LAB 5 – Unsupervised/supervised Learning

Authors :

Ricardo Rei	78047
Luis Henriques	77919
Luis Nunes	77940

Question 1:

The K Means clusters observations into k groups, where k is provided as an input parameter. The following pseudo-code illustrates how this algorithm works:

1) Initialization of k centroids.
2) For each point in the dataset we compute the distance to each centroid and assign the point to the cluster that has a closer centroid to the point.
3) Each centroid is recomputed as the average of the points in that cluster.
4) Repeat step 2 and 3 until the clusters converge or until it reaches a maximum number of iterations.

There are different ways of initializing the centroids in step 1. We used the Forgy method. This method randomly chooses k observations from the data set. Also there are different ways of computing the distance between an observation and a centroid. We used Euclidean distance as the default distance metric but it is possible in our code to change to other metrics.

Since this algorithm is completely Unsupervised Learning and it uses a random initialization the results can be extremely different each time we run it. In order to avoid this issue, we allow, in our code, the possibility of setting a fixed seed.

1.1)

Our main module lab5.py receives 4 input parameters: three integers and a file name. The order of the parameter is the following:

1) G – number of clusters.
2) R – number or rows to consider.
3) C – number of columns to consider.
4) Filename – the name of the file that contains the dataset (arff format)

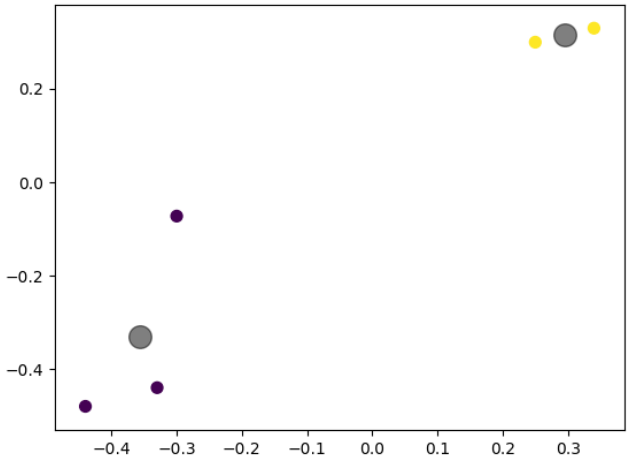
After calling the lab5 module with the correct parameter this module starts by parsing the dataset file with the help of the loadarff[1] function from scipy. Note that some observations in the dataset might have missing values (?). To avoid missing values, we compute de average value for each column and set the missing values to the

[1] https://docs.scipy.org/doc/scipy/reference/generated/scipy.io.arff.loadarff.html

average of the column in which they appear. We also separate the observation values from the observation labels given.

Lastly the lab5 module initializes an instance of the kmeans class from our kmeans.py module with k = G and trains it with all the observation values and a seed = 115 (later we explain why 115, but in this problem the seed value is not relevant). The results obtained are:

- Accuracy: 100%
- Precision: 100%
- Recall: 100%



Note: This example is simple and it is possible to compute precision, recall and accuracy but in a typical Unsupervised approach it doesn't make much sense to do it.

Question 2.1:

a)

The centroids reported by Weka Package are stored in centroids_weka.txt file.

b)

For this problem giving the complexity and the number of features presented the initialization of the centroids is extremely important. Depending on the seed used for the random initialization our results can be completely different.

In order to improve our algorithm for this particular exercise, and since we had the correct labels of each point, we tested different seed values and choose the one with better accuracy. Seed value of 115 gave a total accuracy of 86.66 %, a precision of 78.57% and a recall of 100%.

Running our K-Means algorithm for the dataset we obtained 17 points assigned to the first centroid (GCL) and 28 points assigned to the second centroid (ACL).
Weka K-Means algorithm assigned 22 points to the first centroid and 23 points to the second centroid.

In the dataset there is a total of 22 points corresponding to GCL and a total of 23 points corresponding to ACL which means that the Weka algorithm made the correct separation of the data and our algorithm didn't.
Weka centroids using a Euclidean distance were at 63.65 distance and our centroids using the same metric were only at a 23.84 distance.

