Noname manuscript No.
(will be inserted by the editor)

Global and local distance-based generalized linear models

Eva Boj · Adrià Caballé · Pedro Delicado · Anna Esteve · Josep Fortiana

Received: date / Accepted: date

Abstract This paper introduces local distance-based generalized linear models. These models extend (weighted) distance-based linear models firstly to the generalized linear model framework. Then a nonparametric version of these models is proposed by means of local fitting. Distances between individuals are the only predictor information needed to fit these models. Therefore they are applicable, among others, to mixed (qualitative and quantitative) explanatory variables or when the regressor is of functional type. An implementation is provided by the R package `dbstats`, which also implements other distance-based prediction methods. Supplementary material for this article is available online, which reproduces all the results of this article.

Keywords Distance-based prediction · Functional data analysis · Generalized linear model · Iteratively weighted least squares · Local likelihood · R package `dbstats`.

Mathematics Subject Classification (2000) 62G08 · 62J12

E. Boj
Departament de Matemàtica Econòmica, Financera i Actuarial, Universitat de Barcelona, Barcelona, Spain. E-mail: evaboj@ub.edu

A. Caballé
University of Edimbourgh, Edimbourgh, UK. E-mail: adriacaballe@yahoo.es

P. Delicado
Dept. d'Estadística i I.O., Universitat Politècnica de Catalunya, Barcelona, Spain. E-mail: pedro.delicado@upc.edu

A. Esteve
CEEISCAT-ASPCAT, Badalona, Spain. E-mail: aesteve@iconcologia.net

J. Fortiana
Departament de Probabilitat, Lògica, i Estadística, Universitat de Barcelona, Barcelona, Spain. E-mail: fortiana@ub.edu

1 Introduction

Statistical techniques that are based on distances or similarities between sample units enjoy a long and ample tradition. Among them, cluster analysis (see, for instance, Everitt et al. 2011) and multidimensional scaling (MDS; see, for instance, Borg and Groenen 2005), are used extensively. MDS is a multivariate dimensionality reduction technique valid when the information about data is given by an inter-individual distances matrix. MDS represents in this case an alternative to principal component analysis (PCA, which requires a standard data matrix instead of a distances matrix).

Based on the metric version of MDS, Cuadras (1989) introduced the distance-based linear model (DB-LM), later extended by Cuadras and Arenas (1990) and Cuadras et al. (1996). They assume that for a set of individuals an inter-individual distances matrix is available, as well as the value of a continuous response variable for each individual. The main idea of DB-LM is to use the principal co-ordinates (the output of the metric MDS applied to the inter-individual distances matrix) as explanatory variables in a linear regression model. See Subsection 2.1 for more details on DB-LM. See also the interesting recent work of Faraway (2014b) which considers the case of response variable is also given as a distances matrix.

The usefulness of distance-based statistical methods (as MDS or DB-LM, for instance) comes from the fact that situations often arise where the only possibility of knowing relations between statistical units is to compute a distance matrix between them (then neither PCA or standard LM, for instance, are feasible). Let us mention some examples of these situations.

- In marketing or in psychology, the individuals participating in a study can often say how similar or different are pairs of objects or stimuli but they find it difficult to describe them by a finite number of measurable characteristics. Therefore a common output of these studies is a distance matrix (or a similarity matrix) between objects (or stimuli).
- A social network can be seen as a graph where nodes represent individuals and edges between nodes represent relationships between individuals, such as friendship. In this context a natural distance between individuals is the shortest path metric (the smallest number of steps from one node to another within the graph). MDS is used to depict the social network in a planar graph (Buja et al. 2008). Assume now that a continuous response (for instance, mobile phone consumption last year) is observed for several individuals in the network (for instance, the subscribers to a particular mobile phone company). Then this company could be interested in predicting the potential consumption of other individuals in the network that are not among their subscribers. The company may direct its marketing efforts to attract those potential customers with a higher expected consumption. DB-LM is an appropriate regression technique in this context.
- Assume we have observed a sample of complex random objects belonging to an abstract space where a metric is defined. To fix ideas, consider a sample of random graphs (see Banks and Carley 1994, Butts and Carley

2001 or Butts and Carley 2005) and the Hamming distance between two graphs G_1 and G_2, defined as the number of addition/deletion operations (of edges or nodes) required to turn graph G_1 into G_2. MDS can be used for dimensionality reduction. If, additionally, a continuous variable is measured for each graph (for instance, the average shortest path between two nodes in the graph), DB-LM is useful for fitting a regression model with this characteristic as response variable.
- Multivariate mixed data (some qualitative and some quantitative variables) and functional data constitute two additional examples of data structure that can benefit from distance-based techniques, as we will show with more detail in Subsections 3.1 and 4.1, respectively.

The DB-LM is susceptible of being extended in the same way that the standard LM has been extended to GLM, to local linear regression or to nonparametric versions of GLM (see, for instance, McCullagh and Nelder (1989) and Wood (2006)). In this line, Boj et al. (2010) introduced local DB-LM, a nonparametric prediction technique extending (weighted) DB-LM. In the present paper we introduce two further extensions: the distance-based generalized linear models (DB-GLM) and their nonparametric version (the local DB-GLM) via local likelihood. In general, any statistical technique based on weighted least squares (WLS) can be adapted to data presented as an inter-individual distances matrix by just replacing each WLS step by the corresponding weighted DB-LM. This procedure is easily extended to iterative weighted least squares (IWLS), as applied in many statistical methods, ranging from generalized linear models (GLM) (see McCullagh and Nelder 1989) to robust regression (see, for instance, Green 1984 or Street et al. 1988). This is the procedure used in this paper for the construction of DB-GLM and local DB-GLM.

All the computations in the paper have been done using the recently developed R package (R Development Core Team, 2015), called `dbstats` (Boj et al., 2014), available at `http://CRAN.R-project.org/package=dbstats` from the Comprehensive R Archive Network. `dbstats` contains classes and functions implementing distance-based prediction methods such as DB-LM, local DB-LM, DB-GLM, local DB-GLM and distance-based partial least squares regression (DB-PLSR) (Boj et al., 2007).

The paper is structured as follows. In Section 2 we review the main features of DB-LM and classical GLM, both being the fundamentals of the new models introduced in the paper. In Section 3 we develop DB-GLM as an extension of DB-LM to the framework of GLM. An example of its usage is provided in Subsection 3.1. The nonparametric version of DB-GLM (based on local likelihood) is introduced in Section 4, jointly with an example of its usage (Sub-section 4.1). Conclusions are summarized in Section 5. The implementation of DB-GLM and local DB-GLM by the functions `dbglm` and `ldbglm`, respectively, of `dbstats` package is described in an Appendix. Code excerpts reproducing all the results of this article are provided as online supplementary materials.

2 Review of DB-LM and GLM

In this section we recall the main characteristics of DB-LM and GLM.

2.1 Distance-based linear model: definition and results

DB-LM was introduced by Cuadras (1989) and has been developed in Cuadras and Arenas (1990), Cuadras et al. (1996), Boj et al. (2007), Esteve et al. (2009) and Boj et al. (2010). Here we recall its main concepts, as given in these articles, where the reader is referred to for more details and proofs.

A sketchy description of DB-LM is as follows. We choose at random n independent individuals Ω_i, $i = 1, \ldots, n$, from a given population. For each of them we observe the value of a random *response variable* Y: $Y_1, \ldots, Y_n$. Let $\Omega = \{\Omega_1, \ldots, \Omega_n\}$ and $\boldsymbol{Y} = (Y_1, \ldots, Y_n)^\top$. Let $w_i \in (0, 1)$ be the constant positive weight of Ω_i. The $n \times 1$ weight vector $\boldsymbol{w} = (w_1, \ldots, w_n)^\top$ is standardized to unit sum, i.e., $\mathbf{1}^\top \cdot \boldsymbol{w} = 1$, where $\mathbf{1}$ is the $n \times 1$ vector of ones.

We assume that a distance (metric or semi-metric) $\delta(\cdot, \cdot)$ is defined in Ω. A particular case comes up when individuals in Ω are described by a set $\boldsymbol{Z}$ of variables, possibly including both quantitative and qualitative measurements or, possibly, other nonstandard quantities, such as character strings or functions. Then distance δ is a function of the $\boldsymbol{Z}$ variables.

We denote by $\boldsymbol{\Delta}$ the $n \times n$ matrix, whose entries are the squared distances $\delta^2(\Omega_i, \Omega_j)$. We define the $n \times n$ *inner-products matrix* as

$$\boldsymbol{G_w} = -\frac{1}{2}\, \boldsymbol{J_w} \cdot \boldsymbol{\Delta} \cdot \boldsymbol{J_w}^\top,$$

where $\boldsymbol{J_w}$ is the $\boldsymbol{w}$-centering matrix, defined as $\boldsymbol{J_w} = \boldsymbol{I} - \mathbf{1} \cdot \boldsymbol{w}^\top$. Any $n \times k$ matrix $\boldsymbol{X_w}$ such that $\boldsymbol{G_w} = \boldsymbol{X_w} \cdot \boldsymbol{X_w}^\top$ is called a *Euclidean configuration of* $\boldsymbol{\Delta}$. Observe that $k \geq r \equiv rank\, \boldsymbol{G_w}$ and that $\boldsymbol{w}^\top \cdot \boldsymbol{X_w} = \mathbf{0}$, i.e., $\boldsymbol{X_w}$ is $\boldsymbol{w}$-centered. Such a decomposition exists if and only if $\boldsymbol{G_w}$ is a positive semidefinite matrix, in which case $\boldsymbol{\Delta}$ is called *Euclidean*. In this case it can be proved that the trace of $\boldsymbol{G_w}$ divided by n extends the concept of total variation, so we call *geometric variability* of $\boldsymbol{\Delta}$ to $\mathrm{tr}(\boldsymbol{G_w})/n$. Additionally we denote by $\boldsymbol{g_w}$ a $1 \times n$ row vector containing the (necessarily nonnegative) diagonal entries of $\boldsymbol{G_w}$. It can be proved that the map $\boldsymbol{\Delta} \leftrightarrow \boldsymbol{G_w}$ is bijective, as $\boldsymbol{\Delta}$ can be recovered from $\boldsymbol{G_w}$:

$$\boldsymbol{\Delta} = \mathbf{1} \cdot \boldsymbol{g_w} + \boldsymbol{g_w}^\top \cdot \mathbf{1}^\top - 2\boldsymbol{G_w}.$$

Let us give a precise definition of a DB-LM.

