Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 1

Student Name: Abhishek Kumar
Roll Number: 18111002
Date: August 21, 2018

QUESTION 1

Total number of data points : 800
Total number of data points in class 1 (i.e y=1) : 400
Total number of data points in class 0 (i.e y=0) : 400
Given trees $\mathcal{A}$ and $\mathcal{B}$,

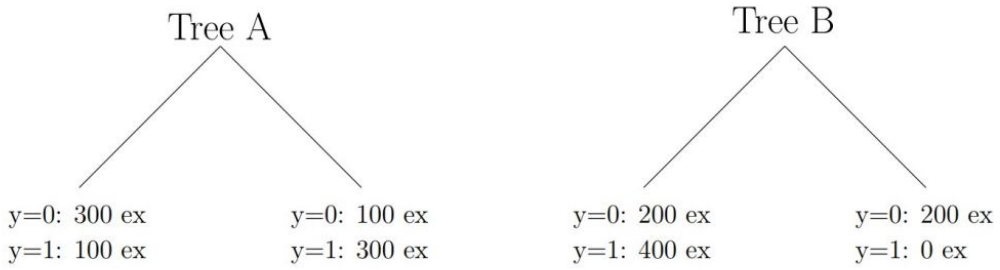

Figure 1: Given tree splits

Part - 1 (Misclassification rate)

Let's denote misclassification rate as $\mathbb{M}$,

$$\mathbb{M} = \frac{No.\ false\ prediction}{No.\ data\ points\ in\ set}$$

here, we have assumed that we are predicting class with higher probability at leaf node.

Tree A

Before splitting at root node,

$$\mathbb{M}_{old} = \frac{400}{800} = 0.50$$

After splitting,

$$\begin{aligned}\mathbb{M}_{new} &= Average(\mathbb{M}_{child}) \\ &= \frac{400}{800} * \frac{100}{400} + \frac{400}{800} * \frac{100}{400} = 0.25\end{aligned}$$

Decrease in misclassification rate,

$$D_{\mathbb{M}_A} = \mathbb{M}_{old} - \mathbb{M}_{new} = 0.50 - 0.25 = 0.25$$

Tree B

Before splitting at root node,

$$\mathbb{M}_{old} = \frac{400}{800} = 0.50$$

After splitting,

$$\begin{aligned}\mathbb{M}_{new} &= Average(\mathbb{M}_{child}) \\ &= \frac{200}{800} * \frac{0}{200} + \frac{600}{800} * \frac{200}{600} = 0.25\end{aligned}$$

Decrease in misclassification rate,

$$D_{\mathbb{M}_B} = \mathbb{M}_{old} - \mathbb{M}_{new} = 0.50 - 0.25 = 0.25$$

We got $D_{\mathbb{M}_A} = D_{\mathbb{M}_B}$, both are equal.

Part - 2 (Information Gain)

Let's denote entropy as $\mathbb{M}$,

$$\mathbb{E} = -\sum_{i=1}^{N} P_i log(P_i)$$

Tree A

Before splitting at root node,

$$\mathbb{E}_{old} = -(\frac{400}{800} log_2(\frac{400}{800}) + \frac{400}{800} log_2(\frac{400}{800})) = 1$$

After splitting,

$$\begin{aligned}\mathbb{E}_{new} &= Average(\mathbb{E}_{child}) \\ &= -2 * (\frac{400}{800} * \frac{100}{400} log_2(\frac{100}{400}) + \frac{400}{800} * \frac{300}{400} log_2(\frac{300}{400})) = 0.81\end{aligned}$$

Information gain in this split,

$$IG_A = \mathbb{E}_{old} - \mathbb{E}_{new} = 1 - 0.81 = 0.19$$

Tree B

Before splitting at root node,

$$\mathbb{E}_{old} = -(\frac{400}{800} log_2(\frac{400}{800}) + \frac{400}{800} log_2(\frac{400}{800})) = 1$$

After splitting,

$$\begin{aligned}\mathbb{E}_{new} &= Average(\mathbb{E}_{child}) \\ &= -(\frac{200}{800} * 0 + \frac{600}{800} * (\frac{200}{600} log_2(\frac{200}{600}) + \frac{400}{600} log_2(\frac{400}{600}))) = 0.69\end{aligned}$$

Information gain in this split,

$$IG_B = \mathbb{E}_{old} - \mathbb{E}_{new} = 1 - 0.69 = 0.31$$

We got $IG_A < IG_B$, so tree B is better split.

