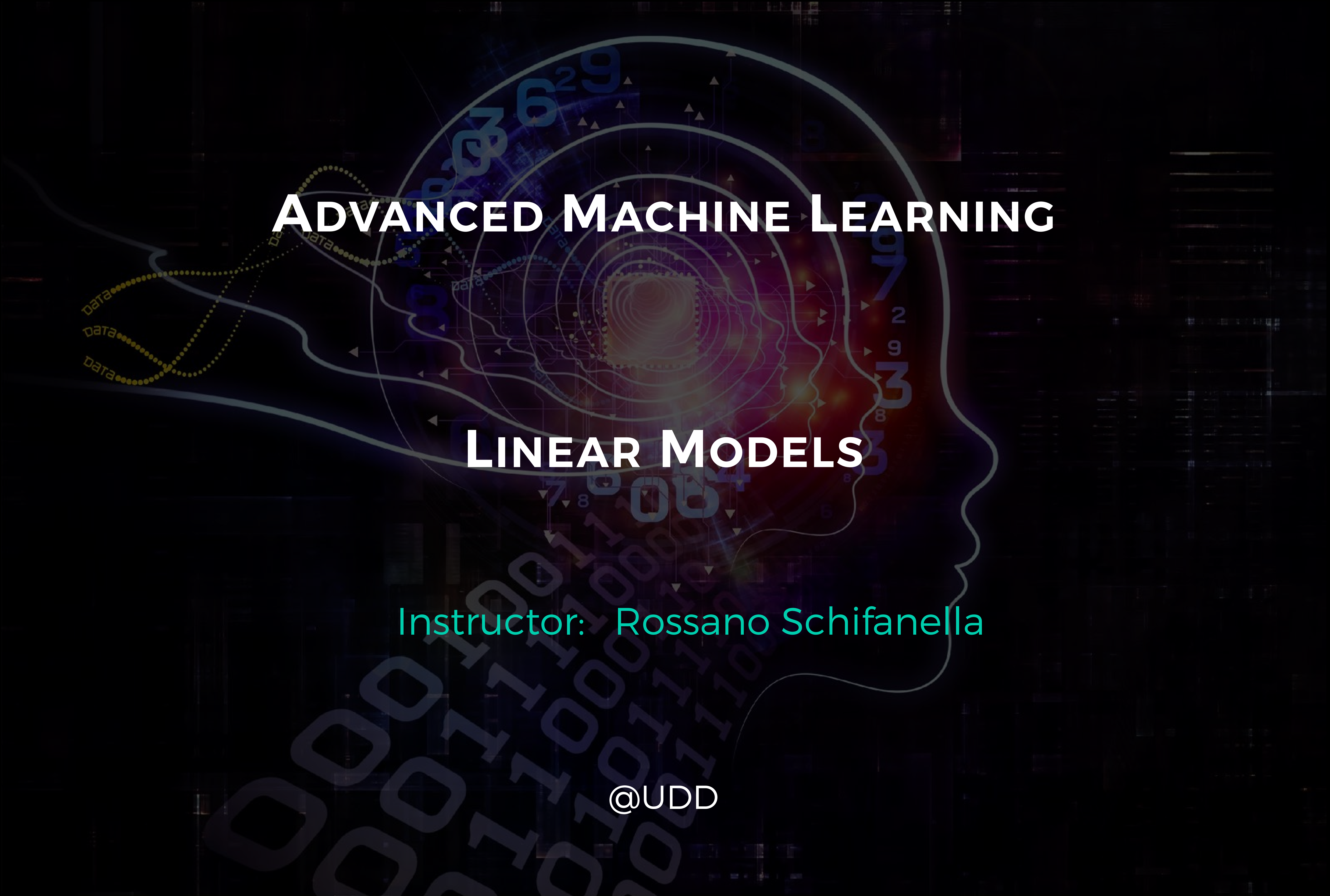


Advanced Machine Learning

Linear Models

Instructor: Rossano Schifanella

@UDD

Simplifying Assumptions

1. For **classification** and **class probability estimation**, we will consider only **binary classes**

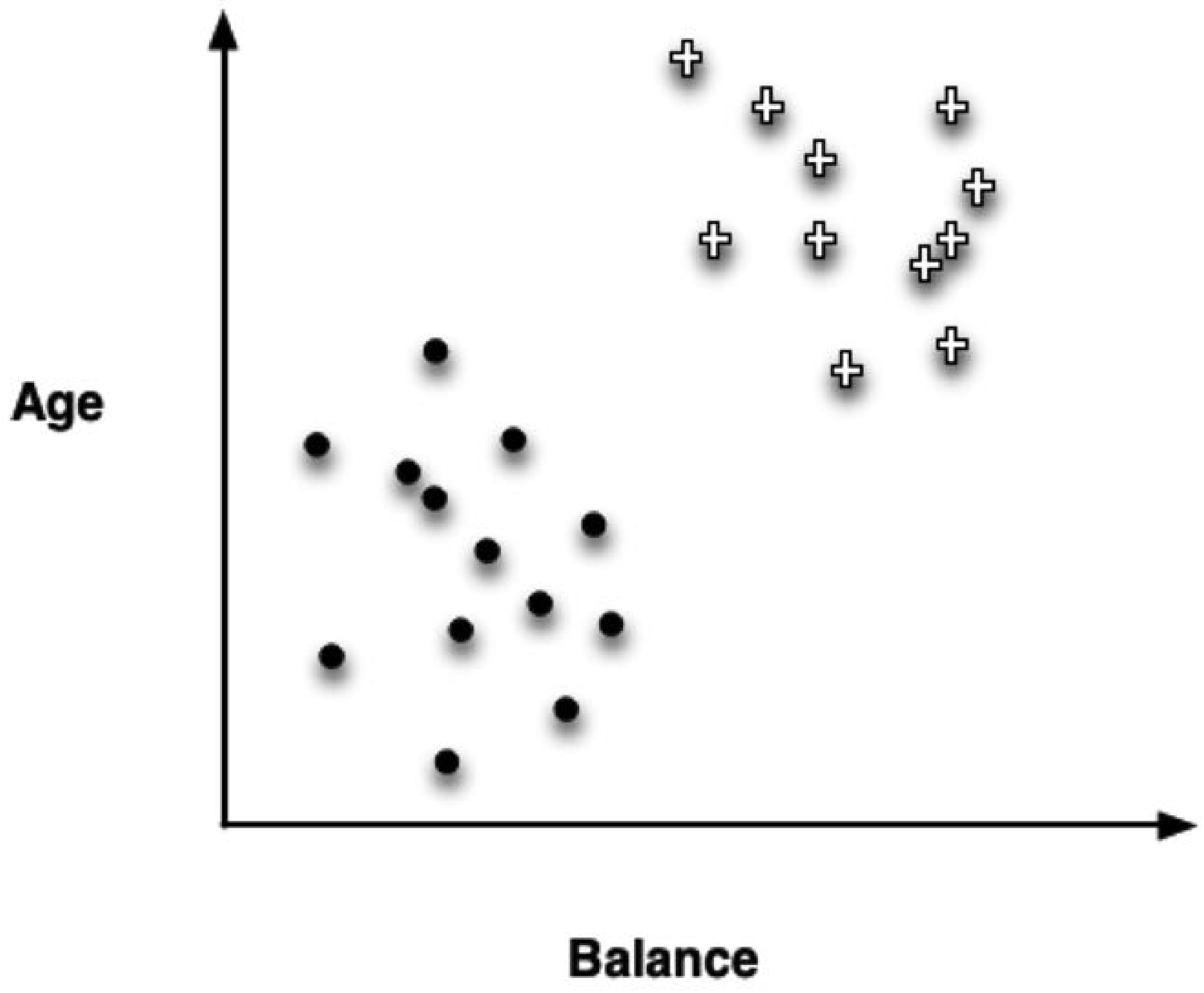
2. We assume that **all attributes are numeric**.

