Mining Preference Relations to Rank Complex Objects

Michelangelo Ceci, Annalisa Appice, Giuseppe De Giosa, Gianluigi Dileo, Alessandro Lallo, and Donato Malerba

Dipartimento di Informatica, Università degli Studi di Bari
via Orabona, 4 - 70126 Bari - Italy
{ceci, appice, malerba}@di.uniba.it,
{degiosa.giuseppe, dileo.gianluigi, alessandro.lallo}@gmail.com

Abstract. One of the key tasks in data mining and information retrieval is to learn preference relations between objects. Approaches reported in the literature mainly aim at learning preference relations between objects represented according to the classical attribute-value representation. However, the growing interest in data mining techniques able to directly mine data represented according to more sophisticated descriptions necessary to model more complex objects, motivates the investigation of relational learning methods for learning preference relations. In this paper, we present a probabilistic relational data mining method that permits to automatically identify preference relations between complex objects. Such preference relations are then used to identify object rankings. Experiments on a real world application are reported.

1 Introduction

The problem of learning preference functions has recently received increasing attention due to its many potential applications to information retrieval problems. Studies reported in the literature for this learning task are mainly based on two different approaches. The first approach aims at learning a function which assigns a numeric value to each item of a set. This numeric preference is then used to rank items. The second approach asks for less: the learned preference function has to make pairwise comparisons in order to define a relative order (if any) between two objects. In a subsequent step this preference function is used to obtain either a total or a partial ordering of objects in a set.

As regards the first approach, some works reformulate the problem of learning to rank as an ordinal regression problem. For instance, Herbrich et al. [13] propose to learn the mapping of an input vector to a member of an ordered set of numerical ranks. They model ranks as intervals on the real line and consider loss functions that depend on pairs of examples and their target ranks. A similar solution is proposed in [6], where learned functions are modeled by perceptrons.

As regards the second approach, Dekel et al. [7] provide a framework for ranking based on directed graphs, where an arc from A to B means that A has to be ranked higher than B. Arcs are computed according to log-linear models.

A drawback of this approach is that it does quantify the degree of preference. As observed in [3] "ranking algorithms often model preferences, and the ascription of preferences is a much more subjective process than the ascription of, say, classes". To overcome this limitation, Freund et al. [11] propose to exploit a probabilistic approach which permits to compute the probability that A follows B. This probability is computed by a learning algorithm that uses decision stumps as weak learners. The probability is a function of the margin over reweighted examples. Burges et. al [3] propose to estimate probabilities on the basis of a cost function computed according to a logistic regression function. Differently, in [5], a naive Bayesian classifier is used to estimate such probabilities.

Although the first approach appears to be more efficient, it is applicable only when a unique total ordering between objects is admissible. When not all the objects have to be necessarily ranked, or more than one ordering is admissible, the second approach is more suitable.

An important common aspect of all methods reported above is that they work on training data represented in a single relational database table, such that each row (or tuple) represents an object and columns correspond to object properties. This tabular representation of data, also known as *propositional* representation, turns out to be too restrictive for several applications, whose units of analysis have a complex structure involving several objects described by different sets of properties and related by one or more relationships. Some of these objects, called *reference* objects, represent the units of analysis and are the main subject of the analysis. The other objects, called *task-relevant* objects, contribute to defining the units of analysis but are not the target of the analysis. The *relational* representation of these units of analysis can be naturally modeled as a set of tables, such that each table describes a specific type of objects involved in the units of analysis, while foreign key constraints model relationships between objects. Examples of units of analysis with a complex structure can be found in spatial domains, which often involve objects of different types, stored in layers (database relations with a geometry), and implicitly related by locational properties of objects (e.g., an object is close to another).

At the best of our knowledge, the only approach that permits to rank complex objects is presented in [16], where the authors propose to apply an Inductive Logic Programming algorithm [15] to learn a logical theory which defines the successor relation. However, learned definitions are "crisp" and do not provide us with a degree of preference. In this paper we present a different method, which can deal with complex objects, i.e., units of analysis with a complex structure representable by multiple database relations, and returns a probability that an object A follows another object B (in preference). The proposed method follows the multi-relational approach to data mining [9].