Part - 3

Yes we got different answers as decrease in classification errors in both case are same but information gain in $\mathcal{B}$ is more as compared to $\mathcal{A}$ tree. This helps us if we wish to further divide the leaf nodes into child nodes.
This does make sense since average misclassification error will remain same in all cases if the number of misclassifications are equal in all splits. This happens because it is linear combination of misclassifiactions of child nodes.
Whereas in Information gain method entropy is a nonlinear bell shaped function of probability of a outcome of class and is more informative. so average will change depending on how we split our data and it will always be less than or equal to the parent entropy due to its concavity (see in figure 2). Let's see the variation for a better intuition of above given statement.

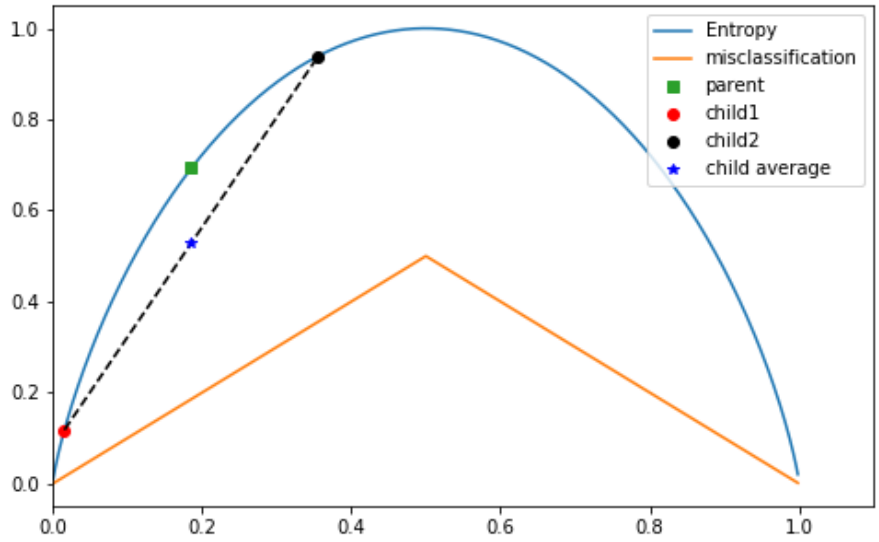

Figure 2: Variation of entropy and classification error with probability of a outcome.

Note that this is just a example plot not related to our data. This concludes the answer for this question.

Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 1

Student Name: Abhishek Kumar
Roll Number: 18111002
Date: August 21, 2018

QUESTION 2

Given that for a 1-NN classifier,

- Bayes optimal rate is zero.(i.e every training data is correctly labeled correctly)
- Number of data points is infinity.
- Let say x is test data and x_{nn} is nearest neighbour of x.

As the number of data points goes to infinity then the point x_{nn} becomes more and more closer to x. we can also say that probability distribution of x over a set of classes is almost same as probability distribution of x_{nn} over a set of classes.

$$\lim_{N\to\infty} p(y_i = c|x) \approx p(y_i = c|x_{nn})$$

Since the x_{nn} is a correctly labeled data then classification of would be correct up to bayes optimal error. **So it will be consistent**.

Also if we do asymptotic analysis for a KNN classifier we get,

$$P_{bayes}(e) \leq P_{kNN}(e) \leq P_{bayes}(e)(2 - \frac{k}{k-1}P_{bayes}(e))$$

here e is error and k is the number of classes, since bayes optimal error is 0,

$$P_{bayes}(e) = P_{kNN}(e) = 0$$

So again we can say **it is consistent**.

Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 1

Student Name: Abhishek Kumar
Roll Number: 18111002
Date: August 21, 2018

QUESTION 3

Nomenclature of dimensions of each matrix just for ease of understanding,

- N : No. of training data objects.
- D : No. of features/dimensions of a data object.
- X : $N \times D$, Training data objects
- Y : $N \times 1$, Training class labels
- x_* : $D \times 1$, Test object.
- $\hat{w}$: $1 \times D$, Weight vector.

For prediction in linear regression model without regularization we can write,

$$f(x_*) = \hat{w}^T x_*$$

since it's scalar we can write,

$$f(x_*) = x_*^T \hat{w}$$

For linear regression $\hat{w} = [X^T X]^{-1} X^T Y$ so we get,

$$\begin{aligned} f(x_*) &= x_*^T [X^T X]^{-1} X^T Y \\ &= x_*^T [X^T X]^{-1} \sum_{n=1}^{N} x_n y_n \\ &= \sum_{n=1}^{N} x_*^T [X^T X]^{-1} x_n y_n \\ &= \sum_{n=1}^{N} w_n y_n \end{aligned}$$

where,

$$w_n = x_*^T [X^T X]^{-1} x_n$$

Here multiplication of $[X^T X]^{-}1$ gives importance to features which are linearly independent of other features and and gives more information about data. (i.e it gives importance to good features dimensions.)

We can see that w_n is kind of similarity between the two points x_n and x_* in dimensions which are linearly independent (i.e it gives a measure of how similar data object is with the test object). Higher the similarity higher will be the chances of given class y_n to be a probable output.

Likewise in K Nearest Neighbour method we generally have,

$$\begin{aligned} f(x_*) &= \sum_{n=1}^{k} w_n y_n \\ &= \sum_{n=1}^{k} \frac{1}{||x_k - x_*||^2} . y_n \end{aligned}$$

Here also w_n is kind of a similarity between the two point. Note that it could also be a cosine similarity, mahalanobis norm etc also.

Conclusion

Both in kNN and Linear Regression the weights are a measure of similarity between the two data objects. But in kNN the weight does not consider which dimensions are more informative and it only depends on data objects, whereas in Linear Regression the $[X^T X]^{-1}$ takes care of that and selects linearly independent features to find similarity on and it depends on all training data objects as well as test object and object with which we are finding similarity.

Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 1

Student Name: Abhishek Kumar
Roll Number: 18111002
Date: August 21, 2018

QUESTION 4

Objective

To regularize each parameter differently, assuming that the hyper-parameters for regularization is given, we define a diagonal Λ matrix width dimensions $D \times D$, where D is number of features.

$$\Lambda = \frac{1}{2}\begin{bmatrix} \lambda_1 & 0 & ... & 0 \\ 0 & \lambda_2 & ... & 0 \\ ... & ... & ... & ... \\ ... & ... & ... & ... \\ 0 & 0 & ... & \lambda_D \end{bmatrix}$$

Now we define out loss/cost function as,

$$\begin{aligned} L(w) &= \sum_{n=1}^{N}(y_n - w^T x_n) + \frac{1}{2}\sum_{d=1}^{D}\lambda_d w_d^2 \\ &= (Y - Xw)^T(Y - Xw) + w^T\Lambda w \end{aligned}$$

To get optimal w in closed form we differentiate our loss function and equate to 0,

$$\frac{\partial L(w)}{\partial w} = -2(Y - Xw)^T X + w^T(\Lambda + \Lambda^T) = 0$$

since RHS is a zero vector we can take transpose and also Λ is a diagonal matrix. so we can write above equation as,

$$\begin{aligned} \frac{\partial L(w)}{\partial w} &= -2X^T(Y - Xw) + 2\Lambda w = 0 \\ &= -2(X^T Y - [X^T X + \Lambda]w) = 0 \end{aligned}$$

From obtained equation we get,

$$w = [X^T X + \Lambda]^{-1} X^T Y$$

This concludes the answer for this question.

Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 1

Student Name: Abhishek Kumar
Roll Number: 18111002
Date: August 21, 2018

QUESTION 5

For a multi-output regression we are proposing a new representation of W to reduce number of parameters to estimate,

$$W = BS$$

where W is a $D \times M$, B is a $D \times K$ and S is a $K \times M$ matrix respectively. so if we keep K small the total number of parameter to estimate reduces.

