CS 260 : Machine Learning Algorithm
Homework #2

Ronak Sumbaly
UID - 604591897

October 15, 2015

Problem 1 - Naive Bayes

(a) **Canonical Setup**

Suppose $X = \{X_i\}_{i=1}^{D} \in \mathbb{R}^D$ represents the features and $Y \in (0, 1)$represents the class labels. Assumptions given,

1. The label variable Y follows a Bernoulli distribution, with parameter $\pi = P(Y = 1)$.
2. For each feature X_j, we have $P(X_j|Y = y_k)$ which follows a Gaussian distribution $N(\mu_{jk}, \sigma_j)$

To show that:

$$P(Y = 1|X) = \frac{1}{1 + exp(-w_0 + w^T X)}$$

Calculation
We know that $P(Y = 1) = \pi$ so using Baye's rule we can re-write the above expression as,

$$P(Y = 1|X) = P(X|Y = 1)\frac{P(Y = 1)}{P(X)} = \pi\frac{P(X|Y = 1)}{P(X)} \tag{1}$$

So we now calculate the probability on the basis of the separate classes,

$$P(X) = \sum_y P(X|Y = y) = \pi P(X|Y = 1) + (1 - \pi)P(X|Y = 0) \tag{2}$$

Equation 1. can therefore be written as,

$$P(Y = 1|X) = \frac{1}{1 + \frac{(1 - \pi)P(X|Y = 0)}{\pi P(X|Y = 1)}} \tag{3}$$

Formulating the above expression in the form of e and log we get,

$$P(Y = 1|X) = \frac{1}{1 + e^{(\log(\frac{1-\pi}{\pi}) + \log(\frac{P(X|Y=0)}{P(X|Y=1)}))}} \tag{4}$$

$$= \frac{1}{1 + e^{(\log(\frac{1-\pi}{\pi}) + \sum_{j=1}^{D} \log(\frac{P(X_j|Y=0)}{P(X_j|Y=1)}))}} \tag{5}$$

We will use the following substitution in the above expression,

$$P(X|Y=y_k) = \prod_{j=1}^{D} \frac{1}{(2\pi\sigma_j^2)} e^{-(\frac{1}{2\sigma_j^2}(x_j-\mu_{jk})^2)}$$

Thus we get substituting for both the classes,

$$P(Y=1|X) = \frac{1}{1+exp(\log(\frac{1-\pi}{\pi}) + \sum_{j=1}^{D}(\frac{-(x_j-\mu_{j0})^2}{2\sigma_j^2} + \frac{(x_j-\mu_{j1})^2}{2\sigma_j^2}))} \tag{6}$$

$$= \frac{1}{1+exp(\log(\frac{1-\pi}{\pi}) + \sum_{j=1}^{D}(\frac{(\mu_{j0}^2-\mu_{j1}^2)}{2\sigma_j^2} - \frac{(\mu_{j0}-\mu_{j1})}{\sigma_j^2}x_j)} \tag{7}$$

Comparing equation 7. to the original term

$$P(Y=1|X) = \frac{1}{1+exp(-w_0+w^TX)}$$

We have,

$$w_0 = -\log(\frac{1-\pi}{\pi}) - \sum_{j=1}^{D}(\frac{(\mu_{j0}^2-\mu_{j1}^2)}{2\sigma_j^2} \tag{8}$$

$$w = \frac{(\mu_{j0}-\mu_{j1})}{\sigma_j^2} \tag{9}$$

Hence we have found the explicit form of w_0 and $\mathbf{w}$

(b) **Maximum Likelihood Estimation for Naive Bayes with Gaussian Assumption**

Calculation

The maximum likelihood estimation L for Naive Bayes will be defined as follows:

$$L = log \prod_{c=0}^{1} \prod_{i=1,y_i=c}^{N} \prod_{j=1}^{D} \Big[\frac{exp\{-\frac{(x_{ji}-\mu_{jc})^2}{2\sigma_j^2}\}}{(2\pi\sigma_j^2)}\Big] \tag{10}$$

$$= -\sum_{c=0}^{1} \sum_{i=1,y_i=c}^{N} \sum_{j=1}^{D} \Big[\frac{1}{2}log(2\pi\sigma_j^2) - \frac{(x_{ji}-\mu_{jc})^2}{2\sigma_j^2}\Big] \tag{11}$$

