HEC MONTRÉAL

École affiliée à l'Université de Montréal

Trees and Random Forests for Nonhomogeneous Count Data Processes

par
Walid Mathlouthi

Thèse présentée en vue de l'obtention du grade de Ph. D en administration
(Option Méthodes quantitatives de gestion)

Octobre 2015

HEC MONTRÉAL

École affiliée à l'Université de Montréal

Cette thèse intitulée :

Trees and Random Forests for Nonhomogeneous Count Data Processes

Présentée par :

Walid Mathlouthi

a été évaluée par un jury composé des personnes suivantes :

François Bellavance
HEC Montréal
Président-rapporteur

Denis Larocque
HEC Montréal
Directeur de recherche

Marc Fredette
HEC Montréal
Directeur de recherche

Didier Chetelat
HEC Montréal
Membre du jury

Thierry Duchesne
Université LAVAL
Examinateur externe

Rajesh Kumar Tyagi
HEC Montréal
Représentant du directeur de HEC Montréal

RÉSUMÉ

Les méthodes d'arbres forment une classe de modèles utiles, polyvalents et populaires auprès des utilisateurs. De plus, les méthodes d'agrégation d'arbres, comme la forêt aléatoire, comptent parmi les outils de prévision les plus performants. Dans cette thèse nous considérons une variable réponse de dénombrement, représentant un nombre d'événements récurrents observés dans une période de temps. Nous développons des méthodes d'arbres et de forêts dans le cas où la fonction de taux du processus est non–homogène dans le temps et/ou dans le cas où il y a un nombre de zéro excédentaires. Des études par simulation démontrent que les méthodes proposées offrent un avantage réel comparativement aux méthodes existantes dans ces cas.

Mots clés : Processus de Poisson non–homogène, Arbre de Poisson, Forêt aléatoire, Processus de Poisson gonflé à zéro, Événements récurrents, Maximum de vraisemblance, Segmentation fonctionnelle.

ABSTRACT

Tree-based methods are a class of useful, versatile and popular models. Moreover, tree aggregation methods, such as random forests, are among the most powerful prediction tools. In this thesis, we consider a count response variable, representing the number of recurrent events observed over a period of time. We develop trees and forests based methods when the rate function of the process is non-homogeneous and/or when the process has an excess number of zeros. Simulation studies show that the proposed methods offer a real advantage over the existing ones in these cases.

Keywords: Non-homogeneous Poisson process, Poisson tree, Random forest, Zero-inflated Poisson process, Recurrent events, Maximum likelihood, Functional clustering.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

REMERCIEMENTS

Tout d'abord, je tiens à adresser mes plus grands remerciements et ma plus profonde reconnaissance à mon professeur et directeur de thèse, Denis Larocque, pour son encadrement, sa patience, sa confiance et sa grande disponibilité tout au long de ces années de recherche. J'ai pris un grand plaisir à travailler avec lui. Merci Denis!

J'adresse un chaleureux merci à mon co-directeur, Marc Fredette, pour l'attention qu'il a toujours porté à mes travaux, pour ses conseils et son écoute qui ont été indispensables pour la bonne réussite de cette thèse. Son énergie et sa confiance ont été des sources de motivation hors de prix.

Tous mes remerciements, également, aux membres du jury pour leur temps consacré à lire et à évaluer ma thèse.

Je tiens à exprimer ma profonde gratitude à mes parents qui m'ont grandement soutenu et encouragé à réaliser mon rêve d'accomplir mes études doctorales.

Le plus grand merci va à mon épouse Wafa, pour son soutien et son encouragement, surtout pour le sprint des derniers mois.

Je suis reconnaissant envers mon ami Arafat, qui fut pour moi d'un soutien inestimable. Je remercie également son épouse Houda pour son encouragement et sa générosité.

Un grand merci à mes amis Hafedh, Oussama, Mohamed-Ali et Lotfi pour leur support, leur encouragement et leurs bonnes paroles.

Merci aussi à Mohamed Jabir pour ses conseils et pour son soutien informatique.

Je remercie tous mes collègues au Centre International de Recherche sur la Finance Coopérative pour leur solidarité, leur soutien et leur flexibilité.

Je remercie bien évidemment les différents organismes qui m'ont appuyé financièrement tout au long du parcours doctoral : Le fonds de recherche du Québec – Nature et technologies (FRQNT), la fondation Édouard-Montpetit, Standard Life et HEC Montréal.

Je dédie finalement ce travail à l'âme de ma chère grand-mère et à mes chers parents qui ont sacrifié leur vie pour moi, et qui ont été mes repères.

Je les remercie pour leur amour, leur affection et leur patience.

INTRODUCTION GÉNÉRALE

Dans cette thèse, nous abordons une classe de méthodes non-paramétriques basées sur le partitionnement récursif des données, qui sont plus communément appelées les méthodes d'arbres. Ces méthodes sont très utilisées en pratique pour l'analyse des données car elles représentent une alternative aux modèles paramétriques, à la fois pour l'estimation et la prévision. Elles ont été popularisées par le paradigme CART (Classification and Regression Trees), introduit par Breiman et al. (1984), dans le cadre d'une variable réponse continue ou catégorielle. Les arbres produisent généralement des règles simples et faciles à interpréter. Un avantage majeur des arbres est qu'ils peuvent être utilisés sans spécifier d'avance le lien entre les variables explicatives et la variable réponse. De plus, ils peuvent détecter automatiquement certains types d'interaction entre les variables explicatives. Combiner plusieurs arbres obtenus selon différents mécanismes de perturbations aléatoires de l'algorithme de base offre généralement une meilleure performance prévisionnelle. C'est pourquoi ces méthodes d'agrégation sont devenues très populaires en pratique et la forêt aléatoire (Breiman, 2001) est possiblement la méthode la plus connue. Dans cette thèse, nous abordons le cas d'une variable réponse de dénombrement, c'est-à-dire, une variable qui prend des valeurs parmi les entiers, incluant 0. Ce type de variable est très fréquent en pratique et est souvent le fruit d'une étude où on s'intéresse au nombre d'événements se produisant dans un intervalle de temps. Nous pouvons citer en guise d'exemples le nombre de réclamations d'assurance d'un client, le nombre de tranches de crédit impayées, le nombre d'accidents sur une autoroute, ou le nombre de visite à l'urgence par une personne. Il existe une vaste littérature traitant des modèles paramétriques pour des variables de dénombrement. Un traitement moderne de ces dernières est présenté dans le livre de Cook et Lawless (2007). Il est également possible de construire un arbre de Poisson avec le package rpart (Therneau et al, 2014) du logiciel R (R Core Team 2014). Le développement des méthodes d'arbres pour modéliser une variable réponse

de dénombrement est néanmoins limité à certaines situations de base. Dans cette thèse, nous développons des méthodes d'arbres pour une variable de dénombrement qui généralisent les méthodes existantes dans différentes directions. Essentiellement, nous allons traiter le cas d'une fonction de taux non-homogène et le cas d'une variable avec des zéros excédentaires. Cette thèse est composée de trois articles. Dans le premier article, nous abordons le cas de processus non-homogènes. En effet, les méthodes d'arbres existantes supposent que le taux d'apparition des événements est constant dans le temps. Nous proposons un algorithme de forêt aléatoire pour processus de Poisson non-homogènes. Nous avons nommé cette approche « Nonhomogeneous Poisson Processes Forests » (NHPPF). L'idée de base consiste à partitionner la période de temps observée en sous-périodes et de permettre aux taux d'événements d'être différents dans chaque sous-période. Cette approche a une grande flexibilité, car elle peut s'adapter aux caractéristiques des données (mensuelles, annuelles, etc.). Cette technique de partitionnement a été utilisée par Lawless et Zhan (1998) pour un modèle paramétrique particulier. Les résultats des simulations montrent la supériorité de la méthode proposée, en termes de pouvoir prédictif, en comparaison avec plusieurs compétiteurs, comme la régression de Poisson et la forêt de Poisson homogène. Le gain prédictif est remarquable lorsque les processus sont non-homogènes mais la performance de la nouvelle méthode est aussi bonne dans le cas de processus homogènes. Dans le deuxième article, nous abordons le cas d'une variable réponse de dénombrement avec des zéros excédentaires. Cette situation survient fréquemment en pratique lorsque le nombre de zéros observés est grand et ne peut pas être adéquatement modélisé ni par les modèles courants (Poisson ou binomial négatif), ni par les variables explicatives disponibles. Seulement Lee et Jin (2006) ont proposé une méthode d'arbre pour une variable réponse distribuée selon la loi de Poisson gonflée à zéro (« Zero Inflated Poisson » ou ZIP). Par contre, leur approche utilise un seul modèle et suppose donc intrinsèquement que les effets des variables explicatives sont les mêmes pour la partie zéro et la partie Poisson du processus. Ceci est différent des approches paramétriques usuelles qui utilisent deux modèles. De plus, la méthode de Lee et Jin (2006) est limitée au cas homogène. C'est pourquoi nous proposons une nouvelle approche qui propose des solutions à ces deux limites. Notre approche utilise deux forêts, l'une pour la partie zéro et l'autre pour la partie Poisson. Ainsi, dans le cadre homogène, notre méthode généralise l'approche de Lee et Jin (2006). Mais nous proposons également une version pouvant s'appliquer au cas de processus non-homogènes avec zéros excédentaires, généralisant ainsi la méthode NHPPF du premier

article. Les résultats des simulations montrent un net avantage en faveur de la nouvelle méthode par rapport à la méthode de Lee et Jin (2006) et par rapport à la méthode NHPPF, en particulier lorsque le nombre de zéros excédentaires est grand. Dans le troisième article nous proposons une extension de la méthode NHPPF afin d'obtenir une estimation lisse de la fonction de taux. En effet, la méthode NHPPF est basée sur l'idée de diviser l'intervalle du temps en plusieurs sous-intervalles et de supposer que la fonction de taux est constante, étant donné les variables explicatives, dans chacun d'eux. Parfois, ces sous-intervalles sont naturellement suggérés par les données ou par les questions de recherche. Mais le choix des intervalles n'est pas toujours évident. Nous développons ici une méthode où il n'est pas nécessaire de choisir les sous-intervalles et qui produit une estimation lisse de la fonction de taux, comparativement à une estimation constante par morceaux pour la méthode NHPPF. Les résultats des simulations montrent la supériorité de cette méthode lorsque la fonction de taux est continue. Une illustration d'utilisation de cette méthode avec des données réelles provenant d'un programme de fidélisation est présentée. Dans cet exemple une technique d'analyse de regroupement fonctionnelle est utilisée afin de segmenter les clients selon leurs courbes de taux estimées par notre méthode.

Chapter 1

Regression Trees and Forests for Non–homogeneous Poisson Processes

Abstract

We propose tree and random forest methods for non–homogeneous Poisson processes. The splitting criterion is derived from a model with a piecewise constant rate function. A simulation study shows that the new method performs well.

Key Words: Non–homogeneous Poisson Processes, Poisson Tree, Random Forests, Maximum Likelihood, Recursive Partitioning, Recurrent Events.

1.1 Introduction

Count data, measuring the number of times that an event of interest happens during a given time period occur frequently in practice. Some examples are the number of warranty or insurance policy claims by a client, the number of unpaid credit installments, the number of accidents on a highway and the number of seizures for an epileptic. When the goal is to relate a set of covariates to the number of events for a sample of subjects, many parametric models are available, including the Poisson and negative binomial regression models. Cook and Lawless (2007) provide a nice treatment covering the modeling of count data.

In this paper, we are more interested in a particular class of nonparametric models based on recursive partitioning also called tree–based models. Tree–based methods, or just trees,

are valuable alternatives to parametric methods and are very popular among practitioners. Some of their advantages are: no need to specify a parametric form, ability to automatically detect interactions, and ease of interpretation and visualization. They were first developed to handle a categorical or a continuous outcome. See Breiman *et al.* (1984) for the early developments of the CART (Classification and Regression Tree) paradigm and the earlier references. This paradigm builds a large tree by selecting the best split among all possible splits at each intermediate node. In order to avoid over–fitting, a subtree is then selected by pruning the large tree using cross–validation. Within a similar framework as CART, Poisson regression trees can be fitted in R (R Core Team 2013) with the package rpart (Therneau *et al.* 2014). In this case, the splitting criterion is basically the likelihood ratio test to compare two Poisson distributions. However, a Bayes estimate of the rate is used in order to avoid an infinite value of the deviance which occurs when the maximum likelihood estimate of the rate is 0. See Therneau and Atkinson (2014) for details.

Chaudhuri *et al.* (1995) proposed another method to build Poisson regression trees. Contrarily to the CART approach, this one proceeds by fitting a Poisson loglinear model with all the covariates in each intermediate node. The adjusted Anscombe residuals of the fitted model are then obtained. The Levene's two sample test is then applied to each covariate to compare the two groups formed by the positive and negative residuals. The selected covariate for the split is the one with the largest absolute statistic value. The split for the selected covariate is the average of the two group means along the covariate. Once a large tree is built, a pruning algorithm can be applied as in CART.

GUIDE (Generalized, Unbiased, Interaction Detection and Estimation) is a general stand alone program for tree building designed and maintained by Wei-Yin Loh

(http://www.stat.wisc.edu/ loh/guide.html). The GUIDE method has many features that are described in Loh (2002). See also Loh (2011) for a recent overview. One of its main feature is that GUIDE performs a test to evaluate the discriminating power of each covariate separately in each node. The covariate with the smallest p–value is then retained and the best split is found with this covariate. This two–step split selection is performed in order to alleviate possible bias in the variable selection. Indeed, with the CART approach, covariates with more potential splits may have a tendency to be selected more often than covariates

with less potential splits. GUIDE can fit Poisson regression trees using the splitting rule of Chaudhuri *et al.* (1995).

Extensions of Poisson regression trees into different directions have been proposed. Lee (2005) generalized the method of Chaudhuri *et al.* (1995) to the case of multivariate outcomes with models using GEE (Generalized Estimating Equations). This method can be used to fit multiple count data responses. In some applications, count data exhibit greater variability than what is expected from a Poisson model. To handle this, Choi *et al.* (2005) extended both the Chaudhuri *et al.* (1995) and the GUIDE approach to the case of a Poisson outcome with extra–variation. In other applications, count data exhibit more zeros than what is expected from a Poisson model. Lee and Jin (2006) proposed a zero–inflated tree model with the CART approach.

