Intro to ML - HW1 - Code discussion

Daniel Volkov, I.D: 330667494

June 11, 2024

Question 1

From the Hoeffding bound, we get that $\mathbb{P}(|\bar{X}_i - \frac{1}{2}| > \varepsilon) \leq 2 \cdot e^{-2n\varepsilon^2}$.
We can see that this bound is pretty loose in this case, although it's the best we can get from the Hoeffding bound here.

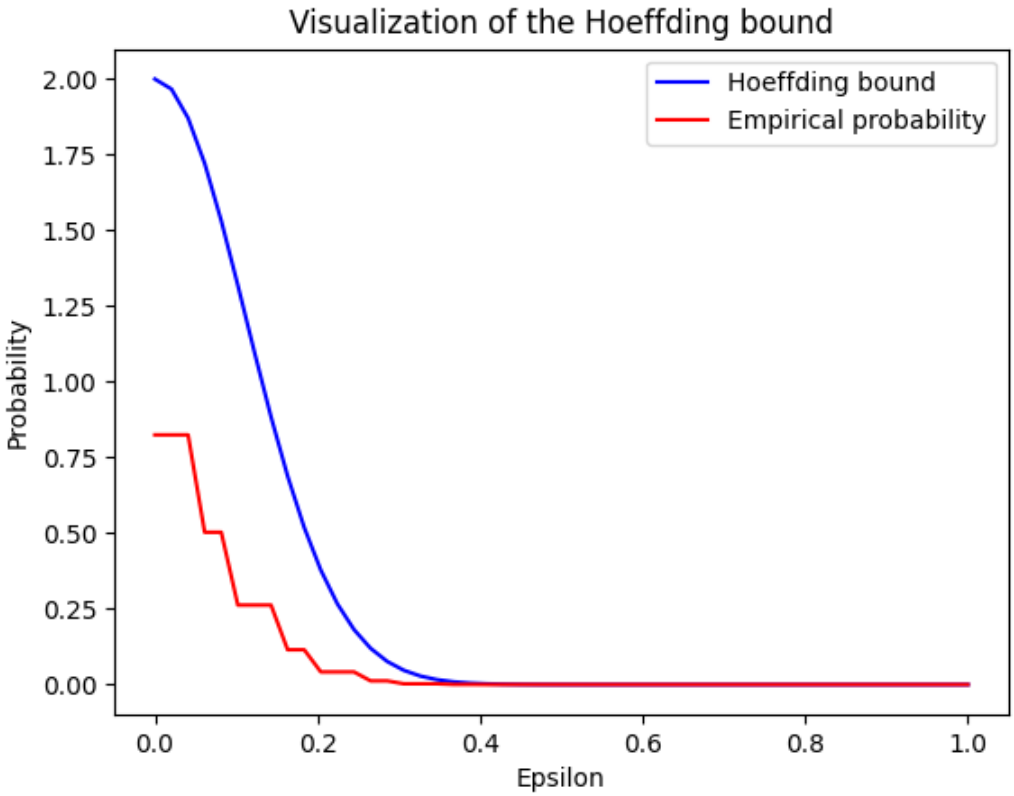

Figure 1: Empirical $\mathbb{P}(|\bar{X}_i - \frac{1}{2}| > \varepsilon)$ as a function of ε

Question 2

(a)

Code submitted, appears as the *classify(images, labels, query, k)* function.

(b)

Running the algorithm on $n = 1000$ training images with $k = 10$, results in an accuracy of 0.846 .
From a completely random predictor, it would be reasonable to expect random classification, which would have resulted in an accuracy of $\frac{1}{10}$, and generally: $\frac{1}{\#\text{labels}}$.

(c)

As we can see, the accuracy **decreases** as a function k. This may suggest that the "closest" sample is mostly (something like 90% empirical probabily) in the same class as the current sample.
This implies that the "best" k (the k value that gives best accuracy), would be $k = 1$.
The conclusion may be somewhat intuitive: making k larger decreases the algorithm's "resolution", meaning that the algorithm has some "useless" (maybe harmful) biases when examining the current sample.
In the extreme, if we make $k = n$, the predictor will make the same prediction for all samples, resulting in very low accuracy.

