Transductive Learning of Logical Structures from Document Images

Michelangelo Ceci, Corrado Loglisci, and Donato Malerba

Abstract. A fundamental task of document image understanding is to recognize semantically relevant components in the layout extracted from a document image. This task can be automatized by learning classifiers to label such components. The application of inductive learning algorithms assumes the availability of a large set of documents, whose layout components have been previously labeled through manual annotation. This contrasts with the more common situation in which we have only few labeled documents and an abundance of unlabeled ones. A further degree of complexity of the learning task is represented by the importance of spatial relationships between layout components, which cannot be adequately represented by feature vectors. To face these problems, we investigate the application of a relational classifier that works in the transductive setting. Transduction is justified by the possibility of exploiting the large amount of information conveyed in the unlabeled documents and by the contiguity of the concept of positive autocorrelation with the smoothness assumption which characterizes the transductive setting. The classifier takes advantage of discovered emerging patterns that permit us to qualitatively characterize classes. Computational solutions have been tested on document images of scientific literature and the experimental results show the advantages and drawbacks of the approach.

1 Introduction

In document image understanding, one of the fundamental tasks is to recognize semantically relevant components in the layout extracted from a document image. This recognition process is based on domain-specific knowledge, which is represented in very different forms (e.g. formal grammars or production rules). Several prototypical document image understanding systems have been developed by

Michelangelo Ceci · Corrado Loglisci · Donato Malerba
Dipartimento di Informatica, Università degli Studi di Bari "Aldo Moro"
via Orabona 4 - 70125 Bari
e-mail: {ceci,loglisci,malerba@}di.uniba.it

M. Biba and F. Xhafa (Eds.): Learning Structure and Schemas from Documents, SCI 375, pp. 121–142.
springerlink.com

manually encoding the required knowledge in specific formalisms (e.g., DeLoS [34]). However, the layout of documents, even for the same publisher, may change considerably with time. Moreover, there might be a drift in the type of documents of interest in a specific context. To prevent obsolescence of the developed systems it is necessary to continuously update the required knowledge, which is unfeasible if based only on manual encoding. *Versatility*, that is, guaranteed competence over a broad and precisely specified class of document images, has been recently recognized as a key requirement for document image analysis systems [7]. In order to deal with this requirement, the application of machine learning methods has been advocated both to build models of image quality, layout and language and to infer the parameters of such models. Starting from a training set of document images, learning methods are able to extract a large number of features relevant for understanding the document structure.

The application of machine learning methods to document image understanding has been investigated for almost two decades [19]. From an operational viewpoint, a human operator provides a document image analysis system with image samples of documents and then detects and labels semantically relevant layout components, from which models of document structures are induced. This supervised learning approach, even though providing some flexibility, still does not ensure the key requirement of versatility. Indeed, to acquire the necessary knowledge on a really broad class of documents, supervised learning methods may require a large set of labeled documents. This contrasts with the more common situation in which only few labeled training documents are available, due to the significant cost of manual annotation. Therefore, it is important to exploit the large amount of information potentially conveyed by unlabeled documents to better estimate the data distribution and to build more accurate recognition models.

Two main settings have been proposed in the literature to exploit information contained in both labeled and unlabeled data: the *semi-supervised* setting and the *transductive* setting [40]. The former is a type of inductive learning, since the learned function is used to make predictions on any possible example. The latter requires less - it is only interested in making predictions for the given set of unlabeled data. When the set of documents to label is known a priori, the transductive setting is more suitable, since it appears to be an easier problem than (semi-supervised) induction.

In this chapter, we investigate the problem of transductive learning of document image understanding models. In the proposed method, unlabeled documents are used to reprioritize models learned from labeled documents alone. Indeed, while discriminative learning methods base their decisions on the posterior probability $p(y|x)$, the transductive learning method uses unlabeled documents to improve the estimate of the prior probability $p(x)$, and hence correct the posterior probability $p(y|x)$ by assuming some form of dependence with $p(x)$.

The proposed learning method follows a logic-based approach in which models are represented by a set of rules expressed in relational (or first-order) logic. In order to “understand” the layout structure of an unlabelled document, rules are matched against the relational description of the document layout. The *relational representation* of the document layout and rules is motivated by the fact that layout objects can be related by a number of spatial relationships, such as distance, directional and topological relationships. Standard feature vectors or equivalent propositional representations do not allow relationships in general, and spatial relationships in particular, to be straightforwardly represented.

The study of relational learning in a transductive setting has received little attention in the research community (notable exceptions are [11] for classification tasks, and [4] for regression tasks), although the transductive setting seems especially suitable for relational datasets which are characterized by positive autocorrelation [30]. The application of transductive relational learning to bootstrap the labelling process of document image collections remains an unexplored research direction.

The paper is organized as follows. In Section 2, the present work is motivated and the problem is defined. In Section 3, related works are discussed. Sections 4 and 5 are devoted to the presentation of the method. Finally, experimental results are reported in Section 6 and then some conclusions are drawn.

2 Motivation and Problem Definition

The recognition of semantically relevant layout components in document images is part of a complex transformation process of document images into a structured symbolic form which facilitate the modification, storage, retrieval, reuse and transmission of documents themselves [33]. This transformation is articulated in several steps. Initial processing steps include binarization, skew detection, and noise filtering. Then the document image is segmented into several layout components, such as text lines, half-tone images, line drawings or graphics (this step is called *layout analysis*). The understanding of document images follows layout analysis. It aims to associate a "logical label" (e.g. title, abstract of a scientific paper, picture in a newspaper) to semantically relevant layout components (called *logical components*), as well as to extract relevant relationships between logical components (e.g., reading order).

Document image understanding is typically based on layout information, such as the relative positioning of layout components or the size of layout components, as well as on content information (e.g., textual, graphical). This is the case of the work reported in this chapter, where the association of logical labels to layout components is based on both layout information and textual information. However, the novelty here is mainly in the strategy applied to build a classifier which can be used to recognize semantically relevant components.

In the literature, several methods have been proposed for building classifiers to be used in document understanding. They are briefly reviewed in the next Section. However, most of them assume that training data are represented in a single table of a relational database, such that each row (or tuple) represents an independent example (a layout component) and the columns correspond to properties of the example (e.g., height of the layout component). This single-table assumption, however, is too strong for at least three reasons. First, layout components cannot be realistically considered independent observations, because their spatial arrangement is mutually constrained by formatting rules typically used in document editing. Second, spatial relationships between a layout component and a variable number of other components in its neighborhood cannot be properly represented by a fixed number of attributes in a table. This is even more the case when layout components are heterogeneous (e.g., half-tone images, text lines) and have different properties (e.g., brightness, font size). Third, logical components may be mutually related (e.g., the title of a paper "is followed" by the author list). Since the single-table assumption limits the representation of relationships (spatial or non) between examples, it also prevents the discovery of these mutual dependencies which can be useful in document image understanding [31].

