# Sparse Poisson regression via mixed-integer optimization

Hiroki Saishu[1], Kota Kudo[1], Yuichi Takano[2*]

**1** Graduate School of Science and Technology, University of Tsukuba, Tsukuba, Ibaraki, Japan
**2** Faculty of Engineering, Information and Systems, University of Tsukuba, Tsukuba, Ibaraki, Japan

* Corresponding author

## Abstract

We present a mixed-integer optimization (MIO) approach to sparse Poisson regression. The MIO approach to sparse linear regression was first proposed in the 1970s, but has recently received renewed attention due to advances in optimization algorithms and computer hardware. In contrast to many sparse estimation algorithms, the MIO approach has the advantage of finding the best subset of explanatory variables with respect to various criterion functions. In this paper, we focus on a sparse Poisson regression that maximizes the weighted sum of the log-likelihood function and the $L_2$-regularization term. For this problem, we derive a mixed-integer quadratic optimization (MIQO) formulation by applying a piecewise-linear approximation to the log-likelihood function. Optimization software can solve this MIQO problem to optimality. Moreover, we propose two methods for selecting a limited number of tangent lines effective for piecewise-linear approximations. We assess the efficacy of our method through computational experiments using synthetic and real-world datasets. Our methods for selecting tangent lines provide better log-likelihood values than do conventional greedy algorithms. In addition, our MIQO formulation delivers better out-of-sample prediction performance than do forward stepwise selection and $L_1$-regularized estimation, especially in low-noise situations.

## Introduction

A count variable, which takes only on nonnegative integer values, reflects the number of occurrences of an event during a fixed time period. Count regression models such as Poisson, overdispersed Poisson, and negative binomial regression are standard methods for predicting such count variables [8, 11, 17]. In particular, Poisson regression is most commonly used for count regression. There are numerous applications of Poisson regression models for predicting count variables, including manufacturing defects [30], disease incidence [40], crowd counting [9], length of hospital stay [48], and vehicle crashes [49].

The aim of sparse estimation is to decrease the number of nonzero estimates of regression coefficients. This method is often used for selecting a significant subset of explanatory variables [10, 20, 32, 34]. Subset selection provides the following benefits:

- data collection and storage costs can be reduced,
- computational load of estimating regression coefficients can be reduced,

- interpretability of regression analysis can be increased, and
- generalization performance of a regression model can be improved.

A direct way of *best* sparse estimation involves evaluating all possible subset regression models. However, the exhaustive search method [31, 33, 36] is often computationally infeasible because the number of possible subsets grows exponentially with the number of candidate variables. In contrast, stepwise selection [14, 36], which repeats addition and elimination of one explanatory variable at a time, is a practical method for sparse estimation. Several metaheuristic algorithms have been applied to subset selection for Poisson regression [1, 27], and various regularization methods have been recently proposed for sparse Poisson regression [16, 19, 23, 24]. Note, however, that these (non-exhaustive) sparse estimation methods are heuristic algorithms, which cannot verify optimality of an obtained subset of explanatory variables (e.g., in the maximum likelihood sense).

In this paper, we focus on the mixed-integer optimization (MIO) approach to sparse estimation. This approach was first proposed for sparse linear regression in the 1970s [2], but has recently received renewed attention due to advances in optimization algorithms and computer hardware [5, 12, 21, 28, 35, 47]. In contrast to many sparse estimation algorithms, the MIO approach has the advantage of finding the best subset of explanatory variables with respect to various criterion functions, including Mallows' $C_p$ [37], adjusted $R^2$ [38], information criteria [18, 26, 38], mRMR [41], and the cross-validation criterion [44]. MIO-based sparse estimation methods can be extended to binary or ordinal classification models [4, 25, 39, 42, 43] and to eliminating multicollinearity [3, 6, 45, 46].

The log-likelihood to be maximized is a concave but nonlinear function, making it hard to apply an MIO approach to sparse Poisson regression. To remedy such nonlinearity, prior studies made effective use of piecewise-linear approximations of the log-likelihood functions, thereby yielding mixed-integer linear optimization (MILO) formulations for binary or ordinal classification [39, 42, 43]. Optimization software can solve the resultant MILO problems to optimality. Greedy algorithms for selecting a limited number of linear functions for piecewise-linear approximations have also been developed [39, 43].

This paper aims at establishing an effective MIO approach to sparse Poisson regression based on piecewise-linear approximations. Specifically, we consider a sparse Poisson regression that maximizes the weighted sum of the log-likelihood function and the $L_2$-regularization term. To that end, we derive a mixed-integer quadratic optimization (MIQO) formulation by applying a piecewise-linear approximation to the log-likelihood function. We also propose two methods for selecting a limited number of tangent lines to improve the quality of piecewise-linear approximations.

We assess the efficacy of our method through computational experiments using synthetic and real-world datasets. Our methods for selecting tangent lines produce better log-likelihood values than do conventional greedy algorithms. For synthetic datasets, our MIQO formulation realizes better out-of-sample prediction performance than do forward stepwise selection and $L_1$-regularized estimation, especially in low-noise situations. For real-world datasets, our MIQO formulation compares favorably with the other methods in out-of-sample prediction performance.

**Notation** Throughout this paper, sets of consecutive integers ranging from 1 to $n$ are denoted as

$$[n] := \begin{cases} \{1, 2, \ldots, n\} & \text{if } n \geq 1, \\ \emptyset & \text{otherwise.} \end{cases}$$

# Methods

This section starts with a brief review of Poisson regression, and then presents our MIO formulations for sparse Poisson regression based on piecewise-linear approximations. We then describe our methods for selecting tangent lines suitable for piecewise-linear approximations.

