Alex Wolf
5/9/17

ML Classifier Model Notes

Naive Bayes Overview

* Assumes Feature Independence
  * ***HOW To CHECK?
* Commonly used for discrete Features
* Runs Fast
* Can Update with new Data Easily
* Strong Bias, Low Variance

Gaussian NB

* Assumes a normal distribution of data
* Non-Multinomial Features
* Good for Small amount of Data
* Strong Bias

Multinomial NB

* Multinomial
  * Good with text data for large documents
  * Data is count of that sub_categorical
* Bernoulli
  * Good for Categorical Data and Small Text Documents (Don't words)
  * Binary Discrete Data

Kmeans

* Can test if work by using data cluster metric called Gap
* Good if noise in data
* Good with text Data
* Bad if data doesn't seem cluster
* Works well if enough Data is Available
* Non-Linear decision boundary
* Stores a lot of parameters
* Use miniBacthClassifier if >100k data points
* Costly computation of distances

RandomForest

* Robust to outliers and noise
* Not pessary to apply Data transformation
* Amount of Data reliant on Complexity of problem
* Feature Selection Not that important

SVM

* overview
  * Work well if have enough data
  * Good if very large data set
  * Not good for imbalanced dataset
  *Doesn't work well on skewed Data

LinearSVM

  * Great if data contains a clear margin
  * Bad if more features that examples\
  * Optimization much faster than kernel use

  * Parameters to tune
    * Penalty
      * 'l2' or 'l1' or elasticNet
      * Try l2 first, then elastic
    * L1 Ratio
      * Make Default Range .15 to .7
    * 'loss'
      * 'hinge'
        * Default- linearModel
      * 'log'
        * Assigns probability for classification instead of making a decision
        * Try for imbalanced data
      * 'modified_huber'
        * Tolerance to outliers
      * 'squared_hinge'
        * Quadratically penalized hinge
    * class_weight
      * ONLY for unbalanced Data
      * Add with dictionary of class labels

NoNLinearSVM

* Use if couldn't get good features from LinearSVM
* Slower optimization

* Parameters
  * kernel
    * 'rbf', 'poly', 'sigmoid'
  * C
    * Soft margin cost
    * Higher C means Higher soft margin
  * gamma
    * Higher leads to more bias
    *id high means noise in data
  * class_weight
    * Set for unbalanced data

My Classifier Function Parameters Methods

Naive Bayes

performGaussianNB(X, y, folds=10, impStrategy= 'mean', preprocess=MaxAbsScaler(), priors=None)

- -performs a Guassian NB
- -Data should have a Normal Distribution
- -can preSet Class priors

performMultiNomNB(X, y, binaryData=False, folds=5, impStrategy= 'mean', preprocess=MaxAbsScaler(), aLow=0, aHigh=1, numAlphas=5, fit_prior=False, class_prior=None)

- -performs a MultiNominal NB
- -if Binary Data set binaryData to False
- -Alpha is smoothness paramater tested in crossfold search
  - 0 means no smoothness applied

KNN

performKNN(X, y, impStrategy= 'mean', preprocess=StandardScaler(), folds=5, nLow=1, nHigh=5, nIter=1):

- -Knn which tests different amount of neighbor criterion per classifier

Forest

performRandomForest(X, y, folds=5, impStrategy= 'mean', class_weight=None, preprocess=StandardScaler(),
treeNumLow=10, treeNumhigh=11, treeNumLowIter=1)

- - Tests different amount of trees per classifier

Linear- SVM

performLinearSVM(X, y, impStrategy= 'mean', preprocess=StandardScaler(), folds=5, penalty='l2', loss='hinge,
aLow = 0.0001, aHigh=.1, numAlphas=10, class_weight=None, l1RatLow=0, l1RatHigh=.5, numL1Ratios=10)
-test different hyparamters L1Ration, and Alphas through Hold out validation, with grid search
-penlatly 'l2', 'l1', or 'elasticnet'
-l2 and net lead to sparse data
-Use when LOOK INTO MORE
loss
- 'hinge'- deafult
- 'log'- logisitic regression
- ‘modified_huber'- good if outliers
- ‘squared_hinge’- quadratcillay penalized

class_weight
-set if class imbalance

L1Ration
- 0 to 1
-percent of L1 versus l2 in model
-high l1 means nonImportant features in Data

Alpha
-regualarization term

NonLinear SVM

performNoNLinearSVM(X, y, kernel='rbf', impStrategy= 'mean', preprocess=StandardScaler(), folds=5, cLow= 1, cHigh=10,
numCs=10, gammaLow=0, gammaHigh=1, numGammas=10, class_weight=None)

- -test different hyparamters c, and gamma through hold out validation, with grid search
  - -kernel
    - -'rbf', 'sigmoid', 'poly'
  - -C
    - -soft margin param
    - -Larger C == bigger soft Margin
    - - Use K cross validation to find
  - -gamma
    - -larger gamma leads to high bias low variance
  - -class_weight
    - -set if class imbalance

Model Selection Steps for a Good Classifier

1. Try Naive Bayes as very quick results and training
    a. Use Multinomial, Guasssian, and Bernounlii respectively from Data
    b. If didn't work means Independence assumption too strong or too much Noise
2. Check for Noise/Outliers in Data
    a. Try RandomForest
    b. Can try Kmeans
3. If Skewed Data
    a. Try Logistic Regression
4. Still didn't work
    a. SVM- not best for skewed/ Imbalanced Data
        i. Linear
            1. 'l2' then 'elasticnet' penalty
            2. Use 'modified_huber' loss func if a lot of outliers
            3. Adjust Class Weights for imbalanced Data
    b. Try NonLinear SVm
        i. Change Class weights for unbalanced
        ii. Try 'poly' and 'sigmoid kernel'