A Theoretical Framework for Web Categorization in Hierarchical Directories using Bayesian Networks

Luis M. de Campos, Juan M. Fernández-Luna, and Juan F. Huete

Departamento de Ciencias de la Computación e Inteligencia Artificial, E.T.S.I. Informática. Universidad de Granada, C.P. 18071, Granada (Spain)
{lci,jmfluna,jhg}@decsai.ugr.es

Summary. In this paper, we shall present a theoretical framework for classifying web pages in a hierarchical directory using the Bayesian Network formalism. In particular, we shall focus on the problem of multi-label text categorization, where a given document can be assigned to any number of categories in the hierarchy. The idea is to explicitly represent the dependence relationships between the different categories in the hierarchy, although adapted to include the category descriptors. Given a new document (web page) to be classified, a Bayesian Network inference process shall be used to compute the probability of each category given the document. The web page is then assigned to those classes with the highest posterior probability.

1 Introduction

Information available on Internet can be accessed by at least two different methods: searching and browsing. When searching, users submit a query to a search engine and obtain a list of web pages which are ranked according to their relevance in terms of the query. When browsing, users navigate through a hierarchy of interlinked concepts or classes until they find relevant documents. A user's information need is less defined in the second method than it is in the first.

This paper shall explore the second method of satisfying information needs. Web pages are organized into a directory or catalog¹ which contains a wide set of categories or classes. These classes are usually arranged hierarchically. Each class contains a group of web pages, the main subject of which is the topic represented by the class to which they have been assigned. Yahoo! [17] is a typical example of a directory which is constructed manually (i.e. by human experts, who are also responsible for web document allocation).

¹ In this paper, we shall use the terms *directory* and *catalog* interchangeably.

As the number of web pages on Internet is constantly growing, the task of determining a category for each new web document is very slow, one reason why techniques related to text classification [13] are being developed and applied to this framework. The main differences between general text classification and web classification methods is that the first is usually applied to a flat set of categories, while in the second, the categories belong to a well-defined structure. Automatic web categorization by means of a directory therefore implies finding the most appropriate class or category in the hierarchy for a given web page, using a computer program, taking into account the content of that candidate document, in order to insert it into that class.

As mentioned above, directories are hierarchies of categories. These are normally structured according to an inclusion relation, or in other words, an IS-A relation [12], usually forming a tree-like structure. The top-most category (placed in the root node) is supposedly the most general, with an increasing level of specificity as the category approaches the leaf nodes (the most specific ones). There are several organizations of the directory, according to the taxonomy presented in [15]. In the model that we propose in this paper, we shall deal with two types of these²: *Virtual Category tree*, where categories are the tree nodes, and documents are assigned to the leaf nodes (the most specific classes); and *Category tree*, where web pages could also be placed in internal nodes.

In this paper, we shall study how to apply the formalism of Bayesian Networks [11] for the classification of web pages in a hierarchical directory. In particular, our objective is to perform *multi-label text categorization* in a *category tree* in a web framework. In order to build the classifier, our inputs are the hierarchical catalog on one side, and a set of documents (each represented as a vector of terms) that have been previously classified under these categories on the other. Figure 1 presents an example of the input data, where for instance *d2* might be a document about a multimedia presentation of the comedy “The Pink Panther”. The hierarchy is usually completed with the inclusion of a new (dummy) general category containing all the classes.

The idea is to construct a Bayesian Network which represents the dependence relationships between the input data (the catalog and the set of pre-classified documents), although adapted to the unique characteristics of the problem. Given a new document (web page) to be classified, the a posteriori probability of relevance for each category is computed using an efficient Bayesian Network inference process. The web page is then assigned to those classes with a higher posterior probability.

Since multi-label categorization might be allowed, the classifier is committed to achieving the greatest possible specificity. If, on the one hand, the document is about how to construct a computer, explaining its components (CPU, hard disks, monitors, etc.), as for instance document *d4* in Figure 1,

² The remaining taxonomy structures present the most general topology of a directed acyclic graph.

Fig. 1. Documents and hierarchical catalogs.

then it is convenient to classify the document under the label “Hardware” and not to classify it under each different subclass. On the other hand, if a document is about the new release of RedHat Linux, such as for example document $d1$ in Figure 1, then the classifier must select the “Operating System” category. Moreover, and as a general rule that must be fulfilled, a document cannot be classified under two different categories in the same branch of the tree, for example document $d2$ cannot be classified as “comedy” and “movie” simultaneously.

In order to describe the previously mentioned model, this paper is divided into the following sections: Section 2 presents a brief introduction to Bayesian Networks; Section 3 explores related work in hierarchical categorization; Section 4 describes the qualitative and quantitative components of the model; Section 5 shows how the inference can be performed efficiently; and finally, Section 6 includes the paper’s conclusions and proposals for future research.

