Network-Induced Supervised Learning: Network-Induced Classification (NI-C) and Network-Induced Regression (NI-R)

Marco S. Reis
CIEPQPF – Dept. of Chemical Engineering, University of Coimbra, Polo 11-Rua Silvio Lima, Coimbra 3030-790, Portugal

DOI 10.1002/aic.13946
Published online December 26, 2012 in Wiley Online Library (wileyonlinelibrary.com).

Current supervised approaches, such as classification and regression methodologies, are strongly focused on optimizing estimation accuracy metrics, leaving the interpretation of the results produced as a secondary concern. However, in the analysis of complex systems, one of the main interests is precisely the induction of relevant associations, to understand or clarify the way the system operates. Two related frameworks for addressing supervised learning problems (classification and regression) are presented, that incorporate interpretational-oriented analysis features right from the onset of the analysis. These features constrain the predictive space, in order to introduce interpretable elements in the final model. Interestingly, such constraints do not usually compromise the methods' performance, when compared to their unconstrained versions. The frameworks, called network-induced classification (NI-C), and network-induced regression (NI-R), share a common methodological backbone, and are described in detail, as well as applied to real-world case studies. © 2012 American Institute of Chemical Engineers *AIChE J,* 59: 1570–1587, 2013

Keywords: partial correlation, clustering, classification, regression, knowledge extraction, generalized topological overlap measure, linear discriminant analysis, partial least squares, ordinary least squares

Introduction

With the increasing ability to collect data from complex systems regarding phenomena going on at different scales of space and time, new opportunities emerge for studying their behavior by adopting a data-driven perspective. In this context, inductive learning from ''well'' collected data, either resulting from carefully planned experimental designs or from observational studies, present itself as a valid alternative to the bottom-up, first-principles-based modeling paradigm, enabling the accumulation of information and the extraction of knowledge from the processes under analysis. Several classes of data-driven methodologies are quite popular nowadays for exploring the potential of information contained in data, such as:

- *Exploratory data analysis methods* (*EDA*), comprising a large toolbox of graphical, tabular and numerical methods, including unsupervised methodologies, such as principal component analysis (PCA[1-3]) and clustering algorithms, which analyze the natural correlated and clustered structured of data, both in the variable and observation modes;
- *Regression* (*or prediction*) *methods,* where the goal is to develop a prediction function for the systems quantitative responses of interest, such as linear regression (e.g., ordinary least squares, OLS[4,5]; principal components regression, PCR;[1,3] partial least squares, PLS[1,3,6–10]; ridge regression, RR[11,12]), nonlinear regression[13] and nonparametric regression (e.g., nearest-neighbor regression, NNR, and Kernel methods;[11] classification and regression trees, CART[14]);
- *Classification methods*, similarly to regression methods, try to estimate a given property of the system (an output variable), but which is now in the form of a finite set of mutually exclusive qualitative levels (called ''labels'' or ''classes''), spanning all the possible states in an exhaustive way. Classification methods can be grouped according to different criteria, such as parametric vs. nonparametric, according to the restrictive assumptions made about the distributions underlying data generation; probabilistic vs. deterministic or algorithmic, if the methods make use or not, of some notions of probability calculus in their derivation and in the analysis and interpretation of results; linear vs. nonlinear methods, if the way variables are combined in order to build the classification components (discriminants) or the boundaries separating class regions (two results that are closely linked) are linear or nonlinear; or even according to absence vs. presence of a capability for providing a no-classification output, as some methods are focused on finding the maximal separating subspaces, i.e., on discriminating classes looking for class dissimilarities (e.g., the linear and quadratic discriminant analysis classifiers, LDA, QDA,[15,16] and partial least squares for discriminant analysis, PLS-DA[17]), while others on modeling individually each class (e.g., soft independent modeling by class analogy, SIMCA,[18] UNEQ,[19] and the general Bayes classifier), focusing on the similarity between objects relative to a given class, and only these have in principle the functionality for providing a no-classification output, in case a sample fails to belong to any of the modeled classes; on the other hand, the former

Correspondence concerning this article should be addressed to M. S. Reis at marco@eq.uc.pt.

© 2012 American Institute of Chemical Engineers

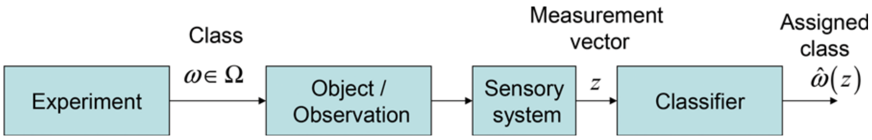


Figure 1. The essential stages in a pattern recognition problem.

(1) Experiment, from which a sample with a class label is obtained $\omega \in \Omega$ where Ω is the finite set containing all possible classes (such labels are assumingly known in the training phase, but unknown in the test or implementation phase), (2) then, such a sample is subject to measurement or observation, using a dedicated sensor or device, (3) from which a vector of features z, is produced, and (4) will support the attribution of a class label to the sample, after being processed by a previously selected and trained classifier. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

methods (i.e., those focused on finding maximal separating subspaces), essentially divide the features space in a set of mutually exclusive regions, corresponding to the available classes, usually not contemplating no-classification regions.

As both regression/prediction and classification methods, require the preliminary knowledge of some sort of output variables during the training stage, in order to estimate the models' parameters, they are known as supervised methodologies (in contrast to methods such as PCA or clustering, that do not require such information about the values of any systems' response or labeling variable, in order to supervise or guide the training stage).

Data-driven methodologies,[20-25] such as those presented previously, are essentially primarily focused on aspects connected with explaining the variability of data, such as total variation in PCA, quality of fitness in OLS, prediction ability in PLS and classification rate in LDA or LQA. Only at a second stage, the issue of model interpretation begins to be addressed, looking to the estimated values of the parameters and some other related quantities. This subordination of interpretation regarding predictability, until now taken for granted, presents a number of drawbacks and hinders certain relevant activities where the analysis of the structure of the system, more than the estimates of its outputs, is the relevant issue. For instance, in process improvement activities, or while analyzing a new complex reaction, one is primarily interested in the way and how variables/reactants interact, in order to design a better system or to conduct the next round of experimental identification trials. Another example occurs in the analysis of natural systems, such as metabolic or gene regulation networks. Here, the focus is also strongly centered on extracting the correct connectivity and causal structure of the system, rather than in predicting accurately the amounts of metabolites/proteins produced. In this article, we present two supervised frameworks, one for classification and another for regression, called network-induced classification (NI-C), and network-induced regression (NI-R), respectively, that are able to bring interpretational features to the forefront of the analysis goals. A brief reference to the main representatives of such classes of supervised methodologies will be provided in the next paragraphs, in order to better contextualize and clarify the contributions made with the proposed approaches.

Classification methods

A classifier is the final component in the whole sequence of stages that constitute a pattern recognition application, which begin with the acquisition of data from a given sample or entity under analysis and end up with the estimation of its class or label (Figure 1). This entire sequence is developed, optimized and tuned for each application, without disregarding or overlooking any stage, especially those closer to data acquisition, as they can determine, in an irreversible way, the quality of what can be done afterward, in the classification stage. Therefore, issues such as experimental design and data collection,[26] random sampling and variable selection[16,27] must be properly addressed before a given classifier is selected, a decision that is closely connected to the scatter patterns exhibited by data for the different classes, and in particular on how they are separated and distributed.

More specifically, a classifier is essentially a map from the N-dimensional measurement space $\mathbb{R}^N$, onto the set of class labels $\Omega = \{\omega_1, \omega_2, \ldots, \omega_g\}$, providing, for each new observation vector $z \in \mathbb{R}^N$, an estimate for the corresponding class $\hat{\omega}(z) : R^N \rightarrow \Omega$.

The proper development of the classifier entails several steps. Among these, one can refer (1) the selection of adequate datasets for selecting, training and testing the classifiers (representative of the range of conditions found in practice, with a balanced representation of all classes), (2) definition and selection of features derived from measurements (features selection)[27] or how they should be combined, in a suitable way (e.g., using PCA, PLS or LDA), in order to better discriminate observations from different classes (feature extraction), (3) development of the classifier (still using the training dataset), which could be either a unique methodology or an ensemble of methodologies, combined, for instance, through a (possibly weighted) majority voting procedure, or in a sequential way, and (4) assessment of the classifier predictive performance, using a test dataset, if such is available (if not, approaches such as cross-validation can still provide a good estimate of its generalization power). Some common examples of classifiers used in practice include: the linear and quadratic discriminant analysis classifiers (LDA, LQA),[15,16] logistic regression (LR),[11] partial least squares for discriminant analysis (PLS—DA),[17] artificial neural networks (ANN),[16] support vector machines (SVM),[28] k-nearest neighbor classification (k-NN),[11] Parzen classification,[11] and soft independent modeling of class analogy (SIMCA),[18] among others.

Regression methods

Several empirical modeling frameworks have been proposed, but the ones based on a linear regression formulation stand out, given their widespread use. This happens to be so, because of the large and widespread body of knowledge regarding the analysis of this class of models, as well as the many computational platforms currently available to implement it. The general *model structure* for a linear regression model can be simply written as

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_m x_m + \varepsilon \tag{1}$$

where, Y represents the output variable, and $\{x_j\}_{j=1:m}$ the input variables. The coefficients are known as partial regression coefficients $\{\beta_j\}_{j=1:m}$, and β_0 is the intercept. The term ε is the model error term or residue, which is a random variable that introduces a stochastic component into the linear regression model, for describing the unstructured variability of the processes, i.e., that part of the Y variability not captured by the input variables. Usually, this term is considered to possess the following properties: zero mean, constant variance and follows a serially independent normal probability distribution, in which case the model parameters $\{\beta_j\}_{j=0:m}$, are estimated optimally through the least-squares method. However, sometimes the constant variance assumption needs to be relaxed in order to meet the data features, and the parameter estimation procedure must be modified accordingly. For instance, when the variance of the output error term varies, but it is known, weighted least squares should be applied instead of ordinary least squares. Some models also include errors in the inputs, besides the output error. Examples of these approaches include multiple least squares[29,30] (MLS, input and output error variances are assumed to be known), orthogonal distance regression (ODR, error variances of inputs and outputs are equal), and constant variance ratio (CVR, error variances of inputs and outputs are different, but their ratio is constant). The last two methods (ODR and CVR) do not assume error variances to be known, falling within the scope of the so called, error-in-variables methods (EIV).[13,31]

On the other hand, methodologies have been proposed that consider the existence of an inner latent variable model structure generating the linear relationship presented in Eq. 1. In others words, such a linear relationship does not represent the generating mechanism underlying observed data, but it is just the external manifestation of an inner model structure of the type

$$\begin{aligned}\mathbf{X} &= \mathbf{TP} + \mathbf{E}\\ \mathbf{Y} &= \mathbf{TQ} + \mathbf{F}\end{aligned} \tag{2}$$

where $\mathbf{X}$ and $\mathbf{Y}$ are the $n \times m$ and $n \times r$ matrices of inputs and outputs, respectively T is the matrix of latent variables (each row corresponds to a different vector of observations of the p latent variables, one variable in each column), $\mathbf{P}$ and $\mathbf{Q}$ are matrices of coefficients (called loadings), and $\mathbf{E}$ and $\mathbf{F}$ are the $n \times m$ and $n \times r$ matrices of residuals, respectively. The parameters of this model can be estimated using different latent variable frameworks, such as principal components regression (PCR)[1,3] or partial least squares (also known as projection to latent structures, PLS),[1,3,6–10] that basically follow different procedures for estimating the internal parameters, in terms of which, the coefficients of the model can be established. The parameter estimation procedure based on latent variables models can also incorporate explicit information about the error structures, if such is available, as reported elsewhere.[32–35]

Other classes of criteria used for estimating linear regression models include the case of restricted estimators (e.g., ridge regression and LASSO) and the class of robust estimators (e.g., the M-, GM- and Siegles repeated median estimators, among others).[36] Nonlinear regression techniques can also be used, when parameters are not linearly related to the output, even though the search of the solution in such cases is subject to the usual problems found in nonlinear optimization problems (namely, the existence of local optima).[13]

Network induced supervised learning frameworks

Looking back at all the classifiers and regression methodologies referred earlier, which represent different categories of approaches available for addressing the classification and regression problems, respectively, one can verify that they either operate by combining all variables together, properly weighting all the input variables according to the criteria of the underlying algorithm, or consider each isolated variable in turn, when developing the final models, in a stage-wise fashion. Despite the quality of results one can already achieve with the current methodologies, there is a clear mismatch between such internal mechanisms and those found in the real world. In fact, in the aforementioned algorithms, only the two extreme situations, occur, namely: either all elements that are considered in the analysis (as some may have been disregarded during the feature selection stage), are simultaneously involved (with different intensities, but still active; e.g., PLS, PCR, PLS-DA, LDA, SIMCA), or they act in an independent way (e.g., CART, k-NN). There is no room to accommodate intermediate situations. However, in most real world systems, the mechanisms are such that there are *groups* of features, with different dimensions and compositions, which are actively involved in the variety of systems' functions, even though sometimes cooperate and act together in certain phenomena. Therefore, the true nature of systems is based on *subsets* of variables cooperatively interacting together, and not on isolated variables acting independently or on the entire set composed by all variables, acting simultaneously and in a coherent way.

