Received: 10 January 2020 | Accepted: 28 March 2020 | Published online: 14 May 2020

DOI: 10.1002/tpg2.20021

The Plant Genome

ORIGINAL RESEARCH

Genome-based prediction of Bayesian linear and non-linear regression models for ordinal data

Paulino Pérez-Rodríguez[1] | **Samuel Flores-Galarza**[1] | **Humberto Vaquera-Huerta**[1] | **David Hebert del Valle-Paniagua**[1] | **Osval A. Montesinos-López**[3] | **José Crossa**[1,2]

[1]Colegio de Postgraduados, CP 56230, Montecillos, Edo. de México

[2]Biometrics and Statistics Unit, International Maize and Wheat Improvement Center (CIMMYT), Apdo. Postal 6–641, 06600, Cd. de México

[3]Facultad de Telemática, Universidad de Colima, Colima, 28040, México

Correspondence
José Crossa, Maize and Wheat Improvement Center (CIMMYT), Apdo. Postal 6–641, 06600, Cd. de México and Colegio de Postgraduados, CP 56230, Montecillos, Edo. de México.
Email: j.crossa@cgiar.org

Abstract

Linear and non-linear models used in applications of genomic selection (GS) can fit different types of responses (e.g., continuous, ordinal, binary). In recent years, several genomic-enabled prediction models have been developed for predicting complex traits in genomic-assisted animal and plant breeding. These models include linear, non-linear and non-parametric models, mostly for continuous responses and less frequently for categorical responses. Several linear and non-linear models are special cases of a more general family of statistical models known as artificial neural networks, which provide better prediction ability than other models. In this paper, we propose a Bayesian Regularized Neural Network (BRNNO) for modelling ordinal data. The proposed model was fitted using a Bayesian framework; we used the data augmentation algorithm to facilitate computations. The proposed model was fitted using the Gibbs Maximum a Posteriori and Generalized EM algorithm implemented by combining code written in C and R programming languages. The new model was tested with two real maize datasets evaluated for Septoria and GLS diseases and was compared with the Bayesian Ordered Probit Model (BOPM). Results indicated that the BRNNO model performed better in terms of genomic-based prediction than the BOPM model.

1 | INTRODUCTION

In genomic selection (GS) and genomic prediction (GP), all markers are fitted simultaneously to the training and testing populations. Practical and simulation studies have favored the use of GS as a way to increase genetic gains in less time (Crossa et al., 2017). For continuous traits, models have been developed to regress phenotypes on all markers using a linear model (Meuwissen, Hayes, & Goddard, 2001). However, in plant and animal breeding, the response variables in many traits are discrete counts ($y = 0, 1, 2, \ldots$). In classical probability theory, counts could indicate a binomial, multinomial or a Poisson distribution. For smaller counts, data analysts recommend using logarithmic or square root transformations. In GS it is still common practice to apply linear regression models to categorical data or transformed data (Montesinos-López

Abbreviations: BGLR, Bayesian Generalized Linear Regression; BOPM, Bayesian ordered probit model; BRNNO, Bayesian Regularized Neural Network for Ordinal data; BS, Brier Score; GBS, Genotype by Sequencing; GEM, Generalized EM algorithm; GLS, Gray Leaf Spot; GMAP, Gibbs Maximum A Posteriori; GS, Genomic Selection; MAE, Mean Absolute Error; MCMC, Markov Chain Monte Carlo; MER, Misclassification Error Rate; RMSE, Root Mean Square Error; SHL, Single Hidden Layer; SHLNN, Single Hidden Layer Neural Network.

et al., 2015a). Transformations do not consider many distributions (of count data) as positively skewed: several observations have a value of 0, and a high number of 0 values in the data does not allow a skewed distribution to be transformed into a normal distribution; it is also possible that the regression model will produce negative predicted values. When a transformation is used, it is not always possible to have normally distributed transformed data, and often times, transformations are counterproductive (Stroup, 2012).

Ordinal data are very common in many research fields, such as econometrics, finance, agriculture, animal and plant breeding, genetics, etc. In agricultural applications, resistance/susceptibility to diseases is usually measured on an ordinal scale (e.g., 1 = no disease, 2 = low infection, 3 = moderate infection, 4 = high infection, and 5 = totally infected). With data on this scale, it is possible to obtain relative or absolute frequencies, but the usual sample mean or sample standard deviation cannot be obtained, since the distance between classes is not defined (Stevens, 1946). An important application with ordinal data is ordinal regression, where a response variable that is measured on an ordinal scale is predicted by using several covariates. Ordinal regression has been widely used in plant and animal breeding (e.g., Gianola, 1982) and is mainly based on linear mixed models. However, ignoring the ordinal nature of the data can cause several problems related to biased estimates, which could potentially lead to wrong and misleading conclusions (Montesinos-López et al., 2015a). In plant breeding, several economically important traits are categorical, and in general, threshold models have not been considered in GS. One of the first studies that introduced the threshold model to GS incorporating genomic × environment interaction was Montesinos-López et al. (2015a) when 278 maize lines scored for resistance to gray leaf spot (GLS) were measured using an ordered categorical scale in three environments.

In GS, a response variable is predicted based on dense molecular markers. Most of the models used to predict the response variable are linear, and the following assumptions hold: (1) the response variable is normally distributed, (2) the variance of the residuals is constant, and (3) the expected value of the response variable is a linear function of covariates (Montesinos-López et al., 2015a; Pérez-Rodríguez et al., 2012). Recently, linear models used in GS have been extended to model ordered phenotypical categories (e.g., Montesinos-López et al., 2015a; Montesinos-López, Montesinos-López, Crossa, Burgueño, & Eskridge, 2015b; Wang et al., 2013).

In order to make predictions on new data, machine learning builds models from sample data using several algorithms (Samuel, 1959). There are some applications of machine learning in genomic selection (González-Camacho et al., 2012; González-Camacho et al., 2018) with connected network architecture, which is a type of machine learning algorithm that uses an artificial neural network with input, hidden and output layers linked non-linearly. The layers consist of stages of non-linear data transformations. Non-linear kernel-based models and neural networks have also been used in the context of GS for continuous traits with promising results (González-Camacho et al., 2012, 2018; Pérez-Rodríguez et al., 2012). Recently, Montesinos-López et al. (2019) compared genome-based prediction accuracies for ordinal traits using several deep machine learning methods and the threshold model for ordinal data on an extensive number of ordinal data; the authors found that the threshold genomic best linear unbiased prediction was the most consistent model in terms of genomic-enabled prediction accuracy. However, the scientific literature does not show results on many non-linear GS models and methods for categorical data.

Core Ideas

- Neural networks are universal approximation machines that can be adapted to predict ordinal responses
- Neural networks outperform the predictive power of linear models
- Data augmentation is a valuable tool for fitting neural network models

Based on the previous considerations, and motivated by the fact that (1) artificial neural networks are considered universal approximations, which in some cases exhibit better predictive ability than linear and non-linear models, and that (2) several traits commonly used in GS are measured on an ordinal scale, the main objective of this study was to introduce a Bayesian Regularized Neural Network for Ordinal data (BRNNO) with a Single Hidden Layer (SHL). We used a data augmentation algorithm (Tanner, 1993) to make part of the computations feasible. We assumed that the process that leads to the observed response variable is a latent variable (unobserved), which is a continuous random variable that follows a standard normal distribution (e.g., Albert & Chib, 1993; Gianola, 1982; Montesinos-López et al., 2015a). We combined Markov Chain Monte Carlo (MCMC) simulation and EM with other optimization algorithms that allowed us to fit the proposed model. The BRNNO model can also be considered a generalization of the probit ordered regression in the context of non-linear models and can also be extended to generalize the logit ordered regression model (Montesinos-López et al., 2015b).

This paper is organized as follows: In the Materials and Methods section we introduce the Bayesian ordered probit model (BOPM) and the Bayesian ordered logit model (BOLM) for ordinal data; next, we introduce the Bayesian Regularized Neural Network for Ordinal data (BRNNO) and describe the procedures used to assess the predictive power of the proposed model. Then, we present an application of the

BRNNO using two real datasets for wheat diseases from CIMMYT maize and wheat breeding trials (https://www.cimmyt.org). Finally, we present the results and discussion.

