Regression

Motivation

H^L is one of the most useful families of hypothesis classes.

Many models that are used in practice rely on linear predictors.

There exist many types of linear models, including:

- Perceptron.
- Linear regression.
- Logistic regression.

Motivation

Let's define the class of affine functions:

$$L_d = \{h_{w,b} : w \in \mathbb{R}^d \;\; and\; b \in \mathbb{R}\}$$

Where:

$$h_{w,b}(x) = \langle w, x\rangle + b = \sum_{i=1}^{d} w_i x_i + b$$

To simplify the notation, we will integrate the bias a s an extra coordinate into w:

$$h_w(x) = \sum_{i=0}^{d} w_i x_i$$

Hence, the class of affine functions is called « homogenous affine functions »

$$L_d = \{h_w : w \in \mathbb{R}^{d+1}\}$$

Motivation

Therefore, we can generate different hypothesis classes H^L, defining different models, by using the composition of φ over L_d such that:

$$\varphi : \mathbb{R} \to Y$$

Perceptron:

$$\varphi_p(x) = sign(x) \text{ and } Y = \{-1, +1\}$$

$$H_p = \varphi_p \circ L_d$$

Linear regression:

$$\varphi_{reg}(x) = x \text{ and } Y = \mathbb{R}$$

$$H_{reg} = \varphi_{reg} \circ L_d$$

Logistic regression:

$$\varphi_{sig}(x) = \frac{1}{1+e^{-x}} \text{ and } Y = [0,1]$$

$$H_{sig} = \varphi_{sig} \circ L_d$$

Linear Regression

Definition:
Linear regression is a type of model used for regression tasks by studying the relationship between some explanatory variables and some real valued outcome.

Here we have:

$$\mathcal{X} \subset \mathbb{R}^d \text{ for some } d$$

And

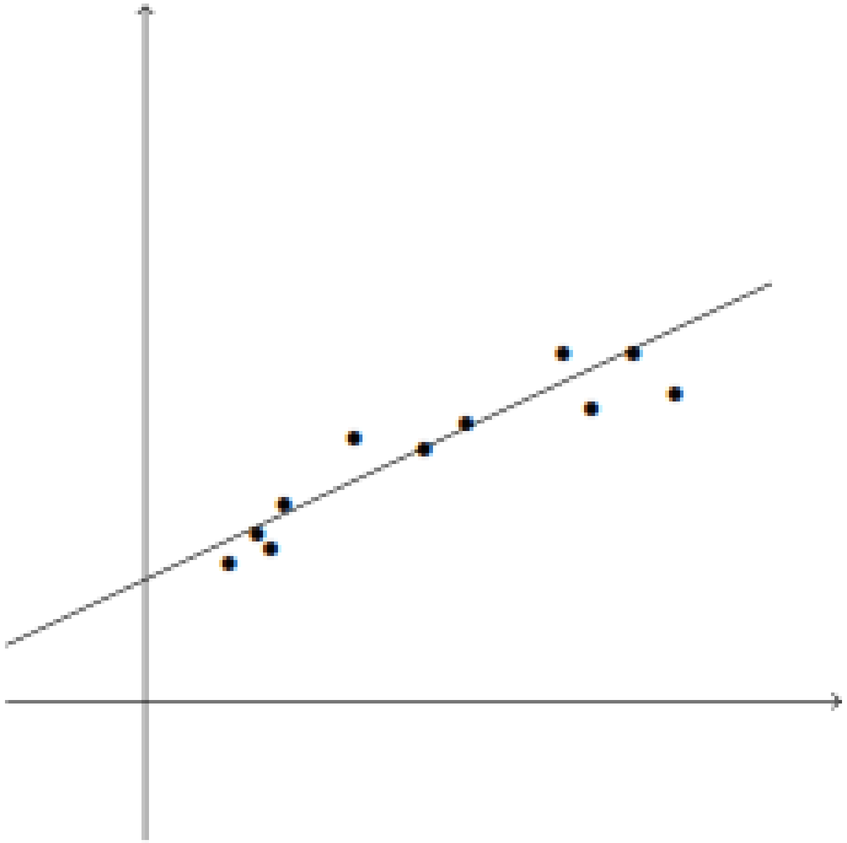
$$Y = \mathbb{R}$$

Objective:
Learn a linear predictor that best approximate the relationship between our variables:

