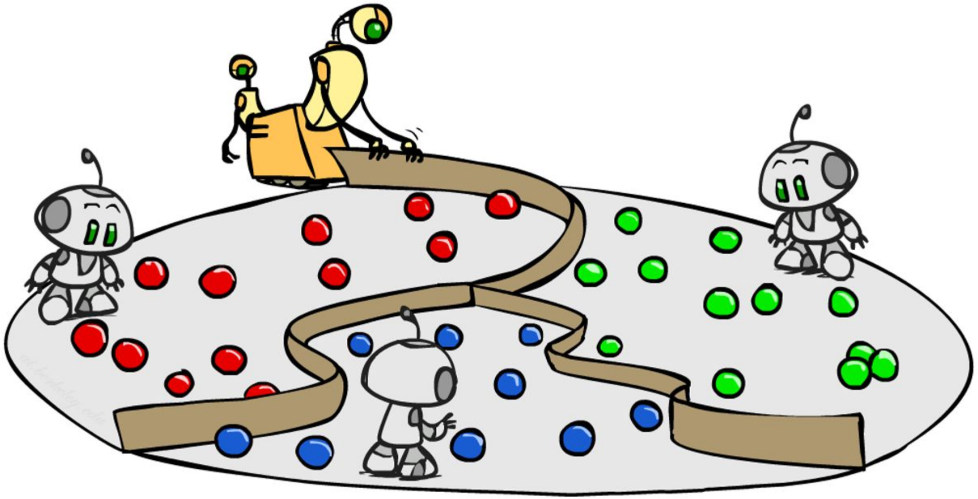
CS-ELEC2C: Machine Learning

Linear Classification and Logistic Regression



What have we learned in the previous lecture?

What have we learned in the previous lecture?

Step-by-step Example

Features

Training Data

Model

Loss

Optimization

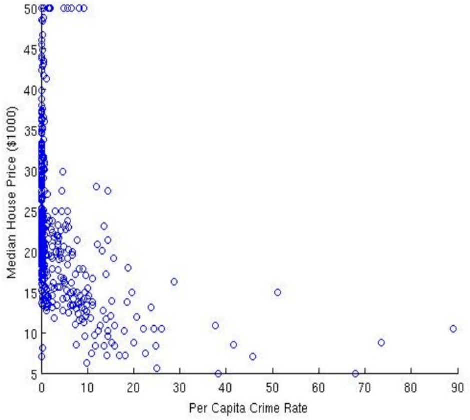
Features: Per Capita Crime Rate and Average Number of Rooms

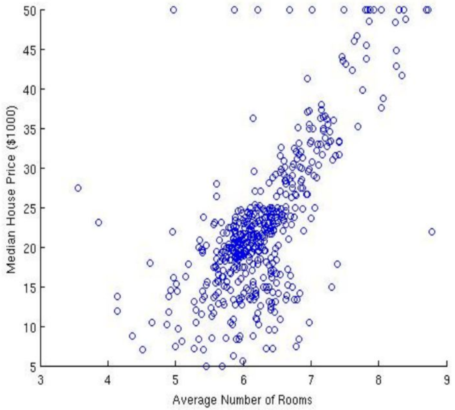
What have we learned in the previous lecture?

Step-by-step Example

- ***Features***
- *Training Data*
- *Model*
- *Loss*
- *Optimization*

Features: **Per Capita Crime Rate** and **Average Number of Rooms**





What have we learned in the previous lecture?

Step-by-step Example

Features

Training Data

Model

Loss

Optimization

Features: **Per Capita Crime Rate** and **Average Number of Rooms**

Training Data Structure

$$D = \{(\mathbf{x}^{(1)}, t^{(1)}), \ldots, (\mathbf{x}^{(n)}, t^{(n)})\}$$

What have we learned in the previous lecture?

Step-by-step Example

Features

Training Data

Model

Loss

Optimization

Features: **Per Capita Crime Rate** and **Average Number of Rooms**

Training Data Structure

$$D = \{(\mathbf{x}^{(1)}, t^{(1)}), \ldots, (\mathbf{x}^{(n)}, t^{(n)})\}$$

such that

What have we learned in the previous lecture?

Step-by-step Example

Features

Training Data

Model

Loss

Optimization

Features: **Per Capita Crime Rate** and **Average Number of Rooms**

Training Data Structure

$$D = \{(\mathbf{x}^{(1)}, t^{(1)}), \ldots, (\mathbf{x}^{(n)}, t^{(n)})\}$$

such that

$$\mathbf{x} = (x_1^{(n)}, x_2^{(n)})$$

What have we learned in the previous lecture?

Step-by-step Example

- *Features*
- ***Training Data***
- *Model*
- *Loss*
- *Optimization*

Features: **Per Capita Crime Rate** and **Average Number of Rooms**

Training Data Structure

$$D = \{(\mathbf{x}^{(1)}, t^{(1)}), \ldots, (\mathbf{x}^{(n)}, t^{(n)})\}$$

such that

$$\mathbf{x} = (x_1^{(n)}, x_2^{(n)})$$

$$x_1 = \text{per capita crime rate}$$

What have we learned in the previous lecture?

Step-by-step Example

Features

Training Data

Model

Loss

Optimization

Features: **Per Capita Crime Rate** and **Average Number of Rooms**

Training Data Structure

$$D = \{(\mathbf{x}^{(1)}, t^{(1)}), \ldots, (\mathbf{x}^{(n)}, t^{(n)})\}$$

such that

$$\mathbf{x} = (x_1^{(n)}, x_2^{(n)})$$

$$x_1 = \text{per capita crime rate}$$
$$x_2 = \text{number of rooms}$$

What have we learned in the previous lecture?

Step-by-step Example

Features

Training Data

Model

Loss

Optimization

Features: **Per Capita Crime Rate** and **Average Number of Rooms**

Training Data Structure

$$D = \{(\mathbf{x}^{(1)}, t^{(1)}), \ldots, (\mathbf{x}^{(n)}, t^{(n)})\}$$

such that

$$\mathbf{x} = (x_1^{(n)}, x_2^{(n)})$$

$$x_1 = \text{per capita crime rate}$$
$$x_2 = \text{number of rooms}$$
$$t = \text{median house price}$$

What have we learned in the previous lecture?

Step-by-step Example

Features

Training Data

Model

Loss

Optimization

Features: **Per Capita Crime Rate** and **Average Number of Rooms**

Training Data Structure

$$D = \{(\mathbf{x}^{(1)}, t^{(1)}), \ldots, (\mathbf{x}^{(n)}, t^{(n)})\}$$

such that

$$\mathbf{x} = (x_1^{(n)}, x_2^{(n)})$$

$$x_1 = \text{per capita crime rate}$$
$$x_2 = \text{number of rooms}$$
$$t = \text{median house price}$$
$$n = \text{number of data points}$$

What have we learned in the previous lecture?

Step-by-step Example

Features

Training Data

Model

Loss

Optimization

Features: **Per Capita Crime Rate** and **Average Number of Rooms**

***Training* Data Structure**

$$D = \{(\mathbf{x}^{(1)}, t^{(1)}), \ldots, (\mathbf{x}^{(n)}, t^{(n)})\}$$

such that

$$\mathbf{x} = (x_1^{(n)}, x_2^{(n)})$$

$$x_1 = \text{per capita crime rate}$$
$$x_2 = \text{number of rooms}$$
$$t = \text{median house price}$$
$$n = \text{number of data points}$$

Sample Data:

Neighborhood #1

$x_1 = 40 \quad x_2 = 3 \quad t = 3,400$

Neighborhood #2

$x_1 = 20 \quad x_2 = 5 \quad t = 4,500$

Neighborhood #3

$x_1 = 30 \quad x_2 = 2 \quad t = 2,800$

What have we learned in the previous lecture?

Step-by-step Example

- *Features*
- *Training Data*
- ***Model***
- *Loss*
- *Optimization*

Features: **Per Capita Crime Rate** and **Average Number of Rooms**

***Training* Data Structure**

$$D = \{(\mathbf{x}^{(1)}, t^{(1)}), \ldots, (\mathbf{x}^{(n)}, t^{(n)})\}$$

Multivariate Linear Regression

$$y(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$$

What have we learned in the previous lecture?

Step-by-step Example

Features

Training Data

Model

Loss

Optimization

Features: **Per Capita Crime Rate** and **Average Number of Rooms**

Training Data Structure

$$D = \{(\mathbf{x}^{(1)}, t^{(1)}), \ldots, (\mathbf{x}^{(n)}, t^{(n)})\}$$

Multivariate Linear Regression

$$y(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$$

Sum of Squares Error

$$\ell(\mathbf{w}) = \sum_{n=1}^{N} [t^{(n)} - (w_0 + w_1 x^{(n)})]^2$$

What have we learned in the previous lecture?

Step-by-step Example

Features

Training Data

Model

Loss

Optimization

Features: **Per Capita Crime Rate** and **Average Number of Rooms**

Training Data Structure

$$D = \{(\mathbf{x}^{(1)}, t^{(1)}), \ldots, (\mathbf{x}^{(n)}, t^{(n)})\}$$

Multivariate Linear Regression

$$y(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$$

Sum of Squares Error

$$\ell(\mathbf{w}) = \sum_{n=1}^{N} [t^{(n)} - (w_0 + w_1 x^{(n)})]^2$$

Optimization using Gradient Descent

$$\mathbf{w} \leftarrow \mathbf{w} + 2\lambda \sum_{n=1}^{N} (t^{(n)} - y(x^{(n)}))x^{(n)}$$

What have we learned in the previous lecture?

Step-by-step Example

Features

Training Data

Model

Loss

Optimization

Multivariate Linear Regression $y(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$

Sample Data:

Neighborhood #1

$x_1 = 40 \quad x_2 = 3 \quad t = 3,400$

Neighborhood #2

$x_1 = 20 \quad x_2 = 5 \quad t = 4,500$

Neighborhood #3

$x_1 = 30 \quad x_2 = 2 \quad t = 2,800$

What have we learned in the previous lecture?

Step-by-step Example

Features

Training Data

Model

Loss

Optimization

Multivariate Linear Regression $y(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$

Gradient Descent

Sample Data:

Neighborhood #1

$x_1 = 40 \quad x_2 = 3 \quad t = 3,400$

Neighborhood #2

$x_1 = 20 \quad x_2 = 5 \quad t = 4,500$

Neighborhood #3

$x_1 = 30 \quad x_2 = 2 \quad t = 2,800$

What have we learned in the previous lecture?

Step-by-step Example

Features

Training Data

Model

Loss

Optimization

Multivariate Linear Regression $y(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$

Gradient Descent

Step #1: Initialize Weights

Sample Data:

Neighborhood #1

$x_1 = 40 \quad x_2 = 3 \quad t = 3,400$

Neighborhood #2

$x_1 = 20 \quad x_2 = 5 \quad t = 4,500$

Neighborhood #3

$x_1 = 30 \quad x_2 = 2 \quad t = 2,800$

What have we learned in the previous lecture?

Step-by-step Example

Features

Training Data

Model

Loss

Optimization

Multivariate Linear Regression $y(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$

Gradient Descent

Step #1: Initialize Weights

Sample Data:

Neighborhood #1

$x_1 = 40 \quad x_2 = 3 \quad t = 3,400$

Neighborhood #2

$x_1 = 20 \quad x_2 = 5 \quad t = 4,500$

Neighborhood #3

$x_1 = 30 \quad x_2 = 2 \quad t = 2,800$

What have we learned in the previous lecture?

Step-by-step Example

Features

Training Data

Model

Loss

Optimization

Multivariate Linear Regression $y(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$

Gradient Descent

Step #1: Initialize Weights

$$y(\mathbf{x}) = 200 + 5(x_1) + 200\,(x_2)$$

Sample Data:

Neighborhood #1

$x_1 = 40 \quad x_2 = 3 \quad t = 3,400$

Neighborhood #2

$x_1 = 20 \quad x_2 = 5 \quad t = 4,500$

Neighborhood #3

$x_1 = 30 \quad x_2 = 2 \quad t = 2,800$

What have we learned in the previous lecture?

Step-by-step Example

Features

Training Data

Model

Loss

Optimization

Multivariate Linear Regression $y(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$

Gradient Descent

Step #1: Initialize Weights

$$y(\mathbf{x}) = 200 + 5(x_1) + 200\,(x_2)$$

Step #2: Update Weights

$$\mathbf{w} \leftarrow \mathbf{w} - \lambda \frac{\partial \ell}{\partial \mathbf{w}} \qquad \mathbf{w} \leftarrow \mathbf{w} + 2\lambda(t^{(n)} - y(x^{(n)}))x^{(n)}$$

Sample Data:

Neighborhood #1

$x_1 = 40 \quad x_2 = 3 \quad t = 3,400$

Neighborhood #2

$x_1 = 20 \quad x_2 = 5 \quad t = 4,500$

Neighborhood #3

$x_1 = 30 \quad x_2 = 2 \quad t = 2,800$

What have we learned in the previous lecture?

Step-by-step Example

Features

Training Data

Model

Loss

Optimization

Multivariate Linear Regression $y(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$

Gradient Descent

Step #1: Initialize Weights

$$y(\mathbf{x}) = 200 + 5(x_1) + 200\,(x_2)$$

Step #2: Update Weights

$$\mathbf{w} \leftarrow \mathbf{w} - \lambda \frac{\partial \ell}{\partial \mathbf{w}} \qquad \mathbf{w} \leftarrow \mathbf{w} + 2\lambda(t^{(n)} - y(x^{(n)}))x^{(n)}$$

Neighborhood #1

Sample Data:

Neighborhood #1

$x_1 = 40 \quad x_2 = 3 \quad t = 3,400$

Neighborhood #2

$x_1 = 20 \quad x_2 = 5 \quad t = 4,500$

Neighborhood #3

$x_1 = 30 \quad x_2 = 2 \quad t = 2,800$

What have we learned in the previous lecture?

