Classifying Web Sites Using Distance-Based Decision Trees

V. Estruch C. Ferri J. Hernández-Orallo
M.J. Ramírez-Quintana

DSIC, Universidad Politécnica de Valencia, Camino de Vera s/n, Apdo. 22012, 46071 Valencia, Spain. Email: {vestruch, cferri,jorallo,mramirez}@dsic.upv.es.

Abstract

In Web classification, web pages are assigned to pre-defined categories mainly according to their content (content mining). However, the structure of the web site might provide extra information about their category (structure mining). Traditionally, both approaches have been applied separately, or are dealt with techniques that do not generate a model, such as Bayesian techniques. Unfortunately, in some classification contexts, a comprehensible model becomes crucial. Thus, it would be interesting to apply rule-based techniques (rule learning, decision tree learning) for the web categorisation task. In this paper we outline how our general-purpose learning algorithm, the so called distance based decision tree learning algorithm (DBDT), could be used in web categorisation scenarios. This algorithm differs from traditional ones in the sense that the splitting criterion is defined by means of metric conditions ("is nearer than"). This change allows decision trees to handle structured attributes (lists, graphs, sets, etc.) along with the well-known nominal and numerical attributes. Generally speaking, these structured attributes will be employed to represent the content and the structure of the web-site.

Keywords: web mining, classification, structured data, decision trees, distance-based methods.

1 Introduction

Etzioni [4] defined Web mining as the use of data mining techniques for extract information from Web documents and services. Given the large amount of documents available in the Web, one of the most common task performed on the Web is the classification of documents into one or more categories. For

[1] This work has been partially supported by the EU (FEDER), the spanish MEC (TIN 2004-7943-C04-02), the ICT for EU-India Cross Cultural Dissemination Project ALA/95/23/2003/077-054, the Acción Integrada Hispano-Austríaca HU2003-0003 and the Generalitat Valenciana (MEDIM, GV04B/477).

instance, this is essential in applications that have to catalog news articles, sort and filter electronic mail, recommend films or music or search information about a topic (search engines). Although some authors distinguish classification from categorisation [2], for the sake of simplicity, in this paper we use both of them as synonyms since a categorisation problem can be solved by several classifiers. A direct approach to the classification of Web documents is to take only the textual part of them into account (Text categorisation). The basic idea is to classify a document as of class c if certain words relevant to the c definition are present in the document.

However, Web documents are more than just plain text; the information contained in other parts like the hyper-links can also be relevant to the categorisation process. For instance, if we are classifying sport news, a more accurate classification can be obtained if our classifier considers that a piece of sport news contains words such as *team*, *play* or *stadium*, or contains links to other sport news. Therefore, recent research solves this problem by merging ideas from Web content mining and Web structure mining. For instance, [7] appends the text of the links to the text of the target page. [1] considers the text of a Web page along with the text and the category of its neighbouring pages. Some other approaches are able to handle both the text components in the pages and the links among them, such as [2], [5], or [6].

In this paper, we study how the DBDT approach fits the web classification problem. Potentially, this method would allow us to integrate both the Web content and the Web structure mining in a unique framework by using structured attributes (lists, sets, etc.) to represent each component or context feature (title, keywords, text, links, ...) found in the pages and then, associate a metric function to each involved data type. This evidence is then used by the DBDT algorithm due to the splitting criterion is defined by means of metric conditions ("is nearer than") and in this way, the structured attributes can be handled. We illustrate how this method works by applying it to a simple example of Web classification and we briefly discuss about how the metric conditions could be expressed in an equivalent but more comprehensible form.

The paper is organised as follows. In Section 2 the DBDT algorithm is outlined. An illustrative example of our approach is shown in Section 3. Then, some experiments comparing our approach to other results reported in the literature can be found in Section 4. Finally, Section 5 presents some conclusions.

[2] The classification is the process of inducing a model in that only one class is assigned to each document, whereas categorisation concerns with the situation in that a document can belong to more than one class.

