Logistic Regression

The City College of New York
CSc 59929 – Introduction to Machine Learning
Spring 2019 – Erik K. Grimmelmann, Ph.D.

Logistic Regression

- A binary classification model
- Developed in the field of Statistics, not Machine Learning
- Easy to implement
- Very widely used
- Easily extended to multiclass classifications using the OvR technique

Ref.: Wikipedia

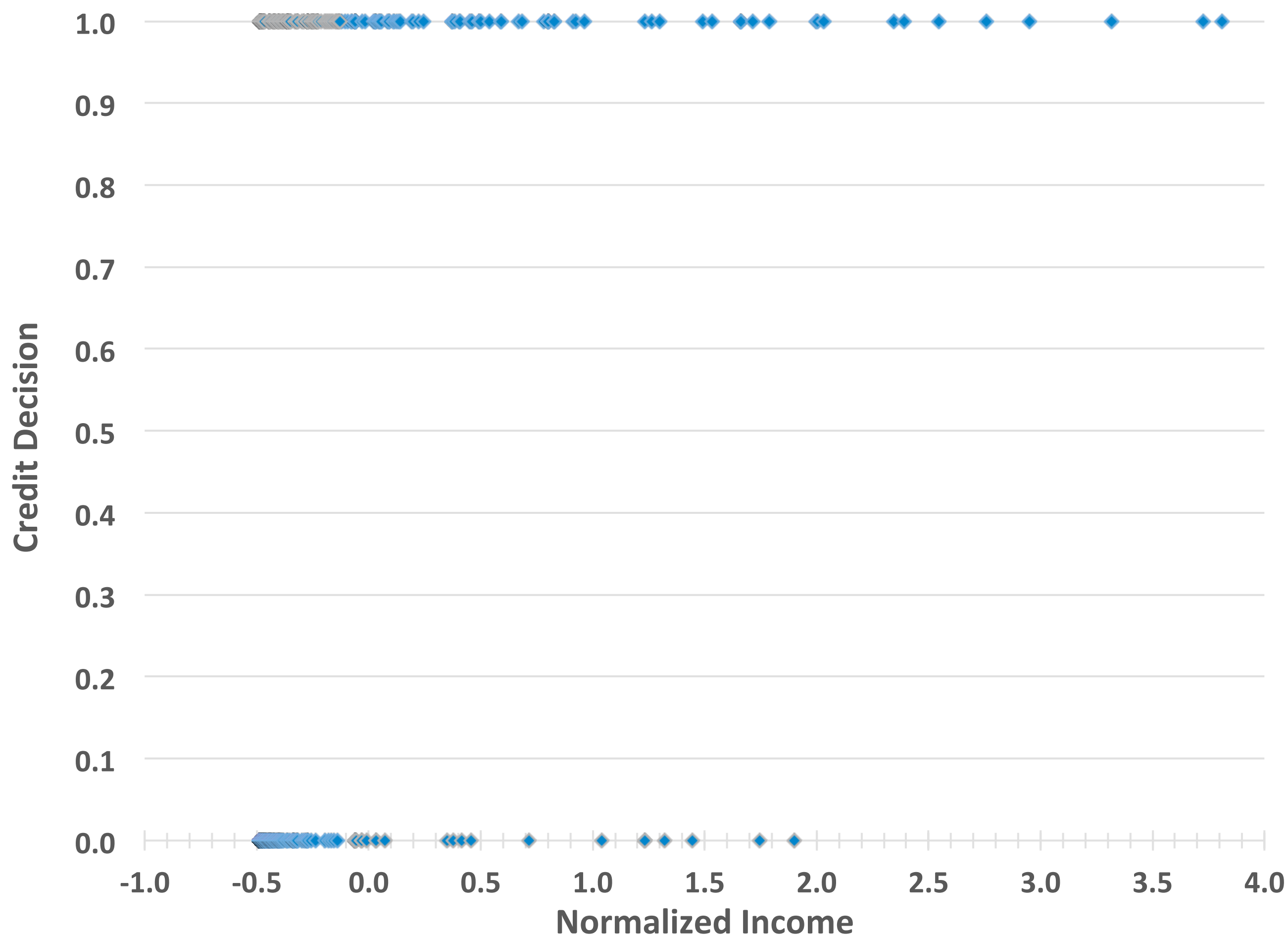
Learning techniques we have seen so far

- Perceptron
  - Divide classes with a hyperplane
  - Converges only if classes are linearly separable
  - Provides no information about the confidence of the classification of a sample
- Adaline
  - Finds the hyperplane that does the best job of dividing the classes
  - Can converge if classes are not linearly separable
  - Provides no information about the confidence of the classification of a sample
- A Better Technique TBD
  - Finds a mechanism that does the best job of dividing the classes
  - Works if classes are not linearly separable
  - Provides information about the confidence of the classification of a sample

Credit decision example

- Based on Real Data (not from the United States)
- Available data
  - Normalized income, x
  - Credit decision y, (0 = denied, 1 = approved)
- Let P$(y=1|\ x))$ be the function whose value is the conditional probability that credit is approved given a normalized income of x.
- Required characteristics of P$(y=1|\ x))$
  - $0 \leqslant$ P$(y=1|\ x)) \leqslant 1$
  - P$(y=0|\ x))+P(y=1|\ x))= 1$
  - P$(y=1|\ x)) \leqslant$ P$(y=1|\ x+\varepsilon)$ for all x and $\varepsilon > 0.$
- Desired characteristic of P$(y=1|\ x))$
  - Defined by as few parameters as possible

Credit decision



Odds ratio

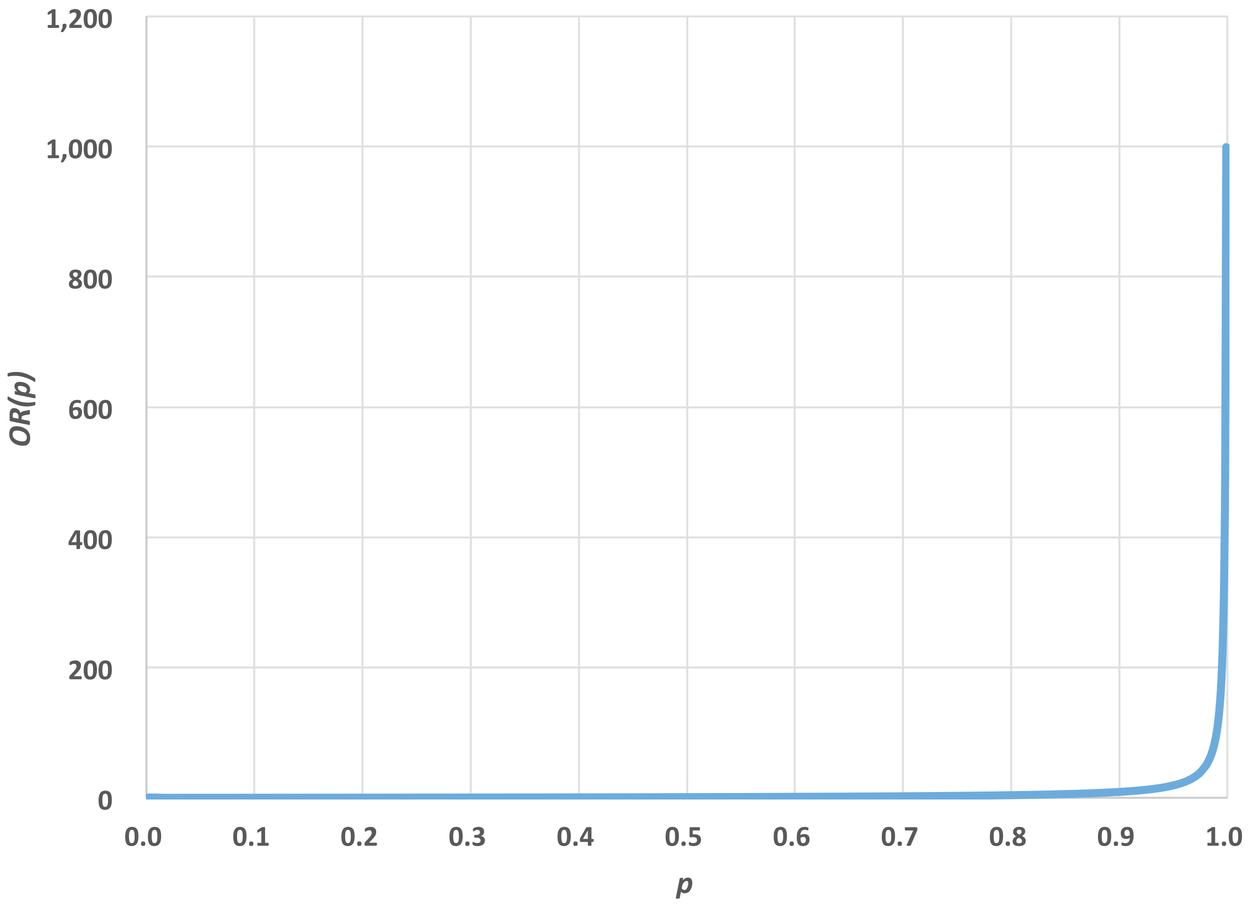
- Let p be the probability of a “positive” event
- We define the **odds ratio** (OR) as the probability of a “positive” events divided by the odds of a “negative” event,

