Mathware & Soft Computing 6 (1999) 277-291

Inference of Fuzzy Regular Grammars from Examples*

I. Fortes[+], R. Morales, J.L. Pérez de la Cruz, F. Triguero, M.A. Comino
[+]Dept. Matemática Aplicada, Dept. Lenguajes y Ciencias Comp.
E.T.S.I. Informática. Campus Teatinos. Univ. Málaga (Spain).
ifortes@ctima.uma.es, morales@lcc.uma.es

Abstract

Let us consider the following situation: An oracle provides us with a finite set of examples considered as words belonging to a regular language. This oracle is not available again. In this paper we study a new and general inference algorithm of fuzzy regular grammars based on this set of words. This algorithm is created by adapting a process discovery method. The main issues in the adaptation are the development of a fuzzy version, the assignation of membership degrees to each production in the grammar, and the treatment of consecutive repeated symbols. In addition to this inference algorithm we present a practical use for automatically generating artistic designs. Specifically, we have collected a set of paintings by Piet Mondrian (1872-1944) and obtained new Mondrian-style paintings. To achieve this, we designed a code to transform the paintings into strings and also to carry out the reverse conversion. We view these strings, which represent the paintings, as words belonging to a regular language and from this finite set of examples infer a fuzzy regular grammar. The entire process has been implemented and some new paintings from the inference algorithm have been obtained. An art expert has judged that these computer- generated paintings are fully in the spirit of those painted by Mondrian.
Keywords: Fuzzy Regular Grammars, Process Discovery, Fuzzy Relations, Computational Learning.

1 Introduction

Computational Learning tries to reproduce human learning in the sense of being able to learn concepts in a natural way. In addition, humans are able to adapt their learning to different situations depending on the availability of the necessary information. So, learning can be exact or approximate. Learning can also be considered in an open or closed context. In the open context a tutor interacts with the learner

*This work has been partially supported by project FACA number PB98-0937-C04-01 of the CICYT-Spain. FACA is a part of FRESCO project.

during the entire learning process. In the closed context there is no external help. One of the first models of computational learning was the PAC model (Probably Approximately Correct) introduced by Valiant [27] with only membership queries. Afterwards, Angluin proved that the DFA (deterministic finite automata) class can be learned and he introduced the exact computational learning model [2] (model with membership and equivalence queries) [3]. Later, Kearns and Vazirani [19] refined the learning algorithm for the DFA by means of classification trees and Balcazar et al. [6] generalized the method.

All these models can be considered in an open context because the oracle is present during the entire learning process. In this paper a more realistic situation is considered, namely that of approximate fuzzy closed learning. The only available information is a finite set of examples and no new elements can be obtained. These examples have subjective features and therefore certain ambiguity. Our goal is to infer a model that fits the data suitably. Specifically, the case where the elements can be considered as words belonging to a regular language. Therefore, a fuzzy regular grammar must be inferred from the examples in order to verify the examples of the sample and generate new words of the language. We consider the use of fuzzy regular grammars to be more appropriate because the concept obtained from the examples is vague.

Various approaches to this problem have been studied in [1, 8, 9]. The DFA can be viewed as a tree that evolves from an initial state and remains in the same state whenever examples provided have the same prefix. When prefixes become different, the DFA has reached a bifurcation. In particular, Biermann and Feldman [7] try to infer a grammar by studying the behavior of the tail of length k of the present state. Also, this kind of problem has been studied from a fuzzy point of view. A general model to obtain deterministic acceptors of regular fuzzy languages is developed in [26].

The model presented here looks both at past and future behavior in order to define a state. As a first step, a deterministic algorithm that focuses on process discovery was presented by Cook and Wolf [10]. This algorithm just looks for the existence of fixed length sequences of events in the sample. In this paper a new, fuzzy version of the algorithm is presented and properties had been not dealt with the original algorithm are studied. In order to achieve this goal, we assign a membership degree to each production in the grammar. Moreover, the new version of the algorithm takes into account consecutive occurrences of the same symbol. In this way, the features of the events are more accurately detected. The fuzzy model presented here has been applied to a practical case and implemented on the Internet [30]: the generation of Mondrian-like paintings. The features of Piet Mondrian's paintings (1872-1944) have been inferred and extracted into a fuzzy regular grammar which, in turn, can be used to generate the basic structures of new paintings. Each structure has an associated membership degree. A final transformation phase provides paintings within Mondrian's style with a realistic shape and size. The membership degree associated to a final painting coincides with the degree of its structure. The membership degree associated with each new painting is the level of credibility of the painting being considered within Mondrian's style. In our opinion, fuzzy sets are the right conceptual tool to describe the "degree of adjustment" of an obtained