Such a mismatch between the internal structure of current supervised learning methods (namely, classifiers and regression methods) and that of real world systems, hinders, from our perspective, a more in-depth extraction of useful information from data when addressing problems in real application scenarios. This is especially relevant in certain key branches of modern science, such as, for instance, in the biosystems and health fields, where increasingly large amounts of data are being gathered from different entities (organs, tissues, cells, etc.) with distinct characteristics (phenotypes, diseases), and the goal is not only to develop a predictive approach to use in future samples, but, on an equal level of importance, to identify which elements under analysis (variables) are involved in such manifestations (for developing markers or infer the molecular origins of the disease).

It is now widely accepted that most complex systems, such as those with which we interact and study on a daily basis, either from natural sources (living systems, such as bacteria, cells, organisms) or artificial origins (industrial plants, computer networks), share some common structural features. In particular, they show organization patterns of *modularity*, *hierarchy* and *specialization*, among others.[37–40] What this evidence tell us, is that, more often than not, the active elements in the expression of a given phenotype, or in the development of a given characteristic, which originate a given class label or response level, are in fact organized in clusters of different sizes, composed by elements that are highly interactive (cooperative) when they are operating. Furthermore, in some cases (for some class labels or response levels) they may be silent (not operating at all), which, in an extreme case means that they may be inactive for all conditions under study (i.e., they are not involved in the conditions studied, even though were contemplated during data collection), whereas for other cases, they may act, along with other clusters, in a synergistic way,

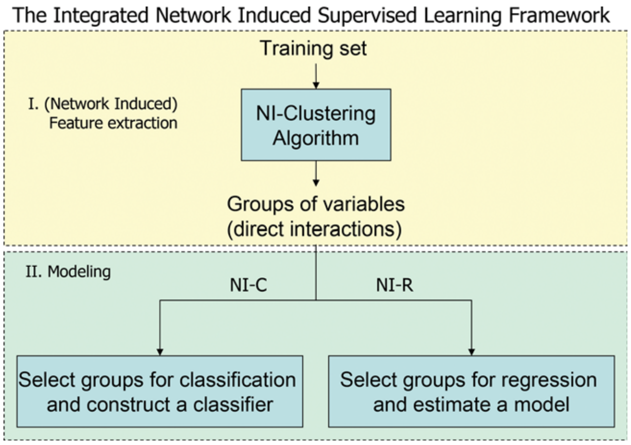

The Integrated Network Induced Supervised Learning Framework
Training set
I. (Network Induced) Feature extraction
NI-Clustering Algorithm
Groups of variables (direct interactions)
II. Modeling
NI-C
NI-R
Select groups for classification and construct a classifier
Select groups for regression and estimate a model

Figure 2. The integrated supervised learning framework, with its two stages: the first stage is common to the classification (NI-C) and regression (NI-R) frameworks, while the second stage is specific of the particular type of problem to address.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

to build the observed manifestations. In this work, classification and regression frameworks are proposed to address the following analysis questions, being also flexible enough to be properly tuned for different application situations: (1) how many clusters are active and what is their composition; (2) in which class manifestations are the clusters/variables more involved in (classification) or to which response are they more strongly associated with (regression), and (3) and how are they involved (e.g., what is the sign/trend of their interaction).

Each one of the two proposed frameworks is composed by two stages. The first stage is common to both frameworks, and involves the identification of the underlying network modules, while the second stage is specific of the type of supervised problem considered, and regards the development of the appropriate selection of modules (or clusters of variables) to use, and the construction of the predictive model for classification or regression. Thus, we can think of these approaches as two realizations of an integrated framework for supervised learning, according to the type of problem to address: classification or regression. Figure 2, schematically summarizes this perspective.

This article is organized as follows. In the next section, a brief overview of the methods used in this work is given, in order to set the necessary theoretical background and clarify the notation used. In the following section, the proposed frameworks, NI-C and NI-R, are described in detail, namely their common backbone (first stage), and distinctive features (second stage). Then, the results of the proposed frameworks are presented, regarding their application to four real world datasets, and compared with those from benchmark methods, that represent the NI-C and NI-R counterparts, if no interpretational-oriented constrains were considered (the benchmarks are the same methods used in our frameworks, but without the preliminary stages of clustering formation and selection). With such comparison, one can assess what is the associated loss/gain in performance, strictly arising from the proposed procedures, other methodological components remaining constant, such as the classifier or regression method adopted. The proposed frameworks are then discussed, and finally their essential features, as well as the results obtained, summarized, in the conclusions section.

Methods

In this section, a brief introduction is provided to the methods underlying the proposed supervised learning frameworks. We start with a revision of the concept of partial correlation, whose computation is instrumental in the first stage of the implementation of the network-induced frameworks. In particular, partial correlations are employed for obtaining a similarity measure between features or variables, from which clusters can be formed. The complete measure of similarity adopted here, will take into account the number of neighbors a given pair of nodes share in common (''features'' or ''variables'' will be here also referred as ''nodes'', which can be connected or not, according to the magnitude of the partial correlation coefficients computed for each pair; the partial correlation magnitude constitutes a measure of the strength of the association for each pair of variables, after controlling for the effects of other variables). Such a similarity measure, called generalized topological overlap measure (GTOM), and the associated clustering algorithm (network-induced clustering), will be presented in detail in the next section. Finally, a brief reference to the well-known methodologies of linear discriminant analysis, ordinary least squares (OLS) and partial least squares (PLS), will also be made, not only because they integrate the proposed methodologies, but are also used as benchmarks, for comparing the performances of the methods proposed, as they provide the results for the unconstrained counterparts of the NI-supervised learning methodologies.

Partial correlation

Partial correlation is a statistical concept developed for evaluating the degree of association between two entities, after the effects associated with others are removed from the analysis (or ''controlled for''), i.e., after the part of the mutual association that can be explained by prespecified third-parties, is discounted or removed from the analysis. Therefore, while the conventional correlation analysis measures the *marginal* linear association between a pair of variables, with the partial correlation, one is able to infer the magnitude of the association that is *unique* to such a pair of variables, when a set of other variables are controlled for, i.e., kept constant, meaning that they are not interfering, in any way, in the establishment of the observed association between the pair of variables under analysis. Let us take for instance the graph represented in Figure 3, where a simulated stochastic process (a ''fork'', in this case) is being driven by node Z, which is naturally associated with nodes A and B, with marginal correlations $r_{AZ} = 0.8$ and $r_{BZ} = 0.6$. This causal relationship leads to the existence of an induced marginal correlation between nodes A and B, of magnitude $r_{AB} = 0.48$, even though they do not actually interact in any direct, causal way. Therefore, it is clear from the analysis of this example, that all variables are significantly associated regarding the marginal correlation or Pearson correlation measure.

However, if one computes all the partial correlations for this case, after controlling for the variable not included in the pair under analysis, i.e., $r_{XY \cdot W}$, where X and Y are the pair of variables whose association we are interested in

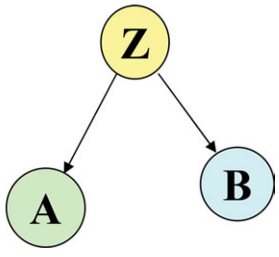


Figure 3. A graph (known as a “fork”) containing a node Z, that is inducing, simultaneously, variation in nodes A and B, therefore, creating, in an indirect way, a marginal association between these two nodes.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

assessing, and W is the variable under control, the following results are found, where we can clearly discern the direct, causal associations from the others $r_{AZ\cdot B} = 0.7295$, $r_{BZ\cdot A} = 0.4104$ and $r_{AB\cdot Z} = 0$. The fact that $r_{AB\cdot Z} = 0.$, means that A and B are not correlated when the effect of Z is previously removed from both variables, which correctly lead us to the conclusion that the marginal correlation of 0.48 between nodes A and B originates from this variable. To sum up, the use of the partial correlation instead of the widely used Pearson correlation, allows for increasing the resolution with which we discern direct and indirect associations between variables, being also bounded to take values between -1 and 1.

$\boldsymbol{r_{AZ\cdot B}}$, $\boldsymbol{r_{BZ\cdot A}}$ and $\boldsymbol{r_{AB\cdot Z}}$, are examples of the computation of first-order partial correlations (because only one variable is being controlled for). If we consider the marginal correlation as a zeroth-order correlation, then the following formulas summarize the equations for computing the partial correlations of orders up to 2:[41]

$$\textit{zeroth-order correlation}: r_{AB} = \frac{\mathrm{cov}(A,B)}{\sqrt{\mathrm{var}(A)\mathrm{var}(B)}} \tag{3}$$

$$\textit{first order-partial correlation}: \ r_{AB\cdot Z} = \frac{r_{AB} - r_{AZ}r_{BZ}}{\sqrt{\left(1-r_{AZ}^2\right)\left(1-r_{BZ}^2\right)}} \tag{4}$$

$$\textit{second-order partial correlation}: \quad r_{AB\cdot ZW} = \frac{r_{AB\cdot Z} - r_{AW\cdot Z}r_{BW\cdot Z}}{\sqrt{\left(1-r_{AW\cdot Z}^2\right)\left(1-r_{BW\cdot Z}^2\right)}} \tag{5}$$

Another way to compute the partial correlation between two variables A and B, when controlling for a finite set of other variables, say **S**, consists of implementing the following sequence of steps:

1. Compute a regression model, where variable A is the response variable and variables in the set **S** are the regressors (or input variables), and save the regression residuals, thus obtained, ε_A;

2. Do the same for variable B, i.e., estimate another regression model, where variable B is now the response variable and variables in the set **S**, the corresponding regressors, and again save the regression residuals ε_B;

3. The partial correlation coefficient between A and B, controlling for **S**, is just the Pearson correlation coefficient between the residuals computed in steps 1 and 2, i.e., ε_A and ε_B, respectively, $r_{AB\cdot\mathbf{S}} = r_{\varepsilon_A,\varepsilon_B}$.

With this procedure, as long as the regression models can be estimated from data, the corresponding higher-order partial correlation coefficients can be determined. It can therefore be easily employed for computing any higher-order partial correlation coefficients. In particular, it can be used to compute the partial correlation between pairs of variables, controlling for all the remaining ones, which will be referred here as full-order partial correlation coefficients (foPC).

