Supervised Learning Methods For Variable Importance And Regression With Uncertainty On Dependent Data

by

Gianluca Gazzola

A dissertation submitted to the
School of Graduate Studies
Rutgers, The State University of New Jersey
In partial fulfillment of the requirements
For the degree of
Doctor of Philosophy in Operations Research
Written under the direction of
Myong K. Jeong
And approved by

_____

_____

_____

_____

_____

_____

New Brunswick, New Jersey
January, 2019

ABSTRACT OF THE DISSERTATION

Supervised Learning Methods For Variable Importance

And Regression With Uncertainty On Dependent Data

By GIANLUCA GAZZOLA

Dissertation Director:

Myong K. Jeong

This dissertation covers a collection of supervised learning methods targeted to data with complex dependence patterns. Part of our work orbits around the concept of variable importance, that is, the relative contribution an input variable to the prediction or the explanation of an output variable. Our interest in variable importance, and its estimation, is two-fold. On the one hand, as a tool for the characterization of data sets produced by multi-stage systems, where variables are related to each other via a network of correlations and causal dependencies. On the other hand, as a tool for the selection of minimal input-variable subsets with optimal predictive performance, in a more general framework involving data sets with an interesting structure of inter-variable dependence and redundancy. The rest of our work focuses on the problem of function approximation in the presence of uncertainty, and, specifically, on the calculation of optimal interpolating hyperplanes from data represented by convex polyhedra, rather than points. In this context, we propose algorithms to determine the spatial orientation of such polyhedra based on the multivariate relationships observed in the data, with particular focus on missing-value scenarios. For all of our methods, we present successful validation on an extensive and diverse array of real-world and simulated problems.

Dedication

A mio padre

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This dissertation focuses on a variety of questions that can be formulated as supervised learning problems, where, broadly speaking, some function is to be optimally estimated from a collection of observations. Some of these questions are derived from the science of data mining and revolve around the concept of variable importance, that is, the measurement of how relevant a certain input variable is for an output variable, for prediction or explanation purposes. Others come from the science of machine learning, and concentrate on the concept of predictive modeling in the presence of uncertainty, that is, dealing with data which is represented by sets, rather than points. These questions are far from lacking answers in the literature. The goal of our work here is to enrich and improve the quality of those answers by leveraging the often complex structure of multivariate dependencies that characterize real-world processes, in a more effective way than state-of-the-art methods do.

Our discussion starts in Chapter 2 with the problem of estimating variable importance within the context of multi-stage processes. In this framework, variables represent the different steps that make up a process or system, and as such are related to each other via a network of correlations and causal dependencies. Our contributions on this topic are two-fold. The first contribution is a new variant of variable importance measure, which builds on an existing measure that relies on data permutations and tree ensemble modeling. The main innovation in this measure is its use of variable dependencies to determine the sequence of data permutations to carry out, with the goal of effectively identifying variables that have a

true impact on a given output. The second contribution is a method for the characterization of a multi-stage process, based on the above measure and targeted at screening the process for stages with substantial relationships with the final stage. This characterization may, for example, be used as a basis to strategize intervention actions on key stages directed at influencing the process outcome.

In Chapter 3, we extend our work on variable importance to tackle problems of variable selection. We focus in particular on the issue of choosing a minimal subset of input variables from a larger set, with the purpose of building the simplest ensemble tree model with optimal prediction performance. The main novelty we propose here is a second permutation-based variable selection measure, in which the sequence of data permutations is chosen by aggregating observations, based on distance metrics that are functions of inter-variable dependencies. We show how this measure can be incorporated in a recursive algorithm to effectively eliminate redundant input variables from a dataset, without affecting the predictability of its output variable.

With Chapter 4, we shift the attention to the problem of building predictive models from uncertain data. Our work concentrates specifically on the estimation of hyperplanes to optimally interpolate numeric data whose uncertainty is represented by the solutions of systems of linear inequalities. For this purpose, we introduce a new generalization of the well-known Support Vector Regression formulation, where we replace point observations with convex polyhedra. We then propose an algorithm to obtain bounding-box-like convex polyhedral representations of data with missing values; our approach is novel in the way it orients such polyhedra in space based on the covariance structure of certain constructs that model the multivariate uncertainty on the unobserved values. We finally show how a characterization of these oriented bounding boxes can be used to define some of the hyper-parameters of our novel interpolating hyperplane formulation.

In all the above chapters, we use simulated experiments to study the behavior of each method as a function of the complexity of the underlying problem, as induced by such factors as dependence structure, dimensionality, amount of available information, and noise perturbations in the data, and as a function of the method's hyper-parameters, wherever

applicable. We also strive to provide extensive evidence of the competitiveness of our approaches with respect to those already discussed in the literature. We give particular emphasis to real-world applications, successfully testing our methods on data sets derived from many diverse fields, such as manufacturing, medical diagnostics, design of experiments, and survey research, among others.

We conclude this dissertation in Chapter 5, by outlining future directions of research, development, and experimental investigation for the methods we propose here.

Chapter 2

Variable importance for multi-stage processes

2.1 Introduction

In this chapter, we study the problem of quantifying the relative contribution that each stage of a multi-stage process (MSP) exhibits toward the others and, in particular, the final stage. In the real world, this problem is often non-trivial since MSPs may involve a large number of stages, which are related to each other via intricate networks of complex relationships. In manufacturing, for example, a given stage may be processing the output of multiple prior stages and, in turn, may be directing its own output to multiple subsequent stages; moreover, the transformation that such stage operates on its inputs to produce an output may be arbitrarily non-linear. Consequently, a given stage usually affects the final stage through a cascade of both direct and indirect input/output contributions, each of which is hard to model parametrically.

Variable importance (VI) in regression is an important topic in applied statistics and machine learning, focused on the assessment of the relative influence that input variables in a given set $X = \{X_1, X_2, \ldots, X_p\}$ have on a certain output variable of interest Y [1, 2, 3]. VI measures usually rely on the estimation of a regression model of the form

$$Y = f(X_1, X_2, \ldots, X_p) + \epsilon, \tag{2.1}$$

where f is a function and ϵ is an error term. Several state-of-the-art VI measures focus on the special case of a linear f; such measures aim at decomposing the predictable fraction of the variance of Y, represented by the coefficient of determination R^2 of the linear regression model, into p shares (non-negative values summing to 100%), one for each input variable [4, 5, 6, 7, 8]. The decomposition of R^2 into individual contribution shares is unique and straightforward only in the unlikely scenario in which all variables in X are mutually uncorrelated, because the variance of Y coincides with a weighted sum of the individual variances of each random variable on the right hand side of (2.1). When correlations within X are present, covariances between input variables also come into play in determining the variance of Y, rising the issue of how to reasonably break down pairwise variable contributions (taking place via covariance) into individual ones. Differences between VI measures for variance decomposition can essentially be traced back to the specific break-down criterion they enforce.

Another family of VI measures that gained popularity over recent years rely on machine-learning methods, where the relationship f is not specified a priori but is learned directly from the data. The most extensively researched regression modeling tools in this area are, to the best of our knowledge, random forests (RFs) [9, 10, 11]. RF-based measures define the importance of an input variable based on how sensitive the prediction performance of a RF is to random perturbations of such input variable. Although these measures do not formally assess relative importance via variance decomposition, some authors suggest their possible interpretability as variance decomposition methods [12]. The non-parametric nature of RFs and the fact that RFs can be applied to both numerical and categorical variables, make RF-based VI measures particularly appealing for our real-world application.

A common element to both the families of VI measures mentioned above is their focus on the contribution that $X_1, X_2, \ldots, X_p$ have toward Y, without considering the contribution that $X_1, X_2, \ldots, X_p$ may be providing to each other, due, for example, to input/output

relationships existing between them. In many practical applications, in fact, the nature and the direction of such relationships are completely unknown and the only feasible line of analysis is the one that considers Y as the dependent variable and $X_1, X_2, \ldots, X_p$ as the independent variables. For MSPs, relationships between stage variables may however be fully known. If we let Y be the output of the final stage and $X_1, X_2, \ldots, X_p$ be the outputs of the p previous stages, (2.1) would be the appropriate regression equation if all such p stages fed directly into the final one; it would not be, however, if we knew, for example, that some stage $j \in \{1, 2, \ldots, p\}$ took as input any of the outputs from stages $k < j$. Several regression-based approaches targeted to MSPs, loosely interpretable as methods for the assessment of VI, have been proposed in the context of process monitoring and product improvement [13, 14, 15, 16]. Most of these approaches, however, rely on assumptions that do not necessarily hold in the real world, such as linearity of the MSP relationships and non-correlation of certain process variables.

In this work, we introduce a virtually assumption-free method for the characterization of an MSP, relying upon a VI approach and aimed at screening the process for stages that exhibit significant relationships with the final stage [17]. We name this method "Integrated Multi-Stage Importance Assessment" (IMSIA). The method starts by deconstructing the MSP into a collection of single-stage building blocks, each of which is defined by a specific stage and by the set of all stages whose output feeds directly into it. Each building block induces a "local" regression problem, that can be modeled with an equation analogous to (2.1). From each building-block problem, the method extracts local relative stage importance measurements, based on a new variant of a conditional-permutation RF-based VI measure. Finally, it integrates these local measurements into "global" ones, by considering how stage contributions propagate along the network of technical relationships that define the MSP. These contribution measurements may, for example, be employed in an economic model to optimally allocate stage investments and intervention actions aimed at influencing the process outcome [13]; the study of this application, however, is beyond the scope of this work and is therefore not carried out here.

IMSIA models an MSP as a network, with stages identified by nodes, input-output

relationships between stages identified by directed arcs, and the strength of such relationships represented by arc and node weights. Within IMSIA's framework, the problem of stage importance assessment can be therefore assimilated to that of ranking nodes in a network, which is a widely researched topic in the field of graph mining. Several diverse applications of this problem can be found in a great deal of recent literature, involving, among others, social [18, 19], criminal [20, 21, 22], bibliographic [23, 24], patent [25, 26], and economic [27, 28] networks. State-of-the art methods generally estimate node importance with measures consisting of functions of certain topological features of a network (as, for example, in the case of measures that build on node or edge centrality [26, 22]). When network weights (typically, edge weights [28, 20]) are a component of the measure, such weights are often either known a priori as a piece of data attached to the network instance, or obtained via moderately simple transformations of such data. Some measures add an extra dimension to the analysis, by considering, besides the topological structure of the network, observations along variables or "attributes" associated to individual nodes; such observations are studied to discover, for example, whether a node exhibits anomalous properties with respect to the others [29, 30]. IMSIA goes one step further, by conjugating topological and attribute analysis with complex, non-parametric supervised modeling, used as the core tool to *learn* network weights from a data set descriptive of node-specific variables. A fundamental basis for this supervised approach is of course the prior knowledge of the *direction* of the inter-node relationships, since that is what allows the key distinction between input and output variables across nodes the network. It must also be emphasized that IMSIA defines the importance of a node (representing some stage in the MSP) with respect to a specific target node in the network (representing the final stage of the MSP); this conceptually differs from typical definitions of node importance, which are given with respect to the entire network.

This chapter is structured as follows. Section 2.2 provides relevant background knowledge on RF modeling and existing RF-based VI measures. Section 2.3 presents our first contribution, consisting of a novel variant of one of the measures described in the previous section, targeted to the assessment of local stage importance. Section 2.4 introduces the theoretical framework of IMSIA as our second contribution, while characterizing the inter-

play between the local and global components of the stage importance assessment problem. Section 2.5 illustrates the features of IMSIA on two artificial MSPs and on a real-world MSP of semiconductor manufacturing.

2.2 Elements of modeling and variable importance assessment with random forests in regression

2.2.1 Regression modeling

A RF is an ensemble of decision tree models [31, 32]. A decision tree is defined by a partition of the training data set, given by a hierarchy of recursive bisections or "splits" of the input variable space. At every recursion, one input variable and one splitting value along such variable are chosen to divide the data in two disjunct subsets, called "nodes". The choice of splitting input variable and corresponding splitting value is carried out by optimizing a certain objective function, which varies depending on the type of decision tree model employed by the RF. For CART [33], the class of trees we use in our framework, the optimization criterion is given by the improvement in homogeneity of the output variable Y in the two subsets resulting from the bipartition, with respect to the same in the unsplit set. Let us call U the unsplit set in the current recursion (with U coinciding with the entire training data set in the first recursion), and let disjunct sets $S_1^{j,v}$ and $S_2^{j,v}$ represent a bipartition of U, induced by a certain input variable X_j and a certain value v of X_j observed in the training data; the recursive optimization problem to solve becomes the following:

$$\max_{j,v} \frac{1}{|U|} \sum_{i \in U} (y_i - \bar{y}_U)^2 - \left[\frac{1}{|S_1^{j,v}|} \sum_{i \in S_1^{j,v}} \left(y_i - \bar{y}_{S_1^{j,v}} \right)^2 + \right.$$
$$\left. + \frac{1}{|S_2^{j,v}|} \sum_{i \in S_2^{j,v}} \left(y_i - \bar{y}_{S_2^{j,v}} \right)^2 \right], \quad (2.2)$$

where y_i is the value of the Y variable in observation i, $\bar{y}_U = \frac{1}{|U|}\sum_{i\in U} y_i$, $\bar{y}_{S_1^{j,v}} = \frac{1}{|S_1^{j,v}|}\sum_{i\in S_1^{j,v}} y_i$, $\bar{y}_{S_2^{j,v}} = \frac{1}{|S_2^{j,v}|}\sum_{i\in S_2^{j,v}} y_i$, and $|\cdot|$ denotes the set cardinality operator. Once (2.2) has been solved for the current recursion, each of the resulting $S_1^{j,v}$ and $S_2^{j,v}$ sets defines a new set U to solve the optimization problem on, and so on. When the solution of (2.2) has a non-positive value, that is, when no improvement in node homogeneity is possible, or when $|U| \leq m$, with $m \geq 1$ parameter, the recursion is stopped and U becomes a "terminal" node. The collection of terminal nodes of a fully grown tree represents, in a set-theoretical sense, a partition of the training data. The predicted output value $\hat{y}_{\boldsymbol{x}_{new}}$ of an unseen (testing) observation $\boldsymbol{x}_{new}$ is computed as follows:

$$\hat{y}_{\boldsymbol{x}_{new},t} = \frac{1}{|U_{\boldsymbol{x}_{new}}|}\sum_{i\in U_{\boldsymbol{x}_{new}}} y_i, \tag{2.3}$$

where $U_{\boldsymbol{x}_{new}}$ is the unique terminal node defined by a set of recursive splitting rules that are entirely satisfied by $\boldsymbol{x}_{new}$.

Within a RF, randomness is injected into the decision-tree building process by means of two sampling techniques known as "tree bagging" and "variable bagging". The first consists of growing each tree on a different independent bootstrap sample of all available observations. The second consists of selecting each split within a tree by solving (2.2) on a different subset of $q \leq p$ indices j, sampled uniformly at random without replacement from the total of p. Due to bagging, data partitioning rules and predictions differ across trees; an overall ensemble prediction $\hat{y}_{\boldsymbol{x}_{new},RF}$ for $\boldsymbol{x}_{new}$ is obtained averaging out the individual tree predictions:

$$\hat{y}_{\boldsymbol{x}_{new},RF} = \frac{1}{T}\sum_{t=1}^{T} \hat{y}_{\boldsymbol{x}_{new},t}, \tag{2.4}$$

where T is the number of trees in the RF and $\hat{y}_{\boldsymbol{x}_{new},t}$ is the prediction returned by tree t. Bagging-induced tree diversity in RFs has been shown to reduce overfitting and prediction variance, and in the specific case of variable bagging, to improve the detection of relevant interactions among input variables compared to standard, unbagged trees.

An estimate of the generalization error of a regression model is usually calculated on a testing data set that only includes observations that were not used for training the model. In RFs, each tree is trained on only a fraction of the overall training data set available to the RF; the remaining fraction of the data (the set of so-called "out of bag", OOB, observations) can serve as a testing set for that tree. Now, if we let: $O = \{(x_{1_i}, x_{2_i}, \ldots, x_{p_i}, y_i),\ i = 1, 2, \ldots n\}$ be the set of training observations available to the RF, along variables $X_1, X_2, \ldots, X_p$ and Y, respectively; $O_t \subset O$ be the set of OOB observations for tree t; and $\hat{y}_{i,t}$ be the prediction of tree t for observation i, we can estimate the generalization error MSE_{RF} of the RF as follows:

$$MSE_{RF} = \frac{1}{n} SSR_{RF}, \tag{2.5}$$

where

$$SSR_{RF} = \sum_{i=1}^{n} (\bar{\hat{y}}_i - y_i)^2, \tag{2.6}$$

$$\bar{\hat{y}}_i = \frac{1}{T_i} \sum_{t:i \in O_t} \hat{y}_{i,t}, \tag{2.7}$$

$$T_i = |\{t \in \{1, 2, \ldots, T\} : i \in O_t\}|. \tag{2.8}$$

In other words, MSE_{RF} corresponds to the mean square error of the n predictions for the observations in set O, each of which obtained as an average of individual tree predictions, computed over the only trees for which the corresponding observation is OOB. Based on this quantity, we can compute for the RF an OOB analog of the coefficient of determination R^2 for linear regression:

$$R^2_{RF} = 1 - \frac{SSR_{RF}}{SST_{RF}}, \tag{2.9}$$

with

$$SST_{RF} = \sum_{i=1}^{n} (y_i - \bar{y})^2 . \tag{2.10}$$

We may note that, like for R^2 in linear regression, the theoretical upper bound of R^2_{RF} is equal to 1 (for a RF with perfect OOB predictions), but unlike for R^2, its theoretical lower bound is not equal to 0 (RFs with very poor OOB performance may yield negative R^2_{RF}). Consequently, we may interpret R^2_{RF} as a goodness-of-fit measure, which is related to, but not coinciding with, the fraction of variability of Y explained by the RF model.

2.2.2 Permutation importance

The assessment of the relative contribution of each input variable to the output variable within a RF model is a problem that has been extensively discussed in the literature on VI. Among the different VI measures for RFs proposed to date, "permutation importance" (PI), first introduced in [31] for classification modeling, is the one that appears to have received the most attention from researchers and practitioners from various applied sciences, and is considered state-of-the-art by numerous authors. There exist two main variants of PI: one which we will refer to as "marginal", and one which we will refer to as "conditional". Marginal PI can be described as the increase in OOB mean square error when a certain input variable of interest is randomly permuted in the OOB data of a tree, averaged out across all trees in the RF [12]. Now, if we let $\hat{y}_{i,t,\pi_j}$ be the prediction of tree t for observation i, $i \in O_t$, after replacing the X_j values in O_t with a random permutation of such values, the PI of variable interest X_j is given by:

$$I^P_j = \frac{1}{T} \sum_{t=1}^{T} \left(MSE_{t,\pi_j} - MSE_t \right), \tag{2.11}$$

where

$$MSE_t = \frac{1}{|O_t|} \sum_{i \in O_t} (y_i - \hat{y}_{i,t})^2 \tag{2.12}$$

and

$$MSE_{t,\pi_j} = \frac{1}{|O_t|} \sum_{i \in O_t} \left(y_i - \hat{y}_{i,t,\pi_j}\right)^2 . \tag{2.13}$$

In PI, the values of X_j are randomly permuted in order to break any dependence existing between X_j and Y. If the two variables are indeed dependent, that is, if X_j is an important input variable, then the expected value of I_j^P is positive, since the permutation will on average deteriorate the predictability of Y from the set of input variables ($MSE_{t,\pi_j} > MSE_t$). Otherwise, the expected value of I_j^P is zero, since the permutation will have on average no significant effect on the predictability of Y ($MSE_{t,\pi_j} = MSE_t$). From an empirical standpoint, important input variables will virtually always have a strictly positive measured PI, whereas it possible, although unusual, for unimportant input variables to have a measured PI that falls slightly below zero.

Some authors showed that PI tends to overestimate the importance of variables X_j that are weakly influential for Y, but that happen to be highly correlated with variables X_k that are strongly influential for Y [34, 35]. There are two factors known to cause this issue. The first one arises when X_j is selected for split as "surrogate" of X_k, for example when X_j appears in the pool of q variables randomly sampled as candidate for splitting at a given recursion of the tree-learning process, while X_k does not. In general, this surrogate effect becomes more probable as the value of q decreases. Another, more interesting factor causing the overestimation of the importance of weakly influential variables involves the notion of statistical independence that the variable permutation scheme in PI is testing. Paraphrasing the reasoning given in [34], we introduce the following:

Proposition 2.1. *X_j is permutation-important $\not\Longrightarrow$ X_j is influential for Y.*

Proof. Let $W^{(j)} = \{X_1, X_2, \ldots, X_p\} \setminus \{X_j\}$ and let $P(X_j, W^{(j)}, Y)$ be the unknown joint

distribution from which observations are sampled. A random permutation of X_j will not affect $P(X_j, W^{(j)}, Y)$ if and only if (a) $X_j \perp Y$ and (b) $X_j \perp W^{(j)}$, since

$$\text{(a)} \wedge \text{(b)} \iff P(X_j, W^{(j)}, Y) = P(X_j)P(W^{(j)}, Y) = P(\Pi_j)P(W^{(j)}, Y), \tag{2.14}$$

where Π_j is a random permutation of X_j. Now, consider MSE_t and MSE_{t,π_j} as statistics calculated on a sample of $P(X_j, W^{(j)}, Y)$ and on a sample of $P(\Pi_j, W^{(j)}, Y)$, respectively. Given (2.14), we have

$$\text{(a)} \wedge \text{(b)} \iff E\left[I_j^P\right] = 0, \tag{2.15}$$

which implies that observing a significantly non-zero I_j^P value may result from the sole violation of (b), which is irrelevant to the influence of X_j on Y.

□

Proposition 2.1 tells us that, if we observe a positive value of I_j^P, we will not be able to conclude if the data violates the condition $X_j \perp Y$ (and therefore if X_j plays a role in predicting Y), or if it violates the condition $X_j \perp W^{(j)}$ (which per se does not say anything about the predictability of Y from X_j), or if it violates both. We note, however, that this issue can be eliminated if we condition X_j on $W^{(j)}$ prior to permuting X_j, in such a way that

$$\begin{aligned}(X_j \perp Y)|W^{(j)} \iff P(X_j, Y|W^{(j)}) = \\ = P(X_j|W^{(j)})P(Y|W^{(j)}) = P(\Pi_j|W^{(j)})P(Y|W^{(j)}).\end{aligned} \tag{2.16}$$

With this type of conditioning, $I_j^P > 0$ would, on average, result only from a violation the interesting condition of independence between Y and X_j. This line of reasoning led

to the proposal of a conditional variant of the marginal permutation measure, where the conditioning scheme (that is, the values of $W^{(j)}$ to condition upon) for each tree of the RF is given by the bisection rules involving $W^{(j)}$ within the tree [34]. The union of all such rules defines a partition $S_t^{(j)}$ of the $W^{(j)}$ space into disjoint subregions $s_{l_t}^{(j)}$, $l = 1, 2, \ldots, |S_t^{(j)}|$. If we now let $O_{s_{l_t}^{(j)}} \subseteq O_t$ be the set of OOB observations located in subregion $s_{l_t}^{(j)}$, conditional permutation importance I_j^{CP} can be formulated as follows:

$$I_j^{CP} = \frac{1}{T} \sum_{t=1}^{T} \left(MSE_{t,\pi,S_t^{(j)}} - MSE_t \right), \tag{2.17}$$

where

$$MSE_{t,\pi,S_t^{(j)}} = \frac{1}{|O_t|} \sum_{l=1}^{|S_t^{(j)}|} \sum_{i \in O_{s_{l_t}^{(j)}}} \left(y_i - \hat{y}_{i,t,\pi,s_{l_t}^{(j)}} \right)^2 \tag{2.18}$$

and $\hat{y}_{i,t,\pi,s_{l_t}^{(j)}}$ is the prediction of tree t for observation i, after randomly permuting X_j within $s_{l_t}^{(j)}$.

2.3 A novel conditional permutation measure

2.3.1 Association-biased bagged random forests for data partitioning

The main rationale behind conditional PI (CPI) lies in the fact that each subregion in $S_t^{(j)}$ contains a group of OOB observations whose values along variables in $W^{(j)}$ are significantly more homogeneous than in the overall OOB set. Therefore, permuting X_j within each group of OOB observations, that is conditioned on specific values of $W^{(j)}$, allows to reduce the effect of the dependences between X_j and $W^{(j)}$ when computing PI. We note that, to more effectively isolate such effect, it is advisable to devise an alternative grouping scheme that favors a higher degree of homogeneity specifically along those variables in $W^{(j)}$ that are most statistically related with X_j. Within CPI, the grouping scheme is a byproduct of RF training, which per se is optimized for predictive accuracy and not for effective isolation of

inter-variable dependences. For this reason, we propose to address the problem of defining a partitioning grid for CPI separately from that of training a predictive model for CPI.

In our proposed approach, the standard RF is kept as predictive modeling tool, whereas a different variant of RF, which we will refer to as dependence-biased-bagged RF (DBB-RF), is used for data partitioning. In DBB-RF, variable bagging is not carried out uniformly at random, but with probabilities that increase with the strength of the dependence between X_j and the input variables in $W^{(j)}$; this will force a larger number of bisection rules within a tree to involve variables in $W^{(j)}$ that are more problematic to the assessment of PI. Since these probabilities are a function of index j, a different DBB-RF is trained for every input variable of interest.

Let us assume the degree of dependence between two variables is estimated from the data by a certain function Ω, such that for any three n-vectors $\boldsymbol{a}_1$, $\boldsymbol{a}_2$, $\boldsymbol{a}_3$ of observations along variables A_1, A_2, A_3, respectively, the following holds:

I. $\Omega(\boldsymbol{a}_1, \boldsymbol{a}_2) = \Omega(\boldsymbol{a}_2, \boldsymbol{a}_1)$.

II. $\Omega(\alpha\boldsymbol{a}_1 + \gamma\boldsymbol{e}, \alpha\boldsymbol{a}_2 + \gamma\boldsymbol{e}) = \Omega(\boldsymbol{a}_1, \boldsymbol{a}_2)$, $\forall\alpha, \gamma \in \mathbb{R}$, $\boldsymbol{e} = (1, 1, \ldots, 1)^T$.

III. The range of $\Omega(\boldsymbol{a}_1, \boldsymbol{a}_2)$ is bounded (w.l.o.g., assumed to be $[0, 1]$).

IV. If $\Omega(\boldsymbol{a}_1, \boldsymbol{a}_2) > \Omega(\boldsymbol{a}_1, \boldsymbol{a}_3)$, then A_1 and A_2 are estimated to be more strongly dependent than A_1 and A_3.

