Efficient Classification Methods for Sparse High-Dimensional Count Data with Model Selection as Motivated by Microbe Finder

Huizi Wang

Advisor: Dr. Joshua D Habiger
Committee: Dr. Lan Zhu, Dr. Ye Liang
Outside committee: Dr. Kitty Cardwell, Institute of Biosecurity and Microbial Forensics, Department of Entomology and Plant Pathology, Oklahoma State University

Table of Contents

Motivation

- It is important to detect pathogens in citrus. For example,
  - the California citrus industry faces various pathogens when grafting with infected materials (Babcock, 2022).
  - the cost of this impact is valued at approximately $3.6 billion, with an economic impact of $7.6 billion in the state of California from the year 2020 to 2021 (Dang et al., 2023; Sohrab et al., 2024).
- Microbe Finder® (MiFi) is an online platform, developed by researchers at the Institute of Biosecurity and Microbial Forensics at Oklahoma State University, which uses high-throughput sequencing (HTS) technology to detect and measure the abundance of a pathogen in a sample (Espindola and Cardwell, 2021; Cardwell et al., 2018; Stobbe et al., 2013).

Data Generation

- The MiFi platform can be used to generate MiFi data for pathogen detection. We summarize the steps below. See Espindola et al. (2022) and Espindola and Cardwell (2021) for details.
  1. E-probe design: First, the unique RNA sequence of a pathogen is decomposed into multiple smaller sequences called e-probes. For example, an e-probe of Citrus Psorosis Virus pathogen is GATAATTCATCTGTTATTGCAGAGAACAGT.
  2. Plant/biological sample collection: The RNA sequence of a biological sample to be tested is collected and uploaded into the MiFi platform.
  3. BLASTn procedure: MiFi scans between the RNA sequence of a sample and a set of validated e-probes of interest using the BLASTn procedure.
  4. The number of hits for each e-probe is recorded, i.e. the number of times that an e-probe sequence was found in a sample is recorded for each e-probe. Also, the "total score" for each sample is recorded. The total score represents the total number of hits across e-probes in a sample and incorporates hit quality. That is, it is a weighted sum, with weights being determined by hit quality.

Data Example

- MiFi was used to generate the Candidatus Liberibacter Asiaticus (CLas) data set.
  - X_{ij}: The number of hits for a sample i in e-probe j.
  - Z_i: The total score for the sample i. It is continuous and incorporates number of hits and quality of each hit.

Sample	Y_i	X_{ij}				Z_i
		X_{i1}	X_{i2}	$\cdots$	X_{i8847}	
1	0	12	13	$\cdots$	0	254.1
2	0	1	1	$\cdots$	0	37.88
3	0	11	19	$\cdots$	0	169.93
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
21	1	2	17	$\cdots$	0	881.2
22	1	3	8	$\cdots$	0	393.7

Table 1: CLas data example. The number of e-probes is 8847. There are 11 healthy ($Y_i = 0$) and 11 pathogenic ($Y_i = 1$) samples.

Challenges

This dissertation focuses on high-dimensional (HD) classification methods with count data.

- One challenge is that data may be zero-inflated and/or over-dispersed.
  - See Figures 1 and 2.
- A second challenge is that data are HD, i.e. the number of variables/e-probes is much higher than the sample size because it can lead to overfitting, i.e. this can result in models that are too complex and highly variable.
  - There are 8847 e-probes, but there are only 22 samples for the CLas data set in Table 1.
  - Table 2 shows that the number of e-probes ranges from 6 to 8847, while the sample size ranges from 15 to 22.
- A third challenge is computational.
  - For example, multiple logistic regression (MLR) requires an iterative procedure to get maximum likelihood estimates (MLEs).

Zero-inflation

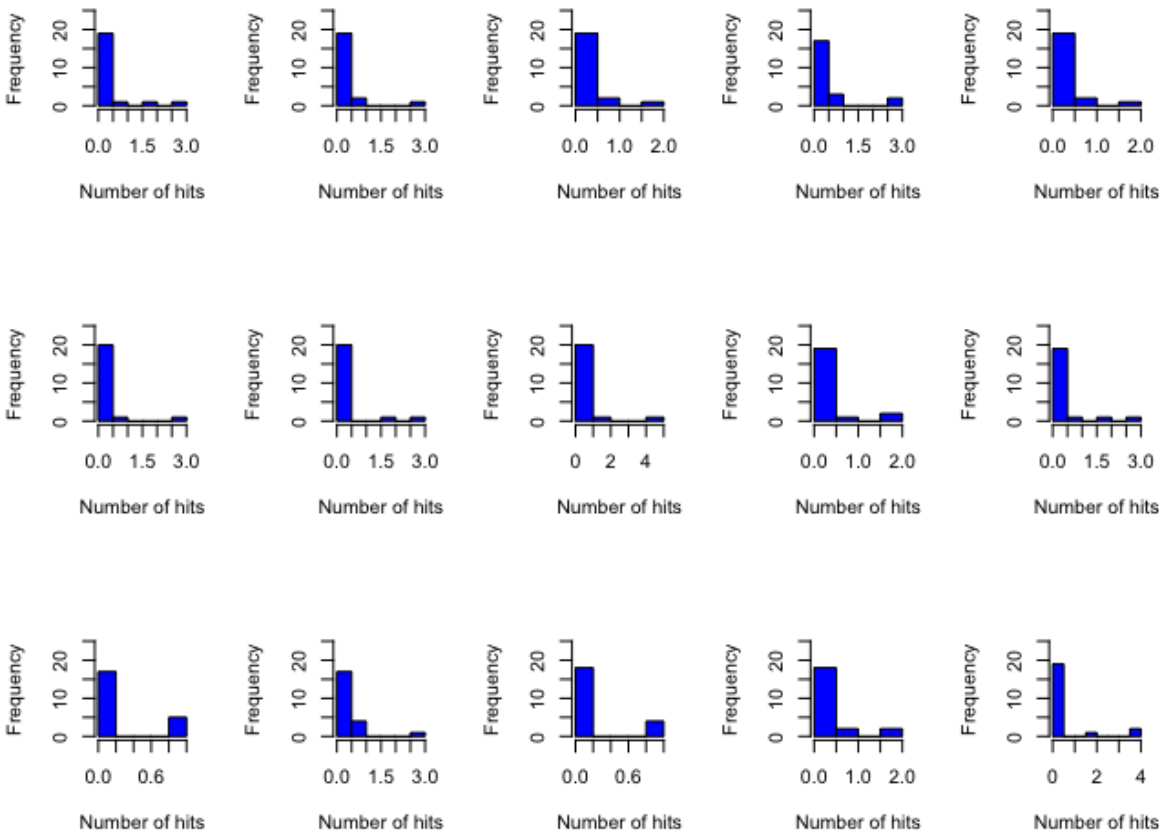

Figure 1: The proportion of hits plot plotted number of hits versus the frequency of hits for e-probe 10 to 24 of the CLas data set in Table 1.

