

INF???: Machine Learning Final Report

**Skin Segmentation**

Sasha Nicolas

Axelle Pochet

Professor: Ruy Milidiu

Pontifical Catholic University of Rio de Janeiro

Rua Marquês de São Vicente, 225, Gávea, Rio de Janeiro

State of Rio de Janeiro

22451-900

INTRODUCTION = motivations

# The Dataset

<https://archive.ics.uci.edu/ml/datasets/Skin+Segmentation>

3 Attributes B, G, R ranged from 0 to 255.

Y = 1 or 2 respectively Skin or Non-Skin

245057 instances

Not really balanced: 50859 skin against 194198 non-skin.

It was obtained from by randomly sampling face images from different age groups, skin colours and gender from FERET database (http://face.nist.gov/colorferet/request.html) and PAL database (<http://agingmind.utdallas.edu/facedb/>).

Points (R,G,B) were displayed in 3D to allow visual appraisal of the data (Figure 1).

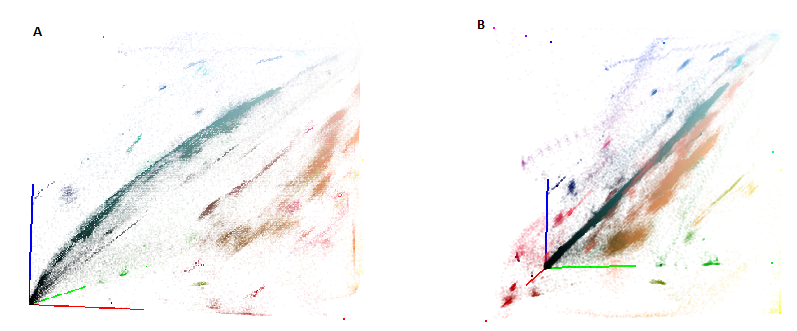


Figure : Dataset in 3D, points coordinates are (R,G,B). Colors of the axis correspond to attributes. (A) View from R axis. (B) View from G axis. Data are gathered in some places (clusters).

# State of the Art

[Bhatt et al., 2009] use a fuzzy decision tree to classify the data.

Indeed, the dataset is fuzzy: skin color varies smoothly between different persons and there is no exact limit between those different colors. Consequently, one point (R,G,B) can belong to several skin types with a certain degree of membership. The term “fuzzy” is a synonym of “blur”: limits between classes are blurred. The fuzzy decision tree allows one point to follow different paths and this represents well this fuzziness.

[Bhatt et al., 2009] arrive at a result of 0,941. Their confusion matrix is given below (Figure 2).

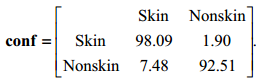


Figure :

# Fuzzy decision tree

## General principle

As introduced in Part 2, fuzzy decision tree takes into account the blur that may occur at the limit between classes. For example, say we have one attribute *Temperature* and we want to know if a given temperature reading is “hot”, “mild” or “cold”. Can we consider a temperature of 15◦C as being mild or cold? This appraisal depends on every person.

Consequently, one data can follow different path in the tree and will eventually be classified with a probability of belonging to the given class.

The first step in building the decision tree is to cluster the data. Here the fuzzy c-mean method was applied. Then the tree is buit on the principle of maximum information gain for every attribute. Finally some rules are extracted in order to read the tree when classifying a new data point.

## Fuzzy clustering

The fuzzy c-mean algorithm was used: it allows one point to belong to several clusters at the same time, with a degree of membership. The method was developed by [Dunn et al., 1973] and consists in minimizing the following objective function:

|  |  |  |
| --- | --- | --- |
|  |  | ( 1 ) |

Where *µij* is the degree of membership of point *xi* to the cluster *j* with center *cj*. *N* is the number of data point and *C* is the number of clusters to extract. *m* is the so-called fuzzifier that will give information on the degree of fuzzification (mixing) of the data. It is generally set to 2 (). The norm here represents the Euclidean distance.

This minimization is carried out through and iterative process that will update *µij* and *cj* until for every *µij*, with *k* the iteration step.

The algorithm is as follows:

Following [Bhatt et al., 2009] the clustering was first done with 5 classes (*C* = 5). Figure 3 shows the cluster centers in the 3D representation and the extracted membership functions drawn as Gaussian curves with parameter σ = 10, for ε = 0.0005.