Linear discriminant analysis (LDA)

Linear discriminant analysis is a multivariate analysis technique, proposed by R. A. Fisher in the late 1930s, for finding the directions in the variables’ space, that most discriminate the samples according to the classes they belong to (hence, the name from which it is also known as Fisher discriminant analysis, FDA). Taken together, these directions will define a subspace, where the projected samples are maximally separated among all the subspaces with the same dimensionality. They are computed by finding the linear combinations that maximize the ratio of the *between groups* variability to the *within groups* variability, under the constraint that each one of the extracted linear discriminants LD_i, satisfy $LD_i^T\mathbf{S}_{pooled}LD_i = 1$, for $i \leq \mathbf{LDmax}$, where $\mathbf{S}_{pooled}$ is the pooled estimator of the common covariance matrix. The discriminant directions are optimal under the assumptions that the conditional probability density functions for the classes follow multivariate normal distributions with *equal covariances*. **LDmax** is the maximum number of linear discriminants that are possible to extract, a number that is upper-bounded, according to the number of variables m, and the number of classes to separate g, more precisely, **LDmax** = min{m, g -1}. The linear discriminants also have the property that, $LD_i^T\mathbf{S}_{pooled}LD_j = 0$, for $i \neq j$.[15]

The linear discriminants provide an adequate summary of the discriminating power contained in the set of variables under analysis, and, therefore, can potentially simplify the task for classifiers, as the observations from different classes appear maximally separated in such subspaces. One example of a classifier that is widely used in practice, being derived from similar assumptions to those considered in FDA, is the *linear classifier*. The linear classifier is a Bayes classifier, which has optimal properties regarding the overall classification risk, a quantity where misclassification costs are also taken into account, through an appropriate cost function (optimality, taken under the assumption that the hypothesis made regarding the class-conditional probability density functions and prior distributions, are valid). The linear classifier is a Bayes classifier with a uniform cost function (i.e., one that gives “0” cost if the estimate is correct, and cost “1”, if it is not), and where the distributions for the classes are assumed to follow a multivariate normal probability density function with the same covariance matrix. Its implementation consists on (1) first estimating the posterior probabilities for all classes $\omega_k \in \Omega$, given the measurement vector z, i.e., $P(\omega_k|z)$:

$$P(\omega_k|z) = \frac{P(z|\omega_k)P(\omega_k)}{P(z)} \tag{6}$$

where $P(z|\omega_k)$ is the class-conditional probability density function for class ω_k (multivariate normal probability density functions with equal covariance matrices), $P\ (\omega_k)$ is the prior probability for class ω_k; and then (2) attributing to a given sample, the class label corresponding to the maximal posterior probability, after it is computed for each and every class:

$$\hat{\omega}(z) = \underset{\omega \in \Omega}{\arg\max}\{P(\omega_k|z)\} \tag{7}$$

After introducing the aforementioned distributions into Eq. 6, and applying logarithms (a monotone increasing transformation that does not change the optimal solution), this procedure simply consists of computing g linear functions of z, and picking the class for the function holding the highest score. The fact that these classification functions are linear, and, therefore, the boundaries separating the regions in the variables' domain corresponding to each class are also linear, justify the name by which this classifier is known, the ''linear classifier''.

Ordinary least squares (OLS)

The formulation of OLS was already presented in the introductory section, whose model structure is given by Eq. 1. If the m inputs are all gathered in a $n \times (m+1)$ matrix $\mathbf{X}$, having in the first column only ones, namely

$$\mathbf{X} = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1m} \\ \vdots & \vdots & \dots & \vdots \\ 1 & x_{n1} & \cdots & x_{nm} \end{bmatrix} \tag{8}$$

and $\mathbf{y}$ represents the $n \times 1$ vector with the responses, while the model coefficients vector $\boldsymbol{\beta}$, is given by $\beta = [\beta_0\, \beta_1 \, \cdots \, \beta_m]^T$, then the OLS estimate for the model coefficients, $\hat{\beta}$, is obtained from the following expression

$$\hat{\beta} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y} \tag{9}$$

More details about OLS can be easily found in the classical linear regression literature.[4,5]

Partial least squares (PLS)

When the input variables are strongly correlated, the OLS method present problems, namely in computing the term $(\mathbf{X}^T\mathbf{X})^{-1}$ of Eq. 9 (matrix $\mathbf{X}^T\mathbf{X}$ becomes ill-conditioned), and the variances of the estimated coefficients, given by the diagonal elements of the variance-covariance matrix var $(\beta) = \sigma^2\ (\mathbf{X}^T\mathbf{X})^{-1}$, also increase sharply in this situation, leading to unreliable solutions for the parameter estimation problem. This is the multicollinearity problem of OLS, for which several approaches can be used to address it, namely (1) variable selection methodologies (e.g., forward addition, backward removal, forward stepwise and best subsets); (2) shrinkage or restricted estimation methodologies (e.g., ridge regression and LASSO); and (3) dimension selection methodologies (e.g., principal components regression, PCR, and partial least squares or projection to latent structures, PLS).

In this context, PLS offers an adequate solution to this problem, by interpreting the presence of collinearity as an external manifestation of an inner latent variable structure, of the type shown in (2). In this way, instead of degrading the solution, collinearity will in fact stabilize the PLS estimates. PLS essentially operates by finding those linear combinations of the input variables that show the largest *covariance* with the output (in case there is only one output, which is not necessarily the case, as PLS can also be applied to multiple outputs situations), while respecting some orthogonality conditions regarding the linear combinations found in the previous stages.[3,6,10,42] As the goal of PLS is to maximize the covariance between the linear combinations of the inputs and the output, it depends on the scale in which the variables are expressed. Therefore, variables need to be properly preprocessed. In this work, all variables were previously centered at the mean and scaled to unit variance (a preprocessing method usually known as *autoscaling*). Each linear combination (also called a ''variate''), say $\mathbf{t}_i = \mathbf{X}\mathbf{w}_i$, corresponds to a latent variable in the PLS model (where $\mathbf{w}_i$ is the ith vector of weights for the linear combinations, and $\mathbf{t}_i$ the associated score vector with the values for the ith latent variable). The number of latent variables (or variates) in use, defines the complexity of the PLS model. This number is usually found by some cross-validation method (e.g., Leave-One-Out, K-fold, or Monte-Carlo, among others). At most, there would be as many latent variables (or PLS dimensions) as the number of inputs, in which case the PLS and OLS solutions are exactly equal. In the end, the PLS model can be recast into a formula such as (1), and implemented straightforwardly in this way.

Network-Induced Supervised Learning Frameworks

In this section, we present and describe the procedures underlying the building blocks of the integrated framework for supervised learning (Figure 2). In particular, the common algorithmic backbone shared by network-induced classification (NI-C) and network-induced regression (NI-R) is addressed, i.e., the network-induced clustering algorithm (Stage I), as well as the distinct features of these two methodologies (Stage II). Each block is described separately, given their marked differences and modular nature.

Network-induced clustering (stage I)

In this subsection, the network-induced clustering, based on the generalized topological overlap measure, is presented. The starting point, is the computation of the pairwise partial correlations, as described previously, which enable us to build a picture of which variables are more directly associated. A map of such associations is obtained by thresholding the coefficients using a criterion for statistical significance, developed in the scope of the hypothesis testing statistical framework. The final result takes the form of an adjacency matrix **Adj**, i.e., a matrix composed only with ones and zeroes, where a ''one'' in the position of **Adj**, means that variable or node ''i'' is linked to (or associated with) variable or node ''j''.

More specifically, the partial correlations for every pair of variables are first computed, while controlling for the other variables. From the options available, we opt primarily to control for the effects of all the remaining variables (full-order partial correlation *foPC*), besides the pair whose direct association is to be inferred from the partial correlation coefficients (the procedure based on regression residuals was adopted), but we have also considered the first- and second-order partial correlations, computed using formulas (4) and (5), in the analysis of the case studies (1*oPC* and 2*oPC*, respectively), in order to get more insight regarding the consequences of using a given order for the partial correlations, something that is not currently firmly established. The ($m \times m$) matrix of partial correlations coefficients **PC**, that contains in the position (i, j) the partial correlation between variable i and j, is then subject to a binarization operation, that sets to 0 all the coefficients whose magnitude falls under the baseline of statistical significance, for a predefined significance level α, while the value of 1 is set for all the other, significant coefficients. Such a binarization procedure[43] consists on first transforming the partial correlation coefficients of order ***ord***, $r_{ij,ord}$, into $Z_{ij,ord}$ through the following expression,

$$Z_{ij,ord} = 0.5 \times \ln\left(\frac{1 + r_{ij,ord}}{1 - r_{ij,ord}}\right) \tag{10}$$

and then into Z-scores, that are approximately normally distributed, using the formula

$$Z = \frac{Z_{ij,ord}}{\sqrt{1/(n-3-ord)}} \tag{11}$$

With such a Z-statistic, the statistical test for significance can be carried out, after which all those coefficients whose Z-score is such that $|Z| < Z_{\alpha/2}$ are set to zero ($Z_{\alpha/2}$ is the upper $\alpha/2 \times 100\%$ percentage point for the normal distribution), while the others, i.e., those that are statistically significant (at a significance level of α), are set to 1. The resulting matrix containing only zeros and ones is an adjacency matrix that reflects the direct associations among pairs of variables.

The adjacency matrix, obtained in this way (**Adj**), is the mathematical codification of an undirected, unweighted graph, and is the basic input to the computation of the topological similarity, according to a criterion that, instead of just looking to what happens between each pair of variables (as is usually done), also incorporates an assessment of the number of common nodes, or variables, that are shared by their neighborhoods. With such an analysis, one tries to increase the robustness of the subsequent clustering procedure against false-positives (related to the identification of spurious significant partial-correlations in the adjacency matrix). In fact, there is some empirical evidence that two highly interacting variables usually extend such interaction among several routes linking them. In this context, Ravasz et al.[37] proposed the use of a topological overlap (similarity) measure (TOM), defined by

$$TOM(i,j) = \frac{l(i,j) + Adj(i,j)}{\min\{k_i, k_j\} + 1 - Adj(i,j)} \tag{12}$$

where $l(i,j) = \sum_{u \neq i,j} Adj(i,u) \times Adj(u,j)$, i.e., is the number of neighbors shared by nodes i and j and $k_i = \sum_{u \neq i} Adj(i,u)$, the number of links in node i. In this sense, two variables are said to a have a high-topological overlap, if they are also connected to approximately the same set of variables in the graph defined by their adjacency matrix. In biosystems, variables/nodes showing a high-topological overlap tend to be involved in the same function, and, therefore, constitute intrinsic system modules. By extending this rational to other systems, we expect to gather those variables belonging to the same functional groups, which act together in the development of the characteristics one aims to correctly classify or predict. This measure was later on extended to higher-order neighborhoods,[44] in order to increase the sensitivity of the similarity measure, giving rise to the generalized topological overlap measure (GTOM). The order of the neighborhood is defined by the parameter l, which determines the l-step neighbors for each input node, being GTOM a measure of the agreement between two such sets (for $l = 1$, the definition reduces to TOM, while the situation for $l = 0$, corresponds to using just the direct links in the adjacency matrix). In this article, we have used the GTOM measure of pairwise interconnectedness, with $l = 2$.

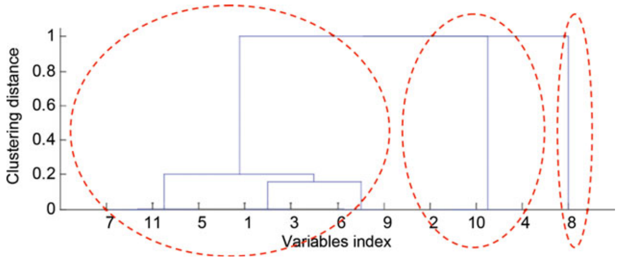


Figure 4. Example of a dendrogram for the NI-Clustering algorithm, based on the GTOM similarity's distance matrix computed from an adjacency matrix derived using partial correlation information (full order; data from the "Roughness" dataset).

The number of groups suggested after analysis of the dendrogam is NCLUST = 3. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Once the GTOM similarity matrix is obtained; it is used to run a hierarchical clustering algorithm, with linkage criteria set to the unweighted average distance. Using the fact that it has 1 as an upper bound; the topological overlap measure gives rise to the following overlap-based distance measure, based on which one can implement a conventional clustering algorithm

$$d_l(i,j) = 1 - GTOM_l(i,j) \tag{13}$$

where the index l defines the order to be used in the GTOM computations. We call this entire procedure for establishing groups of variables/features/nodes, based on a preliminary representation of their association network, network-induced clustering (NI-Clustering).