2 Introduction to Bayesian Networks

Bayesian networks are graphical models which are capable of efficiently representing and manipulating n -dimensional probability distributions [11]. A Bayesian network uses two components to codify qualitative and quantitative knowledge:

- A *Directed Acyclic Graph* (DAG), $G = (V, E)$, where the nodes in V represent the random variables from the problem we want to solve, and the topology of the graph (the arcs in E) encodes conditional (in)dependence relationships between the variables (by means of the presence or absence of direct connections between pairs of variables);
- A set of conditional probability distributions drawn from the graph structure: for each variable $X_i \in V$, we have a family of conditional probability distributions $P(X_i|pa(X_i))$, where $pa(X_i)$ represents any combination of the values of the variables in $Pa(X_i)$, and $Pa(X_i)$ is the parent set of X_i in G .

From these conditional distributions, we can recover the joint distribution over V :

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1}^n P(X_i | pa(X_i)) \quad (1)$$

This decomposition of the joint distribution results in important savings in storage requirements. It also allows probabilistic inference (propagation) to be performed in many cases, i.e. the posterior probability to be computed for any variable given some evidence about the values of other variables in the graph [7, 11]. The dependence/independence relationships enabling this decomposition are graphically encoded (through the d-separation criterion [11]) by means of the presence or absence of direct connections between pairs of variables. Bayesian networks can efficiently perform reasoning tasks: the independences represented in the graph reduce changes in the state of knowledge to local computations.

3 Related Work on Hierarchical Categorization

In [13], there is a very thorough review of the different methods related to general text classification. In this section, however, we shall focus our attention on certain papers related to hierarchical classification.

A first attempt to solve the problem was to ignore the structure and to consider the problem as a flat set of independent categories. However, the common philosophy found in literature when the hierarchical structure of the classes is taken into account is that different classifiers are learnt which deal with smaller problems, and these distinguish first at higher levels, and later in more specific classes.

This methodology is followed by Koller and Sahami [9], who take advantage of the hierarchical nature of the problem, dividing the original classification problem into smaller problems, which could only be solved with a small set of features, and allowing more complex Bayesian classifiers for such a task. More precisely, they build a hierarchy of classifiers which are applied successively.

Mladeníć [10] put a similar idea into practice in order to use Yahoo! to classify documents. She used the structure to carry out a feature selection process as well as to learn a set of independent Naive Bayesian classifiers.

Another example can be found in [20], where Weigend et al. use an approach based on a two-level classification scheme on Reuters-22173 dataset: the first stage attempts to classify the new document into main categories (called “meta-topics”) which group all the classes comprising the hierarchy. Once the meta-topic has been decided, a second stage attempts to assign a more specific category. In both processes, neural networks are employed.

Support Vector Machines have also been applied to the hierarchical classification of web content, as in [4]. These only consider top levels of the hierarchy,

and base their work on learning some classifiers and combining the resulting scores. A second paper where Support Vector Machines are applied is [15].

Another interesting paper was written by Frommholz [5]. This researcher uses a completely different approach in that he uses a non-hierarchical classifier at a first stage. A hierarchical classifier is later used to improve the results obtained by the first, using the classes closest to the class that was selected in the first attempt. Another interesting innovation of his model is that it can assign documents to internal classes.

Finally, [12] is another example that puts this idea into practice, but, in this case, with neural networks.

To sum up, we can say that in all these approaches, classification is performed in a top-down manner. This fact allows the models to scale well with the number of characteristics used [18]. One of the problems of this approach appears when a document is wrongly classified at high levels and cannot be passed to the classifiers at lower levels. Sun et al. [16] introduce the blocking factor measure as a new kind of classifier-centric performance measure to tackle this problem but involving more classifiers and with some degradation in precision.

4 Representing Hierarchical Web Directories using a Bayesian Network

In this section, we shall describe the Bayesian network classifier. In order to do so, we shall proceed incrementally with a step-by-step explanation of the topology of a particular component of the classifier, i.e. we shall identify the set of variables³ and how they are related. Finally, when the topology has been completed, we shall describe how the quantitative values (probability distributions) might be assessed.

- *Hierarchical catalog*: In this case, each category in the catalog will be considered a random variable and therefore represented by a node in the Bayesian Network. For convenience in terms of inference, as we shall explain later, categories are divided into two groups: *basic categories*, which are the most specific and do not contain other classes: $C_b = \{B_1, B_2, \dots, B_m\}$; and *complex categories*, those that are composed of other basic or complex categories: $C_c = \{S_1, S_2, \dots, S_n\}$. Therefore, $C = C_b \cup C_c$. Each node B_j or S_k has an associated binary random variable, which can take its values from the sets $\{b^-, b^+\}$ or $\{s^-, s^+\}$ (the category is not related or is related to a given document, respectively, in the sense that its description matches the document content). In this paper, we shall use C_i to represent a general category (basic or complex) and $\{c^-, c^+\}$ to denote the particular values it takes.

³ In this paper, the random variable and its associated node in the graph will be noted identically.

Regarding the topology of the model, there is an arc from one category to the particular category it belongs to, expressing the fact that the relevance of a given category will depend on the relevance values of the different elements that comprise it. With this criterion, we shall obtain a tree-like topology, more specifically a polytree, since a node can have more than one parent. Figure 2 represents the Bayesian Network associated with the catalog.

