Flexible Matching for Noisy Structural Descriptions

Floriana Esposito[1], Donato Malerba[1], Giovanni Semeraro[2]

(1) Istituto di Scienze deirinformazione - University di Bari
via G. Amendola, 173 - 70126 Bari - Italy

(2) Tecnopolis CSATA Novus Onus
Str. Prov. Casamassima - 70010 Valenzano (BA) - Italy

Abstract

Uncertainty on data often makes the task of perfectly matching two descriptions quite ineffective. In this case, a flexible matching, measuring the similarity of two descriptions rather than their equality, is more useful. According to the convention of connecting similarity to the most common concept of distance, we present a definition of distance measure, based on a probabilistic interpretation of the matching predicate, which can cope with structural deformations. As the problem of matching two formulas of the FOPL is NP-complete, two methods arc presented in order to cope with complexity: firstly, a branch-and-bound algorithm, and secondly, a heuristic method. These ideas are applied to the problem of recognizing office documents in digital form according to their page layout.

1 Introduction

The nature of the problem solving task performed by most expert systems is classification, that is, mapping entities of the world into a set of predetermined solutions or recommendations [Clancey, 1985; Weiss and Kulikowski, 1984]. Typically, expert systems for diagnosis are concerned with selecting an answer from an existing set of diagnoses (solution elements) given the description of a situation. Classification is equally fundamental in nearly all knowledge-based pattern recognition systems, which have to assign appropriate interpretations to objects within a scene [Chandrasekaran and Keuneke, 1987]. Independently from the direction of reasoning, either forward or backward, such systems operate with a description of the current state in the working memory and a description of the conditions to be satisfied in order to select the rule.

Unfortunately, in real applications the descriptions may be both incomplete and also affected by noise. The latter problem is especially felt in those applications in which data are directly detected through sensors or transducers. A scribble on a document or a voice in the background are two common forms of noise. In addition, humans can also introduce errors in the data due to misunderstanding or lack of attention. Another form of noise in a measurement occurs when the measuring instrument shows a poor accuracy. Finally, information may be incomplete due to either human inadequacy or malfunctioning equipment.

When acquiring knowledge from humans, the problem could be solved by multi-expert knowledge acquisition and by applying a cross-validation technique to the rules provided by the experts. In automatic knowledge acquisition the problem is approached by making the machine learning techniques more robust as regards noisy and/or incomplete data [Quinlan, 1986].

Bergadano *et al.* [1988] proposed an approach to learning human concepts which are inherently imprecise and context dependent. The method uses a two-tiered representation of learned concepts and *a flexible matching,* based on a numerical estimation of the typicality or certainty that an instance is a member of a concept, so providing a form of probabilistic inferential extension of a concept In this case, both concept metaknowledge concerning the importance of concept attributes as well as the (joint) probability distributions of these attributes are essential.

To sum up, noisy, imprecise, context-dependent and incomplete descriptions demand a more flexible matching process, also called *partial matching* in [Hayes-Roth, 1979], where two descriptions are compared in order to identify their similarities rather than their equality. Generally, the term *best match* is also used when the rule which maximizes the similarities and minimizes the differences against the current state is selected. The result of a flexible matching should produce a number indicating how well two descriptions match. The number can be a value in the unit interval [0,1] such that 1 indicates a perfect match, 0 no match at all, and any real number r, re (0,1), denotes our confidence in matching. The definition of such a *similarity* measure is strictly connected to the most common concept of *distance,* as the more distant two objects are, the less similar they can be considered.

Several distance measures, or conversely, several similarity measures, have been proposed in the fields of pattern recognition [Sanfeliu and Fu, 1983; Wong and You, 1985; Shapiro and Haralick, 1981] and machine learning [Michalskief a/., 1984; Kodratoff and Tecuci, 1988). They differ in a variety of respects:

- *representation language:* propositional logic, first-order predicate logic, feature vectors, attributed relational graphs;
- *type of problem the distance measures are applied to:* pattern matching in knowledge-based systems, concept acquisition, pattern classification, discriminant analysis, conceptual clustering, numerical taxonomy;
- *theoretical approach:* geometrical, syntactical, probabilistic, entropical, fuzzy, hybrid;
- *type of corrected deformations:* local or structural.

This last point requires further explanation. Generally, an object (or situation) can be decomposed by successive refinements until atomic parts, called *primitives,* are defined. Once these subparts and their mutual relationships are identified, the *structure* is obtained [Stepp, 1987]. The complete description of the object is given by:

- the attributes of the entire structure *(global attributes);*
- the attributes of some subpart *(local attributes);*
- the attributes of the interrelationships between parts *(relations).*

When the differences between the two matching descriptions concern the global/local attributes it is said that *local deformations*occur, while when the differences are at the level of relations then deformations are called *structural.* Not all distance measures take into account structural deformations, particularly those adopting a representation language which does not allow us to represent structural descriptions.

This paper introduces a definition of distance measure suitable for dealing with structural deformations which is based on a probabilistic interpretation of the matching predicate. The three basic characteristics of our definition are: 1) the possibility of dealing with rules whose conditions are not stated as exact descriptions of a particular situation but descri be (complex) properties that the situations must have; 2) the necessity to define, objectively or subjectively, the probability density functions of the features (attributes or relations) used to describe a situation; 3)the possibility of dealing with rules whose conditions are incomplete structural descriptions.

In the following, Section 2 introduces the definition of a flexible matching function for evaluating the goodness of any match between noise-affected structural descriptions. The problem of matching (or unifying) two expressions with commutative and associative operators is NP-complcte [Garey and Johnson, 1979; Siekmann, 1990], moreover the computational cost of a flexible matching procedure increases with the need to calculate the similarity measure. Consequently, we can either try to find algorithms that perform quickly on average or try to find approximate algorithms that produce acceptable answers in an acceptable amount of time. In Section 3 we describe how a branch-and-bound algorithm can be used for reducing the average computational time of the actual similarity between two structural descriptions. Furthermore, for those applications involving complex descriptions and requiring an answer in relatively short time, we discuss the possibility of introducing a heuristic rule which allows us to find an approximate value of similarity. Finally, in Section 4, an application of the proposed distance measure to the recognition of office documents in digital form according to their page layout is illustrated.

