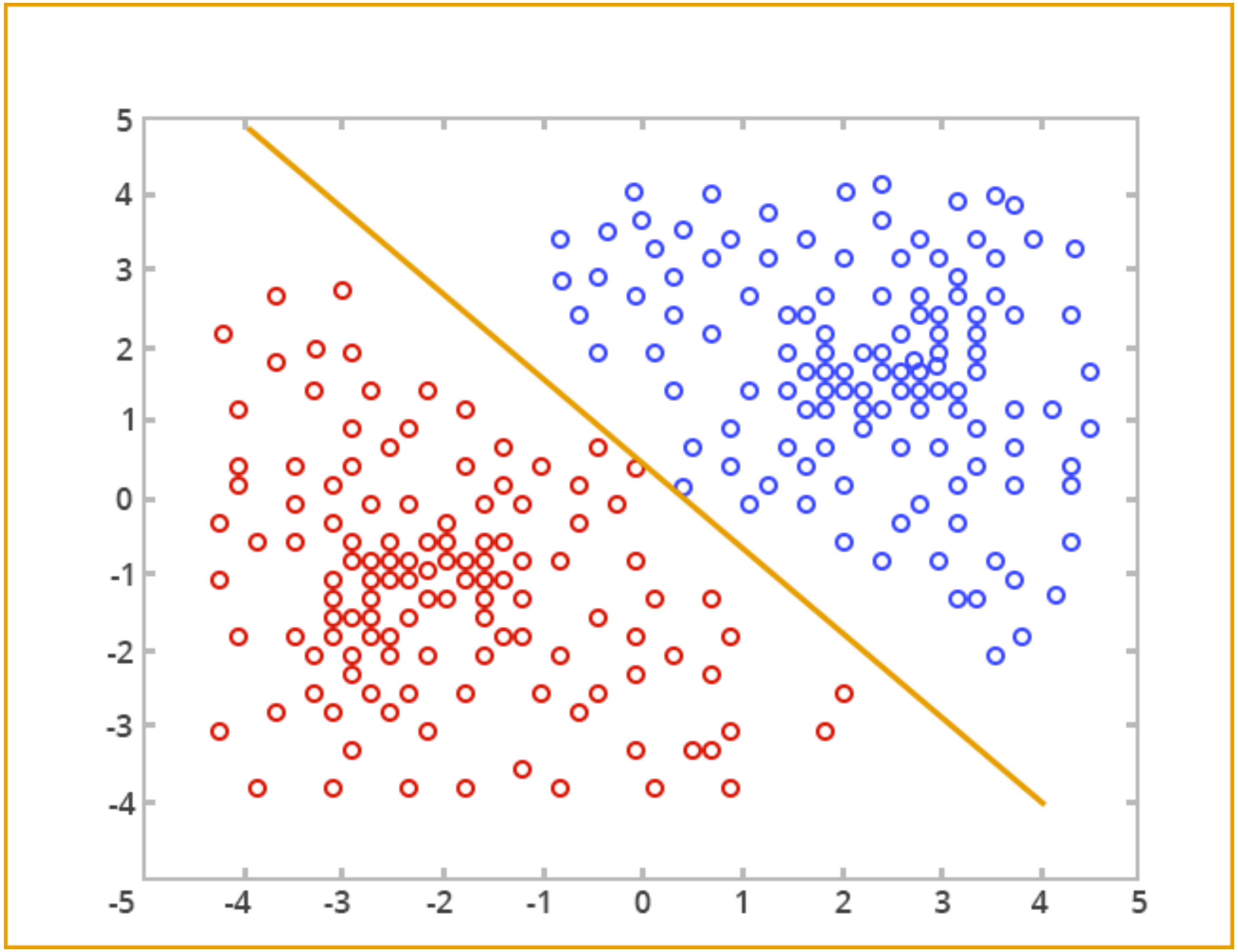


Logistic Regression: Hypothesis Set

Leandro L. Minku

Announcements

- Panopto recordings
- Canvas page
- MS Teams
- Final slides from previous lecture

Outline

- Definition of supervised learning
- Logistic regression hypothesis set
  - What kind of function can logistic regression model?
  - What parameters need to be learned?

Outline

- Definition of supervised learning
- Logistic regression hypothesis set
  - What kind of function can logistic regression model?
  - What parameters need to be learned?

From The Previous Lecture…

Supervised Learning:

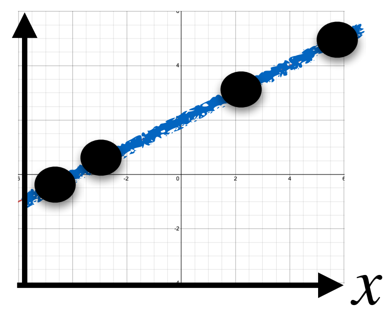
Learns a mapping from inputs $\mathbf{x} = (x_1, \ldots, x_d)^T \in \mathscr{X}$

to outputs $y \in \mathscr{Y}$,

given a training set of input-output pairs
$\mathscr{T} = \{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \cdots, (\mathbf{x}^{(N)}, y^{(N)})\}$.

Components of the Supervised Learning Process

$f(x) = 0.5x + 2$



Unknown Target Function

$f : \mathscr{X} \to \mathscr{Y}$

Training Examples

$\mathscr{T} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$

$\mathscr{T} = \{(4,4), (0,2), (8,6), (2,3)\}$

Learning Algorithm

$\mathscr{A}$

Final Hypothesis

$g \approx f$

Hypothesis Set

$\mathscr{H}$

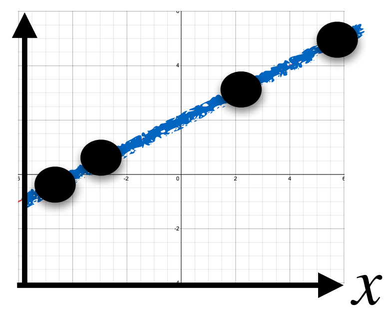
$h(\mathbf{x}) = \mathbf{a}^T\mathbf{x} + b, \quad \forall \mathbf{a} \in R^d, b \in R$

Components of the Supervised Learning Process

Unknown Target Function

$f : \mathscr{X} \rightarrow \mathscr{Y}$

$f(x) = 0.5x + 2$



The learning algorithm will search for the hypothesis $g \in \mathscr{H}$ that it believes best approximates f given $\mathscr{T}$.