2 | MATERIALS AND METHODS

2.1 | Statistical models

According to Albert and Chib (1993), suppose that $Y_1, \ldots, Y_n$ are observed and Y_i can take observations on K ordered values, that is, $Y_i \in \{1, \ldots, K\}$, $1 < 2 < \cdots < K$. Furthermore, suppose that we are interested in modelling the probabilities, $p_{ij} = P(Y_i = j), j = 1, \ldots, K$. Probabilities p_{ij} can be modelled using covariates $\boldsymbol{x}_i^t = (x_{i1}, \ldots, x_{ip})$ with regression techniques, which leads to several very well-known statistical models, e.g., the probit model or the ordered logit model, among others. Below we briefly discuss these two models from a Bayesian perspective.

2.2 | Linear models: Bayesian ordered probit model, Bayesian ordered logistic model

2.2.1 | Bayesian ordered probit model (BOPM)

Albert and Chib (1993) assume that there is a unobserved random variable Z_i normally distributed with mean $\boldsymbol{x}_i^t\boldsymbol{\beta}$ and variance 1, where $\boldsymbol{x}_i^t = (x_{i1}, \ldots, x_{ip})$, a p-dimensional vector of input covariates and $\boldsymbol{\beta}$ is a vector of unknown regression coefficients of dimensions $p \times 1$. We observe that $Y_i = j$ if $\lambda_{j-1} < Z_i < \lambda_j$ with $\lambda_0 = -\infty$, $\lambda_K = \infty$, $\lambda_1, \ldots, \lambda_{K-1}$ are a set of unknown thresholds; note also that $\lambda_0 \leq \lambda_1 \leq \cdots \leq \lambda_K$. The latent variable Z_i, also known as liability (Gianola, 1982), can be represented using the regression framework, that is:

$$Z_i = \boldsymbol{x}_i^t\,\boldsymbol{\beta} + e_i \tag{1}$$

where e_i is a random normal variable with mean 0 and variance 1. Using this representation, it can be shown that:

$$P\left(Y_i = j\right) = \Phi\left(\boldsymbol{x}_i^t\boldsymbol{\beta} - \lambda_j\right) - \Phi\left(\boldsymbol{x}_i^t\boldsymbol{\beta} - \lambda_{j-1}\right) \tag{2}$$

$$P\left(Y_i \leq j\right) = \Phi\left(\lambda_j - \boldsymbol{x}_i^t\boldsymbol{\beta}\right) \tag{3}$$

with $\Phi(\cdot)$ as the standard normal cumulative distribution. Albert and Chib (1993) assigned a diffuse prior for $(\boldsymbol{\beta}, \boldsymbol{\lambda})$ and showed that all conditional distributions necessary to implement the Gibbs Sampler algorithm (Geman & Geman, 1984) have closed form and can be used to draw samples from the posterior distribution $p(\boldsymbol{\beta}, \boldsymbol{\lambda}|\text{data})$.

2.2.2 | Bayesian ordered logit model (BOLM)

Montesinos-López et al. (2015b) argue that the ordinal logistic regression model is often preferred over the ordinal probit model because it provides regression coefficients that can be more easily interpreted as the regression coefficients are connected with the odds ratio. Montesinos-López et al. (2015b) proposed using a latent variable (liability) with logistic distribution, which can be represented as follows:

$$Z_i = \boldsymbol{x}_i^t\,\boldsymbol{\beta} + \tilde{e}_i \tag{4}$$

where $\tilde{e}_i$ is the standard logistic random variable; therefore, the probabilities $P(Y_i = j)$ are computed based on the distribution function of a logistic random variable. The authors propose using Póyla-Gamma data augmentation, which facilitates the computations when sampling from the posterior distribution of the parameters of interest with the Gibbs sampler (Geman & Geman, 1984).

2.3 | The proposed non-linear model: Bayesian Regularized Neural Networks for Ordinal data (BRNNO)

Following Albert and Chib (1993), Gianola (1982), González-Camacho et al. (2012) and Pérez-Rodríguez, Gianola, Weigel, Rosa, and Crossa (2013), in this study, we propose using the latent variable approach in order to model ordinal data with a Single Hidden Layer Neural Network (SHLNN), as shown in Figure 1. The structure of the neural network was analogous to that shown in Pérez-Rodríguez et al. (2013), that is, the input layer has the input variables $\boldsymbol{x}_i^t = (x_{i1}, \ldots, x_{ip})$, which were then combined with linear inputs in neuron $m = 1, \ldots, S$ and transformed by applying a non-linear activation function $g(\cdot)$ in the hidden layer and then combined linearly in the output layer to regress the latent variable, that is:

$$Z_i = \sum_{m=1}^{s} w_m\, g\left(b_m + \boldsymbol{x}_i^t\boldsymbol{\beta}^{[m]}\right) + e_i \tag{5}$$

The activation function is a function that maps the inputs from the real line into the bounded open interval $(-1,1)$, for example, the tangent hyperbolic function $\tanh(\cdot)$ is given by $g\ (u) = \frac{2}{1+\exp(-2u)} - 1$. There are several options for activation functions, but $\tanh(\cdot)$ is a popular alternative implemented in most software packages that fits neural networks. The output from equation (5) is related to the observed data using the approach employed in the probit and logit ordered model, that is, $Y_i = j$ if $\lambda_{j-1} < Z_i < \lambda_j$ with $\lambda_0 = -\infty$, $\lambda_K = \infty$, $\lambda_1, \ldots, \lambda_{K-1}$ the set of unknown thresholds. By using this approach, the probabilities given in equations (2) and (3) can

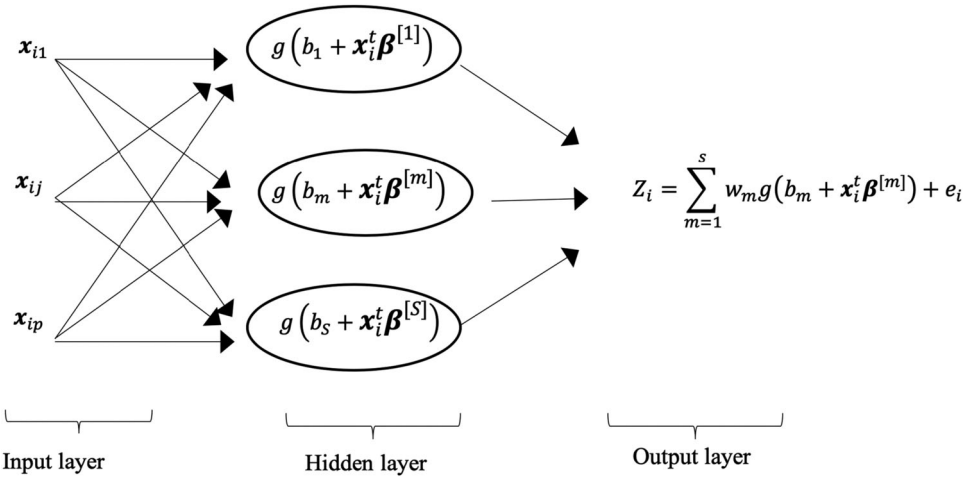

FIGURE 1 Structure of a single-layer feed forward neural network (SLNN) adapted from González-Camacho et al. (2012) and Pérez-Rodríguez et al. (2013) for ordinal data using the latent variable approach

be computed as follows:

$$P\left(Y_i = j\right) = \Phi\left(\sum_{m=1}^{s} w_m g\left(b_m + \boldsymbol{x}_i^t\boldsymbol{\beta}^{[m]}\right) - \lambda_j\right) - \Phi\left(\sum_{m=1}^{s} w_m g\left(b_m + \boldsymbol{x}_i^t\boldsymbol{\beta}^{[m]}\right) - \lambda_{j-1}\right) \tag{6}$$

$$P\left(Y_i \leq j\right) = \Phi\left(\lambda_j - \sum_{m=1}^{s} w_m g\left(b_m + \boldsymbol{x}_i^t\boldsymbol{\beta}^{[m]}\right)\right) \tag{7}$$

Note that the ordered probit model (Albert & Chib, 1993) is a special case of the single layer perceptron, which also is a special case of the proposed Single Layer Neural Network, which is obtained by setting the number of neurons equal to 1 ($S = 1$), $w_1 = 1$, $g(u) = u$ (identity function), $b_1 = 0$, $\boldsymbol{\beta} = \boldsymbol{\beta}^{[1]}$. A Single Hidden Layer Neural Network that generalizes the ordered logit model proposed by Montesinos-López et al. (2015b) can be obtained replacing the random error term in equation (5) by a random error term with standard logistic distribution.