Step-by-step Example

Features

Training Data

Model

Loss

Optimization

Multivariate Linear Regression $y(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$

Gradient Descent

Step #1: Initialize Weights

$$y(\mathbf{x}) = 200 + 5(x_1) + 200\,(x_2)$$

Step #2: Update Weights

$$\mathbf{w} \leftarrow \mathbf{w} - \lambda \frac{\partial \ell}{\partial \mathbf{w}} \qquad \mathbf{w} \leftarrow \mathbf{w} + 2\lambda(t^{(n)} - y(x^{(n)}))x^{(n)}$$

Neighborhood #1

$$w_0 = 200 + 2(1)(3400 - (200 + 5(40) + 200(3)))1$$

Sample Data:

Neighborhood #1

$x_1 = 40 \quad x_2 = 3 \quad t = 3,400$

Neighborhood #2

$x_1 = 20 \quad x_2 = 5 \quad t = 4,500$

Neighborhood #3

$x_1 = 30 \quad x_2 = 2 \quad t = 2,800$

What have we learned in the previous lecture?

Step-by-step Example

Features

Training Data

Model

Loss

Optimization

Multivariate Linear Regression $y(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$

Gradient Descent

Step #1: Initialize Weights

$$y(\mathbf{x}) = 200 + 5(x_1) + 200\,(x_2)$$

Step #2: Update Weights

$$\mathbf{w} \leftarrow \mathbf{w} - \lambda \frac{\partial \ell}{\partial \mathbf{w}} \qquad \mathbf{w} \leftarrow \mathbf{w} + 2\lambda(t^{(n)} - y(x^{(n)}))x^{(n)}$$

Neighborhood #1

$$w_0 = 200 + 2(1)(3400 - (200 + 5(40) + 200(3)))1$$

$$w_1 = 5 + 2(1)(3400 - (200 + 5(40) + 200(3)))40$$

Sample Data:

Neighborhood #1

$x_1 = 40 \quad x_2 = 3 \quad t = 3,400$

Neighborhood #2

$x_1 = 20 \quad x_2 = 5 \quad t = 4,500$

Neighborhood #3

$x_1 = 30 \quad x_2 = 2 \quad t = 2,800$

What have we learned in the previous lecture?

Step-by-step Example

Features

Training Data

Model

Loss

Optimization

Multivariate Linear Regression $y(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$

Gradient Descent

Step #1: Initialize Weights

$$y(\mathbf{x}) = 200 + 5(x_1) + 200\,(x_2)$$

Step #2: Update Weights

$$\mathbf{w} \leftarrow \mathbf{w} - \lambda \frac{\partial \ell}{\partial \mathbf{w}} \qquad \mathbf{w} \leftarrow \mathbf{w} + 2\lambda(t^{(n)} - y(x^{(n)}))x^{(n)}$$

Neighborhood #1

$$w_0 = 200 + 2(1)(3400 - (200 + 5(40) + 200(3)))1$$

$$w_1 = 5 + 2(1)(3400 - (200 + 5(40) + 200(3)))40$$

$$w_2 = 200 + 2(1)(3400 - (200 + 5(40) + 200(3)))3$$

Sample Data:

Neighborhood #1

$x_1 = 40 \quad x_2 = 3 \quad t = 3,400$

Neighborhood #2

$x_1 = 20 \quad x_2 = 5 \quad t = 4,500$

Neighborhood #3

$x_1 = 30 \quad x_2 = 2 \quad t = 2,800$

What have we learned in the previous lecture?

Step-by-step Example

Features

Training Data

Model

Loss

Optimization

Multivariate Linear Regression $y(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$

Gradient Descent

Step #1: Initialize Weights

$$y(\mathbf{x}) = 200 + 5(x_1) + 200\,(x_2)$$

Step #2: Update Weights

$$\mathbf{w} \leftarrow \mathbf{w} - \lambda \frac{\partial \ell}{\partial \mathbf{w}} \qquad \mathbf{w} \leftarrow \mathbf{w} + 2\lambda(t^{(n)} - y(x^{(n)}))x^{(n)}$$

Neighborhood #1

$$w_0 = 200 + 2(1)(3400 - (200 + 5(40) + 200(3)))1 \qquad w_0 = 5,000$$

$$w_1 = 5 + 2(1)(3400 - (200 + 5(40) + 200(3)))40 \qquad w_1 = 192,005$$

$$w_2 = 200 + 2(1)(3400 - (200 + 5(40) + 200(3)))3 \qquad w_2 = 15,400$$

Sample Data:

Neighborhood #1

$x_1 = 40 \quad x_2 = 3 \quad t = 3,400$

Neighborhood #2

$x_1 = 20 \quad x_2 = 5 \quad t = 4,500$

Neighborhood #3

$x_1 = 30 \quad x_2 = 2 \quad t = 2,800$

What have we learned in the previous lecture?

Step-by-step Example

Features

Training Data

Model

Loss

Optimization

Multivariate Linear Regression $y(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$

Gradient Descent

Step #1: Initialize Weights

$$y(\mathbf{x}) = 200 + 5(x_1) + 200\,(x_2)$$

Step #2: Update Weights

$$\mathbf{w} \leftarrow \mathbf{w} - \lambda \frac{\partial \ell}{\partial \mathbf{w}} \qquad \mathbf{w} \leftarrow \mathbf{w} + 2\lambda(t^{(n)} - y(x^{(n)}))x^{(n)}$$

Neighborhood #2

$$w_0 = 5000 + 2(1)(4500 - (5000 + 192005(20) + 15400(5)))1$$

Sample Data:

Neighborhood #1

$x_1 = 40 \quad x_2 = 3 \quad t = 3,400$

Neighborhood #2

$x_1 = 20 \quad x_2 = 5 \quad t = 4,500$

Neighborhood #3

$x_1 = 30 \quad x_2 = 2 \quad t = 2,800$

What have we learned in the previous lecture?

Step-by-step Example

Features

Training Data

Model

Loss

Optimization

Multivariate Linear Regression $y(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$

Gradient Descent

Step #1: Initialize Weights

$$y(\mathbf{x}) = 200 + 5(x_1) + 200\,(x_2)$$

Step #2: Update Weights

$$\mathbf{w} \leftarrow \mathbf{w} - \lambda \frac{\partial \ell}{\partial \mathbf{w}} \qquad \mathbf{w} \leftarrow \mathbf{w} + 2\lambda(t^{(n)} - y(x^{(n)}))x^{(n)}$$

Neighborhood #2

$$w_0 = 5000 + 2(1)(4500 - (5000 + 192005(20) + 15400(5)))1$$

$$w_1 = 192005 + 2(1)(4500 - (5000 + 192005(20) + 15400(5)))20$$

Sample Data:

Neighborhood #1

$x_1 = 40 \quad x_2 = 3 \quad t = 3,400$

Neighborhood #2

$x_1 = 20 \quad x_2 = 5 \quad t = 4,500$

Neighborhood #3

$x_1 = 30 \quad x_2 = 2 \quad t = 2,800$

What have we learned in the previous lecture?

Step-by-step Example

Features

Training Data

Model

Loss

Optimization

Multivariate Linear Regression $y(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$

Gradient Descent

Step #1: Initialize Weights

$$y(\mathbf{x}) = 200 + 5(x_1) + 200\,(x_2)$$

Step #2: Update Weights

$$\mathbf{w} \leftarrow \mathbf{w} - \lambda \frac{\partial \ell}{\partial \mathbf{w}} \qquad \mathbf{w} \leftarrow \mathbf{w} + 2\lambda(t^{(n)} - y(x^{(n)}))x^{(n)}$$

Neighborhood #2

$$w_0 = 5000 + 2(1)(4500 - (5000 + 192005(20) + 15400(5)))1$$

$$w_1 = 192005 + 2(1)(4500 - (5000 + 192005(20) + 15400(5)))20$$

$$w_2 = 15400 + 2(1)(4500 - (5000 + 192005(20) + 15400(5)))5$$

Sample Data:

Neighborhood #1

$x_1 = 40 \quad x_2 = 3 \quad t = 3,400$

Neighborhood #2

$x_1 = 20 \quad x_2 = 5 \quad t = 4,500$

Neighborhood #3

$x_1 = 30 \quad x_2 = 2 \quad t = 2,800$

What have we learned in the previous lecture?

Sample Data:

Neighborhood #1

$x_1 = 40 \quad x_2 = 3 \quad t = 3,400$

Neighborhood #2

$x_1 = 20 \quad x_2 = 5 \quad t = 4,500$

Neighborhood #3

$x_1 = 30 \quad x_2 = 2 \quad t = 2,800$

Step-by-step Example

Features

Training Data

Model

Loss

Optimization

Multivariate Linear Regression $y(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2$

Gradient Descent

Step #1: Initialize Weights

$$y(\mathbf{x}) = 200 + 5(x_1) + 200\,(x_2)$$

Step #2: Update Weights

$$\mathbf{w} \leftarrow \mathbf{w} - \lambda \frac{\partial \ell}{\partial \mathbf{w}} \qquad \mathbf{w} \leftarrow \mathbf{w} + 2\lambda(t^{(n)} - y(x^{(n)}))x^{(n)}$$

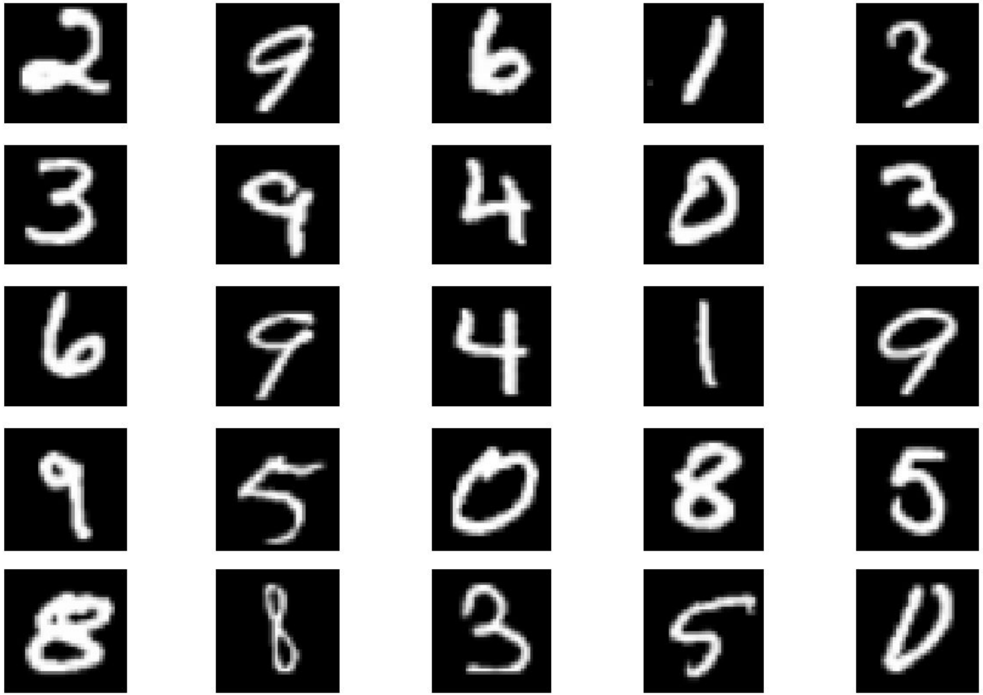
Neighborhood #2

$$w_0 = 5000 + 2(1)(4500 - (5000 + 192005(20) + 15400(5)))1 \qquad w_0 = -7,830,200$$

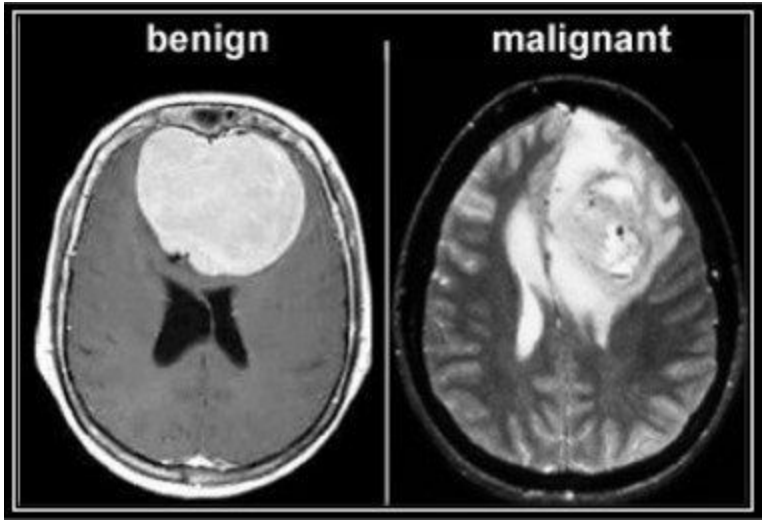
$$w_1 = 192005 + 2(1)(4500 - (5000 + 192005(20) + 15400(5)))20 \qquad w_1 = -156,551,995$$

$$w_2 = 15400 + 2(1)(4500 - (5000 + 192005(20) + 15400(5)))5 \qquad w_2 = -36,160,600$$

What Problem is This?



What Problem is This?



What Problem is This?

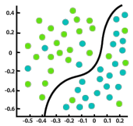
Introduction: Linear Classification

Classification

Introduction: Linear Classification

Classification

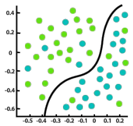
In information science, a classification scheme is ***arranging things according to its kind*** *or into* ***groups or classes***



Introduction: Linear Classification

Classification

In information science, a classification scheme is ***arranging things according to its kind*** *or into* ***groups or classes***

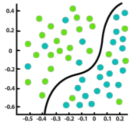


How is it different from Regression?

Introduction: Linear Classification

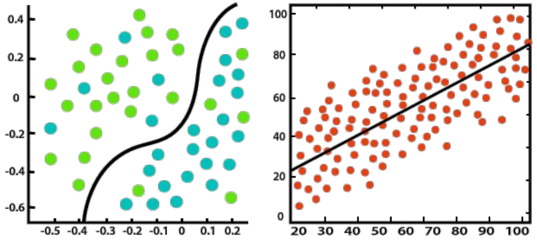
Classification

*In information science, a classification scheme is **arranging things according to its kind** or into **groups or classes***



How is it different from Regression?