$$OR(p) = \frac{p}{(1-p)}$$

Note that the OR $\in [0, \infty)$

Ref.: Wikipedia

Odds Ratio



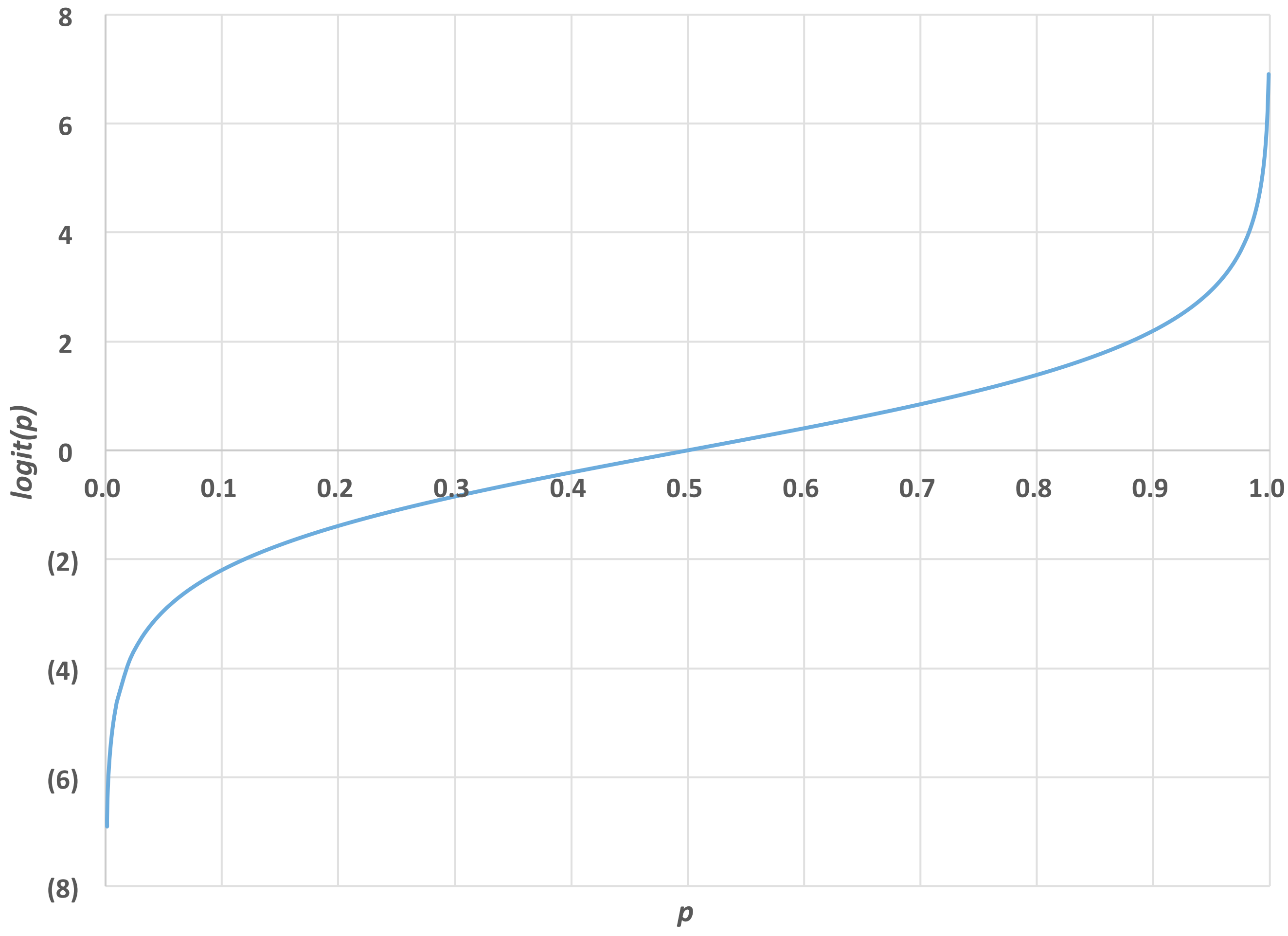
logit function

- Let p be the probability of a "positive" event
- We define the **logit function** as the log of the odds ratio,

$$logit(p) = \ln\left(OR(p)\right) = \ln\left(\frac{p}{(1-p)}\right)$$

Note that $logit(p) \in (-\infty, +\infty)$

logit function



Solve for the inverse of the logit function

$$y = logit(p) = \ln\left(\frac{p}{1-p}\right)$$

$$e^{y} = e^{\ln\left(\frac{p}{1-p}\right)} = \left(\frac{p}{1-p}\right)$$

$$(1-p)e^{y} = p$$

$$e^{y} - pe^{y} = p$$

$$e^{y} = p + pe^{y}$$

$$= p(1+e^{y})$$

Solve for the inverse of the logit function

$$e^{y} = p(1+e^{y})$$

$$p = \left(\frac{e^{y}}{1+e^{y}}\right)$$

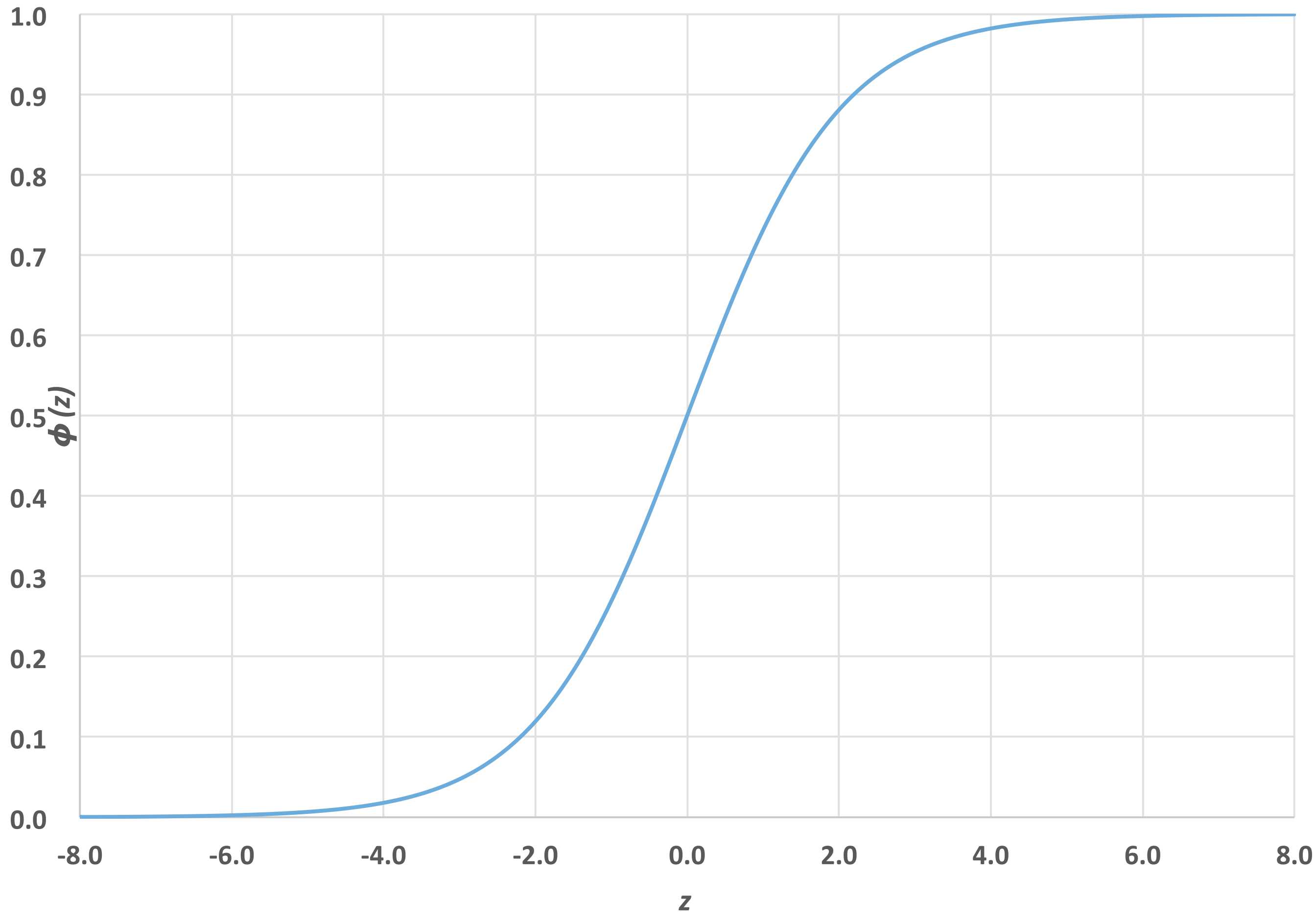
$$= \left(\frac{e^{y}}{1+e^{y}}\right)\left(\frac{e^{-y}}{e^{-y}}\right)$$

$$= \left(\frac{1}{1+e^{-y}}\right)$$

This is called the logistic function and sometimes also the sigmoid function

$$\phi(z) = \left(\frac{1}{1+e^{-z}} \right)$$

logistic function



Limits of the logistic function

$$\phi(z) = \left(\frac{1}{1+e^{-z}}\right)$$

$$\begin{aligned}
\lim_{z \to -\infty} \phi(z) &= \left(\frac{1}{1+e^{--\infty}}\right) \\
&= \left(\frac{1}{1+e^{\infty}}\right) \\
&= \left(\frac{1}{1+\infty}\right) \\
&= \frac{1}{\infty} \\
&= 0
\end{aligned}$$

