Introduction to AI: assignment 3 - EUR DS4H

Logistic regression

Part I: A simple example

A. Linear separator from logistic regression

Consider the following training data:

Class - y	$x^{(1)}$	$x^{(2)}$
1	1	1
1	2	2
1	2	0
0	0	0
0	1	0
0	0	1

1. Draw a scatter plot of these 6 data points. Are these classes separable by a linear classifier?

2. A regularized logistic regression model has been learned from these data. The model parameters are $\beta_0 = -28.55$ (related to the implicit constant feature), $\beta_1 = 19.64$ and $\beta_2 = 18.06$. The linear equation specifying the predicted separating line between the two classes can be written as

$$x^{(2)} = ax^{(1)} + b \tag{1}$$

   for some a and b. Based on the values of β_0, β_1 and β_2 give the values of a and b, then draw approximately the separating line on the scatter plot drawn from the previous question.

3. Based on the drawing, give the accuracy of the corresponding classifier on the training data $\text{Acc}_{\hat{\mathbf{y}}(\mathbf{x})}(\mathbf{X}, \mathbf{y})$ and its corresponding confusion matrix.

4. A new point has been added in the dataset, its feature values are $x^{(1)}_{\text{new}} = 1.25$, $x^{(2)}_{\text{new}} = 1$ and its class is $y_{\text{new}} = 0$. Add this point to the previous scatter plot. Is the updated data set separable by a linear classifier?

5. A logistic regression model has been learned from the updated dataset, its parameters are $\beta_0 = -5.85$, $\beta_1 = 3.87$ and $\beta_2 = 1.50$. Give the values of a and b of the separating line equation and draw approximately the line on the update scatter plot.

6. Give the new values of the accuracy on the training data and the new confusion matrix.

B. Effect of the probability threshold

In binary logistic regression, the output of the logistic function can be interpreted as the probability or the confidence of an observation belonging to class $y = 1$. During the classes, we have set a threshold $p = 0.5$ on this probability for deciding whether the predicted class will be $\hat{y} = 1$ or $\hat{y} = 0$.

1. Intuitively, what would be the effect on the classifier of choosing $p > 0.5$ or $p < 0.5$?
2. Give the expression of a and b in (1) as a function β_0, β_1, β_2 and p. Using the parameter values of A.5, give the values of a and b for both $p = 0.25$ and $p = 0.75$.
3. Draw approximately the separating lines for $p = 0.25$ and $p = 0.75$ and, in each case, give the corresponding accuracy and confusion matrix. Are the results in accordance with intuition?

Part II: Logistic regression and SVM applied to handwritten digit recognition

In this part you are going to use the *Digits* data set which is already present in Scikit-learn[1]. It corresponds to labeled images of handwritten digits from 0 to 9. The images have already been preprocessed: the digits are centered, scaled and the amplitudes of the pixels are normalized between 0 and 16. Each image has 8 pixels of height and 8 pixels of width.

[1] This data set is a part of a larger one which can be found at `http://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits`

The objective of this part of the assignment is to test logistic regression to do binary classification on data from two digits only. Since the feature space is large[2], to better visualize the obtained classifier, you are going to reduce the dimensionality of the features to 2 prior to classifier learning. To do so, you are going to use a dimensionality reduction method called principal component analysis (PCA).

[2] What is its size?

A. Loading and visualizing the data set for two given classes

First, you are going to read and visualize the data for the classes C_0 and C_1 corresponding to digits 0 and 1 respectively.

1. Import *datasets* from *sklearn* and also the library *numpy*.

2. Load the entire data set with the command

```
digits = datasets.load_digits()
```

The features are the vectorized images and they are given in *digits.data*, while the output labels are given in *digits.target*.

3. Print the sizes of the feature matrix and of the output vector. What is the total number of observations in this dataset?

