Imperial College London

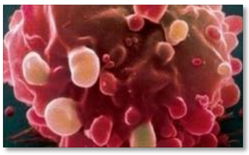
November 28, 2018

MLNC - Machine Learning and Neural Computation, Dr Aldo Faisal

Lab Assignment - Cancer Classification

Bowel cancer classification

Bowel cancer, commonly known as colon cancer or colorectal cancer, results from the uncontrolled cell growth in the large intestine or appendix. Colorectal cancer is the third most commonly diagnosed cancer in the world, but it is more common in industrialised countries. Around 60% of cases were diagnosed in the developed world. About 1.23 million new cases of colorectal cancer are clinically diagnosed each year, and killed 608,000 people per annum. Thus, your mission is to develop an automated way of diagnosing bowel cancer from clinical data, so as to free up time of clinicians for primary care, reduce cost and increase throughput of patient tests. To this end you have been provided with a Bowel Cancer data set, from a recent clinical screen. Expert clinicians manually annotated this data.



We are going to use a standard benchmark for classification, the detection of bowel cancer from microscope images of tumor tissue samples. Each of these tissue samples contains cells which have been used to generate a 30 dimensional real valued vector. Features include parameters of cellular form such as:

- radius (mean of distances from center to points on the perimeter)
- texture (standard deviation of gray-scale values)
- perimeter
- area
- smoothness (local variation in radius lengths)
- compactness ($\frac{perimeter^2}{area-1}$)
- concavity (severity of concave portions of the contour)
- concave points (number of concave portions of the contour)
- symmetry
- fractal dimension (how jagged is the contour?)

Each data points belongs to only one of two classes: malign or benign. The feature descriptions do not ultimately matter, but may help you in getting a feeling for the data and its two classes (Can you perhaps even ”spot” a rule for classification?).

The data is organised inside the structure `cancer` with two fields. 1. A `cancer.input` matrix of feature row vectors extracted from screening images of cells. There are 30 real valued feature per data point. A binary column vector `cancer.output`, with the expert annotations of the images 0=malign, 1=benign diagnosis.

Assignment Use any two classifiers covered in the course to solve classification problem e.g. k-nearest neighbours or the generative classification approach (a Gaussian per class). For each of the two classifiers do the following, where X stands for 1 or 2 respectively.

1. Calculate (by hand) the number of parameters for the classifier. Briefly discuss if you need to take into account unequal number of training data points for each class.

2. Write a function `ClassifyX` that takes as argument an `input` vector and a matrix/vector of `parameters` and returns the predicted `class`. Your first line in the file `ClassifyX.m` should look like this (do not change this format): `function class = ClassifyX(input, parameters)`

3. Write a function `TrainClassifierX` that takes as arguments an `input` matrix of data points (each row one data point, the columns correspond to the dimensions of the data point) and the training class labels (desired class labels) as a column vector of numbers. The function should return the parameters required by your `ClassifyX` function.

   Your first line in the file `ClassifierX.m` should look like this (do not change this format): `function parameters = TrainsClassifierX(inputs, output)` (Sanity check: Your should be able to run `testLabel = Classify(testPoint,TrainClassifierX(inputs, output)` your command and get 0 or 1 answer in `testLabel` for e.g. a `testPoint = zeros(1,30)`.

4. Test your systems performance by splitting the data into equally sized training and test data set. Use only the training data to traing your classifier. Report your classifier's performance (% of correct) on the test data set. Report the confusion matrix[1] If you have hyperparameters use also a validation data set.

5. Compare the results between the two classifiers you implemented. Explain what properties of the data are responsible for each of your classifiers performance. Which one do you expect to generalise better?

[1] Each column of the matrix represents the instances (%) in a predicted class while each row represents the instances in an actual class (%).