Fig. 2. Bayesian Network based catalog.

- *Relating documents and categories:* For each document D_j , we know the set of basic or complex categories where it has been pre-classified, for instance by a human expert. We have two different problems: the first consists in determining which characteristics of those documents relating to category C_i will be used for categorization purposes, and the second consists in modeling how these characteristics should be related to the class. From a Bayesian network perspective, these problems are the selection of the set of variables of interest and the construction of the network topology.
 - Characteristic Selection: The set of characteristics that will be considered relevant to C_i is a subset of the set of terms contained in those documents pre-classified under C_i . We use $\mathcal{T}_j$ to denote the subset of terms summarizing the content of a document D_j . These terms are obtained after “stop word” elimination and the stemming processes. In order to reduce the high dimensionality of the term space, a term selection process should also be applied, for instance by considering only the terms with the highest discrimination power or by using methods based on Information-Theoretical functions [19].

We have also decided to consider the proper set of documents as relevant variables. We think that this approach might be useful in web domains where the set of documents is highly (inter)connected by means of links. For instance, from this information we might determine the quality of a document, in the sense that it might give a better description of the class than others.

The Bayesian Network must therefore include variables representing the set of pre-classified documents, $\mathcal{D}$, and the set of terms obtained after dimensionality reduction, $\mathcal{T}$. The first will be represented by the set $\mathcal{D} = \{D_1, D_2, \dots, D_n\}$, while the second will be denoted as $\mathcal{T} = \{T_1, T_2, \dots, T_l\}$. As before, each node represents a bivaluated variable taking its values from the sets $\{d^-, d^+\}$ and $\{t^-, t^+\}$ (the document/term is not related or is related to the document), respectively.

– Relating characteristics and classes: In order to represent the relationships between terms and documents, we shall include an arc from each term T_i to those documents that have been indexed with T_i . A similar approach could have been used to relate documents and class, i.e. to include an arc between a given document and all the categories which it belongs to. In this case, we would be expressing the fact that the relevance of a category will depend on the relevance values of the different documents classified under it. For basic categories, this criterion seems to be coherent but considering the previous Bayesian network representing the catalog, complex categories should be related to both documents and categories. In order to give a coherent semantics to complex category nodes⁴, we therefore propose that a virtual node, S'_k , be included and which represents the complex category S_k , gathering the information supported by those documents pre-classified under category S_k . It should be noted that virtual nodes will act as basic categories, and in terms of notation, the set of basic categories is therefore extended in the following way: $\mathcal{C}_b = \{B_1, B_2, \dots, B_f, S'_1, \dots, S'_n\}$, where $S'_1, \dots, S'_n$ represent the virtual nodes.

Finally, an arc from a virtual category to the particular complex category it represents will be added, and this expresses the fact that the relevance of the complex category will also depend on the relevance values of the documents belonging to this category. Figure 3 represents the topology of the classifier in this step, where virtual category nodes are represented with filled boxes and their relationships with the associated complex category nodes are represented with dashed lines.

It should be noted that in the Bayesian network, root nodes are terms and the leaf node represents the general category. It is also worth mentioning that the hierarchical structure of the model determines that each category $C \in \mathcal{C}$ has only one category as its child, the single category containing C (except for the most general category, which has no child). We shall use $Ch(C)$ to denote the single child node associated with node C (with $Ch(C) = null$ if C is the general node).

Once the model has been completed, regardless of the assessment of the probability values (see Section 4.2), it can be used to perform classification tasks. Thus, given a new unseen document, D , we must be able to compute the posterior probabilities of relevance of all the categories $C \in \mathcal{C}$, $p(c^+|D)$,

⁴ This fact will be essential when assessing the probability distributions.

Fig. 3. Bayesian Network based classifier.

representing our belief that D can be categorized under C . In order to compute these probability values, we must instantiate all the terms indexing the document to the related value, t^+ , and propagate this information through the network (in Section 5 we show how this process can be performed efficiently). Once we have computed all the probabilities $p(c^+|D)$, the document should then be classified under the category with the maximum value in the case of single label categorization or those categories with higher posterior probability values in the case of multi-label categorization, checking that the document has not been classified under two different categories in the same branch of the catalog. It should be noted that for the set of categories, the probabilities $p(c^+|D)$ might be considered as a kind of measure of vector closeness in the term space.

Perhaps the main advantage of this proposal with respect to classical methods (see Section 3) is that in our case there is no need for a training stage (since the topology is fixed, learning the classifier comes down to computing the probability distributions). This advantage is all the more important when we consider the Web categorization task because the sets $\mathcal{D}$ and $\mathcal{C}$ might not be completely available from the start. New pre-classified documents can therefore be incorporated into the system and new categories can be added and obsolete ones deleted easily. Moreover, our model scales well with the dimensionality of the term space and the dimensionality of the catalog. Another important feature of our model is that we can naturally perform multi-label categorization, explicitly exploiting the hierarchical structure of the category, whereas those methods using a divide and conquer-type approach (which decompose the classification problem into a number of smaller classification problems) have mainly been designed for single label categorization since they select the best branch of the hierarchy at each step.

