Machine Learning in Healthcare

#C05 Linear Models for Classification

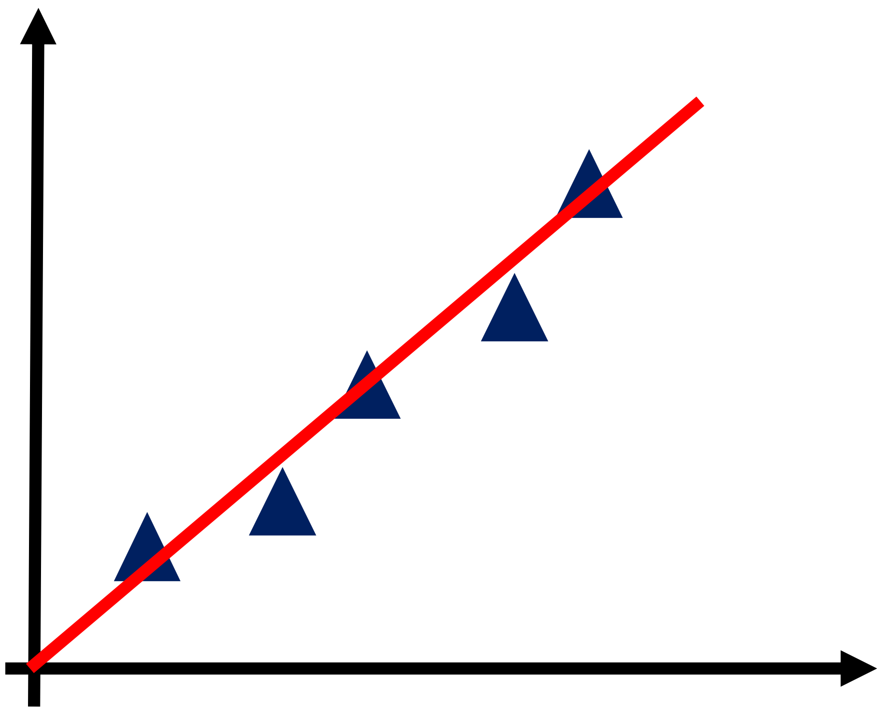
Technion-IIT, Haifa, Israel

Assist. Prof. Joachim Behar
Biomedical Engineering Faculty
Technion-IIT

Classification versus Regression

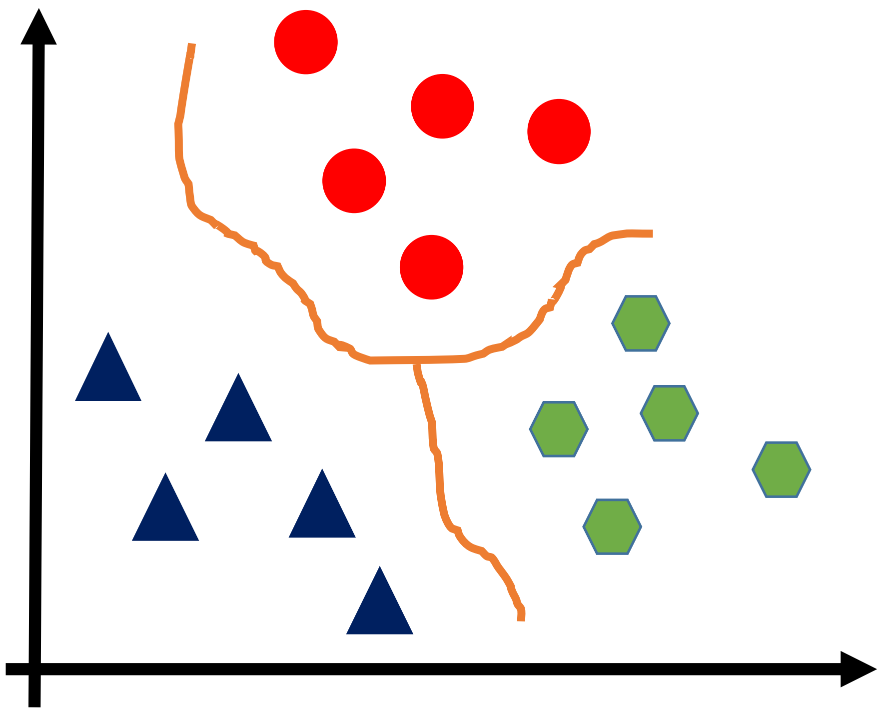
Regression versus classification

<u>Regression</u>



Estimate relationships among usually continuous variables.

<u>Classification</u>



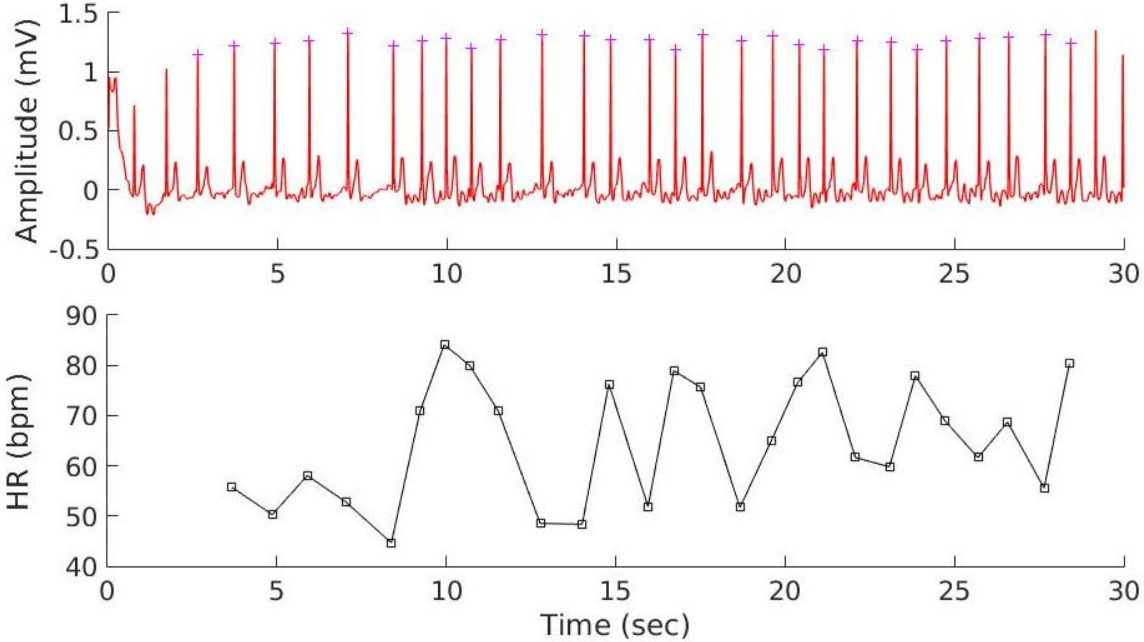
Identify decision boundary between examples of different classes.

Classification

- Examples:
  - Tumor: Malignant/benign?
  - Rhythm: Atrial fibrillation/normal sinus?
- Let’s consider a binary classification problem for now:

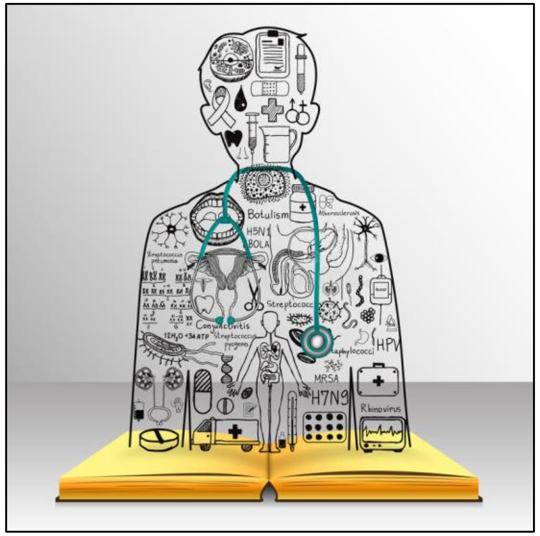
$$y \in \{0,1\}$$

0: negative class (non-AF)
1: positive class (AF)

