A New Algorithm to find best splitting criteria for Web page classification

V. Bharani Priya[1], V. Kamakshi Prasad[2],
[1](CSE dept,Suprabath engineering college,Ibrrahimpatnam, Hyderabad.)
[2](CSE dept,JNTUH)

Abstract: Due to vast availability of data on web, the challenges in retrieving the web data are also increasing. In our paper we discussed about web and how to mould unstructured web documents into structural format. Further we have developed a new algorithm for finding selection attribute to find best splitting criterion for classifying the data using Information Gain. Our algorithm is mainly learner's centric.

Keywords: Java, ID3, Web documents, Title, classification, Moulding.

1. Introduction

Web is popular as the availability of data is vast. It contains several billions of HTML documents, pictures and other multi media files. These documents reside on Internet servers and the information-exchange can be done through HTTP protocols. Individual HTML files having unique address is called Web page and collection of such web pages having same electronic address is called as Web site. The quality of the web page is decided by its title. It is the name of a web page or web site. So giving an appropriate and good title to a web page is very important. To retrieve the required web pages based on the title of the web page, the title of the web page has to be parsed (i.e.) the key words have to be extracted from the given titles. To retrieve the documents efficiently the unstructured web documents have to be moulded into structured format for classifying them using these keywords.

2. Moulding Web Documents

We represent the Web data in the binary format where all of the keywords derived from the schema (Title of the web page) positioned in the columns and some frequent schemas are posited in the rows. If a keyword is in a frequent schema, a 1 is stored in related cell and otherwise a 0 is stored in it. In the following this idea is demonstrated. The attributes of frequent schemas are stated as below.

QI1: Deep Web Content mining = {Deep, Web, content, mining}
QI2: Extracting structured data from Web pages = {Extracting, structured, data, Web, pages}
QI3: Page content Rank: An approach to the web content mining = {Page, content, rank, web, content, mining}
QI4: Web mining: Information and pattern discovery on the World Wide Web = {Mining, information, pattern, discovery, World Wide Web}

We represent the web data in binary format as in below table.

Table 1: A Sample Array with Input Information.

	Deep	Web	Content	Mining	Extracting	Other Keywords	Category
QI1	1	1	1	1	0	0	1
QI2	0	1	0	0	1	1	2
QI3	0	1	1	1	0	1	3
QI4	0	0	0	1	0	1	1

3. Algorithm for Attribute Selection

To attain knowledge out of retrieved data we classify the documents by converting the unstructured documents into structured format. For classification of web documents we have many methods like classification by decision tree induction, Bayesian classification, rule-based classification, support vector machine etc. During late 1970s and 1980s J.Ross Quinlan, a researcher in machine learning has developed a decision tree algorithm known as ID3 [1] (Iterative Dichotomiser). ID3 uses information gain for attribute selection. Information gain Gain(A) is given as

$$\text{Gain}(A) = \text{Info}(D) - \text{Info}_A(D)$$

We have developed a new algorithm to calculate information gain. Methodology wise this algorithm is promising. We have divided the algorithm into two parts. The first part calculates Info(D) and the second part calculates the Gain(A). The web documents which we represent in binary format, constitutes as input array. This algorithm is described as below.

/* First part of the algorithm */

Algorithm to find *Entropy*

```
Input: Set of class values
Output: Entropy of D.
Method:
//Let the input is stored in some array say a ←
{ai, $a_j$ . . . $a_n$}
Sort the input array, a.
 {$b_0$, $b_1$,…$b_m$} ←
frequency_count_of_each_element_of_array_a
 For i ← 0 to m
   Entropy ← Entropy + -($b_i$/n) * (log($b_i$/n)/0.301)
```

/* Second part of the algorithm*/

Algorithm to find *Gain(A)*

Input:

- Data partition D, which is a set of training tuples and their associated class values.
- Entropy value, to calculate information gain.

```
Output: Splitting criterion attribute.
Method:
// Let the data item values D be stored in a two
dimensional array say        a ← {{$a_{i,j}$ ,
$a_{i,j+1}$. . .}, $a_{i+1,j}$ , . . .} . . .$a_{row,col}$}
Declare an array b of length m
Declare an array c of length n
For i ← 0 to (col-1)
{
k←0, k1 ← 0
For j ← 0 to row
{
If ($a_{ji}$ ← 1)
        Array $b_k$ ← $a_{j, col}$
        k ← k+1
Else
        Array $c_{k1}$ ← $a_{j, col}$
        k1 ←k1+1
}
Array b1 ← frequency_count_of_each
element_of_array_b

For i1 ← 0 to ((SizeOfArray_b1) -1)
  int info1 ← info1+(-$b1_{i1}$/m-1) * ((log($b1_{i1}$/m-
1))/0.301)

info1 ← ((m-1) / row) * info1
Array c1 ← frequency_count
_of_each_element_of_array_c

For i2 ← 0 to ((SizeOfArray_c1)-1)
  int info2 ← info2 + (-($c1_{i2}$)/n -1) * (log($c1_{i2}$/n -
1))/0.301

int info2 ← ((n -1)/row) * info2
// Entropy = manual input taken from previous
algorithm.

Gain(A) ← Entropy –(info1 + info2)
Array info ← Gain(A).

}
Find maximum number from the array 'info', to
find the highest information gain among the
attributes.
/* The splitting attribute will reside at the index
equal to the index value of an array having
highest information gain*/
```

In the first algorithm we have calculated entropy. This value has been used in the equation info = entropy – (info1 + info2) of second algorithm. In the following section we have given flow chart of the algorithm. We have tested the algorithm on the following titles.

- Information retrieval with principal component
- Document clustering using locality preserving indexing
- Hierarchy-Regularized Latent Semantic Indexing
- Improving Text classification using local latent semantic indexing
- Genralising Discriminant analysis using the generalized singular value decomposition
- Enhanced hypertext categorization using hyperlinks
- owing the semantic gap-Improved Text-Based web document retrieval using visual features
- Extracting structured data from web pages
- From data mining to knowledge discovery in databases

- Minig knowledge from text using information extraction
- Page content Rank: An approach to the web content mining
- Web Hunting design of a simple Intelligent web search agent
- An introduction to Regression analysis
- Design and Development of " Biomass stove" for onfarm value of Rainfed crops
- Onfarm value addition of rainfed crops with CRIDA Herbal Dryer
- Retrieval Data from CSV files in JSP
- Web site minig: A new way to spot competitors, customers and suppliers in the world wide web.
- A fully automated object extaction system for the world wide web
- Web mining: Information and pattern discovery on the world wide web

After converting the above unstructured data into structural form, the data becomes our input. The output of the algorithm for our input is:

Entropy= 2.8156
max = 39

We got max = 39 which means that the splitting attribute will reside at the index 39 of the keyword array.

4. Conclusions

In our paper we discussed about web and how to mould web documents into a structured format. We have used decision tree for classification and we have given our own version of algorithm to find information gain. This algorithm can be further extended for generating a decision tree.

Reference

[1] Jiawei Han and Micheline Kamber, Data Mining Concepts and Techniques, 2nd edition, chapter no 6.