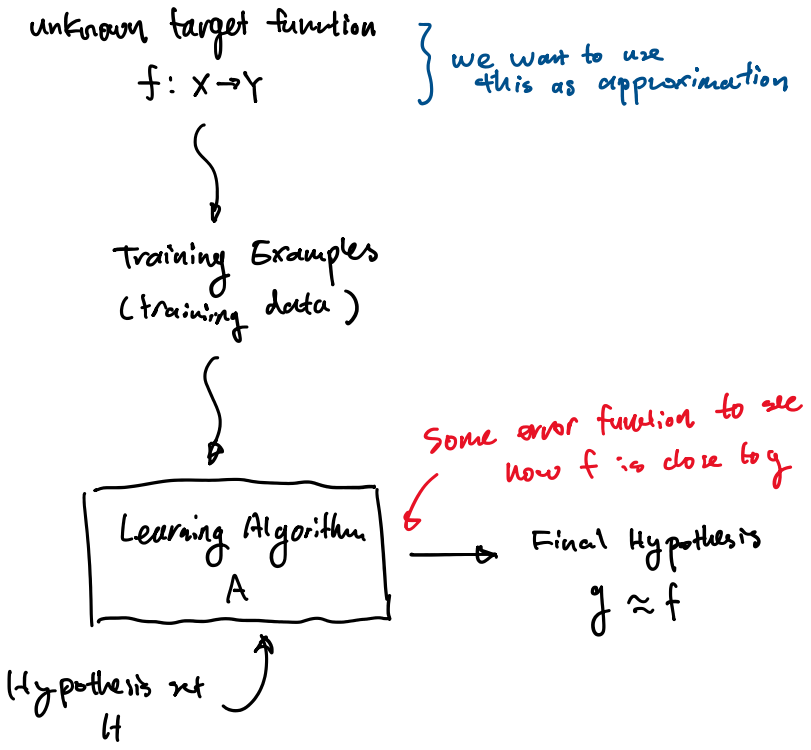
Machine Learning Introduction

Tuesday, January 14, 2020 12:40

How ML works fundamentally



ex. suppose trning example

$$\mathcal{D} = \{(x_1, y_1) \dots (x_N, y_N)\}$$

input vector $\vec{x} \in X = \mathbb{R}^d$, d-dimensional
output vector $\vec{y} \in Y = \{-1, +1\}$

Hypothesis $\mathcal{H}$: $h: \mathbb{R}^d \to \{-1, +1\}$

we can use a weighted sum and compare w/ threshold value (t)

$$\forall x \in \mathcal{R}^d,\ y = h(x) = \begin{cases} +1, & \text{if } \sum_{i=1}^{d} w_i x_i \geq t \\ -1, & \text{if } \sum_{i=1}^{d} w_i x_i < t \end{cases}$$

Simplified to $\vec{w}^T \vec{x}$

$$\text{sign}\left(\sum_{i=1}^{d} w_i x_i + b\right)$$

b was

or alternatively with Perceptron

prepend (1) to input $\vec{x}$ ($\vec{x} = \{1, x_0, x_1, \dots\}$)

prepend (1) to input $\vec{x}$ ($\vec{x} = \{1, x_0, x_1, \ldots\}$)
then start from $i=0$:

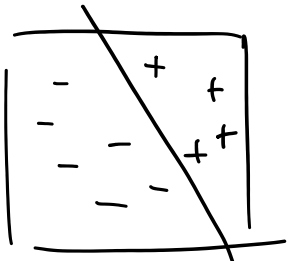
$$y = h(x) = \text{sign}\left(\sum_{i=0}^{d} w_i x_i\right)$$

we're free to choose the weights (w) so that $h(x)$ fits the training data

ex. $d=2$ (2D linear estimation)

then $\vec{x} = (x_1, x_2)$

$\vec{x}_{aug} = (1, x_1, x_2)$



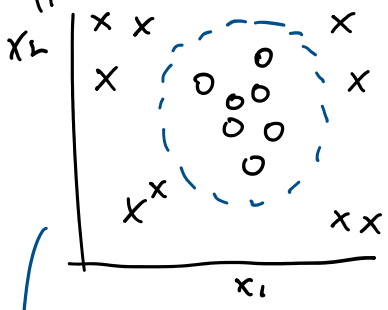
then using algorithm from prev. example:

$$y = \text{sign}\left(\sum_{i=0}^{d} w_i x_i\right)$$

$$= w_0 + w_1 x_1 + w_2 x_2$$

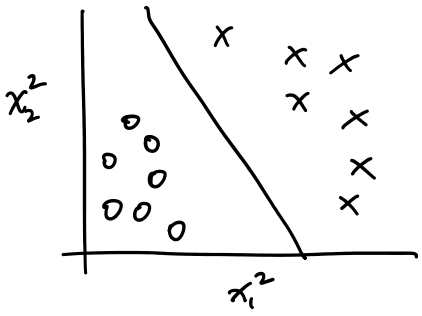
Transformation of Variables / Kernel Trick.

Suppose:



Instead of using x_1, x_2 directly, we can use a function of x_1, x_2. In this case, we can square the inputs.

transformed to



this is an example of SUPERVISED LEARNING