The paper is organized as follows. The problem of learning preference relations between complex objects is faced in Sections 2 and 3. The problem of learning rankings is faced in Section 4. Section 5 is devoted to the presentation of an application of the method to detecting reading orders of layout components extracted from document images. Results are reported and commented.

2 Mining Preference Relations

The problem of mining preference relations can be formalized as follows:
Given:

- a database schema S with h relational tables $S = \{T_1, T_2, \ldots, T_h\}$
- a set PK of primary key constraints on tables in S
- a set FK of foreign key constrains on tables in S
- a target relation $T \in S$ [1]
- a precedence relation $PT \in S$ with two attributes. Each tuple in this table represents an ordered pair of reference objects where the first reference object precedes the second one.

Find: A probability estimation $P(a \prec b|a, b)$ for any couple of reference objects a and b represented according to the schema $S - PT$.

By applying the Bayes theorem, $P(a \prec b|a, b)$ can be computed as:

$$P(a \prec b|a, b) = P(a \prec b)P(a, b|a \prec b)/P(a, b). \quad (1)$$

where:

- $P(a \prec b)$ in (1) denotes the prior probability that an object precedes another. This probability might be different from 0.5, since training reference objects might not be totally ordered.
- $P(a, b) = P(a \prec b)P(a, b|a \prec b) + P(b \prec a)P(a, b|b \prec a)$

In order to simplify the estimation of the likelihood $P(a, b|a \prec b)$, conditional independence is assumed (*naïve Bayes assumption*[8]), according to which $P(a, b|a \prec b)$ can be factorized as follows:

$$\begin{aligned} P(a, b|a \prec b) &= P(a_1, \ldots, a_m, b_1, \ldots, b_m|a \prec b) = \\ &= P(a_1|a \prec b) \times \ldots \times P(a_m|a \prec b) \times P(b_1|a \prec b) \times \ldots \times P(b_m|a \prec b) \end{aligned}$$

where $a_1, \ldots, a_m$ represent the set of attribute values of a and $b_1, \ldots, b_m$ represent the set of attribute values of b.

However, this formulation that exploits the naïve Bayesian assumption is clearly limited to propositional representations. In the case of complex objects representations, some extensions are necessary. The basic idea is that of using a set of relational patterns $\Re(a, b)$ to describe the considered objects, and then to define a suitable decomposition of the likelihood *à la* naive Bayesian classifier to simplify the probability estimation problem.

Before describing how the likelihood is computed, it is necessary to provide a formal definition of relational pattern, which is a conjunction of unary and binary predicates[2] of two different types:

[1] Objects in T play the role of reference objects, while objects in $S - \{T, PT\}$ play the role of task relevant objects

[2] Henceforth, "/n" indicates the predicate arity. Unary (binary) predicates are indicated as /1 (/2).

Definition 1 (Property predicate). *A predicate $p/2$ is a property predicate associated to a table $T_i \in S - PT$ if the first argument of p represents the primary key of T_i and the second argument represents another attribute in T_i which is neither the primary key of T_i nor a foreign key.*

Definition 2 (Structural predicate). *A predicate $p/2$ is a structural predicate associated to a table $T_i \in S - PT$ if a foreign key in $S - PT$ exists that connects T_i to a table $T_j \in S$. The first argument of p represents the primary key of T_j and the second argument represents the primary key of T_i.*

Definition 3 (Relational Pattern).
A Relational Pattern is in the form:

$\langle S\rangle\{,\langle attr(A)\rangle\}_{0..1}\{,\langle attr(B)\rangle\}_{0..1}\ \{,\langle rel(C_k, C_j)\rangle\{,\langle attr(C_j, v)\rangle\}_{0..n}\}_{0..n}$

where $attr/1$ represents the predicate associated to the target relation T (The argument represents the primary key of T); $rel/2$ ($attr/2$) represents a generic structural (property) predicate; S is in the form of preference(A,B).

