UNIVERSITY OF SOUTHERN CALIFORNIA, COMPUTER SCIENCE

Machine Learning: HomeWork 1

Rohit Kondekar
740-581-9473

September 24, 2014

1 Question 1

1.1 Parametric form of Naive Bayes with Gaussian Assumption

Solution:
X = X1, X2 ..... XD is a continuous RV with Gaussian Distribution
Y = 0,1 Binary RV - Bernoulli Distribution with P(Y=1) = π

$$
\begin{aligned}
P(Y=1|X) &= \frac{P(X|Y=1)P(Y=1)}{P(X|Y=1)P(Y=1)P(X|Y=0)P(Y=0)} \\
&= \frac{1}{1+\frac{P(Y=0)}{P(Y=0)}\frac{P(X|Y=0)}{P(X|Y=1)}} \\
&= \frac{1}{1+e^{(\log(\frac{1-\pi}{\pi})+\log(\frac{P(X|Y=0)}{P(X|Y=1)}))}} \\
&= \frac{1}{1+e^{(\log(\frac{1-\pi}{\pi})+\sum_{j=1}^{D}\log(\frac{P(X_j|Y=0)}{P(X_j|Y=1)}))}} \\
&= \frac{1}{1+exp(\log(\frac{1-\pi}{\pi})+\sum_{j=1}^{D}(\frac{-(x_j-\mu_{j0})^2}{2\sigma_j^2}+\frac{(x_j-\mu_{j1})^2}{2\sigma_j^2}))} \\
&= \frac{1}{1+exp(\log(\frac{1-\pi}{\pi})+\sum_{j=1}^{D}(\frac{(\mu_{j0}-\mu_{j1})X_j}{2\sigma_j^2}-\frac{(\mu_{j0}-\mu_{j1})}{2\sigma_j^2})} \\
&= \frac{1}{1+exp(-w_0+w^TX)}
\end{aligned}
$$

1.2 Parameter Estimation for Naive Bayes with Gaussian Assumption

Maximum Likelihood for Naive Bayes-

$$L(\theta) = \sum_{i=1}^{N}\log(P(X_i, Y_i))$$

$$= \sum_{i=1}^{N} \log(P(Y_i)) + \sum_{i=1}^{N} \log(\prod_{j=1}^{d} P(X_i|Y_i))$$
$$= \sum_{i=1}^{N} \log(P(Y_i)) + \sum_{i=1}^{N} \sum_{j=1}^{D} log(P(X_{ji}|Y_i))$$

Two Terms can be estimated separately.

$$\sum_{i=1}^{N} \log(P(Y_i)) = \sum_{c} \log(\Pi_c) \times (\text{data points labeled as c})$$

Therefore

$$\pi^* = \frac{\#\text{No. of data points labeled as } 1}{N}$$
$$1 - \pi^* = \frac{\#\text{No. of data points labeled as } 0}{N}$$

For other term:

$$\sum_{i=1}^{N} \sum_{j=1}^{D} (\frac{-\log(2\pi\sigma^2_{jY_i})}{2} - \frac{(X_{ji} - \mu_{jY_i})^2}{2\sigma^2_{jY_i}})$$

Differentiating wrt $\sigma^2_{jY_i}$

$$\sigma^2_{jY_i} = \sum_{i=1}^{N} \sum_{j=1}^{D} \frac{(X_{ji} - \mu_{jY_i})^2}{N}$$

Differentiating wrt μ_{jY_i}

$$\mu_{jY_i} = \sum_{i=1}^{N} \sum_{j=1}^{D} \frac{X_{ji}}{N}$$

2 Question 2 Nearest Neighbor

2.1

Q. Assume the feature dimensionality D is 3. One way to learn the optimal feature weights is to do an exhaustic search. To enable the search, we often discretize the weights so that the search space is finite. Describe the procedure to search for the optimal feature weights.

Discussion In this situtation, the best way of getting optimal feature weights is to discretize the weights into N equal parts between 0-1. After dividing into N parts, you would have total number of weight combinations as N*N*N.

For each of this combination, run KNN and check out the accuracy on the training data for all the points using Leave One Out Stratergy (LOO). Choose the weights (w1,w2,w3) for which the accuracy is best.