In **regression problems**, the targets or outputs are **continuous variables**. However, in **classification problems**, they have **categorical outputs**. Intuitively, classification can be thought of as **assigning each input to one of a finite number of labels**

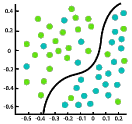


Classification Regression

Introduction: Linear Classification

Classification

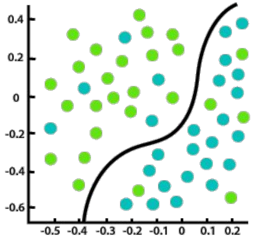
*In information science, a classification scheme is **arranging things according to its kind** or into **groups or classes***



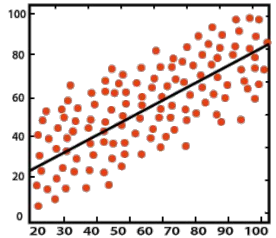
How is it different from Regression?

In **regression problems**, the targets or outputs are **continuous variables**. However, in **classification problems**, they have **categorical outputs**. Intuitively, classification can be thought of as **assigning each input to one of a finite number of labels**

There are two types of classification problems: **binary classification** and **multiclass**



Classification

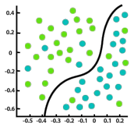


Regression

Linear Classification

Classification

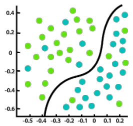
In information science, a classification scheme is ***arranging things according to its kind*** *or into* ***groups or classes***



Linear Classification

Classification

In information science, a classification scheme is ***arranging things according to its kind*** *or into* ***groups or classes***

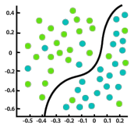


Can we frame classification as a regression problem?

Linear Classification

Classification

*In information science, a classification scheme is **arranging things according to its kind** or into **groups or classes***



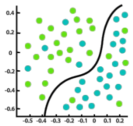
Can we frame classification as a regression problem?

Yes! Our simple hack is ***ignore that the output is categorical***

Linear Classification

Classification

*In information science, a classification scheme is **arranging things according to its kind** or into **groups or classes***



Can we frame classification as a regression problem?

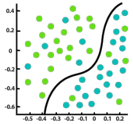
Yes! Our simple hack is ***ignore that the output is categorical***

Suppose we have a binary classification problem…

Linear Classification

Classification

In information science, a classification scheme is ***arranging things according to its kind*** *or into* ***groups or classes***



Can we frame classification as a regression problem?

Yes! Our simple hack is ***ignore that the output is categorical***

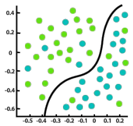
Suppose we have a binary classification problem…

Assume the standard model used for regression $y = f(\mathbf{x}, \mathbf{w}) = \mathbf{x}^T \mathbf{w} \quad t \in \{-1, 1\}$

Linear Classification

Classification

*In information science, a classification scheme is **arranging things according to its kind** or into **groups or classes***



Can we frame classification as a regression problem?

Yes! Our simple hack is ***ignore that the output is categorical***

Suppose we have a binary classification problem…

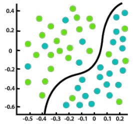
Assume the standard model used for regression $y = f(\mathbf{x}, \mathbf{w}) = \mathbf{x}^T\mathbf{w} \quad t \in \{-1, 1\}$

How do we obtain the weights?

Linear Classification

Classification

*In information science, a classification scheme is **arranging things according to its kind** or into **groups or classes***



Can we frame classification as a regression problem?

Yes! Our simple hack is ***ignore that the output is categorical***

Suppose we have a binary classification problem…

Assume the standard model used for regression $y = f(\mathbf{x}, \mathbf{w}) = \mathbf{x}^T\mathbf{w} \quad t \in \{-1, 1\}$

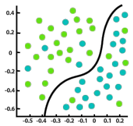
How do we obtain the weights?

What loss are we minimizing?

Linear Classification

Classification

*In information science, a classification scheme is **arranging things according to its kind** or into **groups or classes***



Can we frame classification as a regression problem?

Yes! Our simple hack is ***ignore that the output is categorical***

Suppose we have a binary classification problem…

Assume the standard model used for regression $y = f(\mathbf{x}, \mathbf{w}) = \mathbf{x}^T \mathbf{w} \quad t \in \{-1, 1\}$

How do we obtain the weights?

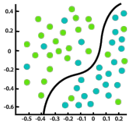
Would this make sense?

What loss are we minimizing?

Linear Classification

Classification

In information science, a classification scheme is ***arranging things according to its kind*** *or into* ***groups or classes***



Can we frame classification as a regression problem?

Yes! Our simple hack is ***ignore that the output is categorical***

Suppose we have a binary classification problem…

Assume the standard model used for regression $y = f(\mathbf{x}, \mathbf{w}) = \mathbf{x}^T\mathbf{w} \quad t \in \{-1, 1\}$

How do we obtain the weights? Would this make sense?

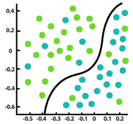
What loss are we minimizing?

$$\ell_{\text{square}}(\mathbf{w}, t) = \frac{1}{N}\sum_{n=1}^{N}(t^{(n)} - \mathbf{w}^T\mathbf{x})^2$$

Linear Classification

Linear Classification

A linear classifier makes a classification decision ***based on the value of a linear combination of the characteristics****.*



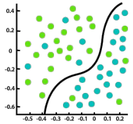
As an example…

Assume that the classifier has the following form $f(\mathbf{x}, \mathbf{w}) = w_0 + \mathbf{w}^T\mathbf{x}$

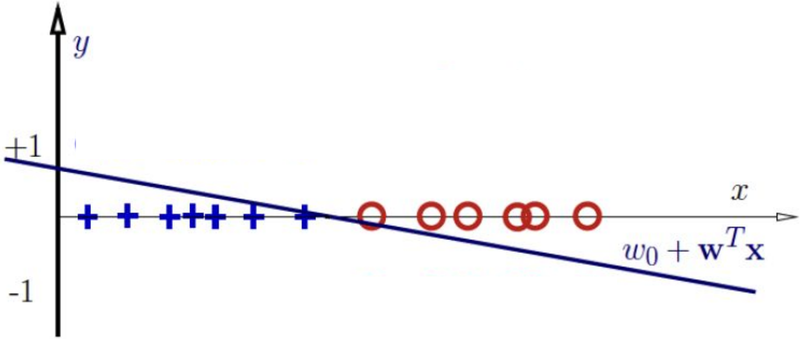
Linear Classification

Linear Classification

A linear classifier makes a classification decision ***based on the value of a linear combination of the characteristics***.



As an example…

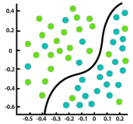


Assume that the classifier has the following form $f(\mathbf{x}, \mathbf{w}) = w_0 + \mathbf{w}^T\mathbf{x}$

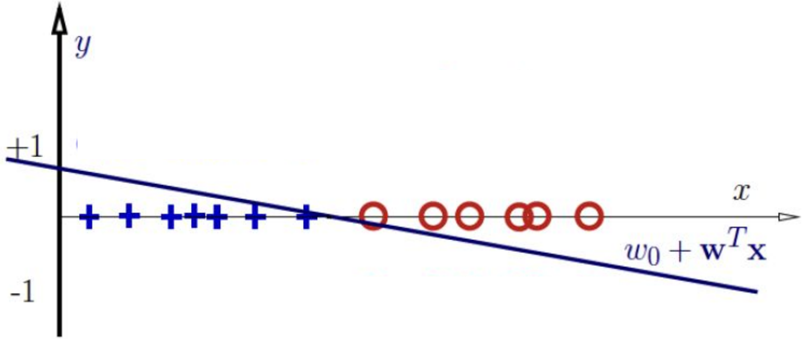
Linear Classification

Linear Classification

*A linear classifier makes a classification decision **based on the value of a linear combination of the characteristics**.*



As an example…



Assume that the classifier has the following form

$$f(\mathbf{x}, \mathbf{w}) = w_0 + \mathbf{w}^T\mathbf{x}$$

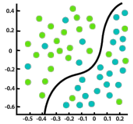
A reasonable **decision rule** would be

$$y = \begin{cases} 1, & \text{if } f(\mathbf{x}, \mathbf{w}) \geq 0 \\ -1, & \text{otherwise} \end{cases}$$

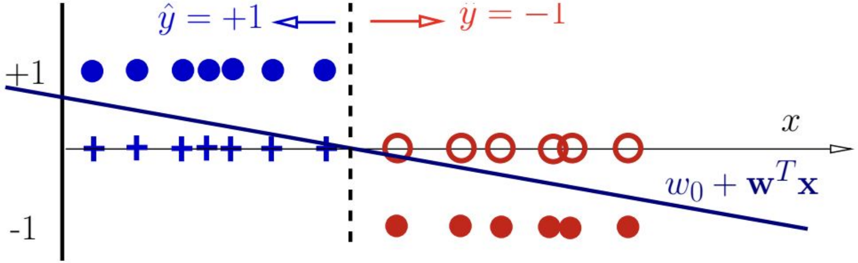
Linear Classification

Linear Classification

A linear classifier makes a classification decision ***based on the value of a linear combination of the characteristics***.



As an example…



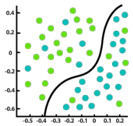
How do we mathematically represent that rule?

$$y = \mathrm{sign}(w_0 + \mathbf{w}^T\mathbf{x})$$

Linear Classification

Linear Classification

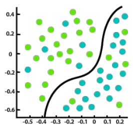
A linear classifier makes a classification decision ***based on the value of a linear combination of the characteristics****.*

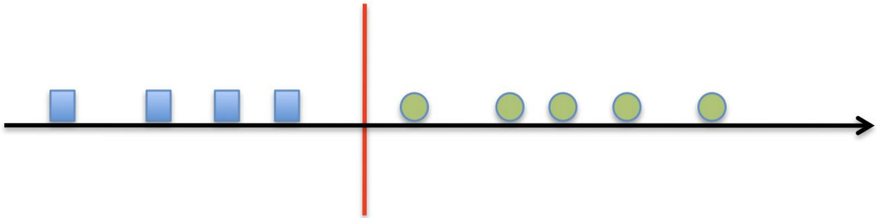


Linear Classification

Linear Classification

A linear classifier makes a classification decision ***based on the value of a linear combination of the characteristics.***

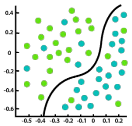


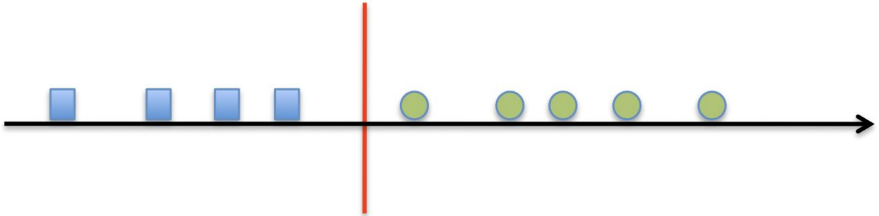


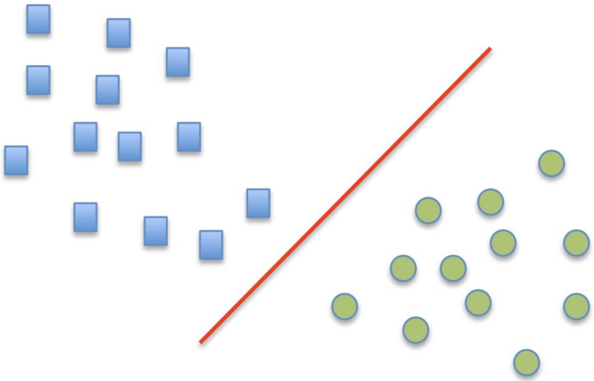
Linear Classification

Linear Classification

A linear classifier makes a classification decision ***based on the value of a linear combination of the characteristics****.*



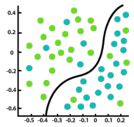


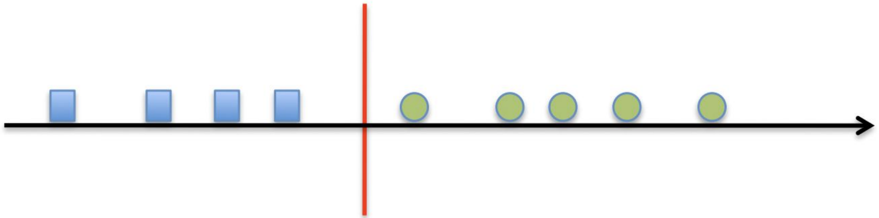


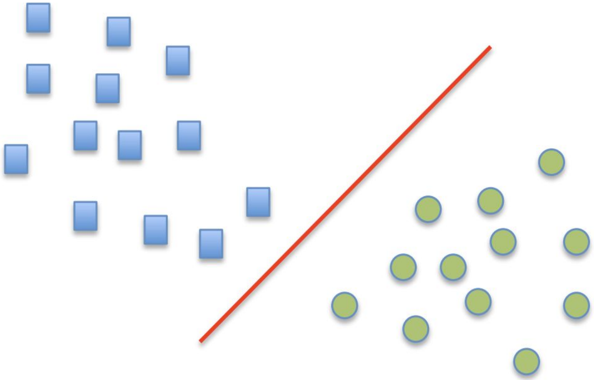
Linear Classification

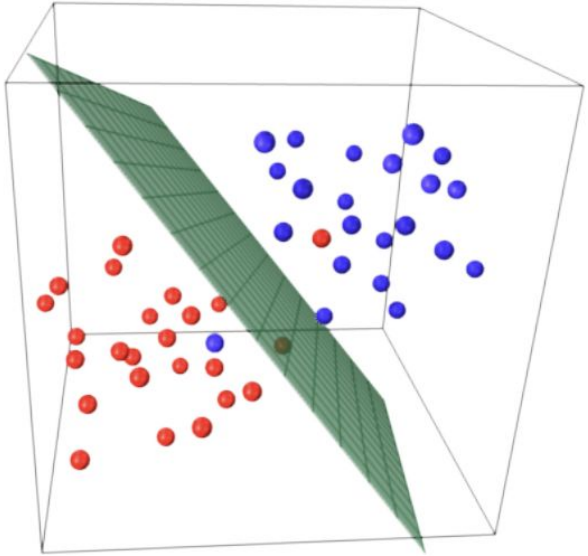
Linear Classification

A linear classifier makes a classification decision ***based on the value of a*** ***linear combination*** ***of the characteristics***.





