CSCC11 Assignment 2

Kevin Li

March 10, 2023

1. Without loss of generality, let i = 1 and let $\sigma_1 = \sigma_2 = \sigma$

$$\begin{aligned}
p(x|c_1) &\sim Normal(\mu_1, \sigma) \\
\Rightarrow p(x|c_1) &= (\sigma\sqrt{2\pi})^{-1} e^{\frac{-1}{2}(\frac{x-\mu_1}{\sigma})^2} \\
\Rightarrow p(c_1|x) &= \frac{p(x|c_1)p(c_1)}{p(x|c_1)p(c_1) + p(x|c_2)p(c_2)} \\
&= \frac{1}{1 + (\frac{p(x|c_2)p(c_2)}{p(x|c_1)|p(c_1)})} \\
&= \frac{1}{1 + e^{-\ln(\frac{p(x|c_1)p(c_1)}{p(x|c_2)p(c_2)})}}
\end{aligned}$$

Suppose the prior probabilities $p(c_1), p(c_2)$ are constant. Then, the decision boundary is:

$$\begin{aligned}
\ln(\frac{p(x|c_1)p(c_1)}{p(x|c_2)p(c_2)}) &= \ln(\frac{p(x|c_1)}{p(x|c_2)}) + \ln(\frac{p(c_1)}{p(c_2)}) \\
&= \ln(\frac{(\sigma\sqrt{2\pi})^{-1} e^{\frac{-1}{2}(\frac{x-\mu_1}{\sigma})^2}}{(\sigma\sqrt{2\pi})^{-1} e^{\frac{-1}{2}(\frac{x-\mu_2}{\sigma})^2}}) + k, k = \ln(Odds(c_1)) \\
&= \ln(e^{\frac{-1}{2}(\frac{x-\mu_1}{\sigma})^2}) - \ln(e^{\frac{-1}{2}(\frac{x-\mu_2}{\sigma})^2}) + k \\
&= -\frac{1}{2\sigma}((x-\mu_1)^2 - (x-\mu_2)^2) + k \\
&= -\frac{1}{2\sigma}(\mu_1^2 - \mu_2^2 + 2\mu_2 x - 2\mu_1 x) + k
\end{aligned}$$

which is a linear function given μ and σ.

2. Minimize

$$\begin{aligned}
Loss(w) &= \sum_{i=1}^{n} -y_i \log(P(c_1|x_i)) - (1-y_i)\log(1 - P(c_1|x_i)) \\
&= \sum_{i=1}^{n} -class(y_i)\log(P(x_i = Orange|x_i)) \\
&- (1 - class(y_i))\log(1 - P(x_i = Orange|x_i))
\end{aligned}$$

The parameter to be estimated is the weight vector w where
$P(Orange|x_i) = (1 + e^{-w^T x - b})^{-1}$
Gradient Descent of Step size $\lambda = 0.01$:

(a) Training data predictions for w_1:

- $g(w^T x_1) = g([0.3, -0.2, 0.7] \begin{bmatrix} 4 \\ 4 \\ 1 \end{bmatrix}) = 0.75$, Guess: Orange, PASS
- $g(w^T x_2) = g([0.3, -0.2, 0.7] \begin{bmatrix} 4 \\ 6 \\ 1 \end{bmatrix}) = 0.84$, Guess: Orange, PASS
- $g(w^T x_3) = 0.81$, Guess: Orange, PASS
- $g(w^T x_4) = 0.71$, Guess: Orange, FAIL
- $g(w^T x_5) = 0.62$, Guess: Orange, FAIL
- $g(w^T x_6) = 0.82$, Guess: Orange, PASS
- $g(w^T x_7) = 0.75$, Guess: Orange, FAIL

[1, 1, 1, 1, 1, 1, 1]
Accuracy of predictions: $\frac{4}{7}$
Initial vector:
$w_1 = \begin{bmatrix} 0.3 \\ -0.2 \\ 0.7 \end{bmatrix}$

$$\begin{aligned} \frac{\partial L(w)}{\partial w} &= -\sum_{i=1}^{n} (y_i - p_i) x_i \\ &= -(1 - g(\begin{bmatrix} 0.3 & -0.2 & 0.7 \end{bmatrix} \begin{bmatrix} 4 \\ 4 \\ 1 \end{bmatrix})) \begin{bmatrix} 4 \\ 4 \\ 1 \end{bmatrix} - \dots \\ &= -(1 - 0.75) \begin{bmatrix} 4 \\ 4 \\ 1 \end{bmatrix} - (1 - 0.85) \begin{bmatrix} 6 \\ 4 \\ 1 \end{bmatrix} - \dots \\ &= \begin{bmatrix} 9.52 \\ 15.43 \\ 1.31 \end{bmatrix} \end{aligned}$$

$$w_2 = \begin{bmatrix} 0.3 \\ -0.2 \\ 0.7 \end{bmatrix} - 0.01 \begin{bmatrix} 9.52 \\ 15.43 \\ 1.31 \end{bmatrix} = \begin{bmatrix} 0.20 \\ -0.35 \\ 0.69 \end{bmatrix}$$

