An Effective Preprocessing Methodology for Textual Data Classification

Dr. Muhammad Shahbaz [1],Dr. Syed Muhammad Ahsen[2], Maryam Shaheen[3], Muhammad Shaheen[4]
, Syed Athar Masood[5]
[1,2,3, 4] Department of Computer Science & Engg, UET Lahore, Pakistan
[5] Department of Engineering Management, NUST College of E&ME, Rawalpindi Pakistan
[1] M.Shahbaz@uet.edu.pk, [2] Ahsencs@hotmail.com, [3]shaheen@uet.edu.pk, [4]maryam.shaheen@gmail.com, [5] atharmasood2000@hotmail.com

Abstract: In the present rapidly changing world, a massive amount of raw data is generated, collected and organized in databases. This data contains lot of useful and important information which is hidden and not directly accessible. There is vital need for in-depth data analysis tools which can turn raw data into knowledge. This situation is known as "data rich but information poor". It is very time consuming, slow and expensive to analyze and understand the huge volume of data manually specially when the data is in the form of text. Textual data need huge resources to preprocess it to make it ready for the data/text mining algorithms. In this paper we have proposed an effect preprocessing methodology for textual data which have produce quality data efficiently and reliably. [Dr. Muhammad Shahbaz, Dr. Syed Muhammad Ahsan, Maryam Shaheen, Muhammad Shaheen, Syed Athar Masood. An Effective Preprocessing Methodology for Textual Data Classification. Journal of American Science 2011;7(6):944-951]. (ISSN: 1545-1003). http://www.americanscience.org.

Keywords: Text Mining, Data Mining, Classification, Knowledge Discovery, Parsing, P-Tree

Introduction

The word data mining has many other synonyms like knowledge extraction, knowledge mining, data archaeology, data dredging and data pattern processing. Data mining is given the name from gold mining. As in gold mining people dig rocks to find gold, similarly analysts use data mining tools to find precious nuggets (knowledge) from raw data.

Data mining is the main step in the KDD process that consists of applying data analysis and discovery algorithms that, under acceptable computational efficiency imitations, produce a particular enumeration of patterns (or models) over the data **[2]**.

The data mining phase of the KDD process basically relies on the algorithms that are useful in extracting and analyzing the patterns. Most of these methods are applied and extensively being used. Data mining step is iterative in nature as the techniques used are tried and tested many times until goal is achieved. The data mining algorithms belong to the machine learning, pattern recognition, statistics and artificial intelligence fields. Data mining algorithms consists of three components **[1]**:

- **Model Representation**

Mined patterns are represented by using some modeling language. The language should be selected very carefully so that correct model of data is formed otherwise it results in poor accuracy.

- **Model Evaluation**

Special quantitative measures (criteria) are defined to test a particular pattern against the KDD goal. It shows how well a model meets its defined criteria.

- **Search**

Its main functionality is performing the optimization task of evaluation criteria.

Data Mining Methods

Data mining tasks can be divided into two groups by the kinds of pattern they discover: descriptive and predictive. Descriptive mining tasks deals with common properties of the data which are human understandable while predictive mining tasks involve the prediction of future values by making inference on current values of data.

Summarization and Visualization

Summarization methods give summaries of data. These methods describe subset of data in compact form like average, standard deviation, means etc. Visualization techniques provide graphical overview of data by which analysts can perceive data more easily that will be more helpful in data analysis. Examples include pie chart, bar chart, histograms, and multidimensional tables. More information can be studied through graphical form of data than to study texts and numerical form **[2]**.

Classification

Classification is a way to divide data into groups which have already defined classes. Classification process is carried out in two steps: learning and classification. In learning phase, model is built by using pre-defined set of classes on data. Subset of data selected for learning purpose is called training data which is randomly selected during classification process. The learned model is then presented in the form of classification rules, decision trees, or mathematical formulae. In the classification phase, the model which presented in some suitable form is then tested against some test data to estimate its accuracy. If accuracy is good enough then this model can be used for classification of future coming data whose class label is not known.