2 A distance measure for flexible matching between wff's

Let $\mathcal{S}$ denote the space of all the possible descriptions (or *well formed formulas (wff's)),* complying with the syntax of the representation language and built according to a given vocabulary of attributes and relations. Here we are interested in defining a *flexible matching* function:

$$\text{Flex\_Match}: \mathcal{S} \times \mathcal{S} \rightarrow [0,1]$$

which could be considered as an *extension* of the canonical (strict) matching predicate:

$$\text{Match}: \mathcal{S} \times \mathcal{S} \rightarrow \{\text{false},\text{true}\}.$$

By *extension* we mean that:

$$\forall s,t \in \mathcal{S} \quad \text{Flex\_Match}(s,t) = 1 \Leftrightarrow \text{Match}(s,t)=\text{true}$$
$$\text{Flex\_Match}(s,t) \in [0,1) \quad \text{otherwise}.$$

The function Flex_Match(s,t) represents a degree of similarity between two descriptions $s,t \in \mathcal{S}$, or even the degree *of fitness* of s on t. The definition of such a function should be based on a theory which is able to quantify the degree of similarity between two descriptions. As probability theory fulfils such requirements, we can assign to each pair of wff's in $\mathcal{S}$ the probability of precisely matching the two formulas provided that a change is made in the description t; formally:

$$\text{Flex Match}(s,t) = P(\text{Match}(s,t))$$

Such a definition marks the transition from syntactic to probabilistic matching. Consequently, it is possible to define a probabilistic distance measure, Δ, between s and t as follows:

$$\Delta(s,t) = 1 - \text{Flex\_Match}(s,t) = 1 - P(\text{Match}(s,t))$$

A more detailed definition of distance measure requires a rather more specific description of the representation language than we have given up to now. In particular, the representation formalism we have chosen is inspired to VL_{21} [Michalski, 1980], The basic component of the VL_{21} is the *selector* or relational statement, written as:

[L = R]

where:

- *L,* called *referee,* is a function symbol with its arguments;
- *R,* called *reference,* is a set of values of the referee's domain;

Function symbols, called *descriptors,* are n-adic functions $(n \geq 1)$ mapping onto one of three different kinds of domains: *nominal, linear* and *tree-structured.*

Selectors can be combined by applying different operators, such as *AND* $(\wedge)$, *OR* $(\vee)$ and *decision operator* $(::>)$ in order to define wff's like:

(d-formula) ::> <c-formula)

where *d-formula* is a disjunction of or-atoms (selector conjunctions), while *c-formula* is a conjunction of selectors. This formalism is adequate to express classification rules in many knowledge-based pattern recognition systems dealing with structural descriptions.

Since the main application of theproposed distance measure is noise-affected concept recognition, from now on s will denote the description of a *concept* and t the *observation* to be classified. Moreover, the specializing isomorphism *(s-isomorphism)* [Larson, 1977] rather than the simple isomorphism is used in concept recognition, therefore the match of s and t consists in searching for a substitution a such that:

$$t \Rightarrow \sigma(s)$$

Flex_match is computed according to the following top-down evaluation scheme:

I) s is a disjunction of conjuncts: $s \equiv Or\_atom_1 \vee Or\_atom_2 \vee \ldots \vee Or\_atom_n$. Then the formulation of the flexible matching is given as follows:

$$\text{Flex\_Match}(s,t) = \max_{i \in [1,n]} \text{Flex\_Match}(\text{Or\_atom}_{i}, t) \qquad (1)$$

This definition corresponds to the idea that when a concept is polymorphic (i>l), we are usually interested in finding the "best" matching between one of its morphisms and the observation t. For instance, if $s \equiv$ [length(sl)=10..100] $\vee$ [width(s2)=5..30] and $t \equiv$ [length(sl)=9] $\wedge$ [width(sl)=45], we say that t "nearly" satisfies s simply because it is "near" to the first morphism of s. When correlations occur among the different morphisms expressed by s, the definition above has to be extended so as to take them into account.

II) s is a conjunction of selectors: $\mathbf{s \equiv Sel_1 \wedge Sel_2 \wedge \ldots \wedge Sel_k (k>0)}$. Thus the computation of the flexible matching is affected by the consistent substitution or of the variables in s. As we are looking for the best matching between s and t, we define:

$$\text{Flex Match(M)} = \max_{\sigma_j} \prod_{i=1}^{k} \text{Flex Match}_j(\text{Se}_i\text{t}) \qquad (2)$$

where Flex_Match. denotes the flexible matching function with the tie of the substitutions fixed by σ_j.

III) s is a selector: $\mathbf{s \equiv [f(v_1,v_2,\ldots,v_1)= g_1, g_2, \ldots, g_M]}$, where/ is a 1-adic descriptor and $\{g_1, g_2, \ldots, g_M\}$ is a subset of the domain D of/. Flex_ Match_iSel.,t) is determined by evaluating the degree of similarity between the selector r(s) = Sel. and the corresponding selector of t, $\mathbf{Sel_t \equiv [f(w_1,w_2,\ldots,w_1)= e_1, e_2, \ldots, e_m]}$, which has the same referee as Sel,. Consequently:

Flex Jvlatch (s,t) = Flex„Match(Sel_i,Sel_t) (3)

and Flex_Matcn(Sel_fSel_t) computes the degree of similarity between the references of Sel_f and Sel_t

Since we are searching for an s-isomorphism, the similarity between the references of Sel, and Sel is equal to 1 if and only if the reference of Sel_t is *more specific* than that of Sel_z. The notion of *specialization* is intended as set inclusion, if the descriptor/ is a nominal or linear one. This interpretation can be easily extended to tree-structured descriptors: each single element in the reference of two selectors is replaced by all the values representing the leaves of the subtree having that particular element as its root.

