

# 183.605

## Machine Learning for Visual Computing

### Assignment 1

Michael Reiter

16. Oktober 2018

## 1 Task1

### 1.1 MNIST Data Set and Feature Selection

We applied a perceptron model to classification of handwritten digits. MNIST is a standard data set with  $28 \times 28$  pixel images of handwritten digits. We selected the digits 0 and 7 and 500 training images for each class for our classification task, as they are reasonably different. To choose the right feature we plotted all regionprops in a scatter plot matrix, see figure 1 and selected solidity and eccentricity. To compare the batch and online version we used an equivalent number of maximal iterations. This means we took the size of a batch times the maximum number of iterations in the batch case as the maximum number of iterations for the online algorithm. In our case we took 500000 and 500.

#### 1.1.1 Features without transform

We used once the batch training algorithm and once the online training algorithm to classify the data directly from these two features. The confusion matrix evaluated on a separate test set gives us the following performance:

For the online training algorithm evaluated on the test set we got 385 correctly classified results of 400 test images. We got the following confusion matrix:

n=400	classified as 1	classified as 7
Digit 0	185	15
Digit 7	0	200

For the batch training algorithm evaluated on the same test set we got 383 correctly classified results of 400 test images. We got the following confusion matrix:

n=400	classified as 1	classified as 7
Digit 0	187	13
Digit 7	4	200

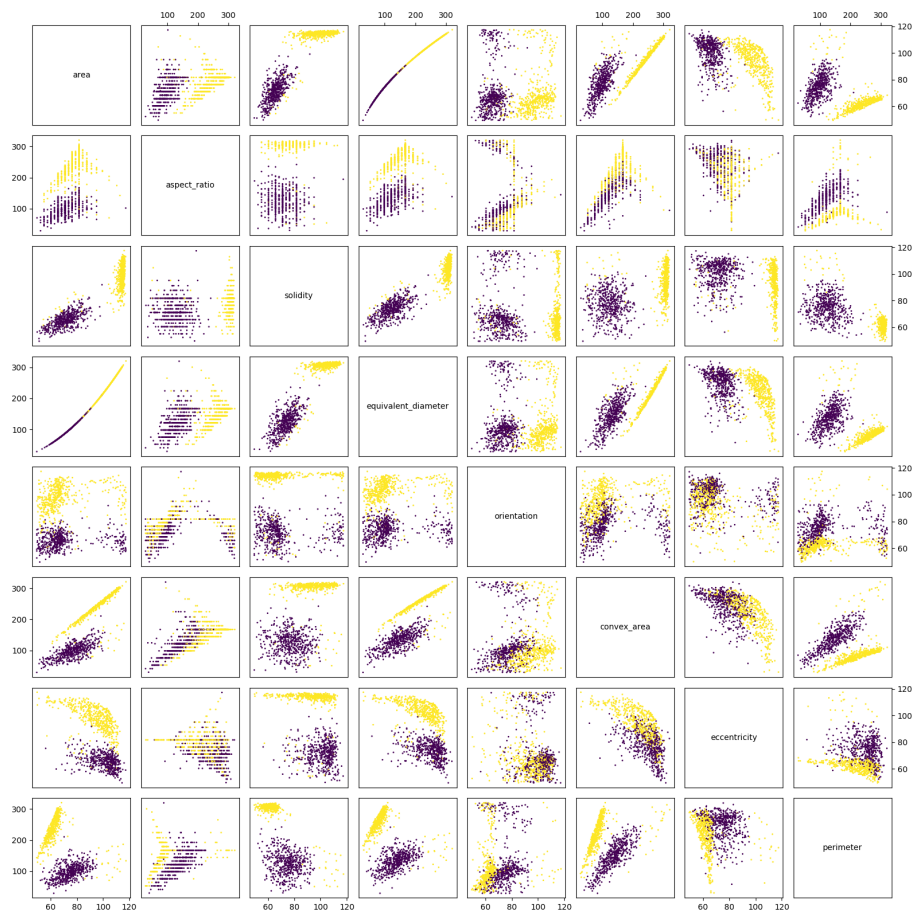


Figure 1: Scatter plot matrix of all region properties

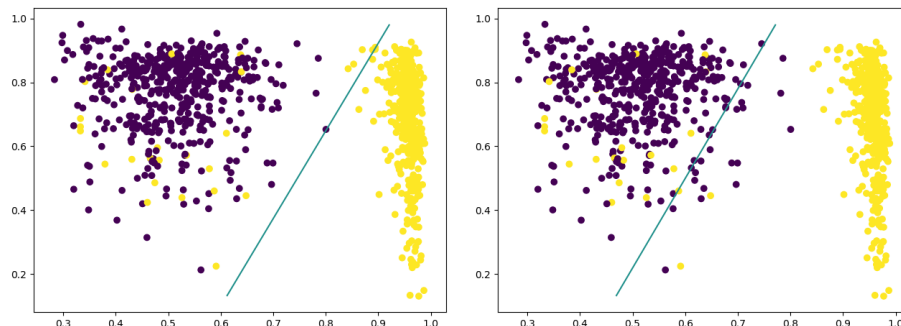


Figure 2: On the decision boundary of the trained perceptron. On the left for the online algorithm on the right for the batch algorithm.

The corresponding decision boundaries are shown in figure 3. We see that the data is not linearly separable, because there are yellow points inside the purple "cloud". In theory a perceptron is able to detect a linearly separable set, because if the set is linearly separable, then the algorithm is guaranteed to converge. Moreover we can bound the number of training steps from above. But as this bound depends on a separating hyperplane and the training set, we can not know beforehand how big this upper bound is.

### 1.1.2 Classification with feature transform

We performed the same experiments as above, but first applied following feature transform to our data.  $(x, y) \mapsto (1, x, y, x^2, y^2, xy)$

For the online training algorithm evaluated on the test set we got 385 correctly classified results of 400 test images. We got the following confusion matrix:

n=400	classified as 1	classified as 7
Digit 0	186	14
Digit 7	0	200

For the batch training algorithm evaluated on the same test set we got 383 correctly classified results of 400 test images. We got the following confusion matrix:

n=400	classified as 1	classified as 7
Digit 0	187	13
Digit 7	4	200

The corresponding decision boundaries are shown in figure 3. It's not very surprising that the feature transform doesn't help the classification. as there are yellow points, corresponding to the digit 0, inbetween purple points. Such a simple feature transform cannot separate this points, as the new decision boundary is just a quadratic.

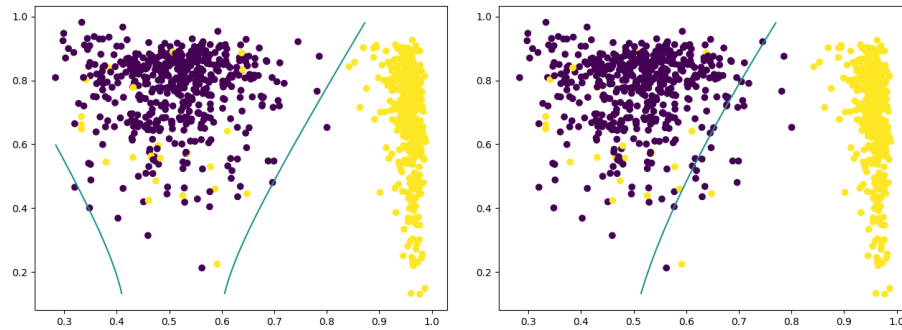


Figure 3: On the decision boundary of the trained perceptron. On the left for the online algorithm on the right for the batch algorithm.

### 1.1.3 Classification directly with image data

## 2 Task2

Here comes docu for part 2