Machine Learning

Fractions

Ranojoy Dutta

Lecture #1

2022

Introduction

sec:intro

Machine Learning is actively being used today, perhaps in many more places that we'd expect. For example,

1. You realise that it's your friend's birthday and want to send them a card via some internet store. You search for funny cards, and the search engine shows you the 10 most relevant links. You click the third link; the search engine learns from this.

2. You check some emails, and without you noticing it, the spam filters catches unsolicited ads for pharmaceuticals and places them in the Spam Folder.

3. You went out shopping for some pizza crusts. When you get to the checkout and purchase the items, the human operating the cash register hands you a coupon for 10% discount off a six pack beer. The cash register's software generated this coupon for you because people who buy diapers also tend to buy beer.

4. You go to a loan agent and ask them if you are eligible for a loan. They don't ask you questions, rather plugs in some financial information about you in the machine and the decision is made.

5. You went to a doctor with some symptoms. Your symptoms are fed to a machine and a decision is made to find out the the possible ailments that you might have.

What is Machine Learning?

Machine Learning is turning data into information. It lies at the intersection of computer science, engineering, and statistics and often appears in other disciplines. Machine Learning is a tool that can be applied to many problems. Any field that needs to interpret and act on data can benefit from machine learning techniques.

Although it is often difficult to model a problem. For example, Do humans not act to maximize their own happiness? Can't we just predict the outcome of events involving humans based on this assumption? But its difficult to define what makes everyone happy, because this may differ greatly from one person to the next. So, even if our assumption are correct about people maximizing their

own happiness, the definition of happiness is too complex to model. Happiness cannot be modelled deterministically.

For example, say we build a cat classification system. This sort of system is an interesting topic often associated with machine learning called *expert systems*.

Terminology : Training and Testing

Weight (kg)	Height (cm)	Age (yr)	Gender	Back pain
64	172	25	Male	No
54	158	18	Female	No
72	178	38	Male	Yes
102	172	28	Male	Yes
60	163	32	Female	No
92	165	42	Female	Yes

table: Human Stat

Table 1: Human Statistics

In table are some basic human statistics that we decided to measure. We chose to measure weight, height, age and gender. In reality, you'd want to measure more than this. It is common practice to measure just about anything you can measure and sort out the important parts later. The four things measured are called *features*, also called *attributes*. The first three features in the table are numeric whereas the fourth feature is binary : it can only be 1 or 0.

One task of machine learning is *classification*. Suppose using the information in Table we want to find out the information whether a person is affected by back pain. Ofcourse we would need much more data, but for the moment assume we have all that information. How would we then decide whether a person has back pain or not ? This task is called *classification*, and there are many machine learning algorithms that are good at classification.

Now let us assume that we have decided on which machine learning algorithm to use for classification. What we would do next is to train the algorithm, or allow it to learn. To train the algorithm we feed it quality data known as *training set*. A training set is a set of training examples we will use to train out machine learning algorithm. In table our training set has six *training examples*. Each training example has four features and one *target variable*. The target variable is what we will be trying to predict with our machine learning algorithm.

In classification the target variable takes on a nominal value, and in task of regression its value could be continuous. In classification the target variables are called *classes*, and there is assumed to be a finite number of classes.

To test ML algorithms, we usually have a training set of data, and a separate dataset called the *test set*.

- Initially the program is fed the training examples; this is when the machine learning takes place.

- Next, the test set is fed to the program. The target variable for each example from the test set isn't given to the program and the program decide which class each example should belong to. The target variable that the training training example belongs to is then compared to the predicted value, and we can get a sense for how accurate the algorithm is .

Different classes of Machine Learning

Applications in which the training data comprises examples of the input vectors along with their corresponding target vectors are known as *supervised learning* problems. If the aim is to assign each input vector to one of a finite number of distinct categories, then they are called classification problem. If the desired output consists of one or more continuous variables, then the task is called regression.

In other pattern recognition problems, the training data consists of a set of input vectors x without any corresponding target values. The goal in such *unsupervised learning* problems may be to discover groups of similar examples within the data, where it is called *clustering*, or to determine the distribution of data within the input space, known as *density estimation*, or to project the data from a high dimensional space down to two or three dimensions for the purpose of *visualization*.

