Scalable approximation of the transformation-free linear simplicial-simplicial regression via constrained iterative reweighted least squares

Michail Tsagris[1] and Omar Alzeley[2]

[1] Department of Economics, University of Crete, Gallos Campus, Rethymnon, Greece
mtsagris@uoc.gr
[2] Department of Mathematics, Umm Al-Qura University, Saudi Arabia
oazeley@uqu.edu.sa

January 6, 2026

Abstract

Simplicia-simplicial regression concerns statistical modeling scenarios in which both the predictors and the responses are vectors constrained to lie on the simplex. Fiksel et al. (2022) introduced a transformation-free linear regression framework for this setting, wherein the regression coefficients are estimated by minimizing the Kullback-Leibler divergence between the observed and fitted compositions, using an expectation-maximization (EM) algorithm for optimization. In this work, we reformulate the problem as a constrained logistic regression model, in line with the methodological perspective of Tsagris (2025), and we obtain parameter estimates via constrained iteratively reweighted least squares. Simulation results indicate that the proposed procedure substantially improves computational efficiency-yielding speed gains ranging from $6\times --326\times$-while providing estimates that closely approximate those obtained from the EM-based approach.

Keywords: compositional data, iteratively reweighted least squares, quadratic programming

1 Introduction

Simplicial, or compositional data, data[1] are non-negative multivariate vectors whose components convey only relative information. When the vectors are scaled to sum to 1, their sample space is the standard simplex,

$$\mathbb{S}^{D-1} = \left\{ (y_1, \ldots, y_D)^\top \middle| y_i \geq 0, \sum_{i=1}^{D} y_i = 1 \right\}, \tag{1}$$

where D denotes the number of components.

Such data arise in a wide range of scientific disciplines, and the extensive literature on their proper statistical treatment attests to their prevalence in practical applications.[2]

---

[1] In econometrics, these are commonly referred to as multivariate fractional data (Mullahy, 2015, Murteira and Ramalho, 2016).

[2] For numerous examples of real-world applications involving simplicial data see Tsagris and Stewart (2020).

The frequent appearance of simplicial variables in regression settings has motivated the development of appropriate regression methodologies, leading to several recent methodological advances. Most contributions in this field focus either on the case of a simplicial response with real-valued predictors (simplicial–real regression) or on the converse setting with simplicial predictors and real-valued responses (real–simplicial regression). By contrast, the simplicial–simplicial regression setting, in which both the response and predictor variables are simplicial, has received comparatively limited attention. This constitutes the main focus of the present work.

Applications involving simplicial responses and simplicial predictors include, among others, Wang et al. (2013), who modeled the relationship between economic outputs and inputs in China, and Di Marzio et al. (2015), who predicted the composition of the moss layer using the composition of the O-horizon layer. Chen et al. (2017) investigated the association between age structure and consumption structure across economic regions. Filzmoser et al. (2018) studied differences in educational composition across countries. Aitchison (2003) and Alenazi (2019) examined relationships between alternative methods of estimating white blood cell type compositions. Chen et al. (2021) explored associations between chemical metabolites of *Astragali Radix* and plasma metabolites in rats following administration. Tsagris (2025) analyzed relationships between crop production and cultivated area in Greece, as well as voting proportion dynamics in Spain. Finally, Rios et al. (2025) investigated associations between high-dimensional multi-omics simplicial datasets.

Most existing approaches to simplicial–simplicial regression apply transformations to both simplicial variables. For example, Hron et al. (2012), Wang et al. (2013), Chen et al. (2017), and Han and Yu (2022) employed log-ratio transformations for both predictors and responses followed by multivariate linear regression. Alenazi (2019) transformed simplicial predictors via the α-transformation (Tsagris et al., 2011), applied principal component analysis, and subsequently fitted a Kullback–Leibler divergence (KLD) regression (multinomial logit) model (Murteira and Ramalho, 2016). In contrast, Fiksel et al. (2022) proposed a transformation-free linear regression (TFLR) framework, in which the regression coefficients lie on the simplex and are estimated by minimizing the KLD between observed and fitted simplicial responses using an Expectation–Maximization (EM) algorithm.

A limitation of the EM-based estimation procedure for the TFLR model is its computational burden, as the EM algorithm is generally slow to converge. To alleviate this issue, we employ a constrained iteratively reweighted least squares (CIRLS) algorithm, incorporating simplex constraints on the regression coefficients. The implementation of the EM algorithm in Tsagris (2025) was shown to be approximately four times faster than that of Fiksel et al. (2022). Our simulation studies demonstrate that CIRLS achieves additional substantial computational gains, yielding speed improvements ranging from $2\times$ to $255\times$, depending on the scenario, while producing estimates that closely approximate those obtained via the EM algorithm. These computational benefits are particularly relevant in contexts involving: (a) analysis of multiple datasets, (b) large-scale simulation studies, (c) high-dimensional compositions (Rios et al., 2025), and (d) permutation- or bootstrap-based inference procedures (Fiksel et al., 2022, Tsagris, 2025).

The next section presents the TFLR model and the EM algorithm used for parameter estimation. Section 3 describes the CIRLS algorithm for logistic regression and its adaptation to the TFLR framework. Section 4 discusses the computational complexity of both approaches, Section 5 reports simulation results, and the final section provides concluding remarks.

2   The TFLR model

Fiksel et al. (2022) developed the TFLR model as a novel framework for modeling simplicial data, addressing methodological limitations inherent in existing approaches.

The TFLR model is defined as

$$\mathbb{E}[\boldsymbol{Y}|\boldsymbol{X}] = \boldsymbol{X}\boldsymbol{B}, \tag{2}$$

where $\boldsymbol{X} \in \mathcal{S}^{D_p-1}$ denotes a D_p-dimensional simplicial predictor and $\boldsymbol{Y} \in \mathcal{S}^{D_r-1}$ denotes a D_r-dimensional simplicial response. The model relates the k-th component of the simplicial response to the predictor through the linear mapping

$$E(\boldsymbol{Y}_k \mid \boldsymbol{X}) = \sum_{j=1}^{D_p} \boldsymbol{X}_j B_{jk}, \tag{3}$$

where the coefficient matrix $\boldsymbol{B}$ itself lies in a product of simplices,

$$\boldsymbol{B} \in \left\{ \mathbb{R}^{D_p \times D_r} \mid B_{jk} \geq 0, \ \sum_{k=1}^{D_r} B_{jk} = 1 \right\}.$$

The elements of $\boldsymbol{B}$ are estimated by minimizing the Kullback–Leibler divergence between the observed and fitted response compositions:

$$\min_{\boldsymbol{B}} \left\{ \sum_{i=1}^{n} \sum_{k=1}^{D_r} y_{ik} \log \left(\frac{y_{ik}}{\sum_{j=1}^{D_p} x_{ij} B_{jk}} \right) \right\} = \max_{\boldsymbol{B}} \left\{ \sum_{i=1}^{n} \sum_{k=1}^{D_r} y_{ik} \log \left(\sum_{j=1}^{D_p} x_{ij} B_{jk} \right) \right\}. \tag{4}$$

Unlike conventional approaches relying on log-ratio transformations (ALR, CLR, ILR), the TFLR model operates directly in the simplex, thereby avoiding interpretability challenges and distributional distortions introduced by transformations. The associated estimation procedure is based on first-moment specifications via estimating equations rather than full likelihood specification, providing robustness to a variety of data-generating mechanisms. Furthermore, the method naturally accommodates zeros and ones in both predictors and responses, overcoming a major limitation of log-ratio-based techniques, which typically require preprocessing such as zero-replacement.