Definition 1 We say that the response $\boldsymbol{Y}$, the weights $\boldsymbol{w}$ and the square distances matrix $\boldsymbol{\Delta}$ follow a DB-LM (a distance-based linear model) when $\boldsymbol{\mu} = E(\boldsymbol{Y})$ $\boldsymbol{w}$-centered (that is $\boldsymbol{J_w} \cdot \boldsymbol{\mu}$) belongs to the column space $\mathscr{G}$ of $\boldsymbol{G_w}$.

Observe that $\mathscr{G}$ is also the column space of any Euclidean configuration $\boldsymbol{X_w}$ of $\boldsymbol{\Delta}$ because $\boldsymbol{G_w} = \boldsymbol{X_w} \cdot \boldsymbol{X_w}^\top$ (Rao, 1973, p. 27). Simple algebra shows that

$$\delta^2(\Omega_i, \Omega_j) = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 = \langle \boldsymbol{x}_i, \boldsymbol{x}_i \rangle + \langle \boldsymbol{x}_j, \boldsymbol{x}_j \rangle - 2\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle,$$

which express equality between $\delta(\Omega_i, \Omega_j)$ and the Euclidean distance between rows $\boldsymbol{x}_i$ and $\boldsymbol{x}_i$ of $\boldsymbol{X_w}$. The standard inner product in $\mathbb{R}^k$ is denoted by $\langle \cdot, \cdot \rangle$.

Let $\boldsymbol{y}$ be the observed values of the response variable $\boldsymbol{Y}$. The estimation of the DB-LM corresponding to responses $\boldsymbol{y}$, weights $\boldsymbol{w}$ and squared distances matrix $\boldsymbol{\Delta}$, is performed by doing WLS regression of $\boldsymbol{y}$ on a $\boldsymbol{w}$-centered Euclidean configuration of $\boldsymbol{\Delta}$, $\boldsymbol{X_w}$, a *latent Euclidean configuration*.

Assume a new case Ω_{n+1} is available, and we are given the $1 \times n$ vector $\boldsymbol{\delta}_{n+1}$ of squared distances from Ω_{n+1} to the n previously known individuals. Ω_{n+1} can be represented as a k-vector $\boldsymbol{x}_{n+1}$ in the row space of $\boldsymbol{X_w}$ using the Gowers interpolation or add-a-point formula (Gower 1968; see Boj et al. 2010 for the weighted version). Then, the predicted Y for Ω_{n+1} is $\boldsymbol{x}_{n+1} \cdot \hat{\boldsymbol{\beta}}$, where $\hat{\boldsymbol{\beta}}$ is the vector of estimated regression coefficients.

The following Theorem (proved in Boj et al. 2010) states that DB-LM does not depend on a specific $\boldsymbol{X_w}$, since the final quantities are obtained directly from the distances. Usually such a configuration needs not be made explicit, and neither do $\hat{\boldsymbol{\beta}}$ or $\boldsymbol{x}_{n+1}$.

Theorem 1 *In DB-LM the hat matrix is*

$$\boldsymbol{H_w} = \boldsymbol{G_w} \cdot \left(\boldsymbol{D_w}^{1/2} \cdot \boldsymbol{F_w}^{+} \cdot \boldsymbol{D_w}^{1/2} \right), \tag{1}$$

where $\boldsymbol{D_w} = diag(\boldsymbol{w})$ is the diagonal matrix whose diagonal entries are the weights $\boldsymbol{w}$,

$$\boldsymbol{F_w} = \boldsymbol{D_w^{1/2}} \cdot \boldsymbol{G_w} \cdot \boldsymbol{D_w^{1/2}},$$

and $\boldsymbol{F_w}^{+}$ is the Moore-Penrose pseudo-inverse of $\boldsymbol{F_w}$. Thus, $\boldsymbol{H_w}$ is an intrinsic quantity, meaning that it can be expressed directly as a function of the distances or, equivalently, the inner products.

The fitted values are

$$\hat{\boldsymbol{y}} = \bar{y}_{\boldsymbol{w}} \cdot \mathbf{1} + \boldsymbol{H_w} \cdot (\boldsymbol{y} - \bar{y}_{\boldsymbol{w}} \cdot \mathbf{1}), \tag{2}$$

where $\bar{y}_{\boldsymbol{w}} = \boldsymbol{w}^\top \cdot \boldsymbol{y}$ is the $\boldsymbol{w}$ mean of $\boldsymbol{y}$.

The predicted Y for a new case Ω_{n+1}, given its $\boldsymbol{\delta}_{n+1}$ vector, is:

$$\hat{y}_{n+1} = \bar{y}_{\boldsymbol{w}} + \frac{1}{2} (\boldsymbol{g_w} - \boldsymbol{\delta}_{n+1}) \cdot \left(\boldsymbol{D_w}^{1/2} \cdot \boldsymbol{F_w}^{+} \cdot \boldsymbol{D_w}^{1/2} \right) \cdot (\boldsymbol{y} - \bar{y}_{\boldsymbol{w}} \cdot \mathbf{1}). \tag{3}$$

In DB-LM the rank r of the hat-matrix in (1), as in an ordinary linear regression, is equal to the number of linearly independent linear predictors. Since for n cases, depending on the chosen metric, r can be as high as $n - 1$, giving an overparametrized model with unstable predictions, a sensible procedure is to replace the pseudo-inverse $\boldsymbol{F}^{+}_{\boldsymbol{w}}$ with a lower-rank approximation. This can be easily implemented by the Singular Value Decomposition which,

by the Schmidt-Eckart-Young Theorem (see, e.g., Stewart 1993), gives the best ℓ^2 approximation of any given rank k, $1 \leq k \leq r$. The rank k used to define the pseudo-inverse $\boldsymbol{F}_{\boldsymbol{w}}^{+}$ is called *effective rank*. Several criteria can be used to select a suitable value for effective rank k: ordinary or generalized cross-validation (OCV or GCV), as well as Akaike or Bayesian information criterion (AIC or BIC), defined as in the ordinary linear model (LM).

An alternative way to choose the effective rank k is based on the fact that the sum of all the singular values of $\boldsymbol{F}_{\boldsymbol{w}}$ is equal to the geometric variability of $\boldsymbol{\Delta}$. So it is possible to fix a given proportion of geometric variability to be achieved by the sum of the k largest singular values of $\boldsymbol{F}_{\boldsymbol{w}}$.

DB-LM contains WLS as a particular instance: if we start from a $n \times r$ $\boldsymbol{w}$-centered matrix $\boldsymbol{X}_{\boldsymbol{w}}$ of r continuous predictors corresponding to n individuals and we define $\boldsymbol{\Delta}$ as the matrix of squared Euclidean distances between rows of $\boldsymbol{X}_{\boldsymbol{w}}$, then $\boldsymbol{X}_{\boldsymbol{w}}$ is trivially a Euclidean configuration of $\boldsymbol{\Delta}$, hence the DB-LM hat matrix, response and predictions coincide with the corresponding WLS quantities of ordinary LM.

Let us finish this subsection discussing about the sensitivity of the DB-LM predictions to the choice of distance. It can be proved that the predictions are continuous on the distance matrix used to fit the DB-LM (see equation 3). In general the advices given on the choice of a distance measure when planning a MDS analysis (Borg and Groenen 2005, Chapter 6) are still valid when fitting a DB-LM. Discussing with an expert on the field of application on the appropriate distance choice is always a good practice. These remarks also apply to other distance-based regression methods that are introduced later in this paper.

2.2 Generalized linear model: basic concepts

We review the basic concepts and notations of GLM, for the sake of an easy reference. As it is well-known (see, eg., McCullagh and Nelder 1989), in a GLM we have a linear predictor $\eta_i = \boldsymbol{x}_i \cdot \boldsymbol{\beta}$, which is related to the response variable Y_i by means of a link function $g(\cdot)$, $\eta_i = g(\mu_i)$, then,

$$\mu_i = g^{-1}(\eta_i)\,;\ i = 1, \ldots, n, \tag{4}$$

where $\mu_i = E(Y_i)$.

In a GLM it is assumed that each component of the response has a distribution in the exponential family, taking the form:

$$f_{Y_i}(y_i; \theta_i, \phi) = \exp\left\{(y_i \cdot \theta_i - b(\theta_i))/a_i(\phi) + c(y_i, \phi)\right\}, \tag{5}$$

for some specific functions $a_i(\cdot)$, $b(\cdot)$ and $c(\cdot)$. The *dispersion parameter* ϕ is constant over observations. If ϕ is known, this is an exponential family model with canonical parameter θ_i.

The log-likelihood function for a GLM is $l(\theta_i; y_i) = (y_i \cdot \theta_i - b(\theta_i))/a_i(\phi) + c(y_i, \phi)$ and the mean and the variance of Y can be derived easily from the

relations $E\left(\frac{\partial l}{\partial \theta_i}\right) = 0$ and $E\left(\frac{\partial^2 l}{\partial \theta_i^2}\right) + E\left(\frac{\partial l}{\partial \theta_i}\right)^2 = 0$. From (5) we have that $\frac{\partial l}{\partial \theta_i} = \{y_i - b'(\theta_i)\}/a_i(\phi)$ and $\frac{\partial^2 l}{\partial \theta_i^2} = -b''(\theta_i)/a_i(\phi)$ and then,

$$E(Y_i) = \mu_i = b'(\theta_i) \text{ and } var(Y_i) = b''(\theta_i) \cdot a_i(\phi), \ i = 1, \ldots, n. \quad (6)$$

The variance of Y is the product of two functions; one, $b''(\theta_i)$, depends on the canonical parameter only (and hence on the mean μ_i, by the first part of (6)) and will be called the *variance function*, while the other is independent of θ_i and depends only on ϕ. The variance function, as a function of μ_i, will be written $V(\mu_i) = b''(\theta_i)$. Commonly $a_i(\phi)$ is of the form $a_i(\phi) = \phi/w_i$ and ϕ, where w_i is a known *prior weight* that varies from observation to observation. Finally, we can write the variance in (6) as

$$var(Y_i) = V(\mu_i) \cdot \frac{\phi}{w_i}. \quad (7)$$

The link function is said to be *canonical* when the linear predictor η_i is the same as the canonical parameter θ_i.

