Proc. 14th Int. Conf. on Global Research and Education, Inter-Academia 2015
JJAP Conf. Proc. **4** (2016) 011610

Fuzzy Classifier Hyper-matrices for Rapid Data Classification

Balázs Tusor[1,2,a], Annamária R. Várkonyi-Kóczy[1,3,4,b]

[1] *Integrated Intelligent Systems Japanese-Hungarian Laboratory, Óbuda University, Budapest, Hungary*
[2] *Doctoral School of Applied Informatics, Óbuda University, Budapest, Hungary*
[3] *Institute of Mechatronics and Vehicle Engineering, Óbuda University, Budapest, Hungary*
[4] *Department of Mathematics and Informatics, J. Selye University, Komarno, Slovakia*

E-mail: [a]balazs.tusor@gmail.com, [b]varkonyi-koczy@uni-obuda.hu

(Received September 28, 2015)

In this paper, a novel fuzzy classification method is presented for very fast evaluation. The idea is the usage of a multidimensional matrix for data classification purposes, in which the attribute values of the input data are used as matrix coordinates. The matrices contain the fuzzy membership function values (i.e. the degree of the matrix elements belonging to each class) mapped to integer coordinates in the whole problem domain. This method is ideal for problems where the input values are integer and the quantity of the possible values they can take is finite. A training algorithm is proposed for tuning the classifier. The performance of the system is shown through an experiment.

1. Introduction

Classification is a widely used tool to sort measured/given data into categories based on its observed characteristics. It is applied in many fields in physics and engineering, like photonics (Refs. 1 and 2), nanotechnology (Refs. 3 and 4), plasma physics (Ref. 5), manufacturing technology (Refs. 6 and 7), etc. Besides efficiency, many problems require the classifier to be fast as well. The fuzzy classifier proposed in this paper is based on a simple idea for enhancing the evaluation speed: instead of calculating the output values of the system during run-time, they can be pre-calculated in the training phrase and stored in a suitable data structure where they can be quickly accessed during the evaluation phrase using the attribute values of the input data. This idea has been successfully applied in (Ref. 8), where the authors proposed a fast fuzzy decision tree architecture for real-time color filtering. The classifier presented in this paper is an improved utilization of that idea, using multi-dimensional arrays (so-called hyper-matrices or tensors) combined with fuzzy logic in order to realize even faster and more robust classification.

The rest of the paper is as follows. Section 2 gives an overview of the classifier. Subsection 2.1 describes the general architecture of the system, while Subsect. 2.2 presents a method to determine its applicability. Subsection 2.3 proposes a training algorithm and Subsect. 2.4 illustrates the operation with a set of experiments. Finally, Sect. 3 draws the conclusions and presents future work.

2. Fuzzy Hyper-matrices

2.1 The architecture of the classifier

The proposed classifier uses hyper-matrices (HMs) in order to describe the known nature of the problem domain. The problem domain has as many dimensions as many attributes the data has, and the extent of each dimension is the number of possible (integer or categorical) values that the related attribute can take. For example, in color filtering problems where the attributes of the input data are the HSV or RGB color coordinates, the HMs of the classifier are 3D arrays with each dimension ranging from 0 to 255.

Since the HMs are addressed by the input data attribute values, they are most suitable for problems where the attributes are of positive integer type and their domain is bounded. However, the same principle is also applicable for data with continuous values, though the data (during the evaluation phase) is needed to be converted into a range of positive integers, which is a simple task with a suitable linear transformation function and bounding.

In general, (Q_c+1) HMs are required for a classifier that can distinguish Q_c number of classes:

- one HM to describe the knowledge of the system, i.e. what classes are present at the given coordinates, let us denote it with C;
- and Q_c HMs for the fuzzy membership function values, i.e. at what degree the cell under the given coordinates fall into the category of each class. Let us denote them with μ_c, where c is a given class. Each μ_c only contains the fuzzy values for one class.

For general cases, the output of the classifier is the ID or sequence number of the class with the highest value under the coordinates gained from the input data. The system can also determine the confidence measure (how sure it is in the answer) considering the said value, and is capable of determining the other possible answers if there are any with value higher than 0.

For cases where the instances of one class significantly outnumbers that of the other classes, the former can be considered a *default* class. In this case it is sufficient to only store the HMs for other classes, if the output of the classifier is 0 or close to 0, then it is most likely that the *default* class is the answer.