3. We **ignore the need to normalize** numeric measurements to a common scale

Classification

+ and **o** values of a **binary target variable**

Age and **Balance** are (numerical) **features**

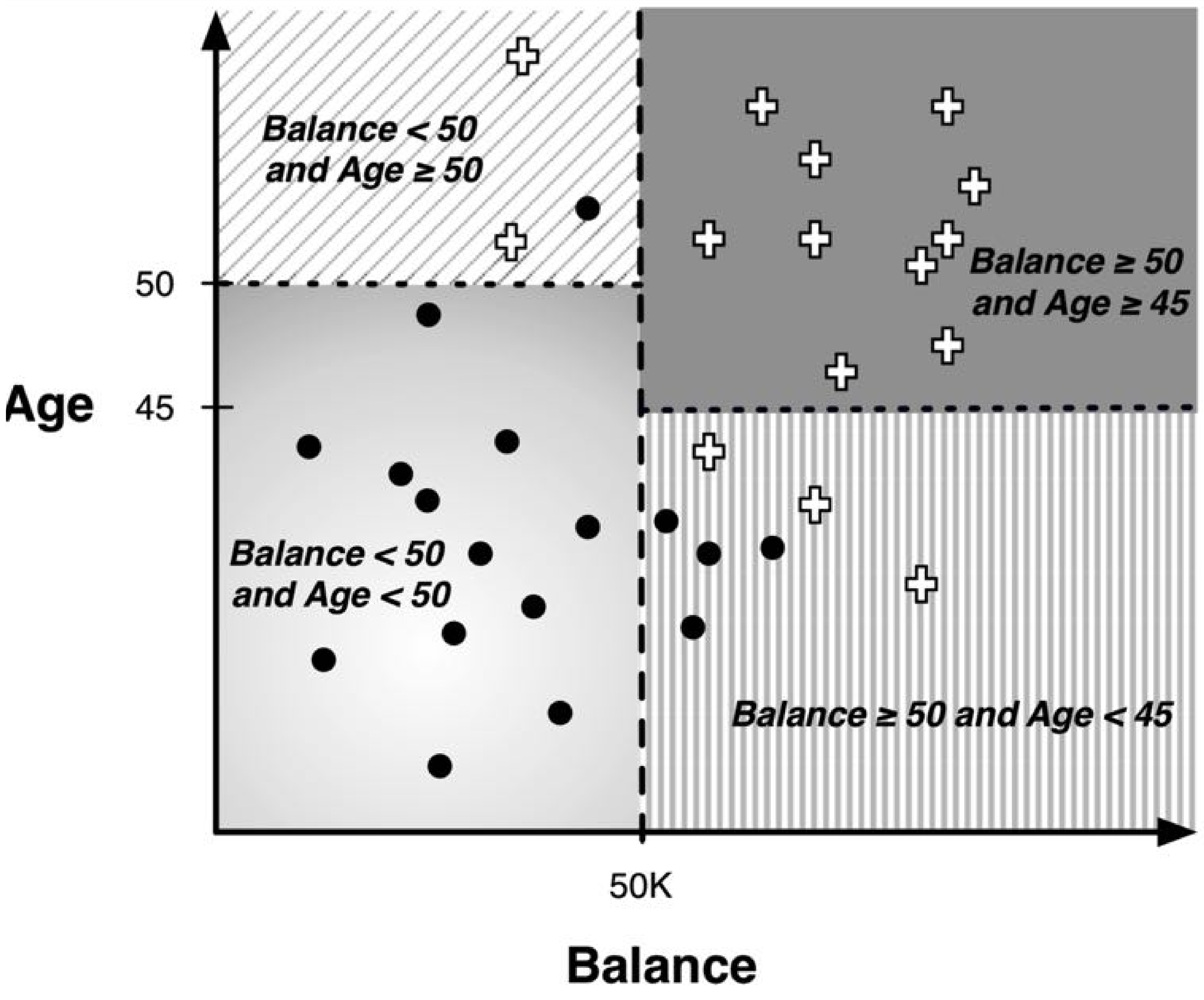


Classification with Decision Trees

+ and **o** values of a **binary target variable**

Age and **Balance** are (numerical) **features**

Dataset split by a tree with four leaves



Classification with a Linear Discriminant

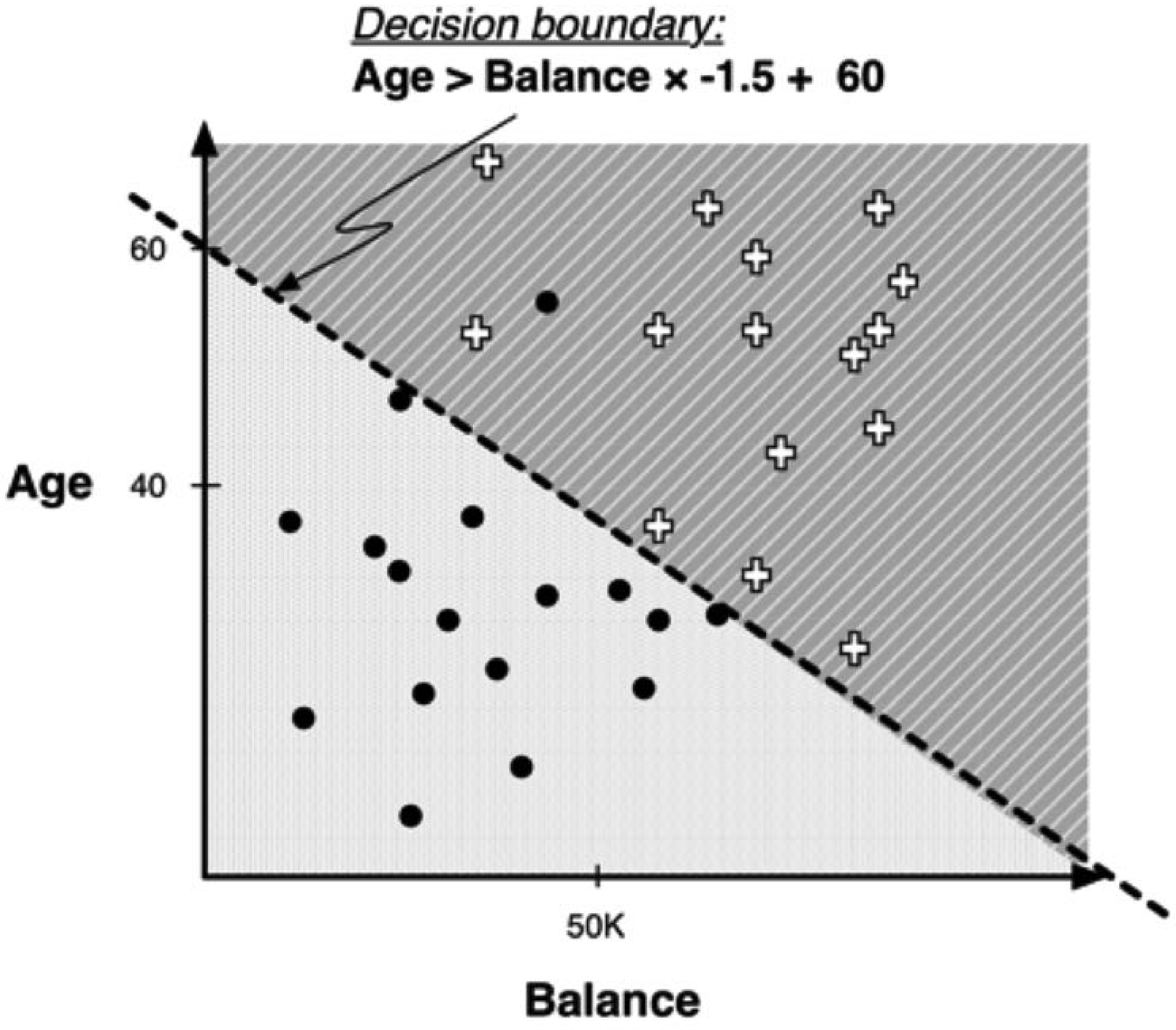
+ and **o** values of a **binary target variable**

Age and **Balance** are (numerical) **features**

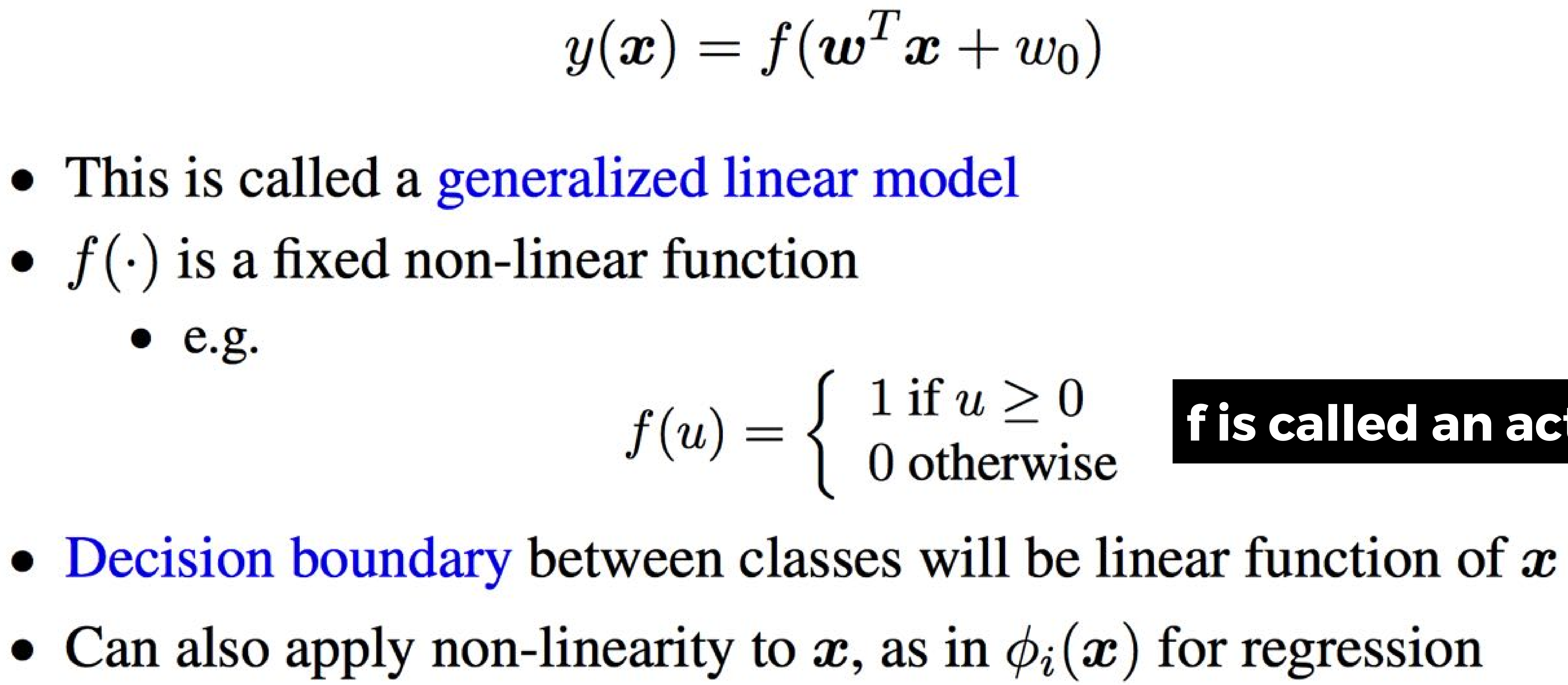
Dataset with a linear split

$$Age = (-1.5) \times Balance + 60$$

$$class(\mathbf{x}) = \begin{cases} + \text{ if } 1.0 \times Age - 1.5 \times Balance + 60 > 0 \\ \bullet \text{ if } 1.0 \times Age - 1.5 \times Balance + 60 \leq 0 \end{cases}$$



Generalized Linear Model



$$y(\boldsymbol{x}) = f(\boldsymbol{w}^T\boldsymbol{x} + w_0)$$

- This is called a generalized linear model
- $f(\cdot)$ is a fixed non-linear function
  - e.g.

$$f(u) = \begin{cases} 1 \text{ if } u \geq 0 \\ 0 \text{ otherwise} \end{cases}$$

f is called an activation function

- Decision boundary between classes will be linear function of $\boldsymbol{x}$
- Can also apply non-linearity to $\boldsymbol{x}$, as in $\phi_i(\boldsymbol{x})$ for regression

Φᵢ is something called a kernel function

Discriminant Functions with 2 classes

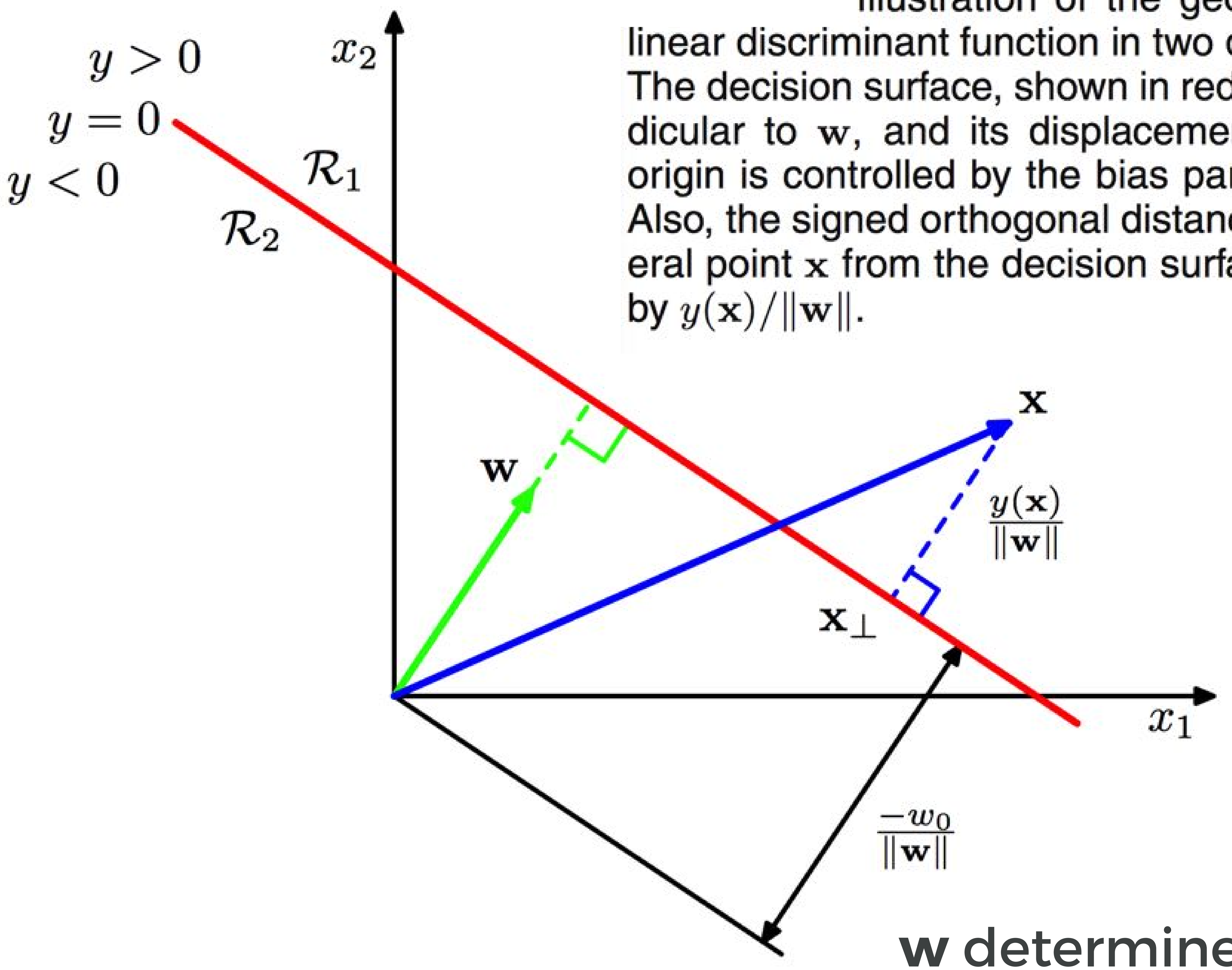


Illustration of the geometry of a linear discriminant function in two dimensions. The decision surface, shown in red, is perpendicular to $\mathbf{w}$, and its displacement from the origin is controlled by the bias parameter w_0. Also, the signed orthogonal distance of a general point $\mathbf{x}$ from the decision surface is given by $y(\mathbf{x})/\|\mathbf{w}\|$.