The above considerations motivate us to consider a relational representation of document layouts. In fact, document layout structures are a kind of spatial data, and relational approaches have been recently advocated as useful in spatial domains [30]. Document layout structures are also subject to spatial autocorrelation, i.e., a property value of a layout component depends on the property values observed for other layout components in the neighborhood. Spatial autocorrelation clearly indicates a violation of the independence assumption of observations usually made in statistics. By considering this in the definition of the learning method, it is possible to improve the performance of the learned classifiers.

A second motivation for this work is to face the usual scarcity of labeled documents which prevent the application of inductive learning algorithms to generate accurate classifiers. Indeed, manual annotation of the many layout components in a document is very demanding. Therefore, it is important to exploit the large amount of information potentially conveyed by unlabeled documents to better estimate the data distribution and to build more accurate classification models [21]. This is possible in transductive learning, which is formalized as follows:

Let D be a dataset labeled according to an unknown target function, whose domain is $\mathbf{X} = X_1, X_2, \ldots, X_m$ and whose range is a finite set $Y = \{C_1, C_2, \ldots, C_L\}$. Given:

- a training set $TS \subset D$, and
- the projection of the working set $WS = D - TS$ on $\mathbf{X}$,

the goal is to predict the class value of each example in the working set WS as accurately as possible.

The learner receives full information (including labels) on the examples in TS and partial information (without labels) on the examples in WS and is required to predict the class values only of the examples in WS. The original formulation of the problem of function estimation in a transductive (*distribution-free*) setting requires TS to be sampled from D without replacement. This means that, unlike the standard inductive setting, the examples in the training (and working) set are supposed to be mutually dependent. Vapnik also introduced a second (*distributional*) transduction setting, in which the learner receives training and working sets, which are assumed to be drawn i.i.d. from some unknown distribution. As shown in ([44], Theorem 8.1), error bounds for learning algorithms in the distribution-free setting also apply to the more popular distributional transductive setting. Therefore, in this work we focus our attention on the first setting.

There is an interesting convergence of opinions which motivates us to investigate relational learning in the transductive setting. On the one hand, it is claimed that transduction is most useful when the standard i.i.d. assumption is violated ([14]). On the other hand, it is observed that statistical independence of examples is contradicted by many relational datasets ([23]). Moreover, it has also been observed that the presence of (positive) spatial autocorrelation in a dataset entails the smoothness assumption of transductive learning [30]. Therefore, the application of transductive learning to sets of spatially related layout components seems appropriate and worth of being investigated.

In the case of relational data, the problem of transductive classification can be more precisely formulated as follows:
Given:

- a database schema SC which consists of a set of h relational tables $\{T_0,\dots,T_{h-1}\}$, a set PK of primary keys on the tables in SC, and a set FK of foreign key constraints on the tables in SC,
- a target relation $T \in SC$ (that permits us to represent layout components) and a target discrete attribute Y in T, different from the primary key of T, whose domain is the finite set $\{C_1,C_2,\dots,C_L\}$ (Logical label),
- the projection T' of T on all attributes of T except Y,
- a training (working) set that is an instance TS (WS) of the database schema SC with known (unknown) values for Y;

Find: the most accurate prediction of Y for examples in WS.

In the proposed approach, the prediction of Y is based on a classification framework that works in the relational data mining setting. It is mainly inspired by an associative classification framework proposed by Ceci and Appice [9] where association rules discovered from training datasets are used by a naïve Bayes classifier which operates on relational representations of spatial data.

More precisely, given an object E to be classified, a classical naïve Bayes classifier assigns E to the class C_i, that maximizes the *posterior probability* $P(C_i|E)$. By applying the Bayes theorem, $P(C_i|E)$ is expressed as follows:

$$P(C_i|E) = \frac{P(C_i)\cdot P(E|C_i)}{P(E)}. \tag{1}$$

In fact, the decision on the class that maximizes the posterior probability can be made only on the basis of the numerator, that is $P(C_i)\cdot P(E|C_i)$, since $P(E)$ is independent of the class C_i.

To work on relational representations, Ceci and Appice proposed considering a set $\Re$ of association rules, expressed as first order definite clauses, which are mined on the training set and can be used to define a suitable decomposition of the likelihood $P(E|C_i)$ *à la* naive Bayes, in order to simplify the probability estimation problem. In particular, if $\Re(E) \subseteq \Re$ is the set of first order definite clauses, whose antecedent covers E, the probability $P(E|C_i)$ is defined as follows:

$$P(E|C_i) = P(\bigwedge_{R_j \in \Re(E)} antecedent(R_j)|C_i). \qquad (2)$$

The straightforward application of the naïve Bayes independence assumption to all literals in $\bigwedge_{R_j \in \Re(E)} antecedent(R_j)$ is not correct, since it may lead to underestimating $P(E|C_i)$ when several similar clauses in $\Re(E)$ are considered for the class C_i. To prevent this problem the authors resort to the logical notion of factorization [39]. Details are reported in [9].

This associative classification framework for relational classification has been subsequently extended in order to use Emerging Patterns (EPs) instead of association rules. EPs are introduced in [17] as a particular kind of pattern (or multi-variate features), whose support significantly changes from one data class to another: the larger the difference of pattern support, the more interesting the pattern. Change in pattern support is estimated in terms of support ratio (or *growth rate*). EPs with sharp change in support (high growth rate) are useful to discriminate a class from the other classes. This observation motivated Ceci et al. [10] to investigate both the discovery of relational EPs and their usage in the associative classification framework. Experimental results proved the effectiveness of this extension. Therefore, in this work, we consider emerging-pattern based associative classifiers to define a new transductive learning algorithm for document image understanding.

3 Related Work

In the literature there are already several works on automatic recognition of semantically relevant layout components. Akindele and Belaïd [2] proposed applying the R-XY-Cuts method on training data, in order to extract the layout structures and to match them against an initial model defined by an expert. The aim of the matching is to discard training documents whose layout structure is very different from the expected one. Then, a generic model of the logical structure is built by means of a tree-grammar inference method applied to validated layout structures with associated labels. Therefore, this approach is based on demanding human intervention, which is not only limited to layout labeling but also involves the specification of an initial model.