The global log-likelihood function is $l(\boldsymbol{\theta}; \boldsymbol{y}) = \sum_{i=1}^{n} l(\theta_i; y_i)$. The model deviance is

$$\text{Dev} = 2\left[l(\boldsymbol{\theta}^*; \boldsymbol{y}) - l(\boldsymbol{\theta}; \boldsymbol{y})\right]\phi = \sum_{i=1}^{n} 2w_i \cdot \left[y_i(\theta_i^* - \theta_i) - b(\theta_i^*) + b(\theta_i)\right], \quad (8)$$

where $\boldsymbol{\theta}^*$ is the maximum likelihood estimator of $\boldsymbol{\theta}$ in the *saturated model* (the model with one parameter per data point).

In a GLM the maximum-likelihood estimates of the parameters $\boldsymbol{\beta}$ in the linear predictor $\boldsymbol{\eta}$ can be obtained by IWLS (see, e.g., McCullagh and Nelder 1989 pp. 40-43 or Wood 2006 pp. 63-66 for a more detailed description and justification of the algorithm). In the IWLS the dependent variable of the regression is not $\boldsymbol{y}$ but $\boldsymbol{z}$, a linearized form of the link function applied to $\boldsymbol{y}$, and the weights $\boldsymbol{W}$ are functions of the fitted values $\hat{\boldsymbol{\mu}}$. The process is iterative because both the adjusted dependent variable $\boldsymbol{z}$ and the weight $\boldsymbol{W}$ depend on the fitted values, for which only current estimates are available. The procedure underlying the iteration is as follows. Let $\hat{\boldsymbol{\eta}}_0$ be the current estimate of the linear predictor, with corresponding fitted value $\hat{\boldsymbol{\mu}}_0$ derived from the link function $\boldsymbol{\eta} = g(\boldsymbol{\mu})$. Form the adjusted dependent variate with typical value

$$\boldsymbol{z}_0 = \hat{\boldsymbol{\eta}}_0 + (\boldsymbol{y} - \hat{\boldsymbol{\mu}}_0) \cdot \left(\frac{d\boldsymbol{\eta}}{d\boldsymbol{\mu}}\right)_0, \quad (9)$$

where the link derivative is evaluated at $\hat{\boldsymbol{\mu}}_0$. The weight vector is defined by

$$\boldsymbol{W}_0^{-1} = \left(\frac{d\boldsymbol{\eta}}{d\boldsymbol{\mu}}\right)_0^2 \cdot \frac{\boldsymbol{V}_0}{\boldsymbol{w}}, \quad (10)$$

where $\boldsymbol{V}_0$ is the variance function evaluated at $\hat{\boldsymbol{\mu}}_0$. Now regress $\boldsymbol{z}_0$ on the covariates with weight $\boldsymbol{W}_0$ to give new estimates of $\hat{\boldsymbol{\beta}}_1$ of the parameters; from

these form a new estimate $\hat{\boldsymbol{\eta}}_1$ of the linear predictor. Repeat until changes are sufficiently small.

Note that $\boldsymbol{z}$ is just a linearized form of the link function applied to the data, for, to first order, $g(\boldsymbol{y}) \simeq g(\boldsymbol{\mu}) + (\boldsymbol{y} - \boldsymbol{\mu}) \cdot g'(\boldsymbol{\mu})$. The variance of $\boldsymbol{z}$ is $\boldsymbol{W}^{-1}$ (ignoring the dispersion parameter), assuming that $\boldsymbol{\eta}$ and $\boldsymbol{\mu}$ are fixed and known.

3 Distance-based generalized linear model

By analogy to the definition of DB-LM, we give the following definition.

Definition 2 A DB-GLM (distance-based generalized linear model) consists of random variables $\boldsymbol{Y} = (Y_1, \ldots, Y_n)^\top$ whose expectation, $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)^\top$, transformed by the link function and $\boldsymbol{w}$-centered, is a vector in the column space $\mathscr{G}$ of $\boldsymbol{G}_{\boldsymbol{w}}$, that coincides with the column space of any Euclidean configuration $\boldsymbol{X}_{\boldsymbol{w}}$ of $\boldsymbol{\Delta}$.

Both DB-GLM and DB-LM share the same elements: a set of n independent individuals Ω_i, $i = 1, \ldots, n$, with associated weights vector $\boldsymbol{w}$ (standardized to unit sum), for which we have observed the response vector $\boldsymbol{y}$. We calculate the $n \times n$ squared distances matrix $\boldsymbol{\Delta}$. Just as GLM with respect to LM, DB-GLM differs from DB-LM in two aspects:

1. We assume the responses distribution is in an exponential dispersion family (5), as in any GLM.
2. The relation between the linear predictor $\boldsymbol{\eta} = \boldsymbol{X}_{\boldsymbol{w}} \cdot \boldsymbol{\beta}$, obtained from the latent Euclidean configuration $\boldsymbol{X}_{\boldsymbol{w}}$, and the response $\boldsymbol{y}$ is given by a link function $g(\cdot)$ as in (4).

Then we have an underlying GLM, with link function g

$$g(\mu_i) = \eta_i, \quad \text{where } \mu_i = E(Y_i), \quad \eta_i = \boldsymbol{x}_i \cdot \boldsymbol{\beta}, \tag{11}$$

where $\boldsymbol{\beta} \in \mathbb{R}^r$ is an $r \times 1$ parameter vector. Model (11) relates each response Y_i to the Euclidean coordinates of Ω_i. That is, the linear predictor η_i is a linear combination of the Euclidean coordinates $\boldsymbol{x}_i$ of Ω_i, the i-th row of the $n \times r$ matrix $\boldsymbol{X}_{\boldsymbol{w}}$ of a Euclidean configuration of $\boldsymbol{\Delta}$.

To fit the DB-GLM we use the IWLS algorithm described above, where DB-LM substitutes LM in formulas (9) and (10) to regress $\boldsymbol{z}_0$ on the covariates with weight $\boldsymbol{W}_0$, in order to obtain the new estimation $\hat{\boldsymbol{\eta}}_1$. Observe that the IWLS estimation process for DB-GLM does not depend on a specific $\boldsymbol{X}_{\boldsymbol{w}}$ since the final quantities are obtained directly from distances. In the first step we need and initial $\hat{\boldsymbol{\mu}}_0$. Then we calculate $\hat{\boldsymbol{\eta}}_0$ and $\left(\frac{d\boldsymbol{\eta}}{d\boldsymbol{\mu}}\right)_0$. These two elements only depend on the link function. Finally, we calculate $\boldsymbol{V}_0$, the function (7) evaluated at $\hat{\boldsymbol{\mu}}_0$, which only depends on the fitted vales $\hat{\boldsymbol{\mu}}$ at each step.

Prediction for new observations is also independent of the choice of $\boldsymbol{X}_{\boldsymbol{w}}$. Given a new case Ω_{n+1}, described by the $1 \times n$ vector $\boldsymbol{\delta}_{n+1}$ of squared distances

from Ω_{n+1} to the n previously known individuals, the predicted $\hat{\eta}_{n+1}$ for Ω_{n+1} is calculated with formula (3) with the quantities of the last IWLS step. Then we can calculate $\hat{\mu}_{n+1} = g^{-1}(\hat{\eta}_{n+1})$.

DB-GLM contains GLM as a particular case: if we start from a $n \times r$ $\boldsymbol{w}$-centered matrix $\boldsymbol{X}_{\boldsymbol{w}}$ of r continuous predictors corresponding to n individuals and we define $\boldsymbol{\Delta}$ as the matrix of squared Euclidean distances between rows of $\boldsymbol{X}_{\boldsymbol{w}}$, then $\boldsymbol{X}_{\boldsymbol{w}}$ is trivially a Euclidean configuration of $\boldsymbol{\Delta}$, hence the DB-GLM hat matrix, response and predictions coincide with the corresponding IWLS quantities of ordinary GLM. In this case, of course, there is no reason for using DB-GLM instead of a regular GLM which would also get an insight into the significance of each covariate. Anyway, remember (Section 1) that situations often arise where the only possibility of knowing relations between statistical units is to compute a distance matrix between them. In these cases it is not possible to use a regular GLM, while it is possible to fit a DB-GLM.

As we said before, the iterative algorithm used to fit the DB-GLM makes internal calls to DB-LM. The estimation of DB-LM requires a value for the effective rank parameter k to be fixed (or a criterion to choose it to be provided). In `dbstats` package the DB-GLM fitting has been implemented (in function `dbglm`, see Sub-section A.1) in such a way that, when a value of the effective rank k is specified, this value is used for all the required internal calls to DB-LM.

The choice of the effective rank k in function `dbglm` is controlled by the argument `method`, that can take five different values: `eff.rank`, `rel.gvar`, `AIC`, `BIC` or `GCV`. When `method` is equal to `eff.rank` the user specifies a value k of the effective rank via the additional argument `eff.rank` of function `dbglm`. When the option is `rel.gvar` the user fixes a proportion of geometric variability and in this way he or she indirectly fixes the value of the effective rank k that provides this proportion. The specific value of proportion of geometric variability is specified with the argument `rel.gvar` of function `dbglm`.

The last three options for argument `method`, `AIC`, `BIC` or `GCV`, allow for an automatic choice of the effective rank k, by the minimization of the criteria (see, e.g., Wood 2006, pages 67 and 174, and Wasserman 2004, page 220)

$$\mathrm{AIC}(k) = -2\sum_{i=1}^{n} \log(f(y_i; g(\hat{\mu}_i(k)), \hat{\phi}(k))) + 2(k+1),$$

$$\mathrm{BIC}(k) = -2\sum_{i=1}^{n} \log(f(y_i; g(\hat{\mu}_i(k)), \hat{\phi}(k))) + \log(n)(k+1),$$

or

$$\mathrm{GCV}(k) = \frac{n\,\mathrm{Dev}(k)}{\left(n - \sum_{i=1}^{n} h_{ii}(k)\right)^2}, \tag{12}$$

respectively, where $\hat{\mu}_i(k)$ and $\hat{\phi}(k)$ are estimated with effective rank equal to k (then $k+1$ is the number of parameters), $\mathrm{Dev}(k)$ is the deviance (8) of this estimated model, and $h_{ii}(k)$ is the i-th element of the diagonal of the hat matrix (1) in the last step of the IWLS when using effective rank k.

Canonical links are assumed in the previous expressions. The likelihood cross-validation criterion (parallel in DB-GLM to the OCV in DB-LM) has not been implemented because it is much more computationally demanding than OCV in DB-LM (see, e.g., page 129 of Wood 2006).