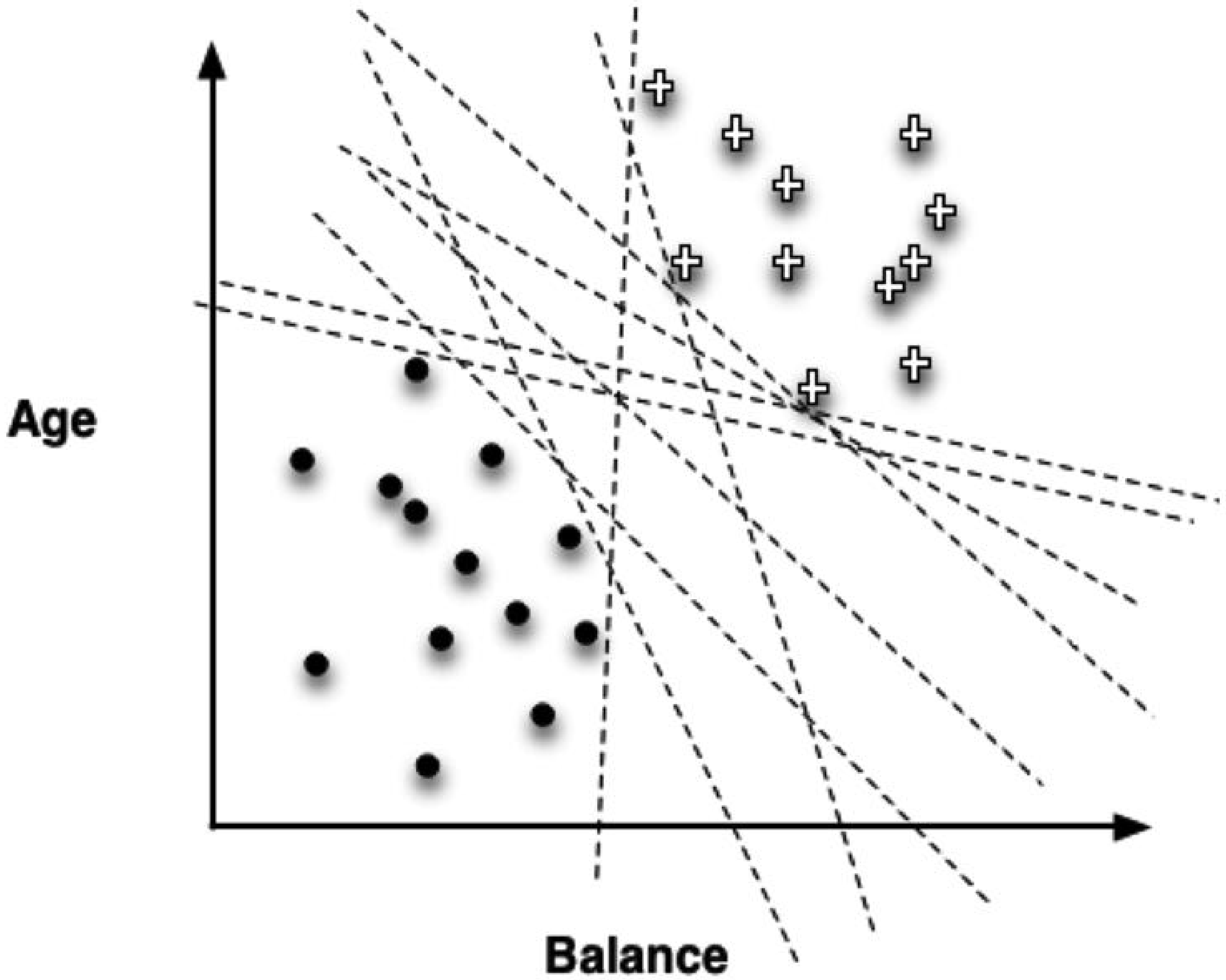
- Start with 2 class problem, $t_i \in \{0, 1\}$
- Simple linear discriminant

$$y(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + w_0$$

apply threshold function to get classification

w determines the **orientation** of the decision surface
(w is orthogonal to the decision surface)
bias parameter **w₀** determines the **location** of the decision surface

Many Linear Discriminants



How to select the best?
Optimize an objective function!

Ordinary Least Squares (OLS)

- **Explanatory** and **Response** Variables are numeric
- Relationship between the mean of the response variable and the level of the explanatory variable assumed to be approximately linear (straight line)

Model:

$$\mathbf{Y = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \varepsilon}$$

vector form

$$y_i = x_i^T \beta + \varepsilon$$

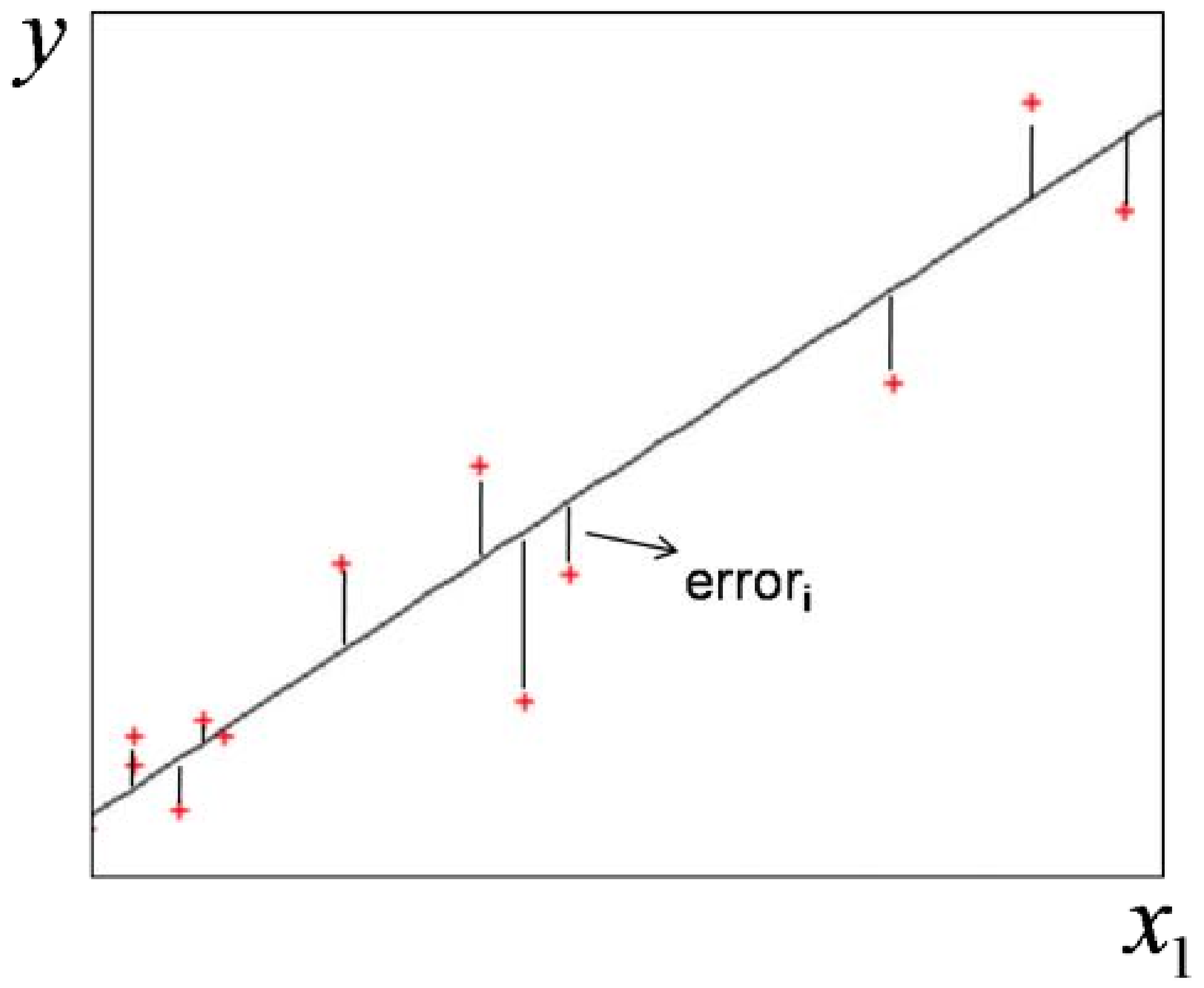
$\beta_1 > 0 \Rightarrow$ Positive Association

$\beta_1 < 0 \Rightarrow$ Negative Association

$\beta_1 = 0 \Rightarrow$ No Association

Ordinary Least Squares (OLS)

residual



$y_i - x_i^T b$ where b is a candidate for the β vector

measures the vertical distance between the data point (x_i y_i) and the hyperplane $y = x^T b$

Intelligibility

- One nice advantage of linear regression models (and linear classification) is the **potential to look at the coefficients to give insight into which input variables are most important in predicting the output**
- The variables with the largest magnitude have the highest correlation with the output
  - A large positive coefficient implies that the output will increase when this input is increased (positively correlated)
  - A large negative coefficient implies that the output will decrease when this input is increased (negatively correlated)
  - A small or 0 coefficient suggests that the input is uncorrelated with the output (at least at the 1st order)
- Linear regression can be used to find best "indicators"
- However, **be careful not to confuse correlation with causality**

Disadvantage of OLS

very sensitive to the data: erroneous or otherwise outlying data points can severely skew the resultant linear function.

Outliers

Least Squares vs Logistic Regression

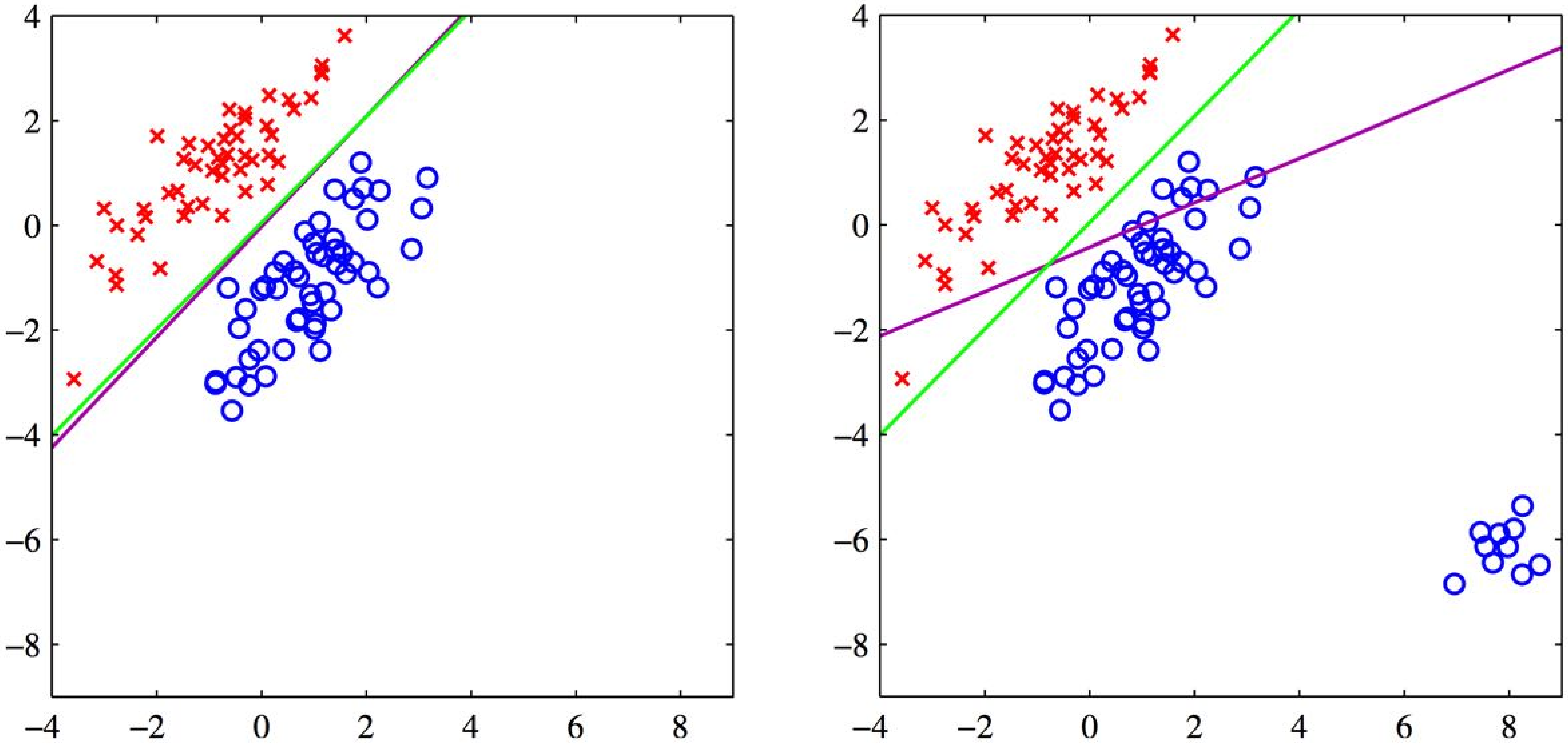


Figure 4.4 The left plot shows data from two classes, denoted by red crosses and blue circles, together with the decision boundary found by least squares (magenta curve) and also by the logistic regression model (green curve), which is discussed later in Section 4.3.2. The right-hand plot shows the corresponding results obtained when extra data points are added at the bottom left of the diagram, showing that least squares is highly sensitive to outliers, unlike logistic regression.

