ML Assignment 2

Arpita Agrawal , Collaborated with: Anjali K Prasad

October 2016

1 Logistic Regression

a) Given n training examples $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$

$$L(w) = -log(\prod_{i=1}^{n} P(Y = y_i|X = x_i)) = -log\prod_{i=1}^{n} \sigma(b + w^T x_n)^{y_n}(1 - \sigma(b + w^T x_n))^{(1-y_n)}$$

$$= -\sum_{i=1}^{n} \Big[y_n log\sigma(b + w^T x_n) + (1 - y_n)log(1 - \sigma(b + w^T x_n))\Big]$$

b)

$$L(w) = -\sum_{i=1}^{n} y_n log\sigma(b + w^T x_n) + (1 - y_n)log(1 - \sigma(b + w^T x_n))$$

$$= \frac{\sigma(a)}{a} = \frac{\partial \frac{1}{1+exp^{-a}}}{\partial a} = \frac{exp^a}{(1 + exp^{-}1)^2} = \frac{1}{1 + exp^{-a}}(1 - \frac{1}{1 + exp^{-a}})$$

$$\frac{\sigma(a)}{a} = \sigma(a)\sigma(1 - a), \frac{\partial log\sigma(a)}{a} = (1 - \sigma a) - (4)$$

Using the values found in equation (4) above and substituting a = w$^T x_n$

$$= -\sum_{i=1}^{n} \frac{\partial L(w)}{\partial w} == -\sum_{i=1}^{n} y_n(1 - \sigma(a))x_n + (1 - y_n)(1 - \sigma(a))x_n$$

$$= -\sum_{i=1}^{n} -y_n\sigma a x_n + x_n y_n - \sigma a x_n + y_n \sigma a x_n$$

$$= -\sum_{i=1}^{n} x_n y_n - \sigma a x_n = -\sum_{i=1}^{n} x_n(y_n - \sigma a) = \sum_{i=1}^{n} x_n(\sigma(w^T x_n) - y_n)$$

update Rule for w(t+1) = w(t) - $\eta \sum_{i=1}^{n} x_n(\sigma(w^T x_n) - y_n)$

$$\frac{\partial L^2(w)}{\partial w w^T} = x_i x^T \sigma(w^T x_i)(1 - \sigma(w^T x_i))$$

The above double derivative equations shows that our function converges to a global minimum, as the double derivate is > 0

c) Negative log likelihood L(w1, ..., wK), where we can simplify the multiclass logistic regression expression above by introducing an additional fixed parameter w$_K$ = 0

$$L(w_1, w_2, ..., w_k) = log\prod_{j=1}^{n} P_k(Y_j = k|X = x_j)^{y_{jk}} = \sum_{j=1}^{n}\sum_{i=1}^{K} log P_k(Y_j = k|X = x_j)^{y_{jk}}$$

where y_{jk} is a vector of class k, which specifies if each training example j belongs to class k or not.

$$P_k(Y_j = k|X = x_j) = \frac{exp(w_k^T x_j)}{1 + \sum_{k=1}^{K-1} exp^{w_i \cdot \mathbf{X}_j}}$$

for k = K, $w_i = 0$, So $exp0 = 1$,1 in the denominator can be replaced as $exp(w_K^T x_j)$, which = 1

$$-L(w) = -y_{ji}\sum_{j=1}^{n}\sum_{i=1}^{k} log\frac{exp(w_i^T x_j)}{exp(w_K^T x_j) + \sum_{i=1}^{K-1} exp^{w_i^T \cdot \mathbf{X}_j}} = -y_{ji}\sum_{j=1}^{n}\sum_{i=1}^{k} log\frac{exp(w_i^T x_j)}{\sum_{i=1}^{K} exp^{w_i^T \cdot \mathbf{X}_j}}$$

$$= -y_{ji}\sum_{j=1}^{n}\sum_{i=1}^{K} logexp(w_i^T x_j) - log\sum_{i=1}^{K} exp^{w_i^T \cdot \mathbf{X}_j} = \sum_{j=1}^{n}\sum_{i=1}^{K} -y_{ji}(w_i^T x_j) + y_{ji}log\sum_{i=1}^{K} exp^{w_i^T \cdot \mathbf{X}_j}$$

d)

$$\frac{\partial L(w_1, w_2, w_3...., w_i)}{\partial w_i} = \frac{\partial}{\partial w_i}\sum_{j=1}^{n} -y_{ji}(w_i^T x_j) + y_{ji}log\sum_{i=1}^{K} exp^{w_i^T \cdot \mathbf{X}_j}$$

