POLITECNICO
MILANO 1863

Machine Learning Methods for Communication Networks and Systems

Francesco Musumeci
Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB)
Politecnico di Milano, Milano, Italy

Part I – 2: Logistic regression

Outline

- Introduction
- Binary classification with logistic regression
- Decision boundary
- Parameter learning
- Logistic regression for multiple classes

Outline

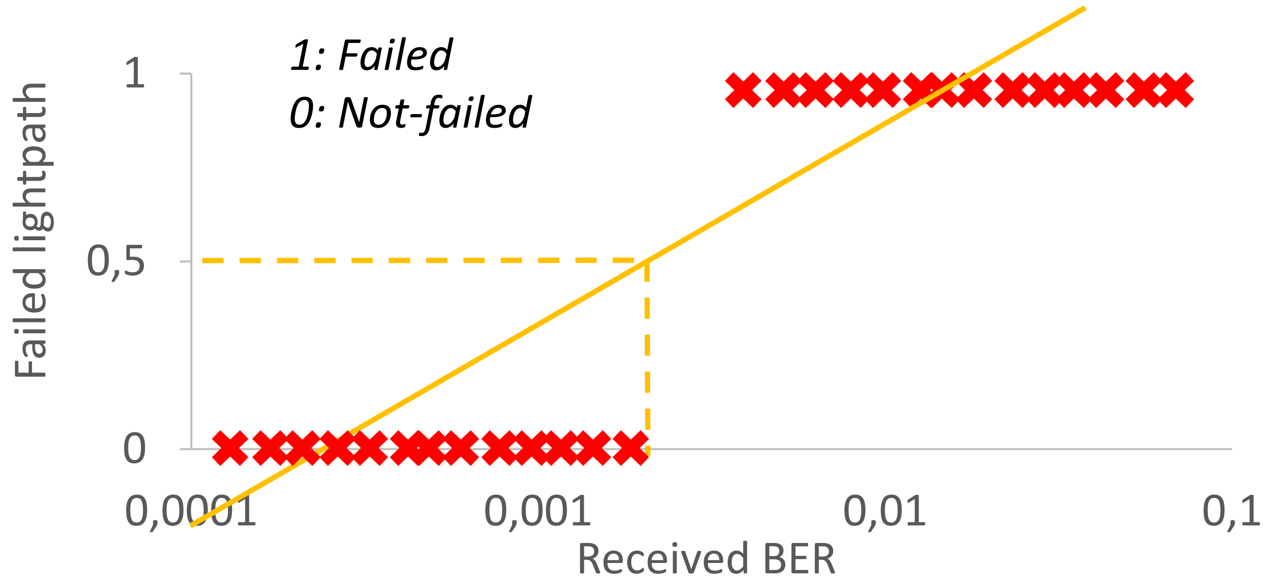
- Introduction
- Binary classification with logistic regression
- Decision boundary
- Parameter learning
- Logistic regression for multiple classes

Introduction

- **Logistic regression** is a *supervised* learning technique used for *classification* problems
- Given the "ground truth" for a set of (labeled) examples $(\underline{x}^{(i)}, y^{(i)})$, $i=1,2,\ldots,m$ (*"training set"* with m *"examples"*)
- Predict the class (category) for new (unlabeled) examples $\underline{x}_{test}$ (i.e., find y_{test})
  - y_{test} takes on **discrete** values
    - Binary classifier: y={0;1}, e.g., yes/no, good/bad, spam/non-spam…
    - Multiclass classifier: y={A,B,C,…}, e.g., colour, shape, character, images…
- General approach:
  - "guess" a model (hypothesis) for function $h(\underline{x})$
  - estimate parameters for function $h(\underline{x})$
  - perform prediction: $h(\underline{x}_{test})=y_{test}$

Introduction

- Why not linear regression to predict also *discrete* values?
  - Example: failed lightpaths vs received BER



