Softmax Regression.

Given training set $\{(x^{(1)}, y^{(1)}), \cdots, (x^{(m)}, y^{(m)})\}$, where $y^{(i)} \in \{1, 2, \cdots, k\}$.

- $$h_\theta(x^{(i)}) = \begin{bmatrix} p(y^{(i)}=1 \mid x^{(i)};\theta) \\ p(y^{(i)}=2 \mid x^{(i)};\theta) \\ \vdots \\ p(y^{(i)}=k \mid x^{(i)};\theta) \end{bmatrix}_{k\times 1} = \underbrace{\frac{1}{\sum_{j=1}^{k} e^{\theta_j^T \cdot x^{(i)}}}}_{\text{Normalize the distribution to 1.}} \cdot \begin{bmatrix} e^{\theta_1^T \cdot x^{(i)}} \\ e^{\theta_2^T \cdot x^{(i)}} \\ \vdots \\ e^{\theta_k^T \cdot x^{(i)}} \end{bmatrix}_{k\times 1}.$$

$$X = \begin{bmatrix} \cdots & x^{(1)T} & \cdots \\ \cdots & x^{(2)T} & \cdots \\ & \vdots & \\ \cdots & x^{(m)T} & \cdots \end{bmatrix}_{m\times(n+1)}, \quad \theta = \begin{bmatrix} \cdots & \theta_1^T & \cdots \\ \cdots & \theta_2^T & \cdots \\ & \vdots & \\ \cdots & \theta_k^T & \cdots \end{bmatrix}_{k\times(n+1)}.$$

Cost Function:

$$J(\theta) = -\frac{1}{m} \cdot \left[\sum_{i=1}^{m} \sum_{j=1}^{k} \underbrace{1\{y^{(i)}=j\}}_{\text{indicator function}} \cdot \log \frac{e^{\theta_j^T \cdot x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_l^T \cdot x^{(i)}}} \right] + \frac{\lambda}{2} \cdot \sum_{i=1}^{k} \sum_{j=0}^{n} \theta_{ij}^2$$

indicator function, $1\{\text{true statement}\}=1$, $1\{\text{false statement}\}=0$

$$\nabla_{\theta_j} J(\theta) = -\frac{1}{m} \cdot \sum_{i=1}^{m} \left[x^{(i)} \cdot \left(1\{y^{(i)}=j\} - \underbrace{p(y^{(i)}=j \mid x^{(i)};\theta)}_{= \frac{e^{\theta_j^T \cdot x^{(i)}}}{\sum_{j=1}^{k} e^{\theta_j^T \cdot x^{(i)}}}} \right) \right] + \lambda \cdot \theta_j .$$

Softmax Regression vs k Binary classifiers.

If the k classes are mutually exclusive $\rightarrow$ softmax classifier

otherwise, k separate classifiers are preferred.

Sparse function in matlab.

$S = \text{sparse}(i, j, v)$ generates a sparse matrix S from the triplets i, j, and v such that $S(i(k), j(k)) = v(k)$. The max(i)-by-max(j) output matrix has space allotted for nonzero elements.

e.g: $y = \begin{bmatrix} 3 \\ 5 \\ 1 \end{bmatrix}_{3\times1}$, sparse(y, 1:3, 1) $\Rightarrow$ (3,1) = 1, (5,2) = 1, (1,3) = 1

($\downarrow$ set the target value to 1.)

full function in matlab: convert sparse matrix to full matrix.

e.g: $\text{full}(\text{sparse}(y, 1{:}3, 1)) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$

$\boxed{\text{groundTruth = full(sparse(labels, 1:numCases, 1));}}$

Implementation of Softmax Regression.

1. Cost Function and gradient.

i). Set 1, get the max number in each column.
ii). Set 2, get the max number in each row

$M = \text{bsxfun}(@\text{minus}, \theta \times \text{data}, \text{max}(\theta \times \text{data}, [\,], 1));$

θ is $k \times (n+1)$, data is $(n+1) \times m$, $\theta \times$ data is $k \times m$.

In this function, M = each element in each column in $\theta \times$data − max element in each column.

$M = \exp(M);$

$p = \text{bsxfun}(@\text{rdivide}, M, \underbrace{\text{sum}(M)}_{\text{sum of each column in } M});$

In this function, p = each element in each column in M divides by each column element in M, respectively, therefore, normalize the probability.

$$\text{cost} = \frac{-1}{\text{numCases}} \cdot \text{groundTruth}(:)^T \cdot \log(p(:)) + \frac{\text{lambda}}{2} \cdot \text{sum}(\text{theta}(:).\hat{}\,2);$$

$$\text{thetagrad} = \frac{-1}{\text{numCases}} \cdot (\text{groundTruth} - p) \cdot \text{data}^T + \text{lambda} \cdot \text{theta};$$