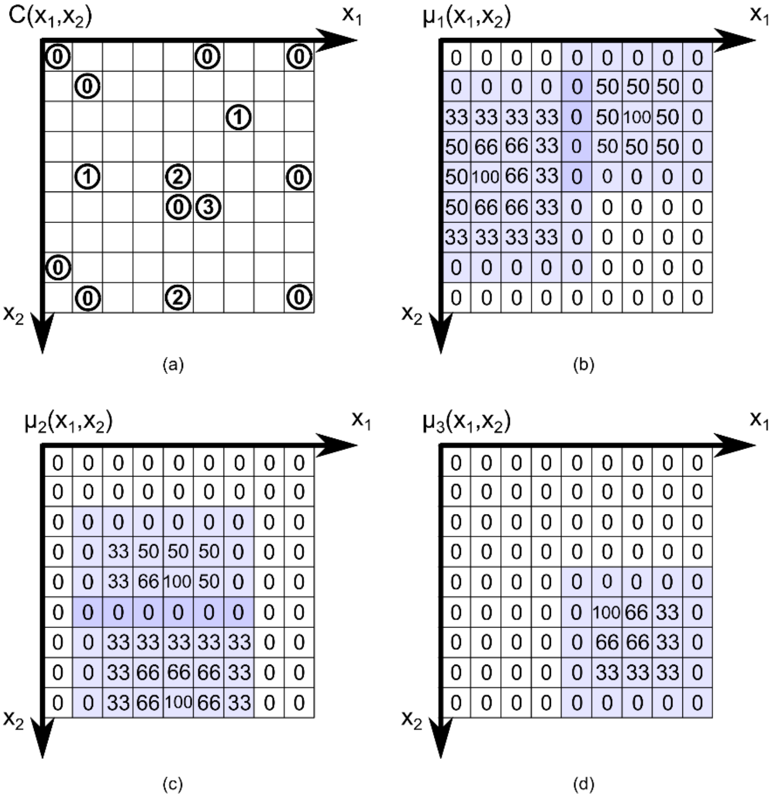
One good example for that is 2 class color filtering, where the goal is to find certain sets of color tones (marked as positive results) while everything else can be regarded as negative. In such cases only the positive class needs a HM, the negative one can be regarded as *default* and thus be discarded. This way the evaluation step is reduced to the accessing of the value under the appropriate coordinates in μ_p and comparing it to an arbitrary threshold parameter (that is used to tune the sensitivity of the system). This idea of course can be extended to the recognition of multiple colors, the background of the training images (i.e. the colors of pixels that are not marked to belong to any of the classes that are needed to be found) can be considered to be *default*.

2.2 General applicability of the classifier

The applicability of the classifier can be easily determined, using the following formula to calculate the amount of memory space (M) that would be required to store the structure:

$$M = (Q_c + 1) \prod_{i=1}^{A} D_i$$

where A is the number of attributes in the input data, Q_c is the number of classes in the problem, and D_i is number of possible integer values that can be taken by attribute i. As it can easily be seen, too many attributes that take too many values can result in an enormous amount of memory need. Let us take the color filtering problem (mentioned in Section 2.1) as an example. In this case A is 3 and D_i is 256 (for all i), thus the structure only takes 16.7 MB of memory space, and additionally 16.7 MB for each distinguished class. (Moreover, the type of the arrays should be accounted for as well. In this research, character type arrays are assumed because they can more efficiently be used to store a value between 0 and 100 using only 1 byte per cell, while floating point numbers (to store real numbers between 0 and 1) would require much more space.)



0					0			0
	0							
						1		
	1			2				0
				0	3			
0								
	0			2				0

(a)

0	0	0	0	0	0	0	0	0
0	0	0	0	0	50	50	50	0
33	33	33	33	0	50	100	50	0
50	66	66	33	0	50	50	50	0
50	100	66	33	0	0	0	0	0
50	66	66	33	0	0	0	0	0
33	33	33	33	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

(b)

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	33	50	50	50	0	0	0
0	0	33	66	100	50	0	0	0
0	0	0	0	0	0	0	0	0
0	0	33	33	33	33	33	0	0
0	0	33	66	66	66	33	0	0
0	0	33	66	100	66	33	0	0

(c)

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	100	66	33	0
0	0	0	0	0	66	66	33	0
0	0	0	0	0	33	33	33	0
0	0	0	0	0	0	0	0	0

(d)

Fig. 1. An example for a 2D FHM classifier with 4 classes (with class 0 considered as a default class): (a) hyper-matrix of the classes and (b,c,d) fuzzy hyper-matrices of the fuzzy membership functions.

2.3 The training of the classifier

The training of the system, in the simplest case can be done statically: first *C* is filled by using all of the available training data.

After that, HMs μ_i are evaluated (for i=1..Q_c) using the contents of *C*. For non-zero class markers in *C*, the area around the marker in the μ_v (where *v* is the class of the marker) is recalculated (using an arbitrary range parameter ρ to restrict the area). In the center of said area the value equals to the highest, at the borders of the area the value is zero, and everywhere in between a simple linear function is used to calculate the value from the center to the border. The advantage of this method lies in its simplicity, while its disadvantage is its poor adaptability, since for further training all the μ_i values have to be recalculated.

