

Heuristics to model the dependencies between Features in Fuzzy Pattern Matching

J.M. Cadenas

M.C. Garrido

J.J. Hernández

Departamento de Ingeniería de la Información y las Comunicaciones
 Universidad de Murcia. 30071-Campus de Espinardo, Murcia. Spain
 jcadenas@dif.um.es mgarrido@dif.um.es juanjoseha@terra.es

Abstract

Fuzzy pattern matching technique represents a group of fuzzy methods for supervised fuzzy pattern recognition. These methods build a prototype for each feature 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 features, 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 features modelled. To show the accuracy of the proposed solution, we have tested the method on several data sets. In this paper, we present the results obtained in a simulated data set and a real data set.

Keywords: Heuristic, Soft Computing, Pattern recognition, fuzzy pattern matching, fuzzy integral, fuzzy systems

1 Introduction

Fuzzy pattern recognition presents one of the largest application areas of fuzzy set theory. Fuzzy pattern matching (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 Naive Bayesian Classifiers perform poorly on some data sets and these authors [7, 12, 11] have proposed different heuristics to model the dependencies between features.

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 be-

tween feature modelled by the classifier. This objective will allow keeping the good properties of FPM.

The paper is organized as follows: in section 2 we will introduce Fuzzy pattern matching technique. In section 3 we will present the dependence between feature problem and we will give the proposed solution to solve this problem. In section 4 we will explain a method based on our solution and parzen window and, finally, before the conclusions, some results of tests will be presented.

2 Fuzzy pattern matching

Fuzzy pattern matching methods, [9], combine partial matching values with respect to a given feature 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 a feature i . We compute by some means the matching degree, ϕ_{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 . the classification procedure for the class C_j is illustrated in Figure 1.

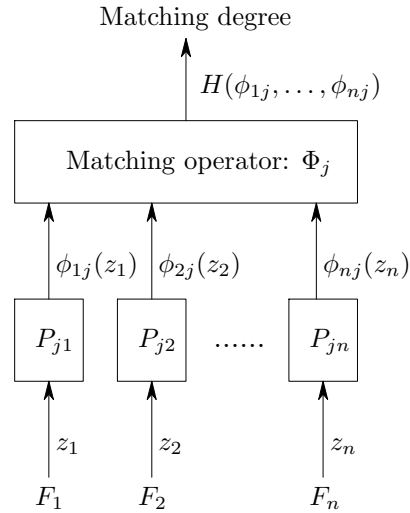


Figure 1: Classification procedure for the class C_j

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].

3 Extending Fuzzy Pattern Matching

If we look at formula (1), 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 feature 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 preprocess the data before using FPM.

The question is: What kind of preprocessing we should perform on the data? We can't learn prototypes of classes based only on one feature, since in this way we are not able to model the dependencies between them. We could learn prototypes of classes from all the features (that is what most learning methods do) but due to the high dependence between features these methods model, we lose the advantages of fuzzy pattern matching technique (such as simpler feature selection methods). Hence, in order to learn the prototypes of classes, we have to consider another objective: minimizing the dependencies between features.

Let's look at figure 1. We have to extend the prototype builder method to use not only the information of one feature but also, as less as possible, the information of the other features. We have to use the information of the other features when the accuracy of the prototype built over one feature is not acceptable. Figure 2 illustrates this idea.

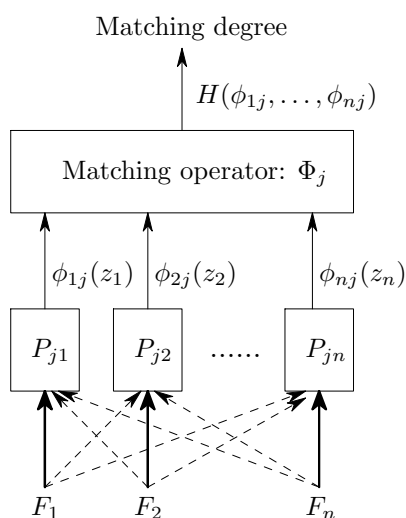


Figure 2: Proposed solution: classification procedure for the class C_j

We represent the important feature in the prototype with the solid line, and the auxiliary features with the dashed line. It must be noticed that the prototype builder attempt to optimize two objectives: minimizing the dependencies between features and maximizing the accuracy of the classi-

fier. We have to choose the importance we give to each objective due to in many real situations we can't reach both.

4 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 Window will be the learning method used to build the prototype of classes. This prototype, in the case of FPM approach, uses just the main feature. In our method, we will decide for each value of the main feature in the learning examples if we need to use more features 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 feature, the best of all. Further research should address the problem of using more than one auxiliary feature 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 ($[10, 1]$) 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).

