

# Data Mining: Heuristics in Fuzzy Pattern Matching

Jose M. Cadenas      María C. Garrido      Juan J. Hernández\*

Dpto. Ingeniería de la Información y las Comunicaciones

Facultad de Informática. Universidad de Murcia.

30100-Espinardo. Murcia. Spain

Fax: +34 968 364151

email: {jcadenas, mgarrido}@dif.um.es      \* email: juanjoseha@terra.es

**ABSTRACT:** Fuzzy pattern matching technique represents a group of fuzzy methods for supervised fuzzy pattern recognition. It has a number of advantages over other pattern recognition methods, including simpler methods of attribute selection or ability to learn in real time environments. These methods build a prototype for each attribute and combine the partial estimations of each prototype by a fusion operator. One of the major problems of this technique is that it is not able to model the dependencies between attributes since fuzzy pattern matching assumes non interactivity between them, and nowadays there is no heuristic in the literature that solves this problem. In this paper we propose a solution to this problem. In order to keep the good properties of fuzzy pattern matching, this heuristic will have the objective of minimizing the dependencies between attributes modeled. To show the accuracy of the proposed solution, we have tested the method on several data sets.

**KEYWORDS:** Data Mining, Heuristics, Soft Computing, Fuzzy Systems, Fuzzy Integral, Fuzzy Pattern Matching, Pattern Recognition, Learning Systems, Dependences between attributes, Optimization.

## INTRODUCTION

The data mining is not more than an essential step of a process whose objective is the knowledge discovery in data sets. This process consists of the iterative sequence of 5 phases:

1. Phase of integration and compilation of information, in where a data warehouse is constructed.
2. Phase of selection, removing and transformation. The quality of the discovered knowledge not only depends on the used method, but also on the quality of the mined data.
3. Phase of Data Mining. This phase produces new knowledge that can use the user. This is made constructing a model based on the data. In this phase it is decided which is the task to make, the type of model and the method.
4. Phase of evaluation and interpretation in which the models obtained by the data mining methods are evaluated.
5. Phase of diffusion in which we use the new knowledge.

In this paper we are centred on the phase of data mining. We apply the phase of data mining to the supervised task of classification and the utilized method is Fuzzy Pattern Matching (FPM).

Fuzzy pattern recognition presents one of the largest application areas of fuzzy set theory. FPM technique represents a group of fuzzy methods for supervised pattern recognition. The most general framework was introduced in [8] as a pattern recognition method based on the fuzzy integral. One of the principal differences between this technique and the others is that FPM, as Naive Bayesian classifiers, works on marginal distribution instead of joint ones, where partial matching values with respect to a given feature are combined together [6]. This way of learning has a number of advantages [2] that should be researched, but one of its main problems is that FPM is not able to model the dependencies between features. However, in the field of Naive Bayesian classifiers, we may find in the literature [4] a lot of people who show that these methods get results competitive with state-of-the-art classifiers such as C4.5. This fact raises the question of whether it is necessary to model the dependencies between features to get better results. In [11], Kononenko said: "It seems that in the data used by human experts there are no strong dependences between attributes because attributes are properly defined". Anyway Naïve Bayesian Classifiers perform poorly on some data sets and these authors [7],[12],[11] have proposed different heuristics to model the dependencies between attributes.

In the field of FPM, we may find in the literature, [3], a heuristic different to the previous ones, which performs firstly a clustering over the data and assumes that there is no dependence in the examples belonging to the same cluster. In this paper we present some heuristics to extend FPM to model the dependencies between features in a similar way to the

Naive Bayesian approaches. This heuristic will take into account an important objective: minimizing the dependence between attributes modelled by the classifier. This objective will allow keeping the good properties of FPM.

The paper is organized as follows: initially, we will introduce Fuzzy pattern matching technique and the fuzzy integral as matching operator. Next, we will present an extension of the Fuzzy pattern matching technique and the dependence between attributes problem. We will give the proposed solution to solve this problem and we will explain a method based on our solution and parzen window. Finally, before the conclusions, some results of tests will be presented.

