ml

Tanvir

November 2025

(data + learning algorithm) → function

1 Supervised learning

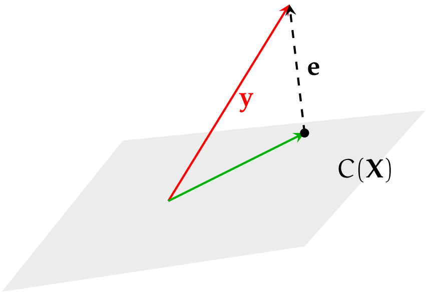
1.1 Regression

model, parameters, cost function, objective

1.1.1 Linear (in w) Regression

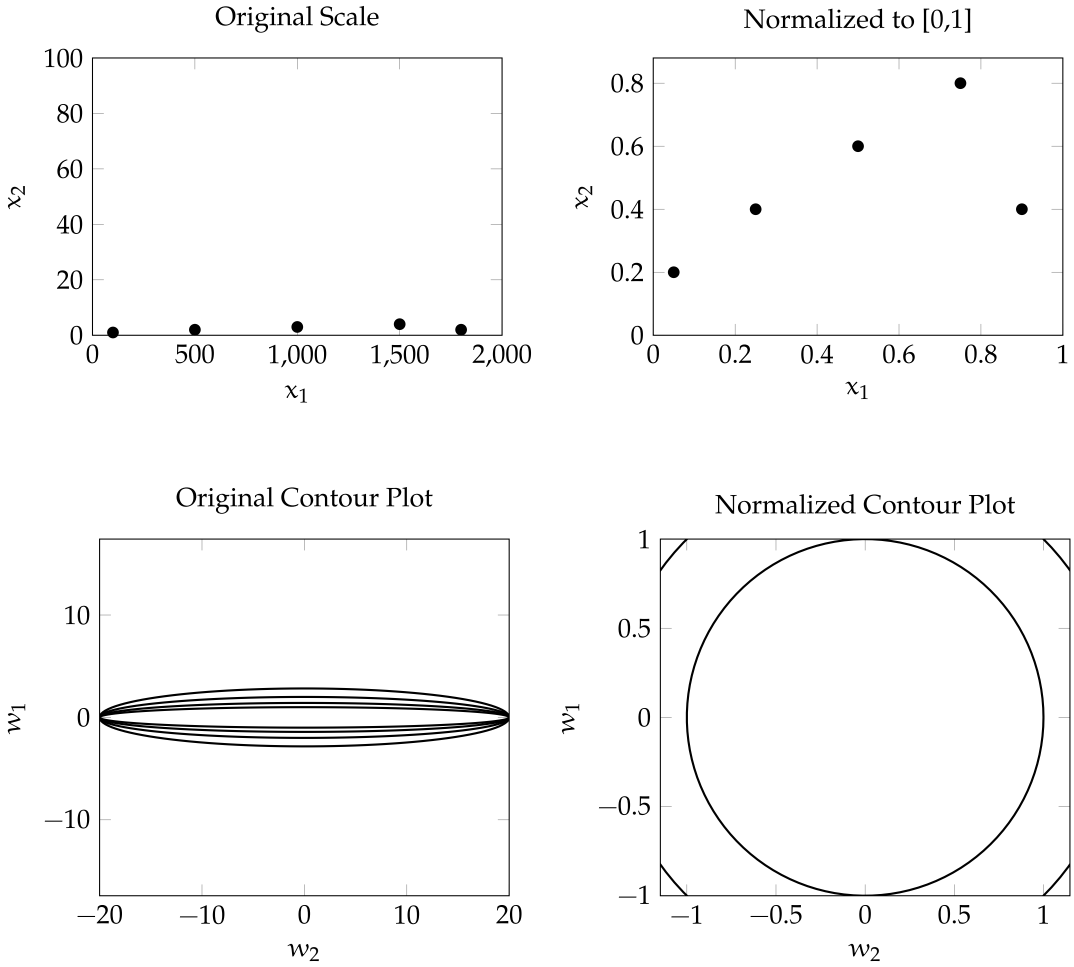
$\mathbf{X}^{m \times n}$ has m examples, each having n features. Usually $m >> n$. $\mathbf{y}^{m \times 1}$ are the corresponding outputs.