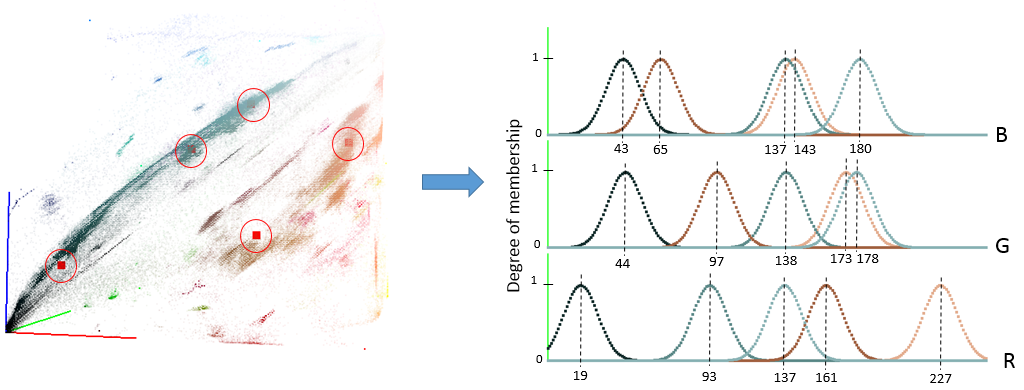


Figure :

Visually, two clusters may correspond to skin color (beige and brown). It seems like there is a wider range of colors for human skin and looking at the initial data it was assumed that more clusters could be found for skin colors. The second clustering then increased to 10 classes. Figure 4 shows the result for *C* = 10, with the same parameters as before.

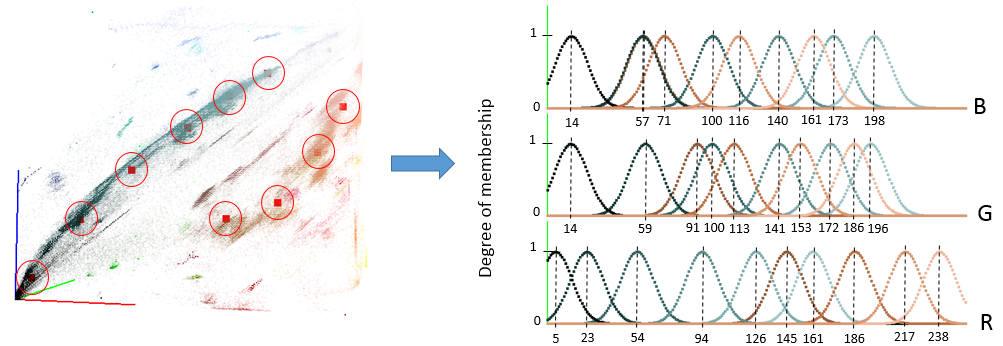


Figure :

The second clustering highlighted new shades for skin color (light beige, dark brown, light brown). However the process was longer, as resumed in Table .... In the following, the two parameterization are used to build the decision tree and show that better results are obtained with 10 clusters.

## Fuzzy decision tree building

A fuzzy decision tree consists in nodes that are attributes, branches that correspond to classes and leaves that correspond to the final decision with its probability to be accurate. An example is given in Figure 5, where attributes are categorical and the concept is easy to understand: the goal is to determine which sport to play according to the current climate.

Decision trees can quickly become hard to follow and to conceptualize with other kind of problems (high number of attributes, classes, ...); the FID3 (Fuzzy Iterative Dichotomiser 3) is one of the algorithms that aim at automatizing the tree construction.

|  |  |  |  |
| --- | --- | --- | --- |
| **Humidity** | **Temperature** | **Wind** | **Sport** |
| Weak | Hot | Not windy | Tennis |
| Strong | Hot | Windy | Football |
| Weak | Hot | Windy | Football |
| Strong | Cold | Windy | Running |
| Weak | Hot | Windy | Running |

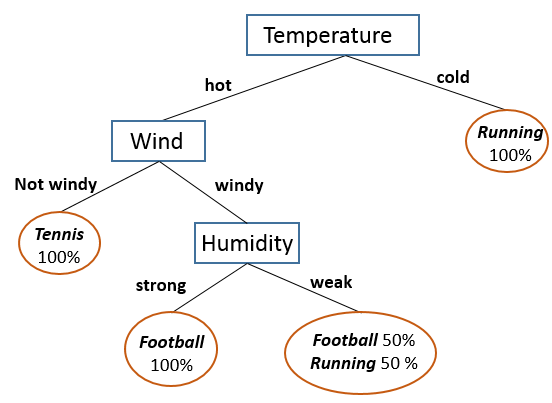


Figure : Attributes are Humidity, Temperature and Wind. Classification concerns the sport to play: Tennis, Football or Running. The reading of the table leads easily to the given decision tree.

In our case, the training table is given by Table 1 in the case of 5 clusters.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **R** | | | | | **G** | | | | | **B** | | | | | **Y** |
| µC1 | µC2 | µC3 | µC4 | µC5 | µC1 | µC2 | µC3 | µC4 | µC5 | µC1 | µC2 | µC3 | µC4 | µC5 | **Skin**/**Non-Skin** |

Table : Attributes are R,G,B. Decision class Y correspond to Skin or Non-Skin. µCi correspond to the degree of membership of the current example to class i.

