Logistic Regression

S. R. M. Prasanna

Dept of Electrical Engineering
Indian Institute of Technology Dharwad

prasanna@iitdh.ac.in

November 26, 2020

Acknowledgements

- Artificial Neural Networks by Prof. B. Yegnanarayana, PHI, 1999
- Machine learning video lectures by Prof. Andrew Ng, Stanford Uty
- Jason Brownlee "Basics of Linear Algebra for Machine Learning", Online book, 2018.

Motivation for Logistic Regression

- Linear vs Non-linear regression.
- Classification: $y = \{0, 1\}$ in binary classification and $y = \{0, 1, 2, \ldots, \}$ in case of multiclass classification.
- Binary Classification: $h_w(x)$ by linear regression and thresholding, if $h_w(x) >= 0.5, y = 1$ and if $h_w(x) < 0.5, y = 0$
- May not work in all cases, may lead to miss-classification.
- $h_w(x)$ may be > 1 or < 0, even though output can be 0 or 1.
- Linear regression is not advisable for classification task.
- Hence need for non-linear regression, ex, logistic regression.

Logistic Regression

- $0 \leq h_w(x) \leq 1$
- $h_w(x) = g(w^T x)$, where, $g(z) = \frac{1}{1+e^{-z}}$
- $g(z)$ is sigmoidal or logistic function.
- $h_w(x) = \frac{1}{1+e^{-w^T x}}$
- $h_w(x) = p(y = 0|x, w)$ vs $h_w(x) = p(y = 1|x, w)$
- $p(y = 0|x, w) + p(y = 1|x, w) = 1.0$
- $p(y = 0|x, w) = 1 - p(y = 1|x, w)$

Decision Boundary from $h_w(x)$

- Predict $y = 1$ when $h_w(x) = g(w^T x) \geq 0.5$
- Similarly, predict $y = 0$ when $h_w(x) = g(w^T x) < 0.5$
- When $z \geq 0$, then $g(z) \geq 0.5$ and $z < 0$, then $g(z) < 0.5$
- When $w^T x \geq 0$, then $g(W^T x) \geq 0.5$ and $w^T x < 0$, then $g(W^T x) < 0.5$

Decision Boundary using $h_w(x)$

- Set of examples containing binary class labelled data are given.
- Let $h_w(x) = w_0 + w_1 x = -1 + x$
- When $W^T x \geq 0$, then $h_w(x) = p(y = 1|x, w) \geq 0.5$
- $-1 + x \geq 0$ or $x \geq 1$, then $h_w(x) = p(y = 1|x, w) \geq 0.5$
- $x = 1$ is the decision boundary.
- Let $h_w(x) = w_0 + w_1 x_1 + w_2 x_2 = -2 + x_1 + x_2$
- When $w^T x \geq 0$, then $h_w(x) = p(y = 1|x, w) \geq 0.5$
- $-2 + x_1 + x_2 \geq 0$ or $x_1 + x_2 \geq 2$, then $h_w(x) = p(y = 1|x, w) \geq 0.5$
- $x_1 + x_2 = 2$ is the decision boundary.

About Decision Boundary

- Decision boundary is a straight line, which is linear.
- How to get non-linear decision surface?
- Let $h_w(x) = -1 + 0x_1 + 0x_2 + x_1^2 + x_2^2$
- $x_1^2 + x_2^2 = 1$
- Decision boundary will be circle with radius 1, which is a non-linear decision boundary.
- Along the same lines we can consider higher order polynomials to consider even more complex non-linear decision boundaries.
- Example, boundary for a plot of land !

Cost Function: Estimating Parameters $h_w(x)$

- Given $h_w(x)$. Is it w dependent or x dependent or both?
- Given, $(X, Y) = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(M)}, y^{(M)})\}$
- $x^{(i)}$ is $(N+1)$ dim vector and $y = 0 or 1$ is a scalar.
- How to estimate parameters of $h_w(x)$?
- Cost fn: $C(w) = \frac{1}{2M}\sum_{i=1}^{M}(h_w(x^{(i)}) - y^{(i)})^2$
- Cost fn: $C(w) = \frac{1}{M}\sum_{i=1}^{M}\frac{1}{2}(h_w(x^{(i)}) - y^{(i)})^2$
- Cost fn: $C(w) = \frac{1}{M}\sum_{i=1}^{M} cost(h_w(x^{(i)}), y^{(i)})$
- where, $cost(h_w(x^{(i)}), y^{(i)}) = \frac{1}{2}(h_w(x^{(i)}) - y^{(i)})^2$