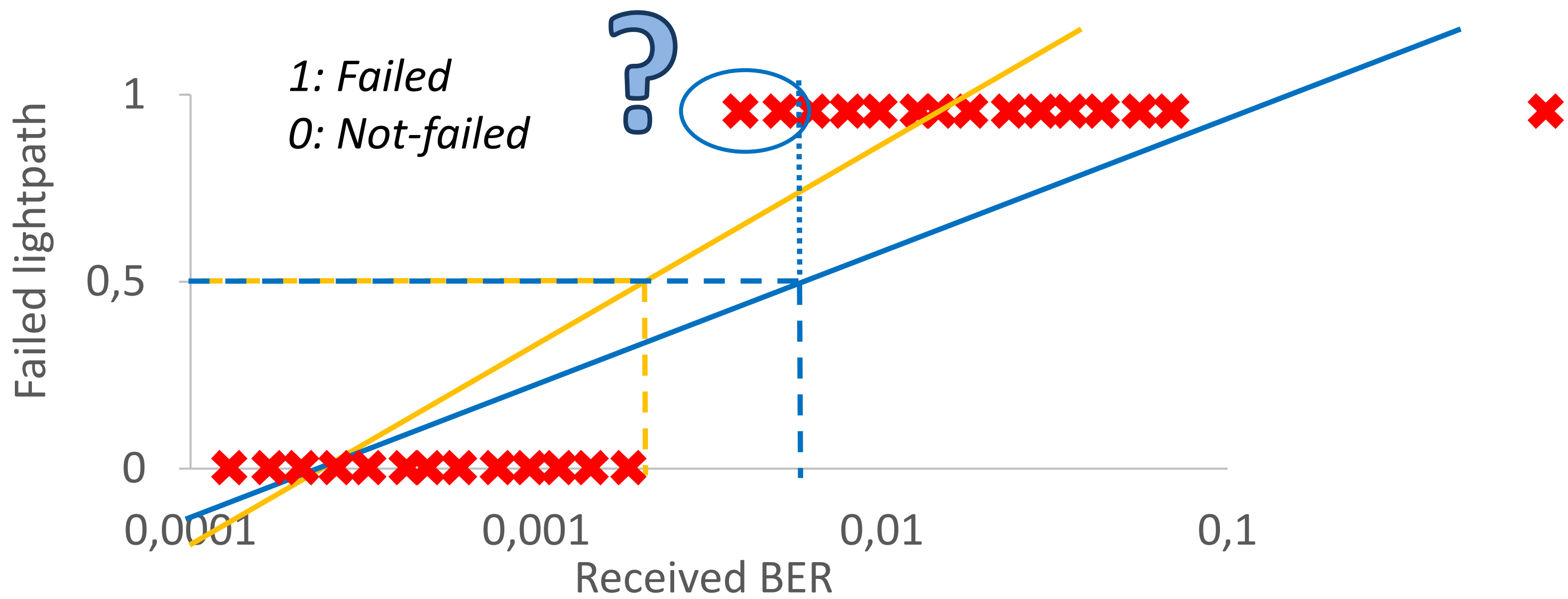
BER

  - Linear hypothesis: $h(x) = \theta_0 + \theta_1 x$ (matrix form: $h = \Theta^T X$)
  - Threshold-based prediction
    - If h(x) > threshold → failed lightpath
    - If h(x) < threshold → non-failed lightpath

N.B. Threshold value can be adapted to our needs

Introduction

- Why not linear regression to predict also *discrete* values?
  - Example: failed lightpaths vs received BER