The presence of multiple values in the reference of Sel_t actually means that the value of an attribute is not known exactly, but it ranges over a subset of the attribute domain. This is a form of uncertainty in data [Dubois and Prade, 1988] and its management, together with the problem of incomplete descriptions, has been extensively described in IEsposito *et al.,* 1991a]. Henceforth, we will assume that m=l, that is we are sure about the value *e* taken by / in Sel_t.

Let EQUAL(x,y) denote the matching predicate defined on any two values x and y of the same domain. Since we are looking for the best mapping from {e} into $\{g_,, g_2,\ldots, g_M]$, then the definition of flexible matching depends on the maximum probability of two matching selectors computed over the set of all possible correspondences between the elements of {e} and *[g_1,* $g_2 \ldots, g_M$), that is:

$$\text{Flex\_Match}(\text{Sel}_f,\ \text{Sel}_t) = \max_{i \in [1,M]} \text{P(EQUAL(g.,e))} \qquad (4)$$

Suchadefinition takes into account the goal of classification by means of event covering, thus when ee $\{g_t, g_2,\ldots, g_M)$ then MF(Sel_f, Sel_t) = 1 because there is a perfect matcn, otherwise MF(Sel_f, Sel,) represents the maximum probability that the value in the reference of Sel, equals one of the M values in the reference of Sel_s.

The probability of the event EQUAL(g_1,e) can be defined as the probability that an observation *e* could be a distortion of g., that is:

$$\mathbf{P(EQUAL(g_i,e)) = P(\delta(g_i,X) \geq \delta(g_i,e))} \qquad (5)$$

where:

- *X* is a random variable assuming values in the domain D of /;
- δ is a distance defined on the domain itself.

In other words, the probability that any two values of the domain D match is defined as the probability that a randon variable X defined on D takes a value farther than e from $g_{\{}$ given that $g_{\{}$ is the centroid. In Figure 1, a geometrica interpretation of this definition is provided.

The definition of 6 must take into account the type of VL_2 descriptor. In particular we propose the discrete metrics fo *nominal descriptors:*

$$\delta([\ x\ \begin{cases} 0 & \textit{ifx-y} \\ . \quad y &) \\ 1 & \textit{otherwise} \end{cases} = \quad (6)$$

for *linear not numerical descriptors:*

$$\mathbf{\delta(x,y) = | ord(x)-ord(y) |} \qquad (7)$$

where *ord(x)* denotes the ordinal number given to $\mathbf{x \in D}$, and for*linear numeri*

$$\mathbf{\delta(x,y) = | x-y |} \qquad (8)$$

It should be observed that other reasonable choices of 8 an possible; nevertheless the value of P(EQUAL(g.,e)) does no change since we compute the probability over distance and no merely geometrical distance. This key point also allows us tc ignore problems with scaling when the similarity is computec over the whole set of features.

Of course, the computation of P(EQUAL(g ,e)) must tak< into account the probability density function or X. When nc information is available on the probability distribution of X wc assume X to have a uniform probability distribution, that is:

$$\mathbf{\forall x_i \in D \quad P(X=x_i) = 1/C}$$

for the descriptors with a finite domain (here C is the cardinality of D), while

$$\mathbf{\forall x \in D\ f_D(x) = 1/(b-a)}$$

if the domain D is an interval [a,b] (here f_D denotes the density function).

Having made such assumptions it can be proved that foi nominal descriptors wc have:

$$\text{P(EQUAL}\begin{cases} 1 & \textit{ifgr*} \\ (\ _{g\ i},\ e)) = (9) & \\ \text{(C-l)/C} & \textit{otherwise} \end{cases}$$

while for linear not-num

$$\text{P(EQUAL(g,,e))=} \begin{cases} \mathbf{[1+ord(e)+(C-2ord(g_i)+ord(e))\cdot} \\ \mathbf{\cdot\ step(C-1 - 2ord(g_i)+ord(e))]/C} \\ \qquad\qquad \textit{if } g_i>e \\ 1 \qquad\qquad\qquad \textit{ifg}_t\textit{=e} \qquad (10) \\ \mathbf{[C-ord(e)+(2ord(g_i)-ord(e)+1)\cdot} \\ \mathbf{\cdot\ step(2ord(g_i)-ord(e))]/C} \\ \qquad\qquad \textit{if } g_i<e \end{cases}$$

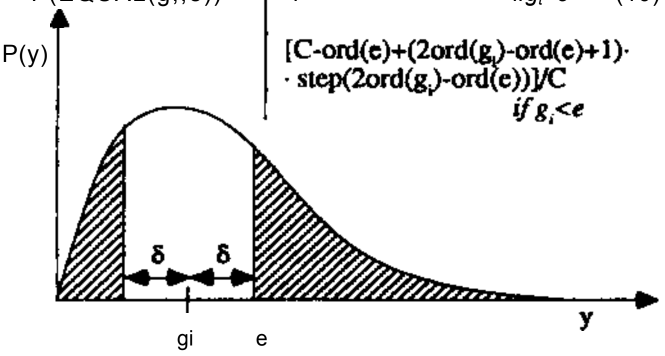


Figure 1. The shaded area represents P(EQUAL(g_i.e)).

where:

$$\mathrm{stcp}(x) = \begin{cases} 0 & \text{if } x<0 \\ 1 & \text{otherwise} \end{cases}$$

A proof of formulas (9) and (10) is given in Appendix A.