Training Examples

$\mathscr{T} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$

$\mathscr{T} = \{(4,4), (0,2), (8,6), (2,3)\}$

Learning Algorithm

$\mathscr{A}$

Final Hypothesis

$g \approx f$

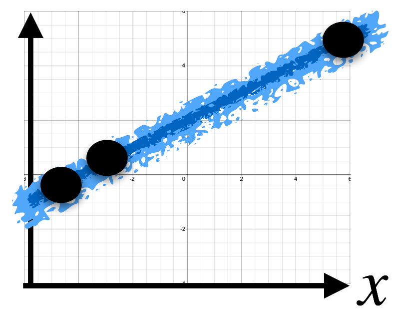
Hypothesis Set

$\mathscr{H}$

$h(\mathbf{x}) = \mathbf{a}^T\mathbf{x} + b, \quad \forall \mathbf{a} \in R^d, b \in R$

Components of the Supervised Learning Process in View of Noise

$f(x) = 0.5x + 2 + noise$



Unknown Target Function

$f : \mathscr{X} \to \mathscr{Y}$

Unknown Target Distribution

$p(y \mid \mathbf{x})$

Unknown Input Distribution

$p(\mathbf{x})$

$\mathbf{x}$

Training Examples

$T = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$

$\cancel{T = \{(4,4), (4,4), (8,6), (2,3)\}}$

$T = \{(4.3,4), (4,4.2), (7.9,5.1), (2.3,3.1)\}$

Learning Algorithm

$\mathscr{A}$

Final Hypothesis

$g \approx f$ or $g \approx p(y \mid \mathbf{x})$

$g(\mathbf{x})$

Hypothesis Set

$\mathscr{H}$

$h(\mathbf{x}) = \mathbf{a}^T\mathbf{x} + b, \quad \forall \mathbf{a} \in R^d, b \in R$

Supervised Learning Problem

- Given a set of training examples

  $$\mathcal{T} = \{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \cdots, (\mathbf{x}^{(N)}, y^{(N)})\}$$

  where $(\mathbf{x}^{(i)}, y^{(i)}) \in \mathcal{X} \times \mathcal{Y}$ are drawn i.i.d. (independently and identically distributed) from a fixed albeit unknown joint probability distribution $p(\mathbf{x}, y) = p(y \mid \mathbf{x})p(\mathbf{x})$.

- Goal: to learn a function g able to generalise to unseen (test) examples of the same probability distribution $p(\mathbf{x}, y)$.
  - $g : \mathcal{X} \to \mathcal{Y}$, mapping input space to output space.
  - g as a probability distribution approximating $p(y \mid \mathbf{x})$.

Equivalent Terms

- x_i: input, input attribute, input feature, independent variable, input variable.
- y: output attribute, output variable, dependent variable, label (for classification).
- mapping: learned function, predictive model, classifier (for classification).
- Learning a function, learning a model, training a model, building a model.
- $\mathcal{T}$: set of training examples, training data.
- $(\mathbf{x}, y)$: example, observation, data point, instance (more frequently used for examples with unknown outputs).
- Different people and books will use different terms and notations!

Notation

- Scalar: lower case, e.g, b.
- Column Vector: lower case, bold, e.g., $\mathbf{x}$.
- Vector element: lower case with subscript, e.g., x_i.
- Matrix: upper case, bold, e.g., $\mathbf{X}$.
- Matrix element: upper case with subscripts, e.g., $X_{i,j}$.
- If enumerating these (e.g., having multiple vectors), superscript will be used to differentiate this from indices, e.g., $\mathbf{x}^{(i)}$.

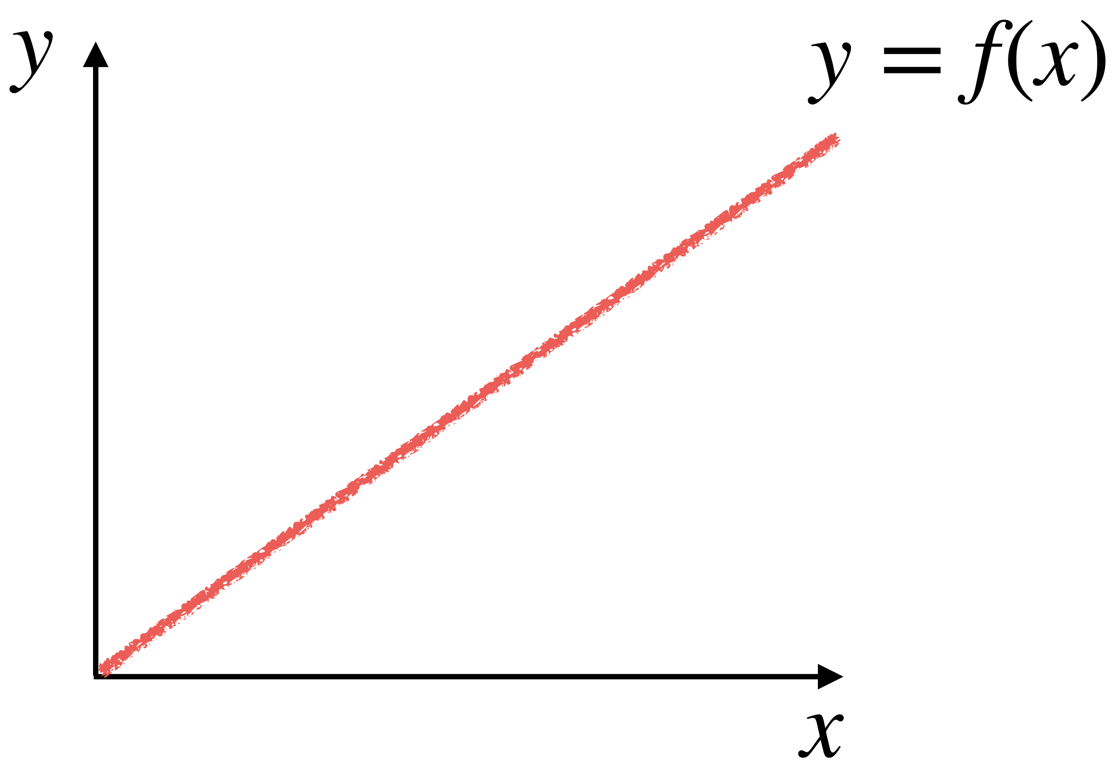
Outline

- Definition of supervised learning
- Logistic regression hypothesis set
    - What kind of function can logistic regression model?
    - What parameters need to be learned?