Estimation of S

Method 1

Our loss function after including new representation of W is,

$$\begin{aligned} L(S) &= \sum_{n=1}^{N}\sum_{m=1}^{M}(y_{nm} - w_m^T x_n)^2 \\ &= \sum_{m=1}^{M}(y_m - Xw_m)^T(y_m - Xw_m) \\ &= Tr((Y - XW)^T(Y - XW)) \\ &= Tr((Y - XBS)^T(Y - XBS)) \end{aligned}$$

Differentiating with respect to S and equating to zero for optima,

$$\begin{aligned} \frac{\partial L(S)}{\partial S} &= \frac{\partial}{\partial S}(-Tr(S^T B^T X^T Y) - Tr(Y^T XBS) + Tr(S^T B^T X^T XB)) = 0 \\ &= -2(B^T X^T Y - [B^T X^T XB]S) = 0 \end{aligned}$$

we get,

$$S = [B^T X^T XB]^{-1} B^T X^T Y$$

Let's take, $\mathbb{X} = XB$

$$S = [\mathbb{X}^T \mathbb{X}]^{-1} \mathbb{X}^T Y$$

which gives similar expression to multi output regression for W but with transformed X as $\mathbb{X}$. Also please note that transformed data has a dimension of $N \times K$ and initial data had $N \times M$, $M > K$ so it is kind of dimensionality reduction of data given some information like B about the data.

Method 2

We could also get above result using pre-obtained solution for W in multioutput regression model as,

$$X^T XW = X^T Y$$
$$X^T XBS = X^T Y$$
$$B^T X^T XBS = B^T X^T Y$$
$$S = [B^T X^T XB]^{-1} B^T X^T Y$$

Estimation of B

Bonus Part

Similarly we can derive closed form solution for B either with first or second method we will get the same expression. so lets go with second method,

$$X^T XW = X^T Y$$
$$X^T XBS = X^T Y$$
$$X^T XBSS^T = X^T YS^T$$
$$B = [X^T X]^{-1} X^T YS^T [SS^T]^{-1}$$

We can now alternatively optimize these two parameters keeping the other one fixed using these equations,

$$S = [B^T X^T XB]^{-1} B^T X^T Y$$

$$B = [X^T X]^{-1} X^T YS^T [SS^T]^{-1}$$

This concludes the answer for this question.

Introduction to ML (CS771), Autumn 2018
Indian Institute of Technology Kanpur
Homework Assignment Number 1

Student Name: Abhishek Kumar
Roll Number: 18111002
Date: August 21, 2018

QUESTION 6

Prototype Classification for unseen classes

Method 1 - Convex Combination

In this method we assume unseen means to be convex combination of means of seen data.

$$\mu_c = \sum_{k=1}^{N} s_{c,k}\mu_k$$

where, μ_c is mean of unseen class c and μ_k is seen mean of class k.
The convex coefficients can be found using similarity followed by normalization between seen and unseen classes.

$$s_{c,k} = \frac{[a_c^T a_k]^p}{\sum_{l=1}^{N}[a_c^T a_l]^p}$$

here, a_c is unseen class attribute vector and a_k is seen class attribute vector, p is the power of this dot product.
Note that if $p = 1$ it is simple measure of similarity but if we increase $p > 1$ it acts as a regularizer (due to normalization) over linear combination model of seen means to calculate unseen means. It relatively tries to decrease the coefficients of seen means with less similarity value and increse coefficients of highly similar value.

