January 9, 2015

Abstract

The class focused on giving an overview of the types of machine learning problems. The instructor defined supervised and unsupervised problems and their subclasses.

1 The Goal

The Goal of the whole process is to learn some concept. In this setting, we have a supervisor, who *knows* the concept, and there is a *learner*, who wants to learn it. A running example used throughout the lecture was that of a baby trying to make sense of the world around it. In this case, the learner is the baby. Different kinds supervisors (e.g. Mother) and learning settings give rise to different types of machine learning problems.

1.1 What does it mean to learn a concept?

In abstract terms, it means learning a mapping from a given input to an output. For example, given 100 images of cats and dogs, with tags assigned to each of them, we want to *learn* what do cat and dog look like so that given a new image, we can assign a tag to it.

1.2 Training Data

Supervisor tries to teach the concept with the help of some examples. This set of examples is called the training data, D, in Machine learning parlance.

$D = (x_1, y_1), (x_2, y_2), ..., (x_m, y_m)$

Here,

- m is the input size,
- x is the input,
- y is the corresponding output.

As we'll see in the coming sections, y may not always be present.

Continuing the example above, i^{th} element of the training set will be (x_i, y_i), where x_i is the i^{th} image (in some representation, say pixel map) and y_i will be the label of the i^{th} image (either cat or dog).

1.3 Hypothesis Set

Various mappings can explain the training data to different degrees of accuracy. A set of all such mappings constitutes the Hypothesis set. Goal of the learner is to pick the one which explains the concept best.

2 Supervised Learning

2.1 Types: Based on type of input

The supervised learning is further classified into the following 2 broad categories based on the nature of y

- **Regression** $y \in \mathbb{R}$.
- **Classification** $y \in S$ where S is a discrete set. (E.g. {cat, dog} in the example above)

  Note that we are not interested in the interaction happening among the elements of S.

There are more fancy categories of problems, like structured learning, but we need to focus on these two only at the moment.

2.2 Active Learning

The standard setup is the one in which we are given **all** the input output examples before the training starts. There is an alternate mechanism wherein only some of the examples are given and you query the supervisor for y variables corresponding to the other X. The idea is to ask good questions to the supervisor so that you can manage to learn the concept fairly well by using just a few examples. This alternate paradigm is known as *Active Learning.*

3 Unsupervised Learning

In contrast with the supervised learning setting, where the training data is of the form (X, y), the training data in this approach consists of just X. As an example, consider the following problem:
Given a new sentence, figure out whether it is from your mother tongue or not.

Think about the kind of training data that is needed to learn such a concept. Clearly, there are no (X, y) pairs involved. A language is slowly learned by getting exposed to a large number of sentences that belong it (Only X!) [1]. In this spirit, unsupervised learning can also be thought of as one class classification.

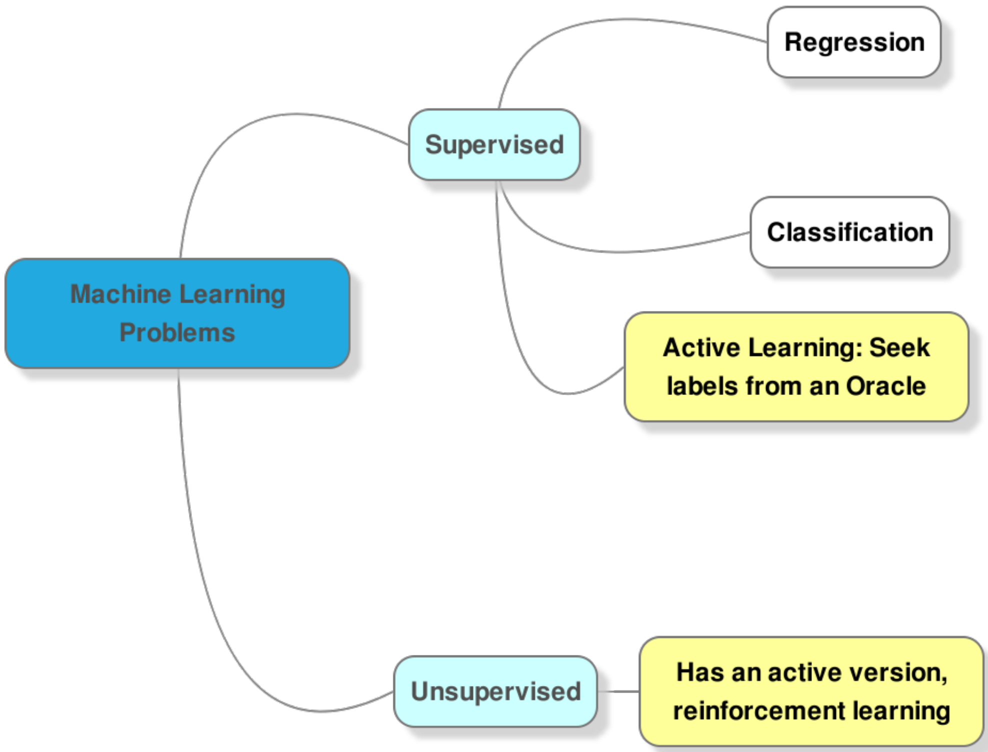
3.1 Reinforcement Learning

There exists an active version of unsupervised learning, called reinforcement learning. As an example, consider how babies figure out what not to do; say what not to touch, like a bucket of hot water. The baby might touch it on

[1](Interested students can look up `http://en.wikipedia.org/wiki/Language_model`)

day 1 and feel the pain. The memories of the temperature of the bucket might be suppressed in 10 days or so, after which he might be tempted to touch the bucket again, only to have his belief that "hot water is bad" reinforced.

A good rule thumb is to say that in supervised setting, we have pairs (X, y), whereas in the unsupervised setting, we just have the X, there is no y



MOTHER

$\downarrow$

$(p_i, t_i) \longrightarrow (\underbrace{\varphi(p_i), t_i}_{\text{memorize}})$

BABY

$\downarrow$

t_j'

Evaluation

$(p_j, t_j) \xrightarrow{\text{hide } t_j} t_j'$

F_{XY}

$\downarrow$

$\{(x_i, y_i)\}_{i=1 \text{ to } m} \; \mathcal{D}$

m times

$D = \{(X_i, Y_i)\}_{i=1 \text{ to } m}$

$\mathcal{T}$ (test net) $\longrightarrow y_j'$

NN model

$\downarrow$

Storing $\mathcal{D}$ (TRAINING)

$\downarrow$

y_j'

$$P\left[Y \neq g_m^{NN}(X) / X = x, X' = x'\right]$$

$$= P\left[Y = 0, Y' = 1 / X = x, X' = x'\right] + P\left[Y = 1, Y' = 0 / X = x, X' = x'\right]$$

$$= P\left[Y = 0 / X = x\right] P\left[Y' = 1 / X = x, X' = x'\right] + \text{''}$$

$$= (1 - \eta(x))(\eta(x')) + \eta(x)(1 - \eta(x'))$$

$$\xrightarrow[m \to \infty]{} \underline{2\eta(x)(1 - \eta(x'))}$$

$X' \to X$ as $m \to \infty$

$\eta(X') \to \eta(X)$ as $m \to \infty$

$$R^{NN}(x) = 2\eta(x)(1-\eta(x))$$

$$\underbrace{E[R^{NN}(x)]}_{R^{NN}} = 2E[\eta(x)(1-\eta(x))]$$

$$\leq 2R^*$$

Also, $R^* = E[\min(\eta(x), 1-\eta(x))]$