3.1 An example of distance-based generalized linear model fitting

We fit DB-GLM to a data set on Swedish third-party motor insurance in 1977 described in Hallin and Ingenbleek (1983). The file is included in `faraway` package with the name *motorins* (Faraway, 2014a). A subset of these data (registers with Zone = 1) can also be found in Andrews and Herzberg (1985, pp. 413-421). These data correspond to Stockholm, Göteburg and Malmo, and were obtained from a committee study of risk premiums in motor insurance. The total number of observations (for Zone = 1) is $n = 295$, all the non-empty risk groups. The following variables were observed for each group: Payment (total of payments in Skr), Claims (number of claims) and Insured (number of insured, in policy-years).

These data could be used to illustrate premium rating, where risk premiums are calculated as the product of claim frequency times claim severity. Detailed descriptions of the procedure can be found, e.g., in Haberman and Renshaw (1996, Section 8), Brockman and Wright (1992) or Kass et al. (2008, pp. 413-421) among other references. In premiums calculation it is a standard practice to model claim frequency as a GLM with Poisson error structure and claim severity as a GLM with Gamma error structure, using the logarithmic link in both models to obtain a multiplicative tariff.

To illustrate the use of `dbstats` we analyze claim severity, where the response Y is the quotient Payment/Claims and the weights $\boldsymbol{w}$ are proportional to the number of claims. We assume a Gamma error structure with logarithmic link.

Three risk factors are usually considered relevant in modeling claim severity or frequency: Distance (Kilometers Traveled), Bonus (No-claims bonus) and Make (specified car makes). The numbers of levels of each factor are 5, 7 and 9, respectively. The continuous numerical predictors Distance and Bonus, appear as discretized in the published version of the dataset. To illustrate the use of `dbstats` for mixed type predictors we substituted sensible representative values for their factor levels. The codes are:

< 1000 Km per year: 750 Kilometers traveled per year,
1000 - 15000 Km per year: 8000 Kilometers traveled per year,
15000 - 20000 Km per year: 17500 Kilometers traveled per year,
20000 - 25000 Km per year: 22500 Kilometers traveled per year,
> 25000 Km per year: 40000 Kilometers traveled per year.

Bonus is represented by the (arbitrary) numerical codes 1 to 7, equal to the number of years, plus one, since last claim. Make is treated as a nominal categorical variable in Gower's formula (13). It is numerically coded (as 1 to 9) just as a convenience. It represents 9 specified car makes.

Model	Residual deviance	Effective rank
GLM	921.69	10
DB-GLM (`rel.gvar = 1`)	**844.97**	18
DB-GLM (`rel.gvar = 0.90`)	898.73	10
DB-GLM (`method = "GCV"`)	845.07	17

Table 1 *Motorins* data. Results for different fittings of the Gamma model with logarithmic link. The lowest residual deviance values has been highlighted.

The first step in the treatment of these data by DB-GLM is the choice of a suitable metric. In principle it is possible to tailor a metric to reflect specific information on predictors and on how their proximity relates to the particular prediction under study. An omnibus popular metric for mixes of numerical, categorical and binary predictors is the one based on Gower's general similarity coefficient (see Gower, 1971, for further details):

$$s_{ij} = \frac{\sum_{h=1}^{p_1} (1 - |x_{ih} - x_{jh}|/G_h) + a_{ij} + \alpha_{ij}}{p_1 + (p_2 - d_{ij}) + p_3} \tag{13}$$

where p_1 is the number of continuous variables, a_{ij} and d_{ij} are the number of positive and negative matches, respectively, for the p_2 binary variables, and α_{ij} is the number of matches for the p_3 multi-state categorical variables. G_h is the range of the h-th continuous variable. The squared distance is computed as: $\delta_{ij}^2 = 1 - s_{ij}$. Gower (1971) proves that this distance satisfies the Euclidean condition. In our example, $p_1 = 2$, $p_2 = 0$ and $p_3 = 1$ in (13).

When we fit ordinary GLM to these data, we use class marks for continuous predictors (Distance and Bonus) and dummies coding the 9 levels of Make, thus the model has 11 parameters: Intercept, Distance, Bonus plus 8 more, for the Make factor. This GLM model has been fitted using the R function `glm`. The corresponding code can be found as supplementary material (see Annex B). The first row in Table 1 shows the deviance corresponding to the fitted GLM.

For DB-GLM we fit the following versions (the R code fitting them can be found as supplementary material; see Annex B):

1. `rel.gvar = 1`, i.e., a DB-GLM taking into account the whole latent Euclidean configuration, with an effective rank of $k = 18$. See the second row in Table 1 for the resulting deviance.
2. `rel.gvar = 0.90`, i.e., up to a 90% of the total geometric variability. The effective rank in this case is $k = 10$, the same value as the number of explanatory variables (except the constant) in the GLM fitted before. Third row in Table 1 contains the corresponding deviance.
3. `method = "GCV"`, i.e., selecting the effective rank optimizing the GCV statistic (12). In this case we retain $k = 17$ dimensions accumulating a total of 99.46% of the total geometric variability. See fourth row in Table 1 for the resulting deviance.

```
Call:  dbglm(formula = y ~ KmC + BonC + factor(Make), data = Motor1,
    family = Gamma(link = "log"), method = "GCV", full.search = TRUE,
    metric = "gower", weights = w, range.eff.rank = c(1, 18))

Deviance Residuals:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-4.5410 -1.4870 -0.2806 -0.2325  0.8918  4.7470

(Dispersion parameter for Gamma family taken to be 3.310162)

Null deviance:     1024.61  on 294 degrees of freedom
Residual deviance:  845.07  on 277 degrees of freedom

Number of Fisher Scoring iterations: 6
Convergence criterion: DevStat

AIC:        379086.1
BIC:        379152.5
GCV:        0.04
```

Table 2 Summary corresponding to the `dbglm` object obtained when fitting the DB-GLM using `method = "GCV"` to the *motorins* data (Gamma model with logarithmic link).

In these three DB-GLM's we obtain lower residual deviances than the residual deviance of the classical GLM. See Table 1.

The R package `dbstats` includes a summary method for objects of class `dbglm` (see `help(summary.dbglm)` for complete details). As an illustration, Table 2 shows the summary corresponding to the `dbglm` object obtained when fitting the DB-GLM using `method = "GCV"`. Note that the reported degrees of freedom corresponding to the residual deviance (277 in this case) are equal to the number of observations (295) minus the model effective rank (17) minus 1.

In addition to the three previous DB-GLM models, the supplementary materials (see Annex B) include the code for fitting a DB-GLM with Gamma response and logarithmic link, using Euclidean distance and full geometric variability. This is a relevant case since their results (fitted values, deviance residuals, degrees of freedom, etc.) coincide with those of a GLM with same response and link, as we argued before, when we said that GLM is a particular case of DB-GLM. In fact the output of summary of this particular DB-GLM and the summary of the standard GLM (not included here, but reproducible using the code in the supplementary materials) are extremely close to each other (as intended by design of the R package `dbstats`), with the exception of the estimated coefficients for observed predictors, absent in DB modeling.

The package `dbstats` also offers a plot method for objects of class `dbglm` (see `help(plot.dbglm)`). To illustrate it, for the DB-GLM using `method = "GCV"` we exhibit the six available plots: `Residuals vs Fitted` (deviance residuals are used), `Normal Q-Q` (of standardized residuals), `Scale-location` (standardized residuals versus fitted values), `Cook's distance`, `Residuals vs Leverage` (it uses standardized Pearson residuals), and `GCV vs Effective`

`rank`, which can be found in Figure 1. The first five plots are useful for residual analysis and are the same as in `plot.lm` of `stats` package. The fitted values used in these plots can be the fitted link values (by default) or the fitted responses (Figure 1 uses fitted responses). This is a difference with respect to method `plot` for `glm` objects in R, that only uses fitted links. The last plot (also specific for `dbglm` objects) allows us to view the `"GCV"`, `"AIC"` or `"BIC"` criterion according to which the effective rank is chosen and it applies only if the parameter `full.search` is `TRUE`. In this example the result for the effective rank according to the `"GCV"` criterion is $k = 17$.

Regarding the available predict method for objects of class `dbglm`, predicted values may be the expected values of the response for a new data (`type.pred = "response"`), or of the linear predictors evaluated at the estimated `dblm` of the last iteration (`type.pred = "link"`). Also, we can choose the type of the new data, according to the type of data used when fitting the `dbglm` model, as explained in Appendix A.1. See `help(predict.dbglm)` for more details.

4 Local distance-based generalized linear model

In this section we introduce a non-parametric version of the DB-GLM, considered in Section 3 that is based on the same ideas that Loader (1999) uses to extend the GLM to the local likelihood based models.

Consider again the DB-GLM framework. Now we examine the *local DB-GLM*, where *local* refers to the fact that, in a neighborhood of any observed individual, a DB-GLM is assumed to be approximately valid. Let us be more precise. Let $f_Y(y;\eta)$, $\eta \in \mathbb{R}$, be a density function in the exponential family, with $\mu = \mu(\eta) = E(Y)$ when $Y \sim f_Y(y;\eta)$ (we assume that the dispersion parameter is $\phi = 1$, as in Loader 1999, page 61). A 1-1 relation between μ and η is assumed: $\eta = g(\mu)$. We assume that for each observed individual Ω_i, $Y_i \sim f_Y(y_i;\eta_i)$. Let $\mu_i = E(Y_i) = g^{-1}(\eta_i)$.

The DB-GLM assumes that the vector $(\eta_1, \ldots, \eta_n)$, once $\boldsymbol{w}$-centered, belongs to the column space $\mathscr{G}$ of $\boldsymbol{G_w}$. This is equivalent to assume that there exists a vector $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_n)$ that, once $\boldsymbol{w}$-centered, is in the column space $\mathscr{G}$ of $\boldsymbol{G_w}$, such that

$$g(\mu_i) = \eta_i,\ i = 1, \ldots, n.$$

The local version of DB-GLM is defined as follows:

Definition 3 We say that the random variables $Y_1, \ldots, Y_n$ follow a local DB-GLM (local distance-based generalized linear model) when, for any additional individual Ω_{n+1}, there exists a vector $\boldsymbol{\eta}^{n+1} = (\eta_1^{n+1}, \ldots, \eta_n^{n+1})$ that, once $\boldsymbol{w}$-centered, is in the column space of $\boldsymbol{G_w}$ and verifies that

$$g(\mu_i) = \eta_i^{n+1} + o(\delta(\Omega_i, \Omega_{n+1})),\ i = 1, \ldots, n,$$

that is, $\lim(|g(\mu_i) - \eta_i^{n+1}|/\delta(\Omega_i, \Omega_{n+1})) = 0$ as $\delta(\Omega_i, \Omega_{n+1})$ goes to zero.