Limits of the logistic function

$$\phi(z) = \left(\frac{1}{1+e^{-z}} \right)$$

$$\begin{aligned}
\lim_{z \to +\infty} \phi(z) &= \left(\frac{1}{1+e^{-\infty}} \right) \\
&= \left(\frac{1}{1+0} \right) \\
&= \left(\frac{1}{1} \right) \\
&= 1
\end{aligned}$$

Attributes of the logistic function

$$\phi(z) = \left(\frac{1}{1+e^{-z}}\right)$$

$$\begin{aligned}\phi(0) &= \left(\frac{1}{1+e^{-0}}\right)\\ &= \left(\frac{1}{1+1}\right)\\ &= \frac{1}{2}\end{aligned}$$

Symmetry of the logistic function

$$\phi(-z)+\phi(z)=\left(\frac{1}{1+e^{--z}}\right)+\left(\frac{1}{1+e^{-z}}\right)$$

$$=\left(\frac{1}{1+e^{z}}\right)+\left(\frac{1}{1+e^{-z}}\right)$$

$$=\frac{\left(1+e^{-z}\right)+\left(1+e^{z}\right)}{\left(1+e^{z}\right)\left(1+e^{-z}\right)}$$

$$=\frac{\left(2+e^{-z}+e^{z}\right)}{\left(1+e^{z}+e^{-z}+e^{z}e^{-z}\right)}$$

$$=\frac{\left(2+e^{-z}+e^{z}\right)}{\left(1+e^{z}+e^{-z}+e^{0}\right)}$$

Symmetry of the logistic function

$$\phi(-z) + \phi(z) = \frac{\left(2 + e^{-z} + e^{z}\right)}{\left(1 + e^{z} + e^{-z} + e^{0}\right)}$$

$$= \frac{\left(2 + e^{-z} + e^{z}\right)}{\left(1 + e^{z} + e^{-z} + 1\right)}$$

$$= \frac{\left(2 + e^{-z} + e^{z}\right)}{\left(2 + e^{z} + e^{-z}\right)}$$

$$= 1$$

Symmetry of the logistic function

$$\phi(-z) + \phi(z) = 1$$

$$\phi(+z) = 1 - \phi(-z)$$

$$\phi(-z) = 1 - \phi(+z)$$

$$\left(\phi(-z) - \frac{1}{2}\right) = \left(\frac{1}{2} - \phi(+z)\right)$$

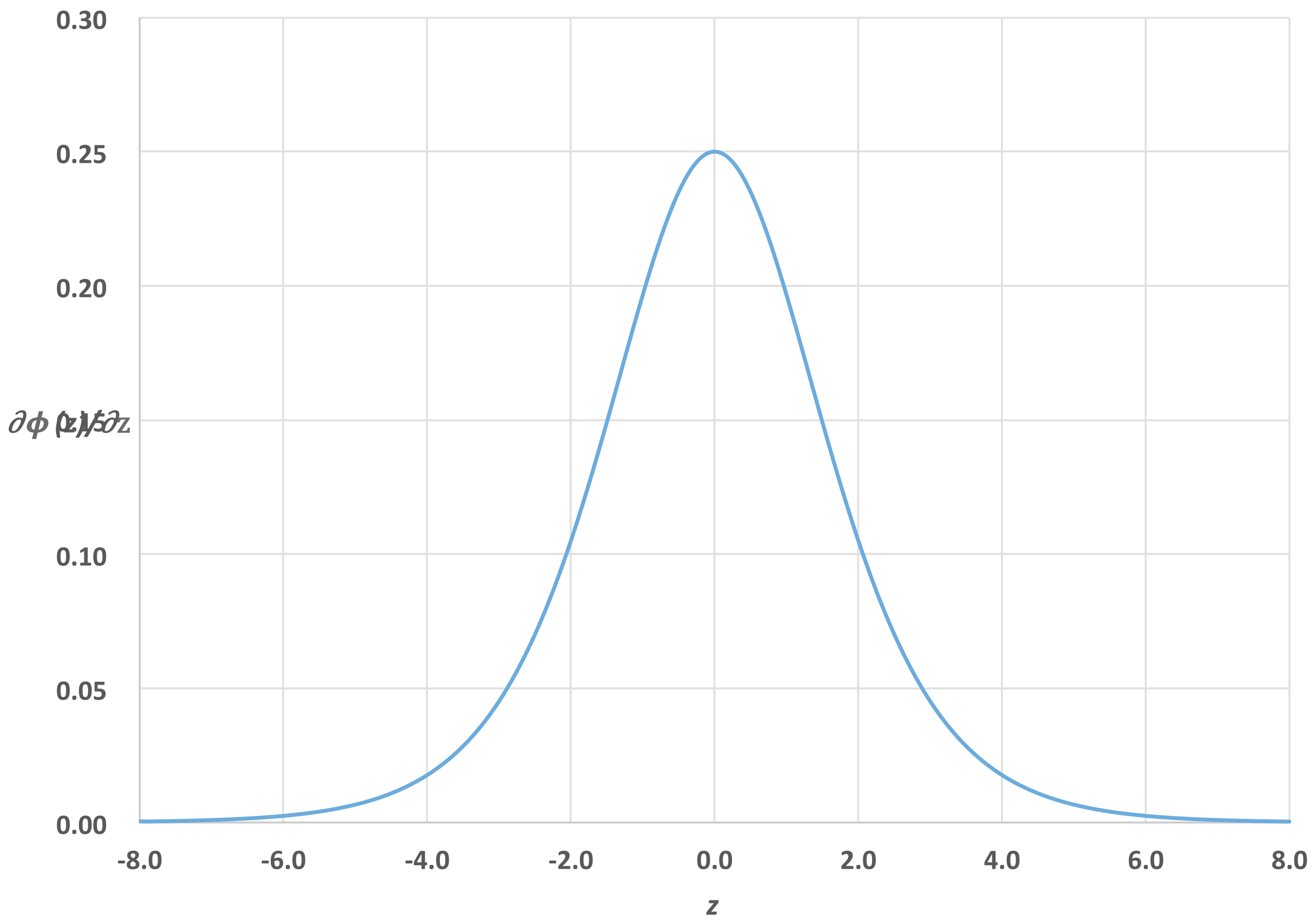
Derivative of the logistic function

$$
\begin{aligned}
\frac{\partial}{\partial z}\phi(z) &= \frac{\partial}{\partial z}\left(\frac{1}{1+e^{-z}}\right) \\
&= \frac{1}{\left(1+e^{-z}\right)^2}\frac{\partial}{\partial z}\left(1+e^{-z}\right) \\
&= \frac{1}{\left(1+e^{-z}\right)^2}e^{-z} \\
&= \frac{1}{\left(1+e^{-z}\right)^2}\frac{\left(1+e^{-z}\right)}{\left(1+e^{-z}\right)}e^{-z}\,\frac{e^{z}}{e^{z}} \\
&= \frac{1}{\left(1+e^{-z}\right)}\frac{1}{\left(1+e^{-z}\right)}\frac{1}{e^{z}}
\end{aligned}
$$

Derivative of the logistic function

$$
\begin{aligned}
\frac{\partial}{\partial z}\phi(z) &= \frac{1}{\left(1+e^{-z}\right)}\frac{1}{\left(1+e^{-z}\right)}\frac{1}{e^{z}} \\
&= \frac{1}{\left(1+e^{-z}\right)}\frac{1}{\left(e^{z}+e^{-z}e^{z}\right)} \\
&= \frac{1}{\left(1+e^{-z}\right)}\frac{1}{\left(e^{z}+1\right)} \\
&= \frac{1}{\left(1+e^{-z}\right)}\frac{1}{\left(1+e^{z}\right)} \\
&= \phi(z)\phi(-z) \\
&= \phi(z)\left(1-\phi(z)\right)
\end{aligned}
$$

Derivative of the logistic function



Derivative of the logistic function

$$\frac{\partial}{\partial z}\phi(z) = \phi(z)\left(1-\phi(z)\right)$$

$$\begin{aligned}\frac{\partial}{\partial z}\phi(0) &= \phi(0)\left(1-\phi(0)\right)\\ &= \left(\frac{1}{2}\right)\left(\frac{1}{2}\right)\\ &= \frac{1}{4}\end{aligned}$$

