An Evaluation of Accuracy, Overdispersion and Zero-Inflation in Count Regression Trees

Gayathri Girish Nair, Kim Nolle, Yongxun Qiao, Yuanpei Teng, Yuqing Zhang, Xiaoyao Zhu, Xiangying Peng and Shaomeng Ji
Trinity College Dublin, Dublin, Ireland
{girishng, nollek, yqiao, yteng, zhangy46, zhux7, pengxi, jis1}@tcd.ie

***Abstract*—In disciplines like economics, health services, and social sciences, processing count data—which consists of non-negative integer values listing occurrences or items—pose substantial obstacles. Because of problems like overdispersion and zero-inflation, traditional count regression models like Poisson and negative binomial frequently perform poorly. To overcome these statistical difficulties with count data, the CORE (COunt REgression Trees) model integrates decision trees with count regression models. This work evaluates the prediction accuracy of CORE as well as its robustness to zero-inflation and overdispersion. The advantages of CORE in managing complicated count data distributions are shown by comparing the results with baseline models and with the MOB (Model-Based recursive partitioning) technique. Our findings reveal that CORE, when applied to a real-world dataset, does not lead to similar predictive accuracy found in other works. We find lesser accuracy and poorer ability to handle zero inflated data and overdispersion than expected. Instead, both CORE and MOB were found to significantly overfit the data which is speculated to be due to a magnification effect curtsy of repeated sub-model fitting at each node. Further through this work, we have also contributed, to the best of our knowledge, the only publicly available runnable implementation of CORE in R.**

***Keywords*—regression tree analysis, CORE, MOB, count data, negative binomial model, hurdle model, zero-inflated model.**

I. Introduction

Count data consists of observations that enumerate events or items, represented by non-negative integer values ranging from 0 to infinity (although practically bounded) [1]. Such data is prevalent in sectors like economics, health services, and the social sciences, to name a few. But, effective modelling of count data can be challenging. For instance, the distribution of such data is often not normal and hence may fail to satisfy the assumption of normality that common linear regression models make. Further, its discrete nature can render some traditional models inadequate. Models like Poisson and negative binomial regression can accommodate characteristics of count data and hence are popular choices w.r.t modelling this type of data. That said, they too may not be ideal in every case, given added challenges, such as overdispersion and zero inflation.

Over-dispersion in count data refers to a situation where the variability (variance) of data exceeds what is expected based on a standard statistical model, like the Poisson distribution [1]. This phenomenon can arise due to heterogeneity of observations, where explanatory variables exhibit low variance while the response variable exhibits high variance. More specifically, a Poisson distribution assumes that the mean and the variance of the data are equal. Data demonstrating overdispersion violates this assumption, resulting in biased parameter estimates and standard errors. This in turn can affect the accuracy and reliability of statistical inference. Meanwhile, zero-inflation in count data arises when observed data contains more zero values than what typical count models, like the Poisson or negative binomial models, would predict [2]. This can distort statistical analyses and lead to biased estimates if not properly addressed. In the real world, there are multiple scenarios wherein accounting for peculiarities like excess zeros, can be very important when modelling count data. For example, industrial gas leaks are rare, but potentially dangerous events. If a production plant that uses/produces harmful gases model the frequency of gas leaks in a residential area over a period of time using a classic model like Poisson that does not account for zero inflation, the large no. of zeros in the data set, could undermine estimated likelihood of dangerous gas leaks which could lead to an under allocation of resources and time towards maintenance or inspection which can have dire consequences. The infamous Bhopal gas tragedy in 1984 killed at least 20K people with at least 500K suffering long term health consequences [3]. Thus, risk assessment in such scenarios requiring companies, governments, etc, to use models that effectively account for characteristics of count data such as zero inflation is one example of the significance of apt modelling of count data.

Different methods have been developed to deal with over-dispersion and zero-inflation. For over-dispersed count data, it is recommended to use alternative methods such as the negative binomial regression model [4]. One common approach to address zero-inflation is to use zero-inflated or hurdle models. Other models like regression trees, which enable interpretability, are also able to handle zero-inflation and over-dispersion [5].

More recently, Liu et al. proposed CORE (Count Regression Trees) [6], which combine decision trees with count regression models to tackle analytical challenges of count data, particularly in addressing overdispersion and zero inflation. In their study, Liu et al. compare CORE to MOB (Model-Based recursive partitioning) [7], which is another state-of-the-art method that applies regression trees to count data. They demonstrated that CORE has more chance of selecting informative covariates than MOB, indicating that the CORE method performs better with regards to prediction accuracy. We aim to evaluate their study by not only considering predictive

accuracy of the algorithm but also the extent to which CORE is able to compensate for overdispersion and zero-inflation.

A. Research Statement

The main research goal of this work is to evaluate the performance of count regression trees. Our hypothesis is as follows:

> ***Hypothesis:*** The CORE variant regression trees is effective at predicting count data and is less affected by overdispersion and zero inflation. The CORE model when leveraged to predict the number of absences of students from school comprising an overdispersed count target variable with excess zeros, leads to lower prediction errors than simpler models (Poisson, negative binomial, zero-inflated Poisson, zero-inflated negative binomial, Poisson hurdle, negative binomial hurdle) and similar prediction error as the MOB variant of count regression trees.

To investigate this hypothesis, we fit the simpler models, MOB, and CORE to a dataset that displays overdispersion and zero-inflation in Poisson and negative binomial models. We then measure the Mean Squared Error (MSE) and Mean Absolute Error (MAE) to determine the predictive performance of the models. Additionally, we measure the dispersion as well as the number of zeros expected by the model, which allows us to evaluate zero-inflation. This is repeated by employing 10-fold cross validation, which allows us to obtain more generalizable results. If the CORE displays low MSE and MAE, low overdispersion and low zero-inflation, then this would confirm our hypothesis and demonstrate that it is a suitable approach to modeling overdispersed and zero-inflated count data.