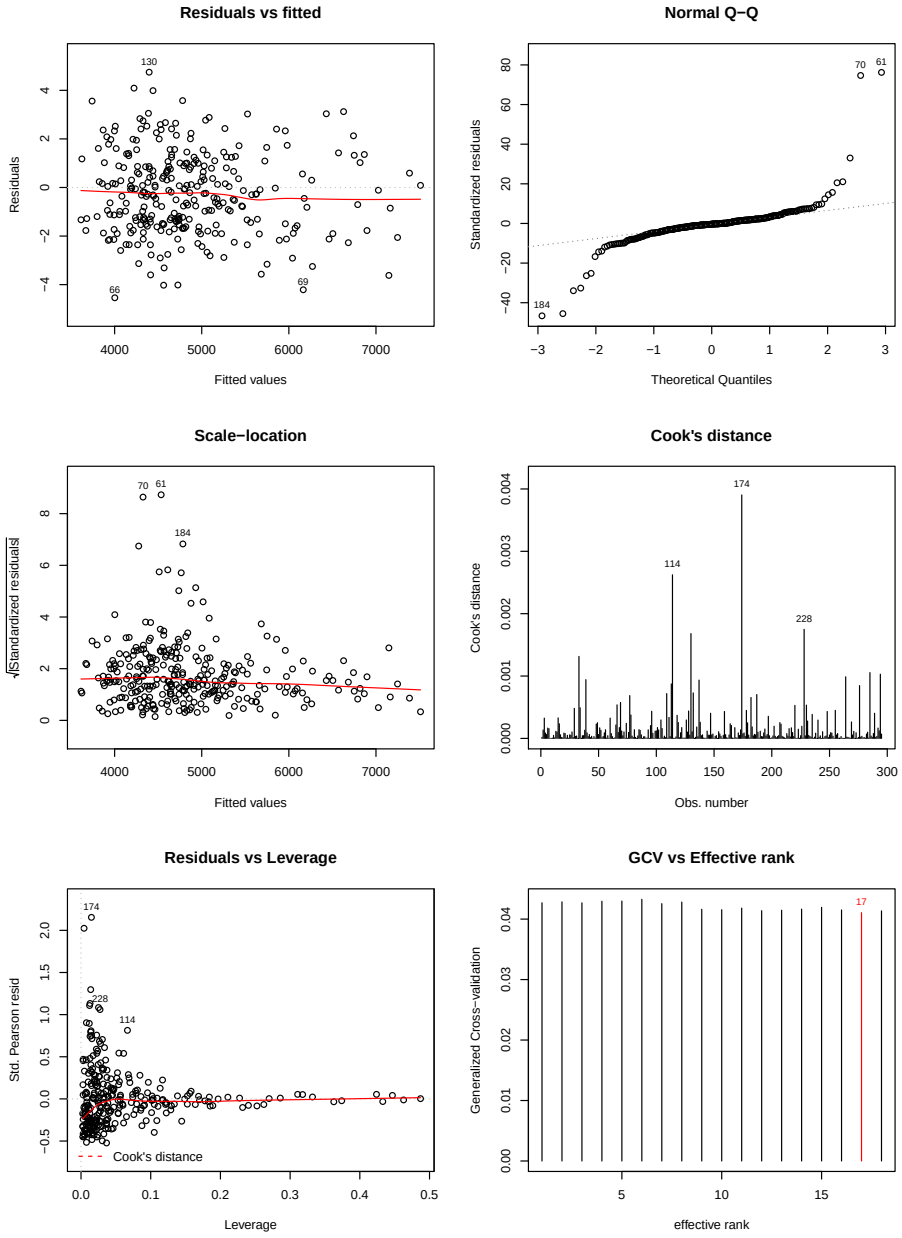

Fig. 1 Plots of `Residuals vs Fitted` (deviance residuals versus fitted responses), `Normal Q-Q`, `Scale-location`, `Cook's distance`, `Residuals vs Leverage`, and `Effective rank of GCV method` for DB-GLM with Gamma response and logarithmic link, using Gower's distance and fitted with `method = "GCV"`.

In other words, Y follows a local DB-GLM when a DB-GLM is a good approximation for the true distribution of Y in the neighborhood of any additional individual Ω_{n+1} .

Let μ_{n+1} be the expected value of the response Y corresponding to the individual Ω_{n+1}. This is the value we want to estimate and we do that by fitting a DB-GLM that only uses the information provided by observed objects Ω_i, $i = 1, \ldots, n$, that are *close* to Ω_{n+1}. The idea is to translate to the DB-GLM context the principles of *local likelihood*, as stated in Loader (1999, Chapter 4) (see also Section 3.4 in Bowman and Azzalini 1997, Section 6.5 in Hastie et al. 2009, Section 5.10 in Wasserman 2006, or the book of Wood 2006). Our approach parallels that used in Boj et al. (2010), where local DB-LM was defined.

We consider the weights

$$\upsilon_i(\Omega_{n+1}) = K(\delta(\Omega_{n+1}, \Omega_i)/h) / \sum_{j=1}^{n} K(\delta(\Omega_{n+1}, \Omega_j)/h), \tag{14}$$

where K is a kernel function (a non-negative function, decreasing in $[0, \infty)$ with $\lim_{u\to\infty} K(u) = 0$) and $h > 0$ is a smoothing parameter (depending on n). Let $\boldsymbol{\upsilon} = (w_i \cdot \upsilon_i(\Omega_{n+1}))_{i=1,\ldots,n}$ (where w_i is the original weight of individual i). The local log-likelihood function is defined as

$$\mathscr{L}_{\Omega_{n+1}}(\boldsymbol{\eta}) = \sum_{i=1}^{n} \upsilon_i(\Omega_{n+1}) \log(f_Y(y_i; \eta_i))$$

for each $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_n)$ such that $(\boldsymbol{\eta} - \mathbf{1}\boldsymbol{\upsilon}^\top \boldsymbol{\eta}) \in \mathscr{G}_{\boldsymbol{\upsilon}}$, the column space of $\boldsymbol{G}_{\boldsymbol{\upsilon}}$. It is easy to prove that

$$(\boldsymbol{\eta} - \mathbf{1}\upsilon^\top \boldsymbol{\eta}) \in \mathscr{G}_{\boldsymbol{\upsilon}} \Leftrightarrow (\boldsymbol{\eta} - \mathbf{1}\boldsymbol{w}^\top \boldsymbol{\eta}) \in \mathscr{G}_{\boldsymbol{w}} \Leftrightarrow (\boldsymbol{\eta} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top \boldsymbol{\eta}) \in \mathscr{G}.$$

To maximize the local likelihood function, we fit the DB-GLM defined by the elements

$$\boldsymbol{y} = (y_i)_{i=1,\ldots,n},\ \boldsymbol{\upsilon} = (w_i \cdot \upsilon_i(\Omega_{n+1}))_{i=1,\ldots,n},\ \boldsymbol{\Delta} = (\delta(\Omega_i, \Omega_j)^2)_{i=1,\ldots,n, j=1,\ldots,n}, \tag{15}$$

as responses, weights and squared distances matrix, respectively. Running the IWLS algorithm we obtain the local maximum likelihood estimator of $\boldsymbol{\eta}$, say $\hat{\boldsymbol{\eta}}^{n+1}$. Let

$$\boldsymbol{\delta}_{n+1} = (\delta(\Omega_{n+1}, \Omega_1)^2, \ldots, \delta(\Omega_{n+1}, \Omega_n)^2) \tag{16}$$

be the squared distances from Ω_{n+1} to other individuals Ω_i. By analogy with equation (3), the prediction $\hat{\eta}_{n+1}^{n+1}$ for the linear predictor $\eta_{n+1} = g(\mu_{n+1})$ is

$$\hat{\eta}_{n+1}^{n+1} = \boldsymbol{\upsilon}^\top \hat{\boldsymbol{\eta}}^{n+1} + \frac{1}{2}(\boldsymbol{g}_{\boldsymbol{\upsilon}} - \boldsymbol{\delta}_{n+1}) \cdot \left(\boldsymbol{D}_{\boldsymbol{\upsilon}}{}^{1/2} \cdot \boldsymbol{F}_{\boldsymbol{\upsilon}}{}^{+} \cdot \boldsymbol{D}_{\boldsymbol{\upsilon}}{}^{1/2}\right) \cdot \left(\hat{\boldsymbol{\eta}}^{n+1} - \mathbf{1}\boldsymbol{\upsilon}^\top \hat{\boldsymbol{\eta}}^{n+1}\right),$$

and the local DB-GLM estimator of μ_{n+1} is

$$\hat{\mu}_{n+1} = g^{-1}(\hat{\eta}_{n+1}^{n+1}).$$

When fitting the local DB-GLM, the distance function $\delta(\Omega_i, \Omega_j)$ has two different roles. First, in equation (14) it is used to define the weights $\upsilon_i(\Omega_{n+1})$ that give the local character to the whole procedure. Second, in equations (15) and (16), distance δ is used to fit the weighted DB-GLM. Both roles are unrelated and in fact it is possible to work with two different distance functions involved in the local DB-GLM: one of them, say δ_1, used in equation (14), and the other, say δ_2, in equations (15) and (16). In the context of local DB-LM, Boj et al. (2010) show that using two distance functions provides more flexibility than using only one ($\delta_1 = \delta_2$). For instance, in the case of univariate explanatory variable, x_i, a local cubic polynomial fitting can be implemented using the local DB-LM with

$$\delta_1(x_i, x_j) = |x_i - x_j| \text{ and } \delta_2(x_i, x_j) = \|(x_i, x_i^2, x_i^3) - (x_j, x_j^2, x_j^3)\|,$$

$\|\cdot\|$ being the Euclidean norm in $\mathbb{R}^3$.