Regression

The process to model and analyze continuous values by using statistical methods is called regression. This technique is based on one independent variable and one or more dependent variables. Independent variable is that which is used to make prediction and dependent variable is the one which is to be predicted. Existing values are used to make prediction for future ones. The simplest form of regression is the linear regression in which data uses straight line model. CART (Classification and regression Trees) algorithm is mainly used for this purpose.

Clustering

Clustering is also a descriptive task which divides the data into groups which have no class information associated with them so that objects in the same clusters are more similar to each other than to the objects in other clusters. It is a form unsupervised learning in which classes is not defined at the start. Most common algorithms used in clustering are K-means and Kohonen feature maps. One major problem with clustering is to define the number of clusters in the beginning which can affect the accuracy.

Link Analysis

Link analysis helps in discovering the relationships and associations between the data values of the given dataset and represents these in the form of rules. These rules represent those values which frequently occur in the data set under consideration. It is also known as association rule mining. Link analysis technique is also descriptive in nature. Market-basket analysis is the most commonly used example of association analysis.

Association rules identified are written in the form A=>B, where A is called antecedent and B is known as consequent. The most commonly used algorithms to find the association rules are Apriori Algorithm and FP-growth.

Outlier Detection

Outlier analysis or change detection tries to find those data values which show different behavior from the previous measured values. These are known as outliers. Some algorithms discard these as noise but in some cases where unusual happening events are of great importance, these are not thrown away. It can help in detecting the misuse of credit cards or telecommunication service. Visualization tools help a lot in detecting the outliers as human eye is very efficient in noticing the unusual behavior or data inconsistencies, but this process remain effective with lower dimensionality as higher dimensions are difficult to perceive.

Text Mining

Textual data consists of raw data or free text which is easily understandable by human beings. The text words are in printed format or written material which can be displayed on a display screen. It is written in any of the natural languages which human beings use to communicate with each other. This kind of data is not readable and understandable by computers or machines. To make this unstructured data usable by computers, natural language algorithms are being applied on it to make it suitable for further processing.

Characteristics of Text Data

- **Large Textual Database**
  In this "Information Age", most of the textual data of companies like reports, emails, web pages etc. is in electronic format. Most of the publications, books, related material are available and easily accessible through the internet.
- **High Dimensionality**
  The textual data is considered to be as of sparse in nature. Each keyword is regarded as a separate dimension. This makes the data very complex to understand.
- **Unstructured Nature**
  The data is in unstructured form which is not readable and understandable by machine. No specific data type and data structure is defined, the words of the text are considered as words only. It's been said that 85% data of the companies is in unstructured form **[5]**.
- **Noisy Data**
  The textual databases are noisy in nature as these contain a lot of grammatical and spelling errors. There is difference between the original text representation and short-hand words used like in SMS, chat, and e-mail **[6]**.

Text Mining

Text mining is a step in the KDT (Knowledge Discovery in Text) process consisting of particular data mining and NLP algorithms that under some acceptable computational efficiency limitations produces a particular enumeration of patterns over a set of unstructured textual data **[3]**.

KDT is same like KDD (knowledge Discovery in Databases) discussed in last chapter except that KDT deals with only textual databases. Similarly, text mining differs from data mining in the case that text mining deals with only unstructured text while data mining is applied on structured databases often numerical in nature. KDT indicates the overall process of conversion of unstructured text into knowledge while text mining is the small step in the whole process of KDT which has the main purpose of extracting the useful patterns from the free text.

Text mining uses data mining, information retrieval and NLP techniques and algorithms to extract useful information hidden in the text. All these areas combined together according to the problem to form a text mining pipeline.

