mathematics

MDPI

Article

Matrix Factorization and Prediction for High-Dimensional Co-Occurrence Count Data via Shared Parameter Alternating Zero Inflated Gamma Model

Taejoon Kim [1] and Haiyan Wang [2,*]

[1] Department of Statistics and Biostatistics, California State University East Bay, Hayward, CA 94542, USA; taejoon.kim@csueastbay.edu
[2] Department of Statistics, Kansas State University, Manhattan, KS 66506, USA
* Correspondence: hwang@ksu.edu

Abstract: High-dimensional sparse matrix data frequently arise in various applications. A notable example is the weighted word–word co-occurrence count data, which summarizes the weighted frequency of word pairs appearing within the same context window. This type of data typically contains highly skewed non-negative values with an abundance of zeros. Another example is the co-occurrence of item–item or user–item pairs in e-commerce, which also generates high-dimensional data. The objective is to utilize these data to predict the relevance between items or users. In this paper, we assume that items or users can be represented by unknown dense vectors. The model treats the co-occurrence counts as arising from zero-inflated Gamma random variables and employs cosine similarity between the unknown vectors to summarize item–item relevance. The unknown values are estimated using the shared parameter alternating zero-inflated Gamma regression models (SA-ZIG). Both canonical link and log link models are considered. Two parameter updating schemes are proposed, along with an algorithm to estimate the unknown parameters. Convergence analysis is presented analytically. Numerical studies demonstrate that the SA-ZIG using Fisher scoring without learning rate adjustment may fail to find the maximum likelihood estimate. However, the SA-ZIG with learning rate adjustment performs satisfactorily in our simulation studies.

Keywords: high-dimensional co-occurrence matrix; matrix factorization; zero-inflated Gamma regression; Adam; recommender system

MSC: 62-08; 62J99

check for updates

Citation: Kim, T.; Wang, H. Matrix Factorization and Prediction for High-Dimensional Co-Occurrence Count Data via Shared Parameter Alternating Zero Inflated Gamma Models. *Mathematics* **2024**, *12*, 3365. https://doi.org/10.3390/math12213365

Academic Editor: Andreas Artemiou

Received: 28 September 2024
Revised: 24 October 2024
Accepted: 25 October 2024
Published: 27 October 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Matrix factorization is a fundamental technique in linear algebra and data science, widely used for dimensionality reduction, data compression, and feature extraction. Recent research expands its use in various fields, including recommendation systems (e.g., collaborative filtering), bioinformatics, and signal processing. Researchers are actively pursuing new types of factorizations. In their effort to discover new factorizations and provide a unifying structure, ref. [1] list 53 systematically derived matrix factorizations arising from the generalized Cartan decomposition. Their results apply to invertible matrices and generalizations of orthogonal matrices in classical Lie groups.

Nonnegative matrix factorization (NMF) is particularly useful when dealing with nonnegative data, such as in image processing and text mining. Ref. [2] survey existing NMF methods and their variants, analyzing their properties and applications. Ref. [3] present a comprehensive survey of NMF, focusing on its applications in feature extraction and feature selection. Ref. [4] summarize theoretical research on NMF from 2008 to 2013, categorizing it into four types and analyzing the principles, basic models, properties, algorithms, and their extensions and generalizations.

There are many advances aimed at developing more efficient algorithms tailored to specific applications. One prominent application of matrix factorization is in recommendation systems, particularly for addressing the cold-start problem. In recommender systems, matrix factorization models decompose user–item, user–user, or item–item interaction matrices into lower-dimensional latent spaces, which can then be used to generate recommendations.

Ref. [5] summarizes the literature on recommender systems and proposes a multifaceted collaborative filtering model that integrates both neighborhood and latent factor approaches. Ref. [6] develop a matrix factorization model to generate recommendations for users in a social network. Ref. [7] proposes a matrix factorization model for cross-domain recommender systems by extracting items from three different domains and finding item similarities between these domains to improve item ranking accuracy.

Ref. [8] develop a matrix factorization model that combines user–item rating matrices and item–side information matrices to develop soft clusters of items for generating recommendations. Ref. [9] propose a matrix factorization that learns latent representations of both users and items using gradient-boosted trees. Ref. [10] provide a systematic literature review on approaches and algorithms to mitigate cold-start problems in recommender systems.

Matrix factorization has also been used in natural language processing (NLP) in recent years. Word2Vec by [11,12] marks a milestone in NLP history. Although no clear matrices are presented in their study, Word2Vec models the co-occurrence of words and phrases using latent vector representations via a shallow neural network model. Another well-known example of matrix factorization in NLP is the word representation with global vectors (GloVe) by [13]. They model the co-occurrence count matrix of words using cosine similarity of latent vector representations via alternating least squares. However, practical data may be heavily skewed, making the sum or mean squared error unsuitable as an objective function. To address this, ref. [13] utilize weighted least squares to reduce the impact of skewness in the data. Manually creating the weight is difficult to make it work well for real data. In GloVe training, the algorithm is set to run a fixed number of iterations without another convergence check. Here, we consider using the likelihood principle to model the skewed data matrix.

In this paper, our goal is to model non-negative continuous sparse matrix data from skewed distributions with an abundance of zeros but lacking covariate information. We are particularly interested in zero-inflated Gamma observations, often referred to as semi-continuous data due to the presence of many zeros and the highly skewed distribution of positive observations. Examples of such data include insurance claim data, household expenditure data, and precipitation data (see [14]). Ref. [15] study and simulate data using actual deconvolved calcium imaging data, employing a zero-inflated Gamma model to accommodate spikes of observed inactivity and traces of calcium signals in neural populations. Ref. [16] examine the amount of leisure time spent on physical activity and explanatory variables such as gender, age, education level, and annual per capita family income. They find that the zero-inflated Gamma model is preferred over the multinomial model. Beyond Gamma distribution, Weibull distribution can also model skewed non-negative data. Unfortunately, when the shape parameter is unknown, the Weibull distribution is not a member of the exponential family. Further, the alternating update is not suitable because its sufficient statistics is not linear in the observed data. A different estimation procedure will need to be developed if Weibull distribution is used. The Gamma distribution not only is a member of the exponential family, but its sufficient statistic is also linear in the observed data. This gives solid theoretical ground for alternating update to find a solution. Section 5 explains more details in this regard.

In all the aforementioned examples, there are explicitly observed covariates or factors. Furthermore, the two parts of the model parameters are pendicular to each other, allowing model estimation by fitting two separate regressions: binomial regression and Gamma regression. Unfortunately, such models cannot be applied to user–item or co-occurrence

count matrix data arising from many practical applications. Examples include user–item or item–item co-occurrence data from online shopping platforms and co-occurring word–word pairs in sequences of texts. One reason is the absence of observed covariates. Additionally, the mechanism that leads to the observed data entries in the data matrix for the binomial and Gamma parts may be of the same nature, making it more appropriate to utilize shared parameters in both parts of the model. Therefore, in this paper, we consider shared parameter modeling of zero-inflated Gamma data using alternating regression. We consider two different link functions for the Gamma part: canonical link and log link.

We believe that our study is the first that utilizes a shared parameter likelihood for zero-inflated skewed distribution to conduct matrix factorization. The alternating least squares (ALS) shares a similar spirit as our SA-ZIG in terms of alternately updating parameters. Most matrix factorization methods in the literature involving ALS are adding an assumption without realizing it. In order for ALS to be valid, the data need to have constant variance. This is because the objective function behind the ALS procedure is the mean squared error, which gives equal emphasis to all observations regardless of how big or small their variations are. If the variations for different observations are drastically different, it is unfair to treat the residuals equally. The real data in the high-dimensional sparse co-occurrence matrix are often very skewed with a lot of zeros and do not have constant variance. The contribution of this paper is the SA-ZIG model that models the positive co-occurrence data with Gamma distribution and attributes the many zeros in the data as being from a Bernoulli distribution. Shared parameters are used in both the Bernoulli and Gamma parts of the model. The latent row and column vector representations in matrix decomposition can be thought of as missing values that have some distributions relying on a smaller set of parameters. Estimating the vector representation for the rows relies on the joint likelihood of the observed matrix data and the missing vector representation for the columns. Due to missingness, the estimation of row vector representations is transformed to using conditional likelihood of the observed data given the column vector representations, and vice versa. This alternating update in the end gives the maximum likelihood estimate if the sufficient statistic for the column vector representation is linear in the observed data and the row vector representations. Both ALS and our SA-ZIG rely on this assumption to be valid.

The remainder of this paper is structured as follows. Section 2 outlines the fundamental framework of the ZIG model. Section 3 focuses on parameter estimation within the SA-ZIG model using the canonical link. Section 4 addresses parameter estimation for the scenario involving the log link in the Gamma regression component. Convergence analysis is presented in Section 5. Section 6 details the SA-ZIG algorithms incorporating learning rate adjustments. Section 7 presents the experimental studies. Finally, Section 8 concludes the paper by summarizing the research findings, contributions, and limitations.

2. Shared Parameter Alternating Zero-Inflated Gamma Regression

Suppose the observed data are $\{y_{ij}, i = 1, \ldots, n, j = 1, \ldots, n\}$ whose distribution depends on some unknown Bernoulli random variable g_{ij} and a Gamma random variable such that

$$g_{ij} = \begin{cases} 0, & \text{with probability } 1 - p_{ij} \\ 1, & \text{with probability } p_{ij} \end{cases}, \qquad y_{ij} = \begin{cases} 0, & \text{if } g_{ij} = 0 \\ \text{Gamma observation}, & \text{if } g_{ij} = 1 \end{cases},$$

The mechanism behind the observed data may be a result of combined contribution from some covariates or factors but none was observed.

We can write the probability mass function (pmf) for the Bernoulli random variable and probability density function (pdf) for the Gamma random variable as follows:

$$\begin{aligned} p(g_{ij}) &= p_{ij}^{g_{ij}} \cdot (1-p_{ij})^{1-g_{ij}}, \\ f(y_{ij} \mid g_{ij}=1) &= \frac{1}{\Gamma(v_i)} \cdot \left(\frac{v_i y_{ij}}{\mu_{ij}}\right)^{v_i} \cdot \frac{1}{y_{ij}} \cdot \exp\left(-\frac{v_i y_{ij}}{\mu_{ij}}\right). \end{aligned}$$

The product of the two functions generates the joint distribution of the Bernoulli and Gamma random variable

$$f(y_{ij}, g_{ij}) = p(g_{ij}) \cdot f(y_{ij} \mid g_{ij}).$$

By summing up the two possibilities of the Bernoulli random variable, we obtain the pdf of the zero-inflated Gamma (ZIG) distribution:

$$f(y_{ij}) = \sum_{x=0}^{1} p(g_{ij}=x) \cdot f(y_{ij} \mid g_{ij}=x) = (1-p_{ij})^{I(y_{ij}=0)} \cdot \left[p_{ij}\, f(y_{ij} \mid g_{ij}=1)\right]^{I(y_{ij}>0)}. \quad (1)$$

For the Bernoulli part, we use the logit link function to connect the probability p_{ij} to the effects of some unknown covariates or factors as follows:

$$\eta_{ij} = \log \frac{p_{ij}}{1-p_{ij}} = \mathbf{w}_i^\top \widetilde{\boldsymbol{w}}_j + b_i + \widetilde{b}_j, \quad (2)$$

where $\mathbf{w}_i$ and b_i are unknown parameters related to row i, while $\widetilde{\boldsymbol{w}}_j$ and $\widetilde{b}_j$ are unknown parameters related to column j. Assume $\mathbf{w}_i$ and $\widetilde{\boldsymbol{w}}_j$ are d-dimensional vectors and b_i, $\widetilde{b}_j$ are scalars. For the Gamma observations,

$$\log f(y_{ij} \mid y_{ij}>0) = -\log\Gamma(v_{ij}) + v_{ij}\log\left(\frac{v_{ij}}{\mu_{ij}}\right) + (v_{ij}-1)\log y_{ij} - \frac{v_{ij}y_{ij}}{\mu_{ij}}.$$

When using the canonical link, the mean of the response variable is connected to the unknown parameters through the following formula:

$$g(\mu_{ij}) = -\mu_{ij}^{-1} = \mathbf{w}_i^\top \widetilde{\boldsymbol{w}}_j + e_i + \widetilde{e}_j, \quad (3)$$

where e_i and $\widetilde{e}_j$ are scalars. With the canonical link, the likelihood function and the score equations are both a function of the sufficient statistic. As the sufficient statistic carries all information about the unknown parameters, we can restrict our attention to the sufficient statistic without losing any information. This link function, however, could have difficulty in the estimation process. The natural parameter space is $\{\mu_{ij} : \mu_{ij} > 0\}$. The right-hand side of Equation (3) may yield an estimate of μ_{ij} that is outside of this parameter space.

Another link function that is popularly used for Gamma distribution is the log link $g(\mu) = \log(\mu)$, which gives the log linear model:

$$g(\mu_{ij}) = \log(\mu_{ij}) = \mathbf{w}_i^\top \widetilde{\boldsymbol{w}}_j + e_i + \widetilde{e}_j. \quad (4)$$

The log link eliminates the non-negativity problem and the model parameters have better interpretation than the canonical link. For each unit increment in $\widetilde{w}_{jk}$ or w_{ik}, the mean increases multiplicatively by $\exp(w_{ik})$ or by $\exp(\widetilde{w}_{jk})$, respectively. Even though the parameters enjoy better interpretation, the estimation could also become a problem in the sense that the dot product on the right-hand side of (4) could become large, leading to μ_{ij} being infinity, and hence, the estimation algorithm diverges.

In both links, the common dot product $\mathbf{w}_i^\top \widetilde{\boldsymbol{w}}_j$ is used to reflect the fact that the cosine similarity is the key driving force behind the observed data in the table. For example, in

natural language processing (NLP), $\mathbf{w}_i$ and $\widetilde{w}_j$ each represent a word in a dense vector. Each one of them contains both linguistic information and word usage information in it. The observations are distance-weighted co-occurrence counts that are linked to relevance between the words and the relevance can be captured with the cosine similarity. In the example of item–item co-occurrence matrix, $\mathbf{w}_i$ and $\widetilde{w}_j$ represent the hidden product information of the items including characteristics, properties, functionality, popularity, users' ratings on them, etc. The dot product again tells how the two items are relevant to each other. Sometimes, the co-occurrence matrix was derived from a time series such as a sequence of watched movies in time order from a customer. In this case, the co-occurrence count tells how often the two items were considered in a similar time frame because the counts were weighted based on the position separation of the two items in the sequence. The relevance of these items summarized by the cosine similarity could reflect how the two products are alike in their property, functionality, etc. Therefore, the dot product serves as a major contributor for the observed weighted count.

In both link functions, the intercepts are allowed to be different from that in the logistic model for flexibility. In some classical statistical models such as in [17], shared parameter modeling is used by assuming one part of model parameters to be proportional to the other. Such proportionally assumed parameters are meaningful for the case where covariates are observed. In our case, both $\mathbf{w}$ and $\widetilde{w}$ are unknown. Different pairs of $\mathbf{w}$ and $\widetilde{w}$ could give the same dot product. Adding an additional proportionality parameter only makes the model even more non-identifiable. This is why we believe it is better to put flexibility in the intercepts instead of using the proportionality parameters.

Due to shared parameters being used, the parameters from both the Gamma regression part and the Logistic regression part should be estimated simultaneously. There are past studies that use a separate set of parameters. Ref. [14] conducted hypothesis testing to compare two groups via mixture of zero-inflated Gamma and zero-inflated log-normal models. However, the parameters from the two model components were modelled separately. In a totally applied setting, ref. [17] considered normal and log-Gamma mixture model for HIV RNA data using shared parameters, assuming that the two parts of the model are proportional to each other. They proposed such an application and simulation, but no inference was given. The shared parameter modeling was employed in some of the literature to achieve some model parsimony. For example, ref. [18] used shared parameters for both a random effects linear model and a probit-modeled censoring process. Ref. [19] used a similar approach for simultaneous modeling of a mean structure and informative censoring. Ref. [20] used shared parameters to model the intensity of a Poisson process and a binary measure of severity.

Denote $\boldsymbol{\theta}_i = (\mathbf{w}_i^\top, b_i, e_i)^\top$, $\widetilde{\boldsymbol{\theta}}_i = (\widetilde{w}_i^\top, \widetilde{b}_i, \widetilde{e}_i)^\top$, and $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n)^\top$, $\widetilde{\boldsymbol{\theta}} = (\widetilde{\boldsymbol{\theta}}_1, \ldots, \widetilde{\boldsymbol{\theta}}_n)^\top$. We estimate $\boldsymbol{\theta}$ and $\widetilde{\boldsymbol{\theta}}$ alternately. In this estimation scheme, the likelihood function can be treated as a function of either $\boldsymbol{\theta}$ or $\widetilde{\boldsymbol{\theta}}$ but not concurrently at a same time. When estimating $\boldsymbol{\theta}$, the likelihood is treated as a function of $\boldsymbol{\theta}$ while $\widetilde{\boldsymbol{\theta}}$ stays fixed. Reversely, when we estimate the $\widetilde{\boldsymbol{\theta}}$, the likelihood is treated as a function of $\widetilde{\boldsymbol{\theta}}$ while $\boldsymbol{\theta}$ is treated as fixed. This resembles block coordinate descent in which one part of the parameters is updated while holding the remaining part of the parameters as fixed values. For clarity, we use separate notations ℓ_i and $\widetilde{\ell}_j$ to refer to these two occasions, i.e.,

$$\ell_i = \ell_i(\boldsymbol{\theta}_i; \widetilde{\boldsymbol{\theta}}) = \sum_{j=1}^{n} \ell_{ij}(\boldsymbol{\theta}_i; \widetilde{\boldsymbol{\theta}}_j), \qquad \widetilde{\ell}_j = \widetilde{\ell}_j(\widetilde{\boldsymbol{\theta}}_j; \boldsymbol{\theta}) = \sum_{i=1}^{n} \ell_{ij}(\widetilde{\boldsymbol{\theta}}_j; \boldsymbol{\theta}_i),$$

where $\ell_{ij}(\boldsymbol{\theta}_i; \widetilde{\boldsymbol{\theta}}_j)$ and $\ell_{ij}(\widetilde{\boldsymbol{\theta}}_j; \boldsymbol{\theta}_i)$ are both equal to $\log(f(y_{ij}))$ but one is treated as a function of $\boldsymbol{\theta}_i$ while the other one is treated as a function of $\widetilde{\boldsymbol{\theta}}_j$.