Actually the distance from our GCL centroid to Weka GCL centroid is bigger than the distance between our 2 centroids, with a total value of 42.11. For the ACL centroids the result was the same with a total distance between our centroid and Weka centroid of 38.42.

Question 2.2:

a)

J48:

```
Instances:    45
Attributes:   4027
              [list of attributes omitted]
Test mode:    6-fold cross-validation

=== Classifier model (full training set) ===

J48 pruned tree
------------------

GENE3330X <= 0.38
|   GENE518X <= -0.23: GCL (2.0)
|   GENE518X > -0.23: ACL (22.0)
GENE3330X > 0.38: GCL (21.0/1.0)

Number of Leaves  :     3

Size of the tree :      5


Time taken to build model: 0.3 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          32               71.1111 %
Incorrectly Classified Instances        13               28.8889 %
Kappa statistic                          0.4236
Mean absolute error                      0.313
Root mean squared error                  0.5356
Relative absolute error                 62.524  %
Root relative squared error            106.9655 %
Total Number of Instances               45

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0.773    0.348    0.680      0.773    0.723      0.427    0.685     0.616     GCL
               0.652    0.227    0.750      0.652    0.698      0.427    0.685     0.660     ACL
Weighted Avg.  0.711    0.286    0.716      0.711    0.710      0.427    0.685     0.638

=== Confusion Matrix ===

  a  b   <-- classified as
 17  5 |  a = GCL
  8 15 |  b = ACL
```

The Precision of the J48 decision Tree is 71.6%

Naïve Bayes:

```
Time taken to build model: 0.16 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances          40               88.8889 %
Incorrectly Classified Instances         5               11.1111 %
Kappa statistic                          0.777
Mean absolute error                      0.1128
Root mean squared error                  0.3335
Relative absolute error                 22.5385 %
Root relative squared error             66.6073 %
Total Number of Instances               45

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0.818    0.043    0.947      0.818    0.878      0.784    0.950     0.926     GCL
               0.957    0.182    0.846      0.957    0.898      0.784    0.912     0.867     ACL
Weighted Avg.  0.889    0.114    0.896      0.889    0.888      0.784    0.930     0.896

=== Confusion Matrix ===

  a  b   <-- classified as
 18  4 |  a = GCL
  1 22 |  b = ACL
```

The Precision of the Naïve Bayes classifier is 89.6%

b)

J48 is the implementation of a Decision Tree algorithm called ID3 (Iterative Dichotomiser 3) developed by WEKA project team.
ID3 algorithm uses a top-down approach to create the final tree.
The pseudo-code is the following:

1. Scan the dataset and compute the entropy
2. Split the dataset using the attribute with the highest Gain of Information (minimum entropy).
3. Make a decision node with that attribute.
4. Repeat for the remaining subsets.

The **Naïve Bayes** Classifier technique is based on the so-called Bayesian theorem and is particularly suited when the dimensionality of the inputs is high. Despite its simplicity, Naive Bayes can often outperform more sophisticated classification methods.
Given a set of variables, $X = \{x_1, x_2, \dots, x_n\}$, we want to construct the posterior probability for the event Cj among a set of possible outcomes $C = \{c_1, c_2, \dots, c_n\}$, In a more familiar language, X is the predictors and C is the set of categorical levels present in the dependent variable.

Using Bayes' rule, the probability of given a set of variables X belong to a class C_j is given by:

$$P(C_j | x_1, x_2, \dots, x_n) \propto P(x_1, x_2, \dots, x_n \mid C_j) * P(C_j)$$

Since Naïve Bayes assumes that variables are independent:

$$P(X \mid C_j) \propto \prod_{k=1}^{n} P(x_k \mid C_j)$$

Rewriting it again, the probability of a given class given a set of variables X is given by:

$$P(C_j | x_1, x_2, \dots, x_n) \propto P(C_j) * \prod_{k=1}^{n} P(x_k \mid C_j)$$

Using Bayes' rule above, we label a new case X with a class level Cj that achieves the highest posterior probability.

