

# Chapter - 3

## Linear Models

Aviral Janveja

### 1 Linear Classification

Let us consider a real world data-set of hand-written digits collected from India-Post offices. It is always better to test your models on real data, to get a better understanding of how your system would actually perform in reality. Here is a sample from the data-set :

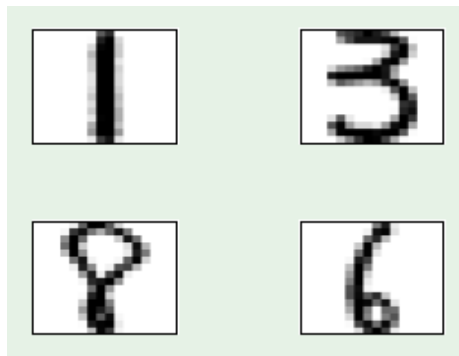


Figure 1: Real Data Example

As shown above, we have a bunch of hand-written digits, collected from postal stamps. We would like to design a model, that can decipher the digits from the given images. People can sometimes write digits in weird ways, making it difficult to understand even for a human operator. Indeed, the error rate for human operators is found to be around 2.5% and we would like to see if our machine learning model can at least equal that or maybe do better.

#### 1.1 Input Representation

Let us begin by looking at the given input data more closely. We are given a set of grayscale images containing hand-written digits, as shown below :

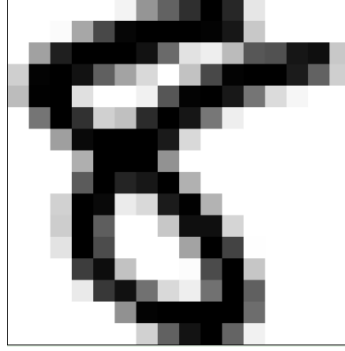


Figure 2: Hand-Written Digit Example

Now, each of these images is  $16 \times 16$  pixels. Meaning, each digit is represented by 256 real-number attributes. The raw-input  $\mathbf{x}$ , would therefore look like :

$$\mathbf{x} = (x_1, x_2, x_3 \dots x_{256})$$

That is a very long input to represent such a simple object. If we take this raw-input and try the Perceptron directly on it, we get too many parameters :

$$\mathbf{w} = (w_1, w_2, w_3 \dots w_{256})$$

The idea of input representation is to simplify the algorithm's life. We know that it is not about the individual pixel values, when trying to recognize a digit. We can instead extract some relevant features from the raw-input and then give those to the learning model and let it figure out the pattern.

## 1.2 Feature Engineering

Features are basically useful information that can be extracted from the given raw-input. For example, average pixel intensity, symmetry-score and curve-score. The digit 1 for instance, will score higher on the symmetry measure compared to a 5, whereas 5 will score higher on the intensity-score, similarly 8 will get a higher curve-score than 1 and so on. Using just these three features, significantly simplifies our input representation :

$$\mathbf{x} = (x_1, x_2, x_3)$$

Admittedly, we have lost information in this process. But chances are, most of that is irrelevant information anyway. From a generalization point of view as well, going from 256 to 3 parameters is a pretty good situation. Plotting a scatter diagram for digits 1 and 5 alongside just two features - symmetry and intensity. We get the following illustration :

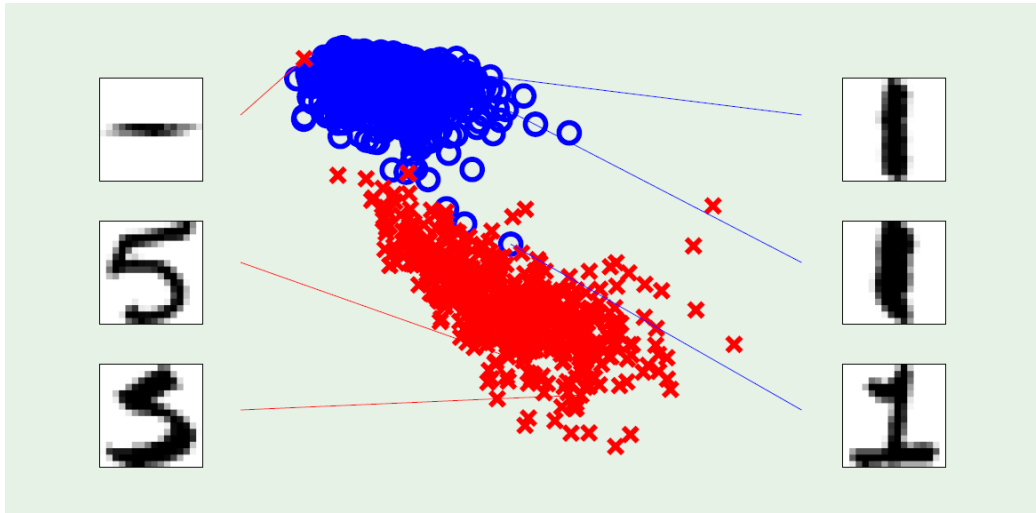


Figure 3: Illustration of Features

The blue points represent 1 and the red points represent 5. The horizontal-axis represents intensity and the vertical-axis represents symmetry. Indeed the red-fives are tilted more towards the right on the horizontal-axis, corresponding to their higher intensity-score. Meanwhile the blue-ones are higher on the vertical-axis owing to their higher symmetry-score. Just by using these two features, we see that the above data is already almost classified.

### 1.3 Pocket Perceptron

## 2 Linear Regression

## 3 References

1. CalTech Machine Learning Course - CS156, Lecture 2.