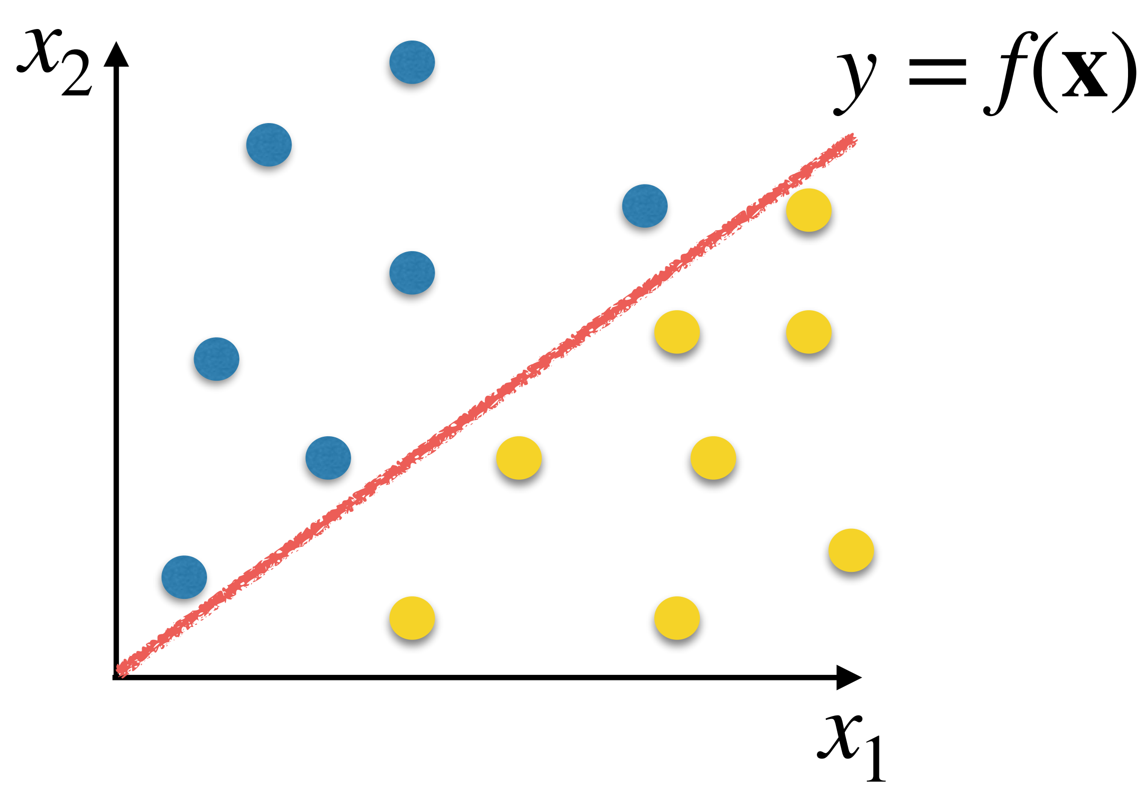
General Idea

- Despite the name, Logistic Regression is an approach for classification problems.
- In Logistic Regression, we will model the probability (actually the log odds) of an instance to belong to a given class as a linear combination of the inputs.

Regression:



Classification:



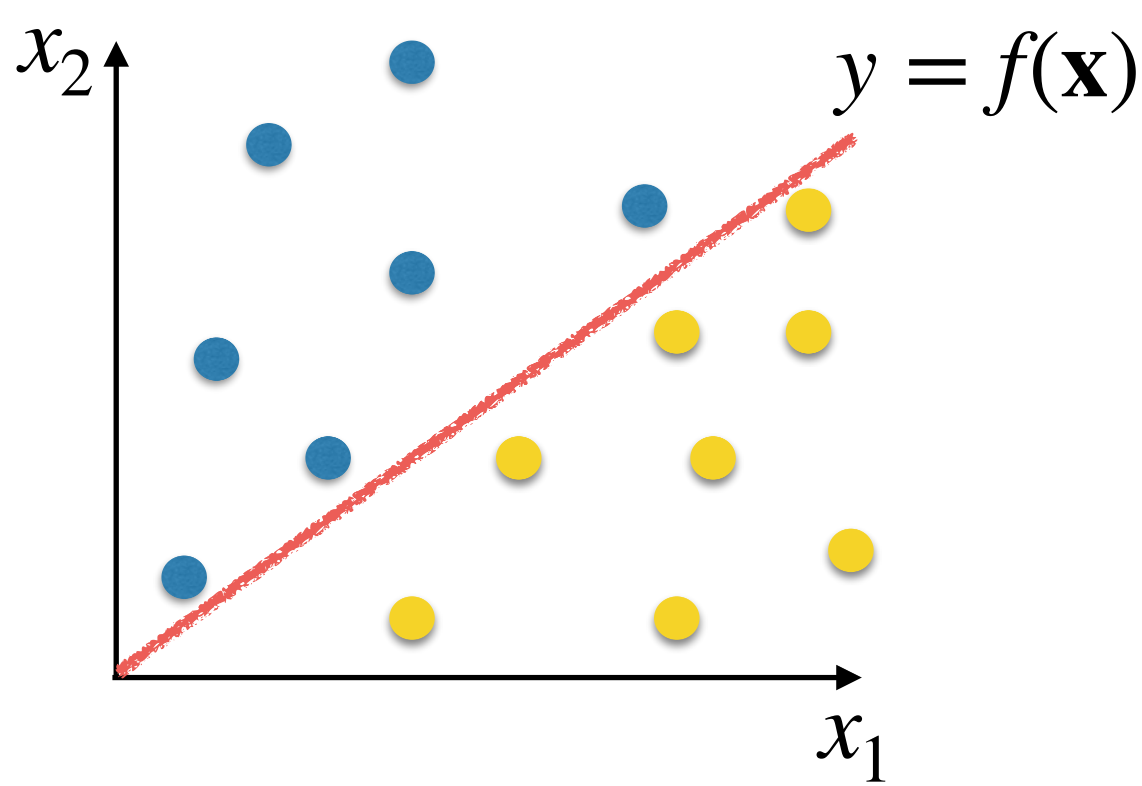
Focus

- We will focus on binary classification problems, i.e., problems where $\mathscr{Y}$ is a set containing two possible categorical values (classes), e.g., $\mathscr{Y} = \{c_0, c_1\} = \{0,1\}$.

- We assume numeric inputs

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \in \mathbb{R}^d$$

Classification:



The Need for the Logit Function

- Consider that we wish to model $P(y = 1 \mid \mathbf{x}) = P(1 \mid \mathbf{x})$ as a function of the input variables:

$$p(1 \mid \mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + \cdots + w_d x_d$$

$$p(1 \mid \mathbf{x}, \mathbf{w}) = w_0 x_0 + w_1 x_1 + \cdots + w_d x_d, \text{ where } x_0 = 1$$

$$p(1 \mid \mathbf{x}, \mathbf{w}) = \mathbf{w}^T \mathbf{x}$$

$$p_1 = \mathbf{w}^T \mathbf{x}$$

- If that was possible, we would be able to deal with this classification problem by learning the coefficients $\mathbf{w} \in \mathbb{R}^{d+1}$ and predicting class 1 if $p_1 \geq 0.5$ and 0 otherwise.