This paper is organized as follows. Section II presents background information for this study. Section III outlines the methodology followed including a brief explanation of the CORE count regression tree algorithm implemented. Section VI presents findings. Finally, Section V discusses the reported metrics and Section VI concludes our work with key takeaways and further research ideas stated.

II. Background

This section gives an overview of different approaches to handling overdispersion and zero-inflation when modelling count data.

A. Models Handling Over-Dispersion in Count Data

One effective way to solve over-dispersion is to adopt a negative binomial distribution. When variance of data is greater than the mean, the negative binomial distribution with more flexible constraints than the Poisson distribution, accommodates this. Due to being able to capture variability of overdispersed count data better, the negative binomial distribution is often associated with more generalizable and accurate models [4].

The negative binomial distribution [4] is a statistically discrete probability distribution that describes the number of failures in a series of independently and identically distributed Bernoulli trials when the number of successes reaches a specified number.

More specifically, two parameters are included in the negative binomial distribution, which are the probability of success and the number of successes, making the negative binomial distribution more flexible than the Poisson distribution as it introduces additional parameters to better characterise the data [4]. Thus, the negative binomial distribution can be viewed as an extension of the Poisson model that better handles diversity in data.

B. Models Handling Zero-Inflation in Count Data

Zero-Inflated Models: Zero-inflated models are particularly useful when the data contains a large number of zero counts that exceed the number predicted by typical counting models, such as the Poisson or negative binomial distribution. These models are hybrid models of binary (Bernoulli) distributions and count distributions, designed to handle excess zeros through two different processes: one that generates zeros (the zero-inflated component) and another that generates them according to a specified distribution The process of counting (counting component) [8] .

The zero-inflated component models the probability that an observation is an additional zero [9]. This component is a logistic regression that predicts the log odds of an outcome of zero due to an additional zero-generating process rather than a counting process. Let Z be a binary random variable, where Z=1 means that the zero result is generated by the zero-inflation process (rather than by the counting process), and Z=0 means the opposite. The probability of model Z=1 is as follows:

$$logit\big(P(Z=1)\big) = \log\left(\frac{P(Z=1)}{1-P(Z=1)}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k$$

where β_0, β_1 ..., β_k are parameters to be estimated, and X_1... X_k are covariates.

The counting component uses a counting distribution (Poisson or negative binomial distribution) to model the number of times an event occurs, provided that the event is generated by a counting process (i.e., Z=0). In a Zero-inflated Poisson (ZIP) model the counting component assumes that given Z=0, the counts Y follow a Poisson distribution:

$Y \mid (Z=0) \sim \text{Poisson}(\lambda)$

The parameter λ represents the average number of occurrences of an event, and typically the model is $\lambda = e^{X'\beta}$, where X includes covariates and β is the coefficient.

A zero-inflated negative binomial (ZINB) model assumes a negative binomial distribution. Here, the counting component model is:

$Y \mid (Z=0) \sim \text{NegativeBinomial}(\mu,\phi)$

The parameters μ and ϕ represent the mean and discrete parameters, respectively, and are usually related to the covariates through some linear combination [9].

Hurdle model: The hurdle model is a two-part model commonly used for count data where zeros occur more frequently than expected under standard count distributions like the Poisson or Negative Binomial. This model is called "hurdle"

because it explicitly models the probability of going from a zero count to a positive count [8]. The components and working process of the model are as follows:

Zero Hurdle Component (Binary Outcome Model)

This model component determines whether the result is zero or a positive number. It is typically modeled using a binary probability distribution, such as the Bernoulli distribution, with probability parameters estimated via logistic regression:

$P(Y=0 \mid X)=1-\pi(X)$

$P(Y>0 \mid X)= \pi(X)$

where: Y is the count variable. X represents the covariates or predictors. $\pi(X)$ is the probability of having a positive count, modeled as $\pi(X) = \frac{1}{1+e^{-X\beta}}$ with β being the parameters to be estimated.

Positive Count Component (Truncated Count Model)

Once the hurdle of zero is crossed, the count distribution is modeled using a truncated version of the Poisson or negative binomial distribution that excludes zero. Truncated distributions modify standard probability mass functions (PMFs) to adjust for the absence of zero outcomes:

Truncated Poisson Distribution:

$P(Y = y|Y > 0, X) = \frac{e^{-\lambda}\lambda^{y}}{(1-e^{-\lambda})y!}$

For y = 1, 2, 3, …

Truncated Negative Binomial Distribution:

$P(Y = y|Y > 0, X) = \frac{\Gamma(y+r)}{\Gamma(r)y!}(\frac{p}{1-p})^{y}(\frac{p}{1-p})^{-r}$

For y = 1, 2, 3, …

where: λ is the rate parameter of the Poisson model, or p and r are the probability and dispersion parameters of the Negative Binomial model, respectively. Γ denotes the gamma function, used in the Negative Binomial distribution.

Estimation and Inference:

The parameters of the two components are typically estimated using maximum likelihood estimation (MLE). The likelihood function across the model is the product of the likelihoods of the two components:

$$L(\theta; Y, X) = \Pi_{i:y_i=0}\big(1-\pi(x_i)\big)\Pi_{i:y_i>0}[\pi(x_i) * P(Y = y_i|Y > 0, X)]$$

where θ encapsulates all the model parameters [10].

Regression Trees: [11] A decision tree is one way to capture rules that govern patterns in data. When a decision tree is leveraged for a classification task, then it is called a classification tree. Likewise, when it is used for a regression task, it is called a regression tree. Irrespective of the kind of decision tree, building one involves recursive division of a data set into smaller subsets according to input feature values based on information gain associated with specific predictors and dataset split conditions such that each split reduces impurity or variance of predictions to ultimately result in lowest response variable variance at a point where the dataset can no longer be split either because its size is too small or because error is minimum.