Test set Accuracy obtained for $p = 1$
Accuracy of normal convex combination method is : 0.46893203883495144

Test set Accuracy obtained for $p > 1$
Accuracy of Convex combination with p = 2 is : 0.5334951456310679
Accuracy of Convex combination with p = 3 is : 0.5966019417475729
Accuracy of Convex combination with p = 4 is : 0.6461165048543689
Accuracy of Convex combination with p = 5 is : 0.686084142394822
Accuracy of Convex combination with p = 6 is : 0.7127831715210357
Accuracy of Convex combination with p = 7 is : 0.7242718446601941
Accuracy of Convex combination with p = 8 is : 0.7270226537216828
Accuracy of Convex combination with p = 8.1 is : 0.7273462783171522
Accuracy of Convex combination with p = 8.2 is : 0.7263754045307443
Accuracy of Convex combination with p = 8.5 is : 0.7258899676375404
Accuracy of Convex combination with p = 9 is : 0.722168284789644
Accuracy of Convex combination with p = 9.5 is : 0.7186084142394822
Accuracy of Convex combination with p = 10 is : 0.7155339805825243

Accuracy of Convex combination with p = 12 is : 0.6906148867313916
Accuracy of Convex combination with p = 20 is : 0.6153721682847897
Maximum achieved accuracy : 0.7273462783171522 at p value : 8.1

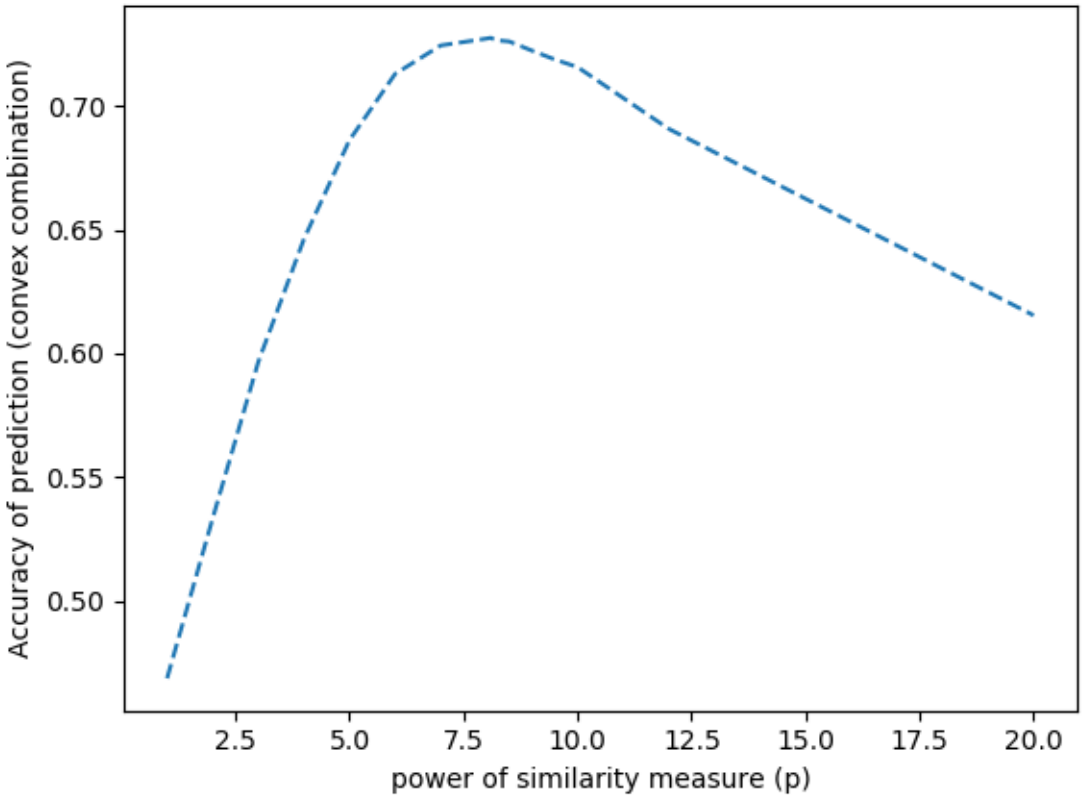

Figure 3: p vs accuracy

Method 2 - Linear Regression

This is a multi-output linear regression model with regularization in which we tried to model means of a class as a linear function of class attributes ($a \epsilon R^{85}$) given.