By analyzing the dendrogram resulting from the hierarchical clustering algorithm (see e.g., Figure 4), as well as the TOM plot (a matrix-like plot, where the topological overlap measure is substituted by colors reflecting the similarity degree between variables, after they are reordered according to the results of the clustering algorithm, appearing now organized by their mutual similarity), a number of clusters or natural variable groups, reflecting the variables direct associations, and, therefore, their potentially similar functional role, can be proposed (**NCLUST**). We have also computed the "silhouette" values $S(i)$, for each variable, that provides a measure of how close a variable from each cluster is from the variables grouped in other clusters. This measure ranges from +1 (for variables that are distant from neighbor clusters, indicating a well-defined clustering structure), to −1 (for variables that are not clearly in one cluster, or were assigned to the wrong cluster). The following formula is used for computing the "silhouette" scores for each variable i

$$S(i) = \frac{(\min(AVER\_BETWEEN(i,k)) - AVER\_WITHIN(i))}{\max(AVER\_WITHIN(i), \min(AVER\_BETWEEN(i,k)))} \tag{14}$$

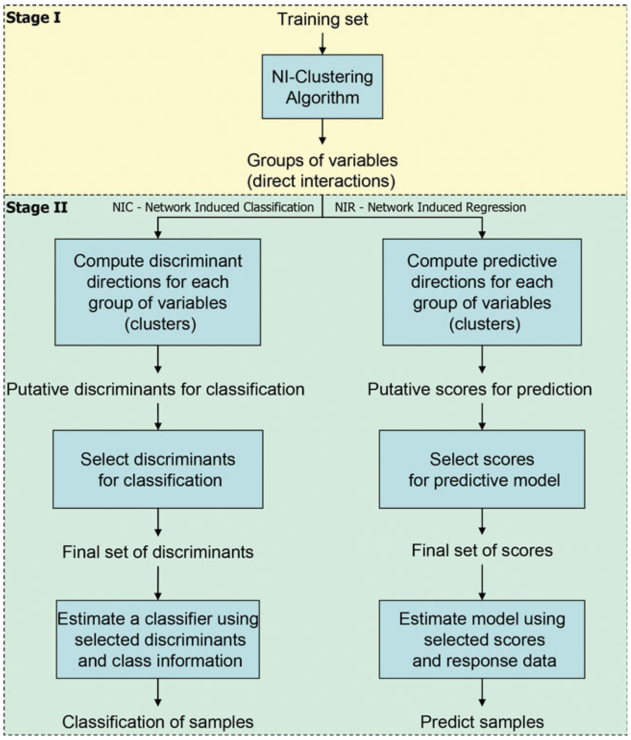

Stage I
Training set
NI-Clustering Algorithm
Groups of variables (direct interactions)
Stage II
NIC - Network Induced Classification
NIR - Network Induced Regression
Compute discriminant directions for each group of variables (clusters)
Compute predictive directions for each group of variables (clusters)
Putative discriminants for classification
Putative scores for prediction
Select discriminants for classification
Select scores for predictive model
Final set of discriminants
Final set of scores
Estimate a classifier using selected discriminants and class information
Estimate model using selected scores and response data
Classification of samples
Predict samples

Figure 5. Block diagram for the training phase of the integrated supervised learning framework, comprising NI-C and NI-R in Stage II.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

where $AVER\_WITHIN(i)$ stands for the average distance from each variable i, to the others in the same cluster, and $AVER\_BETWEEN(i,k)$ is the average distance between the ith variable and all others from cluster k. By representing graphically the silhouette values for each clustering solution, or computing their overall mean, it is possible to make certain qualitative inferences about the group structure for different numbers of clustering conditions, including different values of **NCLUST**, being another supporting tool for defining this parameter. Even though there are some methods proposed in the literature that, more or less, allow for an automatic selection of the number of clusters to use (see Ref. 45 for a list of possible methodologies), we believe that it is both wise and opportune to let the user control this parameter, because this stage is an important source of relevant information about the structure of the underlying system (which variables are associated, how many natural groups they form, etc.), and also because the automatic methods do not function well all the times. Furthermore, the clustering method may also require some fine tuning in some particular applications, something that can only be detected by direct analysis of the clustering results.

Network-induced classification, NI-C (Stage II)

After completing the segmentation of variables into functional groups, using the NI—Clustering procedure, the training stage computations proceed to Stage II (Figure 5). If the problem under analysis is a classification problem, than the network-induced classification branch is selected. In this case, the following step involves the computation of a number of linear discriminants for each cluster of variables. These linear discriminants provide a summary of the discriminatory power associated with each group of variables, being optimal for the case where observations arise from multivariate normal distributions with equal covariance matrices and cluster-dependent mean vectors, but also functioning well in other situations where the observations for different classes are linearly separable. The number of linear discriminants or variates (a variate is a linear combination of variables, e.g., a linear discriminant) that are computed for a given cluster of variables k, is given by the expression $\min\{m(k), g-1, \mathbf{NVC}\}$, where $m(k)$ is the number of variables in group k, g is the number of class categories present in data, and **NVC** is a user defined parameter that establishes an upper bound for the maximum number of variates to compute for each cluster. Usually **NVC** is kept to a small value such as 2 or 3, as the discriminatory dimensions associated with a specific functional group are not expected to be large. However, if a cluster contains a large number of variables, perhaps because they could not be properly clustered into several groups, then this number can be set to higher values, since the several complex phenomena structuring the formation of different class categories (labels), involving all these variables, may require more variates, in order to be properly described. In this work, we set these number to $\mathbf{NVC} = 2$, and only try higher values if some classification limitations are detected, that could be reasonably attributed to a low number for **NVC**. How large this number can be set, is not critical, the only consequence being that the computation time increases in the next step of the method.

In the next step of NI-C, all the discriminants arising from the different variables groups are gathered into a single group, and all their combinations with one, two, three and four discriminants, are tested, regardless of the clusters of variables they are relative, in order to assess which combination and size, presents more potential from the standpoint of classification performance. Combinations of higher order can also be tested. However, from some accumulated experience gathered so far, usually four or fewer combinations are sufficient for achieving good classification performances, representing a good comprise between the goal of exploring all possible combinations that might improve classification performance, and the time spent in the search (as this is a combinatorial problem, and the number of combinations to explore usually increases very significantly after this point).

With such a procedure, we select the variates or discriminants that are really relevant for classification, and, therefore, identify which variable groups are playing major roles in the development of the class labels that one wishes to discriminate. The analysis can then proceed to a finer level, by looking to the discriminant loadings for the selected linear discriminants, whose magnitude and signal allow for a better screening about which variables are more significantly involved in the phenomena under analysis and their interplay in this process. This is an important source of information regarding the variables under study, that can be easily explored within the scope of the proposed method.

Another important aspect of this procedure is that groups of variables that *are not* significantly involved in the generation of the class categories are automatically discarded, as their linear discriminants will not be selected in the combinatorial analysis driven by a classification performance criterion. In other words, the method has a variable selection capacity built-in by design that eliminates variables not correlated with the classification labels.

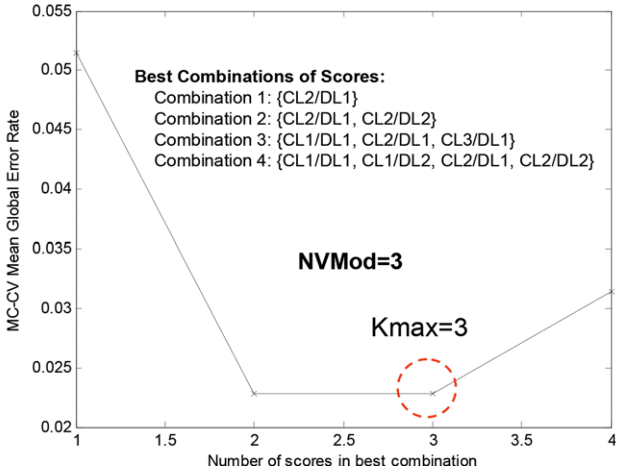


Figure 6. Plot of the performance for the best combination of a given size (one, two, etc.) vs. the size of the combinations.

In this case, it is apparent that the best global performance is achieved with combinations of size 2 or 3. Legend: CL–Cluster identifier; DL-linear discriminant identifier, regarding the cluster it is relative to. (Data from the "Roughness" dataset.) [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

The procedure used for analyzing each combinatorial trial combination of linear discriminants consists of a ''quasi-Monte Carlo cross-validation'' methodology, where, after randomly splitting observations into ''train'' and ''test'' sets, according to a user specified split fraction (here the ''test'' set always contained 20% of the total number of samples available), a classifier involving a particular combination of discriminant scores of a given size is trained and tested. The procedure tests exhaustively all linear discriminants combinations of a given size, and save the combination that achieved the best classification performance on the test sets, before moving to the analysis of another combination size. The performance of each particular combination is computed from the classification results obtained in several successive Monte Carlo cross-validation runs (in our case, 20). Performance was measured using classification accuracy measures. In particular, either the global accuracy (the overall mean of the percentage of correct class predictions obtained in each Monte Carlo cross-validation trial) and the class-mean accuracy (average of the mean accuracies for each class, given by the percentages of correct class attributions in each Monte Carlo trial, for each class label), were used.

In the end of this procedure, when all combination sizes were considered, the results are summarized by collecting the best performance achieved with combinations of size one, two, three, etc., and plotted in a graph, which provides an adequate support to the selection of the maximum number of variates to use in the final model **NVmod**. Figure 6 provides an example of such a plot, for the same dataset used in Figure 4. In this plot, the classification scores (mean global error rates, obtained with Monte Carlo cross validation) are represented for different numbers of linear discriminants considered. Also indicated in this figure, are the best combinations found, when considering one variate (the 1st linear discriminant from cluster 2), two variates (the 1st and the 2nd linear discriminants from cluster 2), three variates (the 1st linear discriminants from clusters 1, 2 and 3) and four variates (the 1st and the 2nd linear discriminants from cluster 1 and 2). Analyzing Figure 6, it can be seen that no predictive advantage can be expected from considering combinations with more than 3 linear discriminants, therefore, **NVMod** was set to 3.

In the case depicted in this figure, cluster 2 seems to be playing a major role in the discrimination of class categories. The term ''quasi'' for this methodology, arises from the fact that the linear discriminants used in it, are those computed from the original, complete dataset. This allows for faster computations, comparing to the situation where the NI-Clustering algorithm is run for every Monte Carlo realization, along with the associated linear discriminants, but it is slightly optimistic in the number of discriminants suggested for achieving the best performance. For this reason, we suggest to test the method with at least one additional linear discriminant than the number suggested by the plot, if the estimated performance between them is not very different.

Finally, in the fourth and last stage of the procedure, a classifier is developed using the best combination of linear discriminants for the selected combination size. In this work**,** we have used the linear classifier, but any other classification approach can be adopted, if expected to be more adequate. Before deploying the classifier, its performance should be more thoroughly evaluated. This can be accomplished using different approaches such as using an independent test set or adopting re-sampling approaches such as cross-validation, bootstrap, etc., some of which will be referred in the next section, where the results achieved with NI-C are comparatively assessed with a reference methodology (benchmark). The reference methodology used only differs from NI-C in the fact that no preliminary variable clustering is made. With such a comparison, it is possible to analyze whether the NI-Clustering and variates selection procedures, that enable the extraction of useful information from the nature of the actions and interactions among the variables in the phenomena under analysis, are limiting, in any significant way, the performance of the classifier.

Once the methodology is properly ''trained'', i.e., its adjustable parameters set (**NCLUST**, **NVC**, **NVMod**) and the required quantities estimated (e.g., the internal parameters of the linear discriminant analysis and the classifier), it is ready for use with new, independent data. In this phase, called, the ''test phase'', the procedure simply consists of picking up the variables clusters involved in the selected linear discriminants, combining them using the linear discriminant loadings from which the associated discriminant scores are easily computed, which will be straightforwardly processed by the (trained) classifier, in order to provide estimates for the class labels.

