Report on generating handwriting using KDE

Hoang Ba Cong

February 2020

1 Introduction

In this report, i will describe and explain the formulas i use for generating new data. There are 6 parts consisted in the report.

2 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 number 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.

3 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:

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

The problem is how to calculate $p(y|x)$ with the given x and y. The key in solving this problem lies in the formula $p(y|x) = \frac{p(x,y)}{p(x)}$ which involves building 2 KDE models to separately calculate the joint probability $p(x, y)$ and $p(x)$. Given this problem formulation, My idea of estimating the optimal bandwidth is:

- Form a vector $\mathbf{z} \in \Re^{784+10}$: The concatenation of $\mathbf{x} \in \Re^{784}$of the flattened image and $\mathbf{y} \in \Re^{10}$ of its label in the one-hot form.
- Then apply Kernel Density Estimation for estimating $p(z)$.
- $p(x)$ can also be estimated using the same technique, for the optimal bandwidth of this part, I use the GridSearch function of the SkLearn.
- Given $p(z)$ and $p(x)$, $p(y|x)$ is naturally derived as above.
- For data generation, random sampling of x such that $p(y|x) > T$ for $T \in \Re^+$ is a chosen threshold.

The choosing of T has to be experimented to guarantee decent results.

4 Estimating bandwidth

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 the condition that x is the best bandwidth. In order to reduce the runtime of implementation, the bandwidths would be already estimated using GridSearch function of the Sklearn. In this case, 2 optimal bandwidths to estimate $p(x)$ and $p(x|y)$ are 35.2970730273065 and 37.64935806792467.

5 Threshold prediction experiment

Apparently, if the chosen threshold is too small, the generated data will not be accurate as we have $p(y|x) >= threshold$. Thus, we have to predict appropriate threshold to increase the accuracy of new generated data. The figure below show how inaccurate it was when choosing small threshold.

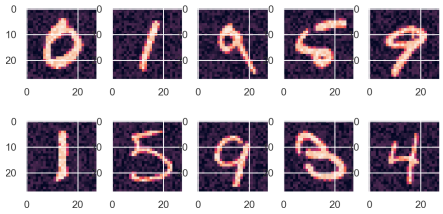

Figure 1: Generated numbers from 0 to 9 with threshold = 0,001

After increasing threshold to 0.1, we get acceptable result showed in the figure below:

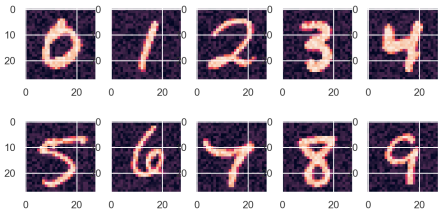

Figure 2: Generated numbers from 0 to 9 with threshold = 0,1

In conclusion, after several experiments, enhancing the accuracy could affect the runtime since generating more data is needed for estimating the data more exactly due to high threshold.

6 Result

After implementing KDE using pure python, my bandwidth estimation is able to meet expectation from the challenge. With $numpy, scipy, sklearn$ libraries in python, the generator achieved decent result. In addition, I figured out that not only generating but we can identify a new number using KDE with high accuracy, although due to time constraint, quantitative experiments has to be spared. One of the emphasized observation from my side is that the value of one-hot form labels $\mathbf{y}$ is crucial for the estimation. This value normally belongs to 0,1 but it doesn't affect much on the distribution. Therefore, in my code, I choose1000000 which I find it most optimized.