For the descriptors with tree-structured domain the computation of P(EQUAL(g.,e)) makes use of the previous formulas. Each element in the references of Sel and Sel_t is replaced by the values representing the leaves of the subtree which has that element as its root. The formulas (9) and (10) are adopted, depending on whether the generalization hierarchy for the descriptor is unordered or ordered, respectively. The only changes to be made both in (9) and (10) consist in replacing C with the number of leaves of the tree-structured domain.

3 Coping with complexity of matching

The computation of the flexible matching when s is a conjunction of selectors requires the evaluation of the maximum conditional probability as in formula (2) as a varies. Unfortunately, if p and q ($p \le q$) are the number of variables in s and t respectively, the number of possible substitutions *a* is given by the permutation of p elements taken from a set of q elements, i.e. P(q,p). Consequently, the computation of Flex_Match(s,t) has a combinatorial cost which should be reduced in some way, particularly when P(q,p) is very large.

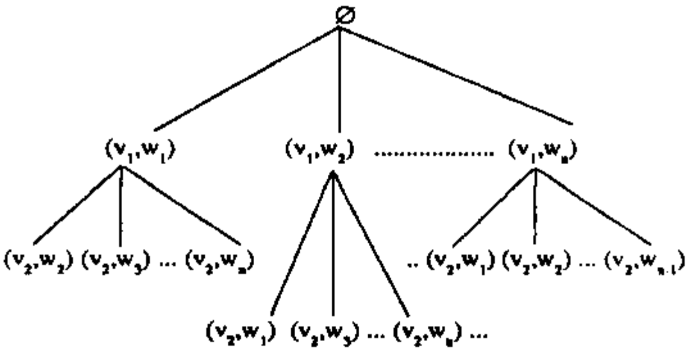
In order to prevent an exponential growth of the computational time, two alternative techniques are presented in the following. Each of them requires that s and t were connected conjunctions of selectors (for a definition of connected formulas see [Larson, 1977]).

Firstly, we can make use of a branch-and-bound algorithm which performs quickly on average. The search space can be represented by a tree where:

- the nodes are variable pairs, $(v.,w_k)$, representing the substitution $v. \leftarrow w_k$ of a variable v. appearing in s with a variable w appearing in t;
- a branch from a node Nl to a node N2 represents the instantiation of a variable of s which has not yet been instantiated in any node along the path from the root to N1.

When all the variables in s have been instantiated, the node of the tree representing the last instantiation can by no means branch (i.e. it is a leaf), and the set of the substitutions along the path from the root represents one possible substitution a (see Figure 2).

Each node of the tree can be labeled with a pair of numbers. The first number represents the partial measure of fitness computed only on those selectors of s whose variables have already been instantiated along the path to the node. The second number represents the exact number of selectors in s which gave a contribution to the computation of the partial measure of fitness. If there is a branch from a node Nl toa node N2 then the value of the partial measure of fitness in N2 must be less or equal to that associated with N1, due to the definition of flexible matching. In other words, walking along a path from the root towards a leaf of the tree, the partial distance measure associated with each node can only decrease or remain the same. A similar (but increasing) monotonic property is also true for the second value reported in node labels. These considerations suggest how a branch-and-bound algorithm can help in finding the best substitution more quickly. In fact, it is sufficient to consider a function cost composed by the partial distance measure and the opposite of the number of selectors in s which contributed to the computation of the partial distance. Minimizing the function cost while the tree is extended allows us to find the best substitution without necessarily exploring all the possible alternatives. When s is a disjunction of or-atoms, the algorithm proceeds exploring alternative consistent instantiations of variables belonging to all the or-atoms, otherwise it could spend too much time trying to evaluate the distance measure concerning a single "bad** or-atom.



∅
(v_1,w_1)
(v_1,w_2)
(v_1,w_n)
(v_2,w_2) (v_2,w_3) ... (v_2,w_n)
(v_2,w_1) (v_2,w_3) ... (v_2,w_n) ...
.. (v_2,w_1) (v_2,w_2) ... (v_2,w_{n-1})

Figure 2. An example of tree explored by the branch-and-bound algorithm.

As second al ternative, it is possible to decompose s into two parts:

$$\mathbf{s \equiv s' \wedge s''}$$

so that:

- $s^* = \mathrm{Sel}_1 > \mathrm{Sel}_2 \wedge \ldots \wedge \mathrm{Sel}_r$, $r \le k$, is a conjunction of selectors such that the referee of Sel, i = 2, 3, ..., r, contains the maximum non-null number of variables not appearing in the referees of Sel_1 $\mathrm{Sel}_2, \ldots ,\mathrm{Scl}_{i-1}$;
- s" is a conjunction of the remaining selectors in s.

The constraint of connection upon s ensures that all the distinct variables in s were in s'. As a consequence, the search for a substitution *a* such that $\mathbf{t \Rightarrow \sigma(s)}$ can be weakened into:

$$\mathbf{t \Rightarrow \sigma(s')} \qquad (11)$$

Under such a hypothesis the events $\mathrm{Flex\_Match}_i(\mathrm{Sel},t)$, i=r+1, r+2, ..., k, become independent since the substitution *a* that verifies (11) has already bounded the variables in s\ As a result, we have:

$$\mathrm{Flex\_Match}(s,t)= \begin{cases} \max\limits_{\sigma_j} \prod\limits_{i=1}^{k} \mathrm{Flex\_Match}_j(\mathrm{Sel}_i,t) & \textit{if there exists at least a} \\ & \sigma_j \textit{ such that: } t \Rightarrow \sigma_j(s') \\ 0 & \textit{otherwise} \end{cases} \qquad (12)$$

This formula must be interpreted as follows: while varying the considered substitution σ_j, the flexible matching between s and t is computed as the highest value given by the product of the degree of similarity between each selector of s" and t.