Limits of the derivative of the logistic function

$$\frac{\partial}{\partial z}\phi(z) = \phi(z)\phi(-z)$$

$$\begin{aligned}
\lim_{z\to+\infty} \frac{\partial}{\partial z}\phi(z) &= \left(\lim_{z\to+\infty} \frac{\partial}{\partial z}\phi(z)\right)\left(\lim_{z\to+\infty} \frac{\partial}{\partial z}\phi(-z)\right) \\
&= \left(\lim_{z\to+\infty} \frac{\partial}{\partial z}\phi(z)\right)\left(\lim_{z\to-\infty} \frac{\partial}{\partial z}\phi(z)\right) \\
&= (1)(0) \\
&= 0
\end{aligned}$$

$$\begin{aligned}
\lim_{z\to-\infty} \frac{\partial}{\partial z}\phi(z) &= \left(\lim_{z\to-\infty} \frac{\partial}{\partial z}\phi(z)\right)\left(\lim_{z\to-\infty} \frac{\partial}{\partial z}\phi(-z)\right) \\
&= \left(\lim_{z\to-\infty} \frac{\partial}{\partial z}\phi(z)\right)\left(\lim_{z\to+\infty} \frac{\partial}{\partial z}\phi(z)\right) \\
&= (0)(1) \\
&= 0
\end{aligned}$$

Back to the credit decision example

- Required characteristics of $P(y=1|\, x))$
  - $0 \leqslant \mathrm{P}(y=1|\, x)) \leqslant 1$
  - $\mathrm{P}(y=0|\, x))+\mathrm{P}(y=1|\, x))= 1$
  - $\mathrm{P}(y=1|\, x)) \leqslant \mathrm{P}(y=1|\, x+\varepsilon)$ for all x and $\varepsilon > 0.$
- Desired characteristic of $\mathrm{P}(y=1|\, x))$
  - Defined by as few parameters as possible

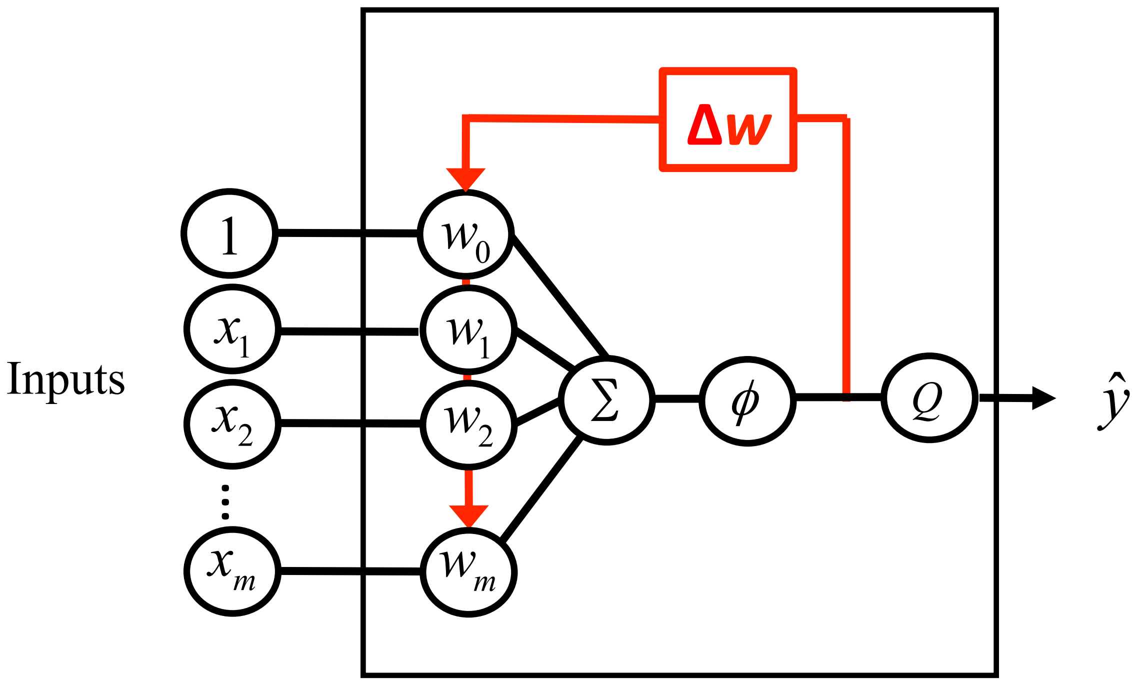
- Let $\operatorname{logit}\left(\mathrm{P}(y=1\,|\,\boldsymbol{x})\right) = w_0 x_0 + w_1 x_1 + \ldots\ + w_m x_m$

$$= \sum_{i=0}^{m} w_i x_i = \boldsymbol{w}^T \boldsymbol{x}$$

$$= z$$

- Then $\mathrm{P}(y=1\,|\,\boldsymbol{x}) = \phi(z)$

Training an Adaline / Logistic Regression



- $\boldsymbol{x}$ input
- $\boldsymbol{w}$ weight
- $\sum$ $z = \boldsymbol{w}^T\boldsymbol{x}$
- ϕ activation function
- Q quantizer
- $\hat{y}$ computed output value

$\boxed{\Delta w}$ = adjustments to w

Activation functions

Adaline $\phi(z) = z$

Logistic Regression $\phi(z) = \dfrac{1}{1 + e^{-z}}$

Cost functions

Adaline $\quad J(\boldsymbol{w}) = \frac{1}{2}\sum_i \left(y^{(i)} - \phi\left(z^{(i)}\right)\right)^2$

Logistic Regression $\quad J(\boldsymbol{w}) = ?$

Likelihood function

$$
\begin{aligned}
\mathrm{L}(\boldsymbol{w}) &= \mathrm{P}(\boldsymbol{y} \mid \boldsymbol{x} : \boldsymbol{w}) \\
&= \prod_{i=1}^{n} \mathrm{P}(\boldsymbol{y}^{(i)} \mid \boldsymbol{x}^{(i)} ; \boldsymbol{w}) \\
&= \prod_{i=1}^{n} \left(\phi\left(z^{(i)}\right)\right)^{y^{(i)}} \left(1-\phi\left(z^{(i)}\right)\right)^{1-y^{(i)}}
\end{aligned}
$$

Log-likelihood and cost functions

$$
\begin{aligned}
l(\boldsymbol{w}) &= \ln\left(L\left(\boldsymbol{w}\right)\right) \\
&= \sum_{i=1}^{n}\left[y^{(i)}\ln\left(\phi\left(z^{(i)}\right)\right)+\left(1-y^{(i)}\right)\ln\left(1-\phi\left(z^{(i)}\right)\right)\right]
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{J}(\boldsymbol{w}) &= -\ l(\boldsymbol{w}) \\
&= -\sum_{i=1}^{n}\left[y^{(i)}\ln\left(\phi\left(z^{(i)}\right)\right)+\left(1-y^{(i)}\right)\ln\left(1-\phi\left(z^{(i)}\right)\right)\right]
\end{aligned}
$$

Cost function for a single-sample instance

$$J(\boldsymbol{w}) = -\ l(\boldsymbol{w})$$

$$= -\sum_{i=1}^{n}\left[y^{(i)}\ln\left(\phi\left(z^{(i)}\right)\right)+\left(1-y^{(i)}\right)\ln\left(1-\phi\left(z^{(i)}\right)\right)\right]$$

$$J(\phi(z),y;\boldsymbol{w}) = -y\ln\left(\phi\left(z\right)\right)-\left(1-y\right)\ln\left(1-\phi\left(z\right)\right)$$

Cost function for a single-sample instance

$$J(\phi(z),y;\boldsymbol{w}) = -y\ln\left(\phi\left(z\right)\right)-\left(1-y\right)\ln\left(1-\phi\left(z\right)\right)$$

$$= \begin{cases} -\ln\left(\phi\left(z\right)\right) \text{ for y} =1 \\ -\ln\left(1-\phi\left(z\right)\right)= -\ln\left(\phi\left(-z\right)\right) \text{ for y} = 0 \end{cases}$$

Derivative of the cost function for a single-sample instance

$$J(\phi(z),y;\boldsymbol{w}) = -y\ln\left(\phi\left(z\right)\right)-\left(1-y\right)\ln\left(1-\phi\left(z\right)\right)$$

$$\frac{\partial J(\phi(z),y;\boldsymbol{w})}{\partial\phi(z)} = -y\frac{\partial\ln\left(\phi\left(z\right)\right)}{\partial\phi(z)}-\left(1-y\right)\frac{\partial\ln\left(1-\phi\left(z\right)\right)}{\partial\phi(z)}$$

$$= -\frac{y}{\phi(z)}-\left(1-y\right)\frac{\partial\ln\left(\phi\left(-z\right)\right)}{\partial\phi(z)}$$