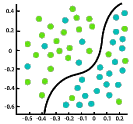




Linear Classification

Linear Classification

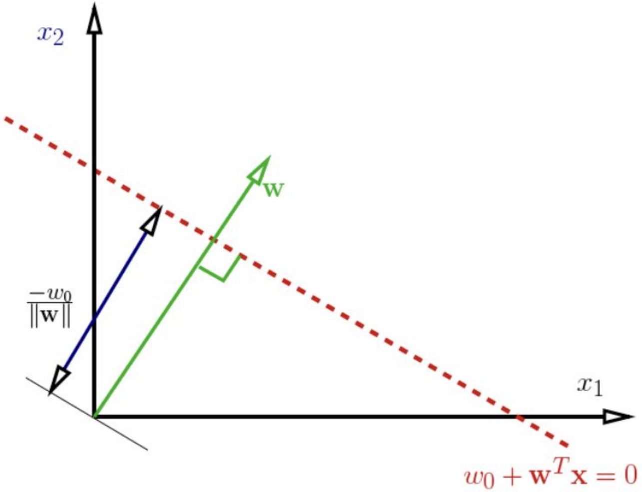
A linear classifier makes a classification decision ***based on the value of a linear combination of the characteristics.***



What does this represent geometrically?

$\mathbf{w}^T\mathbf{x} = 0$ is a line passing through the origin and is orthogonal to $\mathbf{w}$

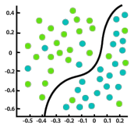
$\mathbf{w}^T\mathbf{x} + w_0 = 0$ shifts the line by w_0



Linear Classification

Linear Classification

A linear classifier makes a classification decision ***based on the value of a linear combination of the characteristics****.*

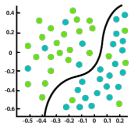


What is the algorithm supposed to learn?

Linear Classification

Linear Classification

A linear classifier makes a classification decision ***based on the value of a linear combination of the characteristics****.*



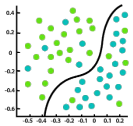
What is the algorithm supposed to learn?

The objective in classification is to **learn "good" decision boundaries**. There is a need to **find the direction** (weights or parameters) and the **location** (bias) of the boundary

Linear Classification

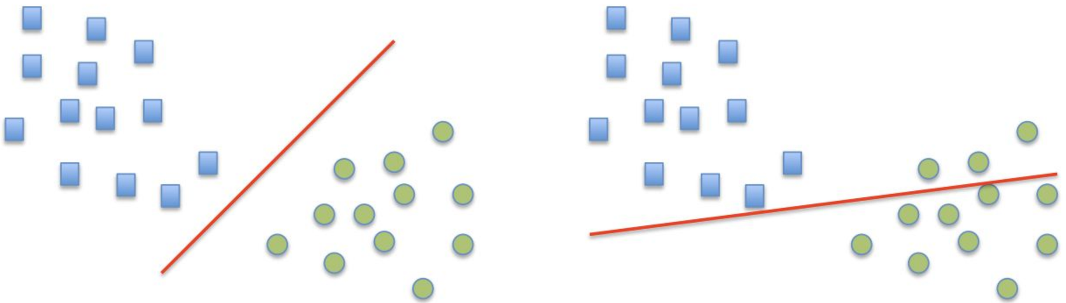
Linear Classification

A linear classifier makes a classification decision ***based on the value of a linear combination of the characteristics****.*



What is the algorithm supposed to learn?

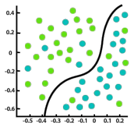
The objective in classification is to **learn “good” decision boundaries**. There is a need to **find the direction** (weights or parameters) and the **location** (bias) of the boundary



Linear Classification

Linear Classification

A linear classifier makes a classification decision ***based on the value of a*** ***linear combination*** ***of the characteristics****.*

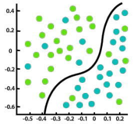


What loss functions can we use?

Linear Classification

Linear Classification

A linear classifier makes a classification decision ***based on the value of a linear combination of the characteristics***.



What loss functions can we use?

Zero/one loss for a classifier

$$L_{0-1}(y(\mathbf{x}), t) = \begin{cases} 0 & \text{if } y(\mathbf{x}) = t \\ 1 & \text{if } y(\mathbf{x}) \neq t \end{cases}$$

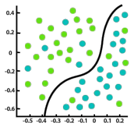
Asymmetric Binary Loss

$$L_{ABL}(y(\mathbf{x}), t) = \begin{cases} \alpha & \text{if } y(\mathbf{x}) = 1 \wedge t = 0 \\ \beta & \text{if } y(\mathbf{x}) = 0 \wedge t = 1 \\ 0 & \text{if } y(\mathbf{x}) = t \end{cases}$$

Linear Classification

Linear Classification

A linear classifier makes a classification decision ***based on the value of a linear combination of the characteristics***.



What loss functions can we use?

Zero/one loss for a classifier

$$L_{0-1}(y(\mathbf{x}), t) = \begin{cases} 0 & \text{if } y(\mathbf{x}) = t \\ 1 & \text{if } y(\mathbf{x}) \neq t \end{cases}$$

Squared (quadratic) loss

$$L_{squared}(y(\mathbf{x}), t) = (t - y(\mathbf{x}))^2$$

Asymmetric Binary Loss

$$L_{ABL}(y(\mathbf{x}), t) = \begin{cases} \alpha & \text{if } y(\mathbf{x}) = 1 \wedge t = 0 \\ \beta & \text{if } y(\mathbf{x}) = 0 \wedge t = 1 \\ 0 & \text{if } y(\mathbf{x}) = t \end{cases}$$

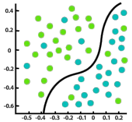
Absolute Error

$$L_{absolute}(y(\mathbf{x}), t) = |t - y(\mathbf{x})|$$

Linear Classification

Linear Classification

A linear classifier makes a classification decision ***based on the value of a*** ***linear combination*** ***of the characteristics****.*

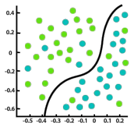


Can classes always be separated?

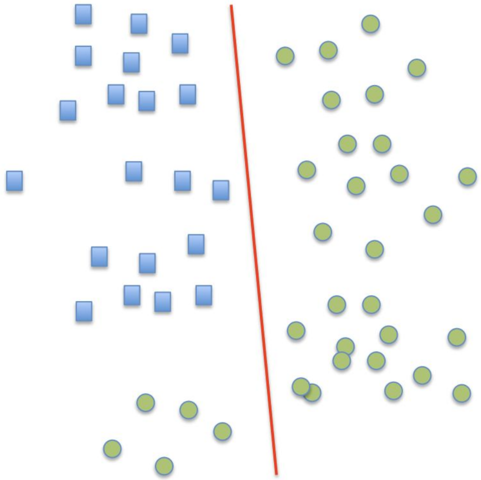
Linear Classification

Linear Classification

A linear classifier makes a classification decision ***based on the value of a*** ***linear combination*** ***of the characteristics****.*



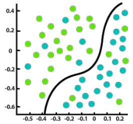
Can classes always be separated?



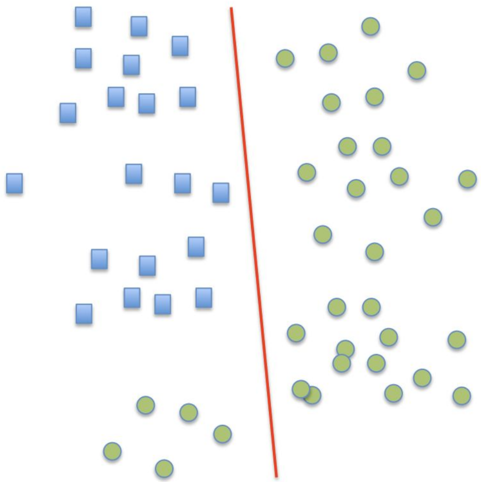
Linear Classification

Linear Classification

A linear classifier makes a classification decision ***based on the value of a linear combination of the characteristics****.*



Can classes always be separated?

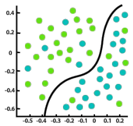


If we can perfectly separate classes, it is a **linearly separable problem**

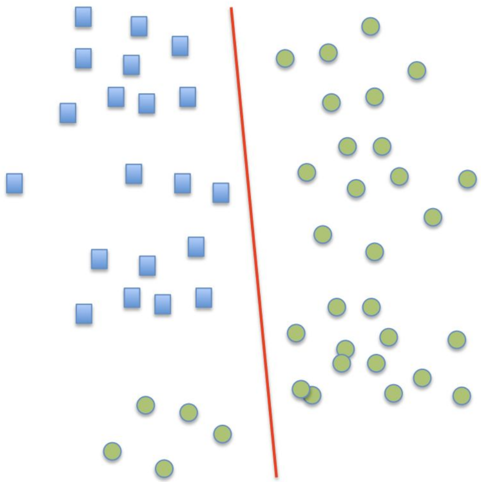
Linear Classification

Linear Classification

A linear classifier makes a classification decision ***based on the value of a linear combination of the characteristics****.*



Can classes always be separated?



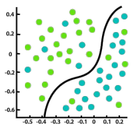
If we can perfectly separate classes, it is a **linearly separable problem**

Causes of Non-Perfect Separation:

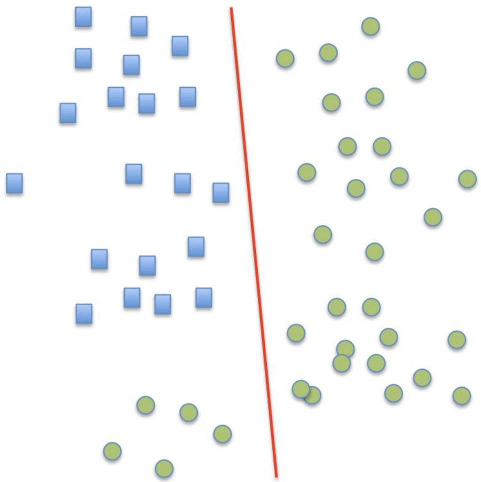
Linear Classification

Linear Classification

A linear classifier makes a classification decision ***based on the value of a linear combination of the characteristics****.*



Can classes always be separated?



If we can perfectly separate classes, it is a **linearly separable problem**

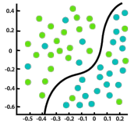
Causes of Non-Perfect Separation:

Model is too simple

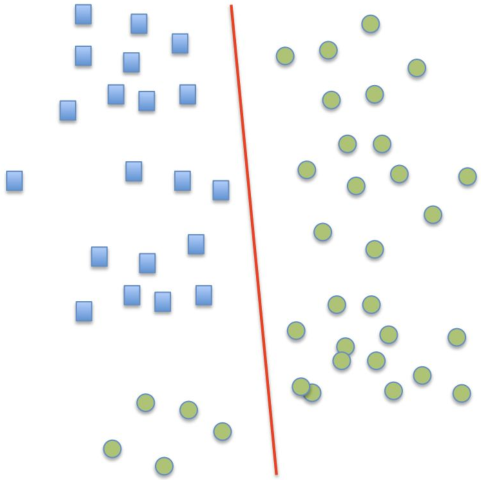
Linear Classification

Linear Classification

A linear classifier makes a classification decision ***based on the value of a linear combination of the characteristics****.*



Can classes always be separated?



If we can perfectly separate classes, it is a **linearly separable problem**

Causes of Non-Perfect Separation:

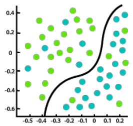
Model is too simple

Noise in the inputs

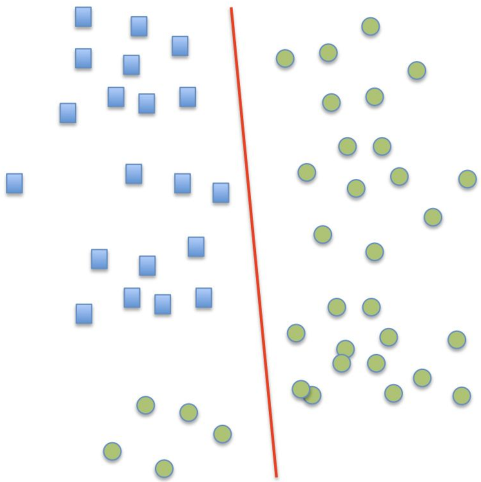
Linear Classification

Linear Classification

*A linear classifier makes a classification decision **based on the value of a linear combination of the characteristics**.*



Can classes always be separated?



If we can perfectly separate classes, it is a **linearly separable problem**

Causes of Non-Perfect Separation:

Model is too simple

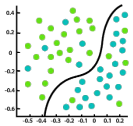
Noise in the inputs

Simple features that do not account for variations

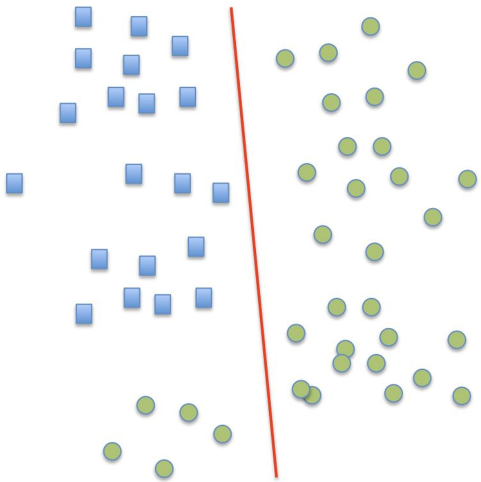
Linear Classification

Linear Classification

A linear classifier makes a classification decision ***based on the value of a linear combination of the characteristics****.*



Can classes always be separated?