The *BRNNO* model can be fitted using the Generalized Expectation-Maximization algorithm (Kärkkäinen & Sillanpää, 2013; Neal & Hinton, 1998) or by using Bayesian methods. Let $\boldsymbol{\theta} = (w_1, \ldots, w_S; b_1, \ldots, b_S; \boldsymbol{\beta}^{[1]}, \ldots, \boldsymbol{\beta}^{[S]})$, the vector of weights, biases and connection strengths, and let $p(\boldsymbol{\theta}|\sigma^2_{\boldsymbol{\theta}})$ be the prior distribution assigned to the elements of this vector. Here we assume that $p\left(\boldsymbol{\theta}|\sigma^2_{\boldsymbol{\theta}}\right) = MN(\mathbf{0}, \sigma^2_{\boldsymbol{\theta}}\boldsymbol{I})$, where $MN(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ stands for multivariate normal distribution with mean $\boldsymbol{\mu}$ and variance-covariance matrix $\boldsymbol{\Sigma}$; in this case, $\boldsymbol{\mu} = \mathbf{0}$ and $\boldsymbol{\Sigma} = \sigma^2_{\boldsymbol{\theta}}\boldsymbol{I}$, with $\sigma^2_{\boldsymbol{\theta}}$ a variance parameter common to all the elements in $\boldsymbol{\theta}$, assuming for the moment that $\sigma^2_{\boldsymbol{\theta}}$ is known. In the data augmentation framework for ordinal data (e.g., Albert & Chib, 1993), the joint posterior distribution of $\{\boldsymbol{\theta}, \lambda_1, \ldots, \lambda_{K-1}, Z_1, \ldots, Z_n\}$, given the observed data, can be obtained as follows:

$$\begin{aligned} p\left(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{Z}|\boldsymbol{y}\right) &\propto p\left(\boldsymbol{y}|\boldsymbol{Z}, \boldsymbol{\lambda}, \boldsymbol{\theta}\right) p\left(\boldsymbol{Z}|\boldsymbol{\lambda}, \boldsymbol{\theta}\right) p\left(\boldsymbol{\theta}, \boldsymbol{\lambda}|\sigma^2_{\boldsymbol{\theta}}\right) \\ &= p\left(\boldsymbol{y}|\boldsymbol{Z}, \boldsymbol{\lambda}\right) p\left(\boldsymbol{Z}|\boldsymbol{\theta}\right) p\left(\boldsymbol{\theta}|\sigma^2_{\boldsymbol{\theta}}\right) p\left(\boldsymbol{\lambda}\right) \end{aligned}$$

By assigning a diffuse prior for $p(\boldsymbol{\lambda})$, this leads to:

$$p\left(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{Z}|\boldsymbol{y}\right) \propto \prod_{i=1}^{n}\left[\phi\left(z_i; \sum_{m=1}^{s} w_m g\left(b_m + \boldsymbol{x}_i^t\boldsymbol{\beta}^{[m]}\right), 1\right) \times \sum_{j=1}^{K} 1\left(Y_i = j\right) 1\left(\lambda_{j-1} < Z_i \leq \lambda_j\right)\right] \times p(\boldsymbol{\theta}|\sigma^2_{\theta}) p\left(\boldsymbol{\lambda}\right) \tag{8}$$

where $\phi(;\mu,\sigma^2)$ is the density function of a normal random variable with mean μ and variance σ^2, and $1(X \in A)$ is the indicator function that takes a value of 1 if X is contained in the A set, and a value of 0 otherwise. Note that the joint posterior distribution of the parameters of interest does not have a closed form, so we must use numerical algorithms and simulation techniques to fit this model. In order to ensure the identification of the $\boldsymbol{\lambda}$ parameters, λ_1 is set to 0 (Albert & Chib, 1993).

Next, we describe two numerical algorithms that can be employed to fit the BRNNO model: GMAP and GEM (Generalized EM algorithm). The GMAP algorithm (Kim, Hall, & Li, 2009) combines the Gibbs Sampler algorithm (G = Gibbs; see Geman & Geman, 1984) and an optimization algorithm to obtain the posterior modes of some parameters of interest in the model (MAP = Maximum A Posteriori). Pursuant to Kim et al. (2009), it is necessary to divide the parameters into

two parts, which are labeled as 'tractable' and 'intractable'. Tractable parameters are those whose full conditional distributions have a closed form, whereas the remaining parameters are 'intractable'. Thus, 'tractable' parameters can be sampled by using the Gibbs sampler and 'intractable' parameters are estimated by MAP and G; MAP steps are repeated until convergence. As for the problem of interest, the tractable parameters are the latent variables and the thresholds $\{\boldsymbol{Z}, \boldsymbol{\lambda}\}$, while the intractable parameters are the weights, biases, connection strengths and variance parameter $\sigma^2_{\boldsymbol{\theta}}$, that is $\{\boldsymbol{\theta}, \sigma^2_{\boldsymbol{\theta}}\}$.

Next, we derive the full conditional distributions for tractable parameters and briefly explain how to obtain the posterior modes of intractable parameters. Appendix 1 shows computational details for implementing the GMAP algorithm.

a. ***Conditional distributions of 'tractable' parameters***

a.1. *Conditional distribution of latent variables,* $p(Z_i|\, Y_i = j, \text{else})$, $i = 1, \ldots, n$.

The fully conditional distribution of the latent variables can be obtained from (8); it has closed forms and can be obtained as follows:

$$p\left(Z_i|\, Y_i = j,\, else\right) \propto \phi\left(Z_i;\, \sum_{m=1}^{s} w_m g\left(b_m + \boldsymbol{x}_i^t \boldsymbol{\beta}^{[m]}\right), 1\right) \times \sum_{j=1}^{K} 1\left(Y_i = j\right) 1\left(\lambda_{j-1} < Z_i \leq \lambda_j\right), \quad (9)$$

which corresponds to a truncated normal random variable truncated at the left (right) by λ_{j-1} (λ_j), location parameter $\sum_{m=1}^{s} w_m g(b_m + \boldsymbol{x}_i^t \boldsymbol{\beta}^{[m]})$ and scale parameter 1.

a.2. Conditional distribution of thresholds, $p(\boldsymbol{\lambda}|\text{else})$

The fully conditional distribution of $\lambda_j|else$ is proportional to:

$$p\left(\lambda_j|else\right) \propto \prod_{i=1}^{n} \left[1\left(Y_i = j\right) 1\left(\lambda_{j-1} < Z_i < \lambda_j\right) + 1\left(Y_i = j+1\right) 1\left(\lambda_j < Z_i < \lambda_{j+1}\right)\right], \quad j = 2, .., K, \quad (10)$$

which corresponds to a uniform distribution in the $[a, b]$ interval, where $a = \max\{\max\{Z_i : Y_i = j\}, \lambda_{j-1}\}$, $b = \min\{\min\{Z_i : Y_i = j+1\}, \lambda_{j+1}\}$; see Albert and Chib (1993) for more details.

a. ***Conditional posterior modes of 'intractable' parameters***

The set of intractable parameters is $\{\boldsymbol{\theta}, \sigma^2_{\boldsymbol{\theta}}\}$. From equation (5), and by taking into account the prior distribution assigned to $\boldsymbol{\theta}$, it is clear that the problem reduces to fitting a Bayesian Regularized Neural Network with a Single Hidden Layer; the details of the algorithm can be found elsewhere, such as in Foresee and Hagan (1997), Gianola, Okut, Weigel, and Rosa (2011), Okut, Gianola, Rosa, and Weigel (2011), and Pérez-Rodríguez et al. (2013), among others. The algorithms used to fit the Bayesian Regularized Neural Network need to be slightly modified to take into account that the residual variance for e_i in equation (5) is set to 1. The Appendix shows the computational details for estimating posterior modes for these parameters. One interesting output from this algorithm is the effective number of parameters η which can give some guidance about the number of neurons to include in the neural network (see, for example, Pérez-Rodríguez et al., 2013). This parameter is computed as follows (see Appendix 1 for more details):

$$\eta = p - 2\alpha Trace\left(\boldsymbol{H}^{-1}\right),$$

where p is the number of elements in $\boldsymbol{\theta}$, $\alpha = \frac{1}{2\sigma^2_{\boldsymbol{\theta}}}$, $\boldsymbol{H} = \frac{\partial^2}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}'} F(\boldsymbol{\theta})$, $F(\boldsymbol{\theta}) = \frac{1}{2\sigma^2_e} \sum_{i=1}^{n} (z_i - \hat{z}_i)^2 + \frac{1}{2\sigma^2_{\boldsymbol{\theta}}} \sum_{l=1}^{p} \theta_l^2$. When this parameter shows little change for S and S+1 neurons, the more parsimonious model is preferred.