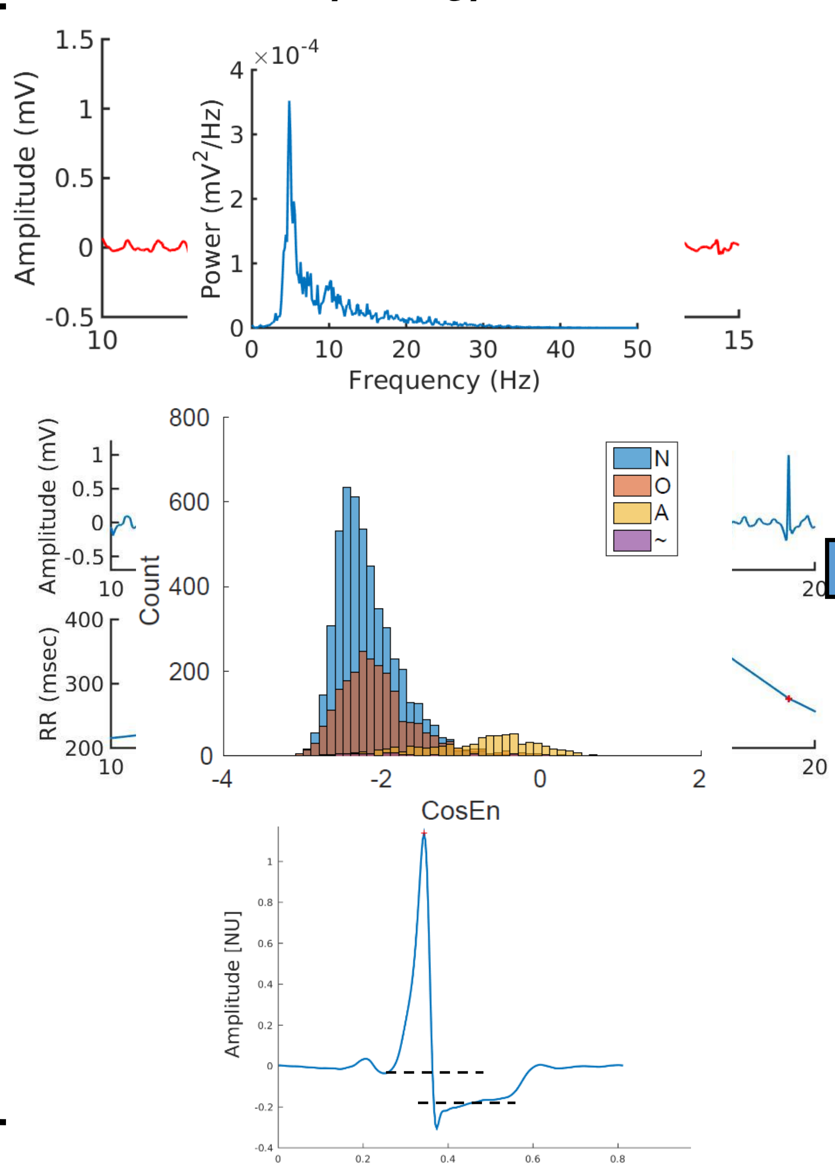


Behar et al., CinC 2017

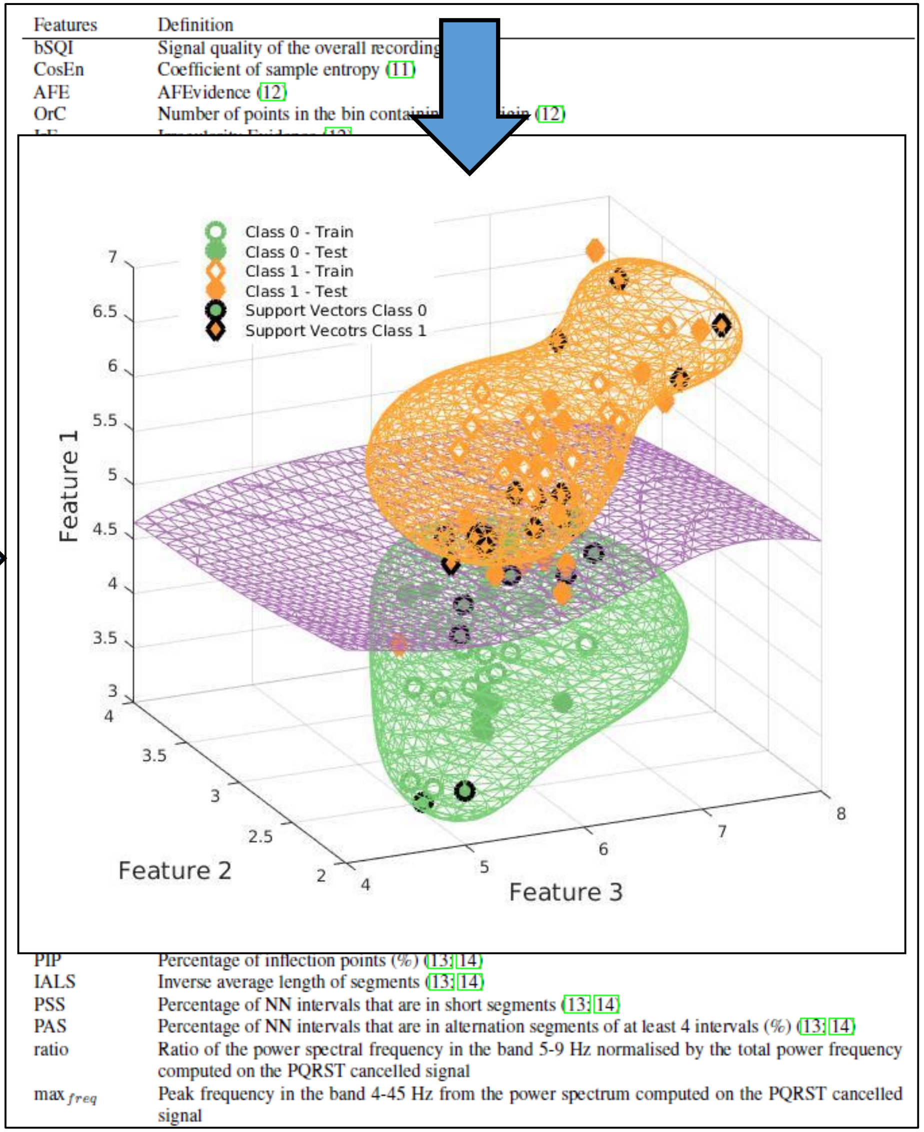
Linear Models for Classification



Physiology



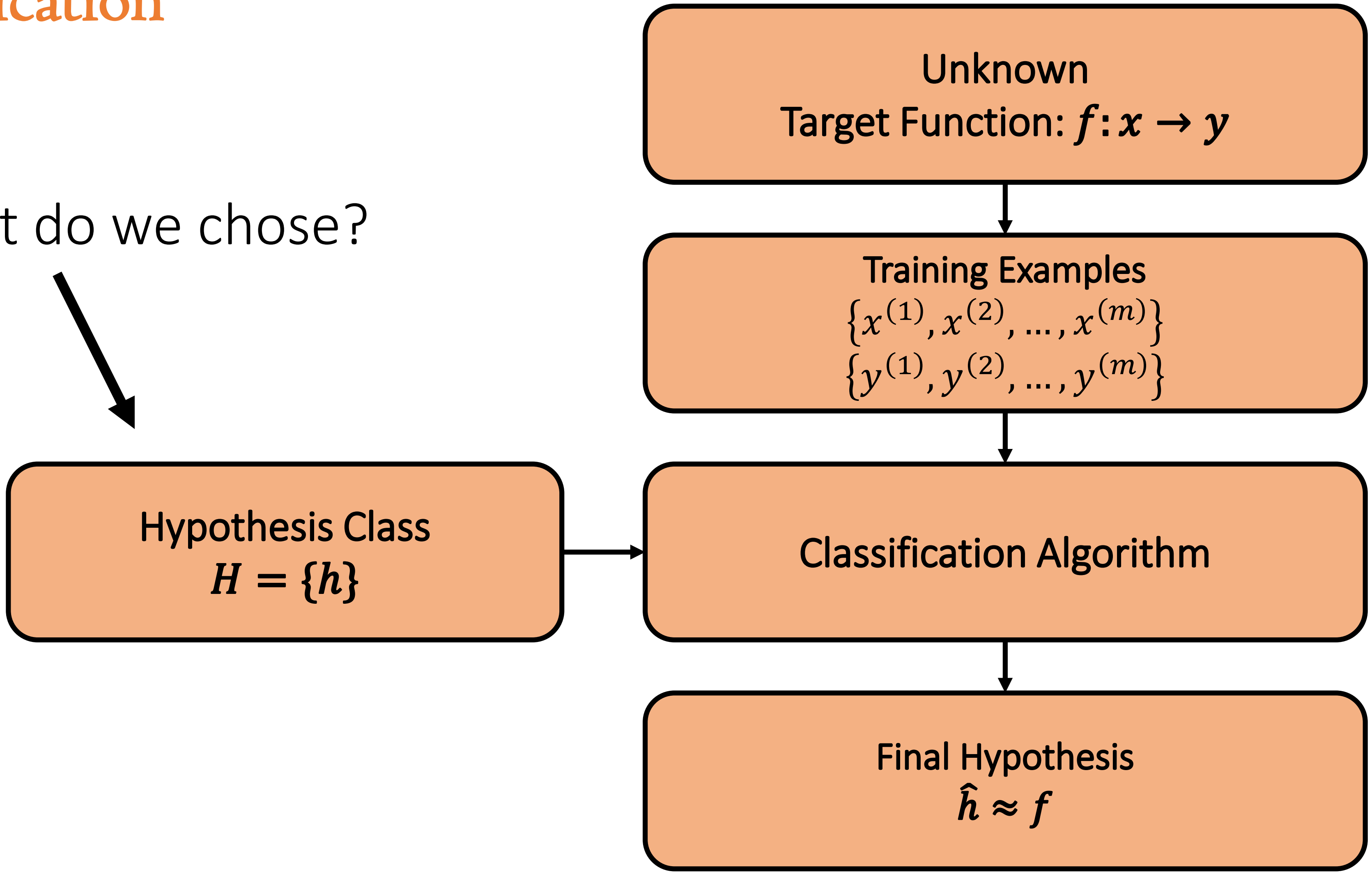
Physio-Features



Behar et al., CinC 2017

Classification

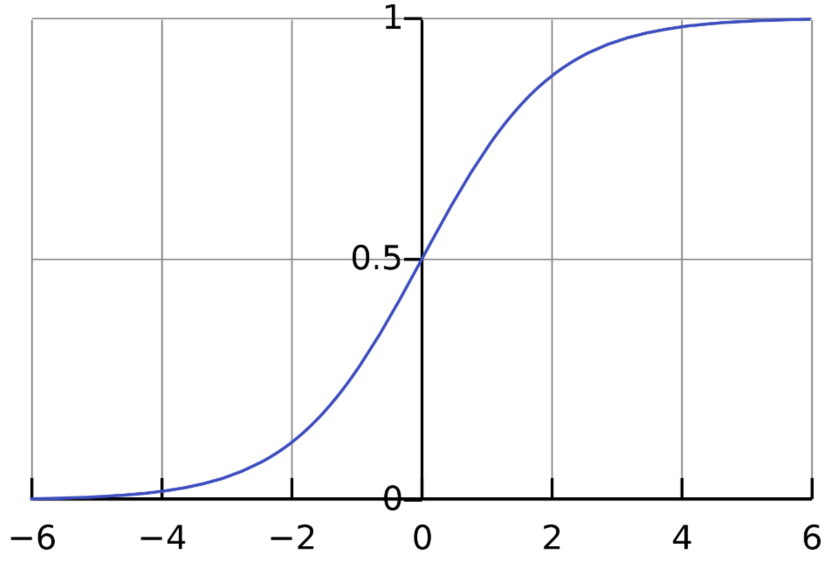
What do we chose?



LR Hypothesis Representation

Hypothesis representation

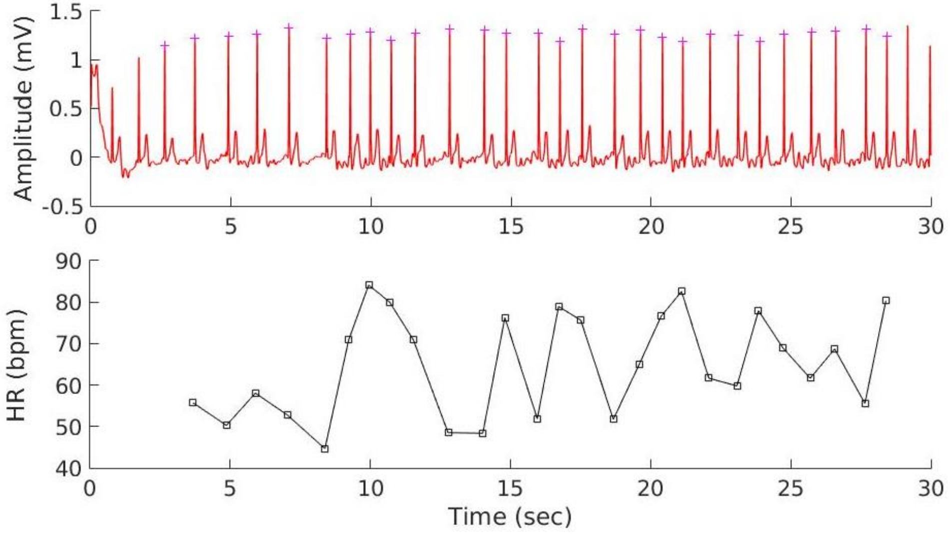
- Linear regression: $h_w(x) = w^T x$
- Logistic regression: $h_w(x) = g(w^T x)$
  - With $g(z) = \frac{1}{1+e^{-z}} = \sigma(z)$ the **sigmoid function**.
  - $\lim_{z \to +\infty} g(z) = 1$,
  - $\lim_{z \to -\infty} g(z) = 0$,
  - $g(0) = 0.5$.

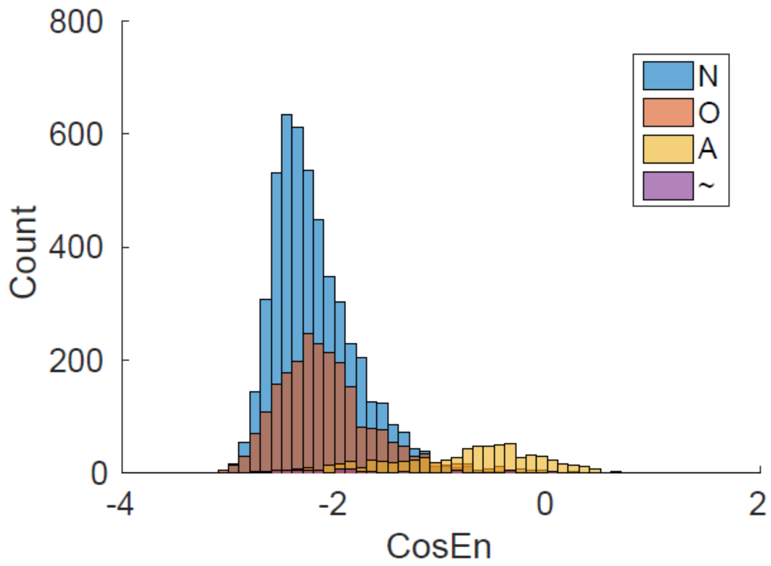


Remark: “Logistic regression” is not a “regression” algorithm but a classification one. The naming is just historical!

