CMPE 462 PROJECT 1

Eray Kurtulus, Sadi Uysal, Nevzat Ersoy

April 2020

1 Feature Extraction for Representation 2

We prepared 4 features;

- Length of longest horizontal line,
- Length of longest vertical line,
- Maximum number of color changes in a column,
- Number of nonwhite pixels.

First two features are very similar to each other, second one does the exact same thing as the first one, only not to rows but to columns. These features record the length of the longest black line inside the image. We count the number of consecutive black pixels in a row, then we look at all the rows while doing the same thing and remembering the maximum number of consecutive black pixels. Of course here what we consider as black is up to us and we chose our threshold as 0.8.
Third feature looks at columns of the image. While looking at a single column, we count how many times the color is changed. Again, our choice of threshold for color change is 0.8. Then we look at every column of the image and feature is the maximum number of changes.
Last feature is fairly simple, simpler than the average, just counting number of nonwhite pixels but this time we considered only the value -1 as black.

2 Logistic Regression

To implement the gradient of the logistic loss with respect to ω, we shall first derive its expression by hand.

$$P(Y|X) = \sigma(Y \cdot W^T \cdot X) \quad where \;\; \sigma(s) = \frac{1}{1 + \exp(-s)}$$

with independence and identical distribution assumptions,

$$P(Y_1, Y_2, ...Y_N|X_1, X_2, ...X_N) = \prod_{n=1}^{N} P(Y_n|X_n)$$

We need to maximize the equation above.

$$max \quad \prod_{n=1}^{N} P(Y_n|X_n)$$

$$\Leftrightarrow \quad max \quad \ln \prod_{n=1}^{N} P(Y_n|X_n) \quad \equiv \quad max \quad \sum_{n=1}^{N} \ln P(Y_n|X_n)$$

$$\Leftrightarrow \quad min \quad \frac{-1}{N} \sum_{n=1}^{N} \ln P(Y_n|X_n) \quad \equiv \quad min \quad \frac{1}{N} \sum_{n=1}^{N} \ln \frac{1}{P(Y_n|X_n)}$$

$$\equiv \quad min \quad \frac{1}{N} \sum_{n=1}^{N} \ln (1 + e^{(-Y_n \omega^T X_n)})$$

Then we take the derivative with respect to ω and arrive at:

$$\frac{-1}{N} \sum_{n=1}^{N} \frac{(Y_n X_n) e^{(-Y_n \omega^T X_n)}}{1 + e^{(-Y_n \omega^T X_n)}}$$

With the regularization parameter, the equation becomes:

$$\frac{-1}{N} \sum_{n=1}^{N} \frac{(Y_n X_n) e^{(-Y_n \omega^T X_n)}}{1 + e^{(-Y_n \omega^T X_n)}} + \omega\lambda$$

This has no closed form solution, we will use the gradient descent algorithm to find iteratively.

3 5-fold Cross Validation

We have only seen minor improvement by using the regularization parameter. As we have marked clearly on the table below; the best configuration we could achieve was with representation 2, with 0.0001 learning rate (we had to lower the learning rate because the program overflowed with higher learning rates) and 0.15 as the regularization parameter.

Repr.	Learning rate	Reg. parameter	Mean Error Count	Standard Deviation
1	0.001	0.1	39	1.6733
		0.15	39	1.6733
		0.2	38.6	1.3565
2	0.0001	0.1	14.4	3.0067
		0.15	14.4	2.4166
		0.2	15.4	3.2

4 Evaluation

For representation 1, the training classification accuracy is:

$$\frac{1524}{1561} \times 100 = 97,63$$

The test classification accuracy is:

$$\frac{404}{424} \times 100 = 95,28$$

For representation 2, the training classification accuracy is:

$$\frac{1548}{1561} \times 100 = 99,17$$

The test classification accuracy is:

$$\frac{412}{424} \times 100 = 97,17$$

The regularization parameter have slightly improved the generalization performance.

Our feature set (representation 2) gave significantly better results. It is more discriminative as it has the same features as representation 1 and additional features.

To further improve the test accuracy, we can optimize hyper-parameters and find new discriminative features. We can test for a reduction in these features, if one of them is redundant we can remove it.