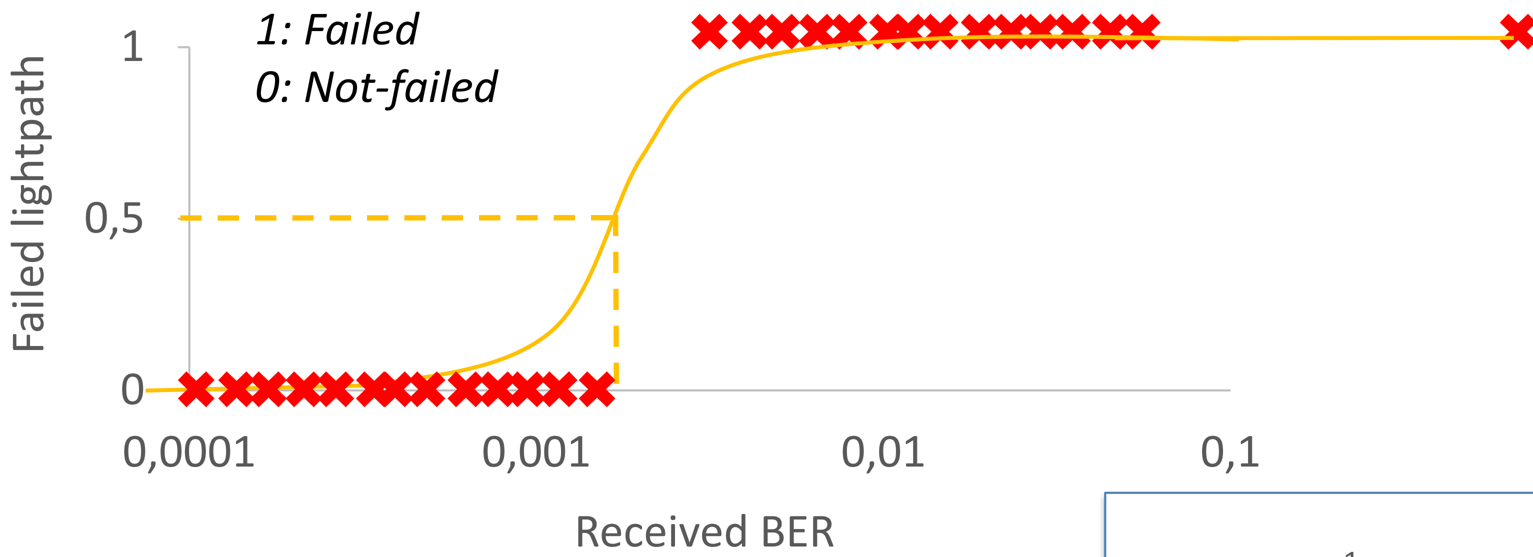
- Problems w/ the linear hypothesis
  - Values of *h(x)* greater than 1 and lower than 0 are meaningless
  - Adding "strong" examples worsen the prediction

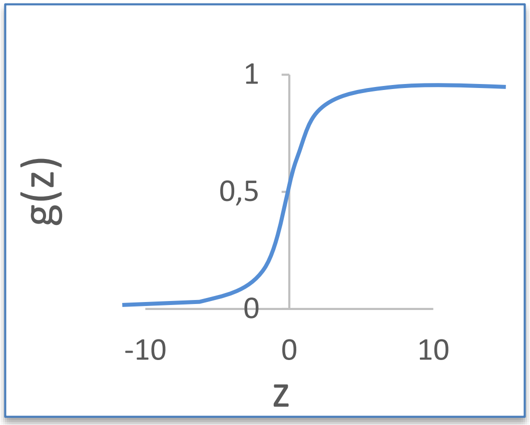
Outline

- Introduction
- Binary classification with logistic regression
- Decision boundary
- Parameter learning
- Logistic regression for multiple classes