However, as noted by Tsagris (2025), this methodology generally allows for zeros in the response but does not fully accommodate zeros in the simplicial predictors. Consider, for example, an observation $\boldsymbol{x}_i = (x_{i1}, x_{i2}, 0, 0)$, where x_{i1} and x_{i2} are positive, and suppose that the k-th column of the estimated matrix $\boldsymbol{B}$ is $(0, 0, B_{3k}, B_{4k})^\top$ with B_{3k} and B_{4k} strictly positive. Alternatively, a column of $\boldsymbol{B}$ may consist entirely of zeros. In both situations, the product $x_{ik} B_{ik} = 0$, rendering (4) undefined. To circumvent this issue, Fiksel et al. (2022) introduce a small positive constant ($\delta = 10^{-8}$) when computing the KLD.

2.1 EM algorithm

Fiksel et al. (2022) employed the EM algorithm to estimate the parameters of the TFLR model, and in particular the column-stochastic transition matrix $\boldsymbol{B}$. The estimation problem is formulated as a missing-data problem, wherein each observed simplicial vector $\boldsymbol{y}_i$ is interpreted as the outcome of a two-stage generative mechanism. Specifically: (a) for each component j of $\boldsymbol{x}_i$, a component k of $\boldsymbol{y}_i$ is selected with probability b_{jk}; and (b) the contributions from all components of $\boldsymbol{x}_i$ are subsequently aggregated according to their proportions x_{ij}. The steps of the EM algorithm are outlined below.

1. **Initialization:** Set $\boldsymbol{B}^{(0)}$ to a valid column-stochastic matrix, i.e., each row is equal to $1/D_p$.

2. **Expectation step (E-step):** For iteration t, compute the expected latent allocations:

$$z_{ijk}^{(t)} = \frac{x_{ij} B_{jk}^{(t-1)}}{\sum_{j'=1}^{p} x_{ij} B_{jk}^{(t-1)}} \cdot y_{ik}$$

where z_{ijk} represents the contribution of predictor component j to response component k for observation i.

3. **Maximization step (M-step):** Update the transition matrix elements:

$$B_{jk}^{(t)} = \frac{\sum_{i=1}^{n} z_{ijk}^{(t)}}{\sum_{k=1}^{q} \sum_{i=1}^{n} z_{ijk}^{(t)}}$$

4. **Convergence check:** Repeat steps 2-3 until a convergence criterion is met:

$$\|\boldsymbol{B}^{(t)} - \boldsymbol{B}^{(t-1)}\|_1 < \epsilon \ \text{ or } \ \text{KLd}^{(t-1)} - \text{KLD}^{(t)} < \epsilon,$$

where $\|.\|_1$ denotes the L_1 norm and ϵ is a small tolerance value.

The EM updates naturally satisfy the non-negativity and sum-to-one constraints on each row of $\boldsymbol{B}$. Moreover, each iteration of the algorithm is guaranteed to increase the (pseudo-)likelihood function. Numerical underflow and overflow issues are mitigated by operating on log-transformed probabilities when necessary, and the algorithm accommodates zeros in the data (particularly in the simplicial response) without requiring zero-value imputation.

In addition, the Initialization Step employs starting values derived from the simplicially constrained least squares estimator of Tsagris (2025). This enhancement yielded an implementation that was reported to be approximately four times faster than that of the R package codalm (Fiksel and Datta, 2021), albeit at the cost of increased memory requirements.

3 TFLR with CIRLS

While the EM algorithm provides a convenient latent-allocation interpretation of the TFLR model, it is not essential for estimating the coefficient matrix $\boldsymbol{B}$. The EM procedure maximizes the KLD objective in (4), up to an additive constant, indirectly by introducing latent variables and iteratively improving a lower bound of the KLD criterion.

An alternative perspective arises by noting that the fitted compositions

$$\hat{y}_{ik}(\boldsymbol{B}) = \sum_j x_{ij} B_{jk}$$

behave as probabilities. Consequently, the optimization problem can be viewed as a constrained regression problem defined on the simplex. This viewpoint motivates the use of iteratively reweighted least squares (IRLS), which replaces the EM lower bound with a quadratic surrogate obtained via a second-order Taylor expansion of the objective function.

Thus, both EM and IRLS yield weighted updates for $\boldsymbol{B}$, although IRLS generates Newton-type additive steps that typically require substantially fewer iterations. Each IRLS update reduces to solving a strictly convex quadratic program subject to simplicial constraints, leading to a computationally efficient approximation to the EM solution while preserving the structure of the parameter space.

In what follows, we present preliminary concepts regarding the TFLR model, the IRLS procedure, and its constrained extension (CIRLS), and demonstrate how the latter can be applied to estimate the regression coefficients of the TFLR framework.

3.1 Iteratively reweighted least squares for logistic regression

The logistic regression is a generalised linear model where the response variable follows a Bernoulli distribution $y_i \sim \text{Bernoulli}(p_i)$, the ordinary link function is the logit $g(p_i) = \text{logit}(p_i) = \ln\left(\frac{p_i}{1-p_i}\right)$, the mean function is $\mathbb{E}[y_i] = p_i$, the variance function is $\text{Var}[y_i] = p_i(1-p_i)$, and the model is expressed as

$$\ln\left(\frac{p_i}{1-p_i}\right) = \boldsymbol{x}_i^T \boldsymbol{\beta} \Leftrightarrow p_i = \frac{\exp(\boldsymbol{x}_i^T \boldsymbol{\beta})}{1 + \exp(\boldsymbol{x}_i^T \boldsymbol{\beta})} = \frac{1}{1 + \exp(-\boldsymbol{x}_i^T \boldsymbol{\beta})},$$

where $\boldsymbol{x}_i$ is a vector containing the values of the predictor variables for the i-th observation.

For n independent observations, the log-likelihood function is

$$\ell(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left[y_i \ln(p_i) + (1 - y_i) \ln(1 - p_i) \right] = \sum_{i=1}^{n} \left[y_i \boldsymbol{x}_i^T \boldsymbol{\beta} - \ln(1 + \exp(\boldsymbol{x}_i^T \boldsymbol{\beta})) \right] \tag{5}$$

To estimate the regression coefficients $\boldsymbol{\beta}$), the Newton-Raphson algorithm that uses the first and second order derivatives may be employed. The score function (gradient of log-likelihood) is

$$\boldsymbol{s}(\boldsymbol{\beta}) = \frac{\partial \ell}{\partial \boldsymbol{\beta}} = \sum_{i=1}^{n} \boldsymbol{x}_i (y_i - p_i) = \boldsymbol{X}^T (\boldsymbol{y} - \boldsymbol{p})$$

The Fisher information matrix (negative expected Hessian of log-likelihood) is

$$\boldsymbol{I}(\boldsymbol{\beta}) = -\mathbb{E} \left[\frac{\partial^2 \ell}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^T} \right] = \sum_{i=1}^{n} p_i (1 - p_i) \boldsymbol{x}_i \boldsymbol{x}_i^T = \boldsymbol{X}^T \boldsymbol{W} \boldsymbol{X},$$

where $\boldsymbol{W} = \mathrm{diag}(w_1, \ldots, w_n)$ is the weight matrix, where $w_i = p_i(1 - p_i)$.