Regression trees are very versatile. They don't make assumptions about distribution of values of predictors and can work with a mixture of categorical and continuous predictors. They also work better than many other models when the relationship between input variables and the target variable is complex and/or non-linear. [12] This type of models has also been found to handle zero-inflated data well, by identifying subgroups within the data where the proportion of zeros is significantly higher or lower [13]. One variant of decision tree is Chi-square Automatic Interaction Detector (CHAID) which uses a Chi-Square test to help determine best predictor and split condition that results in classification with least impurity. Impurity in case of classification may be measured using metrics like Gini impurity or entropy. For regression trees, like the one implemented in this project, goodness of prediction may be quantified by computing Mean Squared Error (MSE) or Mean Average Error (MAE) [14].

Traditional CART however, generally fits a simple constant model in each node. That is, after splitting data based on a feature at a node, a single constant value is usually assigned to all data points within that leaf node (a leaf node refers to a node that cannot be split further). For classification, this constant may be the majority class label of response variable values in the leaf and for regression, it is the mean of response values in that leaf. Because the model at each node is very simple, CART can still sometimes struggle to capture more intricate relationships between features and the target variable. [13]. Thus, Choi et al. [15] proposed a method which utilizes the idea of leveraging additional Poisson regression models in a recursive hierarchy, which works by using a fitted likelihood method at each node to accommodate additional variability. This, along with introduction of a new resampling method and adjusted Anscombe residuals, resulted in improvement in robustness and performance of the model which was also easy to interpret [15].

C. Count Regression Trees

Meanwhile, one of the Counting Regression Trees algorithms called MOB (model-based recursive partitioning) [7] surmounts limitations of the CART model. It can infer effects of different input factors on the response variable by associating each terminal node with a parameter model while still being based on the concept of recursive partitioning, typical of a decision tree. Here, the parametric model is first fitted on the full dataset, considerate of metrics like maximum likelihood. Then, the stability of coefficients is tested via a generalized M-fluctuation test, and the best splitting point is determined by evaluating the segmentation objective function. This process is eventually repeated recursively at each sub-node until model coefficients in each terminal node stabilize, allowing further observations to be made using the fitted parametric regression function in each terminal node [7]. Moreover, MOB method proposed by Zeileis et al. may also be used with models apt for count data (Poisson, Negative Binomial, Hurdle, Zero-Inflated) at each node. Thus, this method is likely to be better adaptable w.r.t count data, compared to classic CART [13].

To this end, Tang et al. [16] compared performance differences between the MOB algorithm and the negative binomial distribution algorithm in predicting road accident frequency. They conclude that the MOB algorithm, with a negative binomial model, provides unique insights into factors

influencing accident frequency under different covariate conditions. They discovered that covariates like speed limit signs, affect accident frequency differently under different subgroup categories, and that the MOB with negative binomial model had the highest prediction accuracy on test data, compared to the vanilla negative binomial regression model [16].

Liu et al. [6] propose an extension of MOB called COunt REgression Trees (CORE). While MOB only uses a single type of count regression model when building the tree, CORE fits the most appropriate model from 4 possible models (Poisson, negative binomial, hurdle, and zero-inflation) in each node. This allows CORE to be more flexible and overcome negative effects of overdispersion and zero-inflation as it enables adaptive model selection based on the features of the subset of data. Tree building in the CORE method, also includes pruning, to reduce overfitting and improve generalizability of the model.

III. Research Methodology

This section details our approach to answering our research question. This includes an overview of the dataset we will use to evaluate the performance of CORE, details of our implementation of the algorithm, and our approach to the evaluation.

A. Dataset

To evaluate the performance of Count Regression Trees, we analyzed a Student Performance dataset [17] containing data about how well some Portuguese secondary school students faired in the subjects of mathematics and the Portuguese language. The dataset includes demographic, social and school related attributes of students and consists of 1044 samples. The target variable of our analysis was the "absences" variable, which is the number of days students were absent from the respective course. Figure 1 shows a histogram of the target variable that gives an idea about the distribution of corresponding counts.

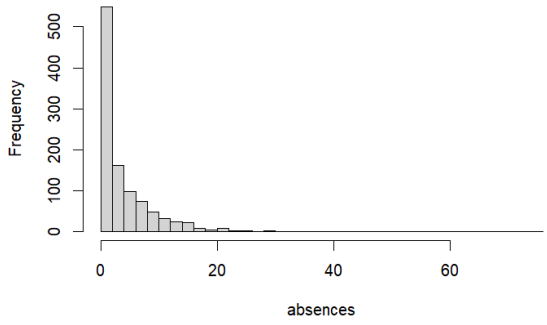

Figure 1: Histogram of target variable *absences*

We considered remaining variables in the dataset as predictors, except for those listed in Table 1. We do not use these because they are expected to have little influence on absence from school and will likely only serve to contribute towards the curse of dimensionality [18], if included. Moreover, most of these dropped features are likely highly correlated with one another (cohabitation status of parents (Pstatus) with family relations (famrel), whether the student is enrolled in post school paid tuition (paid) with study time (studytime), intermediate grades (G1, G2) with final grade G3) and hence can lead to violation of the assumption of independence that decision trees make.

Table 1: Variables in the dataset excluded from analysis (descriptions from [17])

Variable	Description
reason	Why the school was selected (categorical variable)
paid	Whether extra study classes were taken (binary variable)
Pstatus	Co-habitation status of the parents (binary variable)
G1	Student's grade at the end of the first period (continuous variable)
G2	Student's grade at the end of the second period (continuous variable)

To ensure that the dataset is suitable for analysis, we checked for, and confirmed over-dispersion and zero-inflation when fitting Poisson or negative binomial models. The result of this preliminary analysis is presented in Table 2.

We first fitted Poisson and negative binomial distributions to our target variable "absences" using Maximum Likelihood Estimates (MLE). Using the MLE of distribution parameters, we then calculated expected variance. For Poisson distributions, this corresponds to the parameter μ [19]. For the negative binomial distribution, the expected variance is $\mu(1 + \alpha\mu)$, where α is the dispersion parameter [19]. The results show that in both models, the variance in the data exceeds expected variance. This signals over-dispersion in the data.