Over-dispersion

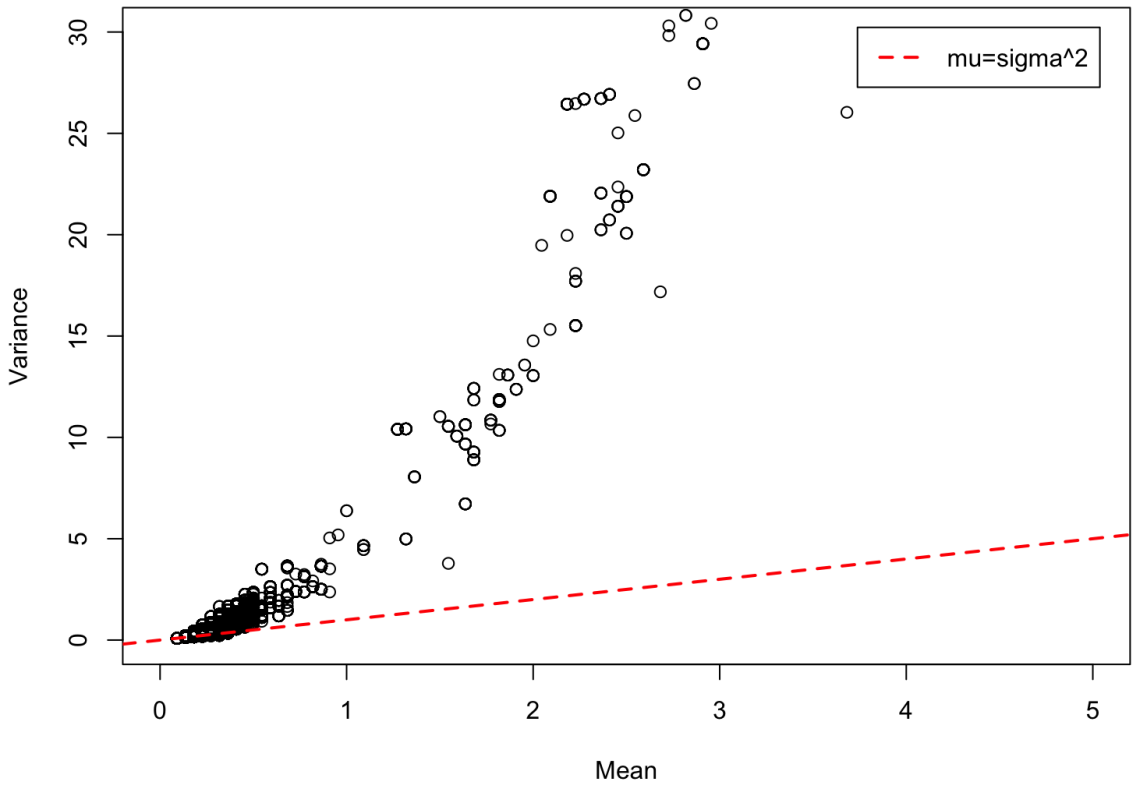

Figure 2: A plot of sample means versus sample variances for all e-probes in the CLas data set in Table 1. The 45-degree line represents the setting when the sample mean is equal to the sample variance.

Data are HD

Pathogen Names	# E-probes	# Samples
Candidatus Liberibacter Asiaticus	8847	22
Citrus Tristeza Virus	823	15
Citrus Psorosis Virus	203	21
Citrus Virus A	151	21
Citrus Vein Enation Virus	90	19
Citrus Variegated Virus	86	22
Citrus Tatter Leaf Virus	51	20
Citrus Leaf Blotch Virus	27	17
Citrus Yellow Vein Associated Virus	24	22
Citrus Exocortis Viroid	6	18

Table 2: The summary of MiFi data sets.

- Two main approaches of HD classification methods with count data are
  - penalized multiple logistic regression (MLR) approach (Hastie et al., 2015, Chapter 3).
  - multivariate discriminant analysis (MDA) approach
    - Poisson linear discriminant analysis (Witten, 2011),
    - Zero-Inflated Poisson logistic discriminant analysis (Zhou et al., 2018),
    - Negative Binomial linear discriminant analysis (Dong et al., 2016),
    - Zero-Inflated Negative Binomial logistic discriminant analysis (Zhu et al., 2021).

Notations

- Let $\boldsymbol{X}$ denote an $n \times p$ matrix of counts with n experiment units and p variables.
- Let $\boldsymbol{Y}$ be a vector of length n with $Y_i = 0$ if the sample is healthy and $Y_i = 1$ otherwise.
- Let X_{ij} be the number of hits for a sample i of e-probe j, for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, p$.
- Denote $Pr(X_1 = x_1, X_2 = x_2, \ldots, X_p = x_p, Y = y)$ by $p(x_1, x_2, \ldots, x_p, y) = p(\boldsymbol{x}, y)$.
- Denote $Pr(X_1 = x_1, X_2 = x_2, \ldots, X_p = x_p \mid Y = y)$ by $p(x_1, x_2, \ldots, x_p \mid y) = p(\boldsymbol{x} \mid y)$.
- Denote $Pr(Y = y \mid X_1 = x_1, X_2 = x_2, \ldots, X_p = x_p)$ by $p(y \mid x_1, x_2, \ldots, x_p) = p(y \mid \boldsymbol{x})$.
- Denote, μ_{ij} by

$$\mu_j(y_i) = \begin{cases} \mu_j(1), & \text{if} \quad y_i = 1, \\ \mu_j(0), & \text{if} \quad y_i = 0. \end{cases}$$

- Denote π_{ij} by

$$\pi_j(y_i) = \begin{cases} \pi_j(1), & \text{if} \quad y_i = 1, \\ \pi_j(0), & \text{if} \quad y_i = 0. \end{cases}$$

Penalized multiple logistic regression (MLR) model

- Specify the **MLR** model $Y \mid \boldsymbol{X} \sim \text{Bern}(p(1 \mid \boldsymbol{x}))$
- The **MLR** classifier is

$$C(\boldsymbol{x}) = \log\left[\frac{p(1 \mid \boldsymbol{x})}{p(0 \mid \boldsymbol{x})}\right] = \beta_0 + \sum_{j=1}^{p} \beta_{1j} x_j.$$

  We state $\hat{Y} = 1$ if $C(\boldsymbol{x}) > K$ for some K and $\hat{Y} = 0$ otherwise.
- The objective function is Log likelihood($\boldsymbol{\beta}$) + Penalty($\boldsymbol{\beta}$)
  - Log likelihood:

$$\ell(\beta_0, \beta) = \frac{1}{n}\sum_{i=1}^{n}\left[y_i\left(\beta_0 + \boldsymbol{x}_i^T\beta\right) - \log\left(1 + e^{\beta_0 + \boldsymbol{x}_i^T\beta}\right)\right] \quad (1)$$

  - Penalty
    - Ridge: $\lambda\|\beta\|_2^2$
    - Lasso: $\lambda\|\beta\|_1$
    - Elastic-net: $\lambda\left[\frac{1}{2}(1-\alpha)\|\beta\|_2^2 + \alpha\|\beta\|_1\right]$

Multivariate Discriminant Analysis (MDA) Approach Classifier

- Assume $Pr(\boldsymbol{X} = \boldsymbol{x} \mid Y = y) = \prod_{j=1}^{p} Pr(X_j = x_j \mid Y = y)$ where $\boldsymbol{X} = (X_1, X_2, \ldots, X_p)$.
- Specify models for $\boldsymbol{X} \mid Y$ such as
  - **Poisson**: $X_j \mid Y \overset{\text{ind}}{\sim} \text{POI}\Big(\mu_j(Y)\Big)$ where $\mu_j(0)$ is mean of X_j when $Y = 0$ and $\mu_j(1)$ is mean of X_j when $Y = 1$.
  - **Zero-Inflated Poisson (ZIP)**:
    $$X_j \mid Y \overset{\text{ind}}{\sim} \begin{cases} \text{POI}\Big(\mu_j(Y)\Big), & \text{with probability } 1 - \pi_j(Y), \\ 0, & \text{with probability } \pi_j(Y), \end{cases}$$
    where $\mu_j(0)$ is mean of X_j when $Y = 0$ and $\mu_j(1)$ is mean of X_j when $Y = 1$. Similarly, $\pi_j(0)$ is probability of $X_j = 0$ when $Y = 0$ and $\pi_j(1)$ is probability of $X_j = 0$ when $Y = 1$.