Note that ℓ_i and $\widetilde{\ell}_j$ simply represent the log likelihood of a row or a column of the data matrix. The ℓ_i can be thought of as the log likelihood function of $\boldsymbol{\theta}_i$ for data in the i^{th} row of the co-occurence matrix and the $\widetilde{\ell}_j$ is the log likelihood function of $\widetilde{\boldsymbol{\theta}}_j$ for data in the j^{th}

column of the matrix. Each log likelihood function could be split into two components; one corresponds to the Bernoulli part and the other corresponds to the Gamma part:

$$\ell_i = \ell_i(\boldsymbol{\theta}_i; \widetilde{\boldsymbol{\theta}}) = \ell_i^{(1)} + \ell_i^{(2)}, \quad \widetilde{\ell}_j = \widetilde{\ell}_j(\widetilde{\boldsymbol{\theta}}_j; \boldsymbol{\theta}) = \widetilde{\ell}_j^{(1)} + \widetilde{\ell}_j^{(2)},$$

where

$$\begin{aligned}
\ell_i^{(1)} &= \ell_i^{(1)}(\boldsymbol{\theta}_i; \widetilde{\boldsymbol{\theta}}) = \sum_{j=1}^{n} \{ I(y_{ij} = 0) \cdot \log(1 - p_{ij}) + I(y_{ij} > 0) \cdot \log p_{ij} \}, \\
\ell_i^{(2)} &= \ell_i^{(2)}(\boldsymbol{\theta}_i; \widetilde{\boldsymbol{\theta}}) = \sum_{j=1}^{n} I(y_{ij} > 0) \cdot \log f(y_{ij} \mid y_{ij} > 0), \\
\widetilde{\ell}_j^{(1)} &= \widetilde{\ell}_j^{(1)}(\widetilde{\boldsymbol{\theta}}_j; \boldsymbol{\theta}) = \sum_{i=1}^{n} \{ I(y_{ij} = 0) \cdot \log(1 - p_{ij}) + I(y_{ij} > 0) \cdot \log p_{ij} \}, \\
\widetilde{\ell}_j^{(2)} &= \widetilde{\ell}_j^{(2)}(\widetilde{\boldsymbol{\theta}}_j; \boldsymbol{\theta}) = \sum_{i=1}^{n} I(y_{ij} > 0) \cdot \log f(y_{ij} \mid y_{ij} > 0).
\end{aligned}$$

Assuming that the observations y_{ij}'s are independent of each other, conditional on unobserved $\boldsymbol{\theta}_i$ and $\widetilde{\boldsymbol{\theta}}_j$, $i = 1, \ldots, n, j = 1, \ldots, n$, the overall loglikelihood function from all observations can be written as

$$\ell(\boldsymbol{\theta}, \widetilde{\boldsymbol{\theta}}) = \sum_{i=1}^{n} \ell_i(\boldsymbol{\theta}_i; \widetilde{\boldsymbol{\theta}}) = \sum_i (\ell_i^{(1)} + \ell_i^{(2)}); \qquad \widetilde{\ell}(\widetilde{\boldsymbol{\theta}}; \boldsymbol{\theta}) = \sum_{j=1}^{n} \widetilde{\ell}_j(\widetilde{\boldsymbol{\theta}}_j; \boldsymbol{\theta}) = \sum_j (\widetilde{\ell}_j^{(1)} + \widetilde{\ell}_j^{(2)}).$$

The alternating regression deals with two separate log likelihood functions $\ell(\boldsymbol{\theta}; \widetilde{\boldsymbol{\theta}})$ and $\widetilde{\ell}(\widetilde{\boldsymbol{\theta}}; \boldsymbol{\theta})$, respectively. The $\ell_i^{(1)}$ and $\widetilde{\ell}_j^{(1)}$ part is the traditional log likelihood for binary logistic regression. The $\ell_i^{(2)}$ and $\widetilde{\ell}_j^{(2)}$ part is the Gamma log likelihood restricted to only positive observations. If the two parts do not share common parameters, then the estimation can be performed separately. However, they share some common parameters $\mathbf{w}_i$ and $\widetilde{\boldsymbol{w}}_j$. In the next two sections, we describe the parameter estimations.

3. ZIG Model with Canonical Links

In this section, we consider the case with canonical links, i.e., the Bernoulli part uses logit function and the Gamma part uses negative inverse link. Using the canonical link with generalized linear models enjoys the benefit that the score equations are a function of sufficient statistics. Here, we consider parameter estimation under the canonical links. The negative inverse link has difficulty interpreting model parameters and also has some restrictions in terms of the support of the link function, which does not match the positive value of Gamma distribution. More details can be seen as we introduce the model.

Recall that the the log likelihood and the link function for the logistic part on the i^{th} row of data are

$$\begin{aligned}
\ell_i^{(1)} &= \sum_{j=1}^{n} \{ I(y_{ij} = 0) \cdot \log(1 - p_{ij}) + I(y_{ij} > 0) \cdot \log p_{ij} \}, \\
\eta_{ij} &= \log \frac{p_{ij}}{1 - p_{ij}} = \mathbf{w}_i^\top \widetilde{\boldsymbol{w}}_j + b_i + \widetilde{b}_j,
\end{aligned}$$

where $\mathbf{w}_i$ and b_i are unknown parameters while treating $\widetilde{\boldsymbol{w}}_j$ and $\widetilde{b}_j$ as fixed. The log likelihood and the negative inverse link function for the Gamma part on the i^{th} row of data are

$$\ell_i^{(2)} = \sum_{j=1}^{n} I\Big(y_{ij} > 0\Big) \cdot \log f\Big(y_{ij} \mid y_{ij} > 0\Big), \tag{5}$$

$$\text{with } \log f\Big(y_{ij} \mid y_{ij} > 0\Big) = -\log\Gamma\Big(v_{ij}\Big) + v_{ij}\log\left(\frac{v_{ij}}{\mu_{ij}}\right) + \Big(v_{ij} - 1\Big)\log y_{ij} - \frac{v_{ij}y_{ij}}{\mu_{ij}},$$

$$\tau_{ij} = g(\mu_{ij}) = -\mu_{ij}^{-1} = \mathbf{w}_i^{\top}\widetilde{\boldsymbol{w}}_j + e_i + \widetilde{e}_j. \tag{6}$$

The log likelihood function for the i^{th} row of data is

$$\ell_i \quad = \quad \ell_i^{(1)} + \ell_i^{(2)}.$$

To obtain partial derivatives, we write $\ell_i^{(1)}$ as follows:

$$\ell_i^{(1)} \quad = \quad \sum_{j=1}^{n}(1 - g_{ij}) \cdot \log(1 - p_{ij}) + g_{ij} \cdot \log p_{ij} = \sum_{j=1}^{n} g_{ij} \cdot \eta_{ij} - \log\{1 + \exp(\eta_{ij})\}. \tag{7}$$

The inverse logit and its partial derivative w.r.t. $\mathbf{w}_i$ are

$$p_{ij} = \frac{\exp\Big(\mathbf{w}_i^{\top}\widetilde{\boldsymbol{w}}_j + b_i + \widetilde{b}_j\Big)}{1 + \exp\Big(\mathbf{w}_i^{\top}\widetilde{\boldsymbol{w}}_j + b_i + \widetilde{b}_j\Big)}, \qquad \frac{\partial p_{ij}}{\partial \mathbf{w}_i} = p_{ij}(1 - p_{ij})\widetilde{\boldsymbol{w}}_j.$$

Therefore, the first-order partial derivatives can be summarized as

$$\begin{gathered}\frac{\partial \ell_i^{(1)}}{\partial \mathbf{w}_i} = \sum_{j=1}^{n}\left\{g_{ij} \cdot \widetilde{\boldsymbol{w}}_j - \frac{\exp(\eta_{ij})}{1 + \exp(\eta_{ij})}\widetilde{\boldsymbol{w}}_j\right\} = \sum_{j=1}^{n}(g_{ij} - p_{ij})\widetilde{\boldsymbol{w}}_j, \\ \frac{\partial \ell_i^{(1)}}{\partial b_i} = \sum_{j=1}^{n}(g_{ij} - p_{ij}).\end{gathered} \tag{8}$$

The negative second-order partial derivatives and their expectations are the same and are given below:

$$-\frac{\partial^2 \ell_i^{(1)}}{\partial \mathbf{w}_i \partial \mathbf{w}_i^{\top}} = \sum_{j=1}^{n} p_{ij}(1 - p_{ij})\widetilde{\boldsymbol{w}}_j\widetilde{\boldsymbol{w}}_j^{\top} = E\left[-\frac{\partial^2 \ell_i^{(1)}}{\partial \mathbf{w}_i \partial \mathbf{w}_i^{\top}}\right] \tag{9}$$

$$\begin{gathered}-\frac{\partial^2 \ell_i^{(1)}}{\partial \mathbf{w}_i \partial b_i} = \sum_{j=1}^{n}\widetilde{\boldsymbol{w}}_j\frac{\partial p_{ij}}{\partial b_i} = \sum_{j=1}^{n}\widetilde{\boldsymbol{w}}_j p_{ij}(1 - p_{ij}) = E\left[-\frac{\partial^2 \ell_i^{(1)}}{\partial b_i^2}\right] \\ -\frac{\partial^2 \ell_i^{(1)}}{\partial b_i^2} = \sum_{j=1}^{n} p_{ij}(1 - p_{ij}) = E\left[-\frac{\partial^2 \ell_i^{(1)}}{\partial b_i^2}\right]\end{gathered} \tag{10}$$

Now, consider the second component of the log likelihood $\ell_i^{(2)}$ from the ith row of data

$$\begin{aligned}\ell_i^{(2)} &= \sum_{j=1}^{n} I(y_{ij}>0)\cdot \log f(y_{ij}\mid y_{ij}>0)\\ &= \sum_{j=1}^{n} g_{ij}\cdot\Big\{v_{ij}\log\Big(-\mathbf{w}_i^\top\widetilde{\boldsymbol{w}}_j-e_i-\widetilde{e}_j\Big)+v_{ij}y_{ij}\Big(\mathbf{w}_i^\top\widetilde{\boldsymbol{w}}_j+e_i+\widetilde{e}_j\Big)\\ &\qquad -\log\Gamma(v_{ij})+v_{ij}\log v_{ij}+(v_{ij}-1)\log y_{ij}\Big\}.\end{aligned}\tag{11}$$

The first-order partial derivatives for the $\ell_i^{(2)}$ are

$$\frac{\partial\ell_i^{(2)}}{\partial\mathbf{w}_i}=\sum_{j=1}^{n} g_{ij}\left\{\frac{v_{ij}\cdot\widetilde{\boldsymbol{w}}_j}{\mathbf{w}_i^\top\widetilde{\boldsymbol{w}}_j+e_i+\widetilde{e}_j}+v_{ij}y_{ij}\widetilde{\boldsymbol{w}}_j\right\},\quad \frac{\partial\ell_i^{(2)}}{\partial e_i}=\sum_{j=1}^{n} g_{ij}\left\{\frac{v_{ij}}{\mathbf{w}_i^\top\widetilde{\boldsymbol{w}}_j+e_i+\widetilde{e}_j}+v_{ij}\cdot y_{ij}\right\}.\tag{12}$$

The second-order partial derivatives and their negative expectations are

$$-\frac{\partial^2\ell_i^{(2)}}{\partial\mathbf{w}_i\partial\mathbf{w}_i^\top}\bigg|_{g_{ij}=1}=\sum_{\substack{j=1\\ g_{ij}=1}}^{n}\frac{v_{ij}\widetilde{\boldsymbol{w}}_j\widetilde{\boldsymbol{w}}_j^\top}{\left(\mathbf{w}_i^\top\widetilde{\boldsymbol{w}}_j+e_i+\widetilde{e}_j\right)^2}=E\left[-\frac{\partial^2\ell_i^{(2)}}{\partial\mathbf{w}_i\partial\mathbf{w}_i^\top}\bigg|g_{ij}=1\right];\tag{13}$$

$$-\frac{\partial^2\ell_i^{(2)}}{\partial\mathbf{w}_i\partial e_i}\bigg|_{g_{ij}=1}=\sum_{\substack{j=1\\ g_{ij}=1}}^{n}\frac{v_{ij}\widetilde{\boldsymbol{w}}_j}{\left(\mathbf{w}_i^\top\widetilde{\boldsymbol{w}}_j+e_i+\widetilde{e}_j\right)^2}=E\left[-\frac{\partial^2\ell_i^{(2)}}{\partial\mathbf{w}_i\partial e_i}\bigg|g_{ij}=1\right]\tag{14}$$

$$-\frac{\partial^2\ell_i^{(2)}}{\partial e_i^2}\bigg|_{g_{ij}=1}=\sum_{\substack{j=1\\ g_{ij}=1}}^{n}v_{ij}\left(\mathbf{w}_i^\top\widetilde{\boldsymbol{w}}_j+e_i+\widetilde{e}_j\right)^{-2}=E\left[-\frac{\partial^2\ell_i^{(2)}}{\partial e_i^2}\bigg|g_{ij}=1\right].\tag{15}$$

All of the aforementioned formulae work with the i^{th} row of data. When we combine the log likelihood from different rows of data, other rows do not contribute to the partial derivative with respect to $\boldsymbol{\theta}_i$. That is,

$$\frac{\partial\ell}{\partial\mathbf{w}_i}=\frac{\partial\ell_i}{\partial\mathbf{w}_i};\quad \frac{\partial\ell}{\partial b_i}=\frac{\partial\ell_i^{(1)}}{\partial b_i};\quad \frac{\partial\ell}{\partial e_i}=\frac{\partial\ell_i^{(2)}}{\partial e_i};\tag{16}$$

$$\frac{\partial^2\ell}{\partial\mathbf{w}_i\partial\mathbf{w}_i^\top}=\frac{\partial^2\ell_i}{\partial\mathbf{w}_i\partial\mathbf{w}_i^\top};\quad \frac{\partial^2\ell}{\partial\mathbf{w}_i\partial b_i}=\frac{\partial^2\ell_i^{(1)}}{\partial\mathbf{w}_i\partial b_i};\quad \frac{\partial^2\ell}{\partial\mathbf{w}_i\partial e_i}=\frac{\partial^2\ell_i^{(2)}}{\partial\mathbf{w}_i\partial e_i};\tag{17}$$

$$\frac{\partial^2\ell}{\partial b_i^2}=\frac{\partial^2\ell_i^{(1)}}{\partial b_i^2};\quad \frac{\partial^2\ell}{\partial b_i\partial e_i}=\frac{\partial^2\ell_i}{\partial b_i\partial e_i}=0;\quad \frac{\partial^2\ell}{\partial e_i^2}=\frac{\partial^2\ell_i^{(2)}}{\partial e_i^2}.\tag{18}$$

As a result, the estimation of the components in $\boldsymbol{\theta}=(\boldsymbol{\theta}_1^\top,\ldots,\boldsymbol{\theta}_n^\top)^\top$ does not need to be performed simultaneously. Instead, we can cycle through the estimation of $\boldsymbol{\theta}_i$, for $i=1,\ldots,n$, one by one iteratively. After $\boldsymbol{\theta}_i$ is updated, the estimated value of $\boldsymbol{\theta}_1,\boldsymbol{\theta}_2,\ldots,\boldsymbol{\theta}_i$ is used to update $\boldsymbol{\theta}_k, k=i+1,\ldots,n$.

The first part of the alternating regression has the following updating equation based on the Fisher scoring algorithm:

$$\boldsymbol{\theta}_i^{(t+1)}=\boldsymbol{\theta}_i^{(t)}+S_{\boldsymbol{\theta}_i^{(t)}}^{-1}U_{\boldsymbol{\theta}_i^{(t)}},\quad i=1,\ldots,n,\tag{19}$$

For each i and t, this update requires the value of $\boldsymbol{\theta}_i$ and their score equations and information matrix at the t^{th} iteration. One iteration alone here is not taking advantage of the data because retrieving the i^{th} row of data takes a significant amount of time when the dimension of the data matrix is huge. Therefore, for each row of data retrieved, it is better to update $\boldsymbol{\theta}_i^{(t)}$ a certain number of times. Specifically, the $\boldsymbol{\theta}_i^{(t)}$ is updated again and again in a loop of E epochs and the updated values are used to recompute the score equations and information matrix, all of which are used for next epoch's update. After all epochs are completed, $\boldsymbol{\theta}_i^{(t)}$ takes the value of $\boldsymbol{\theta}_i^{(t,E)}$ at the end of all iterations from E epochs based on the updating formula below:

$$\boldsymbol{\theta}_i^{(t,k+1)} = \boldsymbol{\theta}_i^{(t,k)} + S^{-1}_{\boldsymbol{\theta}_i^{(t,k)}} U_{\boldsymbol{\theta}_i^{(t,k)}}, \quad k = 1, \ldots, E, \tag{20}$$

This inner loop of updates makes good use of the already loaded data to refine the estimate of $\boldsymbol{\theta}_i^{(t)}$ so that the end estimate is closer to its MLE when the same current $\widetilde{\boldsymbol{\theta}}^{(c)}$ values are used. Note that the updated $\boldsymbol{\theta}_i$ in each epoch from (20) leads to changes in the score equations and Fisher information matrix, whose update will in turn result in a better estimate of $\boldsymbol{\theta}_i$. Such multiple rounds of updates in (20) reduce the variation of update in $\boldsymbol{\theta}_i^{(t)}$ when we go through many iterations based on Equation (19). It effectively reduces the number of times to retrieve the data. Below are the formulae involved in the updating equations:

$$\boldsymbol{\theta}_i^{(t)} = \left(\mathbf{w}_i^{\top(t)}, b_i^{(t)}, e_i^{(t)}\right)^\top; \quad U_{\boldsymbol{\theta}_i^{(t)}} = \left[\left(\frac{\partial \ell_i}{\partial \mathbf{w}_i}\right)^\top, \frac{\partial \ell_i}{\partial b_i}, \frac{\partial \ell_i}{\partial e_i}\right]^\top \Bigg|_{\boldsymbol{\theta}_i^{(t)}}; \quad S_{\boldsymbol{\theta}_i^{(t)}} = \begin{bmatrix} S^{(t)}_{\mathbf{w}_i} & S^{(t)}_{\mathbf{w}_i b_i} & S^{(t)}_{\mathbf{w}_i e_i} \\ S^{(t)}_{b_i \mathbf{w}_i} & S^{(t)}_{b_i} & S^{(t)}_{b_i e_i} \\ S^{(t)}_{e_i \mathbf{w}_i} & S^{(t)}_{e_i b_i} & S^{(t)}_{e_i} \end{bmatrix}$$

with

$$\frac{\partial \ell_i}{\partial \mathbf{w}_i}\Bigg|_{\boldsymbol{\theta}_i^{(t)}} = \sum_{k=1}^{2} \frac{\partial \ell_i^{(k)}(\boldsymbol{\theta}_i^{(t)}; \widetilde{\boldsymbol{\theta}}^{(c)})}{\partial \mathbf{w}_i}, \qquad \frac{\partial \ell_i}{\partial b_i}\Bigg|_{\boldsymbol{\theta}_i^{(t)}} = \frac{\partial \ell_i^{(1)}(\boldsymbol{\theta}_i^{(t)}; \widetilde{\boldsymbol{\theta}}^{(c)})}{\partial b_i}, \qquad \frac{\partial \ell_i}{\partial e_i}\Bigg|_{\boldsymbol{\theta}_i^{(t)}} = \frac{\partial \ell_i^{(2)}(\boldsymbol{\theta}_i^{(t)}; \widetilde{\boldsymbol{\theta}}^{(c)})}{\partial e_i},$$

where these partial derivatives are the $\frac{\partial \ell_i^{(1)}}{\partial \mathbf{w}_i}$, $\frac{\partial \ell_i^{(2)}}{\partial \mathbf{w}_i}$, $\frac{\partial \ell_i^{(1)}}{\partial b_i}$, and $\frac{\partial \ell_i^{(2)}}{\partial e_i}$ evaluated at $\boldsymbol{\theta}_i^{(t)}$ and $\widetilde{\boldsymbol{\theta}}^{(c)}$ using formulae in (8) and (12), and

$$S^{(t)}_{\mathbf{w}_i} = E\left[-\frac{\partial^2 \ell_i^{(1)}(\boldsymbol{\theta}_i^{(t)}; \widetilde{\boldsymbol{\theta}}^{(c)})}{\partial \mathbf{w}_i \partial \mathbf{w}_i^\top}\right] + E\left[-\frac{\partial^2 \ell_i^{(2)}(\boldsymbol{\theta}_i^{(t)}; \widetilde{\boldsymbol{\theta}}^{(c)})}{\partial \mathbf{w}_i \partial \mathbf{w}_i^\top}\Bigg| g_{ij} = 1\right];$$

$$S^{(t)}_{\mathbf{w}_i b_i} = E\left[-\frac{\partial^2 \ell_i^{(1)}(\boldsymbol{\theta}_i^{(t)}; \widetilde{\boldsymbol{\theta}}^{(c)})}{\partial \mathbf{w}_i \partial b_i}\right]; \qquad S^{(t)}_{b_i} = E\left[-\frac{\partial^2 \ell_i^{(1)}(\boldsymbol{\theta}_i^{(t)}; \widetilde{\boldsymbol{\theta}}^{(c)})}{\partial b_i^2}\right]; \qquad S^{(t)}_{b_i e_i} = 0;$$

$$S^{(t)}_{\mathbf{w}_i e_i} = E\left[-\frac{\partial^2 \ell_i^{(2)}(\boldsymbol{\theta}_i^{(t)}; \widetilde{\boldsymbol{\theta}}^{(c)})}{\partial \mathbf{w}_i \partial e_i}\Bigg| g_{ij} = 1\right]; \qquad S^{(t)}_{e_i} = E\left[-\frac{\partial^2 \ell_i^{(2)}(\boldsymbol{\theta}_i^{(t)}; \widetilde{\boldsymbol{\theta}}^{(c)})}{\partial e_i^2}\Bigg| g_{ij} = 1;\right],$$

where the expectations are based on Formulae (9), (10), (13)–(15) evaluated at $\boldsymbol{\theta}_i^{(t)}$ and $\widetilde{\boldsymbol{\theta}}^{(c)}$ if it is in the outer loop of the update and using the values of $\boldsymbol{\theta}^{(t,k)}$ and $\widetilde{\boldsymbol{\theta}}^{(c)}$ when it is in the inner loop of updates.