## Poisson regression model

Suppose we are given a sample of $n$ data instances $(\boldsymbol{x}_i, y_i)$ for $i \in [n]$, where $\boldsymbol{x}_i := (x_{i1}, x_{i2}, \ldots, x_{ip})^\top$ is a vector composed of $p$ explanatory variables, and $y_i \in \{0\} \cup [m]$ is a count variable to be predicted for each instance $i \in [n]$. We define binary labels as

$$\delta_{ik} := \begin{cases} 1 & \text{if } y_i = k, \\ 0 & \text{otherwise} \end{cases} \quad (i \in [n],\ k \in \{0\} \cup [m]). \tag{1}$$

The random count variable $Y$ is assumed to follow the Poisson distribution

$$\Pr(Y = k \mid \lambda) = \frac{\lambda^k \exp(-\lambda)}{k!} \quad (k \in \{0\} \cup [+\infty]), \tag{2}$$

where $\lambda \in \mathbb{R}_+$ is a parameter representing both the mean and variance of the Poisson distribution. The distribution parameter $\lambda$ is explained by the linear regression model

$$\log \lambda_i = \boldsymbol{w}^\top \boldsymbol{x}_i + b = w_1 x_{i1} + w_2 x_{i2} + \cdots + w_p x_{ip} + b \quad (i \in [n]), \tag{3}$$

where $\boldsymbol{w} := (w_1, w_2, \ldots, w_p)^\top$ is a vector of regression coefficients, and $b$ is an intercept term. Then, the occurrence probability of the given sample is expressed as

$$\prod_{i=1}^{n} \Pr(Y = y_i \mid \lambda_i) = \prod_{i=1}^{n} \prod_{k=0}^{m} \Pr(Y = k \mid \lambda_i)^{\delta_{ik}}. \quad \because \text{Eq. (1)}$$

The regression parameters $(b, \boldsymbol{w})$ are estimated by maximizing the log-likelihood function

$$\begin{aligned} L(b, \boldsymbol{w}) &:= \log \left( \prod_{i=1}^{n} \prod_{k=0}^{m} \Pr(Y = k \mid \lambda_i)^{\delta_{ik}} \right) \\ &= \sum_{i=1}^{n} \sum_{k=0}^{m} \delta_{ik} \left( k \log \lambda_i - \lambda_i - \log k! \right) \quad \because \text{Eq. (2)} \\ &= \sum_{i=1}^{n} \sum_{k=0}^{m} \delta_{ik} f_k(\boldsymbol{w}^\top \boldsymbol{x}_i + b), \quad \because \text{Eq. (3)} \end{aligned} \tag{4}$$

where $f_k(u)$ is a nonlinear function defined as

$$f_k(u) = ku - \exp(u) - \log k! \quad (k \in \{0\} \cup [m]). \tag{5}$$

Fig. 1 shows graphs of $f_k(u)$ for $k \in \{0, 5, 10, 15, 20\}$. Since its second derivative $f_k''(u) = -\exp(u)$ is always negative, $f_k(u)$ is a nonlinear concave function.

**Fig 1. Graphs of $f_k(u)$ for $k \in \{0, 5, 10, 15, 20\}$.**

The following theorem gives an asymptote of $f_k(u)$.

**Theorem 1.** When $u$ goes to $-\infty$, $f_k(u)$ has the asymptote

$$\phi_k(u) = ku - \log k! \quad (k \in \{0\} \cup [m]). \tag{6}$$

*Proof.* We have

$$\begin{aligned} \lim_{u \to -\infty} \frac{f_k(u)}{u} &= \lim_{u \to -\infty} \left( k - \frac{\exp(u)}{u} - \frac{\log k!}{u} \right) = k, \\ \lim_{u \to -\infty} (f_k(u) - ku) &= \lim_{u \to -\infty} (-\exp(u) - \log k!) = -\log k!, \end{aligned}$$

which completes the proof. □

## Mixed-integer nonlinear optimization formulation

Before deriving our desired formulation, we introduce a mixed-integer nonlinear optimization (MINLO) formulation for sparse Poisson regression. Let $\boldsymbol{z} := (z_1, z_2, \dots, z_p)^\top$ be a vector composed of binary decision variables for subset selection, namely,

$$z_j = \begin{cases} 1 & \text{if the } j\text{th explanatory variable is selected,} \\ 0 & \text{otherwise (i.e., } w_j = 0) \end{cases} \qquad (j \in [p]).$$

To improve the generalization performance of a resultant regression model, we also introduce the $L_2$-regularization term $\alpha \boldsymbol{w}^\top \boldsymbol{w}$ to be minimized, where $\alpha \in \mathbb{R}_+$ is a user-defined regularization parameter [22]. We therefore address maximizing the weighted sum of the log-likelihood function (4) and the $L_2$-regularization term. This sparse Poisson regression can be formulated as the MINLO problem

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{n} \sum_{k=0}^{m} \delta_{ik} f_k(\boldsymbol{w}^\top \boldsymbol{x}_i + b) - \alpha \boldsymbol{w}^\top \boldsymbol{w} && (7) \\ \text{subject to} \quad & z_j = 0 \ \Rightarrow \ w_j = 0 \quad (j \in [p]), && (8) \\ & \sum_{j=1}^{p} z_j = \theta, && (9) \\ & b \in \mathbb{R}, \ \boldsymbol{w} \in \mathbb{R}^p, \ \boldsymbol{z} \in \{0,1\}^p, && (10) \end{aligned}$$

where $\theta \in [p]$ is a user-defined parameter of the subset size. If $z_j = 0$, then the $j$th coefficient must be zero by logical implication (8). Eq. (9) specifies the number of nonzero regression coefficients, and Eq. (10) lists all decision variables.

The logical implication of Eq. (8) can be imposed by using indicator constraints implemented in modern optimization software. The logical implication of Eq. (8) can also be represented as

$$-Mz_j \le w_j \le Mz_j \quad (j \in [p]),$$

where $M$ is a sufficiently large positive constant.

## Piecewise-linear approximation

It is very difficult to handle the MINLO problem of Eqs. (7)–(10) using MIO software, because Eq. (7) to be maximized is a concave but nonlinear function. Following prior studies [39, 42, 43], we apply piecewise-linear approximation techniques to the nonlinear function (5).

Letting $\{(u_{k\ell}, f_k(u_{k\ell})) \mid \ell \in [h]\}$ be a set of $h$ tangent points for the function $f_k(u)$, the corresponding tangent lines are

$$g_k(u \mid u_{k\ell}) := f'_k(u_{k\ell})(u - u_{k\ell}) + f_k(u_{k\ell}) \quad (\ell \in [h]), \tag{11}$$

where $f'_k(u) = k - \exp(u)$ is the derivative of $f_k(u)$.

As Fig. 2 shows, the graph of a concave function lies below its tangent lines, so $f_k(u)$ can be approximated by the pointwise minimum of a set of $h$ tangent lines. For each $u$, we have

$$\begin{aligned} f_k(u) \approx G_{kh}(u) &:= \min\{g_k(u \mid u_{k\ell}) \mid \ell \in [h]\} \\ &= \max\{t \mid t \leq g_k(u \mid u_{k\ell}) \quad (\ell \in [h])\}, \end{aligned} \tag{12}$$

where $t \in \mathbb{R}$ is an auxiliary decision variable.