When it is not possible to find a substitution satisfying (11) then we can set Flex_Match(s,t) = 0, since s and t have no similarities, not even at a level of variables (components). This interpretation corresponds to the heuristic thats' is a conjunction of *Must-relations* [Winston, 1984], thus the computation of (12) is performed only if a perfect matching can be detected between s' and t Sometimes the choice of s' is not unique, in that case a simple preference criterion based on the sum of weights of the selectors in s' may help to select the best alternative.

4 Application to Document Recognition

The flexible matching algorithm has been employed and tested as a part of PLRS, a system for digitized office document recognition based upon the page layout [Esposito *et al„* 1990]. Within the scope of the ODA/ODIF standards [Horak, 1985], a document presents two hierarchical structures: both the *layout* (or geometric) and the *logical* structure. The former concerns the internal organization of the document, i.e. the areas containing text and images, and some components are: set of pages, pages, frames and basic blocks. The logical structure associates the content of a document with a hierarchy of logical components, such as articles, summaries, sections, paragraphs, page numbers, logotypes, and so on. Furthermore, documents can be grouped into classes according to a specific criterion, such as the kind of processing or the common subject.

PLRS classifies single page documents using only on the page layout structure, i.e. the invariant geometrical characteristics shared by documents belonging to the same class, due to underlying printing standards or writing style. An extension of PLRS exploits the results of the document classification process in order to identify the logical components of a document again using the page layout structure. However, this problem, named *document understanding,* is still under study and it will not be dealt with in this paper.

The rules used for the page layout recognition are produced by means of a process of inductive learning, in which some meaningful examples of document classes, relevant for a specific office, are used to train the system. This allows the "in field" customization of the system, thus avoiding the definition of user-handwritten classification rules for a specific office. The form of a recognition rule is:

<condition> ::> <decision>

where:

- *<condition>* is a VL_{21} wff in disjunctive normal form;
- *<decision>* refers to a document class.

The page layout of a document is automatically described in symbolic form, as a VL_{21} conjunctive formula, by a document processing system performing the following steps:

- *preprocessing* of the digitized document;
- *segmentation* into *basic blocks* through the Run Length Smoothing Algorithm (RLS A);
- *layout analysis,* that groups together blocks satisfying some predefined requirements, such as closeness, alignment, and so on, into larger blocks, called *frames,* and produces numerical tables describing each frame;
- *translation* of the numerical tables produced by the previous step into VL_{21} symbolic descriptions.

The descriptors used in the document description are:

CONTAIN_IN_POS(Doc,Block),WIDTH(Block),
HEIGHT(Block),TO.RIGHT(Blockl ,Block2),
ON_TOP(Blockl ,Block2), ALIGN(Blockl ,Block2)

and a page layout description of a training document is reported in the following:

[contain_n_pos(x 1 ,x2)=north] ∧
[contain_in_pos(xl ,x3)=northjwest] ∧
[contain_in_pos(x 1 ,x4)=centre] ∧ [width(x2)=large] ∧
[width(x3)=medium] ∧ [width(x4)=medium_large] ∧
[height(x2)=medium] ∧ [height(x3)=medium_small] ∧
[height(x4)=very_very_large] ∧ [on_top(x2,x3)] ∧
[on_top(x2,x4)] ∧ [to_right(x3,x4)] ∧
[align(x2,x3)=beginning_column] ∧
[align(x2,x4)=last_column] ∧
[align(x3,x4)=beginning_row] ::> [class=three]

The classification of a new document consists of two steps. Firstly the condition part of each recognition rule generated by the learning system is matched against the symbolic description of the new document. Secondly, the document is assigned to the class specified in the decision part of the matching rule.Due to the presence of noise affecting the VL_{21} descriptions of documents, such as a scribble on a document or sensing problems, it is not possible to use a canonical (strict) matching procedure forclassifying test documents, therefore the proposed flexible matching is adopted.

In order to test our approaches to coping with complexity in flexible matching, we organized an experiment in which a set of 72 single page documents, belonging to nine different classes, has been considered. Four classes are letters, each class containing generic printed letters of the same company, while other four classes are magazine indexes; the ninth class is a *reject* class, representing *the rest of the world.* Fifty instances were selected as training examples, leaving the remaining 22 documents for the testing process.

The results of the application of both branch-and-bound algorithm and the heuristic method in the flexible matching procedure applied to the test documents are reported in Table 1 and 2, respectively. In Table 1 entries containing a "*" mean that the value of the flexible matching (FM) is not known since the search has been interrupted. This happens when the partial similarity measure becomes lower than a fixed threshold (0.3 in our experiment). In Table 2 null "FM" values indicate that a strict matching on s' is not possible (see formula (12)). In both tables, an FM value 1.0 in the column denoted by rule indicates the presence of a perfect matching between the test document and the rule generated for the i-th class. The results concerning a full comparison between the canonical matching procedure and the flexible matching have been reported in [Esposito *et al,* 1991b in press].

As we could theoretically expect, entries in Table 1 are never less than the corresponding ones in Table 2, since the branch-and-bound algorithm finds the highest similarity. It should be observed that the classification results do not change at all if the heuristic method is used and the class corresponding to the highest value of similarity is taken as the membership class. The correct class is reported in the first column of Table 2. Both the tables also present the throughput time, expressed in seconds, for each flexible matching and the total time per document (last column) or per class (last row). We can conclude from a comparison of these time entries that the branch-and-bound method needs much more time than the heuristic method, and this is a great limitation for a real-time document handling system.