structure to a given painting style -which is a fuzzy concept-, so fuzzy grammars can be useful to describe and generate paintings of this kind.
In the following section we introduce some notation and previous concepts. In section 3 the inference algorithm is described. In the following section Mondrian's paintings and style are described and so is the algorithm for transforming paintings into strings and vice versa. In the same section we present some new paintings obtained by the system and how the grammar is employed to verify and generate paintings. Finally, some conclusions are drawn from the work done.

2 Previous Concepts

We now review some basic notions that will be used in the next section; a more detailed development can be seen in [13, 17, 18, 25].

2.1 Basic Notions

We give some basic notions about formal fuzzy language. Informally a fuzzy grammar may be viewed as a set of rules for generating the elements of a fuzzy set. More precisely,

Definition 1 (Fuzzy Right Regular Grammar) *A fuzzy grammar is a quadruple $\mathcal{G} = (N, V, P, A)$ where N is a nonempty finite set or terminals of the alphabet Σ, V is a set of nonterminals ($N \cap V = \emptyset$), P is a finite subset of $(N \times V \times N) \cup (N \times V)$ and A is the initial symbol. The elements $t = (A_i, a_k, A_j)$, $r = (A_i, a_k)$, of P are called rules and they are expressions of this form $\mu_P(t) : A_i \to a_k A_j, \mu_P(r) : A_i \to a_k$ where m is the membership degree.*

Definition 2 (Membership degree of a sequence) *Let be $\alpha_i, \quad i = 1, \dots, m$, and x sequences of $(N \cup V)^*$. Let i be a triangular norm and u its co-norm, then the membership degree $\mu_G(x) = u(i(\mu(A \to \alpha_1), \mu(\alpha_1 \to \alpha_2), \dots, \mu(\alpha_m \to x)))$*

Definition 3 (Generated Fuzzy Language) *A fuzzy grammar $\mathcal{G}$ generates a fuzzy language $L(\mathcal{G})$. A string x of V^* is said to be in $L(\mathcal{G})$ if and only if x is derivable from A.*

We also use deterministic finite automata [18]. A deterministic finite automata can be represented by its state diagram which is a directed graph where the states are represented by vertices, and the edges are labeled with symbols of the alphabet.

2.2 Grammar Inference

The grammar inference problem is usually informally stated as follows: given some sample sentences in a language, and perhaps some sentences specifically not in the language, infer a "good" grammar that represents the language. More formally, we can define the problem in the following terms:
Let Σ be an alphabet and Σ^* the set of all possible sequences (sentences) in Σ. Let

$L \subset \Sigma^*$ be a language. Let $\mathcal{G}$ be a grammar that describes and generates L.
We define S_L as the infinite set of pairs $(s, l) \in \Sigma^* \times \{0, 1\}$ where $l = 1$ if $s \in L$ and $l = 0$ otherwise. S_L^+ is defined as the set of pairs for words belonging to L, and S_L^- is the set for words not belonging to L. So, $S_L = S_L^+ \cup S_L^-$.
Therefore, given some presentation of all or part of S_L, can one infer a good grammar $\mathcal{G}$ describing L?
In the case studied here, we have a finite subset S (sample) of S_L^+, $S = \{(s_k, 1) \in \Sigma^* \times \{1\}; \quad 1 \leq k \leq m\}$ where its elements are considered as words belonging to a regular language.

2.3 Process Discovery

Process discovery is essentially the analysis of data captured from a process in order to generate a formal model of the process behavior that it describes. All the effort is concentrated on the behavioral aspects of the process.
In the framework of the process an *event* is used to characterize the dynamic behavior of a process in terms of identifiable, instantaneous actions. The overlapping activities of a process are represented by a sequence of events, which we refer to as an *event sequence*.
Considering these in terms of grammar inference, the events $e_i \in s_k, \quad i = 1, \ldots, n$ are interpreted as terminals, and event sequences $s_k \in S$ as strings in the language of the process. Note that each e_i can belong to more than one s_k.
A good result is one that produces a model that reflects the inherent structure of the sample. Often the sample does not reflect all the possible events that can be generated. This is due to the fact that collecting the sample data can be difficult and expensive. In addition, the data can be limited (as is the case with Mondrian's paintings).

