Data Analysis

fwang2@ornl.gov

Contents

The reference material for this write-up are:

1. *Make Sense of Data*, by Glenn Myatt, 2007.
2. *Data Analysis and Graphics Using R*, by Maindonald and Braun, 2nd Edition, 2007.

1 Data Preparation

This is the number one task right there with understanding the data you are about to analyze. There are a few things you need to know about data transformation:

1.1 Descriptor and Response

Descriptors refer to the variables that are used as inputs to a model. They are also referred to as X variables.

Response variables are predicted from a predictive model using descriptor as input. These variables will be used to guide the creation of the predictive model.

1.2 Normalization

Normalization is a process where numeric columns are transformed using a mathematical function to a new range. It is important so that analysis of data treat all variables equally so that one column does not have more influence over another because the range are different. The following are common normalization methods:

- **Min-max**: suppose you want to transform data into a range of $NewMin$ and $NewMax$, the following methods works:

$$NewValue = \frac{Value - OrginalMin}{OriginalMax - OriginalMin}(NewMax - NewMin) + NewMin \tag{1}$$

  If $NewMin = 0$ and $NewMax = 1$, then you can have a simplified form of the formula.

- z**-score**: It normalizes the values around the *mean* of the set, with differnces from the mean being recorded as standard units on the basis of frequency distribution of the variable:

$$NewValue = \frac{Value - \bar{x}}{s} \tag{2}$$

  Where $\bar{x}$ is the mean and s is the standard deviation.

- **Decimal sampling**: This transformation moves the decimal to ensure the range is -1 and 1:

$$NewValue = \frac{Value}{10^n} \tag{3}$$

  Where n is the number of digits of the maximum absolute value. For example, if the largest number is 5989, then n would be 4.

1.3 Transform a Variable to Normal

Certain data analysis technique may require a variable to be *normal distribution*. You can try a few methods to transform it to be normal: taking **Log**, **Exponential**, or **Box-Cox** method. The later one is defined as:

$$NewValue = \frac{Value^{\lambda} - 1}{\lambda} \tag{4}$$

1.4 Value Mapping, Discretization, Aggregation

Value mapping refers to the case where you have an ordinal variable in text format (e.g., low, medium, high value), but you need it to be numerical, you can map it to say 0, 1, 2.

Discretization refers to the case where you have continuous variable but the collection method may not warrant such precision scale so you *discretize* it. For example, credit score can be discretized as poor, average, good, excellent. This technique also known as **data smoothing**. Another similar type of conversion is known as **binning**: for example, you can *bin* car's weight to less than 1000, between 1000-2000 lb, and more than 2000 lb. It is useful to study the frequency distribution before binning it.

Finally, aggregation refers to the case where a non-existent column is derived from one or more other columns. For example, a mathematical mean is a kind of aggregation.

2 Grouping Data

2.1 Similarity Measures

Any method of grouping needs to have an understanding for how similar observations are to each other. There are two common methods of computing distance.

- **Euclidean distance**: it handles continuous variable.

$$d = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \tag{5}$$

  It calculates the distance between two observations p and q, where each observation has n variables.

- **Jaccard distance**: it handles binary variables.

  - C_{11}: count of all variables that are 1 in observation p and 1 in observation q.
  - C_{10}: count of all variables that are 1 in observation p and 0 in observation q.
  - C_{01}: count of all variables that are 0 in observation p and 1 in observation q.
  - C_{00}: count of all variables that are 0 in observation p and 0 in observation q.

  The distance is defined as:

$$d = \frac{C_{10} + C_{01}}{C_{11} + C_{10} + C_{01}} \tag{6}$$

  An easy way to doing the count is: write two observation is tabular form, with each variable's value converted to 1 or 0. Then the value of C_{11} will be the number of pairs in the form of $(1,1)$. See *Make*

Sense, Vol 1, P108 for a concrete example.

There are many other distance measures I have no idea about: Mahalanobis, City Block, Minkowski, Cosine, Spearman, Hamming and Chebuchev. Do I need to check out this stuff?

2.2 Clustering

Supervised and Unsupervised

Distinction between the two are if they use response variable to guide how the grouping are generated: methods that do not use any response variable are called *unsupervised methods*, whereas methods that make use of response variable to guide group generation are called *supervised methods*.

When to Use?

Clustering is an unsupervised grouping method.

The advantage of clustering is (1) **flexible**: options to select similarity measures; options to select the size of the cluster etc. (2) you can do **hierarchical** clustering - where other method only generate clusters based on a pre-defined number.

The disadvantage of clustering is (1) **subjective**: different problems require different clustering options (2) **interpretation**: making sense of a particular cluster may require additional analysis. (3) **speed**: clustering large data set can be time-consuming.