Walischewski [45] proposes representing each document layout by a complete attributed directed graph, with one vertex for each layout object. The vertex attributes are pairs (l, c), where l denotes the type of layout component (page, block, line, word, char), while c denotes the logic label of the layout object (e.g. title, author). Edges have thirteen attributes corresponding to Allen's qualitative relations on intervals [3]. An attribute of the edge (v_i, v_j) is a pair (h, v) describing qualitatively the relative horizontal/vertical location between the two vertices v_i and v_j. The learning algorithm returns triples $[(c_i, c_j), (h, v), (w_h, w_v)]$ stating that Allen's relation $h(v)$ holds between c_i and c_j along the horizontal (vertical) axis with strength w_h (w_v). Altogether, the triples define an attributed directed graph representing the model. Recognition is based on an error tolerant subgraph isomorphism between the graphs representing the document and the model. This approach, although relational, only handles qualitative information and has been tested on simple layout structures extracted from envelopes.

In the work by Palmero et al. [35] a document can be considered as a sequence of objects, where the object labels depend both on the geometrical properties of the block (size, position, etc.) and on the decisions made for previous sequence items. As in the work by Walischewski, there is an implicit recognition of the importance of considering autocorrelation on logical labels, although, in this case, the original bidimensional spatial autocorrelation boils down to one-dimensional temporal autocorrelation, which is handled by a recursive neuro-fuzzy learning algorithm. The effect of sequence ordering on blocks is not examined.

Probabilistic relaxation [6] is a general approach to deal with autocorrelation on logical labels. Indeed, objects are initially classified on the basis of their properties and then their classification is iteratively adjusted by using compatibilities with other objects found in the neighborhood. Le Bourgeois et al. [41] tested this approach on blocks delimiting words and compared it to a naïve Bayesian classification, by taking into account both word features and features of neighboring (left/right) words.

Aiello et al. [1] applied the well-known decision tree learning system C4.5 [38] to learn classification rules for textual logical components (body, caption, title, page number). Only seven attributes are considered: two for the geometrical properties of the block (aspect ratio, area ratio), four for the textual content (font size ratio, font style, number of characters, number of lines) and one for spatial closeness to a figure. Experimental results show that these seven features are sufficient to learn a decision tree with a very high ($> 90\%$) recognition rate for body, title and page number. However, the experimentation is mostly based on ground truth data for layout structures and textual content, which is an ideal situation.

Interestingly, although there have been attempts to deal with a relational representation of data [12], studies reported in the literature for document image understanding, do not consider the transductive learning setting. Therefore, we intend this contribution to be a further step towards the investigation of methods which originate from the intersection of these three promising research areas: namely, transduction, relational data mining and document image understanding.

For transductive learning, several methods have been proposed in the literature. They are based on support vector machines ([8] [21] [24] [15]), on k-NN classifiers ([25]) and even on general classifiers ([27]). However, they do not work with relational data. For relational classification in the transductive setting, three methods have been reported in the literature.

Krogel and Scheffer ([26]) investigate a transformation (known as *propositionalization*) of a relational description of gene interaction data into a classical double-entry table and then study transduction with the well-known transductive support vector machines. Therefore, transduction is not explicitly investigated on relational representations and it is based on propositionalization, which is fraught with many difficulties in practice ([16, 22]).

Taskar et al. ([42]) build, on the framework of Probabilistic Relational Models, a *generative* probabilistic model which captures interactions between examples, either labeled or unlabeled. However, given sufficient data, a *discriminative* model generally provides significant improvements in classification accuracy over generative models ([43]). This motivates our interest in designing classifiers based on discriminative models.

Ceci et al. [11] propose a different probabilistic method that is based on an iterative approach that bootstraps classification labels on unlabeled examples. However, this approach does not permit us to exploit a preliminary descriptive phase (e.g. association rule discovery or emerging pattern discovery) that helps to obtain a twofold advantage. First, the user can decide to mine both a descriptive and a classification model in the same data mining process [29]. Second, we can solve the *understandability* problem [36] that may occur with some classification methods. Indeed, many rules produced by standard classification systems are difficult to understand because these systems often use only domain independent biases and heuristics, which may not fulfill users' expectations. With the descriptive classification approach, the problem of finding understandable rules is reduced to a post-processing task [29].

Data mining research has provided several solutions for the task of emerging pattern discovery. In the seminal work by Dong and Li [17], a border-based approach is adopted to discover the EPs discriminating between separate classes. Borders are used to represent both candidates and subsets of Emerging Patterns (EPs); the border differential operation is then used to discover the EPs. Zhang et al. [46] have described an efficient method, called ConsEPMiner, which adopts a level-wise generate-and-test approach to discover EPs which satisfies several constraints (e.g., growth-rate improvement). Recently, Fan and Ramamohanarao [20] have proposed a method which improves the efficiency of EPs discovery by adopting a CP-tree data structure to register the counts of both the positive and negative classes. All these methods assume that data to be mined are stored in a single data table. An attempt to upgrade the emerging pattern discovery to deal with relational data has been reported in [5], where the authors propose adapting the *levelwise* method described in [32] to the case of relational emerging patterns.

4 Extracting Emerging Patterns with SPADA

In this work, we propose a modified version of the system SPADA [28], originally designed for *relational* frequent pattern discovery, that permits us to extract emerging patterns. SPADA represents relational data *à la* Datalog, a logic programming language with no function symbols specifically designed to implement deductive databases. Moreover, it takes into account a background knowledge (*BK*) expressed in Prolog and is able to mine relational patterns at multiple levels of granularity, in order to properly deal with hierarchies of objects. When these are available, it is important to take them into account, since patterns involving more abstract objects are better supported (although less precise), while patterns involving more specific objects have higher confidence values (although lower support values). Hence by efficiently exploring the pattern space at different levels of abstraction (or granularity) it is possible to find the right trade-off between these two conflicting criteria.

SPADA distinguishes between the set S of *reference* (or target) *objects*, which are the main subject of analysis, and the sets R_k, $1 \leq k \leq m$, of *task-relevant* (or non-target) objects, which are related to the former and can contribute to account for the variation. Each unit of analysis includes a distinct reference object and many related task-relevant objects. Therefore, the description of a unit of analysis consists of both the properties of included reference and task-relevant objects as well as their relationships. From a database viewpoint, S corresponds to the target table $T \in SC$ and each R_k corresponds to a different relational table $T_i \in SC$. A unit of analysis corresponds to a tuple in $t \in T$ and to all the tuples in the databases related to t according to foreign key constraints.

In the following sub-sections, the document description problem is presented and the learning strategy is described, as it has been modified in order to mine emerging patterns.