In the last decade, the attention has shifted towards using trees as part of an ensemble. This is mainly due to the fact that combining many trees has often a better predictive capability than a single tree. Random forests (Breiman 2001) is such a combination method among the most popular. The good performance of random forests has been demonstrated in many empirical studies (e.g., Breiman 2001 and Hamza and Larocque 2005) and their theoretical properties have also been studied (Biau *et al.* 2008, Biau and Devroye 2010). This is why they are now part of the standard practitioners' toolbox. In order to get into the literature about random forests and ensemble methods, the surveys by Rokach (2008), Siroky (2009) and Verikas *et al.* (2011) are good starting points.

GUIDE has implemented bagging (Breiman 1996) and random forests and thus it is possible to build ensembles of Poisson regression trees with this software. Boosting, introduced by Freund and Schapire (1997), is another popular and powerful ensemble method that can be used with trees as base learners. Borisov *et al.* (2009) used the gradient–boosting framework (Friedman 2001) to build ZIP (Zero–Inflated Poisson) trees and ensembles in the case of a rare event count, i.e., a response with more zeros than a usual Poisson response.

All the methods discussed above work under the basic assumption that the Poisson process is homogeneous with respect to time. That is, the rate function of the response, given the covariates, does not vary with time. In this paper, we want to introduce a tree–based method and a forest based on it for an outcome from a non–homogeneous Poisson process. The basic

idea is to partition the total time period and to allow a different rate function in each subperiod. This approach allows more flexibility as it can adapt to local (in time) features in the data. Such a piecewise rate function approach is studied in Lawless and Zhan (1998) but with a parametric model. Moreover, since the number of subperiods is arbitrary, it can increase with the sample size, allowing a more and more accurate fit.

The paper is organized as follows. Section 1.2 describes the proposed approach, including the split function, to build trees and forests for non–homogeneous count data. The results from a simulation study are presented in Section 1.3. This study compares homogeneous and non–homogeneous benchmark (no covariates), parametric and forests models over many different data generation scenarios. A real data example is given in Section 1.4. Concluding remarks and possibilities for future work are given in Section 1.5.

1.2 Non–homogeneous Poisson Process Tree and Forest

In this section, we describe the algorithms to build a non–homogeneous Poisson process tree (NHPPT) and a non–homogeneous Poisson process forest (NHPPF) using the NHPPT as base learner.

1.2.1 Non–homogeneous Poisson process model

The basic model that will be used to define a splitting criterion for the tree algorithm is a piecewise constant Poisson process model. Assume that we have N subjects and that we observe the number of times an event of interest occurs during a fixed time period T. Let us assume that this time period is partitioned into K subperiods, $T_1, \ldots, T_K$, such that

$$\bigcup_{k=1}^{K} T_k = T \quad \text{and} \quad T_i \cap T_j = \emptyset \quad \text{for all} \quad i \neq j.$$

Each subperiod T_k may be an interval or a finite union of disjoint intervals. Let N_{ik} be the number of events which occurred for subject i in subperiod T_k. We assume that events occurred according to a homogeneous Poisson process in each subperiod T_k. Each N_{ik} thus follows a Poisson distribution with expectation μ_k and all the N_{ik}'s are independent. Hence, subjects are independent but the number of events in different periods are also independent

for any given subject. This is a direct consequence of the independent increment property of a Poisson process.

In many applications, the K subperiods will simply be adjacent time intervals

$(a_0, a_1], (a_1, a_2], \ldots, (a_{K-1}, a_K]$ of the whole period $T = (a_0, a_K]$. But the above model is more general than that and the subperiods can represent non–adjacent periods. Some examples are days (T_1=all the Mondays in a year, T_2=all the Tuesdays in a year and so on) and months (T_1=all the months of January in a 10–year period and so on).

The log–likelihood of this model is:

$$\sum_{i=1}^{N}\sum_{k=1}^{K}(N_{ik}\ln(\mu_k) - \mu_k - \ln(N_{ik}!)),$$

and the maximum likelihood estimators (MLEs) are simply

$$\hat{\mu}_k = \frac{\sum_{i=1}^{N} N_{ik}}{N}, k = 1, \ldots, K.$$

1.2.2 Splitting criterion for tree building

We assume that the reader is familiar with the mechanics of tree building as described, for instance, in Breiman *et al.* (1984). We assume that we have a vector of p covariates $X = (X_1, \ldots, X_p)$ per subject. Note that these covariates do not vary with time. They will usually be measurements available at time 0 for each subject. Let t be the current node of the tree to be split into two nodes, the left one t^L, and the right one t^R. For a continuous (or at least ordinal) covariate x, the possible splits take the form $x \leq c$ where c is a specified cutpoint. For a categorical covariate x, the possible splits take the form $x \in \{c_1, \ldots, c_l\}$ where $\{c_1, \ldots, c_l\}$ is a subset of the possible values of x. Each possible split with each covariate (one at a time) is evaluated and the best one is retained according to the criterion described below. For a candidate split, the subset of observations in node t is partitioned into two nodes, t^L and t^R, according to whether or not the splitting condition is true. Let N^L and N^R be the sizes of these two nodes. In this two–node model, each node has its own set of

parameters μ_k^L and μ_k^R and the MLEs are given by:

$$\hat{\mu}_k^L = \frac{\sum_{i \in t^L} N_{ik}}{N^L} \quad \text{and} \quad \hat{\mu}_k^R = \frac{\sum_{i \in t^R} N_{ik}}{N^R}.$$

The total observed log–likelihood for a candidate split is then

$$\text{LL} = \sum_{i \in t^L} \sum_{k=1}^{K} (N_{ik} \ln(\hat{\mu}_k^L) - \hat{\mu}_k^L - \ln(N_{ik}!)) + \sum_{i \in t^R} \sum_{k=1}^{K} (N_{ik} \ln(\hat{\mu}_k^R) - \hat{\mu}_k^R - \ln(N_{ik}!)).$$

The best split is the one that maximizes LL among all allowable splits. Note that the $\ln(N_{ik}!)$ terms can be omitted in practical implementations since their total contribution is constant across splits.

Having defined a splitting criterion, the tree building can then proceed as usual. Starting from the root node with all the observations, a first split is performed and then the process is repeated recursively with the two resulting nodes until a stopping criterion is attained. Typically, when a node has less than a predetermined number of observations, splitting is stopped and this node becomes a terminal node. If a single tree is needed, then a pruning algorithm is usually performed to avoid overfitting. We do not describe this since we are more interested in ensembles of trees in this paper, which are built with unpruned trees. The interested reader can refer to Breiman *et al.* (1984) for a cross–validation based pruning method.

For an observation with covariate vector $X = x$, an estimation of $\mu_1, \ldots, \mu_K$ can be obtained as the MLE in the terminal node in which the observation ends up when thrown down the tree. Namely, if $t(x)$ is the terminal node in which an observation with $X = x$ falls, then

$$\hat{\mu}_k(x) = \frac{\sum_{i \in t(x)} N_{ik}}{N^{t(x)}}, k = 1, \ldots, K, \tag{1.1}$$

where $N^{t(x)}$ is the size of node $t(x)$. These quantities can also be used as predictions for the number of events, in each subperiod, for a subject with $X = x$, or they can be summed to obtain a prediction for the total number of events over the whole time period.

Note that the effects of the covariates are assumed to be the same over time (i.e. over the time periods) in the proposed tree model. This is because when a split occurs, the whole

time period is split all at once. But the rates of the time periods within each node can differ, making the model non–homogeneous.

1.2.3 Non–homogeneous Poisson process forest

Random forest is an ensemble method introduced by Breiman (2001). This method consists of building many trees with a subset of covariates selected randomly in each node and with bootstrap data sets obtained from the original data set. More specifically, the random forest algorithm is structured as follows:

1. Draw B bootstrap samples from the original data set.

2. For each bootstrap sample, grow an unpruned NHPPT, with the following modification: at each node, randomly sample p_0 $(0 < p_0 \leq p)$ of the p predictors and choose the best split among those variables.

Estimations of $\mu_1, \ldots, \mu_K$, in each subperiods, for a subject with $X = x$, are obtained by averaging the estimates (predictions) of the individual trees. Namely,

$$\hat{\mu}_k^F(x) = \frac{\sum_{b=1}^{B} \hat{\mu}_k^b(x)}{B}, k = 1, \ldots, K,$$

where $\hat{\mu}_k^b(x)$ is the estimation from the b^{th} tree as given by (1.1).

1.3 Simulation Study

In this section, we will compare the proposed NHPPF to different methods, including parametric models, with artificial data generated from six different processes.

As in Section 1.2, we have N independent subjects and we observe the number of times an event of interest occurs during a fixed time period T. This time period is partitioned into K subperiods, $T_1, \ldots, T_K$. We define N_i to be the total number of events during the whole time period T, for subject i. Moreover, we define N_{ik} to be the number of events for subject i in subperiod T_k, $k = 1, \ldots, K$. We assume that $x_i = (x_{i1}, \ldots, x_{ip})'$ is the vector of covariates for subject i.

1.3.1 Models compared

Six models are fitted to the generated data sets for the comparison. The first two are benchmarks models that do not use the covariates, the next two are parametric models using the covariates and, finally, the last two are forest models. These models are now described.

Model 1: This is simply a homogeneous Poisson process model without covariates. Specifically, this model assumes that N_i is a Poisson random variable with mean μ.

Model 2: This is a non–homogeneous piecewise constant Poisson process model without covariates. Specifically, this model assumes that N_{ik} is given by a Poisson random variable with mean μ_k and that all the N_{ik}'s are independent.

Model 3: This is the usual Poisson regression model. Specifically, this model assumes that N_i is a Poisson random variable with mean μ_i and that $\mu_i = \mu_0 \exp(x_i'\beta)$, where $\beta = (\beta_1, \ldots, \beta_p)'$ is a vector of unknown parameters.

Model 4: This is a piecewise Poisson regression model. Hence it is a non–homogeneous Poisson regression model. Specifically, this model assumes that N_{ik} is a Poisson random variable with mean μ_{ik} and that $\mu_{ik} = \mu_k \exp(x_i'\beta)$.

Model 5: This is a homogeneous forest built with the approach described in Section 1.2. However, the trees are built using the whole time period T, i.e., a single interval.

Model 6: This is the NHPPF as described in Section 1.2 with the K subperiods.

Models 1 to 4 are fitted using the glm() function in R. Models 5 and 6 were implemented in C. When building a tree, we set to 30 the minimum number of observations needed in a node to attempt splitting and any resulting node must have at least 10 observations. Three out of the nine predictors were randomly selected at each node to find the best split (this 1/3 ratio is typically retained with regression forests). Each forest was built with 500 trees.

1.3.2 Simulation design

We consider six different data generating processes (DGP). They come from three different functional forms and we consider two cases, homogeneous and non-homogeneous, for each of

them. In all cases, the nine available covariates $X_1, \ldots, X_9$, are independent and uniformly distributed over the interval $[0, 10]$. The first three variables are related to the outcome and the last six are noise covariates. The whole time period T is divided into 12 disjoint intervals. In each interval, the number of events comes from a Poisson distribution with mean μ_k, $k = 1, \ldots, 12$. Let $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_{12})$.

The first two DGPs are parametric Poisson models with main effects only.

DGP 1H: Homogeneous case

$\ln(\mu_k) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3$, for all $k = 1, \ldots, 12$, where $\beta_0 = -0.8$, $\beta_1 = 0.1$, $\beta_2 = -0.1$ and $\beta_3 = 0.05$. Hence this is a homogeneous Poisson regression model.

DGP 1NH: Non–homogeneous case

$\ln(\mu_k) = \beta_{0k} + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3$, for $k = 1, \ldots, 12$, where $\beta_{0k} = \ln(0.1 \cdot k) - 0.3$, $\beta_1 = 0.1$, $\beta_2 = -0.1$ and $\beta_3 = 0.05$. Hence this is a non–homogeneous Poisson regression model.

The next two DGPs are parametric Poisson models with more complicated effects. Note that models 3 and 4 of Section 1.3.1 will be fitted using the main effects only. This will allow to see if a forest can better capture the covariates' effects compared to a misspecified parametric model.

DGP 2H: Homogeneous case

$\ln(\mu_k) = \beta_0 + \beta_1 X_1 X_2 + \beta_2 X_1 \ln(X_2) + \beta_3 X_2^2 X_3$, for all $k = 1, \ldots, 12$, where $\beta_0 = -0.15$, $\beta_1 = 0.03$, $\beta_2 = -0.03$ and $\beta_3 = -0.01$.

DGP 2NH: Non–homogeneous case

$\ln(\mu_k) = \beta_{0k} + \beta_1 X_1 X_2 + \beta_2 X_1 \ln(X_2) + \beta_3 X_2^2 X_3$, for $k = 1, \ldots, 12$, where $\beta_{0k} = \ln(0.1 \cdot k) + 0.3$, $\beta_1 = 0.03$, $\beta_2 = -0.03$ and $\beta_3 = -0.01$.

The final two DGPs are simple tree models with four leaves.

DGP 3H: Homogeneous case

Leaf 1. If $X_1 \leq 5$ and $X_2 \leq 5$ then $\boldsymbol{\mu} = (0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1)$.

Leaf 2. If $X_1 \leq 5$ and $X_2 > 5$ then $\boldsymbol{\mu} = (0.4, 0.4, 0.4, 0.4, 0.4, 0.4, 0.4, 0.4, 0.4, 0.4, 0.4, 0.4)$.

Leaf 3. If $X_1 > 5$ and $X_3 \leq 5$ then $\boldsymbol{\mu} = (0.9, 0.9, 0.9, 0.9, 0.9, 0.9, 0.9, 0.9, 0.9, 0.9, 0.9, 0.9)$.

Leaf 4. If $X_1 > 5$ and $X_3 > 5$ then $\boldsymbol{\mu} = (1.2, 1.2, 1.2, 1.2, 1.2, 1.2, 1.2, 1.2, 1.2, 1.2, 1.2, 1.2)$.