It is worth to mention that most of the diagnostic tools introduced in Loader (1999, Section 4.3) are valid in the local DB-GLM: deviance definition, different types of residuals, influence function, fitted degrees of freedom, likelihood cross-validation and AIC. In general, all definitions given in local likelihood that are not based directly on the local estimated parameters are still valid in local DB-GLM. In particular the likelihood cross-validation and the AIC criteria can be used to select the value of the bandwidth parameter h in local DB-GLM. For instance, for a given h the corresponding AIC is defined as

$$\text{AIC}(h) = -2\sum_{i=1}^{n} w_i \log(f(y_i; g(\hat{\mu}_i(h)))) + 2\nu(h),$$

where $\nu(h)$ is the number of fitted degrees of freedom and $\hat{\mu}_i(h)$ is the local DB-GLM estimation of μ_i, both corresponding to bandwidth h. The bandwidth choice can be done minimizing AIC(h) in h. Another possibility is using the BIC criterion, that is defined as the AIC, but replacing the penalty $2\nu(h)$ by a larger one: $\log(n)\nu(h)$. Both methods of bandwidth selection, AIC and BIC, are implemented in the package `dbstats`. The likelihood cross-validation criterion, much more computationally demanding, has not been implemented. As an alternative we have included the GCV criterion, following the suggestion of Wood (2006, equation 4.24).

The local DB-GLM has an extra tuning parameter that is not present in the local likelihood model developed in Loader (1999): the *effective rank* k, the number of linearly independent Euclidean coordinates used in prediction at each DB-LM performed throughout the iterations of the IWLS algorithm. This extra parameter lends more flexibility to local DB-GLM at the risk of falling into over-fitting. A possible way to avoid over-fitting is to choose jointly both parameters, h and k, by minimization of the AIC (or BIC or GCV) criterion, that in fact depends on both: AIC(h, k). Another possibility is to fix k and then choose h as explained before. Alternatively, the effective rank k can be indirectly specified by fixing the proportion of geometric variability that each DB-LM should explain. Values between 0.9 and 0.99 for this proportion are

usually adequate. In the package `dbstats` the last two possibilities have been implemented.

4.1 An example of local distance-based generalized linear model fitting

In this section we make an example using local DB-GLM with functional data as explanatory variable and a binary response. We say that an observed variable is functional when a whole function is registered for each individual in the sample (see Ramsay and Silverman 2005 for a general perspective on Functional Data Analysis and Ferraty and Vieu 2006a for a nonparametric approach).

We consider the near infrared (NIR) spectral data set contending with wheat samples that was described in Kalivas (1997). This data set contains data from 100 wheat samples. The available information for each sample consists of two scalar measures (protein and moisture contents; only protein content is used here) and a functional variable, the NIR spectra: samples were measured using diffuse reflection in units of log inverse reflectance $\log(1/R)$ at wavelengths going from 1100 to 2500 nm in 2 nm intervals (reflectance refers to the fraction of incident electromagnetic power that is reflected by the sample; see Brenchley et al. 1997 for more details about NIR measurements). The protein and spectrum data were available for years at `ftp://ftp.clarkson.edu/users/h/o/hopkepk/chemdata/kalivas/` (we accessed in May 2011), at files `protein.asc` and `whtspec.asc`, respectively. Nevertheless Clarkson University stopped running FTP servers around 2013. So the data sets are no longer available there. Therefore we provide these two files as supplementary material with the permission of John H. Kalivas (see Annex B).

Let us define the binary variable y indicating for each wheat sample in the data set whether its protein content is over the median value ($y = 1$) or not ($y = 0$). Our goal is to predict the variable `y` using the NIR spectra function as predictor.

We use the R package `fda.usc` (Febrero-Bande and Oviedo, 2014) to deal with NIR spectra data as functional data. The corresponding code can be found as supplementary material (see Annex B). In particular `fda.usc` has been used to plot the spectra functions, as well as their first and second derivatives, in Figure 2. Wheat samples have been colored according to the value of the binary variable y (gray continuous for $y = 1$, black dotted for $y = 0$). From the figure it is not obvious how NIR spectra functions or their derivatives could allows us to predict the value of y, the indicator of high protein content.

We compare the performance of the following binary prediction tools, all of them using functional predictors:

- FGLM: Functional Generalized Linear Model, a version of the GLM with functional predictor using basis representation. We use the implementation of FGLM available as function `fregre.glm` in package `fda.usc`.
- DB-GLM: Distance based generalized linear model, developed in Section 3.

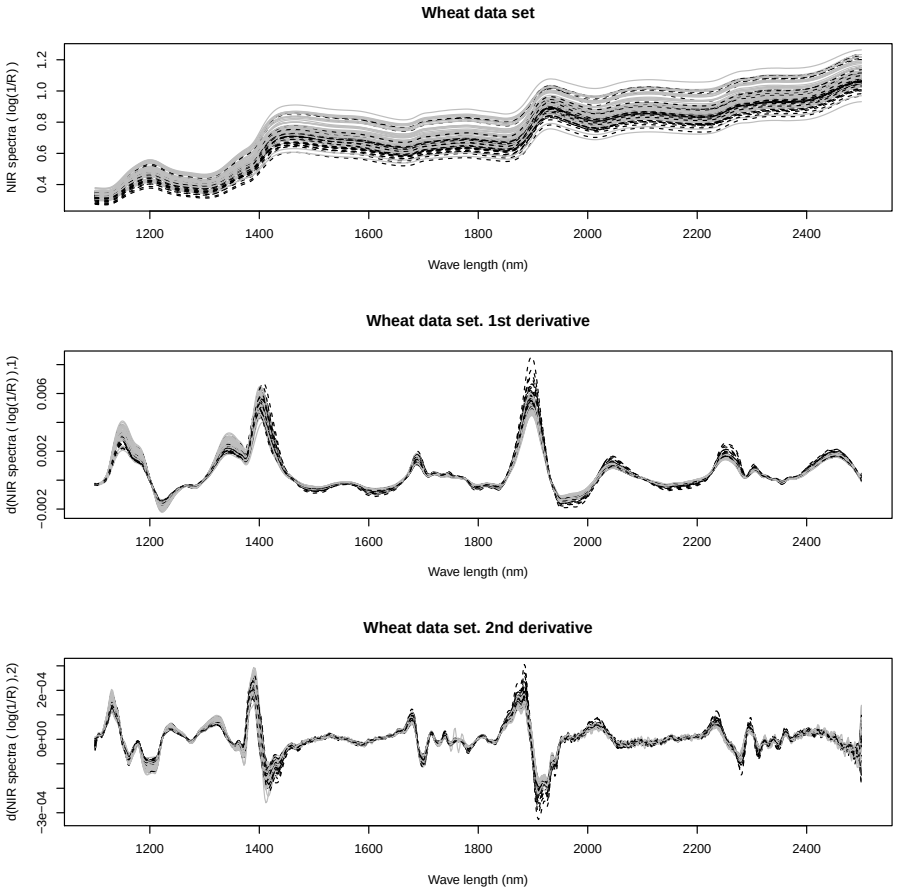

Fig. 2 Wheat data set. NIR spectra functions, jointly with their first and second derivatives. Functions represented with gray continuous lines correspond to wheat samples with protein content over the median. Black dotted lines are used otherwise.

– PSSR: P-spline signal regression, as defined by Marx and Eilers (1999). It is a different way of fitting a generalized linear model with functional explanatory variable. The main difference with the previous alternative is that in P-spline signal regression only the functional coefficient is assumed to be a smooth function (and therefore represented by splines). The work of Marx and Eilers (1999) has its motivation in chemometric data (in fact they use wheat protein data as an example, with the original continuous response variable). We use the function `signal.fit`, downloaded from Marx's web page (`http://www.stat.lsu.edu/faculty/marx/signal.txt`) on May 15, 2014. The default values are used for the parameters of this function.
– FGAM: Functional Generalized Additive Model, introduced by Mclean et al. (2014). This model is the natural extension of the generalized addi-

	Prediction tool				
	FGLM	DB-GLM	PSSR	FGAM	Local DB-GLM
Functional predictor:					
NIR spectra functions	54	29	24	33	27
First derivative	58	25	26	**20**	29
Second derivative	62	24	**20**	30	28

Table 3 Number of bad classified wheat samples by leave-one-out cross-validation. Observe that these figures must be integers numbers between 0 and $n = 100$ (the sample size) and that a random classifier would fail at 50 cases in average. The best performances have been highlighted.

tive models (GAMs) to functional data when there is only one explanatory functional variable, as argued by Mclean et al. (2014). FGAM has been implemented as function `fgam` in R package `refund` Crainiceanu et al. (2014) with its parameters in their default values. For an extension of GAMs to the case of multiple functional covariates see Febrero-Bande and González-Manteiga (2013).

– Local DB-GLM: Local distance based generalized linear model, developed in Section 4.

For each of these five prediction tools we have used three different functional predictors: NIR spectra functions, their first derivatives and their second derivatives (in P-spline signal regression we use first and second differences instead). In order to measure the prediction ability of the 15 prediction rules under consideration we compute the number of bad classified wheat samples when applying leave-one-out cross-validation. The Table 3 shows the results. The R code used to fit these model can be found as supplementary material (see Annex B).

It follows from Table 3 that in this dataset FGLM presents significantly worst results than other methods. A possible explanation for that is our arbitrary choice of the number of elements in the basis of B-splines (we have chosen 18 basis functions) that could be improved by using an automatic model choice criterion as GCV, AIC or BIC.

In order to fit a DB-GLM with the function `dbglm` the first step is to compute the inter-individual distance matrix. When dealing with functional data our choice is to use one of the semimetrics defined in Ferraty and Vieu (2006a) as they are implemented in their own R library NPFDA (Ferraty and Vieu 2006b), `http://www.lsp.ups-tlse.fr/staph/npfda/`, also available at the package `fda.usc` (see functions `metric.lp`, `semimetric. basis` and `semimetric.NPFDA` in this package). In particular here we use L^2 distances between NIR spectra functions or their derivatives calculated after representing functions in a B-spline basis. The effective rank has been chosen according to the GCV criterion using all the data. The resulting values are 6, 8 and 8, respectively, when using the spectra functions, their first or second derivatives. These effective ranks values have been fixed for all the DB-GLM fits in the leave-one-out process. We fit the local DB-GLM with the function `ldbglm` of package `dbstats`. We use again L^2 distances between NIR spectra functions

or their derivatives. The automatic choice of the smoothing parameter h is done with the Generalized Cross Validation criterion for the whole dataset. The resulting optimal bandwidths have been fixed for all the local DB-GLM fits in the leave-one-out process.