5 Conclusions

In the paper a definition of a flexible matching *is* presented: it is based on a probabilistic interpretation of the matching predicate and proves useful to cope both with noisy data and with structural deformations. Unfortunately, computing the

Table 1
Classification results using the Branch-and-Bound algorithm

Ex		rule$_1$	rule$_2$	rule$_3$	rule$_4$	rule$_5$	rule$_6$	rule$_7$	rule$_8$	tot
1	T	1	2	4	1	0	9	25	12	49
	FM	1.0	$0.4\overline{09}$	*	$0.\overline{80}$	$0.\overline{45}$	*	$0.4\overline{90}$	*	
2	T	1	1	4	1	2	7	13	2	32
	FM	1.0	0.431	*	$0.\overline{81}$	$0.\overline{45}$	0.28	$0.4\overline{90}$	*	
3	T	6	1	2	1	1	8	16	3	39
	FM	$0.\overline{45}$	1.0	*	0.8	$0.\overline{45}$	0.3	$0.4\overline{90}$	*	
4	T	1	1	2	1	1	2	7	2	19
	FM	$0.\overline{40}$	1.0	0.09	$0.\overline{72}$	0.5	*	$0.\overline{72}$	*	
5	T	2	0	1	1	1	0	1	1	8
	FM	0.8	*	1.0	0.8	0.5	*	0.9	*	
6	T	3	1	0	1	0	2	0	1	9
	FM	0.8	0.4	$0.\overline{90}$	$0.\overline{81}$	0.5	$0.\overline{28}$	0.9	0.36	
7	T	3	0	4	1	0	10	35	4	59
	FM	$0.5\overline{81}$	0.8	0.09	$0.\overline{81}$	$0.\overline{90}$	$0.\overline{45}$	$0.\overline{72}$	0.32	
8	T	3	2	10	0	2	17	2	15	53
	FM	$0.\overline{58}$	0.8	0.29	1.0	1.0	$0.\overline{45}$	$0.\overline{90}$	*	
9	T	5	5	22	1	1	2	4	8	50
	FM	$0.\overline{64}$	0.45	*	$0.\overline{90}$	1.0	1.0	$0.\overline{88}$	0.81	
10	T	4	1	7	1	0	10	24	22	71
	FM	1.0	$0.4\overline{09}$	0.826	0.90	1.0	0.5	$0.\overline{81}$	0.45	
11	T	6	3	10	1	0	5	25	13	65
	FM	0.5	$0.4\overline{09}$	*	1.0	1.0	0.8	$0.\overline{72}$	0.45	
12	T	6	3	5	2	1	1	21	5	46
	FM	$0.\overline{72}$	$0.\overline{72}$	*	$0.\overline{72}$	0.5	1.0	$0.\overline{81}$	0.72	
13	T	7	1	4	2	1	1	31	18	67
	FM	$0.4\overline{36}$	$0.\overline{72}$	*	$0.\overline{72}$	$0.\overline{45}$	1.0	$0.\overline{64}$	0.40	
14	T	6	2	3	0	1	2	1	2	19
	FM	$0.\overline{88}$	$0.\overline{48}$	*	0.6	0.5	0.5	$0.\overline{90}$	0.72	
15	T	6	0	2	1	1	1	2	1	15
	FM	$0.\overline{72}$	$0.\overline{32}$	$0.\overline{36}$	$0.\overline{72}$	0.5	$0.\overline{32}$	0.9	0.4	
16	T	5	2	2	1	2	3	5	3	25
	FM	$0.\overline{80}$	$0.\overline{81}$	*	$0.\overline{72}$	0.5	0.25	1.0	0.36	
17	T	8	3	4	3	2	3	4	1	29
	FM	$0.\overline{80}$	0.45	*	$0.\overline{72}$	$0.\overline{54}$	$0.\overline{45}$	1.0	0.36	
18	T	9	3	7	2	1	5	33	17	72
	FM	$0.\overline{90}$	$0.\overline{72}$	*	$0.\overline{72}$	$0.\overline{90}$	$0.\overline{44}$	0.581	1.0	
19	T	13	4	9	3	0	10	25	9	75
	FM	$0.\overline{90}$	$0.\overline{72}$	*	$0.\overline{72}$	$0.\overline{90}$	$0.\overline{44}$	0.581	1.0	
20	T	15	4	5	4	1	7	11	7	55
	FM	1.0	$0.\overline{72}$	*	0.8	$0.\overline{45}$	$0.\overline{40}$	0.58	1.0	
21	T	13	2	4	5	1	2	6	2	36
	FM	$0.\overline{60}$	0.8	*	$0.5\overline{81}$	$0.\overline{54}$	0.45	$0.3\overline{27}$	0.45	
22	T	8	2	1	4	1	1	10	7	35
	FM	$0.\overline{72}$	$0.\overline{72}$	*	0.6	0.5	$0.\overline{44}$	$0.3\overline{27}$	0.72	
Tot.		133	45	115	39	22	112	304	158	928