Multivariate Discriminant Analysis (MDA) Classifier

- Specify models for $\boldsymbol{X} \mid Y$ such as
  - **Negative Binomial (NB)**: $X_j \mid Y \overset{\text{ind}}{\sim} \text{NB}\Big(\mu_j(Y), \phi_j\Big)$ where $\mu_j(0)$ is mean of X_j when $Y = 0$ and $\mu_j(1)$ is mean of X_j when $Y = 1$, and ϕ_j is the dispersion of X_j.
  - **Zero-Inflated Negative Binomial (ZINB)**:
    $$X_j \mid Y \overset{\text{ind}}{\sim} \begin{cases} \text{NB}\Big(\mu_j(Y), \phi_j\Big), & \text{with probability} \quad 1 - \pi_j(Y), \\ 0, & \text{with probability} \quad \pi_j(Y), \end{cases} \text{ where}$$
    $\mu_j(0)$ is mean of X_j when $Y = 0$ and $\mu_j(1)$ is mean of X_j when $Y = 1$. Similarly, $\pi_j(0)$ is probability of $X_j = 0$ when $Y = 0$ and $\pi_j(1)$ is probability of $X_j = 0$ when $Y = 1$.
- The **MDA** classifier is $\hat{Y} = 1$ if

$$C(\boldsymbol{x}) \propto \log\left[\frac{p(\boldsymbol{x} \mid 1)}{p(\boldsymbol{x} \mid 0)}\right] > K$$

for some K and $\hat{Y} = 0$ otherwise.

Limitations

No one method exists that is computationally efficient, can model zero-inflation and over-dispersion, and allows for model/variable selection while avoiding overfitting.

- The penalized MLR approach
  - does not directly model zero-inflation or over-dispersion.
  - is not computationally efficient in that an iterative procedure is required.

Limitations

- The penalized MDA:
  - The Poisson Linear Discriminant Analysis (PLDA) classifier (Witten, 2011) CANNOT model zero-inflation and over-dispersion.
    - Plugging in ℓ_1-type penalized maximum likelihood estimators (MLEs) of $\mu_j(y)$'s.
  - The Zero-Inflated Poisson Logistic Discriminate Analysis (ZIPLDA) classifier (Zhou et al., 2018) CANNOT model over-dispersion and allow for model/variable selection and is not computationally efficient.
    - Employed ℓ_2-type penalized MLEs of $\mu_j(y)$'s and $\pi_j(y)$'s to avoid overfitting, but do not allow for model/variable selection (Risso et al., 2018).
    - Is not computationally efficient in that an iterative procedure is required.

Limitations

- The penalized MDA:
  - The Negative Binomial Linear Discriminant Analysis (NBLDA) classifier (Dong et al., 2016) CANNOT model zero-inflation and allow for model/variable selection.
    - Used shrinkage estimation (shrinkage ϕ_js toward a pre-defined target value), but do not allow for model/variable selection (Yu et al., 2013).
  - The Zero-Inflated Negative Binomial Logistic Discriminant Analysis (ZINBLDA) classifier (Zhu et al., 2021) CANNOT allow for model/variable selection and is not computationally efficient.
    - Employed ℓ_2-type penalized MLEs of $\mu_j(y)$'s, $\pi_j(y)$'s and ϕ_js to avoid overfitting, but do not allow for model/variable selection (Risso et al., 2018).
    - Is not computationally efficient in that an iterative procedure is required.

Merit Contributions

- **This dissertation provides methods for HD classification analysis that**
  1. can model zero-inflation and over-dispersion,
  2. allow for model/variable selection while avoiding overfitting, and
  3. are computationally efficient.
- **Idea:**
  - First, we show analytically that multivariate model classifiers can be written as MLR classifiers where the regression coefficients include parameters for zero-inflation and over-dispersion.
  - Then, we develop penalized method of moments (PMOM) estimators of the regression coefficients, where a soft-thresholding function applies an ℓ_1 penalty so that the methods allow for model/variable selection and avoid overfitting.
  - The methodology is computationally efficient because PMOM estimators use soft-thresholding functions and avoid iterative procedures.

Multivariate Model Classifiers can be Written as Multiple Logistic Regression Classifier

Definition 6.1 (Multiple Logistic Regression Classifier)

Assume $Y \mid \boldsymbol{X} \sim \text{Bern}(p(1 \mid \boldsymbol{x}))$ where

$$\log\left(\frac{p(1 \mid \boldsymbol{x})}{p(0 \mid \boldsymbol{x})}\right) = \beta_0 + \sum_{j=1}^{p} \beta_{1j} x_j + \sum_{j=1}^{p} \beta_{2j} \delta_0(x_j), \tag{2}$$

where δ_0 is the Dirac delta measure with the point mass 0. So that when $\beta_{2j} = 0 \ \forall j$,

$$\log\left(\frac{p(1 \mid \boldsymbol{x})}{p(0 \mid \boldsymbol{x})}\right) = \beta_0 + \sum_{j=1}^{p} \beta_{1j} x_j. \tag{3}$$

The classifier is $\hat{Y} = 1$ if $C(\boldsymbol{x}) = \log\left(\frac{p(1|\boldsymbol{x})}{p(0|\boldsymbol{x})}\right) > K$ for some K.

Multivariate Model Classifiers can be Written as Multiple Logistic Regression Classifier

Definition 6.2 (General Multivariate Model Classifier)

Assume $p(\boldsymbol{x} \mid y) = \prod_{j=1}^{p} p(x_j \mid y)$ where

$$p(x_j \mid y) = \Big(1 - \pi_j(y)\Big) p^*(x_j \mid y) + \pi_j(y)\delta_0(x_j) \tag{4}$$

where $\pi_j(y)$ is the probability of "0 inflation" when $Y = y$ taking values in $[0, 1]$ for all j, $p^*(x_j \mid y)$ is a component PMF on $\{0, 1, 2, \dots\}$ for all j, and δ_0 is the Dirac delta measure with the point mass 0. The classifier is $\hat{Y} = 1$ if $C(\boldsymbol{x}) \propto \log\left(\frac{p(\boldsymbol{x}|1)}{p(\boldsymbol{x}|0)}\right) > K$ for some K.