For logistic regression, the iteratively reweighted least squares (IRLS) algorithm works by updating the β_s in an iterative way. Define z_i to be the working response

$$z_i = \boldsymbol{x}_i^T \boldsymbol{\beta} + \frac{y_i - p_i}{w_i}.$$

The update scheme of the β_s comes from the fact that at each iteration, the $\boldsymbol{\beta}^{t+1}$ is produced by an ordinary least squares (OLS) minimization problem

$$\boldsymbol{\beta}^{(t+1)} = \min_{\boldsymbol{\beta}} \sum_{i=1}^{n} w_i \left(z_i - \boldsymbol{x}_i^\top \boldsymbol{\beta} \right)^2 = \min_{\boldsymbol{\beta}} \left[\left(\boldsymbol{z} - \boldsymbol{X}^\top \boldsymbol{\beta} \right)^\top \boldsymbol{W} \left(\boldsymbol{z} - \boldsymbol{X}^\top \boldsymbol{\beta} \right) \right].$$

Hence, the updated $\boldsymbol{\beta}$ is given by[3].

$$\boldsymbol{\beta}^{(t+1)} = (\boldsymbol{X}^T \boldsymbol{W}^{(t)} \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{z}^{(t)}. \tag{6}$$

3.2 Logistic regression with identity link and simplicial constraints

Suppose we wish to maximize Eq. (5), using the identity link, $\mathbb{E}[y_i] = \boldsymbol{x}_i^\top \boldsymbol{\beta}$, subject to the following simplex constraints: $0 \leq \beta_j \leq 1$, for all $j = 1, 2, \ldots, p$ and $\sum_{j=1}^{p} \beta_j = 1$. Note, that in this formulation, we have assumed there is no constant. Two notable differences occur, the working response $\boldsymbol{z}$ equals the original response vector $\boldsymbol{y}$, $z_i = y_i$ and $w_i = [(p_i(1 - p_i)]^{-1}$.

This approach has been used, for example, to estimate predator diet within the quantitative fatty acid signature analysis framework (Iverson et al., 2004). However, the standard IRLS algorithm must be modified to accommodate simplex constraints. In particular, at each iteration, after computing the unconstrained update, one would need to project the estimate back onto the feasible region defined by the simplex. Such projection-based corrections, however, may not be optimal from a likelihood-based perspective. A more principled alternative is to formulate the constrained weighted least squares problem at each IRLS iteration as a quadratic programming (QP) problem, leading to the constrained IRLS (CIRLS) formulation (Masselot et al., 2025).

At iteration t of the IRLS algorithm, we need to solve the following constrained weighted least squares problem

$$\begin{aligned}
\min_{\boldsymbol{\beta}} \quad & \sum_{i=1}^{n} w_i^{(t)} (z_i^{(t)} - \boldsymbol{x}_i^T \boldsymbol{\beta})^2 \\
\text{subject to} \quad & \sum_{j=1}^{p} \beta_j = 1 \\
& 0 \leq \beta_j \leq 1, \quad j = 1, 2, \ldots, p.
\end{aligned}$$

---

[3]This expression explicitly shows that IRLS for logistic regression is equivalent to Newton-Raphson optimization of the log-likelihood function.

This can be rewritten in standard QP form as:

$$\min_{\boldsymbol{\beta}} \quad -\boldsymbol{d}^T\boldsymbol{\beta} + \tfrac{1}{2}\boldsymbol{\beta}^T\boldsymbol{D}\boldsymbol{\beta}$$
$$\text{subject to} \quad \boldsymbol{A}\boldsymbol{\beta} \geq \boldsymbol{b},$$

where:

$$\begin{aligned}
\boldsymbol{D} &= \boldsymbol{X}^T\boldsymbol{W}^{(t)}\boldsymbol{X} \\
\boldsymbol{d} &= \boldsymbol{X}^T\boldsymbol{W}^{(t)}\boldsymbol{z}^{(t)} \\
\boldsymbol{A} &= \begin{pmatrix} \boldsymbol{j}_p^\top \\ \boldsymbol{I}_p \end{pmatrix} \\
\boldsymbol{b} &= (1, \boldsymbol{j}_p^\top)^\top,
\end{aligned}$$

where $\boldsymbol{j}_p = (1, \ldots, 1)^\top$ is the p-dimensional vector of $1s$, and $\boldsymbol{I}_p$ is the p-dimensional identity matrix. The first row of $\boldsymbol{A}$ corresponds to the sum of the β_s that must be equal to 1, which is denoted by the first element of the $\boldsymbol{b}$ vector. Hence the $p+1$ constraints, are in fact one equality constraint and p inequality constraints.

Algorithm 1 IRLS for logistic regression with identity link function and simplex constraints
---
1: **Initialize:** $\boldsymbol{\beta}^{(0)}$ satisfying the simplex constraints
2: $t \leftarrow 0$
3: Define $\boldsymbol{A}$ and $\boldsymbol{b}$
4: **while** not converged **do**
5: $\quad \boldsymbol{\mu}^{(t)} \leftarrow \boldsymbol{X}\boldsymbol{\beta}^{(t)}$ $\qquad\qquad\qquad\qquad\qquad$ ▷ Mean equals linear predictor for identity link
6: $\quad$ Compute weights: $w_i^{(t)} \leftarrow \left[\mu_i^{(t)}(1 - \mu_i^{(t)})\right]^{-1}$
7: $\quad \boldsymbol{W}^{(t)} \leftarrow \operatorname{diag}(w_1^{(t)}, \ldots, w_n^{(t)})$
8: $\quad$ Compute working response: $\boldsymbol{z}^{(t)} \leftarrow \boldsymbol{y}$ $\qquad\qquad\qquad\qquad\qquad\qquad$ ▷ For identity link
9: $\quad$ Set up QP problem:
10: $\quad\quad \boldsymbol{D} \leftarrow \boldsymbol{X}^T\boldsymbol{W}^{(t)}\boldsymbol{X}$
11: $\quad\quad \boldsymbol{d} \leftarrow \boldsymbol{X}^T\boldsymbol{W}^{(t)}\boldsymbol{z}^{(t)}$
12: $\quad$ Solve QP: $\boldsymbol{\beta}^{(t+1)} \leftarrow \arg\min_{\boldsymbol{\beta}} \tfrac{1}{2}\boldsymbol{\beta}^T\boldsymbol{D}\boldsymbol{\beta} - \boldsymbol{d}^T\boldsymbol{\beta}$ subject to the simplicial constraints
13: $\quad t \leftarrow t + 1$
14: **end while**
15: **return** $\boldsymbol{\beta}^{(t)}$

The QP approach offers several advantages over alternative methods for handling simplex constraints. First, the constraints are explicitly incorporated into the optimization problem at each iteration, rather than being imposed post hoc following an unconstrained update. Moreover, QP solvers employ specialized algorithms designed to efficiently manage constraints while maintaining numerical stability. For convex optimization problems—such as the weighted least squares problem arising at each IRLS iteration—QP solvers guarantee convergence to the global optimum under the imposed constraints. In contrast to transformation-based approaches (e.g., softmax reparametrization), QP directly operates on the original parameters, thereby avoiding potential numerical difficulties associated with reparametrization. Finally, modern QP solvers are highly optimized and capable of handling large-scale problems with considerable computational efficiency.

