Logistic Regression

CS60010: Deep Learning

Abir Das

IIT Kharagpur

Jan 19 and 20, 2022

Agenda

- Understand regression and classification with linear models.
- Brush-up concepts of maximum likelihood and its use to understand linear regression.
- Using logistic function for binary classification and estimating logistic regression parameters.

Resources

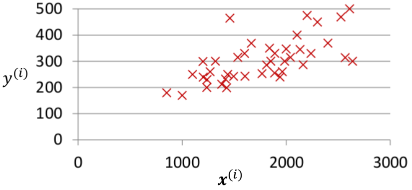
- The Elements of Statistical Learning by T Hastie, R Tibshirani, J Friedman. [Link] [Chapter 3 and 4]
- Artificial Intelligence: A Modern Approach by S Russell and P Norvig. [Link] [Chapter 18]

Linear Regression

- In a regression problem we want to find the relation between some input variables $\mathbf{x}$ and output variables y, where $\mathbf{x} \in \mathbb{R}^d$ and $y \in \mathbb{R}$.
- Inputs are also often referred to as covariates, predictors and features; while outputs are known as variates, targets and labels.
- Examples of such input-output pairs can be
  - {Outside temperature, People inside classroom, target room temperature | Energy requirement}
  - {Size, Number of Bedrooms, Number of Floors, Age of the Home | Price}

Linear Regression

- In a regression problem we want to find the relation between some input variables $\mathbf{x}$ and output variables y, where $\mathbf{x} \in \mathbb{R}^d$ and $y \in \mathbb{R}$.
- Inputs are also often referred to as covariates, predictors and features; while outputs are known as variates, targets and labels.
- Examples of such input-output pairs can be
  - {Outside temperature, People inside classroom, target room temperature | Energy requirement}
  - {Size, Number of Bedrooms, Number of Floors, Age of the Home | Price}
- We have a set of N observations of y as $\{y^{(1)}, y^{(2)}, \cdots, y^{(N)}\}$ and the corresponding input variables $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \cdots, \mathbf{x}^{(N)}\}$.



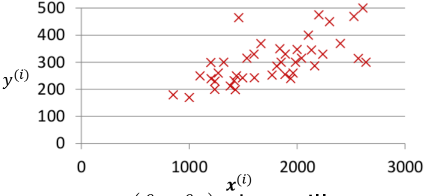
Linear Regression

- The input and output variables are assumed to be related via a relation, known as hypothesis. $\widehat{y} = h_{\boldsymbol{\theta}}(\mathbf{x})$, where $\boldsymbol{\theta}$ is the parameter vector.
- The goal is to predict the output variable $\widehat{y^*} = f(\mathbf{x}^*)$ for an arbitrary value of the input variable $\mathbf{x}^*$.
- Let us start with scalar inputs (x) and scalar outputs (y).

Univariate Linear Regression

- **hypothesis**: $h_{\boldsymbol{\theta}}(x) = \theta_0 + \theta_1 x$.
- **Cost Function**: Sum of squared errors.

$$J(\theta_0, \theta_1) = \frac{1}{2N} \sum_{i=1}^{N} \left(h_{\boldsymbol{\theta}}(x^{(i)}) - y^{(i)}\right)^2$$

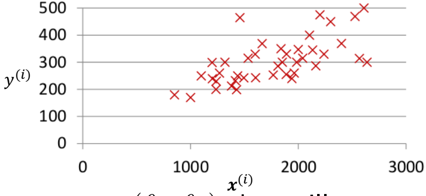


- **Optimization objective**: find model parameters (θ_0, θ_1) that will minimize the sum of squared errors.

Univariate Linear Regression

- hypothesis: $h_{\boldsymbol{\theta}}(x) = \theta_0 + \theta_1 x$.
- Cost Function: Sum of squared errors.

$$J(\theta_0, \theta_1) = \frac{1}{2N} \sum_{i=1}^{N} \left(h_{\boldsymbol{\theta}}(x^{(i)}) - y^{(i)}\right)^2$$



- Optimization objective: find model parameters (θ_0, θ_1) that will minimize the sum of squared errors.
- Gradient of the cost function w.r.t. θ_0:

$$\frac{J(\theta_0, \theta_1)}{\theta_0} = \frac{1}{N} \sum_{i=1}^{N} \left(h_{\boldsymbol{\theta}}(x^{(i)}) - y^{(i)}\right)$$

- Gradient of the cost function w.r.t. θ_1:

$$\frac{J(\theta_0, \theta_1)}{\theta_1} = \frac{1}{N} \sum_{i=1}^{N} \left(h_{\boldsymbol{\theta}}(x^{(i)}) - y^{(i)}\right) x^{(i)}$$

- Apply your favorite gradient based optimization algorithm.

