Report on generating handwriting using KDE

- Hoang Ba Cong

Introduction

In this report, i will describe and explain the formulas i use for generating new data. There are 3 parts consisted in the report.

Presenting the problem

Initially, i will read the data which are vectors of 784 numbers corresponding the brightness of pixels in each image and the labels showing the actual numb3er of image. Then, the way to generate new data would be finding x satisfying the value of distribution p(y|x) ,y is labels from 0 to 9 , bigger than the threshold.

KDE model

In this case, i obtain a smooth density model and the common choice is gaussian, which gives rise to the following Kernel density
model:

$$p(\mathbf{x}) = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{(2\pi h^2)^{1/2}}\exp\left\{-\frac{\|\mathbf{x}-\mathbf{x}_n\|^2}{2h^2}\right\}$$

h is bandwidth, x is data point.

The question is how to calculate p(y|x) with the given x and y. The point here is that we have the equality p(y|x) = p(x,y)/p(x) which involves to build 2 KDE model to separately calculate p(x,y) and p(x). With p(x,y), my idea is to form a vector of 784 + 10 consisting the image and its labels which have 9 zero numbers and the another i position number is x (x should be big enough to emphasize the specific features of each point). Therefore, we could easily implement p(x) either. Finally, the last step

is to generate and choose random points which have p(y|x) >= threshold (the given threshold should be big enough to guaranty the good result.

24

Estimating bandwidth

25

We have to efficiently identify 2 optimized bandwidth for each model,the accurate algorithm we could use is K-cross-validation. Hence, we would call this algorithm like binary search by choosing max and min and find the min <= x <= max satisfying x is the best bandwidth.

Result

After implementing KDE without using libraries in python, my program get the great result. In addition, i figure out that not only generating but we can identify a new number using KDE with high accuracy. And one crucial factor affecting the accuracy is the value of labels y. As we normally observe that this value belongs to {0,1} but it doesn't affect much on the distribution. Therefore, in my code, i choose x = 1000000 which i find it most optimized.