## FUZZY PATTERN MATCHING

Fuzzy pattern matching methods, [9], combine partial matching values with respect to a given attribute into a single one. Here we build fuzzy prototypes of classes under the form of fuzzy sets. The classification of an unknown sample is done by matching the sample with all the prototypes, and then choosing the class with the highest matching degree. Specifically, let us denote by  $P_1, \dots, P_c$  the prototypes of the classes, and let us suppose that the classes are described by  $n$  features. Each prototype  $P_j$  is a collection of  $n$  fuzzy sets  $P_{j1}, \dots, P_{jn}$  expressing the set of typical values of each feature for class  $C_j$ .

When an unknown sample  $Z = (z_1, \dots, z_n)$  is presented, the matching process is done in two steps:

- Matching with respect to an attribute  $i$ . We compute by some means the matching degree,  $\phi_{ij}$ , between the value  $z_i$  and the fuzzy set of typical values  $P_{ji}$ ,  $\forall i, j$ .
- Global matching: all the degrees of matching concerning  $C_j$  are merged into a single one by a fusion operator:

$$\Phi_j = H(\phi_{1j}, \dots, \phi_{nj}) \quad (1)$$

The result of this fusion represents the matching degree between the new sample and the prototype  $P_j$ .

Any classifier based on FPM contains two distinct parts:

- The prototype builder: we need to build the prototypes of the classes from the training data. In this part, any method producing fuzzy sets, possibility or probability distributions can be used here, as fuzzy c-means, Parzen or possibilistic histograms, [5].
- The aggregation part, which use a matching operator to aggregate partial matching degrees. The matching operator can be a multiplication, a minimum, an average or a fuzzy integral, [8].

In this paper, we will use the fuzzy integral as matching operator. We introduce the fuzzy integral briefly:

## FUZZY INTEGRAL

Let's suppose that we have  $l$  data in the following way,  $(\bar{x}_1, y_1), (\bar{x}_2, y_2), \dots, (\bar{x}_l, y_l)$ , where  $\bar{x}_k$  is a  $n$ -dimensional vector  $\bar{x}_k = (x_{k1}, x_{k2}, \dots, x_{kn})$  representing the results obtained by the  $n$  sources (that we must merge) in the  $k$ -th example, and  $y_k$ , the supervised output. The objective of fuzzy integral is to merge the different information sources to approximate as well as possible the supervised output. For this it will be necessary to minimize the quadratic error:

$$E^2(\mu) = \sum_{k=1}^l (y_k - F_\mu(\bar{x}_k))^2, \text{ where } F_\mu \text{ is a particular fuzzy integral with respect to a fuzzy measure } \mu.$$

As the variable here is the fuzzy measure, let us express it in  $(2^n - 2)$ -dimensional vector form, including all the coefficients of the fuzzy measure, except  $\mu_0$  and  $\mu_X$ , since their values are, respectively,  $0$  and  $1$ .

$$\mu = [\mu_1, \mu_2, \dots, \mu_n, \mu_{1,2}, \mu_{1,3}, \dots, \mu_{1n}, \mu_{1,2,3}, \dots, \mu_{2,3, \dots, n}]$$

$F_\mu(\bar{x}_k)$  is calculated in the following way:  $F_\mu(\bar{x}_k) = \sum_{i=1}^n (x_{kper(i)} - x_{kper(i-1)}) \mu_{per(i), \dots, per(n)}$ , where the elements of

$\bar{x}_k$  have been permuted in order to have  $x_{kper(1)} \leq x_{kper(2)} \leq \dots \leq x_{kper(n)}$ , and where  $x_{kper(0)} = 0$ . The coefficients must satisfy the following monotonicity relation  $A \subseteq B \Rightarrow \mu(A) \leq \mu(B) \quad \forall A, B \subseteq X$

In order to minimize  $E^2(\mu)$  in line with the monotony constraints, we will use the Choquet integral as fuzzy integral, which will lead to a quadratic programming problem, [10].

## EXTENDING FUZZY PATTERN MATCHING

If we look at formula (ref{formul1}), this is very similar to the Bayesian classifier with independent features, whose discriminant function is:

$$\Phi(C_j / Z) = \prod_j p(z_i / C_j) P(C_j) \quad (2)$$

where  $p(z_i / C_j)$  is the marginal conditional density of attribute  $i$ , given the class  $j$ , and  $P(C_j)$  is the a priori probability of class  $C_j$ . As all methods working on marginal distribution instead of joint ones, we are not able by FPM to model the dependencies between the features. For example, FPM is not able to solve the xor problem. These remarks narrow the applicability of fuzzy pattern matching methods in a real situation, and suggest, as it is said in [9], to pre-process the data before using FPM.

