Article

Logistic Biplot by Conjugate Gradient Algorithms and Iterated SVD

Jose Giovany Babativa-Márquez ^{1,2,*} and José Luis Vicente-Villardón ¹ ¹ Department of Statistics, University of Salamanca, 37008 Salamanca, Spain; villardon@usal.es² Facultad de Ciencias de la Salud y del Deporte, Fundación Universitaria del Área Andina, Bogotá 1321, Colombia

* Correspondence: jgbativam@usal.es

Abstract: Multivariate binary data are increasingly frequent in practice. Although some adaptations of principal component analysis are used to reduce dimensionality for this kind of data, none of them provide a simultaneous representation of rows and columns (biplot). Recently, a technique named logistic biplot (LB) has been developed to represent the rows and columns of a binary data matrix simultaneously, even though the algorithm used to fit the parameters is too computationally demanding to be useful in the presence of sparsity or when the matrix is large. We propose the fitting of an LB model using nonlinear conjugate gradient (CG) or majorization–minimization (MM) algorithms, and a cross-validation procedure is introduced to select the hyperparameter that represents the number of dimensions in the model. A Monte Carlo study that considers scenarios with several sparsity levels and different dimensions of the binary data set shows that the procedure based on cross-validation is successful in the selection of the model for all algorithms studied. The comparison of the running times shows that the CG algorithm is more efficient in the presence of sparsity and when the matrix is not very large, while the performance of the MM algorithm is better when the binary matrix is balanced or large. As a complement to the proposed methods and to give practical support, a package has been written in the R language called BiplotML. To complete the study, real binary data on gene expression methylation are used to illustrate the proposed methods.

Keywords: binary data; logistic biplot; optimization methods; conjugate gradient algorithm; coordinate descent algorithm; MM algorithm; low rank model; R software

Citation: Babativa-Márquez, J.G.; Vicente-Villardón, J.L. Logistic Biplot by Conjugate Gradient Algorithms and Iterated SVD. *Mathematics* **2021**, *9*, 2015. <https://doi.org/10.3390/math9162015>

Academic Editor: Liangxiao Jiang

Received: 24 June 2021

Accepted: 19 August 2021

Published: 23 August 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 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

In many studies, researchers have a binary multivariate data matrix and aim to reduce dimensions to investigate the structure of the data. For example, in the measurement of brand equity, a set of consumers evaluates the perceptions of quality, perceptions of value, or other brand attributes that can be represented in a binary matrix [1]; in the evaluation of the impact of public policies, the answers used to identify whether the beneficiaries have some characteristics or to identify if some economic or social conditions have changed from a baseline are usually binary [2–4]. Likewise, in biological research—and in particular in the analysis of genetic and epigenetic alterations—the amount of binary data has been increasing over time [5]. In these cases, classical methods to reduce dimensionality, such as principal component analysis (PCA), are not appropriate.

This problem has received considerable attention in the literature; consequently, different extensions of PCA have been proposed. From a probabilistic perspective, Collins et al. [6] provide a generalization of PCA to exponential family data using the generalized linear model framework. This approach suggests the possibility of having proper likelihood loss functions depending on the type of data.

Logistic PCA is the extension of the classic PCA method for binary data and was studied by Schein et al. [7] using the Bernoulli likelihood, where an alternating least squares method is used to estimate the parameters. De Leeuw [8] proposed the calculation

of the maximum likelihood estimates of a PCA on the logit or probit scale, using a MM algorithm that iterates a sequence of weighted or unweighted singular value decompositions. Subsequently, Lee et al. [9] introduced sparsity to the loading vectors defined on the logit transform of the success probabilities of the binary observations and estimated the parameters using an iterative weighted least squares algorithm, but the algorithm is computationally too demanding to be useful when the data dimension is high. To solve this problem, a different method was proposed by Lee and Huang [10] using a combined algorithm with coordinate descent and MM to reduce the computational effort. More recently, Landgraf and Lee [11] proposed a formulation that does not require matrix factorization, and they use an MM algorithm to estimate the parameters of the logistic PCA model. Song et al. [12] proposed the fitting of a logistic PCA model using an MM algorithm across a non-convex singular value threshold to alleviate the overfitting issues. However, neither of these approaches provides a simultaneous representation of rows and columns to visualize the binary data set, similar to what is called a biplot for continuous data [13].

The biplot methods allow the simultaneous representation of the individuals and variables of a data matrix [14]. Biplots have proven to be very useful for analyzing multivariate continuous data [15–19] and have also been implemented to visualize the results of other multivariate techniques such as multidimensional scaling, MANOVA, canonical analysis, correspondence analysis, generalized bilinear models, and the HJ-Biplot, among many others [14,20–24].

In cases where the variables of the data matrix are not continuous, a classical linear biplot representation is not suitable. Gabriel [25] proposed the “bilinear regression” to fit a biplot for data with distributions from the exponential family, but the algorithm was not clearly established and was never used in practice. For multivariate binary data, Vicente-Villardón et al. [26] proposed a biplot called Logistic Biplot (LB), which is a dimension reduction technique that generalizes PCA to cope with binary variables and has the advantage of simultaneously representing individuals and variables. In the LB, each individual is represented by a point and each variable as directed vectors, and the orthogonal projection of each point onto these vectors predicts the expected probability that the characteristic occurs. The method is related to logistic regression in the same way that classical biplot analysis is related to linear regression. In the same way, as linear biplots are related to PCA, LB is related to Latent Trait Analysis (LTA) or Item Response Theory (IRT).

The authors estimate the parameters of the LB model by a Newton–Raphson algorithm, but this presents some problems in the presence of separation or sparsity. In [27], the method is extended using a combination of principal coordinates and standard logistic regression to approximate the LB model parameters in the genotype classification context and is called an external logistic biplot, but the procedure of estimation is quite inefficient for big data matrices. More recently, in [28], the external logistic biplot method was extended for mixed data types, but the estimation algorithm still has problems with big data matrices or in the presence of sparsity. Therefore, there is a clear need to extend the previous algorithms for the LB model because they are not very efficient and none of them present a procedure for choosing the number of dimensions of the final solution of the model.

In the context of supervised learning, some optimization methods have been successfully implemented for logistic regression. For example, Komarek and Moore [29] develops Truncated-Regularized Iteratively-Reweighted Least Squares (TR-IRLS) technique that implements a linear Conjugate Gradient (CG) to approximate the Newton direction. The algorithm is especially useful for large, sparse data sets because it is fast, accurate, robust to linear dependencies, and no data preprocessing is necessary. Furthermore, another of the advantages of the CG method is that it guarantees convergence in a maximum number of steps [30]. On the other hand, when the imbalance is extreme, this problem is known as the rare events problem or the imbalanced data problem, which presents several problems and challenges to existing classification algorithms [31]. Maalouf and Siddiqi [32] developed

a method of Rare Event Weighted Logistic Regression (RE-WLR) for the classification of imbalanced data on large-scale data.

In this paper, we propose the estimation of the parameters of the LB model in two different ways: one of these is to use CG methods, and the other way is to use a coordinate descent MM algorithm. In addition, we incorporate a cross-validation procedure to estimate the generalization error and thus choose the number of dimensions in the LB model.

Taking into account the latent variables and model specification that defines a LB, a simulation process is carried out that allows for the evaluation of the performance of the algorithms and their ability to identify the correct number of dimensions to represent the multivariate binary data matrix adequately. Besides the proposed methods, the BiplotML package [33] was written in R language [34] to give practical support to the new algorithms.

The paper is organized into the following sections. Section 2.1 presents the classical biplot for continuous data. Next, Section 2.2 presents the formulation of the LB model. Section 2.3 introduces the proposed adaptation of the CG algorithm and the coordinate descent MM algorithm to fit the LB model. Section 2.4 describes the model selection procedure (number of dimensions) and introduces our formulation via simulated data. Section 3 presents the performance of the proposed models and an application using real data. Finally, Section 4 presents a discussion of the main results.

2. Materials and Methods

2.1. Biplot for Continuous Data

Classical biplot methods allow the combined visualization of the rows and columns of a data matrix $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)^T$, with $\mathbf{x}_i \in \mathbb{R}^p, i = 1, \dots, n$ being a vector of the observations of an individual i on p variables in low-dimensional space [13,14].