Binary classification with logistic regression

- Solution: Logistic regression

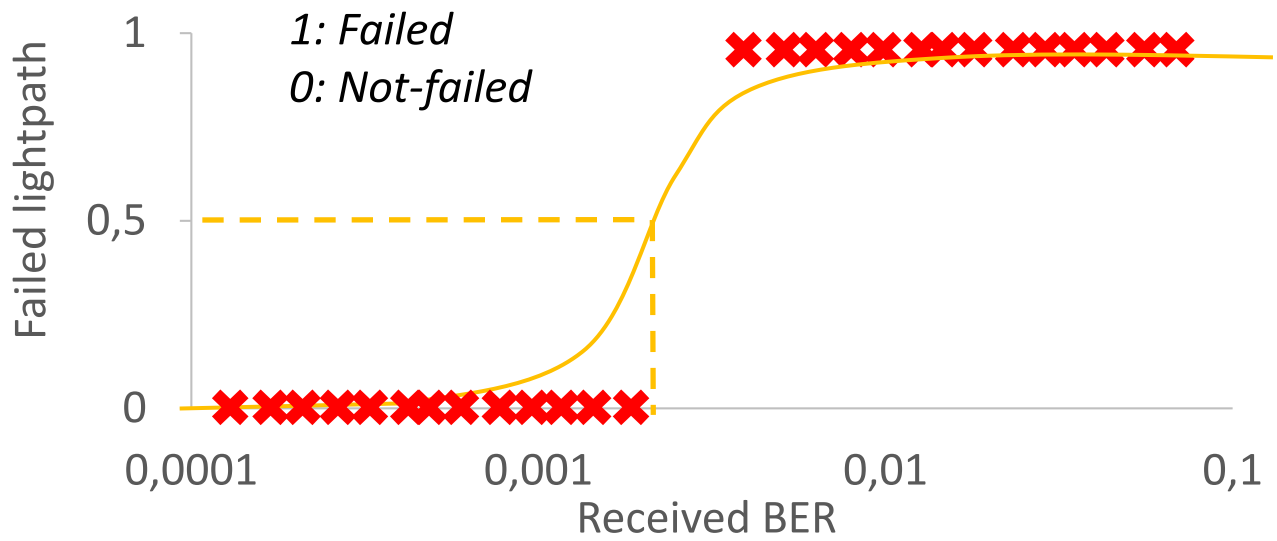




- $h(x) = g(\theta_0 + \theta_1 x)$ (in matrix form: $h = g(\Theta^T X)$)
  - $g(z) = 1/(1+e^{-z})$ is the "logistic" (or "sigmoid") function
    - for $z \rightarrow -\text{inf}$: $g(z) \rightarrow 0$
    - for $z \rightarrow +\text{inf}$: $g(z) \rightarrow 1$
    - for $z=0$: $g(z)=0.5$

Binary classification with logistic regression

- **Interpretation** of logistic regression



- $h(x) = g(\theta_0 + \theta_1 x)$ (in matrix form: $h = g(\Theta^T X)$)
  - $h(x)=p(y=1|\Theta;x)$
    - **probability** that a new example ***x*** belongs to the *positive class* (e.g., failed lightpaths) **given** the parameters θ_0 and θ_1
  - Prediction for new examples is performed via a threshold on this probability (e.g., p ≥ 0.5)

Outline

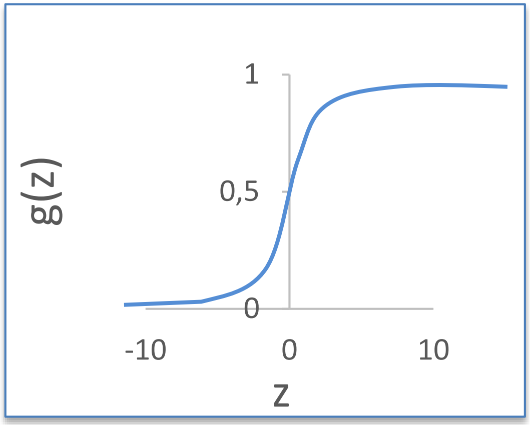
- Introduction
- Binary classification with logistic regression
- Decision boundary
- Parameter learning
- Logistic regression for multiple classes

Decision boundary

- Prediction with logistic regression

$h(x) = g(\theta_0 + \theta_1\, x)$ (in matrix form: $h = g(\Theta^T X)$)

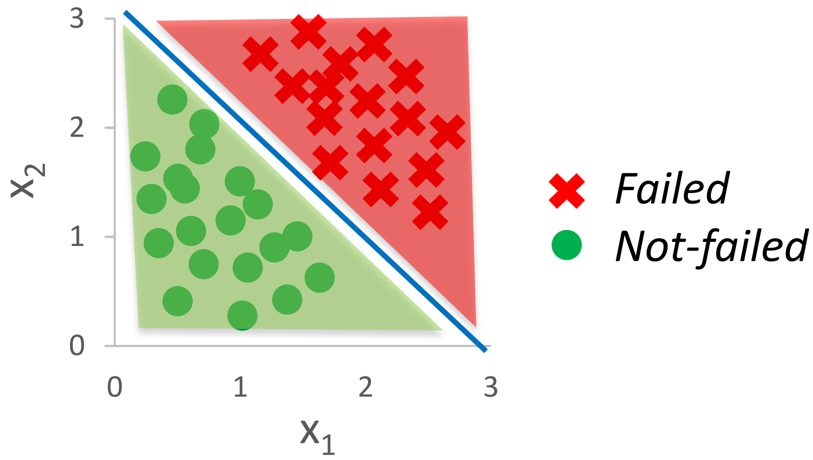
  - $\theta_0 + \theta_1\, x \geq 0$ → *predict* $y=1$
  - $\theta_0 + \theta_1\, x < 0$ → *predict* $y=0$



- *Decision boundary*: straight line w/ equation

$$\theta_0 + \theta_1\, x = 0$$

- In multi-dimensional space (e.g., 2 features x_1 and x_2)