Univariate Linear Regression

§ These being linear equations of $\boldsymbol{\theta}$, have a unique closed form solution too.

$$\theta_1 = \frac{N\sum\limits_{i=1}^{N} y^{(i)}x^{(i)} - \big(\sum\limits_{i=1}^{N} x^{(i)}\big)\big(\sum\limits_{i=1}^{N} y^{(i)}\big)}{N\sum\limits_{i=1}^{N}\big(x^{(i)}\big)^2 - \big(\sum\limits_{i=1}^{N} x^{(i)}\big)^2}$$

$$\theta_0 = \frac{1}{N}\Big\{\sum_{i=1}^{N} y^{(i)} - \theta_1 \sum_{i=1}^{N} x^{(i)}\Big\}$$

Multivariate Linear Regression

- We can easily extend to multivariate linear regression problems, where $\mathbf{x} \in \mathbb{R}^d$
- hypothesis: $h_{\boldsymbol{\theta}}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_d x_d$. For convenience of notation, define $x_0 = 1$.
- Thus h is simply the dot product of the parameters and the input vector.

$$h_{\boldsymbol{\theta}}(\mathbf{x}) = \boldsymbol{\theta}^T \mathbf{x}$$

- Cost Function: Sum of squared errors.

$$J(\boldsymbol{\theta}) = J(\theta_0, \theta_1, \cdots, \theta_d) = \frac{1}{2N} \sum_{i=1}^{N} \left(\boldsymbol{\theta}^T \mathbf{x}^{(i)} - y^{(i)}\right)^2 \tag{1}$$

- We will use the following to write the cost function in a compact matrix vector notation

$$h_{\boldsymbol{\theta}}(\mathbf{x}) = \boldsymbol{\theta}^T \mathbf{x} = \mathbf{x}^T \boldsymbol{\theta}$$

Multivariate Linear Regression

$$\begin{bmatrix} \widehat{y}^{(1)} \\ \widehat{y}^{(2)} \\ \vdots \\ \widehat{y}^{(N)} \end{bmatrix} = \begin{bmatrix} h_\theta(\mathbf{x}^{(\mathbf{1})}) \\ h_\theta(\mathbf{x}^{(\mathbf{2})}) \\ \vdots \\ h_\theta(\mathbf{x}^{(\mathbf{N})}) \end{bmatrix} = \begin{bmatrix} \mathbf{x}_0^{(1)} & \mathbf{x}_1^{(1)} & \mathbf{x}_2^{(1)} & \cdots & \mathbf{x}_d^{(1)} \\ \mathbf{x}_0^{(2)} & \mathbf{x}_1^{(2)} & \mathbf{x}_2^{(2)} & \cdots & \mathbf{x}_d^{(2)} \\ \vdots & \vdots & \ddots & \vdots & \\ \mathbf{x}_0^{(N)} & \mathbf{x}_1^{(N)} & \mathbf{x}_2^{(N)} & \cdots & \mathbf{x}_d^{(N)} \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_d \end{bmatrix} \quad (2)$$

$$\widehat{\mathbf{y}} = \mathbf{X}\boldsymbol{\theta}$$

Here, $\mathbf{X}$ is a $N \times (d+1)$ matrix with each row an input vector. $\widehat{\mathbf{y}}$ is a N length vector of the outputs in the training set.

Multivariate Linear Regression

§ Eqn. (1), gives,

$$J(\boldsymbol{\theta}) = \frac{1}{2N}\sum_{i=1}^{N}\left(\boldsymbol{\theta}^T\mathbf{x}^{(i)} - y^{(i)}\right)^2 = \frac{1}{2N}\sum_{i=1}^{N}\left(\widehat{y}^{(i)} - y^{(i)}\right)^2 \tag{3}$$

$$= \frac{1}{2N}||\widehat{\mathbf{y}} - \mathbf{y}||_2^2 = \frac{1}{2N}\left(\widehat{\mathbf{y}} - \mathbf{y}\right)^T\left(\widehat{\mathbf{y}} - \mathbf{y}\right)$$

$$= \frac{1}{2N}\left(\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\right)^T\left(\mathbf{X}\boldsymbol{\theta} - \mathbf{y}\right) = \frac{1}{2N}\left\{\boldsymbol{\theta}^T\left(\mathbf{X}^T\mathbf{X}\right)\boldsymbol{\theta} - \boldsymbol{\theta}^T\mathbf{X}^T\mathbf{y} - \mathbf{y}^T\mathbf{X}\boldsymbol{\theta} + \mathbf{y}^T\mathbf{y}\right\}$$

$$= \frac{1}{2N}\left\{\boldsymbol{\theta}^T\left(\mathbf{X}^T\mathbf{X}\right)\boldsymbol{\theta} - \left(\mathbf{X}^T\mathbf{y}\right)^T\boldsymbol{\theta} - \left(\mathbf{X}^T\mathbf{y}\right)^T\boldsymbol{\theta} + \mathbf{y}^T\mathbf{y}\right\}$$

$$= \frac{1}{2N}\left\{\boldsymbol{\theta}^T\left(\mathbf{X}^T\mathbf{X}\right)\boldsymbol{\theta} - 2\left(\mathbf{X}^T\mathbf{y}\right)^T\boldsymbol{\theta} + \mathbf{y}^T\mathbf{y}\right\}$$