2 Distance Based Decision Trees

In [3] we defined a learning method named Distance Based Decision Trees (DBDT). This proposal is based on the use of *prototypes* and distances to define the partitions for the decision tree. Our decision tree inference strategy is a modification of the centre splitting method [8] consisting in the computation of a set of attribute prototypes. Unlike the centre splitting, we do not use all the attributes into account for each split. Basically, for each attribute and for each class, a prototype (that value which minimises the sum of all the distances from it to the rest) is calculated, considering only the values belonging to that attribute and that class. Once this process is finished, an attribute is chosen in order to split the data set. The split proceeds by associating every instance to its closest attribute prototype. The splitting attribute is selected according to some of the well-known heuristic functions (information gain, gain ratio [18], GINI index [19], etc.). For this purpose, a metric space is associated to every attribute. Note that the fact of handling all the attributes as a whole entity, just as centre splitting does, turns the comprehensible model extraction into a harder task, even if the involved attributes are nominal or numerical. The result of this adaptation of centre splitting is not very different from classical decision trees (see the algorithm below), when attributes are either nominal and numeric, but in our case, we are able to deal with data containing structured attributes such as sets, lists, or trees.

PROCEDURE DBDT(S, m); // Learns a decision tree based on attribute distances.
INPUT: A training set S as a set of examples of the form: $(x_1, \ldots, x_n), n \geq 1$ where every attribute is nominal, numerical or structured. A metric space is associated to every attribute. m is the maximum num. of children per node.
BEGIN
  $C \leftarrow \{Class(e) : e \in S\}$ // C is the set of existing classes
  If $|C| < 2$ **Then RETURN End If**
  For each attribute x_j:// Computes two (or more) centres for each class using x_j
    If $|Values(x_j, S)| < 2$ **Then CONTINUE End If** //next iteration
    $ProtList \leftarrow$ Compute_Prototypes(x_j, S, m, C).
    If $Size(ProtList) \leq 1$ **Then RETURN End If**
    $Split_j \leftarrow \emptyset$ // Set of possible splits for attribute x_j
    For i $\leftarrow$ 1 **to** $length(ProtList)$ // for all the prototypes
      $\hat{S}_i \leftarrow \{e \in S : i =$ Attracts$(e, ProtList, x_j)\}$ // $\hat{S}_i \equiv$ examples attracted by prot. i
      $Split_j = Split_j \cup \hat{S}_i$ // We add a new child to this split
    End For
  End For
  $BestSplit = Argmax_{Split_j}(Optimality(Split_j))$ // GainRatio, MDL, ...

```
  For each set S_k in BestSplit
    DBDT(S_k, m) // go on with each child
  End For
END
```

The auxiliary functions `Attracts` and `Compute_Prototypes` are inherent to the method. In a nutshell, the function `Attracts` just determines which prototype is assigned with a new example and finally, the function `Compute_Prototypes` obtains a set of prototypes for each attribute.

3 An illustrative example

The first step consists of deciding what information from the data set is going to be modelled and what data types are going to be used in order to model it, as well as their associated metric functions. Let us consider the following example. A user is interested in seeking sport news from the Internet using a search engine. This search engine must "decide" automatically which available documents fit the search parameters. Thus, this task can be addressed as a two class classification problem. The information, extracted from an HTML document for this purpose, can be grouped into these three categories:

- **Structure**: it refers to how the pages from a web site are connected by means of hyper-links. Formally, it is represented as a graph.However, we will use a simpler approach which is a very common proposal in the graph mining literature: we represent a graph as a set of ordered pairs where each pair encodes two linked pages. Concretely, each item in the ordered pair will store a set of key words. Also, for the sake of brevity, we use the well-known symmetric difference between sets as a metric function.
- **Content**: it deals with the information contained in a web page.Traditionally, this information is represented as a bag or a vector of words. In our example, we only consider one attribute, a set, reflecting the whole content (*Content*), and we use an adaptation of the symmetric difference between sets as a metric function.
- **Web use**: we mean by web use information the information derived from the HTTP connection to a web server. All these data is encoded by means of nominal or numerical attributes. For these types we can use the discrete metric or the absolute value difference, respectively. In our example, this attribute is referred by *Connections* and it contains the number of daily connections.

The next step is to infer a classifier by training a model from a processed dataset that contains collected information from some web pages, such as

that included in Table 1.