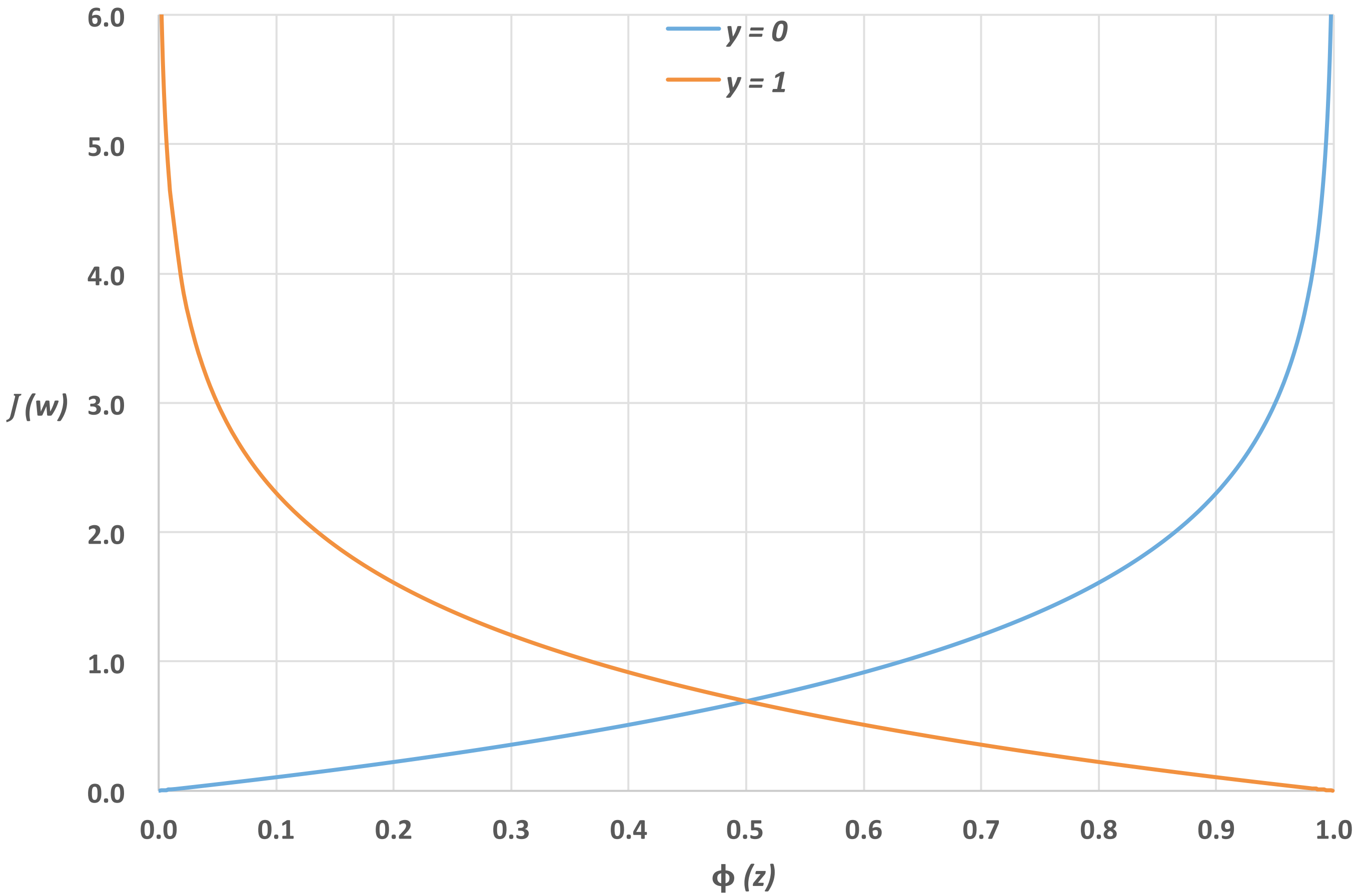
$$= -\frac{y}{\phi(z)}+\frac{\left(1-y\right)}{\phi(-z)}$$

Derivative of the cost function for a single-sample instance

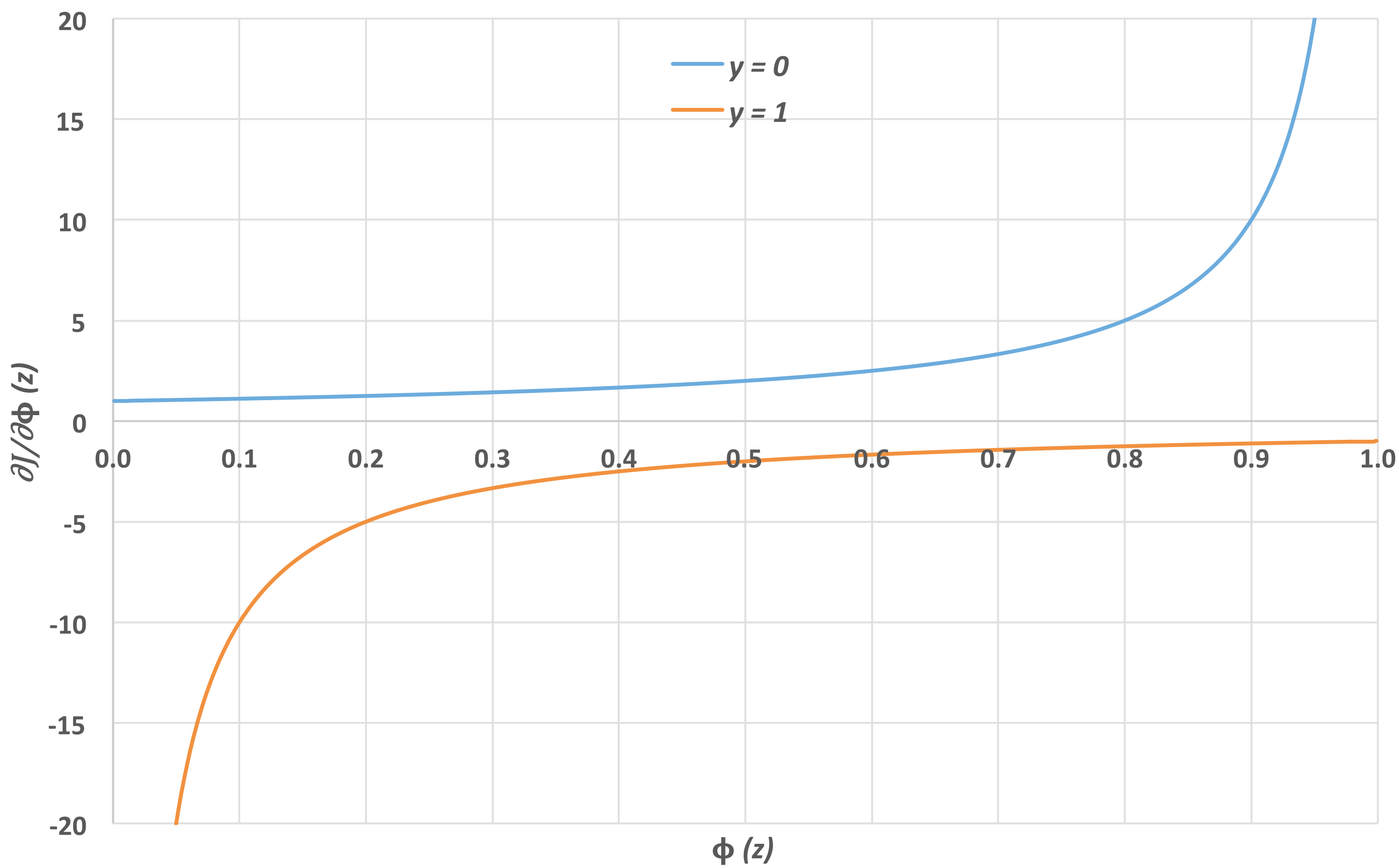
$$\frac{\partial \mathrm{J}(\phi(\mathrm{z}),\mathrm{y};\boldsymbol{w})}{\partial \phi(\mathrm{z})} = -\frac{y}{\phi(\mathrm{z})} + \frac{(1-y)}{\phi(-\mathrm{z})}$$

$$= \begin{cases} -\dfrac{1}{\phi(\mathrm{z})} & \text{for y} = 1 \\ +\dfrac{1}{\phi(-\mathrm{z})} & \text{for y} = 0 \end{cases}$$

Cost function for a single-sample instance



Derivative of cost function for a single-sample instance



Gradient in logistic regression

$$\mathrm{J}(\boldsymbol{w}) = -\left(y\ln\left(\phi\left(z\right)\right)+\left(1-y\right)\ln\left(1-\phi\left(z\right)\right)\right)$$

$$\frac{\partial}{\partial w_j}J(\boldsymbol{w}) = -\left(\left(y\frac{1}{\phi\left(z\right)}\right)\frac{\partial}{\partial w_j}\left(\phi\left(z\right)\right)-\left(\left(1-y\right)\frac{1}{\left(1-\phi\left(z\right)\right)}\right)\frac{\partial}{\partial w_j}\left(\phi\left(z\right)\right)\right)$$

$$= -\left(\left(y\frac{1}{\phi\left(z\right)}\right)-\left(\left(1-y\right)\frac{1}{\left(1-\phi\left(z\right)\right)}\right)\right)\frac{\partial}{\partial w_j}\left(\phi\left(z\right)\right)$$