Multivariate Linear Regression

§ Equating the gradient of the cost function to 0,

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \frac{1}{2N}\left\{2\mathbf{X}^T\mathbf{X}\boldsymbol{\theta} - 2\mathbf{X}^T\mathbf{y} + 0\right\} = 0$$
$$\mathbf{X}^T\mathbf{X}\boldsymbol{\theta} - \mathbf{X}^T\mathbf{y} = 0$$
$$\boldsymbol{\theta} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y} \tag{4}$$

Multivariate Linear Regression

§ Equating the gradient of the cost function to 0,

$$\begin{aligned} \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) &= \frac{1}{2N}\left\{2\mathbf{X}^T\mathbf{X}\boldsymbol{\theta} - 2\mathbf{X}^T\mathbf{y} + 0\right\} = 0 \\ \mathbf{X}^T\mathbf{X}\boldsymbol{\theta} - \mathbf{X}^T\mathbf{y} &= 0 \\ \boldsymbol{\theta} &= \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y} \end{aligned} \tag{4}$$

§ This gives a closed form solution, but another option is to use iterative solution (just like the univariate case).

$$\frac{\partial J(\boldsymbol{\theta})}{\partial \theta_j} = \frac{1}{N}\sum_{i=1}^{N}\left(h_{\boldsymbol{\theta}}(x^{(i)}) - y^{(i)}\right)x_j^{(i)}$$

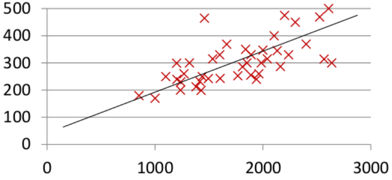
Multivariate Linear Regression

- Iterative Gradient Descent needs to perform many iterations and need to choose a stepsize parameter judiciously. But it works equally well even if the number of features (d) is large.
- For the least square solution, there is no need to choose the step size parameter or no need to iterate. But, evaluating $\left(\mathbf{X}^T\mathbf{X}\right)^{-1}$ can be slow if d is large.

Linear Regression as Maximum Likelihood Estimation

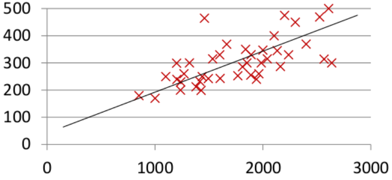
§ So far we tried to fit a "straightline" ("hyperplane" to be more precise) for linear regression problem.

§ This is, in a sense, a "constrained" way of looking at the problem. Datapoints may not be perfectly fit to the hyperplane, but "how uncertain" they are from the hyperplane is never considered.



Linear Regression as Maximum Likelihood Estimation

- So far we tried to fit a "straightline" ("hyperplane" to be more precise) for linear regression problem.
- This is, in a sense, a "constrained" way of looking at the problem. Datapoints may not be perfectly fit to the hyperplane, but "how uncertain" they are from the hyperplane is never considered.



- An alternate view considers the following.
  - $y^{(i)}$ are generated from the $\mathbf{x}^{(i)}$ following a underlying hyperplane.
  - But we don't get to "see" the generated data. Instead we "see" a noisy version of the $y^{(i)}$'s.
  - Maximum likelihood (or in general, probabilistic estimation) models this uncertainty in determining the data generating function.

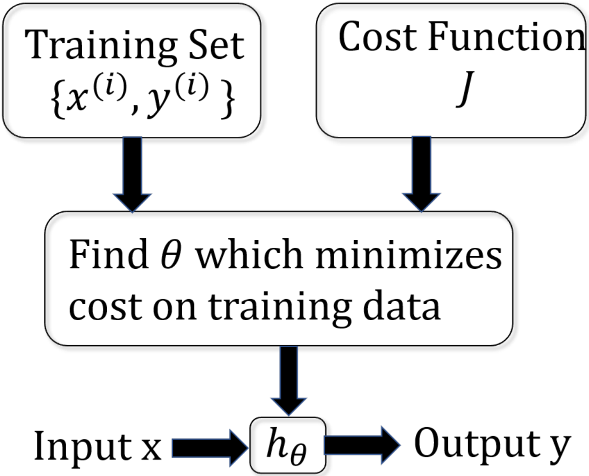
Linear Regression as Maximum Likelihood Estimation

- Thus data are assumed to be generated as follows.