DGP 3NH: Non–homogeneous case

Leaf 1. If $X_1 \leq 5$ and $X_2 \leq 5$ then $\boldsymbol{\mu} = (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2)$.

Leaf 2. If $X_1 \leq 5$ and $X_2 > 5$ then $\boldsymbol{\mu} = (1.2, 1.2, 1.2, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 1.2, 1.2, 1.2)$.

Leaf 3. If $X_1 > 5$ and $X_3 \leq 5$ then $\boldsymbol{\mu} = (0.1, 0.1, 0.1, 1.2, 1.2, 1.2, 1.2, 1.2, 1.2, 0.1, 0.1, 0.1)$.

Leaf 4. If $X_1 > 5$ and $X_3 > 5$ then $\boldsymbol{\mu} = (1.2, 1.1, 1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1)$.

For each DGP, 100 simulation runs are performed. For each run, the training sample size is 500 and the test sample size is 10,000. The six models are fitted with the training data and their performance is assessed with the test data using the predictive mean squared error (PMSE) defined by

$$\text{PMSE} = \frac{\sum_{k=1}^{12} \sum_{i=1}^{10{,}000} (y_{ki} - \hat{y}_{ki})^2}{120,000},$$

where y_{ki} and $\hat{y}_{ki}$ are the real and predicted responses for the i^{th} test observation in interval k.

1.3.3 Results

The results of the simulations are presented in Table 1.1. We start with the first DGP, 1H, which is simply a homogeneous Poisson regression model with only main effects for the covariates. We see that using the covariates improve the predictions since models 3 to 6 all have a smaller PMSE than the benchmark (no covariates) models 1 and 2. Without surprise, the best performer is model 3 which is precisely the true model for this DGP. However, we see that the non–homogeneous, and thus over–specified, model 4 has an almost identical performance. As for the forests (models 5 and 6), we see that their PMSEs are only slightly higher than model 3. Thus, they are able to recover almost all of the predictive power of the covariates without having to guess a specific parametric form beforehand.

Table 1.1 Simulation results. The first number is the average PMSE over the 100 runs and the number between parentheses below is the standard deviation. The smallest PMSE for a given scenario is in bold.

	Model					
	No covariates		Poisson regression		Random forest	
DGP	H (1)	NH (2)	H (3)	NH (4)	H (5)	NH (6)
1H	0.7882	0.7550	**0.6673**	0.6688	0.6729	0.6894
	(0.0078)	(0.0063)	(0.0051)	(0.0052)	(0.0053)	(0.0062)
1NH	1.0331	0.8422	0.8859	**0.7163**	0.8924	0.7383
	(0.0100)	(0.0069)	(0.0068)	(0.0055)	(0.0069)	(0.0069)
2H	0.9997	0.9715	0.8305	0.8324	**0.6998**	0.7175
	(0.0186)	(0.0166)	(0.0173)	(0.0175)	(0.0122)	(0.0124)
2NH	1.2435	1.0902	1.0646	0.9027	0.9315	**0.7525**
	(0.0239)	(0.0198)	(0.0230)	(0.0222)	(0.0189)	(0.0181)
3H	0.8633	0.8342	0.7191	0.7207	**0.6553**	0.6703
	(0.0102)	(0.0075)	(0.0075)	(0.0075)	(0.0064)	(0.0069)
3NH	0.8900	0.8615	0.8602	0.8617	0.8628	**0.6631**
	(0.0065)	(0.0056)	(0.0056)	(0.0057)	(0.0056)	(0.0044)

Looking at the second DGP, 1NH, we see that the models which assume a homogeneous process do not do well compared to their non–homogeneous counterparts. This is not surprising since this DGP is indeed non–homogeneous. Model 4 is the best one but it is followed closely by the NHPPF (model 6). Once again, a forest is able to recover almost all of the predictive power.

The third DGP, 2H, is still a Poisson regression but this time the covariates' effects are more complicated than simple main effects and models 3 and 4 are wrongly specified. We see that the forests are able to use more efficiently the covariates' predictive power. Hence even when the right model family is assumed, using a nonparametric approach like a random forest may be preferable than a wrongly specified parametric model. Moreover, the PMSE of the NHPPF is only slightly higher than that of the homogeneous forest.

Looking at the fourth DGP, 2NH, we see that the NHPPF clearly outperforms the other models. It is the only model which is able to adapt simultaneously to the complicated relation between the covariates and the outcome and to the fact that the process is non–homogeneous.

The fifth DGP, 3H, is a homogeneous tree. The parametric models 3 and 4 are wrongly specified and we see that the forests are the two best models (with a slight advantage to the homogeneous forest).

Finally, the sixth DGP, 3NH, is a non–homogeneous tree. As for DGP 2NH, the clear winner is the NHPPF. Once again, it is able to make the most of this situation by handling simultaneously a hard to guess link between the covariates and the outcome and the non–homogeneous nature of the process. Note that a single (homogeneous and non-homogeneous) tree should have performed as well as the corresponding random forest for DGP 3H and 3NH because the underlying DGP is a single tree.

The main conclusion of this study is that the proposed NHPPF is either the best performer or very close to the best in all situations considered. Moreover, using a method that assumes a non–homogeneous process when the true DGP is homogeneous is not too harmful. However, using a method that assumes a homogeneous process when the true DGP is non–homogeneous leads almost always to a severe deterioration of the performance. Hence, when in doubt it might be more prudent to use a non–homogeneous model.

1.4 Data example

In this section, we illustrate the use of the proposed method using data from Coop HEC Montréal, a co-op university store. Coop HEC Montréal provides books, computing equipment, and various supplies to the university community. Details on the transactions made by the members of the co-op are available for a three year period spanning from June 1st, 2011 to May 31st, 2014. In this example, the response variable is the number of transactions made by a member. The goal is to predict the response variable for a given year based on the values of the covariates from the previous year, using each calendar month as a subperiod. We define year 1 as the one ranging between June 1st, 2011 and May 31st, 2012, year 2 as the one ranging between June 1st, 2012 and May 31st, 2013, and year 3 as the one ranging between June 1st, 2013 and May 31st, 2014. The models are estimated with a training sample (of size 3,828) which contains the responses in year 2 and the covariates in year 1. The performance of the estimated models are then assessed with a test sample (of size 4,076) which contains the responses in year 3 and the covariates in year 2. The following five covariates are used: 1) age of the member; 2) total number of transactions during the previous year for the member;

3) total amount spent during the previous year for the member; 4) time since the person is a member of the co-op; 5) category of the member (undergraduate student, master student, PhD student, etc.).

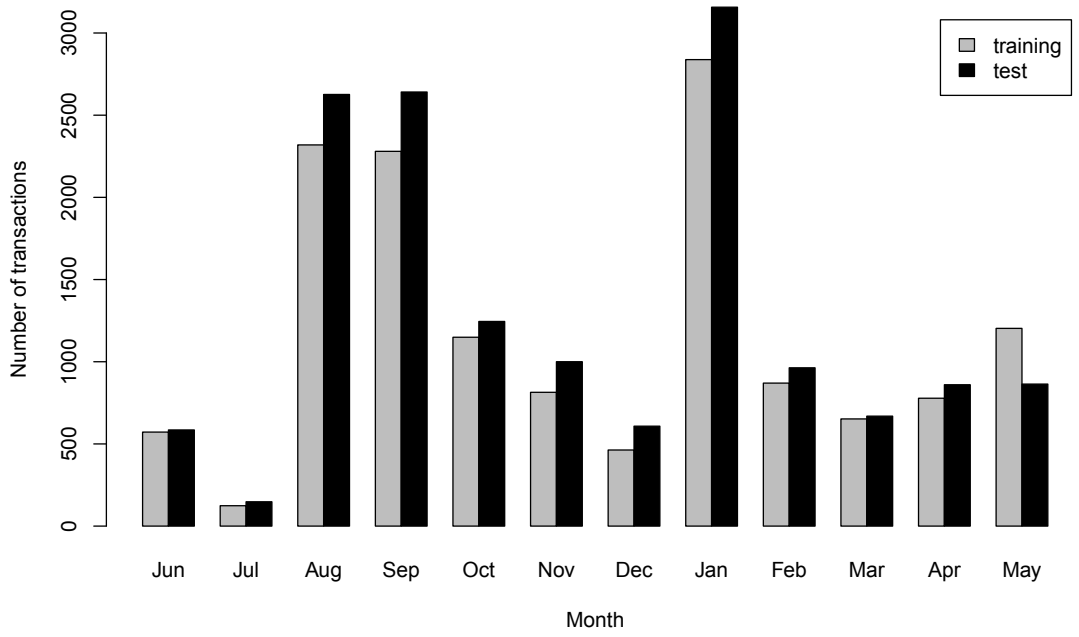


Figure 1.1 Number of transactions for the training and test samples in the co-op store example.

Figure 1.1 provides the monthly number of transactions for the training and test samples, which suggests clearly that the process is non–homogeneous. Homogeneous and non-homogeneous Poisson regressions and forests are compared. Here the homogeneous models simply pool the monthly response data into a single yearly response, while the non-homogeneous models use the monthly response data. The forests are built with 500 trees and all 5 covariates are retained at each node to find the best split. The PMSE over the 12 months for the test data set are provided in Table 1.2.

The results show clearly the need to use a non-homogeneous model in this case. In fact, we have an improvement, in terms of PMSE, of 13% for the forest and of 10% for the Poisson regression, when comparing the non–homogeneous (NH) model to the respective homogeneous (H) model. The best PMSE is obtained by the proposed NHPPF (PMSE=0.392).

Table 1.2 PMSE for the test data set in the co-op store example.

Model			
Poisson regression		Random forest	
H	NH	H	NH
0.454	0.406	0.453	0.392

However, the difference between the NHPPF and the piecewise Poisson regression model (PMSE=0.406) is small with an improvement of 3.4%.

1.5 Concluding remarks

In this paper, we introduced a nonparametric approach to model non–homogeneous count data. A piecewise constant Poisson process model was used to define a splitting criterion to build a non–homogeneous Poisson process tree (NHPPT) which then served as base learner to build a non–homogeneous Poisson process forest (NHPPF). The approach is quite flexible since the partition of the total time period into subperiods can be arbitrary. A simulation study involving different data generating processes was performed to explore the merits of the new approach compared to many competitors. The proposed NHPPF was either the best performer or very close to the best in all situations considered.

The primary use of the methods proposed in this paper is to obtain predictions and we used predefined subperiods because, in many applications, the chosen subperiods will be the ones where predictions are needed. For example, if predictions are needed at the monthly level for next year, then the subperiods will simply be the months and the forest will be built using the available monthly data in the preceding years. In that case, it is not clear that using shorter subperiods (e.g. weeks) and then aggregating the predictions at the monthly level would be beneficial. Indeed, using more subperiods entails more parameters to be estimated and hence more variability. Even if the process is not homogeneous within the months, the tree (forest) would still estimate the average expectation of the process over the month, and the predictions should still be accurate. Nevertheless, it could be interesting to investigate this aspect further. The subperiods could be selected as those that maximize the predictive accuracy over a validation sample or by cross–validation.

However, the focus might be on estimating the, assumed continuous, true rate function. In this case, the number of subperiods would likely increase with the sample size to get an increasingly smoother estimate. This problem is currently being investigated by the authors and will be part of another paper.

As mentioned in the Introduction, the proposed approach uses the CART paradigm of exhaustively searching for the best split by looking at all possible splits in each node to build the trees and hence the forest. This method is known to suffer from possible selection bias when building a single tree. However, it is not clear if the predictive performance of a forest built under this paradigm is adversely affected compared to that of a forest built with trees using an unbiased split criterion. It could be worthwhile to investigate this aspect. One possibility to modify the proposed method to mitigate this potential bias would be to select the best split at a given node in the following manner. First, we fit the piecewise Poisson regression models (Model 4 in the simulation study) with one covariate at a time with the observations of the current node. Second, we select the splitting covariate as the one with the smallest p-value when testing for no covariate effect. Third, we find the best split by performing an exhaustive search among all possible splits with the selected covariate only, and with the split criterion proposed in this paper.

The proposed approach is promising and this paper opens the way to many possible extensions and further investigations. Firstly, all the DGPs used in the simulation study were Poisson processes. It would be interesting to study the performance and robustness of the NHPPF when this no longer holds. Secondly, processes with extra–Poisson variation arising from clustered data occur often in practice. It would be interesting to generalize our approach to this setting. In a related direction, processes exhibiting an unusual number of zeros cannot always be captured appropriately by a Poisson process. Zero–inflated (ZIP) processes can then be used. It would be interesting to generalize our approach to handle such a situation. The NHPPF would then be a particular case of such a ZIP–forest. Thirdly, we only considered covariates that do not vary with time in this paper. Or if they do, we only considered their baseline values, for instance. It would be interesting to generalize our approach to be able to include time–varying covariates.

Acknowledgements

The authors would like to thank two referees for their constructive comments who helped us prepare an improved version of this article. They would also like to thank Joël Dusseault, the Director of the university store, for providing the data set used in the example. The authors acknowledge the financial support of NSERC and FRQNT.

References

• Biau, G., Devroye, L. and Lugosi, G. (2008). Consistency of Random Forests and Other Averaging Classifiers. *Journal of Machine Learning Research* **9**, 2015–2033.

• Biau, G. and Devroye, L. (2010). On the Layered Nearest Neighbour Estimate, the Bagged Nearest Neighbour Estimate and the Random Forest Method in Regression and Classification. *Journal of Multivariate Analysis* **101**, 2499–2518.

• Borisov, A., Runger, G., Tuv, E. and Lurponglukana–Strand, N. (2009). Zero–Inflated Boosted Ensembles for Rare Event Counts. N. Adams et al. (Eds.): IDA 2009, LNCS 5772 pp. 225–236. Springer–Verlag, Berlin.

• Breiman, L., Friedman, J., Olshen, R. and Stone, C. (1984). *Classification and Regression Trees*. Belmont, CA: Wadsworth International Group.

• Breiman, L. (2001). Random Forests. *Machine Learning* **45**, 5–32.

• Breiman, L. (1996). Bagging Predictors. *Machine Learning* **24**, 123–140.

• Chaudhuri, P., Lo, W.-D., Loh, W.-Y., Yang, C.-C. (1995). Generalized Regression Trees. *Statistica Sinica* **5**, 641–666.