Id.	Structure	Content	Conn.	Class
1	{([Olympics,games],[swim]),([swim],[win]), ([Olympics,games],[boxing]) , ([win],[medal])}	{(Olympics,30),(held,10) (summer,40)}	10	No
2	{([Olympics,games],[swim]),([swim],[win]), ([win],[medal])}	{(Olympics,15),(summer,20) (Athens,40)}	20	Yes
3	{([football],[Europe]),([Europe],[final]), ([final],[best,player])}	{(football,20),(champion,10)}	40	No
4	{([football],[match]),([match],[team,players]), ([football],[referees]),([match],[results])}	{(football,20),(Europe,10), (champion,12)}	40	Yes
5	{([football],[match]),([match],[team,players]), ([match],[scores])}	{(football,20),(Europe,10)}	40	Yes

Table 1
Information from a web server sample repository.

The set {([Olympics,games],[swim]),([swim],[win]),([win],[medal])} in the *Structure* attribute is interpreted in the following way. The first component of the list stands for words "Olympics" and "games" appear as keywords in a web page. This page links another one which has "swim" as its only key word. The reasoning is the same for the second and third components of the set.

If we apply the DBDT algorithm (using an accuracy-based heuristic), we find that the first attribute to be selected, as the first split, is *Connection*, being the values 40 (*Conn* value for the 4*th* instance) and 10 (*Conn* value for the 1*st* instance) the prototypes for the class "yes" and "no" respectively. Iterating the process, attributes *Structure* and *Content* are used to split the left and the right first level nodes, respectively. Finally, the new obtained nodes are pure and the process stops, getting the distance based decision tree (see figure below [3]).

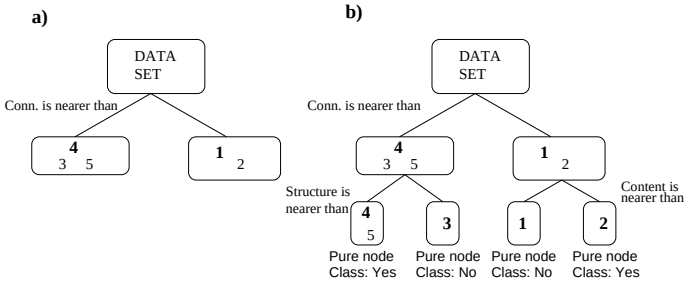

Fig. 1. a) Decision tree after the first split. b) Decision tree after finishing the process.

[3] The numbers correspond to instance id, and the bold numbers stand for the prototype of each class for a particular partition.

Id.	Structure	Content	Conn.	Class
	{([football],[match]),([match],[players]), ([match],[results])}	{(football,30),(held,10) (Europe,7)}	36	No

Table 2
Information from a web server sample repository.

Imagine now that a web site described as in Table 2 is stored in the list along with other web sites which are candidates to be shown to a possible user. Before listing them directly we should classify the web site repository in order to filter non suitable information. First, we look inside the connection attribute. As the number of daily connections is closer to 40 than 10, the instance is linked to the left first-level node. Then, we repeat the same process for the structure attribute, in this case, the structure of this new web site is more similar to the structure of the fourth instance in the table than to the third one. Then, this instance would be classified as a sport news site, and, consequently, listed to the user.

4 Experimental evaluation

In this section we show that our general-purpose learning algorithm can be used to address real web classification problems. The experimental evaluation is divided into two parts. First, DBDT is tested over our own web-data repository in order to check the feasibility of the method. Secondly, DBDT is run over a web repository extracted from UCI [4] and the obtained results are compared to those reported in the literature. For each experimental setting, 10×10 **cross-validation** has been performed.

4.1 Experimental and implementation remarks

The algorithm in Section 2 has been implemented in WEKA [10] [5]. As we stated in Section 3, running the current DBDT system requires a preliminary stage which deals with preprocessing the input data set in order to obtain the distance matrix associated to each attribute.

Several distance and pseudo-distance functions have been implemented. For nominal and numerical data, the employed distances are the well-known discrete and absolute value difference respectively. Regarding structured data types, this issue is not so straightforward. In fact, the definition of a proper distance function for some specific structured data types becomes a research

[4] http://kdd.ics.uci.edu/summary.data.application.html
[5] http://www.dsic.upv.es/users/elp/soft

topic on its own. Thus, these distance or pseudo-distance functions have been looked up in the specific literature as we remark below.