If we can perfectly separate classes, it is a **linearly separable problem**

Causes of Non-Perfect Separation:

Model is too simple

Noise in the inputs

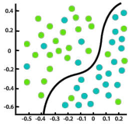
Simple features that do not account for variations

Errors in data targets

Linear Classification

Linear Classification

A linear classifier makes a classification decision ***based on the value of a linear combination of the characteristics****.*

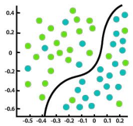


What is the cost of getting things wrong?

Linear Classification

Linear Classification

A linear classifier makes a classification decision ***based on the value of a linear combination of the characteristics****.*



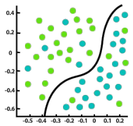
What is the cost of getting things wrong?

For medical diagnosis: For a diabetes screening test is it better to have **false positives** or **false negatives**?

Linear Classification

Linear Classification

A linear classifier makes a classification decision ***based on the value of a linear combination of the characteristics****.*



What is the cost of getting things wrong?

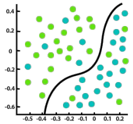
For medical diagnosis: For a diabetes screening test is it better to have **false positives** or **false negatives**?

For movie ratings: The ”truth” is that Alice thinks E.T. is worthy of a 4. How bad is it to predict a 5? How about a 2?

Linear Classification

Linear Classification

A linear classifier makes a classification decision ***based on the value of a linear combination of the characteristics****.*

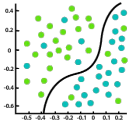


How to Assess Correctness?

Linear Classification

Linear Classification

A linear classifier makes a classification decision ***based on the value of a linear combination of the characteristics.***



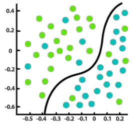
How to Assess Correctness?

		Actual Values	
		Positive	Negative
Predicted Values	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

Linear Classification

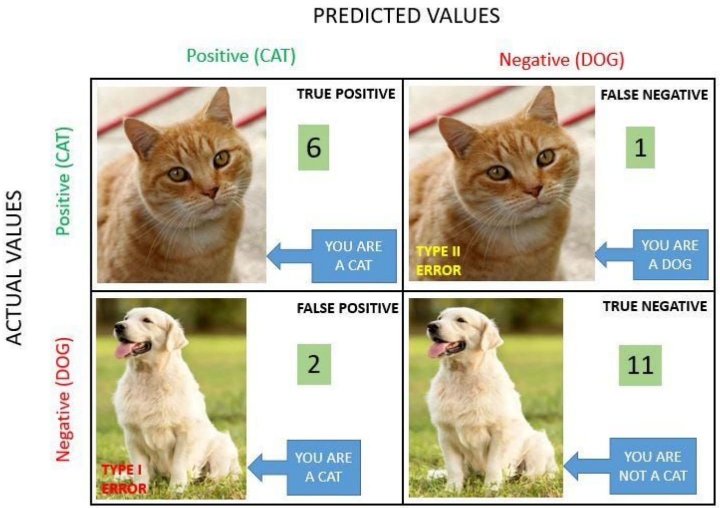
Linear Classification

A linear classifier makes a classification decision ***based on the value of a*** ***linear combination*** ***of the characteristics.***



How to Assess Correctness?

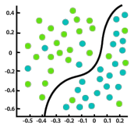
		Actual Values: Positive	Actual Values: Negative
Predicted Values	Positive	True Positive	False Positive
Predicted Values	Negative	False Negative	True Negative



Linear Classification

Linear Classification

A linear classifier makes a classification decision ***based on the value of a linear combination of the characteristics****.*

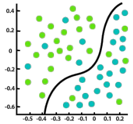


How to Assess Correctness?

Linear Classification

Linear Classification

A linear classifier makes a classification decision ***based on the value of a linear combination of the characteristics.***



How to Assess Correctness?

Recall: is the fraction of relevant instances that are retrieved

$$R = \frac{TP}{TP + FN} = \frac{TP}{\text{all groundtruth instances}}$$

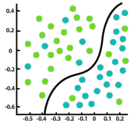
Precision: is the fraction of retrieved instances that are relevant

$$P = \frac{TP}{TP + FP} = \frac{TP}{\text{all predicted}}$$

Linear Classification

Linear Classification

*A linear classifier makes a classification decision **based on the value of a linear combination of the characteristics**.*



How to Assess Correctness?

Recall: is the fraction of relevant instances that are retrieved

$$R = \frac{TP}{TP + FN} = \frac{TP}{\text{all groundtruth instances}}$$

Precision: is the fraction of retrieved instances that are relevant

$$P = \frac{TP}{TP + FP} = \frac{TP}{\text{all predicted}}$$

Accuracy: fraction of correct instances

$$A = \frac{TP+TN}{TP+TN+FP+FN}$$

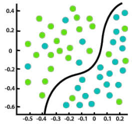
F1 score: harmonic mean of precision and recall

$$F1 = 2\frac{P \cdot R}{P + R}$$

Linear Classification

Linear Classification

A linear classifier makes a classification decision ***based on the value of a*** ***linear combination*** ***of the characteristics****.*

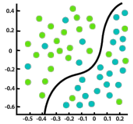


What is the difference between these metrics and loss?

Linear Classification

Linear Classification

A linear classifier makes a classification decision ***based on the value of a linear combination of the characteristics****.*



What is the difference between these metrics and loss?

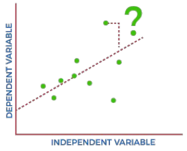
Metrics on the dataset is what we usually care about. This is what is usually referred to as **performance**.

Typically, it is **not possible to directly optimize for these metrics**. The **loss function should reflect the problem we are solving**. We then hope it will yield models that will do well on our dataset

Introduction: Logistic Regression

What is Regression?

***Regression** is a **statistical technique** that **relates a dependent variable** to **one or more independent** variables*

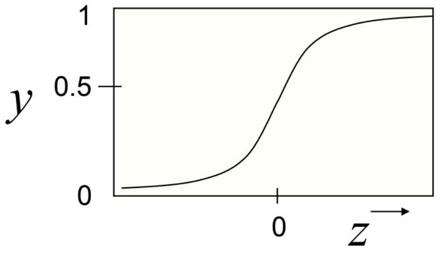


Is there a way to improve Linear Classification?

As an alternative, the *sign()* function can be replaced by the **sigmoid** or **logistic function.** Sigmoid applied to linear function of the data

$$\sigma(z) = \frac{1}{1 + \exp(-z)} \qquad y(\mathbf{x}) = \sigma\left(\mathbf{w}^T\mathbf{x} + w_0\right)$$

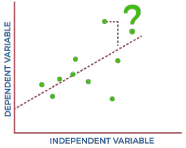
The output is a smooth function of the inputs and the weights. It can be seen as a **smoothed** and **differentiable** alternative to sign()



Introduction: Logistic Regression

What is Regression?

Regression *is a* ***statistical technique*** *that* ***relates a dependent variable*** *to* ***one or more independent*** *variables*

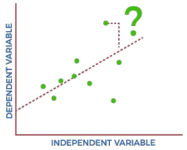


Is there a way to improve Linear Classification?

Introduction: Logistic Regression

What is Regression?

Regression *is a* ***statistical technique*** *that* ***relates a dependent variable*** *to* ***one or more independent*** *variables*



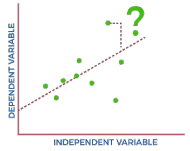
Is there a way to improve Linear Classification?

As an alternative, the *sign()* function can be replaced by the **sigmoid** or **logistic function.** Sigmoid applied to linear function of the data

Introduction: Logistic Regression

What is Regression?

Regression *is a* ***statistical technique*** *that* ***relates a dependent variable*** *to* ***one or more independent*** *variables*



Is there a way to improve Linear Classification?

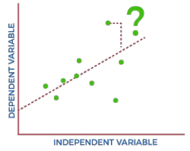
As an alternative, the *sign()* function can be replaced by the **sigmoid** or **logistic function.** Sigmoid applied to linear function of the data

$$\sigma(z) = \frac{1}{1 + \exp(-z)} \qquad y(\mathbf{x}) = \sigma\left(\mathbf{w}^T\mathbf{x} + w_0\right)$$

Introduction: Logistic Regression

What is Regression?

Regression *is a* ***statistical technique*** *that* ***relates a dependent variable*** *to* ***one or more independent*** *variables*

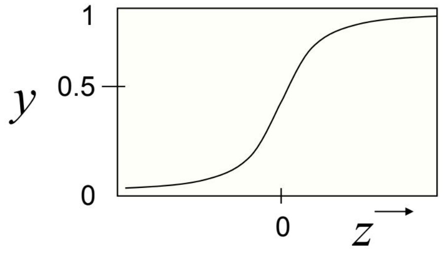


Is there a way to improve Linear Classification?

As an alternative, the *sign()* function can be replaced by the **sigmoid** or **logistic function.** Sigmoid applied to linear function of the data

$$\sigma(z) = \frac{1}{1+\exp(-z)} \qquad y(\mathbf{x}) = \sigma\left(\mathbf{w}^T\mathbf{x} + w_0\right)$$

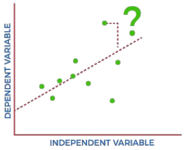
The output is a smooth function of the inputs and the weights. It can be seen as a **smoothed** and **differentiable** alternative to sign()



Introduction: Logistic Regression

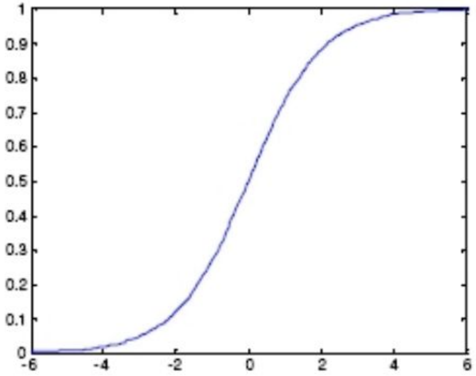
What is Regression?

Regression *is a* ***statistical technique*** *that* ***relates a dependent variable*** *to* ***one or more independent*** *variables*

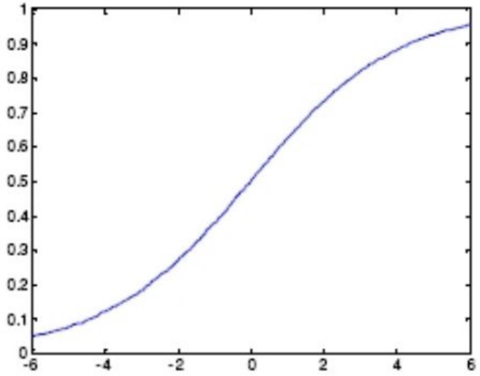


$$y = \sigma\left(w_1 x + w_0\right)$$

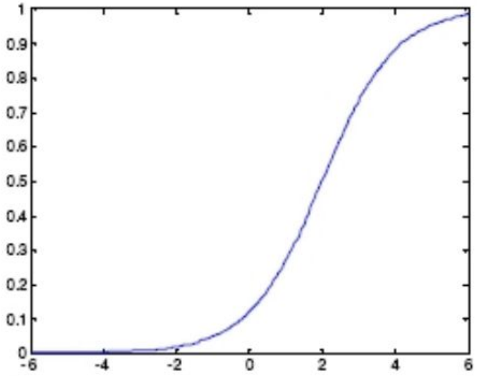
$w_0 = 0, w_1 = 1$



$w_0 = 0, w_1 = 0.5$



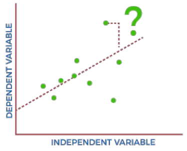
$w_0 = -2, w_1 = 1$



Logistic Regression

Logistic Regression

Models the probability of an event taking place *by having the* ***log-odds for the event be a linear combination*** *of one or more independent variables.*

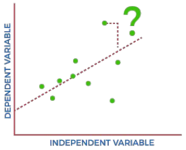


Probabilistic Interpretation…

Logistic Regression

Logistic Regression

Models the probability of an event taking place *by having the* ***log-odds for the event be a linear combination*** *of one or more independent variables.*



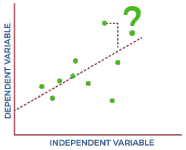
Probabilistic Interpretation…

Since we have values from 0 to 1, we can use it to **model class probability**

Logistic Regression

Logistic Regression

Models the probability of an event taking place *by having the* ***log-odds for the event be a linear combination*** *of one or more independent variables.*



Probabilistic Interpretation…

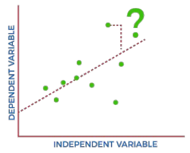
Since we have values from 0 to 1, we can use it to **model class probability**

$$p(C = 0|\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x} + w_0) \quad \text{with} \quad \sigma(z) = \frac{1}{1 + \exp(-z)}$$

Logistic Regression

Logistic Regression

Models the probability of an event taking place *by having the* ***log-odds for the event be a linear combination*** *of one or more independent variables.*



Probabilistic Interpretation…

Since we have values from 0 to 1, we can use it to **model class probability**

$$p(C=0|\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x} + w_0) \quad \text{with} \quad \sigma(z) = \frac{1}{1+\exp(-z)}$$

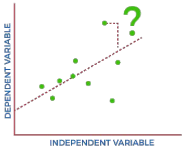
By substitution…

$$p(C=0|\mathbf{x}) = \frac{1}{1+\exp\left(-\mathbf{w}^T\mathbf{x} - w_0\right)}$$

Logistic Regression

Logistic Regression

Models the probability of an event taking place *by having the* ***log-odds for the event be a linear combination*** *of one or more independent variables.*



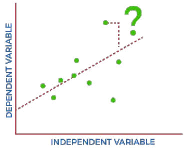
Probabilistic Interpretation…

Suppose we have 2 classes, how to get the probability for the other class?

Logistic Regression

Logistic Regression

Models the probability of an event taking place *by having the* ***log-odds for the event be a linear combination*** *of one or more independent variables.*



Probabilistic Interpretation…

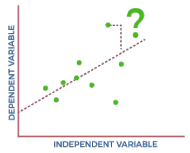
Suppose we have 2 classes, how to get the probability for the other class?