• Choi, Y., Ahn, H. and Chen, J. J. (2005). Regression Trees for Analysis of Count Data with Extra Poisson Variation. *Computational Statistics & Data Analysis* **49**, 893–915.

• Cook, R. J. and Lawless, J. F. (2007). *The Statistical Analysis of Recurrent Events*. Springer. New York.

• Freund, Y. and Schapire, R. (1997). A Decision–theoretic Generalization of On–line Learning and Application to Boosting. *Journal of Computer and System Sciences* **55**, 119–139.

• Friedman, J. H. (2001). Greedy Functon Approximation: A Gradient Boosting Machine. *The Annals of Statistics* **29**, 1189–1232.

• Hamza, M. and Larocque, D. (2005). An Empirical Comparison of Ensemble Methods Based on Classification Trees. *Journal of Statistical Computation and Simulation* **75**, 629–

643.

• Lawless, J. F. and Zhan, M. (1998). Analysis of Interval–grouped Recurrent–event Data Using Piecewise Constant Rate Functions. *The Canadian Journal of Statistics* **26**, 549–565.

• Lee, S. K. (2005). On Generalized Multivariate Decision Tree by Using GEE. *Computational Statistics & Data Analysis*, **49**, 1105–1119.

• Lee, S. K. and Jin, S. (2006). Decision Tree Approaches for Zero–inflated Count Data. *Journal of Applied Statistics*, **33**, 853–865.

• Loh, W.-Y. (2002). Regression Trees With Unbiased Variable Selection and Interaction Detection. *Statistica Sinica* **12**, 361–386.

• Loh, W.-Y. (2011). Classification and Regression Trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **1**, 14–23.

• R Core Team (2013). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

• Rokach, L. (2008). Taxonomy for Characterizing Ensemble Methods in Classification Tasks: A Review and Annotated Bibliography. *Computational Statistics and Data Analysis* **53**, 4046–4072.

• Siroky, D.S. (2009). Navigating Random Forests and Related Advances in Algorithmic Modeling. *Statistics Surveys* **3**, 147–163.

• Therneau, T. M. and Atkinson, B. (2014). An Introduction to Recursive Partitioning Using the RPART Routines.
http://cran.r-project.org/web/packages/rpart/vignettes/longintro.pdf.

• Therneau, T. M., Atkinson, B. and Ripley, B. (2014). rpart: Recursive Partitioning and Regression Trees. R package version 4.1-8.
http://CRAN.R-project.org/package=rpart.

• Verikas, A., Gelzinis, A. and Bacauskiene, M. (2011). Mining Data With Random Forests: A Survey and Results of New Tests. *Pattern Recognition* **44**, 330–349.

Chapter 2

Random Forests for Non-homogeneous Poisson Processes with Excess Zeros

Abstract

We propose a method to build trees and forests when the response is a non-homogeneous Poisson process with excess zeros, based on two forests. The first one is used to estimate the probability of having a zero. The second forest is used to estimate the Poisson parameter using trees built with a splitting criterion derived from the zero truncated non-homogeneous Poisson likelihood. Simulation studies show that the proposed method performs well in hurdle (zero-altered) and zero-inflated settings.

Keywords: Hurdle model, Zero-altered Poisson (ZAP), Zero-inflated Poisson (ZIP), Non-homogeneous Poisson Process, Tree-based Method, Random Forests.

2.1 Introduction

It is often of interest to model the number of times that an event occurs during a given period of time. Such count data, taking only non-negative integer values are encountered in many situations. Some examples are the number of visits to the emergency unit in a hospital, the number or purchases made by a client, and the number of accidents on a road. Many parametric models are available to study the impact of covariates on the count response, including the Poisson and negative binomial models; see Cook and Lawless (2007) and Hilbe (2011). Sometimes, the number of zeros in the response is large and cannot be adequately

accounted by the usual models and the available covariates. Hurdle models, also called zero-altered models (Mullahy, 1986), and zero-inflated models (Lambert, 1992) were designed to handle such situations.

In this paper, we are interested in tree-based methods (Breiman *et al.*, 1984) and more particularly in random forests (Breiman, 2001). The main advantage of these methods is their flexibility, meaning that they can adapt to the data at hand without having to specify a parametric form. Within the CART (Classification and Regression Tree) paradigm, Poisson regression trees can be fitted in R (R Core Team 2015) with the package rpart (Therneau *et al.* 2014). The GUIDE approach (Loh, 2002) provides another way to build Poisson trees using the splitting rule of Chaudhuri *et al.* (1995). But these methods are not aimed at handling excess zeros in the response. Lee and Jin (2006) proposed a tree-based method for the excess zero case. They use the zero-inflated Poisson distribution to derive a splitting criterion. However, all the methods discussed above work under the basic assumption that the Poisson process generating these count data is homogeneous with respect to time. That is, the rate function of the response, given the covariates, does not vary with time. Mathlouthi, Fredette and Larocque (2015) developed tree-based methods for a response from a non–homogeneous Poisson process, but their method is not aimed at the excess zero case. In this paper, we present a method that extends both the Lee and Jin (2006) and Mathlouthi, Fredette and Larocque (2015) methods. We propose a method to build trees and forests for a non-homogeneous Poisson response with excess zeros. Hence, the method can handle simultaneously a non-homogeneous rate function and excess zeros in the response.

The paper is organized as follows. Section 2.2 describes the usual parametric models for a Poisson response with excess zeros. Section 2.3 describes the basic tree and forest methodology and the method of Lee and Jin (2006). Section 2.4 presents the proposed methods. The results from a simulation study are presented in Section 2.5. Concluding remarks and possibilities for future work are given in Section 2.6.

2.2 Zero-altered Poisson (ZAP) and Zero-inflated Poisson (ZIP) Regression Model

We are interested in modeling a count response Y with a set of q covariates $\mathbf{X} = (X_1, \ldots, X_q)'$. In Poisson hurdle, or zero-altered Poisson (ZAP), and zero-inflated Poisson (ZIP) models, it

is assumed that Y follows a Poisson distribution modified to account for an excess number of zeros. We will denote by λ, the parameter of the Poisson distribution, by p, the probability of having an excess (with respect to the Poisson distribution) 0, and by θ, the total probability of having a 0. These parameters can depend on the covariates and it will be clear form the context whether we are talking about the generic parameters or the covariate dependent parameters. We note that the Poisson variable used in this Section could also represent the total number of events obtained from a Poisson process. The parameter λ then represents the integral of the corresponding rate function over a given time interval.

In the ZAP regression model, it is assumed that $Y = 0$ with probability θ $(0 \leq \theta < 1)$, and that Y follows a zero truncated Poisson distribution with parameter λ, $(\lambda > 0)$, given that $Y > 0$. Consequently,

$$P(Y = y) = \begin{cases} \theta & y = 0 \\ \frac{(1-\theta)\exp(-\lambda)\lambda^y}{(1-\exp(-\lambda))y!} & y = 1, 2, \ldots \end{cases} \tag{2.1}$$

Many possibilities are available to link the covariates to the response, through λ and θ. The common link functions are given by:

$$\log(\lambda) = \mathbf{X}'\beta \quad \text{and} \quad \log\left(\frac{\theta}{1-\theta}\right) = \mathbf{X}'\gamma, \tag{2.2}$$

where β, γ are vectors of parameters to be estimated.

In the ZIP regression model, it is assumed that $Y = 0$ with probability p, $(0 \leq p < 1)$ and that Y follows a Poisson distribution with parameter λ, $(\lambda > 0)$, with probability $1 - p$. Hence, there are two sources for the zeros. Consequently,

$$P(Y = y) = \begin{cases} p + (1-p)\exp(-\lambda) & y = 0 \\ (1-p)\exp(-\lambda)\lambda^y/y! & y = 1, 2, \ldots \end{cases} \tag{2.3}$$

Again, many possibilities are available to link the covariates to the response. The common link functions are the same as for the ZAP model:

$$\log(\lambda) = \mathbf{X}'\beta \quad \text{and} \quad \log\left(\frac{p}{1-p}\right) = \mathbf{X}'\gamma, \tag{2.4}$$

where β, γ are vectors of parameters to be estimated. Note that different covariates can be used for the (excess) zero part and the Poisson part of these models, but we use this notation for simplicity.

Note that, without covariates, models (2.1) and (2.3) are just two parameterizations of the same model. The difference between the two approaches lies in the way the covariates are linked to the parameters. In the ZAP model, the covariates are linked directly to the total probability of having a 0 while, for the ZIP model, they are linked directly to the probability of having an excess 0 with respect to the Poisson distribution.

One key observation is that the ZAP model is formed by two models. Consequently, the covariates can have different effects on the zero and the Poisson parts. The same is true for the ZIP model, except that this time, the covariates can have different effects on the excess zero and the Poisson parts.

Assume that a sample of size n is available. Namely, $(Y_i, \mathbf{X}_i)$, $i = 1, \ldots, n$, where $\mathbf{X}_i = (X_{1i}, \ldots, X_{qi})'$. Then the parameters of the ZAP and ZIP regression models can be estimated with maximum likelihood. It is straightforward to do it with the ZAP model because the likelihood for the zero and the zero truncated Poisson parts can be factored. Hence, the two models can be fitted separately. Things are a slightly more complex with the ZIP model but Lambert (1992) proposed an estimation method based on the EM algorithm.

2.3 Trees and Forests

2.3.1 Basic Tree and Forest Methodology

We assume the reader is familiar with the CART paradigm (Breiman *et al.*, 1984) as only a brief description is given here. Tree-based methods partition the covariate space by splitting it recursively with rules based on covariates. The basic ingredient for building a tree is the splitting criterion, which is problem dependent. For example, the least-squares splitting

criterion is the usual one when the response is continuous. Suppose we are at a given node t and we want to split it into two children nodes, t_L (left node) and t_R (right node). The best split is chosen among all possible binary splits obtained from a covariate. If x is continuous (or at least ordinal), the possible splits take the form $I(x \leq c)$. If x is categorical, the possible splits take the form $x \in \{c_1, \ldots, c_s\}$ where $\{c_1, \ldots, c_s\}$ is a subset of the possible values of x. The best split is the one maximizing the splitting criterion. If a single tree is required, then the usual procedure builds a large tree and then uses a pruning algorithm to avoid over-fitting. However, it is now well-established that using an ensemble of trees is generally preferable to a single tree. One of the most popular ensemble method is random forests, introduced by Breiman (2001). Here we describe the generic random forest algorithm that will be used in this paper:

1. Draw B bootstrap samples from the original data.

2. For each bootstrap sample, grow a tree with the selected splitting criterion. At each node, randomly select q_0 out of q covariates where $q_0 \leq q$ and is a user-specified parameter. Splitting ends when a stopping criterion is reached; for instance, when a node has less than a predetermined number of observations. No pruning is performed.

3. To obtain an estimation of a parameter (or a prediction) for a new observation, take the average estimates (predictions) from the B trees.

2.3.2 Maximum Likelihood Splitting Criterion

A simple method for deriving a splitting criterion is to use the log-likelihood of an adequate two-node model; see Su, Wang and Fan (2004) and Bou-Hamad *et al.* (2009) for some examples. Basically, the best split at a given node is the one that maximizes the observed log-likelihood, i.e. the one evaluated at the maximum likelihood estimates, among all allowable splits. Moreover, if the parameters are estimated separately in the two children nodes, then the best split is the one that maximizes

$$\widehat{LL}(\text{left node}) + \widehat{LL}(\text{right node}), \tag{2.5}$$

where $\widehat{LL}$(left node) and $\widehat{LL}$(right node) are the observed log-likelihood in the left and right nodes, respectively.

2.3.3 ZIP Tree of Lee and Jin (2006)

Lee and Jin (2006) proposed a decision tree method for zero-inflated count data based on the CART paradigm. They call it a ZIP tree. They basically fit the ZIP distribution (2.3) separately in the two children nodes, and use (2.5) as the splitting criterion. Let N^+ denote the number of observations such that $Y_i > 0$. The log-likelihood function in that case is

$$\begin{aligned} LL_{ZIP} = (n - N^+)\log(p + (1-p)\exp(-\lambda)) + N^+(\log(1-p) - \lambda) \\ + \sum_{Y_i>0} Y_i \log(\lambda) - \sum_{Y_i>0} \log(Y_i!). \end{aligned} \tag{2.6}$$

One crucial observation is that the covariates are used to find the tree structure for both the excess zero part and the Poisson part jointly. Hence a single model is used to model both parts and a single tree is built. But as we just saw, the ZIP regression model (2.3) uses two models, one for the excess zero part and one for the Poisson part. Hence the covariates can have different effects on both parts. This is why we propose in this paper a new approach, more in the spirit of the ZIP and ZAP models, where two models are used.

2.4 Random Forests for Poisson Data With Excess Zeros

In the ZAP model, assume that, instead of a rigid parametric model like (2.2), we use a general nonparametric model given by

$$\theta = f_\theta(\mathbf{X}) \quad \text{and} \quad \lambda = f_\lambda(\mathbf{X}), \tag{2.7}$$

where f_θ and f_λ are general unknown link functions. Similarly, assume a same general setup for the ZIP model, given by

$$p = g_p(\mathbf{X}) \quad \text{and} \quad \lambda = g_\lambda(\mathbf{X}), \tag{2.8}$$

where g_p and g_λ are general unknown link functions. Since $\theta = g_p(\mathbf{X}) + (1 - g_p(\mathbf{X}))\exp(-g_\lambda(\mathbf{X}))$, we see that in this general nonparametric framework, the ZIP and ZAP models are again only different parameterizations of the same model. Indeed, we can just define $f_\theta(\mathbf{X}) = g_p(\mathbf{X}) + (1 - g_p(\mathbf{X}))\exp(-g_\lambda(\mathbf{X}))$ in (2.7). Hence, it does not matter whether we specify model (2.7) or (2.8). Namely, if a general nonparametric and flexible procedure is used to estimate f_θ and f_λ in (2.7), it can be used to obtain estimates

$$\hat{\theta} = \hat{f}_\theta(\mathbf{X}) \quad \text{and} \quad \hat{\lambda} = \hat{f}_\lambda(\mathbf{X}), \tag{2.9}$$

for a given value of $\mathbf{X}$. But it can also be used to estimate p through the equation $\hat{\theta} = \hat{p} + (1 - \hat{p})\exp(-\hat{\lambda})$. Solving for $\hat{p}$ gives $\hat{p} = (\hat{\theta} - \exp(-\hat{\lambda}))/(1 - \exp(-\hat{\lambda}))$. However, since this value can be less than 0, we will use $\hat{p} = \max(0, (\hat{\theta} - \exp(-\hat{\lambda}))/(1 - \exp(-\hat{\lambda})))$, in this paper. The key point is that the method proposed in this paper is valid for both the ZAP and the ZIP settings.