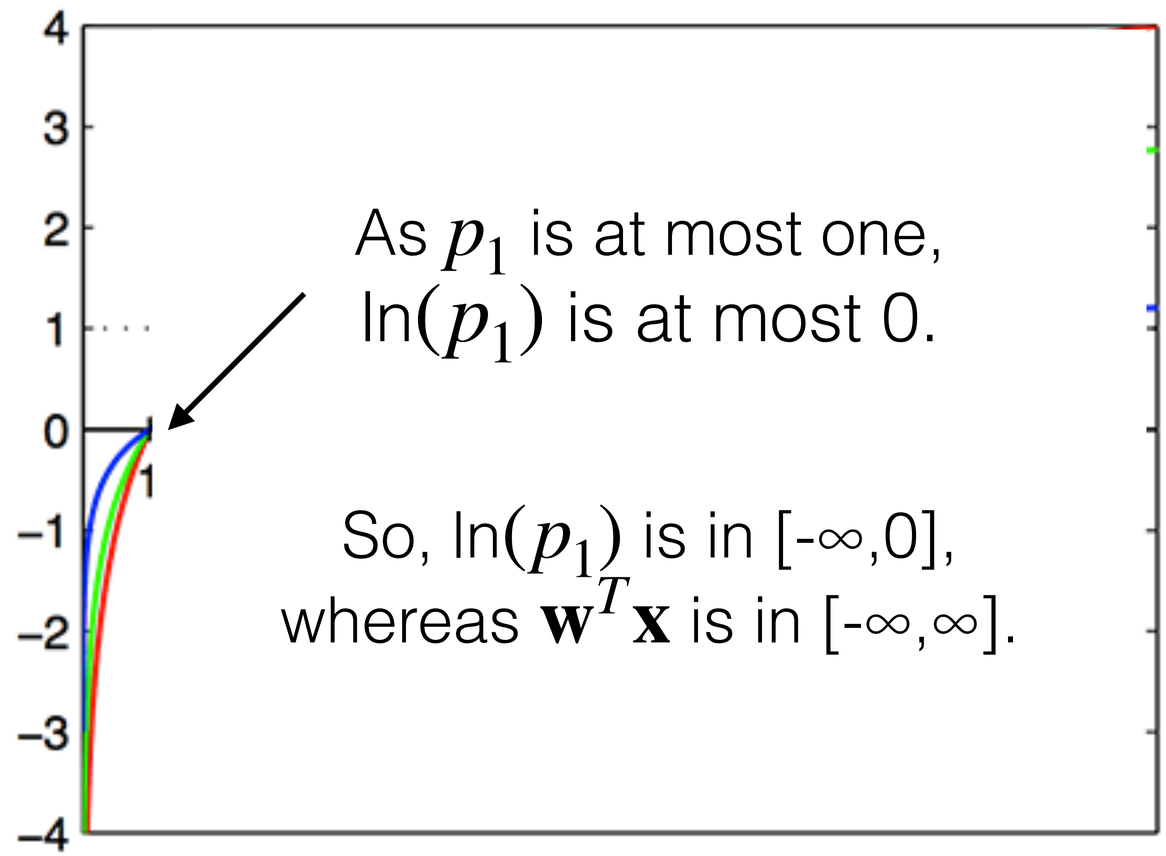
- However, $\mathbf{w}^T \mathbf{x}$ could assume any values in [-∞,∞], whereas p_1 should be in [0,1].

The Need for the Logit Function

- To fix that, one might think of modelling ln(p_1) instead of p_1:

$$\ln(p_1) = \mathbf{w}^T\mathbf{x}$$

- However, logarithms are unbounded only from one direction and linear functions are not.



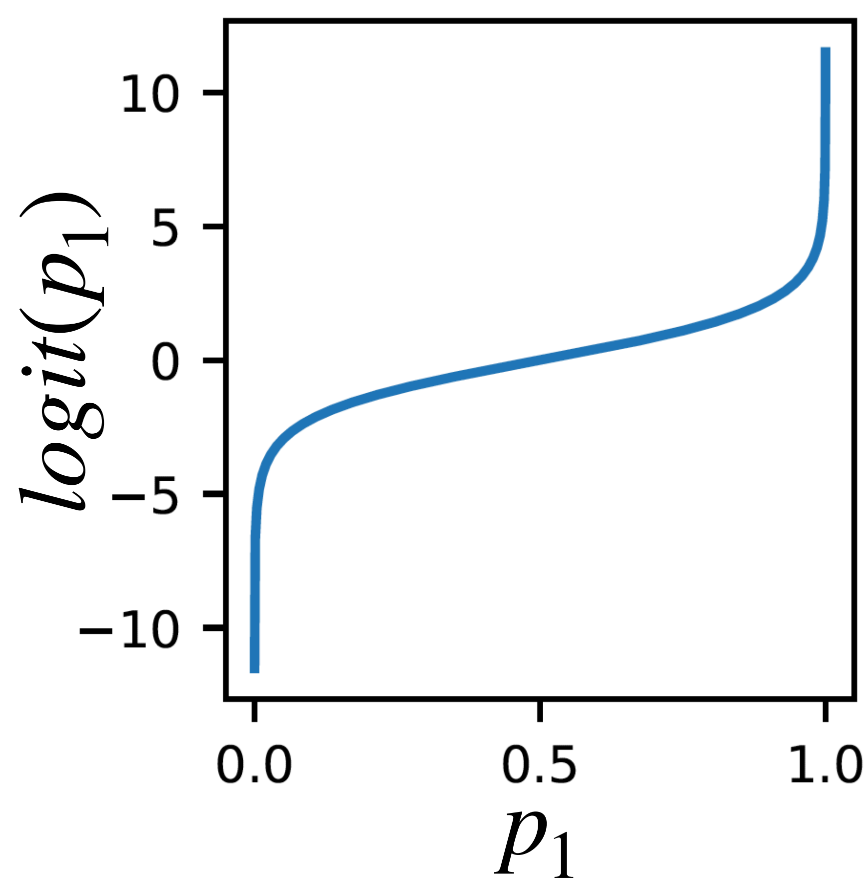
Again, we cannot use a linear combination to model ln(p_1).

The Need for the Logit Function

- A solution would be to create a model logit$(p_1) = \mathbf{w}^T\mathbf{x}$, where

  logit(p_1) = ln $\left(\dfrac{p_1}{1-p_1}\right)$

- Logit enables us to map from [0,1] to [-∞,∞].



So, logit(p_1) is in [-∞,∞],
and $\mathbf{w}^T\mathbf{x}$ is in [-∞,∞].

So, we can model
logit$(p_1) = \mathbf{w}^T\mathbf{x}$

The Odds

$$\text{logit}(p_1) = \ln\left(\frac{p_1}{1-p_1}\right) = \mathbf{w}^T\mathbf{x}$$

- Odds: ratio of probabilities of two possible outcomes:

$$o_1 = \frac{p_1}{p_0} = \frac{p_1}{1-p_1}$$

- For example,

If $p_1 = 0.7$ and $p_0 = 0.3$, $o_1 \approx 2.33$
If $p_1 = 0.5$ and $p_0 = 0.5$, $o_1 = 1$
If $p_1 = 0.3$ and $p_0 = 0.7$, $o_1 \approx 0.43$

- If $o_1 \geq 1$, predict class 1.
- If $o_1 < 1$, predict class 0.

Bernoulli distribution: a discrete probability distribution of a random variable that takes value 1 with probability p_1 and value 0 with probability $p_0 = 1 - p_1$.

Logit

- Logit: logarithm of the odds.

$$\text{logit}(p_1) = \ln\left(\frac{p_1}{1 - p_1}\right)$$

- For example,

If $p_1 = 0.7$ and $p_0 = 0.3$, $\text{logit}(p_1) \approx 0.85$
If $p_1 = 0.5$ and $p_0 = 0.5$, $\text{logit}(p_1) = 0$
If $p_1 = 0.3$ and $p_0 = 0.7$, $\text{logit}(p_1) \approx -0.85$

- If $\text{logit}(p_1) = \mathbf{w}^T\mathbf{x} \geq 0$, predict class 1.
- If $\text{logit}(p_1) = \mathbf{w}^T\mathbf{x} < 0$, predict class 0.

This is the key idea behind logistic regression!

Coefficients $\mathbf{w}$ are "parameters" of the function that we need to learn based on training examples.