Two methods are describe below: hierarchical agglomerative clustering and k-means clustering.

2.2.1 Hierarchical Agglomerative Clustering

It uses a *bottom-up* approach: each observation is a member of a separate cluster and progressively merge clusters together until all observations are a member of a final single cluster. The major limitation of this approach is its speed and size limit (less than 10K observations).

The clustering process is described below:

1. The distance between all combinations of observations is calculated. The two closest observations are identified and merged into a single cluster. These two will be treated as a single group from now on.

2. All observations (minus two observation just got merged) are compared against the newly formed cluster. Again, take the two closest observation or cluster and merge.

   When calculate distance between an observation and cluster, use linkage rule, as described below.

3. Repeat the steps until all is merged into one big cluster.

The linkage rules can be one of the three:

- **Average**: the distance between all members in the cluster and the observation are determined and its average is used as the distance measure.

- **Single**: the distance between all members in the cluster and the observation are determined and the smallest is used as the distance measure.
- **Complete**: the distance between all members in the cluster and the observation are determined and the largest is used as the distance measure.

The nice thing about this process is, at the end of it, you have a clustering scheme ranging from 1 to n, the number of observations. You can select an *optimal* cut-off point that decides number of clusters for the best way of grouping.

2.2.2 K-means Clustering

K-means clustering is a non-hierarchical method for grouping data. In contrast to hierarchical clustering method discussed above, this is a *top-down* approach: the number of clusters is pre-determined, and you assign observations to them. The advantage of this approach is it is computationally faster and can handle large data set. The disadvantage of it is (1) you need to pre-determine the number of clusters; (2) when data set has many outlier, the grouping can be distorted and not optimal.

The grouping process is as follows:

1. Allocate an observation to each pre-defined clusters or groups, usually randomly.
2. All other observations are compared to each of these allocated observations and placed in the group they are most similar to. Then the center point for each of these group is calculated.
3. For all observations, determine its distance to the center of each group. If an observation is closer to the center of another group, move it there. Then, the center of the two groups are re-calculated.
4. Repeat last step s until there is no further need to move any observations.

We use an example to illustrate how cluster center is determined:

```
Cluster 1 (3 observations)

Name      Var1    Var2    Var3    Var4
-----------------------------------------
C            8       9       7       8
D            6       8       7       8
E           10      12       3       4
-----------------------------------------
Center
Average      8    9.66    5.66    6.66
```

The similarity distance has been discussed before, use any appropriate ones.

2.3 Association Rules

The association rule is an example of *unsupervised* grouping method. The goal is to understand links and associations between attributes of the group. The advantage of this method is (1) Easy to interpret: the results are presented in the form of a rule (2) Actionable (3) Handle large data set.

The disadvantages of the method are (1) handle only categorical variables (2) time-consuming (3) Many rules

can be generated,but you have to prioritize and interpret.

The association rules are generated in two phases:

- Grouping (by value combination)
- Extracting Rules from Groups
- Prioritize rules by *support, confidence* and *lift*.

2.3.1 Support

An example: say a group has observations with following character: **Color** is gray, **Border** is thick, **Shape** is circle. Now you can deduce three rules:

Rule 1: IF color=gray AND shape=circle THEN border=thick
Rule 2: IF border=thick AND color=gray THEN shape=circle
Rule 3: IF border=think AND shape=circle THEN color=gray

Support is often defined as a proportion or percentage. Say your data set has 26 observations and the group we described earlier (gray circle with thick border) has 6 observations, then the group has a support value of $6/26 = 0.23$.

2.3.2 Confidence

Each rule is divided into two parts. The IF part or *antecedent* refers to the list of statements linked with AND. The THEN-part of the rule or *consequence* refers to any statements after the THEN.

The confidence score is a measure for how predictable a rule is. It is calculated using the support for the entire group divided by the support for all observations satisfied by the IF-part of the rule:

$$Confidence = \frac{\text{group support}}{\text{IF-part support}} \tag{7}$$

2.3.3 Lift

The confidence value does not indicate the strength of the association between IF-part and THEN-part. The lift score takes this into consideration, often described as the importance of the rule. It is calculated by dividing the confidence value by the THEN-part support.

$$Lift = \frac{\text{confidence}}{\text{THEN-part support}} \tag{8}$$

Lift values greater than 1 indicate a positive association, less than 1 indicate a negative association.

In summary, a rule is considered more interesting if both confidence score and positive lift score are high or higher than the rest of the rules from the group.