2.4.1 Description of the Basic Method

The basic idea is to fit two random forests, one for the zero part to estimate f_θ, and one for the observations that are greater than 0 to estimate f_λ. More specifically, for the zero part, the response is $I(Y > 0)$, that is the binary variable taking a value of 1 if $Y > 0$ and the value 0 if $Y = 0$. This is a standard problem and many implementations are available in R, for example through the packages `randomForest` (Liaw and Wiener, 2002) and `randomForestSRC` (Ishwaran, Kogalur, Blackstone and Lauer 2008, Ishwaran and Kogalur, 2015). For the observations that are greater than 0, we propose a forest of trees built using a splitting criterion derived from the zero truncated Poisson likelihood. Only the observations where $Y > 0$ are used. Assume there are N^+ such observations denoted by $Y_1^+, \ldots, Y_{N^+}^+$. The probability mass function from the truncated Poisson distribution is

$$P(Y^+ = y) = P(Y = y | Y > 0) = \frac{\exp(-\lambda)\lambda^y}{y!(1 - \exp(-\lambda))} \quad y = 1, 2, \ldots \tag{2.10}$$

Hence, the log-likelihood function for the sample is

$$LL^+ = -N^+ \log(1 - \exp(-\lambda)) + \log(\lambda) \sum_{i=1}^{N^+} Y_i^+ - N^+\lambda - \sum_{i=1}^{N^+} \log(Y_i^+!). \tag{2.11}$$

The estimated λ is obtained by solving $\partial LL^+/\partial\lambda = 0$ which reduces to

$$\frac{\sum_{i=1}^{N^+} Y_i^+}{N^+} = \frac{\lambda}{1 - \exp(-\lambda)}. \tag{2.12}$$

For a given candidate split, the zero truncated Poisson model is fitted separately in the two children nodes and the splitting criterion is given by (2.5) with (2.11) as the log-likelihood function.

2.4.2 Extension to the Non-Homogeneous Case

Mathlouthi, Fredette and Larocque (2015) proposed tree and random forest methods for non-homogeneous Poisson processes. It was achieved by considering a model with a piecewise constant rate function. Here we extend this method to the case of a non-homogeneous Poisson process with excess zeros. We are again interested in a count response but this time we want to allow the rate function to vary over time. Assume we have a fixed time period T and assume that it is partitioned into K subperiods, $T_1, \ldots, T_K$, such that

$$\bigcup_{k=1}^{K} T_k = T \quad \text{and} \quad T_i \cap T_j = \emptyset \quad \text{for all} \quad i \neq j.$$

Each subperiod T_k may be an interval or a finite union of disjoint intervals. Denote by N_k the number of events in subperiod T_k. We assume that N_k follows a Poisson distribution with parameter λ_k and that all the N_k's are independent. The K subperiods can be adjacent time intervals $(a_0, a_1], (a_1, a_2], \ldots, (a_{K-1}, a_K]$ covering the whole period $T = (a_0, a_K]$. But the above formulation is more general than that and the subperiods can represent non–adjacent periods. For example, they could represent days (T_1=all the Mondays in a year, T_2=all the Tuesdays in a year and so on). The total number of events observed over T is denoted by Y,

that is

$$Y = \sum_{k=1}^{K} N_k.$$

We are interested in allowing Y to have excess zeros with respect to the non-homogeneous Poisson process described above. Once again, the idea is to fit two random forests, one to estimate $P(Y = 0)$, that is the probability that no events at all occurred, and one for the observations with at least one event. For the zero part, the binary response is $I(Y > 0)$, which can be fitted by standard algorithm as described in the preceding section. For the observations with at least one event, we propose a forest of trees built using a splitting criterion derived from a zero truncated non-homogeneous Poisson model likelihood, as described next. Let $(n_1, \ldots, n_K)$ be non-negative integers such that at least one of them is greater than 0. The joint probability mass function of $(N_1, \ldots, N_K)$ given that at least one event occurred is

$$P(N_1 = n_1, \ldots, N_K = n_k | Y > 0) = \frac{1}{(1 - \exp(-\lambda))} \prod_{k=1}^{K} \exp(-\lambda_k) \lambda_k^{n_k} / n_k! \tag{2.13}$$

where $\lambda = \sum_{k=1}^{K} \lambda_k$.

For a sample, $(N_{i1}, \ldots, N_{iK})$, $i = 1, \ldots, N$, assume we have N^+ observations such that $Y_i = \sum_{k=1}^{K} N_{ik} > 0$, then the log-likelihood function for those observations is

$$LL_{NH}^{+} = -N^+ \log(1 - \exp(-\lambda)) + \sum_{k=1}^{K} \log(\lambda_k) \sum_{i=1}^{N^+} N_{ik} - N^+ \lambda - \sum_{k=1}^{K} \sum_{i=1}^{N^+} \log(N_{ik}!). \tag{2.14}$$

Note that if we have a single time period, i.e., $K = 1$, then $Y_i = N_{i1}$ and we fall back to the setting of Section 2.4.1. The estimated λ_k's are obtained by solving the K equations $\partial LL_{NH}^{+} / \partial \lambda_k = 0$, giving

$$\sum_{i=1}^{N^+} \frac{N_{ik}}{N^+} = \frac{\lambda_k}{1 - \exp\left(-\sum_{j=1}^{K} \lambda_j\right)}, k = 1, \ldots, K.$$

This system can be solved with the Newton-Raphson algorithm. For a given candidate split, the zero truncated non-homogeneous Poisson model is fitted separately in the two children nodes and the splitting criterion is given by (2.5) with (2.14) as the log-likelihood function.

2.5 Simulation Study

In this section, we investigate the performance of the proposed method compared to various competitors including parametric models and forests. In the first set of simulations, the data generating process (DGP) is homogeneous. This will allow comparisons with the usual parametric ZIP and ZAP models, with a forest of Poisson trees but also with a forest built with the Lee and Jin (2006) ZIP tree approach. Then, in the second set of simulations, non-homogeneous DGPs will be considered. This will allow a comparison with the approach of Mathlouthi *et al.* (2015).

2.5.1 Description of the Simulation Study

In all cases, nine covariates $X_1, ..., X_9$, are available. They are independent and uniformly distributed over the interval $[0, 10]$. X_1, X_2, and X_3 are related to the Poisson intensity parameter, while X_1, X_4, and X_5 are related to the zero part of the model. Hence X_6, X_7, X_8 and X_9 are noise covariates unrelated to the outcome.

DGPs from ZAP and ZIP settings with varying proportions of zeros are considered. Consider first the following logistic regression DGP that will be used to generate either the zeros (ZAP) or excess zeros (ZIP):

DGP zero

$$\log\left(\frac{1-\tau}{\tau}\right) = c - 3\log(X_1 + 0.5) + 0.2(10X_4 - X_4^2) + 0.4X_5.$$

The choices of intercepts $c = 2.6$, $c = 0.55$ and $c = -1.1$ produce approximately 15%, 35% and 55% of zeros for this DGP. The parameter τ represents either θ, the total probability of having a 0, for a ZAP DGP or p, the probability of having an excess 0, for a ZIP DGP.

Consider the following three models for the Poisson part of the outcome, governed by the parameter λ.

DGP A: Poisson model with main effects only

$$\ln(\lambda) = -0.105 + 0.1X_1 - 0.1X_2 + 0.1X_3.$$

DGP B: Poisson model with more complicated effects

$$\ln(\lambda) = -0.7 + 0.05X_1 + 2(X_2 > 5) + 0.05(10X_3 - X_3^2) + 0.04(X_2 > 5)(X_1 - 5)^2.$$

DGP C: Poisson model with a tree structure

Leaf 1. If $X_1 \leq 5$ and $X_2 \leq 5$ then $\lambda = 1.5$.

Leaf 2. If $X_1 \leq 5$ and $X_2 > 5$ then $\lambda = 3.0$.

Leaf 3. If $X_1 > 5$ and $X_3 \leq 5$ then $\lambda = 2.5$.

Leaf 4. If $X_1 > 5$ and $X_3 > 5$ then $\lambda = 2.0$.

Nine scenarios are considered for the ZAP DGPs, by crossing the three Poisson DGPs with the three different probabilities of zero. For these scenarios, a binary outcome is first generated from DGP zero. If a 0 is generated, then this is the value of Y. If not, then Y is generated by using the Poisson model (either A, B or C) but truncated at zero. Indeed, for a ZAP DGP, the total probability of having a 0 is governed only by DGP zero. Hence, for these scenarios, the total proportions of zeros are going to be approximately 15%, 35% or 55%.

Nine scenarios are also considered for the ZIP DGPs, again by crossing the three Poisson DGPs with the three different probabilities of zero. For these scenarios, a binary outcome is first generated from DGP zero. If a 0 is generated, then this is the value of Y. If not then Y is generated by using the Poisson model (either A, B or C). This time, a 0 can also be generated from the Poisson part. The binary outcome model only generates excess zeros with respect to the Poisson model. Hence, for these scenarios, the total proportions of zeros are going to be higher than 15%, 35% or 55%.

Nine other scenarios are considered for the non-homogeneous ZAP case. Using the notation of Section 2.4.2, we have $K = 12$ subperiods. The probability that no event at all occurred is still given by DGP zero above. The three following non-homogeneous Poisson DPGs are used for the Poisson part. This time, 12 parameters, $\mathbf{\Lambda} = (\lambda_1, \ldots, \lambda_{12})$ are required.

DGP D: Non-Homogeneous Poisson model with main effects only

$$\ln(\lambda_k) = \log(0.1 * k) - 0.3 + 0.1X_1 - 0.1X_2 + 0.1X_3, k = 1, \ldots, 12.$$

DGP E: Non-Homogeneous Poisson model with more complicated effects

$$\ln(\lambda_k) = -0.5*k+0.05X_1+2(X_2 > 5)+0.05(10X_3-X_3^2)+0.04(X_2 > 5)(X_1-5)^2, k = 1, \ldots, 12.$$

DGP F: Non-Homogeneous Poisson model with a tree structure

Leaf 1. If $X_1 \leq 5$ and $X_2 \leq 5$ then $\mathbf{\Lambda} = (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2)$.

Leaf 2. If $X_1 \leq 5$ and $X_2 > 5$ then $\mathbf{\Lambda} = (1.2, 1.2, 1.2, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 1.2, 1.2, 1.2)$.

Leaf 3. If $X_1 > 5$ and $X_3 \leq 5$ then $\mathbf{\Lambda} = (0.1, 0.1, 0.1, 1.2, 1.2, 1.2, 1.2, 1.2, 1.2, 0.1, 0.1, 0.1)$.

Leaf 4. If $X_1 > 5$ and $X_3 > 5$ then $\mathbf{\Lambda} = (1.2, 1.1, 1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1)$.

For the homogeneous DGP's, five models are compared. They are

1. Parametric ZAP model described in (2.1) and (2.2). The nine covariates are used as main effects only both for the zero and Poisson parts of the model.
2. Parametric ZIP model described in (2.3) and (2.4). The nine covariates are used as main effects only both for the excess zero and Poisson parts of the model.
3. Poisson forest. A forest of basic Poisson trees is built.
4. Lee-Jin forest. A forest with the ZIP tree approach of Lee and Jin (2006).
5. Proposed approach, the ZAP forest (Section 2.4.1).

For the non-homogeneous DGP's, two models are compared. They are the NHPPF method of Mathlouthi *et al.* (2015), and the proposed non-homogeneous ZAP forest (Section 4.2).

The parametric models were fitted with the R package `pcsl` (Jackman, 2015, and Zeileis, Kleiber and Jackman, 2008). The Poisson parts of the forests were implemented in C. However, the R package `randomForest` was used to build the forest for the zero part of the proposed ZAP forest. All forests are built with 500 trees. Three out of the nine covariates

are randomly selected at each node of each tree. This comes from the value $\sqrt{q}$ typically used to build a regression random forest. Thirty observations are needed to attempt splitting and the resulting nodes must have at least ten observations. The models are estimated with a training sample of size 1000. Parameter estimates are then obtained for each observation in a test sample of size 5000. The number of simulation runs is 100.

The mean absolute error (MAE) is used as the performance criterion. It is defined by

$$\text{MAE} = \frac{1}{T}\sum_{t=1}^{T} |\gamma_t - \hat{\gamma}_t|,$$

where T is the size of the test set (5000 here), γ_t and $\hat{\gamma}_t$ are the true and estimated values of the parameter of interest for the t^{th} test observation. Here γ represents one of the three parameters of interests which are λ, θ and p.

2.5.2 Results

The results are presented in Tables 2.1 to 2.3. Table 2.1 presents the average MAE, over the 100 simulation runs, of all methods for the three parameters of interests, λ, θ, p, for the nine ZAP DGPs. For each line in the table, the value of the best (smallest) MAE is in bold. The values between parentheses are the standard deviations of the MAE over the simulation runs. It is striking that the proposed method has the smallest average MAE in all but three of the 27 cases. For DGP A, the Poisson part is a main effect DGP. Hence, the Poisson part of the parametric ZAP model contains the true effects. It is then not surprising that it has the smallest MAE for estimating λ whatever the proportion of zeros is. However, the zero part of the parametric ZAP model is not well-specified and thus the proposed method is better to estimate both θ and p. The Lee-Jin forest is better than a Poisson forest for estimating λ and p but the opposite is true for estimating θ.