Differentiating w.r.t. μ_{jc} we get,

$$\frac{\partial}{\partial\mu_{jc}}L = \sum_{i=1,y_i=c}^{N} \frac{x_{ji}-\mu_{jc}}{\sigma_j^2} = 0 \tag{12}$$

We equate the above to zero and split to summation to get the following terms,

$$\sum_{i=1,y_i=c}^{N} x_{ji} = \sum_{i=1,y_i=c}^{N} \mu_{jc} = N_c \mu_{jc} \tag{13}$$

$$\mu_{jc} = \frac{1}{N_c} \sum_{i=1,y_i=c}^{N} x_{ji} \tag{14}$$

Differentiating w.r.t σ_j we get,

$$\frac{\partial}{\partial \sigma_j^2} L = \sum_{c=0}^{1} \Big[- \sum_{i=1,y_i=c}^{N} \frac{1}{2\sigma_j^2} + \sum_{i=1,y_i=c}^{N} \frac{1}{2\sigma_j^4} (x_{ji} - \mu_{jc})^2 \Big] \tag{15}$$

Like before equate the differentiation to zero and separate

$$\sum_{c=0}^{1} \sum_{i=1,y_i=c}^{N} \frac{1}{\sigma_j^2} = \sum_{c=0}^{1} \sum_{i=1,y_i=c}^{N} \Big[- \frac{1}{\sigma_j^4} (x_{ji} - \mu_{jc})^2 \Big] \tag{16}$$

Removing summation and equating we get,

$$\sigma_j = \sqrt{\frac{1}{N} \sum_{c=0}^{1} \sum_{i=1,y_i=c}^{N} (x_{ji} - \mu_{jc})^2} \tag{17}$$

Hence the maximum likelihood estimation for Naive Bayes with Gaussian Assumption are equations 14 , 17.

Problem 2 - Logistic Regression

(a) **Negative Log Likelihood - Binary Logistic Regression Model**

$$\mathcal{L}(w) = -\sum_n \{y_n \log(\sigma(w^T x_n)) + (1-y) log(1-\sigma(w^T x_n))\}$$

Where, $\sigma[.]$ denotes the sigmoid function and values of x and w have be created by appending ones to the first column of x and the value b to the first column of w

(b) **Is the loss function convex?**

Yes, the function is convex. In order to show that the function is convex we state that since the loss function is a linear combination of two functions we will have to show that both the functions in above equation are convex. Functions to consider,

$$-\log(\sigma(w^T x_n)) \tag{18}$$
$$-log(1-\sigma(w^T x_n)) \tag{19}$$

In order for convexity to hold, the second order derivative should be ≥ 0 and the Hessian matrix H must be positive semi definite i.e $z \cdot H \cdot z^T \geq 0 \; \forall$ non-zero column vector z. Considering the functions individually we take the derivative 18.

$$\nabla_w[-log(\frac{1}{1+e^{-w^t x}})] = \nabla_w \log(1+e^{-w^t x}) \tag{20}$$
$$= \frac{-e^{-w^T x} x}{1+e^{-w^T x}} \tag{21}$$
$$= (\sigma(w^T x) - 1)x \tag{22}$$

We know that,

$$\sigma^{'}(w^T x) = \sigma(w^T x)[1-\sigma(w^T x)] \tag{23}$$

Hence, the Hessian can be calculated by taking the derivative again of equation 2

$$\nabla_w[(\sigma(w^T x) - 1)x] = \sigma(w^T x)[1-\sigma(w^T x)]xx^T \tag{24}$$
$$\tag{25}$$
$$= X^T diag(\sigma(w^T x)[1-\sigma(w^T x)])X \tag{26}$$

Since the diagonal of the matrix is strictly positive, it would make all the eigenvectors strictly positive as well. Make the above expression ≥ 0. Thus the Hessian matrix is Positive Semi-definite. Therefore the function 18 is convex.
Now for function in equation 19

$$\nabla_w[-log(\frac{e^{-w^t x}}{1+e^{-w^t x}})] = \nabla_w[w^T x + \log(1+e^{-w^t x})] \tag{27}$$
$$= x - \frac{e^{-w^t x} x}{1+e^{-w^t x}} \tag{28}$$
$$= (1 - \frac{e^{-w^t x}}{1+e^{-w^t x}})x \tag{29}$$

The above Hessian equation takes the same form as 21 proved in part 1 of this proof.
Therefore, we can conclude that both the functions are convex.
Hence from the above conclusion we can say that the Loss Function is convex in nature.