4.1 Improving the Basic Model

In order to motivate our approach, let us only consider the set $\mathcal{C}_b$ that includes basic categories and virtual nodes (as a representation of the complex categories). In this case, by using $p(c_i^+|D), \forall C_i \in \mathcal{C}_b$, we can perform a flat (non-hierarchical) classification measuring the closeness of document D to the centroid of the positive documents pre-classified under C_i .

A typical error made by flat classifiers is that a large proportion of the false positives (incorrect categories with high probabilities) were on topics which are semantically related to the proper one (mainly because the vocabulary they use is connected, but only incidentally, with the topic). For example, while the term “LINUX” is a good predictor of the category “Operating System”, it also tends to appear in documents relating to “Multimedia”, for instance considering how to set up a particular application in this operating system. The problem worsens when we consider virtual nodes (representing complex classes), where they should have many terms in common with the subclasses which they comprise. The key is that these terms will rarely appear in documents relating to other branches of the catalog. For instance, it will be difficult to find the term “LINUX” in connection with the entertainment category. Although this discussion has only been made for basic categories, it can easily be extended to the remaining categories in the hierarchy.

In this section, we shall therefore present an approach to tackle this problem, and which basically consists in also taking the information supported by the set of *related categories* into account when making decisions. This idea has also been considered in different text classification approaches such as Schapire et al. [14] for flat categorization and Weingend et al. [20] for hierarchical categorization. In our case, and considering the catalog hierarchy, the set of categories related to C_i will be defined as those sibling categories in the catalog plus the category it belongs to. This definition is equivalent to considering those categories represented by nodes with the same child in the Bayesian network. At this point, we must note that the information supported by the set of related categories of C_i is gathered by the complex category it belongs to, $Ch(C_i)$. It therefore seems natural that the final decision will depend on both the relevance of C_i and the relevance of $Ch(C_i)$.

In order to model this fact in the Bayesian Network, a new set of variables, $\mathcal{A}$, will be included representing the *adequacy* of a category to the new document. In particular, for each category variable C_i (excluding the general class and the set of virtual nodes), a new adequacy variable A_i is added, i.e. $\mathcal{A} = \{A_1, A_2, \dots, A_{m+n}\}$ with $A_i \in \{a_i^-, a_i^+\}$ representing the fact that the i -th category is or is not appropriate for the document. The model will be completed by adding two new arcs for each adequacy node: the first arc connects node C_i with A_i , and the second connects node $Ch(C_i)$ also with A_i , expressing the fact that the adequacy of a category to the document will depend on the relevance of C_i and $Ch(C_i)$. Section 4.2 discusses how we can measure the strength of these relationships.

Fig. 4. The final hierarchical classifier.

Figure 4 presents a part of the final Bayesian network, obtained from the one displayed in Figure 3. The adequacy nodes have been represented with squares. For example, the node label with A_OS represents the adequacy of the “Operating Systems” category.

4.2 Assessment of the Probability Distributions

In this section, we shall present some guidelines about how to assess the quantitative information. For each node X_i , we need to estimate the conditional probabilities $p(x_i^+ | pa(X_i))$ with $pa(X_i)$ being a given configuration of the values of the set of parents of X_i , $Pa(X_i)$, in the graph. In general, the estimation of these probability values is not an easy problem because the number of possible configurations $pa(X_i)$ (any assignment of values to all the variables in $Pa(X_i)$), and therefore the number of conditional probabilities that we need to estimate and store, grows exponentially with the number of parents in X_i . For example, if X_i has 20 parents (and this may be a common situation in our model), we need 2^{20} (around one million) probability distributions, hence we cannot use a standard approach. Since computing and storing the conditional probabilities becomes prohibitive, we propose an approach that has been successfully used by the BNR model in the field of Information Retrieval [2]⁵ which is based on the use of canonical models of multicausal interaction [11].

⁵ The subgraph containing only terms and document nodes is the basis for the BNR model [2].

We shall therefore discuss the possible alternatives for each type of node:

- *Term nodes*: Because all the terms are root nodes, marginal distributions need to be stored. We propose that identical probability be used for all the terms, $p(t^+) = p_0, \forall T \in \mathcal{T}$:

$$p(t_i^+) = p_0 \quad \text{and} \quad p(t_i^-) = 1 - p_0 \quad (2)$$

- *Document nodes*: In order to estimate the conditional probabilities of relevance of a document D_j , $p(d_j^+ | pa(D_j))$, we shall consider the canonical model used in [2], i.e.

