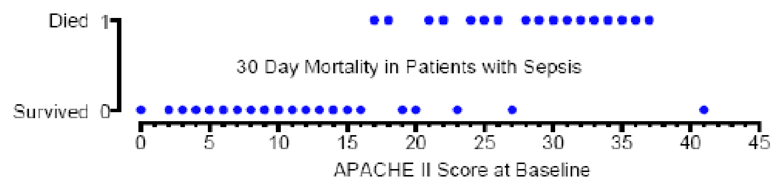
Logistic Regression

Logistic regression in one dimension

a) Example: APACHE II Score and Mortality in Sepsis

The following figure shows 30 day mortality in a sample of septic patients as a function of their baseline APACHE II Score. Patients are coded as 1 or 0 depending on whether they are dead or alive in 30 days, respectively.



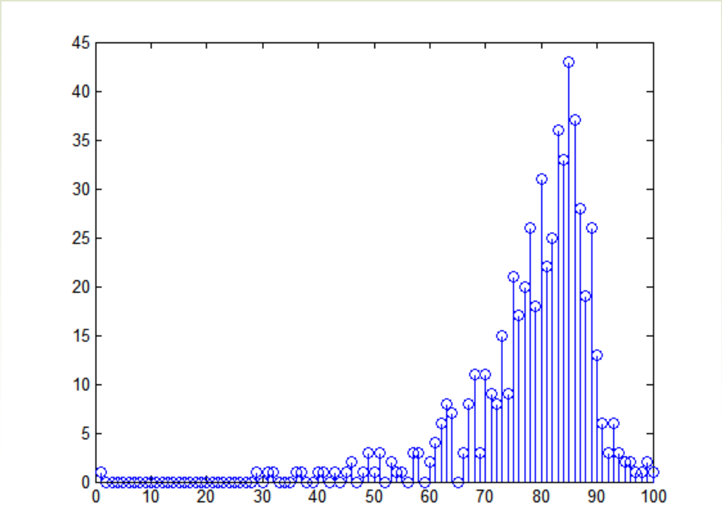
Sigmoid Function

- Bernoulli distribution has 2 possible outcomes
  - PMF: $p(x) = y^x(1-y)^{1-x}$

- Recall for linear regression

$$y(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + \varepsilon \text{ where } \varepsilon \sim \text{Gaussian}$$

- Error Histogram



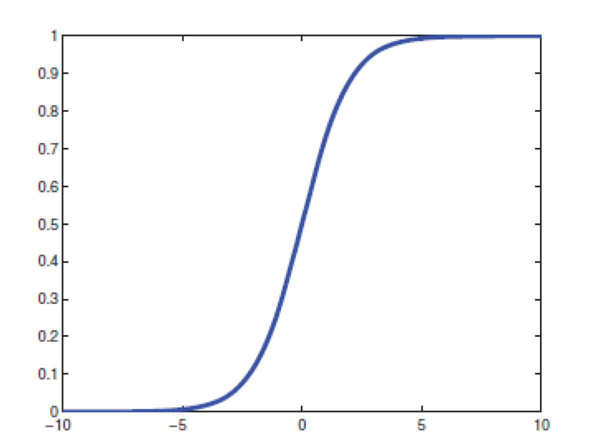
Sigmoid Function

- Bernoulli distribution has 2 possible outcomes
  - PMF: $p(x) = y^x(1-y)^{1-x}$
- If the response is binary, $y \in \{0,1\}$, then $\varepsilon \sim$ Bernoulli

$$p(y|\mathbf{x},\mathbf{w}) = Ber(y|\mathbf{w}^T\mathbf{x})$$

- Pass $\mathbf{w}^T\mathbf{x}$ through function $f(\mathbf{w}^T\mathbf{x})$ such that $0 \le f(\mathbf{w}^T\mathbf{x}) \le 1$
- For logistic regression, choose sigmoid (logistic) function

$$sigm(\mathbf{w}^T\mathbf{x}) = \frac{1}{1+\exp(-\mathbf{w}^T\mathbf{x})}$$



Weight Vector

- How to choose optimal weight vector $\mathbf{w}$?
  - Gradient descent (ascent) method
  - $\mathbf{w} = \text{argmin}_{\mathbf{w}} J(\mathbf{w}) \rightarrow \mathbf{w} = \mathbf{w} - \eta \frac{\partial J(\mathbf{w})}{\partial \mathbf{w}}$ if $J(\mathbf{w})$ is convex
  - $\mathbf{w} = \text{argmax}_{\mathbf{w}} J(\mathbf{w}) \rightarrow \mathbf{w} = \mathbf{w} + \eta \frac{\partial J(\mathbf{w})}{\partial \mathbf{w}}$ if $J(\mathbf{w})$ is not convex

- With 1 training example, formulate the cost function
  - If model predicts correctly, "reward" the model, cost = 0
  - Else "penalize" heavily, cost = ∞
  - Cost = $-[y \log\big(f(x)\big) + (1-y) \log\big(1 - f(x)\big)]$

- With N training examples

$$J = -\sum_{i=1}^{N} y^{(i)} \log\big(f(\mathbf{x}^{(i)})\big) + \big(1 - y^{(i)}\big) \log(1 - f\big(\mathbf{x}^{(i)}\big))$$

- Is J convex ?