(c) **Prove**

Show that the magnitude of the optimal w can go to infinity when the training samples are linearly separable.

Proof

Consider the loss function provided in Problem 2.a. Splitting the loss function on the basis of class gives us,

$$\begin{aligned}\mathcal{L}(w) &= -\sum_{i=1,y_i=1}^{n} \log(\sigma(w^T x_n)) - \sum_{i=1,y_i=0}^{n} log(1-\sigma(w^T x_n)) \\ &= -\sum_{i=1,y_i=1}^{n} \log(\sigma(w^T x_n)) - \sum_{i=1,y_i=0}^{n} log(\sigma(-w^T x_n))\end{aligned}$$

Since $1-\sigma[.] = \sigma[-.]$

We know that the data is linearly separable, which means that there should exist w such that is splits the data into both the class perfectly. This means that for $y_i = 1$ the function would be evaluated only at a positive point and for $y_i = 0$ at a negative point.

So what we require is to minimize the above function $-log(\sigma[.])$. So we want $\sigma[.] \in (0,1)$ to approach value of 1. For this to happen $w^T x$ needs to be very large. Since we know that the value of x are constant, this means that w will need to be very large which basically implies that w "goes to infinity".

(d) **Compute the gradient respect to w_i, i.e $\frac{\partial \mathcal{L}}{\partial w_i}$**

Calculation - Similar to steps followed for Problem 2.b.

$$\left[\frac{\partial \mathcal{L}}{\partial w_i}\right] = -\sum_{i}^{n}\left[y_i \frac{1}{\sigma[w^T x_i]}\frac{d}{dw}\sigma[w^T x_i] + (1-y_i)\frac{1}{1-\sigma[w^T x_i]}\frac{d}{dw}(1-\sigma[w^T x_i])\right] \tag{30}$$

$$= -\sum_{i=1}^{n}\left[y_i(1-\sigma[w^T x_i])x_i + (1-y_i)\sigma[w^T x_i]x_i\right] \tag{31}$$

$$= -\sum_{i=1}^{n}\left[(\sigma[w^T x_i] - y_i)x_i\right] \tag{32}$$

Therefore from the above calculation we get

$$\frac{\partial \mathcal{L}}{\partial w_i} = -\left[(\sigma[w^T x_i] - y_i)x_i\right] + 2\lambda \sum_{i}^{n} w_i$$

(e) **Show that the above equation has one unique solution**

We have proven using 26 that the function is convex in nature. By the linear property of partial derivative operators, addition of a strictly convex function and a convex function is also strictly convex which means that the function has a single minimum.

As there is only a single unique minimum, the gradient of the loss function should have a unique solution.

Problem 3 - Decision Tree

(a) **Canonical Setup**

$$Entropy = E(S) = -\sum_{i=1} p_i \ log_2 \ p_i$$

Calculation

Calculating and comparing the Entropy for all the fields

WEATHER

$$= \frac{28}{100} \times (\frac{-23}{28} \log \frac{23}{28} - \frac{5}{8} \log \frac{5}{28}) + \frac{72}{100} \times (\frac{-50}{72} \log \frac{50}{72} - \frac{22}{72} log \frac{22}{72})$$

$$= 0.28 \times 0.676 + 0.72 \times 0.887$$

$$= 0.827$$

TRAFFIC

$$= \frac{73}{100} \times (\frac{-73}{73} \log \frac{73}{73} + 0) + 0.27 \times (\frac{-27}{27} \log \frac{27}{27})$$

$$= 0$$

Since "Traffic" has a less entropy or basically "Traffic" gives a perfect prediction of "Accident Rate" we split the decision tree using "Traffic".

(b) **Difference between trees $\mathbf{T}_1$ and $\mathbf{T}_2$**

NO DIFFERENCE.

