**Machine Learning Fundamentals by Sanjoy Dasgupta UC San Diego on edX platform**

**DSE 220 X - UCSD**

Maths covered :

Probability ---> Linear Algebra ---> Optimization

**Day 1: 18 November**

* Classification problem
* Training set and test set
* Representing data as vectors
* Distance in Euclidean space
* 1-NN classifier
* Training error versus test error
* The error of a random classifier



Above is a sample dataset of MNIST(National Institute of Standards and Technology ). **Problem:** identify the handwritten digit.

**Approach 1:** Identify the loops and straight-line every digit has, and write a piece of software to identify the loops and lines to identify which digit it is.

**Inference1:** It did not work due to the noise and variability in different handwriting. A massive set of rules were declared for various styles in vain.

**Approach 2:** Machine Learning Approach. Let the machine figure out these patterns with a whole chunk of data. MNIST has a **Training set** **of 60,000 images** of handwritten digits and their labels and a separate **Test set of 10,000 images** with their labels.

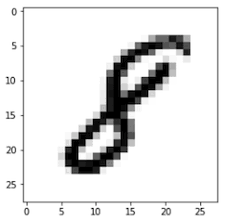
It can be run in **Classifier,** a function that takes an image as input and gives the output as its label.

The method of classification we will be using today is the Nearest neighbor classification.

In order to classify a new image x, we will look for an image that is closest to that particular image, and then we will output the label of that particular neighbor.

For the implementation of this program, we require a notion of distance or closeness among different images. An MNIST image is 28 pixels wide and 28 pixels high. (total of 784 pixels)

Each pixel has a greyscale ( [black] 0 - 255 [white] ).



In order to convert such an image into a vector, we stretch each image into a vector with 784 coordinates. Copy the 28 pixels of the first row, then the next 28 pixels of the second row, and then to the last row is stretched into a vector with 784 coordinates.

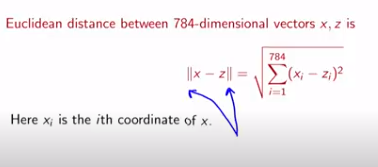
A new data space is created as 784-dimensional Euclidean Space as

X = R ^784

Label space = { 0,1,2,3...7,8,9 }

How are we going to calculate distance between these 784 dimensional vectors?

The most common Euclidean distance in 2 dimensions (x,y) is simply the line connecting the two points. For every dimension, we square the distance between each individual coordinate.



To classify a new image X, we are just going to find its nearest neighbor amongst the x(i) using Euclidean distance in R^784, and return its label as y(i).

How good is this classifier?

Error rate on **training datasets** is completely zero 0 as it is its own nearest neighbor.

In general, **training error** is an overly optimistic predictor of the future performance.

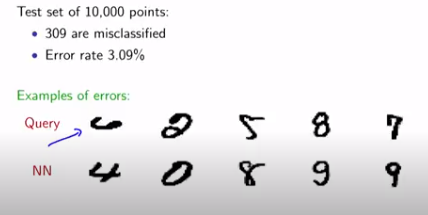
But there will be slight errors when gauged on **Test datasets.**

**Test error** is the fraction of test points incorrectly classified.

What test error would we expect for a random classifier?

It is the one that picks a label of 0-9 at random. Probability of error rate will be 90% as only 1 of the labels among 0-9 will be incorrect.

Test error of the nearest neighbor classifier 3.09% for judging incorrectly 309 points among 10,000 test points.



NN is the judgement of the Nearest Neighbor Classifier among the given Query dataset.

**DAY 2: 19 November**

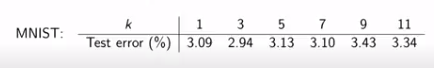
Ideas for improvement :

1. K-NN

To classify a new point, instead of looking for the 1 perfect point

Find the k nearest neighbors of 3-5 points in the training set.

Return the most common label amongst them.



When k = 1, error rate is 3.03 %

But when k = 3, it decreased to 2.94 % and further it keeps on increasing.

The changes were not dramatically different in case of error rate.

Also in practical cases, there is no specific test set.

In ML, many of the parameters such as K needs to be set correctly using the training set alone in order to make the method work correctly.

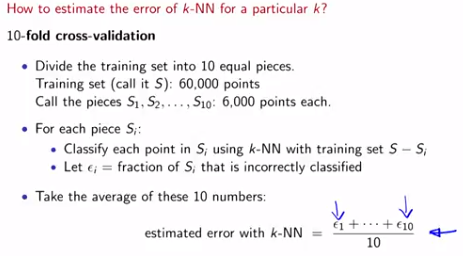
Suppose we want to evaluate a particular choice of k for checking its error rate with only 3 nearest neighbour in only the training set.

We take the 60,000 points in the training set, and divide it into 10 equal parts as S1, S2,... S10 with 6,000 points each.

We will consider 1 chunk as test set, and remaining 9 chunks as the training set.

Each 6,000 points of the points in the S1 set is classified with remaining 54,000 points.

The error rate inferred will be as e1, e2... ,e9,e10..To get the better error estimate, we will do the average.



Similarly 5-fold cross validation can also be implemented.

1. Better distance functions



Vector distance is slightly different in above 2 pics due to slight rotation at the corner, leading to wide gap in the Euclidean distance. Another distance function is the Tangent distance function.

A broader family of natural deformations is the shape context function that consider wider distortion rather than just a slight rotation.



In general, using domain knowledge of using particular function to improve the performance of classifier is necessary. With Shape context, the error rate in using K-nearest neighbour drops below 1%, and the perform of the classifier is improved dramatically.