More generally, if $\text{rank}(\mathbf{X}) = r$ and the data are centered, given a positive integer $k \leq r$, a biplot is an approximation

$$\mathbf{X} = \mathbf{AB}^T + \mathbf{E}, \quad (1)$$

where $\mathbf{A}$ and $\mathbf{B}$ are matrices of rank k , and $\mathbf{E}$ is the matrix that contains the approximation errors. In this way, the matrix $\mathbf{X}$ can be graphically represented using markers (points or vectors) $\mathbf{a}_1, \dots, \mathbf{a}_n$ for its rows and $\mathbf{b}_1, \dots, \mathbf{b}_p$ for its columns, in such a way that the ij -th element of the matrix, x_{ij} , is approximated by the inner product $\mathbf{a}_i^T \mathbf{b}_j$, and the natural space of parameters is determined by $\mathbf{\Theta} = \mathbf{AB}^T$.

It is well known that we can reproduce exactly the matrix in dimension r using its singular value decomposition (SVD),

$$\mathbf{X} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T, \quad (2)$$

where $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_r]$ and $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_r]$ are the matrices of left and right singular vectors and $\mathbf{\Lambda}$ is the r -dimensional diagonal matrix containing the singular values in decreasing order $\lambda_1 \geq \dots \geq \lambda_r > 0$. It is also known that the low (k)-rank approximation for $\mathbf{X}$ is obtained from

$$\mathbf{\Theta} = \mathbf{U}_{(k)}\mathbf{\Lambda}_{(k)}\mathbf{V}_{(k)}^T = \mathbf{AB}^T, \quad (3)$$

where the subscript (k) stands for the first k columns of the matrix. We can define a biplot in the form of Equation (1) taking $\mathbf{A} = \mathbf{U}_{(k)}\mathbf{\Lambda}_{(k)}^\gamma$ and $\mathbf{B} = \mathbf{V}\mathbf{\Lambda}_{(k)}^{(1-\gamma)}, 0 \leq \gamma \leq 1$. Then, $\mathbf{\Theta} = \mathbf{AB}^T$ minimizes the Frobenius norm

$$\|\mathbf{X} - \mathbf{\Theta}\|_F^2 = \sum_{i=1}^n \|\mathbf{x}_i - (\mathbf{a}_{i1}\mathbf{b}_1 + \dots + \mathbf{a}_{ik}\mathbf{b}_k)\|^2, \quad (4)$$

for any value of γ . Note that, for example, if $\gamma = 1$, $\mathbf{A}$ contains the coordinates of individuals on the principal components and $\mathbf{B}$ is the projection of the initial coordinate

axis onto them. There are algorithms other than SVD used to calculate the markers as alternated regressions [13] or NIPALS [35].

2.2. Logistic Biplot

Let $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)^T$ be a binary matrix, with $\mathbf{x}_i \in \{0, 1\}^p$, $i = 1, \dots, n$ and $x_{ij} \sim \text{Ber}(\pi(\theta_{ij}))$, where $\pi(\cdot)$ is the inverse link function. In this paper, the logit link is used, $\pi(\theta_{ij}) = \{1 + \exp(-\theta_{ij})\}^{-1}$, which represents the expected probability that the character j is present in individual i , the log-odds $\pi(\theta_{ij})$ is θ_{ij} with $\theta_{ij} = \log\{\pi(\theta_{ij})/(1 - \pi(\theta_{ij}))\}$, which corresponds to the natural parameter of the Bernoulli distribution expressed in an exponential family form. Using the probability distribution, we have $P(X_{ij} = x_{ij}) = \pi(\theta_{ij})^{x_{ij}}(1 - \pi(\theta_{ij}))^{1-x_{ij}}$, and the loss function is obtained as the negative log likelihood

$$\mathcal{L}(\Theta) = - \sum_{i=1}^n \sum_{j=1}^p [x_{ij} \log(\pi(\theta_{ij})) + (1 - x_{ij}) \log(1 - \pi(\theta_{ij}))]. \quad (5)$$

In this case, it is not appropriate to center the columns because the centered matrix will no longer be made up of elements equal to zero or one. Therefore, we extend the specification of the natural parameter space by introducing variable main effects or the column offset term μ for a model-based centering. The canonical parameter matrix $\Theta = (\theta_1, \dots, \theta_n)^T$ can be represented by a low-dimensional structure for some integer $k \leq r$ that satisfies $\theta_i = \mu + \sum_{s=1}^k a_{is} \mathbf{b}_s$, $i = 1, \dots, n$, which expressed in matrix form is written as

$$\Theta = \text{logit}(\Pi) = \mathbf{1}_n \mu^T + \mathbf{A} \mathbf{B}^T, \quad (6)$$

where $\mathbf{1}_n$ is an n -dimensional vector of ones; $\mu = (\mu_1, \dots, \mu_p)^T$; $\mathbf{A} = (\mathbf{a}_1, \dots, \mathbf{a}_n)^T$ with $\mathbf{a}_i \in \mathbb{R}^k$, $i = 1, \dots, n$; $\mathbf{B} = (\mathbf{b}_1, \dots, \mathbf{b}_k)$ with $\mathbf{b}_j \in \mathbb{R}^p$, $j = 1, \dots, k$; and $\Pi = \pi(\Theta)$ is the predicted matrix whose ij th element equals $\pi(\theta_{ij})$. Now, $\Theta = \text{logit}(\Pi)$ is a biplot in the logit scale and log-odds $\theta_{ij} = \mu_j + \mathbf{a}_i^T \mathbf{b}_j$.

In addition to the canonical parameter matrix Θ , the hyperparameter k must also be estimated to select the LB model. In general, in dimensionality reduction methods, the choice of k is problematic and presents a risk of overfitting [36]. We propose the use of a completely unsupervised procedure based on cross-validation for LB model selection to counteract such overfitting. Cross-validation is operationally simple and allows the selection of the number of dimensions at the point where a numerical criterion is minimized, which will indicate that choosing more dimensions would lead to the overfitting of the model.

Once the parameters have been estimated, the geometry of the biplot regression allows us to obtain the direction vector $\mathbf{g}_j$ that projects a row marker of $\mathbf{A}$ to predict the values in column j [21,26]. For example, to find the coordinates for a fixed probability π when $k = 2$, we look for the point (g_{j1}, g_{j2}) that predicts π and that is on the biplot axis, i.e., on the line joining the points $(0, 0)$ and (b_{j1}, b_{j2}) , which is calculated as

$$g_{j1} = \frac{(\text{logit}(\pi) - \mu_j) b_{j1}}{\sum_{s=1}^2 b_{js}^2}, \quad g_{j2} = \frac{(\text{logit}(\pi) - \mu_j) b_{j2}}{\sum_{s=1}^2 b_{js}^2}. \quad (7)$$

In the logistics biplot, as in the classic PCA biplot, all directions pass through the origin. In a PCA biplot for centered data, the origin represents the mean of each variable and the arrow shows the direction of increasing values. As the binary data cannot be centered (we keep the column offset term in the model), the origin does not represent any particular value of the probability. In the LB model, we represent the variables with arrows (segments) starting at the point that predicts 0.5 and ending at the point that predicts 0.75.

2.3. Parameter Estimation

Expressing the loss function given in (5) as $\mathcal{L}(\Theta) = \sum_{i=1}^n \sum_{j=1}^p f(\theta_{ij})$ and considering that the logit link function is used, $\pi(\theta_{ij}) = (1 - \exp(-\theta_{ij}))^{-1}$, the gradient is

$$\begin{aligned}\nabla f(\theta_{ij}) &= - \left[x_{ij} \frac{1}{\pi(\theta_{ij})} \frac{\partial \pi(\theta_{ij})}{\partial \theta_{ij}} + (1 - x_{ij}) \frac{1}{1 - \pi(\theta_{ij})} \frac{\partial (1 - \pi(\theta_{ij}))}{\partial \theta_{ij}} \right] \\ &= - [x_{ij}(1 - \pi(\theta_{ij})) - (1 - x_{ij})\pi(\theta_{ij})] \\ &= \pi(\theta_{ij}) - x_{ij}.\end{aligned}\quad (8)$$

The second derivative, $\nabla^2 f(\theta_{ij}) = \pi(\theta_{ij})(1 - \pi(\theta_{ij}))$ is a quadratic function satisfying $0 \leq \nabla^2 f(\theta_{ij}) \leq 1/4$ as x_{ij} has a Bernoulli distribution.

