Presentation Scope: Non-Linear Regression

Contributors: Anthony Pagan, Betsy Rosalen, Zach Herold

* Linear vs Non-Linear Regression)
  + Linear Regression (Betsy)
    - High level Description
    - Pros and Cons
  + Non-Linear Regression (Betsy)
    - High Level Description
    - Pros and Cons
* Non-Linear Regression Models
  + Neural Networks (Anthony)
    - Description

Ref: <https://www.youtube.com/watch?v=oYbVFhK_olY>

* + - * Modeled by Neural network of the brain
        + Input-Dendrite (Artificial NN Input values)
        + Nucleus – (Inputs with unique weights and summed together and passed threshold in 1 to many hidden layers (neural network 1 layer, deep) neural network >1 layers)
        + Output – Axon and terminal (Based on about becomes a 0 or 1 with Sigmoid function)
        + Passthrough Synapse to next Dendrite (passed to next neuron)
      * Intermediate Observed or Hidden Variables
        + Linear combinations transformed by sigmoidal function

Sigmoidal functions

Transformed by a nonlinear function g()



* + - * For P predictors there are H(P+1)+H+1 parameters
        + Fix

Pre-filter the predictors

Use PCA prior to modeling to eliminate correlation

Increasing amount of weight decay

* + - Computing examples

Ref: <https://www.youtube.com/watch?v=-Vs9Vae2KI0>

* + - * Normalize all variables to 0 and 1
        + For int (var-min(var)/max(var)-min(var))
      * Partition data :
        + set.seed
        + ind<-sample(2, nrow(data), replace = TRUE, prob=c(.7, .3)
        + Training<-data[ind==1,]
        + Testing<-data[ind==2,]
      * Create Model
        + Library(neuralnet)
        + Nn<-Neuralnet(y~., data=ind, hidden = 1, err.fct=”ce”, linear.output=FALSE)

hidden is number of nodes/neuron in a layer, c(2,1) would be 2layers with 2 nodes/neurons and 1 nodes/neurons

Can add lifesign= ‘full” to get all data points and rep = number of repetitio times to run model. When plotting with Rep can use plot(n, num) to show one with min error. Same for classification line.

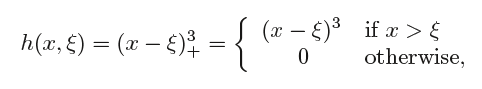
Can add algorithm = one to use

* + - * + Plot(Nn)
      * Prediction
        + output<-compute(n, training[,-1])
      * Calculation
        + iv<-input1value + gre\*weight+gpa\*weight+rank\*weight
        + ov<-1/(1+exp(-iv))
        + -iv2<-input2value+ov\*weight
        + Ov21/(1+exp(-iv2))
      * Confuion Matrix
        + Output<-computer(n, **training**[,-1])
        + P1<- output$net.resut
        + Pred1<-ifelse(pq>.5, 1,0)
        + Tab1<- table(pred1, training$admin)
        + 1-sum(diag(tab1))/sum(tab1) #missclassifications Output<-computer(n, **test**[,-1])
        + P2<- output$net.resut
        + Pred2<-ifelse(pq>.5, 1,0)
        + Tab2<- table(pred2, training$admin)
        + 1-sum(diag(tab2))/sum(tab2) #missclassifications
    - Other NN Models
      * Bayesian Framework
        + Regularization and automatic feature selection
      * Self-Organizing maps
        + Unsupervised, explanatory or supervised prediction
    - Pros
      * Robust with noisy data
    - Cons
      * Less interpretable
      * Need longer training times
      * Neural Networks have a tendency to over-fit the relationship between predictor and response due the large coefficients
      * Fix
        + Early stopping
        + Weight Decay with regularization with lambda values 0-.1
  + Multivariate Adaptive Regression Splines (Anthony)
    - Description
      * Creates 2 contrasted versions to of a predictor
      * 1 or 2 predictors at a time
      * Breaks predictors to 2 groups and models between
      * Hockey-stick(hinges)
        + Left-hand – values > 0 than cut point
        + Right-hand - values < 0 than cut point
        + Piece-wise linear model isolated portion of original data
        + Predictor/cut-point with smallest error
        + *X<a, h*(*x − a*) and *h*(*a − x*)
      * Turning used to remove parameters
        + The degree of the features that are added to the model and the number of retained terms.
        + The latter parameter can be automatically determined using the default pruning procedure (using GCV), set by the user or determined using an external resampling technique.
      * Types

Ref: <https://www.youtube.com/watch?v=UDDXkffB-aE&t=329s>

* + - * + Linear Splines- continuous at the knot

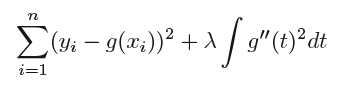


* + - * + Cubic Splines- continuous at the knot. Same as linear splines instead of power of 0 its 3.
        + Smoothing Splines

Splines without knots

Use smooth.splines() function in R

Find the function *g* that minimizes where *λ* is a nonnegative *tuning parameter*.



* + - * Advantages
        + Model automatically conducts feature selection
        + Interpretability, each hinge feature is responsible for modeling a specific region in the predictor space using piecewise linear model.
        + MARS require very little pre-processing, transformation and filtering not needed.
    - Computing examples
  + Support Vector Machines (SVM) (Zach)
    - Description
    - Computing examples
  + K-Nearest Neighbors (SVM) (Zach)
    - Description
    - Computing examples
* Conclusion and any new insights into the future analytics (SVM) (?)