**Fig 2. Piecewise-linear approximation of $f_k(u)$ for $k = 10$.**

We next focus on the approximation gap $g_k(u \mid \bar{u}) - f_k(u)$ arising from a tangent point $(\bar{u}, f_k(\bar{u}))$. By the following theorem, this gap does not depend on $k$; therefore, we can employ the same set $\{u_\ell \mid \ell \in [h]\}$ for all $k \in \{0\} \cup [m]$ when selecting tangent points for piecewise-linear approximations.

**Theorem 2.** $g_k(u \mid \bar{u}) - f_k(u)$ is independent of $k \in \{0\} \cup [m]$.

*Proof.* We have

$$\begin{aligned} & g_k(u \mid \bar{u}) - f_k(u) \\ =\, & (k - \exp(\bar{u}))(u - \bar{u}) + k\bar{u} - \exp(\bar{u}) - \log k! - (ku - \exp(u) - \log k!) \\ =\, & -\exp(\bar{u})(u - \bar{u}) - \exp(\bar{u}) + \exp(u), \end{aligned}$$

which completes the proof. □

## Mixed-integer quadratic optimization formulation

We are now ready to present our desired formulation for sparse Poisson regression. Let $\boldsymbol{T} := (t_{ik})_{(i,k) \in [n] \times (\{0\} \cup [m])}$ be a matrix composed of auxiliary decision variables for piecewise-linear approximations. We substitute Eq. (11) and $u = \boldsymbol{w}^\top \boldsymbol{x}_i + b$ into Eq. (12) to make a piecewise-linear approximation of the objective function (7). Consequently, the MINLO problem of Eqs. (7)–(10) can be reduced to the MIQO problem

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{n} \sum_{k=0}^{m} \delta_{ik} t_{ik} - \alpha \boldsymbol{w}^\top \boldsymbol{w} && (13) \\ \text{subject to} \quad & t_{ik} \leq f'_k(u_\ell)(\boldsymbol{w}^\top \boldsymbol{x}_i + b - u_\ell) + f_k(u_\ell) \\ & \qquad\qquad (i \in [n],\ k \in \{0\} \cup [m],\ \ell \in [h]), && (14) \\ & z_j = 0 \ \Rightarrow \ w_j = 0 \quad (j \in [p]), && (15) \\ & \sum_{j=1}^{p} z_j = \theta, && (16) \\ & b \in \mathbb{R},\ \boldsymbol{w} \in \mathbb{R}^p,\ \boldsymbol{T} \in \mathbb{R}^{n \times (m+1)},\ \boldsymbol{z} \in \{0,1\}^p, && (17) \end{aligned}$$

where Eq. (17) lists all of the decision variables. Note that optimization software can solve this MIQO problem to optimality.

## Previous algorithms for selecting tangent lines

The accuracy of piecewise-linear approximations depends on the associated set of tangent lines. It is clear that with increasingly many appropriate tangent lines, the MIQO problem of Eqs. (13)–(17) approaches the original MINLO problem of Eqs. (7)–(10). In this case, however, solving the MIQO problem (13)–(17) becomes computationally expensive because the problem size grows larger. It is therefore crucial to limit the number of tangent lines for effective approximations.

Sato et al. [43] developed a greedy algorithm for selecting tangent lines to approximate the logistic loss function. This algorithm adds tangent lines one by one so that the total approximation gap (the area of the shaded portion in Fig. 2) will be minimized. Naganuma et al. [39] employed a greedy algorithm that selects tangent planes to approximate the bivariate nonlinear function for ordinal classification. This algorithm iteratively selects tangent points where the approximation gap is largest.

These previous algorithms have two limitations addressed in this paper. First, they totally ignore the properties of the sample distribution, Second, tangent lines are determined one at a time, so the resultant set of tangent lines is not necessarily optimal. In the following sections, we propose two methods for resolving these limitations.

## Adaptive greedy algorithm

Our first method, the *adaptive greedy algorithm*, selects tangent lines depending on the sample distribution.

Suppose we are given $(\bar{b}, \bar{\boldsymbol{w}})$ as regression parameter values. These values can be obtained, for example, through maximum likelihood estimation of the full model (3). We then have an empirical distribution of input values for the nonlinear function (5) as $\bar{u}_i := \bar{\boldsymbol{w}}^\top \boldsymbol{x}_i + \bar{b}$ for $i \in [n]$. Our algorithm sequentially minimizes the sum of squared approximation gaps based on this empirical distribution.

Specifically, we determine the $s$th tangent point on the condition that previous tangent points are fixed. This procedure is formulated as

$$u_s^* \in \operatorname*{arg\,min}_{u_s \in \mathbb{R}} \left\{ \sum_{i=1}^{n} (G_{ks}(\bar{u}_i) - f_k(\bar{u}_i))^2 \;\middle|\; \begin{array}{l} u_\ell = u_\ell^* \quad (\ell \in [s-1]) \\ L \le u_s \le U \end{array} \right\} \quad (s \in [h]), \tag{18}$$

where $[L, U]$ is an input interval of the nonlinear function (5). Notably, by Theorem 2 this algorithm yields the same set of tangent lines for all $k \in \{0\} \cup [m]$.

## Simultaneous optimization method

Our second method, the *simultaneous optimization method*, selects a set of $h$ tangent lines simultaneously, not sequentially.

Suppose the intersection between the $\ell$th and $(\ell + 1)$th tangent lines is specified by $c(\ell, \ell + 1)$, meaning $g_k(u \mid u_\ell) = g_k(u \mid u_{\ell+1})$ holds when $u = c(\ell, \ell + 1)$. We then simultaneously determine a set of $h$ tangent points minimizing the total approximation gap (the area of the shaded portion in Fig. 2). This procedure can be posed as the nonlinear optimization (NLO) problem

$$\begin{aligned} \text{minimize} \quad & \sum_{\ell=1}^{h} \int_{c(\ell-1,\ell)}^{c(\ell,\ell+1)} (g_k(u \mid u_\ell) - f_k(u)) \, \mathrm{d}u && (19) \\ \text{subject to} \quad & L \le u_1 \le u_2 \le \cdots \le u_h \le U, && (20) \\ & (u_1, u_2, \ldots, u_h) \in \mathbb{R}^h, && (21) \end{aligned}$$

where $c(0,1) = L$ and $c(h, h+1) = U$ are fixed. NLO software can handle this problem, yielding a locally optimal set of tangent points. This method also provides the same set of tangent lines for all $k \in \{0\} \cup [m]$.

# Results and discussion

This section describes computational experiments for evaluating the effectiveness of our method for sparse Poisson regression.

## Methods for comparison

We investigate the performance of our MIQO formulation by Eqs. (13)–(17) using tangent lines selected by each of the following methods, where $h$ is the number of tangent lines to be selected.