The GEM algorithm (Kärkkäinen & Sillanpää, 2013; Neal & Hinton, 1998) is similar to the GMAP algorithm. The 'tractable' parameters (latent variables and thresholds) are updated according to the conditional expectations for distributions given in (9) and (10) and the 'intractable' parameters are estimated using MAP, so the updating of 'tractable' and 'untractable' parameters is repeated until convergence. As the number of parameters to estimate increases, the GMAP algorithm becomes slow, and therefore, our preferred algorithm for fitting the proposed model is GEM. Appendix 2 shows computational details for implementing the GEM algorithm.

2.4 | The software

The ordered probit model described above can be fitted using the BGLR library functions in R (Pérez & de los Campos, 2014). The ordered logit model proposed by Montesinos-López et al. (2015b) can also be fitted in R (R Core Team, 2019); the authors provided the code in the supplementary materials included in the published paper. The GEM algorithm for fitting the ordinal Bayesian Regularized Neural Network proposed in this study was implemented using the C programming language (Kernighan & Ritchie, 1988) and the R statistical package (R Core Team, 2019) to speed up computations and facilitate user interaction with the software. We included the resulting routines for fitting the proposed model in the brnn package version 0.8 with the GEM algorithm (Pérez-Rodríguez & Gianola, 2020), while

the code for fitting the model with the GMAP algorithm is included in supplementary materials. We decided not to include it in the brnn package, as it is very slow.

2.5 | Measurements of genome-based prediction accuracy of models for ordinal data

In order to evaluate the proposed model's predictive ability, we proposed partitioning the data at random into the training and testing sets. The basic idea was to fit the model with the training data and then predict phenotypes (observed ordinal response) for the individuals in the testing set. In the case of continuous response, the model's predictive ability was measured by Pearson's correlation coefficient between observed and predicted phenotypical values or mean squared error prediction (MSEP).

However, in the case of categorical data, several metrics are widely used, such as sensitivity, specificity, ROC curve, etc. (Tharwat, 2018). Montesinos-López et al. (2015b) suggested using the Brier score (BS) (Brier, 1950) because these authors argued that this statistic uses all the information contained in the predictive distribution. The Brier score can be computed as follows:

$$BS = \frac{1}{2n}\sum_{i=1}^{n}\sum_{j=1}^{K}\left(\hat{\pi}_{ij} - d_{ij}\right)^2$$

where n is the total number of observations in the set of interest, $\hat{\pi}_{ij}$ is the estimated probability that observation $i = 1,\ldots,n$ belongs to class $j = 1,\ldots,K$, which can be computed using the adjusted model with the equations (6)-(7) and $d_{ij} = 1$, if observation i belongs to class j and $d_{ij} = 0$ otherwise. As defined above, BS takes values between 0 and 1, and small values are associated with better prediction ability.

Other genome-based prediction performance measures that are used with ordinal data are, for example, the misclassification error rate (MER), which is obtained by counting the number of classification errors and then dividing the result by the number of test cases. Mathieson (1995) argued that for ordinal responses, performance measures that take into account the difference between class numbers are preferred over MER due to the lack of better information. In this case, two performance measures that take into account this criterion are Mean Absolute Error (MAE) computed as $\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$ and Root Mean Square Error (RMSE), which can be obtained as $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$ (see da Costa & Cardoso, 2005, for more details). Pearson's correlation coefficient, widely used with continuous data, can be replaced by Sperman's rank correlation coefficient or Kendall's tau coefficient; here, we used Sperman's coefficient. It is important to note that the predicted class $\hat{y}_i$ in the testing set is obtained by first predicting the value of the latent variable $\hat{z}_i$ using the estimated parameters and associated covariates, and then $\hat{y}_i = j$ if $\hat{\lambda}_{j-1} < \hat{z}_i < \hat{\lambda}_j$, where $\hat{\boldsymbol{\lambda}}$ is the vector of estimated threshold parameters.

2.6 | Applying models to real data

2.6.1 | The Septoria dataset

Septoria is a fungus that causes leaf spot diseases in field crops, forage crops and vegetables. The Septoria dataset includes information for 268 wheat lines from CIMMYT (http://www.cimmyt.org) and was previously analyzed by Montesinos-López et al. (2015b). The lines were planted in 2010 in Toluca, Mexico. The dataset includes information about disease severity, which was measured on an ordinal scale with four points. The lines were genotyped using Genotype by Sequencing (GBS; Poland et al., 2012). Markers were filtered by removing markers with more than 50% missing values, imputed using observed allelic frequencies, and further removing markers with minor allele frequency smaller than 0.05. After that, 6787 markers were still available for further analysis. The dataset can be downloaded from http://hdl.handle.net/11529/10254.

2.6.2 | The GLS dataset

Gray Leaf Spot (GLS) is a disease caused by the fungus *Cercospora zeae-maydis*. This dataset consists of genotypic and phenotypic information for 278 maize lines from the Drought Tolerance Maize (DTMA) project of CIMMYT's Global Maize Program. The dataset was originally analyzed by Crossa et al. (2011), and re-analyzed later by González-Camacho et al. (2012), Montesinos-López et al. (2015b) and Pérez-Rodríguez et al. (2018) using different statistical models. The dataset includes information on disease severity measured on an ordinal scale with 5 points: 1 = no disease, 2 = low infection, 3 = moderate infection, 4 = high infection and 5 = totally infected. The lines were initially genotyped with 1,152 SNPs and re-genotyped later with 55k SNPs using the Illumina platform. After removing SNPs with more than 10% missing values and imputing filtering markers with minor allele frequency smaller than 0.05, a total of 46,347 markers were still available for further analysis. The dataset can be downloaded from http://hdl.handle.net/11529/10254.

We evaluated the predictive ability of the proposed model by using 100 partitions generated at random with 90% of observations in the training set and the remaining 10% in the testing set. For each partition, we fitted the Bayesian Regularized Neural Network (BRNN) for ordinal data with two neurons. In order to expedite computations, we first computed a genomic relationship matrix (**G**) between centered

TABLE 1 Performance measures (BS, MAE, RMSE, r and MER) for the Bayesian Ordered Probit Model (BOPM) and Bayesian Regularized Neural Network for Ordinal data (BRNNO) for Septoria and GLS at three locations (Colombia, Harare and Mexico). Standard deviation (sd). GMAP denotes the Gibbs Maximum a Posteriori Algorithm and GEM is the Generalized EM algorithm

Model	Dataset/Location	Statistic	BS	MAE	RMSE	r	MER
BOPM	Septoria	Mean	0.3205	0.6138	0.8926	0.2742	0.5173
		Sd	0.0287	0.1429	0.1507	0.1650	0.1065
	GLS Colombia	Mean	0.3550	0.6741	0.9301	0.6963	0.5804
		Sd	0.0083	0.0719	0.0770	0.0629	0.0503
	GLS Harare	Mean	0.3458	0.6772	0.9342	0.4335	0.5792
		Sd	0.0070	0.0495	0.0440	0.0566	0.0398
	GLS Mexico	Mean	0.3359	0.6998	0.9308	0.4141	0.6142
		Sd	0.0167	0.0851	0.0822	0.0887	0.0640
BRNNO GMAP	Septoria	Mean	0.3562	0.6197	0.8961	0.3078	0.5256
		Sd	0.0518	0.1173	0.1376	0.1703	0.0839
	GLS Colombia	Mean	0.3299	0.5878	0.8734	0.7241	0.5061
		Sd	0.0291	0.0800	0.0967	0.0621	0.0549
	GLS Harare	Mean	0.3340	0.6510	0.9311	0.4971	0.5464
		Sd	0.0131	0.0536	0.0587	0.0694	0.0393
	GLS Mexico	Mean	0.3418	0.6063	0.9087	0.5541	0.5000
		Sd	0.0424	0.1024	0.1163	0.1160	0.0667
BRNNO GEM	Septoria	Mean	0.3355	0.5927	0.8742	0.3176	0.5023
		Sd	0.0513	0.1363	0.1474	0.2059	0.0991
	GLS Colombia	Mean	0.3279	0.5745	0.8683	0.7295	0.4942
		Sd	0.0162	0.0768	0.0935	0.0644	0.0564
	GLS Harare	Mean	0.3430	0.6574	0.9355	0.4916	0.5533
		Sd	0.0071	0.0508	0.0495	0.0616	0.0387
	GLS Mexico	Mean	0.3431	0.5910	0.8704	0.5503	0.5088
		Sd	0.0105	0.0978	0.1014	0.1021	0.0781

BS = Brier Score (the smaller the better); MAE = Mean Absolute Error (the smaller, the better); RMSE = Root Mean Square Error (the smaller, the better); r = Spearman correlation coefficient (the higher, the better); MER = Misclassification Error Rate (the smaller, the better).