Multi-class Example

Least Squares vs Logistic Regression

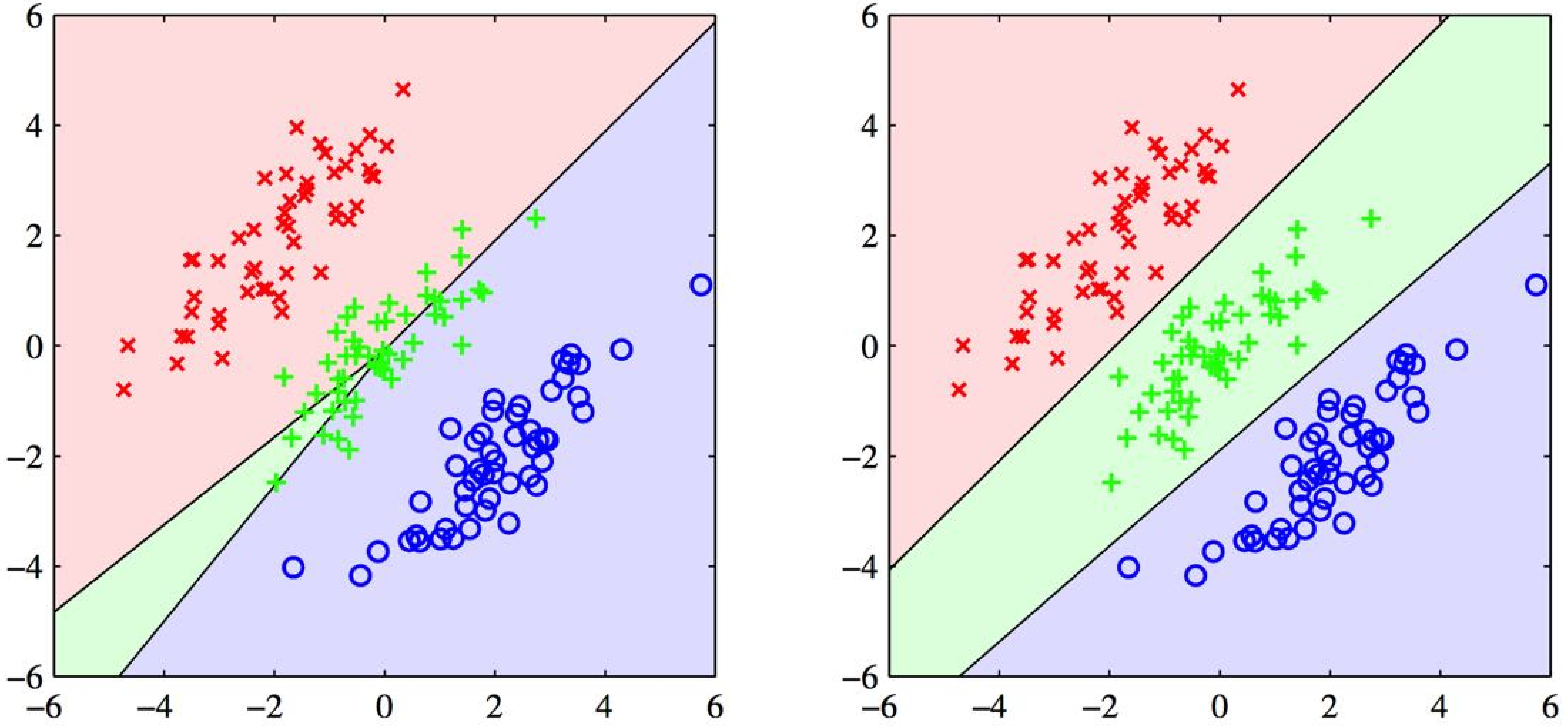
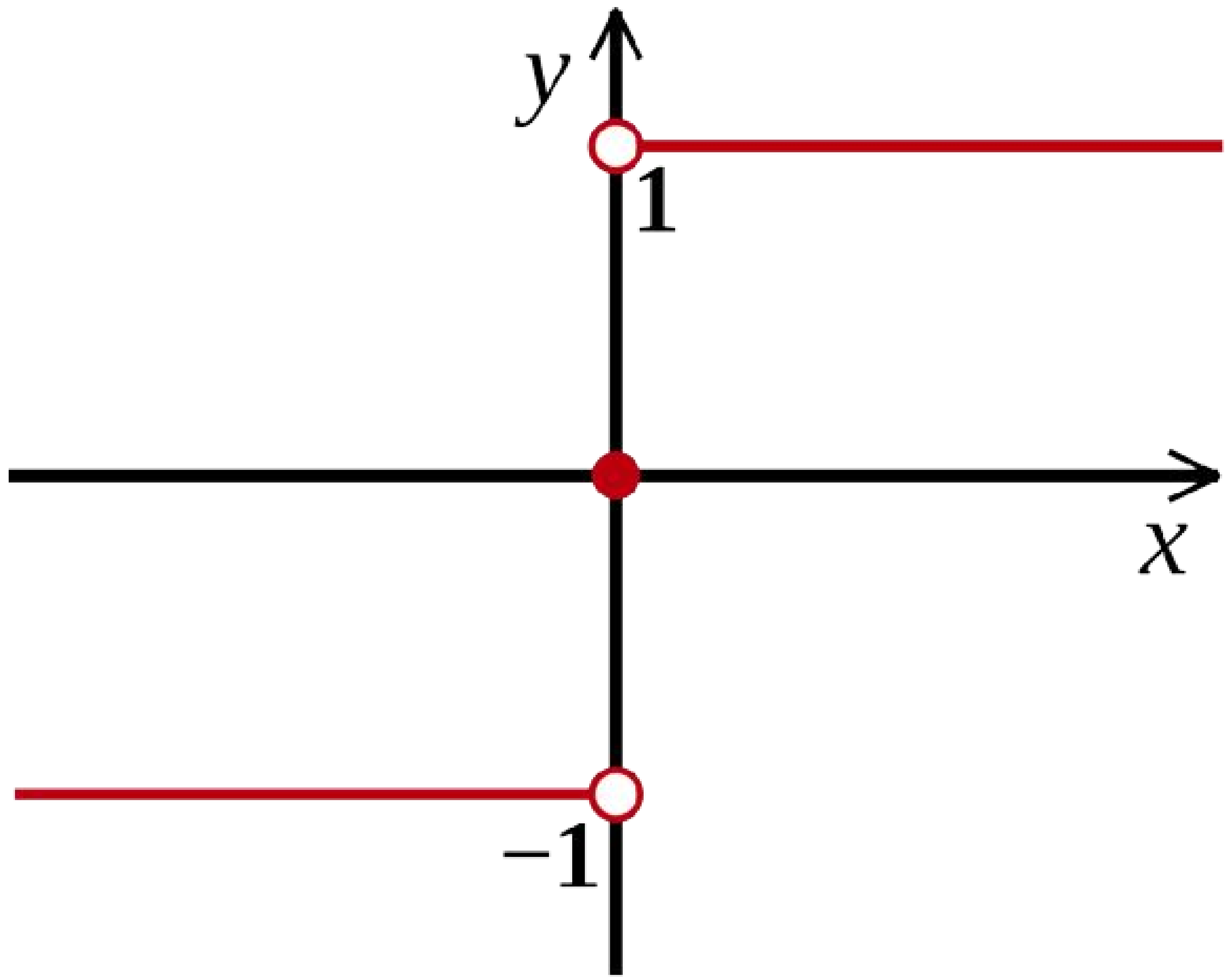


Figure 4.5 Example of a synthetic data set comprising three classes, with training data points denoted in red (×), green (+), and blue (○). Lines denote the decision boundaries, and the background colours denote the respective classes of the decision regions. On the left is the result of using a least-squares discriminant. We see that the region of input space assigned to the green class is too small and so most of the points from this class are misclassified. On the right is the result of using logistic regressions as described in Section 4.3.2 showing correct classification of the training data.

Convert a Problem into Classification: Activation Function

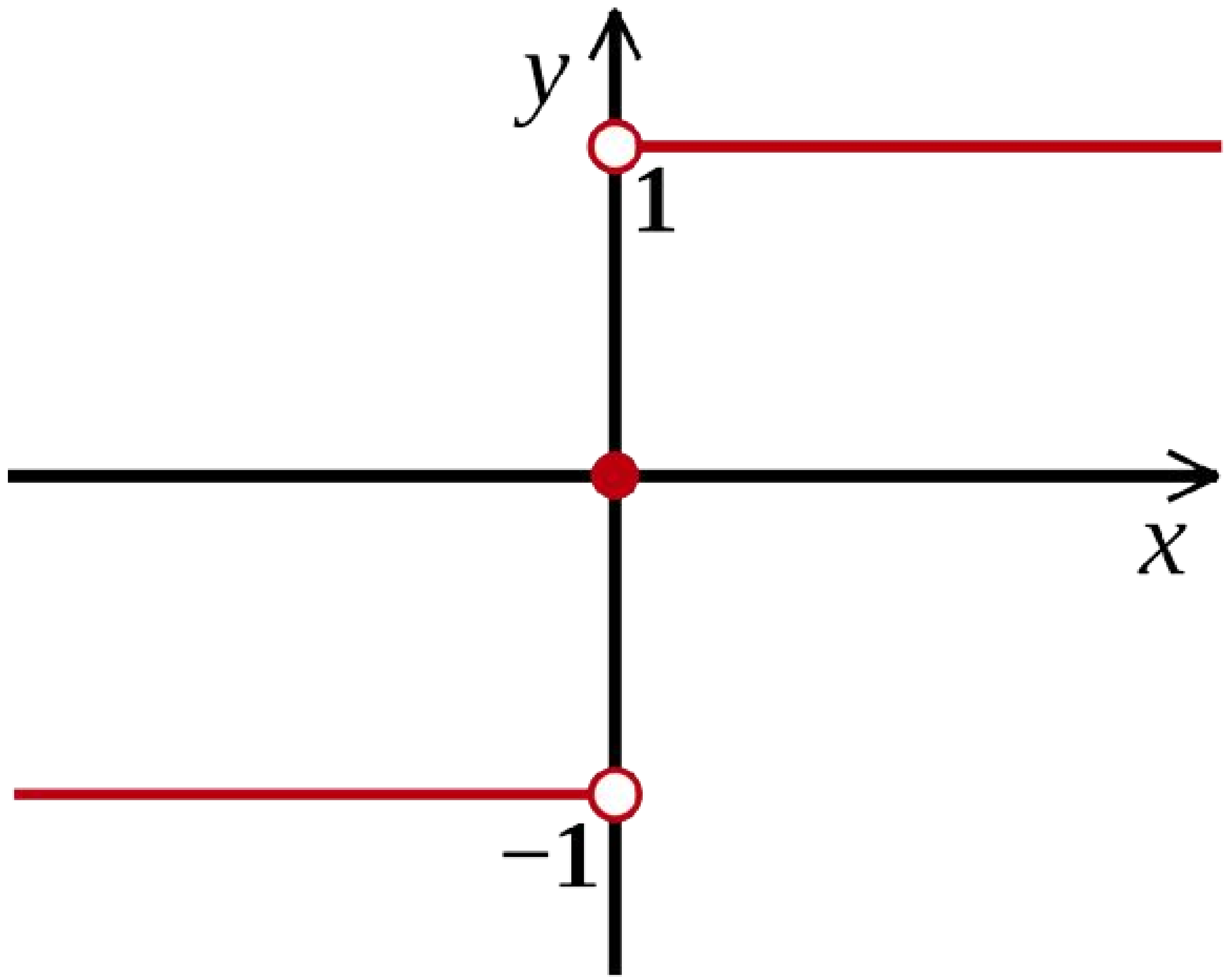
- **From continuous values we need to get a binary value.**
  - For example: **Sign function** for classification



$$\operatorname{sgn}(x) = \begin{cases} -1 & \text{if } x < 0 \\ 0 & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases}$$