This concludes the first part of the alternating ZIG regression. It gives an iterative update of $\boldsymbol{\theta}$ while $\widetilde{\boldsymbol{\theta}}$ is fixed. After we finish updating all $\boldsymbol{\theta}_i$'s, we move on to treat the $\boldsymbol{\theta}$ as fixed to estimate $\widetilde{\boldsymbol{\theta}}$.

Starting with $\widetilde{\ell}_j = \widetilde{\ell}_j^{(1)} + \widetilde{\ell}_j^{(2)}$, we consider the partial derivatives with respect to $\widetilde{\boldsymbol{\theta}}_j$ while holding $\boldsymbol{\theta}$ fixed. The derivations are similar to those for ℓ_i but still we list them for clarity. Note that

$$\begin{aligned}\frac{\partial p_{ij}}{\partial \widetilde{\boldsymbol{w}}_j} &= \frac{\exp(\eta_{ij}) \cdot \mathbf{w}_i \{1 + \exp(\eta_{ij})\} - \exp(2\eta_{ij}) \cdot \mathbf{w}_i}{\{1 + \exp(\eta_{ij})\}^2} = p_{ij}(1 - p_{ij})\mathbf{w}_i, \\ \frac{\partial p_{ij}}{\partial \widetilde{b}_j} &= p_{ij}(1 - p_{ij}).\end{aligned}$$

The first-order partial derivatives regarding $\widetilde{\ell}_j^{(1)}$ are

$$\frac{\partial \widetilde{\ell}_j^{(1)}}{\partial \widetilde{\boldsymbol{w}}_j} = \sum_{i=1}^{n} (g_{ij} - p_{ij}) \cdot \mathbf{w}_i, \qquad \frac{\partial \widetilde{\ell}_j^{(1)}}{\partial \widetilde{b}_j} = \sum_{i=1}^{n} (g_{ij} - p_{ij}).$$

The negative second derivatives and their expectations are

$$-\frac{\partial^2 \widetilde{\ell}_j^{(1)}}{\partial \widetilde{\boldsymbol{w}}_j \partial \widetilde{\boldsymbol{w}}_j^\top} = \sum_{i=1}^{n} \mathbf{w}_i \cdot \left(\frac{\partial p_{ij}}{\partial \widetilde{\boldsymbol{w}}_j^\top} \right) = \sum_{i=1}^{n} \mathbf{w}_i \mathbf{w}_i^\top p_{ij}(1 - p_{ij}) = E\left[-\frac{\partial^2 \widetilde{\ell}_j^{(1)}}{\partial \widetilde{\boldsymbol{w}}_j \partial \widetilde{\boldsymbol{w}}_j^\top} \right], \quad (21)$$

$$-\frac{\partial^2 \widetilde{\ell}_j^{(1)}}{\partial \widetilde{\boldsymbol{w}}_j \partial \widetilde{b}_j} = \sum_{i=1}^{n} \mathbf{w}_i p_{ij}(1 - p_{ij}) = E\left[-\frac{\partial^2 \widetilde{\ell}_j^{(1)}}{\partial \widetilde{\boldsymbol{w}}_j \partial \widetilde{b}_j} \right], \quad (22)$$

$$-\frac{\partial^2 \widetilde{\ell}_j^{(1)}}{\partial \widetilde{b}_j^2} = \sum_{i=1}^{n} p_{ij}(1 - p_{ij}) = E\left[-\frac{\partial^2 \widetilde{\ell}_j^{(1)}}{\partial \widetilde{b}_j^2} \right]. \quad (23)$$

The first-order partial derivatives of the second component are $\widetilde{\ell}_j^{(2)}$.

$$\frac{\partial \widetilde{\ell}_j^{(2)}}{\partial \widetilde{\boldsymbol{w}}_j} = \sum_{i=1}^{n} g_{ij} \left\{ \frac{v_{ij} \cdot \mathbf{w}_i}{\mathbf{w}_i^\top \widetilde{\boldsymbol{w}}_j + e_i + \widetilde{e}_j} + v_{ij} y_{ij} \mathbf{w}_i \right\}, \quad \frac{\partial \widetilde{\ell}_j^{(2)}}{\partial \widetilde{e}_j} = \sum_{i=1}^{n} g_{ij} \left\{ \frac{v_{ij}}{\mathbf{w}_i^\top \widetilde{\boldsymbol{w}}_j + e_i + \widetilde{e}_j} + v_{ij} y_{ij} \right\}.$$

The negative second-order partial derivatives and their expectations are

$$\begin{aligned}-\frac{\partial^2 \widetilde{\ell}_j^{(2)}}{\partial \widetilde{\boldsymbol{w}}_j \partial \widetilde{\boldsymbol{w}}_j^\top}\bigg|_{g_{ij}=1} &= \sum_{\substack{i=1 \\ g_{ij}=1}}^{n} \frac{v_{ij} \mathbf{w}_i \mathbf{w}_i^\top}{(\mathbf{w}_i^\top \widetilde{\boldsymbol{w}}_j + e_i + \widetilde{e}_j)^2} = E\left[-\frac{\partial^2 \widetilde{\ell}_j^{(2)}}{\partial \widetilde{\boldsymbol{w}}_j \partial \widetilde{\boldsymbol{w}}_j^\top} \bigg| g_{ij} = 1 \right], \\ -\frac{\partial^2 \widetilde{\ell}_j^{(2)}}{\partial \widetilde{\boldsymbol{w}}_j \partial \widetilde{e}_j}\bigg|_{g_{ij}=1} &= \sum_{\substack{i=1 \\ g_{ij}=1}}^{n} \frac{v_{ij} \mathbf{w}_i}{(\mathbf{w}_i^\top \widetilde{\boldsymbol{w}}_j + e_i + \widetilde{e}_j)^2} = E\left[-\widetilde{\frac{\partial^2 \ell_j^{(2)}}{\partial \widetilde{\boldsymbol{w}}_j \partial \widetilde{e}_j}} \bigg| g_{ij} = 1 \right], \\ -\frac{\partial^2 \widetilde{\ell}_j^{(2)}}{\partial \widetilde{e}_j^2}\bigg|_{g_{ij}=1} &= \sum_{\substack{i=1 \\ g_{ij}=1}}^{n} v_{ij} \left(\mathbf{w}_i^\top \widetilde{\boldsymbol{w}}_j + e_i + \widetilde{e}_j \right)^{-2} = E\left[-\frac{\partial^2 \widetilde{\ell}_j^{(2)}}{\partial \widetilde{e}_j^2} \bigg| g_{ij} = 1 \right].\end{aligned}$$

Therefore, the other side of the updating equation for the alternating ZIG regression based on the Fisher scoring algorithm is

$$\widetilde{\boldsymbol{\theta}}_j^{(t+1)} = \widetilde{\boldsymbol{\theta}}_j^{(t)} + S^{-1}_{\widetilde{\boldsymbol{\theta}}_j^{(t)}} U_{\widetilde{\boldsymbol{\theta}}_j^{(t)}}, \quad j = 1, \ldots, n; \qquad \widetilde{\boldsymbol{\theta}}_j^{(t,k+1)} = \widetilde{\boldsymbol{\theta}}_j^{(t,k)} + S^{-1}_{\widetilde{\boldsymbol{\theta}}_j^{(t,k)}} U_{\widetilde{\boldsymbol{\theta}}_j^{(t,k)}}, \quad k = 1, \ldots, E.$$

Again, for each j and iteration number t, this equation is iterated for a certain number of epochs to obtain a refined estimate of $\widetilde{\boldsymbol{\theta}}_j$ based on the current value $\boldsymbol{\theta}^{(c)}$ without having to reload the data. The quantities in the updating equation are listed below.

$$\widetilde{\boldsymbol{\theta}}_j^{(t)} = \left(\widetilde{\boldsymbol{w}}_j^{\top(t)}, \widetilde{b}_j^{(t)}, \widetilde{e}_j^{(t)}\right)^{\top}; \quad U_{\widetilde{\boldsymbol{\theta}}_j^{(t)}} = \left[\begin{array}{c} \frac{\partial \widetilde{\ell}_j}{\partial \widetilde{\boldsymbol{w}}_j} \\ \frac{\partial \widetilde{\ell}_j}{\partial \widetilde{b}_j}, \\ \frac{\partial \widetilde{\ell}_j}{\partial \widetilde{e}_j} \end{array}\right]\Bigg|_{\widetilde{\boldsymbol{\theta}}_j^{(t)}}; \quad S_{\widetilde{\boldsymbol{\theta}}_j^{(t)}} = \left[\begin{array}{ccc} S^{(t)}_{\widetilde{\boldsymbol{w}}_j} & S^{(t)}_{\widetilde{\boldsymbol{w}}_j \widetilde{b}_j} & S^{(t)}_{\widetilde{\boldsymbol{w}}_j \widetilde{e}_j} \\ S^{(t)}_{\widetilde{b}_j \widetilde{\boldsymbol{w}}_j} & S^{(t)}_{\widetilde{b}_j} & S^{(t)}_{\widetilde{b}_j \widetilde{e}_j} \\ S^{(t)}_{\widetilde{e}_j \widetilde{\boldsymbol{w}}_j} & S^{(t)}_{\widetilde{e}_j \widetilde{b}_j} & S^{(t)}_{\widetilde{e}_j} \end{array}\right]$$

with

$$\frac{\partial \widetilde{\ell}_j}{\partial \widetilde{\boldsymbol{w}}_j}\Bigg|_{\widetilde{\boldsymbol{\theta}}_j^{(t)}} = \sum_{k=1}^{2} \frac{\partial \widetilde{\ell}_j^{(k)}(\widetilde{\boldsymbol{\theta}}_j^{(t)}; \boldsymbol{\theta}^{(c)})}{\partial \widetilde{\boldsymbol{w}}_j}, \qquad \frac{\partial \widetilde{\ell}_j}{\partial \widetilde{b}_j}\Bigg|_{\widetilde{\boldsymbol{\theta}}_j^{(t)}} = \frac{\partial \widetilde{\ell}_j^{(1)}(\widetilde{\boldsymbol{\theta}}_j^{(t)}; \boldsymbol{\theta}^{(c)})}{\partial \widetilde{b}_j}, \qquad \frac{\partial \widetilde{\ell}_j}{\partial \widetilde{e}_j}\Bigg|_{\widetilde{\boldsymbol{\theta}}_j^{(t)}} = \frac{\partial \widetilde{\ell}_j^{(2)}(\widetilde{\boldsymbol{\theta}}_j^{(t)}; \boldsymbol{\theta}^{(c)})}{\partial \widetilde{e}_j},$$

where these partial derivatives are $\frac{\partial \widetilde{\ell}_j^{(1)}}{\partial \widetilde{\boldsymbol{w}}_j}$, $\frac{\partial \widetilde{\ell}_j^{(2)}}{\partial \widetilde{\boldsymbol{w}}_j}$, $\frac{\partial \widetilde{\ell}_j^{(1)}}{\partial \widetilde{b}_j}$, and $\frac{\partial \widetilde{\ell}_j^{(2)}}{\partial \widetilde{e}_j}$ evaluated at $\widetilde{\boldsymbol{\theta}}_j^{(t)}$ and $\boldsymbol{\theta}^{(c)}$, and

$$S^{(t)}_{\widetilde{\boldsymbol{w}}_j} = E\left[-\frac{\partial^2 \widetilde{\ell}_j^{(1)}(\widetilde{\boldsymbol{\theta}}_j^{(t)}; \boldsymbol{\theta}^{(c)})}{\partial \widetilde{\boldsymbol{w}}_j \partial \widetilde{\boldsymbol{w}}_j^{\top}}\right] + E\left[-\frac{\partial^2 \widetilde{\ell}_j^{(2)}(\widetilde{\boldsymbol{\theta}}_j^{(t)}; \boldsymbol{\theta}^{(c)})}{\partial \widetilde{\boldsymbol{w}}_j \partial \widetilde{\boldsymbol{w}}_j^{\top}}\Bigg| g_{ij} = 1\right];$$

$$S^{(t)}_{\widetilde{\boldsymbol{w}}_j \widetilde{b}_j} = E\left[-\frac{\partial^2 \widetilde{\ell}_j^{(1)}(\widetilde{\boldsymbol{\theta}}_j^{(t)}; \boldsymbol{\theta}^{(c)})}{\partial \widetilde{\boldsymbol{w}}_j \partial \widetilde{b}_j}\right]; \quad S^{(t)}_{\widetilde{b}_j} = E\left[-\frac{\partial^2 \widetilde{\ell}_j^{(1)}(\widetilde{\boldsymbol{\theta}}_j^{(t)}; \boldsymbol{\theta}^{(c)},)}{\partial \widetilde{b}_j^2}\right]; \quad S^{(t)}_{\widetilde{b}_j \widetilde{e}_j} = 0;$$

$$S^{(t)}_{\widetilde{\boldsymbol{w}}_j \widetilde{e}_j} = E\left[-\frac{\partial^2 \widetilde{\ell}_j^{(2)}(\widetilde{\boldsymbol{\theta}}_j^{(t)}; \boldsymbol{\theta}^{(c)})}{\partial \widetilde{\boldsymbol{w}}_j \partial \widetilde{e}_j}\Bigg| g_{ij} = 1\right]; \quad S^{(t)}_{\widetilde{e}_j} = E\left[-\frac{\partial^2 \widetilde{\ell}_j^{(2)}(\widetilde{\boldsymbol{\theta}}_j^{(t)}; \boldsymbol{\theta}^{(c)})}{\partial \widetilde{e}_j^2}\Bigg| g_{ij} = 1\right].$$

The expectations involved in the above formulae were using $\widetilde{\boldsymbol{\theta}}^{(t)}$ at the t^{th} iteration and the current values $\boldsymbol{\theta}^{(c)}$ if it is in the outer loop of the update and using the values of $\widetilde{\boldsymbol{\theta}}^{(t,k)}$ and $\boldsymbol{\theta}^{(c)}$ when it is in the inner loop of updates.

All of the formulae above assume the parameter v_{ij} is known. When it is not known, the parameter can be estimated with MLE, bias corrected MLE, or moment estimator. Simulation studies in [21] suggested that when the sample size is large, all the estimators perform similarly but the moment estimator has the advantage of being easy to compute. If the sample size is medium, then bias corrected MLE is better.

The canonical link for the Gamma regression part could encounter problems. Recall that the canonical link for the Gamma regression is $g(\mu_{ij}) = -\mu_{ij}^{-1} = \mathbf{w}_i^{\top} \widetilde{\boldsymbol{w}}_j + e_i + \widetilde{e}_j$. The left-hand side of the equation is required to be negative because the support of the Gamma distribution is $(0, \infty)$. However, the right-hand side of the equation could freely take any value in $(-\infty, \infty)$. Due to this conflict in natural parameter space and the range of the estimated value, the likelihood function sometimes could become undefined because $\log(\mu_{ij}^{-1})$ appears in it but $\log(\widehat{\mu}_{ij}^{-1})$ cannot be evaluated for negative $\widehat{\mu}_{ij}$ (see Equation (11)).

4. Using Log Link for Gamma Regression

In this section, we consider using the log link to model the Gamma regression part while maintaining the logistic regression part. The settings of the two-part model are similar to those in Section 3 except that modifying the link function from canonical to log link changes the score equations and the Hessian matrix.