Differentiating w.r.t w_i

$$\frac{\partial L(w_1, w_2, w_3...., w_i)}{\partial w_i} = \sum_{j=1}^{n}\Bigg[- y_{ji}(x_j) + \frac{y_{ji}x_j exp^{w_i^T \cdot \mathbf{X}_j}}{\sum_{i=1}^{k} exp^{w_i^T \cdot \mathbf{X}_j}}\Bigg] = \sum_{j=1}^{n}\Bigg[- y_{ji}(x_j) + y_{ji}x_j P_k(Y_j = k|X = x_j)\Bigg]$$

update rule for w(t+1) = w(t) - $\eta \sum_{j=1}^{n}\Bigg[- y_{ji}(x_j) + y_{ji}x_j P_k(Y_j = k|X = x_j)\Bigg]$, where y_{ij}= Identity if j th training example belogs to class i or not.

2 Logistic Regression

Gaussian Discriminant Analysis, $D = (x_n, y_n)_{n-1}^{N}$ with $y_n \epsilon 1, 2$

using joint distribution $p(x_n, y_n) = p(y_n)p(x_n|y_n)$, given to us

$p(x_n, y_n)$, if $y_n = 1 = p_1 \frac{1}{\sqrt{2\pi\sigma_1^2}} \exp^{\frac{-(x-\mu_1)^2}{2\sigma_1^2}}$

if $y_n = 2 = p_2 \frac{1}{\sqrt{2\pi\sigma_2^2}} \exp^{\frac{-(x-\mu_2)^2}{2\sigma_2^2}}$

$$P(Y = y_i|X) = \sum_{i=1}^{N} P(x^i|y_i = k)P(y_i = k)$$

$$L(P(D)) = \sum_{i=1}^{N} logp(x_i, y_i)$$

$$= \sum_{i=1:y_n=1}^{N} logP(x_i|y_i = 1)P(y_i = 1) + \sum_{i=1:y_n=2}^{N} logP(x_i|y_i = 2)P(y_i = 2)$$

$$= \sum_{i=1:y_n=1}^{N} logp_1 \frac{1}{\sqrt{2\pi\sigma_1^2}} \exp^{\frac{-(x-\mu_1)^2}{2\sigma_1^2}} + \sum_{i=1:y_n=2}^{N} logp_2 \frac{1}{\sqrt{2\pi\sigma_2^2}} \exp^{\frac{-(x-\mu_2)^2}{2\sigma_2^2}} - (1)$$

Maximising the likelihood function: $(p_1^*, p_2^*, \mu_1^*, \mu_2^*, \sigma_1^*, \sigma_2^*) = \arg\max logP(D)$
Simplifying and differentiating the above equation (1) w.r.t each parameter,

$\ell(\theta) = \sum_{i=1:y_n=1}^{N} logp_1 - log\sqrt{2\pi\sigma_1^2} - \frac{(x-\mu_1)^2}{2\sigma_1^2} + \sum_{i=1:y_n=2}^{N} logp_2 - log\sqrt{2\pi\sigma_2^2} - \frac{(x-\mu_2)^2}{2\sigma_2^2}$
Differentiating w.r.t p1

$$\frac{\partial(LL)}{\partial p_1} = \sum_{i=1:y_n=1}^{N}\frac{1}{p_1} + \sum_{i=1:y_n=2}^{N}\frac{-1}{1-p_1} = 0$$

$$\frac{\partial(LL)}{\partial p_1} = \frac{n}{p_1} + \frac{-1(N-n)}{1-p_1} = 0$$

where n = total number of data points which belong to class y=1 and N is the total number of points in the data set.