Decision boundary:

$$x_1 + x_2 = 3$$

Positive class (y=1):

$$x_1 + x_2 \geq 3$$

Negative class (y=0):

$$x_1 + x_2 < 3$$

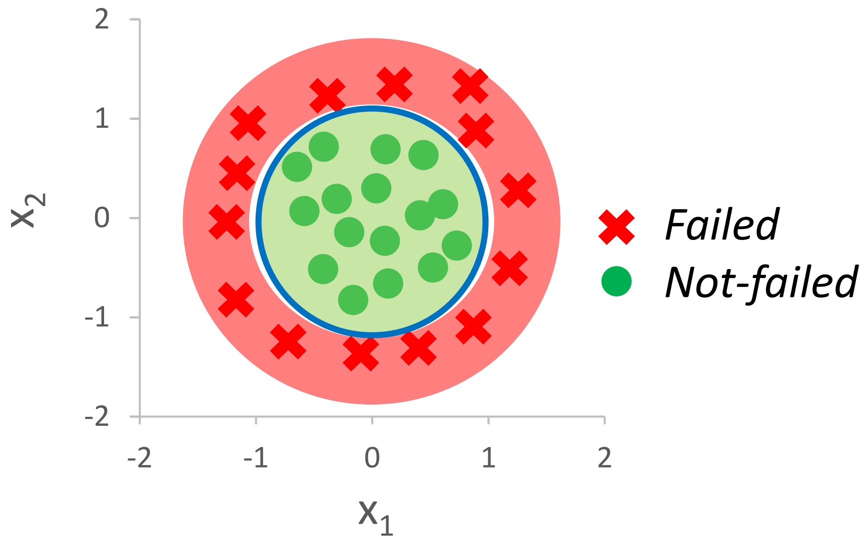
Decision boundary

Nonlinear decision boundaries

- Require adding polynomial features

$$h(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \ldots)$$

- Example: circular decision boundary



Decision boundary:

$$x_1^2 + x_2^2 = 1$$

Positive class (y=1):

$$x_1^2 + x_2^2 \geq 1$$

Negative class (y=0):

$$x_1^2 + x_2^2 < 1$$

Outline

- Introduction
- Binary classification with logistic regression
- Decision boundary
- Parameter learning
- Logistic regression for multiple classes

Parameter learning
Optimization objective

- How do we choose parameters θ_i to have a good fit?
  - "intuitive" choice: minimize MSE

$$MSE(\theta_0,\theta_1)=\frac{1}{2m}\sum_{i=1}^{m}\left(h(x^{(i)})-y^{(i)}\right)^2 \qquad h(x^{(i)})=\frac{1}{1+e^{-(\theta_0+\theta_1 x^{(i)})}}$$

  - problem: MSE is *non-convex* (has local optima)
- <u>Solution</u>: minimize the **new cost function**:

$$J(\theta)=\frac{1}{m}\sum_{i=1}^{m}\mathrm{Cost}(h_\theta(x^{(i)}),y^{(i)})$$

where

$$\mathrm{Cost}(h_\theta(x),y)=\begin{cases}-\log(h_\theta(x)) & \text{if y}=1\\ -\log(1-h_\theta(x)) & \text{if y}=0\end{cases}$$

Parameter learning
Simplified optimization objective

- Cost function

$$\mathrm{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if y} = 1 \\ -\log(1 - h_\theta(x)) & \text{if y} = 0 \end{cases}$$

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)})$$

- Rearranging…

$$\mathrm{Cost}(h_\theta(x), y) = -y \log(h_\theta(x)) - (1 - y) \log(1 - h_\theta(x))$$

…

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} \left[y^{(i)} \log(h_\theta(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)})) \right]$$

Parameter learning
Gradient descent

$$h(x^{(i)}) = \frac{1}{1+e^{-(\theta_0+\theta_1 x^{(i)})}}$$

- Given the cost function

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log(h_\theta(x^{(i)})) + (1-y^{(i)})\log(1-h_\theta(x^{(i)}))\right]$$