Hypothesis representation

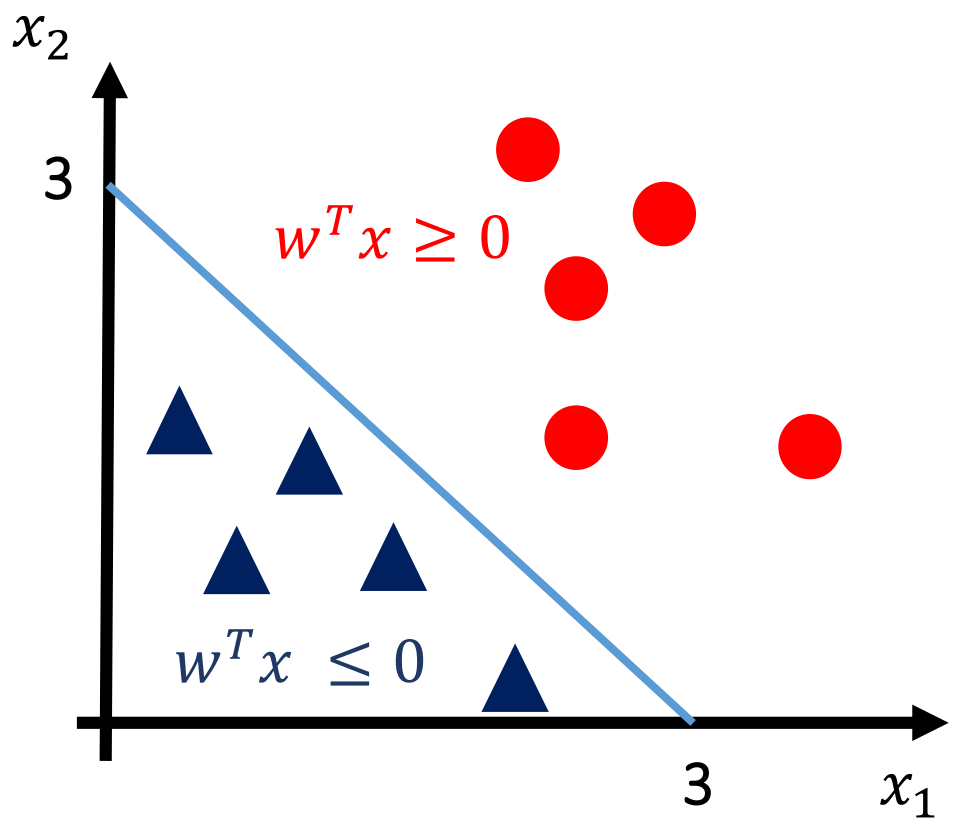
- Interpretation of the **probabilistic output**:
  - $x = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1 \\ CosEn \end{bmatrix},$
  - $h_w(x) = 0.7 = P(y = 1|x, w),$
  - This individual has 70% change to have AF.





Hypothesis representation

- Interpretation of the **decision boundary**:
  - $h_w(x) = g(w^T x) = \frac{1}{1+e^{-w^T x}}$
- Example:
  - $y = 1$ if $h_w(x) \geq 0.5 \iff w^T x \geq 0$
  - Conversely, $y = 0$ if $h_w(x) \leq 0.5 \iff w^T x \leq 0.$
  - This gives the decision boundary.
- $e.g.\ h_w(x) = g(-3 + x_1 + x_2)$



LR Cost Function

Cost function

- We have a training set of
  - m examples: $\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$
  - With target labels: $\{y^{(1)}, y^{(2)}, \dots, y^{(m)}\}$
- How do we find the weights w of the LR model?
  - Reminder
    - Linear regression: $J(w) = \frac{1}{m}\sum_{i=1}^{m}(h_w(x^{(i)}) - y^{(i)})^2$
  - In LR we have $h_w(x) = \sigma(z)$ and the resulting cost function $J(w)$ is **non-convex**. Thus we need to find for another cost function that is convex.

AIMLab.

Reminder: convex function

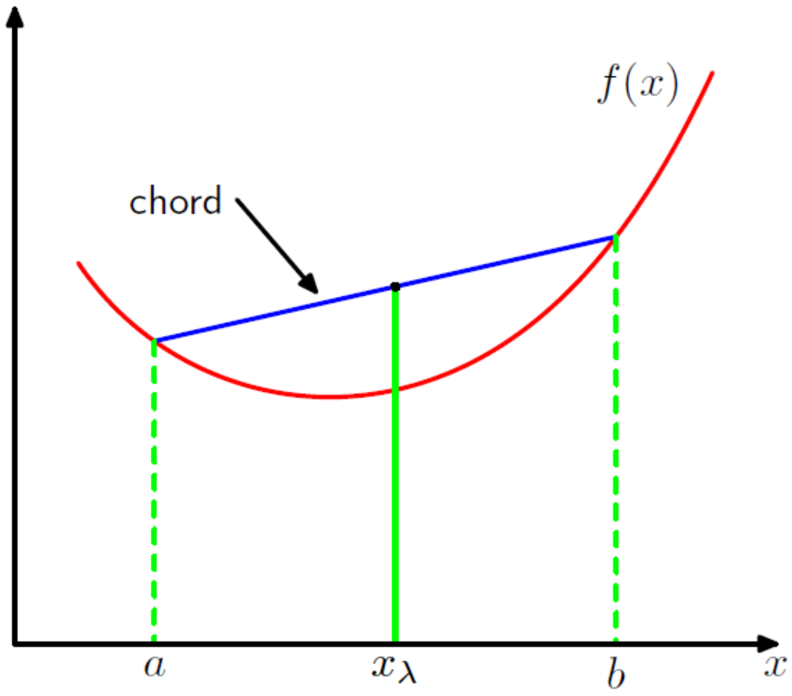
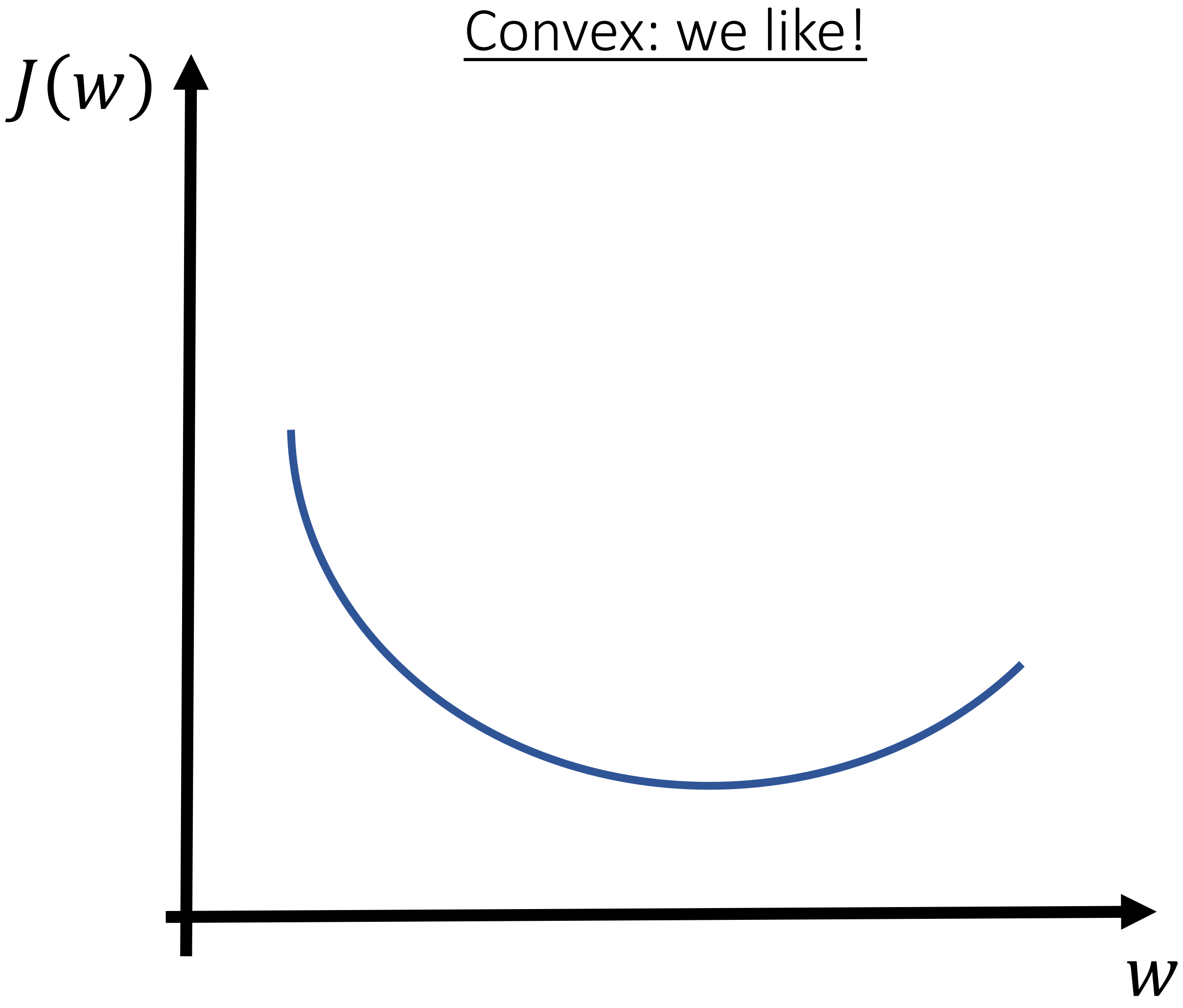


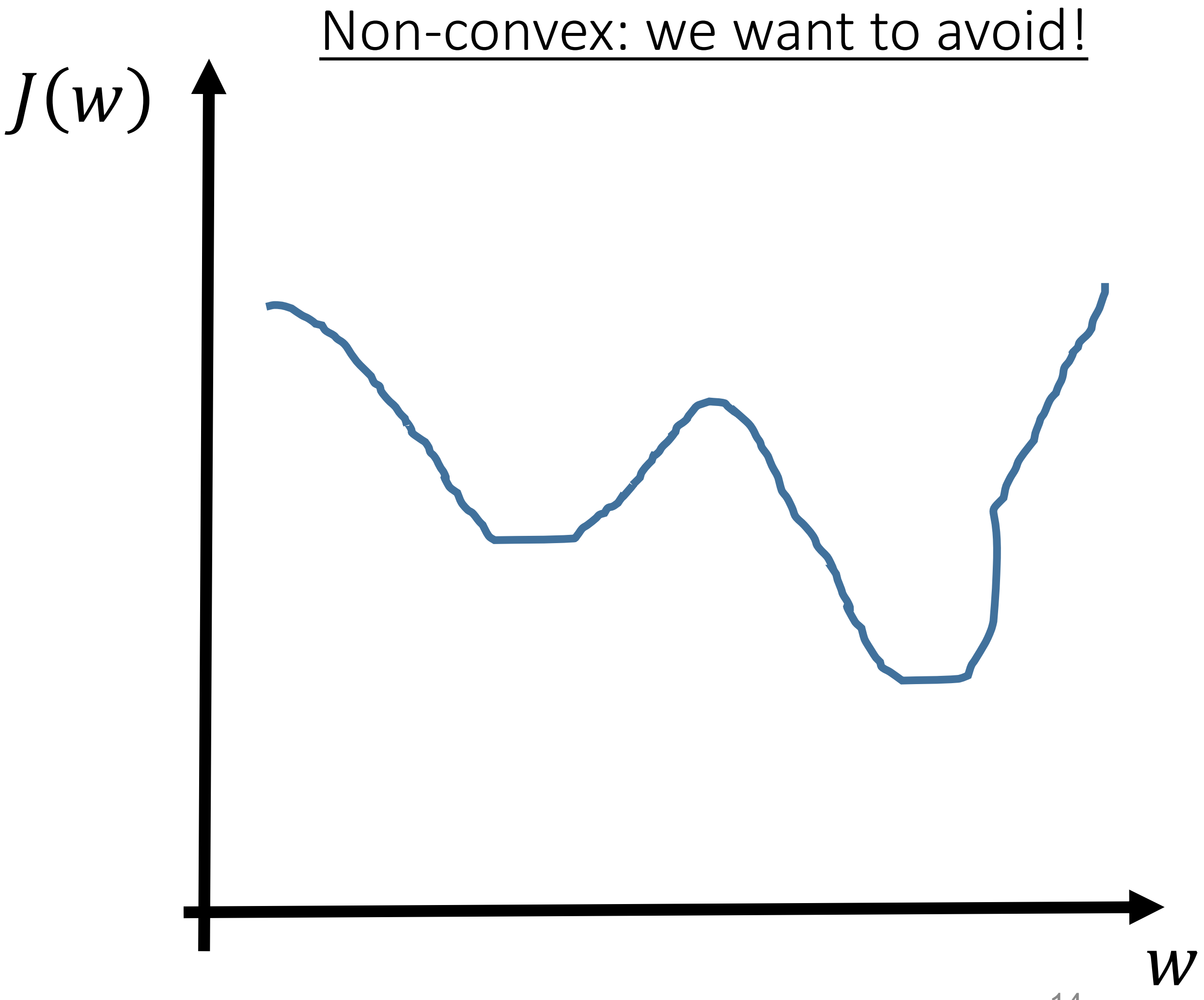
Figure 1.31 A convex function $f(x)$ is one for which every chord (shown in blue) lies on or above the function (shown in red).

$$f(\lambda a + (1-\lambda)b) \leqslant \lambda f(a) + (1-\lambda) f(b).$$

Image: From Bishop's Pattern recognition and machine learning, Figure 1.31

Reminder: convex function





Cost function in LR

- We define the following **error**:

$$E_w(x,y) = \begin{cases} -\,log\left(h_w(x)\right), if\ y = 1 \\ -\,log\left(1 - h_w(x)\right), if\ y = 0, \end{cases}$$

  - If $y = 1$ and $h_w(x) \to 0$ then $E_w(x,y) \to \infty$
  - If $y = 1$ and $h_w(x) \to 1$ then $E_w(x,y) \to 0$.
  - Ibid $y = 0$.