Table 2.2 presents the MAE for the nine ZIP DGPs. This time, the proposed method has the smallest average MAE in 20 cases and is close to the best one in four other cases. Similarly to what we saw with the ZAP DGPs, the parametric model is the best one when the Poisson part of the model contains main effects only, that is for DGP A. The parametric ZIP model is then the best one for estimating λ in these three cases (with either 15%, 35%

and 55% of excess zeros). There are four instances where the Lee-Jin forest is slightly better than the proposed method. But apart from that, the proposed method is globally the best one for the DGPs considered.

Table 2.3 presents the results for the non-homogeneous ZAP DGPs. This time only the proposed method and the NHPPF method of Mathlouthi *et al.* (2015) are compared. The NHPPF fits a forest of non-homogeneous Poisson trees but does not account for excess zeros. The proposed method has the smallest average MAE in all 27 cases. This, combined with the fact that it is also always better than a Poisson forest in Tables 2.1 and 2.2, clearly shows the importance of modeling the potential excess zeros.

2.6 Concluding remarks

We proposed a method to build trees and forests for a non-homogeneous Poisson response with excess zeros, based on two forests. Unlike the ZIP tree proposed by Lee and Jin (2006), our method has two parts, like the usual parametric ZAP and ZIP models. This allows for different covariate effects for the zero part and the Poisson part. Any flexible method to model the probability for a binary response can be used for the zero part. Here we used the traditional random forest for a binary response. For the Poisson part, we used a forest of trees built with a splitting criterion derived from the zero truncated non-homogeneous Poisson likelihood. Our method extends the work of Lee and Jin (2006) in the sense that it can handle a non-homogeneous rate function. Our method also extends the work of Mathlouthi, Fredette and Larocque (2015) by allowing for excess zeros.

The results from extensive simulation studies clearly show the merits of the proposed method. A mix of homogeneous and non-homogeneous ZAP and ZIP models were used to generate artificial data. The proposed method was compared to parametric ZAP and ZIP models, to a basic Poisson forest and to a forest built with the Lee and Jin (2006) ZIP tree approach. The proposed method had the smallest mean absolute error for 71 out of the 81 estimation problems considered, and it was a close second for four others. The six cases where the proposed method was not the best or a close second was for estimating the Poisson intensity when the parametric model was correctly specified for this parameter.

Our approach is very general and many extensions are possible. Firstly, if the Poisson assumption is not appropriate, other models, like the negative binomial, could be used. This

could be useful in cases where extra-Poisson variation is observed, for instance with clustered data. Secondly, we only considered covariates that do not vary with time. It would be interesting to generalize our approach to be able to include time-varying covariates.

Acknowledgements

The authors acknowledge the financial support of NSERC and FRQNT.

References

• Bou-Hamad, I., Larocque, D., Ben-Hameur, H., Mâsse, L. C., Vitaro, F. and Tremblay, R. E. (2009). Discrete-Time Survival Trees. *Canadian Journal of Statistics* **37**, 17–32.

• Breiman, L., Friedman, J., Olshen, R. and Stone, C. (1984). *Classification and Regression Trees*. Belmont, CA: Wadsworth International Group.

• Breiman, L. (2001). Random Forests. *Machine Learning* **45**, 5–32.

• Chaudhuri, P., Lo, W.-D., Loh, W.-Y., Yang, C.-C. (1995). Generalized Regression Trees. *Statistica Sinica* **5**, 641–666.

• Cook, R. J. and Lawless, J. F. (2007). *The Statistical Analysis of Recurrent Events*. Springer. New York.

• Hilbe, J. M. (2011). *Negative Binomial Regression*, 2nd edition. Cambridge University Press. Cambridge.

Ishwaran H., Kogalur U. B., Blackstone E. H. and Lauer M. S. (2008). Random Survival Forests. *Annals of Applied Statistics* **2**, 841–860.

• Ishwaran H. and Kogalur U. B. (2015). Random Forests for Survival, Regression and Classification (RF-SRC), R package version 1.6.1.
URL http://cran.r-project.org/web/packages/randomForestSRC/index.html.

• Jackman, S. (2015). pscl: Classes and Methods for R Developed in the Political Science Computational Laboratory, Stanford University. Department of Political Science, Stanford University. Stanford, California. R package version 1.4.8.
URL http://pscl.stanford.edu/.

• Lambert, D. (1992). Zero-Inflated Poisson Regression, with an Application to Defects in Manufacturing. *Technometrics* , **34**, 1–14.

• Lee, S. K. and Jin, S. (2006). Decision Tree Approaches for Zero–inflated Count Data. *Journal of Applied Statistics*, **33**, 853–865.

• Liaw, A. and Wiener, M. (2002). Classification and Regression by randomForest. *R News* **2**, 18–22.

• Loh, W.-Y. (2002). Regression Trees With Unbiased Variable Selection and Interaction Detection. *Statistica Sinica* **12**, 361–386.

• Mathlouthi, W., Fredette, M. and Larocque, D. (2015). Regression Trees and Forests for Non-Homogeneous Poisson Processes. *Statistics and Probability Letters* **96**, 204–211.

• Mullahy, J. (1986). Specification and Testing of Some Modified Count Data Models. *Journal of Econometrics*, **33**, 341–365.

• R Core Team (2015). *R: A language and environment for statistical computing. R Foundation for Statistical Computing*, Vienna, Austria.
URL http://www.R-project.org/.

• Su, X., Wang, M. and Fan, J. (2004). Maximum Likelihood Regression Trees. *Journal of Computational and Graphical Statistics*, **13**, 586–598.

• Therneau, T. M., Atkinson, B. and Ripley, B. (2014). rpart: Recursive Partitioning and Regression Trees. R package version 4.1-8.
URL http://CRAN.R-project.org/package=rpart.

• Zeileis, A., Kleiber, C. and Jackman, S. (2008). Regression Models for Count Data in R. *Journal of Statistical Software* **27**, Issue 8, 1–25.

Table 2.1 Simulation results for the homogeneous ZAP DGPs. The average MAE are reported for the three parameters of interest: λ, the Poisson intensity; θ, the probability of zero; p, the probability of an excess zero. Standard deviations of the MAE are reported between parentheses. The smallest value of the MAE for a given scenario is in bold. In the first column, the percentage corresponds to the total probability of having a 0.

DGP	Parameter	Parametric Zap	Parametric Zip	Poisson forest	Lee-Jin forest	Zap Forest
A(15%)	λ	**0.1304** (0.0295)	0.3984 (0.0365)	0.5222 (0.0261)	0.3989 (0.0315)	0.2703 (0.0244)
	θ	0.1190 (0.0040)	0.1787 (0.0143)	0.1714 (0.0044)	0.2136 (0.0072)	**0.0719** (0.0045)
	p	0.0889 (0.0032)	0.0799 (0.0117)	0.0991 (0.0027)	0.0817 (0.0030)	**0.0496** (0.0039)
A(35%)	λ	**0.1651** (0.0402)	0.3470 (0.0661)	0.7264 (0.0295)	0.4365 (0.0429)	0.3260 (0.0368)
	θ	0.1741 (0.0032)	0.2163 (0.0084)	0.1994 (0.0078)	0.2303 (0.0066)	**0.1016** (0.0055)
	p	0.1575 (0.0034)	0.1672 (0.0040)	0.2738 (0.0048)	0.1820 (0.0076)	**0.0927** (0.0064)
A(55%)	λ	**0.2098** (0.0506)	0.3260 (0.0488)	1.0019 (0.0313)	0.6341 (0.0584)	0.4397 (0.0565)
	θ	0.1656 (0.0028)	0.1916 (0.0036)	0.1863 (0.0099)	0.2489 (0.0097)	**0.1049** (0.0050)
	p	0.1653 (0.0030)	0.1772 (0.0037)	0.4689 (0.0054)	0.3285 (0.0134)	**0.1165** (0.0081)
B(15%)	λ	0.4748 (0.0167)	0.5919 (0.0220)	0.4672 (0.0261)	0.2994 (0.0343)	**0.2857** (0.0249)
	θ	0.1190 (0.0040)	0.1447 (0.0097)	0.1551 (0.0051)	0.1992 (0.0081)	**0.0718** (0.0046)
	p	0.0852 (0.0033)	0.0671 (0.0072)	0.0906 (0.0026)	0.0747 (0.0027)	**0.0490** (0.0038)
B(35%)	λ	0.4868 (0.0232)	0.5639 (0.0491)	0.6225 (0.0284)	0.3615 (0.0348)	**0.3243** (0.0315)
	θ	0.1741 (0.0032)	0.1848 (0.0181)	0.1729 (0.0078)	0.2043 (0.0067)	**0.1015** (0.0055)
	p	0.1678 (0.0039)	0.1716 (0.0063)	0.2649 (0.0047)	0.1807 (0.0074)	**0.0979** (0.0065)
B(55%)	λ	0.5095 (0.0292)	0.5374 (0.0414)	0.8870 (0.0302)	0.5356 (0.0496)	**0.4008** (0.0420)
	θ	0.1656 (0.0028)	0.1754 (0.0038)	0.1659 (0.0094)	0.2254 (0.0089)	**0.1050** (0.0050)
	p	0.1813 (0.0035)	0.1885 (0.0089)	0.4682 (0.0054)	0.3314 (0.0116)	**0.1233** (0.0084)
C(15%)	λ	0.4748 (0.0167)	0.5919 (0.0220)	0.4672 (0.0261)	0.2994 (0.0343)	**0.2857** (0.0249)
	θ	0.1190 (0.0040)	0.1447 (0.0097)	0.1551 (0.0051)	0.1992 (0.0081)	**0.0718** (0.0046)
	p	0.0852 (0.0033)	0.0671 (0.0072)	0.0906 (0.0026)	0.0747 (0.0027)	**0.0490** (0.0038)
C(35%)	λ	0.4868 (0.0232)	0.5639 (0.0491)	0.6225 (0.0284)	0.3615 (0.0348)	**0.3243** (0.0315)
	θ	0.1741 (0.0032)	0.1848 (0.0181)	0.1729 (0.0078)	0.2043 (0.0067)	**0.1015** (0.0055)
	p	0.1678 (0.0039)	0.1716 (0.0063)	0.2649 (0.0047)	0.1807 (0.0074)	**0.0979** (0.0065)
C(55%)	λ	0.5095 (0.0292)	0.5374 (0.0414)	0.8870 (0.0302)	0.5356 (0.0496)	**0.4008** (0.0420)
	θ	0.1656 (0.0028)	0.1754 (0.0038)	0.1659 (0.0094)	0.2254 (0.0089)	**0.1050** (0.0050)
	p	0.1813 (0.0035)	0.1885 (0.0089)	0.4682 (0.0054)	0.3314 (0.0116)	**0.1233** (0.0084)

Table 2.2 Simulation results for the homogeneous ZIP DGPs. The average MAE are reported for the three parameters of interest: λ, the Poisson intensity; θ, the probability of zero; p, the probability of an excess zero. Standard deviations of the MAE are reported between parentheses. The smallest value of the MAE for a given scenario is in bold. In the first column, the percentage corresponds to the probability of having an excess 0.

DGP	Parameter	Parametric Zap	Parametric Zip	Poisson forest	Lee-Jin forest	Zap Forest
A(15%)	λ	0.1469 (0.0348)	**0.1467** (0.0480)	0.4248 (0.0326)	0.3067 (0.0234)	0.2857 (0.0263)
	θ	0.1237 (0.0047)	0.1022 (0.0075)	0.1044 (0.0057)	**0.1014** (0.0053)	0.1035 (0.0052)
	p	0.1401 (0.0110)	0.1237 (0.0107)	0.1520 (0.0029)	0.1225 (0.0086)	**0.1212** (0.0110)
A(35%)	λ	0.1817 (0.0480)	**0.1783** (0.0460)	0.7471 (0.0330)	0.4152 (0.0433)	0.3474 (0.0419)
	θ	0.1565 (0.0028)	0.1398 (0.0044)	0.1470 (0.0083)	0.1511 (0.0053)	**0.1082** (0.0059)
	p	0.1881 (0.0072)	0.1799 (0.0082)	0.3515 (0.0045)	0.2184 (0.0094)	**0.1437** (0.0101)
A(55%)	λ	0.2385 (0.0594)	**0.2384** (0.0578)	1.0627 (0.0336)	0.6572 (0.0614)	0.4721 (0.0614)
	θ	0.1353 (0.0024)	0.1268 (0.0035)	0.1427 (0.0108)	0.1947 (0.0107)	**0.0992** (0.0056)
	p	0.1729 (0.0054)	0.1710 (0.0064)	0.5493 (0.0048)	0.3715 (0.0141)	**0.1485** (0.0102)
B(15%)	λ	0.4999 (0.0296)	0.5015 (0.0277)	0.3880 (0.0270)	**0.3002** (0.0236)	0.3062 (0.0252)
	θ	0.1259 (0.0040)	0.1184 (0.0057)	0.0992 (0.0056)	0.0982 (0.0060)	**0.0976** (0.0053)
	p	0.1494 (0.0134)	0.1397 (0.0166)	0.1520 (0.0029)	0.1188 (0.0088)	**0.1153** (0.0106)
B(35%)	λ	0.5109 (0.0328)	0.5127 (0.0325)	0.6636 (0.0325)	0.3833 (0.0357)	**0.3446** (0.0314)
	θ	0.1544 (0.0026)	0.1472 (0.0031)	0.1363 (0.0092)	0.1458 (0.0055)	**0.1059** (0.0058)
	p	0.1918 (0.0083)	0.1875 (0.0088)	0.3515 (0.0045)	0.2194 (0.0107)	**0.1406** (0.0093)
B(55%)	λ	0.5310 (0.0332)	0.5368 (0.0352)	0.9655 (0.0334)	0.5822 (0.0514)	**0.4230** (0.0430)
	θ	0.1386 (0.0021)	0.1341 (0.0022)	0.1385 (0.0104)	0.1927 (0.0104)	**0.1011** (0.0055)
	p	0.1750 (0.0051)	0.1731 (0.0052)	0.5493 (0.0048)	0.3741 (0.0145)	**0.1454** (0.0115)
C(15%)	λ	0.4124 (0.0124)	0.4158 (0.0183)	0.4583 (0.0322)	**0.2988** (0.0271)	0.3210 (0.0265)
	θ	0.1303 (0.0040)	0.1221 (0.0063)	0.1049 (0.0054)	**0.0941** (0.0053)	0.0956 (0.0057)
	p	0.1339 (0.0082)	0.1253 (0.0092)	0.1520 (0.0029)	0.1033 (0.0068)	**0.0983** (0.0078)
C(35%)	λ	0.4274 (0.0185)	0.4305 (0.0192)	0.8384 (0.0368)	0.4070 (0.0415)	**0.3664** (0.0303)
	θ	0.1660 (0.0026)	0.1596 (0.0030)	0.1630 (0.0103)	0.1535 (0.0051)	**0.1082** (0.0061)
	p	0.1825 (0.0055)	0.1797 (0.0055)	0.3515 (0.0045)	0.1936 (0.0089)	**0.1208** (0.0078)
C(55%)	λ	0.4519 (0.0295)	0.4580 (0.0363)	1.2576 (0.0378)	0.6704 (0.0591)	**0.4479** (0.0444)
	θ	0.1504 (0.0024)	0.1475 (0.0027)	0.1766 (0.0121)	0.2165 (0.0078)	**0.1041** (0.0054)
	p	0.1695 (0.0040)	0.1692 (0.0040)	0.5493 (0.0048)	0.3363 (0.0136)	**0.1214** (0.0067)

Table 2.3 Simulation results for the non-homogeneous ZAP DGPs. The average MAE are reported for the parameters of interest: $\mathbf{\Lambda} = (\lambda_1, \ldots, \lambda_{12})$, the Poisson intensities; θ, the probability of zero; p, the probability of an excess zero. Standard deviations of the MAE are reported between parentheses. The smallest value of the MAE for a given scenario is in bold. Note that in the case of $\mathbf{\Lambda}$, the average MAE over the 12 parameters are reported. In the first column, the percentage corresponds to the total probability of having a 0.