Table 2: Results of the preliminary data analysis

Model	Variance in Data	Expected Variance	Observed Zeros	Predicted Zeros
Poisson	38.5643	4.4349	359	66
Negative Binomial		15.1141		293

Next, we fitted Poisson and negative binomial GLMs and used the R function `check_zeroinflation()` to check if resulting models were zero-inflated. The results show that in both cases, fewer zeros are predicted by the models than can be observed in the data. This shows that the data is zero-inflated. Therefore, the dataset is suitable for our research.

B. Implementation of CORE

Our R implementation of CORE follows the algorithm proposed by Liu et al. [6]. The following section details the steps in this algorithm and explains any choices we made. For reproducibility, the dataset and our code can be found in our GitHub repository[1].

[1] https://github.com/ggn1/cs7ds1-group-project

1) Model Selection

At each node in the regression tree, a count regression model is fitted to the samples in that node [6]. This model can either be a Poisson, negative binomial, hurdle or zero-inflated model.

Liu et al. [6] do not specify a method to select the model at each node. One approach might be to randomly select a model. However, in the next step, the residuals of this model are used to determine the variable that will be used to split the samples for the child nodes (see section *III-B-2 Split Variable Selection*). If the regression model is randomly selected, then patterns in the residuals might not necessarily be caused by patterns in the data but instead might be due to chance. This could result in a less optimal split of samples. We therefore decided to select the regression model that has the best fit, so that residuals are more significant.

10-fold cross-validation is used to determine which regression model best fits the data in the node. This measures not only how well the model fits but also how well it generalizes [20]. If the model generalizes well, then the residuals are not distorted by over-fitting, which again yields more significant residuals.

2) Split Variable Selection

After selecting the best model for the node samples, the split variable is selected [6]. To do this, the relationship between the predictors and prediction errors of the model (residuals) is analyzed. If a predictor is fitted well in the model, then there should not be any association between the predictor and the residuals. If there is an association, then this indicates that the predictor variable should be split. This process involves two procedures: main effect detection and interaction effect detection.

Main effect detection: A main effect is a relationship between two variables independent of other variables [21]. To detect the main effects between predictors and residuals, Liu et al. [6] propose calculating the association between predictors and residuals using Wilson-Hilferty approximation [22], which approximates the Pearson chi-squared test statistic while being less computationally complex. A test statistic $W_1(\omega_i)$ is calculated for each predictor ω_i and associated partial residuals. The columns of the contingency table used to calculate the test statistic correspond to quantiles of the predictor. In case of categorical predictors, each category is a column. The rows of the contingency table correspond to the sign of the partial residuals. If a zero-inflated or hurdle model is used, the partial residuals are not only split according to their sign but also according to whether they correspond to the count R_i^{β} or zero R_i^{γ} component, resulting in four rows (here, R_i^{β} is the residual associated with the probability of zeros and R_i^{γ} is the residual associated with the counts). Let χ^2_{1,α_1} be the $100(1-\alpha)$th quantile of the χ^2 distribution with 1 degree of freedom and let K be the number of distinct predictors. The critical value for the test statistic of the main effect detection is χ^2_{1,α_1}, where $\alpha_1 = \frac{0.05}{K}$.

Interaction effect detection: An interaction effect is a relationship between two variables that is dependent on other variables [21]. Again, Liu et al. [6] propose using Wilson-Hilferty approximation to detect interaction effects between pairs of predictors and the target count variable. This involves considering all unique pairings (ω_i, ω_j) of predictors and formulating a contingency matrix. Each non-categorical predictor in the pair is divided into two groups or categories by splitting its samples at the median. For categorical variables, the levels (classes) are the groups. These groups form the columns and the sign of the partial residuals of the pair together, form the rows of the contingency matrix which has shape 2×4 in case of the Poisson or negative binomial models and 4×4 in case of the zero inflated or hurdle models (after accounting for the sign of residuals of both count and zero parts). The resulting contingency table is then used to calculate the test statistic $W_2(\omega_i, \omega_j)$. The critical value for the test statistic of interaction effect detection is χ^2_{1,α_2}, where $\alpha_2 = \frac{0.1}{K(K-1)}$.

To select the split variable of a node, main effect detection is carried out for all predictors. If the predictor with the highest test statistic exceeds the critical value, then this predictor is used as the split variable. Otherwise, the interaction effect detection is performed. If the pair of predictors with the highest test statistic exceeds the critical value, then the predictor of that pair with the highest main effect test statistic is used as the split variable. Otherwise, if neither the maximum main effect test statistic nor the maximum interaction effect test statistic exceeds the critical value, then the predictor with the highest main effect test statistic is used as the split variable.

3) Split Set (Condition) Selection

After the split variable ω_s is determined, the samples in the node are split to determine the samples in the child nodes [6]. All possible binary splits that can be generated by ω_s as considered, excluding splits that separate zero and non-zero responses. The split that results in the greatest reduction in impurity is selected.

Lui et al. propose to use deviance to measure the impurity in a node. Deviance is a metric that describes how well an MLE fits the data, with a lower value indicating a better fit [21]. Liu et al. [6] define it based on the likelihood fitted by the count regression model from step *1) Model Selection.* The definition of the deviance in a node t can be seen in (1). Here, θ is the distribution parameters fitted by the count regression model and $\boldsymbol{y}_t$ and $\boldsymbol{x}_t$ are respectively the target and the predictor observations in the node.

$$I(t) = -\log\left\{\max_{\theta}\{\mathcal{L}(\theta|\boldsymbol{y}_t, \boldsymbol{x}_t)\}\right\} \quad (1)$$

To determine the best split, the deviance before the split $I(t)$ and the deviance after the split $I(t_{\xi L}) + I(t_{\xi R})$ is calculated, where $t_{\xi L}$ is the left child node resulting from the split ξ and $t_{\xi R}$ is the right child node. The split that maximizes the difference $\Delta I(t) = I(t) - [I(t_{\xi L}) + I(t_{\xi R})]$ is used to split the samples into the child nodes.