Cost function in LR

- We define the **error**:
  - $E_w(x,y) = \begin{cases} -\log\left(h_w(x)\right), if\ y = 1 \\ -\log\left(1 - h_w(x)\right), if\ y = 0, \end{cases}$
  - We can re-write it:
  - $E_w(x,y\ ) = -ylog\left(h_w(x)\right) - (1-y)\log(1 - h_w(x))$
- The **cost function**:
  - $J(w) = \frac{1}{m}\sum_{i=1}^{m} E_w\left(x^{(i)}, y^{(i)}\right)$
  - $J(w) = \frac{1}{m}\sum_{i=1}^{m} \left[-y^{(i)}log\left(h_w\left(x^{(i)}\right)\right) - \left(1 - y^{(i)}\right)\log(1 - h_w\left(x^{(i)}\right))\right]$
- It is possible to show that the cost function $J(w)$ is convex.
- This is called the **Cross-Entropy** cost function or log loss.

Cost function in LR

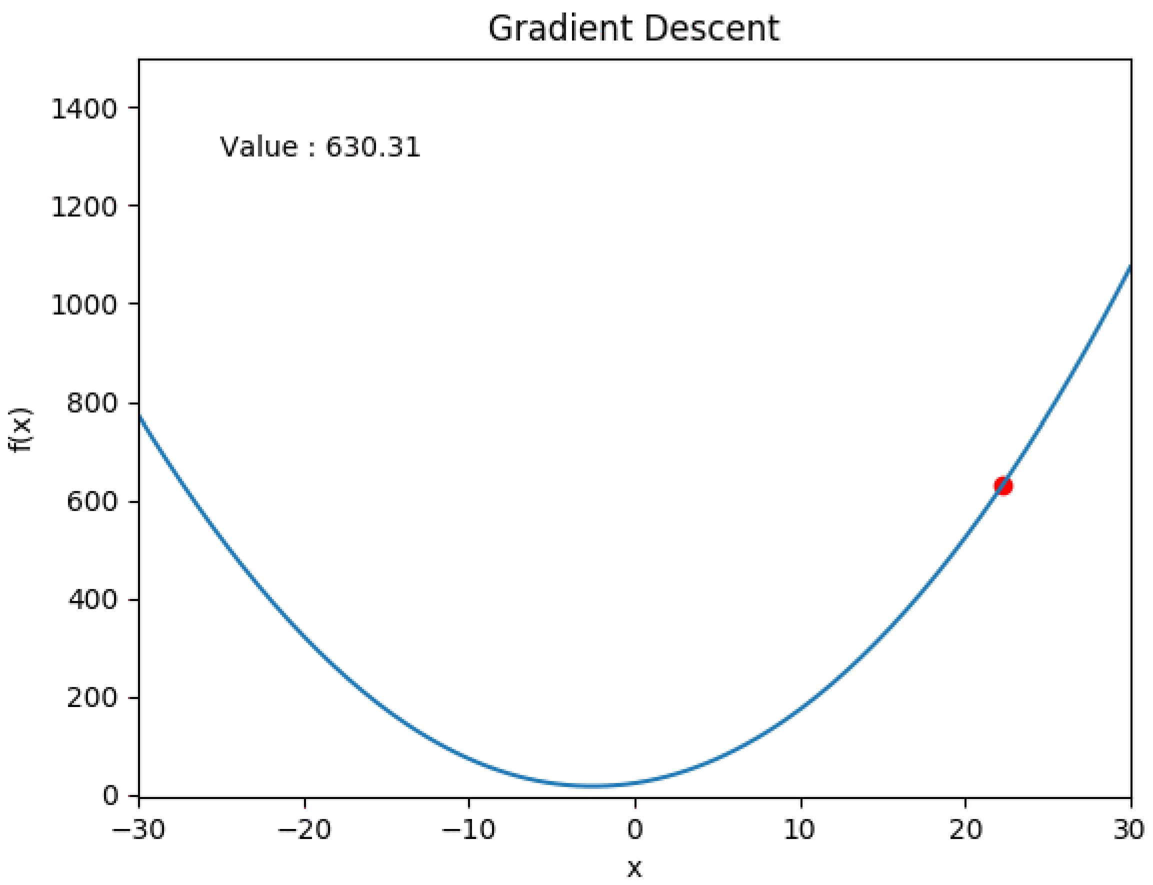
- Why do we choose this particular error definition?
  - Maximum likelihood estimate.
    - $argmax_w \mathcal{L}(w|Y,X) = argmax_w (\prod_{i=1}^{m} \mathcal{L}(w|y^{(i)}, x^{(i)}))$
  - Convex cost function.

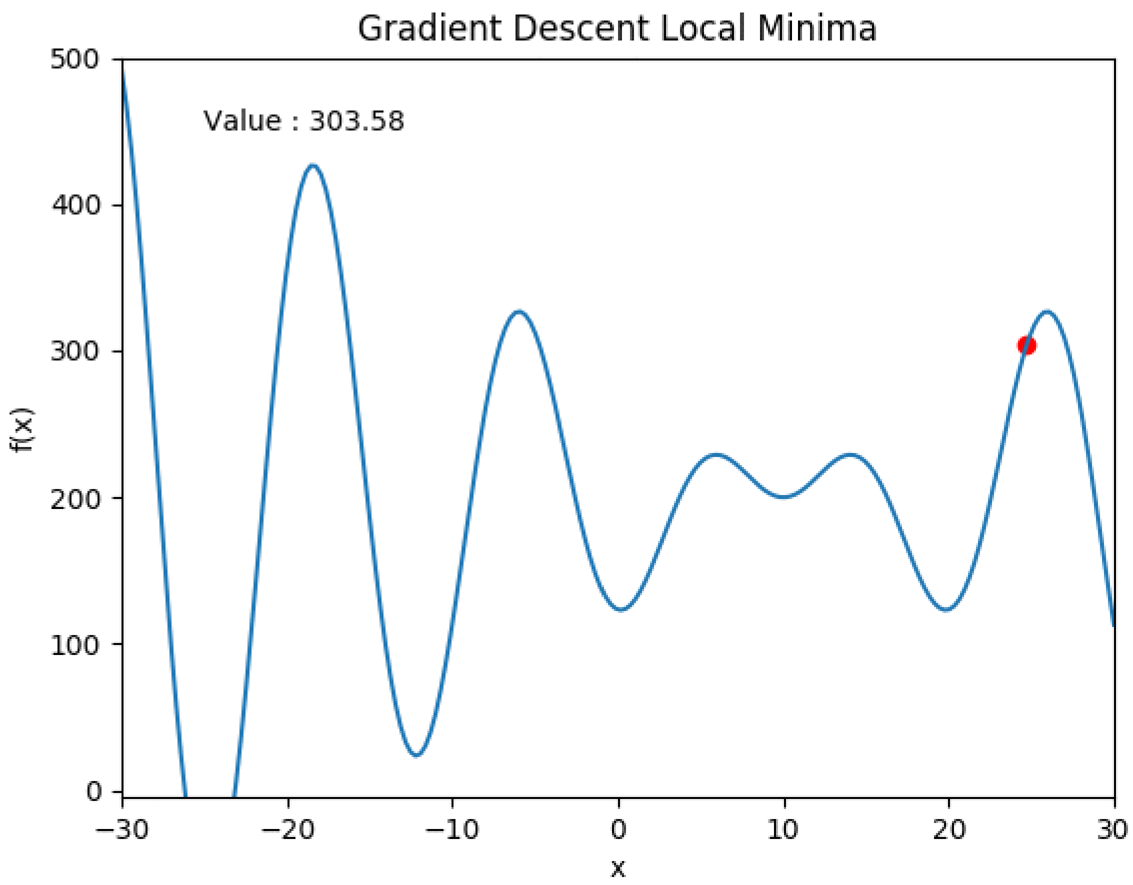
Gradient Descent

Optimization algorithm

- We now want to use some optimization algorithm to solve our optimization problem given the cost function we defined.
- We want to find $min_w\big(J(w)\big)$
- For that purpose we will use **gradient descent**.
- Given w we compute
  - $J(w)$
  - $\frac{\partial J(w)}{\partial w_j}$ for $j \in [1, \dots, n]$
- Gradient descent, update w_j
  - $w_j := w_j$-α$\frac{\partial J(w)}{\partial w_j}$
- More sophisticated alternative to gradient descent exist: conjugate gradient, BFGS, L-BFGS etc.

Optimization algorithm





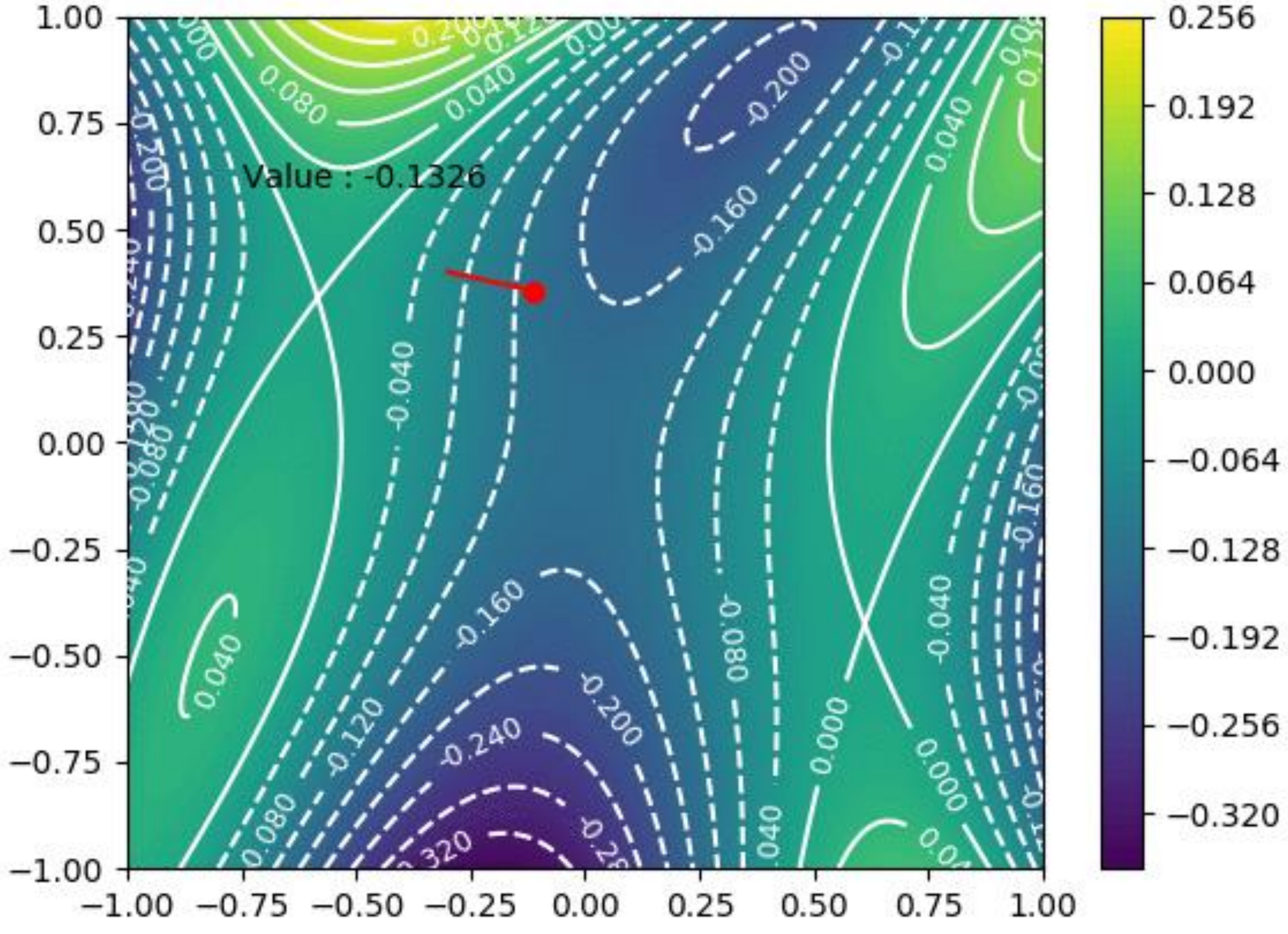
https://github.com/Shathra/gradient-descent-demonstration

Optimization algorithm

- How do we use gradient descent with LR?
  - Snapshot here – more in details in the coming slides.
- The overall cost function:
  - $J(w) = \frac{1}{m}\sum_{i=1}^{m}\left[-y^{(i)}log\left(h_w\left(x^{(i)}\right)\right) - (1 - y^{(i)})\log(1 - h_w\left(x^{(i)}\right))\right]$
- We need to compute $\frac{\partial J(w)}{\partial w_j}$ for $j \in [1, \ldots, n]$
  - $\frac{\partial J(w)}{\partial w_j} = \frac{1}{m}\sum_{i=1}^{m}\left(h_w\left(x^{(i)}\right) - y^{(i)}\right)x_j^{(i)}$
- Gradient descent, update w_j
  - $w_j := w_j\text{-}\alpha\frac{\partial J(w)}{\partial w_j} = w_j - \alpha\frac{1}{m}\sum_{i=1}^{m}\left(h_w\left(x^{(i)}\right) - y^{(i)}\right)x_j^{(i)}$
- What about feature scaling? Yes, we need it!