DGP	Parameter	NHPPF	Non-homogeneous Zap Forest
D(15%)	$\mathbf{\Lambda}$	0.2216 (0.0118)	**0.1334** (0.0055)
	θ	0.1511 (0.0037)	**0.0721** (0.0040)
	p	0.1515 (0.0038)	**0.0721** (0.0040)
D(35%)	$\mathbf{\Lambda}$	0.3962 (0.0135)	**0.1462** (0.0073)
	θ	0.3334 (0.0055)	**0.1012** (0.0056)
	p	0.3499 (0.0050)	**0.1016** (0.0056)
D(55%)	$\mathbf{\Lambda}$	0.5656 (0.0143)	**0.1724** (0.0105)
	θ	0.4534 (0.0126)	**0.1053** (0.0054)
	p	0.5476 (0.0057)	**0.1059** (0.0054)
E(15%)	$\mathbf{\Lambda}$	0.2274 (0.0132)	**0.1481** (0.0062)
	θ	0.1517 (0.0038)	**0.0721** (0.0041)
	p	0.1524 (0.0038)	**0.0721** (0.0041)
E(35%)	$\mathbf{\Lambda}$	0.4131 (0.0141)	**0.1627** (0.0063)
	θ	0.3363 (0.0055)	**0.1011** (0.0056)
	p	0.3512 (0.0051)	**0.1011** (0.0056)
E(55%)	$\mathbf{\Lambda}$	0.6093 (0.0143)	**0.1885** (0.0096)
	θ	0.4682 (0.0114)	**0.1053** (0.0055)
	p	0.5490 (0.0057)	**0.1054** (0.0055)
F(15%)	$\mathbf{\Lambda}$	0.1414 (0.0072)	**0.0923** (0.0043)
	θ	0.1492 (0.0037)	**0.0721** (0.0040)
	p	0.1522 (0.0038)	**0.0721** (0.0040)
F(35%)	$\mathbf{\Lambda}$	0.2496 (0.0082)	**0.1079** (0.0045)
	θ	0.3182 (0.0066)	**0.1012** (0.0056)
	p	0.3510 (0.0051)	**0.1012** (0.0056)
F(55%)	$\mathbf{\Lambda}$	0.3677 (0.0087)	**0.1376** (0.0079)
	θ	0.4164 (0.0128)	**0.1052** (0.0055)
	p	0.5489 (0.0058)	**0.1053** (0.0055)

Chapter 3

A Smooth Forest–Based Model for Nonhomogeneous Poisson Processes

Abstract

This paper proposes a nonhomogeneous Poisson process forest which provides a smooth estimate of the intensity function. Instead of using a fixed time partition, as in Mathlouthi *et al.* (2015a), we vary the intervals from one tree to another. This results in a smoother estimate of the intensity function.

Keywords: Nonhomogeneous Poisson Process, Poisson tree, random forest, smooth intensity function, clustering functions.

3.1 Introduction

In this article, we are interested in data where individuals experience repeated events. More precisely, we model the number and occurrence times of events. Such data are seen in several areas such as marketing, reliability, medicine, and actuarial field. For instance, in marketing, the redemptions made by customers member of a loyalty program will be used as an illustration in this paper.

Many parametric models have been proposed in the literature to deal with count data. Poisson regression models were proposed in Frome *et al.* (1973). But the basic models were developed to handle count data with a time-homogeneous intensity function. Lawless (1987)

designed a parametric model for non-homogeneous count data by adopting a time dependent intensity function with a polynomial form. Zaho an Xie (1992) proposed a maximum likelihood estimation methodology for non-homogeneous Poisson processes.

In this paper, we are concerned with tree-based methods, in particular random forests (Breiman, 2001). Random forests consists in growing an ensemble of trees based on bootstrapped data samples by injecting randomness in the tree growing process. Random forests are non-parametric and flexible, in that they can automatically adapt to many data structures.

Mathlouthi *et al.* (2015a), introduced an algorithm to build a Non–Homogeneous Poisson Process Tree (NHPPT) and a Non–Homogeneous Poisson Process Forest (NHPPF). The NHPPF works basically by partitioning the total time period into subperiods and by treating the number of events in each subperiod as a homogeneous Poisson processes. The partitioning can be quite general and can be applied to a wide variety of situations. In the end, the tree (or forest) produces a piecewise constant estimation of the intensity function. The use of a fixed partition might be reasonable when predictions are needed for a specified partition of time (e.g. days, weeks, months). However, it is not always clear how to choose the partition, and the estimated intensity function can be highly influenced by the chosen partition. In order to diminish the impact of selecting one specific partition, we propose to build a forest by varying the partition from tree to tree. Averaging the resulting trees will provide a smoother estimation of the intensity function, compared to a piecewise constant estimator.

The paper is organized as follows. Section 3.2 describes the proposed approach. Section 3.3 presents the results of a simulation study to evaluate the performance of the method. This study compares homogeneous and non–homogeneous benchmarks (no covariates), as well as parametric and forest models over different data generation scenarios. Section 3.4 illustrates the application of the method with a real data set. Concluding remarks and possibilities for future research are given in Section 3.5.

3.2 Smooth non-homogeneous Poisson process forest

Mathlouthi *et al.* (2015a) proposed a tree model with a piecewise constant intensity function. Consider a fixed horizon T that is partitioned into K subperiods $T_1, T_2, \ldots, T_K$

such that:

$$\bigcup_{k=1}^{K} T_k = T \quad \text{and} \quad T_i \cap T_j = \emptyset \quad \text{for all} \quad i \neq j.$$

Let N_{ik} be the number of events observed in the subperiod T_k for subject i. Assume that N_{ik} follows a Poisson distribution with parameter λ_k and that all the N_{ik}'s are independent. In this paper, the K subperiods will be adjacent time intervals $(a_0, a_1], (a_1, a_2], \ldots, (a_{K-1}, a_K]$ covering the whole period $T = (a_0, a_K]$. The maximum likelihood estimators (MLEs) of λ_k, $k = 1, \ldots, K$ are:

$$\hat{\lambda}_k = \frac{\sum_{i=1}^{N} N_{ik}}{N}, k = 1, \ldots, K,$$

where N is the number of subjects.

Now we assume that we have a vector of p covariates $X = (X_1, \ldots, X_p)$ per subject. For tree building, using the well-known CART paradigm described in Breiman *et al.* (1984), Mathlouthi *et al.* (2015a) proposed to fit the above model separately in the candidate children nodes and to use the observed log–likelihood as the splitting rule. The best split is then the one that maximizes the observed log–likelihood among all allowable splits. Tree building can then proceed recursively as usual. Once a final tree is selected, an estimation of the intensity function can be obtained, for an observation with covariate vector $X = x$, as the MLE in the terminal node in which the observation ends up. Namely, if $t(x)$ is index set of the (training) observations in the terminal node in which the observation with $X = x$ falls, then

$$\hat{\lambda}_k(x) = \frac{\sum_{i \in t(x)} N_{ik}}{N^{t(x)}}, k = 1, \ldots, K, \tag{3.1}$$

where $N^{t(x)}$ is the size of node $t(x)$. This method is called Non–Homogeneous Poisson Process Tree (NHPPT). It is clear that, by construction, the intensity function is a piecewise constant function over T.

Mathlouthi *et al.* (2015a) also proposed to build a forest of trees with the above method. To do this, they applied the classical forest algorithm (Breiman, 2001) by using the same partition of T for each tree in the forest. Thus, the forest estimate of the intensity function is also a piecewise constant function on the same subperiods.

In this paper, we are interested in producing a smooth estimation of the intensity function. The key idea is to vary the partition, both in the number of subperiods and their lengths,

from one tree to another in the forest. Averaging them will then produce a smoother estimate of the intensity function.

The algorithm proceeds as follows:

1. For $b = 1, \ldots, B$, draw a bootstrap sample from the original data set.

2. For the b^{th} bootstrap sample, grow an unpruned NHPPT using the intervals

$(a_0^b, a_1^b], (a_1^b, a_2^b], \ldots, (a_{K^b-1}^b, a_{K^b}^b]$, where $a_0^b = 0 < a_1^b < a_2^b \cdots < a_{K^b}^b = T$. Moreover, at each node of the tree, randomly sample p_0 $(0 < p_0 \leq p)$ of the p predictors and choose the best split among those variables.

The superscript b used in the definition of the intervals indicates that both the number and placement of the intervals can vary from one tree to another.

The estimation of the intensity function $\lambda(t|x)$, for a subject with covariate vector $X = x$, can then proceed as follows. Let $\hat{\lambda}^b(t|x)$ be the estimate of $\lambda(t|x)$ for the b^{th} tree. It is piecewise constant function given by $\hat{\lambda}^b(t|x) = \sum_{k=1}^{K^b} 1(t \in [a_{k-1}^b, a_k^b])\hat{\lambda}_k^b(x)$, where $\hat{\lambda}_k^b(x)$ is the estimate of λ_k for the b^{th} tree. The forest estimate of $\lambda(t|x)$ is given by

$$\hat{\lambda}_F(t|x) = \frac{\sum_{b=1}^{B} \hat{\lambda}^b(t|x)}{B}.$$

We call this general method Smooth Non-Homogeneous Poisson Process Forest (SNHPPF).

One key aspect of this method is the choice of intervals from tree to tree. Many methods to vary the intervals from tree to tree are possible. In this paper, the following method was used to select the intervals in a tree. We first randomly generate the number of intervals from a Poisson random variable with mean $\log_2(N) + 1$, where N is the total number of events experienced by all subjects during the whole time period T. This number is generated independently for each tree. The average of the Poisson variable is motivated by Sturges' rule which is often used to determine the number of intervals in a histogram. Indeed, the MLEs of the λ parameters are the proportions of events in each interval, which is the same as the proportions used to build a histogram. Secondly, we calculate the limits of the intervals

by imposing the constraint that each of them contains the same number of events, to avoid having intervals that are empty or with very few events.

We shall use simulations to study the merits of the proposed approach in the next section.

3.3 Simulation Study

The simulations are designed to compare the SNHPPF to different competitors, with artificial datasets generated from three different data generating processes (DGPs). In particular, we want to see how the variation of intervals from tree to tree allows us to estimate a smooth rate function.

We consider n independent subjects and we observe the times of events that occur during the whole period T. We define t_i the vector of occurrence times during the whole time period T, and $x_i = (x_{i1}, ..., x_{ip})'$ the vector of covariates for subject $i = 1, 2, ..., n$.

3.3.1 Models compared

Twelve competing models are used in the comparison, including the models used in Mathlouthi *et al.* (2015a). They come from the five different approaches.

Approach 1: A non–homogeneous Poisson process model without covariates. This model assumes that N_{ik}, the number of events which occurred for subject i in subperiod T_k, is given by a Poisson random variable with mean λ_k and that all the N_{ik}'s are independent. The three models used from this approach are the ones with 1, 10 and 50 intervals.

Approach 2: A piecewise Poisson regression model. Hence, it is a non–homogeneous Poisson regression model. This model assumes that N_{ik} is given by a Poisson random variable with mean λ_{ik} and that $\lambda_{ik} = \lambda_k \exp(x_i'\beta)$. Again, the three models with 1, 10 and 50 intervals are considered.

Approach 3: A non-homogeneous Poisson process regression model. This model assumes that $\lambda(t|x)$ is a polynomial, specifically that $\lambda(t|x) = \delta t^{\delta-1} \exp(\beta_0 + \beta_1 X_1 + ... + \beta_p X_p)$, see Lawless (1987). This model is the true one for DGP 1 below.

Approach 4: The NHPPF as described in Mathlouthi *et al.* (2015a). The four models with 1, 10, 18 and 50 intervals are considered.

Approach 5: The proposed SNHPPF as described in the preceding section.

For the first three approaches, the models with 1 interval reduces to a homogeneous model. Moreover, for the models with more than one interval, they are selected to have equal length while covering the whole period. The NHPPF model with 18 intervals is used because, on average, the number of intervals used for the SNHPPF (i.e. $\log_2(N)+1$) is approximately 18 over all simulations.

3.3.2 Simulation design

We consider three different DGPs. For each DGP the rate $\lambda(t|x)$ has a different functional form. The simulations design uses nine independent covariates, $X_1, ..., X_9$, uniformly distributed on the interval $[0, 10]$. Only X_1, X_2, and X_3 are related to $\lambda(t|x)$ and the others are noise. The period of interest is [0,12].