$$p(d_j^+ | pa(D_j)) = \sum_{\substack{T_i \in D_j \\ t_i^+ \in pa(D_j)}} w(T_i, D_j) \quad (3)$$

where $w(T_i, D_j)$ represents the weight of term T_i in document D_j and the expression $t_i^+ \in pa(D_j)$ in eq. (3) means that only the weights w_{ij} are included in the sum such that the value assigned to the corresponding term T_i in the configuration $pa(D_j)$ is t_i^+ . Therefore, the greater the number of relevant terms in $pa(D_j)$, the greater the probability of relevance of D_j . In addition, the weights $w(T_i, D_j)$ verify that $0 \leq w(T_i, D_j) \forall i, j$ and $\sum_{T_i \in D_j} w(T_i, D_j) \leq 1 \forall j$ and are computed by means of the following expression:

$$w(T_i, D_j) = \frac{tf_{ij}idf_i^2}{\sum_{T_k \in D_j} tf_{kj}idf_k^2} \quad (4)$$

with tf_{ij} being the frequency of the term in the document and idf_i the inverse document frequency of the term T_i in the document training set. Different alternatives for computing these weights might be the *cosine formula*, probabilistic techniques [6], etc.

- *Basic Categories*: Similar to the previous case, the set of basic categories (including virtual nodes) would have a large number of parents, one for each document that has been pre-classified under the category. It would therefore be convenient to use a canonical model similar to the one used for document nodes, i.e.

$$p(b_j^+ | pa(B_j)) = \sum_{\substack{D_i \in B_j \\ d_i^+ \in pa(B_j)}} w(D_i, B_j) \quad (5)$$

Different approaches might be considered to define the support that each document gives to the class, i.e. the link weights $w(D_i, B_j)$. For instance, we can consider that all the documents equally support the class, or that these weights depend on the set of terms indexing the document. A different approach which is suitable for web categorization is to explore the definition of these weights considering some measure of the quality of web

pages that can be obtained using the links pointing to these pages, and this is similar to HITS [8] or PageRank [1] algorithms.

- *Complex Categories:* In the case of a complex category, S_i , the situation is different since we can distinguish between the information supported by its associated virtual node S'_i , i.e. the information that comes from those documents that have been explicitly pre-classified under the category, and the information provided by those (sub)categories included in it, $Pa(S_j) \setminus S'_j$. Although an approach similar to the previous one can be used, it seems natural to consider that the relevance of the virtual node has a different strength, and therefore a different treatment, than the remaining parents in the category.

We therefore propose that $p(s_j^+ | pa(S_j))$ be computed using the following convex combination where the parameter $\alpha_j, 0 \leq \alpha_j \leq 1$, is used to discriminate between the strength of the relevance supported by its virtual node, S'_j , and the strength of the relevance degree supported by its proper subclasses. For instance, the value α_j might depend on the number of documents pre-classified under $Pa(S_j) \setminus S'_j$. If there is a large number of such documents, we can therefore consider that the (sub)classes are well-informed and α_j should be lower than if we do not have enough documents classified into $Pa(S_j) \setminus S'_j$.

$$p(s_j^+ | pa(S_j)) = \begin{cases} \frac{1-\alpha_j}{Z_j-1} \sum_{\substack{C_i \in Pa(S_j) \setminus S'_j \\ c_i^+ \in pa(S_j)}} w(C_i, S_j) & \text{if } s_j'^+ \notin pa(S_j) \\ \alpha_j + \frac{1-\alpha_j}{Z_j-1} \sum_{\substack{C_i \in Pa(S_j) \setminus S'_j \\ c_i^+ \in pa(S_j)}} w(C_i, S_j) & \text{if } s_j'^+ \in pa(S_j) \end{cases} \quad (6)$$

where $Z_j = \sum_{C_k \in Pa(S_j)} w(C_k, S_j)$. In addition, $w(C_i, S_j)$ with $C_i \in Pa(S_j)$ represents the weight of the parent category C_i in S_j . We assume that $w(S'_j, S_j) = 1$. Different alternatives should be considered to estimate the weights $w(C_i, S_j)$ with $C_i \in Pa(S_j) \setminus S'_j$, for instance to consider that all the (sub)categories are equally probable (uniform distribution) or to consider that these weights depend on the ratio of the number of terms (final features) describing the (sub)class C_i with respect to the number of terms describing the class S_j .

- *Adequacy Nodes:*

The probability distributions stored in these nodes will be used to measure the strength of the information supported by the categories related with a node C_i in that the more relevant $Ch(C_i)$ is, the less adequate C_i is. This estimation is simple since an adequacy node A_i has only two parents: category C_i and its child $Ch(C_i)$. For instance, if $C_j = Ch(C_i)$, these distributions should be computed using

$$\begin{aligned} p(a_i^+ | c_i^+, c_j^+) &= \beta_i & \text{and} & & p(a_i^+ | c_i^+, c_j^-) &= 1 \\ p(a_i^+ | c_i^-, c_j^+) &= 0 & \text{and} & & p(a_i^+ | c_i^-, c_j^-) &= 0 \end{aligned} \quad (7)$$

where β_i could be a value between 0 and 1 in such a way that the smaller β_i is, the greater importance we are giving to the nodes connected with C_i , i.e. $Ch(C_i)$.

5 Categorizing Web Pages: Inference

In this section, we shall examine how web page categorization is performed in the Bayesian network classifier introduced in the previous section. Our objective is therefore to find a category that best represents the content of the new document (in the case of single label categorization) or several categories (in the case of multi-classification). The best class or classes are selected according to the highest posterior probabilities of adequacy of each category given the content of the web page.