Training data predictions for w_2: [1, 1, 1, 0, 0, 0, 0]
Accuracy of predictions: $\frac{6}{7}$

(b) Initial vector:

$$w_2 = \begin{bmatrix} 0.20 \\ -0.35 \\ 0.69 \end{bmatrix}$$

$$\begin{aligned}
\frac{\partial L(w)}{\partial w} &= -\sum_{i=1}^{n} (y_i - p_i) x_i \\
&= -(1 - g(\begin{bmatrix} 0.20 & -0.35 & 0.69 \end{bmatrix} \begin{bmatrix} 4 \\ 4 \\ 1 \end{bmatrix})) \begin{bmatrix} 4 \\ 4 \\ 1 \end{bmatrix} - \dots \\
&= \begin{bmatrix} -7.44 \\ -4.53 \\ -1.27 \end{bmatrix}
\end{aligned}$$

$$w_3 = \begin{bmatrix} 0.20 \\ -0.35 \\ 0.69 \end{bmatrix} - 0.01 \begin{bmatrix} -7.44 \\ -4.53 \\ -1.27 \end{bmatrix} = \begin{bmatrix} 0.28 \\ -0.30 \\ 0.70 \end{bmatrix}$$ Training data predictions for w_3: [1, 1, 1, 0, 0, 1, 0]

Accuracy of predictions: $\frac{7}{7}$

(c) Initial vector:

$$w_3 = \begin{bmatrix} 0.28 \\ -0.31 \\ 0.70 \end{bmatrix}$$

$$\begin{aligned}
\frac{\partial L(w)}{\partial w} &= -\sum_{i=1}^{n} (y_i - p_i) x_i \\
&= -(1 - g(\begin{bmatrix} 0.28 & -0.31 & 0.70 \end{bmatrix} \begin{bmatrix} 4 \\ 4 \\ 1 \end{bmatrix})) \begin{bmatrix} 4 \\ 4 \\ 1 \end{bmatrix} - \dots \\
&= \begin{bmatrix} 0.70 \\ 4.68 \\ -0.03 \end{bmatrix}
\end{aligned}$$

$$w_4 = \begin{bmatrix} 0.28 \\ -0.31 \\ 0.70 \end{bmatrix} - 0.01 \begin{bmatrix} 0.70 \\ 4.68 \\ -0.03 \end{bmatrix} = \begin{bmatrix} 0.27 \\ -0.36 \\ 0.70 \end{bmatrix}$$

Final estimate: $w_4 = \begin{bmatrix} 0.27 \\ -0.36 \\ 0.70 \end{bmatrix}$

Training data predictions for w_4: [1, 1, 1, 0, 0, 1, 0]

Accuracy of predictions: $\frac{7}{7}$

Predictions using w_4:

- $g(w^T x_1) = g([0.27, -0.36, 0.70] \begin{bmatrix} 3 \\ 3 \\ 1 \end{bmatrix}) = 0.61$, Guess: Orange
- $g(w^T x_2) = g([0.27, -0.36, 0.70] \begin{bmatrix} 4 \\ 10 \\ 1 \end{bmatrix}) = 0.14$, Guess: Not Orange
- $g(w^T x_3) = g([0.27, -0.36, 0.70] \begin{bmatrix} 9 \\ 8 \\ 1 \end{bmatrix}) = 0.56$, Guess: Orange
- $g(w^T x_4) = g([0.27, -0.36, 0.70] \begin{bmatrix} 9 \\ 10 \\ 1 \end{bmatrix}) = 0.38$, Guess: Not Orange

Logistic regression is very simple and efficient to train datasets that are linearly seperable. However, it only works for linearly seperable data - nonlinear datasets will require a transformation to a different feature space. It is a discriminative model since the joint probability $P(Orange, x)$ is never considered; only $P(Orange|x)$ is fitted.