Use **marginalization property of probability**:

Logistic Regression

Logistic Regression

Models the probability of an event taking place *by having the* ***log-odds for the event be a linear combination*** *of one or more independent variables.*



Probabilistic Interpretation…

Suppose we have 2 classes, how to get the probability for the other class?

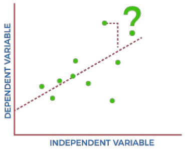
Use **marginalization property of probability**:

$$p(C = 1|\mathbf{x}) + p(C = 0|\mathbf{x}) = 1$$

Logistic Regression

Logistic Regression

Models the probability of an event taking place *by having the* ***log-odds for the event be a linear combination*** *of one or more independent variables.*



Probabilistic Interpretation…

Suppose we have 2 classes, how to get the probability for the other class?

Use **marginalization property of probability**:

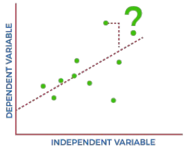
$$p(C = 1|\mathbf{x}) + p(C = 0|\mathbf{x}) = 1$$

Thus…

Logistic Regression

Logistic Regression

Models the probability of an event taking place *by having the* ***log-odds for the event be a linear combination*** *of one or more independent variables.*



Probabilistic Interpretation…

Suppose we have 2 classes, how to get the probability for the other class?

Use **marginalization property of probability**:

$$p(C = 1|\mathbf{x}) + p(C = 0|\mathbf{x}) = 1$$

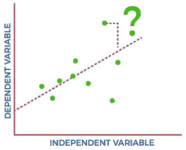
Thus…

$$p(C = 1|\mathbf{x}) = 1 - \frac{1}{1 + \exp\left(-\mathbf{w}^T\mathbf{x} - w_0\right)} = \frac{\exp(-\mathbf{w}^T\mathbf{x} - w_0)}{1 + \exp\left(-\mathbf{w}^T\mathbf{x} - w_0\right)}$$

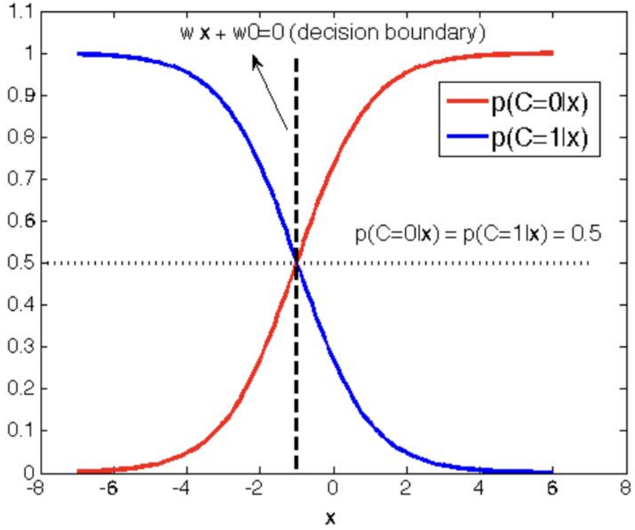
Logistic Regression

Logistic Regression

Models the probability of an event taking place *by having the* ***log-odds for the event be a linear combination*** *of one or more independent variables.*



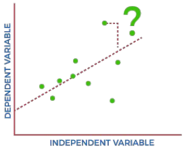
Probabilistic Interpretation…



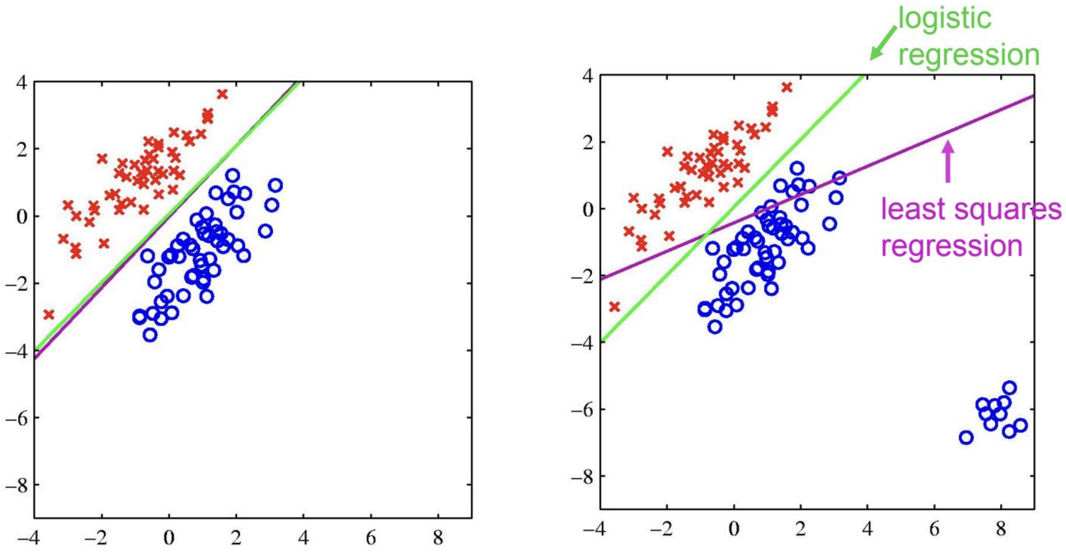
Logistic Regression

Logistic Regression

Models the probability of an event taking place *by having the* ***log-odds for the event be a linear combination*** *of one or more independent variables.*



Logistic Regression vs. Least Squares



Logistic Regression

A working example…

Logistic Regression

A working example…

Problem: Given the number of hours a student spent learning, will (s)he pass the exam?

Logistic Regression

A working example…

Problem: Given the number of hours a student spent learning, will (s)he pass the exam?

Training Data: (Top row are the features $\mathbf{x}^{(i)}$ and the bottom row are the targets $t^{(i)}$)

Logistic Regression

A working example…

Problem: Given the number of hours a student spent learning, will (s)he pass the exam?

Training Data: (Top row are the features $\mathbf{x}^{(i)}$ and the bottom row are the targets $t^{(i)}$)

Hours	0.50	0.75	1.00	1.25	1.50	1.75	1.75	2.00	2.25	2.50	2.75	3.00	3.25	3.50	4.00	4.25	4.50	4.75	5.00	5.50
Pass	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1

Logistic Regression

A working example…

Problem: Given the number of hours a student spent learning, will (s)he pass the exam?

Training Data: (Top row are the features $\mathbf{x}^{(i)}$ and the bottom row are the targets $t^{(i)}$)

Hours	0.50	0.75	1.00	1.25	1.50	1.75	1.75	2.00	2.25	2.50	2.75	3.00	3.25	3.50	4.00	4.25	4.50	4.75	5.00	5.50
Pass	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1

Learn the weights for our model (How to do so will come shortly after this)

Logistic Regression

A working example…

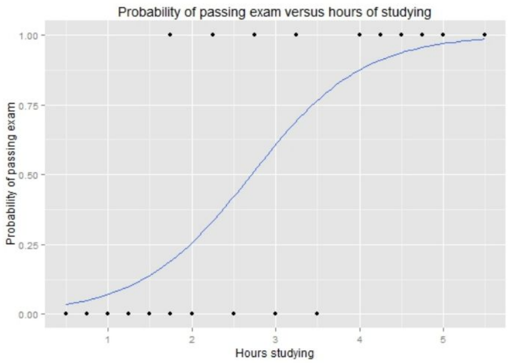
Problem: Given the number of hours a student spent learning, will (s)he pass the exam?

Training Data: (Top row are the features $\mathbf{x}^{(i)}$ and the bottom row are the targets $t^{(i)}$)

Hours	0.50	0.75	1.00	1.25	1.50	1.75	1.75	2.00	2.25	2.50	2.75	3.00	3.25	3.50	4.00	4.25	4.50	4.75	5.00	5.50
Pass	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1

Learn the weights for our model (How to do so will come shortly after this)

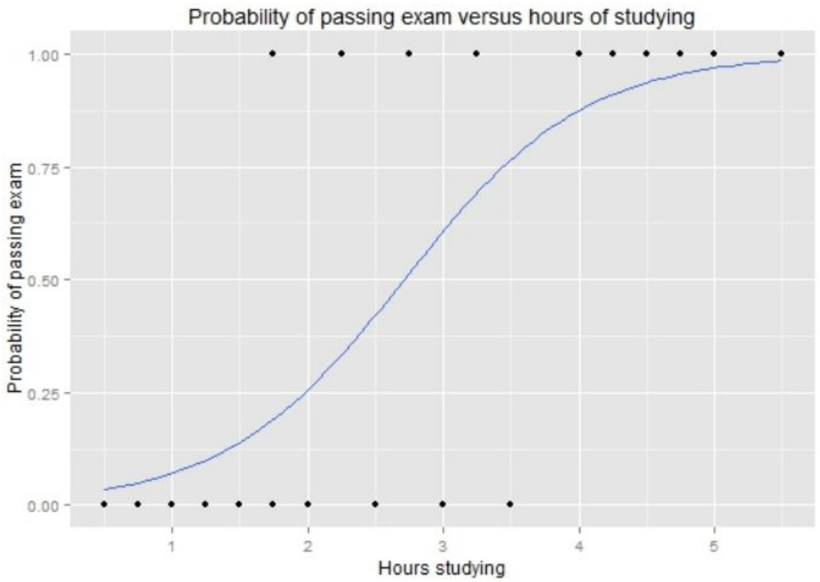
Make predictions:



Hours of study	Probability of passing exam
1	0.07
2	0.26
3	0.61
4	0.87
5	0.97

Logistic Regression

A working example…

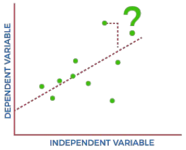


Hours of study	Probability of passing exam
1	0.07
2	0.26
3	0.61
4	0.87
5	0.97

Logistic Regression

Logistic Regression

Models the probability of an event taking place *by having the* ***log-odds for the event be a linear combination*** *of one or more independent variables.*

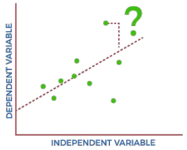


How to correctly learn the weights?

Logistic Regression

Logistic Regression

Models the probability of an event taking place *by having the* ***log-odds for the event be a linear combination*** *of one or more independent variables.*



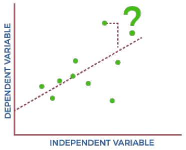
How to correctly learn the weights?

Need to have a **probabilistic model**. For this, we'll use **Maximum Likelihood**

Logistic Regression

Logistic Regression

Models the probability of an event taking place *by having the* ***log-odds for the event be a linear combination*** *of one or more independent variables.*



How to correctly learn the weights?

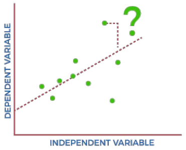
Need to have a **probabilistic model**. For this, we'll use **Maximum Likelihood**

Assume $t = \{0, 1\}$ we can write the probability of the training points

Logistic Regression

Logistic Regression

Models the probability of an event taking place *by having the* ***log-odds for the event be a linear combination*** *of one or more independent variables.*



How to correctly learn the weights?

Need to have a **probabilistic model**. For this, we'll use **Maximum Likelihood**

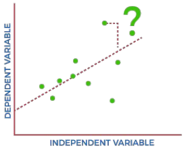
Assume $t = \{0, 1\}$ we can write the probability of the training points

$$p(t^{(1)}, \cdots, t^{(N)} | \mathbf{x}^{(1)}, \cdots \mathbf{x}^{(N)}; \mathbf{w})$$

Logistic Regression

Logistic Regression

Models the probability of an event taking place *by having the* ***log-odds for the event be a linear combination*** *of one or more independent variables.*



How to correctly learn the weights?

Need to have a **probabilistic model**. For this, we'll use **Maximum Likelihood**

Assume $t = \{0, 1\}$ we can write the probability of the training points

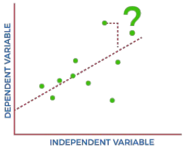
$$p(t^{(1)}, \cdots, t^{(N)} | \mathbf{x}^{(1)}, \cdots \mathbf{x}^{(N)}; \mathbf{w})$$

Assuming that the training examples are sampled IID: **independent and identically distributed**, we can write the likelihood function:

Logistic Regression

Logistic Regression

Models the probability of an event taking place *by having the* ***log-odds for the event be a linear combination*** *of one or more independent variables.*



How to correctly learn the weights?

Need to have a **probabilistic model**. For this, we'll use **Maximum Likelihood**

Assume $t = \{0, 1\}$ we can write the probability of the training points

$$p(t^{(1)}, \cdots, t^{(N)} | \mathbf{x}^{(1)}, \cdots \mathbf{x}^{(N)}; \mathbf{w})$$

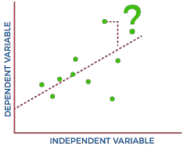
Assuming that the training examples are sampled IID: **independent and identically distributed**, we can write the likelihood function:

$$L(\mathbf{w}) = p(t^{(1)}, \cdots, t^{(N)} | \mathbf{x}^{(1)}, \cdots \mathbf{x}^{(N)}; \mathbf{w}) = \prod_{i=1}^{N} p(t^{(i)} | \mathbf{x}^{(i)}; \mathbf{w})$$

Logistic Regression

Logistic Regression

Models the probability of an event taking place by having the ***log-odds for the event be a linear combination*** *of one or more independent variables.*



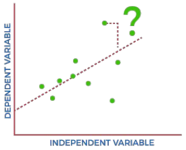
How to correctly learn the weights?

$$L(\mathbf{w}) = p(t^{(1)}, \cdots, t^{(N)} | \mathbf{x}^{(1)}, \cdots \mathbf{x}^{(N)}; \mathbf{w}) = \prod_{i=1}^{N} p(t^{(i)} | \mathbf{x}^{(i)}; \mathbf{w})$$

Logistic Regression

Logistic Regression

Models the probability of an event taking place *by having the* ***log-odds for the event be a linear combination*** *of one or more independent variables.*



How to correctly learn the weights?