2.4 Decision Tree

In contrast with clustering and association rules, decsion trees are an example of a supervised method. The general steps of building a decision tree are:

- Determine **descriptors** and **response variable**. The descriptors are used for setting splitting criteria and response variable is the variable of interest.
- The root node has all observations to start with. Each splitting step will divide the observations into subsets. A splitting criterion has two components: (1) the variable (one of the descriptors you decided early on) to split on (2) values of the variable to split on.
- To determine the best split, all possible splits of all variables must be considered and ranked. One of the ranking methods for categorical response variable is to use **entropy**, a way of quantify *impurity*.

$$\text{Entropy}(S) = -\sum_{i=1}^{c} p_i \log_2 p_i$$

The entropy calculation is performed on a set of observations, S (parent node and child nodes). p_i refers to the fraction of observations that belong to a particular value: for example, 100 observations where response variable are weather, 40 are sunny, 20 are cloudy, and 40 are rainy for one way of splitting. Then the value c is 3 in this case, it is the number of different values that response variable can take. $p_1 = 0.4$ for sunny etc.

The best split consider before and after, represented by a **gain** measurement:

$$\text{Gain} = \text{Entropy(parent)} - \sum_{j=1}^{k} \frac{N(v_j)}{N} \text{Entropy}(v_j)$$

N is the number of observations in the parent node. k is the number of possible resulting nodes and $N(v_j)$ is the number of observations for each of the j child node. v_j is the set of observations for j child node.

- For continuous response variable, sum of square error (SSE) is often used in split criterion. The lower SSE is better split:

$$SSE = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

For each subset of observations, y_i is the individual value of the response variable, $\bar{y}$ is the mean for the subset.

- The process of ranking and splitting continue, until the termination criterion is met.

The advantages of decision tree are (1) easy to understand (2) deal with both categorical and continuous variable (3) represent complex relationships. The disadvantages of decision tree are (1) computationally expensive (2) difficult to optimize a complex tree.

3 Prediction

A predication model is some sort of mathematical process that takes the descriptor variables and calculate an estimate for one or more response variables. In other words, a model try to understand and capture the relationship between input descriptor variables and output response variables.

Rather than thinking any model generated as correct or not, it may be more appropriate to think of these models as useful or not to what you try to accomplish.

The predication model is often characterized by response variable:

- When response variable is categorical, the model is called **classification model**.
- When response variable is continuous, the model is called **regression model**.

3.1 Evaluating Classification Model

A simple way of doing this evaluation is to define:

- c_{00}: number of observations that were false and predicted as false. (true negatives)
- c_{11}: number of observations that were true and predicted as true. (true positives)
- c_{01}: number of observations that were true and predicted as false. (false positives)
- c_{10}: number of observations that were false and predicted as true. (true negatives)

These are the four measures are commonly used to assess the quality of a classification model:

- **Concordance**: Overall measure of the accuracy:

$$\text{Concordance} = \frac{c_{11} + c_{00}}{c_{00} + c_{01} + c_{10} + c_{11}}$$

- **Error Rate**: Overall measure of prediction error:

$$\text{Error Rate} = \frac{c_{10} + c_{01}}{c_{00} + c_{01} + c_{10} + c_{11}}$$

- **Sensitivity**: How well the model predicts the true:

$$\text{Sensitivity} = \frac{c_{11}}{c_{11} + c_{01}}$$

- **Specificity**: How well the model predicts the false:

$$\text{Specificity} = \frac{c_{00}}{c_{10} + c_{00}}$$

3.2 Evaluating Regression Model

It is typical to use r^2 to describe the quality of the relationship between actual response variable and predicted response variable, where r as correlation coefficient is defined as:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{(n-1)s_x s_y} \tag{9}$$

Another value to examine is the *residual value*, which is the difference between actual value (y) and predicted value ($\hat{y}$).

$$residual = y - \hat{y} \tag{10}$$

You can analyze residual values on a number of actors:

- **Response variable**: There should be no trend in residual values over the range of response variables, that is, the distribution should be uniform
- **Frequency distribution**: The frequency distribution of residual values should be normal
- **Observation order**: There should be no discernable trends based on when the observations were measured.

3.3 Simple Regression Model

Here, *simple* refers to one response variable (y), one descriptor variable (x): it can be linear or non-linear relationship. For linear relationship, you want to find a, the intersection point with y-axis; and b, the slope:

$$y = a + bx$$

The follow formula is so called **least square method**:

$$b = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{11}$$

$$a = \bar{y} - b\bar{x} \tag{12}$$

For non-linear regression, if the shape is regular upward, downward curve, then you can try a few transformations first on x, y or both and see if it can form a linear relationship. The common methods are:

- square root
- log
- $-1/x$

Suppose after the transformation, the relationship is deemed linear enough and you apply previously discussed least squares method and get predicted y'. You must then perform a inverse operation and get back y. For shape that is not so regular, or not so *simple*, then check the following methods.

3.4 *k*-Nearest Neighbor (kNN)

The kNN method calculates a predication by looking at similar observations (k of them) and use some function of their response variables to make the predication - in this sense, there is no mathematical model coming out of this process.