3 The Inference Algorithm

A process discovery method was developed in a deterministic algorithm by Cook and Wolf [10]. Now, we develop a new version adapted to fuzzy regular grammars. Given an event sequence sample our inference algorithm generates a fuzzy regular grammar. We focus on event sequences of length two and three. However, there are some problematic situations if the strings xy and yz exist in the examples, because then the algorithm initially allows the string xyz. If the string xyz does not appear in the examples, then the algorithm destroys this overconnectivity. This is the way in which the algorithm reflects the past and future behaviour.
We have adapted the algorithm to assign membership grades to event sequences. Also, consecutive repetition of symbols has been taken into account. We obtain a deterministic finite automata which is transformed into a fuzzy regular grammar. We assign membership degree to the rules and in this way the features of the grammar feature are more precisely gathered. The algorithm has been implemented at all and a version is available on the Web [30].
The algorithm proceeds in six steps. In this section the algorithm is presented

without examples. To smooth this difficulty, each step is explained in detail. Its application to the practical case of Mondrian's paintings is found in the following section.

3.1 The Inference Algorithm

1. Definition of two fuzzy binary relations

   (a) Let E be all the events e_i belonging to the sample $S \subset S_L^+$. The fuzzy binary relation $R(E, E)$ is defined as "$-near \quad to-$" with membership function $\mu_R(e_i, e_j)$. This function amounts to the degree of belief in the fact that event e_i will be followed by the event $e_j \quad i = 1, \ldots, n, \quad j = 1, \ldots, n$.

   (b) Let D be the set of event sequences of length 2 whose membership degree is in R nonzero. The fuzzy binary relation $T(D, E)$ is defined as "$-near \quad to-$" with membership function μ_T. This function indicates the degree of belief in the fact that an event sequence of length 2 will be followed by an event $e_i \quad i = 1, \ldots, n$. To compute the membership degree of each event sequence of length 3 a t-norm is used. To pick up the number of these event sequences the corresponding co-norm is used to accumulate membership degrees.

2. Construction of the event graph G
   After fixing a confidence level, the directed graph G is constructed from the matrix associated with R by the following method. Each event is assigned to a vertex. Then, for each event sequence that exceeds the confidence level, a uniquely labeled edge is created from an event in the sequence to the immediately following event in that sequence. The edges are labeled with natural numbers.
   Each event sequence of the sample can be considered as a path in the graph G. The following condition must be satisfied: each path must be in same region of the graph.

3. Remove overconnectivity
   The previous step can produce overconnected vertices and therefore illegal sequences (event sequences of length three $e_i e_j e_p$ with membership degree zero in the matrix associated with T). Overconnectivity is removed as follows: vertex e_j is split into two vertices, one having an edge to a vertex at the extreme of the illegal sequence, and the other duplicate vertex e_j having an edge from the other extreme to it. The rest of the edges are organized in this way. For each duplicate vertex the same edges that were there with the rest of the vertices are considered again except for the two vertices e_i, e_p that appear in the extremes of the illegal sequence. All these edges are labeled with the same number they had. Repeat this process to obtain all the legal sequences and no illegal sequence in the graph G.

4. Conversion of G into its dual G'
Each edge in G is a vertex in G' labeled with the label of the edge in G. For each pair of in-edge b and out-edge f of a vertex e_j in G two vertices are created in G', labeled with the numbers of the in-edge and out-edge of G (b and f) and also an edge e_j from b to f.
If a vertex e_i in G has just an out-edge b, then an edge is created in G' from a vertex labeled by A (initial state) to the vertex b in G'. The edge in G' is labeled by e_i.
If a vertex e_p in G has just an in-edge f, then a vertex f and an edge labeled by e_p from f to a vertex F (final state) is created in G'.

5. Transformation of the automaton G' into a regular grammar G''
The automaton G' is transformed into a regular grammar in the usual way [18] and for every vertex in G' there will exist a nonterminal symbol in G''; let us denote it with the letter A subindexed by the number of the vertex. For example, A_b or A_f. On the other hand, when writing down the productions of G'' it must be taken into account the fact that there could be additional transitions arising from the initial state A. This is a consequence of the existence of states that can be visited more than once with in-edge u, from which the DFA G must start in order to obtain consistency with all the beginnings of event sequences of S. In this case productions $A \rightarrow \varepsilon A_u$ are added to G''. If all these states are states that can be visited only once, then only the usual productions of the form $A \rightarrow e_i A_b$ are considered in G''.
All states in G' with an in-edge labeled by an event that is consistent with the end of the event sequence of S are final states for this edge. The vertex F of G' is a final state, too. When G' is transformed into the grammar G'', for every state f with transitions to F through e_p only productions in the form $A_f \rightarrow e_p$ will be considered.