TABLE 2 Number of parameters to estimate and the estimated effective number of parameters and corresponding standard deviation

Dataset	Number of parameters to estimate (biases, weights and connection strengths)	Effective number of parameters GMAP algorithm	Effective number of parameters GEM algorithm
Septoria	538	211.28 (4.33)	198.19 (2.27)
GLS Colombia	558	259.23 (1.90)	231.00 (4.63)
GLS Harare	516	237.31 (1.44)	221.63 (2.64)
GLS Mexico	524	230.48 (3.58)	159.76 (5.80)

and standardized genotypes (Lopez-Cruz et al., 2015); we then performed the eigen-value decomposition of $\mathbf{G}$, that is, $\mathbf{G} = \mathbf{\Gamma\Lambda\Gamma}'$, and used $\mathbf{\Gamma\Lambda}^{\frac{1}{2}}$ (principal components) as our matrix of covariates to fit the models (see Gianola et al., 2011, for other computing strategies). The model was fitted using the GMAP and GEM algorithms. In the case of GMAP, inferences were based on 5000 MCMC iterations obtained after discarding 5000 samples that were taken as burn-in, due to high computational times.

In the case of ordered probit regression, we used the same computational strategy as before for speeding up computations. We fitted the model using the BGLR package in R (Pérez & de los Campos, 2014), and inferences were based on 5000 MCMC iterations obtained after discarding 25,000 samples that were taken as burn-in. For each partition, we computed several statistics that allowed us to assess the performance of the models; in this case, Brier score, MAE, RMSE, Spearman's correlation (r) and MER.

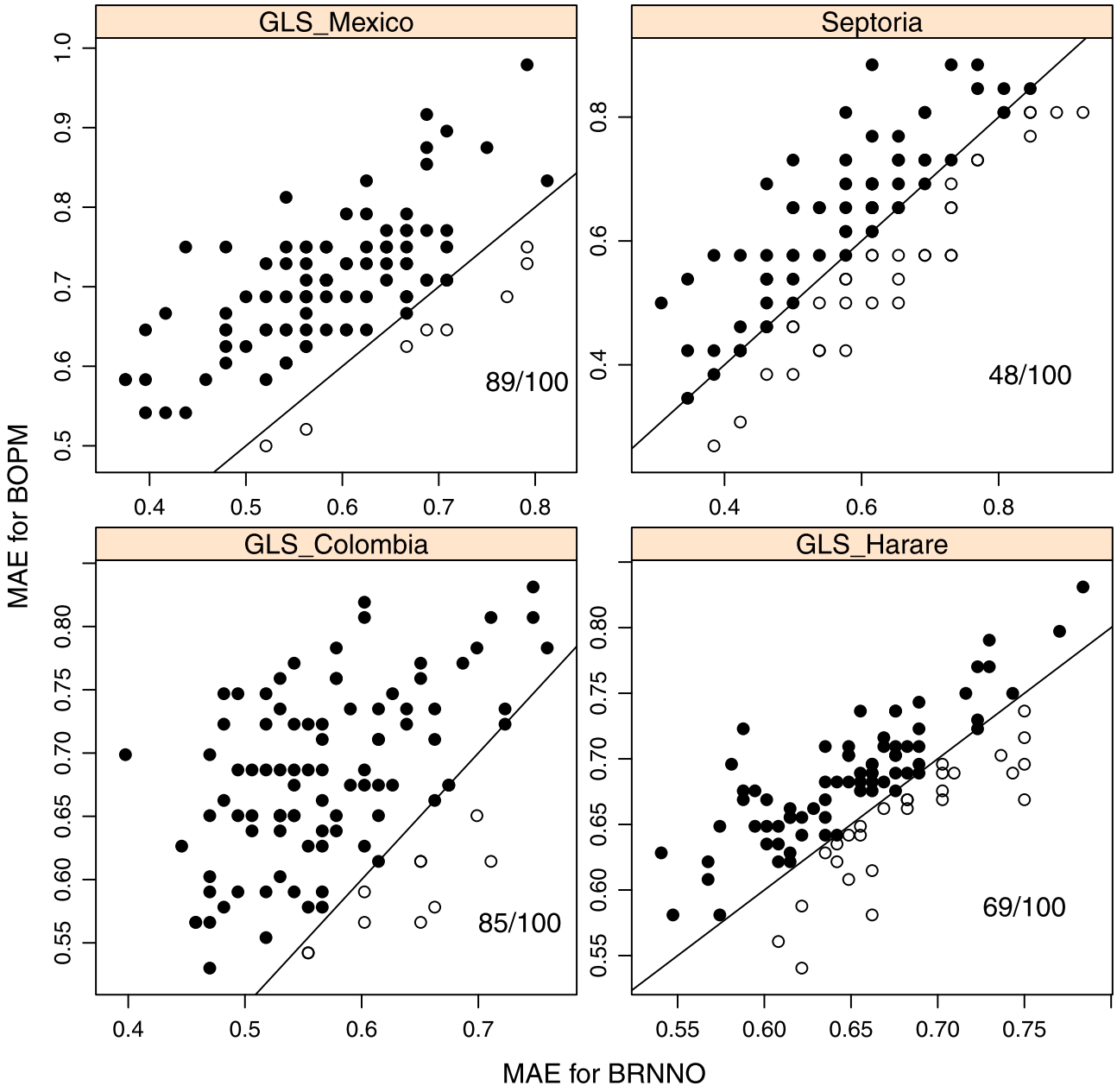

FIGURE 2 Scatter plot matrix of MAE for BRNNO and BOPM. Points above the 45° degree line are associated with the worst performance of the model on the y-axis (BOPM) in comparison with the model presented on the x-axis (BRNNO). When the worst model is BOPM, this is represented by a filled circle; however, when the worst model is BRNNO, it is represented by an open circle. The plots also show the number of times that BOPM is inferior to BRNNO as %

3 | RESULTS

Table 1 shows the results for Septoria and GLS in Colombia, Harare and Mexico. Results in Table 1 show the five criteria for determining the best models for all of them (except the correlations), i.e., the lower the better, whereas for the correlations, the higher the better. Results show that the BRNNO model performed better than the probit model (BOPM), except in the case of BS criterion for the Septoria disease (BS = 0.3205). For all the other cases, the performance measured by criteria MAE, RMSE and MER was always smaller for BRNNO than for the ordered probit model (BOPM). In addition, it should be noted that the associated Spearman's rank correlation coefficient is higher for BRNNO than for the probit model (BOPM). The best prediction of the BRNNO algorithm occurred for disease GLS measured in Colombia using the GEM algorithm with 0.5745, 0.8683, 0.7295, and 0.4942 values for criteria MAE, RMSE, r, and MER,