2.3.1. Estimation Using Nonlinear Conjugate Gradient Algorithms

Let $\nabla \mathcal{L}(\Theta)$ be a matrix whose ij th element equals $\pi(\theta_{ij}) - x_{ij}$, which will be expressed in matrix form as

$$\nabla \mathcal{L}(\Theta) = \Pi - \mathbf{X}. \quad (9)$$

As the $\theta_{ij} = \mu_j + \mathbf{a}_i^T \mathbf{b}_j$, then the $\mathcal{L}(\Theta)$ function involves the matrices $\mathbf{A}$ and $\mathbf{B}$ through $\pi(\theta_{ij}) = \pi(\mu_j + \mathbf{a}_i^T \mathbf{b}_j)$. In this way, it is also possible to calculate the partial derivatives with respect to μ , $\mathbf{A}$, and $\mathbf{B}$ (see Appendix A):

$$\frac{\partial \mathcal{L}}{\partial \mu} = (\Pi - \mathbf{X})^T \mathbf{1}_n \quad (10)$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{A}} = (\Pi - \mathbf{X})\mathbf{B} \quad (11)$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{B}} = (\Pi - \mathbf{X})^T \mathbf{A} \quad (12)$$

For the model to be identifiable and to avoid the elements of $\mathbf{B}$ becoming too small due to changes in the scale of $\mathbf{A}$ and not due to changes in the log-likelihood, it is usually required that $\mathbf{A}^T \mathbf{A} = \mathbf{I}_k$ [9]. However, the maximum likelihood estimation of the model with this constraint tends to overfit the data [12]. In this paper, we use an approach that allows us to control the scale from the simultaneous updating of the parameters.

The initialization of the algorithm starts at point $\Theta_0 = \mathbf{1}_n \mu_0^T + \mathbf{A}_0 \mathbf{B}_0^T$, which can be configured by the user or by using a random initialization. For a random initialization, all elements of $\mathbf{A}_0 = (\mathbf{a}_1^0, \dots, \mathbf{a}_n^0)^T$, $\mathbf{B}_0 = (\mathbf{b}_1^0, \dots, \mathbf{b}_k^0)$, and μ_0 can be sampled from the uniform distribution. We choose the first search direction d_1^T to be the steepest descent direction at the initial point Θ_0 ; then, a line search method is used to calculate the step-length parameter α_l , and using a scalar β_l , we define the rule for updating the direction based on the gradient at every iteration and thus simultaneously update μ , $\mathbf{A}$, and $\mathbf{B}$, thus updating the natural space of parameters Θ . The pseudocode is written formally in Algorithm 1.

This is a method that is considered to have low computational cost because each iteration only requires the evaluation of the gradient and the loss function, so it can become efficient in large data sets [37,38]. In this paper, we use four well-known formulas for the update direction: Fletcher–Reeves (FR), Polak–Ribiere–Polyak (PRP), Hestenes–Stiefel (HS), and Dai–Yuan (DY) [39–42], which can be written as

$$\beta_{l+1}^{FR} = \frac{\|\nabla \mathcal{L}_{l+1}\|^2}{\|\nabla \mathcal{L}_l\|^2}; \quad \beta_{l+1}^{PRP} = \frac{\nabla \mathcal{L}_{l+1}^T \Delta_l}{\|\nabla \mathcal{L}_l\|^2}; \quad \beta_{l+1}^{HS} = \frac{\nabla \mathcal{L}_{l+1}^T \Delta_l}{d_l^T \Delta_l}; \quad \beta_{l+1}^{DY} = \frac{\|\nabla \mathcal{L}_{l+1}\|^2}{d_l^T \Delta_l}, \quad (13)$$

where $\Delta_l = \nabla \mathcal{L}_{l+1} - \nabla \mathcal{L}_l$, and $\|\cdot\|$ is the Euclidean norm. These methods have had modifications that combine the formulas [43–49] and that could be adapted to fit a LB model.

Algorithm 1 CG Algorithm for fitting a LB model

Input $\mathbf{X}$
Output $\mu, \mathbf{A}, \mathbf{B}$

- 1: Initialize $\mu_0, \mathbf{A}_0, \mathbf{B}_0$
- 2: $\Theta_0 = \mathbf{1}_n \mu_0^T + \mathbf{A}_0 \mathbf{B}_0^T$
- 3: $\nabla \mathcal{L}_0 = \nabla \mathcal{L}(\Theta_0)$
- 4: $d_0 = -\nabla \mathcal{L}_0$
- 5: $l = 0$
- 6: **repeat**
- 7: $\Pi_l = \pi(\Theta_l)$
- 8: $\alpha_l = \arg \min_{\alpha > 0} \mathcal{L}(\Theta_l + \alpha d_l)$
- 9: $\mathbf{A}_{l+1} = \mathbf{A}_l + \alpha_l (\Pi_l - \mathbf{X}) \mathbf{B}_l$
- 10: $\mathbf{B}_{l+1} = \mathbf{B}_l + \alpha_l (\Pi_l - \mathbf{X})^T \mathbf{A}_l$
- 11: $\mu_{l+1} = \mu_l + \alpha_l (\Pi_l - \mathbf{X})^T \mathbf{1}_n$
- 12: $\Theta_{l+1} = \mathbf{1}_n \mu_{l+1}^T + \mathbf{A}_{l+1} \mathbf{B}_{l+1}^T$
- 13: $\nabla \mathcal{L}_{l+1} = \pi(\Theta_{l+1}) - \mathbf{X}$
- 14: Compute β_{l+1} according to one of the formulas given in (13)
- 15: $d_{l+1} = -\nabla \mathcal{L}_{l+1} + \beta_{l+1} d_l$
- 16: **until** $(\mathcal{L}(\Theta_l) - \mathcal{L}(\Theta_{l+1})) / \mathcal{L}(\Theta_l) < \epsilon$

To achieve convergence in the implementation of the CG algorithm, one often requires the step-length obtained with a line search to be exact or satisfy the strong Wolfe conditions,

$$\begin{aligned} \mathcal{L}(\Theta_l) - \mathcal{L}(\Theta_l + \alpha_l d_l) &\geq -c_1 \alpha_l \nabla \mathcal{L}_l^T d_l, \\ \left\| \nabla \mathcal{L}(\Theta_l + \alpha_l d_l)^T d_l \right\| &\leq -c_2 \nabla \mathcal{L}_l^T d_l, \end{aligned} \quad (14)$$

with $0 < c_1 < c_2 < 1$. As the FR method has been shown to be globally convergent under strong Wolfe line searches with $c_2 < 1/2$ [50,51], this condition ensures that all d_l directions are descending directions from $\mathcal{L}$.

2.3.2. Estimation Using Coordinate Descendent MM Algorithm

As the optimization problem is non-convex, we also use a MM algorithm to generate solutions that decrease with each iteration. The negative log-likelihood can be majorized by a quadratic function, and the majorized function can then be minimized iteratively.

According to Taylor's theorem, and taking into account the fact that $\nabla^2 f(\theta_{ij}) \leq 1/4$, the loss functions of a single estimated natural parameter given in (5) are quadratically approximated at $\theta_{ij}^{(l)}$ by

$$\begin{aligned} f(\theta_{ij}) &= -[x_{ij} \log(\pi(\theta_{ij})) + (1 - x_{ij}) \log(1 - \pi(\theta_{ij}))] \\ &\approx f(\theta_{ij}^{(l)}) + (\pi(\theta_{ij}^{(l)}) - x_{ij})(\theta_{ij} - \theta_{ij}^{(l)}) + \frac{1}{2} \pi(\theta_{ij}^{(l)})(1 - \pi(\theta_{ij}^{(l)}))(\theta_{ij} - \theta_{ij}^{(l)})^2 \\ &\leq f(\theta_{ij}^{(l)}) + (\pi(\theta_{ij}^{(l)}) - x_{ij})(\theta_{ij} - \theta_{ij}^{(l)}) + \frac{1}{8} (\theta_{ij} - \theta_{ij}^{(l)})^2 \end{aligned} \quad (15)$$

$$\begin{aligned} &= f(\theta_{ij}^{(l)}) + \frac{1}{8} \left(\theta_{ij} - \theta_{ij}^{(l)} + 4(\pi(\theta_{ij}^{(l)}) - x_{ij}) \right)^2 - 2(\pi(\theta_{ij}^{(l)}) - x_{ij})^2 \\ &= \frac{1}{8} \left(\theta_{ij} - \theta_{ij}^{(l)} + 4(\pi(\theta_{ij}^{(l)}) - x_{ij}) \right)^2 + C. \end{aligned} \quad (16)$$

Consequently, the loss function for the whole canonical matrix of parameters is majorized by

$$\mathcal{L}(\Theta) \leq \frac{1}{8} \sum_{i=1}^n \sum_{j=1}^p \left(\theta_{ij} - z_{ij}^{(l)} \right)^2 + C, \quad (17)$$

where C is a constant that does not depend on Θ , and $z_{ij}^{(l)} = \theta_{ij}^{(l)} + 4(x_{ij} - \pi(\theta_{ij}^{(l)}))$.