Classification is initially carried out by instantiating the terms belonging to the new document in the network to the relevant value, acting as evidences, i.e. $p(t^+) = 1$. From that point, a propagation process is performed on the entire network. On account of the size of the network (in particular, the high number of documents and term nodes) and its topology, the application of classic propagation algorithms could be a very time-consuming task.

Taking advantage of the canonical model to estimate the probability distributions in documents and basic and complex category nodes, the propagation process could be reduced to the evaluation of probability functions in documents and basic and complex category nodes. This inference method has been widely and successfully applied in the Information Retrieval framework with the Bayesian Network Retrieval and the Context-based Influence Diagram models [2, 3]. Moreover, this propagation has been proved to be exact, i.e. it computes the same values as a classic algorithm [2] but very efficiently. Finally, and in order to compute the posterior probability of the adequacy of a class given a document, a simple computation of Bayes' rule is only performed in some of the nodes belonging to the set $\mathcal{A}$.

Therefore, given the new unseen document $D = \{T_a, T_b, \dots, T_k\}$, whose terms act as evidences, i.e. $E = D = \{T_a, T_b, \dots, T_k\}$, the classification is carried out in the following stages:

1. Calculate the relevance probability of document nodes given the evidences, i.e. $p(d^+ | E)$:

$$\forall D_j \in \mathcal{D}, p(d_j^+ | E) = \sum_{T_i \in Pa(D_j) \setminus E} w(T_i, D_j) p(t_i^+) + \sum_{T_i \in Pa(D_j) \cap E} w(T_i, D_j). \quad (8)$$

2. Obtain the posterior probability of each basic category, B , or virtual node, S' , i.e. $p(b^+ | E)$ or $p(s'^+ | E)$, respectively. In this case, we need to evaluate the following equation:

$$\forall C_j \in \mathcal{C}_b, \quad p(c_j^+ | E) = \sum_{D_i \in Pa(C_j)} w(D_i, C_j) p(d_i^+ | E). \quad (9)$$

3. Compute the posterior probability of each complex category node, S , i.e. $p(s^+ | q)$ is performed evaluating the following formula:

$$\forall S_j \in \mathcal{C}_c, p(s_j^+ | E) = \alpha_j p(s_j'^+ | E) + \frac{(1 - \alpha_j)}{Z_j - 1} \sum_{C_i \in Pa(S_j) \setminus S_j'} w(C_i, S_j) p(c_i^+ | E), \quad (10)$$

with S_j' being the associated virtual node corresponding to the S_j complex category, and $\alpha_j \in [0, 1]$ a parameter that helps us to give more or less importance to the content of the internal category S_j .

4. Finally, in the penultimate stage, the probability of each adequacy node, A , must be computed, i.e. $p(a^+ | E)$. In this case, if $C_j = Ch(C_i)$, these values are obtained simply by solving this equation:

$$\forall A_i \in \mathcal{A}, \quad P(a_i^+ | E) = \sum_{\mathbf{c}_i, \mathbf{c}_j} p(a_i^+ | \mathbf{c}_i, \mathbf{c}_j) p(\mathbf{c}_i, \mathbf{c}_j | E), \quad (11)$$

where $\mathbf{c}_i$ and $\mathbf{c}_j$ represent the two values that C_i and C_j can take, respectively. In order to put this model into practice, it is therefore necessary to assess the bi-dimensional posterior probabilities corresponding to each category C_i and the category C_j where it is contained, $p(C_i, C_j | E)$. Obtaining these values may be a time-consuming process due to the large amount of calculations required on classification time. This is why we propose the use of a first approximation assuming that both categories are independent given the evidence, i.e.

$$p(C_i, C_j | E) = p(C_i | E) p(C_j | E). \quad (12)$$

We therefore only need to compute the adequacy values for each category given a new document E , using equations (9) and (10). The final expression that enables this value to be calculated is given by the following equation:

$$\begin{aligned} \forall A_i \in \mathcal{A}, \quad P(a_i^+ | E) &= \sum_{\mathbf{c}_i, \mathbf{c}_j} p(a_i^+ | \mathbf{c}_i, \mathbf{c}_j) p(\mathbf{c}_i | E) p(\mathbf{c}_j | E) \\ &= \beta_i p(c_i^+ | E) p(c_j^+ | E) + p(c_i^+ | E) (1 - p(c_j^+ | E)) \\ &= p(c_i^+ | E) [1 - (1 - \beta_i) p(c_j^+ | E)]. \end{aligned} \quad (13)$$

5. Select the classes with the highest posterior probability. As mentioned before, if only a single category is assigned, the one with the highest posterior probability is selected. In case of multi-label classification, we could opt to select the k first categories or those with a posterior probability greater than a threshold, ensuring that two selected classes do not belong to the same branch.

Fig. 5. Example of a Hierarchical Bayesian Network-based Categorization Model.