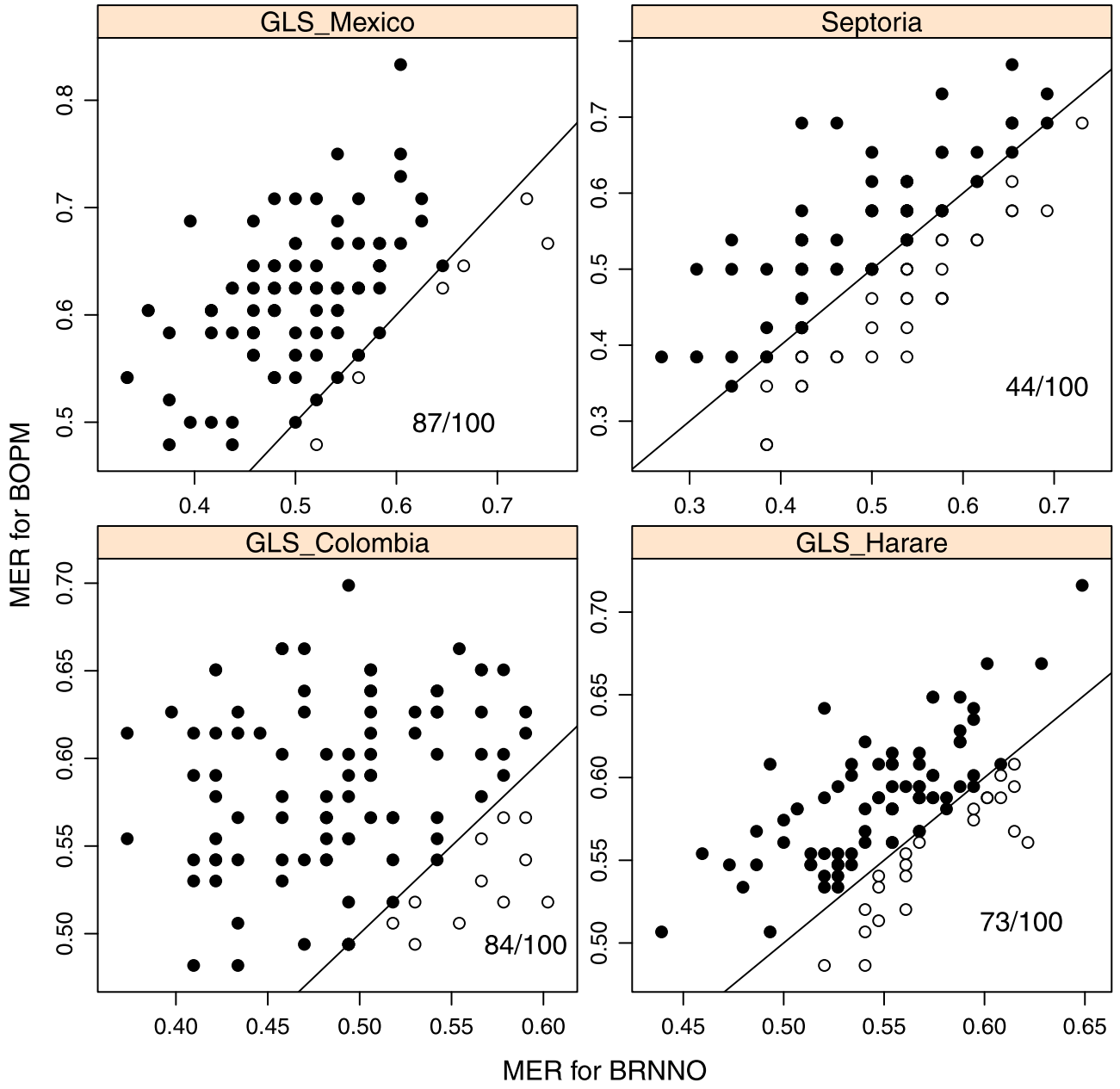

FIGURE 3 Scatter plot matrix of MER for BRNNO and BOPM. Points above the 45° degree line are associated with the worst performance of the model on the y-axis (BOPM) in comparison with the model presented on the x-axis (BRNNO). When the worst model is BOPM, it is represented by a filled circle, and when the worst model is BRNNO, it is represented by an open circle. The plots also show the number of times that BOPM is inferior to BRNNO as %

respectively. Furthermore, usually the second best model was also BRNNO (using either one of the two algorithms GEM or GMAP).

Table 2 shows the number of parameters that should be estimated for the neural network and the estimated effective number of parameters. We fitted the models with $S = 3$ neurons and concluded that it was not necessary to include more neurons.

Figures 2 and 3 show the evaluation of metrics MAE and MER for Septoria and GLS in Colombia, Harare and Mexico, respectively, for each of the 100 partitions generated at random and obtained with the GEM algorithm. Results show similar patterns for the GMAP algorithm. These figures are a graphical representation of the summary shown in Table 1. The 45° line makes it possible to quickly compare the performance of the two models, from where it is clear

that MAE and MER are worse for the probit model than for the proposed (BRNNO) model because they exhibit higher values.

For the MAE criterion, BRNNO was superior to BOPM in GLS because the latter exhibited higher MAE in 89% of times in Mexico, 85% of times in Colombia and 69% of times in Harare. The MAE for Septoria was slightly better for BOPM since BRNNO was worse in 52% of times. Similar patterns were observed in the case of MER, where BOPM was worse than BRNNO for GLS, MER was higher 87% of the times for Mexico, 84% of times in Colombia and 73% of times in Harare. Finally, in the case of Septoria, the BOPM performed slightly better than the proposed model since it had smaller MER values 56% of times.

4 | DISCUSSION

Results suggest that the proposed model BRNNO performs better than the BOPM model for both algorithms, GMAP and GEM; however, GEM is several orders of magnitude faster than GMAP. For example, for the GLS dataset, the process took ∼8 minutes to fit the proposed model with the GEM algorithm in a computer with an intel quad core i7 processor at 2.8 GHz, whereas for 10,000 MCMC iterations with GMAP on the same computer, the process lasted ∼3 hours. Thus it is clear that computing times are reduced substantially by using the GEM algorithm.

As already mentioned, the single hidden layer neural network (BRNNO) proposed in this study generalizes the Bayesian ordered probit model (BOPM), as well as the Bayesian ordered logit model (BOLM) of Montesinos-López et al. (2015b). It should be noted that the Brier score results from Montesinos-López et al. (2015b) are not directly comparable with those obtained for the proposed BRNNO method in this study. However, the pooled BS of the real GLS data in Table 4 of Montesinos-López et al. (2015b) showed a slightly higher BS of around 0.37 for BOLM and BOPM than the BS estimated for the BRNNO ranging from 0.33–0.35 for the three locations in Colombia, Mexico, and Harare (Table 1). The BS values for the Bayesian ordered probit model (BOPM) of this study for the three locations, Colombia, Harare, and Mexico, were 0.3550, 0.3458, and 0.3359, respectively.

Not many studies have been reported on the prediction of ordinal traits in genomic-enabled prediction. Recently, Montesinos-López et al. (2019) proposed a deep learning method for the simultaneous prediction of mixed phenotypes (binary, ordinal and continuous) in plant breeding. The proposed neural network BRNNO and the open-source R package brnn referred in this study for the genomic-enabled prediction of ordinal data is important due to the fact that there is a lack of genomic-enabled studies predicting categorical data in plant breeding and simultaneously offering ready-to-use R software.

5 | CONCLUSIONS

We introduced a neural network that generalizes existing models for the prediction of ordinal responses. We explored two algorithms (GEM and GMAP) that can be used to fit the proposed model. The GMAP algorithm is able to fit the model, but it is very slow because at each iteration, it is necessary to fit a Bayesian Regularized Neural Network using the latent variable as a response; additionally, it is necessary to have several thousands of MCMC iterations to make inferences. In contrast, although the GEM algorithm is also slow in the empirical evaluation, it requires much fewer iterations to converge; these results are in agreement with Kärkkäinen and Sillanpää (2013). Thus, we conclude there is ample room for improvement in both algorithms.

Regarding the genomic-enabled predictive accuracy of the models, most of selected indexes (MAE, RMSE, r, MER), except the Brier score, favored the proposed BRNNO model, fitted either with the GMAP or the GEM algorithm. Improvements are limited, but consistent with the findings of other studies (Montesinos-López et al., 2015b). We should point out that this approach could be applied not only in the GS context, but also in the context of conventional phenotype plant breeding for disease resistance and many other ordinal traits.

ACKNOWLEDGMENTS

We thank all scientists, field workers, and lab assistants from National Programs and CIMMYT who collected the data used in this study. We acknowledge the financial support provided by the Foundation for Research Levy on Agricultural Products (FFL) and the Agricultural Agreement Research Fund (JA) in Norway through NFR grant 267806. We are also thankful for the financial support provided by CIMMYT CRP (maize and wheat), the Bill & Melinda Gates Foundation, as well the USAID projects (Cornell University and Kansas State University) that generated the CIMMYT data analyzed in this study.