Letting $\omega \in [0, 1/p]$ be a parameter, for variable of interest X_j, variables within a tree in DBB-RF are bagged according to probability vector $\boldsymbol{w}_j^T = (w_{j1}, w_{j2}, \ldots, w_{jp})$, defined via the following procedure:

A. Let $w^*_{jj} = 0$.

B. For $k = 1, 2, \ldots, p, k \neq j$, let $w^*_{jk} = \frac{\Omega(\boldsymbol{x}_k, \boldsymbol{x}_j)}{\sum_{l \neq j} \Omega(\boldsymbol{x}_l, \boldsymbol{x}_j)}$.

C. Let $M_j = \{k : w^*_{jk} < \omega\}$. For all $k \in M_j$, let $w_{jk} = \omega$.

D. For $k \notin M_j$ let $w_{jk} = (1 - |M_j|\omega)\frac{w^*_{jk}}{\sum_{l \notin M_j} w^*_{jl}}$.

The procedure starts by assigning a zero probability to X_j in step A; probabilities for

variables in $W^{(j)}$, on the other hand, are defined proportionally to their respective degree of dependence with X_j in step B. If any of the resulting probability values are smaller than ω, such values are replaced with ω in step C, whereas step D takes care or renormalizing the remaining probability values in such a way that the sum of the elements of $\boldsymbol{w}_j$ is equal to 1; note that $\omega \leq 1/p$ implies $1 - |M_j| \geq 0$ and consequently $w_{jk} \geq 0$. The resulting vector $\boldsymbol{w}_j$ enforces a dependence-based probabilistic bias, in that the stronger the dependence between a certain variable $X_k, k \neq j$, and X_j, the larger the probability that X_k will be selected as candidate for split at every recursion. Since DBB-RF is used as a data partitioning tool along variables in $W^{(j)}$, variable X_j can in principle be excluded from bagging altogether, by choosing $\omega = 0$; however, it may be advisable to enforce a small but non-zero probability lower bound on X_j, in order for the DBB-RF to include partitioning rules involving variables X_k that exhibit interactions with X_j. Parameter ω, in fact, serves as a lower bound for all of the probabilities in $\boldsymbol{w}_j$. As per the form of Ω, a natural choice for instances in which all input variables are numerical is the following:

$$\Omega(\boldsymbol{a}_1, \boldsymbol{a}_2) = |\rho(\boldsymbol{a}_1, \boldsymbol{a}_2)|^u, \tag{2.19}$$

where $\rho(\boldsymbol{a}_1, \boldsymbol{a}_2)$ is the sample Pearson's correlation coefficient of $\boldsymbol{a}_1$ and $\boldsymbol{a}_2$, and $u > 0$ is a parameter that allows to control the extent of the probabilistic bias within $\boldsymbol{w}_j$. An alternative function that may be better suited for data sets with non-linear relationships between (all numerical) input variables is the following:

$$\Omega(\boldsymbol{a}_1, \boldsymbol{a}_2) = MIC(\boldsymbol{a}_1, \boldsymbol{a}_2)^u, \tag{2.20}$$

where $MIC(\boldsymbol{a}_1, \boldsymbol{a}_2)$ is the maximal information coefficient [36], and u is as in (2.19). For instances that include categorical variables, we instead propose the following:

$$\Omega(\boldsymbol{a}_1, \boldsymbol{a}_2) = 1 - p_v(\boldsymbol{a}_1, \boldsymbol{a}_2)^u, \tag{2.21}$$

where $p_v(\boldsymbol{a}_1, \boldsymbol{a}_2)$ is the p-value of an appropriate test of statistical independence on vectors

$\boldsymbol{a}_1$ and $\boldsymbol{a}_2$ and $u > 0$ is a parameter that plays the same role as in (2.19).

2.3.2 Conditional permutation importance with dependence-biased bagged random forests

Our proposed revised procedure for calculating CPI, which we will refer to as DBB-CPI, can be summarized by the following steps:

A. Choose ω and Ω, and accordingly compute $\boldsymbol{w}_j$, $j = 1, 2, \ldots, p$.

B. Train a standard RF on the available observations.

C. For $j = 1, 2, \ldots, p$, repeat steps C1–C2:

C1. Train the j-th DBB-RF with $\boldsymbol{w}_j$ as variable bagging probability vector.

C2. Compute and return

$$I_j^{DBB-CP} = \frac{1}{T} \sum_{t=1}^{T} \left(MSE_{t,\pi,S_t^{(j)DBB}} - MSE_t \right), \tag{2.22}$$

where $S_t^{(j)DBB}$ is defined analogously to $S_t^{(j)}$ in section 2.2.2, except for the fact that it is extracted from the j-th DBB-RF.

Remark 1. Within a given subregion, a permutation can take place provided that multiple OOB observations are present; this condition may not always hold true if $S_t^{(j)DBB}$ is defined considering tree t of DBB-RF j from its root all the way down to its terminal nodes. Therefore, some form of tree pruning may be applied prior to carrying out step C2: a possible heuristic pruning criterion is to consider trees up to depth

$$D \geq \lfloor \log_2(cn/m) \rfloor, \tag{2.23}$$

where $c > 0$ is the fraction of OOB observations chosen for the DBB-RFs, and $m \geq 1$ is an integer parameter. With such a D, if the OOB observations were hypothetically distributed

uniformly across the terminal nodes of the pruned tree, each terminal node would contain at least m OOB observations, even if the pruned tree were full (that is, if it had 2^D terminal nodes).

Remark 2. The supervised nature of DBB-RF as a method for data partitioning allows to build the grouping scheme along those variables in $W^{(j)}$ that are not only most statistically related with X_j, but also statistically related with the output variable Y, just like with the original CPI measure. We note that grouping data along variables that have no explanatory/predictive power toward Y is not necessary (and therefore computationally wasteful), since the importance inflation issue identified for marginal PI only involves weakly influential variables that are statistically dependent on strongly influential ones, as explained in section 2.2.2.

2.4 Stage importance within a multi-stage process

2.4.1 Local relative contribution assessment

Let us consider a MSP with $|Q|$ stages indexed by integers forming a set Q. Let us assume that every stage $b \in Q$ yields an output described by numerical variable Z_b, of which n observations are available. Let us assume that the inputs of stage b may include outputs of other stages with index smaller than b, and, if they do, that such information is known. If the output Z_a of stage a is the input of stage b, $a < b$, then we will say that stage a and stage b have a "direct technical relationship" (DTR), that a is a "direct technical predecessor" (DTP) of stage b, and that b is a "direct technical successor" (DTS) of stage a. Let $Q_b = \{a \in Q, a < b : a \text{ is a DTP of } b\}$.

Let us further assume that stage inputs may include sources other than DTP outputs, such as raw materials or other (possibly controllable) process factors in a manufacturing MSP. These further input sources, which we will refer to as "other sources" (OSs), are indexed by integers forming a set S. For simplicity, let us assume that indices in Q induce a partition $\{S_1, S_2, \ldots, S_{|Q|}\}$ on S, where S_b represents the set of OSs of stage $b \in Q$. Let us

assume that every OS h in S is described by variable V_h, which may be either numerical or categorical. Finally, let us assume that n $|Q|+|S|$-dimensional observations are available, each defined by the values of all variables in Q and S for a specific item processed by the MSP.

Stage b, its DTPs, and its OSs altogether define what we call "subsystem" (SUB) b. SUB b identifies a regression equation analogous to (2.1), in which Z_b represents the output variable, and $Z_{\alpha_1}, Z_{\alpha_2}, \ldots, Z_{\alpha_{|Q_b|}} : \alpha_1, \alpha_2, \ldots, \alpha_{|Q_b|} \in Q_b$ (with indices $\alpha_\cdot$ spanning all of b's DTPs) and $V_{\beta_1,b}, V_{\beta_2,b}, \ldots, V_{\beta_{|S_b|},b} : \beta_1, \beta_2, \ldots, \beta_{|S_b|} \in S_b$ (with indices $\beta_\cdot$ spanning all of stage b's OSs) represent the input variables. Therefore, direct contributions to stage b from its DTPs and OSs can be quantified applying a VI measure to the regression equation induced by SUB b. The regression equation of SUB b is of the form

$$Z_b = f_b\left(Z_{\alpha_1}, Z_{\alpha_2}, \ldots, Z_{\alpha_{|Q_b|}}, V_{\beta_1}, V_{\beta_2}, \ldots, V_{\beta_{|S_b|}}\right) + \epsilon_b. \tag{2.24}$$

Let us call RF_b the (standard) RF trained on the n observations along the variables appearing in (2.24) and let us use the quantity

$$r_b^2 = \max(R^2_{RF,b}, 0) \tag{2.25}$$

as a proxy of the proportion of Z_b's variability explained by the overall variability of the ordered set of input variables $X_b = \{Z_{\alpha_1}, Z_{\alpha_2}, \ldots, Z_{\alpha_{|Q_b|}}, V_{\beta_1}, V_{\beta_2}, \ldots, V_{\beta_{|S_b|}}\}$.

For the j-th input variable in X_b, with $j = 1, 2, \ldots, |Q_b|$, corresponding to DTPs and $j = |Q_b|+1, |Q_b|+2, \ldots, |Q_b|+|S_b|$ corresponding to OSs, let us then define

$$I_{j,b} = \max\left(0, I_{j,b}^{DBB-CP}\right) \tag{2.26}$$

and

$$\tilde{I}_{j,b} = \frac{I_{j,b}}{\sum_{k=1}^{|Q_b|+|S_b|} I_{k,b}}. \tag{2.27}$$

The max operator in (2.26) takes care of those rare but possible cases in which random fluctuations push the measured CPI of an irrelevant variable below 0; (2.27) subsequently normalizes the CPI values into proper fractions summing to 1.

Finally, for indices $j \leq |Q_b|$, let us define the "local" (at the SUB level) relative contribution $c^{DTP}_{\alpha_j,b}$ of DTP α_j's output variable Z_{α_j} to Z_b as follows:

$$c^{DTP}_{\alpha_j,b} = r_b^2 \tilde{I}_{j,b}. \tag{2.28}$$

Analogously, for indices $j > |Q_b|$, let us define the local relative contribution $c^{OS}_{\beta_{j-|Q_b|},b}$ of OS $\beta_{j-|Q_b|}$'s variable $Z_{\beta_{j-|Q_b|}}$ to Z_b as follows:

$$c^{OS}_{\beta_{j-|Q_b|},b} = r_b^2 \tilde{I}_{j,b}. \tag{2.29}$$

On a typical instance, (2.28) and (2.29) express the relative contribution of an input variable as a share of $R^2_{RF,b}$, proportional the VI of such input variable, as given by DBB-CPI. For instances in which the measured VI of the j-th input variable is negative, such variable is considered non-explanatory of the variability of Z_b (since (2.26) sets $I_{j,b}$ to 0). For instances in which the prediction performance of RF_b is poor to the point of producing a negative value for $R^2_{RF,b}$, the whole set X_b is considered non-explanatory (since (2.25) sets r_b^2 to 0), and consequently so are all of the individual input variables it contains (since $r_b^2 = 0$ is a multiplier in both (2.28) and (2.29)).

Finally, we define

$$c_b = 1 - r_b^2 \tag{2.30}$$

as the relative contribution to Z_b of any uncontrollable factors that inherently affect stage

b on top of the contributions coming from stage b's DTPs and OSs. Complementarily to r_b^2, c_b can be interpreted as a proxy of the (uncontrollable) portion of Z_b's variability that is not explained by any of the variables in X_b. The sum

$$c_b + \sum_{\beta \in S_b} c_{\beta,b}^{OS} \tag{2.31}$$

can be interpreted as the relative contribution that stage b has toward its own output, since such contributions come from factors that are inherent to stage b itself and do not have to do with previous stages.

In the special case where $|Q_b| + |S_b| = 1$, the right hand side of (2.28) and (2.29) reduces to r_b^2, since all of the predictable portion of the variability of Z_b is explained by the only DTP or OS of stage b. In the special case in which $Q_b = \{\emptyset\}$ and $|S_b| = \{\emptyset\}$, there are no relative contribution shares to be calculated, and we will simply set $c_b = 1$.

2.4.2 Global relative contribution assessment via integration of local relative contribution assessments

Within a MSP, intermediate stages are typically both DTPs for some and DTSs for others, in the sense that they send their output, obtained by transforming the output of previous stages, to later stages. For these intermediate stages, the relative contribution they provide to their DTSs can in principle be explained in terms of relative contributions that they receive from their DTPs, in the sense that the latter can be viewed as the root cause of the former. In other words, the DTPs of an intermediate stage contribute indirectly to the DTSs of such intermediate stage via the intermediate stage itself. Based on this argument, we propose to calculate the relative contribution of a given stage $d \in Q$ to another stage $e \in Q$ by combining the VI measurements locally assessed at each of the SUBs involving intermediate stages existing between d and e, if any. This combination takes place as a recursive decomposition of relative contributions, obtained backtracking from e to d along all possible sequences of DTRs through which the output of d is progressively transformed

into an input of e.

Let us assume the collection of DTRs in the MSP are given as a set of ordered pairs $R = \{(\alpha, b) : \alpha, b \in Q, \alpha < b\}$. Analogously, let $R^{'} = \{(\beta, b) : \beta \in S_b, b \in Q\}$, be the set of ordered pairs defining the collection of OS-to-stage relationships (OSRs). Let us represent the MSP with a directed acyclic network $N = (Q \cup S, R \cup R^{'})$ on stage/node set $Q \cup S$ and DTR/arc set $R \cup R^{'}$. For the sake of illustration, let us suppose that stages d and e have an indirect relationship, given by a sequence of DTRs involving an intermediate stage g, defining a path from d to e in N. Let us further assume that: stage e has DTPs other than g and some OSs; stage g has DTPs other than d, and some OSs; stage d has some DTPs and one OS, say, o (Figure 2.1). Based on what we explained in section 2.4.1, the relative contribution $c_{g,e}^{DTP}$ of stage g to stage e can be interpreted as a proxy of the share of output e's variability explained by output g's variability; analogously, $c_{d,g}^{DTP}$ can be interpreted as a proxy of the share of output g's variability explained by output d's variability; finally, $C_d = c_d + c_o^{OS}$ can be interpreted as a proxy of the share of output d's variability explained by factors that are inherent to stage d itself (that is, by factors not related to stage d's DTPs). Therefore, the product

$$C_{d,e} = c_{g,e}^{DTP} c_{d,g}^{DTP} C_d \tag{2.32}$$

may be used to represent the relative contribution of stage d to stage e, modeled as the sub-sub-share of the relative contribution of stage g to stage e that is attributable to stage d. This sub-sub-share is obtained by a specific sequence of three successive decompositions of the overall contributions to stage e, defined by the three terms of the product, backtracking from stage e to stage d along path $d - g - e$.

Generalizing the above argument to arbitrary MSP topologies, we may define the "global" (at the MSP level) relative contribution $C_{d,e}$ of some stage d to some stage e in N as:

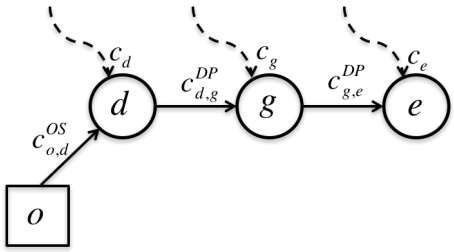

Figure 2.1: Extract of a hypothetical MSP network, where the three circles represent the output variables of stages d, g, and e, the square represents OS variable o of stage d, and solid arcs represent direct relationships between pairs of variables; dotted arcs represent unspecified direct relationships involving the rest of the MSP. Node and arc labels represent local relative contribution coefficients. Assuming o is the only OS of stage d, the relative contribution of stage d to stage e is defined by (2.32).

$$C_{d,e} = \begin{cases} C^{DTP}_{d,e} C_d & \text{if } e \geq d \\ 0 & \text{otherwise} \end{cases}, \tag{2.33}$$

where

$$C_d = c_d + \sum_{\beta \in S_d} c^{OS}_{\beta,d}, \tag{2.34}$$

$$C^{DTP}_{d,e} = \begin{cases} \sum_{i=1}^{n_{d,e}} \prod_{(a,b) \in p^i_{d,e}} c^{DTP}_{a,b} & \text{if } e > d \text{ and } P_{d,e} \neq \{\emptyset\} \\ 1 & \text{otherwise} \end{cases}, \tag{2.35}$$

and $P_{d,e} = \{p^1_{d,e}, p^2_{d,e}, \ldots, p^{n_{d,e}}_{d,e}\}$ is the set of $n_{d,e}$ paths from d to e in N. If any paths from stage d to stage e exist, then each of them represents a different sequence of successive decompositions of relative contributions to stage e, given by a specific chain of products of relative contribution shares in the first row of (2.35); all such sequences end with the same last decomposition, represented by term C_d in the first row of (2.33). If stage d is neither a direct or an indirect predecessor of stage e, then its relative contribution to stage e is obviously 0, unless $d = e$, in which case it reduces to C_d, since $C^{DTP}_{d,e} = 1$ in (2.35).

Based on the definition of the local relative contributions as shares, it is straightforward

to see that the following holds:

$$\sum_{i=1}^{|Q|} C_{i,e} = 1 \ \forall e \in Q \tag{2.36}$$

and that therefore the global relative contribution measurements obtained by the integration of local relative contribution shares as per (2.33) can themselves be interpreted as shares.

As a concluding remark, it must be emphasized that, as noted in [12, 1], the concept of VI is not uniquely defined in the literature, in the sense that there is no agreed upon theoretical "true" quantity that a VI measure should be capable of accurately estimating. Rather, each measure formulates and characterizes importance based on its own "empirical" criteria, which may be more or less appropriate depending on the application on hand. For IMSIA, which is a VI approach targeted to the discovery of key stages affecting a given later stage, these criteria focus: at the local level, on the identification of truly important relationships between inputs and output, via variable conditioning; at the global level, on the location of the root causes of stage contributions, via the topological analysis of the MSP network.

2.4.3 Overall IMSIA procedure for the assessment of stage importance

Given a MSP defined by a set of stages Q with output variables $Z_1, Z_2, \ldots, Z_{|Q|}$, a set of OSs S with variables $V_1, V_2, \ldots, V_{|S|}$, a set of DTRs R, a set of OSRs $R^{'}$, and n $(|Q| + |S|)$-dimensional observations along $Z_1, Z_2, \ldots, Z_{|Q|}, V_1, V_2, \ldots, V_{|S|}$, the overall procedure to assess the relative contribution of stage d to stage e, $d, e \in Q$, can be summarized by the following steps:

A. Represent the MSP with a directed network $N = (Q \cup S, R \cup R^{'})$; choose ω and Ω. For each node $b \in Q$, iterate steps B–E.

B. Consider SUB b, its stage set Q_b, and its OS set S_b. If $|Q_b| + |S_b| = 0$, let $c_b = 1$, label

node b with c_b, and go to the next iteration, otherwise go to step C.

C. Estimate standard RF b from observations along stage b's ordered input variable set $X_b = \{Z_{\alpha_1}, Z_{\alpha_2}, \ldots, Z_{\alpha_{|Q_b|}}, V_{\beta_1}, V_{\beta_2}, \ldots, V_{\beta_{|S_b|}}\}$ and output variable Z_b.

D. Compute $c_b = 1 - r_b^2$ and label node b with c_b.

E. For $j = 1, 2, \ldots, |Q_b| + |S_b|$, repeat steps E1 – E3b

E1. Compute $\boldsymbol{w}_{j,b}$; train DBB-RF j, b with $\boldsymbol{w}_{j,b}$ as variable bagging probability vector.

E2. Compute $\tilde{I}_{j,b}$ with DBB-CPI.

E3. If $j \leq |Q_b|$, go to step E3a; otherwise, go to step E3b:

E3a. Let $c^{DTP}_{\alpha_j,b} = r_b^2 \tilde{I}_{j,b}$, and label the arc connecting node $\alpha_j \in S$ to node b with $c^{DTP}_{\alpha_j,b}$.

E3b. Let $c^{OS}_{\beta_{j-|Q_b|},b} = r_b^2 \tilde{I}_{j,b}$, and label the arc connecting node $\beta_{j-|Q_b|} \in S$ to node b with $c^{OS}_{\beta_{j-|Q_b|},b}$.

F. Compute and return $C_{d,e}$.

This procedure can be used to compute relative contributions between any two stages in the MSP. If the interest is specifically on relative contributions toward the final stage, say stage s, one would simply set $e = s$.

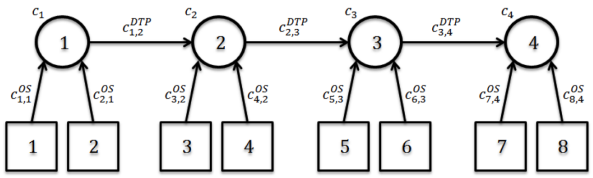

Figure 2.2: Labeled network for the sequential artificial MSP.

2.5 Numerical illustrations

In the following sections, we illustrate IMSIA on two simulated MSPs with different topology, and on one real-world MSP involving semiconductor manufacturing.

2.5.1 Artificial MSPs

Experimental protocol

We consider two artificial MSPs on 4 stages ($Q = \{1, 2, 3, 4\}$), each of which with two OSs ($S = \{1, 2, \ldots 8\}$, $S_b = \{2b - 1, 2b\}$, $b = 1, 2, 3, 4$). The first artificial MSP is defined by a sequential technical-relationship network (Figure 2.2), that is one such that

$$X_b = \begin{cases} \{Z_{b-1}, V_{2b-1}, V_{2b}\} & \text{if } b > 1 \\ \{V_1, V_2\} & \text{otherwise} \end{cases}.$$

For this MSP, each observation i of a total of n is generated via the following true model (unknown to IMSIA):

$$Z_{b_i} = \begin{cases} Z'_{b-1_i} - 0.5V_{2b-1_i} - 0.05V_{2b_i} + \epsilon_{b_i} & \text{if } b > 1 \\ -0.5V_{1_i} - 0.05V_{2_i} + \epsilon_{1_i} & \text{otherwise} \end{cases},$$

where $(V_{1_i}, V_{2_i}, \ldots, V_{8_i})$ and $(\epsilon_{1_i}, \epsilon_{2_i}, \epsilon_{3_i}, \epsilon_{4_i})$ are a random sample from $(V_1, V_2, \ldots, V_8) \sim N(\mathbf{0}, \Sigma_V)$ and $(\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4) \sim N(\mathbf{0}, \Sigma_\epsilon)$, respectively, with

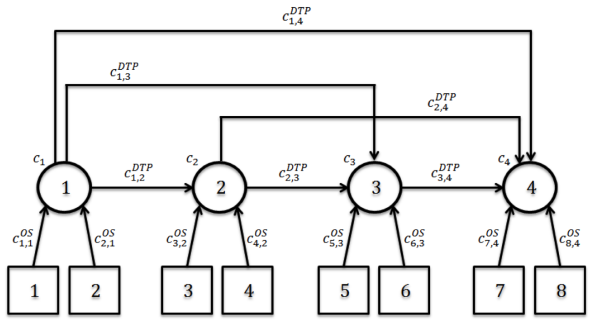

Figure 2.3: Labeled network for the fully-connected artificial MSP.

$$\sigma_{V_{j,k}} = \begin{cases} 1 & \text{if } j = k \\ 0.7 & \text{if } (j,k) \in \{(1,2),(3,4),(5,6),(7,8)\} \\ 0 & \text{otherwise} \end{cases},$$

representing the (j,k) element of Σ_V, $\Sigma_\epsilon = diag(0.1485, 0.6662, 0.6662, 0.6662)$, and Z'_{b-1_i} being the same as Z_{b-1_i} after standardization (subtraction of the mean and division by the standard deviation of the whole sample of n observations). The values of the diagonal elements of Σ_ϵ are calibrated in such a way that approximately 34% of the variance of Y_b is explained by ϵ_b, $b = 1, 2, 3, 4$.

The second artificial MSP is defined by a fully-connected technical-relationship network (Figure 2.3), that is one such that:

$$X_b = \begin{cases} \{\{Z_a : a < b\}, V_{2b-1}, V_{2b}\} & \text{if } b > 1 \\ \{V_1, V_2\} & \text{otherwise} \end{cases}.$$

For this MSP, each observation i of a total of n is generated via the following true model (unknown to the method):

$$Z_{b_i} = \begin{cases} \sum_{a<b} Z'_{a_i} - 0.5V_{2b-1_i} - 0.05V_{2b_i} + \epsilon_{b_i} & \text{if } b > 1 \\ -0.5V_{1_i} - 0.05V_{2_i} + \epsilon_{1_i} & \text{otherwise} \end{cases},$$

where $(\epsilon_{1_i}, \epsilon_{2_i}, \epsilon_{3_i}, \epsilon_{4_i})$ are a random sample from $(\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4) \sim N(\mathbf{0}, \Sigma_\epsilon)$, with $\Sigma_\epsilon = diag(0.1485, 0.6662, 1.9192, 3.9365)$, and the observations along the V and Z' variables are defined the same way as in the sequential artificial MSP. The values of the diagonal elements of Σ_ϵ are calibrated in such a way that approximately 34% of the variance of Y_b is explained by ϵ_b, $b = 1, 2, 3, 4$, as in the sequential artificial MSP.

RF and DBB-RF modeling was carried out via the `randomForest` function in the homonymous R package, whose underlying C code was modified in order to allow the specification of a variable bagging probability vector `prob` ($\boldsymbol{w}_{j,b}$, as per step E1 in IMSIA) as an extra parameter for DBB-RF. All parameter settings were left to their defaults in `randomForest`, except: parameter `mtry` (number of candidate variables for variable bagging), which was set to 2 for all DBB-RFs; parameter `prob`, which was calculated by means of function (2.19) with $u = 1.5$ and lower bound $\omega = 0.5/|X_b|$ (recalculated for every stage b). DBB-RF trees were pruned at depth D, calculated from formula (2.23), setting $c = 0.368$ and $m = 4$.

Results

Tables 2.1 and 2.2 report average numerical results obtained over 100 independent artificial data sets, each composed by $n = 250$ independent observations randomly generated according to the protocol described in section 2.5.1, for the sequential and the fully-connected MSP, respectively. Global contributions are calculated for the special case $e = 4$ (final stage) of formula (2.33).

At the local level, it may be noticed that within SUBs $d > 1$ of both MSPs, estimated contributions to Z_d (stage d) from V variables are significantly smaller than those from Z' variables. This result should not surprise, if one considers that the theoretical contribution of the variance of the V and Z' variables to the variance of Z_d, which of such variables is a (noisy) linear combination, is subject to a quadratic relationship with the variables' coefficients in the linear combination, as given by the true model. Note that in all SUBs d, the contribution of V_{2d} is correctly deemed to be substantially larger than that V_{2d-1},

despite the fairly high degree of correlation existing between the two variables.