For updating formulae on the model parameters based on Fisher scoring algorithm, the first-order and second-order partial derivatives of $\ell_i^{(1)}$ remain the same as in (8)–(10) because the new link function does not appear in the Bernoulli part. Now consider the

second component of the log likelihood $\ell_i^{(2)}$, which was given in (5). In this section, denote $\tau_{ij} = \mathbf{w}_i^\top \widetilde{\boldsymbol{w}}_j + e_i + \widetilde{e}_j$ as in the previous subsection. Then, the log likelihood of the ith row of non-zero observations in the co-occurrence matrix, corresponding to the Gamma distribution, can be expressed as

$$\begin{aligned}\ell_i^{(2)} &= \sum_{j=1}^{n} I(y_{ij} > 0) \cdot \log f(y_{ij} \mid y_{ij} > 0) = \sum_{j=1}^{n} g_{ij} \cdot \log f(y_{ij} \mid y_{ij} > 0) \\ &= \sum_{j=1}^{n} g_{ij} \left\{ -\log \Gamma(v_{ij}) + v_{ij} \log v_{ij} + (v_{ij} - 1) \log y_{ij} - v_{ij}\tau_{ij} - v_{ij} y_{ij} \frac{1}{\exp(\tau_{ij})} \right\}. \end{aligned} \tag{24}$$

The equations below give the first-order partial derivatives of $\ell_i^{(2)}$ with respect to the parameters $\mathbf{w}_i$ and e_i.

$$\begin{aligned}\frac{\partial \ell_i^{(2)}}{\partial \mathbf{w}_i} &= \sum_{j=1}^{n} -g_{ij} v_{ij} \left\{ \frac{\partial \tau_{ij}}{\partial \mathbf{w}_i} + \frac{y_{ij}}{\mu_{ij}} \left(-\frac{\partial \tau_{ij}}{\partial \mathbf{w}_i} \right) \right\} = \sum_{j=1}^{n} g_{ij} v_{ij} \mu_{ij}^{-1} (y_{ij} - \mu_{ij}) \widetilde{\boldsymbol{w}}_j, \\ \frac{\partial \ell_i^{(2)}}{\partial e_i} &= \sum_{j=1}^{n} g_{ij} v_{ij} \mu_{ij}^{-1} (y_{ij} - \mu_{ij}). \end{aligned} \tag{25}$$

To obtain the second-order partial derivatives, first note that

$$\mu_{ij} = \exp(\tau_{ij}) \quad \Rightarrow \frac{\partial \mu_{ij}}{\partial \mathbf{w}_i} = \frac{\partial \mu_{ij}}{\partial \tau_{ij}} \frac{\partial \tau_{ij}}{\partial \mathbf{w}_i} = \mu_{ij} \widetilde{\mathbf{w}}_j, \quad \text{and} \quad \frac{\partial \mu_{ij}}{\partial e_i} = \frac{\partial \mu_{ij}}{\partial \tau_{ij}} \frac{\partial \tau_{ij}}{\partial e_i} = \mu_{ij}.$$

Then, the second-order partial derivatives and their negative expectations which are components of the Fisher information matrix can be derived as

$$\frac{\partial^2 \ell_i^{(2)}}{\partial \mathbf{w}_i \partial \mathbf{w}_i^\top} = -\sum_{j=1}^{n} g_{ij} v_{ij} \left[\frac{y_{ij} - \mu_{ij}}{\mu_{ij}} + 1 \right] \widetilde{\boldsymbol{w}}_j \widetilde{\boldsymbol{w}}_j^\top \Rightarrow E\left[\frac{-\partial^2 \ell_i^{(2)}}{\partial \mathbf{w}_i \partial \mathbf{w}_i^\top} \middle| g_{ij} = 1 \right] = \sum_{j=1}^{n} g_{ij} v_{ij} \widetilde{\boldsymbol{w}}_j \widetilde{\boldsymbol{w}}_j^\top, \tag{26}$$

$$\frac{\partial^2 \ell_i^{(2)}}{\partial \mathbf{w}_i \partial e_i} = -\sum_{j=1}^{n} g_{ij} v_{ij} \left[\frac{y_{ij} - \mu_{ij}}{\mu_{ij}} + 1 \right] \widetilde{\boldsymbol{w}}_j \Longrightarrow E\left[-\frac{\partial^2 \ell_i^{(2)}}{\partial \mathbf{w}_i \partial e_i} \middle| g_{ij} = 1 \right] = \sum_{j=1}^{n} g_{ij} v_{ij} \widetilde{\boldsymbol{w}}_j, \tag{27}$$

$$\frac{\partial^2 \ell_i^{(2)}}{\partial e_i^2} = -\sum_{j=1}^{n} g_{ij} v_{ij} \left[\frac{y_{ij} - \mu_{ij}}{\mu_{ij}} + 1 \right] \Longrightarrow E\left[-\frac{\partial^2 \ell_i^{(2)}}{\partial e_i^2} \middle| g_{ij} = 1 \right] = \sum_{j=1}^{n} g_{ij} v_{ij}. \tag{28}$$

The first part of the alternating regression has the following updating equation based on the Fisher scoring algorithm:

$$\begin{aligned}\boldsymbol{\theta}_i^{(t+1)} &= \boldsymbol{\theta}_i^{(t)} + S_{\boldsymbol{\theta}_i^{(t)}}^{-1} U_{\boldsymbol{\theta}_i^{(t)}}, \quad i = 1, \ldots, n, \\ \boldsymbol{\theta}_i^{(t,k+1)} &= \boldsymbol{\theta}_i^{(t,k)} + S_{\boldsymbol{\theta}_i^{(t,k)}}^{-1} U_{\boldsymbol{\theta}_i^{(t,k)}}, \quad k = 1, \ldots, E. \end{aligned} \tag{29}$$

The updating equation in (29) is iterated E epochs, say E equals 20, for the same i and t, where t is the iteration number. The multiple epochs here allow the estimate of $\boldsymbol{\theta}_i$ to get closer to its MLE for given current value $\widetilde{\boldsymbol{\theta}}^{(c)}$ as the score equations and information matrix are also updated with the estimated parameter value. There is no need to have too many

epochs because the parameter $\widetilde{\boldsymbol{\theta}}^{(c)}$ is not the true value yet and still needs to be estimated later. Below are the formulae involved in the updating equations:

$$\boldsymbol{\theta}_i^{(t)} = \left(\mathbf{w}_i^{\top(t)}, b_i^{(t)}, e_i^{(t)}\right)^{\top}; \quad U_{\boldsymbol{\theta}_i^{(t)}} = \left[\left(\frac{\partial \ell_i}{\partial \mathbf{w}_i}\right)^{\top}, \frac{\partial \ell_i}{\partial b_i}, \frac{\partial \ell_i}{\partial e_i}\right]^{\top}\Bigg|_{\boldsymbol{\theta}_i^{(t)}}; \quad S_{\boldsymbol{\theta}_i^{(t)}} = \begin{bmatrix} S_{\mathbf{w}_i}^{(t)} & S_{\mathbf{w}_i b_i}^{(t)} & S_{\mathbf{w}_i e_i}^{(t)} \\ S_{b_i \mathbf{w}_i}^{(t)} & S_{b_i}^{(t)} & S_{b_i e_i}^{(t)} \\ S_{e_i \mathbf{w}_i}^{(t)} & S_{e_i b_i}^{(t)} & S_{e_i}^{(t)} \end{bmatrix}$$

with

$$\frac{\partial \ell_i}{\partial \mathbf{w}_i}\Bigg|_{\boldsymbol{\theta}_i^{(t)}} = \sum_{k=1}^{2} \frac{\partial \ell_i^{(k)}(\boldsymbol{\theta}_i^{(t)}; \widetilde{\boldsymbol{\theta}}^{(c)})}{\partial \mathbf{w}_i}, \quad \frac{\partial \ell_i}{\partial b_i}\Bigg|_{\boldsymbol{\theta}_i^{(t)}} = \frac{\partial \ell_i^{(1)}(\boldsymbol{\theta}_i^{(t)}; \widetilde{\boldsymbol{\theta}}^{(c)})}{\partial b_i}, \quad \frac{\partial \ell_i}{\partial e_i}\Bigg|_{\boldsymbol{\theta}_i^{(t)}} = \frac{\partial \ell_i^{(2)}(\boldsymbol{\theta}_i^{(t)}; \widetilde{\boldsymbol{\theta}}^{(c)})}{\partial e_i},$$

where these partial derivatives are $\frac{\partial \ell_i^{(1)}}{\partial \mathbf{w}_i}$, $\frac{\partial \ell_i^{(2)}}{\partial \mathbf{w}_i}$, $\frac{\partial \ell_i^{(1)}}{\partial b_i}$, and $\frac{\partial \ell_i^{(2)}}{\partial e_i}$ evaluated at $\boldsymbol{\theta}_i^{(t)}$ and $\widetilde{\boldsymbol{\theta}}^{(c)}$ using Equations (8) and (25), and

$$S_{\mathbf{w}_i}^{(t)} = E\left[-\frac{\partial^2 \ell_i^{(1)}(\boldsymbol{\theta}_i^{(t)}; \widetilde{\boldsymbol{\theta}}^{(c)})}{\partial \mathbf{w}_i \partial \mathbf{w}_i^{\top}}\right] + E\left[-\frac{\partial^2 \ell_i^{(2)}(\boldsymbol{\theta}_i^{(t)}; \widetilde{\boldsymbol{\theta}}^{(c)})}{\partial \mathbf{w}_i \partial \mathbf{w}_i^{\top}}\middle| g_{ij} = 1\right];$$

$$S_{\mathbf{w}_i b_i}^{(t)} = E\left[-\frac{\partial^2 \ell_i^{(1)}(\boldsymbol{\theta}_i^{(t)}; \widetilde{\boldsymbol{\theta}}^{(c)})}{\partial \mathbf{w}_i \partial b_i}\right]; \quad S_{b_i}^{(t)} = E\left[-\frac{\partial^2 \ell_i^{(1)}(\boldsymbol{\theta}_i^{(t)}; \widetilde{\boldsymbol{\theta}}^{(c)})}{\partial b_i^2}\right]; \quad S_{b_i e_i}^{(t)} = 0;$$

$$S_{\mathbf{w}_i e_i}^{(t)} = E\left[-\frac{\partial^2 \ell_i^{(2)}(\boldsymbol{\theta}_i^{(t)}; \widetilde{\boldsymbol{\theta}}^{(c)})}{\partial \mathbf{w}_i \partial e_i}\middle| g_{ij} = 1\right]; \quad S_{e_i}^{(t)} = E\left[-\frac{\partial^2 \ell_i^{(2)}(\boldsymbol{\theta}_i^{(t)}; \widetilde{\boldsymbol{\theta}}^{(c)})}{\partial e_i^2}\middle| g_{ij} = 1;\right].$$

The expectations involved in the above formulae were given in (9)–(10) and (26)–(28) except that the parameters are using the current values $\boldsymbol{\theta}^{(t)}$ and $\widetilde{\boldsymbol{\theta}}^{(c)}$ at the t^{th} iteration if it is in the outer loop of the update and using the values of $\boldsymbol{\theta}^{(t,k)}$ and $\widetilde{\boldsymbol{\theta}}^{(c)}$ when it is in the inner loop of updates.

The aforementioned equations are used in updating the $\boldsymbol{\theta}$ part of the alternating ZIG regression. Now, consider the other side of the alternating ZIG regression in which updates are performed for $\widetilde{\boldsymbol{\theta}}$ while holding $\boldsymbol{\theta}$ fixed. First, consider the first-order partial derivatives

$$\frac{\partial \widetilde{\ell}}{\partial \widetilde{\mathbf{w}}_j} = \sum_{i=1}^{n}\left\{\frac{\partial \widetilde{\ell}_{ij}^{(1)}}{\partial \widetilde{\mathbf{w}}_j} + \frac{\partial \widetilde{\ell}_{ij}^{(2)}}{\partial \widetilde{\mathbf{w}}_j}\right\} = \frac{\partial \sum_{i=1}^{n} \widetilde{\ell}_{ij}^{(1)}}{\partial \widetilde{\mathbf{w}}_j} + \frac{\partial \sum_{i=1}^{n} \widetilde{\ell}_{ij}^{(2)}}{\partial \widetilde{\mathbf{w}}_j} = \frac{\partial \widetilde{\ell}_j^{(1)}}{\partial \widetilde{\mathbf{w}}_j} + \frac{\partial \widetilde{\ell}_j^{(2)}}{\partial \widetilde{\mathbf{w}}_j}.$$

The first-order partial derivatives of $\widetilde{\ell}_j^{(1)}$ are

$$\frac{\partial \widetilde{\ell}_j^{(1)}}{\partial \widetilde{\mathbf{w}}_j} = \sum_{i=1}^{n}(g_{ij} - p_{ij})\mathbf{w}_i, \quad \frac{\partial \widetilde{\ell}_j^{(1)}}{\partial \widetilde{b}_j} = \sum_{i=1}^{n}(g_{ij} - p_{ij}), \quad \frac{\partial \widetilde{\ell}_j^{(1)}}{\partial \widetilde{e}_j} = 0.$$

The second-order partial derivatives $\frac{\partial^2 \widetilde{\ell}_j^{(1)}}{\partial \widetilde{\mathbf{w}}_j \partial \widetilde{\mathbf{w}}_j^{\top}}$, $\frac{\partial^2 \widetilde{\ell}_j^{(1)}}{\partial \widetilde{b}_j \partial \widetilde{e}_j}$, $\frac{\partial^2 \widetilde{\ell}_j^{(1)}}{\partial \widetilde{b}_j^2}$ and their expectations are same as in the canonical link case, which are given in Equations (21)–(23). Additionally,

$$\frac{\partial^2 \widetilde{\ell}_j^{(1)}}{\partial \widetilde{b}_j \partial \widetilde{e}_j} = 0; \quad \frac{\partial^2 \widetilde{\ell}_j^{(1)}}{\partial \widetilde{e}_j^2} = 0; \quad \Longrightarrow \quad E\left[-\frac{\partial^2 \widetilde{\ell}_j^{(1)}}{\partial \widetilde{b}_j \partial \widetilde{e}_j}\right] = 0; \quad E\left[-\frac{\partial^2 \widetilde{\ell}_j^{(1)}}{\partial \widetilde{e}_j^2}\right] = 0.$$

Next, consider the first-order partial derivatives of the second component $\widetilde{\ell}_j^{(2)}$

$$\frac{\partial \widetilde{\ell}_j^{(2)}}{\partial \widetilde{\boldsymbol{w}}_j} = \sum_{i=1}^{n} g_{ij} v_{ij} \mu_{ij}^{-1} (y_{ij} - \mu_{ij}) \mathbf{w}_i, \quad \frac{\partial \widetilde{\ell}_j^{(2)}}{\partial \widetilde{e}_j} = \sum_{i=1}^{n} g_{ij} v_{ij} \mu_{ij}^{-1} (y_{ij} - \mu_{ij}).$$

To derive the second-order partial derivatives, note that two frequently used terms are

$$\mu_{ij} = \exp(\tau_{ij}) \quad \Rightarrow \frac{\partial \mu_{ij}}{\partial \widetilde{\boldsymbol{w}}_j} = \frac{\partial \mu_{ij}}{\partial \tau_{ij}} \cdot \frac{\partial \tau_{ij}}{\partial \widetilde{\boldsymbol{w}}_j} = \mu_{ij} \mathbf{w}_i \quad \text{and} \quad \frac{\partial \mu_{ij}}{\partial \widetilde{e}_j} = \frac{\partial \mu_{ij}}{\partial \tau_{ij}} \cdot \frac{\partial \tau_{ij}}{\partial \widetilde{e}_j} = \mu_{ij}.$$

Using the two terms, the second derivatives can be written as

$$\frac{\partial^2 \ell_i^{(2)}}{\partial \widetilde{\boldsymbol{w}}_j \partial \widetilde{\boldsymbol{w}}_j^\top} = -\sum_{i=1}^{n} g_{ij} v_{ij} \left[\frac{y_{ij} - \mu_{ij}}{\mu_{ij}} + 1 \right] \mathbf{w}_i \mathbf{w}_i^\top \Rightarrow E\left[-\frac{\partial^2 \ell_i^{(2)}}{\partial \widetilde{\boldsymbol{w}}_j \partial \widetilde{\boldsymbol{w}}_j^\top} \middle| g_{ij} = 1 \right] = -\sum_{i=1}^{n} g_{ij} v_{ij} \mathbf{w}_i \mathbf{w}_i^\top,$$

$$\frac{\partial^2 \ell_i^{(2)}}{\partial \widetilde{\boldsymbol{w}}_j \partial \widetilde{e}_j} = -\sum_{i=1}^{n} g_{ij} v_{ij} \left[\frac{y_{ij} - \mu_{ij}}{\mu_{ij}} + 1 \right] \mathbf{w}_i \quad \Rightarrow \quad E\left[-\frac{\partial^2 \ell_i^{(2)}}{\partial \widetilde{\boldsymbol{w}}_j \partial \widetilde{e}_j} \middle| g_{ij} = 1 \right] = \sum_{i=1}^{n} g_{ij} v_{ij} \mathbf{w}_i,$$

$$\frac{\partial^2 \ell_i^{(2)}}{\partial \widetilde{e}_j^2} = -\sum_{i=1}^{n} g_{ij} v_{ij} \left[\frac{y_{ij} - \mu_{ij}}{\mu_{ij}} + 1 \right] \quad \Rightarrow \quad E\left[-\frac{\partial^2 \ell_i^{(2)}}{\partial \widetilde{e}_j^2} \middle| g_{ij} = 1 \right] = \sum_{i=1}^{n} g_{ij} v_{ij}.$$

Therefore, the updating equations for $\widetilde{\boldsymbol{\theta}}$ based on the Fisher scoring algorithm are

$$\widetilde{\boldsymbol{\theta}}_j^{(t+1)} = \widetilde{\boldsymbol{\theta}}_j^{(t)} + S^{-1}_{\widetilde{\boldsymbol{\theta}}_j^{(t)}} U_{\widetilde{\boldsymbol{\theta}}_j^{(t)}}, \quad j = 1, \ldots, n, \tag{30}$$

$$\widetilde{\boldsymbol{\theta}}_j^{(t,k+1)} = \widetilde{\boldsymbol{\theta}}_j^{(t,k)} + S^{-1}_{\widetilde{\boldsymbol{\theta}}_j^{(t,k)}} U_{\widetilde{\boldsymbol{\theta}}_j^{(t,k)}}, \quad k = 1, \ldots, E. \tag{31}$$

The updating Equation (30) is in the outer loop of iterations and the iterations in (31) are in the inner loop of epochs for the same j and t. The quantities involved are

$$\widetilde{\boldsymbol{\theta}}_j^{(t)} = \left(\widetilde{\boldsymbol{w}}_j^{\top(t)}, \widetilde{b}_j^{(t)}, \widetilde{e}_j^{(t)} \right)^\top; \quad U_{\widetilde{\boldsymbol{\theta}}_j^{(t)}} = \left[\begin{array}{c} \frac{\partial \widetilde{\ell}_j}{\partial \widetilde{\boldsymbol{w}}_j} \\ \frac{\partial \widetilde{\ell}_j}{\partial \widetilde{b}_j} \\ \frac{\partial \widetilde{\ell}_j}{\partial \widetilde{e}_j} \end{array} \right] \Bigg|_{\widetilde{\boldsymbol{\theta}}_j^{(t)}}; \quad S_{\widetilde{\boldsymbol{\theta}}_j^{(t)}} = \left[\begin{array}{ccc} S^{(t)}_{\widetilde{\boldsymbol{w}}_j} & S^{(t)}_{\widetilde{\boldsymbol{w}}_j \widetilde{b}_j} & S^{(t)}_{\widetilde{\boldsymbol{w}}_j \widetilde{e}_j} \\ S^{(t)}_{\widetilde{b}_j \widetilde{\boldsymbol{w}}_j} & S^{(t)}_{\widetilde{b}_j} & S^{(t)}_{\widetilde{b}_j \widetilde{e}_j} \\ S^{(t)}_{\widetilde{e}_j \widetilde{\boldsymbol{w}}_j} & S^{(t)}_{\widetilde{e}_j \widetilde{b}_j} & S^{(t)}_{\widetilde{e}_j} \end{array} \right]$$

with

$$\frac{\partial \widetilde{\ell}_j}{\partial \widetilde{\boldsymbol{w}}_j} \Bigg|_{\widetilde{\boldsymbol{\theta}}_j^{(t)}} = \sum_{k=1}^{2} \frac{\partial \widetilde{\ell}_j^{(k)}(\widetilde{\boldsymbol{\theta}}_j^{(t)}; \boldsymbol{\theta}^{(c)})}{\partial \widetilde{\boldsymbol{w}}_j}, \qquad \frac{\partial \widetilde{\ell}_j}{\partial \widetilde{b}_j} \Bigg|_{\widetilde{\boldsymbol{\theta}}_j^{(t)}} = \frac{\partial \widetilde{\ell}_j^{(1)}(\widetilde{\boldsymbol{\theta}}_j^{(t)}; \boldsymbol{\theta}^{(c)})}{\partial \widetilde{b}_j}, \qquad \frac{\partial \widetilde{\ell}_j}{\partial \widetilde{e}_j} \Bigg|_{\widetilde{\boldsymbol{\theta}}_j^{(t)}} = \frac{\partial \widetilde{\ell}_j^{(2)}(\widetilde{\boldsymbol{\theta}}_j^{(t)}; \boldsymbol{\theta}^{(c)})}{\partial \widetilde{e}_j},$$

where these partial derivatives are derivatives $\frac{\partial \widetilde{\ell}_j^{(1)}}{\partial \widetilde{\boldsymbol{w}}_j}$, $\frac{\partial \widetilde{\ell}_j^{(2)}}{\partial \widetilde{\boldsymbol{w}}_j}$, $\frac{\partial \widetilde{\ell}_j^{(1)}}{\partial \widetilde{b}_j}$, $\frac{\partial \widetilde{\ell}_j^{(2)}}{\partial \widetilde{e}_j}$ and

$$S_{\widetilde{\boldsymbol{w}}_j}^{(t)} = E\left[-\frac{\partial^2 \widetilde{\ell}_j^{(1)}(\widetilde{\boldsymbol{\theta}}_j^{(t)}; \boldsymbol{\theta}^{(c)})}{\partial \widetilde{\boldsymbol{w}}_j \partial \widetilde{\boldsymbol{w}}_j^\top}\right] + E\left[-\frac{\partial^2 \widetilde{\ell}_j^{(2)}(\widetilde{\boldsymbol{\theta}}_j^{(t)}; \boldsymbol{\theta}^{(c)})}{\partial \widetilde{\boldsymbol{w}}_j \partial \widetilde{\boldsymbol{w}}_j^\top}\middle| g_{ij} = 1\right];$$

$$S_{\widetilde{\boldsymbol{w}}_j \widetilde{b}_j}^{(t)} = E\left[-\frac{\partial^2 \widetilde{\ell}_j^{(1)}(\widetilde{\boldsymbol{\theta}}_j^{(t)}; \boldsymbol{\theta}^{(c)})}{\partial \widetilde{\boldsymbol{w}}_j \partial \widetilde{b}_j}\right]; \quad S_{\widetilde{b}_j}^{(t)} = E\left[-\frac{\partial^2 \widetilde{\ell}_j^{(1)}(\widetilde{\boldsymbol{\theta}}_j^{(t)}; \boldsymbol{\theta}^{(c)},)}{\partial \widetilde{b}_j^2}\right]; \quad S_{\widetilde{b}_j \widetilde{e}_j}^{(t)} = 0;$$

$$S_{\widetilde{\boldsymbol{w}}_j \widetilde{e}_j}^{(t)} = E\left[-\frac{\partial^2 \widetilde{\ell}_j^{(2)}(\widetilde{\boldsymbol{\theta}}_j^{(t)}; \boldsymbol{\theta}^{(c)})}{\partial \widetilde{\boldsymbol{w}}_j \partial \widetilde{e}_j}\middle| g_{ij} = 1\right]; \quad S_{\widetilde{e}_j}^{(t)} = E\left[-\frac{\partial^2 \widetilde{\ell}_j^{(2)}(\widetilde{\boldsymbol{\theta}}_j^{(t)}; \boldsymbol{\theta}^{(c)})}{\partial \widetilde{e}_j^2}\middle| g_{ij} = 1\right],$$

evaluated at $\widetilde{\boldsymbol{\theta}}^{(t)}$ and $\boldsymbol{\theta}^{(c)}$ during the outer loop of iterations and at $\widetilde{\boldsymbol{\theta}}^{(t,k)}$ and $\boldsymbol{\theta}^{(c)}$ during the inner loop of iterations.