ORCID

Paulino Pérez-Rodríguez https://orcid.org/0000-0002-3202-1784
José Crossa https://orcid.org/0000-0001-9429-5855

REFERENCES

Albert, J. H., & Chib, S. (1993). Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, *88*(422), 669–679. https://doi.org/10.1080/01621459.1993.10476321

Crossa, J., Pérez, P., de los Campos, G., Mahuku, G., Dreisigacker, S., & Magorokosho, C. (2011). Genomic selection and prediction in plant breeding. *Journal of Crop Improvement*, *25*(3), 239–261. https://doi.org/10.1080/15427528.2011.558767

Crossa, J., Pérez-Rodríguez, P., Cuevas, J., Montesinos-López, O., Jarquín, D., de los Campos, G., … Varshney, R. K. (2017). Genomic selection in plant breeding: Methods, models, and perspectives. *Trends in Plant Science*, *22*(11), 961–975. https://doi.org/10.1016/j.tplants.2017.08.011

da Costa, J. P., & Cardoso, J. S. (2005). Classification of ordinal data using neural networks. In J. Gama, R. Camacho, P. B. Brazdil, A. M. Jorge, & L. Torgo (Eds.), *Machine Learning: ECML 2005: 16th European Conference on Machine Learning, Porto, Portugal, October 3–7, 2005. Proceedings* (pp. 690–697). Berlin, Heidelberg: Springer-Verlag.

Foresee, D. F., & Hagan, M. T. (1997). Gauss-Newton approximation to Bayesian learning. *Proceedings of International Conference on Neural Networks (ICNN'97)*, *3*, 1930–1935. https://doi.org/10.1109/ICNN.1997.614194

Geman, S., & Geman, D. (1984). Stochastic relaxation, gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI*, *6*(6), 721–741. https://doi.org/10.1109/TPAMI.1984.4767596

Gianola, D. (1982). Theory and analysis of threshold characters. *Journal of Animal Science*, *54*(5), 1079–1096. https://doi.org/10.2527/jas1982.5451079x

Gianola, D., Okut, H., Weigel, K. A., & Rosa, G. J. (2011). Predicting complex quantitative traits with Bayesian neural networks: A case study with Jersey cows and wheat. *BMC Genetics*, *12*(1), 87. https://doi.org/10.1186/1471-2156-12-87

Brier, G. W. (1950). Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, *78*(1), 1–3.

González-Camacho, J. M., de los Campos, G., Pérez, P., Gianola, D., Cairns, J. E., Mahuku, G., … Crossa, J. (2012). Genome-enabled prediction of genetic values using radial basis function neural networks. *Theoretical and Applied Genetics*, *125*(4), 759–771. https://doi.org/10.1007/s00122-012-1868-9

González-Camacho, J. M., Ornella, L., Pérez-Rodríguez, P., Gianola, D., Dreisigacker, S., & Crossa, J. (2018). Applications of machine learning methods to genomic selection in breeding wheat for rust resistance. *The Plant Genome*, *11*(2), 0. https://doi.org/10.3835/plantgenome2017.11.0104

Kärkkäinen, H. P., & Sillanpää, M. J. (2013). Fast Genomic Predictions via Bayesian G-BLUP and Multilocus models of threshold traits including censored Gaussian data. *G3: Genes|Genomes|Genetics*, *3*(9), 1511–1523. https://doi.org/10.1534/g3.113.007096

Kernighan, B. W., & Ritchie, D. M. (1988). *The C programming language* (2nd ed). Englewood Cliffs, NJ: Prentice Hall.

Kim, S., Hall, S. D., & Li, L. (2009). A Novel Gibbs Maximum a Posteriori (GMAP) Approach on Bayesian Nonlinear Mixed-Effects Population Pharmacokinetics (PK) Models. *Journal of Biopharmaceutical Statistics*, *19*(4), 700–720. https://doi.org/10.1080/10543400902964159

Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, *2*(2), 164–168. https://doi.org/10.1090/qam/10666

Lopez-Cruz, M., Crossa, J., Bonnett, D., Dreisigacker, S., Poland, J., Jannink, J.-L., … de los Campos, G. (2015). Increased prediction accuracy in wheat breeding trials using a marker × environment interaction genomic selection model. *G3: Genes|Genomes|Genetics*, *5*(4), 569–582. https://doi.org/10.1534/g3.114.016097

Marquardt, D. W. (1963). An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, *11*(2), 431–441. https://doi.org/10.1137/0111030

Mathieson, M. J. (1995). Ordinal models for neural networks. *Proc. 3rd Int. Conf. Neural Netw. Capital Markets*, *1995*, 523–536.

Meuwissen, T. H. E., Hayes, B. J. B., & Goddard, M. E. M. (2001). Prediction of total genetic value using genome-wide dense marker maps. *Genetics*, *157*, 1819–1829.

Montesinos-López, O. A., Martín-Vallejo, J., Crossa, J., Gianola, D., Hernández-Suárez, C. M., Montesinos-López, A., … Singh, R. (2019). New deep learning genomic-based prediction model for multiple traits with binary, ordinal, and continuous phenotypes.*G3:Genes|Genomes|Genetics*. *5*(9), 1545–1556, https://doi.org/10.1534/g3.119.300585

Montesinos-López, O. A., Montesinos-López, A., Pérez-Rodríguez, P., de los Campos, G., Eskridge, K., & Crossa, J. (2015a). Threshold models for genome-enabled prediction of ordinal categorical traits in plant breeding. *G3: Genes|Genomes|Genetics*, *5*(2), 291–300. https://doi.org/10.1534/g3.114.016188

Montesinos-López, O. A., Montesinos-López, A., Crossa, J., Burgueño, J., & Eskridge, K. (2015b). Genomic-enabled prediction of ordinal data with bayesian logistic ordinal regression. *G3: Genes|Genomes|Genetics*, *5*(10), 2113–2126. https://doi.org/10.1534/g3.115.021154

Neal, R. M., & Hinton, G. E. (1998). A view of the EM algorithm that justifies incremental, sparse, and other variants. In M. I. Jordan (Ed.), *Learning in graphical models* (pp. 355–368). Springer. https://doi.org/10.1007/978-94-011-5014-9_12

Pérez, P., & de los Campos, G. (2014). Genome-wide regression and prediction with the BGLR statistical package. *Genetics*, *198*(2), 483–495. https://doi.org/10.1534/genetics.114.164442

Pérez-Rodríguez, P., Gianola, D., Weigel, K. A., Rosa, G. J. M., & Crossa, J. (2013). Technical note: An R package for fitting Bayesian regularized neural networks with applications in animal breeding. *Journal of Animal Science*, *91*(8), 3522–3531. https://doi.org/10.2527/jas.2012-6162

Pérez-Rodríguez, P., Acosta-Pech, R., Pérez-Elizalde, S., Cruz, C. V., Suárez-Espinosa, J., & Crossa, J. (2018). A Bayesian genomic regression model with skew normal random errors. *G3: Genes|Genomes|Genetics*, *8*(5), 1771–1785. https://doi.org/10.1534/g3.117.300406

Pérez-Rodríguez, P., & Gianola, D. (2020). *brnn: Bayesian Regularization for Feed-Forward Neural Networks*. R package version 0.8. Retrieved from https://CRAN.R-project.org/package=brnn

Pérez-Rodríguez, P., Gianola, D., González-Camacho, J. M., Crossa, J., Manès, Y., & Dreisigacker, S. (2012). Comparison between linear and non-parametric regression models for genome-enabled prediction in wheat. *G3: Genes|Genomes|Genetics*, *2*(12), 1595–1605. https://doi.org/10.1534/g3.112.003665

Okut, H., Gianola, D., Rosa, G. J. M., & Weigel, K. A. (2011). Prediction of body mass index in mice using dense molecular markers and a regularized neural network. *Genetics Research*, *93*(3), 189–201. https://doi.org/10.1017/S0016672310000662

Poland, J., Endelman, J., Dawson, J., Rutkoski, J., Wu, S., Manes, Y., … Jannink, J.-L. (2012). Genomic selection in wheat breeding using

genotyping-by-sequencing. *The Plant Genome Journal*, *5*(3), 103. https://doi.org/10.3835/plantgenome2012.06.0006

R Core Team. (2019). *R: A Language and Environment for Statistical Computing*. R Core Team, Vienna. Retrieved from https://www.R-project.org/

Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, *3*(3), 210–229. https://doi.org/10.1147/rd.33.0210

Stevens, S. S. (1946). On the theory of scales of measurement. *Science*, *103*(2684), 677–680. https://doi.org/10.1126/science.103.2684.677

Stroup, W. W. (2012). *Generalized linear mixed models: Modern concepts, methods and applications*. Boca Raton, FL: CRC Press, Taylor & Francis Group.