6. Fuzzy Regular Grammar: assignation of membership degrees to the productions of G''
A membership degree is assigned to each production by the following set of rules:

   (a) For each production $A_b \rightarrow e_j A_f$ that corresponds in G with the path from e_i to e_j through the edge b and from e_j to e_p through f, the fuzzy relation T is looked up and the membership degree $\mu_T(e_i e_j, e_p)$ of the event sequence $e_i e_j e_p$ is assigned to the production.

   (b) For each production $A_f \rightarrow e_p$ where f are the states in G' with transitions to F and f is the edge in G from e_j to e_p, the fuzzy relation R is looked up and the membership degree $\mu_R(e_i, e_j)$ of the event sequence $e_j e_p$ is assigned to the production.

   (c) If f is a state in G' consistent with all the ends of the event sequence of S, the rules $A_b \rightarrow e_j A_f$ and $A_b \rightarrow e_j$ are in G''. Then the fuzzy relation T is looked up and the membership degree $\mu_T(e_i e_j, e_p)$ of the event sequence $e_i e_j e_p$ is assigned to the rule $A_b \rightarrow e_j$.

(d) For each production $A \rightarrow e_i A_b$ where b is the unique edge in G from e_i to e_j, the fuzzy relation R is looked up and the membership degree $\mu_R(e_i, e_j)$ of the event sequence $e_j e_p$ is assigned to the production.

(e) For each production $A \rightarrow \varepsilon A_u$ the membership degree is 1.

(f) In the special case that there is exactly one production from A, the membership degree assigned to this rule is 1.

In this way, a regular grammar with membership degrees attached to the rules is obtained.

3.2 Complexity Analysis

Let $|\Sigma| = k$ be the number of symbols in the alphabet. Let $|S| = n$ be the number of examples in the sample and $t + 1$ be the length of the largest example. In these conditions for the total examples there are $t \times n$ event sequences of length 2 and $(t - 1) \times n$ event sequences of length 3 (different or not). The first matrix corresponding to the fuzzy relation R has k^2 elements which need $t \times n$ computations (if the event sequence $e_i e_j$ appears then the membership degree of the component (i, j) of the table is calculated). The second matrix corresponding to the fuzzy relation T has k^3 elements that need $(t - 1) \times n$ computations (analogous to the previous case).
Considering the process directly, to construct the graph G the first table is converted into a matrix M that represents the graph G with k nodes. To remove overconnectivity, a new matrix M' is built with at most $3k^3$ nodes and $9k^6$ edges. The construction of the dual G' is a quadratic process in the number of the nodes of G.
Finally, transformation into a regular grammar and assignation of membership degrees are both of order k^6. So, the algorithm is polynomial with respect to the number of symbols in the alphabet. As mentioned, the algorithm controls length-3 overconnectivity. We can continue the process for lengths 4, 5, etc., with a multiplicative increment in the time, by a factor equal to k.
The level of robustness can be easily controlled by means of the α-cut. This parameter α can also be used to control the complexity of the model for large amounts of data, pruning the execution of the algorithm. If it is known that data are exactly correct (i.e., contain no noise), then every nonzero transition should be taken into account.