Figure 1 shows an example for a 4-class case with 2 attributes (x_1 and x_2). In Fig. 1(a) the model of the problem domain can be seen that was derived from the training data. Class 0 is considered as the default class, and since it does not have to be stored explicitly, it reduces training time that would be spent on calculating HM μ_0. In Figs. 1(b), 1(c) and 1(d) the fuzzy sets of the other 3 classes are represented.

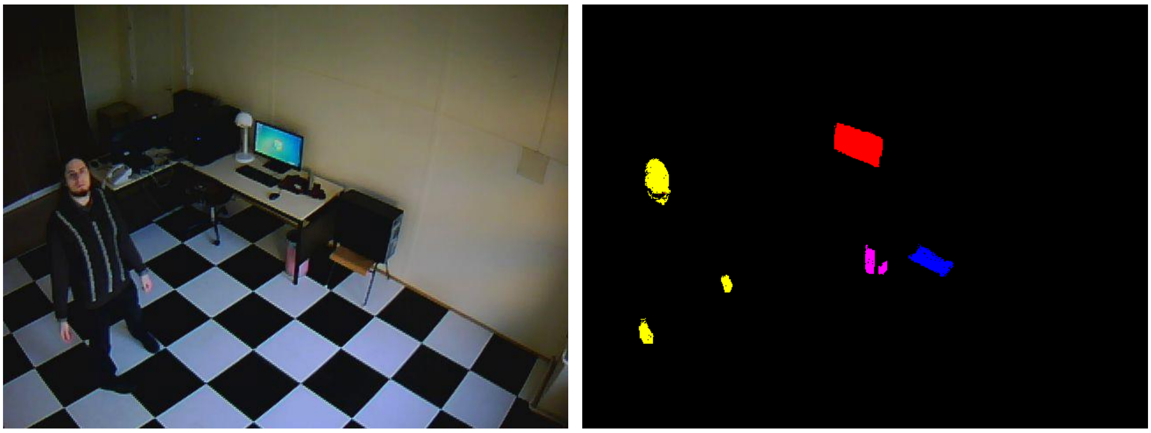


Fig. 2. A training image for the conducted tests (left), and the output of the classifier after the training (using the same image (right)).

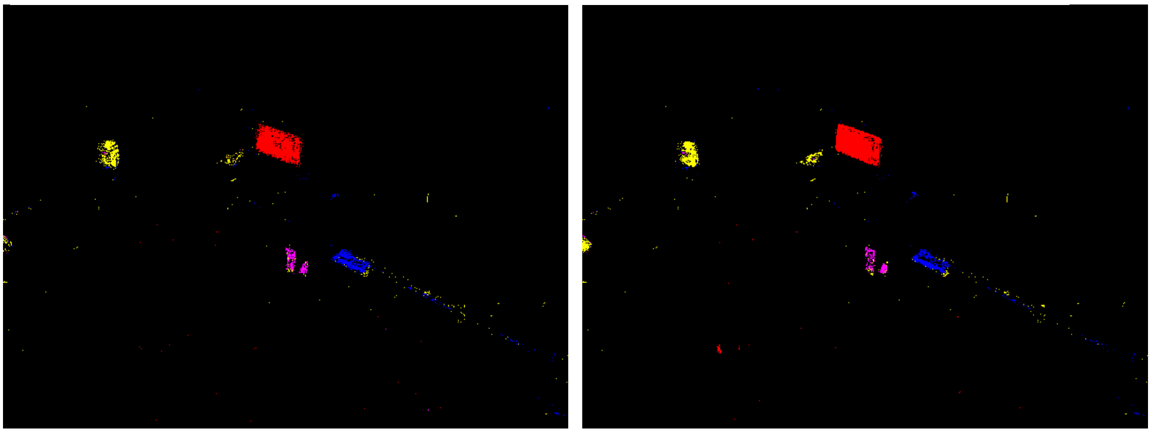


Fig. 3. The classification performance for the FHM$_{\rho=5}$ (left) and FHM$_{\rho=15}$ (right) for a test image.

