Exercises of LDA classifier

Pablo García Fernández
(pablo.garcia.fernandez2@rai.usc.es)

Saturday 15th January, 2022

Abstract

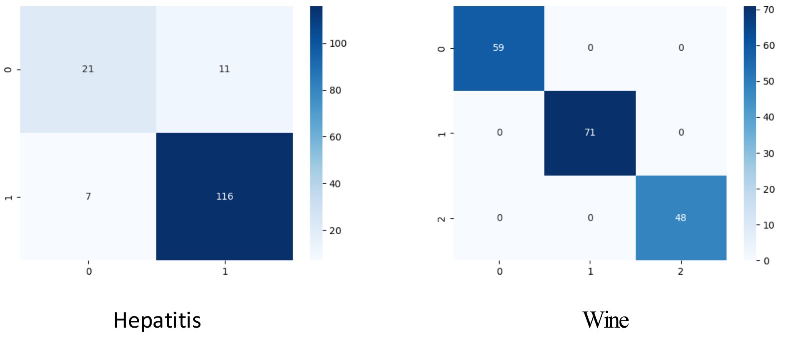
The aim of this report is to present the work carried out during the execution of the exercises of the unit 5 assignment, as well as the obtained results. In addition to this report, we also attach the developed .py code, the obtained figures and a .txt file with the results.

Exercise 2: *Calculate acc., kappa and cm for both datasets using LDA and the whole dataset as training and test set.*

We start from the files *hepatitis.data* and *wine.data*. Hepatitis is a 2-class classification problem (this means that precision, recall and f-score metrics can be obtained). It has 155 patterns and each pattern 19 features. The number of samples of class 1 is 32; and of class 2, 123. Wine is a 3-class classification task. It has 178 13-dimensional patterns with class populations 59, 71 and 48. Applying LDA on both datasets using all samples for training and testing, we obtain the following results. ("Tab. 1" and "Fig. 1").

Table 1: LDA results using whole dataset for training and testing

Dataset	**Metrics**				
	acc. (%)	**kappa (%)**	**precision (%)**	**recall (%)**	**f1 (%)**
Hepatitis.data	88.39	62.84	91.34	94.31	92.80
Wine.data	100.0	100.0	-	-	-



Hepatitis

Wine

Figure 1: LDA confusion matrices using whole dataset for training and testing

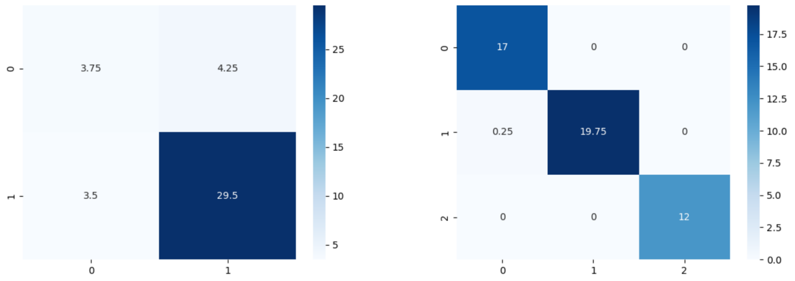
Training and testing with the same data is not a good way to evaluate a model because the results will be positively biased. That is, they will be better than they really are. In fact, in “Tab. 1” we can see how the kappa and precision metrics on the wine dataset are 100% (perfect distinction between the 3 classes), which is almost impossible for non-trivial problems.

Exercise 3: *Repeat the process using cross-validation with 4 folds*

“Fig. 2” shows the metrics obtained by applying a 4-fold cross-validation process on each of the datasets. The corresponding confusion matrices are shown in “Fig. 2”. As no hyperparameters need to be adjusted in LDA, the training and validation partitions obtained with the `cread_folds()` function must be grouped. Since 4 groups were selected in k-folds, 4 training/testing runs are performed (rotating in each one the group used for the test). Finally, the resulting metrics are the averages of the 4 runs.