4.1 Document Description

In the logic framework adopted by SPADA, a relational database is boiled down into a deductive database. Properties of both reference and task-relevant objects are represented in the extensional part D_E, while the domain knowledge is expressed as a normal logic program which defines the intensional part D_I. For example, we report a fragment of the extensional part of a deductive database D which describes spatial and textual information extracted from the document image reported in Figure 1:

block(b1). block(b2). ...
height(b2,[11..54]). width(b1,[7..82]). ...
on_top(b2,b1). ... on_top(b2,b3). ...
part_of(b1,p1). part_of(b2,p1). page_first(p1). ...
abstract(b1). title(b2). ...
text_in_abstract(b1,'base'). text_in_title(b2,'model')....

In this example, $b1$ and $b2$ are two constants which denote as many distinct layout components (reference objects) , while $p1$ denotes a document page (task-relevant object). Predicate *block* defines a layout component, *part_of* associates a block to a document page, *height* and *width* describe geometrical properties of layout components, *on_top* expresses a topological relationship between layout components, *page_first*$(p1)$ refers to the position of the page in the document, *abstract* and *title* associate $b1$ and $b2$ with a logical label, *text_in_abstract* and *text_in_title* permit us to describe the textual content of the logical components.

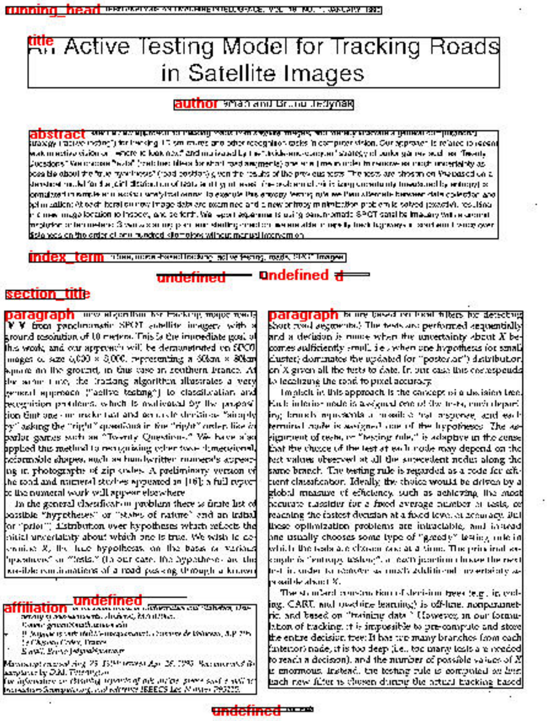
An Active Testing Model for Tracking Roads in Satellite Images

Fig. 1 Document Layout: logical components

The complete list of predicates is reported in Table 1. The aspatial feature *type_of* specifies the content type of a layout component (e.g. image, text, horizontal line). Logical features are used to associate a logical label to a layout object and depend on the specific domain. In the case of scientific papers (considered in this work), possible logical labels are: *affiliation*, *page_number*, *figure*, *caption*, *index_term*, *running_head*, *author*, *title*, *abstract*, *formulae*, *subsection_title*, *section_title*, *biography*, *references*, *paragraph*, *table*.

Textual content is represented by means of another class of predicates, which are true when the term reported as the second argument occurs in the layout component denoted by the first argument. Terms are automatically extracted by means of a text-processing module. All terms in the textual components are tokenized and the set of obtained tokens is filtered out in order to remove punctuation marks, numbers and tokens of less than three characters. Standard text preprocessing methods are used to:

Table 1 Used predicates

Layout structure	Locational features	$x\_pos\_center/2$
		$y\_pos\_center/2$
	Geometrical features	$height/2$
		$width/2$
	Topological features	$on\_top/2$
		$to\_right/2$
	Aspatial feature	$type\_of/2$
Logical structure	Logical features	application dependent (e.g., *abstract/1*)
Text	Textual features	application dependent (e.g., $text\_in\_abstract/2$)

1. remove stopwords, such as articles, adverbs, prepositions and other frequent words;
2. determine equivalent stems (stemming), such as “topolog” in the words “topology” and “topological”, by means of Porter’s algorithm for English texts [37].

Only relevant tokens are used in textual predicates. They are selected by maximizing the product $maxTF \times DF^2 \times ICF$ [13] that scores high terms appearing (possibly frequently) in a single logical component c and penalizes terms common to other logical components. More formally, let c be a logical label associated to a textual component. Let d be the bag of tokens in a component labeled with c (after the tokenizing, filtering and stemming steps), w a term in d and $TF_d(w)$ the relative frequency of w in d. Then, the following statistics can be computed:

1. the maximum value $TF_c(w)$ of $TF_d(w)$ on all logical components d labeled with c;
2. the document frequency $DF_c^2(w)$, i.e., the percentage of logical components labeled with c in which the term w occurs;
3. the category frequency $CF_c(w)$, i.e., the number of labels $c' \neq c$, such that w occurs in logical components labeled with c'.

Then, the score v_i associated to the term w_i belonging to at least one of the logical components labeled with c is:

$$v_i = TF_c(w_i) \times DF_c^2(w_i) \times 1/CF_c(w_i) \tag{3}$$

According to this function, it is possible to identify a ranked list of “discriminative” terms for each of the possible labels. From this list, we select the best n_{dict} terms in $Dict_c$, where n_{dict} is a user-defined parameter. The textual dimension of each logical component d labeled as c is represented in the document description as a set of ground facts that express the presence of a term $w \in Dict_c$ in the specified logical component.

4.2 The Mining Step

In the original version of SPADA, the problem of mining frequent patterns can be formalized as follows:
Given

- a set S of *reference objects*,
- some sets R_k, $1 \leq k \leq m$, of *task-relevant objects*,
- a background knowledge BK including some hierarchies H_k on objects in R_k,
- M granularity levels in the descriptions (1 is the highest while M is the lowest),
- a set of granularity assignments Ψ_k which associate each object in H_k with a granularity level
- a set of thresholds $minsup[l]$ for each granularity level
- a language bias LB that constrains the search space;

Find frequent multi-level patterns, i.e., frequent patterns involving objects at different granularity levels.

Hierarchies H_k define *is-a* (i.e., taxonomical) relations on task-relevant objects. The frequency depends on the granularity level l at which patterns describe data. Therefore, a pattern P with support s at level l is *frequent* if $s \geq minsup[l]$ and all ancestors of P with respect to H_k are frequent at their corresponding levels.