$$h_w : \mathbb{R}^d \longrightarrow \mathbb{R}$$

Linear Regression

Example:
Predicting the weight of a baby as a function of his age and weight at birth.
Here, $d = 1$.



Linear Regression

The hypothesis class for linear regression model:

In linear regression model, we have:

$$\varphi_{reg}(x) = x$$

The hypothesis class of linear regression predictors is simply the set of linear functions:

$$H_{reg} = \varphi \circ L_d = L_d$$

$$H_{reg} = \{x \longmapsto \langle w, x \rangle : w \in \mathbb{R}^{d+1}\}$$

Linear Regression

The loss function for linear regression model:
It measures how much the model should be penalized for the discrepancy between $h_w(x)$ and y. One common way is to use the squared-loss function:

$$l\big(h_w, (x, y)\big) = (h_w(x) - y)^2$$

For this loss function, the empirical risk is called the Mean Squared Error:

$$L_S(h_w) = \frac{1}{m}\sum_{i=1}^{m}(h_w(x_i) - y_i)^2$$

Notice:
There are a variety of other loss functions that one can use, for example, the absolute value loss function:

$$l\big(h_w, (x, y)\big) = |h_w(x) - y|$$

Linear Regression

The learning algorithm for linear regression model:
The learning algorithm follows ERM_H learning rule.

Least squares:
Least squares is the algorithm that solves the ERM_H problem for the hypothesis class of linear regression predictors with respect to squared loss.

$$\underset{w}{\operatorname{argmin}}\, L_S(h_w) = \underset{w}{\operatorname{argmin}} \left(\frac{1}{m} \sum_{i=1}^{m} (\langle w, x_i \rangle - y_i)^2 \right)$$

To solve this problem, we calculate the gradient of the objective function and compare it to zero. That is, we need to solve:

$$\frac{2}{m} \sum_{i=1}^{m} (\langle w, x_i \rangle - y_i) x_i = 0$$

Linear Regression

We can rewrite the problem as the problem :

$$Aw = b$$

Where:

$$A = \left(\sum_{i=1}^{m} x_i . x_i^T \right)$$

And

$$b = \sum_{i=1}^{m} y_i x_i$$

Linear Regression

Or in matrix form:

$$A = \begin{pmatrix} \vdots & & \vdots \\ x_1 & \dots & x_m \\ \vdots & & \vdots \end{pmatrix} \begin{pmatrix} \vdots & & \vdots \\ x_1 & \dots & x_m \\ \vdots & & \vdots \end{pmatrix}^T$$

And

$$b = \begin{pmatrix} \vdots & & \vdots \\ x_1 & \dots & x_m \\ \vdots & & \vdots \end{pmatrix} \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix}$$

If A is invertible, then the solution to the ERM problem is:

$$w = A^{-1}b$$

Linear Regression

If A is not invertible, we require a few standard tools from linear algebra.

A is not invertible when the training data do not cover the entire space of $\mathbb{R}^d$.

Even if A is not invertible, we can always find a solution to the system:

$$Aw = b$$

because b is in the range of A.

Indeed, since A is symmetric, then we can write it using its eigenvalue decomposition as:

$$A = VDV^T$$

Where:
D is a diagonal matrix.
V is an orthonormal matrix (because $V^TV = I$ which is a $d \times d$ matrix).