The question is: What kind of pre-processing we should perform on the data? We can't learn prototypes of classes based only on one attribute, since in this way we are not able to model the dependencies between them. We could learn prototypes of classes from all the attributes (that is what most learning methods do) but due to the high dependence between attributes these methods model, we lose the advantages of fuzzy pattern matching technique (such as simpler attribute selection methods). Hence, in order to learn the prototypes of classes, we have to consider another objective: minimizing the dependencies between attributes. We have to extend the prototype builder method to use not only the information of one attributes but also, as less as possible, the information of the other attributes. We have to use the information of the other attributes when the accuracy of the prototype built over one attribute is not acceptable. Figure 1 illustrates this idea.

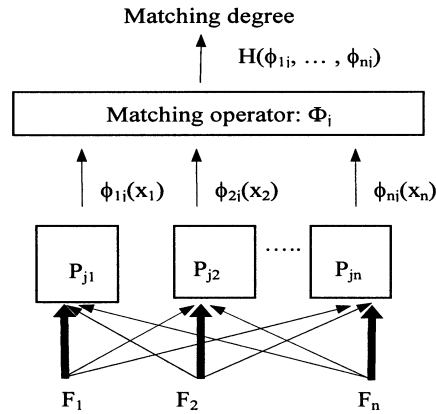


Figure 1: Proposed solution: classification procedure for the class  $C_j$ .

We represent the important feature in the prototype with the thick line, and the auxiliary features with the fine line. It must be noticed that the prototype builder attempts to optimize two objectives: minimizing the dependencies between attributes and maximizing the accuracy of the classifier. We have to choose the importance we give to each objective due to in many real situations we can't reach both.

## A HEURISTIC TO MODEL DEPENDENCES

In this section we present a simple approach using Parzen. This heuristic consists on using one auxiliary feature (the best auxiliary feature) when the accuracy of Parzen Window method over the main feature is not satisfactory.

Therefore Parzen will be the learning method used to build the prototype of classes. This prototype, in the case of FPM approach, uses just the main attribute. In our method, we will decide for each value of the main attribute in the learning examples if we need to use more attributes in the Parzen method in order to improve the accuracy of the prototype. If

we have to do it, this approach addresses the problem of finding just one auxiliary attribute, the best of all. Further research should address the problem of using more than one auxiliary attribute when it is needed to improve accuracy.

We don't say anything about the matching operator because in our method we use it in the same way as FPM. In the experiment we will use the fuzzy integral ([1],[10]) as the matching operator, but any matching operator might be used in this case. Before going into detail of the approach, we are going to explain some things about the parzen window method used. This method classifies an example looking for all the examples of a class that are inside the window defined by one centre (the example) and a distance (this value will depend on each data set). The degree of belonging will be calculated as the number of the examples of the class (inside the window) divided by the total number of the examples (inside the window too).

## EXTENDED FPM WITH PARZEN

Let  $X = [X^1, \dots, X^m]$  be the training set. Let  $X_i = [x_i^1, \dots, x_i^m]$  be the  $i$ -th attribute of each example. Let  $V_i = [v_i^1, \dots, v_i^m]$  be the set of different values we find in  $X_i$ .

We will decide for each value  $v_i^k$  if it needs an auxiliary attribute. If the answer is "yes" then we have to associate to this value the auxiliary attribute that give us the best result.

Let  $X(v_i^k) = X_i^k$  be the subset of examples of  $X_i$  that are inside the window defined by  $v_i^k$ . (let us denote by  $p_w$  the size of the window).

$$X_i^k = \{x_i^h\}, \quad x_i^h \in X_i \text{ and } |x_i^h - v_i^k| \leq p_w \quad (3)$$

Let  $\phi_{ij}(v_i^k)$  be the degree of belonging (calculated through Parzen Window) of  $v_i^k$  to the class  $j$ .

If  $\max_{j=1, \dots, c} \phi_{ij}(v_i^k) \geq p_{\max}$ , where  $p_{\max}$  is a probability close to 1, then  $v_i^k$  doesn't need an auxiliary attribute (in this case there is a prototype of a class that classifies the value  $v_i^k$  with a high probability). Otherwise we have to find the best auxiliary attribute.

$$\phi_{ij}(v_i^k) = \frac{n_j}{|X_i^k|} \quad (4)$$

where  $n_j$  is the number of examples in  $X_i^k$  that belongs to the class  $j$ .