At the global level, estimated contributions from stages $d < 4$ to the final stage along every individual DTR path are expected to decrease with the number of DTR arcs in the path, since such number corresponds to the number of recursive decompositions applied to direct contributions to the final stage. On the other hand, the cardinality of the path set $P_{d,4}$, representing the number of directions along which the contributing stage affects the final stage, is expected to counterbalance the former effect. In the sequential MSP, the recursive-decomposition effect is responsible for the pattern of decreasing measured importance for earlier stages, whereas the path-cardinality effect is not visible, since all stages are connected to the final stage via one only path. In the fully-connected MSP, earlier stages connect to the final stage in more ways than later ones, making the path-cardinality effect dominant with respect to the recursive-decomposition effect, causing the estimated contribution to increase for earlier stages.

Table 2.1: Local and global relative contribution assessments for every stage in the sequential artificial MSP (averages over 100 repeats).

Stage d	**Path set** $P_{d,4}$	**Local contributions to stage** d	**Global contribution** $C_{d,4}$ **to stage** 4
4	–	$c_4 = 0.424$ $c_{7,4}^{OS} = 0.068$ $c_{8,4}^{OS} = 0.010$ $c_{3,4}^{DTP} = 0.498$	$c_4 + c_{7,4}^{OS} + c_{8,4}^{OS} = 0.502$
3	$\left\{\{3,4\}\right\}$	$c_3 = 0.424$ $c_{5,3}^{OS} = 0.072$ $c_{6,3}^{OS} = 0.011$ $c_{2,3}^{DTP} = 0.493$	$c_{3,4}^{DTP}\left(c_3 + c_{5,3}^{OS} + c_{6,3}^{OS}\right) = 0.252$
2	$\left\{\{2,3,4\}\right\}$	$c_2 = 0.422$ $c_{3,2}^{OS} = 0.076$ $c_{4,2}^{OS} = 0.012$ $c_{1,2}^{DTP} = 0.491$	$c_{3,4}^{DTP} c_{2,3}^{DTP}\left(c_2 + c_{3,2}^{OS} + c_{4,2}^{OS}\right) = 0.125$
1	$\left\{\{1,2,3,4\}\right\}$	$c_1 = 0.410$ $c_{1,1}^{OS} = 0.572$ $c_{2,1}^{OS} = 0.019$	$c_{3,4}^{DTP} c_{2,3}^{DTP} c_{1,2}^{DTP}\left(c_1 + c_{1,1}^{OS} + c_{2,1}^{OS}\right) = 0.121$

Table 2.2: Local and global relative contribution assessments for every stage in the fully-connected artificial MSP (averages over 100 repeats).

Stage d	**Path set** $P_{d,4}$	**Local contributions to stage** d	**Global contribution** $C_{d,4}$ **to stage** 4
4	–	$c_4 = 0.388$ $c_{7,4}^{OS} = 0.034$ $c_{8,4}^{OS} = 0.015$ $c_{1,4}^{DTP} = 0.186$ $c_{2,4}^{DTP} = 0.195$ $c_{3,4}^{DTP} = 0.181$	$c_4 + c_{7,4}^{OS} + c_{8,4}^{OS} = 0.437$
3	$\Big\{\{3,4\}\Big\}$	$c_3 = 0.399$ $c_{5,3}^{OS} = 0.056$ $c_{6,3}^{OS} = 0.018$ $c_{1,3}^{DTP} = 0.266$ $c_{2,3}^{DTP} = 0.262$	$c_{3,4}^{DTP}\left(c_3 + c_{5,3}^{OS} + c_{6,3}^{OS}\right) = 0.086$
2	$\Big\{\{2,4\},\{2,3,4\}\Big\}$	$c_2 = 0.422$ $c_{3,2}^{OS} = 0.076$ $c_{4,2}^{OS} = 0.012$ $c_{1,2}^{DTP} = 0.491$	$\left(c_{2,4}^{DTP} + c_{2,3}^{DTP} c_{3,4}^{DTP}\right)\left(c_2 + c_{3,2}^{OS} + \right.$ $\left. + c_{4,2}^{OS}\right) = 0.124$
1	$\Big\{\{1,4\},\{1,2,4\},$ $\{1,3,4\},\{1,2,3,4\}\Big\}$	$c_1 = 0.410$ $c_{1,1}^{OS} = 0.572$ $c_{2,1}^{OS} = 0.019$	$\left(c_{1,4}^{DTP} + c_{2,4}^{DTP} c_{1,2}^{DTP} + c_{3,4}^{DTP} c_{1,3}^{DTP} + \right.$ $\left. + c_{3,4}^{DTP} c_{2,3}^{DTP} c_{1,2}^{DTP}\right) \cdot$ $\cdot \left(c_1 + c_{1,1}^{OS} + c_{2,1}^{OS}\right) = 0.354$

2.5.2 Case study: a real-world multi-stage semiconductor manufacturing process

In semiconductor manufacturing, products are fabricated on wafers, and the manufacture of wafers typically involves tens or hundreds of stages. These stages can be, moreover, divided into sub-modules, each involving a certain number of processes. The performance characteristics of the end product are the result of the combined impact of the output of each sub-module on the output of all of the subsequent sub-modules. Among other important factors, this impact involves variability, which is transferred and/or accumulated from one sub-module to another and from one stage to another, and thus to the final product.

From a quality-control point of view, the identification of the stages that incorporate the main sources of variability in a MSP is of substantial importance. These stages are often the target of process-improvement and production-planning investments, typically aimed at making the outputs of such stages more stable, and, consequently, making the characteristics of the final product more predictable. One possible approach towards the selection of stages that most require intervention actions consists of identifying stages that

most contribute to the characteristics of the final product, as measured by the output of the final stage in the MSP, and their variability. These contributions are a direct result of the interplay between direct and indirect relationships among stages, and it is therefore key that their measurements be defined based on the overall structure of such relationships, in such a way that contributions inherent to a certain stage can be correctly separated from contributions from its preceding stages. This is, in fact, the basic criterion that IMSIA utilizes to construct its results.

We discuss here the application of IMSIA to a MSP aimed at the manufacturing of wafers via multi-patterning photolithograpy, a technology that has been widely adopted by leading companies in the semiconductor micro-fabrication industry. The specific photolithography MSP in our application is composed of 33 stages involving such actions as chemical vapor disposition, photomask patterning, and etching, among others. A specific critical dimension of the processed product, such as the thickness of the remaining wafer mask, was defined as the output variable for each stage in the MSP. DTPs were identified based on expert knowledge about precedence relationships existing between steps across the fabrication process.

Analysis and IMSIA setup

A data set of $n = 182$ observations over the set $Q = \{1, 2, \ldots, 33\}$ of stage output variables was made available to us for analysis. This data set can be conceptually divided in independent 14 batches of 13 independent observations, with each batch representing a different unit of the same of semiconductor wafer, and each observation within a given batch representing a chip in a different location on the wafer. No information about OS variables was given in the data.

Stages 31, 32, and 33 correspond to three distinct final stages, each of which inducing a slightly different technical relationship network. Analysis and results we present here specifically focus on the MSP network induced by final stage 33, unless differently specified below. Table 2.3 provides a high-level overview of the topology of this network by listing

number and length of all existing paths connecting every stage to the respective final stage, as well as the number of DTPs for every stage. Further details are not disclosed to preserve the confidentiality of the data.

Our investigation revealed that, for almost all SUBs b, factors related to wafer-batch and chip-location information may have played an important role in explaining the variability of output Z_b, in that the RF b's prediction performance on the OOB observations significantly improved when such information was used to predict Z_b in conjunction with the variables in Q_b. Three OSs were therefore assumed to exist for each SUB b, in such a way that $S_b = \{3b-2, 3b-1, 3b\}$, $b = 1, 2, \ldots, 33$. The first OS in S_b is related to wafer batch, whereas the other two are related to chip location (Figure 2.4). One categorical variable, given by the wafer-batch index, and two numerical variables, given by the two-dimensional Cartesian coordinates of the chip's center with respect to the the wafer's center, were used as proxies of the three unknown OSs, respectively. The same three proxies were provided as predictors to the RF of every SUB, letting the regression model learn from the data the appropriate transformation that maps each proxy to the corresponding unknown OS variable specific to that SUB.

The parameters settings chosen for IMSIA were the same as those used for the artificial MSPs, except for: (I) `ntree` (number of trees in the RFs/DBB-RFs), which was increased to 1000 from the default of 500; (II) `prob`, which was computed via (2.21), with p_v extracted from an asymptotic, quadratic-type re-randomization test [37, 38], by invoking function `independence_test` with `teststat='quadtype'` in the `coin` R package; (III) u, which was set to 1/10.

Results

Table 2.4 shows that, overall, the OOB performance of the local RF models is reasonably good both for SUBs with no DTPs (median $R^2_{RF,b:|Z_b|=0} = 0.618$) and SUBs with one or more DTPs (median $R^2_{RF,b:|Z_b|>0} = 0.665$), with a substantial boost provided by the inclusion of wafer-batch and chip-location OS variables as predictors. The only exceptions

Table 2.3: Number and length of paths from every stage to stage 33.

Stage d	**Number of paths to stage 33** $n_{d,33}$	**Path lengths** $p^1_{d,33}, p^2_{d,33}, \ldots, p^{n_{d,33}}_{d,e}$	**Number of DTPs** $\lvert Z_d \rvert$
1	1	1	0
2	1	1	0
3	9	2, 3, 3, 5, 5, 5, 5, 5, 5	0
4	9	2, 3, 3, 5, 5, 5, 5, 5, 5	0
5	11	2, 3, 3, 3, 4, 5, 5, 5, 5, 5, 5	0
6	8	3, 3, 5, 5, 5, 5, 5, 5	0
7	14	3, 3, 4, 4, 5, 5, 5, 5, 5, 5, 6, 6, 6, 6	0
8	6	4, 4, 6, 6, 6, 6	0
9	6	4, 4, 6, 6, 6, 6	0
10	6	4, 4, 6, 6, 6, 6	0
11	6	4, 4, 6, 6, 6, 6	0
12	2	3, 5	0
13	2	3, 5	5
14	5	3, 3, 5, 5, 5	5
15	5	3, 3, 5, 5, 5	0
16	3	3, 5, 5	0
17	2	2, 4	9
18	1	4	8
19	2	2, 4	8
20	2	2, 3	1
21	1	3	3
22	1	2	1
23	2	3, 3	0
24	1	3	0
25	0	-	1
26	1	2	2
27	1	2	2
28	1	2	9
29	0	-	9
30	1	1	10
31	0	-	3
32	0	-	3
33	0	-	3

are SUBs 11 and 16, the variability of whose respective outputs, besides not being affected by any DTP variable (since $|Z_{11}| = |Z_{16}| = 0$), does not seem to be driven by wafer-batch- or chip-location-related components.

The three MSP networks showed similar relative-contribution patterns across stages. All stages with significant relative contribution to final stage's output (say, above 5% of the overall relative contributions to such output), involved the later phases of the semiconductor fabrication, focused on wafer etching. Detailed analysis of the results suggested that the recursive-decomposition effect (see section 2.5.1) was dominant with respect to the path-

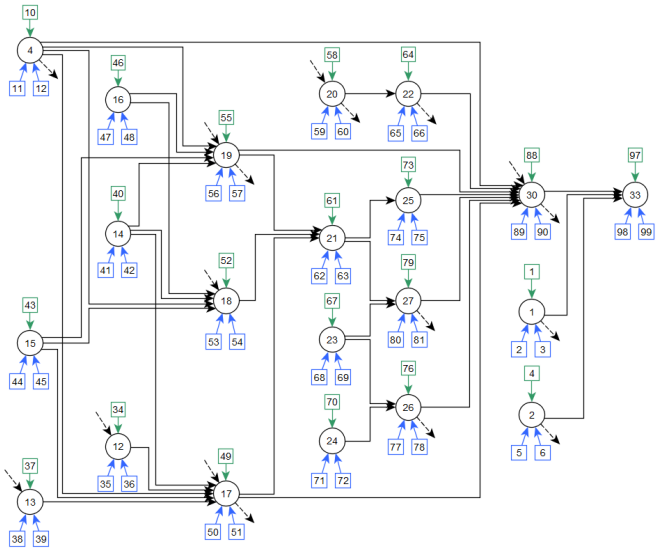

Figure 2.4: Representation of a portion of the MSP network induced by final stage 33, including a subset of 21 out of the 33 stages, along with all DTRs existing within such subset. Dashed arrows represent DTRs between a stage in this subset and one or more stages in the complement of this subset (not shown to preserve the confidentiality of the data). Circular nodes represent stages; square nodes represent OSs, with green ones corresponding to wafer-batch-related OSs and blue ones corresponding to chip-location-related OSs.

cardinality effect in IMSIA's determination of the relative contribution values obtained here; this was somewhat expected, considering the sparsity of the networks (for all of them, the number of stage-connecting arcs were less than 2% the theoretical fully-connected maximum). Significantly contributing stages were typically located on a single short path, corresponding to the terminal portion of a longer path. Specifically for the network induced by final stage 33, if we define $SC_{33} = \{d : d \in Q, C_{d,e} > 0.05\}$ as the set of stages significantly contributing to the output of stage 33, we have $SC_{33} = \{22, 25, 26, 30, 33\}$, connected via paths 22–30-33, 25–30–33, and 26–30-33. Not surprisingly, we observed that for the first stage on all such paths, most of the local relative contributions (see section 2.4.1) are self-contributions (see (2.31)) from wafer-batch- and chip-location-related OS variables, and other uncontrollable factors; for the remaining stages on the path, their respective local relative contributions are a balance of both self-contributions and the contribution they receive from the stage that precedes them on the path.

Table 2.4: OOB RF performance for every SUB, without and with the inclusion of wafer-batch and location OS variables as predictors. Dashes correspond to SUBs b with no DTPs, for which r_b^2 cannot be calculated prior to the inclusion of the OS variables as predictors.

SUB b	r_b^2 **prior to inclusion of OS variables**	r_b^2 **after inclusion of OS variables**
1	–	0.527
2	–	0.619
3	–	0.694
4	–	0.605
5	–	0.412
6	–	0.420
7	–	0.723
8	–	0.620
9	–	0.432
10	–	0.806
11	–	0.031
12	~ 0	0.369
13	0.522	0.623
14	–	0.537
15	–	0.782
16	–	0.080
17	0.264	0.376
18	0.079	0.297
19	0.716	0.846
20	~ 0	0.723
21	0.160	0.382
22	~ 0	0.787
23	–	0.551
24	–	0.585
25	0.273	0.406
26	0.011	0.678
27	0.045	0.539
28	0.596	0.628
29	0.419	0.543
30	0.621	0.643
31	0.719	0.834
32	0.640	0.798
33	0.873	0.907

Table 2.5: Global relative contribution assessments (second column) for every stage in the real-world MSP toward final stage 33 (values multiplied by 1000). Dashed entries correspond to nodes not included in the subnetwork of Figure 2.4.

Stage d	**Global contribution** $C_{d,33}$	**Wafer-lot-OS contribution** $c^{OS}_{3d-2}C^{DTP}_{d,33}$	**Chip-location-OS contribution** $(c^{OS}_{3d-1} + c^{OS}_{3d})C^{DTP}_{d,33}$	**Non-OS contribution** $c_d C^{DTP}_{d,33}$
1	19.479	5.887	4.375	9.217
2	7.229	2.949	1.523	2.757
3	–	–	–	–
4	14.444	7.880	0.853	5.711
5	–	–	–	–
6	–	–	–	–
7	–	–	–	–
8	–	–	–	–
9	–	–	–	–
10	–	–	–	–
11	–	–	–	–
12	0.987	0.217	0.069	0.701
13	0.015	0.003	0.001	0.011
14	0.348	0.003	0.183	0.161
15	0.540	0.191	0.231	0.118
16	0.197	0.015	0.001	0.181
17	4.069	0.407	0.152	3.509
18	1.993	0.161	0.241	1.591
19	9.226	5.144	0.267	3.815
20	23.037	16.263	<0.001	6.774
21	29.709	4.258	4.824	20.628
22	74.348	27.494	27.317	19.537
23	8.591	2.685	2.050	3.857
24	6.637	2.108	1.775	2.755
25	103.127	1.757	20.571	80.799
26	116.701	24.025	51.380	41.296
27	6.024	1.779	1.086	3.159
28	–	–	–	–
29	–	–	–	–
30	386.547	24.440	66.067	296.040
31	–	–	–	–
32	–	–	–	–
33	143.442	25.577	24.480	93.385

Chapter 3

Dependence-biased clustering for variable selection

3.1 Introduction

Variable selection is a well-known problem in supervised learning [39, 40, 41]. For a given data set $(\boldsymbol{X}, \boldsymbol{y})$, with $\boldsymbol{X} \in \mathbb{R}^{nxp}$ containing the values of n observations along input variables $X_1, X_2, \ldots, X_p$, and $\boldsymbol{y} \in \mathbb{R}^n$ containing the values of the same observations along output variable Y, variable selection is the search of a subset W of $X = \{X_1, X_2, \ldots, X_p\}$ that satisfies certain properties, which, in quite general terms, depend on the "relevance" that the variables in W have for Y and, possibly, also on the "redundancy" that the variables in W have toward each other [39, 32]. Relevance may be expressed, for example, as a function of predictability, statistical association, or even causal dependence between Y and the variables in W. Redundancy, on the other hand, refers to the degree of similarity between variables in W, as measured, for example, by linear correlation or mutual information. Depending on the application, the uses of variable selection may be fairly diverse, but they usually boil down to the construction of simpler supervised models that are less prone to data overfitting, quicker to train and deploy, and more straightforward to analyze and interpret, especially when p is very large.

In this chapter, we extend the work we presented in chapter 2 by introducing a new RF-based method aimed at the selection of a small set of strong, non-redundant predictors. This method relies on a novel CPI measure, which, unlike DBB-CPI (section 2.3.2), carries out the data partitioning task separately from that of model training. This is done by replacing the DBB-RF with a clustering procedure, targeted at finding optimal values of $W^{(j)}$ to condition upon when assessing the importance of X_j. The key feature of this clustering approach lies in its definition of optimality, which is formulated to reflect the structure of dependencies existing between the variables in $W^{(j)}$ and X_j itself.

This chapter is organized as follows. Section 3.1.1 provides relevant background on RF-based methods for variable selection. Section 3.2 presents our novel clustering-based CPI measure as the main theoretical contribution of this work, and explains how we incorporate it in a recursive backward-elimination algorithm to solve the underlying variable selection problem. Section 3.3 finally discusses the performance of our algorithm on several artificial and real-world data sets, comparing results with a variety of state-of-the-art RF-based methods for variable selection.

3.1.1 Background: Variable Selection with Random Forests

The problem of variable selection has been widely studied within the RF framework. Most methods appeared in the literature so far build on a PI measure analogous to that described in section 2.2.2, but differ in the approach they employ to solve the variable selection problem. There are essentially three main classes of approaches, aimed at either finding a variable subset that yields optimal prediction performance, or a compact, small-cardinality variable subset with no redundant variables, or a variable subset that contains all predictors whose contributions to the output variable are statistically significant. Below we briefly summarize the basic elements that characterize each of these approaches.

The performance-based approach is primarily geared towards the elimination of variables that are detrimental to RF prediction accuracy, with the goal of training a RF with the best possible testing performance. Methods that employ this approach usually start by calculating PI on all input variables and then iteratively reject a certain fraction of them, cho-

sen from those with the least measured importance. At every iteration, model performance is assessed via OOB or cross-validation error, and the variable subset that yields the best model is finally returned as solution. Although generally not explicitly targeted to the elimination of redundant variables, some of these methods do return small-cardinality solutions, thus showing some overlap with the compactness-based approach [42, 43, 44, 45].

The compactness-based approach seeks a variable subset with small or, ideally, minimal cardinality, by excluding variables that are not essential to the model's predictive power, because they are either irrelevant to the output variable or redundant with variables that are relevant. The inclusion of subset-cardinality minimization in the variable selection criterion results in simpler, more interpretable RF models that are somewhat less prone to data overfitting. Some methods that follow this approach rely on regularization penalties for unimportant variables enforced within the RF training algorithm [46, 47], whereas others complement the RF with tools for the detection of masking relationships among variables [48].

The significance-based approach aims at identifying all input variables that have a relationship with the output variable, without any regard to the redundancies that may exist among those input variables. This approach may be desirable when the underlying analytic goal is that of developing broad intuition on the process that is generating the data. Most methods that use these approach start by measuring the PI of all input variables; they then perform repeated permutations of either the output variable or the input variables, and after each repetition they measure PI again. These repeated measurements are used to build an empirical distribution of PI under a null hypothesis of independence between input variables and output. This distribution is employed to calculate a p-value or a Z-score on the PI measurements obtained prior to the repeated permutations, and all variables with statistically significant PI are finally selected [49, 50, 51, 52].

3.2 A new method for variable selection with conditional permutation importance

We propose a new method for variable selection with CPI, placed at the intersection between the performance-based and the compactness-based approach described in section 3.1.1. The method consists of a RF-based backward-elimination algorithm [53, 54, 55], targeted at the selection of a minimal subset of input variables characterized by high predictive power and low reciprocal redundancy. The core of our method is a novel CPI measure, which relies on a clustering approach for optimally partitioning the $W^{(j)}$ space while leveraging the structure of dependencies existing among input variables.

Broadly speaking, clustering is an optimization problem concerned with dividing observations in a data set into groups, in such a way that observations within each group be more homogeneous than those belonging to different groups [32]. As such, this grouping task is closely related to the goal of devising a partition $S^{(j)}$ of the $W^{(j)}$ space, composed of sets with homogeneous values (see section 2.2.2). Now, let us suppose we can formulate the clustering problem in such a way that, the stronger the dependence between X_k and X_j, $X_k \in W^{(j)}$, the more any two observations are considered dishomogeneous along X_k. Intuitively, with such a formulation, a good clustering solution will be one where within-group homogeneity holds specifically along variables X_k with more substantial dependence on X_j. As mentioned at the end of section 2.2.2, those are indeed the variables whose homogeneity is most critical to the correct assessment of X_j's conditional importance. The use of a clustering formulation of this kind as a partitioning engine for the $W^{(j)}$ space is the basic idea behind the CPI measure we propose. We name this measure "Dependence-Biased-Clustering Conditional Permutation Importance" (DBC-CPI).

3.2.1 Dependence-biased-clustering conditional permutation importance

Let Ω be a function as in section 2.3.1, and let $(\boldsymbol{X}, \boldsymbol{y})$ be a data set as defined in section 3.1, and let us assume that $\boldsymbol{X}$ is numeric and standardized to zero mean and unit variance. Let $\boldsymbol{w}_j$, $j \in \{1, 2, \ldots, p\}$ be a p-dimensional vector defined as in section 2.3.1 after setting

$\omega = 0$. After letting $\boldsymbol{M}_j = diag(\boldsymbol{w}_j)$, let us define a $\boldsymbol{w}_j$-biased Euclidean distance function between any two p-dimensional observations $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$ in $\boldsymbol{X}$:

$$d_j(\boldsymbol{x}_a, \boldsymbol{x}_b) = \sqrt{(\boldsymbol{x}_a - \boldsymbol{x}_b)^T \boldsymbol{M}_j^2 (\boldsymbol{x}_a - \boldsymbol{x}_b)}. \tag{3.1}$$

For (3.1) we can straightforwardly derive the following.

Proposition 3.1. *d_j is a metric.*

Proof. We can rewrite d_j as follows:

$$\begin{aligned} d_j(\boldsymbol{x}_a, \boldsymbol{x}_b) &= \sqrt{\sum_{k=1}^{p} w_{kj}^2 (x_{ka} - x_{kb})^2} \\ &= \sqrt{\sum_{k=1}^{p} (w_{kj} x_{ka} - w_{kj} x_{kb})^2} \\ &= \sqrt{\left(\boldsymbol{x}_a^{[j]} - \boldsymbol{x}_b^{[j]}\right)^T \left(\boldsymbol{x}_a^{[j]} - \boldsymbol{x}_b^{[j]}\right)}, \end{aligned} \tag{3.2}$$

where $\boldsymbol{x}_a^{[j]} = \boldsymbol{M}_j \boldsymbol{x}_a = (w_{1j} x_{1a}, w_{2j} x_{2a}, \ldots, w_{pj} x_{pa})^T$ and $\boldsymbol{x}_b^{[j]} = \boldsymbol{M}_j \boldsymbol{x}_b = (w_{1j} x_{1b}, w_{2j} x_{2b}, \ldots, w_{pj} x_{pb})^T$. Equation (3.2) shows that d_j is equivalent to the standard (unbiased) Euclidean distance metric on a linear transformation of the original p-dimensional variable space, induced by scaling/projection matrix $\boldsymbol{M}_j$, with scaling bias w_{kj} along variables X_k, $k \neq j$, and projection of X_j to 0, given $w_{jj} = 0$.

□

In summation, d_j is defined in such a way that the more strongly X_k is dependent on X_j, $k \neq j$, the farther $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$ (or, equivalently, $\boldsymbol{x}_a^{[j]}$ and $\boldsymbol{x}_b^{[j]}$) are considered to be along X_k; d_j, on the other hand, is not affected by the values that $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$ have along X_j.

Now, let us consider a RF with T trees, trained on $(\boldsymbol{X}, \boldsymbol{y})$, and with cn OOB obser-

vations per tree, $0 < cn < n$. As above, let O_t be the OOB data for the t-th tree. Let Δ be a function that, given a set of grouped observations, somewhat quantifies how heterogeneous observations are within each group, overall. Given a choice of Ω, Δ, and integer K, $1 \leq K \leq cn$, the procedure to calculate DBC-CPI for each variable X_j, $j \in \{1, 2, \dots p\}$ is as follows:

A. Compute $\boldsymbol{w}_j$ from $\boldsymbol{X}$ via Ω, and accordingly define d_j.

B. Cluster $\boldsymbol{X}$ into K groups, by minimizing Δ and using d_j as a measure of within-group heterogeneity.

C. For $t = 1, 2, \dots, T$, partition the observations in O_t according to the group assignments of the corresponding observations in $\boldsymbol{X}$. Let $S_t^{(j)DBC}$ be this partition.