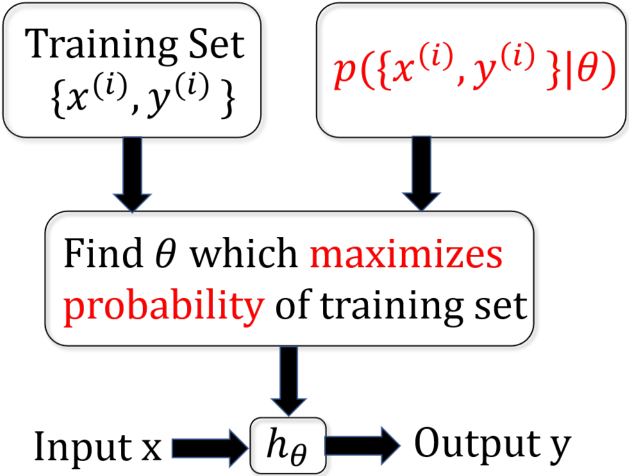
$$y^{(i)} = h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) + \epsilon^{(i)}$$

where $\epsilon^{(i)}$ is an additive noise following some probability distribution.
- So, $\left(\mathbf{x}^{(i)}, y^{(i)}\right)$'s form a joint distribution.
- The idea is to assume a probability distribution on the noise and the probability distribution is parameterised by some additional parameters (*e.g.*, Gaussian with 0 mean and covariance σ^2).
- Then find the parameters (both $\boldsymbol{\theta}$ and σ^2) that is "most likely" to generate the data.

Recall: Cost Function



Alternate View: "Maximum Likelihood"



Maximum Likelihood: Example

- **Intuitive example: Estimate a coin toss**
  I have seen 3 flips of heads, 2 flips of tails, what is the chance of head (or tail) of my next flip?
- **Model:**
  Each flip is a Bernoulli random variable x.
  x can take only *two* values: 1(head), 0(tail)

$$p(x|\theta) = \begin{cases} \theta, & if\ x = 1 \\ 1 - \theta, & if\ x = 0 \end{cases} \tag{5}$$

  where, $\theta \in [0, 1]$, is a parameter to be defined from data
- We can write this probability more succinctly as

$$p(x|\theta) = \theta^x (1 - \theta)^{1-x} \tag{6}$$

Maximum Likelihood: Example

- Let us now assume, that we have flipped the coin a few times and got the results $x_1, ..., x_n$, which are either 0 or 1. The question is what is the value of the probability θ?

Maximum Likelihood: Example

§ Let us now assume, that we have flipped the coin a few times and got the results $x_1, ..., x_n$, which are either 0 or 1. The question is what is the value of the probability θ?

§ Intuitively, one could assume that it is the number of heads we got divided by the total number of coin throws.

Maximum Likelihood: Example

- Let us now assume, that we have flipped the coin a few times and got the results $x_1, ..., x_n$, which are either 0 or 1. The question is what is the value of the probability θ?
- Intuitively, one could assume that it is the number of heads we got divided by the total number of coin throws.
- We will prove in the following that the intuition in this case is correct, by proving that the guess $\theta = \sum_i x_i/n$ is the "most likely" value for the real θ.

Maximum Likelihood: Example

- Let us now assume, that we have flipped the coin a few times and got the results $x_1, ..., x_n$, which are either 0 or 1. The question is what is the value of the probability θ?
- Intuitively, one could assume that it is the number of heads we got divided by the total number of coin throws.
- We will prove in the following that the intuition in this case is correct, by proving that the guess $\theta = \sum_i x_i/n$ is the "most likely" value for the real θ.
- Then the joint probability is

$$f(x_1, ..., x_n; \theta) = \prod_i f(x_i; \theta) = \theta^{\sum_i x_i} (1-\theta)^{n - \sum_i x_i} \quad (7)$$

Maximum Likelihood: Example

- We now want to find the θ which makes this probability the highest.
- It is easier to maximize the *log* of the joint probabilities $log\ \mathcal{L}(\theta) = \sum_i x_i log\ \theta + (n - \sum_i x_i) log\ (1-\theta)$, which yields the same result, since the *log* is monotonously increasing.
- As we may remember, maximizing a function means setting its first derivative to 0.

$$
\begin{aligned}
\frac{\partial log\ \mathcal{L}(\theta)}{\partial \theta} &= \frac{\sum_i x_i}{\theta} - \frac{(n - \sum_i x_i)}{1-\theta} \\
&= \frac{(1-\theta)\sum_i x_i - \theta n + \theta \sum_i x_i}{\theta(1-\theta)} \\
&= \frac{\sum_i x_i - \theta n}{\theta(1-\theta)} = 0 \\
\implies \theta &= \frac{\sum_i x_i}{n}
\end{aligned}
\tag{8}
$$