Multivariate Model Classifiers can be Written as Multiple Logistic Regression Classifier

Theorem 6.3

Suppose $\forall j\ p^*(x_j \mid y)$ is a PMF in Definition 6.2. Then, the classifier in Definition 6.2 is equivalent to the classifier in Definition 6.1, and it can be written as

$$
\begin{aligned}
C(\boldsymbol{x}) \propto & \sum_{j=1}^{p} \log\left(\frac{1-\pi_j(1)}{1-\pi_j(0)}\right) + \sum_{j=1}^{p} \log\left(\frac{p^*(x_j \mid 1)}{p^*(x_j \mid 0)}\right) \\
& + \sum_{j=1}^{p} \left[\log\left(\frac{\big(1-\pi_j(1)\big)p^*(0 \mid 1) + \pi_j(1)}{\big(1-\pi_j(0)\big)p^*(0 \mid 0) + \pi_j(0)}\right) - \log\left(\frac{1-\pi_j(1)}{1-\pi_j(0)}\right) - \log\left(\frac{p^*(x_j \mid 1)}{p^*(x_j \mid 0)}\right)\right] \delta_0(x_j) \\
& \equiv \beta_0 + \sum_{j=1}^{p} \beta_{1j} x_j + \sum_{j=1}^{p} \beta_{2j} \delta_0(x_j).
\end{aligned}
\tag{5}
$$

Multivariate Model Classifiers can be Written as Multiple Logistic Regression Classifier

Lemma 6.4

Equation (4) is equivalent to

$$p(x_j \mid y) = \Big[(1-\pi_j(y))p^*(0 \mid y) + \pi_j(y)\Big]^{\delta_0(x_j)} \Big[(1-\pi_j(y))p^*(x_j \mid y)\Big]^{1-\delta_0(x_j)}. \tag{6}$$

Corollary 6.5

Suppose conditions in Theorem 6.3 are satisfied. If additionally, $\pi_j(y) = 0 \ \forall j$ and $p^*(x_j \mid y)$ has a Poisson PMF. Then, the classifier in Definition 6.2 is equivalent to the classifier in Definition 6.1 in the sense that

$$\beta_{1j} = \log\left(\frac{\mu_j(1)}{\mu_j(0)}\right), \tag{7}$$

$$\beta_{2j} = 0, \tag{8}$$

and

$$\begin{aligned} \beta_0 &= \sum_{j=1}^{p} \beta_{0j} \\ &= \sum_{j=1}^{p} -\Big(\mu_j(1) - \mu_j(0)\Big) \end{aligned} \tag{9}$$

in Equation (5).

Corollary 6.6

Suppose conditions in Theorem 6.3 are satisfied. If additionally, $\pi_j(y) > 0 \ \forall j$ and $p^*(x_j \mid y)$ has a Poisson PMF. Then, the classifier in Definition 6.2 is equivalent to the classifier in Definition 6.1 in the sense that

$$\beta_{1j} = \log\left(\frac{\mu_j(1)}{\mu_j(0)}\right), \tag{10}$$

$$\begin{aligned}\beta_{2j} &= \log\left(\frac{\big(1-\pi_j(1)\big)e^{-\mu_j(1)}+\pi_j(1)}{\big(1-\pi_j(0)\big)e^{-\mu_j(0)}+\pi_j(0)}\right)+\Big(\mu_j(1)-\mu_j(0)\Big)-\log\left(\frac{1-\pi_j(1)}{1-\pi_j(0)}\right)\\ &= \log\left(\frac{\big(1-\pi_j(1)\big)e^{-\mu_j(1)}+\pi_j(1)}{\big(1-\pi_j(0)\big)e^{-\mu_j(0)}+\pi_j(0)}\right)-\beta_{0j},\end{aligned} \tag{11}$$

and

$$\begin{aligned}\beta_0 &= \sum_{j=1}^{p}\beta_{0j}\\ &= \sum_{j=1}^{p}-\Big(\mu_j(1)-\mu_j(0)\Big)+\log\left(\frac{1-\pi_j(1)}{1-\pi_j(0)}\right)\end{aligned} \tag{12}$$

in Equation (5).

Corollary 6.7

Suppose conditions in Theorem 6.3 are satisfied. If additionally, $\pi_j(y) = 0 \ \forall j$ and $p^*(x_j \mid y)$ has a NB PMF. Then, the classifier in Definition 6.2 is equivalent to the classifier in Definition 6.1 in the sense that

$$\beta_{1j} = \log\left(\frac{\mu_j(1)}{\mu_j(0)}\right) - \log\left(\frac{1+\phi_j\mu_j(1)}{1+\phi_j\mu_j(0)}\right), \tag{13}$$

$$\beta_{2j} = 0, \tag{14}$$

and

$$\begin{aligned} \beta_0 &= \sum_{j=1}^{p} \beta_{0j} \\ &= \sum_{j=1}^{p} -\phi_j^{-1} \log\left(\frac{1+\phi_j\mu_j(1)}{1+\phi_j\mu_j(0)}\right) \end{aligned} \tag{15}$$

in Equation (5).

Corollary 6.8

Suppose conditions in Theorem 6.3 are satisfied. If additionally, $\pi_j(y) > 0\ \forall j$ and $p^*(x_j \mid y)$ has a NB PMF. Then, the classifier in Definition 6.2 is equivalent to the classifier in Definition 6.1 in the sense that

$$\beta_{1j} = \log\left(\frac{\mu_j(1)}{\mu_j(0)}\right) - \log\left(\frac{1+\phi_j\mu_j(1)}{1+\phi_j\mu_j(0)}\right), \tag{16}$$

$$\begin{aligned}\beta_{2j} &= \log\left[\frac{\left(1-\pi_j(1)\right)\left(\frac{1}{1+\phi_j\mu_j(1)}\right)^{\phi_j^{-1}} + \pi_j(1)}{\left(1-\pi_j(0)\right)\left(\frac{1}{1+\phi_j\mu_j(0)}\right)^{\phi_j^{-1}} + \pi_j(0)}\right] + \phi_j^{-1}\log\left(\frac{1+\phi_j\mu_j(1)}{1+\phi_j\mu_j(0)}\right) - \log\left(\frac{1-\pi_j(1)}{1-\pi_j(0)}\right) \\ &= \log\left[\frac{\left(1-\pi_j(1)\right)\left(\frac{1}{1+\phi_j\mu_j(1)}\right)^{\phi_j^{-1}} + \pi_j(1)}{\left(1-\pi_j(0)\right)\left(\frac{1}{1+\phi_j\mu_j(0)}\right)^{\phi_j^{-1}} + \pi_j(0)}\right] - \beta_{0j},\end{aligned} \tag{17}$$

and

$$\begin{aligned}\beta_0 &= \sum_{j=1}^{p}\beta_{0j} \\ &= \sum_{j=1}^{p} -\phi_j^{-1}\log\left(\frac{1+\phi_j\mu_j(1)}{1+\phi_j\mu_j(0)}\right) + \log\left(\frac{1-\pi_j(1)}{1-\pi_j(0)}\right)\end{aligned} \tag{18}$$

in Equation (5).

Method of Moments Estimators

- We need to get Method of Moments estimates to use as initial values in the soft-thresholding functions.
- Method of Moments (MOM) estimates can be found by setting the K-th theoretical moment of group y equals to the K-th sample moment of group y, where $y = 0, 1$, i.e. MOM estimates can be found by solving

$$E(X_{ij}^K \mid y_i = 1) \stackrel{\text{set}}{=} \frac{1}{n_1} \sum_{i \in \{i: y_i = 1\}} X_{ij}^K y_i \tag{19}$$

and

$$E(X_{ij}^K \mid y_i = 0) \stackrel{\text{set}}{=} \frac{1}{n_0} \sum_{i \in \{i: y_0 = 0\}} X_{ij}^K (1 - y_i). \tag{20}$$

For the parameter ϕ, solves

$$E(X_{ij}^K) \stackrel{\text{set}}{=} \frac{1}{n} \sum_{i=1}^{n} X_{ij}^K \tag{21}$$

to get MOM estimates $\hat{\phi}$.