D. Compute and return

$$I_j^{DBC-CP} = \frac{1}{T} \sum_{t=1}^{T} \left(E_{t,\pi,S_t^{(j)DBC}} - E_t \right), \tag{3.3}$$

where $E_{\cdot}$ is a mean square error (as in 2.17) if Y is numeric or a misclassification error if Y is categorical. Regarding step B, we note that: (I) minimizing within-group heterogeneity is equivalent to maximizing within-group "compactness", which is a common criterion to measure cluster validity [56]; (II) the standardization of $\boldsymbol{X}$ ensures that variable scale does not affect within-group heterogeneity. Moreover, we note that $S_t^{(j)DBC}$ in (3.3) plays the same role as $S_t^{(j)}$ in (2.17).

We propose a choice of Δ derived from the K-means formulation of the clustering problem [32], which seeks for the partition into K groups that minimizes

$$\Delta = \sum_{l=1}^{K} \sum_{i \in s_l} d^2 \left(\boldsymbol{x}_i, \boldsymbol{\mu}_l \right), \tag{3.4}$$

where s_l is the l-th group, $d(\cdot, \cdot)$ is the Euclidean distance, and $\boldsymbol{\mu}_l$ is the center of the l-th group, given by the mean of the observations it contains. In this work, however, we replace d with d_j, thus changing objective function (3.4) to the following:

$$\begin{aligned}\Delta_j &= \sum_{l=1}^{K} \sum_{i \in s_l^{(j)}} d_j^2 \left(\boldsymbol{x}_i, \boldsymbol{\mu}_l^{(j)} \right) \\ &= \sum_{l=1}^{K} \sum_{i \in s_l^{(j)}} \left(\boldsymbol{x}_i - \boldsymbol{\mu}_l^{(j)} \right)^T \boldsymbol{M}_j^2 \left(\boldsymbol{x}_i - \boldsymbol{\mu}_l^{(j)} \right)\end{aligned} \tag{3.5}$$

where $\boldsymbol{\mu}_l^{(j)}$ is the mean of group $s_l^{(j)}$. Now, if we let n_l be the cardinality of group $s_l^{(j)}$ and $v_{X_k|s_l^{(j)}}$ be the sample variance of variable X_k computed on the observations in group $s_l^{(j)}$, we may rewrite (3.5) as follows:

$$\Delta_j = \sum_{l=1}^{K} n_l \sum_{k=1}^{p} w_{kj}^2 v_{X_k|s_l^{(j)}} \tag{3.6}$$

Equation (3.6) shows that using (3.5) as an objective function pushes step B of DBC-CPI towards finding groups that are more homogeneous along the most critical conditioning variables (that is, those most dependent on X_j), since the variance of such variables is associated to larger bias and has therefore a larger impact on the objective value. As a concluding remark, we note that, given the equivalence of (3.1) and (3.2), we can look for an optimum of (3.5) by deploying a standard algorithm for solving the K-means problem (relying, that is, on the standard Euclidean distance) on the transformed data set $\boldsymbol{X}\boldsymbol{M}_j$ instead of $\boldsymbol{X}$.

A DBC-CPI Variant for Categorical Variables

If $\boldsymbol{X}$ contains any categorical variables, observations cannot be grouped based on their Euclidean distance, and group means are not defined. For this type of data, our biased Euclidean distance may be replaced, for example, by Gower's coefficient of dissimilarity [57], which defines the distance between two observations $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$ as follows:

$$d(\boldsymbol{x}_a, \boldsymbol{x}_b) = \frac{\sum_{k=1}^{p} \zeta_k \delta'_{abk} \delta''_{abk}}{\sum_{k=1}^{p} \zeta_k \delta'_{abk}} \tag{3.7}$$

In (3.7), ζ_k is a bias associated to X_j; δ'_{kab} is equal to 0 if X_j is asymmetric binary and x_{ka} and x_{kb} are both zero, and equal to 1 otherwise; for categorical X_k, δ''_{kab} is equal to 0 if $x_{ka} = x_{kb}$, and equal to 1 otherwise; for numeric X_k, δ''_{kab} is equal to $|x_{ka} - x_{kb}| / R_k$, where R_k is the range of X_k.

By setting $\zeta_k = w_{jk}$, (3.7) may be used as a replacement for (3.1) in objective function Δ_j; note that the scale of any numeric variables in $\boldsymbol{X}$ will not affect the value of such a Δ_j. Observations in $\boldsymbol{X}$ may then be clustered via a K-medoids formulation, which is an analog of K-means where group centers are defined by the medoid of the observations they contain, rather than their mean [32].

3.2.2 Variable selection with DBC-CPI

From a variable importance standpoint, each variable in $X_j \in X$ can be categorized into one of three types, based both on the relationship that X_j has with Y and the relationship it has with the other variables in X: (I) irrelevant for Y, (II) relevant for Y but redundant with (dependent on) some other variables in X, and (III) relevant for Y and non-redundant with all other variables in X. Analogously to standard CPI, we expect DBC-CPI to assign lower important values to variables of type I and II, and higher importance values to variables of type III. Now, if the goal of the variable-selection task is that of choosing a minimal-size, minimally-redundant subset of variables with maximal predictive power, it is obvious that such subset should not include variables of type I; the same does not necessarily hold for variables of type II, since excluding all of them may result in the loss of useful predictors, that is ones that would become of type III if one or more other predictors of type II were eliminated. Consequently, the variable selection criterion cannot rely simply on the ranking of the p variables based on their DBC-CPI values. For such a reason, we propose a recursively-defined criterion, in which DBC-CPI is first calculated on the entire variable set and used to eliminate the least important variable; subsequently, DBC-CPI is recalculated on the $p-1$ remaining variables and the procedure is repeated until one only variable is left. We expect each of these recursions to eliminate one variable of either type I or type II; in the latter case, the elimination will, on average, cause an increase of the relative DBC-CPI

importance of one or more of the remaining variables in the following recursion, since the elimination will rule out one source of redundancy from the current variable subset.

Variable selection algorithm with recursively-calculated DBC-CPI

Given a data set $(\boldsymbol{X}, \boldsymbol{y})$ defined as above, and hyper-parameters Ω, Δ, and K for DBC-CPI, let us let $Q_p = X$, and let $\delta \geq 0$ be an extra hyper-parameter. Our DBC-CPI-based variable selection algorithm consists of the following steps:

A. For $o = 0, 1, \ldots, p-1$:

   A1. Calculate the $(p-o)$-dimensional vector $\boldsymbol{i}_o = \left(I_{j_1}^{DBC-CP}, I_{j_2}^{DBC-CP}, \ldots, \ I_{j_{p-o}}^{DBC-CP}\right)$ of DBC-CPI values, where $J_{p-o} = \{j_1, j_2, \ldots, j_{p-o}\} \subseteq \{1, 2, \ldots, p\}$ is the set of indices pertaining to the variables in Q_{p-o}.

   A2. Train an RF model on the observations along input variables in Q_{p-o} and output variable Y. Let u_{p-0} be the mean OOB prediction error of such model and γ_{p-o} be the corresponding standard error.

   A3. Let $Q_{p-o-1} = \{Q_{p-o} \setminus X_k\}$, $k = \underset{j \in J_{p-o}}{\operatorname{argmin}} \ I_j^{DBC-CP}$.

B. Let $\tilde{o}^* = \underset{\substack{o \leq \tilde{o} \\ u_o \leq u_{\tilde{o}} + \delta\gamma_{\tilde{o}}}}{\min} \ o$, $\quad \tilde{o} = \underset{o \in \{0,1,\ldots,p-1\}}{\operatorname{argmin}} \ u_{p-o}$. Return $Q_{\tilde{o}^*}$.

We will refer to this algorithm as "DBC-RCPI", where the "R" stands for "recursive".

Remark 1. In step A2, given observed output y_i and corresponding OOB prediction $\hat{y}_i^{OOB}$ for observation i, we define OOB prediction error as

$$e_i^{OOB} = \left(\hat{y}_i^{OOB} - y_i\right)^2 \tag{3.8}$$

for regression and

$$e_i^{OOB} = \begin{cases} 1 & \text{if } \hat{y}_i^{OOB} \neq y_i \\ 0 & \text{otherwise} \end{cases} \tag{3.9}$$

for classification. Letting $\bar{e}^{OOB} = \frac{1}{n}\sum_{i=1}^{n} e_i^{OOB}$ be the mean error and s_e^{OOB} the standard deviation of the error, the standard error $s_{\bar{e}^{OOB}} = \frac{s_e^{OOB}}{\sqrt{n}}$ of $\bar{e}^{OOB}$ is given by

$$s_{\bar{e}^{OOB}} = \sqrt{\frac{\sum_{i=1}^{n}\left(e_i^{OOB} - \bar{e}^{OOB}\right)^2}{n(n-1)}} \tag{3.10}$$

for regression and

$$s_{\bar{e}^{OOB}} = \sqrt{\frac{n\bar{e}^{OOB}(1-\bar{e}^{OOB})}{n(n-1)}} \tag{3.11}$$

for classification (in the latter, s_e^{OOB} corresponds to the standard deviation of a Bernoulli distribution with probability $\bar{e}^{OOB}$).

Remark 2. The final solution returned in step B corresponds to the smallest variable subset $Q_{\tilde{o}^*}$ whose corresponding mean RF OOB performance is no more than δ standard errors worse than the overall best obtained in all recursions, as given by $Q_{\tilde{o}}$.

Remark 3. Although K should generally be fixed throughout all recursions, certain scenarios may justify adjusting K as recursions go by. For example:

- If $\boldsymbol{X}$ contains any variables with fewer than n distinct values, its $p-o$-dimensional projection (along the only variables in Q_{p-o}) in step A1 may contain fewer and fewer distinct observations for larger and larger o. If at the o-th recursion the number of distinct observations n_{d_o} is less than K, then obviously there will not exist any feasible clustering solutions. In that case, it is necessary to decrease K to a value no larger than n_{d_o} to proceed to the next recursion. Our approach to this issue is, however, slightly more conservative: if $n_{d_o} < 2K$, that is, there do not exist any feasible clustering solutions in which each of the K groups contains at least 2 distinct observations, we set $K = 1$, that is, we replace DBC-CPI with non-conditional PI for

the current and all subsequent recursions.

- If $\boldsymbol{X}$ is very high-dimensional, setting $K = 1$, that is, again, replacing DBC-CPI with non-conditional PI, for the first $o_{start} \leq p$ recursions, may be an effective strategy to save some of the overall computational effort. This approach would then provide DBC-RCPI with a warm-start solution, which DBC-RCPI would then refine for the remaining $p - o_{start}$ recursions. During the this warm-start phase, further computational boosts can be achieved by: (I) removing multiple input variables, rather than only one, at a time in step A3, as done in [45]; (II) calculating non-conditional PI only once in recursion $o = 0$, and setting $i_o = i_0$ in step A1 in the following recursions, as done in [43].

3.3 Experiments

We evaluated the performance of DBC-RCPI, and of a variety of benchmark methods, via 50 repeats of 10-fold, stratified cross-validation [58, 59]. For experiments on artificial data instances, the data set used in each repeat corresponds to a different independent sample from the same, given theoretical distribution. For experiments on real-world data instances, each repeat used the same data set.

For each repeat and cross-validation fold, we (I) deployed each method on the training subset of the data, (II) trained a RF on the variable subset of the training data selected by that method, and (III) measured the testing performance of the RF on the testing subset of the data. We calculated testing performance as the mean cardinality v^* of the selected variable subset and the corresponding mean testing prediction error e^*. We finally calculated the mean $\bar{v}^*$ and $\bar{e}^*$ of these two performance indicators across the 50 repeats, as well as the corresponding standard errors.

For a few real-world data sets with small n ($\sim 10^1$) and large p ($\sim 10^2 - 10^3$), we replaced 10-fold cross-validation with leave-one-out cross validation, and ran one only repeat, since multiple repeats would have yielded identical results, given our experimental protocol, as clarified in the next paragraph. For these data sets, the values of $\bar{v}^*$ and $\bar{e}^*$

coincide therefore with the values of $\mathfrak{v}^*$ and e^* calculated over the n folds of the single repeat of leave-one-out cross-validation. Since in this special one-repeat scenario the standard errors of $\bar{\mathfrak{v}}^*$ and $\bar{e}^*$ obviously cannot be calculated, we chose to replace them with the standard errors of $\mathfrak{v}^*$ and e^*, calculated over the n cross-validation folds, when reporting our results.

We used version 3.4.3 of the R language for all method implementations and experiment executions. For all methods, we trained the RFs with function `randomForest` of the `randomForest` package, using `ntree=1000` trees and default values for all other hyper-parameters (including `sample=TRUE`, resulting in a proportion of $c \approx (1 - 1/n)^n$ OOB observations per tree, with c thus approaching $1/e \approx 0.368$ as n increases), unless otherwise specified below. In order to rule out any random fluctuations across methods resulting exclusively from the stochastic training process of RFs, for a given training data set, we mapped each subset of input variables to a unique integer, which we then used as seed for R's random number generator immediately prior to model training. This made sure that RFs trained on the same data were identical. We generated stratified cross-validation splits with function `createFolds` of package `caret`, setting `k=10`.

Within DBC-RCPI, for data sets with fully numeric input variables: we set $\Omega(\boldsymbol{x}^{(j)}, \boldsymbol{x}^{(k)}) = |\rho(\boldsymbol{x}^{(j)}, \boldsymbol{x}^{(k)})|^u$, where $\rho(\boldsymbol{x}^{(j)}, \boldsymbol{x}^{(k)})$ is the sample Pearson's correlation coefficient of $\boldsymbol{x}^{(j)}$ and $\boldsymbol{x}^{(k)}$, and $u > 0$ is a hyper-parameter, which we set to 2, unless differently specified below; we carried out K-means clustering with the Hartigan and Wong algorithm [60], as implemented in the `kmeans` function of the `stats` package, called from `nstart=10` random initial solutions, run for a maximum of `iter.max=1000` iterations. For data sets with one or more categorical input variables: we set $\Omega(\boldsymbol{x}^{(j)}, \boldsymbol{x}^{(k)}) = 1 - p_v(\boldsymbol{x}^{(j)}, \boldsymbol{x}^{(k)})^u$, where $p_v(\boldsymbol{x}^{(j)}, \boldsymbol{x}^{(k)})$ is the p-value of an asymptotic, quadratic-type re-randomization test of independence [37, 38] on vectors $\boldsymbol{x}^{(j)}$ and $\boldsymbol{x}^{(k)}$ (obtained invoking the `independence_test` function with `teststat=quadtype` in the `coin` R package), and $u > 0$ is a hyper-parameter, which we set to 1/20; we carried out K-medoids clustering with the Partitioning Around Medoids algorithm [61], as implemented in the `pam` function of the `cluster` package, with dissimilarity matrix `diss` calculated via the `daisy` function of the same package, setting

`metric="gower"` and `bias=` $\tilde{\boldsymbol{w}}_j = \boldsymbol{w}_j / \boldsymbol{e}^T \boldsymbol{w}_j$, where $\boldsymbol{e}$ is a p-dimensional vector of ones (the use of $[0,1]$-normalized bias vector $\tilde{\boldsymbol{w}}_j$, rather than raw bias vector $\boldsymbol{w}_j$, seemed to allow the `pam` function to run numerically more stable on some data instances, although from a theoretical standpoint, $\tilde{\boldsymbol{w}}_j$ and $\boldsymbol{w}_j$ are equivalent). Finally, unless differently specified below, we set $K = \lfloor \lfloor cn \rfloor / m \rfloor$, with $m = 4$ (m is approximately the mean group cardinality if the $\lfloor cn \rfloor$ OOB observations are partitioned into K clusters), and $\delta = 0$. All of the above can be considered as the "default" hyper-parameter values of DBC-RCPI.

The first group of benchmark methods we utilized includes all methods tested in [50], that is: Hapfelmeier et al., without and with Bonferroni correction (Hap and HapB) [50], Altmann et al. (Alt) [49], Diaz et al. with 0- and 1-standard error rule (Diaz0 and Diaz1) [45], Jiang et al. with 0- and 1-standard error rule (Jiang0 and Jiang1) [44], Svetnik et al. (Svet) [43], and Genuer et al. with prediction and interpretation variants (GenP and GenI) [42]. Our implementation of these methods is based on the code provided in [50], with the main exception being our use of CART-based RFs, instead of Conditional Inference RFs, motivated by the computational impracticality of the latter (as implemented in package party) on large data sets, as also reported in other studies [35]. PI calculations were carried out via function `importance` with `scale=FALSE` in the `randomForest` package for all of these methods. For Hap, HapB, and Alt, we set `ntree=100`, as suggested by their respective authors; for all other methods we set `ntree=1000`, for consistency with DBC-RCPI. We set all remaining parameters of these methods as suggested in their respective articles.

The second group of benchmark methods includes: Kursa et al. (Boruta) [52], Deng et al. (GRRF) [46], and Deng (GRF) [47]. For Boruta we used the `Boruta` function in package `Boruta`, with default parameters, except `ntree`, which we set to 1000. For GRRF and GRF we used function `RRF` in the `RRF` package, setting `ntree=1000`. We left all parameters to their default values, except gamma, which defines the penalty associated to unimportant variables during RF training. For GRF we set `gamma=1`, as in [47]. For GRRF we set: on classification instances, `gamma=0.5`, corresponding to the medium value used in [46]; on regression instances, `gamma=0.9`, corresponding to the medium value between 0.8 (minimum value at which we observed the solution of GRRF starting to differ from that of GRF on

our data sets) and the theoretical maximum of 1. Finally, as a "control" method, we also included a variant of Diaz, which maintains all of the features of the method discussed in [45], except it relies on a standard form of CPI, analogous to that presented in [34], rather than on PI. For the calculation of CPI in this method, which we will refer to as Std-CPI, we used function `randomForest` of R package `extendedForest` [62]. For Std-CPI, we set: `maxLevel`, which controls the depth at which trees in the RF are pruned prior to calculation of CPI, to $\lfloor \log_2(cn/m) \rfloor$, with the same value of m as in DBC-RCPI (m here corresponds to the mean number of OOB observations per leaf in each pruned tree); `ntree=1000`; all other parameters to their default values.

3.3.1 Artificial data

We used numeric continuous data sets artificially generated from multiple linear true models, with polynomially-transformed input variables mapping to a Normally-perturbed continuous output variable. We focused in particular on polynomial models of degree 1, with form

$$Y = \beta_0 + \sum_{j=1}^{p} \beta_j X_j + \alpha\epsilon \tag{3.12}$$

and of degree 2 with interactions, with form

$$Y = \beta_0 + \sum_{j=1}^{p} \beta_j X_j + \sum_{j=1}^{p} \beta_{jj} \widetilde{X_j^2} + \sum_{j=1}^{p-1} \sum_{k>j}^{p} \beta_{jk} \widetilde{X_k X_j} + \alpha\epsilon \tag{3.13}$$

where $X_1, X_2, \ldots, X_p$ are jointly standard Normal random variables with correlation matrix $\boldsymbol{R}$, $\widetilde{X_j^2}$ and $\widetilde{X_k X_j}$ are the standardized squared random variable X_j^2 and the standardized product random variable $X_k X_j$, respectively, ϵ is a standard Normal random variable, independent of $X_1, X_2, \ldots, X_p$, and $\alpha, \beta_0, \beta_1, \ldots, \beta_p, \beta_{11}, \beta_{12}, \ldots, \beta_{pp} \in \mathbb{R}$.

The standardization of each term in (3.12) and (3.13) is employed to ensure that terms with the same β value have the same bias in the sum on the right hand side of the

equations, since their scale is identical. Parameter α is used to modulate the scale of the noise perturbations, and consequently the overall relative contribution that the set of input variables has towards Y, given by the coefficient of determination R^2.

As a baseline for our experiments, we generated $n = 100$ independent observations from a true model of form (3.12), almost identical to that employed in study III of [50], with $p = 20$, $\beta_0 = 0$, $\beta_1 = \beta_4 = \beta_7 = 3, \beta_2 = \beta_5 = \beta_8 = 2, \beta_3 = \beta_6 = \beta_9 = 1, \beta_{10} = \beta_{11} = \ldots = \beta_{20} = 0$, with correlation matrix $\boldsymbol{R}$ with $\rho_{X_j X_k}$, $j \neq k$, equal to $\rho = 0.9$ if either $j, k \in \{4, 5, 6\}$ or $j, k \in \{7, 8, 9, 10, 11\}$ or $j, k \in \{12, 13\}$ and equal to 0 otherwise, and α calibrated to yield $R^2 = 0.9$. Note that the 20 input variables can be separated in 5 blocks: the first block involving X_1, X_2, X_3, all causally relevant to the output variable and uncorrelated; the second block involving X_4, X_5, X_6, all causally relevant and mutually correlated; the third block involving $X_7, X_8, \ldots, X_{11}$, all correlated but with only the first three variables causally relevant; the fourth block involving X_{12}, X_{13}, both causally irrelevant and correlated; finally, the fifth block involving $X_{14}, X_{15}, \ldots, X_{20}$, all causally irrelevant and uncorrelated. In section 3.3.3 we will refer to this experimental setup, as well as its variants with different values of n, ρ, and R^2, as 20DLinear. Our experiments also included several extensions of 20DLinear:

- 20DLinearClass, a classification variant of 20DLinear used also in [50], with X_j, $j = 1, 2, \ldots, p$ defined as in 20DLinear, and Y following a Bernoulli distribution with success probability π defined as follows:

$$\pi = \frac{e^{\sum_{j=1}^{p} \beta_j X_j}}{1 + e^{\sum_{j=1}^{p} \beta_j X_j}}. \tag{3.14}$$

- 20DLinearMixed, identical to 20DLinear, except after generating the observations, the values of X_1, X_4, X_7, X_{11}, and X_{13} are discretized into univariate quartilic bins and the resulting discrete values are then treated as unordered categorical levels.

- 20DLinearDecreasCov, identical to 20DLinear, except within the first, second, and third block, the correlation between X_j, and X_k, $j \neq k$, is not constant but decreases with the distance between indices k and j, according to formula

$$\rho_{X_j X_k} = 0.9 - 0.2(|j-k| - 1). \tag{3.15}$$

- 11DLinear+Poly, corresponding to a model of type (3.13), with $R^2 = 0.9$, $p = 11$, $X_1, X_2, \ldots, X_{11}$ and respective β parameters defined as in 20DLinear, and second-degree terms $\widetilde{X_k X_j}$ with $\beta_{jk} = \sqrt{\beta_j \beta_k}$ (always defined since $\beta_j, \beta_k \geq 0$, $j, k = 1, 2, \ldots, 11$), for a total of $2p + p(p-1)/2 = 77$ input variables.
- 11DLinear+HiddenPoly, identical to 11DLinear+Poly, except the second-degree terms are completely hidden from the variable selection methods and the RF models (which therefore process observations only along $X_1, X_2, \ldots, X_{11}$, and Y).

In all these latter experimental setups, the number of generated independent observations was $n = 100$ as for 20DLinear, with the exception of 11DLinear+Poly and 11DLinear+HiddenPoly, where we increased n to 300.

3.3.2 Real-world data

We experimented with real-world data sets from several diverse fields, including, among others, neuroimaging, evolutionary design of experiments, DNA microarrays, and meteorology. Details about each of these data sets are provided below.

fMRI brain-network data

This data set was kindly provided by the Integrated Brain Imaging Center of the University of Washington, during a fruitful collaboration that allowed us to acquire all of the neuro-scientific and technical background necessary for the analysis described below. The data refers to a sample of subjects from the Seattle Longitudinal Study (SLS), a cohort-sequential longitudinal study started in 1956 [63]. The SLS consists of periodical cognitive and behavioral assessments obtained with a battery of test underwent by all available prior participants, plus a new random sample added every 7 years. Out of the 600 currently active participants, 200 individuals with 3 midlife assessments (over a 14-year period) were

selected for a further longitudinal neuroimaging study. These subjects were classified as "decliners", "stable" or "gainers" based on their performance in tests that examined memory, executive function, and psychomotor speed. The brain activity at rest of these subjects was investigated via functional magnetic resonance imaging (fMRI), producing for each subject a set of time series of the so-called "blood oxygenation level dependent" (BOLD) signal, which is a proxy of neural activation. Each BOLD time series is associated with a different (x, y, z) coordinate within the brain, known as volumetric pixel or "voxel". In our study, we focused on the BOLD time series of a specific set of areas or "regions of interest" (ROI) of the brain (known as "medial frontal", "posterior cingulate", "left lateral parietal", "left medial temporal", "right lateral parietal", and "medial temporal" region), with each of these ROIs representing a different group of contiguous voxels in the so-called "Default Mode Network" (DMN).

We characterized the DMN time-series data set via the extraction of features aimed at identifying connectivity patterns within and between ROIs, with the goal of using these features as predictors for the decliner/non-decliner class (the latter being an aggregation of the "stable" and "gainer" classes). Our feature-extraction algorithms (FEAs) involved the iterative manipulation of Pearson's linear correlation values computed between the BOLD time series of a given source or "seed" voxel and that of a given set of target voxels. We extracted a total of 225 numeric features, subdivided in four groups, each of which is in turn presented in two variants. The first variant uses absolute correlations, while the second uses what we call "top positive" correlations, corresponding to a given fraction of top values selected from the only correlations with positive sign.

Further neuroscience background on the data set, as well as the pseudo-code of all our FEAs can be found in appendix B. Below we will refer to this $n = 29$, $p = 225$ data set as fMRI. Given the $n << p$ scenario, as mentioned in section 3.3, for this data set we replaced 10-fold cross validation with leave-one-out cross validation in our experimental protocol.

Evolutionary design of experiments data

These data sets, as well as the computational hardware to process them, were kindly provided by ProtoLife Inc., a company that develops combinatorial optimization methods for evolutionary design of experiments (EDoE). A fruitful collaboration with ProtoLife Inc. allowed us to acquire all of the EDoE and technical background required for the analysis described below. Design of experiments [64] comprises a broad set of statistical techniques for the estimation and study of the mathematical relationship f between the set of p components (input variables) that describe a process and the properties (output or "response" variable) of that process. These relationships are inferred from a collection of observations, with each observation corresponding to a different "experiment", represented by a vector of p values, one for each input variable, and its corresponding response value (typically, a number in $\mathbb{R}$), measured after materially executing that experiment in the real world. Since experiments are usually expensive, DoE aims at minimizing the size of the so-called "experimental space" E, that is, the overall set of experiments that need to be executed. A fundamental feasibility issue arises when p is large, since E typically grows exponentially with p. This issue is further exacerbated when p cannot be straightforwardly reduced by removing components a priori without severely affecting the process' response (for example, when components exhibit complex non-linear interactions).