However, when implementing this, there were cases where non-singularity was present in the systems of linear equations, which were used to fit zero-inflated or hurdle models. To avoid this problem, we therefore opted to measure impurity by means

of MAE instead, which is a common approach when building decision trees [21]. The definition of MAE can be found in (2). Furthermore, we weigh the MAE of the child nodes according to the number of samples in each split to adjust for different sample sizes. The metric we use for selecting a split is therefore:

$$\Delta MAE(t) = MAE(t) - \left(\frac{n_L}{n_L+n_R} MAE(t_{\xi L}) + \frac{n_R}{n_L+n_R} MAE(t_{\xi R}) \right)$$

where n_L is the number of samples in the left child node and n_R is the number of samples in the right child node.

4) Building the Regression Tree

Starting at the root node containing all samples, the aforementioned steps *model selection*, *split variable selection*, and *split set selection* are recursively performed to build the regression tree until a stopping criterion is reached. Liu et al. [6] suggest using the following criteria to stop splitting a node:

- All target values are the same.
- The number of samples in the node is less than 10% of the total samples.

C. Evaluation of Regression Performance

Our study aims to evaluate the prediction accuracy of CORE as well as its capability to handle overdispersion and zero-inflated data. To do this, we compare it to baseline models, namely Poisson, negative binomial, hurdle, and zero-inflated regression models, which are able to compensate for overdispersion and zero-inflation to varying degrees. Furthermore, to compare our findings to the original study by Liu et al., we also compare our implementation of CORE to MOB. In order to not only evaluate the performance but also how well the models generalize, we use 10-fold cross validation.

We use the R packages `stats` and `MASS` to implement Poisson and negative binomial GLMs. The package `pscl` provides implementations for hurdle and zero-inflated models and the package `partykit` implements the MOB model. When fitting the models we use all predictor variables and no depth limitation when fitting MOB or CORE.

Similar to Liu et al. [6], we measure the accuracy of the models using MSE and MAE. The definitions of these can be seen in (2), where $\hat{y}_i$ is the predicted value of the sample y_i.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \qquad MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (2)$$

Overdispersion can be measured using the Pearson-based dispersion statistic [19]. This is defined as the sum of squared Pearson residuals divided by the residual degrees of freedom $(n-p)$, where n is the number of observations and p is the number of parameters. The definition can be seen in (3), where y_i is the observed value and μ_i is the expected value predicted by the model. A dispersion statistic > 1 indicates overdispersion and a statistic < 1 indicates under-dispersion.

$$\phi = \frac{1}{n-p}\sum_{i=1}^{n}\frac{(y_i-\mu_i)^2}{\mu_i} \quad (3)$$

To measure zero-inflation, we will compare how many zeros are observed in the predictions of each model to the number of zeros observed in the data.

IV. Results

This section presents the results of the evaluation of the models using 10-fold cross validation. Figure 2 shows the average MSE for the training and test data for the different models. The corresponding values can be found in Table 3. The baseline models (Poisson, negative binomial, zero-inflated and hurdle models) have similar training MSEs of approx. 25. The Poisson and negative binomial models both have a test MSE of around 37, while the zero-inflated and hurdle models have lower test MSEs around 28. MOB displays much smaller training MSEs but higher test MSEs. Our CORE implementation has lower training MSEs than the baseline models at approx. 17 but a higher MSE than MOB. CORE also has the highest test MSE at around 51. The large discrepancy between training and test MSE in the MOB and CORE models is an indication that they are overfitting.

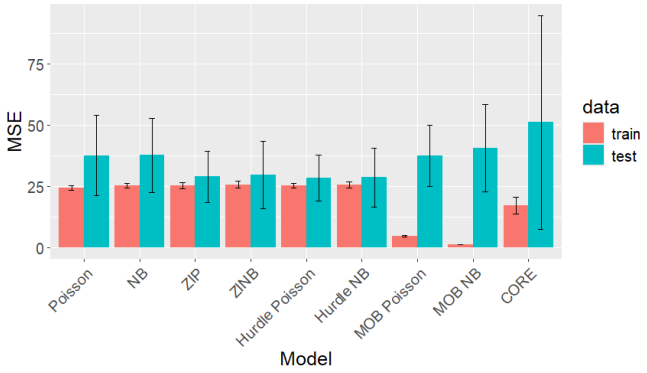

Figure 2: Bar chart showing the mean MSE of the training and test folds. The error bars represent the standard deviation.

Table 3: Results for the MSE of the different models

	Training Data		Test Data	
Model	**Mean**	**Std dev.**	**Mean**	**Std dev.**
Poisson	24.2718	1.0522	37.5067	16.4076
Neg. binomial	25.1875	0.9853	37.6034	15.1111
ZIP	25.1841	1.1763	28.8760	10.5735
ZINB	25.5951	1.3527	29.6864	13.7146
Hurdle (Poisson)	25.2541	0.9648	28.3598	9.5013
Hurdle (neg. bin.)	25.5044	1.2012	28.6120	12.0106
MOB (Poisson)	4.6274	0.3327	37.4594	12.5349
MOB (neg. bin.)	1.1212	0.0040	40.5491	17.9726
CORE	16.9423	3.4411	51.0760	43.6499

Reported MAE metrics as stated in Table 4 and presented in Figure 3 reveal similar levels of overfitting as previously observed in case of MSE wherein highest discrepancy between train and test MAE is associated with the MOB model with our CORE following in at second place.

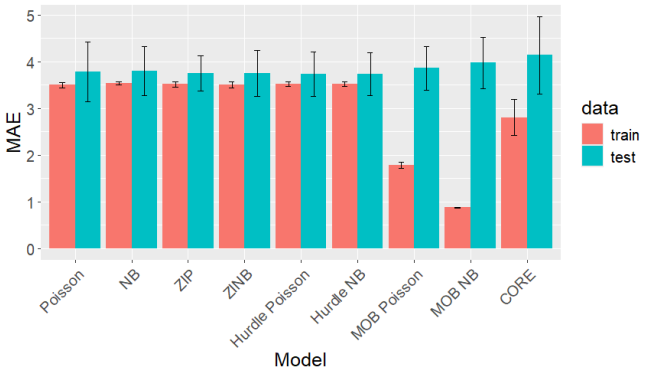

Figure 3: Bar chart showing the MAE of the training and test folds. The error bars represent the standard deviation.