4 Results for Mondrian-Style Paintings

In Holland, around 1917, a group of artists came together who were deeply interested in abstract art. They founded the paper *"De Stijl" ("Style")* in which they expressed their ideas and created a new art movement known as *"neoplasticism"*. The most important of its members was Piet Mondrian (1872-1944). He conceived of his art as an instrument for contributing to true knowledge and happiness throughout the world [9]. Cubism strongly influenced his paintings between the years 1912 to 1914 in Paris. During this period trees were his principal subject. Around 1912 he began abstracting this and other subjects into a geometric version, in which only rectangles between black lines appear in the paintings. In 1917 Mondrian eliminated the relation of depth, working with continuous flat forms, considering harmony to be dependent on orthogonal lines only. By 1920 he had a pure language of rectangles between vertical and horizontal black lines. The colors are always flat and applied to rectangular areas. Mondrian thought that he should only use the primary colors (red, blue and yellow). In the beginning, he had used gray and black in the rectangles but afterwards they were eliminated. In this study we only consider the paintings of this stage. We have collected paintings with primary colors and without gray and black rectangles.In this way, he sought a non-sentimental, non-subjective picture. The paintings were meant to be independent of any historical, cultural or geographical event. Mondrian influenced the French vanguard although his major influence was on the English world. He returned to Paris in 1919 and from 1931 collaborated in the creation of the Abstraction-Création group. He moved from London to New York in 1940. We could say that Mondrian was the most radical artist who aspired to a universally valid art. The paintings with only primary colors of this latter stage (1927-1944) are the ones we have studied with the aim of generating Mondrian-style paintings [4, 5, 11, 12, 14, 15, 21, 22, 23, 24, 29]. The application of our inference algorithm to this case is justified because painter style is a fuzzy concept. This paintings can be clasified in two types. The first paintings of this stage has only a few rectangles and colors are located in big rectangles. The last paintings of this stage has many black lines and therefore more rectangles. Only one or two small rectangles are colored. These paintings can also be viewed on the Internet [29].

4.1 Shapes and Transformation in Mondrian

It has been necessary to develop an algorithm to encode Mondrian's paintings into strings and an algorithm to decode strings into structures of paintings. In a last phase convert these structures into Mondrian-style paintings. We will consider a reference system based on cardinal points. The cardinal points used for the encoding (see figure 1) are: N, NE, NW, NNE, NNW, S, SE, SW, SSE, SSW, E, ENE, ESE, W, WNW, WSW. An additional point rec is used to signal the end of the string. The terminal alphabet has a symbol for every pair $(point, color)$; since there are 17 points and 4 colors (white is considered like color too), there should be 68 terminal symbols. However, since only 39 combinations $(point, color)$ appear in the paintings by Mondrian considered in this study, the set of terminal symbols will be precisely the latter.

To transform the strings into paintings, we start with a white rectangle where all cardinal points are available. The reference system is located at the rectangle centre. When we traverse the string, rectangles with their respective colors are inserted in the cardinal point indicated by the string. Before doing so, the number of available cardinal points decreases; used cardinal points will not be available again until a new white rectangle is created in the painting. Then the reference

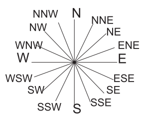

Figure 1: Cardinal Points

system is moved to this new white rectangle and all cardinals points are available again. Note that many strings can represent the same painting.

The process described above does not encode the whole structure of a Mondrian painting; the total dimensions and exact arrangement of vertical and horizontal lines must be tailored. These steps are carried out by an additional algorithm based on the number of rectangles in the generated picture. In fact, as we have mentioned above, two stages can be detected in the development of Mondrian's style during the period 1927-1944 in the paintings considered here. In the second half of this period, paintings are more densely divided and there are more rectangles, so the number of rectangles in the generated painting must be taken into account. Once a painting has been classified according to the number of rectangles, external dimensions are chosen from a table compiled from actual Mondrian paintings in each stage. Once the external dimensions have been fixed, it is time to plot horizontal and vertical lines. If the painting has less than 10 rectangles it belongs to the first stage. In this case an original painting is selected that minimizes the difference between rectangles and orientations in both paintings (original and generated). Then the proportions of rectangles in the generated painting structure are modified in order to conform to proportions in the original one. The treatment for paintings with 10 or more rectangles (second stage) is more straightforward. The number of vertical lines is higher than the number of horizontal lines, horizontal lines cut just a few vertical lines, colors are placed near to the edges of the paintings. Three types of arrangement have been selected and the one chosen corresponds to the original painting which is more similar to the generated one. This similarity is based upon the same criteria previously described.

To transform paintings into strings, the top left corner of each rectangle in the painting is taken into account to fix its orientation. A vector of 5 components in the form $(co-ordinatex, co-ordinatey, width, height, color)$ is associated with each rectangle of the painting. When every rectangle in the painting has this form, it can be transformed into a string. Iteratively, we consider a vector in the list and check whether the premises are correct for substituting the current rectangle for the present co-ordinate $(cardinalpoint)$. If the premises are not correct we check the following co-ordinate of the vector list. This process is repeated until all the vectors of the list are ordered. This ordered vector list is the string associated with the painting. Again a painting can be transformed into many strings. We have generated all the possible strings for each painting. However, in the sample, we have only take into account one string for each painting (i.e. 19 strings). Let us show the encoding process with an example (Composition in Red, Blue and Yellow (1928), 122x79cm, La Haya Gemeentemuseum), see figure 2 and figure 3.