For more exhaustive search, this can be repeated over multiple value of N's i.e. 10,20,30... and the best weight combination for accuracy can be taken into consideration.

The problem with this approach is that, even for small number of partitions the number of combinations increases rapidly i.e. for N=10, there are 1000 combinations we have to check. And as KNN computes distance for all the points, it would be very slow for large datasets.

2.2

Q. What if the feature dimensionality is very high , say D = 10000, does the approach in (a) still work? If not, can you propose a new approach? (Note: your approach does not need to be globally optimal).

No This approach won't work because the number of features are too many and there would be too many possible combinations to make it practicle.

You can do this in two ways:

One of the simple approach which can be used is to consider the features as independent of each other. So bascically, you will be fixing weights of other features as 0 and vary a single features weight and just use that feature (with the weight) to classify and get the accuracies. Choose the weight for which you get maximum accuracy for that particular feature.

This can be done for all the features, so if there are 10000 features and 4 partitions, it would take around 40000 KNN runs to get the best possible weights for all the features. All this can be done using principle of Naive i.e. assuming they are independent.
Other more subtle methods which can be used are: Mutual Information and Correlation between features and class labels. Depending on the correlation values we can decide the weights which should be alloted to different features.

3 Question 3: Logistic Regression

3.1

In Logistic Regression, the conditional probability is given by (Please note the w includes the bias term):

$$P(y_n|x_n;w) = \sigma(w^T x_n)^{y_n}[1-\sigma(w^T x_n)]^{1-y_n}$$

Cross Entropy (Loss Function/Cost Function) is given by taking negative log:

$$L(w) = -\sum_n \{y_n \log(\sigma(w^T x_n)) + (1-y) log(1-\sigma(w^T x_n))\} \tag{3.1}$$

3.2 Is this loss function convex? Provide your reasoning.

Yes it is convex. We need to show that these two functions are convex from eq 3.1:

$$-\log(\sigma(w^T x_n)) \tag{3.2}$$

$$-log(1-\sigma(w^T x_n)) \tag{3.3}$$

As the loss function eq. 3.1 is a linear combination of these two functions.

Second Order condition of convexity states that for the function to be convex the Hessian Matrix should be positive semi definite i.e.

$$\forall x \qquad x^T \nabla_x^2 f(x) x \geqslant 0$$

Taking Gradient of eq. 3.2

$$\begin{aligned}\nabla_w[-log(\frac{1}{1+e^{-w^t x}})] &= \nabla_w \log(1+e^{-w^t x}) \\ &= \frac{-e^{-w^T x} x}{1+e^{-w^T x}} \\ &= (\sigma(w^T x)-1)x\end{aligned}$$

As,

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

Hessian is given by:

$$\nabla_w[(\sigma(w^T x)-1)x] = \sigma(w^T x)[1-\sigma(w^T x)]xx^T$$

$$\begin{aligned}\forall z: \qquad z^T H_w Z &= z^T[\sigma(w^T x)[1-\sigma(w^T x)]xx^T]z \\ &= \sigma(w^T x)[1-\sigma(w^T x)](x^T z)^2 \geq 0\end{aligned}$$

Therefore the function 3.2 is convex.

Now for function in eq 3.3

$$\begin{aligned}\nabla_w[-log(\frac{e^{-w^t x}}{1+e^{-w^t x}})] &= \nabla_w[w^T x + \log(1+e^{-w^t x})] \\ &= x - \frac{e^{-w^t x}x}{1+e^{-w^t x}} \\ &= (1-\frac{e^{-w^t x}}{1+e^{-w^t x}})x\end{aligned}$$

The Hessian of this equation is similar to we solved in the first part, and which we proved is convex. Therefore both the functions are convex.

Therefore the Cross Entropy function is convex.

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

As the pdf for logistic regression is given by:

$$P(y_n|x_n;w) = \sigma(w^T x_n)^{y_n}[1-w(w^T x_n)]^{1-y_n}$$

It is also given that data is linearly separable.
Therefore the sigma value for $y_n = 1$ should be one and 0 otherwise. i.e. the sigma value has to be 0 or 1.