Method of Moments Estimators

- For POI model, MOM estimates are found by solving Equations (19) and (20). They are

$$\hat{\mu}_j(1) = \overline{x}_j(1) \tag{22}$$

and

$$\hat{\mu}_j(0) = \overline{x}_j(0). \tag{23}$$

Method of Moments Estimators

- For ZIP model, MOM estimates are found by solving Equations (19) and (20). They are

$$\hat{\mu}_j(1) = \frac{\overline{x^2}_j(1)}{\overline{x}_j(1)} - 1, \tag{24}$$

$$\hat{\mu}_j(0) = \frac{\overline{x^2}_j(0)}{\overline{x}_j(0)} - 1, \tag{25}$$

$$\hat{\pi}_j(1) = 1 - \frac{\overline{x}_j(1)}{\hat{\mu}_j(1)}, \tag{26}$$

and

$$\hat{\pi}_j(0) = 1 - \frac{\overline{x}_j(0)}{\hat{\mu}_j(0)}. \tag{27}$$

Method of Moments Estimators

- For NB model, MOM estimates are found by solving Equations (19), (20) and (21). They are

$$\hat{\mu}_j(1) = \overline{x}_j(1), \tag{28}$$

$$\hat{\mu}_j(0) = \overline{x}_j(0), \tag{29}$$

and

$$\hat{\phi}_j = \frac{\overline{x^2}_j - \overline{x}_j}{(\overline{x}_j)^2} - 1. \tag{30}$$

Method of Moments Estimators

- For ZINB model, MOM estimates are found by solving Equations (19), (20) and (21). They are

$$\hat{\mu}_j(1) = \frac{\overline{x^3}_j(1)\overline{x}_j(1) + \overline{x^2}_j(1)\overline{x}_j(1) - 2\left(\overline{x^2}_j(1)\right)^2}{\left(\overline{x}_j(1)\right)^2 - \overline{x^2}_j(1)\overline{x}_j(1)}, \tag{31}$$

$$\hat{\mu}_j(0) = \frac{\overline{x^3}_j(0)\overline{x}_j(0) + \overline{x^2}_j(0)\overline{x}_j(0) - 2\left(\overline{x^2}_j(0)\right)^2}{\left(\overline{x}_j(0)\right)^2 - \overline{x^2}_j(0)\overline{x}_j(0)}, \tag{32}$$

$$\hat{\pi}_j(1) = 1 - \frac{\left(\overline{x}_j(1)\right)^3 - \overline{x^2}_j(1)\left(\overline{x}_j(1)\right)^2}{\overline{x^3}_j(1)\overline{x}_j(1) + \overline{x^2}_j(1)\overline{x}_j(1) - 2\left(\overline{x^2}_j(1)\right)^2}, \tag{33}$$

$$\hat{\pi}_j(0) = 1 - \frac{\left(\overline{x}_j(0)\right)^3 - \overline{x^2}_j(0)\left(\overline{x}_j(0)\right)^2}{\overline{x^3}_j(0)\overline{x}_j(0) + \overline{x^2}_j(0)\overline{x}_j(0) - 2\left(\overline{x^2}_j(0)\right)^2}, \tag{34}$$

and

$$\hat{\phi}_j = \frac{\left(\overline{x^2}_j - \overline{x}_j\right)^2}{2\left(\overline{x^2}_j\right)^2 - \overline{x^3}_j\overline{x}_j - \overline{x^2}_j\overline{x}_j} - 1. \tag{35}$$

Penalized Method of Moments Estimators

- First, we need to define soft-thresholding functions. The soft-thresholding functions are

$$\hat{\beta}_{1j}(\lambda_1) = \text{sign}(\hat{\beta}_{1j})(\left|\hat{\beta}_{1j}\right| - \lambda_1)_+ \tag{36}$$

and

$$\hat{\beta}_{2j}(\lambda_2) = \text{sign}(\hat{\beta}_{2j})(\left|\hat{\beta}_{2j}\right| - \lambda_2)_+ \tag{37}$$

where $t_+ = t$ if $t > 0$ and $t_+ = 0$ otherwise.

  - $\hat{\beta}_{1j}$ and $\hat{\beta}_{2j}$ are any estimates, i.e. initial values.
  - $\lambda_1 \geq 0$ and $\lambda_2 \geq 0$ control the complexity of the model and can be chosen by cross-validation. The amount of shrinkage increases as λ_1 and λ_2 increase.
  - This is ℓ_1-type penalization that will force some $\hat{\beta}_{1j}$s and $\hat{\beta}_{2j}$s to 0 i.e., the soft thresholding functions allow the variable selection.

Penalized Method of Moments Estimators

- The penalized method of moment (PMOM) estimates can be found in the following steps.
  1. Specify the model.
  2. Determine whether (λ_1, λ_2) are chosen or estimated.
  3. If (λ_1, λ_2) are chosen, get PMOM estimates using Algorithm 1. Otherwise, get PMOM estimates using Algorithm 2.

Algorithms

- **Algorithm 1** Penalized Method of Moments Estimator for Fixed (λ_1, λ_2).

  1. Select a model from POI, ZIP, NB, ZINB models and compute the corresponding MOM estimates $\hat{\mu}_j(1)$, $\hat{\mu}_j(0)$, $\hat{\pi}_j(1)$, $\hat{\pi}_j(0)$ and $\hat{\phi}_j$ using Equations (22)- (35).
  2. Compute initial estimates $\hat{\beta}_{1j}$ and $\hat{\beta}_{2j}$ using the corresponding Corollaries 6.5- 6.8.
  3. Get PMOM estimates $\hat{\beta}_{1j}(\lambda_1)$ and $\hat{\beta}_{2j}(\lambda_2)$ using Equations (36) and (37).

- **Algorithm 2** Penalized Method of Moments Estimator Maximizes LOOCV Estimate of AUROC.

1. For $i = 1, 2, \ldots, n$ do:
   - (a) Select a model from POI, ZIP, NB, ZINB models and compute the corresponding MOM estimates $\hat{\mu}_j^{(-i)}(1)$, $\hat{\mu}_j^{(-i)}(0)$, $\hat{\pi}_j^{(-i)}(1)$, $\hat{\pi}_j^{(-i)}(0)$ and $\hat{\phi}_j^{(-i)}$ using Equations (22)- (35).
   - (b) Compute initial estimates $\hat{\beta}_{1j}^{(-i)}$ and $\hat{\beta}_{2j}^{(-i)}$ using the corresponding Corollaries 6.5- 6.8.
   - (c) For each pair (λ_1, λ_2),
     - (i) Compute PMOM estimates $\hat{\beta}_{1j}^{(-i)}(\lambda_1)$ and $\hat{\beta}_{2j}^{(-i)}(\lambda_2)$ using Equations (36) and (37).
     - (ii) Compute PMOM classifier by $\hat{C}^{(i)}((\lambda_1, \lambda_2)) = \sum_{j=1}^{p} \hat{\beta}_{1j}^{(-i)}(\lambda_1) x_{ij} + \hat{\beta}_{2j}^{(-i)}(\lambda_2) \delta_0(x_{ij})$.
     - (iii) Compute estimated rank empirical AUROC by

$$\widehat{\text{AUROC}}(\lambda_1, \lambda_2) = \frac{1}{n_0 n_1} \left[\sum_{i \in \{i : y_i = 1\}} r\left(\hat{C}^{(i)}((\lambda_1, \lambda_2)) \right) - \frac{n_1(n_1 + 1)}{2} \right].$$

2. Return the range of $(\hat{\lambda}_1, \hat{\lambda}_2)$ that maximizes $\widehat{\text{AUROC}}$ in (iii).
3. Get PMOM estimates $\hat{\beta}_{1j}(\hat{\lambda}_1)$ and $\hat{\beta}_{2j}(\hat{\lambda}_2)$ using Equations (36) and (37).

Zero-Inflation and Over-Dispersion are Addressed

- Figure 3 shows our method can model zero-inflation.
- Here, $\hat{\pi}_j(Y)$s $(Y = 0, 1)$ and $\hat{\beta}_{2j}$s are found by steps 1 and 2 in Algorithm 1, respectively.