5. Convergence Analysis

In this section, we discuss the convergence behavior of the algorithm analytically. Our algorithm contains two components: a logistic regression part and a Gamma regression part. If these two components' parameter estimations were independent of each other, then it is the situation of the standard Generalized Linear Model (GLM) in each part. In our model, the two components' estimation cannot be separated but the two components in the log likelihood are additive. Hence, the convergence behavior in one component standard GLM case is still relevant. In this section, we first talk about the convergence behavior in the standard case, as this case applies to the situation when we hold the $\widetilde{\boldsymbol{\theta}}$ fixed while estimating $\boldsymbol{\theta}$ or vice versa.

In the standard GLM setting, most of the parameter estimation converges pretty fast. There are also abnormal behaviors that could happen such as when the estimated model component gets out of the valid range of the distribution. For example, in the Gamma regression, if the canonical link (negative inverse) is used, the estimated mean may become negative every now and then even though the distribution requires a positive mean. Refs. [22–24] presented conditions for the existence of MLE in logistic regression models. They proved in the non-trivial case that if there is overlap in the convex cones generated by the covariate values from different classes, the maximum likelihood estimates of the regression parameters exist and are unique. On the other hand, if the convex cones generated by the covariates in different classes have complete separation or quasi complete separation, then the maximum likelihood estimates do not exist or are unbounded. Generally, they recommended inserting a stopping rule if complete separation is found or restarting the iterative algorithm with standardized observations (to have mean 0 and variance 1) if quasi complete separation is found. Ref. [24] also recommended a procedure to check the conditions. For quasi complete separation, the estimation process diverges at least at some points. This makes the estimated probability of belonging to the correct class grow to one. Therefore, ref. [24] recommended checking the maximum predicted probability $p^{(t)}$ for each data point at the t^{th} iteration. If the maximum probability is close to 1 and is bigger than previous iterations' maximum probability, there are two possibilities for which this could happen. They suggested initially printing a warning but continue the iteration because the data point is likely to be an outlier observation in its own class and there is overlap in the two convex cones. In this case, the MLE exists and is unique so the algorithm should be allowed to continue. The other possibility is that there is quasi complete separation in the data. In this case, the process should be stopped and rerun with the observation vectors standardized with zero mean and unit variance.

Ref. [24] also stated that the difficulties associated with complete and quasi complete separation are small sample problems. With large sample size, the probability of observing a

set of separated data points is close to zero. Complete separation may occur with any type of data but it is unlikely that quasi complete separation will occur with truly continuous data.

Ref. [25] illustrated the non-convergence problem with a Poisson regression example. The author proposed a simple solution and implemented it in the R *glm*2 package, version 1.2.1. Specifically, if the iteratively reweighted least squares (IRLS) procedure produces either an infinite deviance or predicted values which fall within an invalid range, then the amount of update of the parameter estimates is repeatedly halved until the update no longer shows the behavior. Moreover, it produced a further step-halving, which checks that the updated deviance is making a reduction compared to that in the previous iteration. If it did not show the reduction, it triggers the step-halving to make the algorithm monotonically reduce the deviance.

Based on the aforementioned studies illustrating the standard case of GLM convergence behavior, we analytically present the convergence behavior of our alternating ZIG regression. For further discussion, we first specify our convex cones. Recall that $\widetilde{\boldsymbol{\theta}}$ serves as data when we estimate $\boldsymbol{\theta}_i$. In our context, the convex cones are defined through $\widetilde{\boldsymbol{\theta}}$, while we estimate $\boldsymbol{\theta}_i$ and through $\boldsymbol{\theta}$ while we estimate $\widetilde{\boldsymbol{\theta}}_j$. That is, for estimating $\boldsymbol{\theta}_i$, the convex cones are

$$G_{\widetilde{\boldsymbol{\theta}}}^{(i)} = \left\{ \sum_{\substack{j=1 \\ y_{ij}>0}}^{n} k_j \widetilde{\boldsymbol{\theta}}_j \mid k_j > 0 \right\}, \quad F_{\widetilde{\boldsymbol{\theta}}}^{(i)} = \left\{ \sum_{\substack{j=1 \\ y_{ij}=0}}^{n} k_j \widetilde{\boldsymbol{\theta}}_j \mid k_j > 0 \right\}. \tag{32}$$

For estimating the $\widetilde{\boldsymbol{\theta}}_j$, the convex cones are denoted as $G_{\boldsymbol{\theta}}^{(i)}$ and $F_{\boldsymbol{\theta}}^{(i)}$, respectively.

Firstly, we consider the case that either complete separation or quasi complete separation exists in the data. We discuss the estimation of $\boldsymbol{\theta}_i$ while holding $\widetilde{\boldsymbol{\theta}}$ fixed.

Suppose there exists complete separation or quasi separation in $G_{\widetilde{\boldsymbol{\theta}}}^{(i)}$ and $F_{\widetilde{\boldsymbol{\theta}}}^{(i)}$. That is, $G_{\widetilde{\boldsymbol{\theta}}}^{(i)} \cap F_{\widetilde{\boldsymbol{\theta}}}^{(i)} = \varnothing$, $G_{\widetilde{\boldsymbol{\theta}}}^{(i)} \neq R^{d+2}$, and $F_{\widetilde{\boldsymbol{\theta}}}^{(i)} \neq R^{d+2}$. Also, suppose $G_{\widetilde{\boldsymbol{\theta}}}^{(i)} \neq \varnothing$ and $F_{\widetilde{\boldsymbol{\theta}}}^{(i)} \neq \varnothing$. Then, there exists a vector direction $\mathbf{c}_i$ such that $\widetilde{\boldsymbol{w}}_j^\top \mathbf{c}_i \geq 0$ for $y_{ij} > 0$ and $\widetilde{\boldsymbol{w}}_j^\top \mathbf{c}_i \leq 0$ for $y_{ij} = 0$. (Note: This vector can be taken to be the perpendicular direction to a vector that lies in between $G_{\widetilde{\boldsymbol{\theta}}}^{(i)}$ and $F_{\widetilde{\boldsymbol{\theta}}}^{(i)}$ but does not belong to either $G_{\widetilde{\boldsymbol{\theta}}}^{(i)}$ or $F_{\widetilde{\boldsymbol{\theta}}}^{(i)}$.) Let $\ell_i^{(1)}(k)$ be the log likelihood of the Bernoulli part when the model parameters are updated toward direction $\mathbf{c}_i$ by k unit. That is, the original log likelihood $\ell_i^{(1)}$ and the updated $\ell_i^{(1)}(k)$ are as follows:

$$\begin{aligned}
\ell_i^{(1)} &= \sum_{j=1}^{n} \left\{ I(y_{ij} = 0) \cdot \log(1 - p_{ij}) + I(y_{ij} > 0) \cdot \log p_{ij} \right\} \\
&\text{where } p_{ij} = \frac{\exp(\eta_{ij})}{1 + \exp(\eta_{ij})}, \quad \eta_{ij} = \mathbf{w}_i^\top \widetilde{\boldsymbol{w}}_j + b_i + \widetilde{b}_j \\
\ell_i^{(1)}(k) &= \sum_{j=1}^{n} \left\{ I(y_{ij} = 0) \cdot \log(1 - p_{ij}(k)) + I(y_{ij} > 0) \cdot \log p_{ij}(k) \right\} \\
&\text{where } p_{ij}(k) = \frac{\exp\left(\eta_{ij} + k\widetilde{\boldsymbol{w}}_j^\top \mathbf{c}_i\right)}{1 + \exp\left(\eta_{ij} + k\widetilde{\boldsymbol{w}}_j^\top \mathbf{c}_i\right)}, \quad \eta_{ij} = \mathbf{w}_i^\top \widetilde{\boldsymbol{w}}_j + b_i + \widetilde{b}_j
\end{aligned} \tag{33}$$

- For $y_{ij} > 0, \widetilde{\boldsymbol{w}}_j^\top \mathbf{c}_i \geq 0$. Hence, as k increases to ∞, $p_{ij}(k)$ increases toward 1. This implies $\sum_{j=1}^{n} I(y_{ij} > 0) \cdot \log p_{ij}(k)$ increases toward 0.
- For $y_{ij} = 0$, $\widetilde{\boldsymbol{w}}_j^\top \mathbf{c}_i \leq 0$. Hence, as k increases to ∞, $p_{ij}(k)$ decreases toward 0. This implies $\sum_{j=1}^{n} I(y_{ij} = 0) \cdot \log(1 - p_{ij}(k))$ increases toward 0.

Putting the two pieces together, we know that as k increases, $\ell_i^{(1)}(k)$ increases for any given $\mathbf{w}_i, b_i$. Therefore, the maximum cannot be reached until k is ∞. This means the

MLE does not exist or the solution set $\{\hat{\boldsymbol{\theta}}_i\}$ is unbounded for the current value of $\widetilde{\boldsymbol{\theta}}$. This perspective can be also seen by looking at the partial derivative of $\ell_i^{(1)}(k)$ with respect to k. Note that

$$\frac{\partial \ell_i^{(1)}(k)}{\partial k} = \sum_{j=1}^{n} \Big\{ I\big(y_{ij}=0\big) \cdot \big(-p_{ij}(k)\big) \widetilde{\boldsymbol{w}}_j^{\top} \mathbf{c}_i \Big\} + \sum_{j=1}^{n} \Big\{ I\big(y_{ij}>0\big) \big(1-p_{ij}(k)\big) \widetilde{\boldsymbol{w}}_j^{\top} \mathbf{c}_i \Big\}. \tag{34}$$

Since $\widetilde{\boldsymbol{w}}_j^{\top} \mathbf{c}_i \leq 0$ when $y_{ij}=0$, we know the first term is non-negative. Similarly, $\widetilde{\boldsymbol{w}}_j^{\top} \mathbf{c}_i \geq 0$ when $y_{ij}>0$ implies the second term is non-negative. Therefore, the partial derivative of $\ell_i^{(1)}(k)$ is non-negative. This indicates that the gradient of $\ell_i^{(1)}$ is positive unless $\widetilde{\boldsymbol{w}}_j^{\top} \mathbf{c}_i = 0$. Therefore, there is no solution for $\ell_i^{(1)} = 0$ except the trivial solution $\mathbf{c}_i = 0$ in binomial regression alone. Of course, this is not exactly our case because we still have the Gamma regression component to be considered together.

Now consider the $\ell_i^{(2)}$ component with its first- and second-order derivatives with respect to k.

$$\ell_i^{(2)}(k) = \sum_{j=1}^{n} I(y_{ij}>0) \left\{ o_{ij} + v_{ij} \left[\log \frac{y_{ij}}{\exp\Big(\tau_{ij} + k\widetilde{\boldsymbol{w}}_j^{\top} \mathbf{c}_i\Big)} \right] - v_{ij} \frac{y_{ij}}{\exp\Big(\tau_{ij} + k\widetilde{\boldsymbol{w}}_j^{\top} \mathbf{c}_i\Big)} \right\},$$

where $o_{ij} = -\Gamma\big(v_{ij}\big) + v_{ij} \log v_{ij} - \log y_{ij}$. Note that

$$\frac{\partial \ell_i^{(2)}(k)}{\partial k} = \sum_{j=1}^{n} I(y_{ij}>0) v_{ij} \widetilde{\boldsymbol{w}}_j^{\top} \mathbf{c}_i \left[\frac{y_{ij}}{\exp\Big(\tau_{ij} + k\widetilde{\boldsymbol{w}}_j^{\top} \mathbf{c}_i\Big)} - 1 \right] = \sum_{j=1}^{n} I(y_{ij}>0) \widetilde{\boldsymbol{w}}_j^{\top} \mathbf{c}_i \sqrt{v_{ij}} u_{ij}(k),$$

where $u_{ij}(k) = \frac{y_{ij} - \exp\left(\tau_{ij} + k\widetilde{\boldsymbol{w}}_j^{\top} \mathbf{c}_i\right)}{\exp\left(\tau_{ij} + k\widetilde{\boldsymbol{w}}_j^{\top} \mathbf{c}_i\right) / \sqrt{v_{ij}}}$ is the standardized Gamma random variable that has mean 0 and variance 1 because the term $\exp\Big(\tau_{ij} + k\widetilde{\boldsymbol{w}}_j^{\top} \mathbf{c}_i\Big)$ is the mean of y_{ij} and v_{ij} is the shape parameter. Further,

$$\frac{\partial^2 \ell_i^{(2)}(k)}{\partial k^2} = -\sum_{j=1}^{n} I(y_{ij}>0) v_{ij} \Big(\widetilde{\boldsymbol{w}}_j^{\top} \mathbf{c}_i\Big)^2 \cdot \frac{y_{ij}}{\exp\Big(\tau_{ij} + k\widetilde{\boldsymbol{w}}_j^{\top} \mathbf{c}_i\Big)} < 0.$$

The negativity of the second-order derivative implies that $\ell_i^{(2)}(k)$ is concave.

Gathering the partial derivatives from both $\ell_i^{(1)}$ and $\ell_i^{(2)}$ together, we obtain the partial derivative of ℓ_i

$$\begin{aligned} \frac{\partial \ell_i(k)}{\partial k} &= \frac{\partial \ell_i^{(1)}(k)}{\partial k} + \frac{\partial \ell_i^{(2)}(k)}{\partial k} \\ &= \sum_{j=1}^{n} -I\Big(y_{ij}=0\Big) p_{ij}(k) \widetilde{\boldsymbol{w}}_j^{\top} \mathbf{c}_i + \sum_{j=1}^{n} I\Big(y_{ij}>0\Big) \big(1-p_{ij}(k)\big) \widetilde{\boldsymbol{w}}_j^{\top} \mathbf{c}_i + \sum_{j=1}^{n} I\Big(y_{ij}>0\Big) \sqrt{v_{ij}} u_{ij}(k) \widetilde{\boldsymbol{w}}_j^{\top} \mathbf{c}_i. \end{aligned} \tag{35}$$

From the above discussion, the first two terms in (35) corresponding to $\frac{\partial \ell_i^{(1)}(k)}{\partial k}$ are greater than 0. The $u_{ij}(k)$ is centered at zero and has variance 1. Therefore, in order for the MLE to exist, the sum of $\sum_{i=1}^{n} I\big(y_{ij}>0\big) \sqrt{v_{ij}} u_{ij}(k) \widetilde{\boldsymbol{w}}_j^{\top} \mathbf{c}_i I(u_{ij}(k) < 0)$ must cancel the total value of $\frac{\partial \ell_i^{(1)}(k)}{\partial k}$ and the positive term $\sum_{i=1}^{n} I\big(y_{ij}>0\big) \sqrt{v_{ij}} u_{ij}(k) \widetilde{\boldsymbol{w}}_j^{\top} \mathbf{c}_i I(u_{ij}(k) > 0)$. If the shape parameter v_{ij} of Gamma distribution is small, the distribution is highly skewed with a long right tail. In this case, there are more observations having values less than its mean. However, a small shape parameter also makes $\sqrt{v_{ij}} u_{ij}(k)$ small such that the $\frac{\partial \ell_i^{(1)}(k)}{\partial k}$

may dominate, and hence, the entire partial derivative $\frac{\partial \ell_i(k)}{\partial k}$ is greater than zero. When the shape parameter is large, the Gamma random variables are approximately normally distributed. In this case, the observations are symmetrically located on either side of its mean. This makes the number of observations satisfying $u_{ij} < 0$ and $u_{ij} > 0$ roughly equal. Consequently, the $\frac{\partial \ell_i^{(2)}(k)}{\partial k}$ is close to zero. Then, there are no extra values left to neutralize $\frac{\partial \ell_i^{(1)}(k)}{\partial k}$. As a result, regardless of whether the shape parameter v_{ij} is large or small, when complete separation or quasi complete separation holds, it is highly likely that the MLE does not exist. When the shape parameter v_{ij} is intermediate such that the Gamma distribution is still skewed, the more values of Gamma observations less than its mean might allow $\sum_{i=1}^{n} I(y_{ij}>0)\sqrt{v_{ij}}u_{ij}(k)\widetilde{\boldsymbol{w}}_j^\top \mathbf{c}_i I(u_{ij}(k) < 0)$ to cancel all other positive terms. This is the case that there could be a solution for $\widetilde{\boldsymbol{\theta}}_i$.