Let $\mathbf{Z}_l$ be the matrix whose ij th element equals $z_{ij}^{(l)}$. According to [52], the iteratively weighted least squares algorithm can be used as an upper bound for a quadratic function, which allows for the minimization of the majorization function as

$$\mathcal{L}(\Theta) \leq \frac{1}{8} \|\Theta - \mathbf{Z}_l\|_F^2 + C, \quad (18)$$

and thus the majorized function to be minimized is written as

$$\min_{\mu, \mathbf{A}, \mathbf{B}} \|\mathbf{1}_n \mu^T + \mathbf{A} \mathbf{B}^T - \mathbf{Z}_l\|_F^2. \quad (19)$$

In this case, the parameters are estimated sequentially. The algorithm is based on the coordinate descent optimization of the majorized function. When fixing $\mathbf{A} \mathbf{B}^T$ in Equation (19), μ can be estimated by the vector of mean scores of each column of matrix $\mathbf{Z}_l$, $\mu = \frac{1}{n} \mathbf{Z}_l \mathbf{1}_n$. In this way, the optimization problem becomes $\min_{\mathbf{A}, \mathbf{B}} \|\mathbf{A} \mathbf{B}^T - \mathbf{P} \mathbf{Z}_l\|_F^2$ where $\mathbf{P} = \mathbf{I} - \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^T$. To initialize $\Theta_0 = \mathbf{1}_n \mu_0^T + \mathbf{A}_0 \mathbf{B}_0^T$, we use $\mathbf{A}_0 = (\mathbf{a}_1^0, \dots, \mathbf{a}_n^0)^T$; $\mathbf{B}_0 = (\mathbf{b}_1^0, \dots, \mathbf{b}_k^0)$. μ_0 can be sampled from the uniform distribution. Algorithm 2 presents the pseudocode of this process.

Algorithm 2 Coordinate descent MM algorithm for fitting a LB model

Input $\mathbf{X}$
Output $\mu, \mathbf{A}, \mathbf{B}$

- 1: Initialize $\mu_0, \mathbf{A}_0, \mathbf{B}_0$
- 2: $\Theta_0 = \mathbf{1}_n \mu_0^T + \mathbf{A}_0 \mathbf{B}_0^T$
- 3: $l = 0$
- 4: **repeat**
- 5: $\Pi_l = \pi(\Theta_l)$
- 6: $\mathbf{Z}_l = \Theta_l + 4(\mathbf{X} - \Pi_l)$
- 7: $\mu_{(l+1)} = \frac{1}{n} \mathbf{Z}_l \mathbf{1}_n$
- 8: $\mathbf{Z}_{l+1}^c = \mathbf{P} \mathbf{Z}_l$, with $\mathbf{P} = \mathbf{I} - \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^T$
- 9: $\mathbf{Z}_{l+1}^c = \mathbf{U} \mathbf{\Lambda} \mathbf{V}^T$
- 10: $\mathbf{A}_{l+1} = \mathbf{U} \mathbf{\Lambda}$
- 11: $\mathbf{B}_{l+1} = \mathbf{V}$
- 12: $\Theta_{l+1} = \mathbf{1}_n \mu_{l+1}^T + \mathbf{A}_{l+1} \mathbf{B}_{l+1}^T$
- 13: **until** $(\mathcal{L}(\Theta_l) - \mathcal{L}(\Theta_{l+1})) / \mathcal{L}(\Theta_l) < \epsilon$

2.4. Simulated and Real Data

2.4.1. Real Data

We used data from the Genomic Determinants of Sensitivity in Cancer 1000 (GDSC1000) [5]. The database contains 926 tumor cell lines with comprehensive measurements of point mutation, CNA, methylation, and gene expression. For the purpose of illustrating the methods, the methylation data were used, and to facilitate the interpretation of the results, three types of cancer were included: breast invasive carcinoma (BRCA), lung adenocarcinoma (LUAD), and skin cutaneous melanoma (SKCM).

2.4.2. Simulation Process

The data sets were simulated from a latent variables model with different levels of sparsity and with a low-dimensional structure according to the model presented in Equation (6). To generate the binary matrix $\mathbf{X}$, we used the procedure presented in Algorithm 3.

The offset term μ was used to control the sparsity, while the log-odds Θ was defined to have a low-dimensional structure.

Algorithm 3 Algorithm to simulate a binary data matrix

Input n, p, k, D .
Output $\mathbf{X}, \boldsymbol{\Theta}, \boldsymbol{\mu}, \mathbf{A}, \mathbf{B}$

- 1: $\mathbf{B} = (\mathbf{b}_1, \dots, \mathbf{b}_k)$ with $\mathbf{b}_j \sim \mathcal{N}(0, 1), j = 1, \dots, k$.
- 2: **procedure** GRAM-SCHMIDT ALGORITHM
- 3: $\mathbf{B}^T \mathbf{B} = \mathbf{I}_k$
- 4: **end procedure**
- 5: $\boldsymbol{\mu} = (\mu_1, \dots, \mu_p)^T$ with $\mu_i = \ln\left(\frac{D}{1-D}\right)$
- 6: $\mathbf{A} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_k)$
- 7: $\boldsymbol{\Theta} = \mathbf{1}_n \boldsymbol{\mu}^T + \mathbf{A} \mathbf{B}^T$
- 8: $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)^T$, with $x_{ij} \sim \text{Ber}(\pi(\theta_{ij})), \pi(\theta_{ij}) = \{1 + \exp(-\theta_{ij})\}^{-1}$.

2.5. Model Performance Assessment

To evaluate the performance of the CG algorithm and the MM algorithm, we used the training error, which is defined as the average misclassification rate when using the low-dimensional structure generated by the LB model with the training set. Each algorithm provided an estimate of $\boldsymbol{\mu}, \mathbf{A}$, and $\mathbf{B}$. We used the predicted probability matrix $\boldsymbol{\Pi} = \pi(\mathbf{1}_n \boldsymbol{\mu}^T + \mathbf{A} \mathbf{B}^T)$ to select p thresholds, $0 < \delta_j < 1, j = 1, \dots, p$ —one for each column of $\mathbf{X}$ —and then perform the binary classification. As the data matrices could be imbalanced, we used the training error defined as Balanced Accuracy (BACC), which is most appropriate in these cases [53,54]

$$BACC = \frac{1}{2} \left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP} \right), \quad (20)$$

where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives, and FN is the number of false negatives.

To decide the classification rule from the predicted value, a threshold must be selected for each variable. This threshold value can be selected by minimizing the BACC rate for each variable in the training set and then applying the rule to a test set.

As the training error can be an optimistic measure of the misclassification rate, we also used a cross-validation procedure that allowed for the testing of the models; thus, we used a test data set that was independent of the training data set. As in the supervised models, cross-validation was used to keep track of the overfitting of the model. In our case, the LB model is a non-supervised method to reduce the dimensionality, and cross-validation helps to find the number of dimensions to maintain [36,55–57]. The procedure allowed us to estimate the value of the hyperparameter k to avoid overfitting and evaluate the capacity of the different estimation algorithms to identify the low-dimensional space.

As in PCA, we expect that a few dimensions would reproduce the original data (binary) as well as possible; then, we excluded some observed values, fit the model without those values, and calculated the expected probabilities for the missing data. Using the rule of classifying a missing value as being present if the fitted probability is higher than a threshold δ_j , we imputed the selected missing data and calculated the classification. We performed this procedure M times for different values of k , and select the value that minimizes the generalization error.