Maximum Likelihood Estimation

We have $n = 3$ data points $y_1 = \mathbf{1}, y_2 = \mathbf{0.5}, y_3 = \mathbf{1.5}$, which are independent and Gaussian with unknown $mean = \theta$ and $variance = 1$:

$$y_i \sim \mathcal{N}(\theta, 1)$$

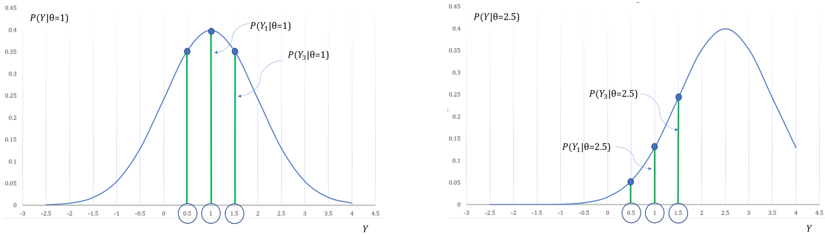
with **likelihood** $\boldsymbol{P}(y_1, y_2, y_3; \theta) = \boldsymbol{P}(y_1; \theta)\boldsymbol{P}(y_2; \theta)\boldsymbol{P}(y_3; \theta)$. Consider two guesses of θ, 1 and 2.5. Which has higher likelihood (probability of generating the three observations)?

Maximum Likelihood Estimation

We have $n = 3$ data points $y_1 = \mathbf{1}, y_2 = \mathbf{0.5}, y_3 = \mathbf{1.5}$, which are independent and Gaussian with unknown $mean = \theta$ and $variance = 1$:

$$y_i \sim \mathcal{N}(\theta, 1)$$

with **likelihood** $\boldsymbol{P}(y_1, y_2, y_3; \theta) = \boldsymbol{P}(y_1; \theta)\boldsymbol{P}(y_2; \theta)\boldsymbol{P}(y_3; \theta)$. Consider two guesses of θ, 1 and 2.5. Which has higher likelihood (probability of generating the three observations)?



Finding the θ that maximizes the likelihood is equivalent to moving the Gaussian until the product of 3 green bars (likelihood)is maximized.

Slide Motivation: *Nando de Freitas* [*Link*]

Maximum Likelihood Estimation of model parameters θ

- In general, we have observations, $\mathcal{D} = \{u^{(1)}, u^{(2)}, \cdots, u^{(N)}\}$
- We assume data is generated by some distribution $U \sim p(U; \theta)$
- Compute the likelihood function

$$\mathcal{L}(\theta) = \prod_{i=1}^{N} p(u^{(i)}; \theta) \leftarrow \texttt{Likelihood Function} \tag{9}$$

$$\begin{aligned} \theta_{ML} &= \arg\max_{\theta} \mathcal{L}(\theta) \\ &= \arg\max_{\theta} \sum_{i=1}^{n} \log p(u^{(i)}; \theta) \leftarrow \texttt{Log Likelihood} \end{aligned} \tag{10}$$

- $\log(f(x))$ is monotonic/ increasing, same $\arg\max$ as $f(x)$

Maximum Likelihood for Linear Regression

- Let us assume that the noise is Gaussian distributed with mean 0 and variance σ^2
$$y^{(i)} = h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) + \epsilon^{(i)} = \boldsymbol{\theta}^T\mathbf{x}^{(i)} + \epsilon^{(i)}$$
- Noise $\epsilon^{(i)} \sim \mathcal{N}(0, \sigma^2)$ and thus $y^{(i)} \sim \mathcal{N}(\boldsymbol{\theta}^T\mathbf{x}^{(i)}, \sigma^2)$.

Maximum Likelihood for Linear Regression

- Let us assume that the noise is Gaussian distributed with mean 0 and variance σ^2
$$y^{(i)} = h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) + \epsilon^{(i)} = \boldsymbol{\theta}^T\mathbf{x}^{(i)} + \epsilon^{(i)}$$
- Noise $\epsilon^{(i)} \sim \mathcal{N}(0, \sigma^2)$ and thus $y^{(i)} \sim \mathcal{N}(\boldsymbol{\theta}^T\mathbf{x}^{(i)}, \sigma^2)$.
- Let us compute the likelihood.

$$\begin{aligned} p(\mathbf{y}|\mathbf{X};\boldsymbol{\theta},\sigma^2) &= \prod_{i=1}^{N} p(y^{(i)}|\mathbf{x}^{(i)};\boldsymbol{\theta},\sigma^2) \\ &= \prod_{i=1}^{N} (2\pi\sigma^2)^{-\frac{1}{2}}\, e^{-\frac{1}{2\sigma^2}\left(y^{(i)}-\boldsymbol{\theta}^T\mathbf{x}^{(i)}\right)^2} \\ &= (2\pi\sigma^2)^{-\frac{N}{2}}\, e^{-\frac{1}{2\sigma^2}\sum\limits_{i=1}^{N}\left(y^{(i)}-\boldsymbol{\theta}^T\mathbf{x}^{(i)}\right)^2} \\ &= (2\pi\sigma^2)^{-\frac{N}{2}}\, e^{-\frac{1}{2\sigma^2}\left(\mathbf{y}-\mathbf{X}\boldsymbol{\theta}\right)^T\left(\mathbf{y}-\mathbf{X}\boldsymbol{\theta}\right)} \end{aligned} \tag{11}$$

