Chapter 7

How to select K in kNN (1Q)

K= 1: By validation data; whichever k gives the lowest validation error.

Binary Classification K With Even K's

DO NOT USE even numbers since it could lead to an tie. XLMiner will pick the lowest probability and can chose an even number but that doesn't mean it should be chosen

K >1: classify by the majority decision rule based on the nearest k records

Low K values:

Capture local structure but may also capture noise. You cant rely on one Neighbour

High K values:

Provide more smoothing but may lose local detail. K can be as large as the training sample

Chose the K that gives you the lowest valid ER

Euclidean Distance (1Q)

sometimes predictors need to be standardized to equalize scales before computing distances. Standarized = normalized (-3, 3)

# of possible partitions in Recursive Partition (2Q)

Continuous: (n-1)*p

Categorical: 1ID- _P, 2ID - _P, 3ID -_P, 4ID - _P, 5ID - 15P

Cut Off Value in Classification

- Cutoff = 0.5 by default becuase the proportion of observation neighbors 1's in the k nearest neighbors. Majority decision rule is related to the cut off value for classifying records

Cut Off Value in Classification (cont)

You can adjust the cut off value to improve accuracy

Y = 1 (if p> cutoff)

Y = 0 (if p < cutoff)

Cut Off Example Question

Example: Suppose cutoff = 0.9, k=7, we observed 5 C1 and 2C0. Y = 1 or 0?

- Probability (Y=1) = 5/7 = 0.71 ---> 0.71 < 0.9 -> Y= 0

Advantages and Disadvantages

Simple and intuitive	Curse of Dimensionality (req size and many predictors
No assumptions required about data -> always correct	# of predictors x 1000 x 100? 50 predictors = need 5 mil observations
Effective with large training data	n/a

General Info

- Makes no assumptions about the data
- Gets classified as whatever the predominant class us among nearby records
- the way to find the k nearest neighbors in Knn is through the Euclidean distance

Rescaling: Only for kNN do you need to rescale because the amount of contribution from each variable. No need for logistic regression since it does not change the P value or RMSE

No need for CART since it doesn't change the order of values in a variable

General Info (cont)

XLMiner can only handle up to K= 10

Chapter 9

Properties of CART (3Q)

- Model Free
- Automatic variable selection
- Needs large sample size (bc its model free)
- Only gives horizontal or vertical splits
- Training error gets smaller and smaller with the tree size
- Validation error decreases and then increases with the tree size
- both methods of CART are BOTH model free

Trees

Best pruned tree: the tree whole validation error equal minimum error plus standard error; usually smaller than minimum error tree. You naturally get overfitting when the natural end of process is 100% purity in each leaf which ends up fitting noise in the data. Slightly overfitted so people partition a bit less to accommodate based of the minimal error tree

Minimum error tree: The tree with lowest validation error.

Full tree: largest tree training error equals zero; overfitted

Note: The full tree can be the same as the minimum error tree BUT usually best pruned tree should be smaller than the other trees

Recursive Partitioning

(1) Enumerate all possible partitions and select the one with the lowest impurity score

Impurity Score: Gini or Entropy Measure

(2) Partition following the first step is a subset partition of the same dataset -> Repeat choosing the lowest impurity score each time and drop

- Identify the midway point of the two lowest values of the output (14.0 & 14.8 -> split at 14.4)

- Repeat with the lowest purity being dropped and therefore compare values of 2nd and 3rd lowest (14.8 & 16.0) -> split at 15.4

(3) Continue the partitioning until ALL regions have either class 1 or class 0

- But must impose an early stop mark to prevent overfitting error since you can split it too much and lower training error to 0 but validation error will be very HIGH

algorithm decides where to partition

Impurity Score

- Metric to determine the homogeneity of the resulting subgroups of observations
- For both, the lower the better
- One has no advantage using one over the other.

Gini Index (0, 0.50 binary)

Entropy Measure: (0, $\log_2^2$ if binary) OR (0, $\log_2(m)$ -> m is the total # of classes of Y)

Overall Impurity Measure: Weighted average of impurity from individuals' rectangles weights being the proportion of cases in each rectangle.