A pattern P in this form is a relational pattern if the property of linkedness [12] is satisfied (e.g. each variable C_k or C_j should be linked to the variables A or B by means of structural predicates).

An example of relational pattern is:

$preference(X, Y), block(X), block(Y), to\_right(X, Z),$
$block\_x\_pos\_centre(Y, [435.1, \ldots, 478.0])$

This pattern states that an object X (of type block) that appears to the left of another object (Z) is preferred to an object Y (of type block) for which property x_pos_centre assumes values in the interval $[435.1, \ldots, 478.0]$.

The likelihood is then computed as follows:

$$P(a, b|a \prec b) = P(\bigwedge_{R_k \in \Re(a,b)} R_k|a \prec b). \quad (2)$$

where $\Re(a, b)$ is the set of relational patterns that cover the tuple $(a, b) \in PT$. The coverage of (a, b) by a relational pattern $R_k \in \Re(a, b)$ demands for matching all variables in R_k against some tuples in the set of relations $S - PT$ according to foreign key constraints. The set $\Re(a, b)$ is a subset of the set $\Re$ of all possible relational patterns ($\Re(a, b) \subseteq \Re$) whose construction is explained in section 3. To prevent this problem we resort to the logical notion of factorization [19] which is given for clauses but we adapt it to the notion of relational pattern.

Definition 4. *Let P be a relational pattern, which has a non-empty subset $Q \subseteq P$ of unifiable literals with most general unifier (*mgu*) θ. Then $P\theta$ is called a* factor *of P.*

A factor of a pattern P is obtained by applying a substitution θ to P which unifies one or more literals in P, and then deleting all but one copy of these unified literals. In our context, we are interested in particular factors, namely those that are obtained by substitutions θ which satisfy three conditions: *i)*

$Domain(\theta) = \bigcup_{R_k \in \Re(a,b)} Vars(R_k)$ that is, the domain of θ includes all variables occurring in the relational pattern $R_k \in \Re(a,b)$; *ii)* $Domain(\theta) \cap Range(\theta) = \emptyset$, that is, θ renames all variables occurring in the relational pattern $R_k \in \Re(a,b)$ with new variable names; *iii)* $\theta_{|Vars(R_k)}$ is injective, that is, the restriction of θ on the variables occurring in R_k is injective.

For each pattern P, a factor always exists. In the trivial case, it coincides with P up to a renomination of variables in P. A factor $P\theta$ is minimal, when there are no other factors of P with less literals than $P\theta$.

From a logic point of view, $\bigwedge_{R_k \in \Re(a,b)} R_k$ is equivalent to one of its factors since only redundant literals are removed in the factorization process. However, as stated before, working with factors permits to avoid that the probability will approach zero. For this reason, we impose in (2) that $P(a,b|a \prec b) = P(F|a \prec b)$ for any minimal factor F of $\bigwedge_{R_k \in \Re(a,b)} R_k$.

By separating the contribution of the conjunctions of literals corresponding to structural predicates ($struct(F)$) from the contribution of the conjunction of literals corresponding to property predicates ($props(F)$) we have:

$$P(a,b|a \prec b) = P(struct(F)|a \prec b) \times P(props(F)|struct(F) \wedge a \prec b) \qquad (3)$$

Under the naïve Bayes independence assumption:

$$P(struct(F)|a \prec b) = \prod_{rel_j(A,B) \in \ struct(F)} P(rel_j(A,B)|a \prec b), \qquad (4)$$

where $P(rel_j(A,B)|a \prec b)$ is computed on the basis of the relative frequency, computed on the training set, that two tuples are associated according to foreign key constraints given the event $a \prec b$.