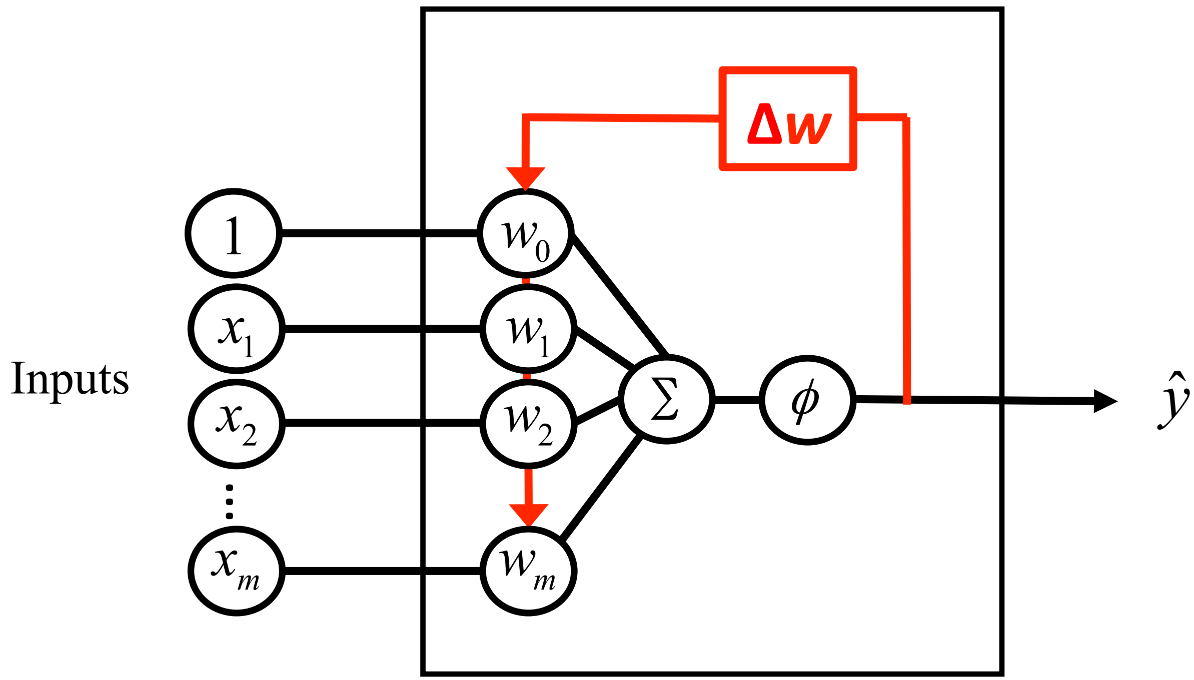
$$\text{since } \frac{\partial}{\partial x}\ln\left(f\left(x\right)\right) = \frac{1}{f\left(x\right)}\frac{\partial}{\partial x}f\left(x\right)$$

Gradient in logistic regression

Using our previous result that $\frac{\partial}{\partial z}\phi(z) = \phi(z)\left(1-\phi(z)\right)$,

$$
\begin{aligned}
\frac{\partial}{\partial w_j} J(\boldsymbol{w}) &= -\left(y\frac{1}{\phi(z)} + (1-y)\frac{1}{\left(1-\phi(z)\right)} \right) \frac{\partial}{\partial w_j}\left(\phi(z)\right) \\
&= -\left(\left(y\frac{1}{\phi(z)} \right) + \left((1-y)\frac{-1}{\left(1-\phi(z)\right)} \right) \right) \frac{\partial}{\partial w_j}\left(\phi(z)\right) \\
&= -\left(\left(y\frac{1}{\phi(z)} \right) + \left((y-1)\frac{1}{\left(1-\phi(z)\right)} \right) \right) \phi(z)\left(1-\phi(z)\right) \frac{\partial}{\partial w_j} z \\
&= -\left(\left(y\left(1-\phi(z)\right) \right) + \left((y-1)\phi(z) \right) \right) \frac{\partial}{\partial w_j} z \\
&= -\left(y - y\phi(z) + y\phi(z) - \phi(z) \right) x_j \\
&= -\left(y - \phi(z) \right) x_j
\end{aligned}
$$

Training a logistic regression model (or a Adaline model or a Perceptron)



- $\boldsymbol{x}$ input
- $\boldsymbol{w}$ weight
- $\sum$ $z = \boldsymbol{w}^T\boldsymbol{x}$
- ϕ activation function
- $\hat{y}$ computed output value

$\boxed{\Delta w}$ = adjustments to w

Training an Logistic regression model (or an Adaline model)

To train a Logistic Regression model we update the weights to minimize the cost function, *J(w),* by moving in direction of the negative of the gradient of *J(w)* by an amount proportional to the magnitude of the gradient.

$$\boldsymbol{w} := \boldsymbol{w} + \Delta\boldsymbol{w} \quad \text{where}$$

$$\Delta\boldsymbol{w} = -\eta\nabla J\left(\boldsymbol{w}\right)$$

and where η is the learning rate such that $0 < \eta < 1$.

Training a logistic regression model

$$\boldsymbol{w} := \boldsymbol{w} + \Delta\boldsymbol{w} \quad \text{where}$$

$$\Delta\boldsymbol{w} = -\eta \nabla J\left(\boldsymbol{w}\right)$$

$$\Delta w_j = -\eta \frac{\partial J\left(\boldsymbol{w}\right)}{\partial w_j}$$

$$= \eta \sum_i \left(y^{(i)} - \phi\left(z^{(i)}\right)\right) x_j^{(i)}$$

and where $z^{(i)} = \boldsymbol{w}^T \boldsymbol{x}^{(i)}$

Logistic regression training algorithm

1. Initialize the weights, $\boldsymbol{w}$, to 0 or small random numbers
2. Compute the gradient of the cost function, $\nabla J\left(\boldsymbol{w}\right)$, by summing over all (or a fixed number of randomly selected) training samples, $\left(\boldsymbol{x}^{(i)}, y^{(i)}\right)$,

$$\frac{\partial J\left(\boldsymbol{w}\right)}{\partial w_j} = -\sum_i \left(y^{(i)} - \phi\left(z^{(i)}\right)\right) x_j^{(i)}$$

Logistic regression training algorithm

3. Update the weights,

$$w_j := w_j + \Delta w_j, \text{ where}$$

$$\Delta w_j = -\eta \frac{\partial J(\boldsymbol{w})}{\partial w_j}$$

$$= \eta \sum_i \left(y^{(i)} - \phi\left(z^{(i)}\right) \right) x_j^{(i)}$$

and where η is the learning rate such that $0 < \eta < 1$.