SPADA operates in two steps for each granularity level: i) pattern generation; *ii)* pattern evaluation. It exploits statistics computed at granularity level l when computing the supports of patterns at granularity level $l+1$. The expressive power of first-order logic is utilized to specify both the background knowledge BK, such as hierarchies and domain specific knowledge, and the language bias LB. The LB is relevant to allow the user to specify his/her bias for interesting solutions, and then to exploit this bias to improve both the efficiency of the mining process and the quality of the discovered rules.

In our case, we modified SPADA in order to discover emerging patterns instead of frequent patterns. Accordingly, the mining problem is modified as follows:
Given

- a set S of *reference objects*,
- a label value $y \in Y = \{C_1, C_2, \ldots, C_L\}$ associated to each reference object,
- some sets R_k, $1 \leq k \leq m$, of *task-relevant objects*,
- a background knowledge BK including some hierarchies H_k on objects in R_k,
- M granularity levels in the descriptions,
- a set of granularity assignments Ψ_k which associate each object in H_k with a granularity level
- a couple of sets of thresholds $minsup[l]$ and $minGR[l]$ for each granularity level
- a language bias LB that constrains the search space;

Find A set of multilevel emerging patterns $\{F | supp_{C_i}(F) \geq minsup[l], GR_{C_i}(F) \geq minGR[l]\}$

In this formulation, $supp_{C_i}(F)$ represents the support of pattern F in the subset of reference objects labeled with C_i, while the growth rate $GR_{C_i}(F)$ is defined as:

$$GR_{C_i}(F) = \frac{supp_{C_i}(F)}{supp_{\neg C_i}(F)}$$

where $supp_{\neg C_i}(F)$ is the support of pattern F in the subset of reference objects labeled with $c \in \{C_1, \ldots, C_{i-1}, C_{i+1}, \ldots C_L\}$.

To efficiently mine frequent patterns, SPADA prunes the search space by exploiting the monotonicity of support. Let F' be a refinement of a pattern F (i.e. F' is more specific than F). If F is an infrequent pattern for the class C_i (i.e. $supp_{C_i}(F) < minsup$), then also $supp_{C_i}(F') < minsup$. This means that F' cannot be an emerging pattern that permits us to distinguish C_i from $\neg C_i$. Hence, SPADA does not refine patterns which are infrequent on C_i.

Unfortunately, the monotonicity property does not hold for the growth rate: a refinement of an emerging pattern whose growth rate is lower than the threshold *minGR* may or may not be an EP. However, also in this case it is possible to prune the search space. According to [46], we modified the mining algorithm originally developed in SPADA in order to avoid generating the refinements of a pattern F in the case that $GR_{C_i}(F) = \infty$ (i.e., $supp_{C_i}(F) > 0$ and $supp_{\neg C_i}(F) = 0$). Indeed, due to the monotonicity of support, for each pattern F' obtained as a refinement of F: $supp_{C_i}(F) \geq supp_{C_i}(F')$, then $supp_{C_i}(F') = 0$. Hence, $GR_{C_i}(F') = 0$ in the case that $supp_{C_i}(F') = 0$, while $GR_{C_i}(F') = \infty$ in the case that $supp_{C_i}(F') > 0$. In the former case, F' is not worth considering. In the latter case, we prefer F to F', based on the Occam's razor principle, according to which all things being equal, the simplest solution tends to be the best one (F has the same discriminating ability as F').

In our application domain, reference objects are all the logical components for which a logical label is specified. Task relevant objects are all the logical components (including undefined components), as well as pages and documents. The *BK* is used to specify the hierarchy of logical components (Figure 2). The *BK* also permits us to automatically associate information on page order to layout components, since the presence of some logical components may depend on the page order (e.g. the author is on the first page). This concept is expressed by means of the following Datalog rules stored in the intensional part D_I of the deductive database D:

$at\_page\_first(X)$:- $part\_of(Y,X)$, $page\_first(Y)$.
$at\_page\_intermediate(X)$:- $part\_of(Y,X)$, $page\_intermediate(Y)$.
$at\_page\_last\_but\_one(X)$:- $part\_of(Y,X)$,$page\_last\_but\_one(Y)$.
$at\_page\_last(X)$:- $part\_of(Y,X)$, $page\_last(Y)$.

Moreover, in the *BK* we can also define the predicate $text\_in\_component$:

$text\_in\_component(X,Y)$:- $text\_in\_index\_term(X,Y)$.
$text\_in\_component(X,Y)$:- $text\_in\_references(X,Y)$.
$text\_in\_component(X,Y)$:- $text\_in\_abstract(X,Y)$.
$text\_in\_component(X,Y)$:- $text\_in\_title(X,Y)$.
$text\_in\_component(X,Y)$:- $text\_in\_running\_head(X,Y)$.

```
article
+ − − heading
|   + −− identification
|   | + −− (title, author, affiliation)
|   + −− synopsis
|     + −− (abstract, index_term)
+ − − content
|   + −− final components
|   | + −− (biography, references)
|   + −− body
|     + −− (section_title, subsect_title, paragraph, caption, figure, formulae, table)
+ − − page_component
|   + −− running_head
|   + −− page_number
+ − − undefined
```

Fig. 2 Hierarchy of logical components

It is noteworthy that hierarchies are defined on task relevant objects. This means that, in theory, it is not possible to consider the same reference object at different levels of granularity. To overcome this limitation, we introduced in the *BK* the fact $specialize(X,X)$, which allows SPADA to consider a reference object as a task-relevant object, and we forced SPADA (by means of *LB* constraints) to include the predicate $specialize/2$ in the emerging patterns.

We also extended the language bias of SPADA in order to deal properly with predicates representing textual features. Indeed, the SPADA language bias requires the user to specify predicates that can be involved in a pattern. For instance, if we are interested in patterns that contain the predicate $text\_in\_abstract(A, paper)$, where A is a variable representing a task-relevant object (tro) already introduced in the pattern and *paper* is a constant value representing the presence of the term “paper” in A, we have to specify the following bias rule:

$$lb\_atom(text\_in\_component(old\,tro, paper))$$

This means that it is necessary to specify a rule for each constant value that could be involved in the predicate. However, there are hundreds of constants representing selected terms (the number depends on the n_{dict} constant and on the number of user-selected logical labels for which the textual dimension is considered). To avoid the manual or semiautomatic specification of different *lb_atom*’s, we extended the SPADA syntax for *LB* in order to support anonymous variables:

$$lb\_atom(text\_in\_component(old\,tro, \_))$$

This means that we intend to consider in the search phase those patterns involving the predicate $text\_in\_component/2$, whose second argument is an arbitrary term.