4.1 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 feature 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 feature. If the answer is "yes" then we have to associate to this value the auxiliary feature 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).

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

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 feature (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 feature.

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

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 feature i' . This is the value that we will use to decide which is the best auxiliary feature.

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

$E_{i'}(x_i^h)$ 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 $j = class(x_i^h)$.

$$n_{i'} = \sum_{x_i^{h'} \in X_i^k} H_{i'}(x_i^h, x_i^{h'})$$

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

and

$$n_{i'}^j = \sum_{x_i^{h'} \in X_i^k} H_{i'}^j(x_i^h, x_i^{h'})$$

with

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

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 feature 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 features used, will be compared with the probability of the best class without auxiliary features ($\max_{j=1,\dots,c} \{\phi_{ij}^j(v_i^k)\}$). If this probability is bigger than $(1 - E_{ibest}(v_i^k))$, then the auxiliary feature does not improve the results obtained by the main feature 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

- Unknown example $Z = [z_1, \dots, z_n]$
- For each feature i do the partial matching:
 - To decide if we need an auxiliary feature
 - * To search the value v_i^k closest to z_i
 - If this value has not associated any auxiliary feature then we apply Parzen method just over the feature i
 - If v_i^k has associated an auxiliary feature i' then we apply Parzen over the features i and i'
- To do the global matching applying the matching operator over the partial matching values.

5 Experiments

We have tested the method on several data sets. Here we present the results obtained in a simulated data set (an extension of the xor problem) and a real data set (the Wisconsin Breast Cancer data set).

5.1 Simulated data set

The simulated data set is an extended version of xor problem using two more features that are completely irrelevant. In this little example we can see how our method can solve problems that FPM methods cannot solve.

Table 1: Simulated data set

F_1	F_2	F_3	F_4	Class
0	0	0	8	1
0	1	2	6	1
1	0	4	4	1
1	1	6	2	1
0,5	0,5	8	0	1
3	3	0	8	1
3	4	2	6	1
4	3	4	4	1
4	4	6	2	1
3,5	3,5	8	0	1
0	3	0	8	2
0	4	2	6	2
1	3	4	4	2
1	4	6	2	2
0,5	3,5	8	0	2
3	0	0	8	2
3	1	2	6	2
4	0	4	4	2
4	1	6	2	2
3,5	0,5	8	0	2

Any method using FPM cannot reach a good solution from this data set because each feature by its self has not any discrimination power. We need to model the dependencies between the features 1 and 2 and that is precisely what our method does. The prototype built over the feature 1 uses as the best auxiliary feature the feature 2, and the prototype of feature 2 uses the feature 1. The prototypes built over features 3 and 4 are not relevant because they can't separate the classes even with the help of one auxiliary feature.

5.2 Real data set

Our real data set is the Wisconsin Breast Cancer data set. This data set has 683 examples, each of one is described by 9 continuous features. The values of each feature are nominal and go from 1 to 10. As we have used the fuzzy integral as the matching operator, and due to the exponential complexity of this method, we have reduced the original set of features from 9 to 7 features, removing initially the 2nd and the 5th features. We have used the same matching operator for both class (hence we have just optimized $2^7 - 2$ coefficients due to we use the same fuzzy measure for both classes), and the quadratic criterion to optimize the coefficients. 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 2 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 features. In table 2 we also present the results obtained with Naive Bayes method and different extensions to find dependencies between features. We can see that these extensions doesn't improve the results obtained by Naive Bayes, while we improve significantly in our method the results obtained by FPM. We also show the results obtained by other learning methods. The column "Reference" show the author and the kind of cross validation test (normally 5 or 10 fold cross validation).

Table 2: Accuracy in WBC

Method	Accuracy %	Reference
FPM	96.65±2.04	Cadenas (10CV)
Our method	97.5±1.57	Cadenas (10CV)
Naive Bayes	96.4	Ster. Dobnikar (10CV)
Semi-NB	96.6	Ster. Dobnikar (10CV)
Naive Bayes	97.36±0.5	Friedman (5CV)
TAN	96.92±0.67	Friedman (5CV)
Naive Bayes	97.3	Pazzani
BSEJ	97.1	Pazzani
C4.5	94.7±2.0	Zarndt (10CV)
SVM	97.2	Bennet and Blue (5CV)

6 Conclusion

In this paper we propose a heuristic to extend fuzzy pattern matching to model the dependencies between features. This heuristic takes into account a new objective: minimizing the dependencies between features. 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 feature and attempting to minimize the dependencies between them.

Acknowledgments

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

References

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