The value of w in this case would be:

$$\begin{aligned}\sigma(w^T x_n) &= 1 \\ \frac{1}{1+e^{-w^T x}} &= 1 \\ e^{-w^T x} &= 0\end{aligned}$$

This is possible when w is very large or $w \to \infty$

3.4

After adding the penalty term to the loss function:

$$L(w) = -\sum_n \{y_n \log(\sigma(w^T x_n)) + (1-y)log(1-\sigma(w^T x_n))\} + \lambda \sum_j w_j^2 \tag{3.4}$$

(3.5)

Using Equations from earler problem, for gradient of two functions:

$$\nabla L(w) = \sum_{i=1}^{n}[y_i[(\sigma(w^T x_i)-1)x_i] + (1-y_i)[(\sigma(w^T x_i)-2)x_i]] + 2\lambda \sum_j w_j \tag{3.6}$$

3.5

As the regularizer given is a l2 norm, which is a well known convex regularizer as: $||w||_2^2 = w1^2 + w2^2 + ...$ gives out a figure in form of :

Figure 3.1: Norms

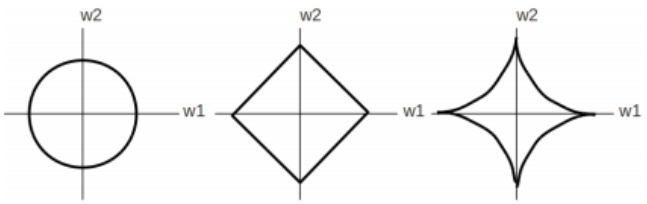

Figure : ℓ_2, ℓ_1, and $\ell_p(p < 1)$ balls in two dimensions

As $f(x) = x^2$ is strictly convex function. And as we proved that the eq. 3.6 is convex function (in earlier part), the conbination of two convex function must be a convex function.

the twice differentiation of w^2 is 2. This shows that the equation is postive definite, as it cannot be 0 after adding this convex regularizer. It shows that the whole function is strictly convex (i.e. as it is positive definite its global minimum is same as local minimum) The value can be obtained using Gradient Descent method.

As there is only a single unique global minimum, the gradient of loss function i.e. the eq. 3.6 should have a unique solution.

4 Question 4: Decision Trees

4.1

$$InfoGain = I(A,B) = H(A) - \sum_{j=a} P(A=a)H(B|A=a)$$

So we just need to compare the values of

$$\sum_{v \epsilon Values(A)} (\frac{S_v}{S}) Entropy(S_v)$$

For Weather

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

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

$$= 0.827$$

For Traffic

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

$$= 0$$

As there is less entropy for Traffic, we will be using traffic to split at the first step of the decision tree.

4.2

The Tree would remain same. As the other student has just normalized the values and has used same parameters for spliting, the probability for values would remain same. For e.g. if he uses entropy to decide the features, the probability would remain same in both the cases, which will ultimately give the same tree. i.e. There is no change in information gain

Normalizing the values only scales the graph, but doesnt change the number of values in the features or the probability. So T1 and T2 obtained would remain same.

4.3

Checking the value of Equation:

$$f(P_k) = -P_k \log(P_k) - P_k \times (1 - P_k) \tag{4.1}$$

It can be proved in two ways:

I) Checking if value of equation 4.1 is always greater than 0.

$$\begin{aligned} f(P_k) &= -P_k \log(P_k) - P_k \times (1 - P_k) \\ &= -P_k \log(P_k) - P_k + P_k^2 \\ &= -P_k (\log(P_k) - 1) + P_k^2 \end{aligned}$$

Now as $0 \leq P_k \geq 1 \Rightarrow \log(P_k)$ is negative. Therefore the term $-P_k(\log(P_k) - 1)$ is always positive. Therefore the whole equation is always positive. Which shows that gini index is always less than cross entropy.

II) Differentiating the equation and equating it to zero gives:

$$\begin{aligned} f'(P_k) = 2P_k - 2 - \log(P_k) &= 0 \\ P_k &= 1 \end{aligned}$$

Checking the value of double derivative at $P_k = 1$

$$f''(P_k) = \frac{-1}{P_k} + 2P_k = 1 \geq 0$$

Therefore it is an increasing function, which means Gini index is less than Cross Entropy in [0,1]

Figure 4.1: Graph Comparison