The SPADA search strategy has been consequently modified in order to support this additional feature.

An additional aspect worth to be considered in this work is related to the possibility of dealing with numerical data. Indeed, although the application requires the manipulation of such data in order to consider geometrical features of a layout component, SPADA, in its original version, is not able to automatically deal with them. To avoid this problem, a simple equal-frequency discretization algorithm has been integrated. This allows the mining algorithm implemented in SPADA to process discrete intervals instead of numerical data.

5 Transductive Classification

The transductive classification implemented in our proposal upgrades the EP-based classifier CAEP [18] to the relational setting. It computes a membership score of an object to each class. The score is computed by means of a growth rate based function of the relational EPs covered by the object to be classified. The largest score determines the object's class.

The score is computed on the basis of the subset of relational emerging patterns that cover the object to be classified. Formally, let o be the description of the object to be classified (an object is represented by a tuple in the target table and all the tuples related to it, according to foreign key constraints), $\Re(o) = \{F \in \Re | \exists \theta \; F\theta \subseteq o\}$ is the set of relational emerging patterns that cover the object o.

The score of o on the class C_i is computed as follows:

$$score(o, C_i) = \sum_{F \in \Re(o)} \frac{GR_{C_i}(F)}{GR_{C_i}(F) + 1} sup_{C_i}(F) \tag{4}$$

This measure may result in an inaccurate classifier in the case of unbalanced datasets, i.e. when training objects are not uniformly distributed over the classes. In order to mitigate this problem, in [18] the authors proposed normalizing this score on the basis of the median of the scores obtained from training examples belonging to C_i. This results in the following classification function:

$$class(o) = argmax_{C_i} \frac{score(o, C_i)}{median_{ro \in TS}(score(ro, C_i))} \tag{5}$$

where TS represents the training set.

Although this normalization solves problems due to unbalanced class distribution, in our case the main problem arises from the different number of emerging patterns that are extracted from different classes. This means that, in our case, a different normalization that weights the number of emerging patterns is necessary:

$$score(o, C_i) = \frac{1}{|\Re(o)|} \sum_{F \in \Re(o)} \frac{GR_{C_i}(F)}{GR_{C_i}(F) + 1} sup_{C_i}(F) \tag{6}$$

Since $sup_{C_i}(F)$ represents the probability that a reference object belonging to class C_i is covered by F, Equation (6) can be transformed as follows:

$$score(o,C_i) = \frac{1}{|\Re(o)|} \sum_{F \in \Re(o)} \frac{GR_{C_i}(F)}{GR_{C_i}(F)+1} P(F|C_i) \tag{7}$$

By applying the Bayes theorem:

$$score(o,C_i) = \frac{1}{|\Re(o)|} \sum_{F \in \Re(o)} \frac{GR_{C_i}(F)}{GR_{C_i}(F)+1} \frac{P(C_i|F)}{P(C_i)} \times P(F) \tag{8}$$

where $P(C_i|F)$ can be estimated as the percentage of examples covering F in TS that belong to C_i, $P(C_i)$ can be estimated as the percentage of examples in TS that belong to C_i, and $P(F)$ is the percentage of examples covering F. According to the transductive learning setting, this factor is estimated by considering the whole set of examples ($TS \cup WS$). This would provide a more reliable estimation of $P(F)$ (since it is obtained from a larger population of examples potentially coming from the same distribution).

$$P(F) = \frac{\#\{ro|ro \in TS \cup WS, \exists\theta\ F\theta \subseteq ro\}}{\#\{ro|ro \in TS \cup WS\}} \tag{9}$$

6 Experiments

To evaluate the viability of the proposed transductive approach, it has been evaluated on a real-world dataset, consisting of multi-page articles published in an international journal. In particular, we considered twenty-four papers, published as either regular or short, in the IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), in the January and February issues of 1996. Each paper is a multi-page document, therefore, we processed 217 document images containing 3611 layout components (examples). Among them, the user manually labeled 2,693 layout components, that is, on average, 112.2 components per document and 12.41 per page. The components that have not been labeled are “irrelevant” for the task in hand or are associated to “noise” blocks: they are automatically considered *undefined*. Overall, there are 918 unlabeled layout components.

The dataset is analyzed by means of a 4-fold cross-validation on documents. Each fold contains 902.75 layout components on average and the algorithm is run in order to collect predictive average accuracy and, for each class, average precision and recall. Unlike the standard cross-validation approach, here one fold at a time is set aside to be used as the *training set* (and not as the *test set*). Small training set sizes allow us to validate the transductive approach, but may result in high error rates as well.

Table 2 reports classes involved in the evaluation (logical labels) and the number of examples belonging to them. As can be seen, classes are highly unbalanced in the number of examples.

Table 2 Class and example distribution

logical label	No. of examples
abstract	39
affiliation	23
author	28
biography	21
caption	202
figure	357
formulae	333
index_term	25
page_number	191
paragraph	968
references	45
running_head	230
section_title	65
subsection_title	27
table	48
title	91
undefined	918
TOTAL	3611

In the extraction of emerging patterns, SPADA has been run with different parameters: $minGR = \{2,8,68\}$ and $minsup = \{10\%,20\%,30\%\}$. In Table 3 the average number of emerging patterns extracted with different parameter values is reported. As expected, by increasing *minsup* and *minGR* values, the number of extracted emerging patterns is drastically reduced.

Table 3 Average number of extracted emerging patterns

	minsup		
minGR	10	20	30
2	22959.25	11769.5	7943.75
8	15608.5	8315.25	5947.25
64	10266.75	5306.5	3939.5

By looking at the distribution of emerging patterns over the classes (see Table 4), we note that classes for which the discrimination is simpler are characterized by a higher number of emerging patterns. It is also noteworthy that the number of patterns is not related to the number of examples. This means that the extracted emerging patterns do not suffer from overfitting problems.

An example of an emerging pattern for the class *abstract* is reported in the following:

$$is\_a\_block(A), specialize(A,B), is\_a(B,abstract), text\_in\_component(B,paper).$$

$$supp_{abstract} = 50\%;\ GR_{abstract} = +\infty$$

Table 4 Average number of extracted emerging patterns per class ($minsup = 10\%$, $minGR = 2$)

logical label	Average No. of emerging patterns
abstract	3235.25
affiliation	738.5
author	1569.25
biography	645
caption	512.5
figure	744.75
formulae	595.75
index_term	2099.25
page_number	1443
paragraph	780.75
references	2782.25
running_head	2751.25
section_title	995.5
subsection_title	773.5
table	1108.75
title	2184
TOTAL	22959.25

This pattern states that 50% of layout components labeled as *abstract* contains the term "paper". Moreover, this pattern is not satisfied by layout components labeled with a different label. This is probably due to the fact that term "paper" is not selected for other logical labels.