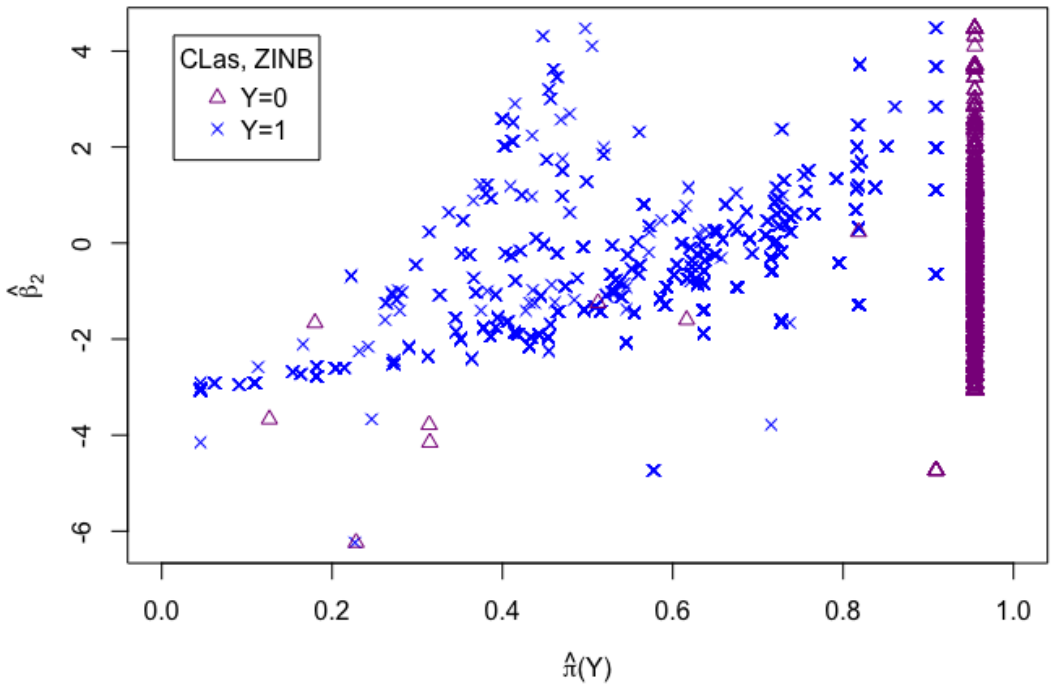

Figure 3: A scatter plot plotted $\hat{\pi}_j(Y)$ versus $\hat{\beta}_{2j}$ using CLas data set in Table 1 under ZINB model, where $Y = 0$ and $Y = 1$ indicated with different characters.

Zero-Inflation and Over-Dispersion are Addressed

- Figure 4 shows that our method can model over-dispersion.
- Here, $\hat{\phi}_j$s and $\hat{\beta}_{1j}$s, $\hat{\beta}_{2j}$s are found by steps 1 and 2 in Algorithm 1, respectively.

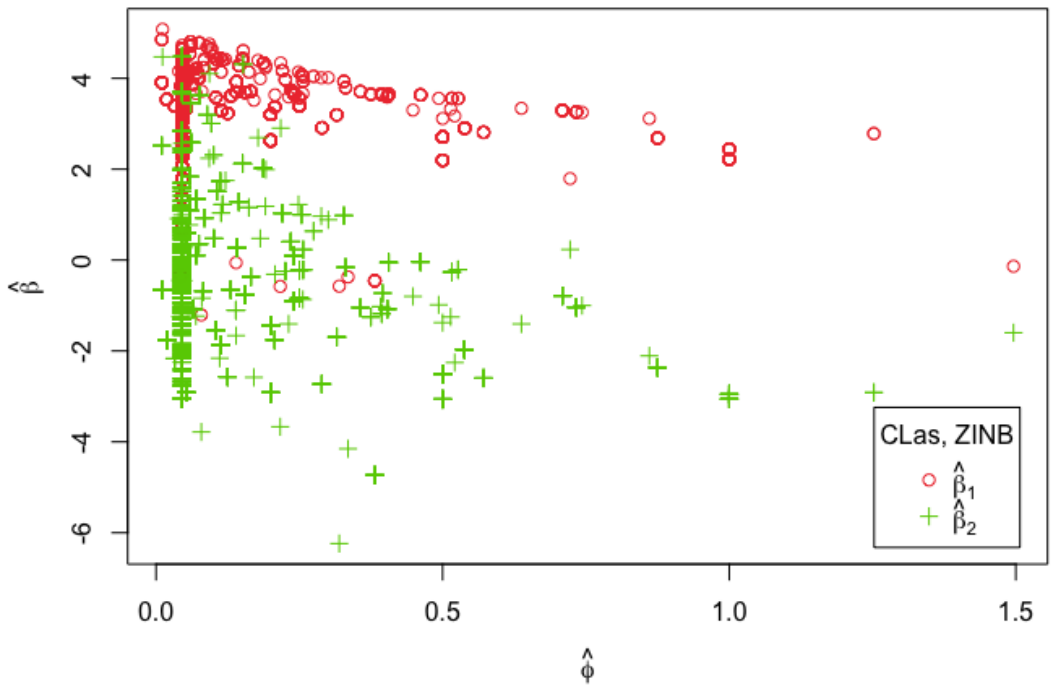

Figure 4: A scatter plot plotted $\hat{\phi}_j$ versus $\hat{\boldsymbol{\beta}}_j$ using CLas data set in Table 1 under ZINB model, where $\hat{\beta}_{1j}$ and $\hat{\beta}_{2j}$ indicated with different characters.

Model/Variable Selection is Addressed

Fixed λ

- Figure 5 shows coefficient paths of the PMOM estimates $\hat{\beta}_{1j}$s and $\hat{\beta}_{2j}$s.
- The PMOM estimates are found by applying the step 3 in Algorithm 1.

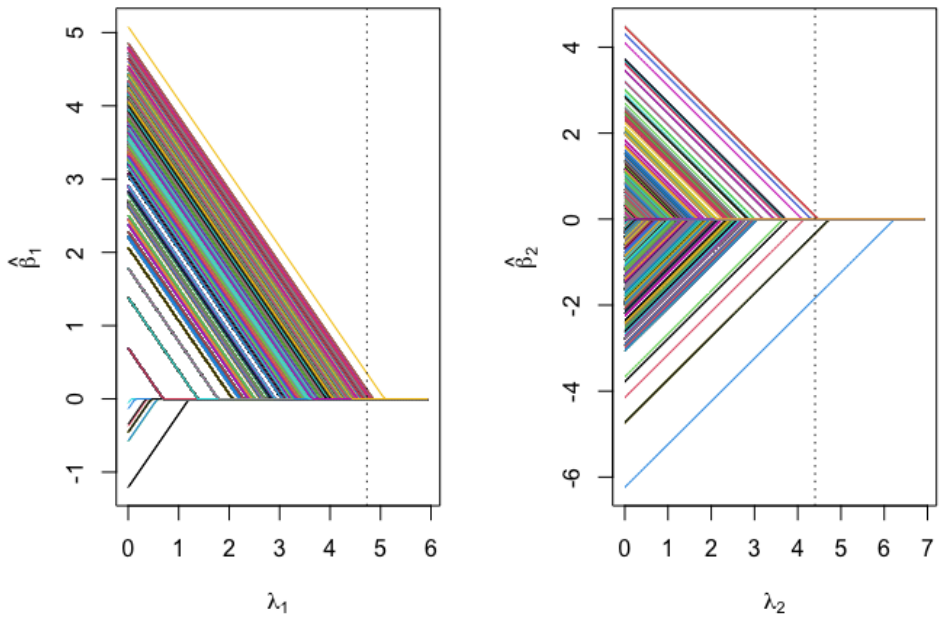

Figure 5: Left: A coefficient path plot plotted λ_1 versus PMOM estimates $\hat{\beta}_{1j}$ using CLas data set in Table 1 under ZINB model. Right: Same for λ_2 versus PMOM estimates $\hat{\beta}_{2j}$. The vertical dotted lines indicate a choice of λ_1 and λ_2.