AIMLab.

Optimization algorithm

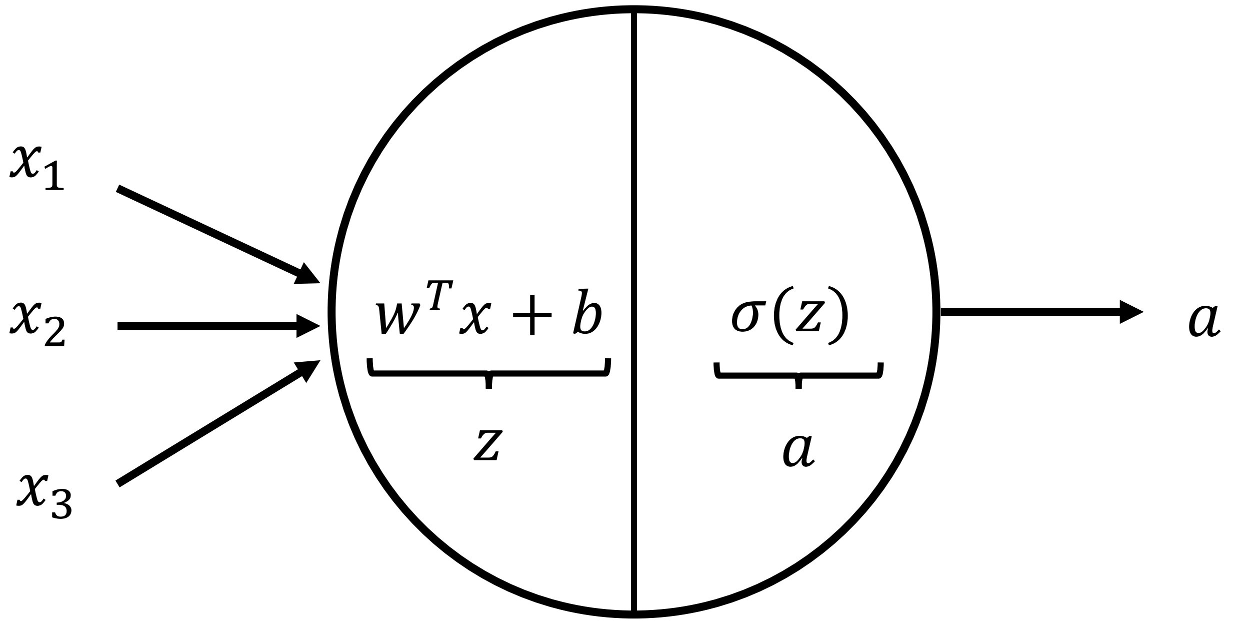


https://github.com/Shathra/gradient-descent-demonstration

Logistic Regression Gradient Descent

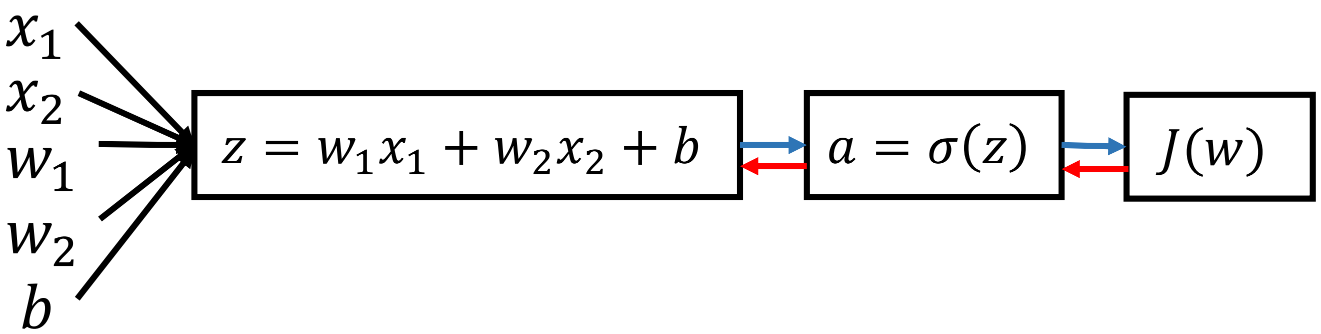
Logistic regression equations

- Equations:
  - $z = w^T x + b$
  - $a = h_w(z) = \sigma(z)$

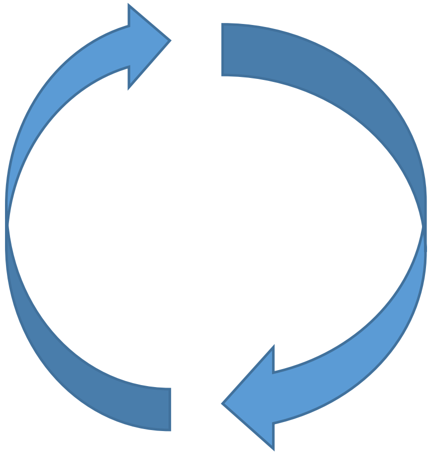


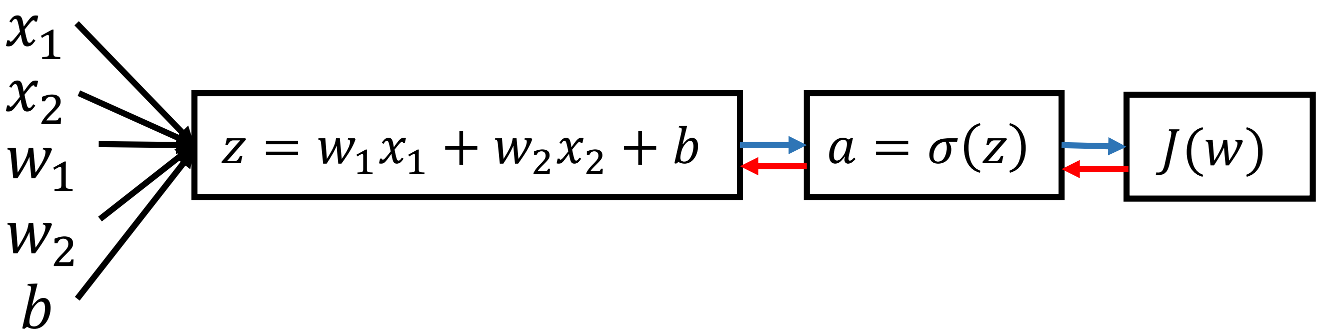
AIMLab.

Logistic regression equations



- $z = w^T x + b$
- $a = h_w(z) = \sigma(z)$
- $J(w) = -y log(a) + (1-y)\log(1-a)$ for a single example.
- Forward propagation: $\rightarrow$
  - $z = w^T x + b$
  - $a = \sigma(z)$
- Backward propagation: $\leftarrow$
  - $w_1 := w_1 - \frac{\partial J(w)}{\partial w_1} = w_1 - \alpha(a-y)x_1$
  - $w_2 := w_2 - \frac{\partial J(w)}{\partial w_2} = w_2 - \alpha(a-y)x_2$
  - $b := b - \frac{\partial J(w)}{\partial b} = b - \alpha(a-y)$
- Iterate between forward and backward step.

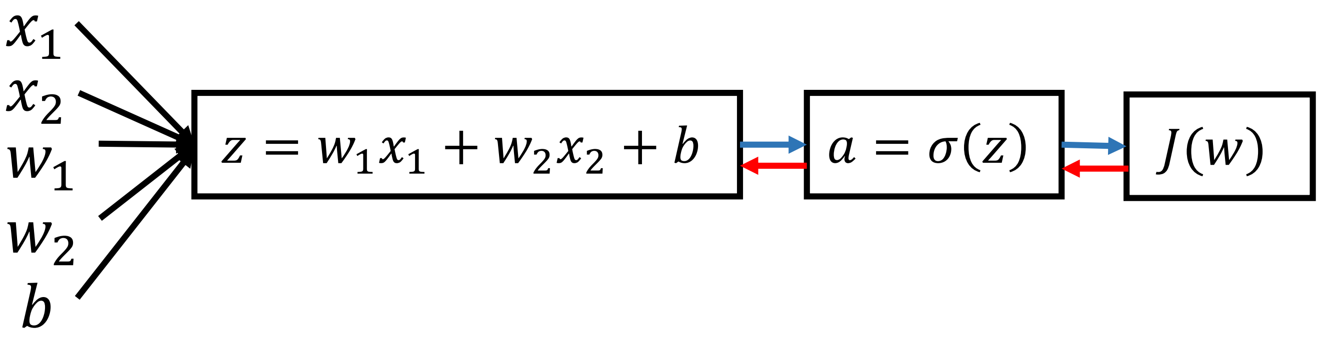




Logistic regression equations

- Now consider m examples
  - Forward propagation,
    - $z^{(i)} = w^T x^{(i)} + b, \forall i \in [1, m]$
    - $a^{(i)} = \sigma(z^{(i)}), \forall i \in [1, m]$
  - Backward propagation:
    - $w_1 := w_1 - \frac{1}{m} \sum_{i=1}^{m} \alpha\left(a^{(i)} - y^{(i)}\right) x_1^{(i)}$
    - $w_2 := w_2 - \frac{1}{m} \sum_{i=1}^{m} \alpha\left(a^{(i)} - y^{(i)}\right) x_2^{(i)}$
    - $b := b - \frac{1}{m} \sum_{i=1}^{m} \alpha\left(a^{(i)} - y^{(i)}\right).$

Logistic regression equations



- Now consider n input features:
  - Forward propagation:
    - $z^{(i)} = w^T x^{(i)} + b, \forall i \in [1, m]$
    - $a^{(i)} = \sigma(z^{(i)}), \forall i \in [1, m]$
  - Backward propagation:
    - $w_j := w_j - \frac{1}{m}\sum_{i=1}^{m} \alpha\left(a^{(i)} - y^{(i)}\right)x_j^{(i)}, \forall j \in [1, n]$
    - $b := b - \frac{1}{m}\sum_{i=1}^{m} \alpha\left(a^{(i)} - y^{(i)}\right).$
- So if we perform k iterations of gradient descent we need to go through two loops of m (forward) and then n (backward) steps. In LR this might be viable but as we will see later on with neural network this becomes not a viable option when there are many training examples and features. We have to vectorise these steps.

How can we vectorise?

- Forward propagation
  - $z^{(1)} = w^T x^{(1)} + b$
  - $a^{(1)} = \sigma(z^{(1)})$
  - $z^{(2)} = w^T x^{(2)} + b$
  - $a^{(2)} = \sigma(z^{(2)})$
  - …
- Vectorized form of forward propagation:
  - $z = w^T X + b$
    - $z \in \mathbb{R}^m, z = \left[z^{(1)}, \dots, z^{(m)}\right]$
    - $w \in \mathbb{R}^{n_x}, w = [w_1, \dots, w_n]$
    - $X \in \mathbb{R}^{n_x \cdot m}, X = \left[x^{(1)}, \dots, x^{(m)}\right]$
    - $b \in \mathbb{R}^m, b = [b, b \dots, b]$

AIMLab.

