Naive Bayes Classification

Machine Learning and Deep Learning

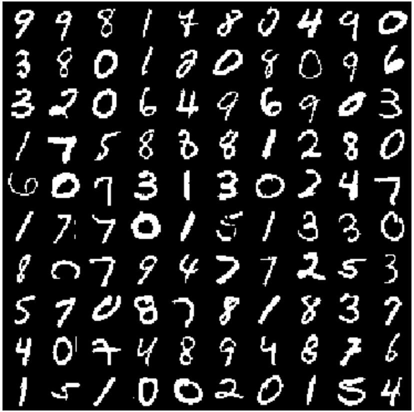
Aniello Panariello, Matteo Boschini, Lorenzo Bonicelli, Angelo Porrello, Simone Calderara

October 04, 2022

University of Modena and Reggio Emilia

MNIST digits classification

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.

Supervised learning setting

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 $i = 1, \ldots, n$.

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

Classification using bayes rule

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

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

Classification using bayes rule

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;

Classification using bayes rule

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;

Classification using bayes rule

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

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

- $P(Y/X)$ 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 likelyhood of the image X under the model Y;

Classification using bayes rule

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 likelyhood 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.

Prior $P(Y)$

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.

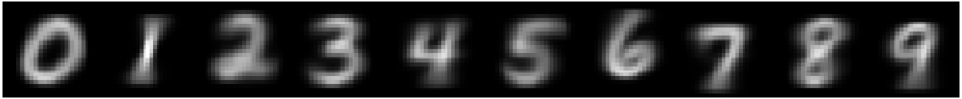
$$P(Y = c) = \frac{\sum_{i=1}^{n} \mathbf{1}\{Y_i == c\}}{n}$$

Likelyhood $P(X/Y)$

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

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

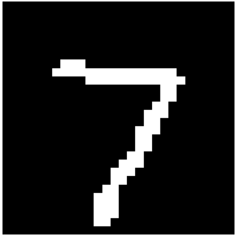
During training, we need to model this for each possible class:



Likelyhood $P(X/Y)$

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;



Inference

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

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

Use logarithms!

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

for $j = 1$ to d **do**
  compute the class prior $P(Y_j)$
  compute a model for $P(X/Y_j)$
end for

Algorithm 2 pseudocode for inference

for $i = 1$ to n **do**
  for $j = 1$ to d **do**
    compute $\log P(X_i, Y_j)$
    compute $\log P(Y_j/X_i) = \log P(X_i, Y_j) + \log P(Y_j)$
  end for
  $Y_i = \arg\max_{Y_j} \log P(Y_j/X_i)$
end for