Model/Variable Selection is Addressed

Fixed λ

- Figure 6 shows histograms of the un-penalized MOM estimates $\hat{\beta}_{1j}$s and $\hat{\beta}_{2j}$s.
- Here, estimates are found by applying step 2 in Algorithm 1.

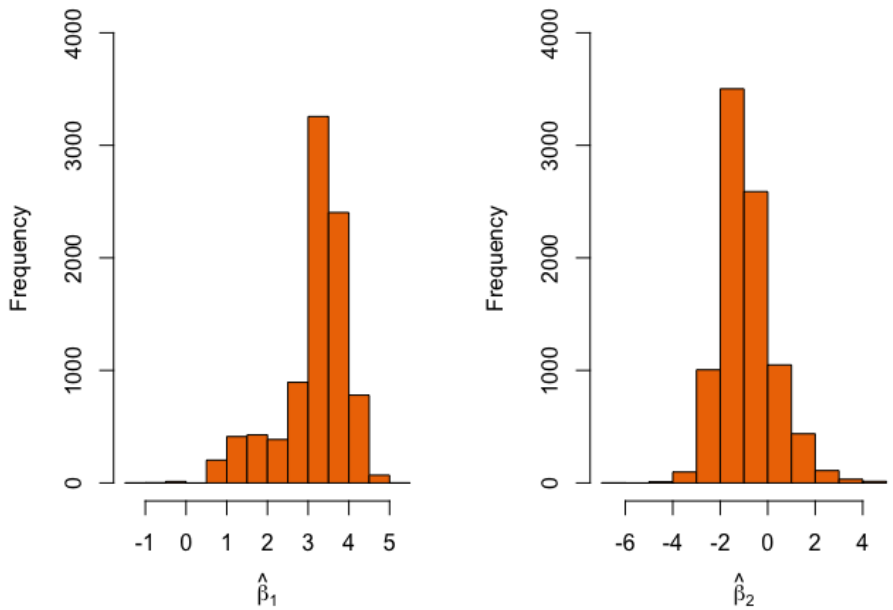

Figure 6: Left: A histogram plotted frequency versus un-penalized MOM estimates $\hat{\beta}_{1j}$ using CLas data set in Table 1 under ZINB model. Right: Same for un-penalized MOM estimates $\hat{\beta}_{2j}$.

Model/Variable Selection is Addressed

Fixed λ

- Figure 7 shows histograms of the PMOM $\hat{\beta}_{1j}(4.73)$s and $\hat{\beta}_{2j}(4.4)$s.
- The PMOM estimates are found by applying step 3 in Algorithm 1.

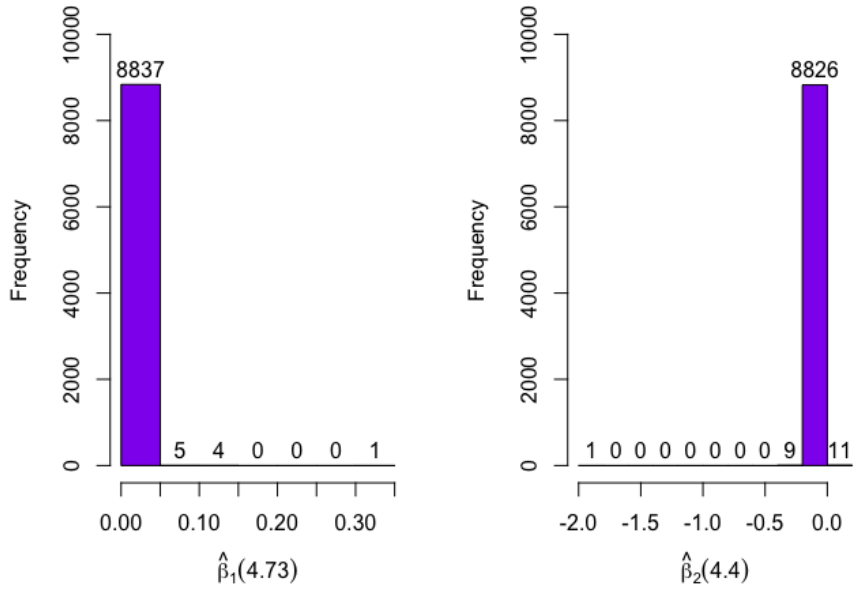

Figure 7: Left: A histogram plotted frequency versus PMOM estimates $\hat{\beta}_{1j}$ with $\lambda_1 = 4.73$ using CLas data set in Table 1 under ZINB model. Right: Same for PMOM estimates $\hat{\beta}_{2j}$ with $\lambda_2 = 4.4$. The numbers on the top of the bars are counts in the range.

Model/Variable Selection is Addressed

- Figure 8 shows a side-by-side boxplot of estimates $\hat{\beta}_{1j}$s and $\hat{\beta}_{2j}$s.
- Here, the MOM estimates are found by applying step 2 in Algorithm 1.
- Here, the PMOM estimates are found by applying step 3 in Algorithm 1.

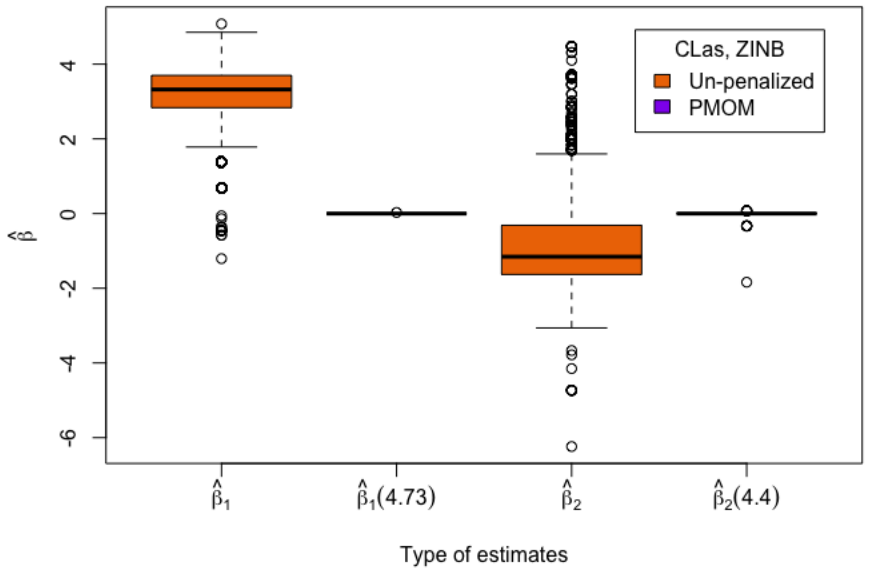

Figure 8: A side-by-side boxplot plotted the type of estimates ($\hat{\beta}_{1j}$ or $\hat{\beta}_{2j}$) versus its estimates values using CLas data set in Table 1 under ZINB model. The un-penalized MOM estimates and the PMOM estimates of $\hat{\beta}_{1j}$s and $\hat{\beta}_{2j}$s are denoted with different colors.