Table 4: Results for the MAE of the different models

	Training Data		Test Data	
Model	**Mean**	**Std dev.**	**Mean**	**Std dev.**
Poisson	3.5019	0.0571	3.7857	0.6387
Neg. binomial	3.5344	0.0356	3.7988	0.5251
ZIP	3.5125	0.0546	3.7489	0.3801
ZINB	3.5037	0.0610	3.7444	0.4884
Hurdle (Poisson)	3.5192	0.0542	3.7325	0.4697
Hurdle (neg. bin.)	3.5236	0.0493	3.7318	0.4550
MOB (Poisson)	1.7831	0.0665	3.8568	0.4708
MOB (neg. bin.)	0.8731	0.0017	3.9711	0.5451
CORE	2.8044	0.3816	4.1348	0.8299

From Figure 4, Poisson and MOB Poisson models are associated with highest dispersion. The CORE model has the seconds highest dispersion. The model with most ideal dispersion (closest to 1) is the Negative Binomial Hurdle model.

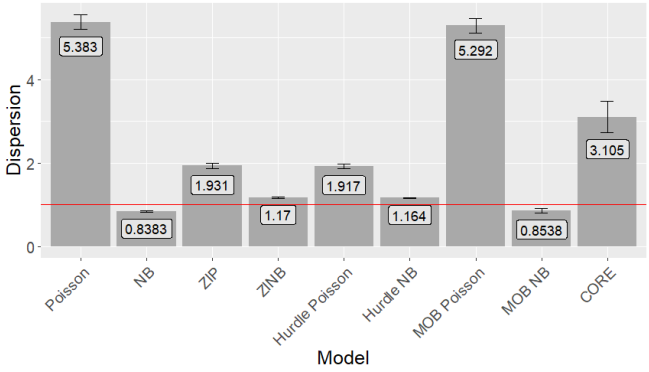

Figure 4: Bar chart showing the mean dispersion statistic of the models. The error bars represent the standard deviation. The red line indicates the ideal value $\phi = 1$.

Figure 5 displays expected/observed zeros and quantifies how well each type of model handled excess zeros in the dataset. This figure states that while the Hurdle models showcased best performance, the Poisson and MOB Poisson performed most poorly. The CORE model reports intermediate prowess at modelling excess zeros.

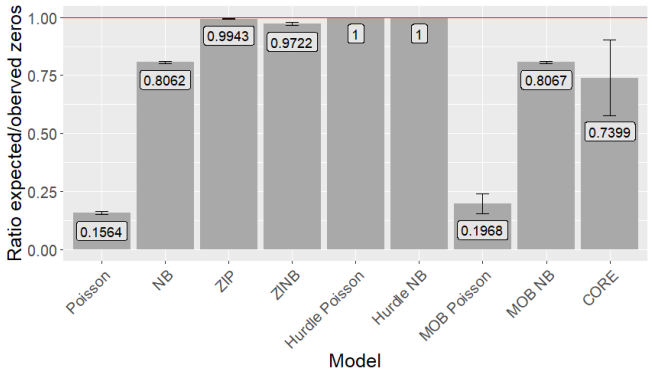

Figure 5: Bar chart showing the ratio of expected zeros predicted by the models vs. observed zeros in the data. The red line indicates the ideal ratio of 1.

D. Varying the Depth Limitation

The results showed that CORE shows symptoms of overfitting. We therefore extend our initially proposed study by investigating the effects of varying the depth limitation on overfitting. Again, we use 10-fold cross validation to evaluate the MSE for different depths. In Figure 6, it can be seen that there is no significant difference in the predictive performance of the model when varying the depth limitation.

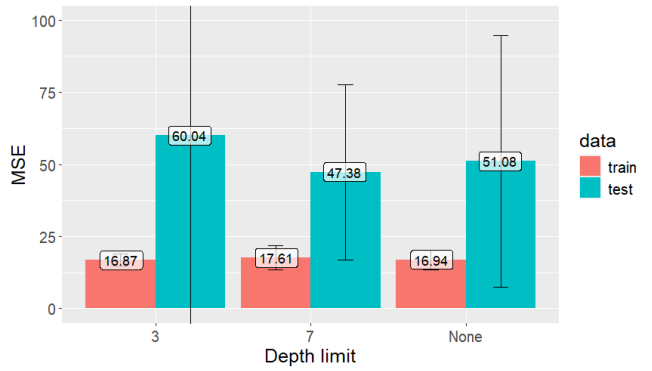

Figure 6: Bar chart showing MSE of training and test folds of CORE models using varying depth limitation. The error bars represent the standard deviation.

V. Discussion

At the start of our study, we hypothesized that the CORE variant of regression trees is effective at predicting count data and is less affected by overdispersion and zero inflation. We expected that it would have lower MSE and MAE than the baseline models and similar results as the MOB variant. We also expected CORE to display less overdispersion and zero-inflation.

The results in Figure 2 and Figure 3 show that CORE is able to model the training data better that the baseline models. However, it performs worse on unseen data, showing that it's ability to generalize is worse. Furthermore, CORE displays higher prediction errors than MOB. While CORE is less overdispersed than the Poisson model and the MOB Poisson model (Figure 4), it exhibits higher overdispersion compared to the remaining models. Similarly, CORE expects more of the observed zeros than the Poisson model and the MOB Poisson

model (Figure 5), which means it is less zero-inflated, but again performs worse in this regard compared to the other models. This means we find little evidence to prove our hypothesis. Nevertheless, it also does not disprove the potential that CORE presents. This further suggests that more research be conducted on variants of CORE.

Our observations also contradict findings by Liu et al. [6], who found that CORE performs similarly to MOB. A reason for this may be the dataset. In their evaluation of CORE, Liu et al. use a simulated dataset comprised of a response variable that has a zero-inflated distribution whose parameters depend on at least one of the simulated predictor variables. Our study on the other hand uses a dataset containing real data, which means that the dataset is inherently noisier and a more suitable candidate to truly access the utility of the model that Liu et al. propose.