DGP 1: A non-homogeneous Poisson regression model with main effects only:

$$\lambda(t|x) = 1.2t^{0.2} \exp\left(0.01 + 0.02X_1 - 0.03X_2 + 0.01X_3\right).$$

DGP 2: A non-homogeneous Poisson regression model with more complicated effects:

$$\lambda(t|x) = 1.5t^{0.5} \exp\left(0.01 + 0.01X_1X_2 + 0.03X_1 \log(X_2) - 0.006X_3X_2^2\right).$$

DGP 3: A tree model with four leaves:

Leaf 1. If $X_1 \leq 5$ and $X_2 \leq 5$ then $\lambda(t|x) = 0.5 \sin(2t) + 1.5$.

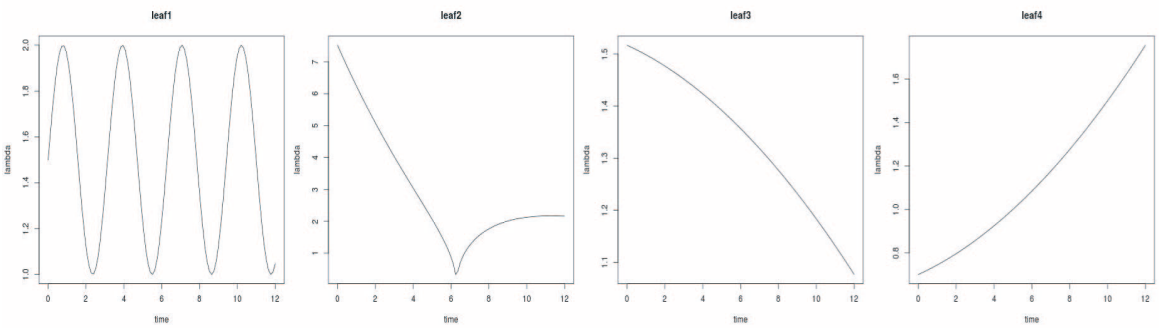
Leaf 2. If $X_1 \leq 5$ and $X_2 > 5$ then $\lambda(t|x) = 3\exp(-\frac{t}{10})\sqrt{|t - 2\pi|}$.

Leaf 3. If $X_1 > 5$ and $X_3 \leq 5$ then $\lambda(t|x) = 1.5 + \frac{2-2t-0.2t^2}{120} = \frac{182-2t-0.2t^2}{120}$.

Leaf 4. If $X_1 > 5$ and $X_3 > 5$ then $\lambda(t|x) = 0.7 + \frac{2t+0.2t^2}{50} = \frac{35+2t+0.2t^2}{50}$.

The tree structure was chosen to include several types of intensity functions. They are depicted in Figure 3.1.

Figure 3.1



The simulations were performed with 100 runs. For each run, the training sample size is $1,000$ and the test sample size is $5,000$. All forests are built with 500 trees and 30 observations are needed to attempt splitting any node. Three out of the nine predictors were randomly selected at each node to find the best split. In order to compare the different models, we use the L_1 as the criterion for comparison. It is the integral of the difference between the true and the estimated intensity function. It can also be seen as the area between the two functions and is defined by:

$$L_1 = \int_0^T |\lambda(t|x) - \hat{\lambda}(t|x)|dt,$$

where $\lambda(t|x)$ and $\hat{\lambda}(t|x)$ are the true and estimated intensity functions. The average over the test sample and all simulation runs gives the final performance criterion for a given model. Note that the trapezoid rule is used to approximate the integral.

3.3.3 Results

The simulation results are presented in Table 3.1. First, we can see that using the covariates improves the estimation of the intensity function for all DGPs. Second, the proposed method (SNHPPF) is always better than a forest with a fixed number of intervals (NHPPF), at least for the number of intervals selected. Third, increasing too much the number of intervals always deteriorates the performance of both the parametric and forest based methods. Hence,

using many intervals is not a good way of achieving a smoother estimation of the intensity function.

Let us now examine each DGP individually. The polynomial Poisson regression is well-specified for DGP 1, and thus has the best performance. Looking at the second DGP, the true model is still a polynomial parametric one but with more complicated effects including interactions. In this case the estimated polynomial parametric model is not well-specified. We see that the forest based methods can more easily, and automatically, detect this more complicated signal. The proposed method is the best one in this case, albeit only slightly better than taking 18 intervals with the NHPPF. Finally, the third DGP is a tree model. As we can see, forest models are generally better than parametric approaches. Moreover, the new method again performs better than the other forest methods.

Table 3.1 Simulation results. The first number is the L_1 criterion and the number between parentheses is its standard deviation.

Model		DGP1	DGP2	DGP3
No covariates	1 interval	4.418/(0.121)	21.52/(0.268)	10.63/(0.278)
	10 intervals	4.419/(0.122)	21.51/(0.267)	10.62/(0.279)
	50 intervals	15.20/(0.048)	27.56/(0.266)	15.77/(0.142)
Piecewise Poisson regression	1 interval	3.924/(0.150)	16.63/(0.256)	9.284/(0.191)
	10 intervals	3.923/(0.150)	16.63/(0.257)	9.284/(0.191)
	50 intervals	15.20/(0.041)	26.46/(0.249)	15.78/(0.120)
Polynomial Poisson regression		**0.390**/(0.087)	11.04/(0.168)	6.884/(0.083)
NHPPF	1 interval	2.649/(0.028)	12.24/(0.129)	6.438/(0.108)
	10 intervals	1.308/(0.053)	7.003/(0.177)	2.367/(0.091)
	18 intervals	1.454/(0.055)	6.830/(0.181)	2.235/(0.092)
	50 intervals	1.879/(0.051)	7.191/(0.177)	2.595/(0.091)
SNHPPF (proposed method)		1.178/(0.051)	**6.811**/(0.176)	**1.896**/(0.094)

3.4 An example

In this section, we provide an example to illustrate how our approach can be used along with a curve clustering method to perform market segmentation. We consider redemption data coming from a rewards program and ranging from January 1st, 2005 to December 31st, 2005. Customers belonging to this data have made at least two redemptions in 2004. The outcome is the time in days of the claims in 2005. Therefore, the values of the outcome range from 0 to 365. We have $4,424$ customers having between 0 and 46 redemptions.

We will use the SNHPPF to obtain a smooth estimate of the intensity curve for each customer. Then a curve clustering method will be applied to segment them.

We consider 28 predictors: 20 continuous and 8 categorical. Some of these variables measure the number of points earned with different partners during the year 2004. Other variables measure the number of different types of transactions during 2004. Another set of predictors includes demographic characteristics.

Figure 3.2 Intensity functions of all subjects in the data.

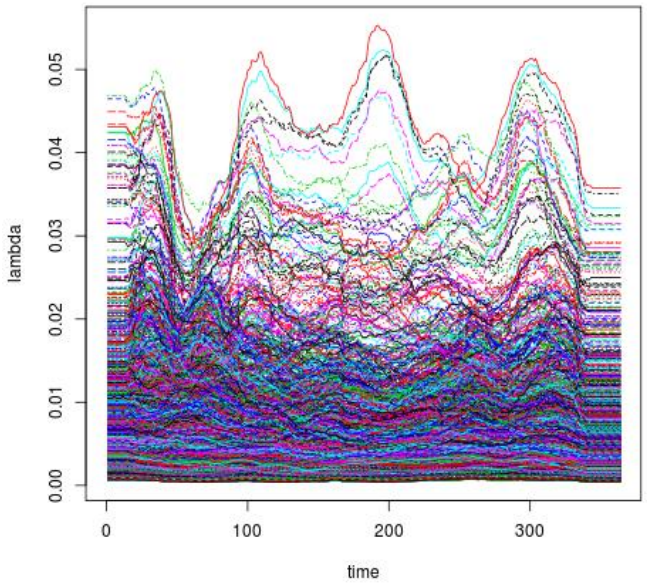
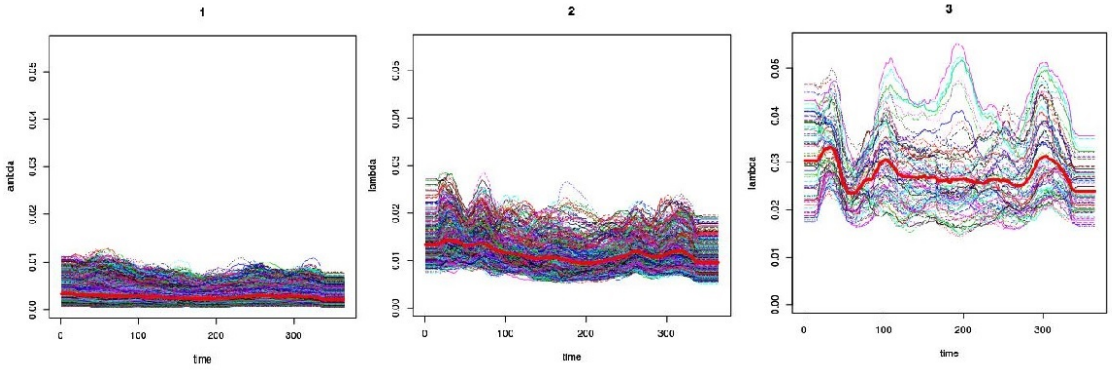


Figure 3.2 illustrates the estimated intensity functions of all the customers present in the data that were obtained with SNHPPF. We can clearly see that different patterns are present.

Giacofci *et al.* (2013) have proposed a method for the clustering of functional data, called model-based clustering. This method assumes a probabilistic model on scores of functional principal component analysis, or on the coefficients of the approximations of the curves into a basis of this function. This technique is available through the R package "curvclust". Using the default settings of the function "getFCM", three clusters of sizes 3,887, 480 and 57 are obtained. They are cluster 1 to 3, respectively, in Figure 3.3. The red curve in each plot is the mean curve of the cluster.

Cluster 1 has two main characteristics: low intensity and little time variations. The intensity is relatively low and practically constant over time. Cluster 2 presents also a fairly constant intensity over time, perhaps slightly decreasing, but with a higher rate. In contrast, the curves in Cluster 3 exhibit more variations with the highest intensities. Looking at the mean curve, we observe a first positive shift at the beginning of the year followed by an abrupt decrease. A second positive peak with lower magnitude occurs around April. The mean rate then remains almost constant until the end of the year when a third positive shift happens.

Figure 3.3 The intensity functions of the three clusters. The red curves are the mean curve for each cluster.



3.5 Concluding remarks

In this paper, we have proposed a random forest algorithm which provides a smooth estimation of the intensity function associated with non-homogeneous count data. The smoothness is obtained by using a forest of non–homogeneous Poisson process trees by varying the partition of the piecewise intensity function from one tree to another. This approach bypasses the non-trivial choice of the intervals in a piecewise constant approach. The proposed smooth non-homogeneous Poisson process forest performed well in a simulation study compared to parametric and other forests based approaches. An original application of the estimated intensity curves produced by our method for market segmentation was presented.

Many possibilities for future work and investigation are available. In this paper, we only used a simple method to select the number and placement of the intervals in the partition. Even though the results were good, it would be interesting to investigate other ways to generate random partitions and see how the final performance is sensible to that aspect.

Mathlouthi *et al.* (2015b) extended the non–homogeneous Poisson process forest method of Mathlouthi *et al.* (2015a) to handle excess zeros. However, the estimated intensity function is still piecewise constant. Hence, it would be interesting to extend the current method to be able to account for excess zeros while producing a smooth estimate of the intensity function.

Finally, our method can only handle baseline covariates. Hence, it would be interesting to generalize it to be able to include time-varying covariates.

References

• Breiman, L., Friedman, J., Olshen, R. and Stone, C. (1984). *Classification and Regression Trees*. Belmont, CA: Wadsworth International Group.

• Breiman, L. (2001). Random Forests. *Machine Learning* **45**, 5–32.

• Frome, E.L., Kutner, M.H., and Beauchamp, J.J. (1973). Regression analysis of Poisson-distributed data. *Journal of the American Statistical Association* **68**, 935–940.

• Giacofci, M., Lambert-Lacroix, S., Marot, G. and Picard, F. (2013). Wavelet-Based Clustering for Mixed-Effects Functional Models in High Dimension. *Biometrics* **69**, 31–40.

• Lawless, J. F. (1987). Regression Methods for Poisson Process Data. *Journal of the American Statistical Association* **82**, 808–815.

• Mathlouthi, W., Fredette, M. and Larocque, D. (2015a). Regression Trees and Forests for Non-Homogeneous Poisson Processes. *Statistics and Probability Letters* **96**, 204–211.

• Mathlouthi, W., Larocque, D. and Fredette, M. (2015b). Random Forests for Nonhomogeneous Poisson Processes with Excess Zeros. *Les Cahiers du GERAD* **G-2015-77**.

• Zaho, M. and Xie, M. (1992). On Maximum Likelihood Estimation for a General Non-Homogeneous Poisson Process. *Scandinavian Journal of Statistics* **23**, 597–607.

CONCLUSION GÉNÉRALE

Dans cette thèse, nous avons proposé des nouvelles méthodes d'arbres et de forêts aléatoires généralisant dans deux directions les méthodes existantes : pour le cas de processus non-homogènes et pour le cas de processus avec zéros excédentaires. Les nouveaux algorithmes ont été codés en C et R. Les performances des nouvelles méthodes ont été étudiées par simulations, qui ont démontré les avantages qu'elles offrent par rapport aux méthodes existantes. Un exemple original d'utilisation des courbes de taux pour segmenter les clients d'un programme de fidélisation a également été présenté. Les différentes méthodes proposées dans cette thèse sont prometteuses et ouvrent la voie à de nombreuses généralisations. Premièrement, les critères de partitionnement utilisés sont basés sur l'hypothèse que le processus est Poisson. Il serait alors intéressant d'étudier d'autres critères basés sur d'autres hypothèses. En effet, il arrive parfois en pratique qu'il y ait sur-dispersion dans le processus. Dans ce cas, utiliser un critère de partitionnement basé sur la loi binomiale négative pourrait être plus adéquat. Deuxièmement, nous avons considéré seulement le cas où les variables explicatives ne varient pas dans le temps ou, à tout le moins, nous avons seulement considérés leurs valeurs initiales au temps 0. Par conséquent, il serait intéressant de généraliser nos méthodes afin de pouvoir incorporer des variables explicatives qui varient dans le temps.