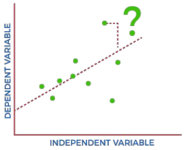
$$L(\mathbf{w}) = p(t^{(1)}, \cdots, t^{(N)} | \mathbf{x}^{(1)}, \cdots \mathbf{x}^{(N)}; \mathbf{w}) = \prod_{i=1}^{N} p(t^{(i)} | \mathbf{x}^{(i)}; \mathbf{w})$$

For binary classification, can write probability as the following:

Logistic Regression

Logistic Regression

Models the probability of an event taking place *by having the* ***log-odds for the event be a linear combination*** *of one or more independent variables.*



How to correctly learn the weights?

$$L(\mathbf{w}) = p(t^{(1)}, \cdots, t^{(N)} | \mathbf{x}^{(1)}, \cdots \mathbf{x}^{(N)}; \mathbf{w}) = \prod_{i=1}^{N} p(t^{(i)} | \mathbf{x}^{(i)}; \mathbf{w})$$

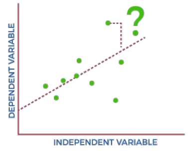
For binary classification, can write probability as the following:

$$p(t^{(i)} | \mathbf{x}^{(i)}; \mathbf{w}) \quad = \quad p(C = 1 | \mathbf{x}^{(i)}; \mathbf{w})^{t^{(i)}} p(C = 0 | \mathbf{x}^{(i)}; \mathbf{w})^{1 - t^{(i)}}$$

Logistic Regression

Logistic Regression

Models the probability of an event taking place *by having the* ***log-odds for the event be a linear combination*** *of one or more independent variables.*



How to correctly learn the weights?

$$L(\mathbf{w}) = p(t^{(1)}, \cdots, t^{(N)} | \mathbf{x}^{(1)}, \cdots \mathbf{x}^{(N)}; \mathbf{w}) = \prod_{i=1}^{N} p(t^{(i)} | \mathbf{x}^{(i)}; \mathbf{w})$$

For binary classification, can write probability as the following:

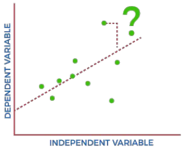
Simply expanded to this

$$p(t^{(i)} | \mathbf{x}^{(i)}; \mathbf{w}) \quad = \quad p(C = 1 | \mathbf{x}^{(i)}; \mathbf{w})^{t^{(i)}} p(C = 0 | \mathbf{x}^{(i)}; \mathbf{w})^{1 - t^{(i)}}$$

Logistic Regression

Logistic Regression

Models the probability of an event taking place *by having the* ***log-odds for the event be a linear combination*** *of one or more independent variables.*



How to correctly learn the weights?

$$L(\mathbf{w}) = p(t^{(1)}, \cdots, t^{(N)} | \mathbf{x}^{(1)}, \cdots \mathbf{x}^{(N)}; \mathbf{w}) = \prod_{i=1}^{N} p(t^{(i)} | \mathbf{x}^{(i)}; \mathbf{w})$$

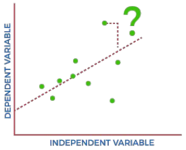
For binary classification, can write probability as the following:

$$\begin{aligned} p(t^{(i)} | \mathbf{x}^{(i)}; \mathbf{w}) &= p(C = 1 | \mathbf{x}^{(i)}; \mathbf{w})^{t^{(i)}} p(C = 0 | \mathbf{x}^{(i)}; \mathbf{w})^{1 - t^{(i)}} \\ &= \left(1 - p(C = 0 | \mathbf{x}^{(i)}; \mathbf{w})\right)^{t^{(i)}} p(C = 0 | \mathbf{x}^{(i)}; \mathbf{w})^{1 - t^{(i)}} \end{aligned}$$

Logistic Regression

Logistic Regression

Models the probability of an event taking place *by having the* ***log-odds for the event be a linear combination*** *of one or more independent variables.*



How to correctly learn the weights?

$$L(\mathbf{w}) = p(t^{(1)}, \cdots, t^{(N)} | \mathbf{x}^{(1)}, \cdots \mathbf{x}^{(N)}; \mathbf{w}) = \prod_{i=1}^{N} p(t^{(i)} | \mathbf{x}^{(i)}; \mathbf{w})$$

For binary classification, can write probability as the following:

Simply expanded to this

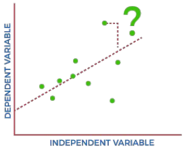
$$\begin{aligned} p(t^{(i)} | \mathbf{x}^{(i)}; \mathbf{w}) &= p(C = 1 | \mathbf{x}^{(i)}; \mathbf{w})^{t^{(i)}} p(C = 0 | \mathbf{x}^{(i)}; \mathbf{w})^{1 - t^{(i)}} \\ &= \left(1 - p(C = 0 | \mathbf{x}^{(i)}; \mathbf{w})\right)^{t^{(i)}} p(C = 0 | \mathbf{x}^{(i)}; \mathbf{w})^{1 - t^{(i)}} \end{aligned}$$

This part is using the property of marginal probability

Logistic Regression

Logistic Regression

Models the probability of an event taking place *by having the* ***log-odds for the event be a linear combination*** *of one or more independent variables.*

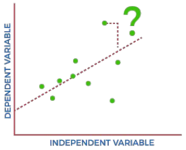


How to correctly learn the weights?

Logistic Regression

Logistic Regression

Models the probability of an event taking place *by having the* ***log-odds for the event be a linear combination*** *of one or more independent variables.*



How to correctly learn the weights?

Learn the parameters by **maximizing likelihood**

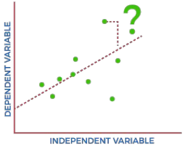
$$\max_{\mathbf{w}} L(\mathbf{w}) = \max_{\mathbf{w}} \prod_{i=1}^{N} p(t^{(i)}|\mathbf{x}^{(i)}; \mathbf{w})$$

$$\begin{aligned} L(\mathbf{w}) &= \prod_{i=1}^{N} p(t^{(i)}|\mathbf{x}^{(i)}) \qquad \text{(likelihood)} \\ &= \prod_{i=1}^{N} \left(1 - p(C=0|\mathbf{x}^{(i)})\right)^{t^{(i)}} p(C=0|\mathbf{x}^{(i)})^{1-t^{(i)}} \end{aligned}$$

Logistic Regression

Logistic Regression

Models the probability of an event taking place *by having the* ***log-odds for the event be a linear combination*** *of one or more independent variables.*



How to correctly learn the weights?

Learn the parameters by **maximizing likelihood**

$$\max_{\mathbf{w}} L(\mathbf{w}) = \max_{\mathbf{w}} \prod_{i=1}^{N} p(t^{(i)}|\mathbf{x}^{(i)}; \mathbf{w})$$

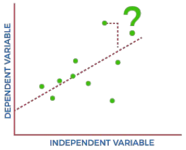
$$\begin{aligned} L(\mathbf{w}) &= \prod_{i=1}^{N} p(t^{(i)}|\mathbf{x}^{(i)}) \qquad \text{(likelihood)} \\ &= \prod_{i=1}^{N} \left(1 - p(C=0|\mathbf{x}^{(i)})\right)^{t^{(i)}} p(C=0|\mathbf{x}^{(i)})^{1-t^{(i)}} \end{aligned}$$

Convert **maximization problem into minimization** to write the loss function (using log to simplify differentiation)

Logistic Regression

Logistic Regression

Models the probability of an event taking place *by having the* ***log-odds for the event be a linear combination*** *of one or more independent variables.*



How to correctly learn the weights?

Learn the parameters by **maximizing likelihood**

$$\max_{\mathbf{w}} L(\mathbf{w}) = \max_{\mathbf{w}} \prod_{i=1}^{N} p(t^{(i)}|\mathbf{x}^{(i)};\mathbf{w})$$

$$\begin{aligned} L(\mathbf{w}) &= \prod_{i=1}^{N} p(t^{(i)}|\mathbf{x}^{(i)}) \qquad \text{(likelihood)} \\ &= \prod_{i=1}^{N} \left(1 - p(C=0|\mathbf{x}^{(i)})\right)^{t^{(i)}} p(C=0|\mathbf{x}^{(i)})^{1-t^{(i)}} \end{aligned}$$

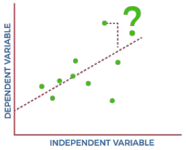
Convert **maximization problem into minimization** to write the loss function (using log to simplify differentiation)

$$\ell_{log}(\mathbf{w}) = -\log L(\mathbf{w})$$

Logistic Regression

Logistic Regression

Models the probability of an event taking place *by having the* ***log-odds for the event be a linear combination*** *of one or more independent variables.*



How to correctly learn the weights?

Learn the parameters by **maximizing likelihood**

$$\max_{\mathbf{w}} L(\mathbf{w}) = \max_{\mathbf{w}} \prod_{i=1}^{N} p(t^{(i)}|\mathbf{x}^{(i)}; \mathbf{w})$$

$$\begin{aligned} L(\mathbf{w}) &= \prod_{i=1}^{N} p(t^{(i)}|\mathbf{x}^{(i)}) \qquad \text{(likelihood)} \\ &= \prod_{i=1}^{N} \left(1 - p(C=0|\mathbf{x}^{(i)})\right)^{t^{(i)}} p(C=0|\mathbf{x}^{(i)})^{1-t^{(i)}} \end{aligned}$$

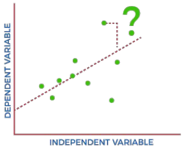
Convert **maximization problem into minimization** to write the loss function (using log to simplify differentiation)

$$\begin{aligned} \ell_{log}(\mathbf{w}) &= -\log L(\mathbf{w}) \\ &= -\sum_{i=1}^{N} \log p(t^{(i)}|\mathbf{x}^{(i)}; \mathbf{w}) \end{aligned}$$

Logistic Regression

Logistic Regression

Models the probability of an event taking place *by having the* ***log-odds for the event be a linear combination*** *of one or more independent variables.*



How to correctly learn the weights?

Learn the parameters by **maximizing likelihood**

$$\max_{\mathbf{w}} L(\mathbf{w}) = \max_{\mathbf{w}} \prod_{i=1}^{N} p(t^{(i)}|\mathbf{x}^{(i)}; \mathbf{w})$$

$$\begin{aligned} L(\mathbf{w}) &= \prod_{i=1}^{N} p(t^{(i)}|\mathbf{x}^{(i)}) \qquad \text{(likelihood)} \\ &= \prod_{i=1}^{N} \left(1 - p(C=0|\mathbf{x}^{(i)})\right)^{t^{(i)}} p(C=0|\mathbf{x}^{(i)})^{1-t^{(i)}} \end{aligned}$$

Convert **maximization problem into minimization** to write the loss function (using log to simplify differentiation)

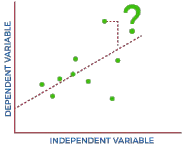
$$\begin{aligned} \ell_{log}(\mathbf{w}) &= -\log L(\mathbf{w}) \\ &= -\sum_{i=1}^{N} \log p(t^{(i)}|\mathbf{x}^{(i)}; \mathbf{w}) \end{aligned}$$

Property of logarithms log(ab) = log(a) + log(b)

Logistic Regression

Logistic Regression

Models the probability of an event taking place *by having the* ***log-odds for the event be a linear combination*** *of one or more independent variables.*



How to correctly learn the weights?

Learn the parameters by **maximizing likelihood**

$$\max_{\mathbf{w}} L(\mathbf{w}) = \max_{\mathbf{w}} \prod_{i=1}^{N} p(t^{(i)} | \mathbf{x}^{(i)}; \mathbf{w})$$

$$\begin{aligned} L(\mathbf{w}) &= \prod_{i=1}^{N} p(t^{(i)} | \mathbf{x}^{(i)}) \qquad \text{(likelihood)} \\ &= \prod_{i=1}^{N} \left(1 - p(C=0|\mathbf{x}^{(i)})\right)^{t^{(i)}} p(C=0|\mathbf{x}^{(i)})^{1-t^{(i)}} \end{aligned}$$

Convert **maximization problem into minimization** to write the loss function (using log to simplify differentiation)

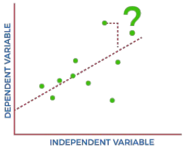
$$\begin{aligned} \ell_{log}(\mathbf{w}) &= -\log L(\mathbf{w}) \\ &= -\sum_{i=1}^{N} \log p(t^{(i)} | \mathbf{x}^{(i)}; \mathbf{w}) \\ &= -\sum_{i=1}^{N} t^{(i)} \log(1 - p(C=0|\mathbf{x}^{(i)}, \mathbf{w})) - \sum_{i=1}^{N} (1 - t^{(i)}) \log p(C=0|\mathbf{x}^{(i)}; \mathbf{w}) \end{aligned}$$

Property of logarithms log(ab) = log(a) + log(b)

Logistic Regression

Logistic Regression

Models the probability of an event taking place *by having the* ***log-odds for the event be a linear combination*** *of one or more independent variables.*



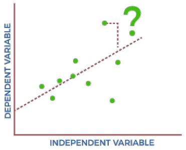
How to correctly learn the weights?

$$\min_{\mathbf{w}} \ell(\mathbf{w}) = \min_{\mathbf{w}} \left\{ -\sum_{i=1}^{N} t^{(i)} \log(1 - p(C=0|\mathbf{x}^{(i)}, \mathbf{w})) - \sum_{i=1}^{N} (1 - t^{(i)}) \log p(C=0|\mathbf{x}^{(i)}, \mathbf{w}) \right\}$$

Logistic Regression

Logistic Regression

Models the probability of an event taking place *by having the* ***log-odds for the event be a linear combination*** *of one or more independent variables.*



How to correctly learn the weights?