Cost Function: Nature of $C(w) vs w$

- In case of logistic regression, $C(w) vs w$ will be **non-convex** function.
- In case of linear regression, $C(w) vs w$ was convex function.
- Gradient descent in logistic regression may lead to local minima.
- Can we have better functions to avoid local minima?
- where, $cost(h_w(x^{(i)}), y^{(i)}) = -log(h_w(x))$ if $y = 1$ and
- $cost(h_w(x^{(i)}), y^{(i)}) = -log(1 - h_w(x))$ if $y = 0$
- $-log(h_w(x)) vs h_w(x)$; 0 at $y = 1$ and ∞ at $y = 0$.
- $-log(1 - h_w(x)) vs h_w(x)$; 0 at $y = 0$ and ∞ at $y = 1$.

Modified Cost Function

- where, $cost(h_w(x), y) = -ylog(h_w(x)) - (1-y)log(1-h_w(x))$.
- $y = 1$ first term and $y = 0$ second term.
- Cost fn: $C(w) = \frac{1}{M}\sum_{i=1}^{M} \frac{1}{2}(h_w(x^{(i)} - y^{(i)})^2$
- Cost fn: $C(w) = \frac{1}{M}\sum_{i=1}^{M} cost(h_w(x^{(i)}), y^{(i)})$
- Cost fn:
  $C(w) = -\frac{1}{M}\sum_{i=1}^{M} y^{(i)}log(h_w(x^{(i)})) + (1-y^{(i)})log(1-h_w(x^{(i)}))$
- $\frac{\partial}{\partial w_j}C(w) = \frac{1}{M}\sum_{i=1}^{M}(h_w(x^{(i)}) - y^{(i)})x_j^{(i)}$

Gradient Descent for Minimization

- $\min_{w} C(w)$
- w by Gradient Descent:
- $w_0 := w_0 - \alpha \frac{1}{M} \sum_{i=1}^{M} (h_w(x^{(i)}) - y^{(i)}) x_0^{(i)}$
- $w_1 := w_1 - \alpha \frac{1}{M} \sum_{i=1}^{M} (h_w(x^{(i)}) - y^(i)) x_1^{(i)}$
- $w_N := w_N - \alpha \frac{1}{M} \sum_{i=1}^{M} (h_w(x^{(i)}) - y^{(i)}) x_N^{(i)}$
- Repeat the above $(N+1)$ steps until convergence
- Looks similar to linear regression, with $h_w(x^{(i)}) = w^T x^{(i)}$ for linear and $h_w(x^{(i)}) = \frac{1}{1+e^{-w^T x^{(i)}}}$

Derivation of $\frac{\partial}{\partial w_j} C(w)$

- $C(w) = -\frac{1}{M} \sum_{i=1}^{M} y^{(i)} log(h_w(x^{(i)})) + (1 - y^{(i)}) log(1 - h_w(x^{(i)}))$

- $\frac{\partial}{\partial w_j} C(w) = -\frac{\partial}{\partial w_j} \frac{1}{M} \sum_{i=1}^{M} y^{(i)} log(h_w(x^{(i)})) + (1 - y^{(i)}) log(1 - h_w(x^{(i)}))$

- $\frac{\partial}{\partial w_j} C(w) =$
  $-\frac{1}{M} \sum_{i=1}^{M} y^{(i)} \frac{\partial}{\partial w_j} (log(h_w(x^{(i)}))) + (1 - y^{(i)}) \frac{\partial}{\partial w_j} (log(1 - h_w(x^{(i)})))$

- $\frac{\partial}{\partial w_j} (log(h_w(x^{(i)}))) = \frac{1}{h_w(x^{(i)})} \frac{\partial}{\partial w_j} (h_w(x^{(i)}))$

- $\frac{\partial}{\partial w_j} (log(1 - h_w(x^{(i)}))) = \frac{1}{1 - h_w(x^{(i)})} \frac{\partial}{\partial w_j} (1 - h_w(x^{(i)}))$