Convert a Problem into Classification: **Activation Function**

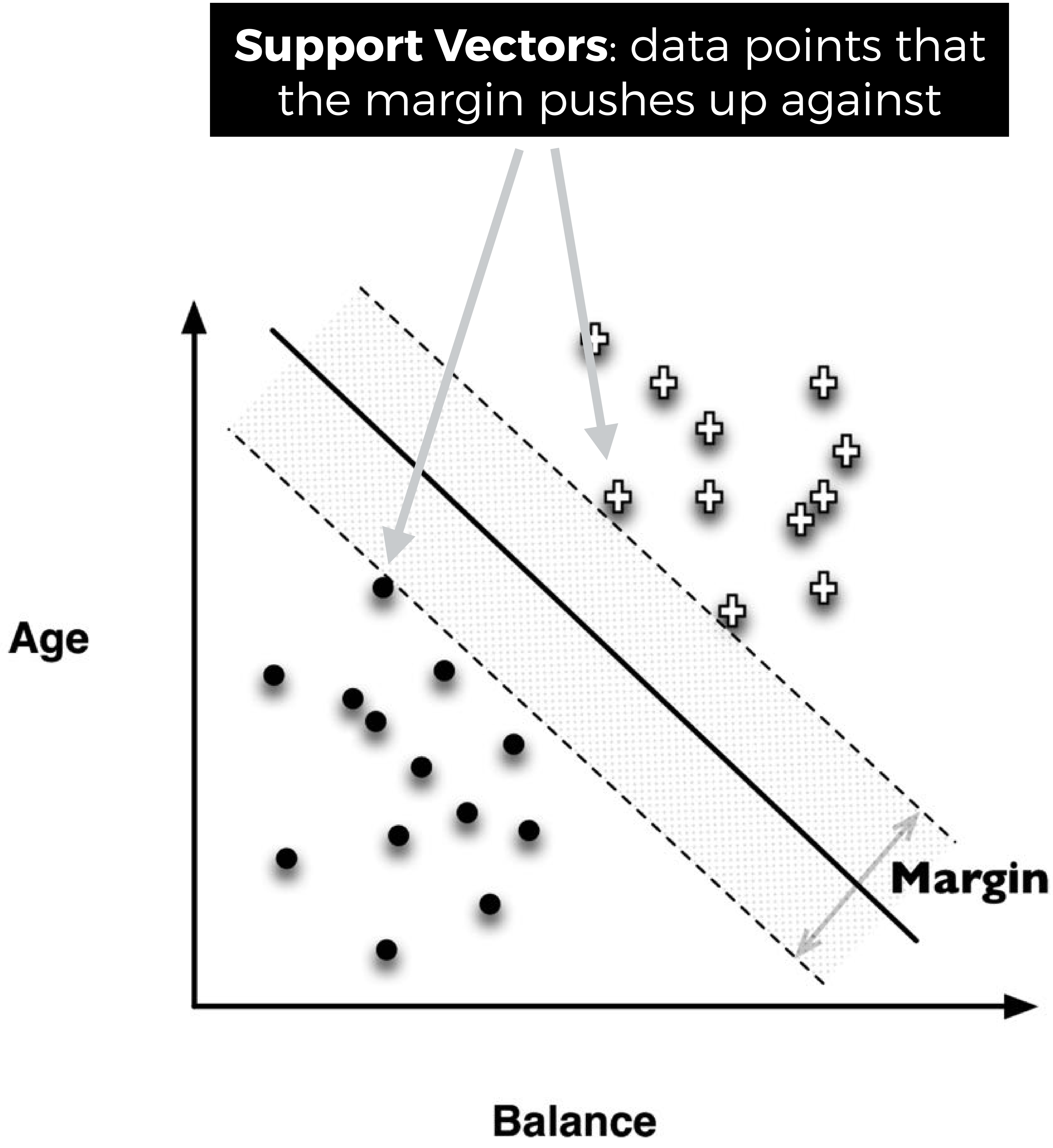
- We have now a **parametrised model**
  - Decision Trees and K-NN weren't!
- We have to **learn** the parameters from the training set



$$\operatorname{sgn}(x) = \begin{cases} -1 & \text{if } x < 0 \\ 0 & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases}$$

Support Vector Machine (SVM)

- In short, SVMs are **linear discriminants**.
- What is the **objective function** that is used to fit an SVM to data?
- Maximize **margin**
- **penalize a training point** for being on the wrong side of **the decision boundary**

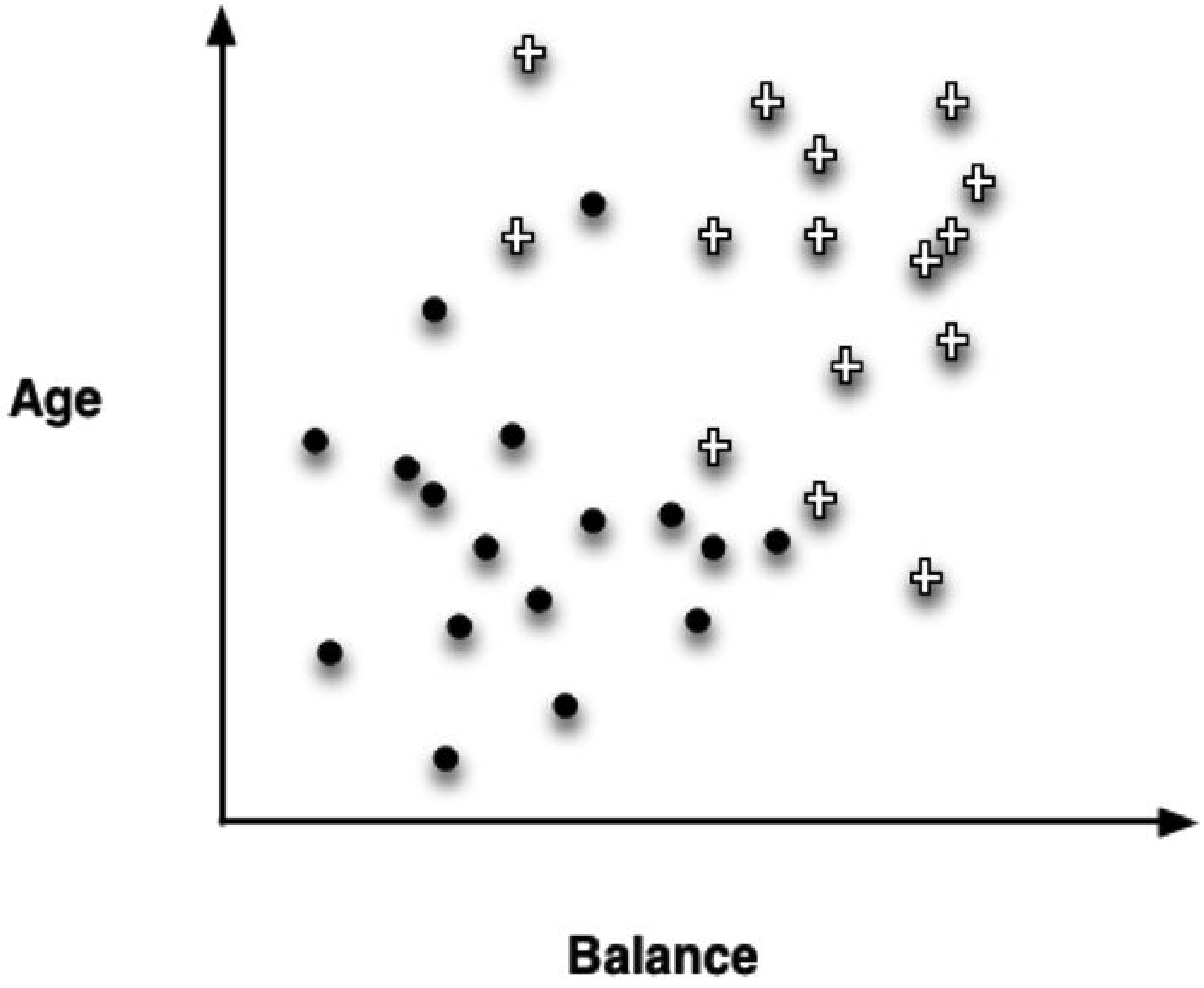


Support Vectors: data points that the margin pushes up against

Support Vector Machine (SVM)

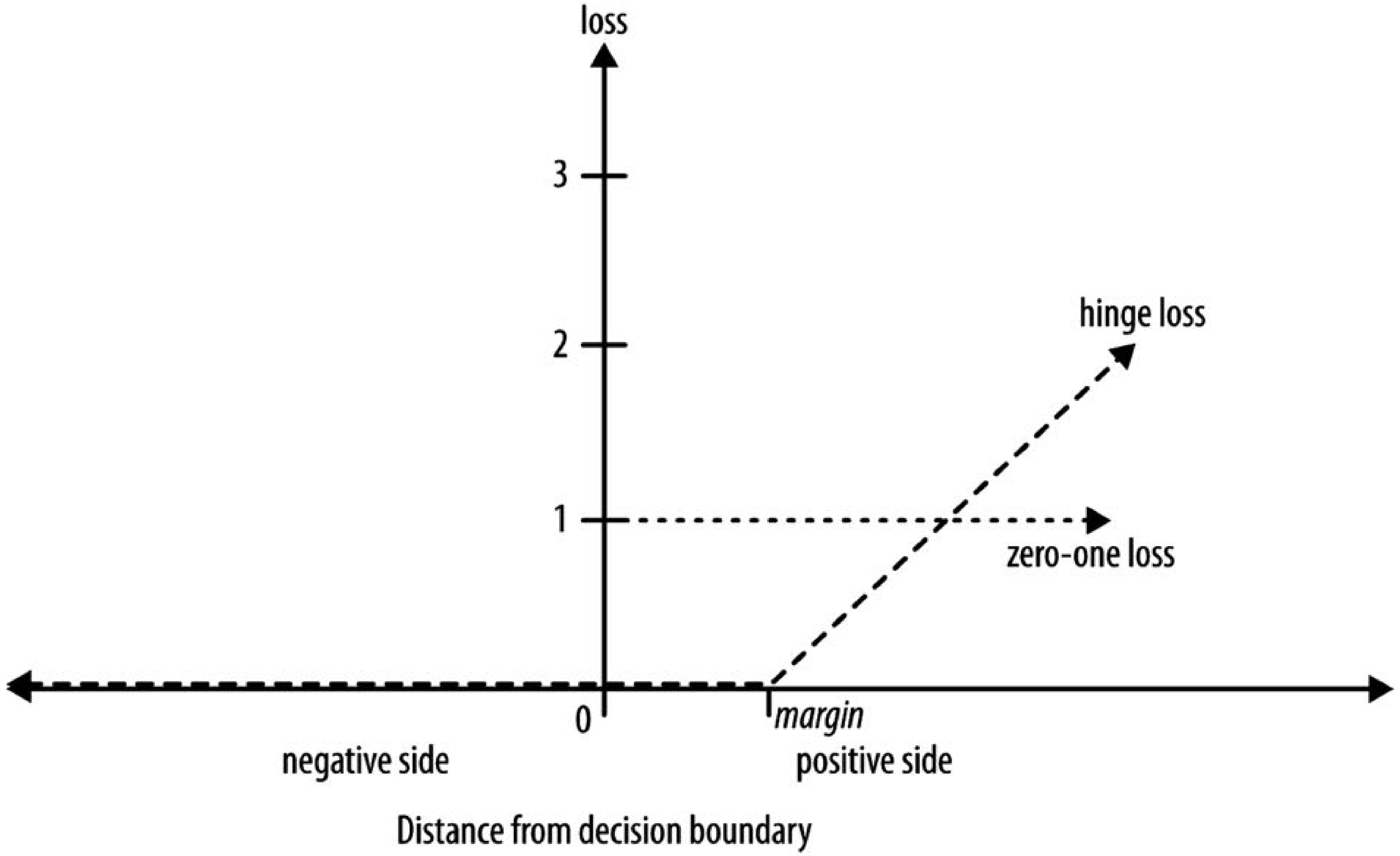
- The best fit is a **balance between a fat margin and a low total error penalty**
- The penalty for a misclassified point is **proportional to the distance** from the decision boundary
  - Technically, error function is a **hinge loss** (you can have different functions though)

not linearly separable



Loss functions

- Determines how much **penalty** should be assigned to an instance based on the **error in the model's predicted value**
  - in our present context, based on its distance from the separation boundary