Test set Accuracy non regularized
Accuracy of non regularized model is : 0.5783171521035598

Test set Accuracy of Regularized model
Accuracy of model with lambda 0.01 is : 0.580906148867314
Accuracy of model with lambda 0.1 is : 0.5954692556634305
Accuracy of model with lambda 1 is : 0.6739482200647249
Accuracy of model with lambda 4 is : 0.7352750809061489
Accuracy of model with lambda 4.5 is : 0.7355987055016181
Accuracy of model with lambda 5 is : 0.7367313915857605
Accuracy of model with lambda 5.5 is : 0.7370550161812298

Accuracy of model with lambda 6 is : 0.7367313915857605
Accuracy of model with lambda 10 is : 0.7328478964401295
Accuracy of model with lambda 20 is : 0.7168284789644013
Accuracy of model with lambda 50 is : 0.6508090614886731
Accuracy of model with lambda 100 is : 0.5647249190938511

Maximum achieved accuracy : 0.7370550161812298 at lambda value : 5.5

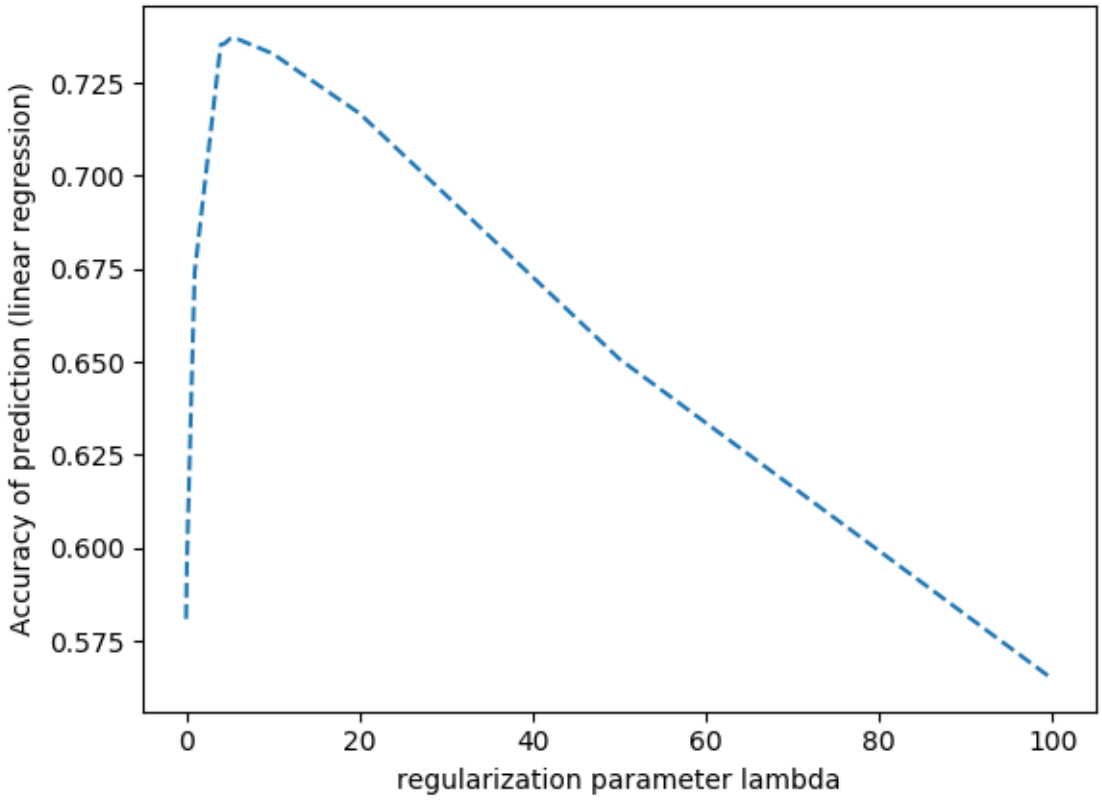

Figure 4: λ vs accuracy

Conclusion

- The convex combination with $p = 1$ gives accuracy about 46.8%
- The convex combination with $p = 8.1$ gives accuracy about 72.73%
- The convex combination with $\lambda = 0$ gives accuracy about 57.83%
- The convex combination with $\lambda = 5.5$ gives accuracy about 73.70%

This concludes the answer for this question you can see code in sent zip file.