E. Overfitting

Overfitting can be observed in both the regression tree algorithms (MOB and CORE) but not in the baseline models (Figure 2). A possible reason for this might be the fact that GLMs are fitted at each node. If these overfit, then the effect may accumulate as the tree grows, resulting in the larger degree of overfitting compared to the baseline models.

Furthermore, it appears that the difference in training and test MSE/MAE is smaller when using CORE compared to MOB, which suggests that the degree of overfitting is smaller when using CORE. This is speculated to be because of the ensemble nature of CORE wherein there is the option to choose from multiple models w.r.t fitting data at each node.

We also considered depth limitation as a possible mechanism to reduce overfitting. However, our results in Figure 6 showed that limiting the depth of the CORE regression tree had no significant impact on the performance. One explanation for such results may be that the GLMs at intermediate steps contribute more towards the overfitting than the tree structure itself.

F. Overdispersion and Zero-Inflation

The results in Figure 4 and Figure 5 show that there is little difference in overdispersion and zero-inflation when comparing the Poisson model and the MOB Poisson model. The negative binomial model and the MOB negative binomial models also display similar overdispersion and zero-inflation. The reason for this may be that the splits do not effectively reduce excess zeros or the heterogeneity of observations that causes overdispersion. This would mean that the GLMs in the terminal nodes face the same problems as the Poisson and negative binomial models, resulting in the similar performance.

On the other hand, the overdispersion and zero-inflation of CORE lies between that of the Poisson model and the negative binomial model (see Figure 4 and Figure 5). This suggests that combining the models results in an averaging effect of the models fitted to the nodes. However, the fact that CORE does not achieve the best performance as observed in the baseline models (i.e. dispersion close to 1 like the negative binomial hurdle model or ratio of expected to observed zeros = 1 like the hurdle models) implies that perhaps less optimal models such as Poisson are selected during the *Model Selection* step. This implies that the strategy for model selection might need to be reconsidered.

G. Future Work

Once the tree has been built, Liu et al. [6] propose to apply cost-complexity pruning. Pruning in general is an approach where the branches of a tree are gradually cut back until the error on hold-out data is minimized, thus reducing over-fitting [21]. Cost-complexity pruning imposes a penalty on the depth of the tree. It uses deviance of a node as the error measure and uses 10-fold cross-validation to determine good values for associated hyperparameter values. There are also other possible post pruning mechanisms like Reduced Error Pruning that may be performed. [6] Our work had primarily relied on pre-pruning / early stopping methods. Thus, future work can explore the effect of introducing cost complexity pruning. This is expected to reduce overfitting.

Also, the dataset explored here had only around 1K data points. This is more than the 720 data points in [6]. Still, it shall be interesting to apply this model on larger datasets to access its feasibility on larger amounts of data. One expectation is the CORE would be significantly slower than other simpler models like the Poisson model due to the repeated need to fit 6 different distributions at each node of the tree. Decision trees in general, due to a need for an underlying hierarchical data structure has higher space complexity than simple linear models. Space complexity is higher still for CORE because unlike traditional decision trees that do not store training related data after construction is complete [23], here the intermediate fitted nodes and their reuse at time of prediction means that model weights need to be stored at each node. CORE may also be harder to vectorize in implementation. Thus, it would be beneficial to investigate different ways of making CORE and other such models faster and more space efficient (perhaps reduce a few types of models that are fit at each node, decide which model to fit based on dataset at the node, vectorize operations wherever possible, and so on).

Further, both MOB and CORE are based on binary splits. One may also consider non-binary splits and determine if that might result in better performance. Additionally, our model selection approach at each node involves 10-Fold Cross validation and picking the one that best fits the subset of data at each intermediate/leaf node. While this is a sound approach, other approaches (Akaike Information Criterion [24], Bayesian Information Criterion (BIC) [25], etc.) are possible and can be experimented with. Lastly, given our aforementioned observation that such hierarchical models with sub-models at each node are prone to overfitting, future work may investigate the cause or extent of overfitting in the intermediate nodes and how this affects child nodes either on this data set or different data.

H. Known Limitations

The biggest known limitation is that it is unclear as to how akin to the work by Liu et al [6], our implementation of CORE, truly is. This is because, while overall concepts to be applied to the model and the general step of the algorithm is explained in Liu et al.'s work, many implementation details are missing. For

instance, while it is stated repeatedly that the best among the 6 models (Poisson, negative binomial, and hurdle models) must be chosen, the method of model selection is not stated. Another piece of missing information includes computation details of residuals in the variable selection phase of the algorithm. The link provided to reference code of CORE in [6] is unreachable. Many a time, this left us wondering as to whether R coding practices and decision decisions that we adopt was as Liu et al intended or not. That said, w.r.t tree building, minor implementation details aside, our version of CORE is believed to have captured all underlying ideas in [6]. The biggest difference between Liu et al.'s implementation and our version is that we chose to omit cost complexity pruning with cross validation to tune hyperparameters as in the former. Instead, we allow for additional pre-pruning options (depth limiting, possibility to set varying values of minimum portion of total nodes that must be present in a node before it can be split). A motivation for this choice stems from another known limitation that our implementation has led to the algorithm being slow. So, additional pre-pruning is more practical as it was found to be much faster than post-pruning. Our implementation is not as efficient as it can be given the unfamiliarity of developers with the R programming language in addition to the lack of reference materials for this model. That said, great care has been taken to write self-explanatory and easy to grasp code. Our implementation is, to the best of knowledge, the only, publicly available runnable manifestation of Liu et al's work [6]. We hope that our implementation sets the stage for more reproduceable experiments around this topic.

VI. Conclusion

Our study was not able to verify that CORE performs similarly well as MOB on a real-world dataset. This means there is the possibility that CORE might not be suitable for modelling count data. Following are some aspects of CORE that warrant further investigation.