A Linear Classifier

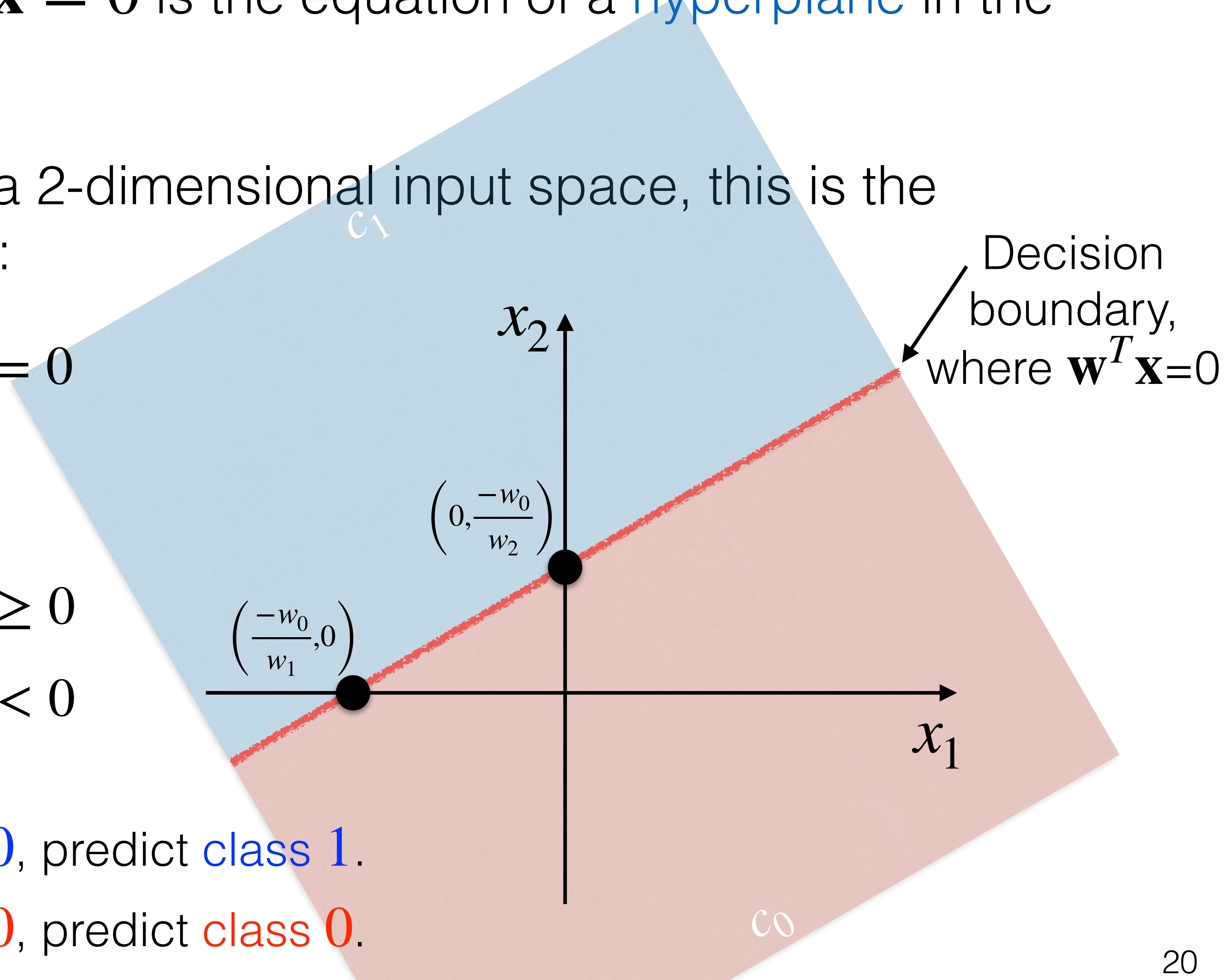
- The equation $\mathbf{w}^T\mathbf{x} = 0$ is the equation of a hyperplane in the input space.
- For example, for a 2-dimensional input space, this is the equation of a line:

$$w_0x_0 + w_1x_1 + w_2x_2 = 0$$

$$w_1x_1 + w_2x_2 = -w_0$$

$$w_0x_0 + w_1x_1 + w_2x_2 \geq 0$$

$$w_0x_0 + w_1x_1 + w_2x_2 < 0$$



- If $\text{logit}(p_1) = \mathbf{w}^T\mathbf{x} \geq 0$, predict class 1.
- If $\text{logit}(p_1) = \mathbf{w}^T\mathbf{x} < 0$, predict class 0.

Computing the Probabilities p_1 and p_0

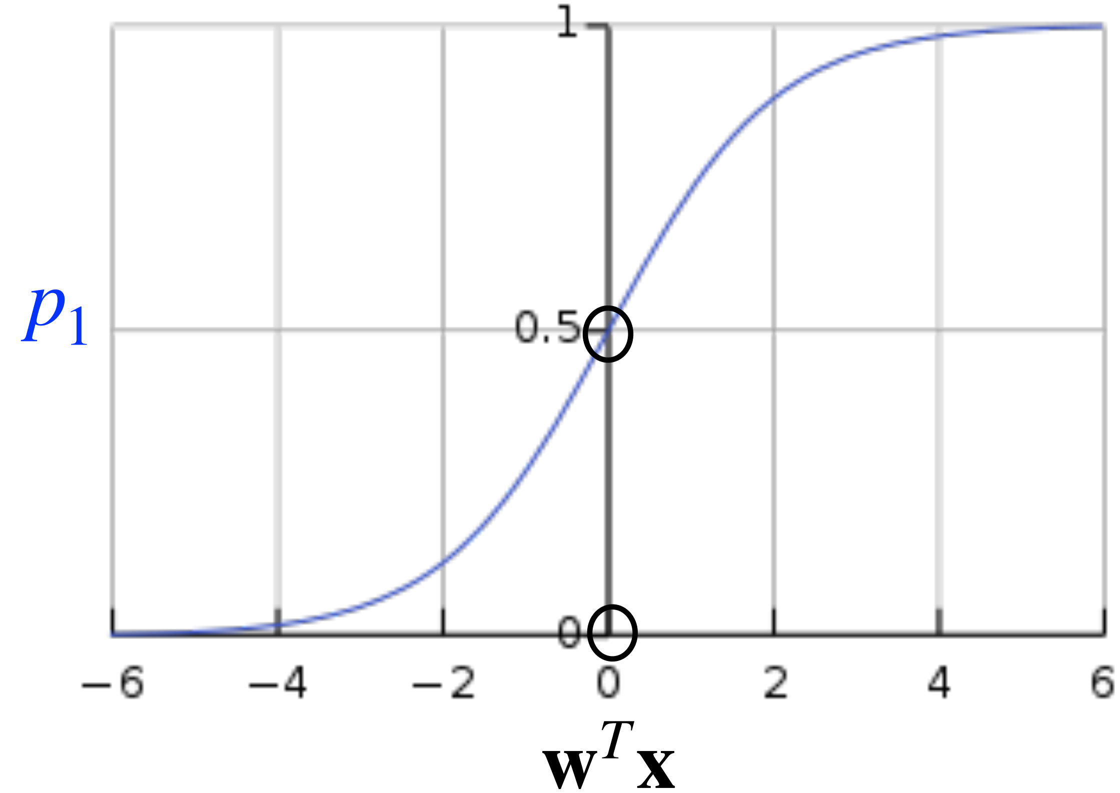
- logit$(p_1) = \mathbf{w}^T\mathbf{x}$
  - $\mathbf{w}^T\mathbf{x} \geq 0 \rightarrow$ class 1
  - $\mathbf{w}^T\mathbf{x} < 0 \rightarrow$ class 0
- If we solve logit$(p_1) = \mathbf{w}^T\mathbf{x}$ for p_1 we get:

$$p_1 = \frac{e^{(\mathbf{w}^T\mathbf{x})}}{1 + e^{(\mathbf{w}^T\mathbf{x})}}$$

$$p_0 = 1 - p_1 = \frac{1}{1 + e^{(\mathbf{w}^T\mathbf{x})}}$$

$p_1 \geq 0.5 \rightarrow$ class 1

Sigmoid logistic function



Computing the Probabilities p_1 and p_0

- $\text{logit}(p_1) = \mathbf{w}^T\mathbf{x}$
  - $\mathbf{w}^T\mathbf{x} \geq 0 \rightarrow$ class 1
  - $\mathbf{w}^T\mathbf{x} < 0 \rightarrow$ class 0
- If we solve $\text{logit}(p_1) = \mathbf{w}^T\mathbf{x}$ for p_1 we get:

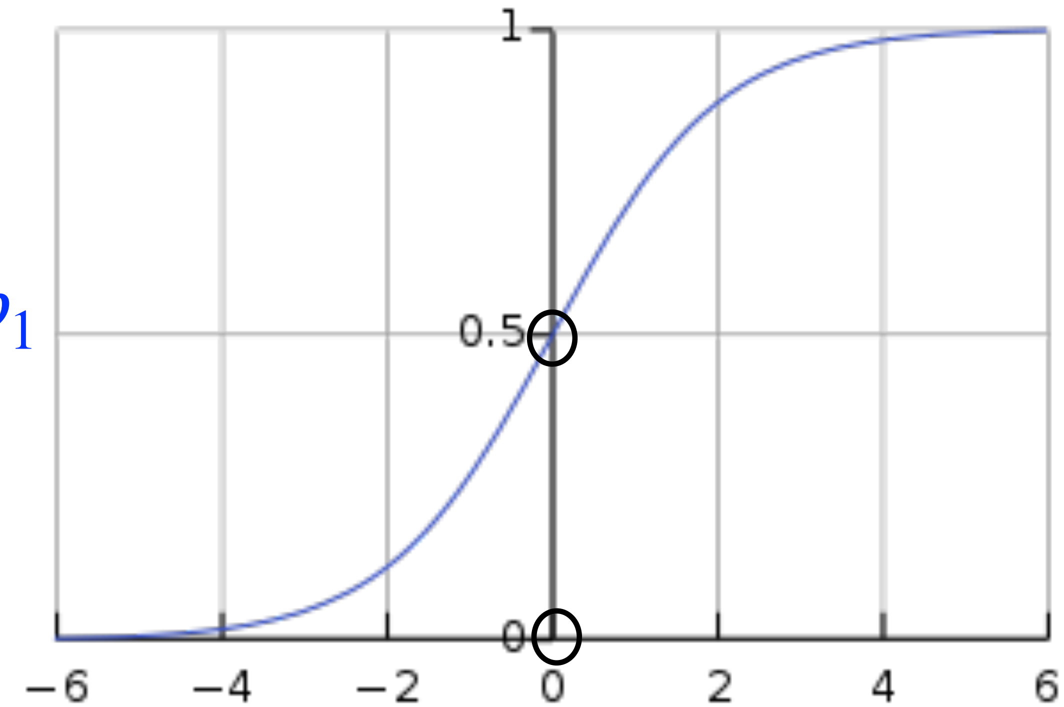
$$p_1 = \frac{e^{(\mathbf{w}^T\mathbf{x})}}{1 + e^{(\mathbf{w}^T\mathbf{x})}}$$