Logistic regression training algorithm

4. Repeat steps 2. and 3. until the weights converge,

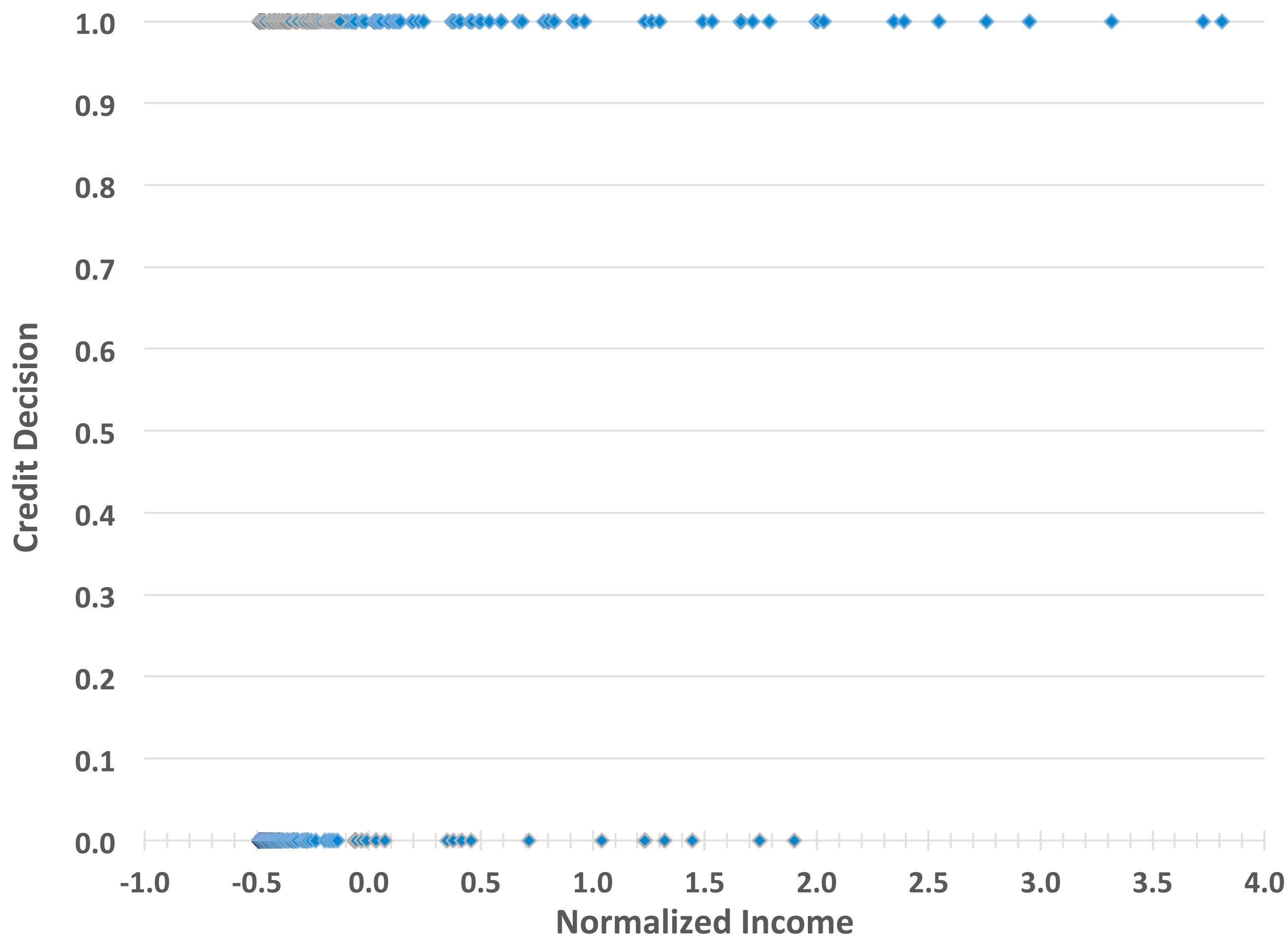
   that is, until

$$\left\|\Delta \boldsymbol{w}\right\| < \varepsilon \ \left(or \left\|\Delta \boldsymbol{w}\right\|_1 < \varepsilon\right), \text{where}$$

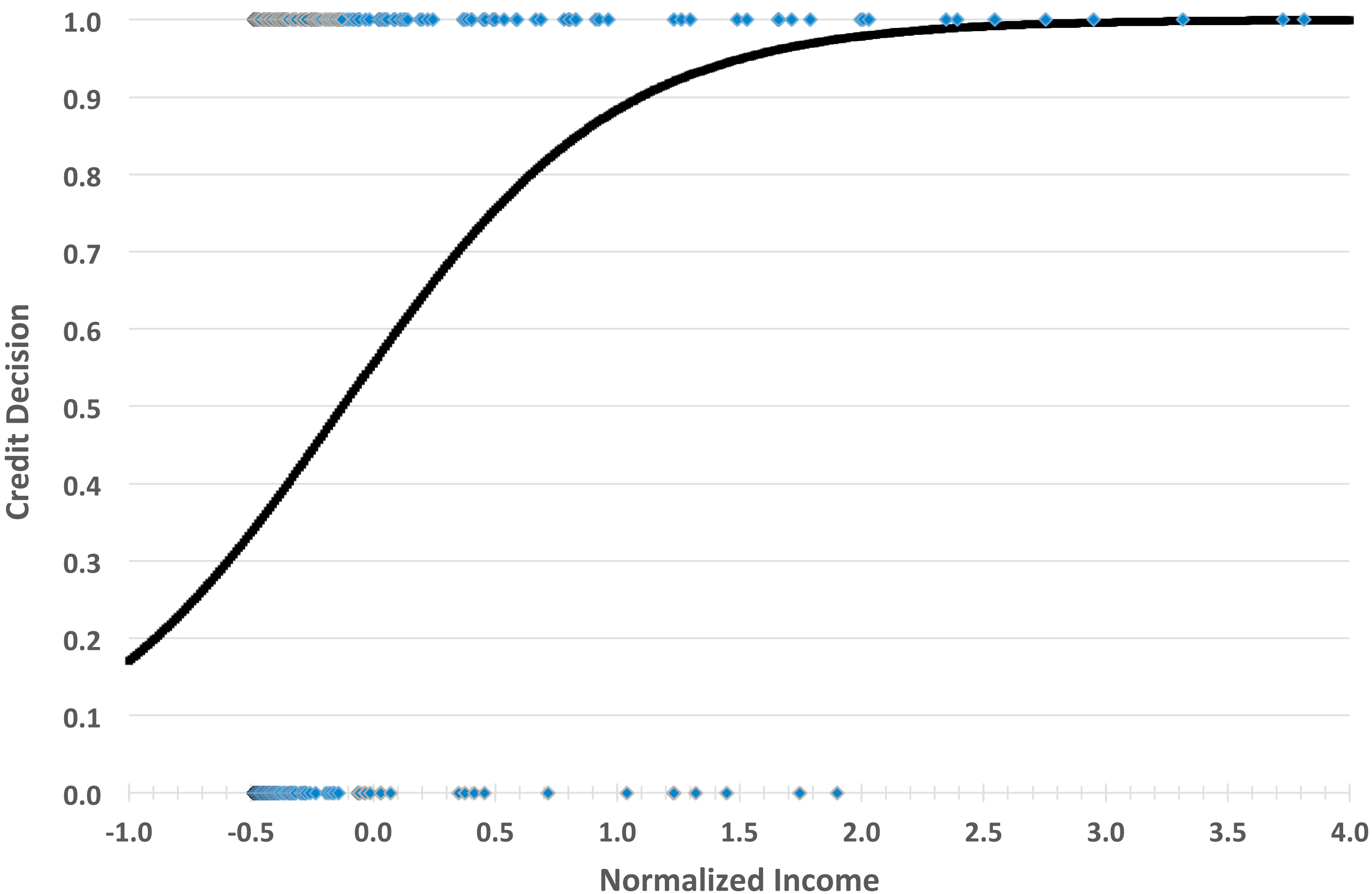
ε is the convergence threshold, $\varepsilon > 0$.

or for a set number of iterations.

