Lecture 1 : Introduction and Nearest Neighbour Algorithm

1 Introduction

Artificial Intelligence (AI) is a broad term used for design of intelligent systems. *Machine learning* (ML) is considered a subfield of it focusing on algorithms for specific subproblems. An AI system will be using multiple machine learning algorithms for solving a high level task like playing a game of chess. Machine learning algorithms can be *data driven* and non data driven (or *logic driven*). Most of the algorithms like sorting, shortest paths etc. that you come across in computer science are non data driven (or logic driven) methods. In such problems, the output for every input is well defined. For example, in case of sorting a list, the output requirements are clear. Either a list is sorted or not sorted.

Data driven methods. For a lot of common problems the distinction between correct outputs and incorrect outputs are blurry. For example, consider the problem of finding a digit from $0, \cdots, 9$, given a handwritten image of the digit. There can be some inputs for which, it is not clear which label is the correct one. However, if we knew the handwriting style of the person, we might be able to get a clearer idea. So if we had a large corpus of hand written digit images for which the correct digit is given, we can get insights in to the handwriting of people, and more accurately find the correct digit in an unseen data point. This type of algorithms are known as data driven algorithms. In a *data driven* method, a vast *dataset* is given and an algorithm needs to be designed that predicts outputs for new unseen data, based on the insights gained from a dataset.

Figure 1: Some examples from the MNIST dataset. In some of the images, there can be a confusion between 2, 3 or 1,7.

Data driven machine learning techniques are further divided into two extreme problems i.) Supervised, ii.) Unsupervised. In *supervised learning*, the dataset provided consists of input, output pairs. For every input in the dataset, the correct output is also provided. However in *unsupervised* learning, we are just given a collection of inputs to the problem without any information about the correct outputs. There are also problems that are intermediate between these two extreme types, where limited supervised dataset is available, or a large data set with partial information regarding the correct outputs are given. Some of the common type of intermediate problems are called *Semi supervised* and *Reinforcement learning* problems.

2 Nearest Neighbour Algorithms

Nearest Neighbour is a very basic machine learning algorithm, that can be used for a wide variety of problems. Lets consider a supervised learning problem where the dataset is of the following form:

$$(X_1, Y_1), (X_2, Y_2), \cdots, (X_n, Y_n) \text{ where each } X_i \in \mathbb{R}^d \text{ and } Y_i \in \{0, \cdots, 9\}.$$

That is the inputs are vectors with dimension d, and output is a number in $\{0, \cdots, 9\}$ (See the handout titled Linear Algebra Primer, to familiarize with some vector definitions and notations). We will see how to convert the input into vectors in a later lecture on *representations*. For example: X_i is a vector representation of the handwritten image in the MNIST dataset and Y_i the correct label. Given a new input X, the nearest neighbour algorithm searches through the dataset and computes the distance $d_i = \|X - X_i\|$ of X from each of the X_i. Then the algorithm predicts the label for X to be the label for the X_i which has the smallest distance d_i among $d_1, d_2, \cdots, d_n$.

A disadvantage of the plain nearest neighbour algorithm described above is that it is highly susceptible to noise in the dataset. That is, if for some reason, a few images in the dataset were given incorrect labels, the algorithm can easily fail. For example, suppose X_3 is a vector representation of the most common way of writing the digit 7 and it was mislabeled as 1. Whenever a new handwritten input of 7 is given, the closest neighbour will most likely be X_3 and the algorithm will predict 1 instead of 7. When large datasets are made, inevitably a few inputs also are mislabeled. Hence this can be a serious problem in practice.

KNN Algorithm. The k-nearest neighbour (KNN) algorithm rectifies the issues regarding noise of the above algorithm, using a simple extension. Then KNN algorithm finds the distances d_i of X from each of the X_i's similar to the nearest neighbour algorithm. But then it finds indices $i_1, i_2, \cdots, i_k$ such that $d_{i_1}, d_{i_2}, \cdots, d_{i_k}$ are the bottom k numbers in the list $d_1, d_2, \cdots, d_n$. Note that $Y_{i_1}, Y_{i_2}, \cdots, Y_{i_k} \in \{0, \cdots, 9\}$ are the correct labels for these indices in the dataset. The KNN algorithm counts the number of times each label in $\{0, \cdots, 9\}$ appears in the list $Y_{i_1}, Y_{i_2}, \cdots, Y_{i_k}$, and then predicts the label Y for X (the new input) as the label with the majority count.

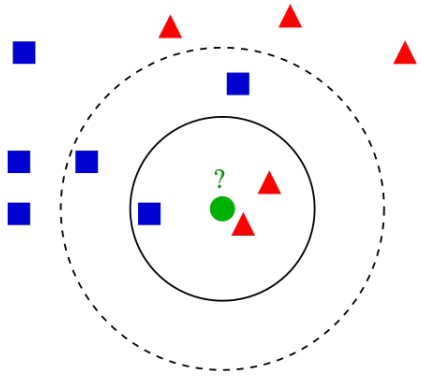

Figure 2: Given an input represented as the circle, the 3NN algorithm, will find the "three nearest neighbours", and choose the majority label among those three. In this case, the label predicted is triangle as there are 2 triangle and only 1 square among the 3 nearest neighbours.