Table 2
Classification results using the heuristic on matching

Cl.		rule$_1$	rule$_2$	rule$_3$	rule$_4$	rule$_5$	rule$_6$	rule$_7$	rule$_8$	tot
1	T	0	0	1	0	0	0	1	0	4
	FM	1.0	$0.3\overline{87}$	0.0	$0.\overline{80}$	$0.\overline{45}$	0.0	$0.4\overline{90}$	0.0	
1	T	0	0	0	0	0	0	1	0	2
	FM	1.0	0.431	0.0	0.0	$0.\overline{45}$	0.0	$0.4\overline{90}$	0.0	
2	T	0	0	2	0	1	1	0	2	8
	FM	0.0	1.0	0.0	$0.\overline{60}$	$0.\overline{45}$	0.0	$0.4\overline{90}$	0.0	
2	T	0	0	0	1	0	0	0	0	3
	FM	0.0	1.0	0.0	0.0	$0.\overline{27}$	0.0	$0.1\overline{63}$	0.0	
3	T	0	0	0	0	0	0	0	0	1
	FM	0.0	$0.2\overline{58}$	1.0	$0.\overline{40}$	$0.\overline{45}$	0.0	0.9	0.0	
3	T	0	0	0	1	0	0	0	1	4
	FM	0.0	0.4	$0.\overline{90}$	$0.\overline{44}$	$0.\overline{45}$	0.0	0.9	0.0	
4	T	1	0	3	1	0	2	1	1	9
	FM	0.0	0.8	0.054	0.0	$0.\overline{90}$	0.0	$0.\overline{72}$	0.0	
4	T	2	0	4	0	0	1	2	1	11
	FM	$0.\overline{08}$	0.8	0.036	1.0	1.0	0.0	$0.\overline{90}$	0.0	
5	T	2	1	4	0	0	1	2	2	12
	FM	$0.\overline{64}$	$0.\overline{32}$	$0.\overline{36}$	$0.\overline{90}$	1.0	1.0	0.65	0.0	
5	T	0	0	3	1	0	2	2	2	11
	FM	1.0	0.387	0.826	$0.\overline{90}$	1.0	0.0	$0.\overline{81}$	0.225	
5	T	1	0	2	0	0	2	1	2	9
	FM	$0.\overline{48}$	$0.\overline{36}$	0.198	1.0	1.0	$0.\overline{88}$	0.65	0.0	
6	T	1	0	3	0	0	0	1	1	6
	FM	$0.\overline{36}$	$0.\overline{72}$	0.198	$0.\overline{48}$	$0.\overline{36}$	1.0	$0.\overline{81}$	0.0	
6	T	1	0	2	0	0	1	1	1	7
	FM	$0.\overline{36}$	$0.\overline{72}$	0.198	$0.\overline{48}$	$0.\overline{18}$	1.0	0.245	0.0	
6	T	0	0	0	0	0	1	0	1	3
	FM	$0.\overline{88}$	$0.\overline{48}$	0.09	0.0	$0.\overline{45}$	0.0	$0.\overline{90}$	0.0	
7	T	0	0	0	0	0	0	0	0	1
	FM	0.0	$0.\overline{32}$	0.0	0.0	$0.\overline{36}$	0.0	0.9	0.0	
7	T	0	0	0	0	0	1	0	1	3
	FM	$0.\overline{80}$	$0.\overline{81}$	0.129	$0.\overline{60}$	$0.\overline{36}$	0.0	1.0	0.0	
7	T	0	0	0	0	0	1	0	1	3
	FM	$0.\overline{80}$	$0.\overline{32}$	0.162	$0.\overline{72}$	$0.\overline{54}$	0.0	1.0	0.0	
8	T	1	0	2	0	0	2	1	1	8
	FM	$0.\overline{90}$	$0.\overline{72}$	0.297	$0.\overline{64}$	$0.\overline{90}$	0.0	$0.2\overline{45}$	1.0	
8	T	1	1	2	0	0	2	2	1	10
	FM	$0.\overline{90}$	$0.\overline{72}$	0.297	$0.\overline{64}$	$0.\overline{90}$	0.0	$0.2\overline{45}$	1.0	
8	T	0	1	0	0	0	1	0	1	4
	FM	1.0	$0.\overline{72}$	0.297	$0.\overline{40}$	$0.\overline{45}$	0.0	$0.2\overline{45}$	1.0	
R	T	2	0	4	1	0	1	2	2	13
	FM	$0.\overline{60}$	0.8	0.240	$0.\overline{54}$	$0.\overline{54}$	0.0	0.327	0.0	
R	T	0	0	0	0	0	0	0	0	1
	FM	0.0	$0.\overline{48}$	0.148	0.0	$0.\overline{36}$	0.0	0.0	0.0	
Tot.		15	6	35	8	3	22	20	24	133

similarity of two descriptions is computationally impractical, therefore two distinct methods are adopted to reduce the complexity: firstly, branch-and-bound algorithm, and secondly, a heuristic method. The flexible matching has been applied to the recognition of digitized office documents and the results of both the algorithms are presented.

A Proof of formulas (9) and (10)

Let us recall the definition (5) given above:

$$P(EQUAL(g_i,e)) = P(\delta(g_i,X) \geq \delta(g_i,e)) \qquad (IB)$$

Henceforth, in order to simplify our notation, we will use g instead of g_i As already said, formula (IB) takes into account both the type of domain which g and e belong to and the probability distribution of the domain values.

By assuming that the probability distribution is uniform, and remembering the definition of 5 for nominal domains, we have:

$$P(EQUAL(g,e)) = \begin{cases} P(\delta(g,X) \geq 0) & \textit{if } e=g \\ & \textbf{(2B)} \\ P(\delta(g,X) \geq 1) = (C-1)/C & \textit{if } e \neq g \end{cases}$$

where C is the number of elements of the domain.

For ordinal domains, (IB) becomes:

$$P(EQUAL(g,e)) = P(|ord(g)-ord(X)| \geq |ord(g)-ord(e)|) =$$

which can be rewritten in a simpler form by denoting ord(g), ord(e) and ord(X) with g,e, and X, respectively:

$$= P(|g-X| \geq |g-e|) \tag{3B}$$

1st case: g=e

$$P(EQUAL(g,e)) = P(|g-X| \geq 0) = 1 \tag{4B}$$

2nd case: g > e

$$\begin{aligned}
&P(|g-X| \geq |g-e|) = P(g-X<e-g \cup g-X=g-e \cup g-X>g-e \cup g-X=e-g) = \\
&= P(g-X<e-g) + P(g-X=g-e) + P(g-X=e-g) + P(g-X>g-e) = \\
&= P(X>2g-e) + P(X=e) + P(X=2g-e) + P(X<e) = \\
&= P(X\geq 2g-e) + P(X\leq e) = \\
&= [(C-2g+e)\cdot step(C-1-2g+e) + e+1]/C \qquad (5B)
\end{aligned}$$

3rd case: g < e

$$\begin{aligned}
&P(|g-X| \geq |g-e|) = P(g-X<g-e \cup g-X=e-g \cup g-X>e-g \cup g-X=g-e) = \\
&= P(g-X<g-e) + P(g-X=e-g)+ P(g-X>e-g) + P(g-X=g-e) = \\
&= P(X>e) + P(X=2g-e) + P(X<2g-e) + P(X=e) = \\
&= P(X\leq 2g-e) + P(X\geq e) = \\
&= [(2g-e+1)\cdot step(2g-e) + C - e]/C \qquad (6B)
\end{aligned}$$

where step(x) is the following function:

$$step(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{otherwise} \end{cases}$$

Finally, resubstistuting ord(g) and ord(e) to g and c, respectively, we have formula (10).