When a response variable is continuous, the predication is the mean of the k nearest neighbors; if categorical, then a voting scheme may be used to select the most common classification term.

Learning

There are three things to consider when applying kNN model:

- **Similarity Method**: this is used to determine how similar or close between two observations. For example, Euclidean or Jaccard distance. Prior to calculating similarity, it is important to normalize the variable to common range so that no variables are considered more important.
- **k value**: When k is too high, the kNN model will over-generalize; too small, kNN model is too sensitive and lead to large variation in predication.
- **Combination of descriptors**: to determine which combination can result in best predication.

To generate a proper k, we need (1) range of k to evaluate (2) one or more pair of training set and test set.

Predicting

New observation x comes in, we calculate distances between x and **every** observation in the training set and pick k closest neighbors. The mean of the response variables from the k neighbors is used as predication value.

3.5 Neural Networks

3.5.1 Topology

- **Input layer**: each node corresponds to a numeric input descriptor variable.
- **Hidden layer**: the number of hidden layers can normally range from 0 to 5. This number has an impact on prediction accuracy of the network.
- **Output layer**: each output node corresponds to an output response variable.
- **Weight**: each connection (from input nodes to hidden nodes to output nodes, usually fully connected) has a number associated with it. Prior to learning, it is assigned with a random number from -1 to 1.

3.5.2 Feed Forward

Given an input (an observation from the training set), each node in the neural network calculates a single output value, starting from first hidden layer all the way to the output layer. This is usually a **two-phase**

process for each node:

1. Assuming I_j is are the individual input values and w_j are the individual weights:

$$Input = \sum_{j=1}^{n} I_j w_j \tag{13}$$

2. Transform through **activation function**, there are two common choices:

$$\textbf{Sigmoid:} \qquad Output = \frac{1}{1+e^{-\text{input}}} \tag{14}$$

$$\textbf{Tanh:} \qquad Output = \frac{e^{\text{input}} - e^{-\text{input}}}{e^{\text{input}} + e^{-\text{input}}} \tag{15}$$

   These type of activation functions allow the neural network to develop non-linear models. The **Sigmoid** function produces output between 0 and +1, and **Tanh** function produces output between -1 and +1.

3.5.3 Learning through Backpropagation

Assuming all input/output values in the training set has been normalized prior to learning. Here is the rough process for learning: for each observation from training set, feed it to input layer, using feed forward formula to calculate the output, at the end of feed forwarding process, the output nodes will produce an output, which will be the initial predication value. Now you need to use this value and do so-called **backpropagation** to adjust those initially random assigned weight. The process will repeat itself for another observation and so on. The number of repeated training is known as **cycles** or **epoches**.

Now we describe how exactly the weight is updated:

- **Calculate Error:**: There are two kinds of error: one for output layer, the other for hidden layers. The first kind of error is calculated by:

$$Error_i = Output_i(1 - Output_i)(Actual_i - Output_i) \tag{16}$$

  Here, $Output_i$ is predicted response value, $Actual_i$ is the actual response value and $Error_i$ is the error resulting from node i. The second kind of error is calculated by:

$$Error_i = Output_i(1 - Output_i) \sum_{j=1}^{n} Error_j w_{ij} \tag{17}$$

  Here, $Error_i$ is the error resulting from hidden node, $Output_i$ is the value of output from the hidden node, $Error_J$ is the error already calculated from the jth node connected to the output and w_{ij} is the weight on this connection.

```
              w(ij)
          i ----------> j
(hidden node)       (either a output node or hidden node)
```

- **Update Weight**: w_{ij} is the weight of the connection between i and j, and $Error_j$ is the calculated error for node j and $Ouput_i$ is the computed output from node i.

$$w_{ij} = w_{ij} + L \times Error_j \times Out put_i \tag{18}$$

  L is known as learning rate, between 0 and 1. The smaller of L, the slower of the learning process. Often a learning rate is set high initially, then reduced as network fine-tunes its weight.

3.5.4 Considerations

When using neural network, there are following factors to consider: (1) how many hidden layers? fewer might be inadequate, more might be over training (prediction accuracy may decease) (2) how many cycles? (3) What should be the learning rate? (4) what combination of input descriptors to use?

Some of these parameters can only be determined with trail and errors - by examining r^2 between predicated values and actual values.

The **advantages** of neural network is (1) it can support linear and non-linear models (2) flexible input and output, can deal with both categorical and continuous variables (3) less sensitive to noise compare to statistical regression models.

The **disadvantages** of neural work is (1) Black box learning - you can't explain the results in a meaningful way; (2) Optimizing parameter (such as above) is largely trial and error, and there is always this over-training in play to avoid.