SVM: Linearly Separable

y=1 (full circles), if **w·x-b>0**

y=-1 (empty circles), if **w·x-b<0**

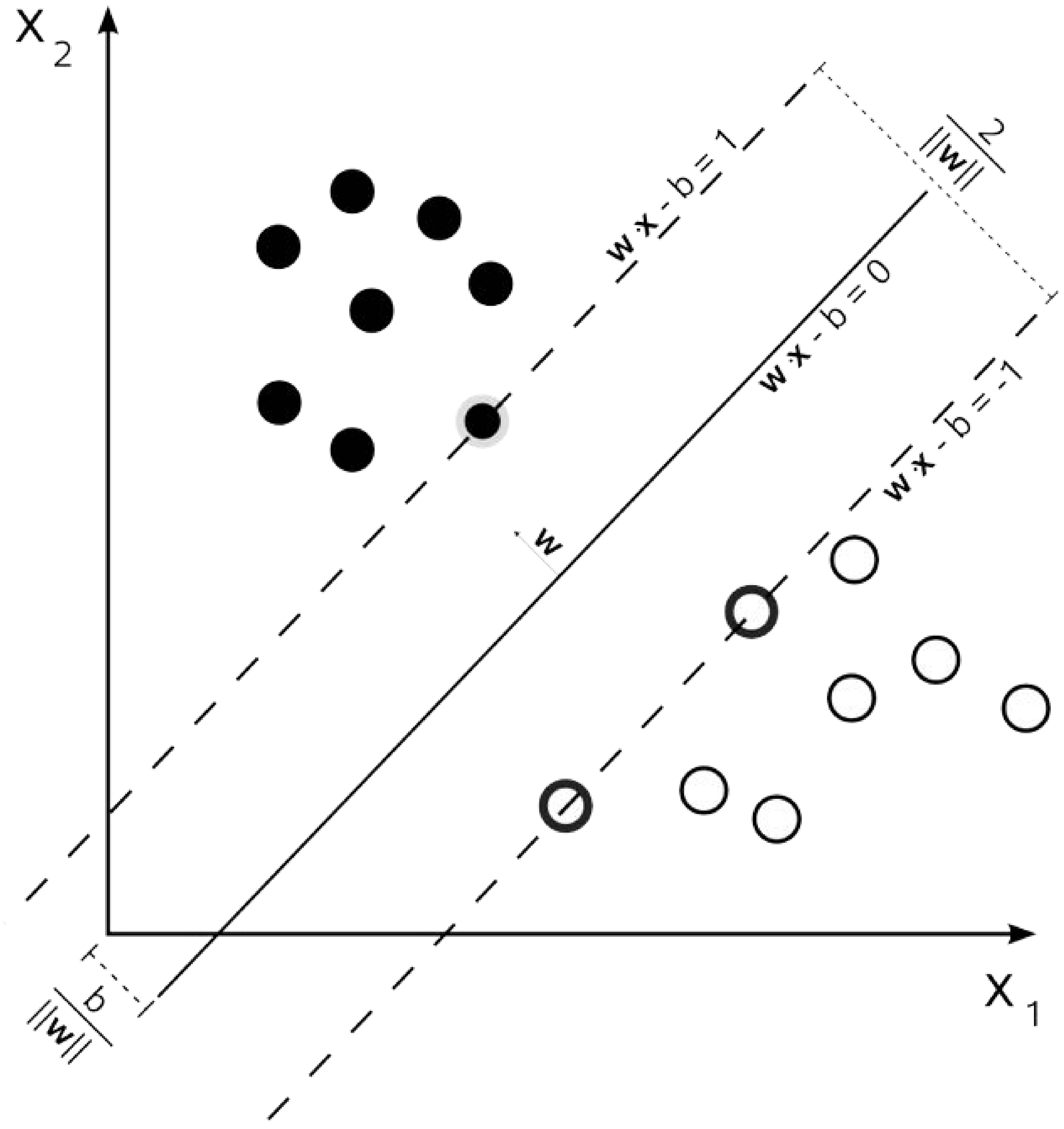
if $\mathbf{x_1}$ and $\mathbf{x_2}$ are the two **closest support vectors**, then:

Maximizing the margin is equivalent to minimizing ||w||

Final optimization problem:

1. minimize ||w||

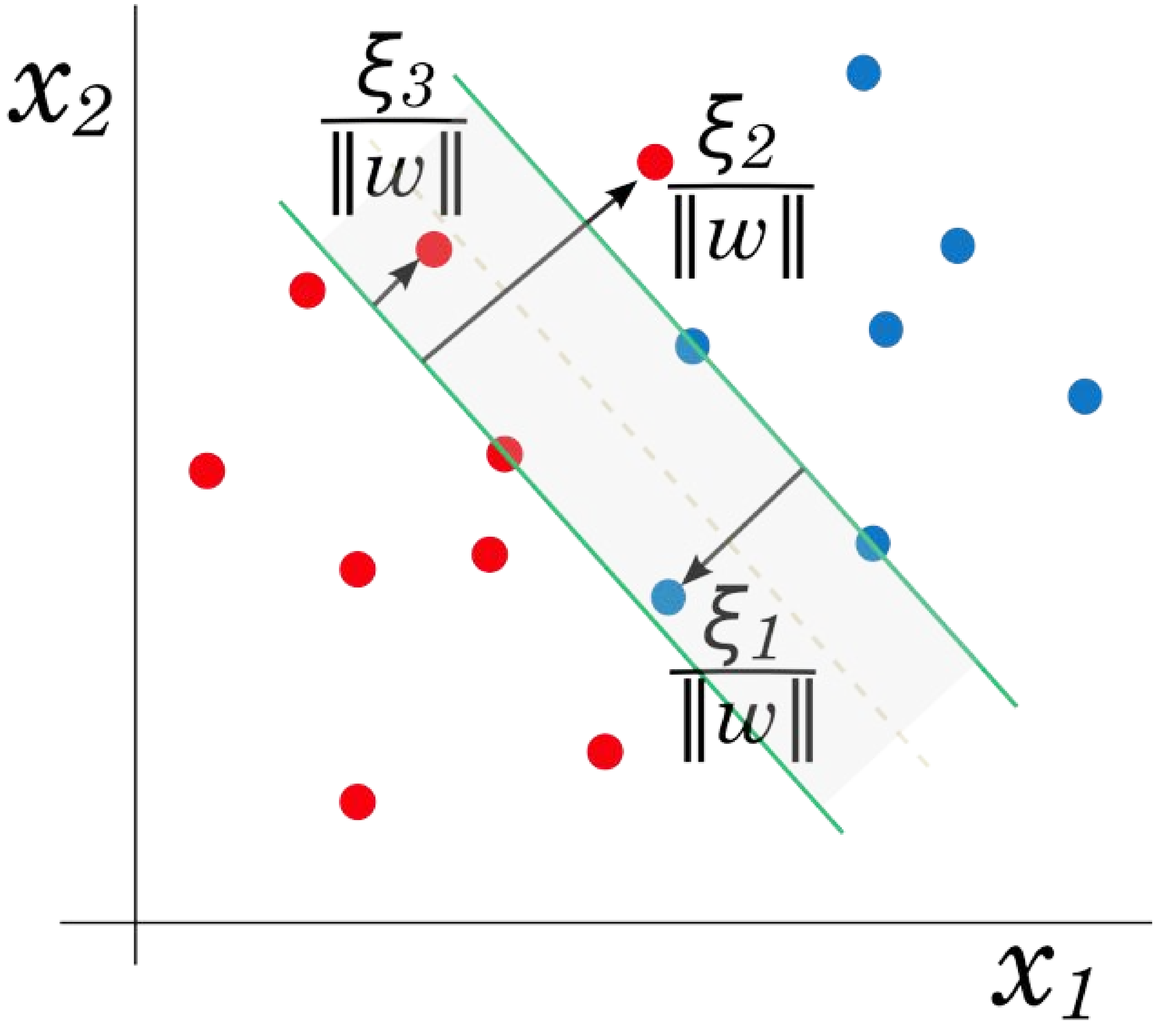
2. y_i (w·x_i - b) ⩾ 1
(correct classification on the training set)



SVM: Not Linearly Separable

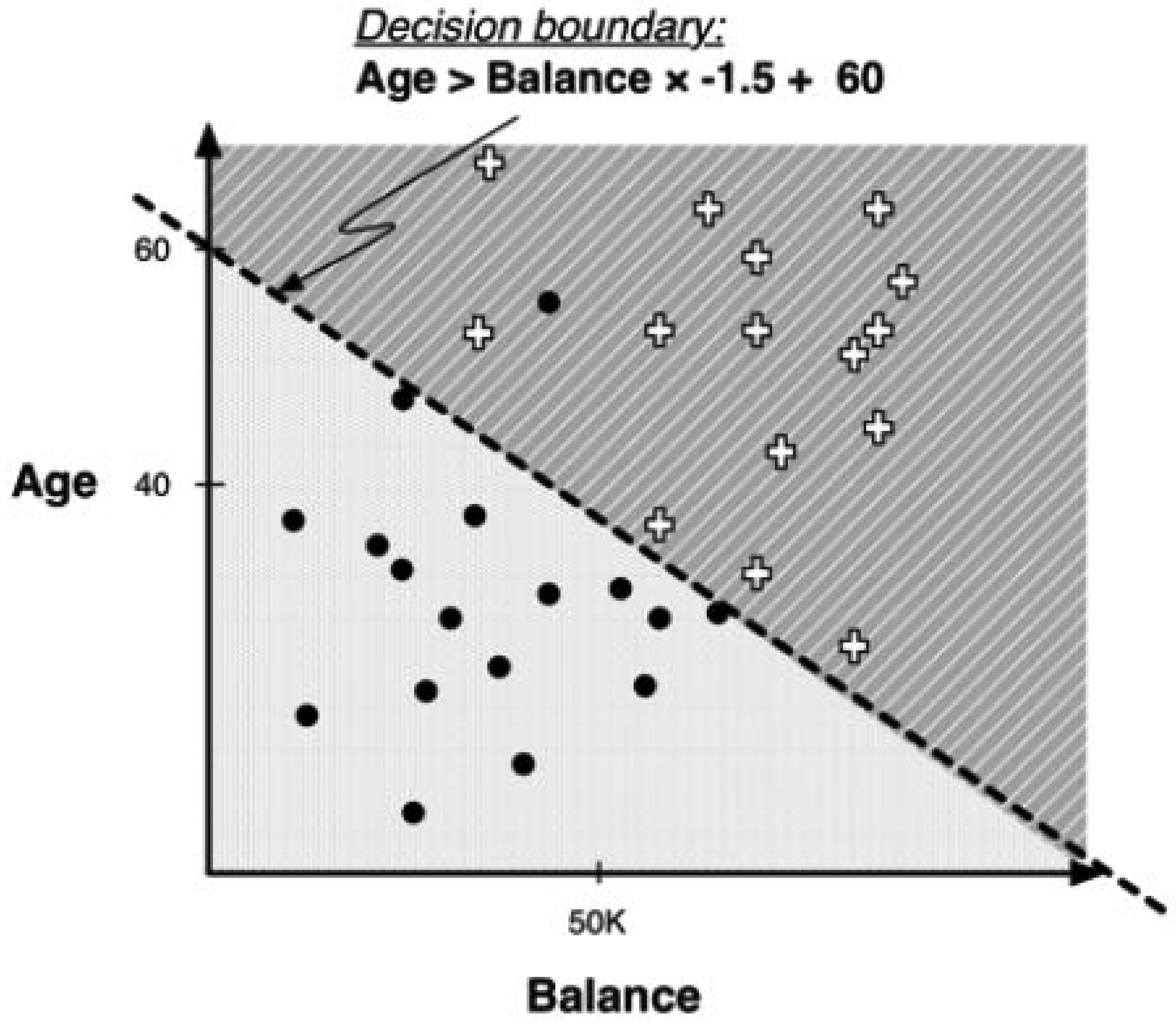
We give a **penalty** (or loss) to points that **fall in the margin or on the wrong side**