**EqlSpc($h$):** setting equally spaced tangent points

**AreaGrd($h$):** the greedy algorithm developed by Sato et al. [43]

**GapGrd($h$):** the greedy algorithm developed by Naganuma et al. [39]

**AdpGrd($h$):** our adaptive greedy algorithm (18)

**SmlOpt($h$):** our simultaneous optimization method by Eqs. (19)–(21)

We implemented these algorithms in the Python programming language. We set the input interval $[L, U] = [-5, 5]$ and use the asymptote by Eq. (6) as the initial tangent line. We use the Python `scipy.optimize` package (`method='SLSQP'`) to solve the NLO problem by Eqs. (19)–(21). We use Gurobi Optimizer 8.1.1 (`https://www.gurobi.com/`) to solve the MIQO problem by Eqs. (13)–(17), and the indicator constraint to impose the logical implication of Eq. (15). We fix the $L_2$-regularization parameter to $\alpha = 0$ in Tables 1 and 4, and tune it through hold-out validation using the training instances in Tables 2 and 5.

We compare the performance of our method with the following sparse estimation algorithms:

**FwdStep:** forward stepwise Poisson regression [14, 36]

**L1-Rgl:** $L_1$-regularized Poisson regression [15]

We implemented these algorithms using the `step` function and the `glmnet` package [15] in the R programming language. We tune the $L_1$-regularization parameter such that the number of nonzero regression coefficients equals $\theta$, then select the corresponding subset of explanatory variables. All computations occurred on a Windows computer with an Intel Core i3-8100 CPU (3.50 GHz) and 8 GB of memory.

We use the following evaluation metrics to compare the performance of sparse estimation methods. Let $\hat{\lambda}_i$ be a predicted value based on Eq. (3) for $i \in N$, where $N$ is the index set of test instances. The magnitude of out-of-sample prediction errors is

$$\mathrm{RMSE} := \sqrt{\frac{1}{|N|} \sum_{i \in N} (y_i - \hat{\lambda}_i)^2},$$

and the number of correct class labels is

$$\mathrm{Accuracy} := \frac{|\{i \in N \mid y_i = k_i\}|}{|N|},$$

where $k_i$ is an integer maximizing $\Pr(Y = k_i \mid \hat{\lambda}_i)$; see also Eq. (2). Let $S^*$ and $\hat{S}$ respectively be true and selected subsets of explanatory variables. Note that the true subset by Eq. (22) is specified for only synthetic datasets. The accuracy of subset selection is quantified as

$$\text{Recall} := \frac{|S^* \cap \hat{S}|}{|S^*|}.$$

## Experimental design for synthetic datasets

Following prior studies [5, 21], we prepared synthetic datasets via the following steps. Here, we set the number of candidate explanatory variables as $p = 30$ and the maximum value of the count variable as $m = 10$.

First, we defined a vector of true regression coefficients as

$$\begin{aligned} &\boldsymbol{w}^* := (1, 0, 0, 1, 0, 0, 1, 0, 0, \dots, 1, 0, 0)^\top \in \mathbb{R}^{30}, \\ &S^* := \{1, 4, 7 \dots, 28\} \quad (\text{i.e., } |S^*| = 10). \end{aligned} \tag{22}$$

We next sampled explanatory variables from a normal distribution as $\boldsymbol{x}_i \sim \mathrm{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma} \in \mathbb{R}^{30 \times 30}$ is the covariance matrix. The $(i, j)$th entry of $\boldsymbol{\Sigma}$ is $\rho^{|i-j|}$, where $\rho$ represents the correlation strength between explanatory variables. We also sampled the error term from a normal distribution as $\varepsilon_i \sim \mathrm{N}(0, \sigma^2)$, where $\sigma$ is the standard deviation. We then generated the count variable $y_i \in \{0\} \cup [10]$ by rounding

$$\exp\left(\frac{(\boldsymbol{w}^*)^\top \boldsymbol{x}_i}{\sqrt{(\boldsymbol{w}^*)^\top \boldsymbol{\Sigma} \boldsymbol{w}^*}} + \varepsilon_i\right)$$

to the nearest integer. We tested $\rho \in \{0.35, 0.70\}$ and $\sigma^2 \in \{0.01, 0.10, 1.00\}$ in the experiments.

Training instances were used to train sparse Poisson regression models, with 100 training instances. We estimated prediction performance by applying the trained regression model to sufficiently many test instances. The tables show average values for 10 repetitions, with standard errors in parentheses.

## Results for synthetic datasets

Table 1 shows the results of our MIQO formulation for the synthetic training instances with subset size $\theta = 10$. The column labeled "LogLkl" shows the log-likelihood value (4), which was maximized using a selected subset of explanatory variables. The largest log-likelihood values for each problem instance $(\sigma^2, \rho)$ are shown in bold. The columns labeled "Time (s)" show computation times in seconds required for solving the MIQO problem (MIQO) and for selecting tangent lines (TngLine).

Our adaptive greedy algorithm (AdpGrd) attained the largest log-likelihood values for all problem instances but required long computation times to select tangent lines. Our simultaneous optimization method (SmlOpt), on the other hand, selected tangent lines very quickly and also provided the second-best log-likelihood values except for $(\sigma^2, \rho) = (0.01, 0.70)$. These results clearly show that our AdpGrd and SmlOpt methods can find sparse regression models of better quality than do the conventional AreaGrd and GapGrd methods.

Table 2 shows the prediction performance of sparse Poisson regression models for synthetic test instances with subset size $\theta = 10$. The best RMSE, accuracy, and recall values for each problem instance $(\sigma^2, \rho)$ are shown in bold.

When $\sigma^2 \in \{0.01, 0.10\}$, our AdpGrd and SmlOpt methods delivered better prediction performance than did the FwdStep and L1-Rgl algorithms for all problem