If our variables are continuous we need to compute the distribution associated with each variable (normally we assume it's a Normal distribution) and the probability of a variable x_k given a class C_j is given by:

$$P(x_k|C_j) = \frac{1}{\sigma_{kj}\sqrt{2\pi}} \exp\left(-\frac{\left(x_k - \mu_{kj}\right)^2}{2{\sigma_{kj}}^2}\right)$$

Where μ_{kj} is the mean of the feature values x_k in which $C = C_j$and σ_{kj} is the standard deviation.

The rest remains the same.

c)

The clustering method used in 2.1 obtained a 100% precision. These methods had less precision.

Question 3:

a) Train:

Running nose:

- $P\,(\,running\ nose\,+\,|\,classification\,+)\,=\frac{2}{3}$
- $P\,(\,running\ nose\,-\,|\,classification\,+)\,=\frac{1}{3}$
- $P\,(\,running\ nose\,+\,|\,classification\,-)\,=\frac{1}{3}$
- $P\,(\,running\ nose\,-\,|\,classification\,-)\,=\frac{2}{3}$

Coughing:

- $P\,(\,coughing\,+\,|\,classification\,+)\,=\frac{2}{3}$
- $P\,(\,coughing\,-\,|\,classification\,+)\,=\frac{1}{3}$
- $P\,(\,coughing\,+\,|\,classification\,-)\,=\frac{1}{3}$
- $P\,(\,coughing\,-\,|\,classification\,-)\,=\frac{2}{3}$

Reddened skin:

- $P\,(\,reddened\ skin\,+\,|\,classification\,+)\,=\frac{2}{3}$
- $P\,(\,reddened\ skin\,-\,|\,classification\,+)\,=\frac{1}{3}$
- $P\,(\,reddened\ skin\,+\,|\,classification\,-)\,=\frac{1}{3}$
- $P\,(\,reddened\ skin\,-\,|\,classification-)\,=\frac{2}{3}$

Fever:

- $P\,(\,fever\,+\,|\,classification\,+)\,=\frac{1}{1}$
- $P\,(\,fever\,-\,|\,classification\,+)\,=\frac{2}{5}$
- $P\,(\,fever\,+\,|\,classification\,-)\,=0$
- $P\,(\,fever\,-\,|\,classification\,-)\,=\frac{3}{5}$

b) Test:

In order to classify a sample with a missing feature we can ignore it. Ignoring a feature value is the same as if we assume any possible value for that particular feature which gives a probability of 1.

$$D7 = \{\, running\ nose -, reddened\ skin\ - \}$$

$$P(+|D7) = P(+) * P(running\ nose -|+) * P(reddened\ skin - |+) = \frac{1}{2} * \frac{1}{3} * \frac{1}{3} \cong 0.056$$

$$P(-|D7) = P(-) * P(running\ nose -|-) * P(reddened\ skin - |-) = \frac{1}{2} * \frac{2}{3} * \frac{2}{3} \cong 0.222$$

$$0.222 > 0.056 \implies D7\ classification\ is\ negative\ (healthy)$$

$$D8 = \{coughing -, reddened\ skin - \}$$

$$P(+|D8) = P(+) * P(coughing -|+) * P(reddened\ skin - |+) = \frac{1}{2} * \frac{1}{3} * \frac{1}{3} \cong 0.056$$

$$P(-|D8) = P(-) * P(coughing -|-) * P(reddened\ skin - |-) = \frac{1}{2} * \frac{2}{3} * \frac{2}{3} \cong 0.222$$

$$0.222 > 0.056 \implies D8\ classification\ is\ negative\ (healthy)$$

$$D9 = \{coughing -, fever - \}$$

$$P(+|D9) = P(+) * P(coughing -|+) * P(fever - |+) = \frac{1}{2} * \frac{1}{3} * \frac{2}{5} \cong 0.067$$

$$P(-|D9) = P(-) * P(coughing -|-) * P(fever - |-) = \frac{1}{2} * \frac{2}{3} * \frac{3}{5} \cong 0.2$$

$$0.2 > 0.067 \implies D9\ classification\ is\ negative\ (healthy)$$