The naïve Bayes conditional independence can also be assumed for the computation of $P(props(F)|struct(F) \wedge a \prec b)$, in which case

$$P(props(F)|struct(F) \wedge a \prec b) = \\ = \prod_{attr_j(A,v) \in \ props(F)} P(attr_j(A,v)|struct(F) \wedge a \prec b). \qquad (5)$$

where $P(attr_j(A,v)|struct(F) \wedge a \prec b)$ is computed on the basis of the relative frequency, computed on the training set, that the property predicate is satisfied given $struct(F)$ and $a \prec b$.

3 Patterns construction

The relational pattern discovery is performed by exploring level-by-level the lattice of relational patterns ordered according to a generality relation ($\geqslant$) between patterns. Formally, given two patterns $P1$ and $P2$, $P1 \geqslant P2$ denotes that $P1$ ($P2$) is more general (specific) than $P2$ ($P1$). Hence, the search proceeds from the most general pattern and iteratively alternates the candidate generation and candidate evaluation phases (levelwise method). In [2], the authors propose an enhanced version of the level-wise method [17] to discover patterns from data scattered in multiple tables of a relational database. Candidate patterns are searched in the space of linked relational patterns, which is structured according to the θ-subsumption generality order [18].

Definition 5 (θ-subsumption). *Let $P1$ and $P2$ be two relational patterns on a data schema S. $P2$ θ-subsumes $P1$ if and only if a substitution θ exists such that $P1\ \theta \subseteq P2$.*

Definition 6 (Generality order under θ-subsumption). *Let $P1$ and $P2$ be two relational patterns. $P1$ is more general than $P2$ under θ-subsumption, denoted as $P1 \geqslant_\theta P2$, if and only if $P2$ θ-subsumes $P1$, that is $P1\ \theta \subseteq P2$ for some substitution θ.*

θ-subsumption defines a quasi-ordering, since it satisfies the reflexivity and transitivity property but not the anti-symmetric property. The quasi-ordered set spanned by $\geqslant_\theta$ can be searched according to a downward refinement operator:

Definition 7 (Downward refinement operator under θ-subsumption). *Let $\langle G, \geqslant_\theta \rangle$ be the space of relational patterns ordered according to $\geqslant_\theta$. A downward refinement operator under θ-subsumption is a function ρ such that $\rho(P) \subseteq \{Q \in G | P \geqslant_\theta Q\}$.*

The downward refinement operator is a refinement operator under θ-subsumption. In fact, it can be easily proved that $P \geqslant_\theta Q$ for all $Q \in \rho(P)$. This makes possible to perform a levelwise exploration of the lattice of relational patterns ordered by θ-subsumption. In particular, patterns are discovered by generating the pattern space one level at a time starting from the most general pattern (the pattern that contains only the $preference/2$ predicate) and then by applying a breadth-first evaluation in the lattice of relational patterns ordered according to $\geqslant_\theta$.

When a level of the lattice is explored, the candidate pattern search space is represented as a set of enumeration trees (SE-trees)[20]. The idea is to impose an ordering on atoms such that all patterns in the search space are enumerated. Practically, a node g of a SE-tree is represented as a group comprising: the *head* ($h(g)$) that is the pattern enumerated at g, and the *tail* ($t(g)$) that is the ordered set consisting of the atoms which can potentially be appended to g by ρ in order to form a pattern enumerated by some sub-node of g. A child g_c of g is formed by taking an atom $i \in t(g)$ and appending it to $h(g)$, $t(g_c)$ contains all atoms in $t(g)$ that follow i (see Figure 1). In the case i is a structural predicate (i.e., a new relation is introduced in the pattern), $t(g_c)$ contains both atoms in $t(g)$ that follow i and new atoms that can be introduced only after i has been introduced (according to linkedness property). Given this child expansion policy, without any pruning of nodes or pattern, the SE-tree enumerates all possible patterns and avoids generation and evaluation of candidates equivalent under θ-subsumption to some other candidate.

```
()
+ - - a
|      + - - b
|      |     + - - c
|      + - - c
+ - - b
|      + - - c
+ - - c
```

Fig. 1. The enumeration tree over the atoms $A = \{a, b, c\}$ to search the atomsets a, b, c, ab, ac, bc, abc.