Stochastic Gradient Descent

shuffle data set randomly

repeat {

for i = 1, …, N {

$w_j = w_j - \eta \nabla_{w_j} J \ where \ \nabla_{w_j} J = \left[sigm\left(\mathbf{w}^T \mathbf{x}^{(i)} \right) - y^{(i)} \right] x_j^{(i)}$

$(j = 0, 1, \dots, d)$

}

}

- $x_j^{(i)}$: j-th feature of the i-th training example

- Recall the update equation for Linear Regression

$$w_j = w_j - \eta \nabla_{w_j} J \ where \ \nabla_{w_j} J = \left[f\left(\mathbf{x}^{(i)} \right) - y^{(i)} \right] x_j^{(i)}$$

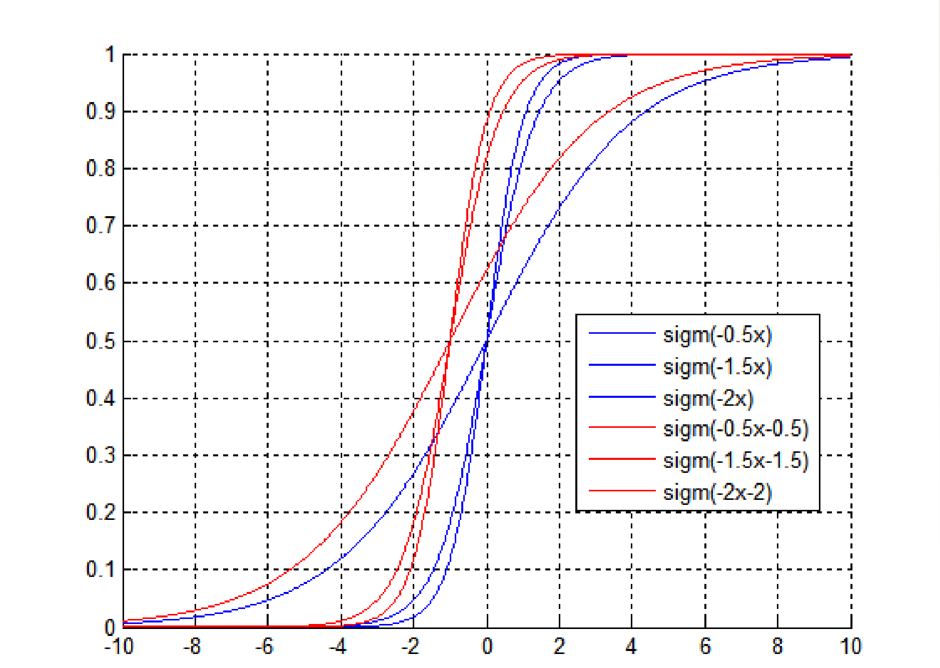
- Once the optimal weight vector is obtained, use the sigmoid function to classify a test data

The bias

- The bias plays a role in shifting the sigmoid curve
  - $s = sigm(z) = \frac{1}{1+\exp(-z)}$

    Where $z = \mathbf{w}^T\mathbf{x} = w_0x_0 + w_1x_1 + w_2x_2 + \cdots$

    Note: $x_0 = 1$



Logistic Regression Classification Example with 2-D Dataset

- Assume $\mathbf{w}^T = [w_0 \quad w_1 \quad w_2] = [0.5 \quad -1.3 \quad 3.2]$
- If $s > 0.5$ → Class 1, else → Class 0
  - $s = sigm(z) = \frac{1}{1+\exp(-z)}$

    $z = \mathbf{w}^T\mathbf{x} = w_0x_0 + w_1x_1 + w_2x_2$

    $x_0 = 1$

x_1	x_2	y	z	s	$\hat{y}$
4.1	1.3	0	-0.67	0.34	0
4.5	1.5	1	-0.55	0.38	0*
1.7	0.4	0	-0.43	0.39	0
0.5	0.7	1	2.09	0.89	1

* Misclassified

Softmax Regression

- Generalization of logistic regression for multi-class classification
- a.k.a. Multinomial Logistic Regression, Maximum Entropy Classifier
- Logistic regression