Linear Regression

Let's define D^+ to be the diagonal matrix such that:

$$\begin{cases} D^+_{i,i} = 0 \quad if \quad D_{i,i} = 0 \\ D^+_{i,i} = \dfrac{1}{D_{i,i}} \quad if \quad D_{i,i} \neq 0 \end{cases}$$

Now, define:

$$A^+ = VD^+V^T \text{ and } \hat{w} = A^+b$$

Let v_i denote the ith column of V. Then we have:

$$A\hat{w} = AA^+b = VDV^TVD^+V^Tb = VDD^+V^Tb = \sum_{i\,:\,D_{i,i}\neq 0} v_i v_i^T b$$

This means that $A\hat{w}$ is the projection of b on the space of vectors v_i for which $D_{i,i} \neq 0$.

Linear Regression

Since the linear space of $(x_1, \ldots, x_m)$ is the same as the linear space of those v_i.

And, since b is in the linear space of x_i.

We obtain that:

$$A\hat{w} = b$$

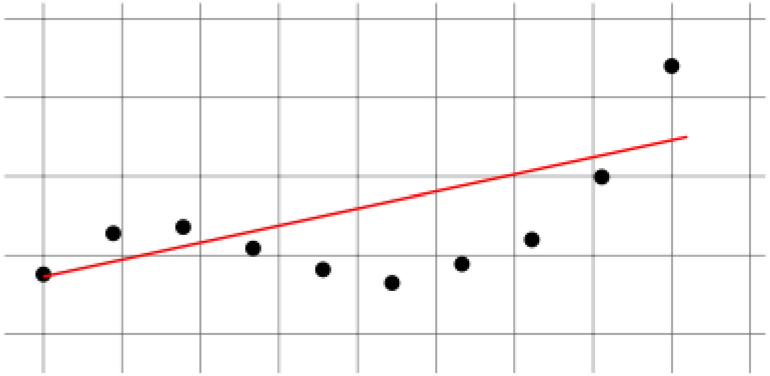
Then, $\hat{w}$ is a solution of $Aw = b$.

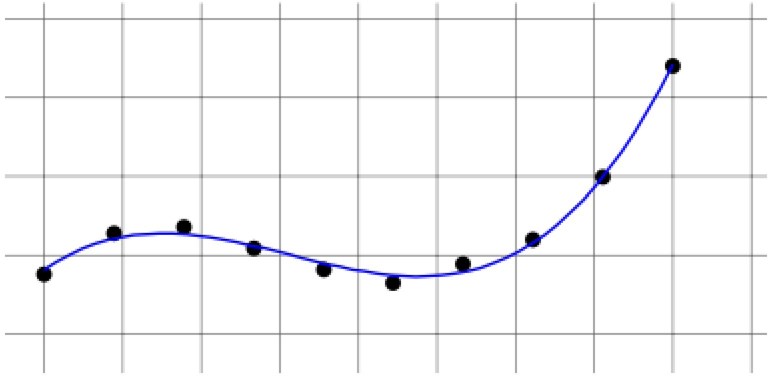
Linear Regression for polynomial regression tasks

Some learning tasks call for nonlinear predictors, such as polynomial predictors. Let's consider a one dimensional polynomial function of degree n:

$$p(x) = w_0 + w_1 x + w_2 x^2 + \cdots + w_n x^n$$

Where $(w_0, \ldots, w_n)$ is a vector of coefficients of size $n + 1$.





Linear Regression for polynomial regression tasks

We will focus on the class of one dimensional, n-degree, polynomial regression hypotheses. Therefore, the class of polynomial hypotheses is:

$$H_{poly}^{n} = \{x \longmapsto p(x)\}$$

Where p is a one dimensional polynomial of degree n, parameterized by a vector of coefficients $(w_0, \dots, w_n)$.

In that case, we have:

$$\mathcal{X} = \mathbb{R} \quad \text{and} \quad Y = \mathbb{R}$$

One way to learn the class H_{poly}^{n} is by reduction to the problem of linear regression.

