Information Theory and Data Mining

Final Project

Work done by:
Bernardo Gomes
Matricola: 0240134

11-02-2017

Table of Contents

Introduction

The aim of this project is to apply measures from information theory to extract any useful information from the downloaded dataset.

Two possible datasets were given as a choice: "Breast Cancer Wisconsin Original" (available at " https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29") and "Mammographic Mass" (available at "https://archive.ics.uci.edu/ml/datasets/Mammographic+Mass"). Since there was the choosing option, the first dataset was chosen for processing.

For the purpose of meaningful information extraction, it is considered the selection of meaningful features and meaningful combination of features. As an upgrade to check if the information was correctly extracted, the developed Matlab program includes the construction of decision trees in order to evaluate the correctness of the obtained results.

Dataset Description

The "Breast Cancer Wisconsin Original" dataset contains eleven features. Each feature is disposed in a column of the given data file in the following order:

1. *Sample Code Number* → contains information of the id number, which makes every row of the matrix unique
2. *Clump Thickness* → attribute range is 1-10;
3. *Uniformity of Cell Size* → attribute range is 1-10;
4. *Uniformity of Cell Shape* → attribute range is 1-10;
5. *Marginal Adhesion* → attribute range is 1-10;
6. *Single Epithelial Cell Size* → attribute range is 1-10;
7. *Bare Nuclei* → attribute range is 1-10;
8. *Bland Chromatin* → attribute range is 1-10;
9. *Normal Nucleoli* → attribute range is 1-10;
10. *Mitoses* → attribute range is 1-10;
11. *Class* → has value "2" if is benign or value "4" if is malign.

It is also known the following information:

- There are sixteen instances from instances 1-6 that contain unknown values/measurements errors that are specified with a "?";
- There are 458 instances representing benign data and 241 representing malign data.

The dataset description is provided on "https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/breast-cancer-wisconsin.names".

Pre-Process of the data

After loading the data, missing values are converted to a non-existing integer "0". Two options are available for the processing of these instances: omission (delete the missing instances) or filling these with values computed based on the other records. Since the data has not yet been processed and according to the instructions given on the assignment, these instances were deleted.

Due to the nature of the data and the same ranges for each feature, it is not needed to perform Normalization, Discretization or Dimensionality Reduction.

À priori, and considering the description provided by the owners of the dataset, it is possible to remove the first column of the data matrix which only purpose is to identify each entry as unique.

Mutual Information (MI)

As seen in class, the Mutual Information between two random variables can be defined as the measure of mutual dependence between them and quantifies the amount of information that can be obtained about a variable through the other. It is highly linked with the concept of entropy of a random variable, which defines the amount of information held in a random variable.

The Mutual Information can be obtained using the following formula:

$$I(X;Y) = \sum_{i=1}^{N_X} \sum_{j=1}^{N_Y} P(x_i; y_j) \log_2 \frac{P(x_i; y_j)}{P(x_i)P(y_j)}$$

This measure of information can also be displayed in a Venn diagram as shown in *figure 1*.

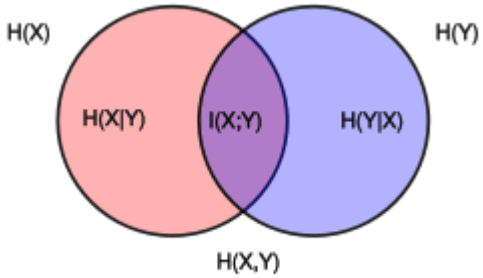

Figure 1 - Venn diagram of Mutual Information

Meaningful Features and Feature Selection

In order to maximize the performance of the data mining method while reducing the number of features, it is important to perform feature selection and identify the meaningful features. Therefore, Mutual Information proves to be a good indicator of relations between the input feature and the target variable.

Meaningful Single Feature evaluation

Meaningful features can be ranked according to their entropy and their MI with the class feature. By running the Matlab code it is possible to obtain the following information about the single features:

Table 1 - Entropy of each feature

Feature	Entropy
Clump Thickness	3.049588
Uniformity of Cell Size	2.343874
Uniformity of Cell Shape	2.489039
Marginal Adhesion	2.212998
Single Epithelial Cell Size	2.290806
Bare Nuclei	1.992490
Bland Chromatin	2.769368
Normal Nucleoli	2.051699
Mitoses	1.129896

Table 2 - Mutual Information between each feature and the class feature

Rank	Feature	MI
1	Uniformity of Cell Size	0.702333
2	Uniformity of Cell Shape	0.676771
3	Bare Nuclei	0.603095
4	Bland Chromatin	0.555260
5	Single Epithelial Cell Size	0.534426
6	Normal Nucleoli	0.487187
7	Marginal Adhesion	0.464424
8	Clump Thickness	0.463995
9	Mitoses	0.211958

According to the obtained information, if only one feature is available, *Uniformity of Cell Size* is the feature that gives the user a most accurate prediction of the class feature. In this case, is the random variable that can by its own given a most accurate prediction about the Breast Cancer being benign or malign.