A valid starting point for $\boldsymbol{\beta}^{(0)}$ that satisfies the simplex constraints could be

$$\beta_j^{(0)} = \frac{1}{p} \quad \text{for } j = 1, 2, \ldots, p.$$

This approach assigns uniform weights to each regression coefficient. As an alternative initialization strategy, starting values can be obtained by fitting ordinary least squares under simplicial constraints

on the coefficient vector. We employ an absolute difference convergence criterion based on successive log-likelihood values. Under certain parametric configurations, the optimal solution may reside on the boundary of the simplex, resulting in one or more coefficients attaining their boundary values of 0 or 1. Such boundary solutions can induce degeneracy in the optimization problem. While quadratic programming solvers are typically equipped to handle degenerate cases, implementation with solvers employing interior point methods or alternative algorithms designed for robust handling of degeneracy may prove more reliable and computationally stable.

3.3 CIRLS formulation of the TFLR

To formulate the TFLR as a univariate logistic regression model we follow Tsagris (2025). They constructed the $\boldsymbol{D}$ matrix is a $D_{rp} \times D_{rp}$ diagonal matrix, where $D_{rp} = D_r \times D_p$, and is related to the $\boldsymbol{X}^\top \boldsymbol{X}$ in the following manner

$$\boldsymbol{D} = \begin{pmatrix} \boldsymbol{X}^\top \boldsymbol{V}_1 \boldsymbol{X} & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{0} & \ddots & \boldsymbol{0} & \vdots \\ \vdots & \boldsymbol{0} & \ddots & \boldsymbol{0} \\ \boldsymbol{0} & \dots & \boldsymbol{0} & \boldsymbol{X}^\top \boldsymbol{V}_{D_r} \boldsymbol{X} \end{pmatrix}, \tag{7}$$

where $V_k = \text{diag}\,(W_{1k}, \dots W_{nk})$ is a diagonal matrix containing the k-column of the weighting matrix $\boldsymbol{W}$, and $\boldsymbol{I}_{D_r}$ is the $D_r \times D_r$ identity matrix. The vector $\boldsymbol{d}$ is given by the vectorization of the product between $\boldsymbol{X}$ and $\boldsymbol{W} \cdot \boldsymbol{Y}$,

$$\boldsymbol{d} = \text{vec}\left(\boldsymbol{X}^\top \boldsymbol{W} \cdot \boldsymbol{Y}\right), \tag{8}$$

where "·" implies the elementwise multiplication of the two matrices.

Notably, we do not vectorize the simplicial responses or the simplicial predictors, nor the weighting matrix $\boldsymbol{W}$. The matrix $\boldsymbol{B}$ is obtained via quadratic programming, from which we subsequently compute the fitted response matrix and the corresponding KLD value. The weighting matrix is then updated, and this iterative process continues until the KLD converges. Initial values for $\boldsymbol{B}$ are obtained using the constrained least squares method of Tsagris (2025).

The construction of matrix $\boldsymbol{D}$ (7) and vector $\boldsymbol{d}$ (8) imposes substantial memory requirements. Unfortunately, no straightforward approach exists to simultaneously reduce memory demands and parallelize the quadratic programming step—a critical consideration when analyzing high-dimensional simplicial data.

Solutions obtained via TFLR may reside on the boundary of the simplex, with certain coefficients attaining their limiting values of exactly 0 or 1. Under such circumstances, the active-set approach of Goldfarb and Idnani (1983) proves particularly well suited, as it handles degeneracy by maintaining a working set of active constraints whose feasibility is rigorously enforced at each iteration. This mechanism ensures that IRLS updates remain within the simplex and that convergence is not compromised by rank deficiencies in the reduced Hessian.

We employ the dual active-set implementation provided by the R package quadprog (Turlach et al., 2019), which exhibits high computational efficiency when the number of active constraints at the optimum is small—a scenario frequently observed empirically when TFLR solutions lie near the interior of the simplex. While alternative interior-point methods could be deployed, particularly in high-dimensional regimes where the $O(D_p^3)$ factorization cost becomes dominant, we find the active-set method to be both numerically stable and highly effective across the dimensionalities considered in this work.

4 Computational complexity of EM and CIRLS algorithms

We analyze the computational and memory complexity of both the EM and CIRLS algorithms to understand their scalability properties and explain the observed speed-up factors.

4.1 Per-iteration Time Complexity

- EM Algorithm: The E-step computes the latent allocations $z_{ijk}^{(t)}$ for all $i \in n$, $j \in D_p$, and $k \in D_r$. For each triplet (i, j, k), the E-step costs $\mathcal{O}(nD_p^2 D_r)$. The M-step performs normalization in $\mathcal{O}(nD_p D_r)$ time. Thus, the complexity of the t-th iteration is $T_{\text{EM}}^{(t)} = \mathcal{O}(nD_p^2 D_r)$.

- CIRLS Algorithm: After vectorization, the main computational steps are: a) computing the linear predictor $\boldsymbol{\mu}^{(t)} = \tilde{\mathbf{X}}\tilde{\boldsymbol{\beta}}^{(t)}$ in $\mathcal{O}(nD_p D_r)$ time; b) computing the weight matrix $\mathbf{W}^{(t)}$ in $\mathcal{O}(nD_r)$ time; c) forming $\mathbf{D} = \tilde{\mathbf{X}}^\top \mathbf{W}^{(t)} \tilde{\mathbf{X}}$, which is block diagonal with D_r blocks of size $D_p \times D_p$, requiring $\mathcal{O}(nD_p^2 D_r)$ operations; and d) solving the QP problem with D_{rp} variables and $\mathcal{O}(D_{rp})$ constraints. Using interior-point methods, the QP solve costs $\mathcal{O}(D_{rp}^3) = \mathcal{O}(D_p^3 D_r^3)$. The complexity of the t-th iteration is $T_{\text{CIRLS}}^{(t)} = \mathcal{O}(nD_p^2 D_r + D_p^3 D_r^3)$.

For typical problems where $n \gg D_p D_r$, both algorithms have similar per-iteration complexity $\mathcal{O}(nD_p^2 D_r)$, as the matrix operations dominate the QP overhead. In this regime, the speed-up of CIRLS over EM arises primarily from requiring fewer iterations to converge (typically $K_{\text{CIRLS}} \approx \frac{1}{2}K_{\text{EM}}$ to $\frac{1}{5}K_{\text{EM}}$). For problems with large $D_p \times D_r$ relative to n, the QP solving becomes the computational bottleneck in CIRLS. However, this can be mitigated by exploiting the fact that the $\mathbf{B}$ matrix has D_r independent rows (each satisfying simplex constraints), allowing decomposition into D_r separate QP problems of size D_p that can be solved in parallel.

If EM requires K_{EM} iterations and CIRLS requires K_{CIRLS} iterations to converge, the total complexities are $T_{\text{EM}} = \mathcal{O}(K_{\text{EM}} \cdot nD_p^2 D_r)$ and $T_{\text{CIRLS}} = \mathcal{O}(K_{\text{CIRLS}} \cdot [nD_p^2 D_r + D_p^3 D_r^3])$.

