracting association patterns, clustering of web documents and classification of web pages. Research activities on this topic have drawn heavily on techniques developed in other disciplines such as Information Retrieval (IR) and Natural Language Processing (NLP). While there exists a significant body of work in extracting knowledge from images in the fields of image processing and computer vision, the application of these techniques to web content mining has been limited.

12.7.2 Web Structure Mining

The structure of a typical web graph consists of web pages as nodes, and hyperlinks as edges connecting related pages. Web structure mining is the process of discovering structure information from the web. This can be further divided into two kinds based on the kind of structure information used.

Hyperlinks

A hyperlink is a structural unit that connects a location in a web page to a different location, either within the same web page or on a different web page. A hyperlink that connects to a different part of the same page is called an intra-document hyperlink, and a hyperlink that connects two different pages is called an inter-document hyperlink.

Document Structure

In addition, the content within a Web page can also be organized in a tree-structured format, based on the various HTML and XML tags within the page. Mining efforts here have focused on automatically extracting document object model (DOM) structures out of documents

12.7.3 Web Usage Mining

Web usage mining is the application of data mining techniques to discover interesting usage patterns from web usage data, in order to understand and better serve the needs of web-based applications. Usage data captures the identity or origin of web users along with their browsing behavior at a web site. Web usage mining itself can be classified further depending on the kind of usage data considered:

Web Server Data

User logs are collected by the web server and typically include IP address, page reference and access time.

Application Server Data

Commercial application servers such as Weblogic, StoryServer have significant features to enable E-commerce applications to be built on top of them with little effort. A key feature is the ability to track various kinds of business events and log them in application server logs.

Application Level Data

New kinds of events can be defined in an application, and logging can be turned on for them — generating histories of these events. It must be noted, however, that many end applications require a combination of one or more of the techniques applied in the above the categories.

12.8 MINING MULTIMEDIA DATA ON THE WEB

The websites are flooded with the multimedia data like, video, audio, images, and graphs. This multimedia data has different characteristics. The videos, images, audio, and pictures have different methods of archiving and retrieving the information. The multimedia data on the web has different properties this is the reason the typical multimedia data mining techniques cannot be applied. This web-based multimedia has texts and links. The text and links are the important features of the multimedia data to organize web pages. The better organization of web pages helps in effective search operation. The web page layout mining can be applied to segregate the web pages into the set of multimedia semantic blocks from non-multimedia web pages. There are few web-based mining terminologies and algorithms to understand.

PageRank: This measure is used to count the number of pages the webpage is connected to other websites. It gives the importance of the webpage. The Google search engine uses the algorithm PageRank and rank the web page very significant if it is frequently connected with the other webpages on the social network. It works on the concept of probability distribution representing the likelihood that a person on random click would reach to any page. It is assumed the equal distribution in the beginning of the computational process. This measure works on iterations. Iterating or repetition of page ranking process would help rank the web page closely reflecting to its true value.

HITS: This measure is used to rate the webpage. It was developed by Jon Kleinberg. It uses hubs and authorities to be determined from a web page. Hubs and Authorities define a recursive relationship between web pages.

- This algorithm helps in web link structure and speeds up the search operation of a web page. Given a query to a Search Engine, the set of highly relevant web pages are called Roots. They are potential Authorities.
- Pages that are not very relevant but point to pages in the Root are called Hubs. Thus, an Authority is a page that many hubs link to whereas a Hub is a page that links to many authorities.

Page Layout Analysis: It extracts and maintains the page-to-block, block-to-page relationships from link structure of web pages.

Vision page segmentation (VIPS) algorithm: It first extracts all the suitable blocks from the HTML Document Object Model (DOM) tree, and then it finds the separators between these blocks. Here separators denote the horizontal or vertical lines in a Web page that visually cross with no blocks. Based on these separators, the semantic tree of the Web page is constructed. A Web page can be represented as a set of blocks (leaf nodes of the semantic tree). Compared with DOM-based methods, the segments obtained by VIPS are more semantically aggregated. Noisy information, such as navigation, advertisement, and decoration can be easily removed because these elements are often placed in certain positions on a page. Contents with different topics are distinguished as separate blocks.

You can understand simply by considering following points:

- The web page contains links and links contained in different semantic blocks point to pages of different topics.

- Calculate the significance of web page using algorithms PageRank or HITS.

Split pages into semantic blocks

Apply link analysis on semantic block level. For example, in the below Fig 6, it is clearly shown. We can see the links in different blocks point to the pages with different topics. In this example, one link points to a page about entertainment and another link points to a page about sports.



Figure 6: Example of a sample web page (new.yahoo.com), showing web page with different semantic blocks (red, green, and brown rectangular boxes). Every block has different importance in the web page. The links in different blocks points to the pages with different topics.

To analyze the web page containing multimedia data there is a technique known as Link analysis. It uses two most significant algorithms PageRank and HITS to analyze the significance of web pages. This technique uses each page as a single node in the web graph. But since, web page with multimedia has lot of data and links. So, cannot be considered as a single node in the graph. So, in this case the web page is partitioned into blocks using vision page segmentation also called VIPS algorithm. So, now after extracting all the required information the semantic graph can be developed over world wide web in which each node represents a semantic topic or semantic structure of the web page.