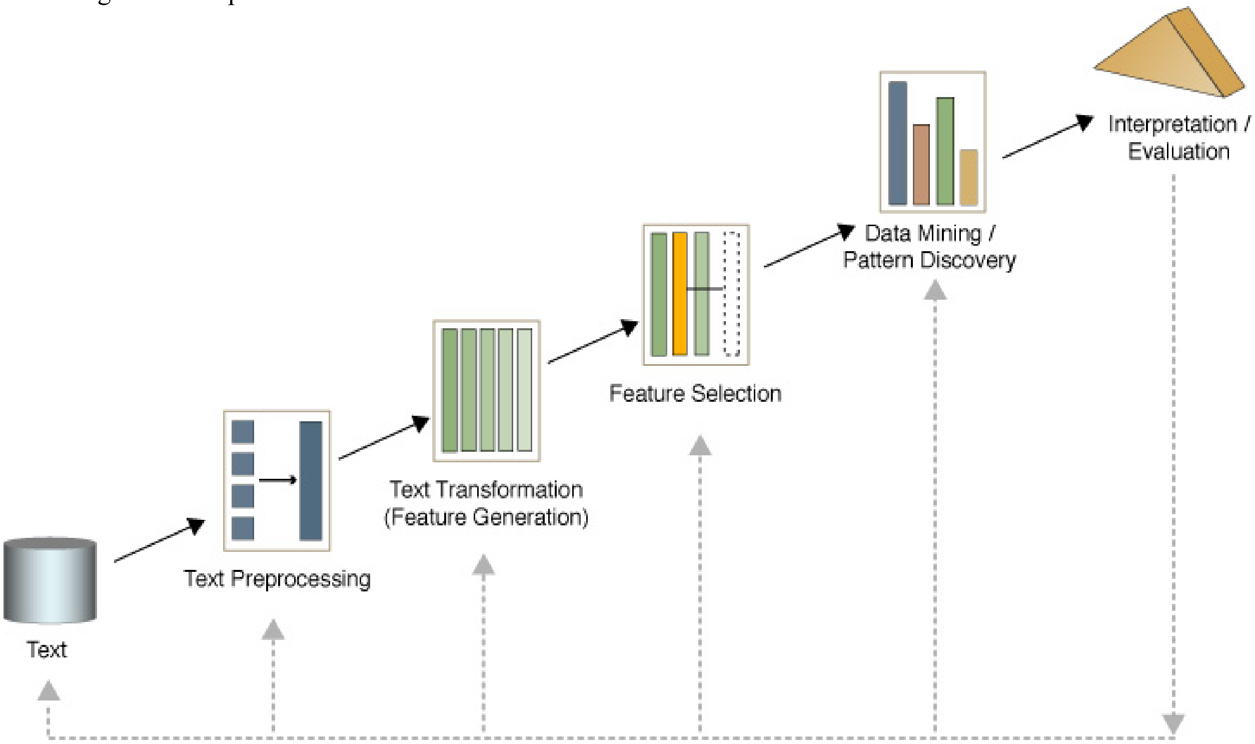


Figure 1. Text Mining Process **[4]**

According to Marti Hearst, "Text mining is the discovery by the computer of new, previously unknown information by automatically extracting information from different written resources" **[7].** The most important thing is the bonding of the mined information together to form new piece of information or knowledge which can be investigated more by using the same usual algorithmic methods.

The limitations of text mining are: firstly, it is very difficult to write a program which can infer text data for a very long time, because with a passage of time new things are being added in the vocabulary. Secondly, sometimes novel information which has to be extracted from the text is not in mentioned in the textual data **[7]**.

Text Mining Process

A text mining process is a multistep procedure which is shown in figure 1. It is an iterative method where the output of one stage becomes the input of the stage following it. The process which completes a text mining process consists of following stages **[4]**:

Text Preprocessing

The first step of text mining process is text preprocessing in which the document collection is analyzed syntactically or semantically. The meaning and grammatical structure of the language in which document collection is written is analyzed and recognized. The algorithms being applied here Part Of Speech tagging, Word Sense Disambiguation and Parsing.

Text Transformation

Text transformation deals with the feature generation part of the text analysis. The text document is considered as bag of words because the words and its occurrences are used to represent the document. The algorithms applied at this stage are stemming and stop word removal. Sometimes feature generation task is also included in the text preprocessing step of the text data mining.

Feature Selection

This stage consists of specialized techniques to select the fewer terms which can best represent the text collection. It will help out in reducing the large dimensions to small number. Feature selection parameters can be of like information gain, chi square, mutual ratio etc. Only selected features are then kept to represent the document collection, the remaining ones are discarded.

Text/Data Mining

This is the part of process where the actual patterns are to be revealed from the textual data. At this stage the actual process of text mining or data mining is being applied on unstructured data. The interesting patterns are extracted from the textual data to convert it into valuable knowledge.