Maximum Likelihood for Linear Regression

- So we have got the likelihood as,

$$p(\mathbf{y}|\mathbf{X};\boldsymbol{\theta},\sigma^2) = (2\pi\sigma^2)^{-\frac{N}{2}} e^{-\frac{1}{2\sigma^2}\left(\mathbf{y}-\mathbf{X}\boldsymbol{\theta}\right)^T\left(\mathbf{y}-\mathbf{X}\boldsymbol{\theta}\right)}$$

- The log likelihood is

$$l(\boldsymbol{\theta},\sigma^2) = -\frac{N}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\left(\mathbf{y}-\mathbf{X}\boldsymbol{\theta}\right)^T\left(\mathbf{y}-\mathbf{X}\boldsymbol{\theta}\right)$$

Maximum Likelihood for Linear Regression

§ So we have got the likelihood as,

$$p(\mathbf{y}|\mathbf{X};\boldsymbol{\theta},\sigma^2) = (2\pi\sigma^2)^{-\frac{N}{2}}\, e^{-\frac{1}{2\sigma^2}\left(\mathbf{y}-\mathbf{X}\boldsymbol{\theta}\right)^T\left(\mathbf{y}-\mathbf{X}\boldsymbol{\theta}\right)}$$

§ The log likelihood is

$$l(\boldsymbol{\theta},\sigma^2) = -\frac{N}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\left(\mathbf{y}-\mathbf{X}\boldsymbol{\theta}\right)^T\left(\mathbf{y}-\mathbf{X}\boldsymbol{\theta}\right)$$

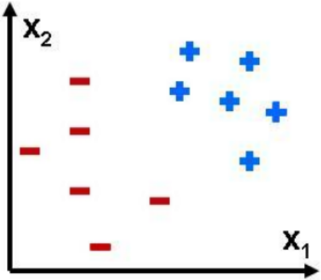
§ Maximizing the likelihood w.r.t. $\boldsymbol{\theta}$ means maximizing $-\left(\mathbf{y}-\mathbf{X}\boldsymbol{\theta}\right)^T\left(\mathbf{y}-\mathbf{X}\boldsymbol{\theta}\right)$ which in turn means minimizing $\left(\mathbf{y}-\mathbf{X}\boldsymbol{\theta}\right)^T\left(\mathbf{y}-\mathbf{X}\boldsymbol{\theta}\right)$.

§ Note the similarity with what we did earlier.

§ Thus linear regression can be equivalently viewed as minimizing error sum of squares as well as maximum likelihood estimation under zero mean Gaussian noise assumption.

Classification

- $y \in \{0, 1\}$, where 0 : "Negative class" (*e.g.*, benign tumor), 1 : "Positive class" (*e.g.*, malignant tumor)
- Some more examples:
  - Email: Spam/ Not Spam?
  - Video: Viral/Not Viral?
  - Tremor: Earthquake/Nuclear explosion?



Linear classifiers with hard threshold

- Linear functions can be used to do classification as well as regression.
- For example,

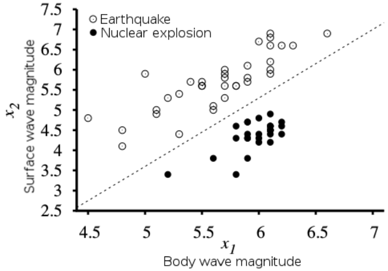


Figure credit: *AIMA: Russell, Norvig*

- A **decision boundary** is a line (or a surface, in higher dimensions) that separates the two classes.
- A linear function gives rise to a **linear separator** and the data that admit such a separator are called **linearly separable**.

Linear Classifier with Hard Threshold

§ The linear separator in the associated fig is given by,

$$x_2 = 1.7x_1 - 4.9$$
$$\implies -4.9 + 1.7x_1 - x_2 = 0$$
$$\implies [-4.9, 1.7, -1.0] \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} = 0$$
$$\boldsymbol{\theta}^T \mathbf{x} = 0$$