- $h_w(x^{(i)}) = \frac{1}{1 + e^{-w^T x^{(i)}}} \implies \frac{1}{h_w(x^{(i)})} = 1 + e^{-w^T x^{(i)}}$ and

- $\frac{1}{1 - h_w(x^{(i)})} = \frac{1 + e^{-w^T x^{(i)}}}{e^{-w^T x^{(i)}}}$

Derivation of $\frac{\partial}{\partial w_j} C(w)$

- $\frac{\partial}{\partial w_j} h_w(x^{(i)}) = \frac{\partial}{\partial w_j}(\frac{1}{1+e^{-w^T x^{(i)}}}) = \frac{0-e^{-w^T x^{(i)}}(-x_j^{(i)})}{(1+e^{-w^T x^{(i)}})^2} = \frac{e^{-w^T x^{(i)}} x_j^{(i)}}{(1+e^{-w^T x^{(i)}})^2}$
- $\frac{\partial}{\partial w_j}(1 - h_w(x^{(i)})) = \frac{\partial}{\partial w_j}(1 - \frac{1}{1+e^{-w^T x^{(i)}}}) = \frac{\partial}{\partial w_j} \frac{e^{-w^T x^{(i)}}}{1+e^{-w^T x^{(i)}}} =$
  $\frac{e^{-w^T x^{(i)}}(-x_j^{(i)})(1+e^{-w^T x^{(i)}})-e^{-w^T x^{(i)}}(-x_j^{(i)})e^{-w^T x^{(i)}}}{(1+e^{-w^T x^{(i)}})^2} = \frac{-e^{-w^T x^{(i)}} x_j^{(i)}}{(1+e^{-w^T x^{(i)}})^2}$
- $\frac{1}{h_w(x^{(i)})} \frac{\partial}{\partial w_j}(h_w(x^{(i)})) = (1 + e^{-w^T x^{(i)}}) \frac{e^{-w^T x^{(i)}} x_j^{(i)}}{(1+e^{-w^T x^{(i)}})^2} = \frac{e^{-w^T x^{(i)}} x_j^{(i)}}{(1+e^{-w^T x^{(i)}})}$
- $\frac{1}{1-h_w(x^{(i)})} \frac{\partial}{\partial w_j}(1 - h_w(x^{(i)})) = (\frac{1+e^{-w^T x^{(i)}}}{e^{-w^T x^{(i)}}}) \frac{-e^{-w^T x^{(i)}} x_j^{(i)}}{(1+e^{-w^T x^{(i)}})^2} = \frac{-x_j^{(i)}}{(1+e^{-w^T x^{(i)}})}$

Derivation of $\frac{\partial}{\partial w_j} C(w)$

- $\frac{\partial}{\partial w_j} C(w) =$
  $-\frac{1}{M} \sum_{i=1}^{M} y^{(i)} \frac{\partial}{\partial w_j} (log(h_w(x^{(i)}))) + (1 - y^{(i)}) \frac{\partial}{\partial w_j} (log(1 - h_w(x^{(i)})))$

- $\frac{\partial}{\partial w_j} C(w) = -\frac{1}{M} \sum_{i=1}^{M} y^{(i)} \frac{e^{-w^T x^{(i)}} x_j^{(i)}}{(1+e^{-w^T x^{(i)}})} + (1 - y^{(i)}) \frac{-x_j^{(i)}}{(1+e^{-w^T x^{(i)}})}$

- $\frac{\partial}{\partial w_j} C(w) = -\frac{1}{M} \sum_{i=1}^{M} y^{(i)} x_j^{(i)} \frac{(1+e^{-w^T x^{(i)}})}{(1+e^{-w^T x^{(i)}})} + \frac{-x_j^{(i)}}{(1+e^{-w^T x^{(i)}})}$

- $\frac{\partial}{\partial w_j} C(w) = \frac{1}{M} \sum_{i=1}^{M} (h_w(x^{(i)}) - y^{(i)}) x_j^{(i)}$

Multiclass Classification using Logistic Regression

- Logistic regression can be used for binary classification.
- Multi-class class classification can be realized using one vs all (or rest) binary classification.
- Train a logistic regression classifier $h_w^{(i)}(x)$ for each class i to predict the prob. $y = i$.
- During testing, for given x, pick the class that maximizes the following: $\max_i h_w^{(i)}(x)$

Thank You