Result Evaluation

At the end the resulting features are then evaluated to check the accuracy of the applied algorithms. The knowledge is then also used to predict the future values of the data.

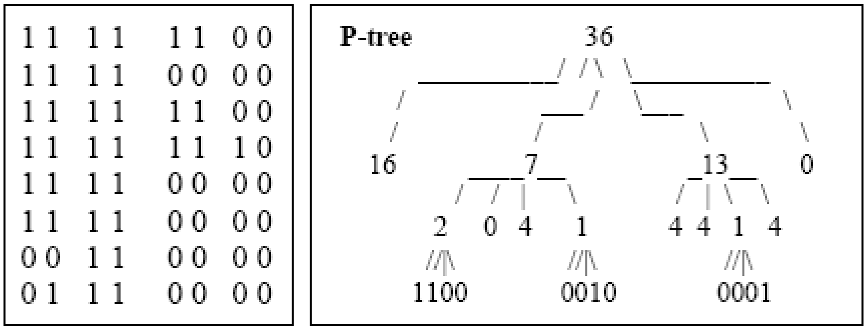
Peano-Count Trees Data Structure

P-Trees basically use vertical partitioning where data is divided into columns per bit rather than column per attribute. If data stored in the columns is in particular sequence, following benefits can be achieved through this technology.

- Compression
- Hardware Optimization (AND operation)
- Efficient index implementation
- Only necessary data in memory

Basic P-trees

P-Trees resemble old data structures like Quad-Trees and HHcodes in the sense that these organize the data into quadrants. The idea behind this is, to divide the image into four quadrants recursively until single bits are left at leaf nodes and then count the number of ones in each quadrant, leads to the formation of P-trees. Each bit file of bSQ format is then converted into P-trees by following the recursive raster order. An 8x8 bSQ file and its P-tree is shown in the Figure 2

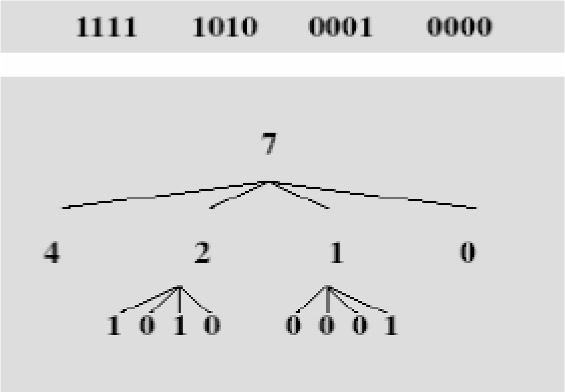


1 1	1 1	1 1	0 0
1 1	1 1	0 0	0 0
1 1	1 1	1 1	0 0
1 1	1 1	1 1	1 0
1 1	1 1	0 0	0 0
1 1	1 1	0 0	0 0
0 0	1 1	0 0	0 0
0 1	1 1	0 0	0 0

Figure 2. 8x8 bSQ File and it's P-tree **[8]**

In the example shown above, 36 is the root count (number of one's) of whole image. This is labeled as level-0. 16, 7, 13 and 0 are the root counts of the level-1 quadrants traversed in Peano-order. 16 represent the number of ones in the first quadrant, 7 represent the number of ones in second quadrant, 13 is the root count of third quadrant while the fourth quadrant shows that there all zeros in it, number of ones are zero here. The first quadrant is composed of only 1-bit so it is known as pure-1 quadrant. Similarly if the quadrant is composed entirely of 0-bits then it is called pure-0 quadrant. As these are pure quadrants, there is no need for further branches. At last, each branch terminates i.e. leaf sequences are reached which are also pure as these consist of only one bit.

Considering the spatial data, basic P-trees are constructed for each bit position in each band. For n bands 8n basic P-trees are constructed assuming 8-bit values in bands. These basic P-trees are labeled as $P_{i,1}$, $P_{i,2}$, $P_{i,3}$, … ,$P_{i,8}$ for all 8 bits for ith band. The basic P-trees are considered as "data-mining ready" and "lossless format" **[8]** for storing spatial data as it provide more information about data.