$\boldsymbol{wx}_i + b > 1 - \xi_i$ and $\boldsymbol{wx}_i + b \leq -1 + \xi_i$



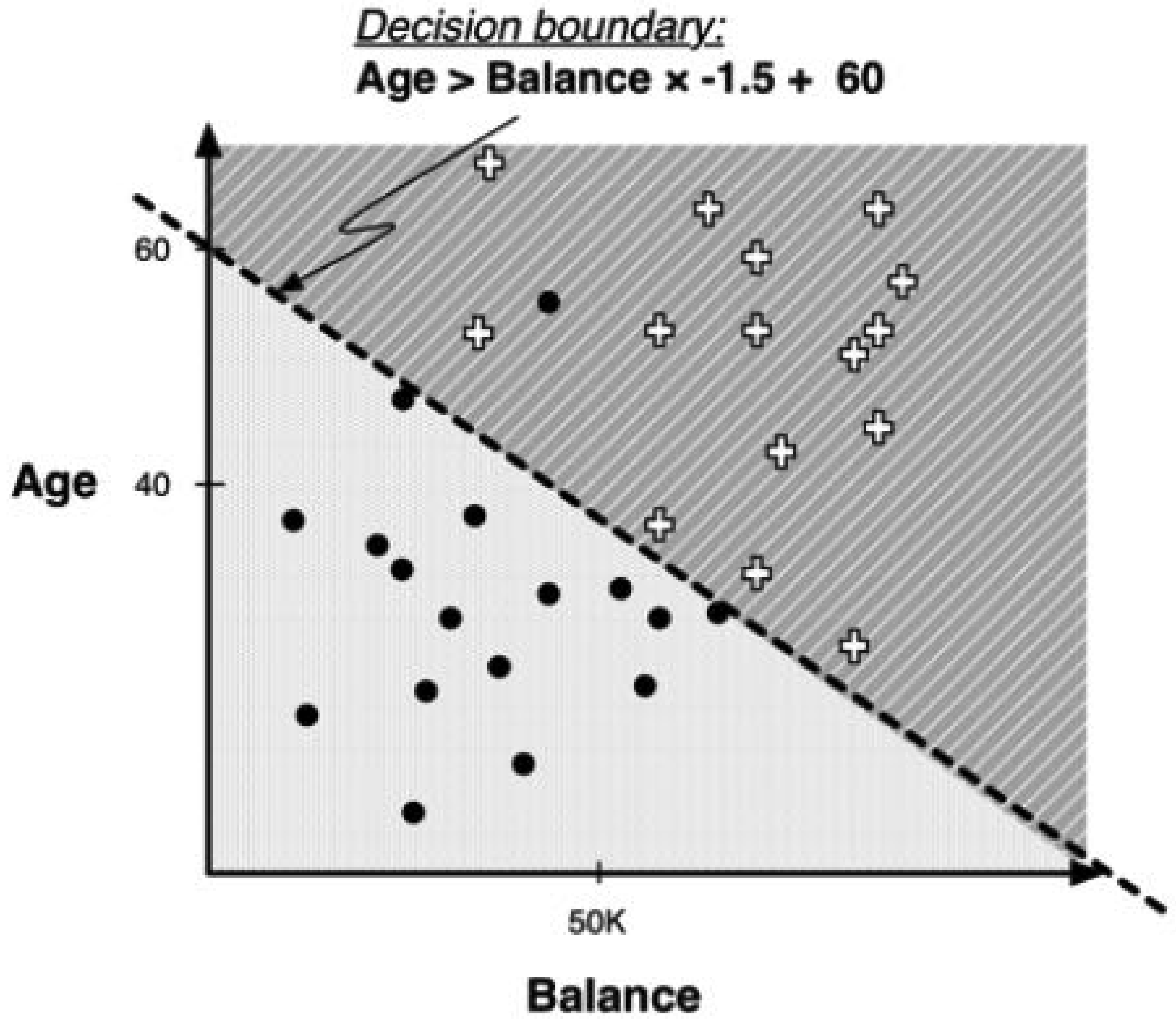
Linear Discriminant Functions for Scoring and Ranking

- **Near the decision boundary we would be most uncertain about a class**. Far away from the decision boundary, on the + side we would expect the highest likelihood of response.
- **f(x) will be zero when x is sitting on the decision boundary f(x)** will be relatively small when x is near the boundary. And f(x) will be large (and positive) when x is far from the boundary in the + direction.

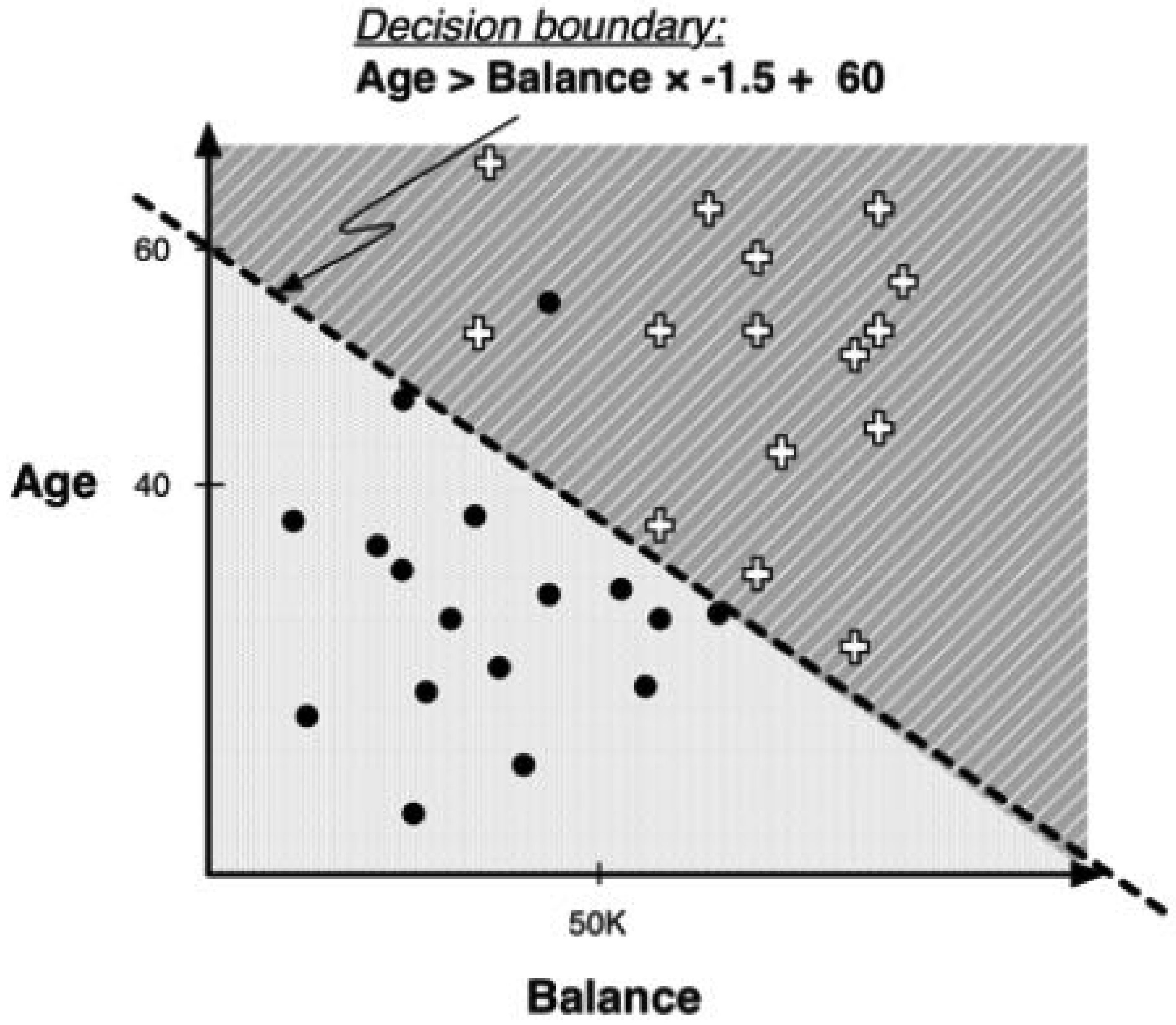


Linear Discriminant Functions for Scoring and Ranking

- f(x) itself gives an intuitively satisfying **ranking of the instances by their (estimated) likelihood of belonging to the class of interest**.



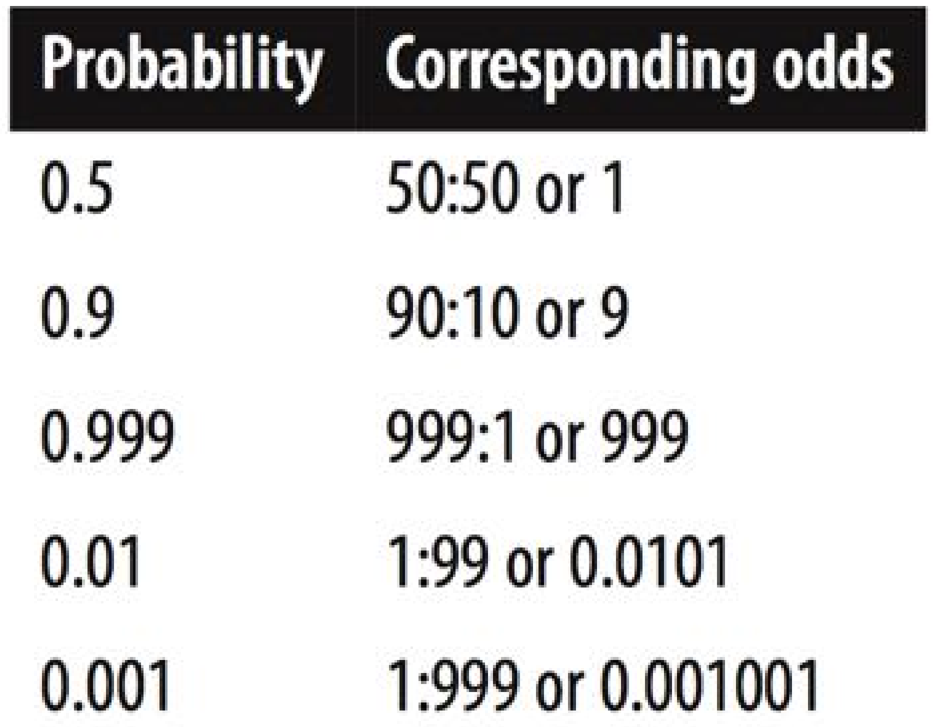
Logistic Regression



- Possible confusion:
  - **it is not really a regression**
  - **it is a linear discriminant too**
- Idea: we would like to estimate the **probability that a new instance belongs to the class of interest**
- The **higher the distance** from the boundary (on the right side) **the higher the probability** of belonging to the right class.
  - **This is not a probability**, because not between 0 and 1

Logistic Regression

- **Is there another representation of the likelihood of an event that we use in everyday life?**
- If we could come up with one that ranges from –∞ to ∞, then we might model this other notion of likelihood with our linear equation.
- One very useful notion of the **likelihood of an event** is the **odds**