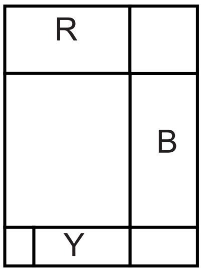

Figure 2: Composition in Red, Blue and Yellow (1928)

4.2 Obtaining Results

4.2.1 Definition of fuzzy binary relations

Now, we define two necessary membership functions in the algorithm. Let E be the set of all terminals appearing in the event sequence of S.
Specifically the elements of E are the cardinal points with their respective associated colors.
The membership function μ_R has been defined in this way: to assign membership degrees to event sequences of length 2 that appear in S the distance in degrees between cardinal points is taken into account. There are grammar rules that display symmetry with respect to the two edges. These grammar rules can be indiscriminately applied and the possible structures generated are symmetric too. Therefore it is reasonable to reduce the analysis to the first quadrant. A new cardinal point appears in each rule. We want to give a value to the " $-$ *near to* $-$ " concept. For these reasons we consider the angles between cardinal points. There are four groups of rules that are equivalent because they produce a new complete rectangle, therefore all of them have the same membership degree. On the other hand, there are three groups of rules produces the same type of structure too. For this reason the membership also is the same. To measure these angles we need a increasing function with range between zero and one. Experiments with several measures carry through the sinus function. We can think of the construction process of the structure painting like a robot moving along the rectangle [28]. In short, we consider the sinus of angles reduced to the first quadrant.
Through an interpolation and rounding up process we obtain the values: 1, 0,97, 0,95, 0,92, 0,70. Event sequences of length 2 that do not appear in S have an associated membership degree equal to 0. Also, we must to calculate membership function μ_T to take into account event sequences of length 3. Its membership degree is calculated from a t-norm and the corresponding co-norm. If they do not exist its membership degree is 0. If they exist in S its membership degree is calculated from Dombi's t-norm [20]. A fuzzy binary relation is considered to compute intersection of degrees through Dombi's t-norm (2). The number of occurrences of an event sequence of length 3 is accumulated through corresponding Dombi's t-conorm (1).

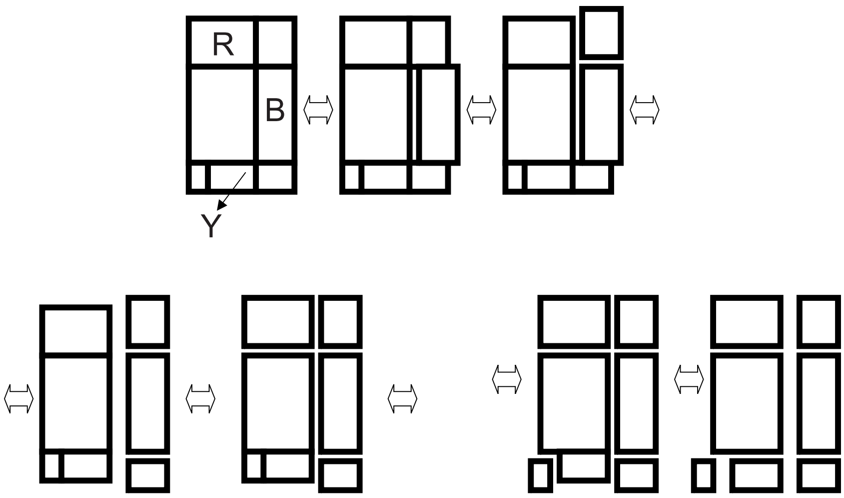

Figure 3: (EB)(NNE)(ESE)(NR)(SW)(SSEY)(rec)

$$\frac{1}{1+\left[\left(\frac{1}{a}-1\right)^{-\lambda}+\left(\frac{1}{b}-1\right)^{-\lambda}\right]^{-\frac{1}{\lambda}}} \qquad \lambda \in (0,\infty) \qquad (1)$$

$$\frac{1}{1+\left[\left(\frac{1}{a}-1\right)^{\lambda}+\left(\frac{1}{b}-1\right)^{\lambda}\right]^{\frac{1}{\lambda}}} \qquad \lambda \in (0,\infty) \qquad (2)$$

4.2.2 Computing the fuzzy regular grammar

From the sample of 19 Mondrian paintings and by applying the described inference algorithm we have obtained a fuzzy regular grammar with 159 rules and 93 non-terminals. (See [30]).