The second student performed normalization of the values before construction of the decision tree. This results in only scaling of the graph but doesn't change the number of output class values or the probability of getting these features. Since decision trees are partitioning the space of observations along each axis. We can transform the feature values for the first student by taking each decision boundary and then subtracting the mean and dividing by the variance for the corresponding features. Thereby getting the same values of the second student with no difference in the probability. Hence in other words, both the trees will have the same structure since the probability would remain same in both the cases.

(c) **Prove**

Prove that, for any discrete probability distribution p with K classes, the value of the Gini index is less than or equal to the corresponding value of the cross-entropy.

Proof

Consider the following formulas:

$$Gini\ Index = \sum_{k=1}^{K} p_k(1 - p_k)$$

$$Cross\ Entropy = \sum_{k=1}^{K} p_k(log\ p_k)$$

Consider the difference between the Gini Index and Cross Entropy

$$\begin{aligned} Gini\ Index - Cross\ Entropy &= \sum_{k=1}^{K} p_k(1 - p_k) - \sum_{k=1}^{K} p_k(log\ p_k) \\ &= \sum_{k=1}^{K} p_k(1 - p_k + log\ p_k) \end{aligned}$$

Examining the function $f(x) = 1 - x + log(x)$ to simply the task of considering probability. We know that f is continuous and $\in +\mathbb{R}$. Taking the derivative we get,

$$\frac{df}{dx} = -1 + \frac{1}{x\ log(a)}$$

where we're considering a to be the base of the log

Like $f(x)$ the derivative also is continuous and $\in +\mathbb{R}$. This condition is satisfied for all $a \leq e$, $log(a) \leq 1$ which means that $\frac{1}{x\ log(a)} \leq 1$ for all $x \in (0, 1)$ and for $x = 1$, we get $\frac{1}{x\ log(a)} = 1$

We can thus imply that $\frac{df(x)}{dx} > 0$ for all $x \in (0, 1)$ and $a < e$ which makes it a decreasing function.

Thus, using the difference result, $1 - p_k + log\ p_k < 0 \Rightarrow Gini\ Index - Cross\ Entropy < 0$. Hence proving that the Gini Index is always less than the Cross Entropy.

Problem 4 - Comparing Classifier

kNN Classifier

kNN Classification Accuracy			
k	Training	Testing	Validation
1	77.7895	79.4344	75.5784
3	85.7895	88.6889	83.8046
5	88.9474	92.0308	85.8612
7	90.5263	90.7455	86.8895
9	90.3158	91.5167	87.4036
11	90.3158	89.7172	88.9460
13	89.4737	88.9460	87.4036
15	89.2632	87.6607	85.6041
17	87.8947	87.6607	85.6041
19	86.7368	87.4036	85.6041
21	86.3158	86.8895	84.3188
23	85.6842	86.8895	83.8046

Decision Tree Classifier

Decision Tree Accuracy - Gini			
s	Training	Testing	Validation
1	93.7895	89.2031	86.3753
2	93.7895	86.3753	87.1465
3	93.7895	87.6607	89.9743
4	93.8947	87.1465	89.7172
5	93.3684	87.4036	88.4319
6	92.6316	85.3470	84.8329
7	90.9474	86.1183	83.8046
8	91.4737	82.7763	83.5476
8	85.0900	87.6607	82.0051
9	91.0526	84.5758	84.8329
10	90.7368	83.5476	82.0051

Decision Tree Accuracy - Entropy			
s	Training	Testing	Validation
1	94.1053	91.0026	85.8612
2	94.3158	89.7172	87.1465
3	94.1053	88.1748	85.0900
4	94.1053	89.4602	89.2031
5	93.4737	88.4319	88.6889
6	92.4211	87.6607	86.8895
7	91.5789	89.2031	84.8329
8	89.2632	87.6607	85.0900
8	92.1053	87.1465	83.8046
9	90.8421	86.1183	86.3753
10	91.1579	83.5476	83.0334

Naive Bayes Classifier

Naive Bayes Accuracy		
Training	Testing	Validation
87.4737	82.7763	82.2622

Logistic Regression Classifier

Logistic Regression Accuracy		
Training	Testing	Validation
94.53	91.52	91.52