$$p_0 = 1 - p_1 = \frac{1}{1 + e^{(\mathbf{w}^T\mathbf{x})}}$$

- $p_1 \geq 0.5 \rightarrow$ class 1
- $p_1 < 0.5 \rightarrow$ class 0

Sigmoid logistic function

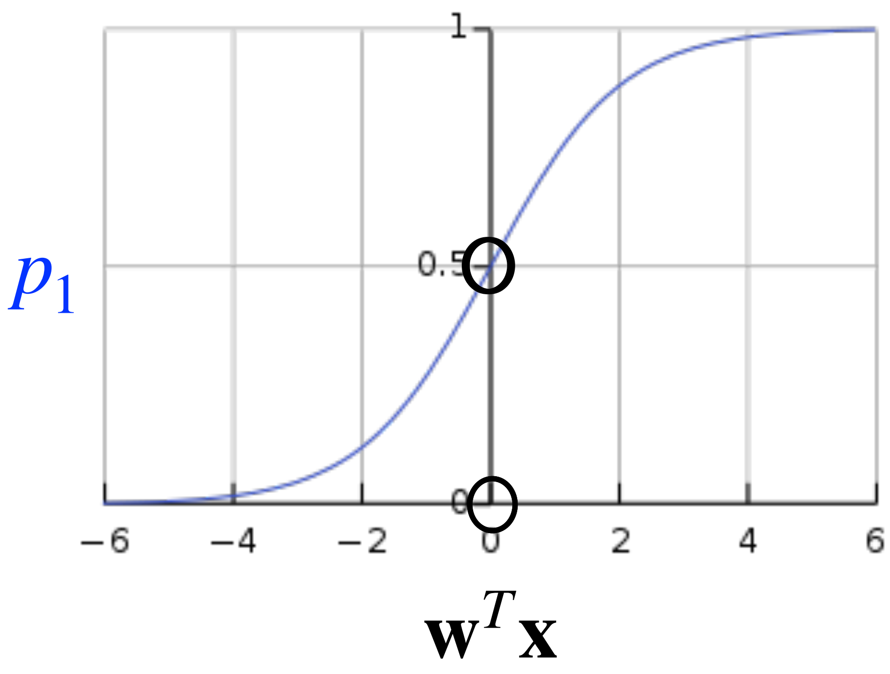
p_1

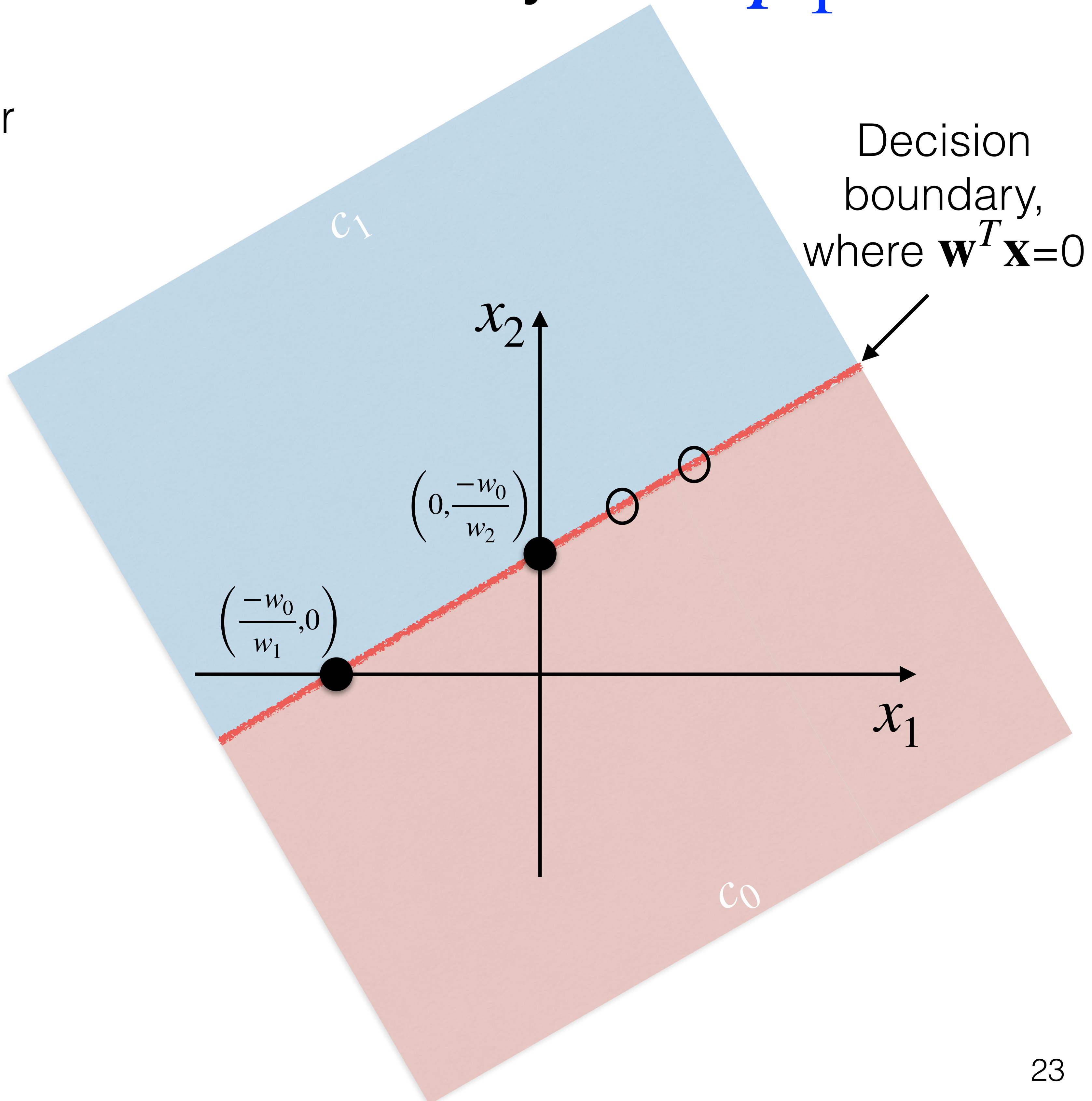


$\mathbf{w}^T\mathbf{x}$

The Relationship Between the Distance To The Decision Boundary and p_1

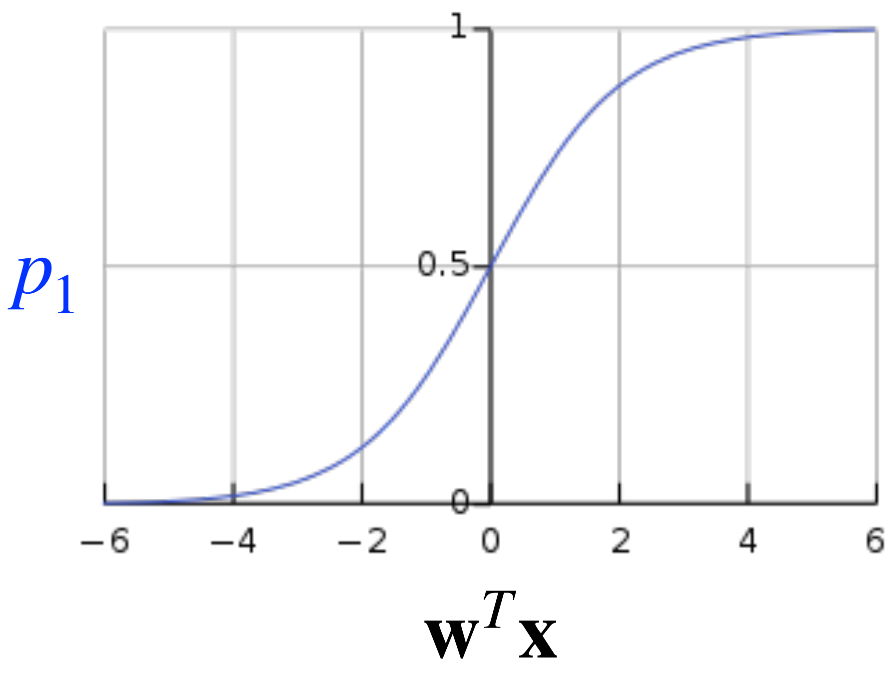
- The larger $|\mathbf{w}^T\mathbf{x}|$, the further away from the decision boundary the example $\mathbf{x}$ is.
- The larger $\mathbf{w}^T\mathbf{x}$, the higher p_1.
- The more negative $\mathbf{w}^T\mathbf{x}$, the smaller the p_1 (and the larger the p_0).