Focusing on the experiments, sets and lists have been the structured data which we have employed. Each document is represented by a finite set (sequence) of unique words, called summary, drawn from the title and the body section. Of course, *html* tags or script instructions are avoided during this process. Additionally, prepositions and adverbs are discarded as well. Therefore, an instance (document) is described by means of two attributes: the summary and the class label. As for the first model (set of words), the Hausdorff distance is applied for sets along with the alignment distance for words belonging to sets [13,14]. As for the second one, a (pseudo)-distance derived from a kernel function (subsequence kernel) has been utilised. This kernel is specially defined over sequences of data and successfully tested in text classification and DNA pattern recognition tasks [11]. The connection between distance and kernel function is explained in [15].

Each word, presented in the summary, is selected according to a significance measure which quantifies how important it is for classification. Several significance measures can be found in the literature [16]. All the experiments have been performed using an entropy-based measure.

The general setting for all experiments is as follows. We start with summaries containing the most 50 significant words, adding 25 words in each setting, up to summaries of 150 words length. Regarding to the parameters of the DBDT, one prototype per class is computed, being each class distribution data represented by its median, and the information gain heuristic is employed as splitting criterion.

4.2 Classifying web sites by topic

At this point, we tackle again the classification problem of *html* documents, which has been used to illustrate DBDT in section above, but in a more realistic way. For this purpose, we collected a total of 83 *html* documents downloaded directly from Internet. The documents are grouped into two different topics: mathematics (biographies, technical pages, personal web sites, lectures, etc.) and sports (biographies, news, events, championships, etc.). Thus, the goal is to learn a classifier which informs about the topic of a new unclassified document. Both models (set and list of words) have been studied. The obtained results are stored in the table below:

From the experimental results, two conclusions can be extracted. First, more accurate results are obtained by employing lists along with the so-mentioned pseudo-distance. A possible explanation of this is that Hausdorff

Num of words	List Acc. %.	Set Acc. %
50	100.0	93.6
75	98.1	91.5
100	95.9	91.4
125	98.4	94.8
150	97.6	92.5

Table 3
Accuracy achieved varying the number of words included in the summary, and the structured data type along with its associated distance.

distance is quite sensitive to outliers, and these strongly affect its performance. Secondly, according to the performance achieved, we can think that DBDT can be used for web mining purposes. In order to confirm that, DBDT is run over a public domain benchmark in the next subsection.

Before concluding, more experiments are presented. The aim is to study the influence of using more information than the summary. In this case, every document is represented not only by its summary (sequence of words), but also by two extra attributes: a list of keywords and the number of pictures appearing in the web page. The obtained results are as follows:

Num of words	Acc. (%).
50	100.0
75	99.7
100	100.0
125	100.0
150	91.4

Table 4
Accuracy achieved employing the summary, a list of keywords and the number of pictures appearing in the web page.

As we can appreciate, excepting for summaries of 150 words length, the accuracy is slightly increased by incorporating these two extra attributes.

4.3 Learning user profiles

The *Syskill & Webert* data set is a repository of web documents organised into several topics: clinical information (*Biomedical*), music events (*Bands*), biochemistry (*Proteins*), etc. Each document is ranked, according to the preferences of the user, in "hot", "medium" or "cold". A document labelled with "hot" stands for a high interesting web page, being uninteresting when

it is labelled with "cold". Therefore, learning a user profile leads to learning a classifier which labels unseen pages with the proper tag.

In order to compare the obtained results to those reported in [12], only "hot" and "cold" documents will be taken into account and these will be represented by only their respective summary. All the experimental process is focused on *Bands* and *Biomedical* repositories. In these experiments, the summary is modelled by a sequence of words using the pseudo-distance derived from the subsequence kernel (see Table below).

Num.words	*Bands* Acc. %	*Biomedical* Acc. %
50	74.5	71.5
75	79.7	81.3
100	77.9	83.0
125	**82.0**	**84.0**
150	81.7	79.6
200	82.1	82.0

Table 5
Accuracy achieved on *Bands* and *Biomedical* data sets varying the number or words included in the summary.

As [12] points out and as we can also observe from the results above, it seems that the optimal number of words is around 125. In contrast, fixing a low number or a high number of words can motivate that some really meaningful words are not included in the summary, or that noisy words take part of it. Keeping on the same work, a collection of learning techniques (ID3, Bayesian methods, neural networks, among others [17]) to address the proposed problem are compared. Although the evaluation method used here is slightly different to cross validation, the best accurate results on both data sets is performed by a Bayesian classifier (78.2% and 74.6% for *Biomedical* and *Bands* respectively). In addition, the authors take a further step by trying to improve the performance by incorporating background knowledge. It consists of a set of key words, in this case, explicitly introduced by the user. Using this strategy, the Bayesian classifier performs $82-83\%$ and $78-79\%$ for *Biomedical* and for *Bands*. Furthermore, according to the outcomes reached by ID3, (remember that DBDT can be viewed as an extension of it) they are still lower, obtaining $70.2-75.9\%$ and $68.6-70.7\%$ for *Biomedical* and for *Bands*. Thus, in the light of these comparisons, we can conclude that DBDT can be applied for profile user learning and its competitive or even better to other methods.