Linear Regression for polynomial regression tasks

To translate a polynomial regression problem to a linear regression problem, we define the mapping:

$$\psi : \mathbb{R} \longrightarrow \mathbb{R}^{n+1}$$

Such that:

$$\psi(x) = (1, x, x^2, \dots, x^n)$$

Then, we have that:

$$p\big(\psi(x)\big) = w_0 + w_1 x + w_2 x^2 + \cdots + w_n x^n = \langle w, \psi(x) \rangle$$

Finally, we can find the optimal vector of coefficients w by using the Least Squares Algorithm.

Logistic Regression

Definition:
Logistic regression is a type of model used for classification tasks by studying the relationship between some explanatory variables and some binary outcome.

Here we have:

$$\mathcal{X} \subset \mathbb{R}^d \text{ for some } d$$

And

$$Y = \{-1, +1\}$$

Objective:
Learn a linear predictor that best approximate the relationship between our variables:

$$h_w\colon \mathbb{R}^d \longrightarrow [0,1]$$

We can interpret $h_w(x)$ as the probbability that the label of x is 1:

$$h_w(x) = P(y = 1|x)$$

Logistic Regression

The hypothesis class for logistic regression model:

In logistic regression model, we have:

$$\varphi_{sig}(x) = \frac{1}{1 + e^{-x}}$$

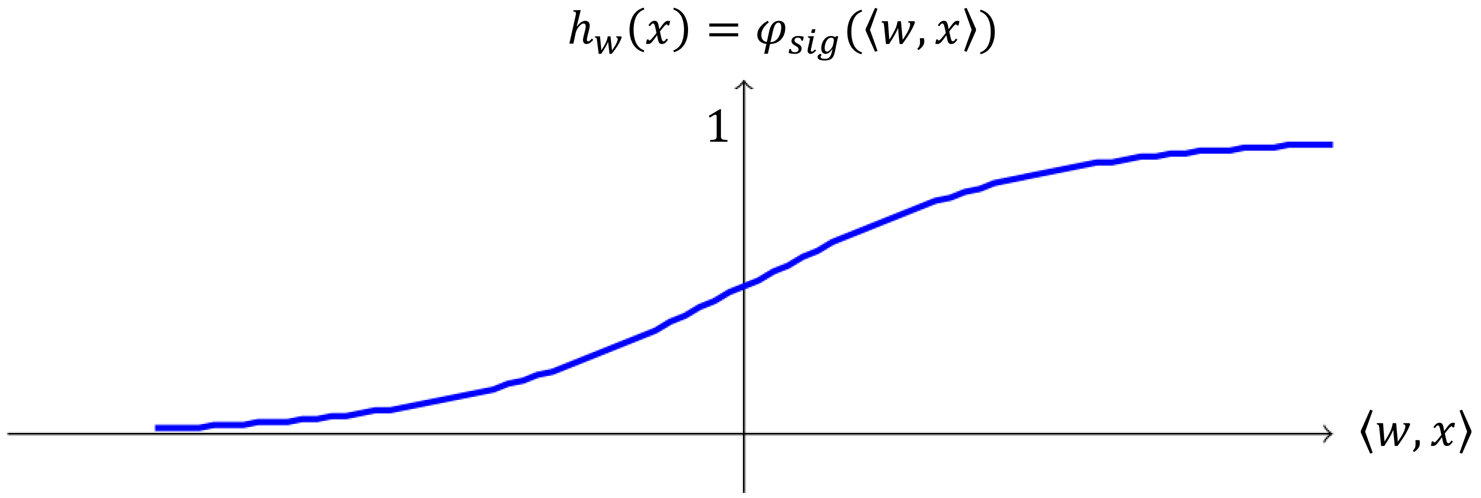
The hypothesis class of logistic regression predictors is the composition of a sigmoid function over the set of linear functions:

$$H_{sig} = \varphi \circ L_d$$

$$H_{sig} = \{x \mapsto \varphi(\langle w, x \rangle) = \frac{1}{1 + e^{-\langle w,x \rangle}} : w \in \mathbb{R}^{d+1}\}$$