EDoE is a form of DoE targeted to the discovery of f-optimizing experiments without the need of exhaustively performing all experiments in E [65]. In several variants of EDoE, experiments are iteratively executed in batches, corresponding to tiny subsets of E, called "generations" [66]. After each generation, a statistical or machine-learning model of the experimental process is trained on the observations collected at all previous generations, and is then used to predict the response of the remaining experiments in E, with the purpose of selecting new experiments in promising regions of E [67]. Further experiments are then selected in unexplored regions of E, usually disregarding their predicted response, with the purpose to collect observations where the currently available data is more sparse. The combination of these two experiment-selection criteria is essentially a design-of-experiments analog of exploration and exploitation strategies in approximate dynamic programming

[68]. Notably, observations collected based on these EDoE criteria are often dependent across input variables, unlike in DoE where inter-variable orthogonality is usually explicitly enforced.

We considered 4 data sets of experimental observations (all with numeric output variable) obtained with an EDoE approach:

- Liposome: with $n = 450$ and $p = 28$, with 2 binary and 26 numeric input variables.
- Amphiphile: with $n = 180$ and $p = 16$, with all numeric input variables.
- Robot: with $n = 529$ and $p = 22$, with all numeric input variables.
- Protein: with $n = 215$ and $p = 11$, with 1 categorical and 10 numeric input variables.

DNA microarray data

These data sets are available in the public domain, and were collected via DNA microarray technology [69]. A DNA microarray consists of an ensemble of DNA molecules attached to a rigid surface in a matrix-like arrangement. DNA microarrays allow to analyze in parallel the expression levels of a very large number of genes within a tissue or an organism. One of the typical goals of this analysis is the comparison of the expression levels of sick individuals with those of healthy ones, in order to identify genes that are responsible for a certain pathology. We considered 4 DNA microarray data sets, all with numeric continuous input variables and binary output variable [45]:

- Leukemia: with $n = 38$ observations and $p = 3051$ input variables.
- NCI: with $n = 61$ observations and $p = 5244$ input variables.
- Brain: with $n = 42$ observations and $p = 5597$ input variables.
- Breast2: with $n = 77$ observations and $p = 4869$ input variables.

For these $n << p$ microarray data sets, as mentioned in section 3.3, we replaced 10-fold cross validation with leave-one-out cross validation in our experimental protocol. Moreover: we excluded Hap and HapB from the list of benchmark methods, due to their computational

impracticality for problems of this high dimension; for DBC-RCPI, we evaluated the performance resulting from variations of the default hyper-parameter values, by increasing the value of δ to 1 (DBC-RCPI,$\delta = 1$), decreasing the value of m to 2 (DBC-RCPI,$m = 2$), or both of the above (DBC-RCPI,$\delta = 1$,$m = 2$).

Other data

This final collection includes miscellaneous public-domain data sets that were used to validate other methods for variable selection with RFs, such as [42] and [50]:

- Ozone: originally with $n = 366$ observations, but reduced to $n = 203$ in our study due to the removal of all observations containing missing values; $p = 12$ input variables, 9 of which numeric and 3 time-resolved, which we treated as numeric, and numeric output variable [70].
- BostonHousing: with $n = 506$ observations and originally $p = 13$ input variables, of which 11 numeric and 1 binary, but augmented to $p = 15$ in our study by including numeric "LAT" and "LON" variables from the corrected version of the data set [71]. The numeric continuous output variable was given by variable "CMEDV" from the same corrected version.
- Heart: with $n = 270$ observations and $p = 13$ input variables, of which 6 numeric, 4 categorical (of which 1 ordered), and 3 binary, and binary output variable [50].
- Parkinson's: with $n = 195$ observations and $p = 22$ input variables, all of which numeric, and binary output variable [72].

3.3.3 Results

The tables below report the values of $\bar{v}^*$ and $\bar{e}^*$ (see section 3.3) for all methods, as well as that of two further indicators, defined as follows: $r_{\bar{v}^*} = (\bar{v}^*_{LEBM} - \bar{v}^*)/\bar{v}^*_{LEBM}$ and $r_{\bar{e}^*} = (\bar{e}^*_{LEBM} - \bar{e}^*)/\bar{e}^*_{LEBM}$, where $\bar{v}^*_{LEBM}$ and $\bar{e}^*_{LEBM}$ are the $\bar{v}^*$ and $\bar{e}^*$ values of the "least error benchmark method" (LEBM), that is the benchmark method that yields the

smallest $\bar{e}^*$. In case multiple benchmark methods have minimal $\bar{e}^*$, the LEBM is chosen to be the method with both the smallest $\bar{e}^*$ and the smallest $\bar{v}^*$.

Positive values of these indicators represent relative improvements over the cardinality of the solution and over the error performance of the LEBM, respectively. For cases in which $\bar{e}^*_{LEBM} = 0$ would make $r_{\bar{e}^*}$ undefined, we define $r_{\bar{e}^*} = 0$ if $\bar{e}^* = 0$, and simply write $r_{\bar{e}^*} < 0$ otherwise. Note this indefiniteness issue does not apply to $r_{\bar{v}^*}$, since $\bar{v}^*_{LEBM}$ is always a strictly positive quantity. We will refer to $\bar{v}^*_{DBC-RCPI}$ and $\bar{e}^*_{DBC-RCPI}$ as the $\bar{v}^*$ and $\bar{e}^*$ values for DBC-RCPI, and to $r_{\bar{v}^*_{DBC-RCPI}}$ and $r_{\bar{e}^*_{DBCI-RCPI}}$ as the corresponding $r_{\bar{v}^*}$ and $r_{\bar{e}^*}$ values.

Artificial data

The analysis of the results obtained on the artificial data sets suggests that, in most cases, DBC-RCPI with hyper-parameters set to their default values (see section 3.3) yields testing error performance comparable to that of the LEBM, but through variable-subset solutions with significantly smaller cardinality. A similar conclusion can be drawn when comparing DBC-RCPI with control benchmark Std-CPI. With 20DLinear (Table 3.1) and on the vast majority of its variants with different values of n (Tables 3.2, 3.3, 3.4), R^2 (Tables 3.8, 3.9, 3.10) and ρ (Tables 3.5, 3.6, 3.7), $r_{\bar{v}^*_{DBC-RCPI}}$ fluctuates around 0.3 and $r_{\bar{e}^*_{DBCI-RCPI}}$ around 0. Larger values of n appear to result in larger values of $r_{\bar{v}^*_{DBC-RCPI}}$ and $r_{\bar{e}^*_{DBCI-RCPI}}$. We observe that $\bar{v}^*_{DBC-RCPI}$ tends to decrease with n, unlike with most of the other methods, which tend to return solutions with larger cardinality when n grows. The values of $r_{\bar{v}^*_{DBC-RCPI}}$ and $r_{\bar{e}^*_{DBCI-RCPI}}$ appears to also increase with R^2, which however does not seem to affect $\bar{v}^*_{DBC-RCPI}$; on the other hand, most of the other methods yield solutions with larger cardinality as R^2 grows. Larger values of ρ are associated with significantly larger values of $r_{\bar{v}^*_{DBC-RCPI}}$, as a consequence of decreasing values of $\bar{v}^*_{DBC-RCPI}$ (unlike with most other methods, whose solutions tend to increase in size as ρ grows). This is not surprising, since the clustering criterion within DBC-RCPI is specifically engineered to leverage strong inter-variable dependencies. While this behavior is desirable, we notice that it comes at the price of a slight decrease in $r_{\bar{e}^*_{DBCI-RCPI}}$. With the exception of 20DLinearClass (Table 3.11),

in which DBC-RCPI scores its worst error performance relative to the LEBM across all artificial data sets ($r_{\bar{e}^*_{DBCI-RCPI}} = -0.119$), the competitiveness of our method is confirmed on the various extensions of 20DLinear involving mixed input variables (20DLinearMixed, Table 3.12), varying correlation levels within input variable blocks (20DLinearDecreasCov, Table 3.13), and hidden non-linear transformations of the input variables (11DLinear+HiddenPoly, Table 3.15). Particularly encouraging is, moreover, the result on the high-dimensional, non-linear 11DLinear+Poly case (Table 3.14), where DBC-RCPI virtually matches the error performance of the LEBM via a solution a fifth of the size of that returned by the LEBM ($\bar{v}^*_{DBC-RCPI} = 0.8$).

Table 3.1: 20DLinear data.

Method	$\bar{\boldsymbol{v}}^*$	$\bar{\boldsymbol{e}}^*$	$\boldsymbol{r}_{\bar{\boldsymbol{v}}^*}$	$\boldsymbol{r}_{\bar{\boldsymbol{e}}^*}$
DBC-RCPI	6.578 ± 0.138	25.18 ± 0.476	0.324	-0.024
Hap	10.134 ± 0.099	25.129 ± 0.389	-0.042	-0.022
HapB	8.456 ± 0.078	26.435 ± 0.484	0.131	-0.075
Alt	6.046 ± 0.112	30.253 ± 0.662	0.379	-0.23
Diaz0	9.944 ± 0.188	24.936 ± 0.369	-0.022	-0.014
Diaz1	6.564 ± 0.287	27.575 ± 0.49	0.325	-0.121
Jiang0	9.776 ± 0.164	24.755 ± 0.372	-0.005	-0.007
Jiang1	6.374 ± 0.214	27.384 ± 0.485	0.345	-0.113
Svt	9.23 ± 0.209	25.393 ± 0.39	0.051	-0.032
GenP	3.086 ± 0.114	41.252 ± 1.475	0.683	-0.677
GenI	8.248 ± 0.235	25.66 ± 0.445	0.152	-0.043
Boruta	9.73 ± 0.09	24.595 ± 0.397	0	0
GRF	20 ± 0	27.387 ± 0.428	-1.055	-0.114
GRRF	15.08 ± 0.101	26.384 ± 0.429	-0.55	-0.073
Std-CPI	9.662 ± 0.191	24.52 ± 0.373	0.007	0.003

Table 3.2: 20DLinear data, variant with $n = 50$.

Method	$\bar{\boldsymbol{v}}^*$	$\bar{\boldsymbol{e}}^*$	$\boldsymbol{r}_{\bar{\boldsymbol{v}}^*}$	$\boldsymbol{r}_{\bar{\boldsymbol{e}}^*}$
DBC-RCPI	6.024 ± 0.168	32.959 ± 0.824	0.313	-0.068
Hap	8.82 ± 0.11	31.677 ± 0.849	-0.005	-0.026
HapB	5.24 ± 0.204	38.457 ± 1.478	0.403	-0.246
Alt	4.714 ± 0.092	39.194 ± 1.146	0.463	-0.27
Diaz0	8.114 ± 0.283	32.019 ± 0.853	0.075	-0.037
Diaz1	4.14 ± 0.197	37.466 ± 1.031	0.528	-0.214
Jiang0	8.388 ± 0.224	31.089 ± 0.796	0.044	-0.007
Jiang1	4.662 ± 0.115	35.776 ± 0.887	0.469	-0.159
Svt	9.17 ± 0.212	31.655 ± 0.75	-0.045	-0.026
GenP	2.61 ± 0.099	48.947 ± 1.763	0.702	-0.586
GenI	7.048 ± 0.236	31.989 ± 0.834	0.197	-0.036
Boruta	8.772 ± 0.108	30.863 ± 0.788	0	0
GRF	20 ± 0	33.858 ± 0.782	-1.28	-0.097
GRRF	14.734 ± 0.129	32.551 ± 0.739	-0.67	-0.055
Std-CPI	8.21 ± 0.259	31.352 ± 0.813	0.064	-0.016

Table 3.3: 20DLinear data, variant with $n = 200$.

Method	$\bar{v}^*$	$\bar{e}^*$	$r_{\bar{v}^*}$	$r_{\bar{e}^*}$
DBC-RCPI	6.824 ± 0.111	21.193 ± 0.233	0.346	0.015
Hap	10.758 ± 0.084	21.783 ± 0.24	-0.031	-0.012
HapB	9.646 ± 0.049	22.13 ± 0.243	0.076	-0.029
Alt	7.116 ± 0.062	28.27 ± 0.36	0.318	-0.314
Diaz0	10.342 ± 0.089	21.605 ± 0.236	0.009	-0.004
Diaz1	9.346 ± 0.109	22.111 ± 0.283	0.104	-0.028
Jiang0	10.434 ± 0.091	21.515 ± 0.233	0	0
Jiang1	8.624 ± 0.097	22.756 ± 0.266	0.173	-0.058
Svt	9.97 ± 0.03	21.565 ± 0.233	0.044	-0.002
GenP	3.31 ± 0.106	35.299 ± 1.02	0.683	-0.641
GenI	9.508 ± 0.112	21.915 ± 0.247	0.089	-0.019
Boruta	10.254 ± 0.061	21.547 ± 0.228	0.017	-0.001
GRF	20 ± 0	23.828 ± 0.257	-0.917	-0.107
GRRF	15.616 ± 0.071	23.024 ± 0.245	-0.497	-0.07
Std-CPI	10.22 ± 0.154	21.543 ± 0.228	0.021	-0.001

Table 3.4: 20DLinear data, variant with $n = 400$.

Method	$\bar{v}^*$	$\bar{e}^*$	$r_{\bar{v}^*}$	$r_{\bar{e}^*}$
DBC-RCPI	6.844 ± 0.073	17.425 ± 0.188	0.352	0.046
Hap	10.982 ± 0.081	18.419 ± 0.17	-0.041	-0.009
HapB	10.252 ± 0.049	18.285 ± 0.172	0.029	-0.002
Alt	7.38 ± 0.062	26.23 ± 0.266	0.301	-0.437
Diaz0	10.2 ± 0.058	18.337 ± 0.18	0.034	-0.004
Diaz1	9.328 ± 0.119	18.761 ± 0.181	0.116	-0.028
Jiang0	10.47 ± 0.06	18.28 ± 0.172	0.008	-0.001
Jiang1	8.982 ± 0.137	18.951 ± 0.189	0.149	-0.038
Svt	10 ± 0	18.301 ± 0.176	0.052	-0.002
GenP	3.52 ± 0.106	29.701 ± 0.532	0.666	-0.627
GenI	9.234 ± 0.153	18.719 ± 0.187	0.125	-0.025
Boruta	10.554 ± 0.071	18.257 ± 0.171	0	0
GRF	20 ± 0	20.133 ± 0.179	-0.895	-0.103
GRRF	15.976 ± 0.062	19.513 ± 0.175	-0.514	-0.069
Std-CPI	10.028 ± 0.106	18.351 ± 0.168	0.05	-0.005

Table 3.5: 20DLinear data, variant with $\rho = 0.5$.

Method	$\bar{v}^*$	$\bar{e}^*$	$r_{\bar{v}^*}$	$r_{\bar{e}^*}$
DBC-RCPI	6.792 ± 0.1	24.696 ± 0.416	0.239	0.012
Hap	9.33 ± 0.142	25.382 ± 0.427	-0.045	-0.015
HapB	6.516 ± 0.13	27.012 ± 0.474	0.27	-0.08
Alt	4.054 ± 0.075	32.555 ± 0.697	0.546	-0.302
Diaz0	7.634 ± 0.177	25.463 ± 0.466	0.145	-0.018
Diaz1	4.97 ± 0.11	28.858 ± 0.567	0.443	-0.154
Jiang0	7.402 ± 0.152	25.114 ± 0.422	0.171	-0.004
Jiang1	5.052 ± 0.09	28.255 ± 0.508	0.434	-0.13
Svt	8.13 ± 0.283	26.278 ± 0.43	0.089	-0.051
GenP	3.908 ± 0.104	31.09 ± 0.488	0.562	-0.243
GenI	6.396 ± 0.15	25.904 ± 0.489	0.283	-0.036
Boruta	8.926 ± 0.132	25.005 ± 0.409	0	0
GRF	20 ± 0	28.495 ± 0.451	-1.241	-0.14
GRRF	15.074 ± 0.095	27.468 ± 0.454	-0.689	-0.099
Std-CPI	7.12 ± 0.138	25.039 ± 0.452	0.202	-0.001

Table 3.6: 20DLinear data, variant with $\rho = 0.7$.

Method	$\bar{\boldsymbol{v}}^*$	$\bar{\boldsymbol{e}}^*$	$\boldsymbol{r}_{\bar{\boldsymbol{v}}^*}$	$\boldsymbol{r}_{\bar{\boldsymbol{e}}^*}$
DBC-RCPI	6.712 ± 0.126	25.693 ± 0.358	0.207	0.002
Hap	10.028 ± 0.102	25.898 ± 0.357	-0.184	-0.006
HapB	7.77 ± 0.139	27.373 ± 0.408	0.082	-0.063
Alt	4.7 ± 0.08	32.509 ± 0.7	0.445	-0.263
Diaz0	8.694 ± 0.18	25.858 ± 0.348	-0.027	-0.004
Diaz1	5.5 ± 0.211	29.776 ± 0.563	0.35	-0.157
Jiang0	8.468 ± 0.171	25.744 ± 0.353	0	0
Jiang1	5.576 ± 0.145	29.007 ± 0.489	0.342	-0.127
Svt	9.39 ± 0.17	26.149 ± 0.385	-0.109	-0.016
GenP	3.436 ± 0.099	35.837 ± 0.797	0.594	-0.392
GenI	7.254 ± 0.19	26.667 ± 0.399	0.143	-0.036
Boruta	9.59 ± 0.078	25.786 ± 0.359	-0.132	-0.002
GRF	20 ± 0	28.518 ± 0.369	-1.362	-0.108
GRRF	15.002 ± 0.087	27.361 ± 0.353	-0.772	-0.063
Std-CPI	8.092 ± 0.206	25.883 ± 0.371	0.044	-0.005

Table 3.7: 20DLinear data, variant with $\rho = 0.99$.

Method	$\bar{\boldsymbol{v}}^*$	$\bar{\boldsymbol{e}}^*$	$\boldsymbol{r}_{\bar{\boldsymbol{v}}^*}$	$\boldsymbol{r}_{\bar{\boldsymbol{e}}^*}$
DBC-RCPI	5.786 ± 0.196	25.931 ± 0.443	0.399	-0.012
Hap	10.094 ± 0.091	25.772 ± 0.443	-0.048	-0.006
HapB	8.702 ± 0.068	27.715 ± 0.534	0.097	-0.082
Alt	7.326 ± 0.09	31.677 ± 0.645	0.24	-0.237
Diaz0	10.314 ± 0.083	25.618 ± 0.45	-0.071	0
Diaz1	7.674 ± 0.274	28.019 ± 0.536	0.203	-0.094
Jiang0	10.244 ± 0.08	25.64 ± 0.45	-0.063	-0.001
Jiang1	7.21 ± 0.207	28.255 ± 0.501	0.252	-0.103
Svt	9.31 ± 0.218	26.277 ± 0.512	0.034	-0.026
GenP	2.598 ± 0.108	54.181 ± 1.704	0.73	-1.115
GenI	8.884 ± 0.169	26.314 ± 0.463	0.078	-0.027
Boruta	9.634 ± 0.087	25.615 ± 0.467	0	0
GRF	20 ± 0	28.76 ± 0.502	-1.076	-0.123
GRRF	16.016 ± 0.118	27.76 ± 0.502	-0.662	-0.083
Std-CPI	10.372 ± 0.105	25.584 ± 0.461	-0.077	0.001

Table 3.8: 20DLinear data, variant with $R^2 = 0.7$.

Method	$\bar{\boldsymbol{v}}^*$	$\bar{\boldsymbol{e}}^*$	$\boldsymbol{r}_{\bar{\boldsymbol{v}}^*}$	$\boldsymbol{r}_{\bar{\boldsymbol{e}}^*}$
DBC-RCPI	6.544 ± 0.164	56.668 ± 1.212	0.296	-0.055
Hap	9.49 ± 0.099	54.005 ± 0.934	-0.021	-0.006
HapB	6.65 ± 0.178	58.684 ± 1.211	0.285	-0.093
Alt	5.358 ± 0.109	61.813 ± 1.219	0.424	-0.151
Diaz0	9.672 ± 0.269	54.358 ± 1.023	-0.04	-0.012
Diaz1	4.448 ± 0.208	60.647 ± 1.113	0.522	-0.129
Jiang0	9.57 ± 0.227	54.079 ± 0.982	-0.029	-0.007
Jiang1	5.05 ± 0.152	59.38 ± 1.049	0.457	-0.106
Svt	9.28 ± 0.201	53.893 ± 0.941	0.002	-0.003
GenP	3.094 ± 0.102	72.134 ± 1.757	0.667	-0.343
GenI	8.024 ± 0.226	54.78 ± 1.011	0.137	-0.02
Boruta	9.298 ± 0.097	53.708 ± 0.975	0	0
GRF	20 ± 0.133	55.464 ± 0.975	-1.151	-0.033
GRRF	16.728 ± 0	54.894 ± 0.988	-0.799	-0.022
Std-CPI	9.658 ± 0.279	54.475 ± 1.091	-0.039	-0.014

Table 3.9: 20DLinear data, variant with $R^2 = 0.8$.

Method	$\bar{v}^*$	$\bar{e}^*$	$r_{\bar{v}^*}$	$r_{\bar{e}^*}$
DBC-RCPI	6.558 ± 0.137	38.013 ± 0.783	0.312	-0.024
Hap	9.882 ± 0.1	37.383 ± 0.631	-0.037	-0.007
HapB	7.738 ± 0.112	39.801 ± 0.723	0.188	-0.072
Alt	5.762 ± 0.117	43.648 ± 0.939	0.395	-0.176
Diaz0	10.042 ± 0.203	37.456 ± 0.635	-0.054	-0.009
Diaz1	5.318 ± 0.24	42.236 ± 0.828	0.442	-0.138
Jiang0	9.746 ± 0.187	37.338 ± 0.635	-0.023	-0.006
Jiang1	5.61 ± 0.202	41.083 ± 0.763	0.411	-0.107
Svt	9.23 ± 0.221	37.696 ± 0.609	0.031	-0.015
GenP	3.064 ± 0.091	53.961 ± 1.471	0.678	-0.453
GenI	8.342 ± 0.217	37.911 ± 0.634	0.124	-0.021
Boruta	9.528 ± 0.097	37.128 ± 0.669	0	0
GRF	20 ± 0	39.502 ± 0.655	-1.099	-0.064
GRRF	15.888 ± 0.115	38.697 ± 0.651	-0.667	-0.042
Std-CPI	9.724 ± 0.211	37.206 ± 0.666	-0.021	-0.002

Table 3.10: 20DLinear data, variant with $R^2 = 0.99$.

Method	$\bar{v}^*$	$\bar{e}^*$	$r_{\bar{v}^*}$	$r_{\bar{e}^*}$
DBC-RCPI	6.548 ± 0.118	16.353 ± 0.295	0.334	0.01
Hap	10.208 ± 0.096	16.875 ± 0.249	-0.038	-0.021
HapB	8.88 ± 0.072	17.835 ± 0.306	0.097	-0.079
Alt	6.294 ± 0.103	21.491 ± 0.478	0.36	-0.301
Diaz0	9.938 ± 0.123	16.65 ± 0.254	-0.01	-0.008
Diaz1	7.49 ± 0.297	18.291 ± 0.305	0.239	-0.107
Jiang0	9.824 ± 0.115	16.536 ± 0.245	0.001	-0.001
Jiang1	7.026 ± 0.243	18.22 ± 0.318	0.286	-0.103
Svt	9.37 ± 0.189	17.18 ± 0.305	0.047	-0.04
GenP	2.886 ± 0.103	33.293 ± 1.508	0.707	-1.015
GenI	8.402 ± 0.215	17.274 ± 0.277	0.146	-0.046
Boruta	9.836 ± 0.078	16.522 ± 0.239	0	0
GRF	20 ± 0	19.337 ± 0.295	-1.033	-0.17
GRRF	14.424 ± 0.078	18.175 ± 0.275	-0.466	-0.1
Std-CPI	9.752 ± 0.117	16.541 ± 0.24	0.009	-0.001

Table 3.11: 20DLinearClass data.

Method	$\bar{v}^*$	$\bar{e}^*$	$r_{\bar{v}^*}$	$r_{\bar{e}^*}$
DBC-RCPI	6.51 ± 0.28	0.174 ± 0.006	0.3	-0.119
Hap	9.78 ± 0.109	0.158 ± 0.004	-0.052	-0.015
HapB	6.802 ± 0.148	0.174 ± 0.006	0.269	-0.117
Alt	5.71 ± 0.107	0.185 ± 0.007	0.386	-0.188
Diaz0	8.264 ± 0.37	0.167 ± 0.005	0.111	-0.071
Diaz1	4.662 ± 0.194	0.185 ± 0.007	0.499	-0.185
Jiang0	8.344 ± 0.267	0.166 ± 0.005	0.103	-0.063
Jiang1	5.078 ± 0.15	0.176 ± 0.005	0.454	-0.13
Svt	12.99 ± 0.482	0.159 ± 0.005	-0.397	-0.02
GenP	2.362 ± 0.076	0.264 ± 0.009	0.746	-0.69
GenI	9.79 ± 0.334	0.164 ± 0.005	-0.053	-0.052
Boruta	9.3 ± 0.097	0.156 ± 0.004	0	0
GRF	17.118 ± 0.089	0.16 ± 0.005	-0.841	-0.029
GRRF	6.798 ± 0.159	0.171 ± 0.005	0.269	-0.094
Std-CPI	8.416 ± 0.367	0.165 ± 0.005	0.095	-0.055

Table 3.12: 20DLinearMixed data.

Method	$\bar{v}^*$	$\bar{e}^*$	$r_{\bar{v}^*}$	$r_{\bar{e}^*}$
DBC-RCPI	6.884 ± 0.14	28.206 ± 0.526	0.288	-0.047
Hap	9.816 ± 0.114	27.697 ± 0.435	-0.016	-0.028
HapB	7.78 ± 0.087	29.226 ± 0.513	0.195	-0.085
Alt	5.998 ± 0.089	31.574 ± 0.653	0.379	-0.172
Diaz0	9.902 ± 0.222	27.95 ± 0.472	-0.024	-0.038
Diaz1	5.508 ± 0.245	31.403 ± 0.52	0.43	-0.166
Jiang0	9.67 ± 0.181	27.711 ± 0.469	0	-0.029
Jiang1	5.852 ± 0.165	30.361 ± 0.474	0.395	-0.127
Svt	7.7 ± 0.281	29.734 ± 0.534	0.203	-0.104
GenP	3.052 ± 0.091	39.386 ± 1.29	0.684	-0.462
GenI	7.884 ± 0.227	28.655 ± 0.485	0.184	-0.064
Boruta	9.666 ± 0.094	26.937 ± 0.428	0	0
GRF	20 ± 0	30.033 ± 0.489	-1.069	-0.115
.GRRF	16.59 ± 0.089	29.433 ± 0.472	-0.009	0.007
Std-CPI	9.828 ± 0.133	27.093 ± 0.434	-0.017	-0.006

Table 3.13: 20DLinearDecreasCov data.