4.2 Memory Complexity

- EM Algorithm: The main memory requirements are: a) the data matrices $\mathbf{X}$ and $\mathbf{Y}$, requiring $\mathcal{O}(n(D_p + D_r))$ space; b) the coefficient matrix $\mathbf{B}$, requiring $\mathcal{O}(D_p D_r)$ space; and c) the latent allocation array z_{ijk}, requiring $\mathcal{O}(nD_p D_r)$ space. Thus, $M_{\text{EM}} = \mathcal{O}(nD_p D_r)$.

- CIRLS Algorithm: In addition to the data and coefficients, CIRLS requires: a) the block-diagonal matrix $\mathbf{D}$, which can be stored efficiently as D_r blocks of size $D_p \times D_p$, requiring $\mathcal{O}(D_p^2 D_r)$ space; b) the QP solver workspace, typically requiring $\mathcal{O}(D_p^2 D_r^2)$ space for storing constraint matrices and factorizations. Thus, $M_{\text{CIRLS}} = \mathcal{O}(nD_p D_r + D_p^2 D_r^2)$.

Table 1 summarizes the key results. Based on the complexity analysis and empirical results, with small to medium problems ($n \leq 50{,}000$, $D_p D_r \leq 100$): CIRLS is strongly recommended, offering 5–85× speed-up with identical accuracy. With large sample sizes, ($n > 50{,}000$): CIRLS remains faster but with diminishing speed-up (2–10×) due to different scaling rates. For typical problems with small to moderate D_p and D_r, the memory requirements are similar. EM is more memory-efficient when $D_p \times D_r$ is large.

4.3 Implementation details in R

Table 1 presents the theoretical complexities of the two algorithms. The bottleneck of the CIRLS is the computation of the block matrix $\boldsymbol{D}$ (7) that involves performing $2 \times D_r$ cross-product matrix multiplications. Second, computational optimizations have been applied. Instead of computing the full KLD, at each iteration of either algorithm, only the parts involving the fitted values are computed, i.e.

Table 1: Computational and memory complexity comparison of EM and CIRLS algorithms.

	EM	**CIRLS**
Per-iteration time	$\mathcal{O}(nD_p^2 D_r)$	$\mathcal{O}(nD_p^2 D_r + D_p^3 D_r^3)$
Iterations to converge	K_{EM}	$K_{\text{CIRLS}} < K_{\text{EM}}$
Total time	$\mathcal{O}(K_{\text{EM}} n D_p^2 D_r)$	$\mathcal{O}(K_{\text{CIRLS}}[nD_p^2 D_r + D_p^3 D_r^3])$
Memory	$\mathcal{O}(nD_p D_r)$	$\mathcal{O}(nD_p D_r + D_p^2 D_r^2)$

the right hand side of Eq. (4) is computed. The matrix $\boldsymbol{D}$ (7), required by QP, is initialised once, and its elements are updated at every iteration.

Both algorithms are available in the R package Compositional (Tsagris et al., 2025). The relevant functions are `tflr()` and `tflr.irls()` and they return the same output, the running time, the number of iterations required, the KLD value, the estimated regression coefficients and optionally, the fitted values for new observations.

5 Simulation studies

Simulation studies[4] were conducted to compare the computational cost (running time) of the EM and the CIRLS estimation procedures for the regression coefficients of the TFLR. For all cases, both simplicial responses and predictors were generated from a Dirichlet distribution. The sample sizes considered were 1,000 up to 10,000 with an increasing step size of 1,000 and from then the step size increased to 5,000 up to 50,000. For all cases, the tolerance value to check convergence was $\epsilon = 10^{-6}$, and the results were averaged over 100 replicates for each combination of dimensionality and sample size. Two scenarios were considered: a) the responses and the predictor variables are independent and b) they are linearly dependent, with the following configurations:

- The simplicial response consisted of $D_r = (3, 5, 7, 10)$ components, while the simplicial predictor contained $D_p = (5, 10)$ components.

- The simplicial response consisted of $D_r = (20, 30, 40, 50)$ components, while the simplicial predictor contained $D_p = (15, 20)$ components.

EM-1 refers to the implementation in the package codalm, EM-2 refers to the implementation in the package Compositional. The tolerance value to terminate the EM and CIRLS algorithms was set to 10^{-8}. The simulations were performed in a desktop with Intel Core i9-14900K CPU at 3.2GHz, 128 GB RAM and Windows 11 Pro installed.

5.1 Speed-up factors

Figure 1 visualizes the speed-up factors of CIRLS to EM when the two compositions are independent. When $p = 5$, the speed-up factors of CIRLS compared to EM-1 range from 6 up to 99, whereas compared to EM-2 they range from 6 up to 65. When $p = 10$, the EM-1 is 20–157 times slower, whereas EM-2 is 13–76 times slower. When $p = 15$ EM-1 is 13–30 times slower than CIRLS, and EM-2 is 24–62 times slower. Finally, when $p = 20$ EM-1 is 14–51 times slower than CIRLS, and EM-2 is 27–75 times slower. Regarding the KLD, the discrepancy between EM and CIRLS was $\approx 10^{-7} - 10^{-6}$.

Figure 2 visualizes the speed-up factors when the compositions are dependent. When $p = 5$, the speed-up factors of CIRLS compared to EM-1 range from 4 up to 84, whereas compared to EM-2 they range from 8 up to 326. When $p = 10$, the EM-1 is 15 up 257 times slower, whereas EM-2 is 27-445 times

---

[4]The experiments were conducted utilizing a Dell laptop equipped with Intel Core i5-1035G1 CPU at 1GHz, with 256 GB SSD, 8 GB RAM and Windows installed.

slower. With $p = 15$ and $p = 20$, EM-1 becomes extremely slow, slower than EM-2, due to large memory requirements and the speed-up factors are not presented. EM-2 is 40-165 times slower than CIRLS when $p = 15$, and 22-155 times slower than CIRLS when $p = 20$. Notably, there was a discrepancy in the produced KLD values. CIRLS always produced slightly higher KLD than the EM algorithm, the discrepancy was $\approx 10^{-5} - 10^{-4}$.

5.2 Scalability of the EM and CIRLS algorithms

To further examine and compare the scalability of the two algorithms, we estimated their scalability rate by fitting the following model $t = \alpha n^\beta$, or $\log(t) = \alpha + \beta \log(n)$ (Goldsmith et al., 2007), and report the β coefficient, where t and n denote the running time and sample size. Tables 2 and 3 present the empirically estimated scalability rates of both algorithms. Notably, all implementations exhibit linear or sublinear scalability rates, depending on the dimensionalities of the compositions.

Table 2: Estimated scalability rates of the EM and CIRLS algorithms when the responses and the predictor variables are independent.

	Algorithm	$D_r = 3$	$D_r = 5$	$D_r = 7$	$D_r = 10$
$p = 5$	EM-1	0.89	0.90	0.89	0.96
	EM-2	0.77	0.78	0.77	0.83
	CIRLS	0.74	0.92	0.93	0.86
$p = 10$	EM-1	0.76	0.78	0.82	0.76
	EM-2	0.79	0.75	0.75	0.71
	CIRLS	1.01	1.02	0.82	0.92
$p = 15$	EM-1	0.64	0.54	0.41	0.34
	EM-2	0.64	0.54	0.41	0.34
	CIRLS	0.75	0.71	0.67	0.57
$p = 20$	EM-1	0.30	0.18	0.14	0.13
	EM-2	0.62	0.62	0.63	0.62
	CIRLS	0.72	0.65	0.55	0.45

Table 3: Estimated scalability rates of the EM and CIRLS algorithms when the responses and the predictor variables are dependent.