Indeed, we are not interested in all possible patterns, but only in those which satisfy the following property:

$$(supp_{a \prec b}(P) > minSup \vee supp_{b \prec a}(P) > minSup) \wedge$$
$$(GR_{a \prec b}(P) > minGR \vee GR_{b \prec a}(P) > minGR).$$

where:

- $minSup \in [0, 1)$ and $minGR \in [1, +\infty)$ are user defined thresholds
- $supp_{a \prec b}(P)$ represents the support of the pattern P with respect to a preference relation.
- $GR_{a \prec b}(P)$ represents the growth rate computed as $supp_{a \prec b}(P)/supp_{b \prec a}(P)$.

This restriction of the search space permits us to apply different pruning criterion. The monotonicity property of the generality order $\geqslant_\theta$ with respect to the support value (i.e., a superset of an infrequent pattern cannot be frequent) [1] can be exploited to avoid generation of infrequent relational patterns. The monotonicity property does not hold for the growth rate: a refinement of a pattern whose growth rate is lower than the threshold *minGR* may or may not be a pattern with growth rate lower than *minGR*. However, the growth rate can be used for pruning as well. In particular, it is possible to stop the search when it is not possible to increase the growth rate with additional refinements [2]. Finally, as stopping criterion, the number of levels in the lattice to be explored can be limited by the user-defined parameter $MAX_L \geq 1$ which limits the maximum number of predicates in a candidate emerging pattern.

4 Ranking Reconstruction

In our approach, we are able to identify both partial orders and total orders (ranking) over a set of examples. In this paper we consider the case of total orders. In this case, the goal is to build a ranking of reference objects. Formally:
Given: A database with schema $S - PT$ (the same schema used for training)
Find: a total ordering of reference objects in the target table T.

The algorithm follows the proposal reported in [14] and we aim at iteratively evaluating the most promising object to be appended to the resulting ranking.

Algorithm 1 ranking identification algorithm

findranking ($G = \langle V, E \rangle$)**: Ranking L**
L$\leftarrow \emptyset$;
while ($\#L <> \#V$) **do**
  $L.add\left(\arg\max_{b_i \in V/L} SUMPREF_G(b_i)\right)$;
end while

– Let: $G = \langle V, E \rangle$ be a *labeled* directed graph where $V = \{b \in T\}$ and $E = \{(a, b, w_{a,b}) \in V^2 \times [0, 1] | w_{a,b} = P(a \prec b | a, b)\}$ is the set of weighted edges where weights are the probabilities $P(a \prec b | a, b)$ computed according to (1).
– Let: $SUMPREF_G : V \rightarrow [0, \#V]$ be a preference function defined as:
$SUMPREF_G(a) = \sum_{b \in V, b \neq a} w_{a,b}$

Algorithm 1 fully specifies the method for the ranking identification. The rationale is that at each step, an object is added to the final ranking. Such an object is that for which $SUMPREF_G(\_)$ is the highest. Higher values of $SUMPREF_G(\_)$ are given to objects which have a high sum of probabilities to precede other.

5 Experiments

To evaluate the applicability of the proposed approach, we considered the specific domain of document image understanding which denotes the recognition of semantically relevant components and relations in the layout extracted from a document image. This recognition process is based on domain-specific knowledge that can be acquired automatically by applying data mining techniques. In particular, we focus our attention on the specific task of reading order detection [5] where units of analysis are layout components. Determining the reading order for layout components extracted from a document image can be a crucial problem for several applications. In fact, it enables the reconstruction of a single textual element from texts associated to multiple layout components and makes both information extraction and content-based retrieval of documents more effective.

We argue that the spatial dimension of page layout makes a multi-relational data mining approach the most suitable candidate for this specific task.