We have the same pattern at a higher level of the hierarchy:

$$is\_a\_block(A), specialize(A,B), is\_a(B,synopsis), text\_in\_component(B,paper).$$

$$supp_{synopsis} = 23.5\%,\ GR_{synopsis} = +\infty.$$

As we can see, support decreases since the term "paper" is not selected for the class *index_term*.

Another example of an emerging pattern for the class *abstract* is:

$$is\_a\_block(A), specialize(A,B), is\_a(B,abstract), only\_left\_col(B,C), C \neq B,$$
$$text\_in\_component(C,index).$$

$$supp_{abstract} = 100\%,\ GR_{abstract} = +\infty.$$

This emerging pattern shows the advantage of exploiting the relational nature of data. Indeed, it shows that layout components labeled as abstract are always aligned with components that contain the term "index" (probably belonging to the class "index_term").

Finally, the emerging pattern:

$$is\_a\_block(A), specialize(A,B), is\_a(B,section\_title), height(B,[6..12]),$$
$$at\_page\_first(B).$$

$$supp_{section\_title} = 55.5\%,\ GR_{section\_title} = 42.67$$

considers other types of task relevant objects (i.e. pages) by exploiting information in *BK*. In this pattern, height is expressed in number of pixels.

In Table 5, average classification accuracy is reported. Results are collected for different values of *minGR* and *minsup*. As we can see, more interesting results are obtained with small values of *minGR* and with high values of *minsup* or, alternatively, with high values of *minGR* and with low values of *minsup*. This means that predictive accuracy significantly depends on the number of extracted patterns. With a high number of patterns, probabilities are flattened and the system loses its discriminative capabilities. On the other hand, when the number of patterns decreases, the system has not enough information to discriminate among classes.

Table 5 Average classification accuracy

	minGR		
minsup	2	8	64
10	43.47	56.67	60.39
20	63.84	58.32	54.8
30	64.63	57.69	44.82

A different perspective of results is reported in Table 6, where precision and recall are reported for each class. Results are collected for $minsup = 10\%$ and $minGR = 2$. As can be noted, results vary significantly from one class to another. Indeed, some classes are more difficult to identify since they do not show regularities (e.g. *subsection_title* vs. *section_title*). Other classes can be more easily identified because the use of text is effective and/or because they are subject to formatting regularities.

Table 6 Average precision and recall per class ($minsup = 10\%$, $minGR = 2$)

logical label	Average precision	Average recall
abstract	0.28593	0.89157
affiliation	0.11167	0.94759
author	0.54731	0.75189
biography	0.16565	0.87101
caption	0.47886	0.07607
figure	0.88800	0.794598
formulae	0.55592	0.31498
index_term	0.45324	0.96687
page_number	0.81994	0.92180
paragraph	0.94674	0.44204
references	0.85855	0.850724
running_head	0.92105	0.80608
section_title	0.47162	0.29978
subsection_title	0.07779	0.74408
table	0.15290	0.23070
title	0.40062	0.68959
Average	*0.5085*	*0.6625*

7 Conclusions

In this work, the induction of a classifier for the automated recognition of semantically relevant layout components has been investigated. In particular, we have investigated the combination of transductive inference with principled relational classification, in order to face the challenges posed by the application domain, characterized by complex and heterogeneous data, which are naturally modeled as several tables of a relational database and characterized by the availability of a small (large) set of labeled (unlabeled) data.

The experiments provide interesting qualitative and quantitative results. For future work, we intend to compare our approach with other competitive approaches proposed in the literature and to employ a different classification strategy that permits us to exploit labels confidently associated to working examples in the classification of other working examples.

Acknowledgements. This work is partial fulfillment of the research objectives of the project "ATENEO 2009 - Estrazione, Rappresentazione e Analisi di Dati Complessi". The authors gratefully acknowledge Dr. Lynn Rudd for reading the final version.

References

1. Aiello, M., Monz, C., Todoran, L.: Document understanding for a broad class of documents. IJDAR 5(1), 1–16 (2002)
2. Akindele, O.T., Belaïd, A.: Construction of generic models of document structures using inference of tree grammars. In: ICDAR 1995: Proceedings of the Third International Conference on Document Analysis and Recognition, vol. 1, p. 206. IEEE Computer Society, Washington, DC, USA (1995)
3. Allen, J.F.: Maintaining knowledge about temporal intervals. Commun. ACM 26(11), 832–843 (1983)
4. Appice, A., Ceci, M., Malerba, D.: Transductive learning for spatial regression with co-training. In: Shin, S.Y., Ossowski, S., Schumacher, M., Palakal, M.J., Hung, C.-C. (eds.) SAC, pp. 1065–1070. ACM Press, New York (2010)
5. Appice, A., Ceci, M., Malgieri, C., Malerba, D.: Discovering relational emerging patterns. In: Basili, R., Pazienza, M.T. (eds.) AI*IA 2007. LNCS (LNAI), vol. 4733, pp. 206–217. Springer, Heidelberg (2007)
6. Rosenfeld, A., Hummel, R., Zucker, S.: Scene labeling by relaxation operations. J IEEE Transactions SMC 6(6), 420–433 (1976)
7. Baird, H.S., Casey, M.R.: Towards versatile document analysis systems. In: Bunke, H., Spitz, A.L. (eds.) DAS 2006. LNCS, vol. 3872, pp. 280–290. Springer, Heidelberg (2006)
8. Bennett, K.P.: Combining support vector and mathematical programming methods for classification, pp. 307–326. MIT Press, Cambridge (1999)
9. Ceci, M., Appice, A.: Spatial associative classification: propositional vs. structural approach. Journal of Intelligent Information Systems 27(3), 191–213 (2006)
10. Ceci, M., Appice, A., Malerba, D.: Emerging pattern based classification in relational data mining. In: Bhowmick, S.S., Küng, J., Wagner, R. (eds.) DEXA 2008. LNCS, vol. 5181, pp. 283–296. Springer, Heidelberg (2008)