5 Conclusions

In this paper, we have studied the DBDT proposal to tackle web classification problems. DBDT has been tested over structured and non structured well-known classification problems, showing a good performance [3] in both. For this reason, we consider that this algorithm can be applied for more concrete scenarios, such as classification web.

Currently, we are thinking over how the metric conditions could be expressed into terms of patterns associated to the metric function (for instance, *"belongs to"* could be a pattern for sets), and obtain a transformed (and more comprehensible) model containing rules (see Section 3) as this one:

IF the word "football" appears in *Content* **and** the connections {([football],[match]), ([match],[team,players])} are found in *Structure* **THEN** this web-site is a sport web-site.

Although some progress has already been made in this line [9], the underlaying formalism needs to be developed a bit further in order to derive an algorithm which allows us to "transform" the metric conditions into more comprehensible patterns.

References

[1] S. Chakrabarti, B. Dom and P. Indyk. Enhanced hypertext categorization using hyperlinks. In *SIGMOD Conference*, pages 307–318, 1998.

[2] M. Craven and S. Slattery. Relational learning with statistical predicate invention: Better models for hypertext. *Journal of Machine Learning*, 43(1/2), pages 97–119, 2001.

[3] V. Estruch, C. Ferri, J. Hernández-Orallo and M.J. Ramírez-Quintana. Distance-based decision trees for structured data. Technical report, DSIC-UPV, 2005, *submitted to ECML 2005*.

[4] O. Etzioni. The world-wide web: Quagmire or gold mine? *Communications of the ACM*, 39(11), pages 65–68, 1996.

[5] W. Hu. Webclass: Web document classification using modified decision trees.

[6] S. Slattery and T. M. Mitchell. Discovering test set regularities in relational domains. In *Proc. of the Seventeenth International Conference on Machine Learning (ICML 2000)*, pages 895–902, 2000.

[7] A. Sun, E. P. Lim, and W. K. Ng. Web classification using support vector machine. In *Proc. of the 4th Int. Workshop on Web Information and Data Management*, pages 96–99, 2002.

[8] C. Thornton. *Truth from Trash: How Learning Makes Sense*. The MIT Press, Cambridge, Massachusetts, 2000.

[9] V. Estruch, C. Ferri, J. Hernández-Orallo and M. J. Ramírez-Quintana. Identifying generalisation patterns for distance-based methods. In *Proc. of the 15th International Conference on Inductive Logic Programming (ILP 2005)*, to appear, 2005.

[10] I. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java implementations*, Ed. Morgan Kaufmann Publishers, 2000.

[11] T. Gaertner. A survey of Kernels for Structured Data. *SIGKDD Exploration Newsletter*, 5(1): pages 49–58, 2003.

[12] M. Pazzani and D. Billsus. Learning and Revising User Profiles. The Identification of Interesting Web Sites. In *Journal of Machine Learning*, 27, pages 313–331, 1997.

[13] B. Mendelson. *Introduction to Topology*. Dover Pubn., 3rd edition, 1990.

[14] T.F. Smith and M.S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147, pages 195–197, 1981.

[15] T. Gaertner, J. W. Lloyd and P. A. Flach. Kernels and distances for structured data. *Journal of Machine Learning*, 57, 2004.

[16] Y. Yang and J. O. Perdersen. A Comparative Study on Feature Selection in Text Categorisation. In *Proc. of the 14th International Conference on Machine Learning (ICML 1997)*, pages 412–420, 1997.

[17] T. Mitchell. Machine Learning. *McGraw Hill*, 1997.

[18] J. R. Quinlan. C4.5. Programs for Machine Learning. *Morgan Kaufmann Publishers*, 1993.

[19] L. Breiman, J. H. Friedman, R. A. Olshen and C. J. Stone. Classification and Regression Trees. *Wadsworth*, 1984.