Network-induced regression, NI-R (Stage II)

In case the problem under analysis falls in the category of a regression problem, then the network-induced regression branch in Figure 5 should be selected and followed, for continuing the Stage II computations of the training phase. In this situation, the methodology proceeds by computing the linear combinations (variates) for the groups of variables identified with the NI-Clustering algorithm (during Stage I), which present predictive potential for explaining the observed variability of the output variable y. They correspond to the first PLS X-scores $\mathbf{t}_i = \mathbf{X}\mathbf{w}_i$, after using the variables for each cluster of inputs, along with the output variable y ($\mathbf{w}_i$ represents the ith PLS weighing vector). The number of variates computed for each cluster is given by $\min\{m(k), \mathbf{NVC}\}$, where m (k) is the number of variables in group k. Then, all such X-scores or variates are gathered together, and a procedure is run in order to select those with the highest potential predictive power. In this work, we have used two "variates selection" procedures, depending on the number of combinations of variates to consider in the final model (**NVMod**). If **NVMod<5**, the procedure consists of exhaustively considering all possible combinations of variates with size 1, 2, …, **NVMod** variates, and then picking the one leading to the lowest root mean square error of cross-validation (RMSECV), following a procedure similar to the selection of linear discriminants in NI-C. One can notice that, after a certain number of variates the predictive performance levels-off or degrades. Analyzing such behavior, it is possible to set the final value for the parameter **NVMod**. However, for **NVMod≥5**, such procedure tends to be too time-consuming to be used in practice, in a routine fashion, and a good alternative is to adopt, under these circumstances, a "forward stepwise" variable selection protocol. This protocol consists of successively selecting variates, as long as they bring a statistically significant contribution to the amount of y-variability that is explained by the model (evaluated with a partial F-test). On the other hand, variates that have already been selected to incorporate the model in earlier stages, may also be discarded later on, if their contribution becomes redundant or not significant, after the introduction of other variates.[5,46] This procedure is fast and usually lead to good models (although not necessarily optimal), despite its criteria is not so much centered on the prediction ability as in the case of the cross-validation methodology, but rather more on the quality of fitness of the estimated model to the collected data.

Once the "best" set of variates to adopt is selected, it will be used to build the model, providing the new set of predictive regressors. The final model may consist of an OLS or PLS model involving such set of variates, depending on the choice of the user, after analyzing the nature of the variates and, in particular, the eventual presence of multicollinearity in such set. Clusters whose variates were not selected for incorporating the model, correspond to variables effectively discarded from the analysis. This variable selection functionality is naturally built into the NI-R procedure, leading to models that tend to be rather parsimonious, and easily interpretable.

After the training stage, the model estimated as described previously, can be analyzed and interpreted, or used for prediction of a new dataset (test conditions). Regarding interpretation, some points may be worthwhile analyzing, such as, for instance (1) the composition of clusters formed in the features extraction module and, in particular, the ones regarding clusters whose variates have a dominant role in the final model, and (2) the weights of the variables in each variate (especially in the dominant ones) whcih contain information about their individual importance and mutual relationships, when explaining the variability of the response. For a more global analysis of the role of each variable, one can also estimate the global effect of each one of them, in the final model, by computing the following "Variable Importance in Projection" type of metric $VIP_{NI\text{-}R}$, defined as

$$VIP_{NI-R}(k) = \sum_{j \in \Omega_k} \left\{ \beta_j^2 \times w_{k,j}^2 \right\} \qquad (15)$$

where Ω_k stands for set of variates containing variable k, and $w_{k,j}$ is the corresponding kth entry of the PLS weighting vector used to compute the jth variate. β_j stands for the regression coefficient affecting the jth variate, in the final model involving the selected variates. For proper interpretation, this metric requires variables to be previously "autoscaled" (as happened in this case), so that the coefficients only reflect their roles in the model, and not the units in which variables are expressed or the variability they present.

For applying the estimated model to future situations (test conditions), the input variables of the new dataset must first be preprocessed with the same parameters used for the training set, after which the same clusters are gathered, their variates computed with the same PLS weighting vectors and, finally, the selected variates are picked up and composed with the regression vector estimated during the training stage, leading to the prediction of the outputs.

Case Studies

The methodologies described in the previous section were tested with several datasets arising from different application scenarios, in order to illustrate their main features and highlight their potential, namely in extracting useful information from data, without compromising classification or prediction performance. In fact, we have found out that sometimes such performance is even improved, meaning that the two goals are, in fact, compatible, rather than antagonistic. In order to put into test this statement, we analyze the effect of constraining the application of benchmark methods with our network-induced approaches (NI-C and NI-R), that eliminate some variables and model the selected ones in a certain restricted way (as variates). This is accomplished by comparing the NI approaches with their unconstrained counterparts, corresponding to the use of the same benchmark methods, but with complete freedom to use all the available variables, without any additional external constraint. To make results comparable, it is indeed important that the same method (the benchmark) is adopted in both situations, in order to clarify the impact of introducing the NI constrains. For instance, in the study of NI-C, the linear classifier (described in section Methods) was adopted for this purpose, therefore, playing the role of benchmark method. As for NI-R, we have used the OLS and PLS methods as benchmarks.

All computations were conducted in the Matlab environment, mostly using code developed in-house, but some functions were also employed from other sources, namely: the Matlab's Statistics toolbox (for hierarchical clustering and linear classification); the computation of the GTOM similarity metric was based on the code developed by Joaquin Goñi and Iñigo Martincorena (University of Navarra); the methodology suggested in Ref. 47 and used in Ref. 43 (code available at http://cheed.nus.edu.sg/∼ chels/DOWNLOADS.htm) for developing an adjacency matrix considering only reduced order partial-correlations (e.g., first-order or second-order partial correlations), was also used.

Datasets

Several datasets were employed to analyze and test the proposed classification and regression procedures. Among the several cases studies analyzed, we will report here the results concerning the datasets referred bellow.

Wine Dataset. Is a $X(178 \times 13)$ dataset, i.e., it is composed of 13 descriptors and 178 samples. Variables consist of analytically measured wine constituents, and the class labels regard three different cultivators from the same region of Italy. The dataset (available at http://archive.ics.uci.edu/ml/datasets/Wine), does not usually raise significant problems in the development of a proper classifier, and, therefore, constitutes one situation where the benchmark classifier is expected to outreach the proposed NI-C methodology. The challenge here is rather to see if the performance degradation imposed with the purpose of enhancing interpretation, is significant or not.

Roughness Dataset. This is a $X(36 \times 11)$ dataset, where variables regard geometrical-oriented features that summarize different aspects of an accurate profile taken from a sheet of paper, at the roughness scale (which is a fine scale), using high-resolution mechanical stylus profilometry.[48,49] For these dataset, one has, in addition, two variables available: a qualitative variable and a quantitative one. The qualitative variable regards an evaluation made by a panel of experts about the quality of article sheets in what concerns to their sensorial perception of surface roughness. Three class labels are used in the assessment, and the panel is basically trying to reproduce, in a controlled way, the evaluation made by real final users. The purpose of the classification study is to see if such an evaluation can be predicted on the basis of objective geometrical features of the roughness profile. On the other hand, the quantitative variable is relative to the measurements made by the so called "Bendtsen tester", a device used in process monitoring routines, for inferring article roughness. Its values will be also predicted, using the profiles' geometrical features as model inputs.

Pulp and Paper Dataset. These dataset contains an extensive collection of process measurements from several sections of a paper production facility (X-variables), as well as measurements of the tensile stiffness orientation angle (TSO) in several positions (nine) of the transversal direction (usually called "cross-direction") of the paper being produced in the paper machine (Y-variables). TSO shows good correlation with fiber orientation, which is a structural parameter of central importance in most paper applications (even though it also depends on other factors besides fiber orientation, such as the paper drying conditions). Each profile of TSO measurements, corresponds to a given data acquisition time, for which the process conditions are also known. The purpose of this study is to develop a regression model that predicts the basic patterns of the TSO profiles in terms of process variables, in order to provide process engineers with a tool that allows for a better control of the fiber orientation profile and to stabilize their variability, two problems with significant relevancy in practice. More information about this industrial process and the data collected is provided in the results section.

Results for network-induced classification (NI-C)

The comparative assessment of NI-C and its benchmark method (the linear classifier), is based on the analysis of the figure of merit, *global accuracy* (overall percentage of correct class assignments), which was computed in three different ways, as described below.

Resubstitution Accuracy. In this case, the classification accuracy is computed with the same dataset used to "train" the methodology. It is a measure of the method's self-consistency, i.e., its ability for classifying the same observations that were used to develop it. Resubstitution accuracy does not provide much information about the classification performance with new, never before seen data, and may even lead to rather poor models in this sense, if one blindly seeks to optimize it by overfitting the training dataset. However, a method cannot be accepted has good, if it fails this self-consistency test, meaning that it is a necessary condition, but not sufficient, for methods selection.

Monte-Carlo Cross-Validation Accuracy. In Monte-Carlo cross-validation (MC-CV), a random train/test data split is performed a number of times, say, $k = 1,\ldots,$N_CV_TRIALS, where the training set is used to estimate the model parameters and the test set to evaluate its performance (i.e., to compute the accuracy for trial k). In each trial k, the adjustable parameters are fixed to values previously estimated (**NCLUST**, **NVC**, **NVMod**), but the algorithm estimate the exact clusters composition, linear discriminants and classifier parameters, in each trial. In the end, a classification measure is obtained by averaging the accuracy scores obtained for all trials, which characterizes the predictive accuracy of the methodology for the specific set of adjustable parameters adopted (its standard deviation is also computed, in order to provide a measure of the uncertainty, to complement the information conveyed by the mean). As a portion of the observations is left out in each trial for testing the methodology, MC-CV provides an indication about the performance of the method with new data, being a good alternative for situations where data is scarce, and no test dataset is available. The method has some limitations however, when the number of classes is high and with rather few samples per class. The reason for this limitation lies in the following: during the Monte Carlo trials, in these circumstances, one of the classes may not be properly represented in the training datasets, leading to poor classification models, and, therefore, to bad classification performances for such trials (a nonstratified sampling cross-validation approach was adopted in this work, as it tends to provide more conservative estimates of the methods accuracy). This may lead to quite unreliable estimates of the methods' performance. In these situations, the following methodology is advisable.

Leave-One-Out Cross-Validation Accuracy. This methodology is similar to the previous one, but now the splitting is not random, but deterministic. In each trial, exactly one sample is left out for testing the method, while the remaining ones are used to estimate it, following the same procedure adopted in the Monte Carlo cross-validation approach. Therefore, in the end, we have exactly as many estimates as the number of samples in the dataset, from which it is possible to compute the overall accuracy (this figure has no variability, when taken for a given dataset). Leave-one-out cross-validation (LOO-CV) usually provides a more optimistic estimate for the classification performance, when compared to MC-CV, but is more reliable for situations when fewer observations are available per class. As our case studies comprehend situations where the datasets used vary significantly in size, we will use both types of validation methodologies (MC-CV and LOO-CV), in order to characterize the methods' performances.

For the benefit of space, not all the outputs produced and analyzed in each case study will be presented in this article, although some will be shown to better illustrate the methods' procedures and analysis features. The summary of the classification results obtained for each dataset is presented in Table 1, where it also appears specified the order of the partial correlations used in the computations, as well as the method parameters selected, according to the methodologies described in the preceding section.

Table 1. Results for the NI-C Framework and the Benchmark Method (Linear Classifier): Global Accuracy (%) Computed using Resubstitution (consistency test), Monte-Carlo Cross-Validation (MC-CV, the Standard Deviation appears inside Parenthesis) and Leave-One-Out Cross-Validation (LOO-CV)

		Method's adjustable parameters			Global accuracy measures (%)1st line: Proposed method (NI-C) 2nd line: Benchmark		
Case study	Partial correlation order	NCLUST	NVC	NVMod	Re-subs.	MC-CV	LOO-CV
Wine	foPC	2	2	3	98.3	98.2 (1.9)	97.2
					100	98.3 (1.5)	98.9
	2oPC	2	2	3	99.4	97.1 (2.3)	97.8
					100	98.3 (1.5)	98.9
Roughness	foPC	3	2	3	97.2	89.3 (13.8)	94.4
					100	82.9 (11.0)	88.9
	2oPC	4	2	3	97.2	85.0 (10.8)	77.8
					100	82.0 (11.0)	88.9

Note: The method adjustable parameters used, and the order of the partial correlation adopted for computing the adjacency matrix, are also shown.