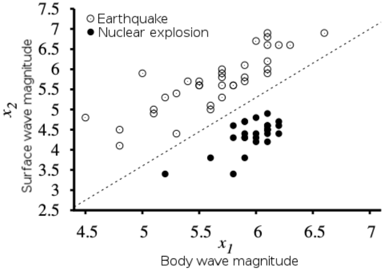


Figure credit: *AIMA: Russell, Norvig*

Linear Classifier with Hard Threshold

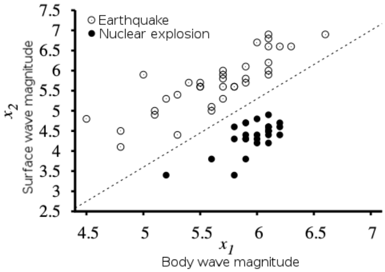


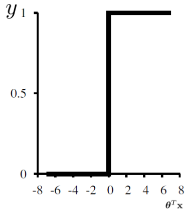
Figure credit: *AIMA: Russell, Norvig*

- The explosions ($y = 1$) are to the right of this line with higher values of x_1 and lower values of x_2. So, they are points for which $\boldsymbol{\theta}^T\mathbf{x} \geq 0$
- Similarly earthquakes ($y = 0$) are to the left of this line. So, they are points for which $\boldsymbol{\theta}^T\mathbf{x} < 0$
- The classification rule is then,

$$y(\mathbf{x}) = \begin{cases} 1 & \text{if } \boldsymbol{\theta}^T\mathbf{x} \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

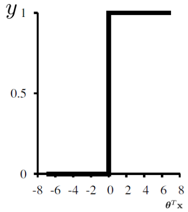
Linear classifiers with hard threshold

§ Alternatively, we can think y as the result of passing the linear function $\boldsymbol{\theta}^T\mathbf{x}$ through a threshold function.



Linear classifiers with hard threshold

- Alternatively, we can think y as the result of passing the linear function $\boldsymbol{\theta}^T\mathbf{x}$ through a threshold function.



- To get the linear separator we have find the θ which minimizes classification error on the training set.
- For regression problems, we found θ in both closed form and by gradient descent. But both approaches required us to compute the gradient.
- This is not possible for the above threshold function as the gradient is undefined when the *value* at $x - axis = 0$ and 0 elsewhere.

Linear classifiers with hard threshold

- Perceptron Rule - This algorithm doesnot compute the gradient to find θ.

Linear classifiers with hard threshold

- Perceptron Rule - This algorithm doesnot compute the gradient to find θ.
- Perceptron Learning Rule can find a linear separator given the data is linearly separable.
- For data that are not linearly separable, the Perceptron algorithm fails.

Linear classifiers with hard threshold

- Perceptron Rule - This algorithm doesnot compute the gradient to find θ.
- Perceptron Learning Rule can find a linear separator given the data is linearly separable.
- For data that are not linearly separable, the Perceptron algorithm fails.
- So, we need to go for a gradient based optimization approach
- Thus, we need to approximate hard threshold function with something smooth.

$$\sigma(u) = \frac{1}{1+e^{-u}}$$
$$y = \sigma(h_\theta(x)) = \sigma(\boldsymbol{\theta}^T \mathbf{x})$$

- Notice that the output is a number between 0 and 1, so it can be interpreted as a probability value belonging to Class 1.
- This is called a logistic regression classifier. The gradient computation is tedious but straight forward.

Maximum Likelihood Estimation of Logistic Regression

- Mathematically, the probability that an example belongs to class 1 is $P(y^{(i)} = 1|\mathbf{x}^{(i)}; \boldsymbol{\theta}) = \sigma\big(\boldsymbol{\theta}^T\mathbf{x}^{(i)}\big)$
- Similarly, $P(y^{(i)} = 0|\mathbf{x}^{(i)}; \boldsymbol{\theta}) = 1 - \sigma\big(\boldsymbol{\theta}^T\mathbf{x}^{(i)}\big)$
- Thus, $P(y^{(i)}|\mathbf{x}^{(i)}; \boldsymbol{\theta}) = \Big(\sigma\big(\boldsymbol{\theta}^T\mathbf{x}^{(i)}\big)\Big)^{y^{(i)}}\Big(1 - \sigma\big(\boldsymbol{\theta}^T\mathbf{x}^{(i)}\big)\Big)^{(1-y^{(i)})}$

Maximum Likelihood Estimation of Logistic Regression

- Mathematically, the probability that an example belongs to class 1 is $P(y^{(i)} = 1|\mathbf{x}^{(i)}; \boldsymbol{\theta}) = \sigma\big(\boldsymbol{\theta}^T\mathbf{x}^{(i)}\big)$
- Similarly, $P(y^{(i)} = 0|\mathbf{x}^{(i)}; \boldsymbol{\theta}) = 1 - \sigma\big(\boldsymbol{\theta}^T\mathbf{x}^{(i)}\big)$
- Thus, $P(y^{(i)}|\mathbf{x}^{(i)}; \boldsymbol{\theta}) = \Big(\sigma\big(\boldsymbol{\theta}^T\mathbf{x}^{(i)}\big)\Big)^{y^{(i)}} \Big(1 - \sigma\big(\boldsymbol{\theta}^T\mathbf{x}^{(i)}\big)\Big)^{(1-y^{(i)})}$
- The joint probability of all the labels $\prod_{i=1}^{N} \Big(\sigma\big(\boldsymbol{\theta}^T\mathbf{x}^{(i)}\big)\Big)^{y^{(i)}} \Big(1 - \sigma\big(\boldsymbol{\theta}^T\mathbf{x}^{(i)}\big)\Big)^{(1-y^{(i)})}$