More formally, we used the exclusion pattern proposed by Wold [55], which consists of a selected sequence of elements x_{ij} being left out using a diagonal pattern and treated as missing values; in this way, we avoided the problems of excluding a complete row or column as described in [36,57]. Let $\mathbf{W}$ be a binary matrix, where $w_{ij} = 0$ if the value is

excluded (belongs to the validation set) and $w_{ij} = 1$ is not (belongs to the training set). The minimization problem thus became

$$\begin{aligned} \min_{\mu, \mathbf{A}, \mathbf{B}} -\log(p(\mathbf{X}; \boldsymbol{\Theta}, \mathbf{W})) \\ &= -\log\left(\prod_{i=1}^n \prod_{j=1}^p [p(x_{ij}; \theta_{ij})]^{w_{ij}}\right) \\ &= -\sum_{i=1}^n \sum_{j=1}^p w_{ij} [x_{ij} \log(\pi(\theta_{ij})) + (1 - x_{ij}) \log(1 - \pi(\theta_{ij}))]. \end{aligned} \quad (21)$$

The loss function now depends only on the training set. Solving the problem with one of the proposed algorithms, we created a matrix of expected probabilities $\mathbf{\Pi} = \pi(\boldsymbol{\Theta})$. Then, $\hat{x}_{ij} = 1$ when $\pi(\boldsymbol{\Theta}) > \delta_j$, and $\hat{x}_{ij} = 0$ otherwise, where $0 < \delta_j < 1$ is the threshold for minimizing the BACC rate for variable j in the training set. We used the elements of $\hat{\mathbf{X}}$ when $w_{ij} = 0$ to calculate the BACC. This procedure was performed M times, and the generalization error given by the cross-validation was calculated using the mean of the BACC; we considered $M = 7$ folds as in [57].

We also measured the models' ability based on each algorithm to recover the log-odds $\boldsymbol{\Theta}$; for this, we used the relative mean squared error (RMSE), defined as

$$RMSE(\boldsymbol{\Theta}) = \frac{\|\boldsymbol{\Theta} - \hat{\boldsymbol{\Theta}}\|_F^2}{\|\boldsymbol{\Theta}\|_F^2}. \quad (22)$$

where $\boldsymbol{\Theta}$ is the true parameter and $\hat{\boldsymbol{\Theta}}$ is estimated by the LB model using one of the proposed algorithms.

3. Results

3.1. Monte Carlo Study

Binary matrices were simulated with $n = 100, 300, 500$; $p = 50, 100$; $k = 3$; and $D = 0.5, 0.3, 0.2, 0.1$, where the parameter D represents the proportion of ones in matrix $\mathbf{X}$. The different sparsity levels were simulated to check if the performance with the algorithms to find the low-dimensional structure was affected for this reason. The combinations of n , p , k , and D generated the different scenarios; in each scenario, R matrices were simulated independently, and the measures were calculated to evaluate the performance of the algorithms. Finally, the mean of the cross-validation error (cv error) was calculated, as well as the mean of the training error (BACC) and the mean of the relative mean squared error (RMSE) with their respective standard errors. We used a value of $R = 30$ that generated standard errors less than 1% in the estimation of the BACC and cv error.

Figure 1a presents the cross-validation error of the algorithms based on the conjugate gradient and the MM algorithm when the matrix $\mathbf{X}$ is balanced. From the cv error, we can see that all models began to overfit when $k > 3$, so the five estimation methods identified the three underlying dimensions for all balanced scenarios that were simulated. Figure 1b shows the Balanced Accuracy (BACC); it is highlighted that the slope decelerated when the three underlying dimensions were reached in the matrix, and thus an elbow criterion could be used as a signal of the number of dimensions to choose. Finally, Figure 1c shows the estimation of the relative mean squared error (RMSE) for the matrix $\boldsymbol{\Theta}$; we see that the algorithms based on the conjugate gradient and the MM algorithm showed similar results when the number of dimensions was less than or equal to 3. Whereas, when the model had more than the three predefined dimensions in the simulation, the algorithms based on the conjugate gradient presented a lower RMSE than the MM algorithm, although, for a fixed value of p , these gaps closed as n increased.

Figure 1. Model selection and performance with balanced data. (a) Cross-validation error. (b) Training error. (c) RMSE.

Figure 2a–c show the cross-validation errors when the data are imbalanced with $D = 0.3, 0.2$, and 0.1 , respectively. In all the scenarios studied, it is observed that the error was minimized when the number of underlying dimensions in the space of the variables was reached, so our method identified that a value of $k = 3$ in the LB model was appropriate to avoid overfitting. As this occurred with all imbalanced data sets, then we see that the level of dispersion does not affect the performance of the algorithms in terms of correctly finding the low-dimensional space.

Figure 2. Cross-validation error plot for imbalanced data sets with different levels of sparsity. (a) $D = 0.3$. (b) $D = 0.2$. (c) $D = 0.1$.

The training error for imbalanced data sets with different levels of sparsity is shown in Figure 3a–c. In all the studied scenarios, it is observed that the percentage of loss in the training error stabilized from the third dimension. In this way, the different algorithms allowed the low-dimensional space to be appropriately selected using the elbow method.

The RMSE of the estimation of the log-odds Θ for the different levels of sparsity is shown in Figure 4a–c; we can see that the algorithms presented similar performances. In the scenarios of $n = 100$ and $p = 50$, it is observed that the RMSE increased notably when the number of dimensions was greater than 3, so there were some important gaps between the two approaches, although these differences decreased as the value of p or the value of n increased.

Figure 3. Training error plot for unbalanced data sets with different levels of sparsity. (a) $D = 0.3$. (b) $D = 0.2$. (c) $D = 0.1$.

Figure 4. RMSE of estimating Θ for imbalanced data sets with different levels of sparsity. (a) $D = 0.3$. (b) $D = 0.2$. (c) $D = 0.1$.

On the other hand, the computational performance of the algorithms was measured on a computer with an Intel Core i7-3517U processor with 6 GB of RAM. Table 1 shows the the running time in seconds with 100 replications for $k = 3$ and a stopping criterion of $\epsilon = 10^{-4}$. We see that the performances of the different algorithms were competitive, and they converged relatively quickly when the maximum of the absolute changes of the estimated parameters in two subsequent iterations was less than 10^{-4} . In general, it is observed that the CPU times of the CG algorithms were similar and presented a better performance than the MM algorithm when $p = 50$ and the sparsity level began to be high ($D \leq 0.2$), or when $n = 100$, $p = 100$, and $D = 0.1$. In the other cases, the MM algorithm performed better, especially when n and p increased, resulting in up to six times faster performance than CG algorithms in balanced scenarios when $n = 500$ and $p = 100$.

Table 1. Running time in seconds to estimate the LB model with $k = 3$ and $\epsilon = 10^{-4}$.

n	p	D	DY	FR	HS	PR	MM
100	50	0.1	29.9	30.1	29.9	29.9	166.1
300	50	0.1	84.6	85.4	85.0	85.3	276.3
500	50	0.1	141.2	141.5	141.9	141.1	356.3
100	100	0.1	58.2	57.8	57.9	58.6	115.1
300	100	0.1	168.6	167.4	168.9	169.4	155.3
500	100	0.1	302.4	279.6	330.8	322.3	215.5
100	50	0.2	30.1	30.1	30.3	30.1	60.8
300	50	0.2	102.8	103.1	102.5	100.8	133.3
500	50	0.2	141.7	159.3	142.4	141.9	140.1
100	100	0.2	62.4	58.4	63.6	58.6	35.7
300	100	0.2	169.8	169.0	168.4	168.9	67.3
500	100	0.2	283.0	288.0	281.9	283.1	99.5
100	50	0.3	30.1	30.6	30.2	30.4	36.6
300	50	0.3	86.2	86.6	86.2	86.2	60.1
500	50	0.3	143.1	143.2	143.0	143.9	99.3
100	100	0.3	58.7	59.0	58.8	58.8	26.8
300	100	0.3	170.5	170.5	170.3	169.7	50.1
500	100	0.3	284.2	284.5	284.4	284.6	77.3
100	50	0.5	30.2	31.1	32.6	30.9	27.6
300	50	0.5	87.1	98.7	87.5	86.9	55.8
500	50	0.5	145.6	144.6	145.4	144.8	103.0
100	100	0.5	58.7	58.9	59.2	59.2	18.8
300	100	0.5	170.8	171.2	171.0	171.1	30.2
500	100	0.5	286.4	327.9	290.6	341.6	56.4

3.2. Application

To apply the proposed methodology, we used data from the Genomic Determinants of Sensitivity in Cancer 1000 (GDSC1000) [5]. The database contains 926 tumor cell lines with comprehensive measurements of point mutation, CNA, methylation, and gene expression. To illustrate the methods, the methylation data were used, and to facilitate the interpretation of the results, three types of cancer were included: breast invasive carcinoma (BRCA), lung adenocarcinoma (LUAD), and skin cutaneous melanoma (SKCM).