Although this information is a lot useful, we should also check whether a combination of the available random variables could result in better prediction.

Meaningful Feature combinations

With the purpose of checking the information provided by a group of random variables, there are two possible approaches:

1. Consider that all variables are independent from one another and therefore use the previously obtained MI to group the *n* random variables that are better correlated with the *class*.
2. Consider that the random variables can have some dependency among each other and calculate the MI for each pair of random variables with the class feature.

Since we do not know anything à priori and no information is provided in these terms on the description file given by the dataset creators, the program is built according to the second possibility. Also, this situation is justified since other diseases also cause a group of symptoms that are correlated between them, so breast cancer probably has similar effects on people.

Some aggregations were made and the best MI achieved for each case is displayed on table 3.

Table 3 - Best results achieved in MI for a certain feature group

Number of combined features	Combined Features	MI
2	Uniformity of Cell Size & Bare Nuclei	0.846539
3	Clump Thickness & Uniformity of Cell Size & Bare Nuclei	0.920467
4	Uniformity of Cell Shape & Marginal Adhesion & Bare Nuclei & Normal Nucleoli	0.934003
5	Clump Thickness & Uniformity of Cell Size & Bare Nuclei & Bland Chromatin & Mitoses	0.934003
6	Clump Thickness & Uniformity of Cell Size & Marginal Adhesion & Bare Nuclei & Bland Chromatin & Mitoses	0.934003

In these terms, there should be a compromise between the achieved MI and the number of features. It is possible that a higher number of features, with a higher MI with the class feature gives a worst prediction since the results are too much linked with the data extracted and not the universe of possible results. To do an evaluation of the obtained results, the following graph was plotted:

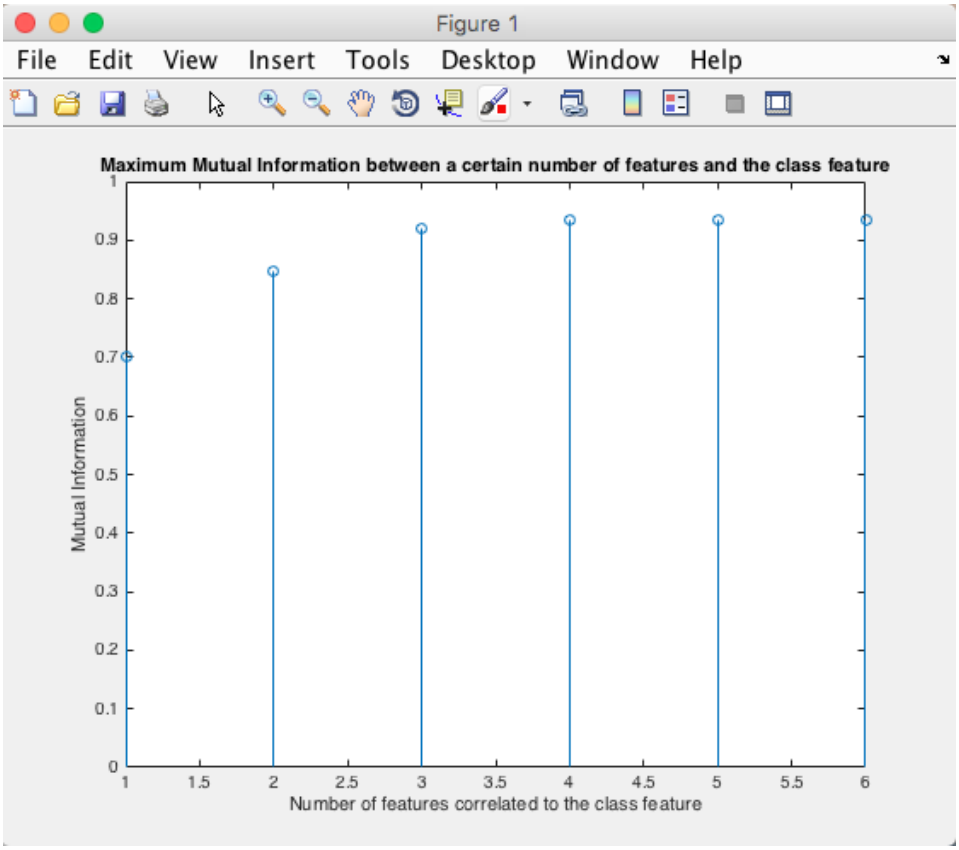

Figure 2 - Relationship between the MI with the class features and the number of features aggregated

There is no bound to the best correlation between the variables and the desired classification. Therefore, the method used to determine the number of features to be used is the "knee method". It is expected that by looking at the figure 2 graph, the existent "knee" at 3 variables indicates that this aggregation gives the needed information to predict if the cancer is malign or benign.