VIPS algorithm helps in determining the text for web pages. This is the closely related text that provides content or text description of web pages and used to build image index. The web image search can then be performed using any traditional search technique. Google, Yahoo still uses this approach to search web image page.

Block-level Link Analysis: The block-to-block model is quite useful for web image retrieval and web page categorization. It uses kinds of relationships, i.e., block-to-page and page-to-block. Let's see some definitions. Let P denote the set of all the web pages,

$P = \{p_1, p_2, \dots, p_k\}$, where k is the number of web pages.

Let B denote the set of all the blocks,

$B = \{b_1, b_2, \dots, b_n\}$, where n is the number of blocks.

It is important to note that, for each block there is only one page that contains that block. $b_i \in p_j$ means the block i is contained in the page j .

Block-Based Link Structure Analysis: This can be explained using matrix notations. Consider Z is the block-to-page matrix with dimension $n \times k$. Z can be formally defined as follows:

$$Z_{ij} = \begin{cases} 1/s_i & \text{if there is a link from block } i \text{ to page } j \\ 0 & \text{otherwise} \end{cases}$$

where s_i is the number of pages that block i links to. Z_{ij} can also be viewed as a probability of jumping from block i to page j .

The block-to-page relationship gives a more accurate and robust representation of the link structures of the web unlikely, HITS as at times it deviates from the web text information. It is used to organize the web image pages. The image graph deduced can be used to achieve high-quality web image clustering results. The web page graph for web image can be constructed by considering measuring which tells the relationship between blocks and images, block-to-image, image-to-block, page-to-block and block-to-pages.

12.9 AUTOMATIC CLASSIFICATION OF WEB DOCUMENTS

The categorization of web pages into the respective subjects or domains is called classification of web documents. For example, in the following Fig 7, it has shown various categories like, books, electronics etc. let's say you are doing online shopping on the Amazon website and there are so many webpages so when you search for electronics the respective web page containing the information of electronics is displayed. This is the classification of products which is done on the textual and image contents.

Categories

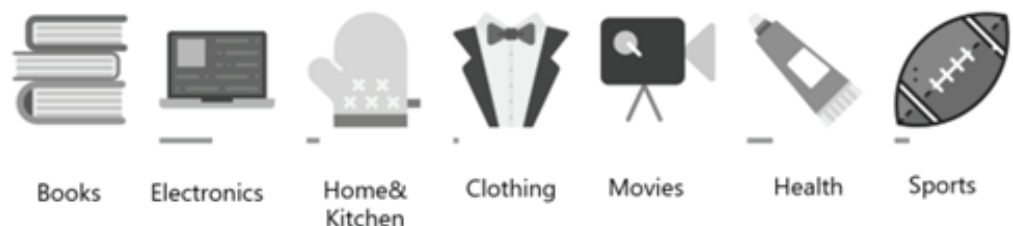


Figure 7: Types of Web Documents Containing Different Types of Data

The problem with the classification of web documents is that every time the model is to be constructed by applying some algorithms to classify the document is mammoth task. The large number of unorganized web pages may have redundant documents.

The automated document classification of web pages is based on the textual content. The model requires initial training phase of document classifiers for each category based on training examples.

In the Fig 8 it is shown that the documents can be collected from different sources. After the collection of documents data cleansing is performed using extraction transformation and loading techniques. The documents can be grouped according to the similarity measure (grouping of the documents according to the similarity between the documents) and TF-IDF. The machine learning model is created and executed, and different clusters are generated.