The tree is built on the principle of maximizing the information gain (reps. minimizing the entropy) at the creation of each node. Equations ( 2 ), ( 3 ) and ( 4 ) give the formulas in the case of fuzzy classification, for a set S={x1, x2,...,xk} with *N* decision classes *c*.

|  |  |  |
| --- | --- | --- |
| **Global Entropy** | , with | ( 2 ) |
| **Attribute Entropy** | ,  with membership value of *j*th pattern at class *i*. | ( 3 ) |
| **Attribute Information Gain** | ,  with the size of subset *Sv* ϵ *S* of training examples *xj* with attribute *v*. | ( 4 ) |

At each node creation, the attribute with the maximum information gain is selected. Another important issue in the tree construction is the condition to create a leaf:

* The β *cut* is the minimal frequency of the given class to stop tree expansion. It is usually given between [0, 0.4] and was here chosen as 0.2.
* The number of remaining training examples at each node has to be above a given *n*. Here it was set to 1.
* The information gain has to be above an ε chosen here as 0.2.

Using those definitions, the iterative algorithm proceeds as follows:

1. Calculate information gain for each remaining attribute;
2. If one of this condition is met :
   1. the frequency of one class is inferior to the β cut,
   2. the number of instances is inferior *n*,
   3. all information gains are inferior to ε;

Then create a leaf and give the frequency of each class.

1. Else select the Attribute with the higher information gain and create a node.
2. Create one branch for each membership class. If no instances are left for one class, do not create a branch for it.
3. For each branch, if some attributes remain, go to step (1).

In the current work, the tree was built with the free software FID3.4 (LINK). To do so, three types of files were generated:

* data.file gives the information about the name, number and type (categorical or numerical) of each attribute, the number of fuzzy subsets with their name, type and range, and the name and type of the decision class.;
* par.template contains all the parameters and options to construct the tree;
* train.file gives the training set with values for each attribute and given decision class. Here we used ... instances for training.

The output files are the following:

* tree.file remains some information on the chosen parameters and details the tree in a compact form;
* test.file is an optional file that gives results of a testing if a test file was given.

Example of each file are given in annex. A visual simplification of the resulting tree for 10 clusters is presented on Figure 6. The tree presents 75 leaves.

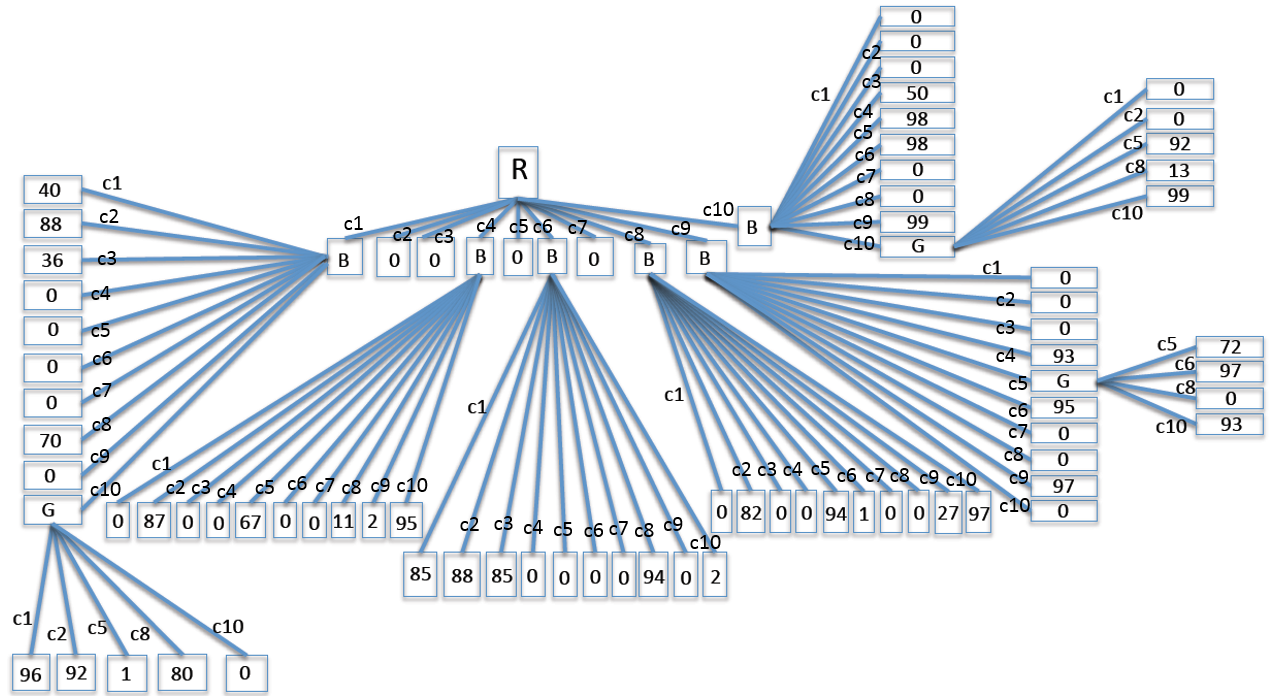


Figure : Resulting fuzzy decision tree with 10 clusters. The attribute with the global maximum information gain is R. Values on the leaf correspond to the frequency of decision class Skin. β-cut =0.2; n = 1; ε = 0.2.

## Testing

# Results

Conclusion

References

Rajen Bhatt, Abhinav Dhall, 'Skin Segmentation Dataset', UCI Machine Learning Repository