1111 1010 0001 0000

Figure 3. A 16-bit bSQ file converted into P-tree **[9]**

Value P-trees

A value P-tree ($P_{b,v}$) is basically the representation of value v at band b in the form of P-tree. Value v can be of any precision from 1-bit to 8-bit, so P-trees formed have value of more than one bit. Value P-trees can be constructed by combining the basic P-trees using the logical operations. For example, $P_{b,101}$ can be constructed by combining the basic P-trees as:

$P_{b,101} = P_{b,1}$ AND $P'_{b,0}$ AND $P_{b,1}$

Where $P_{b,101}$ gives the number of one's of band b bit 1 equal to 1, bit 2 equal to 0 and bit 3 equal to 1. Here AND operation do the pixel wise ANDing of the bits.

Tuple P-trees

The value P-trees for any combination, are then combined to form tuple P-tree. It is constructed for any value combination (v_1, v_2 ,…, v_n) of band I, shown as:

P (v1, v2, …, vn) = $P_{1,v1}$ AND $P_{2,v2}$ AND … AND $P_{n,vn}$

It gives quadrant wise count of occurrences.

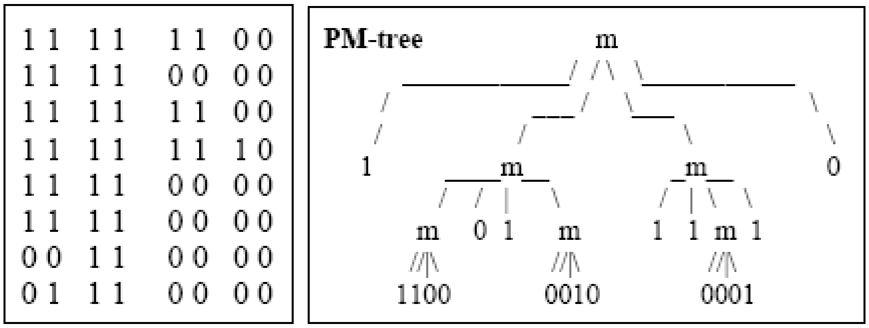
Interval P-trees

The P-Trees created by the values lying in specific intervals [v1,v2] of band i are known as interval P-trees. These are shown as follows:

P (v_1, v_2) = OR P (v), for all v in [v_1, v_2]

P-Tree Variations

Peano Mask Tree (PM-tree) is a variation of P-Tree in which masks are considered rather than the counts by using the 3-value logic for representation of pure-1, pure-0 and mixed quadrants. Pure-1 is represented by 1, pure-0 by 0 and mixed quadrant is denoted by m.



```
11 11 11 00
11 11 00 00
11 11 11 00
11 11 11 10
11 11 00 00
11 11 00 00
00 11 00 00
01 11 00 00
```

Figure 4. PM-Tree **[8]**

P1-tree uses 1 to represent pure-1 quadrant while 0 is used to represent pure-0 quadrant.

In P0-tree, 1 is used to represent pure-0 quadrant and 0 represents the other quadrants.

In non-pure-0-tree, 1 represents non-pure-0 quadrant which indicates pure-1 quadrant and mixed quadrant and 0 to represent pure-0 quadrant.Non-pure-1-tree is the vice versa of non-pure-0-tree.

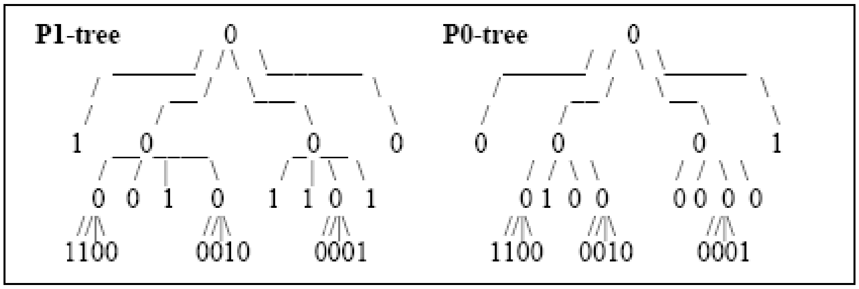


Figure 5. P1-tree and P0-tree **[8]**

Operations of P-Trees

AND

AND is the basic operation performed on the bits represented in the P-Tree form. It is the most widely used operation to check the equality of bit sequences with constant. AND operation can be performed in different ways but most important ones are level wise ANDing and Pure-1 path ANDing.