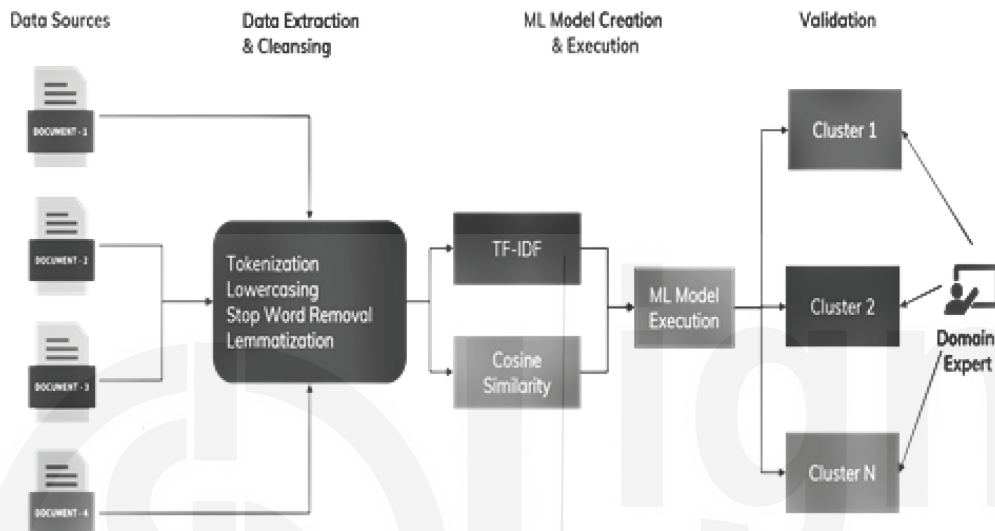


Figure 8: Automatic Classification of Web documents

Automated document classification identifies the documents and groups the relevant documents without any external efforts. There are various tools available in the market like RapidMiner, Azure, Machine Learning Studio, Amazon Sage maker, KNIME and Python. The trained model automatically reads the data from documents (PDF, DOC, PPT) and classifies the data according to the category of the document. This trained model is already trained with the Machine Learning and Natural Language Processing techniques. There are domain experts who perform this task efficiently.

Benefits of Automatic Document Classification System

- 1) It is more efficient system of classification as produces improved accuracy of results and speed up the process of classification.
- 2) The system incurs in less operational costs
- 3) Easy data store and retrieval.
- 4) It organizes the files and documents in a better streamlined way.

Check Your Progress 3

- 1) What are the techniques to analyze the web usage pattern?

.....

.....

.....

- 2) What are the other applications of Web Mining which were not mentioned?
.....
.....
.....
- 3) What are the differences between Block HITS and HITS?
.....
.....
.....
- 4) List some challenges in Web Mining.
.....
.....
.....

12.10 SUMMARY

In this unit we had studied the important concepts of Text Mining and Web Mining.

Text mining, also referred to as text analysis, is the process of obtaining meaningful information from large collections of unstructured data. By automatically identifying patterns, topics, and relevant keywords, text mining uncovers relevant insights that can help you answer specific questions. Text mining makes it possible to detect trends and patterns in data that can help businesses support their decision-making processes. Embracing a data-driven strategy allows companies to understand their customers' problems, needs, and expectations, detect product issues, conduct market research, and identify the reasons for customer churn, among many other things.

Web mining is the application of data mining techniques to extract knowledge from web data, including web documents, hyperlinks between documents, usage logs of web sites, etc..

12.11 SOLUTIONS/ANSWERS

Check Your Progress 1:

- 1) **Structured data:** This data is standardized into a tabular format with numerous rows and columns, making it easier to store and process for analysis and machine learning algorithms. Structured data can include inputs such as names, addresses, and phone numbers.

Unstructured data: This data does not have a predefined data format. It can include text from sources, like social media or product reviews, or rich media formats like, video and audio files.

Semi-structured data: As the name suggests, this data is a blend between structured and unstructured data formats. While it has some organization, it doesn't have enough structure to meet the requirements of a relational database. Examples of semi-structured data include XML, JSON and HTML files.

- 2) The terms, text mining and text analytics, are largely synonymous in meaning in conversation, but they can have a more nuanced meaning. Text mining and text analysis identifies textual patterns and trends within

unstructured data through the use of machine learning, statistics, and linguistics. By transforming the data into a more structured format through text mining and text analysis, more quantitative insights can be found through text analytics. Data visualization techniques can then be harnessed to communicate findings to wider audiences.

Check Your Progress 2:

- 1) Techniques used to analyze the web usage patterns are as follows:
 - Session and web page visitor analysis: The web log file contains the record of users visiting web pages, frequency of visit, days, and the duration for how long the user stays on the web page.
 - OLAP (Online Analytical Processing): OLAP can be performed on different parts of log related data in a certain interval of time.
 - Web Structure Mining: It produces the structural summary of the web pages. It identifies the web page and indirect or direct link of that page with others. It helps the companies to identify the commercial link of business websites.
- 2) Applications of Web Mining are:
 - Digital Marketing
 - Data analysis on website and application accomplishment.
 - User behavior analysis
 - Advertising and campaign accomplishment analysis.
- 3) The main difference between BLHITS (Block HITS) and HITS are:

BLHITS	HITS
Links are from blocks to pages	Links from pages to pages
Root is top ranked blocks	Root is top ranked pages
Analyses only top ranked block links	Analyses all the links of all the pages
Content analysis at block level	Content analysis at page level

- 4) Challenges in Web Mining are:
 - The web page link structure is quite complex to analyze as the web page is linked with many more other web pages. There exists lot of documents in the digital library of the web. The data in this library is not organized.
 - The web data is uploaded on the web pages dynamically on regular basis.
 - Diversity of client networks having different interests, backgrounds, and usage purposes. The network is growing rapidly.
 - Another challenge is to extract the relevant data for subject or domain or user.

12.12 FURTHER READINGS

1. Mining The Web: Discovering Knowledge From Hypertext Data, Chakrabarti Soumen, Elsevier Science, 2014.
2. Data Mining, Charu C. Aggarwal, Springer, 2015.
3. Data Mining: Concepts and Techniques, 3rd Edition, Jiawei Han, Micheline Kamber, Jian Pei, Elsevier, 2012.