To generate a new painting, we apply the rules consecutively. In this way we obtain a string with a membership degree η (calculated in the usual way from Dombi's t-norm). Then the string is transformed into a painting through the conversion algorithms. Membership degree η is the level of credibility that the painting can be considered within the style of Mondrian. Some new Mondrian-style paintings are showed, see figures 4, 5, 6 and 7.

Considering the paintings in artistic terms the expert in art Professor Rosario Camacho judges that these paintings can be considered within of the style of Mondrian. The features of Mondrian's own style are present in the generated paintings Examples of use of the inference algorithm can be done in [30].

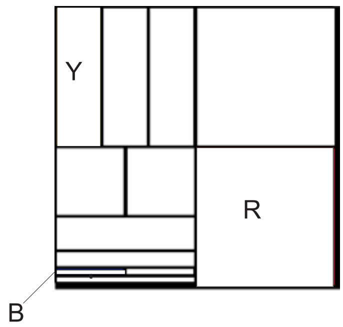

Figure 4: $\eta = 0.3090$

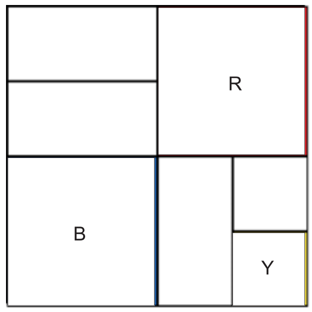

Figure 5: $\eta = 0.4535$

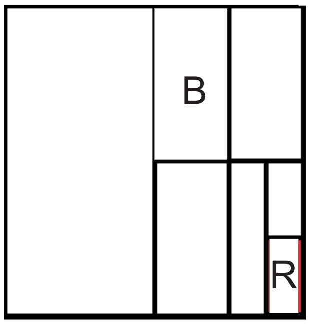

Figure 6: $\eta = 0.3265$

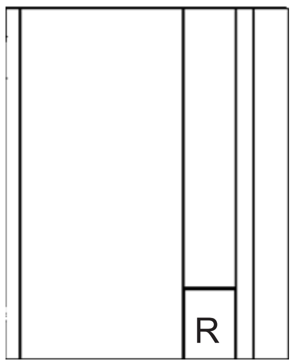

Figure 7: $\eta = 0.9417$

5 Conclusions

An inference algorithm has been developed that can infer a grammar from a set of words belonging to a regular language. The algorithm infers a fuzzy regular grammar whose productions have attached membership degrees that measure the level of credibility of each production. In this way the algorithm infers a fuzzy

language.
The algorithm has been tested and applied to infer Mondrian's style and generate new paintings. In fact, the concept of "being in Mondrian's style" is, in our opinion, an essentially fuzzy one, so fuzzy grammars are the correct tool for the problem of describing it: the algorithm generates new words (paintings) in a regular language and assigns to each word (painting) the degree of closeness to Mondrian's style. In addition, the algorithm can be used to verify known paintings.
In the present version of the algorithm, for each original painting all possible strings that code the painting have been generated; but, in order to infer the grammar, a particular string has been chosen. It would be interesting to study the influence of this choice on the membership degrees of inferred productions.

Acknowledgments

We are grateful to Professor Rosario Camacho Martínez for the time she spent in the careful evaluation of machine-generated paintings. The authors would lite to thank the anonymous referees whose comments greatly improved the readability of the paper.

References

[1] ANGLUIN D. Inference of reversible languages, *Journal of the ACM*, Vol. 29(3), pp. 741 – 765. 1982.

[2] ANGLUIN D.: Learning regular sets from queries and counterexamples, *Information and Computation*, Vol. 75, pp. 87 – 106. 1987.

[3] ANGLUIN D.: Queries and concept learning, *Machine Learning. 2*, pp. 319 – 342. 1988.

[4] ARGAN G.C.*El arte moderno 1770-1970.* Tomo II. Valencia: Doménech-Burjasot. pp 492 – 497, 1977.

[5] ARNALDO J.*Historia del Arte: Las vanguardias históricas* (1).Vol. 45. Madrid:Historia 16. , pp 112 – 118, 157 – 159.1991.

[6] BALCÁZAR J.L., DÍAZ J., GAVALDÀ R., WATANABE O.: Algorithms for Learning Finite Automata from Queries : A Unified View. *Advances in Algorithms, Languages and Complexity. D. -Z. and K. -I. Ko (Eds.)*, Kluwer Academic Publishers pp. 73 – 91. 1997.