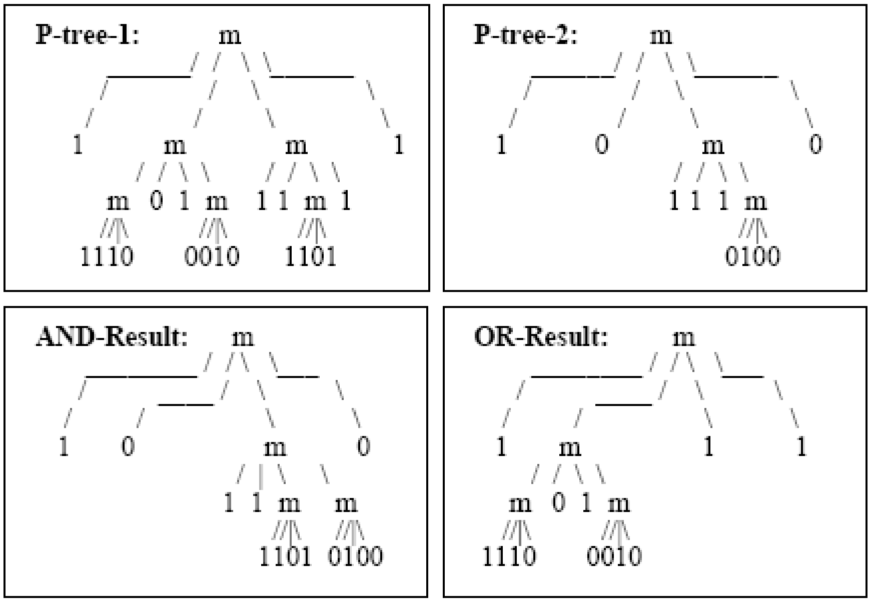


Figure 6. P-Tree AND and OR Operations **[8]**

In level-wise ANDing **[8]**, rules are being specified to perform the AND operation. Operands are given in the P-tree form with mentioned roots. These rules can be of the form: ANDing of pure-1 P-tree with any P-tree give result in the form of P-tree showing second operand. Pure-0 P-tree will be generated when a pure-0 P-tree is ANDed with any P-tree.

In pure-1 path ANDing process **[8]**, only basic P-trees are stored then the value and tuple P-trees are generated when these are needed. Here assumption is that P-trees use the depth first scheme for the paths to the pure-1 quadrant. Quadrants are identified by hierarchical scheme, that an id number is being attached to each quadrant at each level. These numbers are 0 for upper-left quadrant, 1 for upper-right quadrant, 2 for lower-left quadrant and 3 for lower-right quadrant. The same quadrant numbers are being tagged on at each level.

For an 8x8 image, the pixel at (3,6) has values (011, 110), so quadarant id is calculated as by combining two bits such as (01.11.10)=1.3.2. The quadrant can also be written as 132 instead of 1.3.2 for easiness as shown in the figure 6.

In this algorithm, the paths to the pure-1 quadrant are considered to be in the depth-first sequence. In this scheme, the paths are being represented in the form of quadrants which obey Peano-ordering. Peano Quadrant is the process of identifying the quadrant id as shown above. So ANDing is done by scanning the operands and then output the pure-1 paths which are matched.

In the example shown in the below figure, two mask P-trees for two operands are given. The ANDing result of these two P-trees is shown in a form of P-tree. Considering the depth-first order the first value is 0 which means it is 0 quadrants. Second quadrant is mixed, so it will first retrieve the leaf node of this quadrant which has one value such as: 100, 101 and 102. Similarly, it will identify the other quadrants and then AND the values.