	Algorithm	$D_r = 3$	$D_r = 5$	$D_r = 7$	$D_r = 10$
$p = 5$	EM-1	0.92	0.91	0.83	0.66
	EM-2	0.96	0.93	0.86	0.79
	CIRLS	0.74	1.01	1.01	1.01
$p = 10$	EM-1	0.80	0.63	0.44	0.40
	EM-2	0.81	0.71	0.66	0.66
	CIRLS	0.91	0.99	0.82	0.89
$p = 15$	EM-1	0.69	0.68	0.65	0.64
	CIRLS	0.91	0.70	0.48	0.29
$p = 20$	EM-1	0.91	0.79	0.80	0.80
	CIRLS	0.88	0.63	0.40	0.18

5.3 Theoretical justification for KLD discrepancy

The small discrepancy between EM and CIRLS arises from CIRLS's use of local quadratic approximations via Taylor expansion. While EM maximizes the exact KLD objective, CIRLS solves a sequence of quadratic programs that approximate the KLD surface around the current iterate. The negligible magnitude of this discrepancy ($\approx 10^{-5} - 10^{-4}$) demonstrates that the second-order Taylor approximation is highly accurate in the TFLR setting, making CIRLS a valid and efficient alternative to EM. This is a well-understood trade-off in optimization: Newton-type methods (like IRLS/CIRLS) converge faster but use approximations, while EM guarantees exact monotonic improvement but requires more iterations.

A critical observation is that the Hessian matrix $\mathbf{H}$ is **singular** due to the simplex constraints. Specifically, for each row of $\boldsymbol{B}$, we have the constraint $\sum_{k=1}^{D_r} B_{jk} = 1$, which means the parameters lie in a $(D_r - 1)$-dimensional subspace. This introduces one zero eigenvalue per row, making the full Hessian rank-deficient. The rank of the Hessian is $\mathrm{rank}(\boldsymbol{H}) = D_p \cdot (D_r - 1) < D_p \cdot D_r$. This means $\boldsymbol{H}$ has exactly D_p zero eigenvalues (one per simplex constraint).

Both EM and CIRLS solve a constrained optimization problem, for which the Karush-Kuhn-Tucker (KKT) conditions (Gale et al., 1951, Karush, 1939) at optimality are

$$\nabla \ell(\boldsymbol{B}^*) + \sum_{j=1}^{D_p} \lambda_j \nabla g_j(\boldsymbol{B}^*) + \boldsymbol{\nu} = \mathbf{0},$$

where:

- $g_j(\boldsymbol{B}) = \sum_{k=1}^{D_r} B_{jk} - 1 = 0$ is the j-th equality constraint (the simplex constraint for the j-th row).

- $\lambda_j \in \boldsymbol{R}$ is the Lagrange multiplier for the j-th equality constraint.

- $\nabla g_j(\boldsymbol{B})$ is the gradient of the j-th constraint; when $\boldsymbol{B}$ is vectorized, this is a vector with 1's in positions corresponding to $B_{j1}, \ldots, B_{jD_r}$ and 0's elsewhere.

- $\boldsymbol{\nu} \geq \mathbf{0}$ is the vector of Lagrange multipliers for the inequality constraints $B_{jk} \geq 0$.

The proper framework is to consider the projected Hessian $\boldsymbol{H}_{\mathrm{proj}}$ onto the tangent space of the constraints. For the simplex constraints, this is the Hessian restricted to the $(D_r - 1)$-dimensional subspace for each component. Let $\boldsymbol{P}$ be the projection matrix onto the constraint manifold. Then:

$$\boldsymbol{H}_{\mathrm{proj}} = \boldsymbol{P}^T \boldsymbol{H} \boldsymbol{P}.$$

For problems involving simplicial data, $\boldsymbol{H}_{\mathrm{proj}}$ is typically positive definite with minimum eigenvalue $\lambda_{\min}^{\mathrm{proj}} > 0$ in the constraint subspace. For two feasible points (matrices) $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$ (both satisfying the simplex constraints), the strong convexity in the constraint subspace gives

$$\ell(\boldsymbol{B}_1) - \ell(\boldsymbol{B}_2) \geq \nabla \ell(\boldsymbol{B}_2)^\top (\boldsymbol{B}_1 - \boldsymbol{B}_2) + \frac{\lambda_{\min}^{\mathrm{proj}}}{2} \|\boldsymbol{P}(\boldsymbol{B}_1 - \boldsymbol{B}_2)\|^2.$$

Since both solutions are feasible, $\boldsymbol{P}(\boldsymbol{B}_1 - \boldsymbol{B}_2) = \boldsymbol{B}_1 - \boldsymbol{B}_2$ (the difference vector is already in the constraint subspace). Let $\Delta \ell = \ell(\boldsymbol{B}_{\mathrm{EM}}^*) - \ell(\boldsymbol{B}_{\mathrm{CIRLS}}^*)$ be the observed KLD difference. Then:

$$\|\boldsymbol{B}_{\mathrm{EM}}^* - \boldsymbol{B}_{\mathrm{CIRLS}}^*\| \leq \sqrt{\frac{2|\Delta \ell|}{\lambda_{\min}^{\mathrm{proj}}}}.$$

For the TFLR model, the projected Hessian can be approximated by

$$\boldsymbol{H}_{\mathrm{proj}} \approx \mathrm{diag}\left(\boldsymbol{X}^\top \boldsymbol{V}_1^{\mathrm{proj}} \boldsymbol{X}, \ldots, \boldsymbol{X}^\top \boldsymbol{V}_{D_r-1}^{\mathrm{proj}} \boldsymbol{X} \right),$$

where $\boldsymbol{V}_k^{\mathrm{proj}}$ are the projected weight matrices, and the minimum eigenvalue is aprrpximately given by

$$\lambda_{\min}^{\mathrm{proj}} \approx \mathcal{O}\left(n \cdot \min_{i,j,k} \left[\frac{x_{ij}^2 \cdot y_{ik}(1 - y_{ik})}{(\sum_j x_{ij} B_{jk})^2} \right] \right).$$

For well-conditioned simplicial data (no components too close to 0 or 1), this is approximately $\lambda_{\min}^{\text{proj}} \approx \mathcal{O}(n)$. Given the observed KLD differences of $|\Delta\ell| \approx 10^{-5} - 10^{-4}$ in the dependent case, and using $\lambda_{\min}^{\text{proj}} \approx c \cdot n$ where c is a problem-dependent constant (typically $c \in [0.1, 1]$)

$$\|\boldsymbol{B}_{\text{EM}}^* - \boldsymbol{B}_{\text{CIRLS}}^*\|_F \lesssim \mathcal{O}\left(\sqrt{\frac{10^{-4}}{c \cdot n}}\right)$$

For example, for $n = 1,000$ and assuming $c \approx 0.5$ we obtain

$$\|\boldsymbol{B}_{\text{EM}}^* - \boldsymbol{B}_{\text{CIRLS}}^*\|_F \lesssim \mathcal{O}\left(\sqrt{\frac{10^{-4}}{500}}\right) \approx 4.5 \times 10^{-4}$$