- Use gradient descent to minimize cost function *J(θ)*
  - start with (random) initialization of *θ* (θ_0, θ_1 if we have one feature)
  - iteratively update *θ* to reduce *J(θ)*

$$\theta_j = \theta_j - \alpha\frac{\partial}{\partial\theta_j}J(\theta) \quad \mathrm{j}=0,1$$

simultaneous update

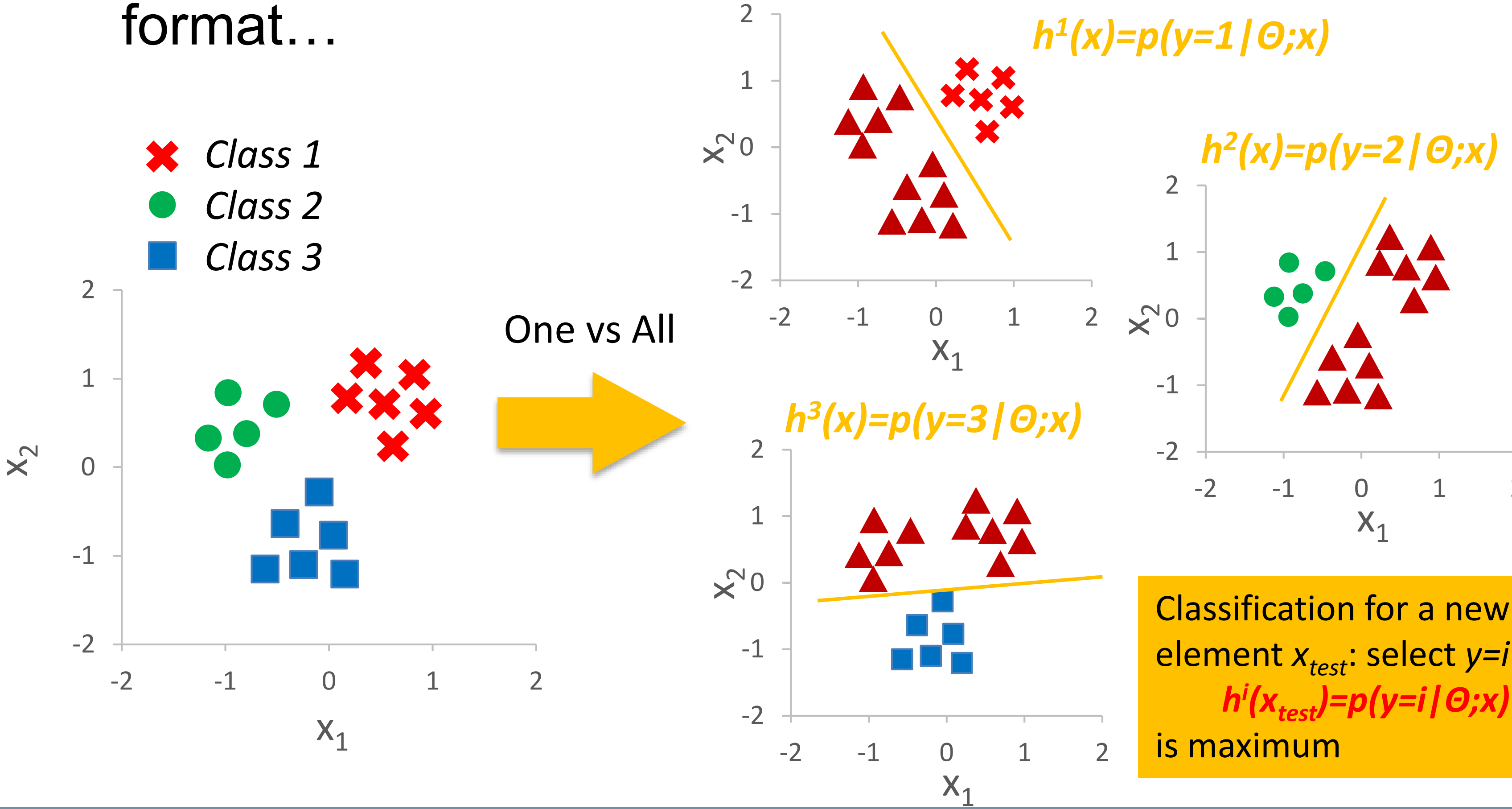
  - STOP when convergence is reached
- To make a prediction on (i.e., to classify) a new example *x*:
  - Use probability interpretation of: $h(x) = \frac{1}{1+e^{-(\theta_0+\theta_1 x)}}$
  - Predict y=1 if *h* ≥ *threshold* (0 otherw.)

Outline

- Introduction
- Binary classification with logistic regression
- Decision boundary
- Parameter learning
- Logistic regression for multiple classes

Logistic regression for multiple classes

- Classification with more than two classes
  - Examples: distinguish traffic flows, recognize modulation format…



Classification for a new element x_{test}: select $y=i$ s.t.

$h^i(x_{test})=p(y=i|\Theta;x)$

is maximum