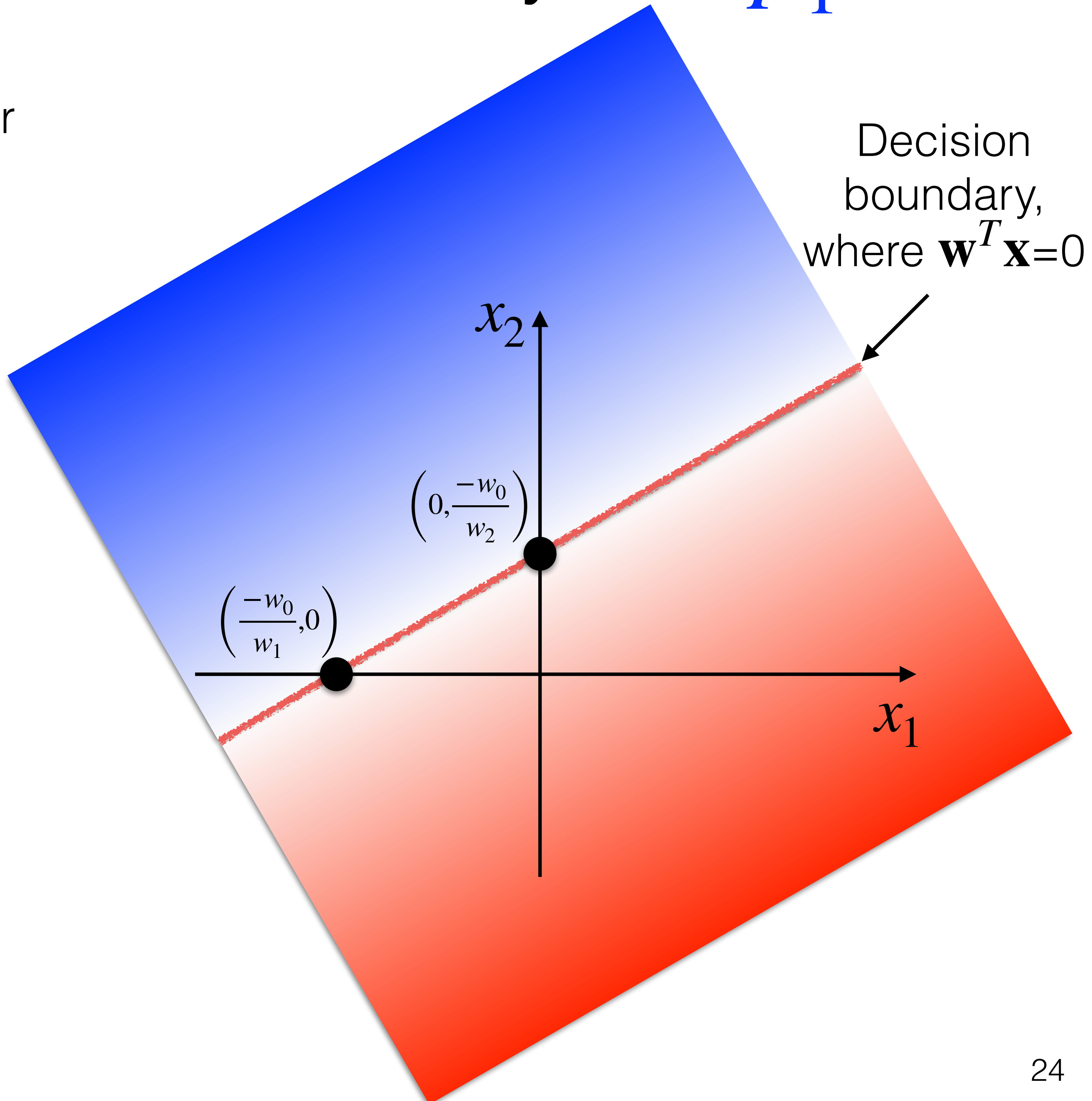




The Relationship Between the Distance To The Decision Boundary and p_1

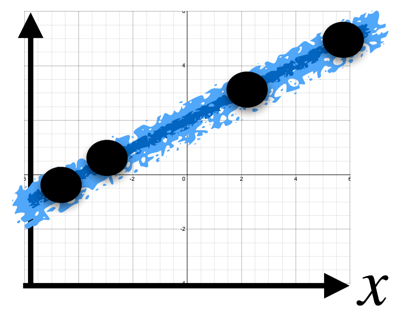
- The larger $|\mathbf{w}^T\mathbf{x}|$, the further away from the decision boundary the example $\mathbf{x}$ is.
- The larger $\mathbf{w}^T\mathbf{x}$, the higher p_1.
- The more negative $\mathbf{w}^T\mathbf{x}$, the smaller the p_1 (and the larger the p_0).





Components of the Supervised Learning Process in View of Noise

$f(x) = 0.5x + 2 + noise$



Unknown Target Distribution

$p(y \mid \mathbf{x})$

Unknown Input Distribution

$p(\mathbf{x})$

$\mathbf{x}$

Training Examples

$T = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$

Learning Algorithm

$\mathscr{A}$

Final Hypothesis

$g \approx f$

$T = \{(4,4), (0,2), (8,6), (2,3)\}$

$T = \{(4.3,4), (0,2.2), (7.9,5.1), (2.3,3.1)\}$

$g(\mathbf{x})$

Hypothesis Set

$\mathscr{H}$

$h(\mathbf{x}) = \mathbf{a}^T\mathbf{x} + b, \quad \forall \mathbf{a} \in R^d, b \in R$

Hypothesis Set

- $\text{logit}(p_1) = \mathbf{w}^T\mathbf{x}$
  - $\mathbf{w}^T\mathbf{x} \geq 0 \rightarrow$ class 1
  - $\mathbf{w}^T\mathbf{x} < 0 \rightarrow$ class 0
- If we solve $\text{logit}(p_1) = \mathbf{w}^T\mathbf{x}$ for p_1 we get:

$$p_1 = \frac{e^{(\mathbf{w}^T\mathbf{x})}}{1 + e^{(\mathbf{w}^T\mathbf{x})}}$$

$$p_0 = 1 - p_1 = \frac{1}{1 + e^{(\mathbf{w}^T\mathbf{x})}}$$

- $p_1 \geq 0.5 \rightarrow$ class 1
- $p_1 < 0.5 \rightarrow$ class 0

$$h(\mathbf{x}) = \begin{cases} 1 & \text{if } \text{logit}(p_1) \geq 0 \\ 0 & \text{otherwise} \end{cases}, \quad \forall \mathbf{w} \in R^{d+1}$$

$$h(\mathbf{x}) = \begin{cases} 1 & \text{if } p_1 = p(1 \,|\, \mathbf{x}, \mathbf{w}) \geq 0.5 \\ 0 & \text{otherwise} \end{cases}, \quad \forall \mathbf{w} \in R^{d+1}$$

$$h(\mathbf{x}) = p_1 = p(1 \,|\, \mathbf{x}, \mathbf{w}), \quad \forall \mathbf{w} \in R^{d+1}$$

Summary

- Supervised learning aims at learning a function g that generalises well to examples from the underlying $p(\mathbf{x}, y)$ of the problem.

- Logistic regression models $\text{logit}(p_1)$ as a linear combination of the input variables, $\text{logit}(p_1) = \mathbf{w}^T\mathbf{x}$.

- The probability p_1 is thus modelled by a sigmoid function $p_1 = \dfrac{e^{(\mathbf{w}^T\mathbf{x})}}{1 + e^{(\mathbf{w}^T\mathbf{x})}}$.

- The hypothesis set can be seen as $h(\mathbf{x}) = p_1 = p(1 \mid \mathbf{x}, \mathbf{w}), \quad \forall \mathbf{w} \in R^{d+1}$.

- The parameters to be learned are $\mathbf{w}$.

- Next: How to learn $\mathbf{w}$?

Further Reading

The reading materials can be found at the module's resource list and elsewhere on the web, except for Iain Style's notes, which can be found in the links provided below.

Essential reading: Abu-Mostafa et al.'s Learning from Data: A Short Course. Section 1.1.1 (Components of Learning), Section 1.4.2 (Noisy Targets), Section 3.3 (Logistic Regression) until page 90. **==> Note that the authors are using -1 and +1 to represent the different categories, instead of 0 and 1 like in this lecture.**

Recommended reading: Iain Styles's Notes on Logistic Regression, Section 1 (Modelling the Logit):

https://canvas.bham.ac.uk/files/15585285/download?download_frd=1