Table 2: LDA results with 4-folds cross validation

Dataset	**Metrics**				
	acc. (%)	**kappa (%)**	**precision (%)**	**recall (%)**	**f1 (%)**
Hepatitis.data	81.10	35.83	87.71	89.39	88.35
Wine.data	99.49	99.22	-	-	-



Hepatitis

Wine

Figure 2: LDA confusion matrices with 4-folds cross validation

By comparing “Tabs. 1, 2” we can see how the evaluation results are now lower (particularly pronounced for the hepatitis dataset). As we have already said, this is normal because the results of table 1 are biased upwards.

Another interesting thing is that for the hepatitis dataset, the kappa is much lower than the accuracy. This is because the accuracy is very sensitive to the imbalance between classes. If we look at the distribution of classes inside hepatitis.data, we see that there are many more examples of “0” class (32) than of “1” class (123).

Exercise 4: Repeat using leave-one-pattern-out cross validation

In a LOOCV approach, the training/testing loop is repeated N times, where N is the number of samples in the dataset. In each iteration, the model (in this case LDA) is trained with N-1 patterns and tested with the remaining one. The tested pattern is rotated over the N iterations. The final metrics are the average over the N runs.

This process is manually implemented in the attached *2-lda.py* file, specifically in the `ej4()` function. To check that the result obtained is correct, we also re-run this exercise using scikit-learn's `LeaveOneOut()` function. The obtained accuracies are the same ("Tab. 3").

Table 3: LDA results with LOOCV

Dataset	**Manually implemented**	**Scikit-learn function**
	acc. (%)	**acc. (%)**
Hepatitis.data	80.00	80.00
Wine.data	98.88	98.88

Exercise 5: Use LDA classifier in CooCurr and LBP datasets.

Since these datasets are already divided into a training set and a test set, cross-validation may not be applied (however, we also include its results in the following table). Without cross-validation the LDA model is trained directly with the training patterns and evaluated with the test patterns. In addition, since the number of classes is greater than two, only the accuracy, kappa and confusion matrix are reported.

As in all the exercises, patterns are standardized with the mean and the standard deviation of the training patterns. This is important because if the features of the different samples have different scales, those that are larger have a greater influence on the decision, which causes bias in the results.

"Tab. 4" shows the results.

Table 4: LDA over CooCur and LBP datasets.

Not cross validation applied

Dataset	**No. features**	**Accuracy (%)**	**Kappa (%)**
Co-occurrence matrices	12	87.96	87.74
Local binary patterns	10	87.04	86.79

4-folds cross validation results

Dataset	**No. features**	**Accuracy (%)**	**Kappa (%)**
Co-occurrence matrices	12	85.42	85.14
Local binary patterns	10	85.65	85.38

We can see that both feature descriptors have sufficient discriminatory power to discern between different textures. Interestingly, without cross-validation, CooCurr obtains slightly better results than LBP; and with 4-fold cross-validation, it is the other way around.

Finally, “Fig. 3” presents the confusion matrices for the LDA clasification without cross-validation.

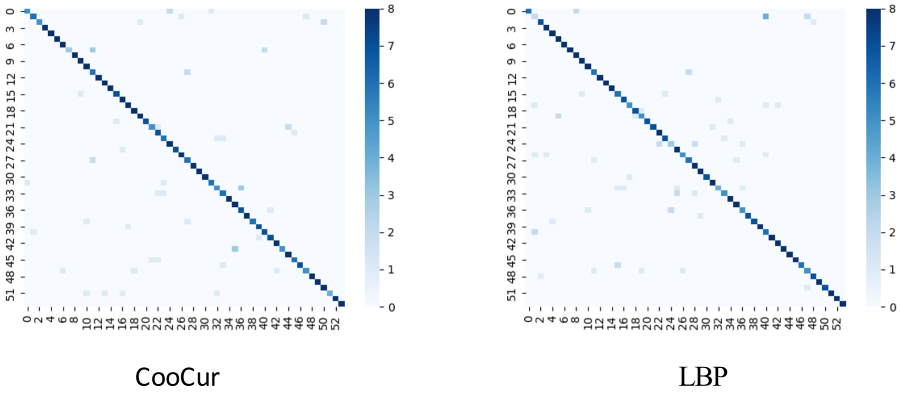


Figure 3: Confusion matrix for CooCur and LBP. LDA