Reference objects correspond to descriptions of the layout components extracted from document images and are described according to the database schema provided in figure 2 (where the target table is *blocks* and the preference table is not reported). Properties or attributes of layout components are:

– Locational: *x_pos_centre* (*y_pos_centre*): position of the centroid of the layout component w.r.t. the x (y) axis.
– Geometrical: *height*: the height in pixels of a layout component. *width*: the width in pixels of a layout component.
– Logical: "logical label" associated to a layout component.

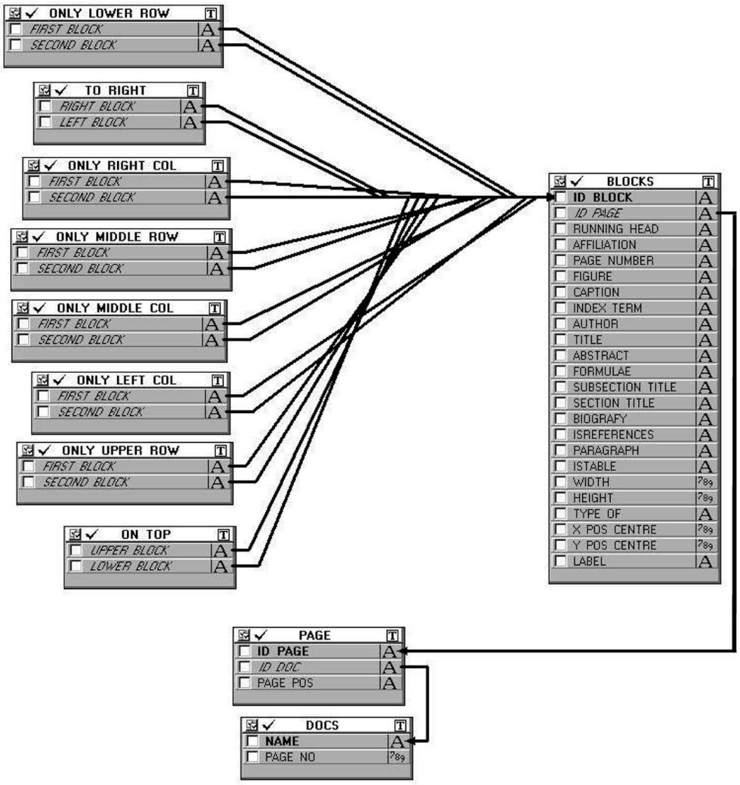

Fig. 2. Logical view of the database schema

– Topological: *on_top*: a layout component is on top/above another layout component. *to_right*: a layout component is to the right of another layout component. *alignment*: defines the type of vertical (col) or horizontal (row) alignment between two layout components. Possible alignments are: right_col, left_col, middle_col, both_columns, middle_row, lower_row,upper_row, both_rows.
– Content type: *type_of*: content type of a layout component. Possible values are: {image, text, horizontal line, vertical line, graphic, mixed}
– Page position: Position of the page in the document. Possible values are: {first, intermediate, last_but_one, last}

For the experiments we considered 24 papers, published as either regular or short articles, in the IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) in two issues of 1996. Each paper is a multi-page document, therefore, we processed 211 document images. Initially, document images were pre-processed by WISDOM++[3] in order to segment them, perform layout analysis, identify the membership class and identify the logical label of a

[3] http:/www.di.uniba.it/%7Emalerba/wisdom++/

Concept	$\prec$ relation		$\prec$ relation in [16]	
	Precision %	Recall%	Precision %	Recall%
FOLD1	76.32	81.44	76.90	64.10
FOLD2	77.24	79.19	74.10	65.20
FOLD3	81.69	83.29	81.00	66.10
FOLD4	77.97	87.63	67.80	56.30
FOLD5	77.26	84.75	78.40	68.70
FOLD6	80.46	85.38	79.40	62.90
AVG	78.49%	83.61%	76.27%	63.88%

Table 1. Precision and Recall results

Algorithm	**Average**	**Standard deviation**
In this paper	0.180	0.03
In [16]	0.491	0.03
In [5]	0.240	0.07