4. Print the matrix of features and the outputs.

5. Define the two classes labels that we are going to visualize $C_0 = 0$ and $C_1 = 1$.

6. Define in separate arrays the feature matrices $\mathbf{X}_0$, $\mathbf{X}_1$ and the outputs $\mathbf{y}_0$ and $\mathbf{y}_1$ for the 2 classes.

7. Import *matplotlib.pyplot* and generate 2 subplots, one subplot with an image from the first class and one subplot with an image from the second class. Use the command *imshow* from *pyplot* to display the images. You should see a figure similar to Fig. 1.

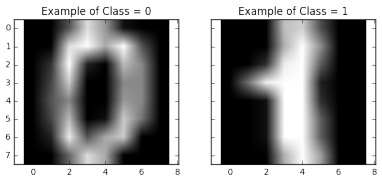

Figure 1: One image from each class.

8. Stack the previously defined features matrices and outputs to have only 1 feature matrix $\mathbf{X}$ and only 1 output vector $\mathbf{y}$.

9. Import the dimensionality reduction method PCA with

```
from sklearn.decomposition import PCA
```

Look at Scikit-learn documentation to see how it should be applied to retrieve only 2 features representing most of the variation of the data.

10. Visualize the data from the 2 classes in a $2D$ scatter plot using the features given by PCA. You should obtain a figure similar to Fig. 2. Are the classes linearly separable?

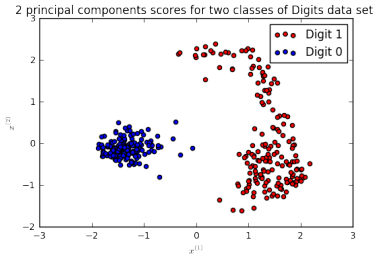

Figure 2: Scatter plot of features obtained after dimensionality reduction with PCA.

B. Logistic regression for digits 0 *and* 1

1. Import logistic regression from Scikit-learn (*linear_model*).

2. Define a logistic regression model with the command

```
logistic = linear_model.LogisticRegression(penalty="l2", C=1e6)
```

The logistic regression model implemented in Scikit-learn includes a regularization term on the l_2-norm of $\boldsymbol{\beta}'$ in order to avoid diverging solutions when solving the optimization problem corresponding to parameter learning. The parameter C divides this regularization term, so if you want to approximately test the standard logistic regression method (without regularization), you should set C to a high value.

3. Fit the logistic regression model to the $2D$ features, predict on the training data and compare with the training output labels.

4. Evaluate the accuracy of the prediction on the training data with the fitted logistic regression model. Print the result. Does it correspond to the expected result?

5. By using the method *predict_proba* from logistic regression model and *pyplot*'s command contour, draw the separating line. Draw the scatter plot with the observations on the top of the separating line.

6. Display the confusion matrix for this classifier when it is applied to the training data. A function for evaluating the confusion matrix can be found in the module *metrics* from *sklearn*[3].

[3] https://scikit-learn.org/stable/modules/model_evaluation.html

C. *Logistic regression for digits* 3 *and* 8

1. In this part, you will set $C_0 = 3$ and $C_1 = 8$. Apply the same steps from the previous section to generate a dataset with only 2 features. Then redo steps C.1-6 on this new dataset.

   Are the classification results as good as for the dataset in B? Why?

2. Draw in the same picture the scatter plot of the dataset, the separating line from the previous exercise and the separating line obtained with threshold $p = 0.2$.

   Generate the confusion matrix for the classifier obtained with $p = 0.2$ and comment on the results.

3. Redo the previous item for $p = 0.8$.

D. *Logistic regression for digits* 8 *and* 9

1. In this part, you will set $C_0 = 8$ and $C_1 = 9$. Apply the same steps from section C.

   Are the classification results as good as for the datasets in B and C? Why?

2. Test nonlinear feature transformations using multivariate monomials of different degree ($d = 2$ or $d = 3$) of the input features. Transformation of features can be found in the module *preprocessing* from *sklearn*[4].

[4] https://scikit-learn.org/stable/modules/classes.html#module-sklearn.preprocessing