Results for the Wine Dataset. As mentioned before, the wine dataset does not pose, in general, significant problems to classifiers for achieving classification scores of good quality, and, therefore, our goal in using it, is to analyze the potential performance degradation of our methodology relative to the benchmark in this simpler situation, which is expected to arise due the constraints imposed by considering only a maximum of three linear discriminants (**NVMod** = 3). Looking to the results presented in Table 1, and in particular to what concerns the resubstitution accuracy, it is possible to see that the benchmark method gets the full score of 100% of correct classifications, whereas our method falls behind by 1,7% or 0.6% (depending on the partial correlation order considered). The fact that the benchmark method always overtakes the proposed methodology in this consistency test hardly can be directly extrapolated as meaning a better predictive classification performance, as it may well be due to some overfitting of the training data, in which case it may, in part, compromise its prediction accuracy. Therefore, one should always interpret the resubstitution test results with some reserve, using them in order to check mainly whether the methods under analysis did not fail on such basic testing condition, but without giving much relevance to small performance differences.

Regarding the tests that provide more information about the methods predictive ability, we can see that the MC-CV accuracy is slightly lower for the proposed methodology (however perfectly within the error bands for the mean values of the benchmark method), being virtually the same for the case of second-order partial correlations. The results for the more optimistic (less conservative) LOO-CV accuracy, again provides a very slight edge of the benchmark method.

To sum-up, one can see from the analysis of the results obtained in this case study, that the performance degradation was really small, even though that NI-C potentially led to a lower number of variables when using full-order partial correlations. In fact, in this situation, when selecting the number of variates to retain in our methodology, there is one particular cluster whose discriminants are more prevalently selected for the classification model (Cluster 2, Figure 7). This means that the other cluster is not playing such a significant role in the classification task, which may indicate that its wine compounds are not very relevant for discriminating the three wine producers from the same region. This information not only carries a significant interpretational value, but also enables the selection of candidate compounds as ‘‘producer markers’’, which can act as their specific production fingerprints. (Note that during the MC- and LOO- cross-validation trials for computing the predictive accuracies, one does not control the selections made by the method regarding the clusters formed and linear discriminants selected, but only the methods parameters **NCLUST**, **NVC**, **NVMod**, which are kept constant, and were preliminarily selected using Figure 7, and, therefore, cannot guarantee that the aforementioned comments will be valid in every cross-validation trial, as they are for the full dataset.)

Results for the Roughness Dataset Some preliminary plots obtained in the analysis of these dataset were already presented earlier (Figures 4 and 6). As can be seen from Table 1, our method tends to present now slightly better mean prediction accuracies than the benchmark, under cross-validation conditions, meaning that the constraints raised by the topological-driven organization of variables and the limited number of variates allowed to build the model, are not compromising the performance of the classifier. It can also be seen that the performance obtained with the second-order partial correlation was, in this case, slightly lower. The

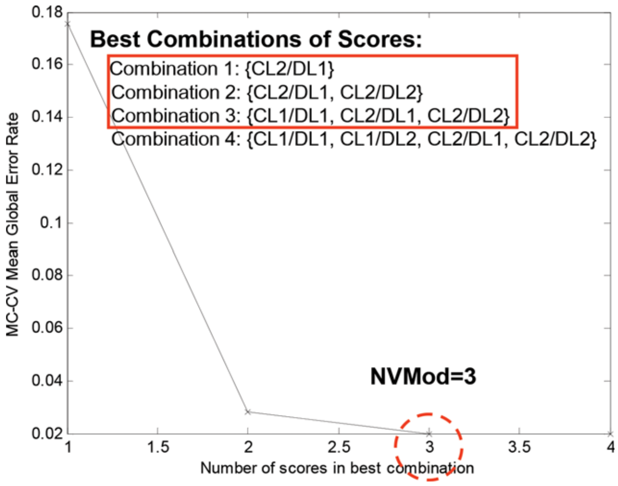


Figure 7. Plot of the performance for the best combination of a given size (one, two, etc.) vs. the size of the combinations.

In this case, the size chosen was NVMod = 3, as after this point the improvement in performance is minor. Legend: CL–Cluster identifier; DL-linear discriminant identifier, regarding the cluster it is relative to. (Data from the "Wine" dataset.) [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary. com.]

variables that are more relevant for classification are those pertaining to cluster 2 (in the case where full-order partial correlations are employed), as discriminants from this cluster are always selected, independently of the combination size considered (see Figure 6). This cluster is formed by the profilometry descriptors [50] [Ra, Rq, Rt, RS, RSm, Rdq, RKu], which were already found to be good candidates for describing the roughness behavior of the paper surface.[48]

Results for network-induced regression (NI-R)

In this section, we analyze two real world case studies, which illustrate the main features, flexibility and application potential of NI-R. Several graphical outputs and information are presented for each situation, in order to exemplify its implementation in practice, and the predictive performance of the estimated models are computed, for establishing a proper comparison with that achieved with the benchmark methods (OLS and PLS).

The predictive performance is evaluated through Monte Carlo cross-validation and leave-one-out cross-validation. In Monte Carlo cross-validation, a random set with approximately 20% of the samples in the training dataset is left aside and, with the remaining samples, a model is estimated and used to predict the values for the output variables in the samples left aside. The following quantities are then computed, in each cross-validation trial (k), the first one relative to a measure of the prediction error incurred in that trial, while the second represents an extension of the well-known coefficient of determination R^2, to the cross-validation context

$$RMSE_{CV}(k) = \sqrt{\frac{\sum_{i=1}^{n_{out}} (y_i - \hat{y}_i)^2}{n_{out}}} \tag{16}$$

$$R^2_{CV}(k) = 1 - \frac{\sum_{i=1}^{n_{out}} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n_{out}} (y_i - \bar{y}_{out})^2} \tag{17}$$

where n_{out} represents the number of observations left out in the kth cross-validation run ($k = 1,\ldots,20$) and $\bar{y}_{\text{out}}$ the respective mean of the output variable. This procedure is repeated 20 times, in order to mitigate distortions from the random allocation of observations in the two groups. In the end, the overall mean and standard deviation for these two quantities, $RMSE_{CV}$ and R^2_{CV}, are computed and reported.

As for LOO-CV, the procedure is similar, but only one observation is left aside in turn. This procedure stops when all observations were left aside once and only once, after which the following quantities are computed for evaluating the predictive ability of the method:

$$RMSE_{LOO-CV}(k) = \sqrt{\frac{\sum_{i=1}^{n} \left(y_{(i)} - \hat{y}_{(i)}\right)^2}{n}} \tag{18}$$

$$R^2_{LOO-CV}(k) = 1 - \frac{\sum_{i=1}^{n} \left(y_{(i)} - \hat{y}_{(i)}\right)^2}{\sum_{i=1}^{n} \left(y_{(i)} - \bar{y}\right)^2} \tag{19}$$

where, $y_{(i)}$ and $\hat{y}_{(i)}$ stand for the value of the output variable and its estimate, respectively, regarding the trial where the ith observation is left out. Notice that, with LOO-CV, these two quantities are fixed, for a given training dataset (no mean and standard deviation are then reported, as happens in the Monte Carlo cross-validation approach).

For comparison purposes, the predictive performances obtained with PLS and OLS (when appropriate) will also be presented, as these methods represent the best approximations of the unconstrained counterparts of NI-R, i.e., they represent the solutions that would be obtained if no interpretational-oriented constraints were enforced in NI-R. These results will enable the evaluation of the impact of introducing the interpretational-oriented restrictions in the modeling stage, on the method's prediction ability, relatively to the unconstrained situation, where all variables are employed and their parameters estimated freely.

Also reported is the fitting ability of the methods, given by the root mean square error of calibration ($RMSE_C$), obtained with the same dataset used for estimating the model (i.e., under resubstitution conditions), and the respective coefficient of determination R^2:

$$RMSE_C = \sqrt{\frac{\sum\limits_{i=1}^{n} (y_i - \hat{y}_i)^2}{n}} \tag{20}$$

$$R^2 = 1 - \frac{\sum\limits_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum\limits_{i=1}^{n} (y_i - \bar{y})^2} = \frac{\sum\limits_{i=1}^{n} (\hat{y}_i - \bar{y}_i)^2}{\sum\limits_{i=1}^{n} (y_i - \bar{y})^2} \tag{21}$$

These quantities provide a simple measure of how well the estimated model fit the data that were used to develop it, therefore offering a straightforward analysis of its quality of fitness, but they are of very limited value for assessing the method's predictive performance.

Results for the Roughness Dataset This case study is the continuation of the one presented in the previous subsection. The same 11 input variables regarding geometrical-oriented features at the roughness scale (the finer scale of analysis), and computed from accurate profiles taken over a sheet of paper, using high-resolution mechanical stylus profilometry, correspond also to the set of X-variables adopted here. However, for each of the 36 observations available, besides the classification labels used to implement the classification methods, another type of measurement was collected, for each sample, using an equipment called, the "Bendtsen tester". The measurement principle of this instrument, consists of quantifying the amount of air that escapes, during a given period of time, between a standardized ring put over the paper sheet and the sheet itself, at constant pressure.[50] This equipment is able to provide an inferential measure of surface roughness, in a short period of time, but its interpretation is not clear, as many aspects of the paper structure can interfere. Therefore, it would be interesting to explain the variability of Bendtsen measurements in terms of fundamental geometrical features of the paper surface; in order to understand from which structural characteristics it mainly depends on. This can be achieved by defining the Bendtsen tester values as the output variable of the analysis y, and develop a regression model to explain its variability. The final Bendtsen value used for each sample is the mean of five replicated measurements.

In order to apply the NI-R framework to these dataset, 4 clusters of variables were considered now (**NCLUST** = 4) for the case where the order of the partial correlation is 2, as a result of the analysis of the variables dendrogram for

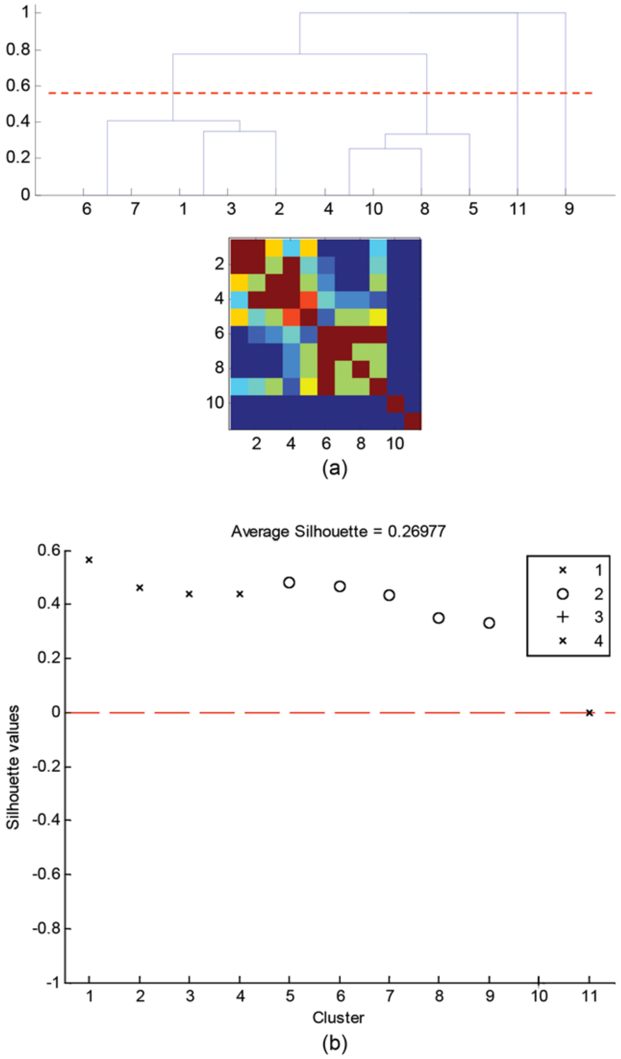


Figure 8. Auxiliary plots for selecting the number of clusters (NCLUST) in the "roughness" case study.

(a) Dendrogram and TOM plot, and (b) Silhouette plot. (They correspond to case where second-order partial correlations are used for computing the generalized topological similarity measure, GTOM). [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

the NI-clustering algorithm and the associated TOM plot, presented in Figure 8a). The silhouette plot (Figure 8b) confirms the adequacy of the clusters formed, as all values for the input variables are positive, indicating that they tend to be more distant from others belonging to the neighbor clusters, when comparing to those on the same cluster.

