EE2211 : Introduction to Machine Learning (Fall 2020)
Different Types of Machine Learning Problems

Vincent Y. F. Tan

In this document, we will summarize three main problems in machine learning, reinforcing some notions from the lecture.

Vectors (e.g., $\mathbf{x}$) will be denoted by boldface while scalars (e.g., y) will be denoted by san serif font. The set of all real numbers (resp. natural numbers) is denoted as $\mathbb{R}$ (resp. $\mathbb{N}$).

1 Classification

The most canonical task is classification. This problem takes the following form. We have a *dataset* $\mathcal{D}$ which consists of a certain number n of *data examples* $(\mathbf{x}_i, y_i), i = 1, \ldots, n$. Each data example $(\mathbf{x}_i, y_i)$ consists of a *feature vector* (also called *training samples*)

$$\mathbf{x}_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{id} \end{bmatrix} \quad \text{or} \quad \mathbf{x}_i = \begin{bmatrix} x_{i1} & x_{i2} & \ldots & x_{id} \end{bmatrix}^T$$

(usually an element of d-dimensional Euclidean space $\mathbb{R}^d$) and a label y_i. The crux of classification is that the label y_i can only take on *finitely many values* so $y_i \in \{1, 2, \ldots, K\}$ for some integer $K \geq 2$. We do not allow y_i to take on infinitely many values.

The classification problem consists in finding a *classifier* which is a function $f : \mathbb{R}^d \to \{1, 2 \ldots, K\}$ such that it accurately predicts labels given new samples, called *test samples*. This function f is constructed or learned based on the dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i) : 1 \leq i \leq n\}$.

A couple of examples will illustrate this point.

- We have records of $n = 100$ emails. Let $d = 2$. Thus, there are two components in each $\mathbf{x}_i$. Furthermore the first component of $\mathbf{x}_i$, denoted as x_{i1}, counts the number of times the word "love" appears. The second component of $\mathbf{x}_i$, denoted as x_{i2}, counts the number of times the word "money" appears. Each label $y_i \in \{0, 1\}$ where $y_i = 1$ means that the i^{th} email is spam while $y_i = 0$ means that the i^{th} email is non-spam. Based on the dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i) : 1 \leq i \leq 100\}$ of $n = 100$ emails, we are tasked to learn a classifier $f : \mathbb{R}^2 \to \{0, 1\}$ such that we can accurately predict whether the next email you receive (which is of course not in the training dataset) is spam or not. See Fig. 1.

- Consider an image classification problem. We are given a dataset of size $n = 10^6$, namely, $\mathcal{D} = \{(\mathbf{x}_i, y_i) : 1 \leq i \leq 10^6\}$ where each $\mathbf{x}_i$ represents an image that is vectorized into a vector. For example, each image contains $5 \times 5 = 25$ pixels and for the sake of simplicity, each pixel value is 0 or 1. Thus $\mathbf{x}_i \in \{0, 1\}^{5 \times 5} \cong \{0, 1\}^{25}$ and $d = 25$. Of course, we can have more complicated images that have multiple quantization levels (not restricted to binary) and colors as well, but the treatment will be the same. To each image, there is a label y_i which can take on 10 values so $K = 10$. These 10 values signify what object is in the image, e.g., $y_i \in \{\text{dog}, \text{cat}, \ldots, \text{snake}\}$. We would now like to design a image classifier $f : \{0, 1\}^{25} \to \{\text{dog}, \text{cat}, \ldots, \text{snake}\}$ that "does well" (in some sense) on new images (or test images) that contain one of the 10 animals.