The various types of Machine Learning Systems are :

1. Whether or not they are trained with human supervision (*supervised* , *unsupervised*, and *Reinforcement Learning*)
2. Whether or not they can learn incrementally on the fly (*online* vs *batch learning*)
3. Whether they work by simply comparing new data points to known data points , or instead by detecting patterns in the training data and building a predictive model, much like scientists do (*instance based* versus *model based learning*)

   For example, a state of the art spam filter may learn of the fly using a deep neural networks model trained using examples of spam and ham; this makes it an online, model based, supervised learning system.

Key task of machine learning

Machine Learning is usually great for :

- Problem for which existing solutions require a lot of fine-tuning or long lists of rules : one ML algorithm can often simplify code and perform better than the traditional approach.
- Complex problems for which using a traditional approach yields no good solution : the best ML techniques can perhaps find a solution.
- Fluctuating environments : a ML system can adapt to new data.

In this section , we will outline the key tasks of machine learning and set a framework that allows us to easily turn a machine learning algorithm into a solid working application.

The example covered before was for the task of classification. In classification, the task is to predict what class an instance of data should fall into.

Another task in machine learning is *regression*. Regression is the prediction of a numeric value.

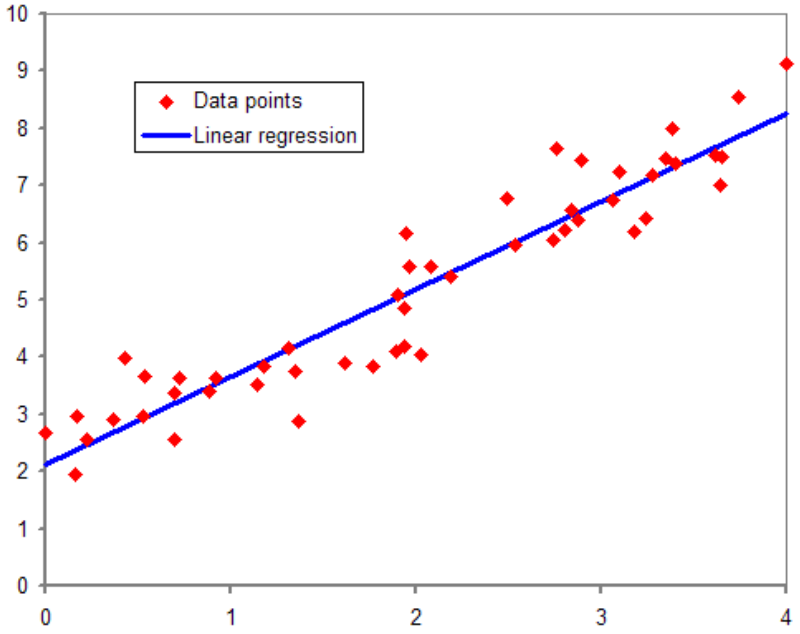

Figure 1: One ball.

Classification and regression are examples of *supervised learning*.

Summary of Developing an ML algorithm

We will be understanding and developing machine learning algorithm by the following procedure :

1. *Collect Data*. We would be collecting the samples by scraping a website and extracting data, or can retrieve information using an API . Also, we could use publicly available data.
2. *Preparing the input data*. Once we have the data, we need to make sure that it is in usable format. The format depends upon the languauge we are going to use to the analysis.
3. *Analyze the input data*. We would look at the data to see if we could recognise any patterns or if there is anything obvious, like a few data points that are vastly different from the rest of the set.

4. *Training the algorithm.* In this step, we feed the algorithm good clean data from the first two steps and extract knowledge or information. The knowledge is often stored in a format that's readily useable by a machine.

5. *Testing the algorithm.* When we are evaluating an algorithm, we are going to test it to see how well it does. For supervised learning, we may already know some values that we could use to evaluate the algorithm. In the case of unsupervised algorithm, we might need to use some other metrics to evaluate the success.

6. *Using the algorithm.* After the checking, we can write a program to do some task.