Let us denote by  $E_{i'}(v_i^k)$  the mean error that each example of  $X_i^k$  has in using the auxiliary attribute  $i'$ . This is the value that we will use to decide which is the best auxiliary attribute.

$$E_{i'}(v_i^k) = \sum_{x_i^h \in X_i^k} E_{i'}^2(x_i^h) / \|X_i^k\| \quad (5)$$

$E_{i'}(v_i^h)$  is the error that the example  $x_i^h$  has in using the auxiliary feature  $i'$ .

$$E_{i'}(x_i^h) = \begin{cases} 1 - (n_{i'}^j / n_{i'}) & \text{if } n_{i'} \geq n_{\min} \\ 1/c & \text{otherwise} \end{cases} \quad (6)$$

where  $n_{i'}$  is the number of examples that are inside the window defined by  $x_i^h$  ( $x_i^h$  is the value of the feature  $i'$  in the example where  $x_i^h$  belongs to) and  $p_w$  and  $n_{i'}^j$  are the number of these examples that belong to class  $j$ , with  $j = \text{class}(x_i^h)$ .

$$n_{i'} = \sum_{x_i^h \in X_i^k} H_{i'}(x_i^h, x_i^{h'}) \quad \text{with } H_{i'}(x_i^h, x_i^{h'}) = \begin{cases} 1 & \text{if } |x_i^{h'} - x_i^h| \leq p_w \\ 0 & \text{otherwise} \end{cases}$$

and

$$n_{i'}^j = \sum_{x_i^h \in X_i^k} H_{i'}^j(x_i^h, x_i^{h'}) \quad \text{with } H_{i'}^j(x_i^h, x_i^{h'}) = \begin{cases} 1 & \text{if } |x_i^{h'} - x_i^h| \leq p_w \text{ and } \text{class}(x_i^{h'}) = j \\ 0 & \text{otherwise} \end{cases}$$

In order to improve reliability, if  $n_i$  is small (we use the parameter  $n_{\min}$  to check it) then we will not take into account the relative frequencies of each class in the window and we will set the same error for each class. The best auxiliary attribute will be the one that:

$$E_{ibest}(v_i^k) = \min_{\substack{i=1,\dots,n \\ i \neq i}} \{E_{i'}(v_i^k)\} \quad (7)$$

This value, in order to minimize the number of auxiliary attributes used, will be compared with the probability of the best class without auxiliary attributes ( $\max_{j=1,\dots,c} \phi_{ij}(v_i^k)$ ). If this probability is bigger than  $(1 - E_{ibest}(v_i^k))$ , then the auxiliary attribute does not improve the results obtained by the main attribute by itself, and we will not associate it to the value  $v_i^k$ .

To classify an unknown example, the following algorithm is applied:

#### Algorithm of classification

1. Unknown example  $Z = (z_1, \dots, z_n)$
2. For each attribute  $i$  do the partial matching:
  - a. To decide if we need an auxiliary attribute
    - i. To search the value  $v_i^k$  closest to  $z_i$
  - b. If this value has not associated any auxiliary attribute then we apply Parzen method just over the attribute  $i$
  - c. If  $v_i^k$  has associated an auxiliary attribute  $i'$  then we apply Parzen over the attributes  $i$  and  $i'$
3. To do the global matching applying the matching operator over the partial matching values

## EXPERIMENTS

We have tested the method on several data sets. Here we present the following results: The simulated data set is an extended version of xor problem using two more attributes that are completely irrelevant. In this little example we can see how our method can solve problems that FPM methods cannot solve. Any method using FPM cannot reach a good solution from this data set because each attribute by its self has not any discrimination power. We need to model the dependencies between the attributes 1 and 2 and that is precisely what our method does. The prototype built over the attribute 1 uses as the best auxiliary attribute the attributes 2, and the prototype of attribute 2 uses the attribute 1. The prototypes built over attributes 3 and 4 are not relevant because they can't separate the classes even with the help of one auxiliary attribute.