$$
\begin{aligned}
\mathbf{X}\mathbf{w} &\approx \mathbf{y} \\
\mathbf{e} &= \mathbf{X}\mathbf{w} - \mathbf{y} \\
\underset{\mathbf{w}}{\text{minimize}}\ J(\mathbf{w}) &= \|\mathbf{e}\|^2 = (\mathbf{X}\mathbf{w} - \mathbf{y})^T(\mathbf{X}\mathbf{w} - \mathbf{y}) \\
\text{gradient descent: } \mathbf{w} &= \mathbf{w} - 2\alpha\mathbf{X}^T(\mathbf{X}\mathbf{w} - \mathbf{y})
\end{aligned}
$$



1.2 Normalization

Changes scale keeping the shape of distribution same. Gradient descent now works, with more ease, in the world of concentric circular contours.



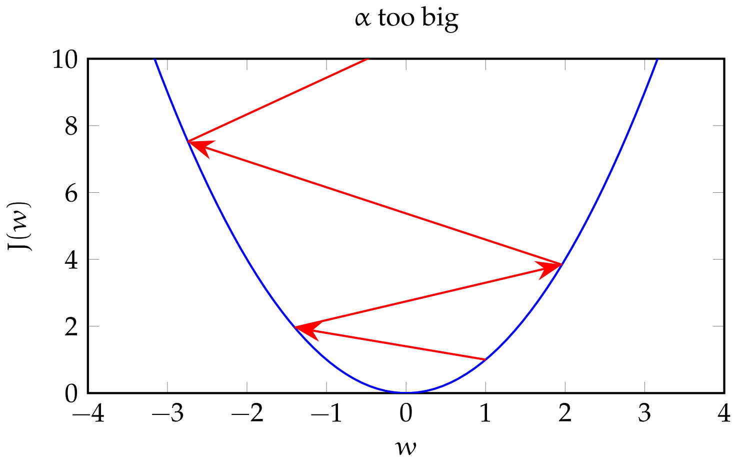
Original Scale

Normalized to [0,1]

Original Contour Plot

Normalized Contour Plot

1.3 Learning rate, α

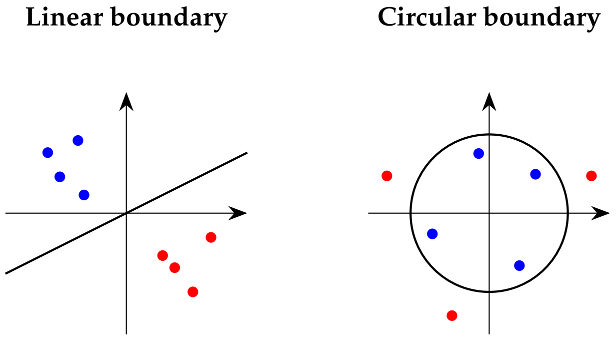

α too big

1.4 Classification

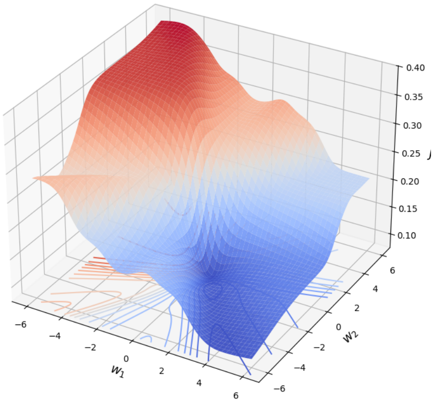
1.4.1 Logistic Regression

$$
\begin{aligned}
\mathbf{z} &= \mathbf{X}\mathbf{w}, && \texttt{// linear regression} \\
\sigma(\mathbf{z}) &= \frac{1}{1+e^{-\mathbf{z}}} && \texttt{// maps z to probability-like (0,1)}
\end{aligned}
$$

At $\mathbf{z} = 0$, $\sigma(\mathbf{z}) = 0.5$. So, with threshold 0.5, $\mathbf{z} = 0$ is the decision boundary. As linear regression doesn't have to be straight line, logistic regression's decision boundary does not have to be a straight line.