**Table 1. Results of our MIQO formulation for synthetic training instances ($\theta = 10$).**

| | | | | | Time (s) | | | |
|---|---|---|---|---|---|---|---|---|
| $\sigma^2$ | $\rho$ | Method | LogLkl | | MIQO | | TngLine | |
| 0.01 | 0.35 | EqlSpc(10) | −105.00 | (±0.62) | 0.36 | (±0.01) | 0.00 | (±0.00) |
| | | AreaGrd(10) | −105.16 | (±0.78) | 0.48 | (±0.06) | 0.23 | (±0.00) |
| | | GapGrd(10) | −106.69 | (±0.84) | 0.54 | (±0.07) | 0.53 | (±0.00) |
| | | AdpGrd(10) | −**102.25** | (±0.53) | 0.40 | (±0.01) | 18.46 | (±0.03) |
| | | SmlOpt(10) | −103.99 | (±0.63) | 0.39 | (±0.02) | 0.08 | (±0.00) |
| | 0.70 | EqlSpc(10) | −107.37 | (±0.96) | 2.37 | (±0.88) | 0.00 | (±0.00) |
| | | AreaGrd(10) | −109.83 | (±0.74) | 5.03 | (±1.26) | 0.23 | (±0.00) |
| | | GapGrd(10) | −111.34 | (±1.04) | 3.98 | (±0.79) | 0.53 | (±0.00) |
| | | AdpGrd(10) | −**105.22** | (±0.86) | 0.55 | (±0.06) | 18.48 | (±0.03) |
| | | SmlOpt(10) | −107.78 | (±1.02) | 3.44 | (±1.09) | 0.08 | (±0.00) |
| 0.10 | 0.35 | EqlSpc(10) | −109.65 | (±1.19) | 0.47 | (±0.03) | 0.00 | (±0.00) |
| | | AreaGrd(10) | −110.51 | (±1.16) | 0.65 | (±0.06) | 0.24 | (±0.00) |
| | | GapGrd(10) | −113.05 | (±0.59) | 1.06 | (±0.17) | 0.53 | (±0.00) |
| | | AdpGrd(10) | −**107.30** | (±1.26) | 0.46 | (±0.02) | 18.46 | (±0.03) |
| | | SmlOpt(10) | −108.81 | (±1.27) | 0.55 | (±0.05) | 0.08 | (±0.00) |
| | 0.70 | EqlSpc(10) | −108.93 | (±1.37) | 2.98 | (±0.92) | 0.00 | (±0.00) |
| | | AreaGrd(10) | −110.82 | (±1.42) | 6.33 | (±1.00) | 0.23 | (±0.00) |
| | | GapGrd(10) | −112.60 | (±1.32) | 5.28 | (±1.12) | 0.52 | (±0.00) |
| | | AdpGrd(10) | −**106.20** | (±1.17) | 1.31 | (±0.25) | 18.44 | (±0.04) |
| | | SmlOpt(10) | −107.96 | (±1.29) | 3.55 | (±0.69) | 0.08 | (±0.00) |
| 1.00 | 0.35 | EqlSpc(10) | −148.55 | (±4.03) | 4.61 | (±1.57) | 0.00 | (±0.00) |
| | | AreaGrd(10) | −150.45 | (±3.75) | 5.88 | (±1.99) | 0.23 | (±0.00) |
| | | GapGrd(10) | −155.41 | (±3.54) | 2.98 | (±0.86) | 0.52 | (±0.01) |
| | | AdpGrd(10) | −**146.51** | (±3.84) | 3.52 | (±1.76) | 18.50 | (±0.03) |
| | | SmlOpt(10) | −148.41 | (±3.88) | 4.35 | (±1.52) | 0.08 | (±0.00) |
| | 0.70 | EqlSpc(10) | −151.37 | (±3.67) | 6.38 | (±1.43) | 0.00 | (±0.00) |
| | | AreaGrd(10) | −153.25 | (±3.56) | 8.58 | (±1.41) | 0.23 | (±0.00) |
| | | GapGrd(10) | −154.34 | (±4.24) | 4.21 | (±0.90) | 0.53 | (±0.00) |
| | | AdpGrd(10) | −**149.30** | (±3.55) | 6.48 | (±0.78) | 18.47 | (±0.04) |
| | | SmlOpt(10) | −150.80 | (±3.51) | 6.37 | (±1.04) | 0.08 | (±0.00) |

instances. In contrast, L1-Rgl performed very well when $(\sigma^2, \rho) = (1.00, 0.70)$. These results suggest that especially in low-noise situations, our MIO-based sparse estimation methods can deliver superior prediction performance as compared with heuristic algorithms such as stepwise selection and $L_1$-regularized estimation. This observation is consistent with the simulation results reported by Hastie et al. [21].

## Experimental design for real-world datasets

Table 3 lists real-world datasets downloaded from the UCI Machine Learning Repository [13], where $n$ and $p$ are numbers of data instances and candidate explanatory variables, respectively. In a preprocessing step, we divided the total number of rental bikes by $d$, rounding down to the nearest integer to be an appropriate scale for the count variable to be predicted. We transformed each categorical variable into a set of dummy variables. Note that variables "dteday," "casual," and "registered" are not suitable for prediction purposes and thus were

**Table 2. Prediction performance for synthetic test instances ($\theta = 10$)**

| $\sigma^2$ | $\rho$ | Method | RMSE | | Accuracy | | Recall | | Time (s) | |
|---|---|---|---|---|---|---|---|---|---|---|
| 0.01 | 0.35 | AdpGrd(30) | **0.524** | (±0.042) | **0.502** | (±0.019) | **1.000** | (±0.000) | 455.61 | (±2.96) |
| | | SmlOpt(30) | 0.566 | (±0.055) | 0.492 | (±0.018) | **1.000** | (±0.000) | 38.42 | (±2.97) |
| | | FwdStep | 0.644 | (±0.059) | 0.490 | (±0.018) | 0.980 | (±0.013) | 0.67 | (±0.02) |
| | | L1-Rgl | 0.908 | (±0.043) | 0.474 | (±0.010) | 0.910 | (±0.028) | 0.08 | (±0.00) |
| | 0.70 | AdpGrd(30) | 0.497 | (±0.032) | 0.520 | (±0.029) | **1.000** | (±0.000) | 1664.84 | (±225.86) |
| | | SmlOpt(30) | **0.490** | (±0.024) | **0.526** | (±0.032) | **1.000** | (±0.000) | 1166.14 | (±184.21) |
| | | FwdStep | 0.733 | (±0.053) | 0.497 | (±0.020) | 0.870 | (±0.021) | 0.73 | (±0.02) |
| | | L1-Rgl | 0.885 | (±0.040) | 0.479 | (±0.015) | 0.620 | (±0.055) | 0.07 | (±0.00) |
| 0.10 | 0.35 | AdpGrd(30) | **0.888** | (±0.021) | **0.492** | (±0.022) | **1.000** | (±0.000) | 468.09 | (±6.20) |
| | | SmlOpt(30) | 0.911 | (±0.022) | 0.487 | (±0.017) | **1.000** | (±0.000) | 40.94 | (±4.13) |
| | | FwdStep | 1.147 | (±0.157) | 0.461 | (±0.016) | 0.990 | (±0.010) | 0.70 | (±0.04) |
| | | L1-Rgl | 1.169 | (±0.103) | 0.444 | (±0.011) | 0.890 | (±0.028) | 0.07 | (±0.00) |
| | 0.70 | AdpGrd(30) | **1.087** | (±0.137) | **0.479** | (±0.013) | **0.940** | (±0.031) | 1742.37 | (±354.82) |
| | | SmlOpt(30) | 1.144 | (±0.142) | 0.467 | (±0.011) | 0.930 | (±0.033) | 959.33 | (±230.95) |
| | | FwdStep | 1.312 | (±0.158) | 0.446 | (±0.007) | 0.820 | (±0.025) | 0.71 | (±0.02) |
| | | L1-Rgl | 1.169 | (±0.039) | 0.455 | (±0.008) | 0.610 | (±0.043) | 0.07 | (±0.00) |
| 1.00 | 0.35 | AdpGrd(30) | 2.342 | (±0.145) | **0.356** | (±0.006) | **0.700** | (±0.030) | 584.74 | (±35.61) |
| | | SmlOpt(30) | 2.378 | (±0.153) | 0.352 | (±0.006) | 0.690 | (±0.031) | 100.76 | (±19.78) |
| | | FwdStep | 2.293 | (±0.096) | **0.356** | (±0.006) | 0.690 | (±0.041) | 0.86 | (±0.04) |
| | | L1-Rgl | **2.133** | (±0.055) | 0.352 | (±0.008) | 0.610 | (±0.043) | 0.07 | (±0.00) |
| | 0.70 | AdpGrd(30) | 2.530 | (±0.096) | 0.354 | (±0.005) | 0.460 | (±0.022) | 804.62 | (±72.09) |
| | | SmlOpt(30) | 2.457 | (±0.086) | 0.356 | (±0.004) | 0.470 | (±0.026) | 296.92 | (±52.32) |
| | | FwdStep | 2.307 | (±0.067) | 0.363 | (±0.004) | 0.540 | (±0.027) | 0.84 | (±0.05) |
| | | L1-Rgl | **2.097** | (±0.040) | **0.375** | (±0.003) | **0.550** | (±0.027) | 0.07 | (±0.00) |