We performed preprocessing to sort the data sets and separate the methylation information into one data set. After preprocessing, the methylation dataset has 160 rows and 38 variables, each variable is a CpG island located in the gene promoter area. In this case, code 1 indicates a high level of methylation, and 0 indicates a low level; approximately 27% of the binary data matrix are ones.

Figure 5 shows the cross-validation error and training error using the conjugate gradient algorithms and the coordinate descent MM algorithm. If $k = 0$, the model (6) only considered the term μ , meaning that $\Theta = \mathbf{1}_n \mu^T$, where μ_j is the proportion of ones in column j and was used as a reference to observe the performance of the algorithms when more dimensions were included by incorporating the row and column markers, $\Theta = \mathbf{1}_n \mu^T + \mathbf{A} \mathbf{B}^T$.

The cross-validation error was minimized in three dimensions for the four formulas (FR, HS, PR, and DY) based on the CG algorithm, so $k = 3$ was the appropriate value to avoid overfitting when using these methods of estimation. When using the MM algorithm, it was found that the LB model generated overfitting for $k > 2$, so using two dimensions in the LB model is suitable when using this estimation method.

Figure 5. Cross-validation plots for CG algorithms and the coordinate descent MM algorithm with methylation data. (a) FR. (b) HS. (c) PR. (d) DY. (e) MM.

An advantage of using a biplot approach is that it allows for a simultaneous representation of rows and columns, which are plotted with points and directed vectors, respectively. Figure 6 shows the biplot obtained for the methylation data using the Fletcher–Reeves conjugate gradient algorithm; the vectors of the variables are represented by arrows (segments) that start at the point that predicts 0.5 and end at the point that predicts 0.75. Therefore, short vectors indicate a rapid increase in probability and the orthogonal projection of the row markers on the vector approximates the probability of finding high levels of methylation in the cell line.

The position of the segment, which corresponds to the point that predicts a probability of 0.5, can start any side around the origin. For example, in Figure 6, the variable DUSP22 points to the origin, when doing the orthogonal projection of the points in the direction of the vector, most of them will be projected after the reference point where the segment starts; this means that almost all cell lines of the three groups have high fitted probabilities of having high levels of methylation in that variable.

The cell lines are separated into three clearly identified clusters. In the BRCA type of cancer, variables such as NAPRT1, THY1, or ADCY4 are directed towards the positive part of dimension 1 and therefore have a greater probability of presenting high levels of methylation. The LUAD cell lines are located in the negative part of dimension 2, so these have a high propensity to present high levels of methylation in variables such as HIST1H2BH, ZNF382, and XKR6. Finally, the cell lines for the SKCM cancer type are

located in the negative part of dimension 1 and have a greater probability of presenting high levels of methylation in variables such as LOC100130522, CHRFAM7A, or DHRS4L2.

Figure 6. Logistic biplot using a Fletcher–Reeves conjugate gradient algorithm for methylation data.

Table 2 shows the rate of correct classifications for each variable using the measures of sensitivity and specificity; these measures allowed us to determine if the model exhibited a good classification for the two types of data in each variable. Sensitivity measured the true positives rate, specificity measured the true negatives rate, and the global measure corresponded to the total rate of correct classifications for each variable.

In general, the model with three dimensions and using the CG algorithm with the FR formula generated high values for sensitivity; only the GSTT1 gene presented a relatively low sensitivity, with 72% true positives. Regarding specificity, the LOC391322 gene obtained the lowest true negatives rate, with 80.9%. Thus, the results of the model are satisfactory.

Table 2. Sensitivity and specificity for each variable when fitting the LB model with $k = 3$ and a CG-FR algorithm.

Variable	Sensitivity	Specificity	Global
GSTM1	97.1	94.8	96.2
C1orf70	100.0	100.0	100.0
DNM3	100.0	94.9	96.2
COL9A2	100.0	100.0	100.0
VAR5	100.0	94.3	95.6
VAR6	100.0	88.6	91.2
THY1	100.0	95.8	96.9
VAR8	100.0	93.6	95.0
DNAH10	100.0	100.0	100.0
VAR10	100.0	90.8	92.5
DHRS4L2	100.0	88.8	91.2
ADCY4	100.0	98.3	98.8
CHRFAM7A	100.0	92.4	94.4
VAR14	100.0	99.1	99.4
FAM174B	100.0	94.8	96.2
VAR16	100.0	99.1	99.4
VAR17	100.0	86.3	87.5
ARL17A	100.0	98.6	99.4
HOXB8	100.0	98.2	98.8
LOC100130522	100.0	89.3	91.9
ZNF714	100.0	100.0	100.0
ZNF382	97.9	94.6	95.6
VAR23	100.0	96.5	97.5
FRZB	100.0	99.1	99.4
VAR25	100.0	94.7	95.0
TBX1	100.0	98.3	98.8
LOC391322	100.0	80.9	81.2
GSTT1	72.0	87.1	80.0
VAR29	100.0	96.6	97.5
FILIP1L	97.1	93.6	94.4
VAR31	100.0	97.5	98.1
HIST1H2BH	100.0	89.7	92.5
DUSP22	100.0	98.4	99.4
VAR34	95.0	97.5	96.9
XKR6	100.0	93.2	95.0
NAPRT1	96.9	87.5	89.4
VAR37	95.8	95.6	95.6
VAR38	97.7	98.3	98.1

4. Conclusions and Discussion

The Logistic Biplot (LB) model is a dimensionality reduction technique that generalizes the PCA to deal with binary variables and has the advantage of simultaneously representing individuals and variables (biplot).

In this paper, we propose and develop a methodology to estimate the parameters of the LB model using nonlinear conjugate gradient algorithms or the descending coordinate MM algorithm. For the selection of the LB model, we have incorporated a cross-validation procedure that allows the choice of the number of dimensions of the model in order to counteract overfitting.

As a complement to the proposed methods and to give practical support, a package has been written in the R language called *BiplotML* package [33], which is available from CRAN; this is a valuable tool that enables the application of the proposed algorithms to data analysis in every scientific field. Our contribution is important because we provide

alternatives to solve some problems encountered in the LB model in the presence of sparsity or a big data matrix [23,26–28]. Additionally, a procedure is presented that allows the choice of the number of dimensions, which until now had not been investigated for an LB model.

The proposed algorithms are iterative and have the property that the loss function decreases with each iteration. To study the properties of the proposed algorithms to fit a LB model, low rank data sets with $k = 3$ and different levels of sparsity were generated for $n = 100, 300, 500$ rows and $p = 50, 100$ columns. The accuracy of the algorithms was measured using the training error, generalization error (cv-error), and mean square error (RMSE) of the log-odds. According to the Monte Carlo study, we established that the cross-validation criterion is successful in the estimation of the hyperparameter of the number of dimensions. This allows the model to be specified and thus avoid overfitting; in this way, we obtain the best performance of the proposed algorithms in terms of recovering the underlying low rank structure.

The comparison of the running times showed that the algorithms converge quickly. The CG algorithm is more efficient when the matrices are sparse and not very large, while the performance of the MM algorithm is better when the number of rows and columns tends to increase; thus, it is preferable for large matrices.

Finally, we used real data on gene expression methylation to show our approach. The LB model allowed us to carry out a simultaneous projection between rows and columns, where a grouping of three classes was observed, formed by the cell lines of the three types of cancer analyzed. Furthermore, the vectors that represented the variables allowed us to identify those cell lines that were more likely to achieve high levels of methylation in the different genes.

Author Contributions: Conceptualization, J.G.B.-M. and J.L.V.-V.; methodology, J.G.B.-M. and J.L.V.-V.; software, J.G.B.-M.; validation, J.L.V.-V. and J.G.B.-M.; formal analysis, J.G.B.-M. and J.L.V.-V.; investigation, J.G.B.-M.; resources, J.L.V.-V.; data curation, J.G.B.-M.; writing—original draft preparation, writing—review and editing, J.G.B.-M. and J.L.V.-V.; visualization, J.G.B.-M.; supervision, J.L.V.-V.; funding acquisition, J.G.B.-M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Ethical review and approval were waived for this paper due to GDSC1000 being open data and completely anonymized; thus, the privacy of any patient data is not compromised.