$$\frac{\partial(LL)}{\partial p_1} = (1-p_1)n - p_1(N-n) = n - np_1 - Np_1 + np_1 = 0$$

$$n = Np_1, p_1 = n/N$$

$$p_2 = 1 - p_1 = 1 - \frac{n}{N} = \frac{N-n}{N}$$

Differentiating equation (1) w.r.t μ_1

$$\frac{\partial(LL)}{\partial\mu_1} = -\sum_{i=1:y_n=1}^{N} \frac{-2(x-\mu_1)}{2\sigma_1^2} = \sum_{i=1:y_n=1}^{N} (x-\mu_1) = 0$$

$$= \sum_{i=1:y_n=1}^{N} x_i = \sum_{i=1:y_n=1}^{N} \mu_1$$

$$\mu_1 = \frac{\sum_{i=1:y_n=1}^{N} x_i}{n}$$

where n is the number of training examples where $y_n = 1$

Similarly differentiating equation (1) w.r.t μ_2

$$\mu_2 = \frac{\sum_{i=1:y_n=2}^{N} x_i}{N-n}$$

Differentiating equation (1) w.r.t σ_1

$$\frac{\partial(LL)}{\partial\sigma_1} = \sum_{i=1:y_n=1}^{N} -\frac{\partial log\sqrt{2\pi\sigma_1^2}}{\partial\sigma_1} - \frac{\partial\frac{(x_i-\mu_1^2)}{\sigma_1^2}}{\partial\sigma_1} = \sum_{i=1:y_n=1}^{N} -\frac{\sqrt{2\pi}}{\sigma_1\sqrt{2\pi}} + 2\frac{(x_i-\mu_1)^2}{2\sigma_1^3} = 0$$

$$\sum_{i=1:y_n=1}^{N} -\frac{1}{\sigma_1} + \frac{(x_i-\mu_1)}{\sigma_1^3} = \sum_{i=1:y_n=1}^{N} -1 + \frac{(x_i-\mu_1)^2}{\sigma_1^2} = 0$$

$$n = \sum_{i=1:y_n=1}^{N} \frac{(x_i-\mu_1)^2}{\sigma_1^2}$$

$$\sigma_1^2 = \frac{\sum_{i=1:y_n=1}^{N}(x_i-\mu_1)^2}{n}$$

$$\sigma_1 = \sqrt{\frac{\sum_{i=1:y_n=1}^{N}(x_i-\mu_1)^2}{n}}$$

Similarly differentiating w.r.t σ_2 we will get,

$$\sigma_2 = \sqrt{\frac{\sum_{i=1:y_n=2}^{N}(x_i-\mu_2)^2}{N-n}}$$

b)

$$p(x|y=c1) = \mathcal{N}(\mu_1,\sigma), p(x|y=c2) = \mathcal{N}(\mu_1,\sigma)$$

$$P(Y=k|X) = \sum_{i=1,k=0/1}^{N} \frac{P(X|y=k)P(y=k)}{P(X)}$$

$$P(Y=c_1|X) = \frac{P(x|y=c_1)P(y=c_1)}{P(x|y_i=c_2)P(y=c_2)+P(x|y=c_1)P(y=c_1)}$$

Dividing both numerator and denominator by numerator

$$P(Y=c_1|X) = \frac{1}{1+\frac{P(X|Y=c_2)P(Y=c_2)}{P(X|Y=c_1)P(Y=c_1)}} = \frac{1}{1+\exp log\frac{P(X|Y=c_2)P(Y=c_2)}{P(X|Y=c_1)P(Y=c_1)}}$$

$$= \frac{1}{1+\exp logP(X|Y=c_2)P(Y=c_2) - logP(X|Y=c_1)P(Y=c_1)}$$

$$= \frac{1}{1 + \exp log\frac{P(Y=c_2)}{P(Y=c_1)} + logP(X|Y=c_2) - logP(X|Y=c_1)} - (1)$$