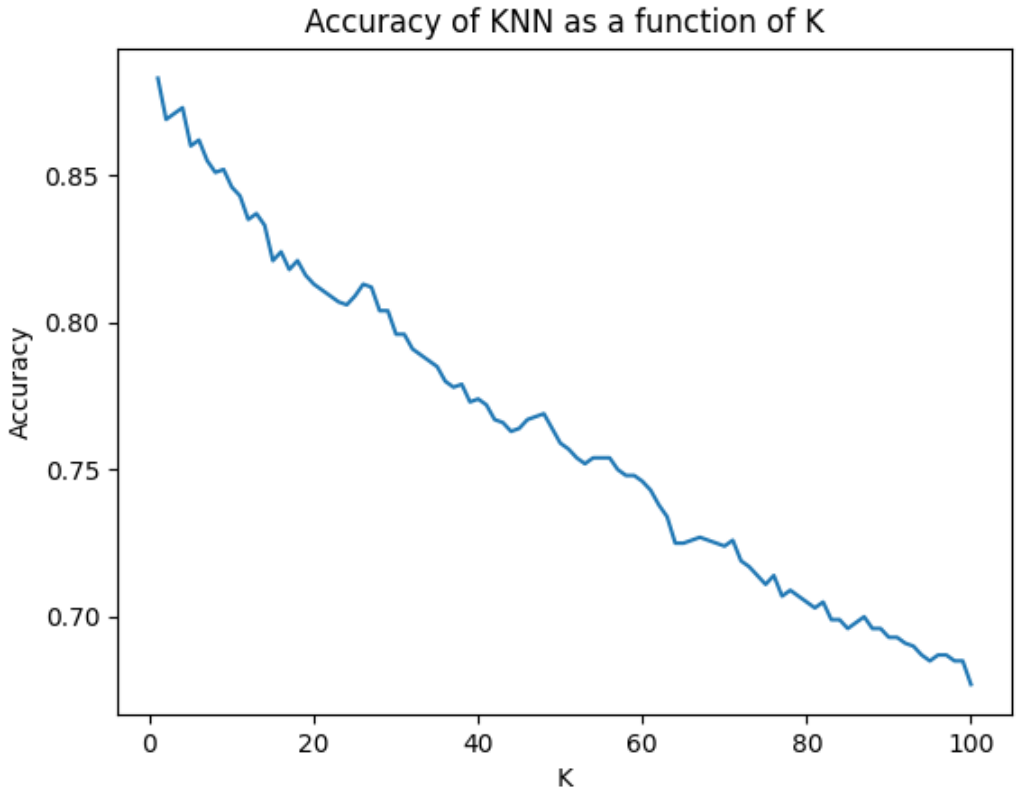

Figure 2: Accuracy of KNN as a function of K

(d)

As we can see, the accuracy generally **incraeses** as a function of training set size.
We can also see that the graph that we get resembles a "logarithmic" shape - meaning it grows rapidly at first, and the rate of change slows down at some point.
This behaviour is a result of the algorithm of course: When n is too small, the algorithm "doesn't know enough" about the data, and thus has low accuracy. As the number of training samples grows, the algorithm has more data-points for comparison, and thus (probably) has closer datapoints to the sample it is trying to classify.

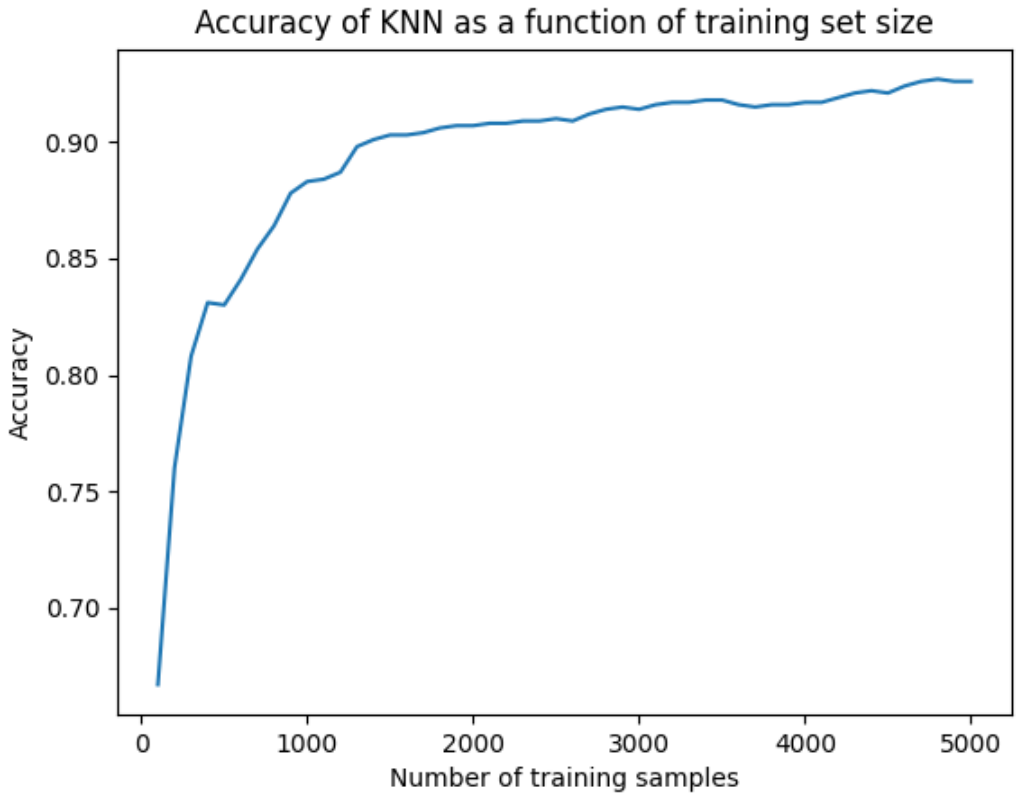

Figure 3: Accuracy of KNN as a function of n