Choose the split that reduces impurity the most (split points becomes nodes on the tree)

Check notes for that distance = to weighted average ratio

Dimensional Predictors Q's

With 21 observations, 2 dimensional continuous predictors, how many partitions can we have?

> # of partition = # of observation -1 -> 20 P

Continous Partitions

> (n-1) x P -> p dimentional predictors (more than 2 dimensional predictors)

Categorical Partitions

> abcd split. (3Levels, 3P), (4Levels, 7P)

XLMiner only supports binary categorical variables

When to Stop Partitioning

- Error rate as a function of the number of splits for training vs validation data -> Indicates overfitting

We can continue partitioning the tree so a **FULL tree** will be obtained in the end. A full tree is usually overfitted so we have to impose and EARLY STOP ...

- Stop when training error rate is approaching 0 as you partition further but you must have an early stop before letting it touch 0

Early - Stop (Minimum Error Tree or Best Prune tree):

OR

Stop based off **Chi-square tests**: (not commonly used for CART. They use min error tree or best prune tree)

- if the improvement of the additional split is **statistically significant** -> continue. If not, STOP.

Largest to Smallest: Full Tree > Min error tree > Best prune tree (Std usually smaller than min error). *Keep in mind: Full tree CAN BE THE SAME as your Min error tree*

Regression Tree

- Used with **continuous** outcome variables. Many splits attempted, chose the one that minimizes impurity
- Prediction is computed as the **average** of numerical target variables in the rectangle
- **Impurity measured by the sum of squared deviation from leaf mean**
- Performance measured by RMSE

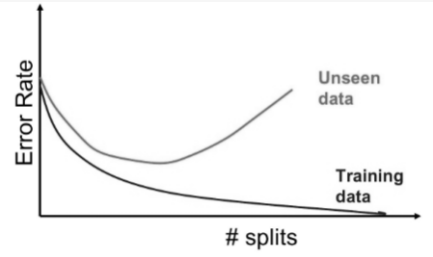
Regression Tree is used for prediction. Compared to classification tree, we only have to ...

Replace impurity measure by the sum of squared deviation everything else will be the same.

Split by irrelevant variables = Bad impurity score

Only split with relevant variables

Error Rate as you continue Splitting



Performance Evaluation

(1)Partition the data into Training and Validation Sets

Training set Used to grow tree

Validation set used to assess classification performance

(2) More than 2 classes (M >2)

Same structure except that the terminal nodes would take one of the m-class labels

Chapter 10

Assumptions For Logistic Regression (1Q)

- Generalized linearity

Logistic Regression Equation(2Q)

NOT model free -> based on following equations

$log\ odds = beta_0 + beta_1X_1 + ... + beta_qX_q$

$log\ p/(1-p) = beta_0 + beta_1X_1 + ... beta_qX_q$

$P = 1/(1+exp(-(beta_0 + beta_1X_1 +... + beta_qX_q)))$

Direct interpretation of beta 1 is that per unit increase of X1, log odds will increase by beta 1 -> not clear so thus you must say

The Log odds are going to increase by beta 1

- 3 equations equivalent to each other/
- All regression models can be called generalized linear modes
- Y= 0 in MLR is never true if Y is binary and thus cannot use this mode

Since Y is continuous, change Y into P (probability) and it eliminates the error term since you add some randomness

The Odds

Odds of ration is the exponential form of beta

- Beta is your coefficient number on your regression model

Comparing 2 Models

First criteria, pick the model with the lowest validation error

Second criterion, when the validation errors are comparable, pick the one with few variables

E.g suppose models 1 and 2 have a validation errors 26.2% and 26.3%. Their model sizes are

Comparing 2 Models (cont)

10 and, respectively. Which model is better?

- Initially go based of lowest validation error but when its too similar (23% and 26% -> its comparable) and thus you go based off of LOWEST model Size