The results of DB-GLM, PSSR, FGAM and local DB-GLM fits (second to fourth columns of Table 3) are better than those of FGLM and they are comparable with each other. It seems that the performance of P-spline signal regression is slightly better than FGLM, FGAM or DB-GLM, mainly when using derivatives.

Observe that, among these four functional regression, two of them are global logistic estimators (DB-GLM and PSSR) and the other two (FGAM and local DB-GLM) have local (or non-parametric) character. The fact that all of them behave in a similar way indicates that a global logistic fit is appropriate for this particular example. FGLM (also a global logistic method) does not perform as well as DB-GLM or PSSR because it is less flexible than the other two (at least with the chosen combination of tuning parameters).

In order to explore the performance of these functional regression methods when the relationship between the explanatory functional data and the binary response is not global logistic, we have simulated alternative responses according to a non-logistic pattern as follows. First we compute the first and second functional principal components of the wheat spectra functions (we use the function `fdata2pc` from package `fda.usc`). Let $w_{i,j}$ the score of functional data i in the principal direction j, for $i = 1, \ldots, n = 100$ and $j = 1, 2$, once standardized to have variance 1 among individuals. Then, for $i = 1, \ldots, n = 100$, we define y_i equal to 1 when

$$\left(\frac{1}{2} + \frac{1}{2}\cos(2w_{1,i})\sin(2w_{2,i})\right)^{\alpha} > \frac{1}{2},$$

and 0 otherwise. The value α is fixed at the value 0.95 in order to have 50 values of $y_i = 1$ and the other 50 $y_i = 0$.

The results of the leave-one-out cross-validation procedure are shown in Table 4. The results corresponding to functional GLM are not included because this model (as well as the DB-GLM with effective rank equal to 1) has a very bad performance (almost 100% of bad classified observations) that is due to their lack of flexibility. Both estimators are unable to fit a such so highly non-linear binary regression function. Then both provide a regression function that are almost constant: the estimated conditional probability of y_i being 1 is approximately equal to the proportion of 1's in the training set. That is, in this non-logistic example GLM and DB-GLM (with effective rank equal to 1) are equivalent to the majority vote rule. It is easy to check that in a data set with exactly 50% of the data in each of two classes, the majority vote rule provide a 100% of bad classified cases when applying leave-one-out cross-validation (when a data is removed from the sample the majority of the remaining data are always of the other class).

Table 4 shows that, in this non-logistic situation, the local (or nonparametric) methods, FGAM and local DB-GLM, has better performance that the

	Prediction tool			
	DB-GLM	PSSR	FGAM	Local DB-GLM
Functional predictor:				
NIR spectra functions	48	43	24	**19**
First derivative	45	47	25	**19**
Second derivative	57	57	51	43

Table 4 Non-logistic simulated data from wheat spectra functions. Number of bad classified samples by leave-one-out cross-validation. Observe that these figures must be integers numbers between 0 and $n = 100$ (the sample size) and that a random classifier would fail at 50 cases in average. The best performances have been highlighted.

globals methods DB-GLM (in this case the chosen effective rank is 2) and PSSR. Moreover DB-GLM seems to be preferable to FGAM.

An additional advantage of local DB-GLM over FGAM or PSSR is that local DB-GLM can be applied to any kind of data, as long as we are able to compute distances between them, but FGAM or PSSR are useful only for functional data. For instance, we can fit a local DB-GLM to the *motorins* insurance data introduced in Section 3.1. In this example the local DB-GLM obtains a residual deviance equal to 733.68 (much lower than the results obtained with parametric fits; see Table 1).

5 Conclusions

We have extended distance-based regression in two directions. First, we introduce the distance-based generalized linear model, in an analogous way to that that brings from the linear model to the generalized linear model. Second, we use local likelihood ideas to propose a nonparametric version of the distance-based generalized linear model. The R package `dbstats` implements these an others distance-based regression methods. We have used this library to fit the proposed models to two real data examples. They show that both (global and local) distance-based generalized linear models compete well with alternative methods. Moreover our proposals have the additional advantage that they are applicable to a wide range of data types, while some of the alternative models have being designed for specific contexts (for instance, functional GLM works only for functional data).

A The `dbstats` package

The `dbstats` package (Boj et al., 2014) for R (R Development Core Team, 2015) implements several distance-based prediction methods. The main functions of `dbstats` are: `dblm` for DB-LM, `ldblm` for local DB-LM, `dbglm` for DB-GLM, `ldbglm` for local DB-GLM and `dbplsr` for DB-PLSR.

In sub-section A.1 we describe the usage of function `dbglm` whereas `ldbglm` is described in sub-section A.2. Two examples illustrating the usage of these functions from a user perspective have been presented, respectively, in sub-sections 3.1 and 4.1. For details of `dblm`, `ldblm` and `plsr` we refer to Boj et al. (2014).

A.1 Function dbglm

Function **dbglm** fits DB-GLM. In this function distances can be directly provided as: an interdistances matrix (class **dist** or **dissimilarity** as in **stats** package); a squared interdistances matrix (class **D2**); or an inner-products matrix (class **Gram**). Classes **D2** and **Gram** have been implemented in the **dbstats** package. It is also possible to compute distances directly from observed explanatory variables (using a class **formula** object in the call to **dbglm**).

The **dbstats** package does not provide specific methods for computing distances, depending instead on other available functions and packages such as **dist** in the **stats** package, **daisy** in the **cluster** package (Maechler, 2015) or **dist** in the **proxy** package (Meyer and Buchta, 2015). Utility functions such as **as.D2**, **as.Gram**, **D2toDist**, **D2toG**, **distoD2** and **GtoD2** allows the user mutual interconversions (see Boj et al. 2014 for details).

Response and link function are as in the **glm** function of **stats** for ordinary GLM.

The usage of **dbglm** is:

```
dbglm(distance, y, family = gaussian, method = "GCV", full.search = TRUE,
    weights, maxiter = 100, eps1 = 1e-10, eps2 = 1e-10, rel.gvar = 0.95,
    eff.rank = NULL, offset, mustart = NULL, range.eff.rank, ...)
```

where the argument *distance* is of class **dist** or **dissimilarity**. The same information can be provided replacing *distance* by an object of class either **D2** or **Gram**.

When calling **dbglm** using an object of class **formula**, the first and second arguments in the previous call to **dbglm** are replaced by other three arguments: **formula** (of the form $\boldsymbol{y}\sim\boldsymbol{Z}$), **data** (a data frame containing the variables in the model: both response $\boldsymbol{y}$ and explanatory variables $\boldsymbol{Z}$) and **metric** (that indicates how to compute distances between the rows of $\boldsymbol{Z}$; it must be one of the strings **"euclidean"** (the default), **"manhattan"** or **"gower"** to be passed to function **daisy** of **cluster** package).

In addition to the response **y**, the distance matrix **distance** (or equivalent information), the formula **formula**, data **data** and **metric**, it is worth mentioning the following arguments of **dbglm**:

family, weights, offset, mustart are arguments with the same role that they have in the **glm** function.

method sets the method to be used in deciding the *effective rank*. There are five different methods: **"AIC"**, **"BIC"**, **"GCV"** (default), **"eff.rank"** and **"rel.gvar"**. See Section 3 (before Subsection A.1) for details on these criteria.

range.eff.rank a vector defining the range of possible values for the effective rank in the dblm iterations to be evaluated when **method** is **"AIC"**, **"BIC"** or **"GCV"**. It should be restricted between $c(1, n-1)$.

full.search sets the optimization procedure to be used to minimize the modelling criterion specified in **method** when **"AIC"**, **"BIC"** or **"GCV"** criteria are specified. See the help of of **dbstats** package Boj et al. (2014) for details.

rel.gvar relative geometric variability (a real number between 0 and 1; default is 0.95). More details can be found at the end of Section 3 (before Subsection A.1).

eff.rank integer between 1 and $n-1$. If specified its value overrides **rel.gvar**. When **eff.rank = NULL** (default), calls to **dblm** are made with **method = "rel.gvar"**. More details can be found in Section 3 (before Subsection 3.1).

maxiter, eps1, eps2 are stopping criteria for the iterative algorithm that fits the DB-GLM.

The function returns a list of class **dbglm** containing the following components:

Common elements with the output of **glm** function for R: residuals, fitted.values, family, deviance, aic.model, null.deviance, iter, prior.weights, weights, df.residual, df.null, y, call.

H hat matrix projector of the last **dblm** iteration.

convcrit convergence criterion. One of: **"DevStat"** (stopping criterion 1: when the relative decrement of deviance in one step is less than **eps1**), **"muStat"** (stopping criterion 2: when the relative change of the estimated expected values of the responses in one step is less than **eps2**), **"maxiter"** (maximum allowed number of iterations has been exceeded).

eff.rank, rel.gvar effective rank and relative geometric variability that have been finally used.

bic.model, gcv.model BIC and GCV criteria of the final DB-GLM.

dev.resids deviance residuals (the way they are computed depends on the specified **family**).

varmu vector of estimated variance of each observation (that depends on the estimated vector of expected values and on the specified **family**).

A.2 Function `ldbglm`

Function `ldbglm` is a localized version of a DB-GLM. As in the global model `dbglm`, explanatory information is coded as distances between individuals, that can either be computed from observed explanatory variables or directly provided to function `ldbglm` as a (possibly squared) interdistances matrix or as a inner products matrix (a `Gram` matrix).

Remember that in local DB-GLM there appear two distance functions, δ_1 and δ_2, playing different roles. Accordingly, function `ldbglm` has two different arguments, `dist1` and `dist2` of class `dist` or `dissimilarity`, where distances δ_1 and δ_2 are specified: `dist1` defines the neighborhood delimiting what observations (and with what weight) are used when locally fitting a DB-GLM, whereas `dist2` (which may coincide with `dist1`) is used specifically for fitting this weighted DB-GLM.

The usage of `ldbglm` is:

```
ldbglm(dist1, dist2 = dist1, y, family = gaussian(), kind.of.kernel = 1,
       method.h = "GCV", weights, user.h = quantile(dist1, .25) ^ .5,
       h.range = quantile(as.matrix(dist1), c(.05,.25)) ^ .5, noh = 10,
       k.knn = 3, rel.gvar = 0.95, eff.rank = NULL, maxiter = 100,
       eps1 = 1e-10, eps2 = 1e-10, ...)
```

In the same way that it was explained in the overview of function `dbglm`, the predictive information contained in distance matrices `dist1` and `dist2` can be provided to function `ldbglm` in three alternative ways: two squared distances matrices (`D2.1` and `D2.2`), two inner products matrices (`G1` and `G2`), or a formula jointly with a dataset and two metrics (`formula`, `data`, `metric1` and `metric2`).