Model/Variable Selection is Addressed

Estimated λ

- We may use LOOCV estimated (λ_1, λ_2) to address the limitation 2: allows for model/variable selection.
- One way to estimate (λ_1, λ_2) is to maximize the LOOCV estimate of AUROC.

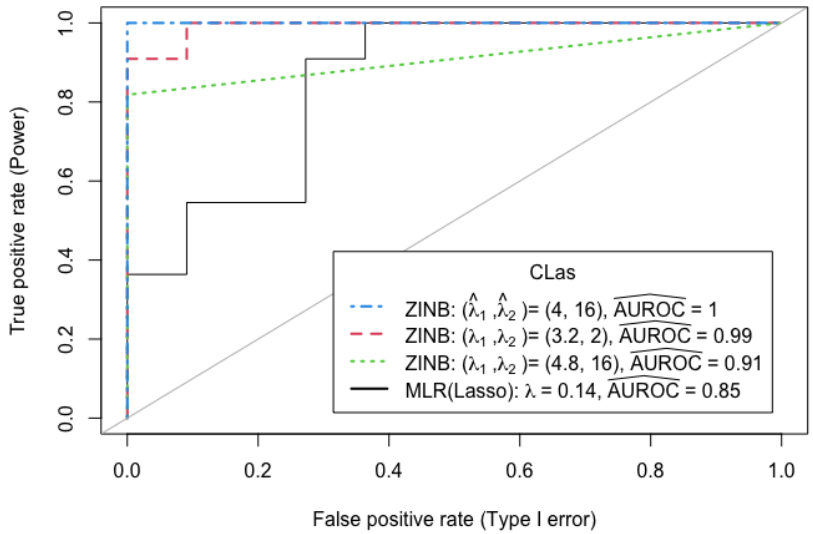

Figure 9: An ROC graph plotted FPR versus TPR using CLas data set in Table 1 under ZINB model. There are four classifiers indicated with different colors and types of lines with their (λ_1, λ_2) and AUROCs. The gray solid line represents a classifier that is as good as random guessing and AUROC $=0.5$.

Computationally Efficient

- Our methods are computationally efficient because the PMOM estimates are closed-form expressions. Thus, an iterative procedure is not required.
  - See steps 1 and 2 in Algorithm 1 for fixed (λ_1, λ_2). Similarly, see steps 1(a) and 1(b) in Algorithm 2 if (λ_1, λ_2) are estimated.
- The MLR approach fails when the two groups have perfect separation because the logistic regression fit is undefined, i.e. there will be $\mp\infty$ ($\log(\frac{0}{1})$, $\log(\frac{1}{0})$) if $p(1 \mid x) = 0$ and $p(0 \mid x) = 1$, respectively.

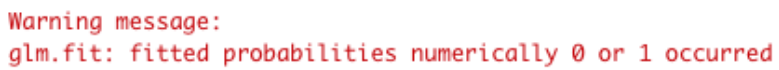

Figure 10: A picture showed warning messages of the MLR approach.

- The divergence could occur because the Newton-Raphson algorithm is needed and is not guaranteed to converge without step-size optimization in the quadratic approximation step (Hastie et al., 2015, Chapter 5.4.3).

Future Work

- In applications,
  - we will provide scatter plots of MOM estimates $\hat{\beta}_{1j}$s and $\hat{\beta}_{2j}$s vs PMOM estimates $\hat{\beta}_{1j}(\lambda_1)$s and $\hat{\beta}_{2j}(\lambda_2)$s. Thus, we can see many $\hat{\beta}_{1j}(\lambda_1)$s and $\hat{\beta}_{2j}(\lambda_2)$s are exactly zeros. Thus, our method addresses the limitation 2, that is, our method allows for model/variable selection.
  - we will consider the applications of ZIP, NB, and POI models.
  - we will consider smaller p specified data.
- In simulations, compare test $\widehat{\text{AUROC}}$s
  - p varies
    - $\pi_j(y)$ varies
    - ϕ_j varies

Acknowledgment

- I would like to express my deepest appreciation to my outside committee Dr. Kitty Cardwell for your support.
- I would like to express my deepest appreciation to my committee members, Dr. Lan Zhu and Dr. Ye Liang, for their valuable guidance and insightful feedback.
- I would like to express my most profound gratitude to my advisor Dr. Joshua D Habiger.
- I would like to acknowledge my parents.

B. A. Babcock. Economic impact of california's citrus industry in 2020. *Journal of Citrus Pathology*, 9(1), 2022.

K. Cardwell, G. Dennis, A. R. Flannery, J. Fletcher, D. Luster, M. Nakhla, A. Rice, P. Shiel, J. Stack, C. Walsh, et al. Principles of diagnostic assay validation for plant pathogens: A basic review of concepts. *Plant Health Progress*, 19(4):272–278, 2018.

T. Dang, H. Wang, A. S. Espindola, J. Habiger, G. Vidalakis, and K. Cardwell. Development and statistical validation of e-probe diagnostic nucleic acid analysis (edna) assays for the detection of citrus pathogens from raw high-throughput sequencing data. *PhytoFrontiers™*, 3(1):113–123, 2023.

K. Dong, H. Zhao, T. Tong, and X. Wan. Nblda: negative binomial linear discriminant analysis for rna-seq data. *BMC bioinformatics*, 17:1–10, 2016.

A. S. Espindola and K. F. Cardwell. Microbe finder (mifi®): Implementation of an interactive pathogen detection tool in metagenomic sequence data. *Plants*, 10(2):250, 2021.

A. S. Espindola, K. Cardwell, F. N. Martin, P. R. Hoyt, S. M. Marek, W. Schneider, and C. D. Garzon. A step towards validation of high-throughput sequencing for the identification of plant pathogenic oomycetes. *Phytopathology®*, 112(9):1859–1866, 2022.

T. Hastie, R. Tibshirani, and M. Wainwright. Statistical learning with sparsity. *Monographs on statistics and applied probability*, 143(143):8, 2015.

D. Risso, F. Perraudeau, S. Gribkova, S. Dudoit, and J.-P. Vert. A general and flexible method for signal extraction from single-cell rna-seq data. *Nature communications*, 9(1):284, 2018.

B. Sohrab, T. Dang, and H. Wang. E-probes targeting citrus pathogens as a new diagnostic standard, 2024. URL `https://citrus-research-board-static.sfo2.digitaloceanspaces.com/citrograph/pdf/CRB-Citrograph-Mag-Q2-Spring-2024-Web.pdf`.

A. H. Stobbe, J. Daniels, A. S. Espindola, R. Verma, U. Melcher, F. Ochoa-Corona, C. Garzon, J. Fletcher, and W. Schneider. E-probe diagnostic nucleic acid analysis (edna): a theoretical approach for handling of next generation sequencing data for diagnostics. *Journal of microbiological methods*, 94(3):356–366, 2013.

D. M. Witten. Classification and clustering of sequencing data using a poisson model. 2011.

D. Yu, W. Huber, and O. Vitek. Shrinkage estimation of dispersion in negative binomial models for rna-seq experiments with small sample size. *Bioinformatics*, 29(10):1275–1282, 2013.

Y. Zhou, X. Wan, B. Zhang, and T. Tong. Classifying next-generation sequencing data using a zero-inflated poisson model. *Bioinformatics*, 34(8):1329–1335, 2018.

J. Zhu, Z. Yuan, L. Shu, W. Liao, M. Zhao, and Y. Zhou. Selecting classification methods for small samples of next-generation sequencing data. *Frontiers in Genetics*, 12:642227, 2021.