FTML practical session 14: 2023/06/15

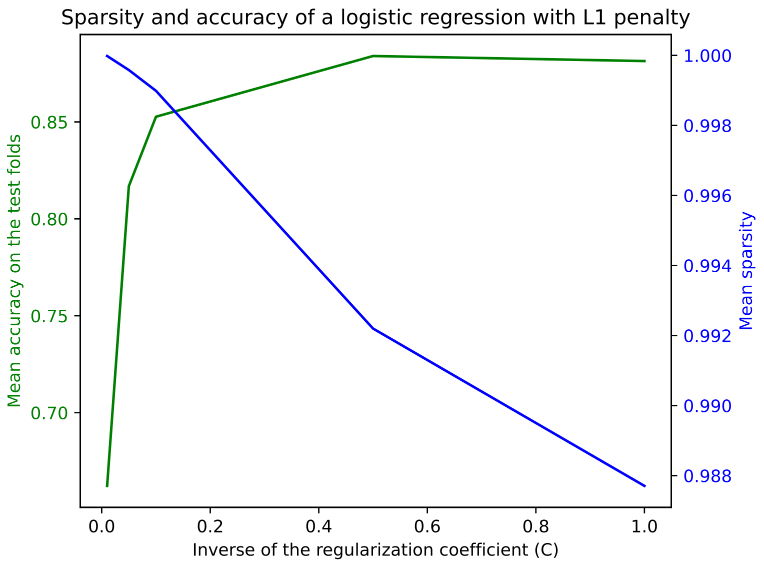

TABLE DES MATIÈRES

1 FEATURE SELECTION PART II

We first finish working with the dataset from the previous session, exploring alternative methods in order to obtain sparse estimators.

1.1 Embedded algorithm

Perform a Pipieline like the one from the first exercise of the former session (one-hot encoding, scaling, logistic regression), but with a L1 regularization in the logistic regression step, similar to the Lasso.

Save the vocabulary effectively used by the obtained estimator, along with the weights corresponding to each word. Rank the used words according to the weight, and interpret the meaning of this ranking. Evaluate the influence of the regularization parameter C (which corresponds to the inverse of the regularization strength) on the sparsity and on the accuracy (e.g. by plotting the cross validated accuracry).

File : **exercice_3_l1_regularization.py**

1.2 Wrappers

As a last method, we will now use a wrapper, in order to reduce the number of features used by the estimators. When using a wrapper, we score feature subsets, based on a previously trained estimator (as opposed to univariate filtering, where we remove features of the inputs without taking an estimator into account). We will use the recursive feature elimination (RFE) method, which is one possibility out of several other available choices. Features are removed iteratively : at each iteration, a chosen number of features is removed (this is a parameter of the algorithm), and the estimator is trained again. This iteration is performed until the number of features kept is equal to a specified number (which is another parameter of the RFE).

By default, in the case of a linear estimator, the score for each feature used by the scikit implementation of RFE is the squared value of the coefficient that corresponds to this feature. In **utils_data_processing.py**, there is a function **LinearPipeline** that makes it possible to apply RFE to a pipeline directly, with this same feature scoring.

Apply RFE with a chosen number of final features and a step size, and save the vocabulary used to a file, again sorted according to the weights.

With this method on this example, it is possible to keep the 0.9 test accuracy with a vocabulary of 10000 words.
File : **exercice_5_wrapper.py**

1.2.1 *Ressources*

https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html#sklearn.feature_selection.RFE

1.3 Conclusion

We have seen several methods in order to obtain a sparse estimator for this binary classification task. These methods lead to slightly different vocabularies being used, but all work well as they keep an accurarcy that is as high or almost as high as the vanilla logistic regression that uses all the vocabulary.

2 DENSITY ESTIMATION WITH GAUSSIAN MIXTURES

In this exercise, we will learn a probability distribution from a dataset (density estimation), here the digits dataset. We use Gaussian mixtures as parametric models : as we haven't discussed them during the lectures, you will need to read some documentation to understant them and to explore the meaning of some parameters.

2.1 Density estimation

One of the main parameters of a Gaussian mixture is its number of components. Intuitively, it can be seen as an analogous to the number of centroids in a k-means clustering algorighm.

Estimate a Gaussian mixture distribution from the dataset, choosing the number of components that minimizes the Akaike information criterion (AIC) (in order to save compoutation time, you might try stepped numbers of components, below 200). Plot the mean of each component.

2.2 Generation of digits

One interesting aspect of having fitted a probabilistic model to a dataset, is that we can sample this model !

Use your previously learned Gaussian mixture model in order to generate images and compare them to images from the dataset. Explore the influence of some chosen hyperparameters on the generated images.

2.3 Ressources

https://scikit-learn.org/stable/modules/mixture.html#gmm
https://fr.wikipedia.org/wiki/Crit%C3%A8re_d%27information_d%27Akaike