Tanner, M. A. (1993). *Tools for statistical inference: Methods for the exploration of posterior distributions and likelihood functions* (2nd ed). New York: Springer-Verlag.

Tharwat, A. (2018). Classification assessment methods. *Applied Computing and Informatics*. https://doi.org/10.1016/j.aci.2018.08.003

Wang, C. L., Ding, X. D., Wang, J. Y., Liu, J. F., Fu, W. X., Zhang, Z., … Zhang, Q. (2013). Bayesian methods for estimating GEBVs of threshold traits. *Heredity*, *110*(3), 213–219. https://doi.org/10.1038/hdy.2012.65

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section at the end of the article.

How to cite this article: Pérez-Rodríguez P, Flores-Galarza S, Vaquera-Huerta H, del Valle-Paniagua DH, Montesinos-López OA, Crossa J. Genome-based prediction of Bayesian linear and non-linear regression models for ordinal data. *Plant Genome*. 2020;13:e20021. https://doi.org/10.1002/tpg2.20021

APPENDIX 1: GMAP ALGORITHM

Here we describe the algorithm for fitting the proposed Bayesian Regularized Neural Network for Ordinal Data (BRNNO). The proposed algorithm is known in statistical literature as GMAP (Kim et al., 2009); it combines the well-known Gibbs Sampler algorithm (G) to sample from some random variables and an optimization algorithm to obtain the posterior mode (Maximum A Posteriori = MAP) of other parameters of the model. The algorithm can be summarized as follows:

1. Initialization step.

$$\text{Initialize}\,\boldsymbol{\theta} = \left(w_1, \ldots, w_S; b_1, \ldots, b_S; \boldsymbol{\beta}^{[1]}, \ldots, \boldsymbol{\beta}^{[S]}\right),$$

$$\sigma^2_{\boldsymbol{\theta}}, \boldsymbol{\lambda} = \left\{\lambda_1, \ldots, \lambda_K\right\},$$

$$\boldsymbol{Z} = \left\{Z_1, \ldots, Z_n\right\}.$$

G-Step

1. Sample $Z_i | Y_i = j, \text{else}, i = 1, \ldots, n$; see equation (9).
2. Sample $\lambda_j | \text{else}, j = 2, \ldots, K$; see equation (10).

MAP-Step

1. Fit for the regularized neural network, using as a response variable the pseudo-data generated in step 2 (see equation 5):

$$Z_i = \sum_{m=1}^{s} w_m \, g\left(b_m + \boldsymbol{x}_i^t \boldsymbol{\beta}^{[m]}\right) + e_i,$$

with $g(u) = \frac{2}{1+\exp(-2u)} - 1$ the tangent hyperbolic activation function.

Steps 2–4 are repeated B times until convergence.

The steps necessary to fit the regularized neural network with pseudo-data are essentially the same as those used to fit a Bayesian Regularized Neural Network for a continuous response; see, for example, Foresee and Hagan (1997), Okut et al. (2011), Gianola et al. (2011), Pérez-Rodríguez et al. (2013). Here we summarize the basic algorithm:

1. Obtain the conditional posterior mode of the elements in $\boldsymbol{\theta}$, setting $\sigma^2_e = 1$ and assuming that $\sigma^2_{\boldsymbol{\theta}}$ is known. These modes are obtained by minimizing the augmented sum of squares:

$$F(\boldsymbol{\theta}) = \frac{1}{2\sigma^2_e} \sum_{i=1}^{n} e_i^2 + \frac{1}{2\sigma^2_{\boldsymbol{\theta}}} \sum_{l=1}^{p} \theta_l^2,$$

where $e_i = z_i - \hat{z}_i$ is the prediction error and p is the number of elements in vector $\boldsymbol{\theta}$. The augmented sum of squares can be minimized using the Levenberg-Marquardt algorithm (Levenberg, 1944; Marquardt, 1963).

1. Update the variance component $\sigma^2_{\boldsymbol{\theta}}$ by maximizing $p(\boldsymbol{z}|\sigma^2_{\boldsymbol{\theta}})$. Because of non-linearity, the marginal log-likelihood, $\log p(\boldsymbol{z}|\sigma^2_{\boldsymbol{\theta}})$, is not expressible in closed form, but can be approximated as:

$$\log p(\boldsymbol{z}|\sigma^2_{\boldsymbol{\theta}}) \propto \frac{p}{2}\log\alpha - \frac{1}{2}\log|\boldsymbol{\Sigma}|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{MAP}} - F(\boldsymbol{\theta})\,|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{MAP}},$$

with $\boldsymbol{\Sigma} = \frac{\partial^2}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^t} F(\boldsymbol{\theta})$, $\alpha = \frac{1}{2\sigma^2_{\boldsymbol{\theta}}}$. The α parameter that maximizes the marginal likelihood is updated iteratively, $\alpha_{new} = \frac{\eta}{\sum_{l=1}^{p}\theta_l^2}$ and $\eta = p - 2\alpha_{old} Trace(\boldsymbol{H}^{-1})$, with α_{new} the

updated value of the parameter and α_{old} the value of the parameter in the previous iteration.

The algorithm described above was implemented in R and the C programming language (Kernighan & Ritchie, 1988), and is included as electronic supplementary material, but was not included in the brnn package because it is very slow. The user can fit a Bayesian Regularized Neural Network by providing data to the R function **brnn_ordinal_mcmc**, which internally combines interpreted and compiled codes to fit the model. Upon successful execution, the routine returns an R object with MCMC samples for $\{\boldsymbol{Z}, \boldsymbol{\lambda}\}$ and the posterior mode for $\{\boldsymbol{\theta}, \sigma^2_{\boldsymbol{\theta}}\}$ obtained by fitting the Bayesian regularized neural network with $\boldsymbol{Z}$ and $\boldsymbol{\lambda}$ fixed to the posterior mean of those parameters obtained from MCMC samples. MCMC samples can be obtained for posterior distributions for thresholds, i.e., $p(\lambda_j|\text{data})$. The output object also allows the prediction of new observations (e.g., in a testing set for cross-validation) using estimated network parameters.

APPENDIX 2: GEM ALGORITHM

The GEM algorithm (Neal & Hinton, 1998; Kärkkäinen & Sillanpää, 2013) is similar to the GMAP algorithm. The 'tractable' parameters (latent variables and thresholds) are updated according to the conditional expectations for distributions given in (9) and (10) and the 'intractable' parameters are estimated using MAP, so updating of 'tractable' and 'untractable' parameters is repeated until convergence.

The algorithm can be summarized as follows:

1. Initialization step.

$$\text{Initialize } \boldsymbol{\theta} = \left(w_1, \ldots, w_S; b_1, \ldots, b_S; \boldsymbol{\beta}^{[1]}, \ldots, \boldsymbol{\beta}^{[S]}\right),$$
$$\sigma^2_{\boldsymbol{\theta}}, \boldsymbol{\lambda} = \left\{\lambda_1, \ldots, \lambda_K\right\},$$
$$\boldsymbol{Z} = \left\{Z_1, \ldots, Z_n\right\}.$$

E-Step

1. Update the latent variables $Z_i|\, Y_i = j, \text{else},\ i = 1, \ldots, n$, by replacing the current values with the expected values from truncated normal distribution given in (9).
2. Update the thresholds Sample $\lambda_j|\text{else},\ j = 2, \ldots, K$, by replacing the current values with the expected values from the uniform distribution given in (10).

MAP-Step

1. Fit the regularized neural network using the pseudo-data generated in step 2 as a response variable (see equation 5):

$$Z_i = \sum_{m=1}^{s} w_m\, g\left(b_m + \boldsymbol{x}_i^t \boldsymbol{\beta}^{[m]}\right) + e_i,$$

with $g(u) = \frac{2}{1+\exp(-2u)} - 1$ the tangent hyperbolic activation function.

Steps 2–4 are repeated until convergence.