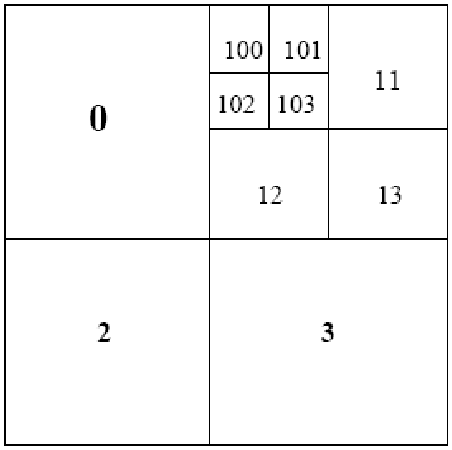


Figure 13. Quadrant Numbers **[8]**

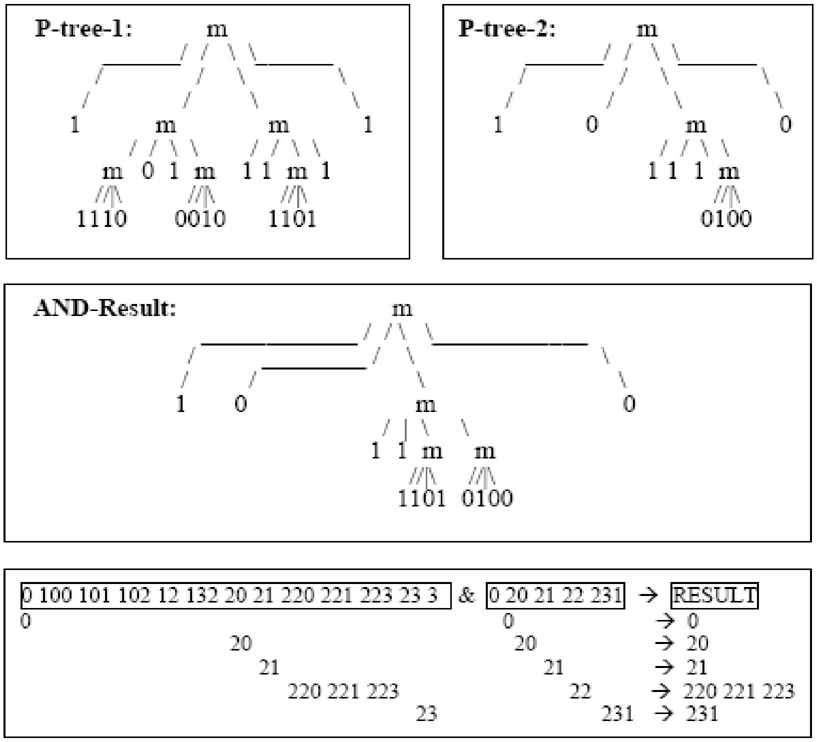


Figure 8. ANDing of PM-trees using Pure-1 Path **[8]**

OR

OR operation can be implemented in the similar way as the AND operation is being implemented. Most of the times it is implemented as ANDing the first operand with the complement of second one.

Complement

Every P-tree shows the natural property of complement. The complement process is very simple; it just takes basic P-tree and then complements the counts at each level (this is done by subtracting the counts from pure-1 counts at that level). Complement gives the number of zeros for each quadrant in a P-tree.

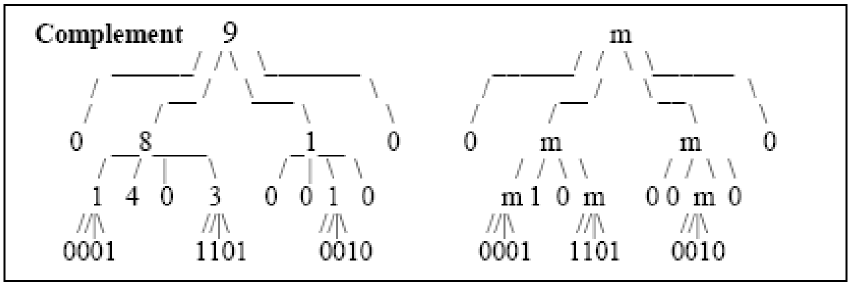


Figure 9. Complement Operation of P-Tree **[8]**

Count

Count operation basically counts the number of ones in each quadrant for a given P-tree in Peano order. This helps in identifying the number of bit sequences that can satisfy a particular condition. Counting algorithm can be implemented as a shift operation on integer taking one bit at a time OR it can be implemented by counting the occurrences of one's directly from the look up table taking multiple bits at a time.