After defining the number of variable clusters, the number of variates to include in the final model must be set, looking to the evolution of the cross-validated errors for increasing values of this parameter, leading to **NVMod** = 4. The NI-R results obtained for such a model are presented in the first line of Table 2. As can be appreciated, the fitting ability of the NI-R model is very good (re-substitution). It also leads to the best prediction accuracy scores when evaluated with the LOO-CV method, relatively to the benchmark methods, even though its variability is higher under MC-CV, due to its more complex structure and the limited amount of data available (only 36 observations), which naturally leads to some variation in the models developed, when 20% of the data is removed from the training set in each trial of the MC-CV procedure. Therefore, introducing interpretational constrains in the model, as proposed in the NI-R methodology, does not seem to have any tangible and significant effect in prediction accuracy in this case study, and can help the interpretation of results. For instance, the $VIP_{NI\text{-}R}$ scores seem to indicate an important contribution of the parameter RSm, the mean width of profile elements, and Rdq, the RMS slope of profile, in the explanation of the variability of measurements provided by the Bendtsen tester (Figure 9). As this instrument operates according to the principle of air leakage, it indeed makes sense that the profile elements' width (an element is a sequence of a "mountain"/"valley" in the profile) has an important impact in the amount of air flow, the same applying to other profile element's features, such as their slope.

This case study illustrates the ability of the NI-R method to introduce interpretational elements in the analysis, without compromising prediction ability.

Results for the Pulp and Paper Dataset As referred before, the outputs considered in this case study are the measurements of the TSO angle at 9 positions in the cross-direction of a paper machine. The TSO profiles analyzed are presented in Figure 10a). They correspond to process conditions, for which one also has available the corresponding X-variables (process variables). In order to implement NI-R, these profiles were first decomposed using principal component analysis (PCA), and the first two scores, T_1 and T_2, which explain approximately 82.5% of the whole variability of the profiles, were used as the output variables for analysis y_1 and y_2. The loads for these first two principal components (PC) are presented in Figure 10b), where we can see that the first PC (PC1) represents a contrast between the TSO angles in the two sides of the paper machine, while the second PC (PC2) gives more weight to the central measurements, being able to explain the presence and variability of bias in the TSO angle, in such an important section of the paper machine. Therefore, the two PCs represent different aspects of the fiber orientation profile, and the goal here is to find out which of the 37 process variables studied, are most involved in the variability of each PC, in order to develop actuation policies for better controlling the cross-directional uniformity of this parameter.

NI-R models were developed for PC1 and PC2 as output variables. During this process, where the various methods' parameters were tuned, it was verified that the order of the partial correlation used in NI-clustering algorithm leading to the best predictions, differ according to the output variable being addressed. For PC1, the best results were achieved with the full-order version, whereas for PC2, the second-order coefficients lead to the best prediction scores (Table 2). Given the number of clusters identified for each situation (>5), the "forward stepwise" variable selection algorithm was employed in order to select the variates that will integrate each one of the two models. The $VIP_{NI\text{-}R}$ plots for the final models, i.e., for each PC of the TSO profile, are presented in Figure 11, where it can be seen that there are more variables affecting the tilting of the TSO profile (PC1), than its bias at the central region of the paper machine (PC2). This result means that there is an opportunity to manipulate

Table 2. Results for NI-R Framework and the Benchmark Methods (PLS, OLS): R^2 (Coefficient of Determination) and RMSE (Root Mean Square Error) for the Training Set (Resubstitution), Monte Carlo Cross-Validation (MC─CV) and Leave-One-Out Cross-Validation (LOO─CV)

		Method';s adjustable parameters			RMSE / R^2 1st line: Proposed method (NI-R) 2nd line: Benchmark (OLS) 3rd line: Benchmark (PLS)		
Case study	Partial correlation order*	NCLUST	NVC	NVMod	Re-subs.	MC-CV	LOO-CV
Roughness	2oPC	4	2	4	*NI-R* 31.32/0.95 *OLS* 28.45/0.96 *PLS* 34.25/0.94	*NI-R* 60.81 (37.51) / 0.6240 (0.6158) *OLS* 46.56 (20.44) / 0.7775 (0.1814) *PLS* 48.22 (23.74) / 0.7569 (0.2515)	*NI-R* 49.74 / 0.8739 *OLS* 52.36 / 0.8602 *PLS* 55.83 / 0.8412
Pulp & Paper, PC1	foPC	9	2	6	*NI-R* 0.646 / 0.909 *OLS* 0.2693 / 0.9842 *PLS* 0.7723 / 0.8704	*NI-R* 0.798 (0.2909) / 0.813 (0.1876) *OLS* 2.962 (3.421) / −4.742 (13.58) *PLS* 0.9442 (0.2718) / 0.7878 (0.095)	*NI-R* 0.917 / 0.818 *OLS* 1.901 / 0.2150 *PLS* 0.9512 / 0.8034
Pulp & Paper, PC2	2oPC	14	2	3	*NI-R* 0.769 / 0.780 *OLS* 0.5788 / 0.8753 *PLS* 0.8021 / 0.7606	*NI-R* 0.813 (0.1527) / 0.721 (0.1693) *OLS* 4.8328 (5.262) / −23.25 (79.96) *PLS* 0.910 (0.1769) / 0.661 (0.1340)	*NI-R* 0.830 / 0.744 *OLS* 5.096 / −8.661 *PLS* 0.945 / 0.668

In the case of Monte Carlo cross-validation results (MC─CV), both the mean and standard deviation are presented (the standard deviation appears inside parenthesis)

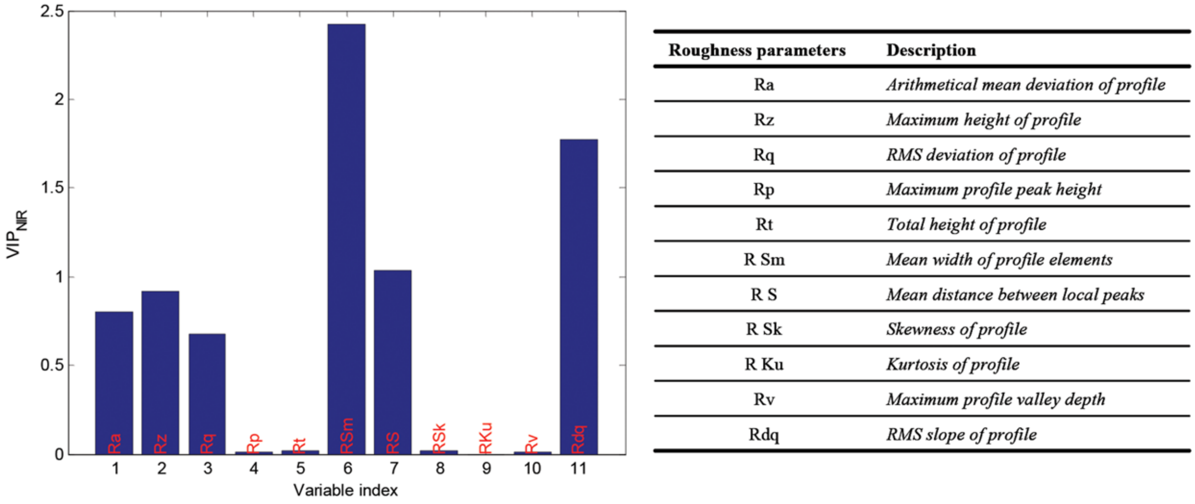
these two features independently, in order to achieve a better uniformity in fiber orientation in the cross-direction of the paper machine.

Analyzing the results presented in Table 2, one can see that, despite the good fitting ability, the OLS method fails in the prediction task (MC-CV and LOO-CV results), a limitation that is well-known in this method, when applied to situations where the input variables may present high levels of cross-correlation among themselves. On the other hand, the NI-R models developed performed quite well in prediction, never being worse than the benchmark methods (namely PLS), and often leading to better mean scores. Furthermore, the values obtained for the prediction metrics are, by themselves, quite interesting when considering the industrial environment where these dataset arises, which typically raises many problems in the development of models with some prediction accuracy.

Discussion

In this section, we address some additional features of the cluster-oriented classification framework, not referred before during the presentation of the case studies in the previous section.

The clustering methodology used, even though designed to be informative and robust by properly combining the concepts of partial correlation, topological overlap and hierarchical clustering, does not exclude other possibilities. In fact, the NI-Clustering "slot" in the overall analysis scheme (see Figures 2 and 5), may be replaced by any other clustering technique that is found to be more suitable to the problem under analysis, without changing the remaining parts of the algorithm, and with very little programming effort associated. In an extreme situation, where *a priori* knowledge about the variables or discriminating entities is such that the groups or clusters are already known beforehand, than these



Roughness parameters	Description
Ra	*Arithmetical mean deviation of profile*
Rz	*Maximum height of profile*
Rq	*RMS deviation of profile*
Rp	*Maximum profile peak height*
Rt	*Total height of profile*
R Sm	*Mean width of profile elements*
R S	*Mean distance between local peaks*
R Sk	*Skewness of profile*
R Ku	*Kurtosis of profile*
Rv	*Maximum profile valley depth*
Rdq	*RMS slope of profile*

Figure 9. ***$VIP_{NI\text{-}R}$* plot for the input variables of the roughness dataset, along with their description.**

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

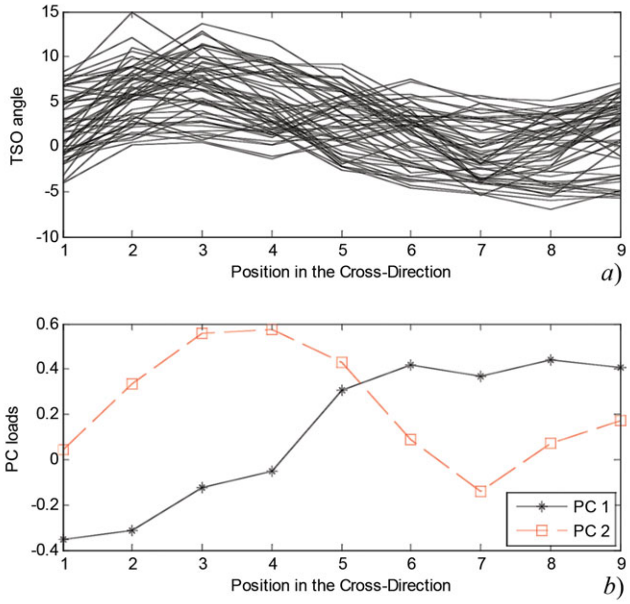


Figure 10. A set of real TSO profiles from the pulp and paper dataset (a) and the loads for the first two principal components obtained from a PCA decomposition, applied to such a set (b).

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

can immediately be used as the final result from the clustering operation. Therefore, NI-Clustering can be implemented either in ‘‘data-driven’’ mode, strictly based on collected data, or in ‘‘knowledge-driven’’ mode, where background knowledge is used to introduce interpretational constrains in the form of variable clusters. The important issue, is that clusters or groups of variables reflect some deeper association among the variables, that could be recognized and may be associated with particular roles, operations, functions, etc., so that, after implementing NI-C or NI-R, one can be able to extract and combine knowledge from the two relevant data modes, namely the variables structure (from clustering), and the observations structure (from the classification or regression tasks).

Naturally, sometimes, clustering approaches do not lead to clear and identifiable clusters and some variables might be assigned to the wrong groups. The way the proposed classification framework deals with this problem, is through the selection of linear discriminants also belonging to other clusters where relevant variables for classification are dispersed, if such improves the estimated classification performance of the method. Of course this leads to a less parsimonious solution, but brings the capability of mitigating the effects of a not so well succeeded clustering operation.

The same flexibility applies to the classifier and linear regression methods adopted. Any methodology can be used instead of the linear classifier and linear regression approaches used in this work. For example, in situations where the distribution of the observations from the several classes, in the multivariate space, present peculiar shapes, such might really be necessary, but to start with the simple, and so often effective, linear framework, is recommended.

We would also like to point out that the second stage of the proposed NI approaches, where the respective modeling tasks are conducted (Figure 2), consists also of a two-steps process: in the first step, the best models within each cluster are obtained, giving rise to a number of variates; in the second step, variates from all clusters are combined, in order to derive the final classification or regression model. In other words, the first modeling step regards developing models within each cluster, while in the second step the problem of finding the best synergistic association between the variates is considered.