The following are other arguments of `ldbglm` that are specific of this function because they control its local character:

kind.of.kernel integer number between 1 and 6 which determines the user's choice of smoothing kernel K (see equation 14): (1) Epanechnikov (Default), (2) Biweight, (3) Triweight, (4) Normal, (5) Triangular, (6) Uniform.

method.h sets the method to be used when choosing the *bandwidth* h to be used in equation (14). There are four different methods, `AIC`, `BIC`, `GCV` (default) and `user.h`. `AIC`, `BIC` and `GCV` take the bandwidth minimizing the Akaike or Bayesian Information Criterion or the generalized cross-validation, respectively. When method is `user.h`, the bandwidth is explicitly set by the user through the `user.h` optional parameter which, in this case, becomes mandatory.

user.h global bandwidth set by the user. The default value is the first quartile of all the distances $d(i, j)$ in matrix `dist1`. It applies only if `method = "user.h"`.

h.range a vector of length 2 giving the range for automatic bandwidth choice. (Default value: quantiles 0.05 and 0.5 of $d(i, j)$ in matrix `dist1`). It applies when `method != "user.h"`.

noh number of bandwidth `h` values within `h.range` for automatic bandwidth choice. It applies when `method != "user.h"`.

k.knn minimum number of observations with positive weight in any local fit of a DB-GLM model. A too small value of bandwidth `h` could originate a neighborhood with only one observation producing a runtime error when trying to fit a local fit of a DB-GLM model. Choosing `k.knn > 1` prevents from this problem. By default `k.knn = 3`.

The function returns a list of class `ldbglm` containing the following components:

Common elements with the output of `dbglm`: residuals, fitted.values, family, weights, y, call.

dist1, dist2 the distances matrices used to calculate the local weights of the observations and to locally fit the `dbglm`'s, respectively.

h.opt the optimal bandwidth `h` used in the fitting process (if `method != user.h`).

S the smoothing matrix in the last iteration of the IRWLS. See Boj et al. (2010) for details on the definition of the smoothing matrix.

B Supplementary material

Code for fitting DB-GLM and local DB-GLM is available in the R package `dbstats` (`http://CRAN.R-project.org/package=dbstats`). Additional code for reproducing the computations and graphics in the paper are included in the R script `ExamplesDB.R`. The data files `protein.asc` and `whtspec.asc` are provided as supplementary material with the permission of John H. Kalivas, Idaho Satate University.

Acknowledgments

We appreciate very much the efforts that Philip K. Hopke, Clarkson University, is doing to make again publicly available the data sets described in Kalivas (1997). We are very grateful to John H. Kalivas for allowing us to add the `protein.asc` and `whtspec.asc` data files as supplementary material of this paper. Work supported in part by the Spanish Ministerio de Educación y Ciencia and FEDER, grants MTM2010-17323, MTM2010-14887, MTM2013-43992-R and MTM2014-56535-R.

References

Andrews, D. F. and A. M. Herzberg (1985). *Data. A Collection of Problems From Many Fields for the Student and Research Worker*. New York: Springer-Verlag.

Banks, D. and K. Carley (1994). Metric inference for social networks. *J. Classif. 11*(1), 121–149.

Boj, E., A. Caballé, P. Delicado, and J. Fortiana (2014). `dbstats`*: Distance-Based Statistics (dbstats)*. R package version 1.0.4.

Boj, E., M. M. Claramunt, and J. Fortiana (2007). Selection of predictors in distance-based regression. *Commun. Stat. A-Theor. 36*, 87–98.

Boj, E., M. M. Claramunt, A. Grané, and J. Fortiana (2007). Implementing pls for distance-based regression: Computational issues. *Computation. Stat. 22*, 237–248.

Boj, E., P. Delicado, and J. Fortiana (2010). Local linear functional regression based on weighted distance-based regression. *Comput. Stat. Data An. 54*, 429–437.

Borg, I. and P. Groenen (2005). *Modern Multidimensional Scaling: Theory and Applications (2nd ed)*. New York: Springer-Verlag.

Bowman, A. and A. Azzalini (1997). *Applied Smoothing Techniques for Data Analysis*. Oxford: Oxford University Press.

Brenchley, J. M., U. Hörchner, and J. H. Kalivas (1997). Wavelength selection characterization for nir spectra. *Appl. Spectrosc. 51*, 689–699.

Brockman, M. J. and T. S. Wright (1992). Statistical motor rating: Making effective use of your data. *J. Inst. Actuar. 119*(3), 457–543.

Buja, A., D. F. Swayne, M. L. Littman, N. Dean, H. Hofmann, and L. Chen (2008). Data visualization with multidimensional scaling. *J. Comput. Graph. Stat. 17*(2), 444–472.

Butts, C. T. and K. M. Carley (2001). Multivariate methods for inter-structural analysis. Casos working paper, Carnegie Mellon University.

Butts, C. T. and K. M. Carley (2005). Some simple algorithms for structural comparison. *Comput. Math. Organ. Th. 11*(4), 291–305.

Crainiceanu, C., P. Reiss, J. Goldsmith, L. Huang, L. Huo, and F. Scheipl (2014). `refund`*: Regression with Functional Data Computing (refund)*. R package version 0.1-11.

Cuadras, C. and C. Arenas (1990). A distance-dased regression model for prediction with mixed data. *Commun. Stat. A-Theor. 19*, 2261–2279.

Cuadras, C. M. (1989). Distance analysis in discrimination and classification using both continuous and categorical variables. In Y. Dodge (Ed.), *Statistical Data Analysis and Inference*, Amsterdam, pp. 459–473. North-Holland.

Cuadras, C. M., C. Arenas, and J. Fortiana (1996). Some computational aspects of a distance-based model for prediction. *Commun. Stat. B-Simul. 25*, 593–609.

Esteve, A., E. Boj, and J. Fortiana (2009). Interaction terms in distance-based regression. *Commun. Stat. A-Theor. 38*, 3498–3509.

Everitt, B. S., S. Landau, M. Leese, and D. Stahl (2011). *Cluster Analysis* (5th ed.). Wiley.

Faraway, J. (2014a). `faraway`*: Functions and Datasets for Books by Julian Faraway*. R package version 1.0.6.

Faraway, J. (2014b). Regression for non-Euclidean data using distance matrices. *J. Appl. Stat. 41*(11), 23422357.

Febrero-Bande, M. and W. González-Manteiga (2013). Generalized additive models for functional data. *Test 22*(2), 278–292.

Febrero-Bande, M. and M. Oviedo (2014). `fda.usc`*: Functional Data Analysis and Utilities for Statistical Computing (fda.usc)*. R package version 1.2.1.

Ferraty, F. and P. Vieu (2006a). *Non Parametric Functional Data Analysis. Theory and Practice*. New York: Springer-Verlag.

Ferraty, F. and P. Vieu (2006b). *Reference Manual for Implementing NonParametric Functional Data Analysis (NPFDA)*. Companion manual of the book: NonParametric Functional Data Analysis: Theory and Practice. Springer-Verlag, New York.

Gower, J. C. (1968). Adding a point to vector diagrams in multivariate analysis. *Biometrika 55*, 582–585.

Gower, J. C. (1971). A general coefficient of similarity and some of its properties. *Biometrics 27*, 857–874.

Green, P. J. (1984). Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives. *J. Roy. Stat. Soc. B Meth. 46*(2), 149–192.

Haberman, S. and A. E. Renshaw (1996). Generalized linear models and actuarial science. *J. Roy. Stat. Soc. D Sta. 45*(4), 407–436.

Hallin, M. and J. F. Ingenbleek (1983). The Swedish automobile portfolio in 1977. a statistical study. *Scand. Actuar. J. 83*, 49–64.

Hastie, T., R. Tibshirani, and J. Friedman (2009). *The Elements of Statistical Learning. Data Mining, Inference, and Prediction (2nd ed.)*. New York: Springer-Verlag.

Kalivas, J. H. (1997). Two data sets of near infrared spectra. *Chemometr. Intell. Lab. 37*, 255–259.

Kass, R., M. Goovaerts, J. Dhaene, and M. Denuit (2008). *Modern Actuarial Risk Theory Using R (2nd ed)*. Berlin: Springer-Verlag.

Loader, C. (1999). *Local Regression and Likelihood*. New York: Springer-Verlag.

Maechler, M. (2015). `cluster`*: Cluster Analysis Extended Rousseeuw et al.* R package version 2.0.1.

Marx, B. D. and P. H. C. Eilers (1999). Generalized linear regression on sampled signals and curves: A p-spline approach. *Technometrics 41*(1), 1–13.

McCullagh, P. and J. A. Nelder (1989). *Generalized Linear Models (2nd ed)*. London: Chapman and Hall.

Mclean, M. W., G. Hooker, A. M. Staicu, F. Scheipl, and D. Ruppert (2014). Functional generalized additive models. *J. Comput. Graph. Stat. 23*(1), 249–269.

Meyer, D. and C. Buchta (2015). `proxy`*: Distance and Similarity Measures*. R package version 0.4-14.

R Development Core Team (2015). *R: A Language and Environment for Statistical Computing*. Vienna, Austria.

Ramsay, J. O. and B. W. Silverman (2005). *Functional Data Analysis (2nd ed)*. New York: Springer-Verlag.

Rao, C. R. (1973). *Linear Statistical Inference and Its Applications*. New York: John Wiley & Sons.

Stewart, G. W. (1993). On the early history of the singular values decomposition. *SIAM Rev. 35*, 551–566.

Street, J. O., R. J. Carroll, and D. Ruppert (1988). A note on computing robust regression estimates via iteratively reweighted least squares. *Am. Stat. 42*(2), 152–154.

Wasserman, L. (2004). *All of Statistics. A Concise Course in Statistical Inference*. New York: Springer-Verlag.

Wasserman, L. (2006). *All of Nonparametric Statistics*. New York: Springer-Verlag.

Wood, S. N. (2006). *Generalized Additive Models: An Introduction with R*. Boca Raton: Chapman & Hall.