Proposed Solution

In this chapter, a solution is being given to the problem that document should be represented in some structured form so that data mining algorithms can be directly applied to it. This proposed methodology discussed below, converts the document into ready form after applying some necessary steps.

Preprocessing step:

After removing the stop words (most common words like the, a, an, the, of, or and so on) and applying stemming (truncating the words to their common root such as "printing", "prints", "printed" to the root "print"), the important terms from each document in the collection are collected, this can best describe the documents in the whole collection.

Document Model:

Each document in the collection is now represented as a vector by using the vector space model. Terms are the dimensions and documents are the points in the space. This whole document-term collection is now represented as term-document matrix.

The most common used method to represent the relation between terms and documents is tf-idf weighting scheme, which combines the local and global weights together. It discriminates the term from the collection and gives importance to those terms which mostly occur in a document and less occurring in the whole document collection.

$$\mathrm{weight}_{ij} = \mathrm{freq}_{ij} * \log\left(\frac{N}{docf_i}\right)$$

Where N= total number of documents in collection
docf$_i$= number of documents in which term i occur
freq$_{ij}$= frequency of term i in document j

This is most widely used weighting scheme to discriminate the term from the document collection point of view.

Tf-ITF measurement is proposed for the term-document data weights, which gives importance to the terms occurring frequently in the document and document is more specific about the topic. It is a local weighting scheme only as it considers the document in consideration not the whole collection of documents. ITF measurement (already proposed) gives high weights to the documents which contain less terms i.e., these documents are more specific about the topics; if documents contain more terms then the document is generic in nature.

$$tf - ITF = tf \times \log\left(\frac{M}{doc_{ti}}\right)$$

Where M= total number of terms extracted from the document collection
doc $_{ti}$=number of terms in a particular document

This weighting scheme shows document richness (relative information about the terms in a document) information. When multiplied with term frequency it gives high weight to those terms which are frequent in the topic specific document. ITF is same for one document representing its length and richness. When term frequency factor is being added in it, then this weighting scheme shows the high weighting for the terms occurring mostly in the document scenario and low weights to the terms occurring less frequently.

Conclusion

This weighting scheme can be used for the shortening purpose of documents as only frequently occurring terms are being extracted from each document. These terms show the document main theme and topic. These frequently occurring terms or important ones are being used to represent the documents in a convenient and useful manner. These important terms in each document can represent some relationships between the documents which is helpful in further analysis.

This is more useful in knowledge extraction part of the whole mining process. As essential features of a text document are extracted by the pre-processing technique and are represented by the form a model. This model will further analyzed to mine the information being hidden in the data like relationships between the documents, importance of each term in particular document scenario, document main theme and so on. This will be done the future analysts.

References

[1] Usama Fayyad, Gregory Piatetsky-Shapiro and Padhraic Smyth, "From Data Mining to Knowledge Discovery in Databases", in American Association for Artificial Intelligence, 1996.

[2] "Introduction to Data Mining and Knowledge Discovery", Third Edition, by Two Crows Corporation, 2005.

[3] Haralampos Karanikas and Babis Theodoulidis, "Knowledge Discovery in Text and Text Mining Software", Technical Report, UMIST-CRIM, Manchester, 2002.

[4]algdocs.ncsa.uiuc.edu/PR-20021116-2.ppt, Accessed on July 8, 2008, 10:00 AM.

[5]http://en.wikipedia.org/wiki/Unstructured_data, Accessed on July 12, 2008, 8:00 PM.

[6]http://en.wikipedia.org/wiki/Noisy_text, Accessed on July 12, 2008, 8:00 PM.

[7] Marti Hearst, "What is Text Mining?" SIMS, UC Berkeley October 17, 2003,

[8] Qin Ding, Maleq Khan, Amalendu Roy and William Perrizo, "The P-tree Algebra", ACM SA2002.

[9] Thomas Rölleke, Theodora Tsikrika, Gabriella Kazai, "A general matrix framework for modelling
Information Retrieval", in Information Processing and Management, issue 42, 2006, pp 4-30.

3/18/2011