Table 2. 6-fold CV results. Normalized Spearman footrule distance.

layout component. In all, 206 reading orderings were manually specified and 1,629 layout components were involved in such orderings. Possible logical labels for each layout component, in this class of documents, are: {*abstract*, *affiliation*, *author*, *biography*, *formulae*, *index_term*, *reference*, *section_title*, *paragraph*, *subsection_title*, *title*, *caption*, *figure*, *table*, *page_no*, *running_head*}. From those, reading ordering is identified only on {*abstract*, *affiliation*, *author*, *biography*, *formulae*, *index_term*, *reference*, *section_title*, *paragraph*, *subsection_title*, *title*}. Remaining components have not been considered relevant for the reading order.

We evaluated the performance of the proposed approach by means of a 6-fold cross-validation, that is, the dataset of 24 documents was divided into six *folds* and then, for every fold, the learner was trained on the remaining folds and tested on it. Used parameters are $minSup = 0.1$, $minGR = 1.5$ and $MAX_L = 4$.

For each learning problem, statistics on precision and recall were recorded. Such measures refer to the $\prec$ relation. In our case, a reference object a precedes another b when $P(a \prec b|a, b) > P(b \prec a|a, b)$. This permits us to *locally* evaluate the method. To *globally* evaluate the ordering returned by the proposed approach, we resorted to metrics used in information retrieval for the evaluation of the returned rankings [10]. In particular, we considered the *normalized Spearman footrule distance* which, given two complete lists L and L_1 on a set S (L and L_1 are two different permutations without repetition of all the elements in S), is defined as $F(L, L_1) = 2/|S|^2 \sum_{b \in S} abs(pos(L, b) - pos(L_1, b))$ where the function $pos(L, b)$ returns the position of the element b in L.

Results reported in Table 1 permit to compare our approach with the multi-relational approach proposed in [16] on the same dataset with equivalent representations. It is noteworthy that although the proposed approach shows comparable results in terms of precision to those obtained in [16], results in terms of recall are significantly in favour of the present approach. This can be explained by the high degree of adaptivity to noise of probabilistic approaches.

Experimental results concerning the reconstruction of the ranking (reading order) are reported in Table 2. We recall that the lower the distance value, the better the reconstruction of the original ranking. Also in this case, the proposed approach outperforms other approaches. In particular, since the algorithm proposed in [5] is probabilistic and, as in our case, exploits the naive Bayesian learner, it is possible to say that the multi-relational approach is beneficial since it permits to capture the spatial dimension of the document layout.

From a qualitative point of view, 6,229 relational patterns have been in averege automatically extracted for each learning task. Two examples of patterns with high growth rate are reported in the following:

$preference(X,Y), block(X), block(Y), to\_right(X,Z),$
$block\_x\_pos\_centre(Y,[435.1,\ldots,478.0])$ $(supp : 0.1 \quad GR : +\infty)$

This pattern states that a block X that appears to the left of another block (Z) is preferred to a block Y that is approximately located close to the right margin of the document page. The second example of pattern is:

$preference(X,Y), block(X), block(Y), only\_lower\_row(X,Z),$
$block\_biography(X,false), block\_author(X,false), block\_section\_title(X,false)$
$(supp : 0.12 \quad GR : 6.46)$

This pattern includes information on the logical label associated to a block and states that a block X (which represents neither the biography block nor an author block nor a section title block) which is aligned on the bottom margin to another block (Z) is preferred to a block Y.

Patterns with lower growth rate are less interesting from a qualitative point of view, but are still useful for computing probabilities.

6 Conclusions

In this paper we proposed a multi-relational data mining method for ranking complex objects. The method first defines the probability that an object can be preferred to another. The probability is computed by extending the naive Bayes assumption to relational representations of complex objects. Then this probability is used to rank a set of objects. The method has been applied to relational data generated by describing the layout extracted from a set of document images. The aim of the application was that of detecting the correct reading order of layout components. Experimental results prove the advantages of the proposed method with respect to other methods reported in the literature. As future work, we intend to evaluate the proposed approach on other document corpora with a different layout. In addition, we intend to test the proposed method on other problems, such as, the automatic identification of a summary from a text, where the summary is defined as the set of the most relevant (according to a ranking function) sentences in the text, or complex objects retrieval.