Method	$\bar{v}^*$	$\bar{e}^*$	$r_{\bar{v}^*}$	$r_{\bar{e}^*}$
DBC-RCPI	6.074 ± 0.119	25.339 ± 0.465	0.238	0.006
Hap	8.998 ± 0.1	25.841 ± 0.444	-0.129	-0.014
HapB	6.894 ± 0.078	27.181 ± 0.535	0.135	-0.066
Alt	4.95 ± 0.061	30.636 ± 0.624	0.379	-0.202
Diaz0	8.252 ± 0.122	25.982 ± 0.486	-0.035	-0.019
Diaz1	5.442 ± 0.191	28.935 ± 0.595	0.317	-0.135
Jiang0	7.97 ± 0.116	25.493 ± 0.454	0	0
Jiang1	5.538 ± 0.112	27.752 ± 0.506	0.305	-0.089
Svt	6.79 ± 0.281	28.5 ± 0.61	0.148	-0.118
GenP	2.818 ± 0.095	38.961 ± 1.159	0.646	-0.528
GenI	6.86 ± 0.133	25.857 ± 0.501	0.139	-0.014
Boruta	8.76 ± 0.102	25.84 ± 0.462	-0.099	-0.014
GRF	20 ± 0	29.137 ± 0.45	-1.509	-0.143
GRRF	14.808 ± 0.087	27.924 ± 0.453	-0.858	-0.095
Std-CPI	8.038 ± 0.134	25.319 ± 0.467	-0.008	0.007

Table 3.14: 11DLinear+Poly data.

Method	$\bar{v}^*$	$\bar{e}^*$	$r_{\bar{v}^*}$	$r_{\bar{e}^*}$
DBC-RCPI	15.058 ± 0.437	348.538 ± 8.84	0.804	-0.019
Alt	11.408 ± 0.172	485.072 ± 11.2	0.852	-0.419
Hap	34.256 ± 0.761	352.688 ± 8.408	0.551	-0.031
HapB	17.866 ± 0.566	411.453 ± 11.312	0.768	-0.203
Diaz0	41.266 ± 0.849	343.084 ± 8.558	0.464	-0.003
Diaz1	17.724 ± 0.741	412.559 ± 15.098	0.77	-0.207
Jiang0	33.386 ± 1.045	347.74 ± 8.623	0.566	-0.017
Jiang1	11.094 ± 0.469	419.006 ± 13.449	0.856	-0.225
Svt	42.098 ± 1.23	346.127 ± 8.753	0.453	-0.012
GenP	3.888 ± 0.098	554.951 ± 14.613	0.95	-0.623
GenI	34.928 ± 0.599	349.905 ± 8.707	0.546	-0.023
Boruta	39.166 ± 0.737	345.349 ± 8.406	0.491	-0.01
GRF	77 ± 0	341.918 ± 8.499	0	0
GRRF	28.608 ± 0.084	357.425 ± 8.411	0.628	-0.045
Std-CPI	37.138 ± 0.914	342.686 ± 8.233	0.517	-0.002

Table 3.15: 11DLinear+HiddenPoly data.

Method	$\bar{v}^*$	$\bar{e}^*$	$r_{\bar{v}^*}$	$r_{\bar{e}^*}$
DBC-RCPI	7.572 ± 0.13	413.556 ± 9.444	0.312	-0.02
Hap	9.714 ± 0.068	413.249 ± 8.868	0.117	-0.019
HapB	7.674 ± 0.21	450.057 ± 15.613	0.302	-0.11
Alt	3.302 ± 0.086	596.644 ± 20.24	0.7	-0.472
Diaz0	10.07 ± 0.128	409.596 ± 8.957	0.085	-0.01
Diaz1	4.624 ± 0.2	478.025 ± 11.626	0.58	-0.179
Jiang0	9.798 ± 0.111	408.717 ± 8.908	0.109	-0.008
Jiang1	4.864 ± 0.133	466.249 ± 10.622	0.558	-0.15
Svt	9.93 ± 0.225	417.878 ± 8.852	0.097	-0.031
GenP	3.462 ± 0.121	560.762 ± 18.985	0.685	-0.383
GenI	8.088 ± 0.195	424.348 ± 9.284	0.265	-0.047
Boruta	9.82 ± 0.06	410.51 ± 8.76	0.107	-0.012
GRF	11 ± 0	405.458 ± 8.656	0	0
GRRF	11 ± 0	405.458 ± 8.656	0	0
Std-CPI	9.778 ± 0.109	408.88 ± 9.172	0.111	-0.008

Real-world data

The results on the real-world data sets overall confirm the competitiveness of DBC-RCPI over the benchmark methods, as observed with the artificial data sets.

DBC-RCPI clearly outperforms all other methods on the fMRI data set (Table 3.16), returning a solution that typically contains only one variable (corresponding to the IeTC proportion between the medial frontal and posterior cingulate ROIs, see section B.2), yielding a classification accuracy of about 83% (with 73% sensitivity and 89% specificity). On three out four of the EDoE data sets (Liposome, Amphiphile, and Robot, Tables 3.17 3.18 3.19), DBC-RCPI produces solutions with substantially smaller size than that obtained with the LEBM, although in the Liposome case getting outperformed by three of the other benchmark methods, and in the Amphiphile case at the cost of a relatively large loss in error performance ($r_{\bar{e}^*_{DBCI-RCPI}} = -0.076$). In the remaining Protein data set (Table 3.20), while the advantage of DBC-RCPI over the LEBM in terms of solution size is just moderate, it also comes with a small but significant improvement over the LEBM's error ($r_{\bar{e}^*_{DBCI-RCPI}} = 0.054$, corresponding to an advantage over the LEBM of more than 6 standard errors).

On the Ozone (Table 3.21), BostonHousing (Table 3.22), and Heart (Table 3.23) data sets, DBC-RCPI's error performance is essentially identical to that of the LEBM, while it is obtained with a significantly, and sometimes drastically, smaller solution (e.g., about a third

as large in the Heart case). A similar gap in solution size is observed in the Parkinson's data set ($\bar{v}^*_{DBC-RCPI} = 0.68$), where however DBC-RCPI's error is 6.5% larger than the LEBM's (Table 3.24); it is worth noting, nonetheless, that all of the benchmarks with comparably small solutions yield substantially larger error compared to DBC-RCPI.

Within the collection of DNA microarray data sets, DBC-RCPI obtains its best results on Leukemia and NCI with variant DBC-RCPI,$m = 2, \delta = 1$. In the former data set, such variant yields a zero error as the LEBM, but with a solution about a tenth of the size. A similar result is obtained in the latter data set, with error performance analogous to that of the LEBM, but via a solution about a twentieth as large. On the Brain (variants DBC-RCPI,$\delta = 1$ and DBC-RCPI,$m = 2, \delta = 1$) and Breast2 (variant DBC-RCPI,$\delta = 1$) data sets, DBC-RCPI's error is slightly larger than the LEBM's, although it is in line with that of most benchmark methods; in terms of solution size, DBC-RCPI ranks roughly average with respect to benchmark methods with similar error performance.

Table 3.16: fMRI data.

Method	$\bar{v}^*$	$\bar{e}^*$	$r_{\bar{v}^*}$	$r_{\bar{e}^*}$
DBC-RCPI, $\delta = 2$	1.034 ± 0.034	0.172 ± 0.071	0.250	0.169
Hap	9.655 ± 0.537	0.31 ± 0.087	-6.001	-0.498
HapB	0.621 ± 0.135	0.483 ± 0.094	0.550	-1.333
Alt	10.241 ± 0.538	0.379 ± 0.092	-6.426	-0.831
Diaz0	3.069 ± 0.530	0.241 ± 0.081	-1.226	-0.164
Diaz1	1.379 ± 0.160	0.207 ± 0.077	0	0
Jiang0	2.483 ± 0.459	0.207 ± 0.077	-0.801	0
Jiang1	1.966 ± 0.312	0.207 ± 0.077	-0.426	0
Svt	16.897 ± 4.919	0.379 ± 0.092	-11.253	-0.831
GenP	1.207 ± 0.077	0.241 ± 0.081	0.125	-0.164
GenI	6.69 ± 0.951	0.379 ± 0.092	-3.851	-0.831
Boruta	4.069 ± 0.488	0.276 ± 0.093	-1.951	-0.333
GRF	81.207 ± 9.677	0.448 ± 0.151	-57.888	-1.164
GRRF	4.724 ± 0.155	0.31 ± 0.128	-2.426	-0.498
Std-CPI	3 ± 0.489	0.31 ± 0.087	-1.175	-0.498

Table 3.17: Liposome data.

Method	$\bar{v}^*$	$\bar{e}^*$	$r_{\bar{v}^*}$	$r_{\bar{e}^*}$
DBC-RCPI	16.22 ± 0.147	1 ± 0.005	0.421	-0.02
Hap	9.995 ± 0.06	0.99 ± 0.003	0.643	-0.01
HapB	5.25 ± 0.06	1.028 ± 0.005	0.813	-0.049
Alt	7.455 ± 0.065	1 ± 0.003	0.734	-0.02
Diaz0	18.395 ± 0.197	0.99 ± 0.004	0.343	-0.01
Diaz1	6.61 ± 0.063	1.016 ± 0.004	0.764	-0.037
Jiang0	18.1 ± 0.281	0.988 ± 0.005	0.354	-0.008
Jiang1	6.795 ± 0.088	1.026 ± 0.003	0.757	-0.047
Svt	9.5 ± 0.093	1.001 ± 0.005	0.661	-0.021
GenP	4.495 ± 0.028	1.035 ± 0.003	0.839	-0.056
GenI	9.21 ± 0.125	1.011 ± 0.004	0.671	-0.032
Boruta	10.485 ± 0.066	0.999 ± 0.004	0.626	-0.019
GRF	28 ± 0	0.98 ± 0.004	0	0
GRRF	28 ± 0	0.98 ± 0.004	0	0
Std-CPI	18.495 ± 0.2	1.006 ± 0.005	0.339	-0.027

Table 3.18: Amphiphile data.

Method	$\bar{v}^*$	$\bar{e}^*$	$r_{\bar{v}^*}$	$r_{\bar{e}^*}$
DBC-RCPI	10.501 ± 0.101	0.969 ± 0.015	0.343	-0.076
Hap	4.73 ± 0.041	1.056 ± 0.012	0.704	-0.173
HapB	2.495 ± 0.037	1.244 ± 0.011	0.844	-0.381
Alt	2.2 ± 0.023	1.282 ± 0.008	0.862	-0.424
Diaz0	8.055 ± 0.178	1.02 ± 0.012	0.497	-0.134
Diaz1	4.175 ± 0.028	1.01 ± 0.011	0.739	-0.122
Jiang0	7.915 ± 0.16	1.028 ± 0.014	0.505	-0.142
Jiang1	4.045 ± 0.014	0.958 ± 0.011	0.747	-0.064
Svt	2.4 ± 0.04	1.268 ± 0.008	0.85	-0.408
GenP	3.305 ± 0.027	1.168 ± 0.014	0.793	-0.297
GenI	3.765 ± 0.033	1.052 ± 0.018	0.765	-0.168
Boruta	7.675 ± 0.042	1.022 ± 0.012	0.52	-0.135
GRF	16 ± 0	0.9 ± 0.008	0	0
GRRF	16 ± 0	0.9 ± 0.008	0	0
Std-CPI	11.435 ± 0.126	0.93 ± 0.011	0.285	-0.033

Table 3.19: Robot data.

Method	$\bar{v}^*$	$\bar{e}^*$	$r_{\bar{v}^*}$	$r_{\bar{e}^*}$
DBC-RCPI	17.72 ± 0.075	0.988 ± 0.007	0.114	-0.012
Hap	19.845 ± 0.023	0.987 ± 0.008	0.008	-0.011
HapB	19.065 ± 0.024	0.987 ± 0.008	0.047	-0.011
Alt	3.99 ± 0.016	1.436 ± 0.011	0.8	-0.471
Diaz0	17.64 ± 0.132	1 ± 0.007	0.118	-0.025
Diaz1	11.135 ± 0.055	1.067 ± 0.007	0.443	-0.093
Jiang0	17.88 ± 0.14	0.999 ± 0.007	0.106	-0.024
Jiang1	11.38 ± 0.062	1.078 ± 0.007	0.431	-0.105
Svt	6.85 ± 0.042	1.151 ± 0.008	0.657	-0.179
GenP	4.985 ± 0.008	1.209 ± 0.008	0.751	-0.239
GenI	5.05 ± 0.017	1.209 ± 0.008	0.747	-0.239
Boruta	19.995 ± 0.005	0.976 ± 0.007	0	0
GRF	20 ± 0	0.976 ± 0.007	0	0
GRRF	20 ± 0	0.976 ± 0.007	0	0
Std-CPI	18.565 ± 0.102	0.986 ± 0.008	0.072	-0.01

Table 3.20: Protein data.

Method	$\bar{v}^*$	$\bar{e}^*$	$r_{\bar{v}^*}$	$r_{\bar{e}^*}$
DBC-RCPI	7.01 ± 0.02	0.858 ± 0.006	0.039	0.054
Hap	7.63 ± 0.039	0.935 ± 0.008	-0.046	-0.031
HapB	6.32 ± 0.039	1.008 ± 0.009	0.134	-0.111
Alt	3.915 ± 0.036	1.294 ± 0.022	0.463	-0.427
Diaz0	7.3 ± 0.066	0.909 ± 0.006	-0.001	-0.002
Diaz1	6.16 ± 0.052	1.011 ± 0.011	0.156	-0.115
Jiang0	7.295 ± 0.067	0.907 ± 0.007	0	0
Jiang1	6.195 ± 0.063	1 ± 0.008	0.151	-0.103
Svt	3 ± 0	1.773 ± 0.005	0.589	-0.955
GenP	2.825 ± 0.019	1.872 ± 0.015	0.613	-1.064
GenI	3 ± 0	1.773 ± 0.005	0.589	-0.955
Boruta	7.005 ± 0.032	0.932 ± 0.009	0.040	-0.028
GRF	11 ± 0	0.988 ± 0.005	-0.508	-0.089
GRRF	11 ± 0	0.988 ± 0.005	-0.508	-0.089
Std-CPI	7.075 ± 0.027	0.872 ± 0.006	0.030	0.039

Table 3.21: Ozone data.

Method	$\bar{v}^*$	$\bar{e}^*$	$r_{\bar{v}^*}$	$r_{\bar{e}^*}$
DBC-RCPI	4.6 ± 0.051	16.986 ± 0.084	0.489	0.011
Hap	8.768 ± 0.016	17.391 ± 0.062	0.026	-0.012
HapB	8.058 ± 0.018	17.927 ± 0.062	0.105	-0.044
Alt	2.978 ± 0.009	23.331 ± 0.093	0.669	-0.358
Diaz0	10.388 ± 0.038	17.565 ± 0.055	-0.153	-0.022
Diaz1	6.196 ± 0.034	19.63 ± 0.095	0.312	-0.143
Jiang0	9.516 ± 0.041	17.439 ± 0.059	-0.057	-0.015
Jiang1	5.168 ± 0.039	19.619 ± 0.104	0.426	-0.142
Svt	12 ± 0	17.604 ± 0.054	-0.332	-0.025
GenP	4.45 ± 0.066	23.569 ± 0.211	0.506	-0.372
GenI	9.212 ± 0.022	17.302 ± 0.057	-0.023	-0.007
Boruta	9.006 ± 0.003	17.179 ± 0.059	0	0
GRF	12 ± 0	17.604 ± 0.054	-0.332	-0.025
GRRF	11.996 ± 0.003	17.603 ± 0.053	-0.332	-0.025
Std-CPI	7.104 ± 0.05	17.245 ± 0.076	0.211	-0.004

Table 3.22: BostonHousing data.

Method	$\bar{v}^*$	$\bar{e}^*$	$r_{\bar{v}^*}$	$r_{\bar{e}^*}$
DBC-RCPI	9.46 ± 0.078	9.311 ± 0.065	0.208	-0.003
Hap	12.81 ± 0.03	9.346 ± 0.06	-0.073	-0.007
HapB	11.458 ± 0.027	9.382 ± 0.061	0.04	-0.011
Alt	3.732 ± 0.037	13.785 ± 0.126	0.687	-0.485
Diaz0	11.288 ± 0.088	9.396 ± 0.065	0.054	-0.012
Diaz1	5.468 ± 0.038	10.955 ± 0.068	0.542	-0.18
Jiang0	11.938 ± 0.072	9.282 ± 0.068	0	0
Jiang1	5.93 ± 0.022	10.716 ± 0.078	0.503	-0.154
Svt	7.152 ± 0.045	10.73 ± 0.101	0.401	-0.156
GenP	4.122 ± 0.024	12.316 ± 0.093	0.655	-0.327
GenI	5.386 ± 0.03	10.962 ± 0.078	0.549	-0.181
Boruta	15 ± 0	9.358 ± 0.057	-0.256	-0.008
GRF	15 ± 0	9.358 ± 0.057	-0.256	-0.008
GRRF	13.68 ± 0.026	9.283 ± 0.061	-0.146	0
Std-CPI	11.288 ± 0.092	9.326 ± 0.07	0.054	-0.005

Table 3.23: Heart data.

Method	$\bar{v}^*$	$\bar{e}^*$	$r_{\bar{v}^*}$	$r_{\bar{e}^*}$
DBC-RCPI	4.422 ± 0.101	0.168 ± 0.002	0.66	0.007
Hap	8.798 ± 0.034	0.178 ± 0.002	0.323	-0.051
HapB	7.688 ± 0.027	0.176 ± 0.001	0.409	-0.04
Alt	4.992 ± 0.026	0.176 ± 0.002	0.616	-0.041
Diaz0	7.588 ± 0.17	0.176 ± 0.002	0.416	-0.044
Diaz1	4.494 ± 0.096	0.189 ± 0.002	0.654	-0.117
Jiang0	11.562 ± 0.069	0.175 ± 0.001	0.111	-0.035
Jiang1	7.592 ± 0.067	0.2 ± 0.002	0.416	-0.185
Svt	12.958 ± 0.024	0.169 ± 0.001	0.003	-0.003
GenP	1.702 ± 0.045	0.271 ± 0.004	0.869	-0.602
GenI	6.988 ± 0.131	0.178 ± 0.002	0.462	-0.052
Boruta	9.368 ± 0.022	0.18 ± 0.002	0.279	-0.063
GRF	13 ± 0	0.169 ± 0.001	0	0
GRRF	8.866 ± 0.025	0.192 ± 0.001	0.318	-0.138
Std-CPI	3.938 ± 0.098	0.163 ± 0.002	0.697	0.037

Table 3.24: Parkinson's data.

Method	$\bar{v}^*$	$\bar{e}^*$	$r_{\bar{v}^*}$	$r_{\bar{e}^*}$
DBC-RCPI	7.038 ± 0.164	0.094 ± 0.002	0.68	-0.065
Hap	14.702 ± 0.064	0.09 ± 0.001	0.331	-0.016
HapB	7.662 ± 0.05	0.1 ± 0.002	0.651	-0.13
Alt	3.798 ± 0.028	0.113 ± 0.002	0.827	-0.272
Diaz0	11.554 ± 0.19	0.096 ± 0.001	0.474	-0.086
Diaz1	6.038 ± 0.077	0.109 ± 0.002	0.725	-0.227
Jiang0	8.784 ± 0.166	0.096 ± 0.002	0.6	-0.085
Jiang1	4.824 ± 0.046	0.1 ± 0.002	0.781	-0.126
Svt	15.328 ± 0.219	0.098 ± 0.001	0.303	-0.101
GenP	3.658 ± 0.053	0.136 ± 0.002	0.834	-0.534
GenI	9.97 ± 0.16	0.101 ± 0.002	0.546	-0.137
Boruta	21.978 ± 0.006	0.089 ± 0.001	0	0
GRF	19.998 ± 0.038	0.089 ± 0.001	0.09	-0.001
GRRF	5.36 ± 0.03	0.109 ± 0.002	0.756	-0.233
Std-CPI	12.006 ± 0.18	0.097 ± 0.002	0.454	-0.09

Table 3.25: Leukemia data.

Method	$\bar{v}^*$	$\bar{e}^*$	$r_{\bar{v}^*}$	$r_{\bar{e}^*}$
DBC-RCPI	5.895 ± 1.222	0.053 ± 0.029	0.856	< 0
DBC-RCPI,$\delta = 1$	5.895 ± 1.222	0.053 ± 0.029	0.856	< 0
DBC-RCPI,$m = 2$	3.658 ± 0.602	0 ± 0	0.911	0
DBC-RCPI,$m = 2$,$\delta = 1$	3.658 ± 0.602	0 ± 0	0.911	0
Alt	104.5 ± 0.853	0.053 ± 0.029	-1.547	< 0
Diaz0	2.316 ± 0.233	0.053 ± 0.029	0.944	< 0
Diaz1	2.316 ± 0.233	0.053 ± 0.029	0.944	< 0
Jiang0	2 ± 0.03	0.053 ± 0.029	0.951	< 0
Jiang1	2 ± 0.03	0.053 ± 0.029	0.951	< 0
Svt	31.421 ± 2.865	0.053 ± 0.029	0.234	< 0
GenP	1.5 ± 0.064	0.026 ± 0.021	0.963	< 0
GenI	1.579 ± 0.70	0.026 ± 0.021	0.962	< 0
Boruta	41.026 ± 0.356	0 ± 0	0	0
GRF	237.895 ± 0.716	0 ± 0	-4.799	0
GRRF	2.921 ± 0.116	0.026 ± 0.021	0.929	< 0
Std-CPI	1.342 ± 0.035	0 ± 0	0.967	0

Table 3.26: NCI data.

Method	$\bar{v}^*$	$\bar{e}^*$	$r_{\bar{v}^*}$	$r_{\bar{e}^*}$
DBC-RCPI	39.508 ± 3.207	0.393 ± 0.063	0.897	-0.143
DBC-RCPI,$\delta = 1$	21.984 ± 1.488	0.377 ± 0.063	0.942	-0.095
DBC-RCPI,$m = 2$	32.279 ± 2.003	0.393 ± 0.063	0.915	-0.143
DBC-RCPI,$m = 2$,$\delta = 1$	19.967 ± 1.064	0.344 ± 0.061	0.948	0
Alt	381.82 ± 1.377	0.344 ± 0.061	0	0
Diaz0	87.557 ± 7.921	0.393 ± 0.063	0.771	-0.143
Diaz1	47.984 ± 3.351	0.393 ± 0.063	0.874	-0.143
Jiang0	47.115 ± 3.209	0.459 ± 0.064	0.877	-0.333
Jiang1	30.525 ± 2.693	0.475 ± 0.064	0.92	-0.381
Svt	1045.279 ± 136.122	0.344 ± 0.061	-1.738	0
GenP	2.787 ± 0.194	0.754 ± 0.056	0.993	-1.19
GenI	210.508 ± 27.24	0.426 ± 0.064	0.449	-0.238
Boruta	32.246 ± 0.422	0.426 ± 0.064	0.916	-0.238
GRF	458.41 ± 1.012	0.361 ± 0.062	-0.201	-0.048
GRRF	10.803 ± 0.321	0.59 ± 0.063	0.972	-0.714
Std-CPI	139.377 ± 14.272	0.443 ± 0.064	0.635	-0.286

Table 3.27: Brain data.

Method	$\bar{v}^*$	$\bar{e}^*$	$r_{\bar{v}^*}$	$r_{\bar{e}^*}$
DBC-RCPI	12.048 ± 0.74	0.31 ± 0.072	0.96	-0.625
DBC-RCPI,$\delta = 1$	12.048 ± 0.74	0.31 ± 0.072	0.96	-0.625
DBC-RCPI,$m = 2$	15.881 ± 1.454	0.286 ± 0.071	0.947	-0.505
DBC-RCPI,$m = 2$,$\delta = 1$	15.881 ± 1.454	0.286 ± 0.071	0.947	-0.505
Alt	301.286 ± 1.492	0.19 ± 0.061	0	0
Diaz0	23.714 ± 2.326	0.262 ± 0.069	0.921	-0.375
Diaz1	22.405 ± 2.035	0.286 ± 0.071	0.926	-0.5
Jiang0	9.19 ± 0.331	0.286 ± 0.071	0.969	-0.5
Jiang1	9.19 ± 0.331	0.286 ± 0.071	0.969	-0.5
Svt	333.405 ± 40.775	0.214 ± 0.064	-0.107	-0.125
GenP	4.167 ± 0.186	0.429 ± 0.077	0.986	-1.25
GenI	16.619 ± 2.448	0.286 ± 0.071	0.945	-0.5
Boruta	40.548 ± 0.836	0.286 ± 0.071	0.865	-0.5
GRF	407.143 ± 1.116	0.19 ± 0.061	-0.351	0
GRRF	6.119 ± 0.178	0.262 ± 0.069	0.98	-0.375
Std-CPI	32.048 ± 3.53	0.31 ± 0.072	0.894	-0.625

Table 3.28: Breast2 data.