Choosing a number of random variables after the "knee" to describe the class feature might induce error in future classifications, although it provides a higher MI. This way, classification of the training data will be improved but when classifying external data from the general universe, results will have more errors. This phenomenon is called overfitting.

The "knee" method is largely used for PCA and can be transposed to the current problem analysis.

It is possible to conclude through the graph observation that the most meaningful combination of features is: **Clump Thickness & Uniformity of Cell Size & Bare Nuclei**.

Parallel confirmation

Apart from the confirmation that will be performed by the construction of the decision tree, there is a built in function in Matlab to perform feature selection. It was decided by me to include a section of code dedicated to this function that is used only for comparison purposes with the MI method.

The functions used are *classify* and *sequentialfs* that rely on machine learning algorithms. Therefore, data was divided into training set and test set. The information about the functions can be obtained in the respective webpages "https://www.mathworks.com/help/stats/classify.html" and "https://www.mathworks.com/help/stats/sequentialfs.html".

It is important to refer that the existing loop in the code is due to the fact that the *sequentialfs* function relies on the *Monte Carlo method* and therefore is highly likely that two different runs of the function provide different main features. The result used for comparison rely only on the features that were chosen by the function in 75% of the runs. As expected, the features match the ones obtained in the previous method.

No further explanation of this method will be provided since it has only comparison importance and do not match the course contents.

Classification tree

At this point, the relevant features and their best combination was already achieved, which means that the dataset useful information was already extracted. However, to check the feasibility of the information, a decision tree was built.

Matlab Built-in functions test

Decision trees can be easily built using Matlab functions. Since the aim of this section is the proof of the feasibility of the previous results, a classification tree was built instead of a

regression tree. The built-in function name is called *fitctree*. After building the regression tree, the dataset was submitted to the classification through the *predict* function. The results are shown in table 4.

Table 4 - Number of classification errors according to the number of combined features

Number of combined features	Number of classification errors
2	40
3	25
4	22
5	10

The above result might seem that the previously obtained information was not correct since the number of errors keeps to be reduced as a new feature is introduced in the built of the decision tree. However, the reader should keep in mind that the above results have its origin in a tree that was built and tested with the same data points. Therefore, the reduction of the number of errors are a consequence of overfitting and not a result of a better correspondence between the variables and the class feature.

ID3 Algorithm

A decision tree can also be built according to entropy measures. This algorithm is called ID3 and is "a greedy search top-down divide and conquer algorithm to build a decision tree, picking the best attribute at each node and never looking back to reconsider early choices in the past".

Unfortunately, it was not possible for me to concretize this algorithm in Matlab due to deadlines to deliver the project. Matlab is a language directed to programming using vectors and matrices and the best way to code ID3 is using recursive construction of a tree made with pointers and node structures. Although the language provides the possibility to create data structures ("https://www.mathworks.com/help/matlab/ref/struct.html"), it was not possible for me in the given time to overcome the pointers inexistence problem.

The pseudo-code for the given algorithm is the following:

Function ID3

- Create a root node for the tree
- If all members are classified as benign (class=2), return single-node tree root with label 2
- If all members are classified as malign (class=4), return single-node tree root with label 4
- If there are no more features, but members do not belong to the same class, return single node tree root with label=most common value of class feature
- Otherwise:
  A←Attribute that provides better classification
  Decision tree attribute for root=A
  For each possible value of A (k_i) do:
  Add a new tree branch bellow root corresponding to the test A=k_i
  Call ID3 function for each branch with the correspondent new subset of samples (subset of samples that have k_i as a value for A)

Figure 3 - Pseudo-code for ID3 algorithm

The way of checking which attribute/feature that provides better classification is done by checking its MI with the class feature.

Although the algorithm was not implemented, it is expected that it would provide similar results as the ones provided by Matlab built-in functions.

Conclusions

With the development of this project, it was possible to verify that among the available characteristics of the cancer, it is highly feasible to predict whether it is malign or benign by just considering *Clump Thickness*, *Uniformity of Cell Size* and *Bare Nuclei* characteristics.

Regarding the course contents, it was possible to verify the similar results provided by information theory methods and the Matlab built-in functions that rely on Machine learning algorithms.