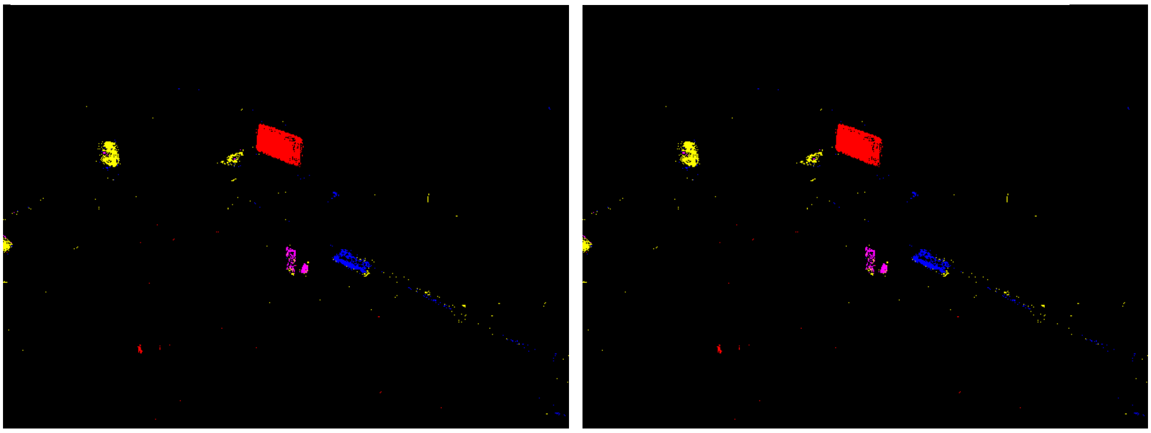


Fig. 4. The classification performance for the FHM$_{\rho=15}$ (left) and FHM$_{\rho=30}$ (right) for a test image.

2.4 Experimental results

In the conducted experiments the classifier has to differentiate the colors of 3 different objects and human skin regions from the background, using images for training where the pixels containing different objects are marked manually (with resolution 640x480, 307200 pixels in total per image). One training image can be seen on the left hand side of Fig. 2, while on the right hand side the figure shows the output of the classifier for that particular image as input. Three different classifiers have been trained using different range parameters (ρ=5, 15, 30). The training times are 5.454, 16.75 and 88.438 seconds per image on average, respectively. As it is expected, the necessary training time scales with ρ. The evaluation time is more or less equally fast, ranging from 0.152 to 0.155 seconds per image on average. Fig. 2 illustrates the difference between the results of $\text{FHM}_{\rho=5}$ and $\text{FHM}_{\rho=15}$ for a test image (where the human user is spreading his arms). As it can be seen, the second one produces a slightly better classification, because the true positive areas are more densely marked, which makes the identification of the areas more reliable (more dense areas are more likely to contain the objects in question). Fig. 4 compares the output of classifiers $\text{FHM}_{\rho=15}$ and $\text{FHM}_{\rho=30}$, with barely any difference in performance, apart from the necessary training time (of which the latter is about 5 times slower than the former). Overall, the classifier with range 15 gives the best performance of the three, being a bit slow during the training but also proving to be more accurate than the ones with lower or higher range.

2. Conclusion

In this paper, a novel fuzzy classifier is presented that uses hyper-matrices for fast data classification. The idea behind the approach is the pre-calculation and storage of the fuzzy membership function values so that they are readily available during the evaluation phrase. The attributes of the input data are used to access the data of the hyper-matrices. This enhances the evaluation speed which is important for many real-time applications. The classifier is most suitable for problems with positive integer data attributes, but it can be used for the classification of continuous data as well. The performance of the classifier has been tested in image processing problems, e.g. to locate human skin areas or finding objects using their color tones, where it provided a quick evaluation process (~0.15 seconds per images with resolution 640x480) with a high accuracy rate. In future work, the determination of the most appropriate values for the arbitrary parameters of the system will be investigated and further research will be done to extend the applicability of the classifier for more complex problems.

Acknowledgment

This work has been supported by the Hungarian National Scientific Fund (OTKA 105846).

References

[1] C.J. Cochrane and J. Blacksberg, IEEE Trans. on Geoscience and Remote Sensing **53**, 4259 (2015).

[2] M. Khodadadzadeh, L. Jun, A. Plaza, and J.M. Bioucas-Dias, Proc. 2015 Joint Urban Remote Sensing Event, Lausanne, 2015.

[3] K. Sergii and Y. Yuriy, Proc. 2014 IEEE 34th Int. Conf. on Electronics and Nanotechnology, Kyiv, 2014.

[4] A.P. Selvam, A. Kumar, and S. Prasad, Proc. 2014 IEEE 14th Int. Conf. on Nanotech., Ontario, 2014.

[5] N. Pattanadech and P. Nimsanong, Proc. 2014 IEEE Region 10 Conference, Bangkok, 2014.

[6] F. Adly, P.D. Yoo, S. Muhaidat, Y. Al-Hammadi, L. Uihyoung, and M. Ismail, IEEE Trans. on Semiconductor Manufacturing **28**, 145 (2015).

[7] P.F. Wagler, U. Tangen, J. Ott, and J.S. McCaskill, IEEE Trans. on Components, Packaging, and Manufacturing Technology **5**, 291 (2015).

[8] B. Tusor, A.R. Várkonyi-Kóczy, M. Takács, and J.T. Tóth, Proc. 2015 IEEE Int. Symp. on Intelligent Signal Processing, Siena, 2015.