Linear boundary **Circular boundary**

The squared error loss function $L(\mathbf{w}, y_i) = \left(\sigma(\mathbf{w}^T\mathbf{x}_i) - y_i\right)^2$, makes the cost function non-convex.

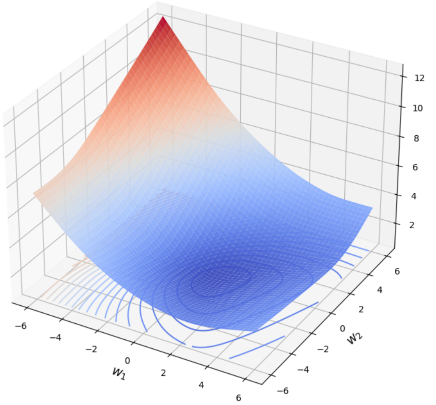


Negative log likelihood loss:

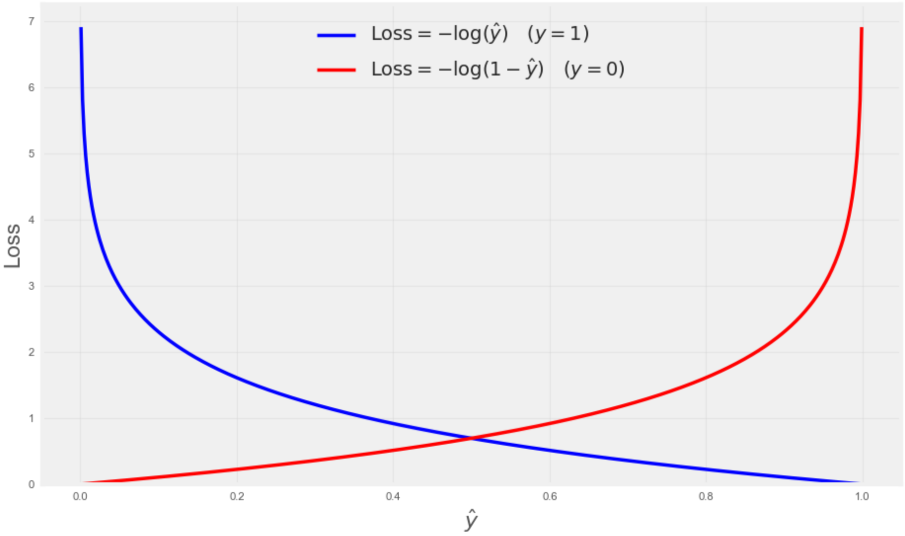
$$\hat{y}_i = \sigma\left(\mathbf{w}^T\mathbf{x}_i\right) \in (0, 1)$$

$$J(\mathbf{w}) = \frac{1}{m}\sum_{i=1}^{m} \begin{cases} -\log(\hat{y}_i), & \text{if } y_i = 1, \\ -\log(1 - \hat{y}_i), & \text{if } y_i = 0 \end{cases}$$

The negative log likelihood loss function with $\sigma(z)$ makes the cost function convex.



Negative log likelihood also incentivizes appropriately: big mismatch $\implies$ big loss, small mismatch $\implies$ small loss.


Loss $= -\log(\hat{y})$ $(y = 1)$
Loss $= -\log(1 - \hat{y})$ $(y = 0)$
Loss
$\hat{y}$