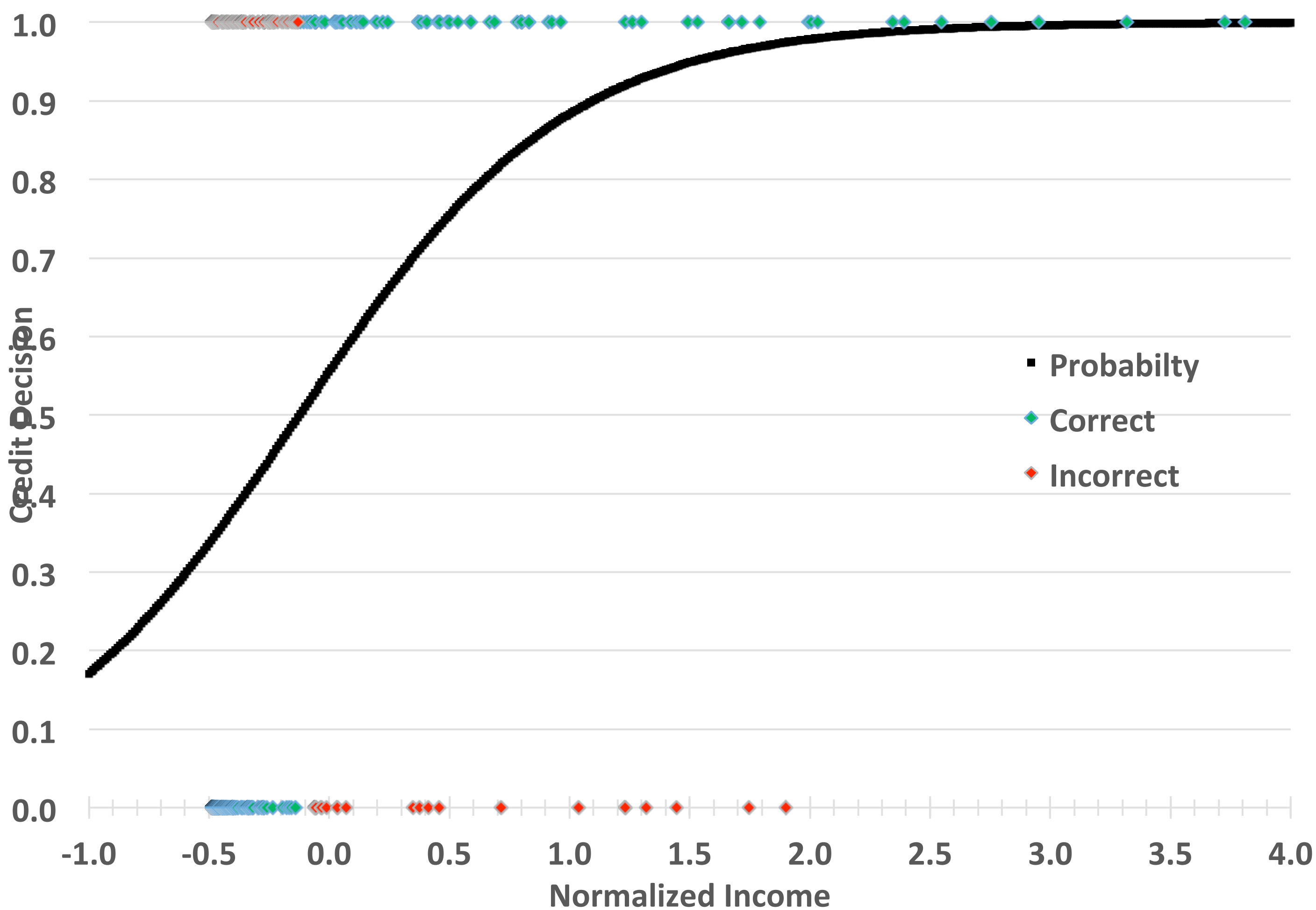
Credit decision



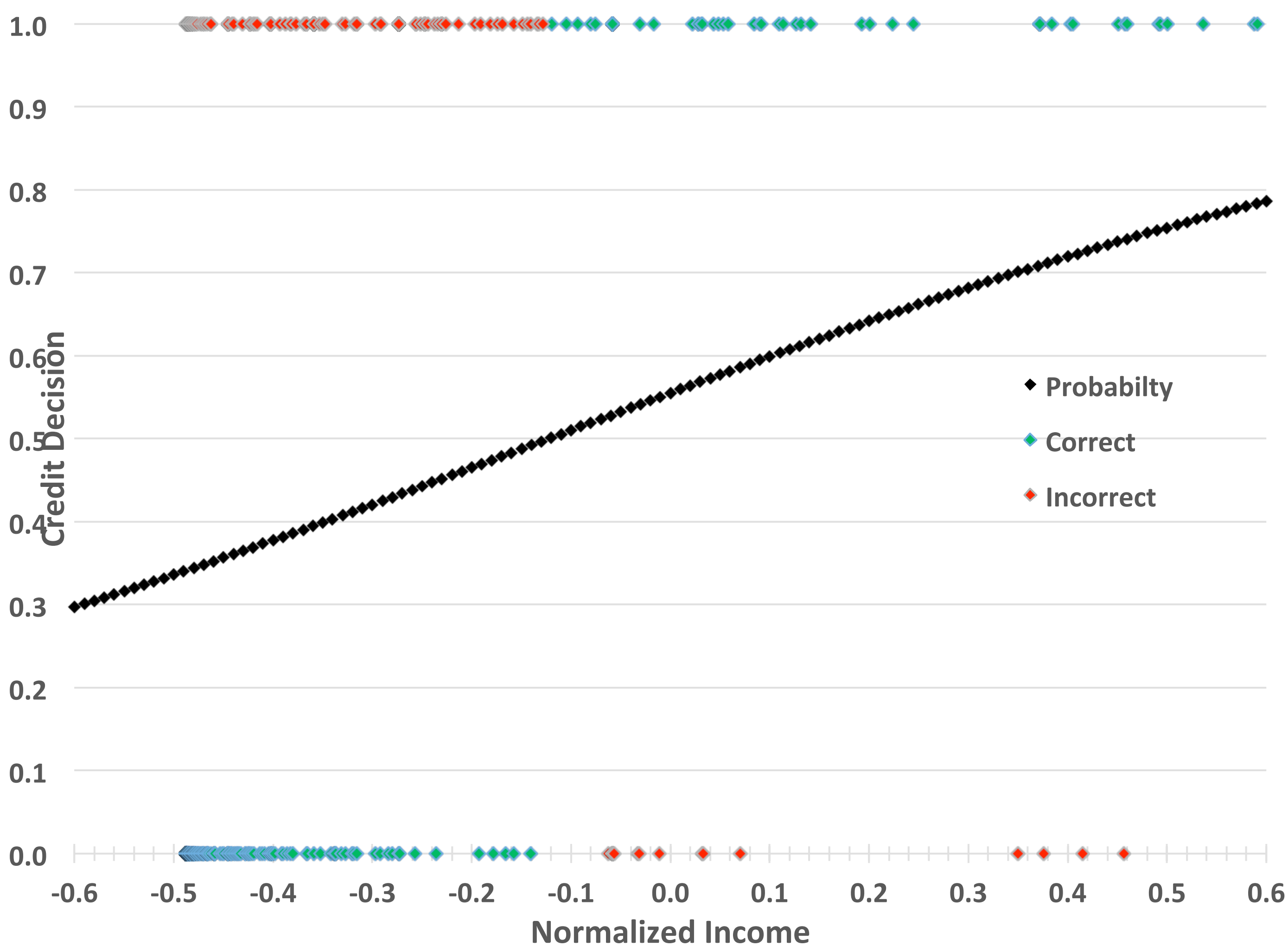
Credit decision using logistic regression



Credit decision using logistic regression



Credit decision using logistic regression



Regularization

Regularization is the process of introducing additional information in order to solve an ill-posed problem of to prevent **overfitting**.

Regularization can enhance the prediction accuracy and/or interpretability of the model it produces.

Regularization often encourages (or requires) models to be simpler rather than more complex.

Logistic regression cost functions

Without Regularization

$$J(\boldsymbol{w}) = -\sum_{i=1}^{n}\left[y^{(i)}\ln\left(\phi\left(z^{(i)}\right)\right)+\left(1-y^{(i)}\right)\ln\left(1-\phi\left(z^{(i)}\right)\right)\right]$$

With L2 Regularization

$$J_{\lambda}(\boldsymbol{w}) = -\sum_{i=1}^{n}\left[y^{(i)}\ln\left(\phi\left(z^{(i)}\right)\right)+\left(1-y^{(i)}\right)\ln\left(1-\phi\left(z^{(i)}\right)\right)\right]+\frac{\lambda}{2}\left\|\boldsymbol{w}\right\|^{2}$$

$$J_{L2}(\boldsymbol{w}) = -C\sum_{i=1}^{n}\left[y^{(i)}\ln\left(\phi\left(z^{(i)}\right)\right)+\left(1-y^{(i)}\right)\ln\left(1-\phi\left(z^{(i)}\right)\right)\right]+\frac{1}{2}\boldsymbol{w}^{T}\boldsymbol{w}$$

where $C = \dfrac{1}{\lambda}$

With L1 Regularization

$$J_{L1}(\boldsymbol{w}) = -C\sum_{i=1}^{n}\left[y^{(i)}\ln\left(\phi\left(z^{(i)}\right)\right)+\left(1-y^{(i)}\right)\ln\left(1-\phi\left(z^{(i)}\right)\right)\right]+\left\|\boldsymbol{w}\right\|_{1}$$

Gradient of cost functions

Without Regularization

$$\frac{\partial}{\partial w_j} J(\boldsymbol{w}) = -\left(y - \phi\left(z\right)\right) x_j$$

With L2 Regularization

$$\frac{\partial}{\partial w_j} J_{L2}(\boldsymbol{w}) = -C\left(y - \phi\left(z\right)\right) x_j + \left|w_j\right|$$

With L1 Regularization

$$\frac{\partial}{\partial w_j} J_{L1}(\boldsymbol{w}) = -C\left(y - \phi\left(z\right)\right) x_j + 1$$

Alternate cost functions

Textbook

$$J(\boldsymbol{w}) = -\sum_{i=1}^{n}\left[y^{(i)}\ln\left(\phi\left(z^{(i)}\right)\right)+\left(1-y^{(i)}\right)\ln\left(1-\phi\left(z^{(i)}\right)\right)\right]$$

$$= -\sum_{i=1}^{n}\left[y^{(i)}\ln\left(\frac{1}{1+\exp\left(-z^{(i)}\right)}\right)+\left(1-y^{(i)}\right)\ln\left(\frac{1}{1+\exp\left(+z^{(i)}\right)}\right)\right]$$

Scikit - learn Documentation

$$J(\boldsymbol{w}) = \sum_{i=1}^{n}\left[\ln\left(\exp\left(-y^{(i)}z^{(i)}\right)+1\right)\right]$$

These alternatives are equivalent but not identical.

Scikit-learn Logistic Regression Class

class sklearn.linear_model.LogisticRegression(…)

Parameter	Default		Parameter	Defaults
penalty	’l2’		random_state	None
dual	False		solver	’liblinear’
tol	0.0001		max_iter	100
C	1.0		multi_class	‘ovr’
fit_intercept	True		verbose	0
intercept_scaling	1		warm_start	False
class_weight	None		n_jobs	1

Scikit-learn Logistic Regression Attributes

Attribute	Description
coef_	Coefficient of the features in the decision function.
intercept_	Intercept (a.k.a. bias) added to the decision function
n_iter_	Actual number of iterations for all classes.

Scikit-learn Logistic Regression Methods

Method	Description
decision_function(X)	Predict confidence scores for samples.
densify()	Convert coefficient matrix to dense array format.
fit(X, y[, sample_weight])	Fit the model according to the given training data.
get_params([deep])	Get parameters for this estimator.
predict(X)	Predict class labels for samples in X.
predict_log_proba(X)	Log of probability estimates.
predict_proba(X)	Probability estimates.
score(X, y[, sample_weight])	Returns the mean accuracy on the given test data and labels.
set_params(**params)	Set the parameters of this estimator.
sparsify()	Convert coefficient matrix to sparse format.