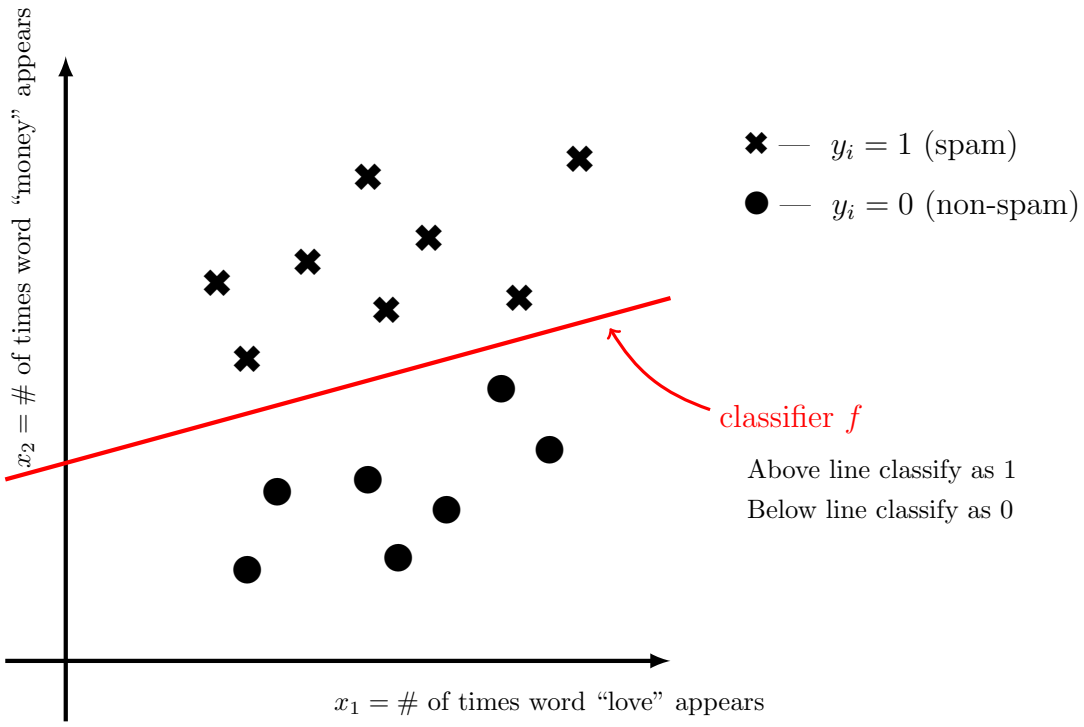

Figure 1: Spam classification

- Is the following a classification problem? There are a total of $M \geq 2$ football teams. Each football team $i \in \{1, 2, \ldots, M\}$ plays L games against every other football team j. Say there are no draws. The number of times i beats j is denoted as $b_{ij} \in \{0, 1, \ldots, L\}$ so necessarily $b_{ij} + b_{ji} = L$ (convince yourself that this is the case). We set $b_{ii} = 0$ for all i. Clearly, if one team wins all its games against other team, it is the best. When this is not case, which is common, it is not so clear what is the best strategy to *rank* the team. More precisely, given the matrix $B = [b_{ij}]_{1 \leq i \leq M, 1 \leq j \leq M}$ we would like to rank the teams from top (best) to bottom (worst). This is a non-standard machine learning problem. However, notice that the total number or rankings is $M!$, which is finite, so in some sense this is like a classification problem. In what way is it not a classification problem?

Classification belongs to the class of *supervised* learning methods.

2 Regression

Regression is just like classification except that y_i are no longer restricted to belong to a finite set. Rather it can take on uncountably[1] many values. In the case of regression, we typically do not say that y_i is the label. Rather we use the term *target variable* or *outcome variable* or *dependent variable* to refer to the y_i's. For instance y_i could take any value in the interval $[0, 1]$ (which is uncountable) or it could take values in $\mathbb{R}$. In either case, the data examples in the dataset $\mathcal{D}$, denoted as $(\mathbf{x}_i, y_i)$ belong to $\mathbb{R}^d \times \mathbb{R}$, which means that each feature vector or training sample $\mathbf{x}_i \in \mathbb{R}^d$ (d-dimensional Euclidean space) and each target variable $y_i \in \mathbb{R}$ is real.