Logistic Regression

The name « sigmoid » means «S-shaped », referring to the plot of this function shown in the figure:



Logistic Regression

Logistic regression Vs Perceptron:

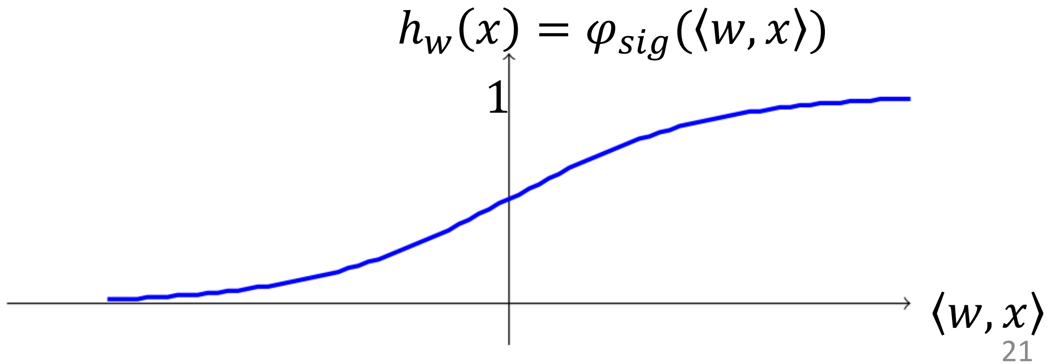
Whenever, $|\langle w, x \rangle|$ is large, the predictions of logistic regression hypothesis and perceptron hypothesis are similar.

However, whenever $|\langle w, x \rangle|$ is close to zero, we have that:

$$\varphi_{sig}(\langle w, x \rangle) \approx \frac{1}{2} \quad \text{and} \quad \varphi_p(\langle w, x \rangle) = \operatorname{sign}(\langle w, x \rangle)$$

The logistic regression hypothesis is not sure about the value of the label.

The perceptron hypothesis always outputs a deterministic prediction $\{-1, +1\}$, even if $|\langle w, x \rangle|$ is very close to zero.



Logistic Regression

The loss function for logistic regression model:
It measures how bad it is to predict some $h_w(x) \in [0,1]$ given that the true label is $y = \{\pm 1\}$.

Clearly, we want that:

$$P(y|x) = \begin{cases} h_w(x) \;\; if \;\; y = +1 \\ 1 - h_w(x) \;\; if \; y = -1 \end{cases}$$

to be large.
We have:

$$h_w(x) = \frac{1}{1+e^{-\langle w,x \rangle}} \text{ and } 1 - h_w(x) = \frac{1}{1+e^{\langle w,x \rangle}}$$

Generally:

$$P(y|x) = \frac{1}{1 + e^{-y\langle w,x \rangle}}$$

Logistic Regression

It is clear that the loss function will increase monotonically if the probability $P(y|x)$ decreases.

This implies that, it will increse monotonically if $1 + e^{-y\langle w,x\rangle}$ increases.

Therefore, the loss function used in logistic regression penalizes h_w based on the log of $1 + e^{-y\langle w,x\rangle}$, that is:

$$l\big(h_w, (x, y)\big) = log\big(1 + e^{-y\langle w,x\rangle}\big)$$

(recall that the log is a monotonic function).

Therefore, given a training set $S = (x_1, y_1), \dots, (x_m, y_m)$, the ERM problem associated with logistic regression is:

$$\underset{w}{\operatorname{argmin}}\, L_S(h_w) = \underset{w \in \mathbb{R}^d}{\operatorname{argmin}} \left(\frac{1}{m} \sum_{i=1}^{m} log\big(1 + e^{-y\langle w,x\rangle}\big) \right)$$

Logistic Regression

Notice:
It is clear that the loss function of the logistic regression is a convex function with respect to w.

So, the ERM_H problem for logisitic regression model can be solved using a gradient descent algorithm.