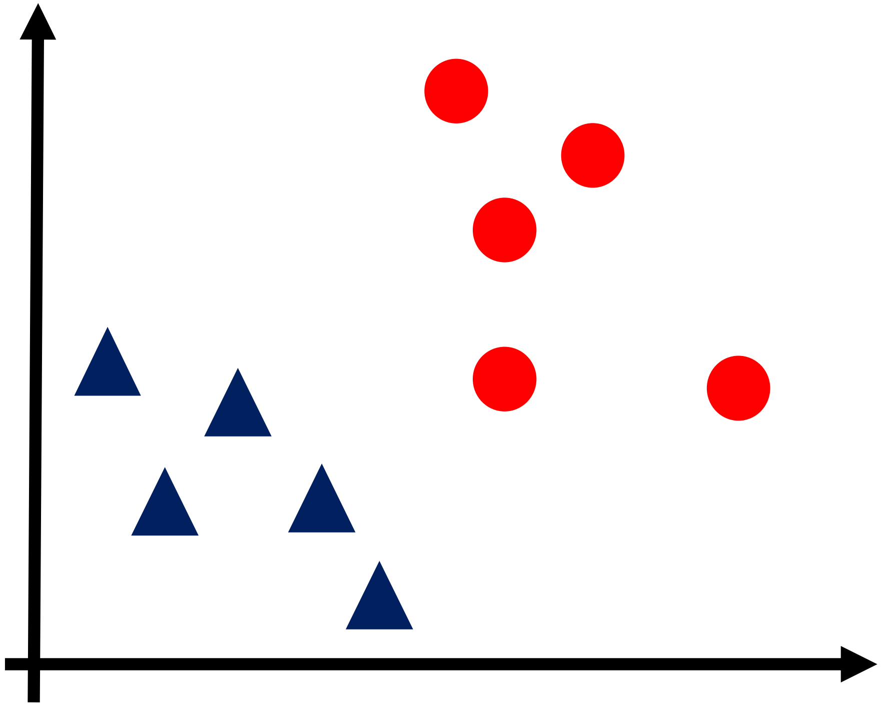
How can we vectorise?

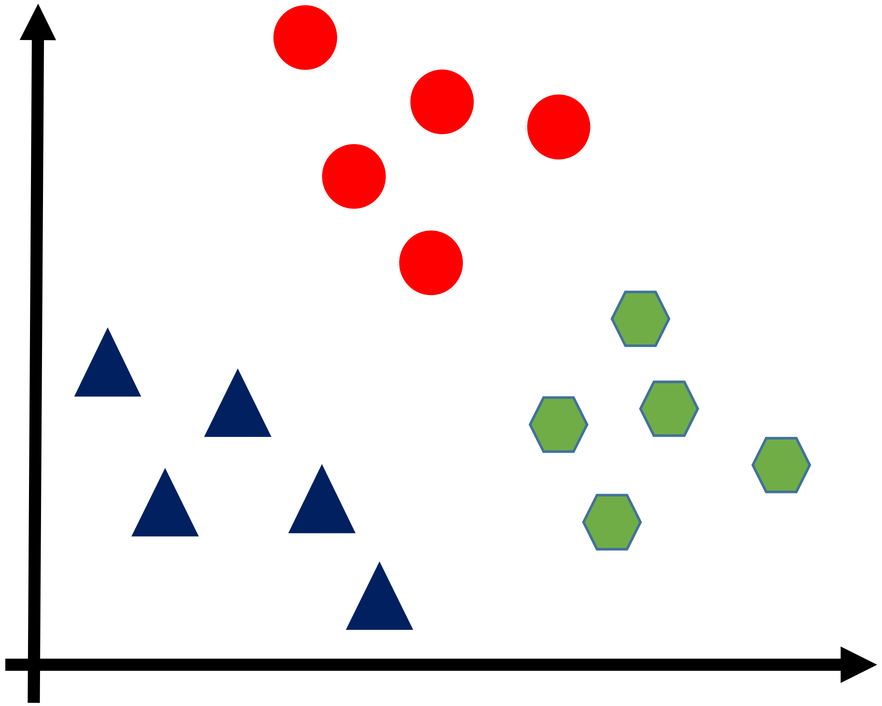
- Backward propagation:
  - $w_j := w_j - \frac{1}{m}\sum_{i=1}^{m} \alpha\left(a^{(i)} - y^{(i)}\right)x_j^{(i)}$, $\forall j \in [1, n]$
- Vectorized backward propagation:
  - $w := w - \frac{1}{m} X \left(\underline{a} - \underline{y}\right)$
  - $\underline{a} \in \mathbb{R}^m, \underline{y} \in \mathbb{R}^m, X \in \mathbb{R}^{n_x \cdot m}$
- In conclusion, **the Vectorized form of LR gradient descent**:
  - $z = w^T X + b$ $\quad (forward\ step)$
  - $w := w - \frac{1}{m} X \left(\underline{a} - \underline{y}\right)$ $\quad (backward\ step)$
  - $b := b - \frac{1}{m}\sum_{i=1}^{m} \alpha\left(a^{(i)} - y^{(i)}\right)$ $\quad (backward\ step)$

Multiclass Classification

Multiclass classification

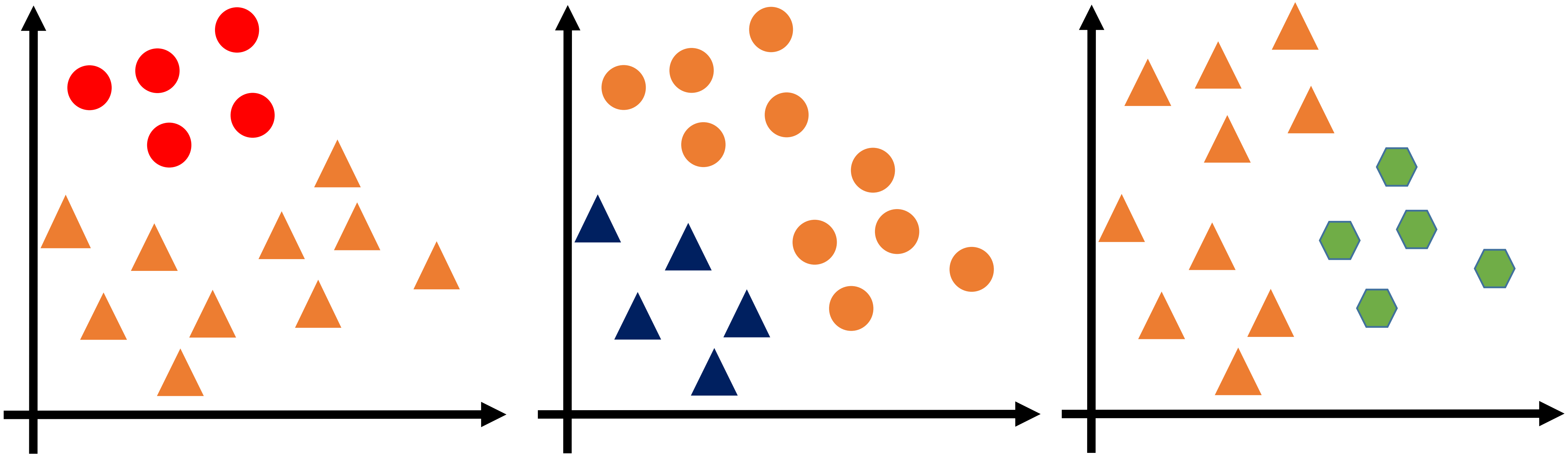
- We are now interested in a problem where the output is not binary.
- Consider the arrhythmia example. Say we now want to distinguish between categories: AF, other ARR and NSR.





Multiclass classification

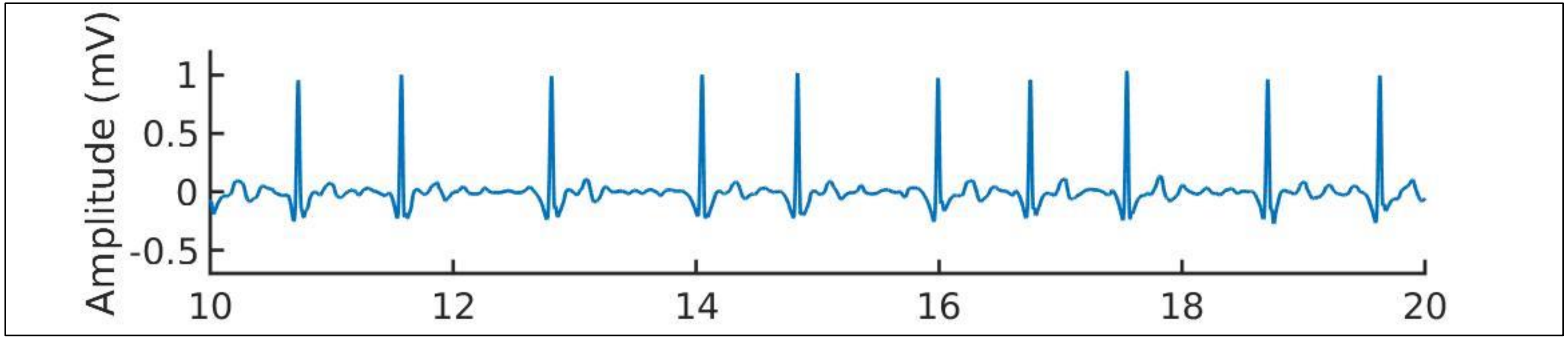
- How do we do that?
- “One vs. all” also called “one vs. the rest” approach.
- Transform the problem into a set of 2-class classification problems:



$h_w^{(1)}(x) = P(y = 1|x, w)$ $\quad$ $h_w^{(2)}(x) = P(y = 2|x, w)$ $\quad$ $h_w^{(3)}(x) = P(y = 3|x, w)$

Multiclass classification

- So we come up with 3-classifiers, each classifier is trained to recognize one of the three classes.
- How do we classify a new observation x?
    - $max_i \left(h_w^{(i)}(x) \right)$
- This is called the **one-vs-all approach**.
- There a exist other approaches.



https://scikit-learn.org/stable/auto_examples/linear_model/plot_logistic_multinomial.html

Multiclass classification

- We saw the one-vs-all approach.
- Another option is using a multinomial LR:
  - Two class classification: $z = w^T x + b$
  - Multinomial: $z = Wx + b$
    - $z, b \in \mathbb{R}^{n_y}$,
    - $W \in \mathbb{R}^{n_y \cdot n_x}$,
- From the z **vector** how do we classify? Softmax activation function:
  - The usual: $z = Wx + b$
  - Now activation function: $\mathrm{a} = softmax(\mathrm{z})$
  - $\mathrm{a} = e^{\mathrm{z}} / \sum_{i=1}^{K} e^{z_i}$
- The softmax is also called normalized exponential function. It is a function that takes an input vector of size K and normalizes it into K probability distribution proportional to the exponentials of the input numbers.

Multiclass classification

- Cost function we add for the two-class classification:
  - $J(w) = \frac{1}{m}\sum_{i=1}^{m}\left[-y^{(i)} log\left(h_w\left(x^{(i)}\right)\right) - (1 - y^{(i)})\log(1 - h_w\left(x^{(i)}\right))\right],$
- For the multinomial LR:
  - $J(w) = \frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{n_y}[1\{y^{(i)} = k\}\log(h_{w_k}\left(x^{(i)}\right))],$
  - $J(w) = \frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{n_y}\left[1\{y^{(i)} = k\}\log(\frac{\exp\left(w^{(k)T}x^{(i)}\right)}{\sum_{j=1}^{K}\exp\left(w^{(k)T}x^{(i)}\right)})\right].$
  - Thus two differences with what we had previously:
    - The cost function sums over the number of classes.
    - We use the $softmax$ activation function and not σ.