Maximum Likelihood Estimation of Logistic Regression

- Mathematically, the probability that an example belongs to class 1 is $P(y^{(i)} = 1|\mathbf{x}^{(i)}; \boldsymbol{\theta}) = \sigma\big(\boldsymbol{\theta}^T\mathbf{x}^{(i)}\big)$
- Similarly, $P(y^{(i)} = 0|\mathbf{x}^{(i)}; \boldsymbol{\theta}) = 1 - \sigma\big(\boldsymbol{\theta}^T\mathbf{x}^{(i)}\big)$
- Thus, $P(y^{(i)}|\mathbf{x}^{(i)}; \boldsymbol{\theta}) = \Big(\sigma\big(\boldsymbol{\theta}^T\mathbf{x}^{(i)}\big)\Big)^{y^{(i)}} \Big(1 - \sigma\big(\boldsymbol{\theta}^T\mathbf{x}^{(i)}\big)\Big)^{(1-y^{(i)})}$
- The joint probability of all the labels $\prod_{i=1}^{N} \Big(\sigma\big(\boldsymbol{\theta}^T\mathbf{x}^{(i)}\big)\Big)^{y^{(i)}} \Big(1 - \sigma\big(\boldsymbol{\theta}^T\mathbf{x}^{(i)}\big)\Big)^{(1-y^{(i)})}$
- So the log likelihood for logistic regression is given by,

$$l(\boldsymbol{\theta}) = \sum_{i=1}^{N} y^{(i)} \log \sigma\big(\boldsymbol{\theta}^T\mathbf{x}^{(i)}\big) + (1 - y^{(i)}) \log \big(1 - \sigma\big(\boldsymbol{\theta}^T\mathbf{x}^{(i)}\big)\big)$$

Maximum Likelihood Estimation of Logistic Regression

§ Derivative of log likelihood w.r.t. one component of $\boldsymbol{\theta}$,

$$
\begin{aligned}
\frac{\partial l(\boldsymbol{\theta})}{\partial \theta_j} &= \frac{\partial}{\partial \theta_j} \sum_{i=1}^{N} y^{(i)} \log \sigma\left(\boldsymbol{\theta}^T \mathbf{x}^{(i)}\right) + (1 - y^{(i)}) \log \left(1 - \sigma\left(\boldsymbol{\theta}^T \mathbf{x}^{(i)}\right)\right) \\
&= \sum_{i=1}^{N} \left[\frac{y^{(i)}}{\sigma\left(\boldsymbol{\theta}^T \mathbf{x}^{(i)}\right)} - \frac{1 - y^{(i)}}{1 - \sigma\left(\boldsymbol{\theta}^T \mathbf{x}^{(i)}\right)}\right] \frac{\partial}{\partial \theta_j} \sigma\left(\boldsymbol{\theta}^T \mathbf{x}^{(i)}\right) \\
&= \sum_{i=1}^{N} \left[\frac{y^{(i)}}{\sigma\left(\boldsymbol{\theta}^T \mathbf{x}^{(i)}\right)} - \frac{1 - y^{(i)}}{1 - \sigma\left(\boldsymbol{\theta}^T \mathbf{x}^{(i)}\right)}\right] \sigma\left(\boldsymbol{\theta}^T \mathbf{x}^{(i)}\right) \left(1 - \sigma\left(\boldsymbol{\theta}^T \mathbf{x}^{(i)}\right)\right) \mathbf{x}_j^{(i)} \\
&= \sum_{i=1}^{N} \left[\frac{y^{(i)} - \sigma\left(\boldsymbol{\theta}^T \mathbf{x}^{(i)}\right)}{\sigma\left(\boldsymbol{\theta}^T \mathbf{x}^{(i)}\right) \left(1 - \sigma\left(\boldsymbol{\theta}^T \mathbf{x}^{(i)}\right)\right)}\right] \sigma\left(\boldsymbol{\theta}^T \mathbf{x}^{(i)}\right) \left(1 - \sigma\left(\boldsymbol{\theta}^T \mathbf{x}^{(i)}\right)\right) \mathbf{x}_j^{(i)} \\
&= \sum_{i=1}^{N} \left[y^{(i)} - \sigma\left(\boldsymbol{\theta}^T \mathbf{x}^{(i)}\right)\right] \mathbf{x}_j^{(i)} \qquad (12)
\end{aligned}
$$

§ This is used in an iterative gradient ascent loop.