Method	$\bar{v}^*$	$\bar{e}^*$	$r_{\bar{v}^*}$	$r_{\bar{e}^*}$
DBC-RCPI	22.169 ± 1.401	0.351 ± 0.054	-1.052	-0.227
DBC-RCPI,$\delta = 1$	12.39 ± 1.152	0.338 ± 0.056	-0.147	-0.182
DBC-RCPI,$m = 2$	20.026 ± 1.132	0.416 ± 0.057	-0.853	-0.455
DBC-RCPI,$m = 2$,$\delta = 1$	9.221 ± 0.596	0.416 ± 0.058	0.147	-0.455
Alt	229.87 ± 0.938	0.338 ± 0.054	-20.274	-0.182
Diaz0	27.558 ± 2.246	0.39 ± 0.056	-1.55	-0.364
Diaz1	11.117 ± 0.775	0.351 ± 0.055	-0.029	-0.227
Jiang0	21.065 ± 1.302	0.299 ± 0.053	-0.95	-0.045
Jiang1	10.805 ± 0.360	0.286 ± 0.052	0	0
Svt	114.597 ± 17.437	0.312 ± 0.053	-9.606	-0.091
GenP	2.052 ± 0.031	0.416 ± 0.057	0.81	-0.455
GenI	20.571 ± 0.877	0.338 ± 0.054	-0.904	-0.182
Boruta	8.325 ± 0.236	0.351 ± 0.055	0.23	-0.227
GRF	413.182 ± 0.919	0.351 ± 0.055	-37.239	-0.227
GRRF	7.221 ± 0.155	0.377 ± 0.056	0.332	-0.318
Std-CPI	25.61 ± 2.535	0.377 ± 0.056	-1.37	-0.318

Effect of hyper-parameters on DBC-RCPI's performance

We studied the effect that DBC-RCPI's hyper-parameters have on its perfomance by varying the value of one between u, m, and δ, while keeping the other two fixed to their default values; we focused our analysis on the 20DLinear case and its variant with $\rho = 0.99$, and on the 11DLinear+Poly case. As expected, increasing the value of u tends to produce solutions with smaller $\bar{v}^*_{DBC-RCPI}$ and slightly larger $\bar{e}^*_{DBCI-RCPI}$ (Tables 3.29, 3.30), although the latter effect is not observed in 11DLinear+Poly (Table 3.31). Not surprisingly, the most significant decrease in $\bar{v}^*_{DBC-RCPI}$ is observed when increasing u from 0 (an extremal value, which causes the clustering objective within DBC-RCPI to be non-biased) to any positive value. This behavior is more apparent in 20DLinear with $\rho = 0.99$ and 11DLinear+Poly, where positive values of u result in a 20% improvement on $\bar{v}^*_{DBC-RCPI}$ over the same with $u = 0$, while essentially not affecting $\bar{e}^*_{DBCI-RCPI}$. The effect of m appears to be data set-specific: larger values tend to increase both $\bar{v}^*_{DBC-RCPI}$ and $\bar{e}^*_{DBCI-RCPI}$ in the two 20DLinear variants (Tables 3.32, 3.33), while, somewhat suprisingly, the opposite behavior is observed in 11DLinear+Poly (Table 3.34). By definition, larger values of δ result in smaller values of $\bar{v}^*_{DBC-RCPI}$, and, as expected, larger values of $\bar{e}^*_{DBCI-RCPI}$, causing $r_{\bar{e}^*_{DBCI-RCPI}}$ to decrease to significantly negative values (Tables 3.35, 3.36, 3.37). We notice, however, that a positive δ allows to reduce $\bar{v}^*_{DBC-RCPI}$ to values that are substantially smaller than the size of the solutions provided by benchmark methods with comparable error performance. This behavior may result useful, for example, in applications where solution cardinality is more crucial than error performance. As an overall conclusion, we note that, albeit all hyper-parameters play a significant role in determining the performance of DBC-RCPI, their default values appear near-optimal in most situations, thus, in general, relieving the user from potentially computationally expensive hyper-parameter tuning operations.

Table 3.29: 20DLinear data, and DBC-RCPI with different u values.

Method	$\bar{v}^*$	$\bar{e}^*$	$r_{\bar{v}^*}$	$r_{\bar{e}^*}$
DBC-RCPI, $u = 0$	6.81 ± 0.149	24.385 ± 0.402	0.3	0.009
DBC-RCPI, $u = 1$	6.646 ± 0.121	25.141 ± 0.448	0.317	-0.022
DBC-RCPI, $u = 2$	6.578 ± 0.138	25.18 ± 0.476	0.324	-0.024
DBC-RCPI, $u = 4$	6.52 ± 0.125	25.101 ± 0.456	0.33	-0.021

Table 3.30: 20DLinear data, variant with $\rho = 0.99$, and DBC-RCPI with different u values.

Method	$\bar{\boldsymbol{v}}^*$	$\bar{\boldsymbol{e}}^*$	$\boldsymbol{r}_{\bar{\boldsymbol{v}}^*}$	$\boldsymbol{r}_{\bar{\boldsymbol{e}}^*}$
DBC-RCPI, $u = 0$	7.408 ± 0.2	25.477 ± 0.488	0.231	0.005
DBC-RCPI, $u = 1$	5.794 ± 0.174	25.674 ± 0.467	0.399	-0.002
DBC-RCPI, $u = 2$	5.786 ± 0.196	25.931 ± 0.443	0.399	-0.012
DBC-RCPI, $u = 4$	5.73 ± 0.157	25.856 ± 0.499	0.405	-0.009

Table 3.31: 11DLinear+Poly data, and DBC-RCPI with different u values.

Method	$\bar{\boldsymbol{v}}^*$	$\bar{\boldsymbol{e}}^*$	$\boldsymbol{r}_{\bar{\boldsymbol{v}}^*}$	$\boldsymbol{r}_{\bar{\boldsymbol{e}}^*}$
DBC-RCPI, $u = 0$	18.71 ± 0.516	349.319 ± 8.815	0.751	-0.024
DBC-RCPI, $u = 1$	14.658 ± 0.419	349.35 ± 8.995	0.805	-0.024
DBC-RCPI, $u = 2$	15.058 ± 0.437	348.173 ± 8.786	0.8	-0.021
DBC-RCPI, $u = 4$	15.046 ± 0.401	348.602 ± 8.878	0.8	-0.022

Table 3.32: 20DLinear data, and DBC-RCPI different m values.

Method	$\bar{\boldsymbol{v}}^*$	$\bar{\boldsymbol{e}}^*$	$\boldsymbol{r}_{\bar{\boldsymbol{v}}^*}$	$\boldsymbol{r}_{\bar{\boldsymbol{e}}^*}$
DBC-RCPI, $m = 2$	6.61 ± 0.131	25.271 ± 0.463	0.321	-0.027
DBC-RCPI, $m = 4$	6.578 ± 0.138	25.18 ± 0.476	0.324	-0.024
DBC-RCPI, $m = 8$	6.604 ± 0.118	24.959 ± 0.444	0.321	-0.015
DBC-RCPI, $m = 16$	7.37 ± 0.204	24.986 ± 0.442	0.243	-0.016

Table 3.33: 20DLinear data, variant with $\rho = 0.99$, and DBC-RCPI with different m values.

Method	$\bar{\boldsymbol{v}}^*$	$\bar{\boldsymbol{e}}^*$	$\boldsymbol{r}_{\bar{\boldsymbol{v}}^*}$	$\boldsymbol{r}_{\bar{\boldsymbol{e}}^*}$
DBC-RCPI, $m = 2$	5.144 ± 0.152	25.995 ± 0.467	0.466	-0.015
DBC-RCPI, $m = 4$	5.786 ± 0.196	25.931 ± 0.443	0.399	-0.012
DBC-RCPI, $m = 8$	6.302 ± 0.194	25.462 ± 0.45	0.346	0.006
DBC-RCPI, $m = 16$	8.338 ± 0.252	25.606 ± 0.477	0.135	0

Table 3.34: 11DLinear+Poly data, and DBC-RCPI with different m values.

Method	$\bar{\boldsymbol{v}}^*$	$\bar{\boldsymbol{e}}^*$	$\boldsymbol{r}_{\bar{\boldsymbol{v}}^*}$	$\boldsymbol{r}_{\bar{\boldsymbol{e}}^*}$
DBC-RCPI, $m = 2$	15.378 ± 0.449	344.088 ± 8.516	0.796	-0.009
DBC-RCPI, $m = 4$	15.058 ± 0.437	348.173 ± 8.786	0.8	-0.021
DBC-RCPI, $m = 8$	14.45 ± 0.425	350.782 ± 8.788	0.808	-0.029
DBC-RCPI, $m = 16$	13.972 ± 0.38	353.908 ± 9.022	0.814	-0.038

Table 3.35: 20DLinear data, and DBC-RCPI with different δ values.

Method	$\bar{\boldsymbol{v}}^*$	$\bar{\boldsymbol{e}}^*$	$\boldsymbol{r}_{\bar{\boldsymbol{v}}^*}$	$\boldsymbol{r}_{\bar{\boldsymbol{e}}^*}$
DBC-RCPI, $\delta = 0$	6.578 ± 0.138	25.18 ± 0.476	0.324	-0.024
DBC-RCPI, $\delta = 0.25$	5.6 ± 0.113	25.596 ± 0.499	0.424	-0.041
DBC-RCPI, $\delta = 0.5$	4.832 ± 0.096	26.268 ± 0.524	0.503	-0.068
DBC-RCPI, $\delta = 1$	3.96 ± 0.074	27.613 ± 0.598	0.593	-0.123

Table 3.36: 20DLinear data, variant with $\rho = 0.99$, and DBC-RCPI with different δ values.

Method	$\bar{\boldsymbol{v}}^*$	$\bar{\boldsymbol{e}}^*$	$\boldsymbol{r}_{\bar{\boldsymbol{v}}^*}$	$\boldsymbol{r}_{\bar{\boldsymbol{e}}^*}$
DBC-RCPI, $\delta = 0$	5.786 ± 0.196	25.931 ± 0.443	0.399	-0.012
DBC-RCPI, $\delta = 0.25$	4.746 ± 0.134	26.044 ± 0.443	0.507	-0.017
DBC-RCPI, $\delta = 0.5$	4.116 ± 0.081	26.329 ± 0.465	0.573	-0.028
DBC-RCPI, $\delta = 1$	3.38 ± 0.066	27.848 ± 0.506	0.649	-0.087

Table 3.37: 11DLinear+Poly data, and DBC-RCPI with different δ values.

Method	$\bar{v}^*$	$\bar{e}^*$	$r_{\bar{v}^*}$	$r_{\bar{e}^*}$
DBC-RCPI, $\delta = 0$	15.058 ± 0.437	348.173 ± 8.786	0.8	-0.021
DBC-RCPI, $\delta = 0.25$	9.052 ± 0.235	364.168 ± 11.552	0.88	-0.068
DBC-RCPI, $\delta = 0.5$	7.288 ± 0.196	378.873 ± 12.402	0.903	-0.111
DBC-RCPI, $\delta = 1$	5.514 ± 0.169	407.768 ± 14.548	0.927	-0.196

Chapter 4

Support vector regression for polyhedral data

4.1 Introduction

Support vector regression (SVR) is a well-known supervised learning method for the estimation of an unknown function from a data set of observations, each of which represented by a point with multiple input values and one output value [32]. Such estimation is carried out via a hyperplane, which is optimally fit on the data set, possibly after a non-linear transformation of the input values by means of a kernel function [73, 74, 75]. Its remarkable performance as a predictive model has gained SVR considerable popularity in a variety of fields of application, including finance [76], transportation [77], genetics [78], neuroimaging [79], and medicine [80], among others.

After the first introduction of SVR in [74], numerous variants of the original formulation have been introduced, involving different combinations of loss functions [81, 82, 83], regularization criteria [84], robustness properties [85, 86], and approximation capabilities [87, 88]. Of particular interest for this work, however, are certain variants of SVR that have been proposed for the purpose of handling non-point data. Some of these non-point variants represent observations, and the possible uncertainty existing on their values, as stochastic

distributions with given mean and variance, and use a probabilistic approach to enforce constraints on the model's prediction errors [89, 90]. Often, these formulations focus on the special case of Normal distributions, therefore actually representing observations as ellipsoids [91, 92]. Examples of SVR models where observations are represented by triangular fuzzy numbers have also been introduced [93, 94]. Other authors have proposed to represent observations as box polyhedra, that is, hyper-rectangles whose axes of symmetry coincide with the Cartesian axes [95, 96]. Convex polyhedra have also been used to model prior knowledge, rather than information on specific observations, in the form of linear bounds imposed on the predictive hyperplane [97, 98].

In this work, we introduce "polyhedral SVR" (PSVR), a novel, computationally-efficient extension of SVR, where arbitrary convex polyhedra are used as a geometrical construct to represent non-point information (for example, uncertainty) about an observation. Unlike [95], which models observations via the Cartesian product of independent intervals, we use convex polyhedra to capture multivariate dependencies that may exist across variables in an observation. Our formulation also introduces a novel criterion for weighting prediction errors based on certain characteristics of the polyhedral observations, which may reflect, for example, the degree of uncertainty associated to such observations.

This chapter is structured as follows. Section 4.2 provides the basic background elements on the standard primal formulation of SVR. Section 4.3 introduces PSVR, our generalization of SVR targeted to observations modeled by convex polyhedra. Section 4.4 illustrates how PSVR can be applied to problems of regression with missing data; section 4.4.3 and 4.4.4 emphasize, in particular, how data-uncertainty estimates can be used towards both the construction of convex polyhedral representations of the data and the calibration of the error hyper-parameters in the formulation. Section 4.5 finally discusses the performance of PSVR on both artificial and real-world data sets, comparing results with those obtained with different benchmark methods.

4.2 Support vector regression: standard formulation

Given a data set $\boldsymbol{D}$ of n observations $(\boldsymbol{x}_i, y_i)^T$, $\boldsymbol{x}_i \in \mathbb{R}^p, y_i \in \mathbb{R}$, SVR seeks to find a hyperplane with gradient $\boldsymbol{w} \in \mathbb{R}^p$ and offset $w_0 \in \mathbb{R}$, which predicts y_i from $\boldsymbol{x}_i$ for each observation i in $\boldsymbol{D}$. In the standard variant of SVR, such predictions are required to deviate no more than a certain quantity $\epsilon \geq 0$, specified as a hyper-parameter, from the true value, while minimizing some norm (typically, the L_2-norm) of the gradient vector $\boldsymbol{w}$. This yields the following convex quadratic program:

$$\begin{aligned} \min_{\boldsymbol{w}, w_0} \ & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} \\ & y_i - \boldsymbol{w}^T\boldsymbol{x}_i - w_0 \leq \epsilon \\ & \boldsymbol{w}^T\boldsymbol{x}_i + w_0 - y_i \leq \epsilon \\ & i = 1, 2, \ldots, n \end{aligned} \tag{4.1}$$

Since program (4.1) may not be feasible, slack variables ξ_i^+ and ξ_i^- are introduced in the constraints for each observation i, and their magnitude enters the objective function as a penalty for constraint violation:

$$\begin{aligned} \min_{\boldsymbol{w}, w_0, \xi_i^+, \xi_i^-} \ & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + c\sum_{i=1}^{n}\left(\xi_i^+ + \xi_i^-\right) \\ & y_i - \boldsymbol{w}^T\boldsymbol{x}_i - w_0 \leq \xi_i^+ + \epsilon \\ & \boldsymbol{w}^T\boldsymbol{x}_i + w_0 - y_i \leq \xi_i^- + \epsilon \\ & \xi_i^+, \xi_i^- \geq 0, \ i = 1, 2, \ldots, n \end{aligned} \tag{4.2}$$

In (4.2), $c \geq 0$ is a hyper-parameter that strikes a balance between model regularization (minimization of the L_2-norm of $\boldsymbol{w}$) and model accuracy (minimization of the L_1-norm of $\boldsymbol{\xi} = \left(\xi_1^+ + \xi_1^-, \xi_2^+ + \xi_2^-, \ldots, \xi_n^+ + \xi_n^-\right)^T$). Note that, at the optimal solution, either ξ_i^+ or ξ_i^-, $i = 1, 2, \ldots, n$, can be non-zero, but not both. Note, moreover, that the enforcement of the constraints is equivalent to calculating prediction errors according to the following loss function:

$$e(y_i, \hat{y}_i) = \begin{cases} 0 & \text{if } |y_i - \hat{y}_i| < \epsilon \\ |y_i - \hat{y}_i| - \epsilon & \text{otherwise} \end{cases} \tag{4.3}$$

where $\hat{y}_i = \boldsymbol{w}^T \boldsymbol{x}_i + w_0$, and with ϵ interpretable as a measure of tolerance or robustness to noise.

4.3 A generalization of support vector regression for convex polyhedral observations

We now wish to extend (4.2), where each observation i is defined by an individual vector $(\boldsymbol{x}_i, y_i)^T$, to a new formulation where each observation i is defined by an infinite set of vectors Λ_i, given as follows:

$$\Lambda_i = \left\{ \begin{pmatrix} \boldsymbol{x} \\ y \end{pmatrix} : \boldsymbol{x} \in \mathbb{R}^p,\ y \in \mathbb{R},\ \begin{pmatrix} \boldsymbol{x} \\ y \end{pmatrix} \in P_i \right\}, \tag{4.4}$$

where P_i is a convex non-empty polyhedron defined by the intersection of a finite number of half spaces, that is:

$$P_i = \left\{ \begin{pmatrix} \boldsymbol{x} \\ y \end{pmatrix} : \boldsymbol{A}_i \begin{pmatrix} \boldsymbol{x} \\ y \end{pmatrix} \leq \boldsymbol{a}_i \right\} \tag{4.5}$$

with $\boldsymbol{A}_i \in \mathbb{R}^{m_i \times (p+1)}$, $\boldsymbol{a}_i \in \mathbb{R}^{m_i}$, $m_i > 0$. Set Λ_i, $i = 1, 2, \ldots, n$, may be viewed as an algebraic representation of the possible locations of observation i in the $\mathbb{R}^{p+1}$-dimensional space, as a result, for example, of uncertain (imprecise or not fully observed) measurements.

In order to incorporate polyhedral-set observations in the model, we propose to reformulate SVR as follows:

$$
\begin{aligned}
\min_{\boldsymbol{x},\boldsymbol{w},w_0,\xi_i} \ & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \sum_{i=1}^{n} c_i\xi_i \\
& \max_{(\boldsymbol{x},y)\in P_i} \left\{ y - \boldsymbol{w}^T\boldsymbol{x} - w_0 \right\} \leq \xi_i + \epsilon_i \\
& \max_{(\boldsymbol{x},y)\in P_i} \left\{ \boldsymbol{w}^T\boldsymbol{x} + w_0 - y \right\} \leq \xi_i + \epsilon_i \\
& \xi_i \geq 0, \ \ i = 1, 2, \ldots, n
\end{aligned}
\tag{4.6}
$$

with $c_i, \epsilon_i \geq 0, \ \ i = 1, 2, \ldots, n$, hyper-parameters, and where the prediction error on observation i is defined by the largest prediction error obtained in P_i (figure 4.1). Note that (4.2) is a special case of (4.6), by setting $c_i = c$, $\epsilon_i = \epsilon$, and assuming that the system of linear inequalities in (4.5) has a unique solution $(\boldsymbol{x}_i, y_i)^T$ for each i. It is also clear that (4.6), as it is formulated, is a harder problem to solve than (4.2), due to the non-linearity of its constraints. It is however possible to transform (4.6) to an equivalent linearly-constrained quadratic program via the following:

Theorem 4.1. *Given $\boldsymbol{A}_i$ and $\boldsymbol{a}_i$ as in (4.5), (4.6) is equivalent to*

$$
\begin{aligned}
\min_{\boldsymbol{u}_i,\boldsymbol{v}_i,\boldsymbol{w},w_0,\xi_i} \ & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \sum_{i=1}^{n} c_i\xi_i \\
& \boldsymbol{a}_i^T\boldsymbol{u}_i - w_0 \leq \xi_i + \epsilon_i \\
& \boldsymbol{a}_i^T\boldsymbol{v}_i + w_0 \leq \xi_i + \epsilon_i \\
& \boldsymbol{A}_i^T\boldsymbol{u}_i = \begin{pmatrix} -\boldsymbol{w} \\ 1 \end{pmatrix} \\
& \boldsymbol{A}_i^T\boldsymbol{v}_i = \begin{pmatrix} \boldsymbol{w} \\ -1 \end{pmatrix} \\
& \boldsymbol{u}_i, \boldsymbol{v}_i, \geq \boldsymbol{0}, \ \ \xi_i \geq 0 \\
& i = 1, 2, \ldots, n
\end{aligned}
\tag{4.7}
$$

Proof. In order to make the two max constraints in (4.6) feasible, the following systems of linear inequalities must be both infeasible:

$$\begin{cases} y - \boldsymbol{w}^T\boldsymbol{x} - w_0 > \xi_i + \epsilon_i \\ \boldsymbol{A}_i \begin{pmatrix} \boldsymbol{x} \\ y \end{pmatrix} \leq \boldsymbol{a}_i \end{cases} \tag{4.8}$$

$$\begin{cases} \boldsymbol{w}^T\boldsymbol{x} + w_0 - y > \xi_i + \epsilon_i \\ \boldsymbol{A}_i \begin{pmatrix} \boldsymbol{x} \\ y \end{pmatrix} \leq \boldsymbol{a}_i \end{cases} \tag{4.9}$$

By the affine Farkas' Lemma [99], alternative systems (4.10) and (4.11) must therefore be both feasible:

$$\boldsymbol{u}_i \geq \boldsymbol{0} \quad \text{and} \quad \begin{cases} \boldsymbol{A}_i^T\boldsymbol{u}_i = \begin{pmatrix} -\boldsymbol{w} \\ 1 \end{pmatrix} \\ \boldsymbol{a}_i^T\boldsymbol{u}_i - w_0 \leq \xi_i + \epsilon_i \end{cases} \quad \text{or} \quad \begin{cases} \boldsymbol{A}_i^T\boldsymbol{u}_i = \boldsymbol{0} \\ \boldsymbol{a}_i^T\boldsymbol{u}_i < 0 \end{cases} \tag{4.10}$$

$$\boldsymbol{v}_i \geq \boldsymbol{0} \quad \text{and} \quad \begin{cases} \boldsymbol{A}_i^T\boldsymbol{v}_i = \begin{pmatrix} \boldsymbol{w} \\ -1 \end{pmatrix} \\ \boldsymbol{a}_i^T\boldsymbol{v}_i + w_0 \leq \xi_i + \epsilon_i \end{cases} \quad \text{or} \quad \begin{cases} \boldsymbol{A}_i^T\boldsymbol{v}_i = \boldsymbol{0} \\ \boldsymbol{a}_i^T\boldsymbol{v}_i < 0 \end{cases} \tag{4.11}$$

The feasibility of (4.10) and (4.11), and the non-emptiness of P_i imply the following:

$$\begin{cases} \boldsymbol{u}_i^T \left(\boldsymbol{A}_i \begin{pmatrix} \boldsymbol{x} \\ y \end{pmatrix} - \boldsymbol{a}_i \right) \leq 0 \\ \boldsymbol{v}_i^T \left(\boldsymbol{A}_i \begin{pmatrix} \boldsymbol{x} \\ y \end{pmatrix} - \boldsymbol{a}_i \right) \leq 0 \end{cases} . \tag{4.12}$$

Now, since both

$$\begin{cases} \boldsymbol{A}_i^T\boldsymbol{u}_i = \boldsymbol{0} \\ \boldsymbol{a}_i^T\boldsymbol{u}_i < 0 \end{cases} \tag{4.13}$$

and

$$\begin{cases} \boldsymbol{A}_i^T\boldsymbol{v}_i = \boldsymbol{0} \\ \boldsymbol{a}_i^T\boldsymbol{v}_i < 0 \end{cases} \tag{4.14}$$

contradict (4.12), neither can hold true, which leads to formulation (4.7).

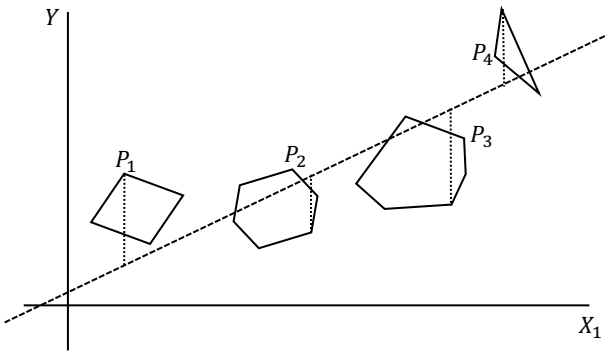

Figure 4.1: Illustration of 2-dimensional convex polyhedra and interpolating plane. The length of each dotted vertical line represents the maximum prediction error obtained in the corresponding polyhedron.

□

4.4 PSVR for regression with missing data

We consider the problem of finding an optimal interpolating hyperplane from a data matrix $\boldsymbol{D}$ with missing values; specifically, we assume that each observation i in $\boldsymbol{D}$ is missing a measurement along any q_i of the $p+1$ variables, with $0 \leq q_i \leq p$. In this application, $P_i \subset \mathbb{R}^{q_i}$ represents a set of "beliefs" or "guesses" as to what the true, unavailable coordinates of the i-th observation are. This use and interpretation of P_i raises the issue of (I) how to derive from $\boldsymbol{D}$ a "construct" of guesses on the possible true values taken on by each partially unavailable observation, as well as of (II) how such construct can be represented in convex polyhedral form. Our approach to this two-fold issue relies on multiple imputation (MI) and principal component analysis (PCA), of which we summarize the most relevant basic notions in sections 4.4.1 and 4.4.2. In section 4.4.3 we introduce a novel procedure that uses a combination of these two techniques to derive P_i as a representation of uncertainty. In section 4.4.4 we then propose a characterization of such P_i, aimed at setting reasonable values for hyper-parameters c_i and ϵ_i in (4.7).

4.4.1 Elements of multiple imputation

Our approach to the above-mentioned issue (I) builds on multiple imputation (MI) [100, 101], a Monte Carlo method for handling missing data. This method consists of three steps. In the first step, a set of plausible estimates is obtained for the missing values in the data. Each estimate in this set, which represents an individual imputation of all missing values, is a different sample of a posterior predictive distribution of the missing values. This distribution is built based on the multivariate structure of the available data, and incorporates stochastic components that characterize the variability and the uncertainty on the values to estimate. A different complete data set can therefore be obtained replacing the missing values with a different individual imputation. The choice of predictive distribution or "imputation model" used to generate the estimates depends on the type of variables to impute, as well as on the distribution that is assumed to underlie the data. This step is key to our approach to issue (I), since it provides us with a tool to generate our construct of guesses on the missing data in the form of a sampling distribution of estimates. The second and third step involve the analysis of the individual complete data sets and the pooling of the corresponding results, but we do not discuss them here since the are not relevant to our approach.

In this work, we employ a popular MI technique known as "fully conditional specification" or "chained equations" (CE) [102]. The basics elements of this technique are as follows. Let us consider a generic data set $\boldsymbol{Z}$ defined on variables $Z_1, Z_2, \ldots, Z_m$, and let Z_j^{A}, Z_j^{M} be the available and the missing portion of Z_j, $j = 1, 2, \ldots, m$, respectively. CE stochastically generates imputations in cycles, each cycle consisting of a sequence of m imputations, one for each Z_j^{M}. A total of C cycles are performed, with C chosen large enough to allow imputations to converge. Let $Z_j^{\mathrm{I},(k)}$ be the imputation of Z_j^{M} generated in the k-th repeat, and let $Z_j^{\mathrm{A,I},(k)}$ be the same as Z_j after replacing Z_j^{M} with $Z_j^{\mathrm{I},(k)}$. $Z_j^{\mathrm{I},(k)}$ is obtained as a sample from the estimated density function (predictive distribution) $f_j^{(k)}$ of Z_j conditioned on the available values of Z_j, and both the available and the imputed values of all other variables:

$$Z_j^{\mathrm{I},(k)} \sim f_j^{(k)}(Z_j | Z_1^{\mathrm{A,I},(k-1)}, Z_2^{\mathrm{A,I},(k-1)}, \ldots, Z_{j-1}^{\mathrm{A,I},(k-1)}, Z_j^{\mathrm{A}}, Z_{j+1}^{\mathrm{A,I},(k-1)}, \ldots, Z_m^{\mathrm{A,I},(k-1)}, \boldsymbol{\phi}_j^{(k)}), \tag{4.15}$$

where initial imputations $Z_j^{\mathrm{I},(0)}$ may be obtained as a random sample with replacement drawn from Z_j^A. In (4.15), the functional form of $f_j^{(k)}$ is fixed, but its parameter vector $\boldsymbol{\phi}_j^{(k)}$, obtained by regressing Z_j^{A} on $Z_1^{\mathrm{A,I},(k-1)}, Z_2^{\mathrm{A,I},(k-1)}, \ldots, Z_m^{\mathrm{A,I},(k-1)}$, varies as a function of k. For details beyond this high-level overview on CE, we refer the reader to [103, 104, 105].