Informed Consent Statement: Any research article describing a study involving humans should contain this statement.

Data Availability Statement: The data analyzed in this paper can be found at https://www.cancerrxgene.org/gdsc1000/GDSC1000_WebResources/Home.html (accessed on 14 May 2021).

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

Abbreviations

The following abbreviations are used in this manuscript:

LB	Logistic Biplot
MM	Majorization–Minimization
CG	Conjugate Gradient
PCA	Principal Component Analysis
BACC	Balanced Accuracy
RMSE	Relative Mean Squared Error
SVD	Singular Value Decomposition
NIPALS	Nonlinear estimation by Iterative Partial Least Square
LTA	Latent Trait Analysis
IRT	Item Response Theory

RE-WLR Rare Event Weighted Logistic Regression
 TR-IRLS Truncated Regularized Iteratively Re-weighted Least Squares

Appendix A. Derivatives

The loss function given in (5) can be expressed as $\mathcal{L}(\Theta) = \sum_{i=1}^n \sum_{j=1}^p f(\theta_{ij})$, where $\pi(\theta_{ij}) = (1 - \exp(-\theta_{ij}))^{-1}$ and $\theta_{ij} = \mu_j + \sum_{s=1}^k a_{is}b_{js} = \mu_j + \mathbf{a}_i^T \mathbf{b}_j$, thereby

$$\frac{\partial f(\theta_{ij})}{\partial a_{is}} = - \left[x_{ij} \frac{1}{\pi(\theta_{ij})} \frac{\partial \pi(\theta_{ij})}{\partial a_{is}} + (1 - x_{ij}) \frac{1}{1 - \pi(\theta_{ij})} + \frac{\partial(1 - \pi(\theta_{ij}))}{\partial a_{is}} \right] \quad (\text{A1})$$

since $\frac{\partial \pi(\theta_{ij})}{\partial a_{is}} = b_{js} \pi(\theta_{ij})(1 - \pi(\theta_{ij}))$ so that $\frac{\partial(1 - \pi(\theta_{ij}))}{\partial a_{is}} = -b_{js} \pi(\theta_{ij})(1 - \pi(\theta_{ij}))$ then

$$\frac{\partial f(\theta_{ij})}{\partial a_{is}} = b_{js}(\pi(\theta_{ij}) - x_{ij}), \quad s = 1, \dots, k. \quad (\text{A2})$$

Analogously, we obtain

$$\frac{\partial f(\theta_{ij})}{\partial b_{js}} = - \left[x_{ij} \frac{1}{\pi(\theta_{ij})} \frac{\partial \pi(\theta_{ij})}{\partial b_{js}} + (1 - x_{ij}) \frac{1}{1 - \pi(\theta_{ij})} \frac{\partial(1 - \pi(\theta_{ij}))}{\partial b_{js}} \right] \quad (\text{A3})$$

$$= a_{is}(\pi(\theta_{ij}) - x_{ij}), \quad s = 1, \dots, k, \quad (\text{A4})$$

where $\frac{\partial \pi(\theta_{ij})}{\partial b_{js}} = a_{is} \pi(\theta_{ij})(1 - \pi(\theta_{ij}))$. Finally, the offset term

$$\frac{\partial f(\theta_{ij})}{\partial \mu_j} = - \left[x_{ij} \frac{1}{\pi(\theta_{ij})} \frac{\partial \pi(\theta_{ij})}{\partial \mu_j} + (1 - x_{ij}) \frac{1}{1 - \pi(\theta_{ij})} \frac{\partial(1 - \pi(\theta_{ij}))}{\partial \mu_j} \right] \quad (\text{A5})$$

$$= (\pi_{ij} - x_{ij}). \quad (\text{A6})$$

In matrix terms, we can write it as

$$\frac{\partial \mathcal{L}}{\partial \mu} = (\Pi - \mathbf{X})^T \mathbf{1}_n; \quad \frac{\partial \mathcal{L}}{\partial \mathbf{A}} = (\Pi - \mathbf{X})\mathbf{B}; \quad \frac{\partial \mathcal{L}}{\partial \mathbf{B}} = (\Pi - \mathbf{X})^T \mathbf{A}. \quad (\text{A7})$$

References

- Keller, K. *Strategic Brand Management: Building, Measuring, and Managing Brand Equity*; Pearson/Prentice Hall: Hoboken, NJ, USA, 2008.
- Murray, D.; Pals, S.; Blitstein, J. Design and Analysis of Group-Randomized Trials: A Review of Recent Methodological Developments. *Am. J. Public Health* **2004**, *94*, 423–432. [\[CrossRef\]](#)
- Moerbeek, M.; Breukelen, G.; Berger, M. Optimal Experimental Designs for Multilevel Logistic Models. *J. R. Stat. Soc. Ser. D Stat.* **2001**, *50*, 17–30. [\[CrossRef\]](#)
- Moerbeek, M.; Maas, C. Optimal Experimental Designs for Multilevel Logistic Models with Two Binary Predictors. *Commun. Stat. Theory Methods* **2005**, *34*. [\[CrossRef\]](#)
- Iorio, F.; Knijnenburg, T.A.; Vis, D.J.; Bignell, G.R.; Menden, M.P.; Schubert, M.; Aben, N.; Gonçalves, E.; Barthorpe, S.; Lightfoot, H.; et al. A landscape of pharmacogenomic interactions in cancer. *Cell* **2016**, *166*, 740–754. [\[CrossRef\]](#) [\[PubMed\]](#)
- Collins, M.; Dasgupta, S.; Schapire, R.E. A generalization of principal components analysis to the exponential family. In *Advances in Neural Information Processing Systems 14*; The MIT Press: Cambridge, MA, USA, 2001; pp. 617–624.
- Schein, A.I.; Saul, L.K.; Ungar, L.H. A Generalized Linear Model for Principal Component Analysis of Binary Data. In Proceedings of the 9th International Workshop on Artificial Intelligence and Statistics, Key West, FL, USA, 3–6 January 2003; Volume 38.
- De Leeuw, J. Principal component analysis of binary data by iterated singular value decomposition. *Comput. Stat. Data Anal.* **2006**, *50*, 21–39. [\[CrossRef\]](#)
- Lee, S.; Huang, J.Z.; Hu, J. Sparse logistic principal components analysis for binary data. *Ann. Appl. Stat.* **2010**, *4*, 1579–1601. [\[CrossRef\]](#)
- Lee, S.; Huang, J. A coordinate descent MM algorithm for fast computation of sparse logistic PCA. *Comput. Stat. Data Anal.* **2013**, *62*, 26–38. [\[CrossRef\]](#)