Our real data set is the IRIS data set. This data set has 150 examples, each of one is described by 5 attributes, of which 4 are continuous (sepal length, sepal width, petal length and petal width) and one, which is the class, is discrete. This set of data contains 3 classes with 50 examples in each (Setosa, Versicolor and Virginica). Class refers to the type of iris plant. One class (Setosa) is linearly separable from the other two, but these are not linearly separable between themselves. In this experiment, we will utilize the two classes versicolor and virginica. Therefore, we will have 100 examples. As the prototype builder method we have used Parzen Window with a size of 0.1. We have also set  $p_{\max} = 0,95$  and  $n_{\max} = 6$ . In Table I we show the results obtained with cross-validation (% of accuracy) in FPM method and our method. As FPM method, we have used the same idea as in our method (Parzen window as the prototype builder method and fuzzy integral as the matching operator). The difference is (as we explained before) FPM doesn't use auxiliary attributes. In table I we also present the results obtained with Naive Bayes method. We can see that this method doesn't improve the results obtained by our method (EFMP). We also show the results obtained by other learning methods. The column "Reference" show the kind of cross validation test (5 fold cross validation).

Method	Accuracy %	Reference
FPM	92 ± 4,12	Cadenas (5CV)
Our method (EFPM)	94 ± 2	Cadenas (5CV)
Naive Bayes	93 ± 2,45	Cadenas (5CV) (with WEKA)
C4.5	91 ± 4,9	Cadenas (5CV) (with WEKA)

Table I.: Accuracy in IRIS

## CONCLUSION

In this paper we propose a heuristic to extend fuzzy pattern matching to model the dependencies between attributes. This heuristic takes into account a new objective: minimizing the dependencies between attributes. This objective allows us developing methods that solve problems better than FPM can do but at the same time keeping the good properties of this technique.

The experiments show that, using a simple approach based on Parzen method, we have built a model with a high accuracy, improving the results obtained by FPM method. Further research should develop methods based on other learning techniques (such as decision trees or possibilistic histograms), building a model for each attribute and attempting to minimize the dependencies between them.

## ACKNOWLEDGMENTS

Work supported by project TIC2002-04021-C02-01.

## REFERENCES

- [1] Cadenas, J.M; Garrido, M.C.; Hernandez, J.J., 2003, "Fuzzy integral in systems modeling", IEEE International Conference on Systems, Man & Cybernetics, pp. 3182 - 3187.
- [2] Cadenas, J.M; Garrido, M.C.; Hernandez, J.J., 2004, "Improving fuzzy pattern matching technique to deal with non discrimination ability features", IEEE International Conference on Systems, Man & Cybernetics, pp. 5708-5713.
- [3] Devillez, A., 2004, "Four fuzzy supervised classification methods for discriminating classes of non-convex shape", Fuzzy Sets and Systems, Vol. 141, pp. 219 – 240.
- [4] Domingos, P.; Pazzani, M., 1997, "On the optimality of the simple Bayesian classifier under zero-one loss", Machine Learning, Vol. 29, pp. 103-130.
- [5] Dubois, D.; Prade, H. 1983, "Unfair coins and necessity measures: towards a possibilistic interpretation of histograms", Fuzzy Sets and Systems, Vol. 10, No.1, pp. 15 - 20.
- [6] Dubois, D.; Prade, H.; Testemale, C., 1988, "Weighted fuzzy pattern matching", Fuzzy Sets and Systems, Vol 28, No.3, pp. 313 - 331.
- [7] Friedman, N.; Geiger, D.; Goldszmidt, M., 1997, "Bayesian Network Classifiers. Machine Learning", Vol. 29, pp. 131-163.
- [8] Grabish, M; Sugeno, M, 1992, "Multi-attribute classification using fuzzy integral", Proc. of Fuzzy IEEE, pp. 47-54.
- [9] Grabisch, M.; Nicolas, J.M., 1994, "Classification by fuzzy integral: Performance and tests", Fuzzy Sets and Systems, Vol. 65, No.2-3, pp. 255 - 271.
- [10] Grabisch, M.; Nguyen, H.T.; Walker, E.A., 1995, "Fundamentals of uncertainty calculi with applications to fuzzy inference", Kluwer Academic Publishers.
- [11] Kononenko, I., 1993, "Inductive and Bayesian Learning in Medical Diagnosis". Applied Artificial Intelligence, Vol. 7, pp. 317 – 337.
- [12] Pazzani, M., "Searching for dependencies in Bayesian classifiers". In D. Fisher & H.-J. Lenz (Eds.), Learning from data: Artificial intelligence and statistics V, pp. 239 - 248. New York, NY: Springer-Verlag.