Odds and odds ratio

Interpreting LR

- Say we learned our model:
  - $h_w(x) = \sigma(w^T x) = \frac{1}{1+e^{-w^T x}}$
  - This means that we have learned the weights w from the dataset.
- How do we interpret the weights value?
- Let's take the former example of AF diagnosis and the positive class as AF and negative class as no-AF.
- Let's write $p = \frac{1}{1+e^{-w^T x}}$
- Then we have:
  - $\frac{p}{1-p} = \frac{1}{e^{-w^T x}}$
  - $\log\left(\frac{p}{1-p}\right) = w^T x$

Interpreting LR

- If we assume a single feature, say blood pressure (BP) then
  - $\log\left(\frac{p}{1-p}\right) = w_0 + w_1 \cdot BP$
- Let's define the following quantity $\frac{p}{1-p}$ that we will call the *odds*.
  - p is the probability of AF and $p - 1$ is the probability of no-AF.
  - $\frac{p}{1-p}$ is the **likelihood of some events to happen** e.g. the likelihood of AF.
- On a more perhaps intuitive example, if we roll an even dice and look for the chance of obtaining a 4 then we can say that the probability of 4 is 1/6=17% or equivalently that the odds of a 4 is (1/6)/(5/6)=0.2 or odds is 1:5.
  - Probabilities: "The probability of rolling a four is 17%"
  - Odds: "For one roll of a 4 you will have 5 non-4." The odds is 1:4.
- On the AF example, say $p = 20\%$ then:
  - Probabilities: "The probability of a patient being AF is 20%".
  - Odds: "For 1 patient having AF 3 will have no AF." The odds is 1:3

Interpreting LR

- If With the previous definition of the odds, we can write:
  - $\log\left(\frac{p}{1-p}\right) = \log(odds) = w_0 + w_1 \cdot BP$
- If we write this equation for a given value of BP and one increment to the variable i.e. $BP + 1$.
  - $(1)\ \log(odds_{BP}) = w_0 + w_1 \cdot BP,$
  - $(2)\ \log(odds_{BP+1}) = w_0 + w_1 \cdot (BP + 1).$
  - $(2) - (1) = \log(odds_{BP+1}) - \log(odds_{BP}) = w_1$
    - Thus w_1 corresponds to the difference between the log odds for one unit increase in BP.
  - $e^{w_1} = \frac{odds_{BP+1}}{odds_{BP}} = \textbf{odds ratio}$
    - **Odds ratio**: relative chance od an event happening under different conditions (here one unit increase of BP).

Interpreting LR

- Say $e^{w_1} = 1.06$, how do we read that?
  - "For one increase in BP the odds of having AF will increase by 6%."
- Why do we use odds and odds ratio?
  - We use the concept of odds and odds ratio here to provide some interpretation to the weights we have learned in the LR model.
- What is we standardize features?
  - In this case the "one unit increase" will read as "one standard deviation increase".
- What about if you have many features?

AIMLab.

Interpreting LR

Contents lists available at ScienceDirect

Clinical Nutrition

journal homepage: http://www.elsevier.com/locate/clnu

ELSEVIER

Original Article

Chocolate consumption is inversely associated with prevalent coronary heart disease: The National Heart, Lung, and Blood Institute Family Heart Study

Luc Djoussé [a,b,*], Paul N. Hopkins [c], Kari E. North [d], James S. Pankow [e], Donna K. Arnett [f], R. Curtis Ellison [g]

[a] Department of Medicine, Brigham and Women's Hospital and Harvard Medical School, Boston, MA, USA
[b] Massachusetts Veterans Epidemiology and Research Information Center and Geriatric Research, Education and Clinical Center, Boston Veterans Affairs Healthcare System, Boston, MA, USA
[c] Cardiovascular Genetics, University of Utah, Salt Lake, UT, USA
[d] Department of Epidemiology, School of Public Health, University of North Carolina, Chapel Hill, NC, USA
[e] Division of Epidemiology and Community, University of Minnesota, Minneapolis, MN, USA
[f] Department of Epidemiology, University of Alabama, Birmingham, AL, USA
[g] Section of Preventive Medicine & Epidemiology, Evans Department of Medicine, Boston University School of Medicine, Boston, MA, USA

Table 2
Prevalence odds ratios (95% confidence intervals) of coronary heart disease according to chocolate consumption in 4970 participants in the NHLBI Family Heart Study[a].

Frequency of chocolate intake	Cases/N	Crude	Model 1[b]	Model 2[c]
0	168/1093	1.0	1.0	1.0
1–3 per month	147/1167	0.79 (0.62–1.01)	1.01 (0.76–1.37)	1.05 (0.77–1.43)
1–4 per week	182/1931	0.57 (0.46–0.72)	0.74 (0.56–0.98)	0.75 (0.56–1.01)
5+ per week	43/779	0.32 (0.23–0.45)	0.43 (0.28–0.67)	0.43 (0.27–0.68)
P for linear trend		<0.0001	<0.0001	0.0002

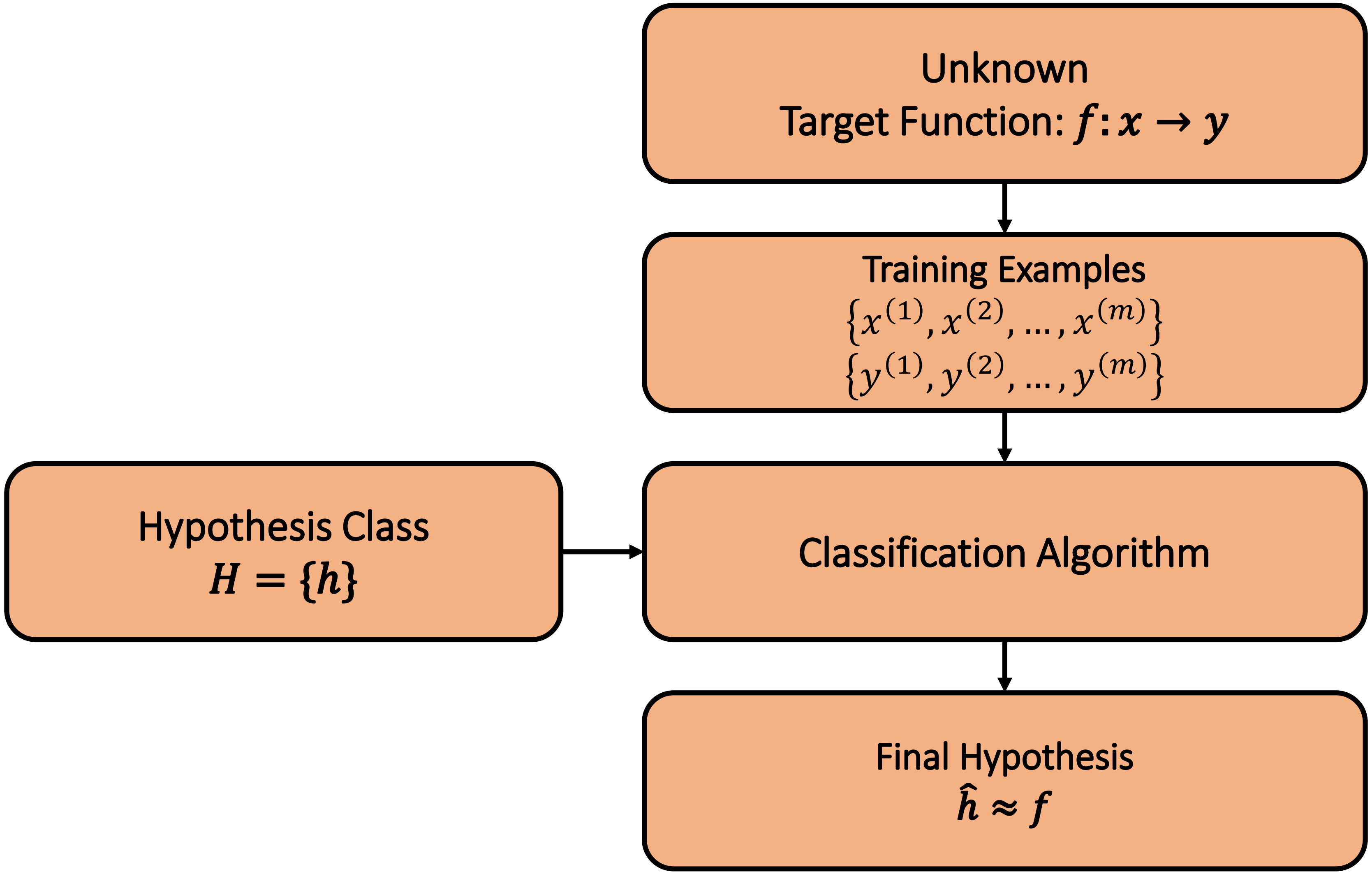
[a] Coronary heart disease was defined as history of myocardial infarction, PTCA, or CABG.
[b] Adjusted for age, sex, and risk group (random vs. high risk) using generalized estimating equations (GEE).
[c] Variables in Model 1 plus additional adjustment for dietary linolenic acid, education, exercise (min/d), smoking (yes/no), alcohol intake (yes/no), fruit and vegetables, energy intake, and non-chocolate candy (4 groups) consumption.

Linear Discriminant Analysis

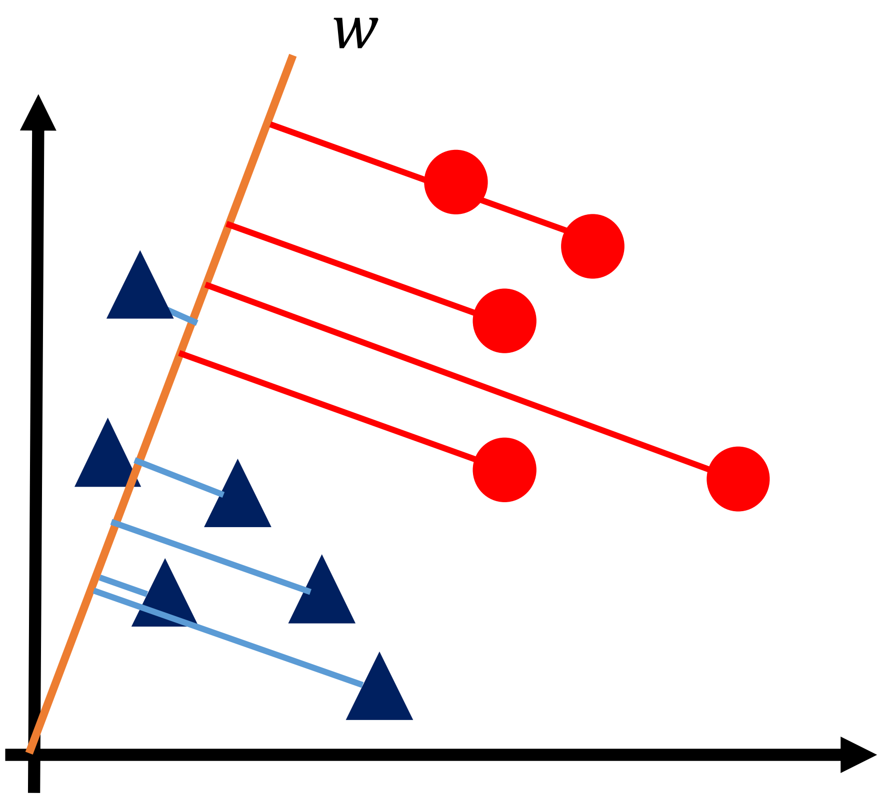
LDA

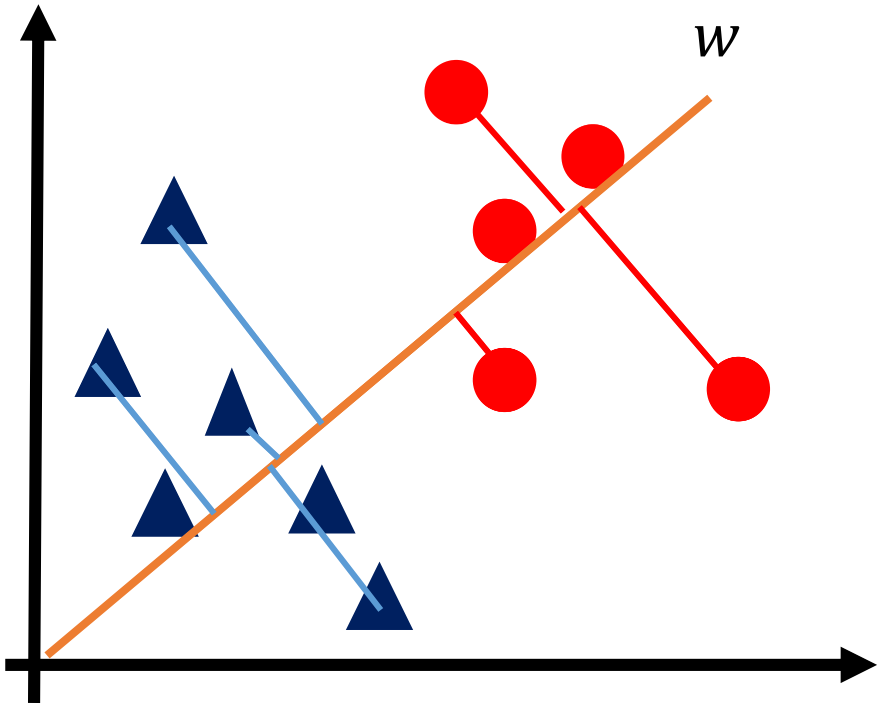
- LDA is used for classification and dimensionality reduction.
- It aims to preserve as much of the class discriminatory information as possible.
- It is often used for dimensionality reduction followed by a classification step.
- We have a training set of:
  - m examples $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$,
  - With target labels: $\{y^{(1)}, y^{(2)}, \ldots, y^{(m)}\}$.
- Hypothesis function: $h_w(x) = w^T x$
- We need to find the w that **maximize the separability of the observations**.