removed. Data instances having outliers or missing values were eliminated.

**Table 3. Real-world datasets**

| Abbr. | $n$ | $p$ | $d$ | Original dataset [13] |
|---|---|---|---|---|
| Bike-H | 17,379 | 33 | 100 | Bike-sharing dataset (hour) |
| Bike-D | 731 | 33 | 1000 | Bike-sharing dataset (day) |

Training instances were randomly sampled, with 500 training instances for the Bike-H dataset and 365 for the Bike-D dataset. We used the remaining instances as test instances. The tables show averaged values for 10 trials, with standard errors in parentheses.

## Results for real-world datasets

Table 4 gives the results of our MIQO formulation for the real-world training instances with subset size $\theta \in \{5, 10\}$. As with the synthetic training instances (Table 1), our adaptive greedy algorithm AdpGrd achieved the largest log-likelihood values, but with long computation times. Our simultaneous optimization method SmlOpt was much faster than AdpGrd and provided good log-likelihood values for both the Bike-H and Bike-D datasets.

Table 5 shows the prediction performance of sparse Poisson regression models for the real-world test instances with subset size $\theta \in \{5, 10\}$. Our AdpGrd and SmlOpt

**Table 4. Results of our MIQO formulation for real-world training instances.**

| | | | | Time (s) | |
|---|---|---|---|---|---|
| Dataset | $\theta$ | Method | LogLkl | MIQO | TngLine |
| Bike-H | 5 | EqlSpc(10) | −744.91 (±7.70) | 5.87 (±0.72) | 0.00 (±0.00) |
| | | AreaGrd(10) | −785.15 (±28.70) | 6.27 (±0.75) | 0.23 (±0.00) |
| | | GapGrd(10) | −938.96 (±22.97) | 1.61 (±0.59) | 0.53 (±0.00) |
| | | AdpGrd(10) | −**742.98** (±7.58) | 8.23 (±0.87) | 94.13 (±1.11) |
| | | SmlOpt(10) | −745.66 (±7.70) | 5.54 (±0.49) | 0.08 (±0.00) |
| | 10 | EqlSpc(10) | −730.67 (±7.97) | 69.47 (±23.99) | 0.00 (±0.00) |
| | | AreaGrd(10) | −739.34 (±7.82) | 116.71 (±30.54) | 0.23 (±0.00) |
| | | GapGrd(10) | −896.40 (±29.85) | 10.42 (±4.22) | 0.53 (±0.00) |
| | | AdpGrd(10) | −**728.35** (±7.77) | 67.75 (±15.86) | 93.40 (±0.86) |
| | | SmlOpt(10) | −731.52 (±7.90) | 54.56 (±13.63) | 0.08 (±0.00) |
| Bike-D | 5 | EqlSpc(10) | −784.89 (±3.18) | 1.55 (±0.31) | 0.00 (±0.00) |
| | | AreaGrd(10) | −795.69 (±15.86) | 0.74 (±0.28) | 0.23 (±0.00) |
| | | GapGrd(10) | −755.64 (±28.97) | 0.96 (±0.11) | 0.54 (±0.01) |
| | | AdpGrd(10) | −**634.00** (±17.10) | 6.84 (±0.62) | 71.24 (±2.39) |
| | | SmlOpt(10) | −720.46 (±7.90) | 2.32 (±0.46) | 0.08 (±0.00) |
| | 10 | EqlSpc(10) | −783.87 (±3.19) | 2.98 (±1.79) | 0.00 (±0.00) |
| | | AreaGrd(10) | −780.44 (±2.53) | 4.35 (±4.01) | 0.23 (±0.00) |
| | | GapGrd(10) | −754.38 (±29.08) | 0.50 (±0.13) | 0.54 (±0.01) |
| | | AdpGrd(10) | −**626.22** (±16.72) | 123.06 (±23.66) | 70.77 (±2.39) |
| | | SmlOpt(10) | −698.47 (±14.19) | 9.69 (±4.42) | 0.08 (±0.00) |

methods were superior to the FwdStep and L1-Rgl algorithms in terms of RMSE values for the Bike-H dataset and accuracy values for the Bike-D dataset. FwdStep gave the best accuracy values for the Bike-H dataset, whereas there was no clear best or worst method regarding RMSE values for the Bike-D dataset.