Acknowledgment

This work is partial fulfillment of the research objective of ATENEO-2009 project "Estrazione, Rappresentazione e Analisi di Dati Complessi".

References

1. R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. In P. Buneman and S. Jajodia, editors, *International Conference on Management of Data*, pages 207–216, 1993.
2. A. Appice, M. Ceci, C. Malgieri, and D. Malerba. Discovering relational emerging patterns. In R. Basili and M. Pazienza, editors, *AI*IA 2007: Artificial Intelligence and Human-Oriented Computing*, volume 4733 of *LNAI*, pages 206–217, GERMANY, 2007. Springer.
3. C. J. C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. N. Hullender. Learning to rank using gradient descent. In L. D. Raedt and S. Wrobel, editors, *ICML*, volume 119 of *ACM International Conference Proceeding Series*, pages 89–96. ACM, 2005.
4. M. Ceci, A. Appice, and D. Malerba. Emerging pattern based classification in relational data mining. In S. S. Bhowmick, J. Küng, and R. Wagner, editors, *DEXA*, volume 5181 of *Lecture Notes in Computer Science*, pages 283–296. Springer, 2008.
5. M. Ceci, M. Berardi, G. Porcelli, and D. Malerba. A data mining approach to reading order detection. In *ICDAR*, pages 924–928. IEEE Computer Society, 2007.
6. K. Crammer and Y. Singer. Pranking with ranking. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *NIPS*, pages 641–647. MIT Press, 2001.
7. O. Dekel, C. D. Manning, and Y. Singer. Log-linear models for label ranking. In S. Thrun, L. K. Saul, and B. Schölkopf, editors, *NIPS*. MIT Press, 2003.
8. P. Domingos and M. Pazzani. On the optimality of the simple bayesian classifier under Zeo-Ones loss. *Machine Learning*, 28(2-3):103–130, 1997.
9. S. Džeroski and N. Lavrač. *Relational Data Mining*. Springer-V., 2001.
10. C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *WWW '01: Proceedings of the 10th international conference on World Wide Web*, pages 613–622, New York, NY, USA, 2001. ACM Press.
11. Y. Freund, R. D. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:933–969, 2003.
12. N. Helft. *Progress in Machine Learning*, chapter Inductive generalization: a logical framework, pages 149–157. Sigma Press, 1987.
13. R. Herbrich, T. Graepel, and K. Obermayer. *Large margin rank boundaries for ordinal regression*. MIT Press, Cambridge, MA, 2000.
14. T. Kamishima and S. Akaho. Learning from order examples. In *ICDM*, pages 645–648. IEEE Computer Society, 2002.
15. D. Malerba. Learning recursive theories in the normal ilp setting. *Fundam. Inform.*, 57(1):39–77, 2003.
16. D. Malerba and M. Ceci. Learning to order: A relational approach. In Z. W. Ras, S. Tsumoto, and D. A. Zighed, editors, *MCD*, volume 4944 of *Lecture Notes in Computer Science*, pages 209–223, 2007.
17. H. Mannila and H. Toivonen. Levelwise search and borders of theories in knowledge discovery. *Data Mining and Knowledge Discovery*, 1(3):241–258, 1997.
18. G. D. Plotkin. A note on inductive generalization. *Machine Intelligence*, 5:153–163, 1970.
19. J. A. Robinson. A machine oriented logic based on the resolution principle. *Journal of the ACM*, 12:23–41, 1965.
20. X. Zhang, G. Dong, and K. Ramamohanarao. Exploring constraints to efficiently mine emerging patterns from large high-dimensional datasets. In *Knowledge Discovery and Data Mining*, pages 310–314, 2000.