5.4 Permutation testing

We examined the effect of this discrepancy on the permutation testing. Using the same simulation studies described earlier, we performed permutation testinf for independence between the simplicial responses and the simplicial predictors. We performed 100 iterations and for each iteration we computed the p-value based on the EM algorithm and based on the CIRLS algorithm using 300 permutations. For each iteration, we computed the correlation coefficient between the permuted KLD values of the two algorithms. The correlation coefficient was always equal to 1 and the p-values were always the same.
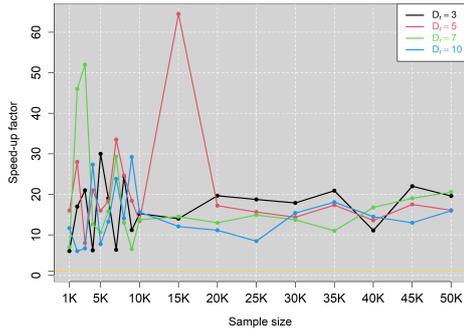
(a) EM-1 to CIRLS when $p = 5$
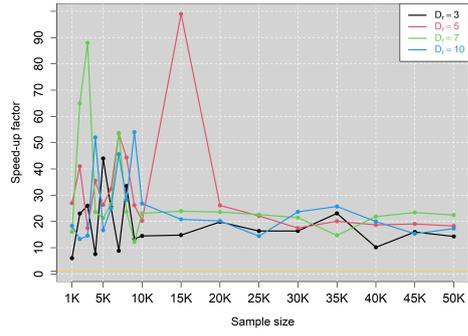
(b) EM-2 to CIRLS when $p = 5$
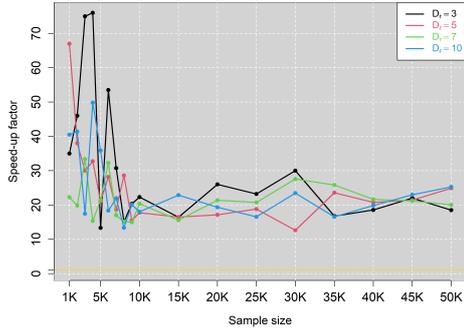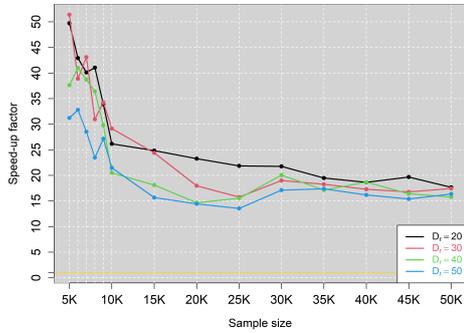
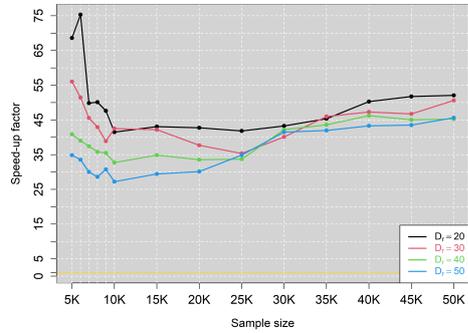(c) EM-1 to CIRLS when $p = 10$
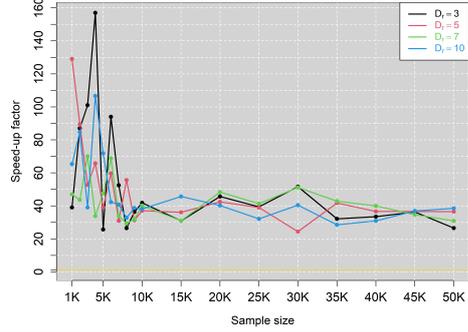
(d) EM-2 to CIRLS when $p = 10$
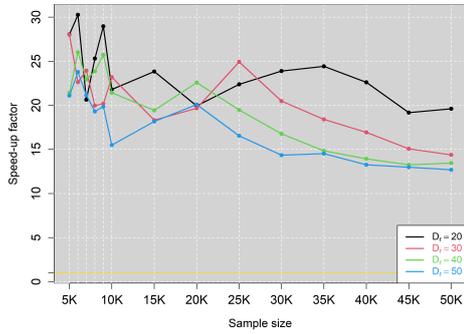
(e) EM-1 to CIRLS when $p = 15$
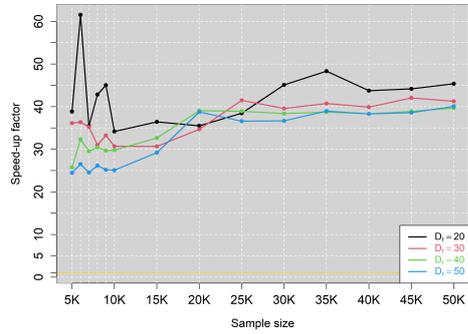
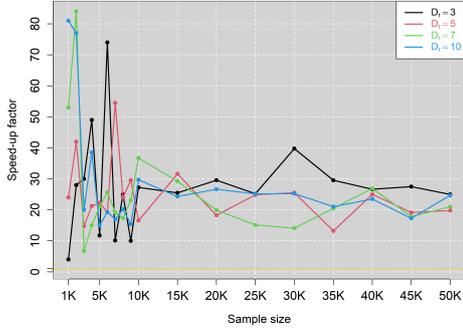(f) EM-2 to CIRLS when $p = 15$

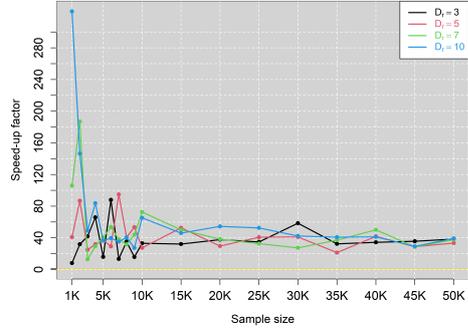(g) EM-1 to CIRLS when $p = 20$
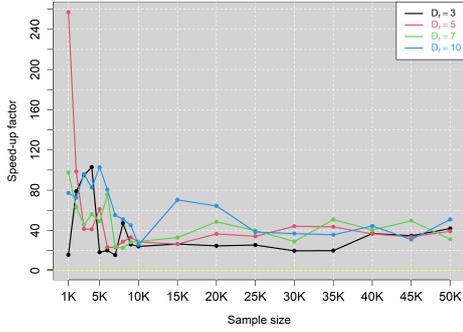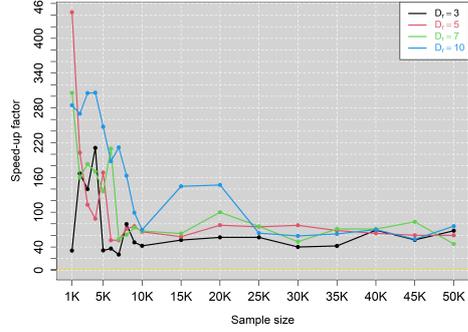
(h) EM-2 to CIRLS when $p = 20$