- Model selection strategy could potentially be refined to ensure suitable models are selected in each node.
- Split set selection does not consider which model will be fit in the child node but only fits the model of the parent node. This strategy, as described in [6] contradicts the suggestion, also from the same work to fit node specific models for each child node as well.
- Repeated fitting of models at each node in the decision tree likely leads to magnification of effects of overfitting.

Furthermore, we find that both CORE and MOB do not fully mitigate overdispersion and zero-inflation. Our results also showed that CORE is more prone to overfitting than simpler GLMs. Furthermore, we provide a runnable implementation of CORE for reproducibility and to support future works in addition several plausible directions for future work on the subject.

VII. References

[1] A. Agresti, An Introduction to categorical data analysis, Gainesville: John Wiley & Sons, Inc., Hoboken, New Jersey, 2019.

[2] M. Ridout, C. G. Dem´etrio and J. Hinde, "Models for count data with many zeros," in *International Biometric Conference*, Cape, 1998.

[3] The Editors of Encyclopaedia, "Bhopal disaster," Britannica, Encyclopedia Britannica, 29 March 2024. [Online]. Available: https://www.britannica.com/event/Bhopal-disaster. [Accessed 9 April 2024].

[4] G. B. Shengping Yang, "The Negative Binomial regression," *THE SOUTHWEST RESPIRATORY AND CRITICAL CARE CHRONICLES*, vol. 3, no. 10, pp. 50-54, 2015.

[5] J. A. Green, "Too many zeros and/or highly skewed? A tutorial onmodelling health behaviour as count data with Poisson andnegative binomial regression," *HEALTH PSYCHOLOGY AND BEHAVIORAL MEDICINE*, vol. 9, no. 1, pp. 436-455, 2021.

[6] N.-T. Liu, F.-C. Lin and Y.-S. Shih, "Count regression trees," *Advances in Data Analysis and Classification,* vol. 14, pp. 5-27, 2020.

[7] A. Zeileis, T. Hothron and K. Hornik, "Model-Based Recursive Partitioning," *Journal of Computational and Graphical Statistics,* vol. 17, no. 2008, pp. 492-514, 2008.

[8] P. K. T. A. Colin Cameron, Regression Analysis of Count Data, Davis: Cambridge University Press, 2013.

[9] D. B. Hall, "Zero-Inflated Poisson and Binomial Regression with Random Effects: A Case Study," 2004.

[10] J. M. Hilbe, Modeling Count Data, Arizona: Cambridge University Press, 2014.

[11] W.-Y. Loh, "Classification and Regression Trees," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, pp. 14 - 23, 2011.

[12] J. E. a. G. Robert Nisbet, "Regression tree, Handbook of Statistical Analysis and Data Mining Applications," Science Direct, 2009. [Online]. Available: https://www.sciencedirect.com/topics/mathematics/regression-tree.

[13] Y. B. Wah, N. Nasaruddin, W. S. Voon and M. A. Lazim, "Decision Tree Model for Count Data," in *Proceedings of the World Congress on Engineering*, London, 2012.

[14] L. Breiman, J. Friedman, R. Olshen and C. J. Stone, Classification and Regression Trees, Monterey: Wadsworth and Brooks, 1984.

[15] Y. Choi, H. Ahn and J. J. Chen, "Regression trees for analysis of count data with extra Poisson variation,"

Computational Statistical & Data Analysis, vol. 49, no. 3, pp. 893-915, 2005.

[16] H. Tang and E. T. Donnell, "Application of a model-based recursive partitioning algorithm to predict crash frequency," *Accident Analysis & Prevention*, vol. 132, 2019.

[17] P. Cortez, "Student Performance," UCI Machine Learning Repository, 26 November 2014. [Online]. Available: https://archive.ics.uci.edu/dataset/320/student+performance. [Accessed 21 March 2024].

[18] N. Venkat, "The Curse of Dimensionality: Inside Out," September 2018.

[19] J. M. Hilbe, Modelling Count Data, Cambridge: Cambridge University Press, 2014.

[20] T. A. Runkler, Data Analytics: Models and Algorithms for Intelligent Data Analysis, 2nd ed., Wiesbaden: Springer Fachmedien Wiesbaden, 2020.

[21] P. Bruce and A. Bruce, Practical Statistics for Data Scientists: 50 Essential Concepts, Sebastopol, CA: O'Reilly Media, Inc., 2017.

[22] E. B. Willson and M. M. Hilferty, "The Distribution of Chi-Square," *Proc Natl Acad Sci USA*, vol. 17, no. 12, p. 684–688, 1931.

[23] E. Alpaydin, in *Introduction to Machine Learning*, 3 ed., The MIT Press, 2014.

[24] R. Bevans, "Akaike Information Criterion | When & How to Use It (Example)," scribbr, 26 March 2020 . [Online]. Available: https://www.scribbr.com/statistics/akaike-information-criterion/. [Accessed 10 May 2024].

[25] S. Bauldry, "Bayesian Information Criterion," ScienceDirect, 2015. [Online]. Available: https://www.sciencedirect.com/topics/social-sciences/bayesian-information-criterion.

[26] C. Paulo, "UC Irvine Machine leanring repository," 26 11 2014. [Online]. Available: https://archive.ics.uci.edu/dataset/320/student+performance.

[27] K. H. Lee, C. Pedroza and E. B. C. Avritscher, "Evaluation of negative binomial and zero-inflated negative binomial models for the analysis of zero-inflated count data: application to the telemedicine for children with medical complexity trial.," Trials, 2023.

[28] N.-T. Liu, F.-C. Lin and Y.-S. Shih, "Count regression trees," *Advances in Data Analysis and Classification*, vol. 14, no. 10, pp. 5-27, 2020.

[29] C. G. D. J. H. Martin Ridout, "Models for count data with many zeros," in *International Biometric Conference*, Cape, 1998.

[30] C. X. Feng, "A comparison of zero-inflated and hurdle models for modeling zero-inflated count data," *Journal of Statistical Distributions and Applications*, vol. 8, 2021.