$$P(X|Y=c_1)P(Y=c_1) = p_1\frac{1}{\sqrt{2\pi|\Sigma^{-1}|}}exp^{\frac{-1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)} - (2)$$

$$P(X|Y=c_2)P(Y=c_2) = p_2\frac{1}{\sqrt{2\pi|\Sigma^{-1}|}}exp^{\frac{-1}{2}(x-\mu_2)^T\Sigma^{-1}(x-\mu_2)} - (3)$$

substituting 2 and 3 in equation 1:

$$= \frac{1}{1 + \exp log\frac{p_2}{p_1} - \frac{1}{2}(x-\mu_2)^T\Sigma^{-1}(x-\mu_2) + \frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)} - (1)$$

using Transpose properties, $(A+B)^T = A^T + B^T$ and simplifying the terms by opening brackets, canceling equal and opposite terms,

$$= \frac{1}{1 + \exp log\frac{p_2}{p_1} + \frac{1}{2}\mu_2^T\Sigma^{-1}\mu_2 - \frac{1}{2}\mu_1^T\Sigma^{-1}\mu_1 + (\mu_1-\mu_2)^T\Sigma^{-1}X} = \frac{1}{1+\exp^{-(b+\theta^T x)}}$$

$$\theta^T = (\mu_1-\mu_2)^T\Sigma^{-1}, b = log\frac{p_2}{p_1} + \frac{1}{2}\mu_2^T\Sigma^{-1}\mu_2 - \frac{1}{2}\mu_1^T\Sigma^{-1}\mu_1$$

3 Programming Assignment :

```
Pearson cofficient of feature  CRIM  :: -0.387696987621
Pearson cofficient of feature  ZN  :: 0.362987295831
Pearson cofficient of feature  INDUS  :: -0.483067421758
Pearson cofficient of feature  CHAS  :: 0.203600144696
Pearson cofficient of feature  NOX  :: -0.424829675619
Pearson cofficient of feature  RM  :: 0.690923334973
Pearson cofficient of feature  AGE  :: -0.390179110401
Pearson cofficient of feature  DIS  :: 0.252420566225
Pearson cofficient of feature  RAD  :: -0.385491814423
Pearson cofficient of feature  TAX  :: -0.468849385373
Pearson cofficient of feature  PTRATIO  :: -0.505270756892
Pearson cofficient of feature  B  :: 0.343434137151
Pearson cofficient of feature  LSTAT  :: -0.73996982063
```

1

Linear Regression::

```
MSE for Linear Regression with all the features (Training Data): 20.950144508
MSE for Linear Regression with all the features (Test Data): 28.4179164975
```

Ridge Regression::

```
Ridge Regression for original 7th data point test sample strategy

Lamda =  0.01
MSE for Training Data ::: 20.9501449001
MSE for Testing Data ::: 28.4182915618


Lamda =  0.1
MSE for Training Data ::: 20.9501836546
MSE for Testing Data ::: 28.42168497


Lamda =  1.0
MSE for Training Data ::: 20.9539918317
```