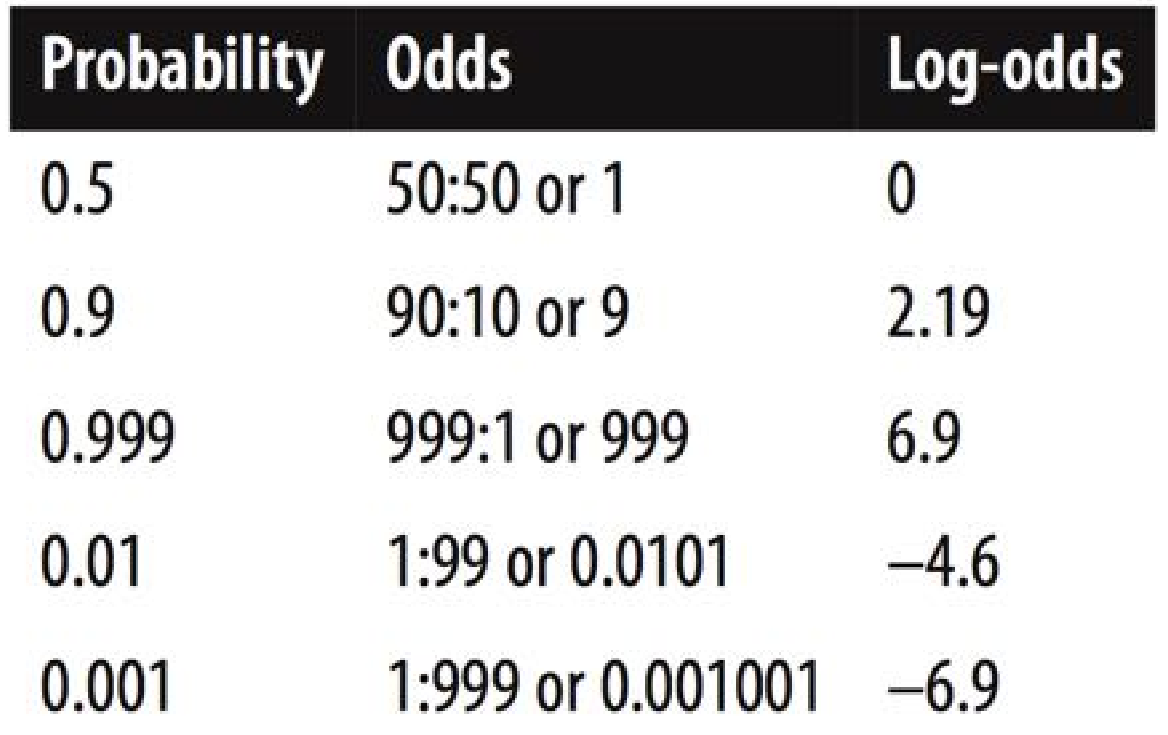


Probability	Corresponding odds
0.5	50:50 or 1
0.9	90:10 or 9
0.999	999:1 or 999
0.01	1:99 or 0.0101
0.001	1:999 or 0.001001

$$\frac{p}{1-p}$$

[0, ∞]

Logistic Regression



Probability	Odds	Log-odds
0.5	50:50 or 1	0
0.9	90:10 or 9	2.19
0.999	999:1 or 999	6.9
0.01	1:99 or 0.0101	–4.6
0.001	1:999 or 0.001001	–6.9

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

[-∞, ∞]

The same linear function f(x) that we've examined throughout the class is used as a measure of the log-odds (logit) of the "event" of interest

in other words, f(x) is the model's estimation of the log-odds that x belongs to the positive class

Logistic Regression (summary)

- For probability estimation, logistic regression uses the **same linear model** as do our linear discriminants for classification and linear regression for estimating numeric target values

- The output of the logistic regression model is **interpreted as the log-odds of class membership**

- These **log-odds can be translated directly into the probability of class membership**

Logistic Regression (advanced)

p_+(x): model's estimate of the probability of class membership of a data item represented by feature vector x

1-p_+(x): estimated probability of the event not occurring

The Equation:

$$\log\left(\frac{p_+(\mathbf{x})}{1 - p_+(\mathbf{x})}\right) = f(\mathbf{x}) = w_0 + w_1x_1 + w_2x_2 + \cdots$$

specifies that for a particular data item, described by feature-vector **x**, **the log-odds of the class is equal to our linear function, f(x)**

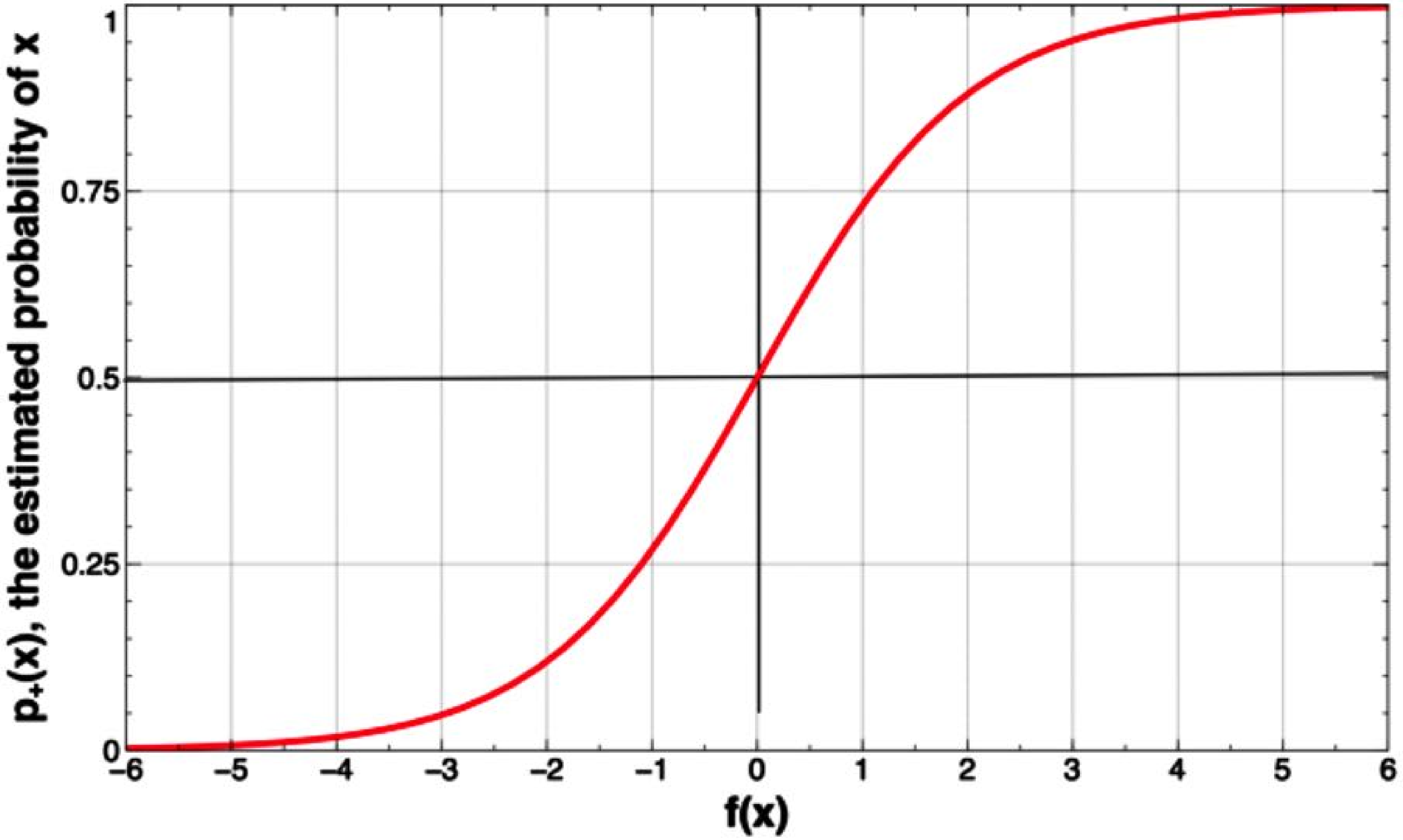
We actually want to **estimate p_+(x)** from this formula so we have:

Logistic Regression (advanced)

We actually want to **estimate $p_+(x)$** from this formula so we have:

$$p_+(\mathbf{x}) = \frac{1}{1 + e^{-f(\mathbf{x})}}$$

logistic activation function



plots the estimated probability **$p_+(x)$** (vertical axis) as a function of the **distance from the decision boundary** (horizontal axis)

This curve is called a “sigmoid” curve because of its “S” shape, which squeezes the probabilities into their correct range (between zero and one).

Regularization

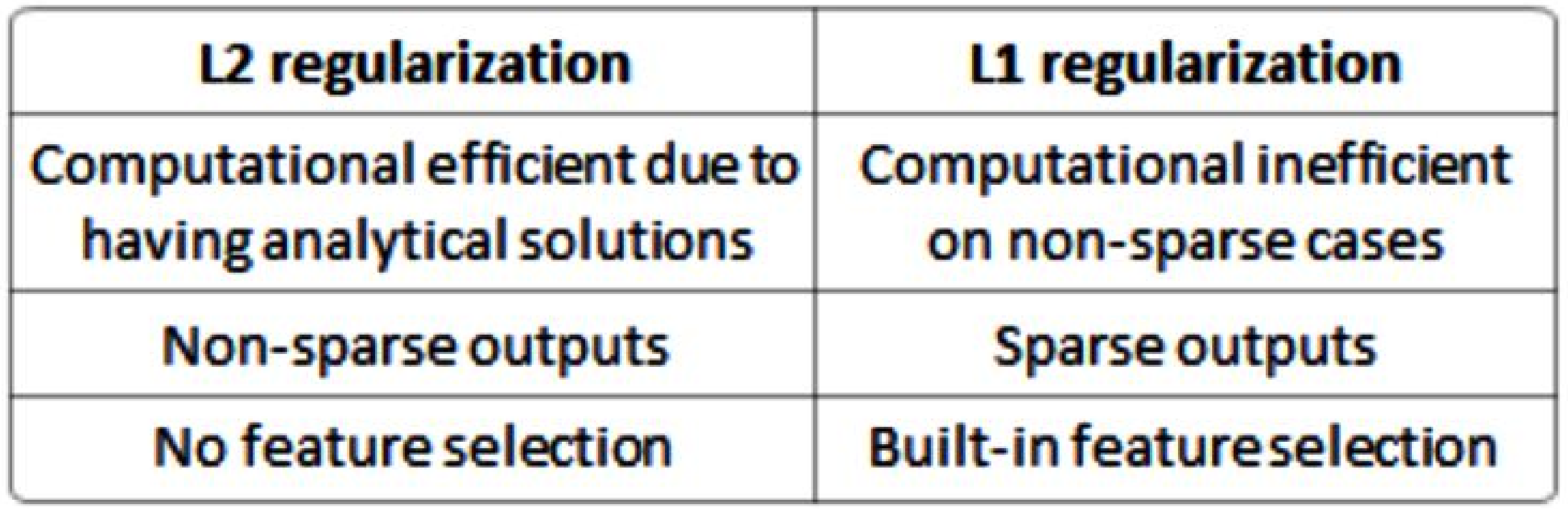
$$\underset{\mathbf{w}}{\arg\max}\ \text{fit}(\mathbf{x},\ \mathbf{w})$$

Say this is our objective function. Like g before, and we want to find w that gives the maximum

$$\underset{\mathbf{w}}{\arg\max}\ [\text{fit}(\mathbf{x},\ \mathbf{w}) - \lambda \cdot \text{penalty}(\mathbf{w})]$$

Complexity control work by adding a penalty

Against overfitting!



L2 regularization	L1 regularization
Computational efficient due to having analytical solutions	Computational inefficient on non-sparse cases
Non-sparse outputs	Sparse outputs
No feature selection	Built-in feature selection

Ridge **Lasso**

Common penalty is the sum of the squares of the weights (L2-norm).

Functions can fit data better if they are allowed to have large positive and negative weights. L2 gives a large penalty when weights have large absolute values.

Not linear SVM and kernel functions

- **Kernel functions** transform non-linear spaces into linear ones
  - sigmoid
  - radial
  - polynomial
  - others
- Applied in combination with SVM in scikit-learn

References and readings

- Chapter 4 [Provost]
- Chapter 4, Bishop “Pattern Recognition and Machine Learning” (freely available online) [advanced]

Questions?

@rschifan

schifane@di.unito.it

http://www.di.unito.it/~schifane