References

[Bergadano *et al.,* 1988] Francesco Bergadano, Stan Matwin Ryszard S. Michalski, and Jianping Zhang. Representing and Acquiring Imprecise and Context-dependent Concepts in Knowledge-based Systems. In Zbigniew.R. Ras, and Lorenza Saitta, (Eds.) *Methodologies for Intelligent Systems, 3,* pages 270-280, Amsterdam, The Netherlands, Elsevier Science Publishers B. V., 1988.

[Chandrasekaran and Kcuneke, 1987] Bruce Chandrasekaran, and Anne Kcuneke. Classification problem solving.A tutorial from an AI perspective. In Pierre A.Devijver, and Josef Kittler (Eds.) *Pattern Recognition Theory and Applications,* Berlin, Germany, Springer-Verlag,1987.

[Clancey, 1985] William J. Clancey. Heuristic Classification. *Artificial Intelligence.* 27(4):289-350,1985.

[Dubois and Prade, 1988] Didier Dubois, and Henry Prade. An Introduction to Possibilistic and Fuzzy Logics. In Philippe Smets, E. H. Mamdani, Didier Dubois, and Henry Prade (Eds.) *Non-Standard Logics for Automated Reasoning,* pages 315-316, London, England, Academic Press, 1988.

[Esposito *et al.,* 1990] Floriana Esposito, Donato Malerba, Giovanni Semcraro, Enrico Annese, and Giovanna Scafuro. Empirical Learning Methods for Digitized Document Recognition: an Integrated Approach to Inductive Generalization. In *Proceedings of the Sixth IEEE Conference on Artificial Intelligence Applications,* pages 37-45, Santa Barbara, California, March 1990.

[Esposito *et* al., 1991a] Floriana Esposito, Donato Malerba, and Giovanni Semeraro. Classification of incomplete structural descriptions using a probabilistic distance measure. To appear in *Proceedings of the International Conference on Symbolic-Numeric Data Analysis and Learning,* Paris, France, September 1991.

[Esposito *et al,* 1991b in press] Floriana Esposito, Donato Malerba, and Giovanni Semeraro. Classification in Noisy Environments Using a Distance Measure Between Structural Symbolic Descriptions. To appear *inIEEETrans. on Pattern Analysis and Machine Intelligence,* 1991.

[Garey and Johnson, 1979] Michael R. Garey, and David S. Johnson. *Computers and Intractability,* page 252, San Francisco, California, W.H. Freeman & Co., 1979.

[Horak, 1985] Wolfgang Horak. Office Document Architecture and Office Document Interchange Formats: Current Status of International Standardization. In *IEEE Computer,* 18(10):50-60, October 1985.

[Kodratoff and Tecuci, 1988] Yves Kodratoff, and Gheorghe Tecuci, Learning Based on Conceptual Distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* PAMI-10(6):897-909, November 1988.

[Larson, 1977] James B. Larson, Inductive Inference in the Variable Valued Predicate Logic System VL_{21}: Methodology and Computer Implementation. Doctoral dissertation, Dept of Computer Science, University of Illinois, Urbana, Illinois, May 1977.

[Michalski, 1980] Ryszard S. Michalski. Pattern Recognition as Rule-Guided Inductive Inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* PAMI-2(4):349-361, July 1980.

[Michalski *et al.,* 1986] Ryszard S. Michalski, Ivan Mozetic, J. Hong, and Nada Lavrac. The AQ15 Inductive Learning System: An Overview and Experiments. Intelligent Systems Group, Dept. of Computer Science, University of Illinois, Urbana, Illinois, 1986.

[Quinlan, 1986] J.Ross Quinlan. Induction of Decision Trees. *Machine Learning,* 1(1):81-106,1986.

[Sanfcliu and Fu, 1983] Alberto Sanfeliu, and King Sun Fu. A distance measure between attributed relational graphs for Pattern Recognition. *IEEE Trans, on Systems, Man, and Cybernetics,* SMC-13(5):353-362, May-June 1983.

[Shapiro and Haralick, 1985] LindaG. Shapiro, and Robert H. Haralick. Structural descriptions and inexact matching. *IEEE Transactions Pattern Analysis and Machine Intelligence,* PAMI-3(5):504-519, September 1981.

[Siekmann, 1990] Jorg H. Siekmann. An Introduction to Unification Theory. In Ranan B. Banerji (Ed.) *Formal Techniques in Artificial Intelligence: A Sourcebook,* pages 369-424, Amsterdan, The Netherlands, Elsevier Science Publishers B. V., 1990.

[Stepp, 1987] Robert E. Stepp, Machine Learning from Structured Objects. *Proceedings of the Fourth International Workshop on Machine Learning,* pages 353-363, Irvine, California, 1987.

[Weiss and Kulikowski, 1984] Sholom M.Weiss, and Casimir Kulikowski. *A Practical Guide to Designing Expert Systems.* Totowa, New Jersey, Rowman and Allanheld, 1984.

[Winston, 1984] Patrick Henry Winston, *Artificial Intelligence (2nd Ed.),* pages 391-414, Reading, Massachusetts, Addison-Wesley, 1984.

[Wong and You, 1985] Andrew K.C. Wong, and Manlai You. Entropy and distance of random graphs with application to structural pattern recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence,* PAMI-7(5):599-609,1985.