All the computations of the posterior probabilities in Steps 1 to 3 can be performed very efficiently using a single traversal of the graph, considering that all the nodes in $\mathcal{A}$ have been removed, as well as the arcs pointing to them, starting only from the instantiated terms in E , provided that the prior probabilities of relevance have been calculated and stored within the structure. An algorithm that computes all the posterior probabilities $p(d^+|E)$, $p(b^+|E)$, $p(s'^+|E)$ and $p(s^+|E)$, starts from the terms in E and carries out a width graph traversal until it reaches the documents that require updating, computing $p(d^+|E)$ using eq. (8). Starting from these nodes, it carries out a depth graph traversal to compute $p(c^+|E)$, only for those basic and complex units that require updating, using eq. (9) and (10).

Example:

In order to illustrate the behavior of the proposed model, let us consider the example in Figure 5. So as to clarify the figure, some term nodes have been omitted⁶, although they have been used for inference tasks. In order to set

⁶ We think that in order to show how the evidences are related to the different categories, it is enough to consider the a posteriori probability of relevance for the set of pre-classified documents. This is why most term nodes are omitted in Figure 5.

the quantitative values, we use the scheme proposed in Subsection 4.2, where the used weights are displayed near the arcs. The prior probabilities of all the terms have been set to 0.5, the values α in Equation 6 have been set as $\alpha_{S1} = \alpha_{S2} = 0.6$ and $\alpha_{S3} = 0.5$. Finally, the values β in Equation 7 have been set to: $\beta_{ABi} = 0.75, i \in \{1, 2, 3, 4\}$; $\beta_{AS1} = \beta_{AS2} = 0.8$ and $\beta_{AS3} = 0.85$. It should be noted that with this criterion, the strength given to those categories relating to a general class is lower than the strength given to those categories relating to the more specific ones. This performance seems to be reasonable since, given a hierarchy, the relationships between categories at the bottom (general classes) might be more spurious than at the top (specific classes).

Firstly, it is worth remembering that when there is no evidence, the probability for all the nodes is 0.5, except for adequacy nodes taking the values $ABi = 0.44, i \in \{1, 2, 3, 4\}$, $AS1 = AS2 = 0.45$ and $AS3 = 0.46$. These values might be used as reference in the next discussion. It should also be noted that, a priori, a document tends to be classified under the most general category.

In the example, we shall classify two new documents, illustrating different situations that should appear. The different rows of Table 1 show the relevance probability for those pre-classified documents, basic categories or virtual nodes, complex categories and adequacy nodes that require updating. These values have been obtained by propagating the evidence through the network using the above equations. It should be noted that by showing these probability values, we can also discuss the results obtained by a flat classifier (considering only nodes in C_b) and the basic hierarchical model (without the improvements discussed in Section 4.1, i.e. without adequacy nodes and considering the probabilities of simple and complex categories). In order to perform the final classification, we shall consider the three most relevant categories.

Let us assume that the first unseen document to be classified, i.e. the evidence, is $E_1 = \{T3, T5, T6\}$. In this case, E_1 is only related to documents $D4, D6, D9$ and $D10$ that have been pre-classified under $B1, B2$ and $S2$ categories. In this case, considering the adequacy, the new document will be classified (in order) under $B1, B2$ and $S2$. It should be noted that the class $S1$, with an adequacy greater than $S2$, has not been selected because some of its (sub)classes were previously assessed. The same results will also be obtained for flat and basic hierarchical classification.

Finally, we use a second document, E_2 , with 16 terms being considered. Looking at row E_2 in Table 1, it can be seen that this new set of terms provide strong evidences to half the set of pre-classified documents, being related to all the categories. In this case, considering the adequacy values, the document will only be classified under $S1$ and $S2$. It should be noted that the document should not be classified under $S3$ because it has been previously classified under $S1$. Although the model selects two different categories on the same level of the hierarchy, we can distinguish two different situations. On one hand, E_2 is closely related to the categories $B1$ and $B2$ ($P(B1|E_2) = 0.81$ and $P(B2|E_2) = 0.82$) but it is also similarly related to

Table 1. $p(\cdot|E_i)$ for the updated nodes

E_1	$D5 = D6 = 0.9; D9 = 0.95; D10 = 0.7$ $B1 = 0.77; B2 = 0.73; S2' = 0.6$ $S1 = 0.60; S2 = 0.56; S3 = 0.54$ $A_{B1} = 0.68; A_{B2} = 0.64; A_{B3} = A_{B4} = 0.44; A_{S1} = 0.53; A_{S2} = 0.50; A_{S3} = 0.48$
E_2	$D1 = 1; D3 = 0.6; D4 = 0.6; D5 = 0.9; D6 = 0.93; D7 = 0.85;$ $D9 = 1; D10 = 0.7; D11 = 0.8; D16 = 0.65; D21 = 0.79$ $B1 = 0.81; B2 = 0.82; B3 = 0.52; B4 = 0.58; S1' = 0.78; S2' = 0.73; S3' = 0.60$ $S1 = 0.79; S2 = 0.64; S3 = 0.66$ $A_{B1} = 0.65; A_{B2} = 0.66; A_{B3} = 0.44; A_{B4} = 0.49; A_{S1} = 0.68; A_{S2} = 0.56; A_{S3} = 0.59$