**Table 5. Prediction performance for real-world test instances**

| Dataset | $\theta$ | Method | RMSE | Accuracy | Time (s) |
|---|---|---|---|---|---|
| Bike-H | 5 | AdpGrd(30) | **1.491** (±0.004) | 0.408 (±0.004) | 2530.03 (±64.29) |
| | | SmlOpt(30) | **1.491** (±0.004) | 0.407 (±0.004) | 240.69 (±31.57) |
| | | FwdStep | 1.494 (±0.005) | **0.414** (±0.002) | 1.61 (±0.07) |
| | | L1-Rgl | 1.495 (±0.004) | 0.405 (±0.003) | 0.08 (±0.00) |
| | 10 | AdpGrd(30) | **1.488** (±0.007) | 0.410 (±0.003) | 8504.38 (±951.32) |
| | | SmlOpt(30) | 1.489 (±0.007) | 0.410 (±0.003) | 2189.76 (±478.19) |
| | | FwdStep | 1.509 (±0.007) | **0.416** (±0.003) | 1.61 (±0.07) |
| | | L1-Rgl | 1.491 (±0.005) | 0.415 (±0.002) | 0.05 (±0.00) |
| Bike-D | 5 | AdpGrd(30) | 0.996 (±0.011) | 0.334 (±0.009) | 1806.09 (±13.37) |
| | | SmlOpt(30) | 0.991 (±0.011) | **0.338** (±0.007) | 146.13 (±6.46) |
| | | FwdStep | **0.989** (±0.009) | 0.335 (±0.008) | 1.13 (±0.03) |
| | | L1-Rgl | 1.011 (±0.008) | 0.319 (±0.008) | 0.08 (±0.00) |
| | 10 | AdpGrd(30) | 0.963 (±0.011) | **0.353** (±0.004) | 6451.01 (±438.40) |
| | | SmlOpt(30) | 0.958 (±0.010) | **0.353** (±0.005) | 1758.75 (±284.93) |
| | | FwdStep | 0.964 (±0.010) | 0.349 (±0.006) | 1.13 (±0.03) |
| | | L1-Rgl | **0.956** (±0.011) | 0.349 (±0.005) | 0.05 (±0.00) |

# Conclusion

This paper presented an MIO approach to sparse Poisson regression, which we formulated as an MIQO problem by applying piecewise-linear approximation to the nonlinear objective function. We also developed the adaptive greedy algorithm and the simultaneous optimization method to select a limited number of tangent lines that work well for piecewise-linear approximations.

We conducted computational experiments using synthetic and real-world datasets. Our methods for selecting tangent lines clearly outperformed conventional methods in terms of the quality of piecewise-linear approximations. For the synthetic datasets, our MIQO formulation delivered better prediction performance than did stepwise selection and $L_1$-regularized estimation, especially in low-noise situations. Our MIQO formulation also compared favorably in terms of prediction performance with the other algorithms for real-world datasets.

Although our method can potentially find good-quality sparse regression models, applying it to large datasets is computationally expensive. It is more practical to choose between our method and heuristic algorithms according to the task at hand. We also expect our framework for piecewise-linear approximations to work well for various decision-making problems involving univariate nonlinear functions.

A future direction of study will be to develop an efficient algorithm specialized for solving our MIQO problem. We are now working on extending several MIO-based high-performance algorithms [5, 7, 29] to sparse Poisson regression. Another direction of future research is to improve the computational performance of our methods for selecting tangent lines.

# References

1. Algamal, Z. (2019). Variable selection in count data regression model based on firefly algorithm. Statistics, Optimization & Information Computing, 7(2), 520–529.

2. Arthanari, T. S., & Dodge. Y. (1981). Mathematical Programming in Statistics, New York: Wiley.

3. Bertsimas, D., & King, A. (2016). An algorithmic approach to linear regression. Operations Research, 64(1), 2–16.

4. Bertsimas, D., & King, A. (2017). Logistic regression: From art to science. Statistical Science, 32(3), 367–384.

5. Bertsimas, D., King, A., & Mazumder, R. (2016). Best subset selection via a modern optimization lens. The annals of statistics, 813-852.

6. Bertsimas, D., & Li, M. L. (2020). Scalable holistic linear regression. Operations Research Letters.

7. Bertsimas, D., Pauphilet, J., & Van Parys, B. (2019). Sparse regression: Scalable algorithms and empirical performance. arXiv preprint arXiv:1902.06547.

8. Cameron, A. C., & Trivedi, P. K. (2013). Regression analysis of count data (Vol. 53). Cambridge university press.

9. Chan, A. B., & Vasconcelos, N. (2009, September). Bayesian Poisson regression for crowd counting. In 2009 IEEE 12th international conference on computer vision (pp. 545–551). IEEE.

10. Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. Computers & Electrical Engineering, 40(1), 16–28.

11. Coxe, S., West, S. G., & Aiken, L. S. (2009). The analysis of count data: A gentle introduction to Poisson regression and its alternatives. Journal of personality assessment, 91(2), 121–136.

12. Cozad, A., Sahinidis, N. V., & Miller, D. C. (2014). Learning surrogate models for simulation-  based optimization. AIChE Journal, 60(6), 2211–2227.

13. Dua, D., & Graff, C. (2019). UCI Machine Learning Repository [`http://archive.ics.uci.edu/ml`]. Irvine, CA: University of California, School of Information and Computer Science.

14. Efroymson, M. A. (1960). Multiple regression analysis. Mathematical methods for digital computers, 191–203.

15. Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. Journal of Statistical Software, 33(1), 1.

16. Frommlet, F., & Nuel, G. (2016). An adaptive ridge procedure for $L_0$ regularization. PloS one, 11(2), e0148620.

17. Gardner, W., Mulvey, E. P., & Shaw, E. C. (1995). Regression analyses of counts and rates: Poisson, overdispersed Poisson, and negative binomial models. Psychological bulletin, 118(3), 392–404.

18. Gómez, A., & Prokopyev, O. (2018). A mixed-integer fractional optimization approach to best subset selection. Optimization Online.

19. Guastavino, S., & Benvenuto, F. (2019). A consistent and numerically efficient variable selection method for sparse Poisson regression with applications to learning and signal recovery. Statistics and Computing, 29(3), 501–516.

20. Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. Journal of Machine Learning Research, 3(Mar), 1157–1182.

21. Hastie, T., Tibshirani, R., & Tibshirani, R. J. (2017). Extended comparisons of best subset selection, forward stepwise selection, and the lasso. arXiv preprint arXiv:1707.08692.

22. Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. Technometrics, 12(1), 55–67.

23. Ivanoff, S., Picard, F., & Rivoirard, V. (2016). Adaptive Lasso and group-Lasso for functional Poisson regression. The Journal of Machine Learning Research, 17(1), 1903–1948.

24. Jia, J., Xie, F., & Xu, L. (2019). Sparse Poisson regression with penalized weighted score function. Electronic Journal of Statistics, 13(2), 2898–2920.