[7] BIERMANN A.W., FELDMAN J.A.: On the synthesis of finite state machines from samples of their behaviour. *IEEE Transactions on Computers*, Vol. 21:, pp. 592 – 597, 1972.

[8] CARRASCO R. C., ONCINA J.: *Grammatical Inference and Applications. Lecture Notes in Artificial Intelligence (subseries of LNCS)* New York: Springer-Verlag, Vol.862,pp. 139 – 152. 1994.

[9] Castellanos A. Galiano I., Vidal E.: *Grammatical Inference and Applications. Lecture Notes in Artificial Intelligence (subseries of LNCS)* New York: Springer-Verlag, Vol.862,pp. 93 – 105. 1994.

[10] Cook J. E. Wolf A. L.:Discovering models of Software Processes from Event-Based Data. Technical Report CU-CS-820 – 96, Department of Computer Science, University of Colorado, Boulder, CO 803009 USA. 1996.

[11] Dee Reynolds*Symbolist aesthetics and early abstract art. Sites of imaginary space.* Cambridge: Cambridge University Press. 1995.

[12] Deicher S.*Piet Mondrian 1872-1944. Composición sobre el vacío.* Köln-Germany: Benedikt Taschen. 1995.

[13] Dubois D., Prade H. *Fuzzy sets and systems: theory and applications* vol. 144 of Mathematics in Science and Engineering, Academic Press 1980.

[14] Elgar F. Muller J-É.*La peinture moderne.* París: Fernand Hazan éditeur. 1979 Paintings 173 – 176.

[15] Heard Hamilton G.*Pintura y escultura en Europa* 1880 – 1940. Madrid: Manuales arte cátedra. 1980.pp 332 – 345.

[16] Fortes I., Morales Bueno R., Pérez de la Cruz J.L., Triguero F., Comino M.A.: Inferencia de Gramáticas Regulares Difusas a partir de Ejemplos, *VIII Congreso Español sobre Tecnologías y Lógica Fuzzy,*Pamplona (España), pp. 91 – 96. 1998.

[17] Honda N. et al.F-Recognition of Fuzzy Languages in Gupta M.M. Saridis G. Gaines B.R (eds): *Fuzzy Automata and Decision Process,*. North-Holland, 1977. Cap. 9

[18] Hopcroft J.E., Ullman J.D.: *Introduction to Automata Theory Languages and Computation.* Reading, Massachusetts: Addison Wesley 1993.

[19] Kearns M.J., Vazirani U.V.: *An Introduction to Computational Learning Theory.* Cambridge: MIT Press, 1994.

[20] Klir G.J. Folger. T.A.:*Fuzzy Sets, Uncertainty and Information.* Prentice Hall. 1988.

[21] Lambert J-C.*Historia general de la pintura: Pintura abstracta.* Vol. 23. Verona: Aguilar. 1969. pp 11 – 17

[22] Maillard R.*Dictionnaire Universel de la Peinture.* Vol. 5. París: S.N.L.-Dictionnaires Robert.1975.

[23] Marchan Fiz S.*Las vanguardias históricas y sus sombras* 1917 – 1930 in *Enciclopedia Summa Artis.* Vol. 39. Madrid: Espasa-Calpe. 1995. p 227.

[24] RAMIREZ J.A. *Biblioteca básica de arte: El arte de las vanguardias.* Madrid: Anaya. 1991. pp 48 – 55.

[25] SANTOS E.S. Regular Fuzzy Expression in GUPTA M.M. SARIDIS G. GAINES B.R (EDS) *Fuzzy Automata and Decision Process*, Cap. 10. North-Holland, 1977. pp. 169 – 175.

[26] THOMASON M. MARINOS P. Deterministic Acceptors of Regular Fuzzy Languages, *IEEE Transactions on Systems, Man and Cybernetics n^o3* pp. 228 – 230, 1974.

[27] VALIANT L.G A Theory of the Learnable. *Communications of the ACM.* 27(11), 1984, pp. 1134 – 1142.

[28] WAN FOO KIM, JOONG HYUPKO, MYUNG JIN CHUNGUncertain robot enviroment modelling using fuzzy numbers. *Fuzzy Sets and Systems* Vol. 61, n1,Jan. 10, 1994.

[29] http://www.aukword.demon.co.uk/piran/mondrian/resources.html

[30] http://www.lcc.uma.es/personal/morales/mondrian/mondrian.html.