Next, consider the case that there is overlap in the two convex cones $G_{\widetilde{\boldsymbol{\theta}}}^{(i)}$ and $F_{\widetilde{\boldsymbol{\theta}}}^{(i)}$ (i.e., there is neither complete separation nor quasi complete separation in the data). Recall that the ZIG model is a two-part model with a Bernoulli part and a Gamma part. The log likelihood function $\ell_i^{(1)}$ corresponding to the Bernoulli part can be shown to be strictly concave in $\boldsymbol{\theta}_i = (\mathbf{w}_i^\top, b_i)^\top$. This is because the first component $g_{ij}\eta_{ij} = g_{ij}(\mathbf{w}_i^\top \widetilde{\boldsymbol{w}}_j + b_i + \widetilde{b}_j)$ in $\ell_i^{(1)}$ is an affine function of $\boldsymbol{\theta}_i$ (see Equation (7) for the expression of $\ell_i^{(1)}$), which is both convex and concave, if we hold $\widetilde{\boldsymbol{w}}_j$ and $\widetilde{b}_j$ fixed. The second component $-\log\{1+\exp(\eta_{ij})\}$ is strictly concave, as its second derivative is less than 0, as shown below.

$$\frac{\partial^2}{\partial x^2}[-\log\{1+\exp(x)\}] = -\frac{\exp(x)}{[1+\exp(x)]^2} < 0.$$

Thus, the log likelihood corresponding to the Bernoulli part $\ell_i^{(1)}$ is strictly concave.

Now, consider the log likelihood corresponding to the Gamma part $\ell_i^{(2)}$ in (24). We only need to consider the summation over the last two terms $-v_{ij}\tau_{ij}$ and $-v_{ij}y_{ij}\exp(-\tau_{ij})$ because the other terms do not involve the regression parameters, where $\tau_{ij} = \mathbf{w}_i^\top \widetilde{\boldsymbol{w}}_j + e_i + \widetilde{e}_j$. Note that $-\tau_{ij}$ is an affine function of $\boldsymbol{\theta}_i$, which is both convex and concave and $\exp(g(x))$ is convex if $g(x)$ is convex (see ref. [26]). This leads to the term $-v_{ij}y_{ij}\exp(-\tau_{ij})$ being strictly concave in $\boldsymbol{\theta}_i$. Hence, we have the $\ell_i^{(2)}$ strictly concave in $\boldsymbol{\theta}_i = (\mathbf{w}_i^\top, b_i, e_i)^\top$. Combining the two concave components $\ell_i^{(1)}$ and $\ell_i^{(2)}$, we know that the entire log likelihood for the i^{th} row ℓ_i is a strictly concave function with respect to the parameter being estimated for any row i. Additionally, there is overlap in the two convex cones. Therefore, for any direction in the overlapping area of the convex cones, updating the parameter along that direction will lead to the two components of $\frac{\partial \ell_i^{(1)}(k)}{\partial k}$ in Formula (34) being of opposite sign to each other. In this case, $-\ell_i$ has a unique minimum when there is neither complete separation nor quasi complete separation in $G_{\widetilde{\boldsymbol{\theta}}}^{(i)}$ and $F_{\widetilde{\boldsymbol{\theta}}}^{(i)}$. Similar arguments apply when we estimate $\widetilde{\boldsymbol{\theta}}_j$ while holding $\boldsymbol{\theta}$ fixed. That is, a unique MLE of $\widetilde{\boldsymbol{\theta}}_j$ exists when there is neither complete separation nor quasi complete separation in $G_{\boldsymbol{\theta}}^{(j)}$ and $F_{\boldsymbol{\theta}}^{(j)}$. This means that the alternating procedure will find the MLE of $\widetilde{\boldsymbol{\theta}}_j$ when $\boldsymbol{\theta}$ is fixed and will find the MLE of $\boldsymbol{\theta}_i$ when $\widetilde{\boldsymbol{\theta}}$ is fixed in this non-separation scenario.

Next, we consider the convergence behavior for the estimation of $\boldsymbol{\theta}$ without fixing the value of $\widetilde{\boldsymbol{\theta}}$. For estimating $\boldsymbol{\theta}$, consider the complete data $\mathbf{x} = (\mathbf{y}, \widetilde{\boldsymbol{\theta}})$. The entire $\widetilde{\boldsymbol{\theta}}$ matrix is missing. Due to too many missing values, a logical approach is to regard the $\widetilde{\boldsymbol{\theta}}$ as randomly drawn from a distribution which has relatively few parameters. Assume the distribution of $\mathbf{x}$ is in the exponential family.

Let $\tilde{f}(\mathbf{x}|\boldsymbol{\theta})$ be the unconditional density of the complete data $\mathbf{x} = (\mathbf{y}, \widetilde{\boldsymbol{\theta}})$ and $K(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta})$ be the conditional density given $\mathbf{y}$. Denote the marginal density of $\mathbf{y}$ given $\boldsymbol{\theta}$ as $g(\mathbf{y}|\boldsymbol{\theta})$. In

the next few paragraphs, we explain that the alternating updates in ZIG leads ultimately to a value of $\boldsymbol{\theta}$ that maximizes $\check{\ell}(\boldsymbol{\theta}) = \log g(\mathbf{y}|\boldsymbol{\theta})$.

For exponential families, the unconditional density $\tilde{f}(\mathbf{x}|\boldsymbol{\theta})$ and conditional density $K(\mathbf{x}|\mathbf{y},\boldsymbol{\theta}) = \tilde{f}(\mathbf{x}|\boldsymbol{\theta})/g(\mathbf{y}|\boldsymbol{\theta})$ both have the same natural parameter $\boldsymbol{\theta}$ and the same sufficient statistic $\mathbf{t}(\mathbf{x})$ except that they are defined over different sample spaces $\Omega(\mathbf{x})$ versus $\Omega(\mathbf{y})$. We can write $\tilde{f}(\mathbf{x}|\boldsymbol{\theta})$ and $K(\mathbf{x}|\mathbf{y},\boldsymbol{\theta})$ in general exponential family format as

$$\begin{aligned}\tilde{f}(\mathbf{x}|\boldsymbol{\theta}) &= \exp(\boldsymbol{\theta}\mathbf{t}(\mathbf{x})^T + c(\mathbf{x}))/a(\boldsymbol{\theta}),\\ K(\mathbf{x}|\mathbf{y},\boldsymbol{\theta}) &= \exp(\boldsymbol{\theta}\mathbf{t}(\mathbf{x})^T + c(\mathbf{x}))/b(\boldsymbol{\theta}|\mathbf{y}),\end{aligned}$$

where

$$a(\boldsymbol{\theta}) = \int_{\Omega(\mathbf{x})} \exp(\boldsymbol{\theta}\mathbf{t}(\mathbf{x})^T + c(\mathbf{x}))d\mathbf{x}, \quad b(\boldsymbol{\theta}|\mathbf{y}) = \int_{\Omega(\mathbf{y})} \exp(\boldsymbol{\theta}\mathbf{t}(\mathbf{x})^T + c(\mathbf{x}))d\mathbf{x}.$$

Then, $\check{\ell}(\boldsymbol{\theta}) = -\log a(\boldsymbol{\theta}) + \log b(\boldsymbol{\theta}|\mathbf{y})$. The first- and second-order derivatives of $\check{\ell}(\boldsymbol{\theta})$ are

$$\begin{aligned}\frac{\partial \check{\ell}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} &= -\frac{\partial \log a(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} + \frac{\partial \log b(\boldsymbol{\theta}|\mathbf{y})}{\partial \boldsymbol{\theta}} = E(\mathbf{t}(\mathbf{x})|\mathbf{y},\boldsymbol{\theta}) - E(\mathbf{t}(\mathbf{x})|\boldsymbol{\theta}),\\ \frac{\partial^2 \check{\ell}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}\partial \boldsymbol{\theta}^T} &= -\frac{\partial^2 \log a(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}\partial \boldsymbol{\theta}^T} + \frac{\partial^2 \log b(\boldsymbol{\theta}|\mathbf{y})}{\partial \boldsymbol{\theta}\partial \boldsymbol{\theta}^T} = E[\mathrm{Var}(\mathbf{t}(\mathbf{x})|\mathbf{y},\boldsymbol{\theta})|\boldsymbol{\theta}] - \mathrm{Var}(\mathbf{t}(\mathbf{x})|\boldsymbol{\theta}),\end{aligned}$$

where $E(\mathbf{t}(\mathbf{x})|\boldsymbol{\theta}])$ and $\mathrm{Var}(\mathbf{t}(\mathbf{x})|\boldsymbol{\theta})$ are the expectation and variance under the complete data likelihood from $\mathbf{y}$ and $\widetilde{\boldsymbol{\theta}}$. And $E(\mathbf{t}(\mathbf{x})|\mathbf{y},\boldsymbol{\theta}])$ and $\mathrm{Var}(\mathbf{t}(\mathbf{x})|\mathbf{y},\boldsymbol{\theta})$ are the conditional expectation and variance of the sufficient statistics. $E[\mathrm{Var}(\mathbf{t}(\mathbf{x})|\mathbf{y},\boldsymbol{\theta})|\boldsymbol{\theta}]$ is the expected value of the conditional covariance matrix when $\mathbf{y}$ has sampling density $g(\mathbf{y}|\boldsymbol{\theta})$. The last equality of both equations assumes the order of expectation and derivative can be exchanged. The previous equation means the derivative of the log likelihood is the difference between the conditional and unconditional expectation of the sufficient statistics.

Meanwhile, the updating equation based on the Fisher scoring algorithm can be written as

$$\begin{aligned}\widehat{\boldsymbol{\theta}}^{(k+1)} &= \widehat{\boldsymbol{\theta}}^{(k)} + \left\{\mathrm{Var}(\mathbf{t}(\mathbf{x})|\widehat{\boldsymbol{\theta}}^{(k)}) - E[\mathrm{Var}(\mathbf{t}(\mathbf{x})|\mathbf{y},\widehat{\boldsymbol{\theta}}^{(k)})|\widehat{\boldsymbol{\theta}}^{(k)}]\right\}^{-1} \frac{\partial \check{\ell}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\bigg|^T_{\boldsymbol{\theta}=\widehat{\boldsymbol{\theta}}^{(k)}}\\ &= \widehat{\boldsymbol{\theta}}^{(k)} + \mathrm{Var}^{-1}(E(\mathbf{t}(\mathbf{x})|\mathbf{y},\widehat{\boldsymbol{\theta}}^{(k)})|\widehat{\boldsymbol{\theta}}^{(k)})[E(\mathbf{t}(\mathbf{x})|\mathbf{y},\widehat{\boldsymbol{\theta}}^{(k)}) - E(\mathbf{t}(\mathbf{x})|\widehat{\boldsymbol{\theta}}^{(k)}])],\end{aligned}$$

where in the limit, $\widehat{\boldsymbol{\theta}}^{(k+1)} = \widehat{\boldsymbol{\theta}}^{(k)} = \boldsymbol{\theta}^*$, for some $\boldsymbol{\theta}^*$, which leads to $E(\mathbf{t}(\mathbf{x})|\mathbf{y},\boldsymbol{\theta}^*) - E(\mathbf{t}(\mathbf{x})|\boldsymbol{\theta}^*)$ or $\partial\check{\ell}(\boldsymbol{\theta})/\partial\boldsymbol{\theta} = 0$ at $\boldsymbol{\theta}^*$.

The complete data log likelihood based on the joint distribution of $\mathbf{x} = (\mathbf{y},\widetilde{\boldsymbol{\theta}})$ can be written as $\log \tilde{f}(\mathbf{x}|\boldsymbol{\theta}) = \ell(\boldsymbol{\theta};\widetilde{\boldsymbol{\theta}}) + \log(P_{\widetilde{\boldsymbol{\theta}}}(\widetilde{\boldsymbol{\theta}}|\boldsymbol{\theta}))$, where $\ell(\boldsymbol{\theta};\widetilde{\boldsymbol{\theta}})$ is defined in the end of Section 2, and $P_{\widetilde{\boldsymbol{\theta}}}(\widetilde{\boldsymbol{\theta}}|\boldsymbol{\theta}))$ is the probability density function of $\widetilde{\boldsymbol{\theta}}$ given $\boldsymbol{\theta}$. We know $P_{\widetilde{\boldsymbol{\theta}}}(\widetilde{\boldsymbol{\theta}}|\boldsymbol{\theta})$ is also a member of the exponential family. Assume its sufficient statistics are linear in $\widetilde{\boldsymbol{\theta}}$. Given $\widetilde{\boldsymbol{\theta}}$, the observed data likelihood $\ell(\boldsymbol{\theta};\widetilde{\boldsymbol{\theta}})$ based on $\mathbf{y}$ contains the Bernoulli and Gamma parts, in which the sufficient statistic for $\boldsymbol{\theta}$ is linear in $\mathbf{y}$. Then, the sufficient statistics for the complete data problem are linear in the data $\mathbf{y}$ and $\widetilde{\boldsymbol{\theta}}$. In this case, calculating $E(\mathbf{t}(\mathbf{x})|\mathbf{y},\boldsymbol{\theta}^{(k)})$ is equivalent to a procedure which first fills in the individual data points for $\widetilde{\boldsymbol{\theta}}$ and then computes the sufficient statistics using filled-in values. With the filled-in value for $\widetilde{\boldsymbol{\theta}}$, the computation of the estimator for $\boldsymbol{\theta}$ follows the usual maximum likelihood principle. This results in iterative update of $\boldsymbol{\theta}$ and $\widetilde{\boldsymbol{\theta}}$ back and forth. Essentially, the problem is a transformation from an assumed parameter-vector to another parameter-vector that maximized the conditional expected likelihood.

One of the difficulties in the ZIG model is that the parameterization allows arbitrary orthogonal transformations on both $\mathbf{w}$ and $\widetilde{w}$ without affecting the value of the likelihood. Even for cases where the likelihood for the complete data $(\mathbf{y},\widetilde{\boldsymbol{\theta}})$ problem is concave, the

likelihood for the ZIG may not be concave. Consequently, multiple solutions of the likelihood equations can exist. An example is a ridge of solutions corresponding to orthogonal transformations of the parameters.

For complete data problems in exponential family models with canonical links, the Fisher scoring algorithm is equivalent to the Newton–Raphson algorithm, which has a quadratic rate of convergence. This advantage is due to the fact that the second derivative of the log-likelihood does not depend on the data. In these cases, the Fisher scoring algorithm has a quadratic rate of convergence when the starting values are near a maximum. However, we do not have the complete data $\mathbf{y}$ and $\widetilde{\boldsymbol{\theta}}$ when estimating $\boldsymbol{\theta}$. Fisher scoring algorithms often fail to have quadratic convergence in incomplete data problems, since the second derivative often does depend upon the data. Further, the scoring algorithm does not have the property of always increasing the likelihood. It could in some cases move toward a local maximum if the choice of starting values is poor.

In summary, we conclude that our alternating ZIG regression has an unique MLE in each side of the regressions when either $\boldsymbol{\theta}$ or $\widetilde{\boldsymbol{\theta}}$ stays fixed and the algorithm converges when there is overlap in data. The data refer to $\widetilde{\boldsymbol{\theta}}$ while we estimate $\boldsymbol{\theta}_i$ and refer to $\boldsymbol{\theta}$ while we estimate $\widetilde{\boldsymbol{\theta}}_j$. When there is overlap in data, both $\ell_i^{(1)}$ and $\ell_i^{(2)}$ are concave functions with unique maximum. However, when there is complete separation or quasi complete separation, the alternating ZIG regression will fail to converge with high chance. This is because the first-order partial derivative of $\ell_i^{(1)}$ is non-negative and $\ell_i^{(1)}$ increases with k. Even though the $\ell_i^{(2)}$ component is a well-behaved concave curve, the entire log likelihood, unfortunately, may not have MLE exist or the solution set may be unbounded especially when the Gamma observations have large or too small shape parameter. The overall convergence behavior for estimating $\boldsymbol{\theta}$ without holding $\widetilde{\boldsymbol{\theta}}$ fixed can treat $\widetilde{\boldsymbol{\theta}}$ as missing data. The alternating update ultimately finds the maximum likelihood estimate of $\boldsymbol{\theta}$ based on the sampling distribution of the observed matrix $\mathbf{y}$ if the solution is in the interior of the parameter space. This requires the joint distribution of $\mathbf{y}$ and $\widetilde{\boldsymbol{\theta}}$ to be in the exponential family with sufficient statistic that is linear in $\mathbf{y}$ and $\widetilde{\boldsymbol{\theta}}$.

6. Adjusting Parameter Update with Learning Rate

In the convergence analysis section, our discussion is based on holding either $\boldsymbol{\theta}$ or $\widetilde{\boldsymbol{\theta}}$ fixed while estimating the other one. The Fisher scoring algorithm is a modified version of Newton's method. In general, Newton's method solves $g'(x) = 0$ by numerical approximation. This algorithm starts with an initial value x_0 and computes a sequence of points via $x_{n+1} = x_n - g''(x_n)^{-1} g'(x_n)$. Newton's method converges fast because the distance between the estimate and its true value shrinks quickly such that the distance in the next step of iteration is asymptotically equivalent to the squared distance in the previous iteration. That is, Newton's method has quadratic convergence order (cf. [27] p. 29 and [28]). This convergence order holds when the initial value of the iteration is in the neighborhood of the true value and the third derivative $g'''(x)$ is continuous and $g''(x)$ is non-zero.

The Fisher scoring algorithm is slightly different from Newton's method. In the Fisher scoring algorithm, we replace the $-g''(x_n)$ by its expected value. This algorithm is asymptotically equivalent to Newton's method and therefore enjoys the same asymptotic property such as consistency of the estimate of the parameter. As sample size becomes large, the convergence order increases [28]. It has some advantages over Newton's method in that the expected value of the Hessian matrix is positive definite which guarantees the update is uphill toward the direction of maximizing the log likelihood function assuming that the model is correct and the covariates are true explanatory variables.

As the Fisher scoring algorithm assumes that $\widetilde{\boldsymbol{\theta}}$ is the true value when we estimate $\boldsymbol{\theta}$ or vice versa, there could be complications when the parameter being fixed is not equal to the true parameter value. To see this point, note that our updating equations are all written in the context of using the Fisher scoring algorithm, which relies on the expectation of the Hessian matrix using correct distribution at the true parameter value. In particular, the

algorithm uses $\widetilde{\boldsymbol{\theta}}$ as a fixed value while estimating $\boldsymbol{\theta}$. The resulting estimate of $\boldsymbol{\theta}$ determines the distribution because the distribution is a function of $\boldsymbol{\theta}$ and $\widetilde{\boldsymbol{\theta}}$. When the parameter is being fixed at a value far from its true value during the intermediate steps, the distribution is wrong even though it is in the right family. The consequence of using a wrong distribution to compute the expectation of the Hessian matrix could lead to a sequence of parameter updates that converges to a limiting value unequal to the true parameter [28]. In this case, the algorithm might diverge. Our simulation study in later section confirms this point.