Again some examples would make this clear.

[1] A set A is *countable* if there is a bijection between A and the set of natural numbers $\mathbb{N}$. A set A is *uncountable* if its cardinality is strictly larger than that of the natural numbers. Check that the set of rationals is countable. A well-known, but non-trivial, fact (due to Cantor) is that the interval $[0, 1]$ is uncountable.

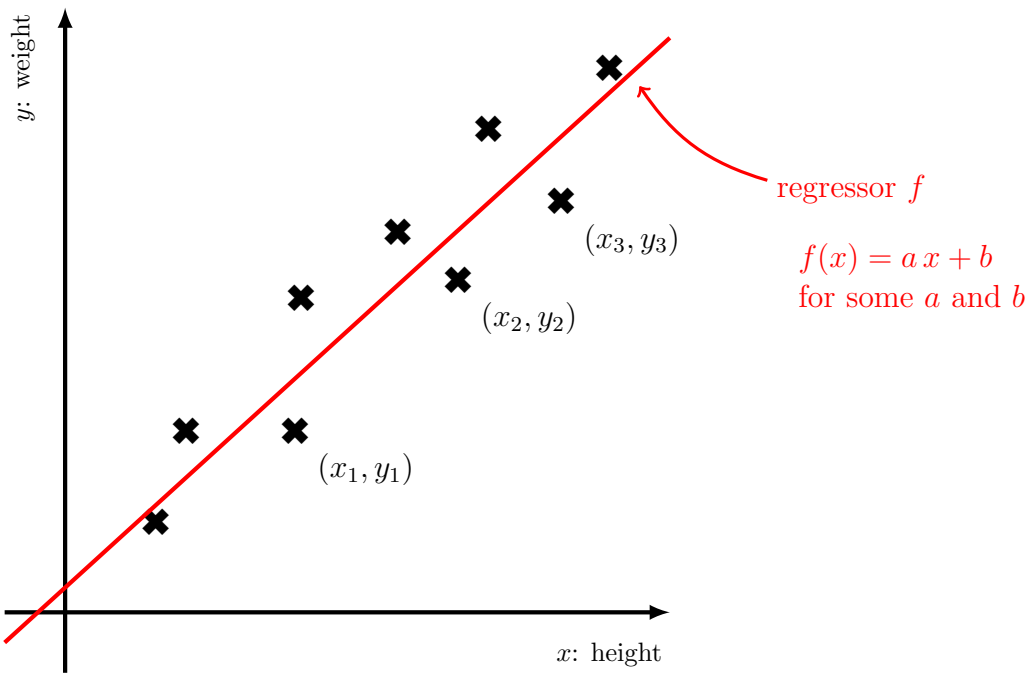

Figure 2: Height and weight regression example

- Say each feature vector $x_i \in \mathbb{R}$ is scalar (so $d = 1$) and represents the height of a student. The target variable y_i represents the weight of the same student. Given the data of $n = 600$ students in EE2211 captured in the dataset $\mathcal{D} = \{(x_i, y_i) : 1 \leq i \leq 600\}$, I am tasked to learn a regressor $f : \mathbb{R} \to \mathbb{R}$ so that I can use the height of Alice x' in a new class EE9999 to predict her weight $y' = f(x')$. See Fig. 2.
- Say each feature vector $\mathbf{x}_i \in \mathbb{R}^3$ is 3-dimensional and x_{i1} represents the air pressure at location i, x_{i2} represents the amount of rainfall at location i and x_{i3} represents the "amount of greenery" at location i. Each y_i corresponds to the temperature at location i. The temperature can take on uncountably many values – it is a real number. Given the dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i) : 1 \leq i \leq n\}$, we would like to learn a regressor $f : \mathbb{R}^3 \to \mathbb{R}$ such that if I give you the feature vector of new location $\mathbf{x}'$, you can tell me the temperature $y' = f(\mathbf{x}')$.

Regression also belongs to the class of *supervised* learning methods.