Figure 1: Speed-up factors of the two algorithms, when the two compositions are independent: running time of EM divided by the running time of CIRLS. The horizontal golden line indicates the unity speed-up factor, when the running times are equal.
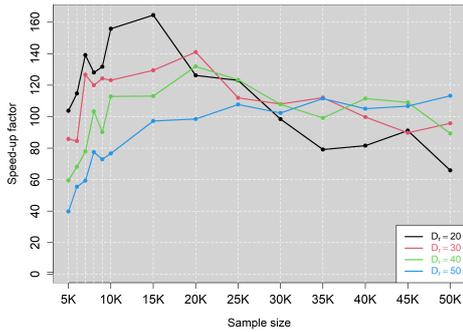
(a) EM-1 to CIRLS when $p = 5$
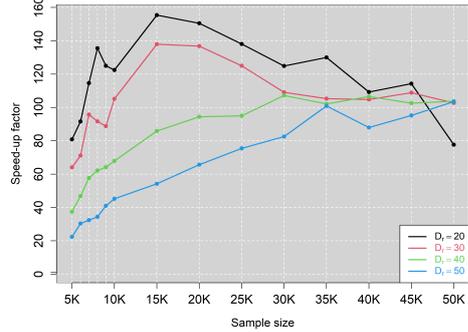
(b) EM-2 to CIRLS when $p = 5$

(c) EM-1 to CIRLS when $p = 10$

(d) EM-2 to CIRLS when $p = 10$

(e) EM-1 to CIRLS when $p = 15$

(f) EM-1 to CIRLS when $p = 20$

Figure 2: Speed-up factors of the two algorithms, when the two compositions are dependent: running time of EM divided by the running time of CIRLS. The horizontal golden line indicates the unity speed-up factor, when the running times are equal.

14

6 Conclusions

We adapted the IRLS methodology employed in generalized linear models to the TFLR framework for simplicial-simplicial regression. This adaptation exploits the formal equivalence between TFLR and logistic regression with simplicial constraints imposed on the coefficient matrix. Irrespective of the dimensionalities of both response and predictor simplices, this approach yields substantial computational improvements, with speed-up factors ranging from $6\times$ to over $300\times$. Both algorithms exhibit linear or sublinear scalability with respect to sample size and dimensionality. However, the exceptional computational efficiency of CIRLS entails a modest trade-off: when compositions exhibit dependence structure, CIRLS attains KLD values that deviate marginally ($\approx 10^{-6}$–10^{-4}) from the global minimum.

This discrepancy is negligible for most practical applications, particularly for permutation-based and bootstrap hypothesis testing procedures. Our simulation studies demonstrate that permutation-based p-values derived from EM and CIRLS exhibit strong concordance.

The reduced computational burden of CIRLS confers substantial advantages in these contexts, especially when analyzing large-scale data (with respect to sample size, dimensionality, or both). These benefits extend to scenarios involving multiple datasets or extensive simulation studies, where computational efficiency becomes paramount. Moreover, CIRLS is naturally amenable to parallelization, and through judicious programming strategies (e.g., pre-computation of invariant quantities), permutation and bootstrap testing procedures can achieve exceptional computational efficiency.

The CIRLS framework enables several promising methodological extensions in simplicial-simplicial regression. Fiksel et al. (2022) proposed minimizing the KLD from observed to fitted compositions, while Tsagris (2025) explored objective functions based on L_2 and L_1 norms. ϕ-divergence regression represents another natural extension, encompassing the symmetric KLD and Jensen-Shannon divergence (a symmetrized variant of the KLD), among others. The CIRLS algorithm can provide approximate solutions to such regression problems, yielding computationally efficient initial values for subsequent refinement via the EM algorithm.

Declarations

Conflict of interest statement: The author declares no conflict of interest.
Funding Information: This research received no funding.
Data availability: This research has used no data.

References

Aitchison, J. (2003). The statistical analysis of compositional data. New Jersey: Reprinted by The Blackburn Press.

Alenazi, A. (2019). Regression for compositional data with compositional data as predictor variables with or without zero values. Journal of Data Science, 17(1):219–237.

Chen, J., Zhang, X., and Hron, K. (2021). Partial least squares regression with compositional response variables and covariates. Journal of Applied Statistics, 48(16):3130–3149.

Chen, J., Zhang, X., and Li, S. (2017). Multiple linear regression with compositional response and covariates. Journal of Applied Statistics, 44(12):2270–2285.

Di Marzio, M., Panzera, A., and Venieri, C. (2015). Non-parametric regression for compositional data. Statistical Modelling, 15(2):113–133.

Fiksel, J. and Datta, A. (2021). codalm: Transformation-Free Linear Regression for Compositional Outcomes and Predict R package version 0.1.2.

Fiksel, J., Zeger, S., and Datta, A. (2022). A transformation-free linear regression for compositional outcomes and predictors. Biometrics, 78(3):974–987.

Filzmoser, R., Hron, K., and M., T. (2018). Applied compositional data analysis: with worked examples in R. Taylor & Francis.

Gale, D., Kuhn, H. W., and Tucker, A. W. (1951). Linear programming and the theory of games. Activity Analysis of Production and Allocation, 13:317–335.

Goldfarb, D. and Idnani, A. (1983). A numerically stable dual method for solving strictly convex quadratic programs. Mathematical Programming, 27(1):1–33.

Goldsmith, S. F., Aiken, A. S., and Wilkerson, D. S. (2007). Measuring empirical computational complexity. In Proceedings of the the 6th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on The Foundations of Software Engineering, pages 395–404.

Han, H. and Yu, K. (2022). Partial linear regression of compositional data. Journal of the Korean Statistical Society, 51(4):1090–1116.

Hron, K., Filzmoser, P., and Thompson, K. (2012). Linear regression with compositional explanatory variables. Journal of Applied Statistics, 39(5):1115–1128.

Iverson, S. J., Field, C., Don Bowen, W., and Blanchard, W. (2004). Quantitative fatty acid signature analysis: a new method of estimating predator diets. Ecological monographs, 74(2):211–235.

Karush, W. (1939). Minima of functions of several variables with inequalities as side constraints.

Masselot, P., Nenon, D., Vanoli, J., Chalabi, Z., and Gasparrini, A. (2025). Estimation and inference in generalised linear models with constrained iteratively-reweighted least squares. arXiv preprint arXiv:2509.18406.

Mullahy, J. (2015). Multivariate fractional regression estimation of econometric share models. Journal of Econometric Methods, 4(1):71–100.

Murteira, J. M. R. and Ramalho, J. J. S. (2016). Regression analysis of multivariate fractional data. Econometric Reviews, 35(4):515–552.

Rios, N., Shi, Y., Chen, J., Zhan, X., Xue, L., and Li, Q. (2025). Composition-on-composition regression analysis for multi-omics integration of metagenomic data. Bioinformatics, 41(7):btaf387.

Tsagris, M. (2025). Constrained least squares simplicial-simplicial regression. Statistics and Computing, 35(2):27.

Tsagris, M., Athineou, G., Alenazi, A., and Adam, C. (2025). Compositional: Compositional Data Analysis. R package version 7.9.

Tsagris, M., Preston, S., and Wood, A. (2011). A data-based power transformation for compositional data. In Proceedings of the 4rth Compositional Data Analysis Workshop, Girona, Spain.

Tsagris, M. and Stewart, C. (2020). A folded model for compositional data analysis. Australian & New Zealand Journal of Statistics, 62(2):249–277.

Turlach, B. A., Weingessel, A., and Moler, C. (2019). quadprog: Functions to Solve Quadratic Programming Problems. R package version 1.5-8.

Wang, H., Shangguan, L., Wu, J., and Guan, R. (2013). Multiple linear regression modeling for compositional data. Neurocomputing, 122:490–500.