11. Landgraf, A.J.; Lee, Y. Dimensionality reduction for binary data through the projection of natural parameters. *J. Multivar. Anal.* **2020**, *180*, 104668. [\[CrossRef\]](#)
12. Song, Y.; Westerhuis, J.A.; Smilde, A.K. Logistic principal component analysis via non-convex singular value thresholding. *Chemom. Intell. Lab. Syst.* **2020**, *204*, 104089. [\[CrossRef\]](#)
13. Gabriel, K.R. The biplot graphic display of matrices with application to principal component analysis 1. *Biometrika* **1971**, *58*, 453–467. [\[CrossRef\]](#)
14. Gower, J.C.; Lubbe, S.G.; Le Roux, N.J. *Understanding Biplots*; John Wiley and Sons: Hoboken, NJ, USA, 2011.
15. Scrucca, L. Graphical tools for model-based mixture discriminant analysis. *Adv. Data Anal. Classif.* **2014**, *8*, 147–165. [\[CrossRef\]](#)
16. Groenen, P.J.; Le Roux, N.J.; Gardner-Lubbe, S. Spline-based nonlinear biplots. *Adv. Data Anal. Classif.* **2015**, *9*, 219–238. [\[CrossRef\]](#)
17. Kendal, E.; Sayar, M. The stability of some spring triticale genotypes using biplot analysis. *J. Anim. Plant Sci.* **2016**, *26*, 754–765.
18. Amor-Esteban, V.; Galindo-Villardón, M.P.; García-Sánchez, I.M. A multivariate proposal for a national corporate social responsibility practices index (NCSRPI) for international settings. *Soc. Indic. Res.* **2019**, *143*, 525–560. [\[CrossRef\]](#)
19. González-García, N.; Nieto-Librero, A.B.; Vital, A.L.; Tao, H.J.; González-Tablas, M.; Otero, Á.; Galindo-Villardón, P.; Orfao, A.; Taberner, M.D. Multivariate analysis reveals differentially expressed genes among distinct subtypes of diffuse astrocytic gliomas: Diagnostic implications. *Sci. Rep.* **2020**, *10*, 1–12. [\[CrossRef\]](#)
20. Galindo Villardón, M.P. Una alternativa de representación simultánea: HJ-Biplot. *Questiio* **1986**, *10*, 13–23.
21. Gower, J.C.; Hand, D.J. *Biplots*; CRC Press: Boca Raton, FL, USA, 1995; Volume 54.
22. Greenacre, M.; Blasius, J. *Multiple Correspondence Analysis and Related Methods*; Chapman and Hall/CRC: London, UK, 2006.
23. Hernández-Sánchez, J.C.; Vicente-Villardón, J.L. Logistic biplot for nominal data. *Adv. Data Anal. Classif.* **2017**, *11*, 307–326. [\[CrossRef\]](#)
24. Cubilla-Montilla, M.; Nieto-Librero, A.B.; Galindo-Villardón, M.P.; Torres-Cubilla, C.A. Sparse HJ Biplot: A New Methodology via Elastic Net. *Mathematics* **2021**, *9*, 1298. [\[CrossRef\]](#)
25. Gabriel, K.R. Generalised Bilinear Regression. *Biometrika* **1998**, *85*, 689–700. [\[CrossRef\]](#)
26. Vicente-Villardón, J.; Galindo-Villardón, M.; Blazquez-Zaballos, A. Logistic Biplots. In *Multiple Correspondence Analysis and Related Methods*; Chapman-Hall: London, UK, 2006; Chapter 23, pp. 503–521.
27. Demey, J.; Vicente Villardón, J.L.; Galindo Villardón, M.P.; Zambrano, A. Identifying molecular markers associated with classification of genotypes by External Logistic Biplots. *Bioinformatics* **2008**, *24*, 2832–2838. [\[CrossRef\]](#) [\[PubMed\]](#)
28. Vicente-Villardón, J.L.; Hernández-Sánchez, J.C. External Logistic Biplots for Mixed Types of Data. In *Advanced Studies in Classification and Data Science*; Springer: New York, NY, USA, 2020; pp. 169–183.
29. Komarek, P.; Moore, A.W. Fast Robust Logistic Regression for Large Sparse Datasets with Binary Outputs. In Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics, Key West, FL, USA, 3–6 January 2003; pp. 163–170.
30. Lewis, J.M.; Lakshmivarahan, S.; Dhall, S. *Dynamic Data Assimilation: A Least Squares Approach*; Cambridge University Press: Cambridge, UK, 2006; Volume 13.
31. King, G.; Zeng, L. Logistic Regression in Rare Events Data. *Political Anal.* **2001**, *9*, 137–163. [\[CrossRef\]](#)
32. Maalouf, M.; Siddiqi, M. Weighted logistic regression for large-scale imbalanced and rare events data. *Knowl. Based Syst.* **2014**, *59*, 142–148. [\[CrossRef\]](#)
33. Babativa-Marquez, J.G. Package BiplotML: Biplots Estimation with Machine Learning Algorithms. Available online: <https://cran.r-project.org/package=BiplotML> (accessed on 24 June 2021).
34. R Core Team. *R: A Language and Environment for Statistical Computing*; R Foundation for Statistical Computing: Vienna, Austria, 2020.
35. Wold, H. Estimation of principal components and related models by iterative least squares. In *Multivariate Analysis*; Academic Press: New York, NY, USA, 1966; pp. 391–420.
36. Owen, A.B.; Perry, P.O. Bi-cross-validation of the SVD and the nonnegative matrix factorization. *Ann. Appl. Stat.* **2009**, *3*, 564–594. [\[CrossRef\]](#)
37. Pytlak, R. *Conjugate Gradient Algorithms in Nonconvex Optimization*; Springer Science & Business Media: Berlin, Germany, 2008; Volume 89.
38. Nocedal, J.; Wright, S. *Numerical Optimization*; Springer Science & Business Media: Berlin, Germany, 2006.
39. Fletcher, R.; Powell, M.J. A rapidly convergent descent method for minimization. *Comput. J.* **1963**, *6*, 163–168. [\[CrossRef\]](#)
40. Polak, E.; Ribiere, G. Note sur la convergence de méthodes de directions conjuguées. *ESAIM Math. Model. Numer. Anal. Model. Math. Anal. Numer.* **1969**, *3*, 35–43. [\[CrossRef\]](#)
41. Polyak, B.T. The conjugate gradient method in extremal problems. *USSR Comput. Math. Math. Phys.* **1969**, *9*, 94–112. [\[CrossRef\]](#)
42. Dai, Y.H.; Yuan, Y. A nonlinear conjugate gradient method with a strong global convergence property. *SIAM J. Optim.* **1999**, *10*, 177–182. [\[CrossRef\]](#)
43. Dai, Y.H.; Yuan, Y. An efficient hybrid conjugate gradient method for unconstrained optimization. *Ann. Oper. Res.* **2001**, *103*, 33–47. [\[CrossRef\]](#)
44. Zhang, L.; Zhou, W.; Li, D.H. A descent modified Polak–Ribière–Polyak conjugate gradient method and its global convergence. *IMA J. Numer. Anal.* **2006**, *26*, 629–640. [\[CrossRef\]](#)
45. Andrei, N. A hybrid conjugate gradient algorithm for unconstrained optimization as a convex combination of Hestenes–Stiefel and Dai–Yuan. *Stud. Inform. Control* **2008**, *17*, 57.

-
46. Yuan, G.; Zhang, M. A modified Hestenes–Stiefel conjugate gradient algorithm for large-scale optimization. *Numer. Funct. Anal. Optim.* **2013**, *34*, 914–937. [[CrossRef](#)]
 47. Liu, J.; Li, S. New hybrid conjugate gradient method for unconstrained optimization. *Appl. Math. Comput.* **2014**, *245*, 36–43. [[CrossRef](#)]
 48. Dong, X.L.; Liu, H.W.; He, Y.B.; Yang, X.M. A modified Hestenes–Stiefel conjugate gradient method with sufficient descent condition and conjugacy condition. *J. Comput. Appl. Math.* **2015**, *281*, 239–249. [[CrossRef](#)]
 49. Yuan, G.; Wei, Z.; Yang, Y. The global convergence of the Polak–Ribiere–Polyak conjugate gradient algorithm under inexact line search for nonconvex functions. *J. Comput. Appl. Math.* **2019**, *362*, 262–275. [[CrossRef](#)]
 50. Al-Baali, M. Descent property and global convergence of the Fletcher–Reeves method with inexact line search. *IMA J. Numer. Anal.* **1985**, *5*, 121–124. [[CrossRef](#)]
 51. Dai, Y.; Yuan, Y.X. Convergence properties of the Fletcher–Reeves method. *IMA J. Numer. Anal.* **1996**, *16*, 155–164. [[CrossRef](#)]
 52. Kiers, H.A. Weighted least squares fitting using ordinary least squares algorithms. *Psychometrika* **1997**, *62*, 251–266. [[CrossRef](#)]
 53. Velez, D.R.; White, B.C.; Motsinger, A.A.; Bush, W.S.; Ritchie, M.D.; Williams, S.M.; Moore, J.H. A balanced accuracy function for epistasis modeling in imbalanced datasets using multifactor dimensionality reduction. *Genet. Epidemiol. Off. Publ. Int. Genet. Epidemiol. Soc.* **2007**, *31*, 306–315. [[CrossRef](#)]
 54. Wei, Q.; Dunbrack, R.L., Jr. The role of balanced training and testing data sets for binary classifiers in bioinformatics. *PLoS ONE* **2013**, *8*, e67863. [[CrossRef](#)]
 55. Wold, S. Cross-validatory estimation of the number of components in factor and principal components models. *Technometrics* **1978**, *20*, 397–405. [[CrossRef](#)]
 56. Gabriel, K.R. Le biplot-outil d’exploration de données multidimensionnelles. *J. Soc. Fr. Stat.* **2002**, *143*, 5–55.
 57. Bro, R.; Kjelldahl, K.; Smilde, A.K.; Kiers, H. Cross-validation of component models: A critical look at current methods. *Anal. Bioanal. Chem.* **2008**, *390*, 1241–1251. [[CrossRef](#)] [[PubMed](#)]