11. Ceci, M., Appice, A., Malerba, D.: Transductive learning for spatial data classification. In: Koronacki, J., Raś, Z.W., Wierzchoń, S.T., Kacprzyk, J. (eds.) Advances in Machine Learning I. SCI, vol. 262, pp. 189–207. Springer, Heidelberg (2010)
12. Ceci, M., Berardi, M., Malerba, D.: Relational data mining and ILP for document image understanding. Applied Artificial Intelligence 21(4&5), 317–342 (2007)
13. Ceci, M., Malerba, D.: Classifying web documents in a hierarchy of categories: a comprehensive study. J. Intell. Inf. Syst. 28(1), 37–78 (2007)
14. Chapelle, O., Schölkopf, B., Zien, A.: A discussion of semi-supervised learning and transduction. In: Chapelle, O., Schölkopf, B., Zien, A. (eds.) Semi-Supervised Learning, pp. 457–462. MIT Press, Cambridge (2006)
15. Chen, Y., Wang, G., Dong, S.: Learning with progressive transductive support vector machines. Pattern Recognition Letters 24, 1845–1855 (2003)
16. De Raedt, L.: Attribute-value learning versus inductive logic programming: the missing links. In: Page, D.L. (ed.) ILP 1998. LNCS (LNAI), vol. 1446, pp. 1–8. Springer, Heidelberg (1998)
17. Dong, G., Li, J.: Efficient mining of emerging patterns: Discovering trends and differences. In: International Conference on Knowledge Discovery and Data Mining, pp. 43–52. ACM Press, New York (1999)
18. Dong, G., Zhang, X., Wong, L., Li, J.: CAEP: Classification by aggregating emerging patterns. In: Arikawa, S., Nakata, I. (eds.) DS 1999. LNCS (LNAI), vol. 1721, pp. 30–42. Springer, Heidelberg (1999)
19. Esposito, F., Malerba, D., Semeraro, G.: Multistrategy learning for document recognition. Applied Artificial Intelligence 8(1), 33–84 (1994)
20. Fan, H., Ramamohanarao, K.: An efficient singlescan algorithm for mining essential jumping emerging patterns for classification. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 456–462 (2002)
21. Gammerman, A., Azoury, K., Vapnik, V.: Learning by transduction. In: Proc. of the 14th Annual Conference on Uncertainty in Artificial Intelligence, UAI 1998, pp. 148–155. Morgan Kaufmann, San Francisco (1998)
22. Getoor, L.: Multi-relational data mining using probabilistic relational models: research summary. In: Knobbe, A., Van der Wallen, D.M.G. (eds.) Proc.of the 1st Workshop in Multi-relational Data Mining, Freiburg, Germany (2001)
23. Jensen, D., Neville, J.: Linkage and autocorrelation cause feature selection bias in relational learning. In: Proc. of the Nineteenth International Conference on Machine Learning (2002)
24. Joachims, T.: Transductive inference for text classification using support vector machines. In: Proc. of the 16th International Conference on Machine Learning, ICML 1999, pp. 200–209. Morgan Kaufmann, San Francisco (1999)
25. Joachims, T.: Transductive learning via spectral graph partitioning. In: Proc. of the 20th International Conference on Machine Learning, ICML 2003, Morgan Kaufmann, San Francisco (2003)
26. Krogel, M.-A., Scheffer, T.: Multi-relational learning, text mining, and semi-supervised learning for functional genomics. Machine Learning 57(1-2), 61–81 (2004)
27. Kukar, M., Kononenko, I.: Reliable classifications with machine learning. In: Elomaa, T., Mannila, H., Toivonen, H. (eds.) ECML 2002. LNCS (LNAI), vol. 2430, pp. 219–231. Springer, Heidelberg (2002)
28. Lisi, F.A., Malerba, D.: Inducing multi-level association rules from multiple relations. Machine Learning 55, 175–210 (2004)
29. Liu, B., Hsu, W., Ma, Y.: Integrating classification and association rule mining. In: Knowledge Discovery and Data Mining KDD 1998, New York, pp. 80–86 (1998)

30. Malerba, D.: A relational perspective on spatial data mining. IJDMMM 1(1), 103–118 (2008)
31. Malerba, D., Ceci, M., Berardi, M.: Machine learning for reading order detection in document image understanding. In: Marinai, S., Fujisawa, H. (eds.) Machine Learning in Document Analysis and Recognition. SCI, vol. 90, pp. 45–69. Springer, Heidelberg (2008)
32. Mannila, H., Toivonen, H.: Levelwise search and borders of theories in knowledge discovery. Data Min. Knowl. Discov. 1(3), 241–258 (1997)
33. Nagy, G.: Twenty years of document image analysis in pami. IEEE Trans. Pattern Anal. Mach. Intell. 22(1), 38–62 (2000)
34. Niyogi, D., Srihari, S.N.: Knowledge-based derivation of document logical structure. In: ICDAR 1995: Proceedings of the Third International Conference on Document Analysis and Recognition, vol. 1, p. 472. IEEE Computer Society Press, Washington, DC, USA (1995)
35. Palmero, G.I.S., Dimitriadis, Y.A.: Structured document labeling and rule extraction using a new recurrent fuzzy-neural system. In: ICDAR 1999: Proceedings of the Fifth International Conference on Document Analysis and Recognition, p. 181. IEEE Computer Society Press, Washington, DC, USA (1999)
36. Pazzani, M.J., Mani, S., Shankle, W.R.: Beyond concise and colorful: Learning intelligible rules. In: KDD, pp. 235–238 (1997)
37. Porter, M.F.: An algorithm for suffix stripping. Readings in information retrieval, 313–316 (1997)
38. Quinlan, J.R.: C4.5: programs for machine learning. Morgan Kaufmann Publishers Inc., San Francisco (1993)
39. Robinson, J.A.: A machine oriented logic based on the resolution principle. Journal of the ACM 12, 23–41 (1965)
40. Seeger, M.: Learning with labeled and unlabeled data. Technical report, Institute for Adaptive and Neural Computation. University of Edinburgh (2001)
41. Souafi-Bensafi, S., Parizeau, M., Lebourgeois, F., Emptoz, H.: Bayesian networks classifiers applied to documents. In: ICPR (1), p. 483 (2002)
42. Taskar, B., Segal, E., Koller, D.: Probabilistic classification and clustering in relational data. In: Nebel, B. (ed.) IJCAI, pp. 870–878. Morgan Kaufmann, San Francisco (2001)
43. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1995)
44. Vapnik, V.: Statistical Learning Theory. Wiley, New York (1998)
45. Walischewski, H.: Automatic knowledge acquisition for spatial document interpretation. In: ICDAR, pp. 243–247. IEEE Computer Society Press, Los Alamitos (1997)
46. Zhang, X., Dong, G., Ramamohanarao, K.: Exploring constraints to efficiently mine emerging patterns from large high-dimensional datasets. In: KDD, pp. 310–314 (2000)