3 Clustering

In clustering, the labels or target variables y_i are no longer available and we *only* have access to the feature vectors $\mathcal{D} = \{\mathbf{x}_i : 1 \leq i \leq n\}$. There is often reason to believe that these feature vectors can be grouped or clustered into different clusters.

- Say each $\mathbf{x}_i = [x_{i1}, x_{i2}]^T$ represents the height and weight of students in EE2211. There is reason to believe that girls are shorter and lighter than boys. So given $\mathcal{D} = \{\mathbf{x}_i : 1 \leq i \leq n\}$, can we assign each partition this set into 2 groups automatically? See Fig. 3. Note that while the clustering algorithm knows that there are 2 clusters, it does not and cannot assign semantic meanings (such as "boys" and "girls") to the clusters it discovers. It can only tell us that $\mathcal{D}$ is partitioned into two non-empty disjoint subsets $\mathcal{D}_1$ and $\mathcal{D}_2$ (i.e., $\mathcal{D}_1 \cap \mathcal{D}_2 = \emptyset$ and $\mathcal{D}_1 \cup \mathcal{D}_2 = \mathcal{D}$).

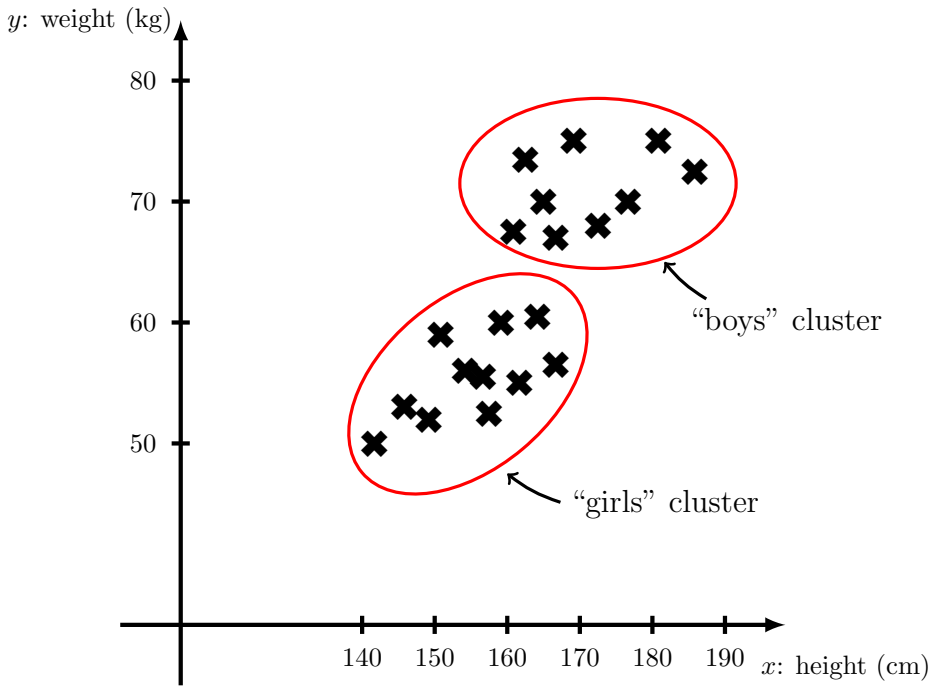

Figure 3: Height and weight clustering example

- Say each $\mathbf{x}_i = [x_{i1}, x_{i2}, x_{i3}, x_{i4}]^T$ and $x_{i1}, x_{i2}, x_{i3}, x_{i4}$ respectively represent the number of times the words "winner", "victory", "stock", "prices" appeared in a bag of n news articles. There is reason to believe that sports articles would contain more of the first 2 words but finance articles would contain more of the last 2 words. Hence, it seems possible to design a clustering algorithm to group the n feature vectors into two clusters, one representing sports articles while the other representing finance articles.

Clustering belongs to the class of *unsupervised* learning methods.