

# Naive Bayes Classification

Pattern Recognition and Machine Learning - MuMeT 2017

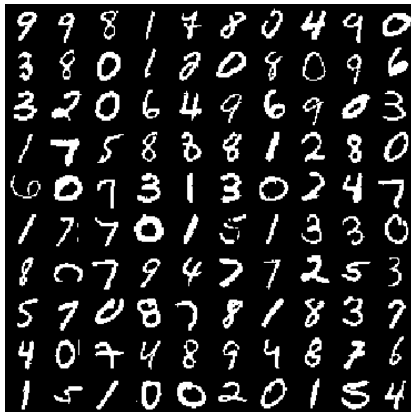
---

Davide Abati

October 12th, 2017

University of Modena and Reggio Emilia

Today we meet MNIST for the first time:



- 70000 images of handwritten images;
- available grayscale, we will use a binarized version;
- the task is to classify each image into the right digit.

We are given a training set  $\{X_i, Y_i\}_{i=1}^n$ , with  $X_i \in \mathbb{R}^m$  and  $Y_i \in \mathbb{R}$  for each  $i = 1, \dots, n$ .

- $n$  is the number of training images;
- each image  $X_i = \{x_i^{(1)}, \dots, x_i^{(m)}\}$  is a vector of  $m$  pixels;
- each label  $Y_i$  is just a number among  $\{1, \dots, d\}$ .

We will use the Bayes rule to build a classifier. The classification rule will be the following:

$$\arg \max_Y P(Y/X) = \frac{P(X/Y)P(Y)}{P(X)}$$

We will use the Bayes rule to build a classifier. The classification rule will be the following:

$$\arg \max_Y \boxed{P(Y/X)} = \frac{P(X/Y)P(Y)}{P(X)}$$

- $P(Y/X)$  is the probability of the image  $X$  of being labeled as  $Y$ ;

We will use the Bayes rule to build a classifier. The classification rule will be the following:

$$\arg \max_Y P(Y/X) = \frac{P(X/Y) \boxed{P(Y)}}{P(X)}$$

- $P(Y/X)$  is the probability of the image  $X$  of being labeled as  $Y$ ;
- $P(Y)$  is the prior probability of the class  $Y$ ;

We will use the Bayes rule to build a classifier. The classification rule will be the following:

$$\arg \max_Y P(Y/X) = \frac{P(X/Y) P(Y)}{P(X)}$$

- $P(Y/X)$  is the probability of the image  $X$  of being labeled as  $Y$ ;
- $P(Y)$  is the prior probability of the class  $Y$ ;
- $P(X/Y)$  is the likelihood of the image  $X$  under the model  $Y$ ;

We will use the Bayes rule to build a classifier. The classification rule will be the following:

$$\arg \max_Y P(Y/X) = \frac{P(X/Y)P(Y)}{\boxed{P(X)}}$$

- $P(Y/X)$  is the probability of the image  $X$  of being labeled as  $Y$ ;
- $P(Y)$  is the prior probability of the class  $Y$ ;
- $P(X/Y)$  is the likelihood of the image  $X$  under the model  $Y$ ;
- $P(X)$  is the probability of the image  $X$  under the whole dataset. Does not influence the argmax so we can drop it.



Finding the prior of a class  $Y$  is easy.

Simply count the number of examples of each class and divide by the number of total examples.

$$P(Y = c) = \frac{\sum_{i=1}^n \mathbf{1}\{Y_i == c\}}{n}$$

**Naive assumption:** all pixels are independent given the class. The probability of an image is the product of the probability of every single pixel.

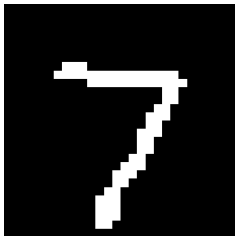
$$P(X_i/Y_c) = \prod_{j=1}^m P(x_i^{(j)}/Y_c)$$

During training, we need to model this for each possible class:



During testing, the likelihood of an image under a class is built by taking the class model (the one built before, during training) and:

- for each active pixel, account for the probability of being active given class;
- for each zero pixel, account for one minus the probability of being active given class;



Once we can model  $P(Y/X)$  for each class, we can classify an image by choosing the class that maximizes it:

$$\tilde{Y} = \arg \max_Y P(Y/X) = P(X/Y)P(Y)$$

All those products of probabilities can result in nothing. Use log-probabilities instead!

$$\log P(X_i/Y_c) = \sum_{j=1}^m \log P(x_i^{(j)}/Y_c)$$

$$\log P(Y/X) = \log P(X/Y) + \log P(Y)$$

---

**Algorithm 1** pseudocode for training

---

```
1: for  $j = 1$  to  $d$  do  
2:   compute the class prior  $P(Y_j)$   
3:   compute a model for  $P(X/Y_j)$   
4: end for
```

---

---

**Algorithm 2** pseudocode for inference

---

```
1: for  $i = 1$  to  $n$  do  
2:   for  $j = 1$  to  $d$  do  
3:     compute  $\log P(X_i, Y_j)$   
4:     compute  $\log P(Y_j/X_i) =$   
        $\log P(X_i, Y_j) + \log P(Y_j)$   
5:   end for  
6:    $Y_i = \arg \max_{Y_j} \log P(Y_j/X_i)$   
7: end for
```

---