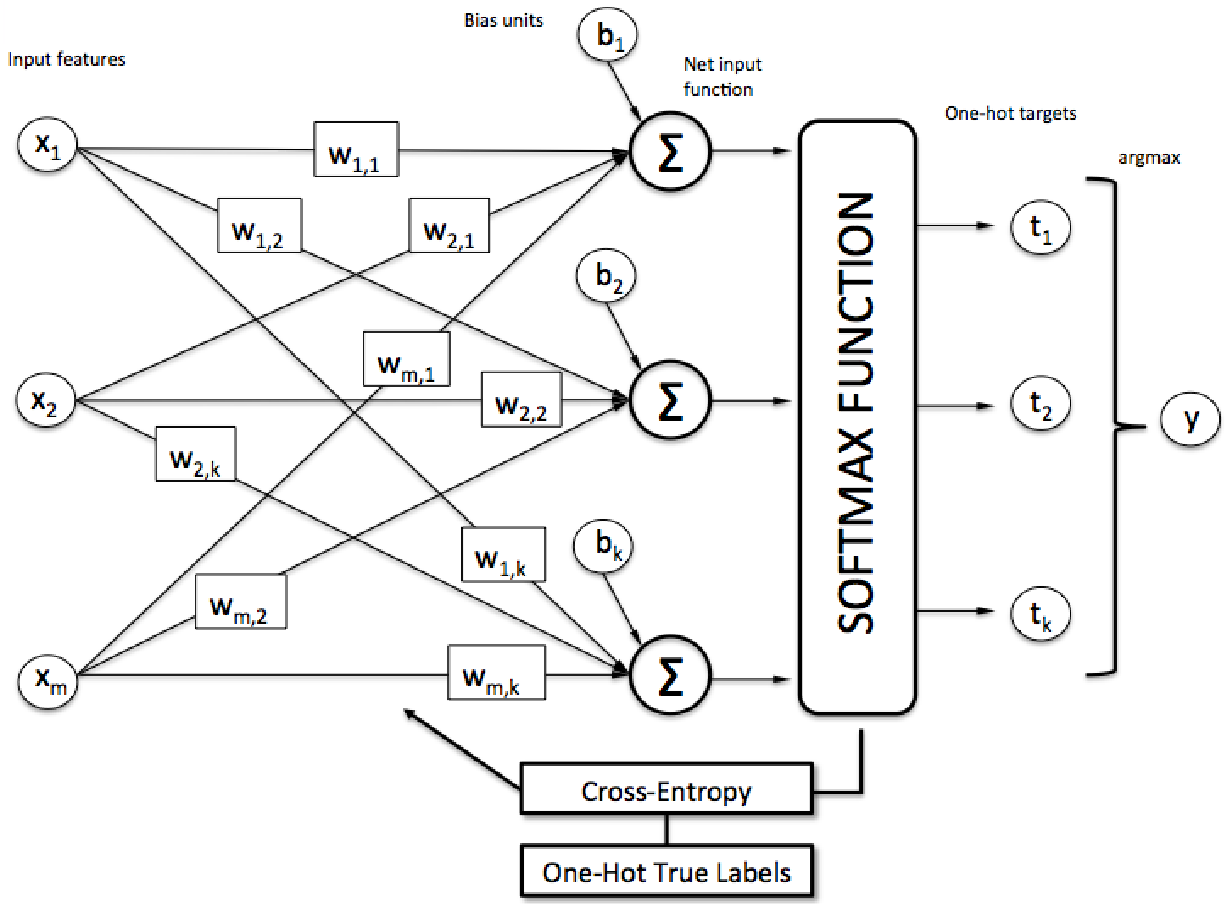
$$s = sigm(z) = \frac{1}{1+\exp(-z)} = \frac{\exp(z)}{1+\exp(z)}$$

Where $z = \mathbf{w}^T\mathbf{x}$

- Softmax regression

$$\phi = softmax(\boldsymbol{z}) = \frac{\exp(z_j)}{\sum_{i=1}^{K}\exp(z_i)} = p(y = j|\mathbf{z})$$

$p(y = j|\mathbf{z})$ → probability of class j



Softmax Regression

Reference by S. Raschka

Training

- Cross entropy

$$H(p,q) = -\sum_i p_i \log(q_i)$$

- Update equation for each class

$$\mathbf{w}_j = \mathbf{w}_j - \eta\, \nabla_{\mathbf{w}_j} J$$

$$J = -\frac{1}{N}\sum_{i=0}^{N} H\left(y_i, \phi\left(\mathbf{x}^{(i)}\right)\right)$$

$$\nabla_{\mathbf{w}_j} J = \frac{1}{N}\sum_{i=0}^{N}\left[\left\{\phi\left(\mathbf{x}^{(i)}\right) - y^{(i)}\right\}x_j^{(i)}\right] \quad j \in \{1, \ldots, D\}$$

  - $y^{(i)}$: true class label
  - $\phi\left(\mathbf{x}^{(i)}\right)$: softmax output (not the predicted label)
  - J : average of all cross entropies over all training examples
  - $\mathbf{w}_j$: weight vector for j^{th} feature
  - $x_j^{(i)}$: j^{th} feature in i^{th} training example

Prediction

- Compute a score for each class

$$s_j(\mathbf{x}) = p(y = j|\, \boldsymbol{z}) = \frac{\exp(z_j)}{\sum_{i=1}^{K} \exp(z_i)}$$

Where $z = \mathbf{w}^T \mathbf{x}$

- Choose the class with the highest score

$$\hat{y} = \text{argmax}_j\ s_j(\mathbf{x})$$