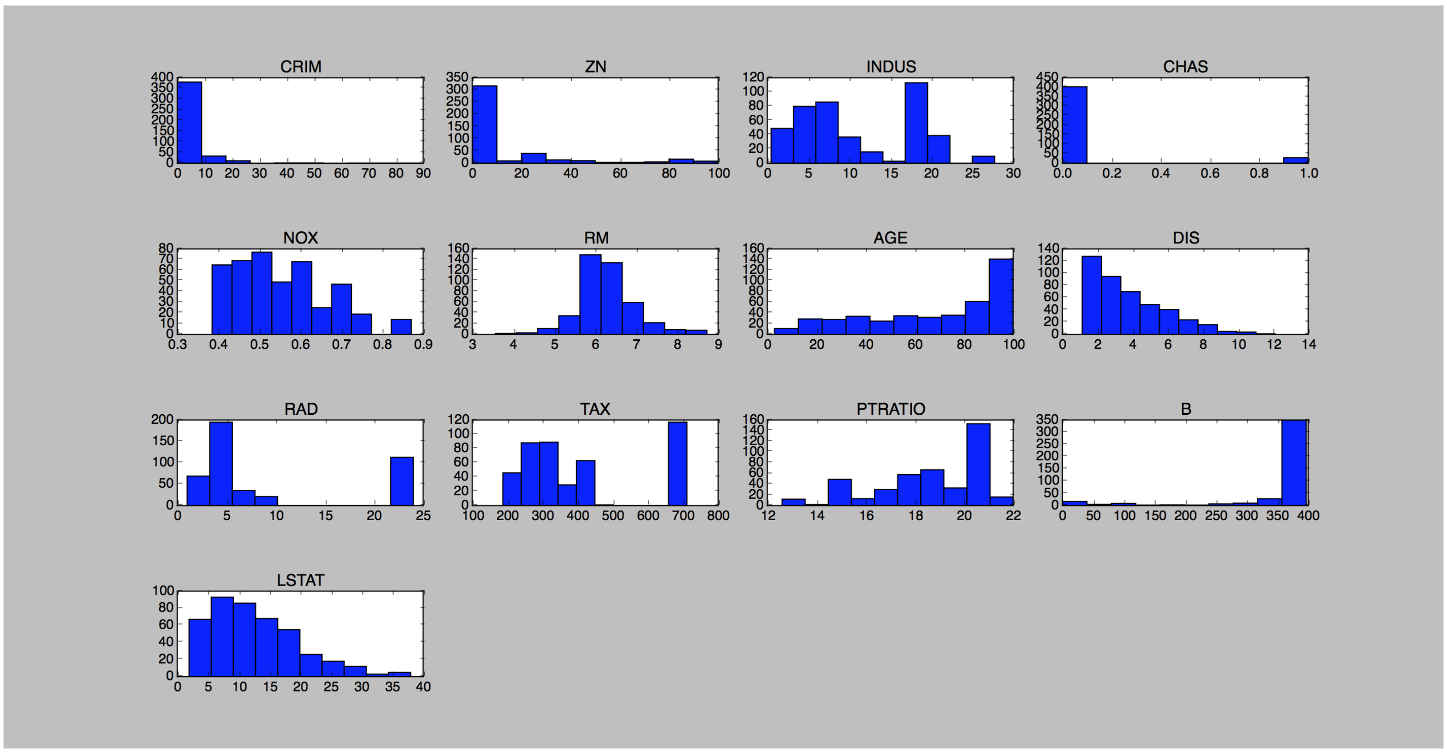

Figure 1: Features Histogram

```
MSE for Testing Data ::: 28.4573733573

Ridge Regression for Cross Validation strategy

winning lamda::::: 1.3901
MSE for CV Training Data ::: 20.9575255035
MSE for CV Testing Data ::: 28.473825784

with each run, my lamda ranges from (0.5 - 1.5)
```

Linear Regression(Trying different feature Selection strategies)::

```
Strategy 1: Select the 4 highest correlated features and then train the  linear regressor
INDUS, RM , PTRATIO, LSTAT

MSE for Linear Regression with 4 highest correlated features (Training Data):
26.4066042155
MSE for Linear Regression with 4 highest correlated features (Testing Data):
31.4962025449


Strategy 2: Select the 4 features iteratively and then train the  linear regressor
LSTAT, RM, PTRATIO, CHAS

MSE for Linear Regression with 4 features calculated iteratively (Training Data):::
25.1060222464
MSE for Linear Regression with 4 features calculated iteratively (Testing Data):::
34.6000723135


Strategy 3: Brute force search on all combinations of 4 and then train the linear regressor
CHAS,RM,PTRATIO,LSTAT
MSE for Linear Regression with brute force best 4 features (Training Data):
25.1060222464
MSE for Linear Regression with brute force best 4 features (Testing Data):
34.6000723135


Polynomial Expansion of Features

MSE for Linear Regression with all the combinations of features (Training Data):
5.05978429711
MSE for Linear Regression with all the combinations of (Test Data): 14.5553049733
```