Finally, as in the implementation of any predictive framework, one should always exercise some care regarding extrapolation, by checking if the measurements under analysis fall within the region used to develop them. This can be accomplished, for instance, by computing the estimated Mahalanobis distances to the center of the training dataset, and verifying if they are of the same magnitude as those used in the training stage, as well as looking at the residuals around the selected variates, and check if they are of the same magnitude as those computed in the training stage.

Conclusions

We have presented in this article, an integrated supervised learning framework with the ability to explore the natural

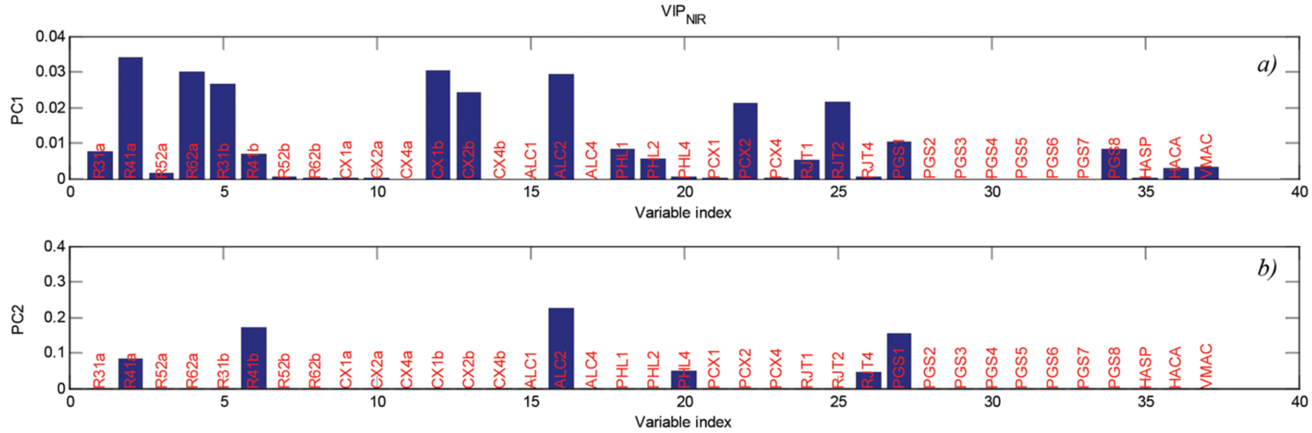


Figure 11. *VIP*$_{NI-R}$ plot for the process variables affecting the TSO profiles in the pulp and paper dataset (a) NI-R model for PC1, and (b) NI-R model for PC2.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

structure of variables, for predicting class labels, through the methodology called network-induced classification (NI-C), and for estimating the level of a continuous response, using network-induced regression (NI-R). The variables structure is described by their direct mutual affinity, inferred from partial correlation coefficients. The connective arrangement, resulting from retaining only the most relevant (strongest) links is coded in the form of an adjacency matrix, from which the topological overlap between any pair of variables is computed, as a measure of the variables affinity. This is finally employed for running a clustering algorithm, in order to identify groups of variables with strong mutual topological similarities (network-induced clustering). Then, at a second stage, these clusters are subjected to a selection operation, according to their discrimination or predictive power, depending on the type of application (classification or regression). Clusters found relevant will provide variates to be used in the final model, whereas others are tacitly removed from the analysis.

This framework elucidates which are the natural clusters of variables present, associated with some sort of systems function, operation, etc., that are furthermore driving the prediction of outputs (class labels or a continuous response). Therefore, NI-C and NI-R enable the interception of the two sources of knowledge arising from data: variables structure and observations structure. The results presented illustrate that such knowledge can indeed be extracted, without compromising the classification and prediction accuracies.

The proposed methodology finds interesting applications in the analysis of datasets from a variety of sources, such as those arising from biosystems, where one is concerned in explaining both the variable structure and natural groups of specimens, and industrial systems, where data is increasingly complex, and models that are both parsimonious and informative are required to support the analysis. Future work will address applications in other scenarios, as well as try to better clarify the role of the partial correlation order in the scope of the proposed methodology.

Acknowledgments

The author acknowledges financial support through project PTDC/EQU-ESI/108374/2008, cofinanced by the Portuguese FCT and European Union's FEDER, through "Eixo I do Programa Operacional Factores de Competitividade (POFC)" of QREN (with ref. FCOMP-01-0124-FEDER-010397).

Literature Cited

1. Jackson JE. *A User's Guide to Principal Components*. New York: Wiley; 1991.
2. Johnson RA, Wichern DW. *Applied Multivariate Statistical Analysis*. 3rd ed. Englewood Cliffs, NJ: Prentice Hall; 1992.
3. Martens H, Naes T. *Multivariate Calibration*. Chichester: Wiley; 1989.
4. Chaterjee S, Price B. *Regression Analysis by Example*. 2nd ed. New York:Wiley; 1998.
5. Draper NR, Smith H. *Applied Regression Analysis*. 3rd ed. NY: Wiley; 1998.
6. Geladi P, Kowalski BR. Partial least-squares regression: a tutorial. *Anal Chimi Acta*. 1986;185:1–17.
7. Haaland DM, Thomas EV. Partial least-squares methods for spectral analysis. 1. relation to other quantitative calibration methods and the extraction of qualitative information. *Anal Chem*. 1988;60:1193–1202.
8. Helland IS. On the structure of partial least-squares regression. *Commun Statist Simula*. 1988;607–17(2):581.
9. Höskuldsson A. *Prediction Methods in Science and Technology*. Ventura, CA: Thor Publishing; 1996.
10. Wold S, Sjöström M, Eriksson L. PLS—regression: A basic tool of chemometrics. *Chemo Intell Lab Syst*. 2001;58:109–130.
11. Hastie T, Tibshirani R, Friedman J. *The Elements of Statistical Learning*. New York, NY: Springer; 2001.
12. Hoerl AE, Kennard RW. Ridge regression: biased estimation for nonorthogonal problems. *Technometrics*. 1970;12(1):55–67.
13. Seber GAF, Wild CJ. *Nonlinear Regression*. New York: Wiley; 1989.
14. Breiman L, Friedman JH, Olshen RA, Stone CJ. *Classification and Regression Trees*. Belmont, CA: Wadsworth International Group; 1984.
15. Johnson RA, Wichern DW. *Applied Multivariate Statistical Analysis*. 5th ed. Upper Sadle River, NJ: Prentice Hall; 2002.
16. Van der Heijden F, Duin RPW, De Ridder D, Tax DMJ. *Classification, Parameter Estimation and State Estimation*. Chichester: Wiley; 2004.
17. Barker M, Rayens W. Partial least squares for descrimination. *J Chemometrics*. 2003;17:166–173.
18. Wold S, Sjöström M. SIMCA: a method for analyzing chemical data in terms of similarity and analogy. *Chemon Theory Appl*. 1977;243–282
19. Derde MP, Massart DL. UNEQ: A disjoint modelling technique for pattern recognition based on normal distribution. *Anal Chimi Acta*. 1986;184:33–51.
20. Kadlec P, Gabrys B, S. S. Data-driven Soft Sensors in the process industry. *Comp Chem. Eng*. 2009;33(4):795–814.
21. Kresta JV, MacGregor JF, Marlin TE. Multivariate statistical monitoring of process operating performance. *Can J Chem Eng*. 1991;69:35–47.
22. Lin B, Recke B, Knudsen JKH, Jørgensen SB. A systematic approach for soft sensor development. *Comp Chem Eng*. 2007;31(5):419–425.
23. MacGregor JF, Kourti T. Multivariate Statistical Treatment of Historical Data for Productivity and Quality Improvements. Paper presented at: Foundation of Computer Aided Process Operations - FOCAPO 98; 1998.
24. Reis MS. A Multiscale Empirical Modeling Framework for System Identification *J Process Contr*. 2009;19(9):1546–1557.
25. Saraiva PM, Stephanopoulos G. Continuous Process Improvement through Inductive and Analogical Learning. *AIChE Journal*. 1992;38(2):161–183.
26. Box GEP, Hunter JS, Hunter WG. *Statistics for Experimenters: Design, Innovation, and Discovery*. 2nd ed. Hoboken, NJ: Wiley; 2005.
27. Guyon I, Elisseeff A. An Introduction to Variable and Feature Selection. *J Machine Learn Res*. 2003;3:1157–1182.
28. Vapnik V. *The Nature of Statistical Learning Theory*. 2nd ed. New York: Springer-Verlag; 2000.
29. Lisý JM, Cholvadová A, Kutej J. Multiple Straight-line Least Squares Analysis with Uncertainties in all Variables. *Comput Chem*. 1990;14:189–192.
30. Martínez À, Riu J, Rius FX. Evaluating Bias in Method Comparison Studies Using Linear Regression with Errors in Both Axes. *J Chemom*. 2002;16:41–53.
31. Mandel J. Fitting Straight Lines when both Variables are subject to Error. *J Quality Technol*. 1984;19(1):1–14.
32. Reis MS, Saraiva PM. A comparative study of linear regression methods in noisy environments. *J Chemom*. 2004;18(12):526–536.
33. Reis MS, Saraiva PM. Integration of data uncertainty in linear regression and process optimization. *AIChE J*. 2005;51(11):3007–3019.
34. Wentzell PD, Andrews DT, Hamilton DC, Faber K, Kowalski BR. Maximum likelihood principal component analysis. *J Chemom*. 1997;11:339–366.
35. Wentzell PD, Andrews DT, Kowalski BR. Maximum likelihood multivariate calibration. *Anal Chem*. 1997;69:2299–2311.
36. Møller SF, Von Frere J, Bro R. Robust methods for multivariate data analysis. *J Chemom*. 2005;19:549–563.
37. Ravasz E, Somera AL, Mongru DA, Oltvai ZN, Barabási A-L. Hierarchical organization of modularity in metabolic networks. *Science*. 2002;297:1551–1555.
38. Clauset A, Moore C, Newman MEJ. Hierarchical structure and the prediction of missing links in networks. *Nature*. 2008;453:98–101.
39. Guimerà R, Amaral LAN. Functional cartography of complex metabolic netwoks. *Nature*. 2005;433(24):895–900.
40. Newman JAS. Modularity and community structure in networks. *Proc Nat Acad Sci*. 2006;103(23):8577–8582.
41. Sokal RR, Rohlf FJ. *Biometry - The Principles and Practice of Statistics in Biological Research*. 3rd ed. New York: W.H. Freeman & Co.; 1994.
42. Höskuldsson A. PLS regression methods. *J Chemom*. 1988;2:211–228.
43. Rao KR, Lakshminarayanan S. Partial correlation based variable selection approach for multivariate data classification methods. *Chemom Intell Lab Syst*. 2007(86):68–81.
44. Yip AM, Horvath S. Gene network interconnectedness and the generalized topological overlap measure. *BMC Bioinform*. 2007;8(22):1–14.

45. Dillon WR, Goldstein M. *Multivariate Analysis - Methods and applications*. New York: Wiley; 1984.

46. Montegomery DC, Peck EA, Vining GG. *Introduction to Linear Regression Analysis*. 4th ed. Hoboken, NJ: Wiley; 2006.

47. de la Fuente A, Bing N, Hoeschele I, Mendes P. Discovery of meaningful associations in genomic data using partial correlation coefficients. *Bioinformatics*. 2004;20(18):3565–3574.

48. Reis MS, Saraiva PM. Analysis and classification of the paper surface. *Indust Eng Chem Res*. 2010;49(5):2493–2502.

49. Angélico D, Reis MS, Costa R, Saraiva PM, Ataíde J. *Profilometry: A Technique to Characterize Paper Surface*. Paper presented at: Tecnicelpa - XIX Encontro Nacional, 2005;Tomar, Portugal.

50. Costa R, Angélico D, Reis MS, Ataíde J, Saraiva PM. Paper superficial waviness: conception and implementation of an industrial statistical measurement system. *Anal Chimi Acta*. 2005;544:135–142.

Manuscript received Aug. 3, 2011, revision received Apr. 4, 2012, and final revision received Oct. 10, 2012.