Count Regression Trees

The COUNT regression tree algorithm is comprised of following.

1. At each node in the regression tree, fit a Poisson, negative binomial, hurdle, and zero-inflated model to samples corresponding to the node. Select best model (**model selection**).
   Model selection usually involves evaluation of model performance using a suitable criterion or metric after fitting the models. Common criteria for model selection include goodness-of-fit measures (e.g., AIC, BIC), likelihood ratio tests, cross-validation, or other statistical tests. The best model would be the one that provides the best balance between goodness of fit and model complexity, effectively capturing the patterns in the data without overfitting. [ChatGPT]
2. Select the **split variable** (**feature selection**) ω_s.

   The aim here is to follow the GUIDE approach and select the split variable based on partial score residuals of the fitted model for each input feature (a.k.a. covariate). If a covariate is fitted well in the model, the sample correlation coefficient between its values and associated partial score residuals would be 0. A violation of this identity suggests that the covariate should be split. Thus, based on this proposition as well as the signs patterns of the partial score residuals of the fitted model, a test of independence ($\chi^2\ test$) between the covariate and corresponding partial score residuals can be conducted to reveal the best split variable. This process involves 2 procedures, being ***main effect detection*** and ***interaction effect detection***.

   Simply put, these procedures are as follows.

   Main Effect Detection [ChatGPT]

   - Fit the best count regression model to the sample data at the node.
   - Obtain the partial score residuals corresponding to both zero and non-zero values in the data.
   - Divide the residuals into two groups based on their signs (positive and non-positive).
   - For each non-categorical covariate (feature), divide the data into groups based on sample quartiles.
   - Build a contingency table with the covariate levels as columns and the signs of the partial score residuals as rows.
   - Remove entries with zero column totals and compute the Pearson chi-squared test statistic for independence.
   - For each categorical covariate, use the categories of the covariate to form the columns of the contingency table.
   - Omit entries with zero column totals and compute the chi-squared test statistic.

Interaction Effect Detection [ChatGPT]

- For each pair of ordered non-categorical covariates (ω_i, ω_j), divide the space into four quadrants by splitting each variable at its sample median.
- Formulate a contingency table with the quadrants as columns and the sign patterns of the partial score residuals as rows.
- Count the number in each cell and compute the chi-squared test statistic.
- For each pair of categorical covariates, use their categorical pairs to form the columns of the contingency table.
- Omit entries with zero column totals and compute the chi-squared test statistic.

Best split variable ω_s = If there exists a variable in interaction effect detection results such that the chi-squared test statistic exceeds a predetermined threshold, then return this variable. Else, return the variable associated with the largest chi-squared test statistic from main effect detection results.

The ***algorithm*** for split variable selection in more detail, as in the paper, is as follows.

2.1 Let

- K = no. of features.
- $\alpha_1 = \frac{0.05}{K}$
- $\alpha_2 = \frac{0.1}{K(K-1)}$
- $\chi^2_{1,\alpha} = 100(1-\alpha)^{th}$ percentile of the χ^2 distribution with 1 degree of freedom χ^2_1.

2.2 Obtain the main effect of each covariate $W_1(\omega_i)$ for $i = 1, \ldots, K$. Main effect detection is as follows.

- Fit the best count regression model to the sample.
- Get partial score residuals (corresponding to both zeros and non-zeros in case of the zero inflated or hurdle models).
- Divide each set of residuals into 2 groups as per their signs (positive and non-positive). This yields 2 (for Poisson and Negative Binomial models) or at most 4 possible patterns (for Zero Inflated or Hurdle models) for the residual sign vector.
- For each covariate (feature), divide data into 4 groups (levels) based on sample quartiles and if the model is Poisson or negative binomial, build a 2×4 contingency table; else (if the model is zero inflated or hurdle) construct a 4×4 one, both with levels as columns and signs of the partial score residuals as rows. Each cell in this table shall contain counts of samples that fall within that category.

- Remove entries with zero column totals and compute the Pearson chi-squared test statistic χ_v^2 of independence. If ν > 1, use Wilson-Hilferty approximation to convert χ_v^2 to χ_1^2 with 1 degree of freedom value $W_1(\omega)$.
- The same procedure is to be repeated for each categorical covariate, using the categories of the covariate to form the columns of the contingency table and omitting entries with zero column totals.

2.3 If $\max_i W_1(\omega_i) > \chi^2_{1,\alpha_1}$, then choose the variable associated with the largest value of $W_1(\omega_i)$ and exit.

2.4 Else, obtain the interaction effect $W_2(\omega_i, \omega_j)$ for each covariate pair. **Interaction effect detection** is as follows.

- For a pair of ordered covariates (ω_i, ω_j), where both covariates are x-covariate or z-covariate, divide the (ω_i, ω_j) space into four quadrants by splitting the range of each variable into two halves at the sample median. The result can then be formulated into a two-way 2×4 or 4×4 contingency table with the quadrants as columns and the sign patterns of the partial score residual vectors as rows; count the number in each cell.
- Do the same for each pair of categorical covariates, using their categorical pairs to form the columns of the contingency table and omitting entries with zero column totals.
- For each pair of covariates (ω_i, ω_j) where ω_i is ordered and ω_j is categorical, divide the ω_i space into two categories at the sample median such that their categorical pairs may be used to form the columns of the contingency table, Once again, entries with zero column totals are omitted.
- Compute its chi-squared test statistic and transform it to using the Wilson-Hilferty approximation to a χ_1^2 value $W2(\omega_i, \omega_j)$.

2.5 If $\max_{i \neq j} W_2(\omega_i, \omega_j) > \chi^2_{1,\alpha_2}$, if $W_1(\omega_i) > W_1(\omega_j)$, select ω_i. Else, select ω_j and exit.

2.6 Else, select the variable associated with the largest value of $W_i(\omega_i)$.

3. Select the **split set** (**split condition selection**). A split condition is the condition based on which we decide how the datapoints in each node can be divided into subsets that will form branches (e.g., all data points with selected feature value $\omega_s \geq 5.7$ goes into the right branch while others go into the left branch). The split set is determined by comparing all possible binary splits generated by ω_s on node deviance (impurity). Split conditions resulting in all zero responses in one group and all non-zero ones in the other group are ignored. The set which reduces the maximum amount of node impurity is selected (minimum entropy, maximum information gain). Following sub-steps comprise this step. [ChatGPT]

3.1. **Binary Split Generation:** For the selected split variable (ω_s), generate all possible binary splits that divide the data into two subsets.

3.2. **Split Set Evaluation:**

- For each binary split, evaluate its effectiveness in reducing node impurity. Node impurity can be measured using a suitable metric such as deviance, entropy, or Gini index.
- Ignore split sets where all zero responses are at one node and all non-zero responses are at the other node. This is because such splits do not effectively separate the data.

3.3. **Best Split Set Selection:** Choose the split set that maximally reduces the node impurity. This split set effectively separates the data into two subsets that are as homogeneous (pure) as possible with respect to the target variable.

4. **Tree Building:** Using steps so far, the tree is split recursively until a stopping criterion (e.g., all covariates or response values are the same, no. of cases in it is $< 0.05 \times$ total no. of cases, etc.) is reached.
5. **Tree pruning** is carried out using the cost complexity pruning algorithm with 10-fold cross validation.
6. **Prediction accuracy** of the resulting tree is accessed **(model evaluation)** using 10-fold cross validation and average prediction Mean Absolute Error (MAE) or Mean Squared Error (MSE).