LDA


Unknown
Target Function: $f: x \to y$
Training Examples
$\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$
$\{y^{(1)}, y^{(2)}, \dots, y^{(m)}\}$
Hypothesis Class
$H = \{h\}$
Classification Algorithm
Final Hypothesis
$\hat{h} \approx f$

LDA





LDA

- How do we measure the **separation**?
- We need to define a measure of separation
- Let's try the difference between the means of the projected observations:
  - $J(w) = |\widetilde{\mu_1} - \widetilde{\mu_2}| = w^T(\mu_1 - \mu_2)$
  - But not so good because it does not take into account the intra classes variance.

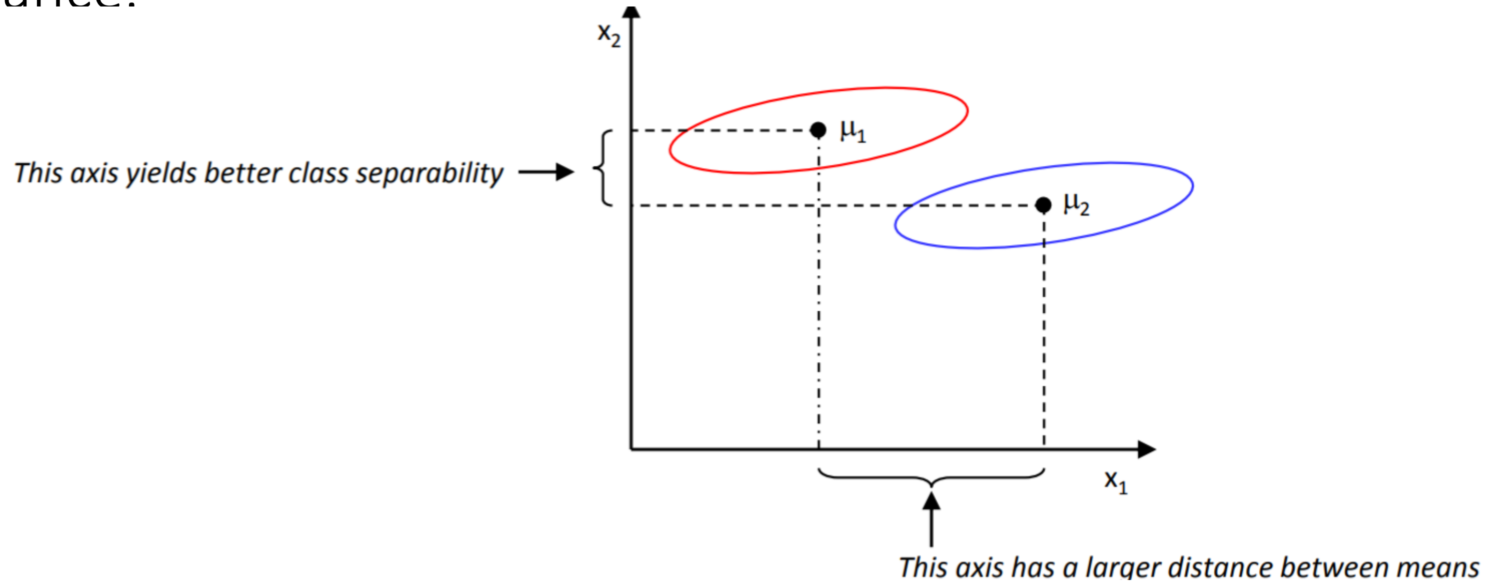


Image: CSCE 666 Pattern Analysis | Ricardo Gutierrez-Osuna | CSE@TAMU

LDA

- We can do better.
- Fisher suggested **maximizing the difference between the means, normalized by a measure of the within-class scatter**.
- For each class:
  - $\tilde{s}_j^2 = \sum_{i=1}^{m_j}(y_j^{(i)} - \tilde{\mu}_j)^2$
- Overall cost function:
  - $J(w) = \frac{|\tilde{\mu}_1 - \tilde{\mu}_2|^2}{\tilde{s}_1^2 + \tilde{s}_2^2}$
  - We look for a projection where observations from the same class are projected very close to each other but where the projected means are as far as possible from each other.

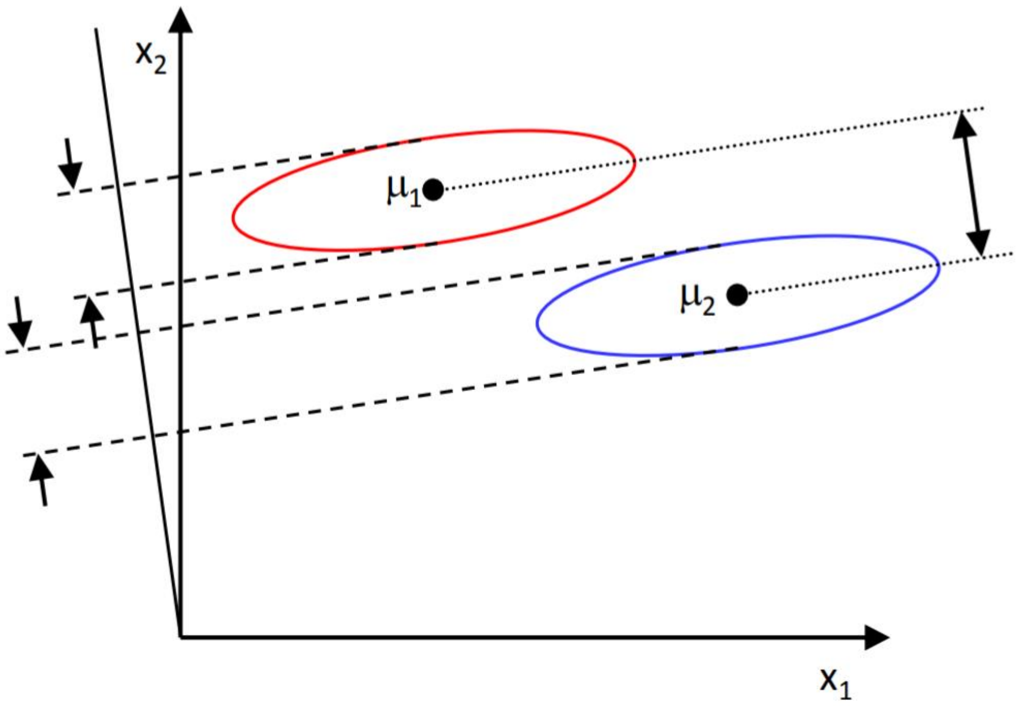
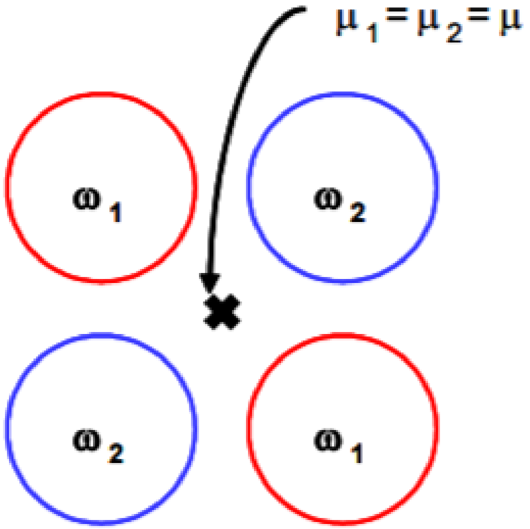


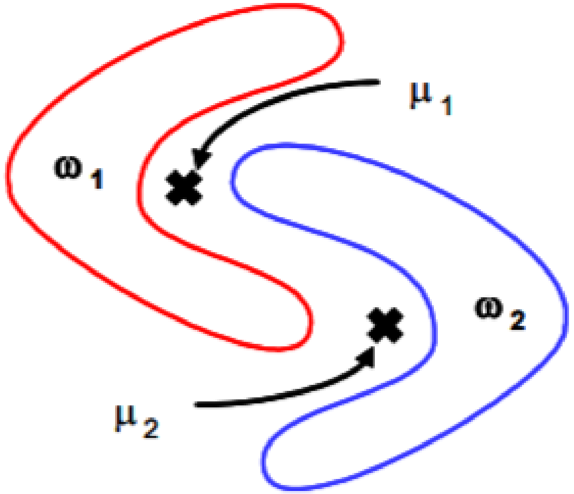
Image: CSCE 666 Pattern Analysis | Ricardo Gutierrez-Osuna | CSE@TAMU

LDA

- Now that we have defined the overall cost function, how do we solve for w?
- We can show (make the demo) that the solution is given by:
  - $w = S_w^{-1}(\mu_1 - \mu_2)$
  - $S_w = S_1 + S_2$
  - $S_j = \sum_{i=1}^{m_j}(x_j^{(i)} - \mu_j)(x_j^{(i)} - \mu_j)$
  - So we have a closed form solution of the LDA problem.
- If we want to use LDA for classification"
  - $h_w(x) = w^T x$
  - An example x belongs to class c if $h_w(x) > t$ where t is the decision threshold.
- Limitation. By assuming $J(w) = \frac{|\tilde{\mu}_1 - \tilde{\mu}_2|^2}{\tilde{s}_1^2 + \tilde{s}_2^2}$ we intrinsically assumed that the independent variables are normally distributed and this is a fundamental assumption of LDA and a fundamental limitation of it!
- LDA can be generalized to multiple classes.

LDA Limitations





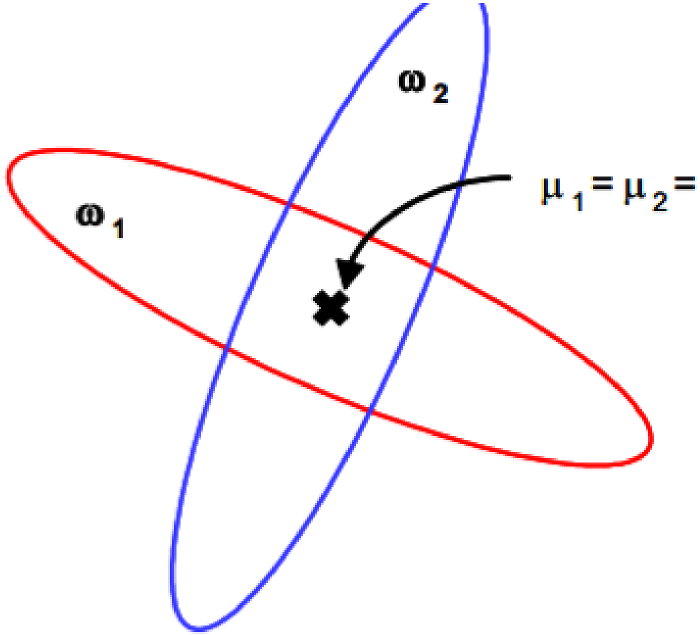


Image: CSCE 666 Pattern Analysis | Ricardo Gutierrez-Osuna | CSE@TAMU

Take home

- Classification versus regression
- Logistic Regression (LR) **hypothesis representation, cost function**
- **Convexity** of the overall cost function
- Gradient descent
- Multiclass classification: **one vs. all**
- LR is one of the most popular classification algorithm. Use it as a baseline before moving to more complex models.
  - Advantages: efficient, interpretable, outputs probabilities.
  - Drawback: cannot solve non-linear problems since the LR decision surface is linear.
- **Linear Discriminant Analysis (LDA)**
  - Reduce dimensionality while preserving as much of the class discriminatory information as possible.
  - Used for dimensionality reduction and classification.
  - Assumes independent variables are normally distributed.

References

[1] Andrew Ng, Coursera, Machine Learning. Coursera.
[2] Andrew Ng, Coursera, Neural Networks and Deep Learning. Coursera.
[3] CSCE 666 Pattern Analysis | Ricardo Gutierrez-Osuna | CSE@TAMU
URL: http://research.cs.tamu.edu/prism/lectures/pr/pr_l10.pdf

Classification versus Regression

- Given HRV features decide whether the individual has AF or not.
- Regression model and threshold at the hypothesis function being 0.5 i.e. $h_w(x) = 0.5$?
- Sometime it may be ok.
- But most of the time this is not appropriate.