the category $S1$ ($P(S1'|E2) = 0.78$). The model takes this fact into account and selects the category $S1$. On the other hand, $S2$ has been selected because $E2$ is strongly related to $S2$ ($P(S2'|E2) = 0.73$) but slightly related with the subcategories $B3$ and $B4$. It is interesting to note that although $S1'$ and $S2'$ are quite relevant, the lower adequacy of $S3$ can be explained by considering the strength of $S3'$ in $S3$ and that there is not enough evidence for those documents pre-classified under $S3$ ($P(S3'|E2) = 0.6$). Finally, and regarding the categorization obtained by the flat and the basic hierarchical classifiers, both will classify the document under $B1$, $B2$ and $S2$. These models can not classify the document under $S1$ unless $P(S1'|E2) > \max\{P(B1|E2), P(B2|E2)\}$.

6 Concluding Remarks and Future Works

In this paper, we have presented a theoretical framework for classifying web pages in an existing hierarchy of categories belonging to a directory. This classifier is based on Bayesian networks. Using a very efficient propagation algorithm in the network, given a new web document, the model determines the most appropriate class or category. One of the main advantages of this classifier compared to others facing the same problem is that there is no need to perform a training stage because the topology is given by the structure of the directory and the probability distributions could be computed directly from the data; only the parameters α and β should be tuned. Another important aspect is that the size of the directory could increase without this representing a problem for the classifier. In such a case, it will not be re-trained again and only new weights must be computed. The model can also assign multiple classes to a web document, only by selecting the highest k classes in the ranking.

By way of future work, we intend to evaluate the model in a real web environment (such as for example, the Yahoo directory) in order to test its performance and make any necessary modifications.

Acknowledgments

This work has been supported by the Spanish Fondo de Investigación Sanitaria, under Project PI021147.

References

1. S. Brin and L. Page, (1998). The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30:1–7, pp. 107–117.
2. L.M. de Campos, J.M. Fernández-Luna, J.F. Huete, (2003) The BNR model: Foundations and performance of a Bayesian network retrieval model, *International Journal of Approximate Reasoning* 34:265–285.
3. L.M. de Campos, J.M. Fernández-Luna, J.F. Huete, (2004) Using context information in structured document retrieval: an approach based on influence diagrams, *Information Processing and Management* 40(5):829–847.
4. S. Dumais, H. Chen (2000). Hierarchical Classification of Web Content. In *Proceedings of the SIGIR Conference* 256–263.
5. I. Frommholz (2001). Categorizing Web Documents in Hierarchical Catalogues. In *Proceedings of the 23rd Conference on Information Retrieval*
6. N. Govert, M. Lalmas and N. Furf (1999). A probabilistic description-oriented approach for categorising web documents. In *Proc. of the ACM Intern. Conference on Information Knowledge and Management*, 475–482.
7. F. V. Jensen (1996). *An Introduction to Bayesian Networks*. University College London Press, London.
8. J. Kleinberg. (1999) Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46:5, pp. 604–632.
9. D. Koller, M. Sahami (1997). Hierarchically classifying documents using very few words. In *Proceedings of the 14th International Conference on Machine Learning* 170–178.
10. D. Mladenić (1998). Turning Yahoo into an Automatic Web-page Classifier. In *Proceedings of the 13th European Conference on Artificial Intelligence* 473–474.
11. J. Pearl (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan and Kaufmann, California.
12. M. E. Ruiz, P. Srinivasan (2002). Hierarchical Text Categorization Using Neural Networks. *Information Retrieval* 5:87–118, 2002
13. F. Sebastiani (2002). Machine Learning in Automated Text Categorization. *ACM Computing Surveys* 34(1):1–47.
14. R. Schapier, E. Singer and A. Singhal (1998) Boosting and Rocchio applied to text filtering. In *Proc. of the SIGIR'98, 21st Intern. Conference on Research and Development in Information Retrieval*. 215–223.
15. A. Sun, E. Lim (2001). Hierarchical Text Classification and Evaluation. In *Proceedings of the IEEE International Conference on Data Mining* 521–528.
16. A. Sun, E. Lim, W. Ng, J. Srivastava (2004). Blocking Reduction Strategies in Hierarchical Text Classification. In *IEEE Transactions on Knowledge and Data Engineering*, 18 (10), 1305–1308.
17. www.yahoo.com

18. Y. Yang, J. Zhang and B. Kisiel. (2003). A scalability of classifiers in text categorization. In Proc. SIGIR'03, Intern. Conference on Research and Development in Information Retrieval. 96–103.
19. Y. Yang and J. Pedersen (1997). A comparative study on feature selection in text categorization. In Proc. of International Conference on Machine Learning. 412–420.
20. A.S. Weigend, E.D. Weiener, J.O. Pedersen (1999). Exploiting Hierarchy in Text Categorization. Information Retrieval 1:193–216.