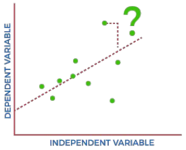
$$\min_{\mathbf{w}} \ell(\mathbf{w}) = \min_{\mathbf{w}} \left\{ -\sum_{i=1}^{N} t^{(i)} \log(1 - p(C = 0|\mathbf{x}^{(i)}, \mathbf{w})) - \sum_{i=1}^{N} (1 - t^{(i)}) \log p(C = 0|\mathbf{x}^{(i)}, \mathbf{w}) \right\}$$

Gradient Descent: iterate and at each iteration, compute the steepest direction towards optimum. Move towards that direction.

Logistic Regression

Logistic Regression

Models the probability of an event taking place *by having the* ***log-odds for the event be a linear combination*** *of one or more independent variables.*



How to correctly learn the weights?

$$\min_{\mathbf{w}} \ell(\mathbf{w}) = \min_{\mathbf{w}} \left\{ - \sum_{i=1}^{N} t^{(i)} \log(1 - p(C = 0|\mathbf{x}^{(i)}, \mathbf{w})) - \sum_{i=1}^{N} (1 - t^{(i)}) \log p(C = 0|\mathbf{x}^{(i)}, \mathbf{w}) \right\}$$

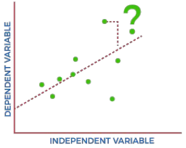
Gradient Descent: iterate and at each iteration, compute the steepest direction towards optimum. Move towards that direction.

$$w_j^{(t+1)} \leftarrow w_j^{(t)} - \lambda \frac{\partial \ell(\mathbf{w})}{\partial w_j}$$

Logistic Regression

Logistic Regression

Models the probability of an event taking place *by having the* ***log-odds for the event be a linear combination*** *of one or more independent variables.*



How to correctly learn the weights?

$$\min_{\mathbf{w}} \ell(\mathbf{w}) = \min_{\mathbf{w}} \left\{ -\sum_{i=1}^{N} t^{(i)} \log(1 - p(C = 0|\mathbf{x}^{(i)}, \mathbf{w})) - \sum_{i=1}^{N} (1 - t^{(i)}) \log p(C = 0|\mathbf{x}^{(i)}, \mathbf{w}) \right\}$$

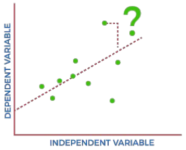
Gradient Descent: iterate and at each iteration, compute the steepest direction towards optimum. Move towards that direction.

$$w_j^{(t+1)} \leftarrow w_j^{(t)} - \lambda \frac{\partial \ell(\mathbf{w})}{\partial w_j} \qquad \nabla \ell(\mathbf{w}) = \left[\frac{\partial \ell(\mathbf{w})}{\partial w_0}, \cdots, \frac{\partial \ell(\mathbf{w})}{\partial w_k} \right]^T, \qquad \text{and} \qquad \triangle(\mathbf{w}) = -\lambda \nabla \ell(\mathbf{w})$$

Logistic Regression

Logistic Regression

Models the probability of an event taking place *by having the* ***log-odds for the event be a linear combination*** *of one or more independent variables.*



How to correctly learn the weights?

$$\min_{\mathbf{w}} \ell(\mathbf{w}) = \min_{\mathbf{w}} \left\{ -\sum_{i=1}^{N} t^{(i)} \log(1 - p(C=0|\mathbf{x}^{(i)}, \mathbf{w})) - \sum_{i=1}^{N} (1 - t^{(i)}) \log p(C=0|\mathbf{x}^{(i)}, \mathbf{w}) \right\}$$

Gradient Descent: iterate and at each iteration, compute the steepest direction towards optimum. Move towards that direction.

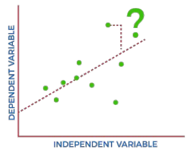
$$w_j^{(t+1)} \leftarrow w_j^{(t)} - \lambda \frac{\partial \ell(\mathbf{w})}{\partial w_j} \qquad \nabla \ell(\mathbf{w}) = \left[\frac{\partial \ell(\mathbf{w})}{\partial w_0}, \cdots, \frac{\partial \ell(\mathbf{w})}{\partial w_k} \right]^T, \qquad \text{and} \qquad \triangle(\mathbf{w}) = -\lambda \nabla \ell(\mathbf{w})$$

where are the weights?

Logistic Regression

Logistic Regression

Models the probability of an event taking place *by having the* ***log-odds for the event be a linear combination*** *of one or more independent variables.*



How to correctly learn the weights?

$$\min_{\mathbf{w}} \ell(\mathbf{w}) = \min_{\mathbf{w}} \left\{ -\sum_{i=1}^{N} t^{(i)} \log(1 - p(C=0|\mathbf{x}^{(i)}, \mathbf{w})) - \sum_{i=1}^{N} (1 - t^{(i)}) \log p(C=0|\mathbf{x}^{(i)}, \mathbf{w}) \right\}$$

Gradient Descent: iterate and at each iteration, compute the steepest direction towards optimum. Move towards that direction.

$$w_j^{(t+1)} \leftarrow w_j^{(t)} - \lambda \frac{\partial \ell(\mathbf{w})}{\partial w_j} \qquad \nabla \ell(\mathbf{w}) = \left[\frac{\partial \ell(\mathbf{w})}{\partial w_0}, \cdots, \frac{\partial \ell(\mathbf{w})}{\partial w_k} \right]^T, \qquad \text{and} \qquad \triangle(\mathbf{w}) = -\lambda \nabla \ell(\mathbf{w})$$

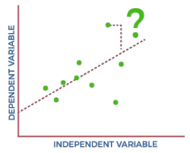
where are the weights?

$$p(C=0|\mathbf{x}) = \frac{1}{1 + \exp(-\mathbf{w}^T \mathbf{x} - w_0)}, \quad p(C=1|\mathbf{x}) = \frac{\exp(-\mathbf{w}^T \mathbf{x} - w_0)}{1 + \exp(-\mathbf{w}^T \mathbf{x} - w_0)}$$

Logistic Regression

Logistic Regression

Models the probability of an event taking place *by having the **log-odds for the event be a linear combination** of one or more independent variables.*



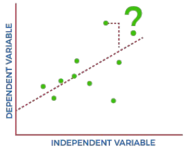
How to correctly learn the weights?

The **loss** is $$\ell_{log-loss}(\mathbf{w}) = -\sum_{i=1}^{N} t^{(i)} \log p(C=1|\mathbf{x}^{(i)}, \mathbf{w}) - \sum_{i=1}^{N} (1-t^{(i)}) \log p(C=0|\mathbf{x}^{(i)}, \mathbf{w})$$

Logistic Regression

Logistic Regression

Models the probability of an event taking place *by having the* ***log-odds for the event be a linear combination*** *of one or more independent variables.*



How to correctly learn the weights?

The **loss** is

$$\ell_{log-loss}(\mathbf{w}) = -\sum_{i=1}^{N} t^{(i)} \log p(C=1|\mathbf{x}^{(i)},\mathbf{w}) - \sum_{i=1}^{N}(1-t^{(i)}) \log p(C=0|\mathbf{x}^{(i)},\mathbf{w})$$

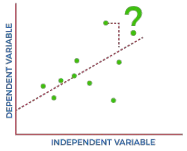
Such that:

$$p(C=0|\mathbf{x},\mathbf{w}) = \frac{1}{1+\exp(-z)} \qquad p(C=1|\mathbf{x},\mathbf{w}) = \frac{\exp(-z)}{1+\exp(-z)} \qquad z = \mathbf{w}^T\mathbf{x} + w_0$$

Logistic Regression

Logistic Regression

Models the probability of an event taking place *by having the* ***log-odds for the event be a linear combination*** *of one or more independent variables.*



How to correctly learn the weights?

The **loss** is

$$\ell_{log-loss}(\mathbf{w}) = -\sum_{i=1}^{N} t^{(i)} \log p(C=1|\mathbf{x}^{(i)}, \mathbf{w}) - \sum_{i=1}^{N} (1-t^{(i)}) \log p(C=0|\mathbf{x}^{(i)}, \mathbf{w})$$

Such that:

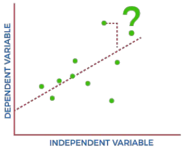
$$p(C=0|\mathbf{x}, \mathbf{w}) = \frac{1}{1+\exp(-z)} \qquad p(C=1|\mathbf{x}, \mathbf{w}) = \frac{\exp(-z)}{1+\exp(-z)} \qquad z = \mathbf{w}^T\mathbf{x} + w_0$$

Can **simplify** by using properties of logarithms:

Logistic Regression

Logistic Regression

Models the probability of an event taking place *by having the* ***log-odds for the event be a linear combination*** *of one or more independent variables.*



How to correctly learn the weights?

The **loss** is

$$\ell_{log-loss}(\mathbf{w}) = -\sum_{i=1}^{N} t^{(i)} \log p(C=1|\mathbf{x}^{(i)}, \mathbf{w}) - \sum_{i=1}^{N} (1-t^{(i)}) \log p(C=0|\mathbf{x}^{(i)}, \mathbf{w})$$

Such that:

$$p(C=0|\mathbf{x},\mathbf{w}) = \frac{1}{1+\exp(-z)} \qquad p(C=1|\mathbf{x},\mathbf{w}) = \frac{\exp(-z)}{1+\exp(-z)} \qquad z = \mathbf{w}^T\mathbf{x} + w_0$$

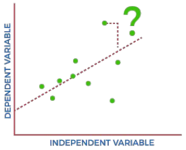
Can **simplify** by using properties of logarithms:

$$\begin{aligned}
\ell(\mathbf{w})_{log-loss} &= \sum_i t^{(i)} \log(1+\exp(-z^{(i)})) + \sum_i t^{(i)} z^{(i)} + \sum_i (1-t^{(i)}) \log(1+\exp(-z^{(i)})) \\
&= \sum_i \log(1+\exp(-z^{(i)})) + \sum_i t^{(i)} z^{(i)}
\end{aligned}$$

Logistic Regression

Logistic Regression

Models the probability of an event taking place *by having the* ***log-odds for the event be a linear combination*** *of one or more independent variables.*



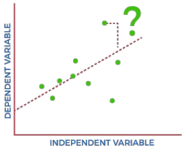
How to correctly learn the weights?

$$\ell(\mathbf{w}) = \sum_i t^{(i)} z^{(i)} + \sum_i \log(1 + \exp(-z^{(i)}))$$

Logistic Regression

Logistic Regression

Models the probability of an event taking place *by having the* ***log-odds for the event be a linear combination*** *of one or more independent variables.*



How to correctly learn the weights?

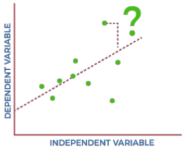
$$\ell(\mathbf{w}) = \sum_i t^{(i)} z^{(i)} + \sum_i \log(1 + \exp(-z^{(i)}))$$

Remember that $z = \mathbf{w}^T \mathbf{x} + w_0$

Logistic Regression

Logistic Regression

Models the probability of an event taking place *by having the* ***log-odds for the event be a linear combination*** *of one or more independent variables.*



How to correctly learn the weights?

$$\ell(\mathbf{w}) \quad = \quad \sum_i t^{(i)} z^{(i)} + \sum_i \log(1 + \exp(-z^{(i)}))$$

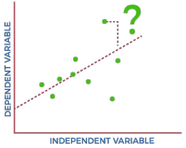
Remember that $z = \mathbf{w}^T \mathbf{x} + w_0$

$$\frac{\partial \ell}{\partial w_j} = \sum_i \left(t^{(i)} x_j^{(i)} - x_j^{(i)} \cdot \frac{\exp(-z^{(i)})}{1 + \exp(-z^{(i)})} \right)$$

Logistic Regression

Logistic Regression

Models the probability of an event taking place *by having the* ***log-odds for the event be a linear combination*** *of one or more independent variables.*



How to correctly learn the weights?

$$\ell(\mathbf{w}) \quad = \quad \sum_i t^{(i)} z^{(i)} + \sum_i \log(1 + \exp(-z^{(i)}))$$

Remember that $z = \mathbf{w}^T \mathbf{x} + w_0$

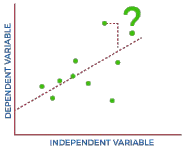
$$\frac{\partial \ell}{\partial w_j} = \sum_i \left(t^{(i)} x_j^{(i)} - x_j^{(i)} \cdot \frac{\exp(-z^{(i)})}{1 + \exp(-z^{(i)})} \right)$$

Simplifying to get

Logistic Regression

Logistic Regression

Models the probability of an event taking place *by having the* ***log-odds for the event be a linear combination*** *of one or more independent variables.*



How to correctly learn the weights?

$$\ell(\mathbf{w}) = \sum_i t^{(i)} z^{(i)} + \sum_i \log(1 + \exp(-z^{(i)}))$$

Remember that $z = \mathbf{w}^T \mathbf{x} + w_0$

$$\frac{\partial \ell}{\partial w_j} = \sum_i \left(t^{(i)} x_j^{(i)} - x_j^{(i)} \cdot \frac{\exp(-z^{(i)})}{1 + \exp(-z^{(i)})} \right)$$

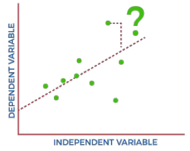
Simplifying to get

$$\frac{\partial \ell}{\partial w_j} = \sum_i x_j^{(i)} \left(t^{(i)} - p(C = 1 | \mathbf{x}^{(i)}; \mathbf{w}) \right)$$

Logistic Regression

Logistic Regression

Models the probability of an event taking place *by having the* ***log-odds for the event be a linear combination*** *of one or more independent variables.*



How to correctly learn the weights?

Gradient descent for logistic regression:

$$w_j^{(t+1)} \leftarrow w_j^{(t)} - \lambda \sum_i x_j^{(i)} \left(t^{(i)} - p(C = 1 | \mathbf{x}^{(i)}; \mathbf{w}) \right)$$

where:

$$p(C = 1 | \mathbf{x}^{(i)}; \mathbf{w}) = \frac{\exp(-\mathbf{w}^T \mathbf{x} - w_0)}{1 + \exp\left(-\mathbf{w}^T \mathbf{x} - w_0\right)} = \frac{1}{1 + \exp\left(\mathbf{w}^T \mathbf{x} + w_0\right)}$$