To avoid this parameter update divergence problem, we introduce learning rate adjustment so that the change in parameter estimate is scaled by $lr/t^{1/4}$, where lr is a small constant learning rate such as 0.1 or 0.01, and t is the iteration number. That is, the general updating formula is $\boldsymbol{\theta}_i^{(t+1)} = \boldsymbol{\theta}_i^{(t)} + \frac{lr}{t^{1/4}} \cdot S^{-1}_{\boldsymbol{\theta}_i^{(t)}} \cdot U_{\boldsymbol{\theta}_i^{(t)}}$ and the adjustment is applied to both inner loop and outer loop iterations. The algorithm follows the same work flow as those listed in Algorithm 1 except that the epoch update in the inner loop (Algorithm 2) is replaced with the Algorithm 3 using the learning rate adjustment $\frac{lr}{t^{1/4}}$. How we decide to use this learning rate adjustment comes from modifying the popularly used adaptive moment estimation (Adam) and the stochastic gradient descent. In the stochastic gradient descent, the parameter update has learning rate adjustment to make small moves so that it compensates for the random nature of selecting only one observation to compute the gradient. Specifically, given parameters $w^{(t)}$ and a gradient function evaluated at one randomly selected observation $g^{(t)}$, the update is based on formula $w^{(t+1)} = w^{(t)} - lr \cdot a(t) g^{(t)}$, where $a(t)$ satisfies $\sum_{t=1}^{\infty} a(t) = \infty$ and $\sum_{t=1}^{\infty} a^2(t) < \infty$. The $a(t)$ corresponds to our $t^{-1/4} S^{-1}_{\boldsymbol{\theta}_i^{(t)}}$. However, our parameter update does not use just one $\widetilde{\boldsymbol{\theta}}$. Instead, all $\widetilde{\boldsymbol{\theta}}_1, \ldots, \widetilde{\boldsymbol{\theta}}_n$ were used in computing the gradient and the information matrix. Given parameters $w^{(t)}$ and a loss function $L^{(t)}$ at the t^{th} training iteration, the Adam update takes the form $w^{(t+1)} = w^{(t)} - lr \cdot \hat{m}_w / (\sqrt{\hat{v}_w} + \epsilon)$, where $\hat{m}_w$ and $\hat{v}_w$ are the exponential moving average of the gradients and the second moments of the gradients in the past iterations, respectively, and ϵ is a small scalar (e.g., 10^{-8}) used to prevent division by 0. Our use of the Fisher information $S^{-1}_{\boldsymbol{\theta}_i^{(t)}}$ should provide a better mechanism than Adam's exponential moving average of second moments to achieve the effect of increasing the learning rate for sparser parameters and decreasing the learning rate for ones that are less sparse. This is because Adam only uses the diagonal entries of $S_{\boldsymbol{\theta}_i^{(t)}}$ and ignores the covariance between the estimated parameters existing in the off-diagonal entries. Refs. [29,30] both pointed out that Adam may not converge to optimal solutions even for some simple convex problems, although it is overwhelmingly popular in machine learning applications.

Using the learning rate adjustment makes smaller steps in each update before changing directions. This learning rate adjustment turns out to be crucial. The simulation study in the next section examines the effect of the learning rate adjustment.

Algorithm 1 SA-ZIG regression

Require: $S^{-1}_{\boldsymbol{\theta}_i^{(t)}}$ and $S^{-1}_{\widetilde{\boldsymbol{\theta}}_j^{(t)}}$ exist.

$t \leftarrow 0$, overall $Loss^{(t)}$ = total negative log likelihood $\leftarrow \infty$
$converged = False$
while $t \leq maxit$ **do**
  while $converged = False$ **do**
    $i \leftarrow 1$
    while $i \leq n$ **do**
      do n_epoch update of $U_{\boldsymbol{\theta}_i^{(t)}}$ and $S_{\boldsymbol{\theta}_i^{(t)}}$ and $\boldsymbol{\theta}_i^{(t+1)}$ using i^{th} row of data. ▷ see Algorithm 2 or 3
      do n_epoch update of $U_{\widetilde{\boldsymbol{\theta}}_i^{(t)}}$ and $S_{\widetilde{\boldsymbol{\theta}}_i^{(t)}}$ and $\widetilde{\boldsymbol{\theta}}_i^{(t+1)}$ using i^{th} column of data.
      $i \leftarrow i + 1$

    for $k = 1, \ldots, n$ **do**
      retrieve kth row of data
      recompute $U_{\boldsymbol{\theta}_k^{(t+1)}}$, $U_{\widetilde{\boldsymbol{\theta}}_k^{(t+1)}}$ and their L_2 norms using $\boldsymbol{\theta}^{(t+1)}$ and $\widetilde{\boldsymbol{\theta}}^{(t+1)}$ values.
      recompute $loss_k$ and $\widetilde{loss}_k$ for k^{th} row and column respectively using $\boldsymbol{\theta}^{(t+1)}$ and $\widetilde{\boldsymbol{\theta}}^{(t+1)}$ values.
    compute the overall $Loss(\boldsymbol{\theta}^{(t+1)}, \widetilde{\boldsymbol{\theta}}^{(t+1)})$.
    check if the relative change in Loss is less than a predefined threshold ϵ.
  $t \leftarrow t + 1$

Algorithm 2 Epoch update in inner loop of Algorithm 1 without learning rate adjustment

for `epoch` $\in$ `1, ..., n_epoch` **do**
  compute $U_{\boldsymbol{\theta}_i^{(t)}}$ and $S_{\boldsymbol{\theta}_i^{(t)}}$.
  update $\boldsymbol{\theta}_i^{(t+1)}$ using current value of $\{\, \widetilde{\boldsymbol{\theta}}_j^{(t)}, j = 1, \ldots, n \,\}$ and $\boldsymbol{\theta}_i^{(t)}$ based on formulae in (29)
for `epoch` $\in$ `1, ..., n_epoch` **do**
  compute $U_{\widetilde{\boldsymbol{\theta}}_i^{(t)}}$ and $S_{\widetilde{\boldsymbol{\theta}}_i^{(t)}}$.
  update $\widetilde{\boldsymbol{\theta}}_i^{(t+1)}$ using current value of $\{\, \boldsymbol{\theta}_j^{(t)}, j = 1, \ldots, n \,\}$, and $\widetilde{\boldsymbol{\theta}}_i^{(t)}$ based on formulae in (31)

Algorithm 3 Updating equations in SA-ZIG regression with learning rate adjustment

for $i = 1, \ldots, n$ **do**
  for $k = 1, \ldots, E$ **do**
    $\boldsymbol{\theta}_i^{(t,k+1)} = \boldsymbol{\theta}_i^{(t,k)} + \frac{lr}{t^{1/4}} \cdot S^{-1}_{\boldsymbol{\theta}_i^{(t,k)}} \cdot U_{\boldsymbol{\theta}_i^{(t,k)}}$
    $\widetilde{\boldsymbol{\theta}}_i^{(t,k+1)} = \widetilde{\boldsymbol{\theta}}_i^{(t,k)} + \frac{lr}{t^{1/4}} \cdot S^{-1}_{\widetilde{\boldsymbol{\theta}}_i^{(t,k)}} \cdot U_{\widetilde{\boldsymbol{\theta}}_i^{(t,k)}}$
  $\boldsymbol{\theta}_i^{(t)} \leftarrow \boldsymbol{\theta}_i^{(t,E)}$
  $\widetilde{\boldsymbol{\theta}}_i^{(t)} \leftarrow \widetilde{\boldsymbol{\theta}}_i^{(t,E)}$
  $\boldsymbol{\theta}_i^{(t+1)} = \boldsymbol{\theta}_i^{(t)} + \frac{lr}{t^{1/4}} \cdot S^{-1}_{\boldsymbol{\theta}_i^{(t)}} \cdot U_{\boldsymbol{\theta}_i^{(t)}}$
  $\widetilde{\boldsymbol{\theta}}_i^{(t+1)} = \widetilde{\boldsymbol{\theta}}_i^{(t)} + \frac{lr}{t^{1/4}} \cdot S^{-1}_{\widetilde{\boldsymbol{\theta}}_i^{(t)}} \cdot U_{\widetilde{\boldsymbol{\theta}}_i^{(t)}}$

7. Numerical Studies

7.1. A Simulation Study with ZIG Using Log Link

In this section, we present a simulation study to assess the performance of the alternating ZIG regression. We found through some experiments that data generation has to be very careful because the mean of the Gamma distribution was given by $\mu_{ij} = \exp(\mathbf{w}_i^\top \widetilde{\boldsymbol{w}}_j + e_i + \widetilde{e}_j)$. When we randomly generate the $\mathbf{w}_i$'s and $\widetilde{\boldsymbol{w}}_j$'s independently from Uniform distribution with each of them having dimension 50, their dot product could easily become so large that exponentiated value μ_{ij} and the variance of the distribution ($\propto \mu_{ij}^2$) become infinity or undefined. Keeping this point in mind, we generate the data as follows:

- The w_{ik}'s were generated independently from Uniform $(-0.25, 0.25)$ with seed 99, $i = 1, \ldots, 300$ and $k = 1, \ldots, 50$. Set $\mathbf{w}_i = (w_{i1}, \ldots, w_{i50})'$.
- The $\widetilde{w}_{ik} = w_{ik}$, $i = 1, \ldots, 300$ and $k = 1, \ldots, 50$. That is, $\widetilde{\boldsymbol{w}}_i = \mathbf{w}_i$.
- The b_i's and $\widetilde{b}_i$'s were generated independently from Uniform (0, 0.05) with seed 97 and 96, respectively, $i = 1, \ldots, 300$.
- The e_i's and $\widetilde{e}_i$'s were generated independently from Uniform (0.1, 0.35) with seed 1 and 2, respectively, $i = 1, \ldots, 300$.
- The success probability p_{ij} of positive observations was calculated as $p_{ij} = (1 + \exp(-\eta_{ij}))^{-1}$, where η_{ij} is based on the logit Formula (2).
- Generate B_{ij} independently from the Bernoulli distribution with success probability p_{ij}, $i = 1, \ldots, 300$ and $j = i, \ldots, 300$.
- If $B_{ij} = 0$, set $Y_{ij} = 0$. Otherwise, generate Y_{ij} from the Gamma distribution with mean $\mu_{ij} = \exp(\tau_{ij})$ and shape parameter $\nu_{ij} = 4$, where τ_{ij} is computed based on Formula (4) for $i = 1, \ldots, 300$ and $j = 1, \ldots, 300$.

The matrix containing Y_{ij}, $i, j = 1, \ldots, 300$, is the data we used for the alternating ZIG regression. With the generated data matrix $\mathbf{Y}$, we applied our algorithm to the data with two different sets of initial values for the unknown parameters. One setting is to check whether the algorithm would perform well when the parameter estimation is less difficult when part of the parameters was initialized with true values. The other setting demands more accurate estimation in both $\mathbf{w}_i$ and $\widetilde{\boldsymbol{w}}_j$ in the right direction.

In Setting 1, we set all the parameters' initial values equal to their true values except for $\widetilde{w}_{ik}, i = 1, \ldots, 300, k = 1, \ldots, 50$. The initial values of $\widetilde{w}_{ik}$'s were randomly generated from Uniform $(-0.25, 0.25)$ with seed number 98. Even though initial values of $\mathbf{w}, b, \widetilde{b}, e$ and $\widetilde{e}$ were set to be equal to the true values, the algorithm was not aware of this and these parameters were still estimated along with $\widetilde{\boldsymbol{w}}$.

In Setting 2, we generated the initial values for all the parameters randomly as follows:

- $\mathbf{w}$'s initial values were generated independently from Uniform $(-0.25, 0.25)$ with dimension 300 by 50 and seed 102.
- $\widetilde{\boldsymbol{w}}$'s initial values were generated independently from Uniform $(-0.25, 0.25)$ with dimension 300 by 50 and seed 103.
- b_i's and $\widetilde{b}_j$'s initial values were generated independently from Uniform (0, 0.05) with seed 104 and 105, respectively, $i, j = 1, \ldots, 300$.
- e_i's and $\widetilde{e}_j$'s were independently generated from Uniform (0.1, 0.35) with seed 106 and 107, respectively, $i, j = 1, \ldots, 300$.

We consider update both with learning rate adjustment (Algorithm 3) and without learning rate adjustment (Algorithm 1).

The results of Setting 1 are presented in Table 1 and Figure 1. We can see in Figure 1 that even though the norm of the score vectors went up after initial reduction, the loss function (i.e., the negative log-likelihood) consistently decreases as the iteration number increases. In the last five iterations, the algorithm with no learning rate adjustment (Algorithm 1) reached a lower value in overall loss compared to that with learning rate adjustment (Algorithm 3). However, the norms of the score vectors from the algorithm with learning

rate adjustment are much smaller than those from the algorithm with no learning rate adjustment (see Table 1).

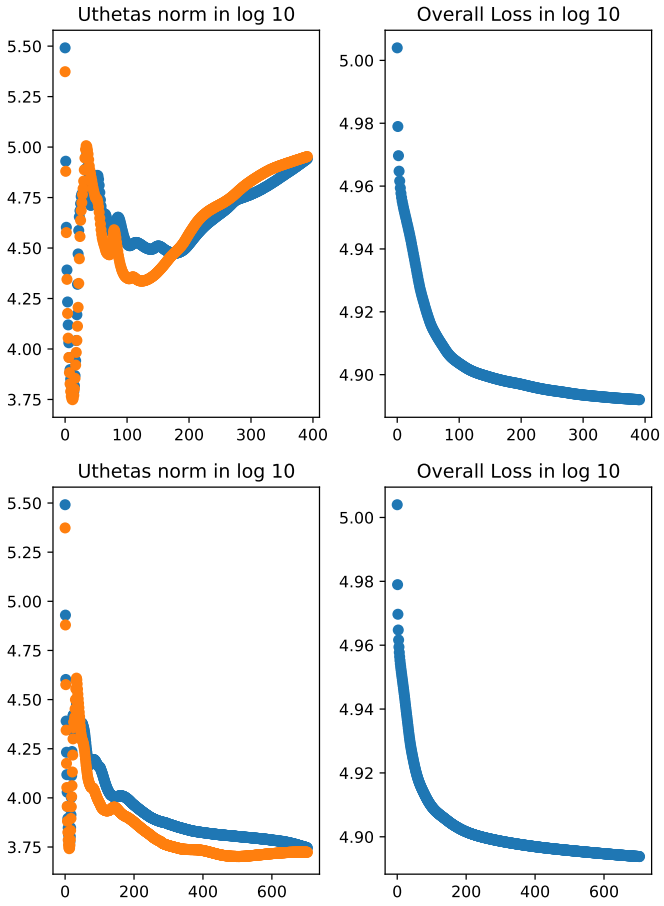


Figure 1. L_2 norm of the score functions $\mathbf{U}_{\boldsymbol{\theta}}$ and $\mathbf{U}_{\tilde{\boldsymbol{\theta}}}$ (Uthetas norm) in blue and orange colors, respectively, and overall loss of the ZIG model in $\log_{10}$ scale. Top two panels: without learning rate adjustment; bottom two panels: with learning rate adjustment. All parameters were initialized with true parameter values except for $\tilde{w}$. In the top panels, the norms of the score vectors decrease drastically in early iterations but increase over later iterations even though the overall loss is consistently reduced. In the bottom panels, the norms of the score vectors first show similar pattern as the top panel but consistently decrease in later iterations. The overall loss shows the desired reducing trend throughout all iterations.

The results for Setting 2 are given in Figure 2. The last column shows the algorithm with learning rate adjustment alone because the scale of the third column is too high to see the reducing trend. In this case, with the initial values of all parameters generated randomly from Uniform distribution, the uncertainty in the parameters caused the algorithm without learning rate to breakdown. We see that the algorithm with no learning rate adjustment using updates in Algorithm 2 struggles to reduce the norms and headed toward wrong directions throughout the optimization process. On the other hand, the algorithm with learning rate adjustment had a little fluctuation at the beginning but continued toward the right direction by reducing both the norm of the score vectors and the loss function as the iterations proceed. Note that the dataset was identical for these two applications and both algorithms started with the same initial values. The only difference between the two algorithms is that learning rate adjustment is used in one algorithm but not in the other algorithm. Hence, the failure of the algorithm with no learning rate adjustment is due to the update being too big, which leads to the algorithm going toward wrong directions. The numerical study confirms that the alternating updates are capable of finding the maximum

likelihood estimate, and learning rate adjustment in Algorithm 3 is an important component of the algorithm.

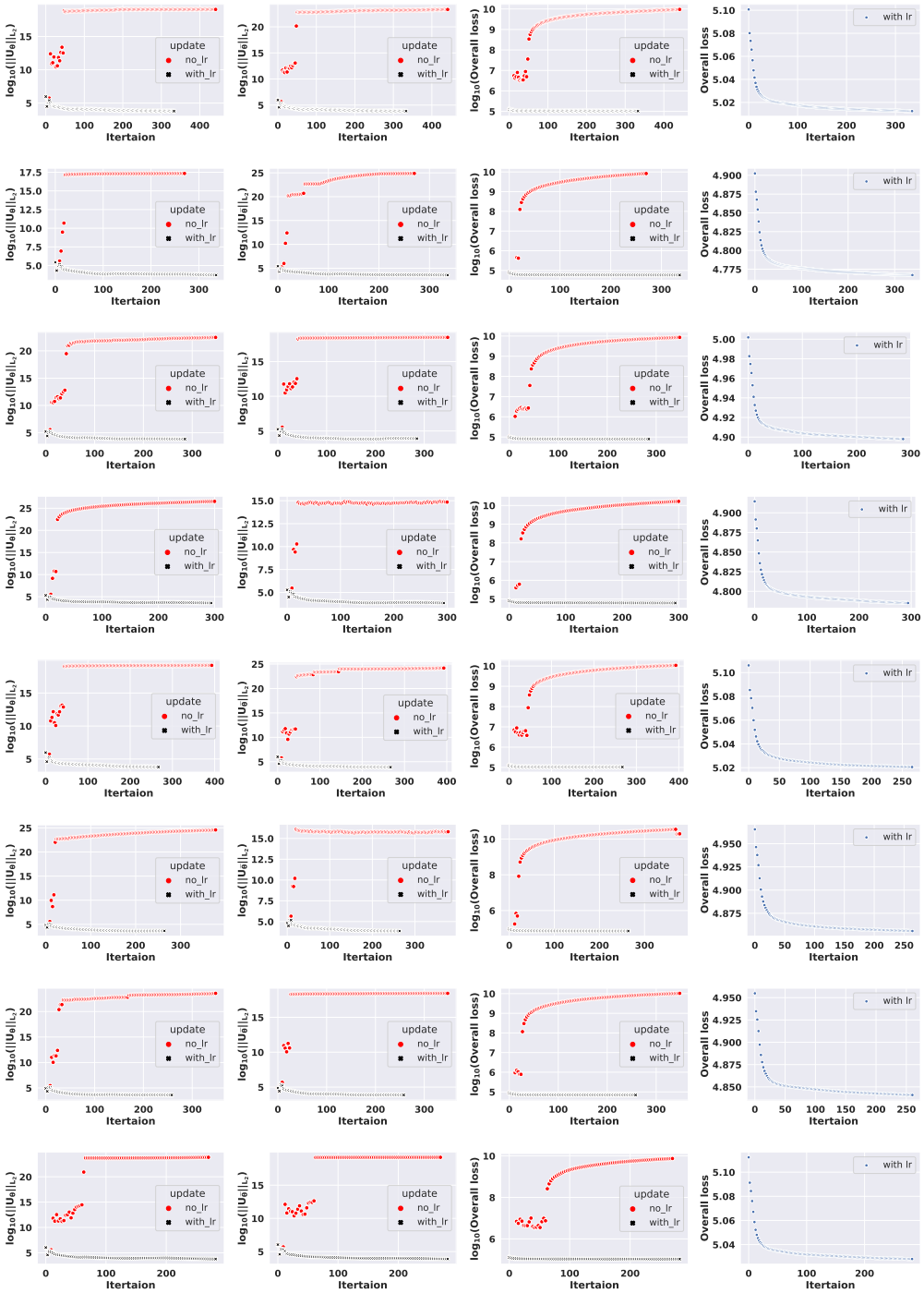


Figure 2. Comparing performance of the alternating ZIG regression using log link with or without learning rate adjustment over 8 simulated datasets. Each row is for one dataset. First column: $\log_{10}(L_2$ norm of $U_{\boldsymbol{\theta}})$; second column: $\log_{10}(L_2$ norm of $U_{\tilde{\boldsymbol{\theta}}})$; third column: $\log_{10}$(overall loss); fourth column: overall loss of the algorithm with learning rate adjustment.