25. Kimura, K. (2019). Application of a mixed integer nonlinear programming approach to variable selection in logistic regression. Journal of the Operations Research Society of Japan, 62(1), 15–36.

26. Kimura, K., & Waki, H. (2018). Minimization of Akaike's information criterion in linear regression analysis via mixed integer nonlinear program. Optimization Methods and Software, 33(3), 633–649.

27. Koç, H., Dünder, E., Gümüştekin, S., Koç, T., & Cengiz, M. A. (2018). Particle swarm optimization-based variable selection in Poisson regression analysis via information complexity-type criteria. Communications in Statistics—Theory and Methods, 47(21), 5298–5306.

28. Konno, H., & Yamamoto, R. (2009). Choosing the best set of variables in regression analysis using integer programming. Journal of Global Optimization, 44(2), 273–282.

29. Kudo, K., Takano, Y., & Nomura, R. (2020). Stochastic discrete first-order algorithm for feature subset selection. IEICE Transactions on Information and Systems, 103(7), 1693–1702.

30. Lambert, D. (1992). Zero-inflated Poisson regression, with an application to defects in manufacturing. Technometrics, 34(1), 1–14.

31. Lawless, J. F., & Singhal, K. (1978). Efficient screening of nonnormal regression models. Biometrics, 318–327.

32. Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., & Liu, H. (2017). Feature selection: A data perspective. ACM Computing Surveys (CSUR), 50(6), 1–45.

33. Lindsey, C., & Sheather, S. (2015). Best subsets variable selection in nonnormal regression models. The Stata Journal, 15(4), 1046–1059.

34. Liu, H., & Motoda, H. (Eds.). (2007). Computational methods of feature selection. CRC Press.

35. Maldonado, S., Pérez, J., Weber, R., & Labbé, M. (2014). Feature selection for support vector machines via mixed integer linear programming. Information sciences, 279, 163–175.

36. Miller, A. (2002). Subset selection in regression. CRC Press.

37. Miyashiro, R., & Takano, Y. (2015). Subset selection by Mallows' $C_p$: A mixed integer programming approach. Expert Systems with Applications, 42(1), 325–331.

38. Miyashiro, R., & Takano, Y. (2015). Mixed integer second-order cone programming formulations for variable selection in linear regression. European Journal of Operational Research, 247(3), 721–731.

39. Naganuma, M., Takano, Y., & Miyashiro, R. (2019). Feature subset selection for ordered logit model via tangent-plane-based approximation. IEICE Transactions on Information and Systems, 102(5), 1046–1053.

40. Nakaya, T., Fotheringham, A. S., Brunsdon, C., & Charlton, M. (2005). Geographically weighted Poisson regression for disease association mapping. Statistics in medicine, 24(17), 2695–2717.

41. Park, Y. W., & Klabjan, D. (2020). Subset selection for multiple linear regression via optimization. Journal of Global Optimization, 1–32.

42. Sato, T., Takano, Y., & Miyashiro, R. (2017). Piecewise-linear approximation for feature subset selection in a sequential logit model. Journal of the Operations Research Society of Japan, 60(1), 1–14.

43. Sato, T., Takano, Y., Miyashiro, R., & Yoshise, A. (2016). Feature subset selection for logistic regression via mixed integer optimization. Computational Optimization and Applications, 64(3), 865–880.

44. Takano, Y., & Miyashiro, R. (2020). Best subset selection via cross-validation criterion. TOP, 28(2), 475–488.

45. Tamura, R., Kobayashi, K., Takano, Y., Miyashiro, R., Nakata, K., & Matsui, T. (2017). Best subset selection for eliminating multicollinearity. Journal of the Operations Research Society of Japan, 60(3), 321–336.

46. Tamura, R., Kobayashi, K., Takano, Y., Miyashiro, R., Nakata, K., & Matsui, T. (2019). Mixed integer quadratic optimization formulations for eliminating multicollinearity based on variance inflation factor. Journal of Global Optimization, 73(2), 431–446.

47. Ustun, B., & Rudin, C. (2016). Supersparse linear integer models for optimized medical scoring systems. Machine Learning, 102(3), 349–391.

48. Wang, Z., Ma, S., Zappitelli, M., Parikh, C., Wang, C. Y., & Devarajan, P. (2016). Penalized count data regression with application to hospital stay after pediatric cardiac surgery. Statistical methods in medical research, 25(6), 2685–2703.

49. Ye, X., Wang, K., Zou, Y., & Lord, D. (2018). A semi-nonparametric Poisson regression model for analyzing motor vehicle crash data. PloS one, 13(5), e0197338.

**Fig. 1**

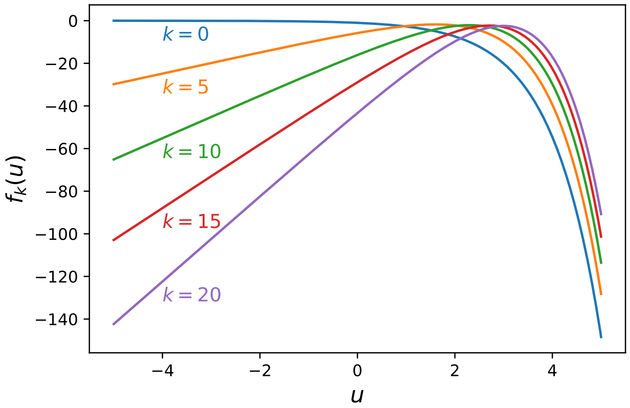
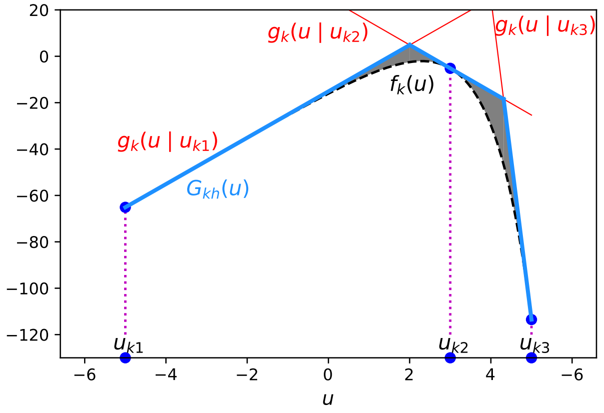


**Fig. 2**



$g_k(u \mid u_{k2})$
$g_k(u \mid u_{k3})$
$f_k(u)$
$g_k(u \mid u_{k1})$
$G_{kh}(u)$
$u_{k1}$
$u_{k2}$
$u_{k3}$
20
0
−20
−40
−60
−80
−100
−120
−6
−4
−2
0
2
4
−6
$u$