Table 1. Result of the last 5 iterations for the L_2 norm of the score vectors and overall loss in ZIG model with log link. The parameters were initialized with true parameters except $\widetilde{w}$, which was randomly initialized. The negative iteration number means counting from the end. The algorithm with no learning rate adjustment did not reduce the score vectors' norms as fast as the algorithm with learning rate adjustment even though it achieved slightly lower loss.

	No Learning Rate Adjustment			**With Learning Rate Adjustment**		
Iteration	$\|\|U_{\theta}\|\|_{L_2}$	$\|\|U_{\widetilde{\theta}}\|\|_{L_2}$	**Overall Loss**	$\|\|U_{\theta}\|\|_{L_2}$	$\|\|U_{\widetilde{\theta}}\|\|_{L_2}$	**Overall Loss**
−5	85,805.79	88,896.77	77,998.77	5596.06	5298.31	78,313.71
−4	86,123.05	89,012.56	77,996.44	5596.00	5303.81	78,312.30
−3	86,614.46	89,236.36	77,994.11	5581.83	5294.08	78,310.92
−2	87,309.84	89,758.44	77,991.79	5579.14	5293.48	78,309.59
−1	87,574.27	89,699.66	77,989.41	5556.29	5286.12	78,308.15

7.2. An Application of SA-ZIG on Word Embedding

In this section, we demonstrate the application of SA-ZIG using log link on a small dataset. This dataset comprises news articles sourced from the Reuters news archive. Using a Python script to simulate user clicks, we downloaded approximately 2000 articles from the Business category, spanning from 2019 to the summer of 2021. These articles are stored in an SQLite database. From this corpus, we created a small vocabulary consisting of the $V = 300$ most frequently used words. We trained a $d = 20$-dimensional dense vector representation for each of the 300 words using the downloaded business news.

To apply the SA-ZIG model, we first obtain weighted word–word co-occurrence count, computed as follows:

$$Y_{ij} = \begin{cases} \sum_{\text{s} \in \text{ all sentences}} 1/d_{ij}(s), & \text{if } d_{ij}(s) \leq \text{k} \\ 0, & \text{otherwise} \end{cases}$$

where $d_{ij}(s)$ = separation between word i and word j in sentence s, and k is pre-determined window size 10.

Figure 3 shows the histogram of the counts from the first 12 rows. The counts for each row are highly skewed and have a lot of zeros. It can be seen that zero-inflated Gamma may be suitable for the data.

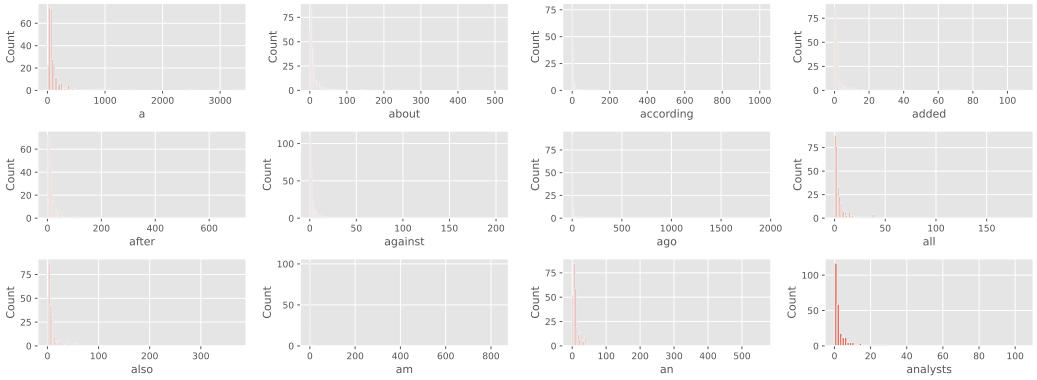


Figure 3. Histogram of weighted count from each of the first 12 rows of the data matrix.

The co-occurrence matrix with sparse format was fed into the SA-ZIG model with learning rate adjustment. The parameters are initialized based on the following description:

- The entries in $V \times d$ matrix **w** were independently generated from uniform distribution between $-0.5/(Vd)$ and $0.5/(Vd)$ with seed 99.
- The entries in $V \times d$ matrix $\widetilde{w}$ were generated from uniform distribution between $-0.5/(Vd)$ and $0.5/(Vd)$ with seed 98.

- The V bias terms in $\mathbf{b}$ were independently generated from uniform $(-0.1, 0.1)$ with seed 97.
- The V bias terms in $\widetilde{\boldsymbol{b}}$ were independently generated from uniform $(-0.1, 0.1)$ with seed 96.
- The V bias terms in $e_1, \ldots, e_V$ were independently generated from uniform $(0.1, 0.6)$ with seed 1.
- The bias terms in $\widetilde{e}_1, \ldots, \widetilde{e}_V$ were independently generated from uniform $(0.1, 0.6)$ with seed 2.
- The learning rate at t^{th} iteration is set to $lr/(t^{1/4})$ as given in Algorithm 3 with $lr = 0.5$.

We let the model parameter update run 60 iterations in the outlier loop and 20 epochs in the inner loop. These numbers are used because the GloVe model training in [13] ran 50 iterations for vector dimensions ≤ 300. We want to see how the estimation proceeds with the iteration number. At the 60th iteration, the algorithm has not converged yet. This can be seen from the loss curve in Figure 4, which still sharply moves downward. The cosine similarity between word pairs is shown in the heatmap on the right panel of Figure 4.

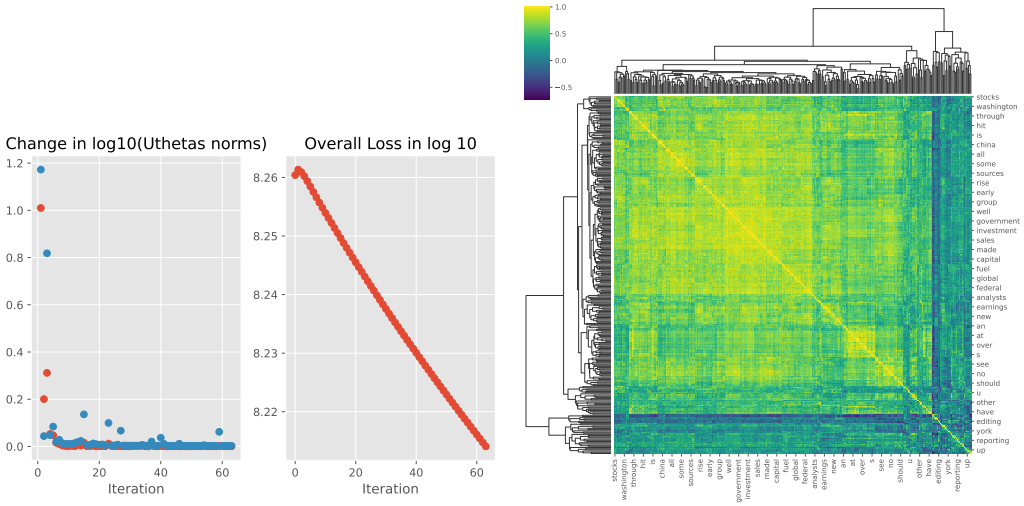


Figure 4. (**Left**): The absolute value of change in the $\log_{10}$ norms of $\mathbf{U}_{\boldsymbol{\theta}}$ (red) and $\mathbf{U}_{\widetilde{\boldsymbol{\theta}}}$ (blue) from successive iterations and overall loss curve. (**Right**): Cosine similarity of word vector representations shown as heatmap.

To visualize the relationships between different words, we first project the learned vector representations to the first 10 principal components (PCs). These 10 PCs cumulatively explained 97% of variations in the word vectors. We then conduct further dimension reduction with T-Distributed Stochastic Neighbor Embedding (t-SNE) to two-dimensional space. The t-SNE minimizes the divergence between two distributions: a distribution that measures pairwise similarities of the input objects and a distribution that measures pairwise similarities of the corresponding low-dimensional points in the embedding. The words on t-SNE coordinates are shown in Figure 5. Each point on the plot represents a word in the low-dimensional space. The closer two words are on the plot, the more similar they are in meaning or usage. For example, the top five most similar words to 'capital' from the 300 words are 'firm', 'management', 'fund', 'board', 'financial'. These are based on the cosine similarity. The five most similar words to 'stocks' from the 300 words are 'companies', 'bitcoin', 'banks', 'prices' and 'some'. The top five most similar words to 'finance' are 'technology', 'energy', 'trading', 'consumer', 'analysts'. Words 'monday', 'tuesday', 'wednesday' are clustered together. This scatter plot captures some semantic similarities.

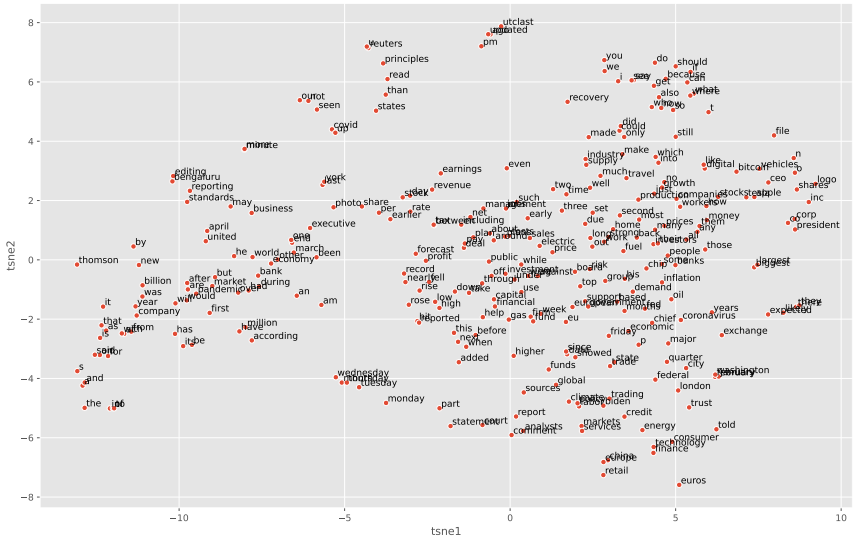


Figure 5. Plot of word vector representations using t-SNE on first 10 principal components.

8. Conclusions and Discussion

In summary, we presented the shared parameter alternating zero-inflated Gamma (SA-ZIG) regression model in this paper. The SA-ZIG model is designed for highly skewed non-negative matrix data. It uses a logit link to model the zero versus positive observations. For the Gamma part, we considered two link functions: the canonical link and the log link, and derived updating formulas for both.

We proposed an algorithm that alternately updates the parameters $\boldsymbol{\theta}$ and $\widetilde{\boldsymbol{\theta}}$ while holding one of them fixed. The Fisher scoring algorithm, with or without learning rate adjustment, was employed in each step of the alternating update. Numerical studies indicate that learning rate adjustment is crucial in SA-ZIG regression. Without it, the algorithm may fail to find the optimal direction.

After model estimation, the matrix is factorized into the product of a left matrix and a right matrix. The rows of the left matrix and the columns of the right matrix provide vector representations for the rows and columns, respectively. These estimated row and column vector representations can then be used to assess the relevance of items and make recommendations in downstream analysis.

The SA-ZIG model is inherently similar to factor analysis. In factor analysis, both the loading matrix and the coefficient vector are unknown. The key difference between SA-ZIG and factor analysis is that SA-ZIG uses a large coefficient matrix, whereas factor analysis uses a single vector. Additionally, SA-ZIG assumes a two-stage Bernoulli–Gamma model, while factor analysis assumes a normal distribution.

In both models, likelihood-based estimation can determine convergence behavior by linking the complete data likelihood with the conditional likelihood. For factor analysis, the normal distribution and the linearity of the sufficient statistic in the observed data allow the use of ALS to estimate both the loading matrix and the coefficient vector ([31]). However, SA-ZIG cannot use ALS because the variance of the Gamma distribution is not constant.

In both SA-ZIG and factor analysis, the unobserved row (or column) vector representation and the factor loading matrix can be treated as missing data, assuming the data are missing completely at random. For missing data analysis, the well-known Expectation Maximization (EM) algorithm can be used to estimate parameters. The EM algorithm has the advantageous property that successive updates always move towards maximizing the log likelihood. It works well when the proportion of missing data is small, but it is notoriously slow when a large amount of data is missing.

For SA-ZIG, the alternating scheme with the Fisher scoring algorithm offers the benefit of a quadratic rate of convergence if the true parameters and their estimates lie within the interior of the parameter space. However, in real applications, the estimation process might diverge at either stage of the alternating scheme because the Fisher scoring update does not always guarantee an upward direction, especially in cases of complete or quasi complete separation. Additionally, the algorithm may struggle to find the optimal solution due to the non-identifiability of the row and column matrices under orthogonal transformations, leading to a ridge of solutions. The learning rate adjustment in Algorithm 3 helps by making small moves during successive updates in later stage of the algorithm and thereby is more likely to find a solution.

Future research on similar problems could explore alternative distributions beyond Gamma. Tweedie and Weibull distributions, for instance, are capable of modeling both symmetric and skewed data through varying parameters, each with its own associated link functions. However, new algorithms and convergence analyses would need to be developed specifically for these distributions. In practical applications, the most suitable distribution for the observed data is often uncertain, making diagnostic procedures an important area for further investigation.

Author Contributions: Methodology, T.K. and H.W.; Software, T.K. and H.W.; Validation, T.K.; Writing—original draft, T.K. and H.W.; Writing—review & editing, T.K. and H.W.; Supervision, H.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Edelman, A.; Jeong, S. Fifty three matrix factorizations: A systematic approach. *arXiv* **2022**, arXiv:2104.08669. [CrossRef]
2. Gan, J.; Liu, T.; Li, L.; Zhang, J. Non-negative Matrix Factorization: A Survey. *Comput. J.* **2021**, *64*, 1080–1092. [CrossRef]
3. Saberi-Movahed, F.; Berahman, K.; Sheikhpour, R.; Li, Y.; Pan, S. Nonnegative matrix factorization in dimensionality reduction: A survey. *arXiv* **2024**, arXiv:2405.03615.
4. Wang, Y.-X.; Zhang, Y.-J. Nonnegative matrix factorization: A comprehensive review. *IEEE Trans. Knowl. Data Eng.* **2013**, *25*, 1336–1353. [CrossRef]
5. Koren, Y. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '08, Las Vegas, NV, USA, 24–27 August 2008; Association for Computing Machinery: New York, NY, USA, 2008; pp. 426–434.
6. Zhang, Z.; Liu, H. Social recommendation model combining trust propagation and sequential behaviors. *Appl. Intell.* **2015**, *43*, 695–706. [CrossRef]
7. Fernández-Tobías, I.; Cantador, I.; Tomeo, P.; Anelli, V.W.; Di Noia, T. Addressing the user cold start with cross-domain collaborative filtering: Exploiting item metadata in matrix factorization. *User Model. User-Adapt. Interact.* **2019**, *29*, 443–486. [CrossRef]
8. Puthiya Parambath, S.A.; Chawla, S. Simple and effective neural-free soft-cluster embeddings for item cold-start recommendations. *Data Min. Knowl. Discov.* **2020**, *34*, 1560–1588. [CrossRef]
9. Nguyen, P.; Wang, J.; Kalousis, A. Factorizing lambdaMART for cold start recommendations. *Mach. Learn.* **2016**, *104*, 223–242. [CrossRef]
10. Panda, D.K.; Ray, S. Approaches and algorithms to mitigate cold start problems in recommender systems: A systematic literature review. *J. Intell. Inf. Syst.* **2022**, *59*, 341–366. [CrossRef]
11. Mikolov, T.; Chen, K.; Corrado, G.S.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
12. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*; Burges, C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2013; Volume 26.
13. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
14. Mills, E.D. Adjusting for Covariates in Zero-Inflated Gamma and Zero-Inflated Log-Normal Models for Semicontinuous Data. Ph.D. Dissertation, University of Iowa, Iowa City, IA, USA, 2013. [CrossRef]

15. Wei, X.-X.; Zhou, D.; Grosmark, A.D.; Ajabi, Z.; Sparks, F.T.; Zhou, P.; Brandon, M.P.; Losonczy, A.; Paninski, L. A zero-inflated gamma model for post-deconvolved calcium imaging traces. *Neural Data Sci. Anal.* **2020**, *3*. [CrossRef]
16. Nobre, A.A.; Carvalho, M.S.; Griep, R.H.; Fonseca, M.d.J.M.d.; Melo, E.C.P.; Santos, I.d.S.; Chor, D. Multinomial model and zero-inflated gamma model to study time spent on leisure time physical activity: An example of elsa-brasil. *Rev. Saúde Pública* **2017**, *51*, 76. [CrossRef] [PubMed]
17. Moulton, L.H.; Curriero, F.C.; Barroso, P.F. Mixture models for quantitative hiv rna data. *Stat. Methods Med. Res.* **2002**, *11*, 317–325. [CrossRef] [PubMed]
18. Wu, M.C.; Carroll, R.J. Estimation and comparison of changes in the presence of informative right censoring by modeling the censoring process. *Biometrics* **1988**, *44*, 175–188. [CrossRef]
19. Have, T.R.T.; Kunselman, A.R.; Pulkstenis, E.P.; Landis, J.R. Mixed effects logistic regression models for longitudinal binary response data with informative drop-out. *Biometrics* **1998**, *54*, 367–383. [CrossRef]
20. Albert, P.S.; Follmann, D.A.; Barnhart, H.X. A generalized estimating equation approach for modeling random length binary vector data. *Biometrics* **1997**, *53*, 1116–1124. [CrossRef]
21. Du, J. Which Estimator of the Dispersion Parameter for the Gamma Family Generalized Linear Models Is to Be Chosen? Master's Thesis, Dalarna University, Falun, Sweden, 2007. Available online: https://api.semanticscholar.org/CorpusID:34602767 (accessed on 23 October 2024).
22. Haberman, S. *The Analysis of Frequency Data*; Midway reprint; University of Chicago Press: Chicago, IL, USA, 1977.
23. Silvapulle, M.J. On the existence of maximum likelihood estimators for the binomial response models. *J. R. Stat. Soc. Ser. B-Methodol.* **1981**, *43*, 310–313. [CrossRef]
24. Albert, A.; Anderson, J.A. On the existence of maximum likelihood estimates in logistic regression models. *Biometrika* **1984**, *71*, 1–10. [CrossRef]
25. Marschner, I.C. glm2: Fitting generalized linear models with convergence problems. *R J.* **2011**, *3*, 12–15. [CrossRef]
26. Boyd, S.; Vandenberghe, L. *Convex Optimization*; Cambridge University Press: Cambridge, UK, 2004
27. Givens, G.H.; Hoeting, J.A. *Computational Statistics*, 2nd ed.; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2013.
28. Osborne, M.R. Fisher's method of scoring. *Int. Stat. Rev. Rev. Int. Stat.* **1992**, *60*, 99–117. [CrossRef]
29. Reddi, S.J.; Kale, S.; Kumar, S. On the convergence of adam and beyond. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
30. Shi, N.; Li, D.; Hong, M.; Sun, R. RMSprop converges with proper hyper-parameter. In Proceedings of the 9th International Conference on Learning Representations (ICLR 2021), Online, 3–7 May 2021.
31. Dempster, A.P.; Laird, N.M.; Rubin, D.B. Maximum likelihood from incomplete data via the em algorithm. *J. R. Stat. Soc. Ser. B (Methodol.)* **1977**, *39*, 1–22. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.