4.4.2 Elements of principal component analysis

Our approach to the above-mentioned issue (II) relies on principal component analysis (PCA) [106, 107, 108]. PCA is a method aimed at algebraically and geometrically characterizing the structure of the sample covariance $\boldsymbol{S}$ of a generic data matrix $\boldsymbol{Z} \in \mathbb{R}^{r \times m}$. This characterization is given by m uncorrelated linear combinations of the m variables, called "principal components" (PC), representing directions of the m-dimensional space along which the data has the most variability. Assuming w.l.o.g. that $\boldsymbol{Z}$ has mean zero (implying $\boldsymbol{S} = \boldsymbol{Z}^T\boldsymbol{Z}$), the linear-combination coefficient vector $\boldsymbol{c}_1$ of the first PC is given by

$$\boldsymbol{c}_1 = \underset{\boldsymbol{c}^T\boldsymbol{c}=1}{\arg\max}\, \mathrm{Var}[\boldsymbol{Z}\boldsymbol{c}] = \underset{\boldsymbol{c}^T\boldsymbol{c}=1}{\arg\max}\, \boldsymbol{c}^T\boldsymbol{S}\boldsymbol{c}, \tag{4.16}$$

that is, by the unit coefficient vector that results in the linear combination of the columns of $\boldsymbol{Z}$ with the largest variance. The coefficient vector $\boldsymbol{c}_j$ of the j-th PC is then obtained iterating the following for $j = 2, 3, \ldots, m$:

$$\boldsymbol{c}_j = \underset{\substack{\boldsymbol{c}^T\boldsymbol{c}=1 \\ \mathrm{Cov}[\boldsymbol{Z}\boldsymbol{c}, \boldsymbol{Z}\boldsymbol{c}_k]=0 \\ k<j}}{\arg\max}\, \mathrm{Var}[\boldsymbol{Z}\boldsymbol{c}] = \underset{\substack{\boldsymbol{c}^T\boldsymbol{c}=1 \\ \mathrm{Cov}[\boldsymbol{c}^T\boldsymbol{S}\boldsymbol{c}, \boldsymbol{c}_k^T\boldsymbol{S}\boldsymbol{c}_k]=0 \\ k<j}}{\arg\max}\, \boldsymbol{c}^T\boldsymbol{S}\boldsymbol{c} \tag{4.17}$$

where $\boldsymbol{c}_j$ therefore corresponds to the unit coefficient vector that results in the linear combinations of the columns of $\boldsymbol{Z}$ with the largest variance, provided it is uncorrelated with all of the coefficient vectors calculated in the previous iterations. The elements of vector $\boldsymbol{c}_j$, $j = 1, 2, \ldots, m$ are called the "loadings" the j PC; the elements of vector $\boldsymbol{Z}\boldsymbol{c}_j$, $j = 1, 2, \ldots, m$ are called the "scores" of the observations in $\boldsymbol{Z}$ along the j-th PC.

Letting $(\boldsymbol{e}_1, \lambda_1), (\boldsymbol{e}_2, \lambda_2), \ldots, (\boldsymbol{e}_m, \lambda_m)$ be the eigenvector-eigenvalue pairs of S, with $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m$, it is possible to obtain vector $\boldsymbol{c}_j$ defined as in (4.16)-(4.17) by setting $\boldsymbol{c}_j = \boldsymbol{e}_j$. Equivalently, $\boldsymbol{c}_j$ can be obtained from the j-th right singular vector of $\boldsymbol{Z}$, which can, in turn, be derived from singular value decomposition

$$\boldsymbol{Z} = \boldsymbol{U}\boldsymbol{T}\boldsymbol{C}^T, \tag{4.18}$$

where $\boldsymbol{U}$ is the orthogonal $\mathbb{R}^{r \times m}$ matrix whose columns are the left singular vectors of $\boldsymbol{Z}$, $\boldsymbol{T}$ is the $\mathbb{R}^{m \times m}$ diagonal matrix whose diagonal elements $t_1 \geq t_2 \geq \ldots \geq t_m$ are the singular values of $\boldsymbol{Z}$, and $\boldsymbol{C} \in \mathbb{R}^{m \times m}$ is the orthogonal matrix whose columns are the right singular vectors of $\boldsymbol{Z}$ [32, 109, 110].

In most applications, PCA is used to obtain a low-rank approximation $\boldsymbol{Z}_{m^*} = \boldsymbol{Z}\boldsymbol{C}_{m^*}$ of $\boldsymbol{Z}$, with $\boldsymbol{C}_{m^*}$ coinciding with the matrix formed by the left-most $m^* < m$ columns of $\boldsymbol{C}$. This approximation minimizes the Frobenius norm of $\boldsymbol{Z} - \boldsymbol{Z}_{m^*}$ among all possible approximations of rank m^*, and has therefore desirable properties for data-compression applications. In this context, however, we are interested in PCA as a tool to obtain a full-rank characterization $\boldsymbol{Z}$, by means of its directions of (co-)variation $\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_m$. We will, in fact, use $\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_m$ to define the axes of a convex, hyper-rectangular bounded polyhedron that encloses a sampling distribution of guesses (obtained by MI), which we will store in matrix $\boldsymbol{Z}$. This polyhedral enclosure will involve all of $\boldsymbol{Z}$'s dimensions (columns), and will be devised to reflect the directional patterns of the data that $\boldsymbol{Z}$ contains [111]. A different $\boldsymbol{Z}$ with $m = q_i$ dimensions will be defined for each observation i. Note that the fact that any hyper-rectangle can be defined as the intersection of two half spaces per dimension implies that $\boldsymbol{A}_i \in \mathbb{R}^{2q_i \times (p+1)}$, $\boldsymbol{a}_i, \boldsymbol{u}_i, \boldsymbol{v}_i \in \mathbb{R}^{2q_i}$ in (4.7).

4.4.3 Derivation of convex polyhedra from data uncertainty

In this section, we describe in detail our proposed MI/PCA-based procedure (referred to as "bounding polyhedron estimation" (BPE) below), targeted at the derivation of convex polyhedral representations of the uncertainty that involves the missing values in a data set. Let $\boldsymbol{D}$ be an n x $(p+1)$ data matrix, containing up to p missing values per row. Let I be some MI technique (such as CE) and M be a choice of hyper-parameter values for I, including the number of MIs r, which we will assume at least as large as p. Let $s \in [0,1]$ be a "shrinkage" coefficient. Our proposed procedure for the estimation of convex polyhedron P_i from a set of MIs for each observation i in $\boldsymbol{D}$ consists of the following steps:

1. Obtain r complete n x $(p+1)$ matrices $\boldsymbol{D}_1, \boldsymbol{D}_2, \ldots \boldsymbol{D}_r$ by generating r MIs of the missing values of $\boldsymbol{D}$ with I, M.
2. Calculate the median matrix $\bar{\boldsymbol{D}}$ of $\boldsymbol{D}_1, \boldsymbol{D}_2, \ldots \boldsymbol{D}_r$.
3. Calculate $p+1$-dimensional vectors $\boldsymbol{m}_{\bar{\boldsymbol{D}}}$ and $\boldsymbol{s}_{\bar{\boldsymbol{D}}}$, corresponding to the column mean and standard deviation of $\bar{\boldsymbol{D}}$.
4. Let $\bar{\boldsymbol{D}}', \boldsymbol{D}'_1, \boldsymbol{D}'_2, \ldots \boldsymbol{D}'_r$ be the same as of $\bar{\boldsymbol{D}}, \boldsymbol{D}_1, \boldsymbol{D}_2, \ldots \boldsymbol{D}_r$ after standardization by means of $\boldsymbol{m}_{\bar{\boldsymbol{D}}}$ and $\boldsymbol{s}_{\bar{\boldsymbol{D}}}$.

For $i = 1, 2, \ldots, n$, repeat steps 5–10:

5. Let $\boldsymbol{Z}_i$ be the r x q_i matrix formed by stacking together the i-th row of each of $\boldsymbol{D}'_1, \boldsymbol{D}'_2, \ldots \boldsymbol{D}'_r$, projected down to the q_i columns corresponding to the missing values of the i-th observation in $\boldsymbol{D}$, and subsequently subtracting the column mean of this matrix from each row.
6. Calculate the PC matrix $\boldsymbol{C}_i$ of $\boldsymbol{Z}_i$, and the corresponding linear transformation $\boldsymbol{Z}^*_i = \boldsymbol{Z}_i \boldsymbol{C}_i$.
7. Calculate q_i-dimensional vectors $\boldsymbol{\lambda}_{i_{(1-s)/2}}$ and $\boldsymbol{\lambda}_{i_{(1+s)/2}}$, corresponding to the column quantiles of order $(1-s)/2$ and $(1+s)/2$ of $\boldsymbol{Z}^*_i$.

8. Let $\boldsymbol{Z}_i^{**}$ be the matrix formed by rows $\boldsymbol{n}_i^j$ of $\boldsymbol{Z}_i'$, for all $j \in \{1, 2, \ldots, q_i\}$ such that $\boldsymbol{\lambda}_{i_{(1-s)/2}} \leq \boldsymbol{n}_i^j \leq \boldsymbol{\lambda}_{i_{(1+s)/2}}$.

9. Let P_i be the intersection of the half spaces defining the smallest q_i-dimensional hyper-rectangle enclosing all observations in $\boldsymbol{Z}_i^{**}$ and with axes of symmetry parallel to the columns of $\boldsymbol{C}_i$.

10. Return P_i.

The above procedure starts with the calculation of multiply-imputed data matrices $\boldsymbol{D}_1, \boldsymbol{D}_2, \ldots, \boldsymbol{D}_r$ and of an "average" $\bar{\boldsymbol{D}}$ of those matrices; the latter is then used to center and scale the former (steps 1-4). The r MIs of the q_i missing values of observation i in $\boldsymbol{D}$ are collected from $\boldsymbol{D}_1, \boldsymbol{D}_2, \ldots \boldsymbol{D}_r$ and centered, to form matrix $\boldsymbol{Z}_i$ (step 5). After mapping the data in $\boldsymbol{Z}_i$ into PC space (step 6), the medium $s \cdot 100$ percent of the mapped data $\boldsymbol{Z}_i^*$ is retained (steps 7-8) and enclosed with the tightest bounding hyper-rectangle P_i whose axes of symmetry coincide with the axes of $\boldsymbol{Z}_i$'s PC space (step 9) (figure 4.2).

4.4.4 Definition of error cost and insensitivity from data uncertainty

We now want to define c_i and ϵ_i in (4.7) as quantities that reflect the degree of uncertainty of observation i, as modeled by its corresponding polyhedron P_i, calculated by BPE. In the special case of a singleton P_i (which typically occurs when observation i has no missing values), we simply set $c_i = c$ and $\epsilon_i = \epsilon$, where $c, \epsilon \geq 0$ are the same hyper-parameter as in the standard SVR formulation (4.2). For a general, non-singleton P_i, our procedure (referred to as "uncertainty-based cost definition" (UCD) below) is as follows:

A. Letting $\boldsymbol{\nu}_i$ be a vector containing the indices of the q_i variables with missing values in observation i, calculate an "overall" bounding hyper-rectangle $P^{\boldsymbol{\nu}_i}$ by running steps analogous to steps 6–10 of BPE on the corresponding q_i columns of $\bar{\boldsymbol{D}}'$, rather than on $\boldsymbol{Z}_i$.

B. Set and return:

$$\Psi_i = \frac{q_i}{p} \min \left(\frac{\sum_{j=1}^{q_i} l_i^j}{\sum_{j=1}^{q_i} L_{\boldsymbol{\nu}_i}^j}, 1 \right) \tag{4.19}$$

$$c_i = c^{\mathrm{u}} \left(1 - \Psi_i\right) \tag{4.20}$$

$$\epsilon_i = \epsilon + \epsilon^{\mathrm{u}} \Psi_i \tag{4.21}$$

where l_i^j is the span of hyper-rectangle P_i along the j-th dimension of its own PC space, $L_{\boldsymbol{\nu}_i}^j$ is the span of hyper-rectangle $P^{\boldsymbol{\nu}_i}$ along the j-th dimension of its own PC space, ϵ is the same as in the singleton P_i case, and $c^{\mathrm{u}}, \epsilon^{\mathrm{u}} \geq 0$ are hyper-parameters.

Note that in (4.19) if observation i_1 and observation i_2 share the same missing variables, then $P^{\boldsymbol{\nu}_{i_1}} = P^{\boldsymbol{\nu}_{i_2}}$, which implies that the computational cost of this procedure will typically be paid less than n times (figure 4.2).

With the parameter definitions given by step B, we can conclude that the objective function in (4.7) becomes less sensitive to the prediction error on observation i the larger its degree of uncertainty Ψ_i, the smaller c^{u}, and the larger ϵ^{u}. In (4.19), $\Psi_i \in [0, 1]$ is a measure of *relative* uncertainty, since q_i/p is the ratio of the number of missing variables for observation i over the theoretical maximum number of variables that may be missing, and $\frac{\sum_{j=1}^{q_i} l_i^j}{\sum_{j=1}^{q_i} L_{\boldsymbol{\nu}_i}^j}$ is the ratio between the amount of uncertainty of observation i (as measured by the sum of the univariate spans of P_i along its own PC dimensions) to the overall amount of uncertainty in the data set along the (only) q_i variables with missing values in observation i (as measured by the sum of the univariate spans of $P^{\boldsymbol{\nu}_i}$ along its own PC dimensions). Note that the min operator in (4.19) makes sure that Ψ_i be within $[0, 1]$ even in the unlikely case in which the sum of the univariate PC spans of P_i is larger than that of $P^{\boldsymbol{\nu}_i}$.

4.5 Experiments

We evaluated the performance of PSVR via 20 repeats of nested 5-fold, stratified cross-validation [58, 59]; for two only large real-world data sets, however, we decreased the number of repeats to a smaller value, due to high computational cost (see section 4.5.2 for details). For experiments on artificial data instances, the data set used in each repeat was a different independent sample from the same, given theoretical distribution. For experiments on real-world data instances, each repeat processed the same data set, unless

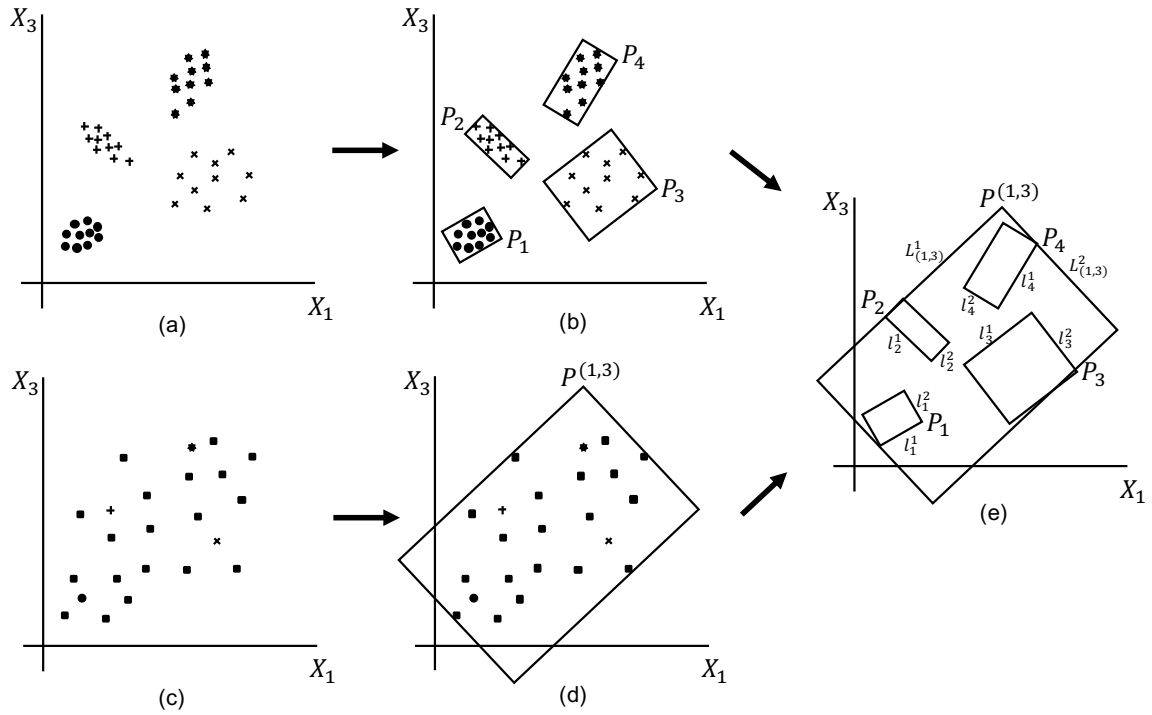

Figure 4.2: Illustration of BPE and UCD on a toy example with 4 observations with missing values along X_1 and X_3. (a): BPE generation of MIs; circles, plus signs, crosses, and stars represent the points/rows of matrices $\boldsymbol{Z}_i$, $i = 1, 2, 3, 4$, respectively. (b) BPE enclosure of $\boldsymbol{Z}_i$'s points with PC-oriented bounding rectangle P_i (assuming $s = 1$). (c): UCD projection of $\bar{\boldsymbol{D}}'$ on the X_1 and X_3 dimensions; the circle, plus sign, cross, and star, represent the median point of $\boldsymbol{Z}_i$, $i = 1, 2, 3, 4$, respectively, whereas squares represent points in the data with no missing values along X_1, X_3. (d): UCD enclosure of $\bar{\boldsymbol{D}}'$ with oriented bounding rectangle $P^{(1,3)}$. (e): UCD calculation of Ψ_i, c_i, and ϵ_i based on the univariate spans of P_i (denoted by the length of sides l_i^1 and l_i^2) and of $P^{(1,3)}$ (denoted by the length of sides $L^1_{(1,3)}$ and $l^2_{(1,3)}$).

differently specified below. For each outer fold we used the inner training-validation folds to tune hyper-parameters c and c^{u} (both with possible values 0, 0.05, 0.1, 0.5, 1, 2, 5), ϵ and ϵ^{u} (both with possible values 0, 0.25, 0.5, 1), and s (with possible values 0, 0.0001, 0.001, 0.01, 0.1, 0.25, 0.5, 0.75, 1). We carried out hyper-parameter tuning once for each of several prediction error measures: root mean square error, mean absolute error, and quantiles of order 0.8 and 1 (maximum) of the absolute error, calculated on all observations ($e_{rms}, e_{ma}, e_{q.8a}, e_{maxa}$, respectively); the same calculated on the certain (missing-value-free) observations only ($e^c_{rms}, e^c_{ma}, e^c_{q.8a}, e^c_{maxa}$, respectively); and the same calculated on the uncertain (with missing values) observations only ($e^u_{rms}, e^u_{ma}, e^u_{q.8a}, e^u_{maxa}$, respectively). For each error measure, we selected the hyper-parameter values that yielded the smallest mean of such measure's values, calculated across the inner-fold validation data sets. We then trained the corresponding PSVR on the training data of the outer fold, and applied the same error measure to the testing predictions. We finally obtained a testing grand mean for such error measure, calculated across all outer folds of all cross-validation repeats. In section 4.5.3, we will report these grand means as $\bar{e}_{rms}, \bar{e}_{ma}, \bar{e}_{q.8a}, \bar{e}_{maxa}, \bar{e}^c_{rms}, \bar{e}^c_{ma}, \bar{e}^c_{q.8a}, \bar{e}^c_{maxa}$ $\bar{e}^u_{rms}, \bar{e}^u_{ma}, \bar{e}^u_{q.8a}, \bar{e}^u_{maxa}$. ; moreover, we will refer to $\bar{e}_{rms}, \bar{e}_{ma}, \bar{e}_{maxa}$ as a whole as the "$\bar{e}_{\cdot}$ measures", to $\bar{e}^c_{rms}, \bar{e}^c_{ma}, \bar{e}^c_{maxa}$ as a whole as the "$\bar{e}^c_{\cdot}$ measures", and to $\bar{e}^u_{rms}, \bar{e}^u_{ma}, \bar{e}^u_{maxa}$ as the "$\bar{e}^u_{\cdot}$ measures".

We obtained $r = 40$ MIs via $C = 20$ cycles of CE for each uncertain training observation, using a predictive-mean-matching imputation model [112, 113], as implemented in the `mice` function of the homonymous R package [114], using the training data only as the `data` input to the function, and setting `m=40`, `maxit=20`, and `method='pmm'`, We enclosed such MIs with PC hyper-rectangles computed via the `prcomp` function of the `stats` R package, setting `center=TRUE` and `scale.=TRUE`, and trained the resulting PSVR model with the Gurobi software, version 7.5.2, through the interface provided by the `gurobi` function of the homonymous R package, after standardizing the data to zero mean and unit variance. With the same approach, we then obtained MIs for the validation data sets, providing both the training and the validation data as input to the `mice` function, and their enclosing PC hyper-rectangles. We obtained output predictions for the uncertain validation data by computing the value of the fitting hyperplane at the median point of the corresponding PC

hyper-rectangle, as calculated in PC space. We used the same approach to process uncertain testing observations, with the exception of their MIs, which we obtained providing all the data to the `mice` function.

We employed the same cross-validation procedure as above to test four benchmark methods, all of which implemented in R and solved with Gurobi as for PSVR. The first benchmark method consists of a standard SVR (with formulation as in (4.2)), trained on the median of the r MIs calculated in PC space (MSVR), that is, a model that disregards the uncertainty represented by the distribution of those MIs. Note that MSVR is equivalent to PSVR with $s = 0$, $c_i = c$, and $\epsilon^{\mathrm{u}} = 0$. The second benchmark method consists, again, of a standard SVR, but this time trained on the only certain observations (CSVR). Note that CSVR is equivalent to PSVR with $c_i = c$ and $\epsilon^{\mathrm{u}} \to \infty$. For the calculation of $\bar{e}^u_{rms}, \bar{e}^u_{ma}, \bar{e}^u_{q.8a}$, and $\bar{e}^u_{maxa}$ we used CSVR models with hyper-parameters tuned via $e^c_{rms}, e^c_{ma}, e^c_{q.8a}$, and e^c_{maxa}, respectively. For both MSVR and CSVR, the possible values for the two only hyper-parameters c and ϵ were the same as for PSVR. The remaining two benchmark methods are the two maximum-distance variants of Carrizosa's et al.'s non-oriented box polytope SVRs. One variant relies on box polytopes estimated from the mean and standard deviation of the available data (BoxSD) and the other one on box polytopes estimated from the quantiles of the available data (BoxQ). Both include a "shrinkage" hyper-parameter (called k in BoxSD and $2a$ in BoxQ), which we tuned using the same possible values as s. For both variants, we obtained output-value predictions for validation and testing observations as in in [95], to which we refer for further details. As mentioned in [95], setting $k = 0$ and $2a = 1$ makes BoxSD and BoxQ equivalent to a standard SVR formulation where each missing value is replaced with a single univariate imputation, equal to the mean or median of the available values along the corresponding missing variable, respectively. Therefore, our experiments also implicitly include these two single-imputation, standard SVR benchmarks.

4.5.1 Artificial data

We employed artificial data sets $\boldsymbol{D}$ of $n = 100$ independent observations on $p = 10$ input variables, generated from a linear true model with form (3.12), setting $\beta_0 = 0, \beta_1 = 1, \beta_2 =$

$2, \ldots, \beta_{10} = 10$, correlation matrix $\boldsymbol{R}$ with $\rho_{X_j X_k} = \rho$, for all $j \neq k$, and $R^2 = 0.9$. We considered scenarios with either low ($\rho = 0.3$) or high ($\rho = 0.9$) inter-variable correlations. We artificially injected missingness into the data by removing a proportion of $\pi_2 p$ of the p input values from a (expected) proportion of π_1 of the n observations, with $\pi_1, \pi_2 \in (0, 1)$ parameters, enforcing a missing-at-random (MAR) mechanism, where the probability of a value to be missing depends on the only available values of an observation [115]. We considered scenarios with either low ($\pi_1 = \pi_2 = 0.3$), medium ($\pi_1 = \pi_2 = 0.6$), or high ($\pi_1 = \pi_2 = 0.9$) missingness. For a given choice of $p + 1$-dimensional vector parameter $\boldsymbol{\gamma}$, our procedure to inject MAR missingness into the data is as follows:

1. For each observation i, $i = 1, 2, \ldots, n$, sample uniformly at random without replacement $\lfloor \pi_2 p + 0.5 \rfloor$ input-variable indices from $\{1, 2, \ldots, p\}$. Let $\boldsymbol{\nu}_i$ be the vector containing such indices.

2. Let $\boldsymbol{d}_i$ be the i-th row of the standardized $\boldsymbol{D}$. For all $j \in \boldsymbol{\nu}_i$, replace the j-th element of $\boldsymbol{d}_i$ with 0. Let $\tilde{\boldsymbol{D}}$ be the resulting matrix, and $\tilde{\boldsymbol{d}}_i$ its i-th row.

3. Find the unique root δ^* of equation

$$\boldsymbol{e}^T \left[e^{[-(\tilde{\boldsymbol{D}}\boldsymbol{\gamma} + \delta \boldsymbol{e})]} + \boldsymbol{e} \right]^{-1} - n\pi_1 = 0, \tag{4.22}$$

   where $\boldsymbol{e}$ is an n-dimensional vector of ones, and $e^{[.]}$ and $[.]^{-1}$ are exponential and power functions applied element-wise to their argument.

4. For each observation i, $i = 1, 2, \ldots, n$, remove the values of all variables $j \in \boldsymbol{\nu}_i$ with probability

$$\pi_{1,i} = \frac{e^{\boldsymbol{\gamma}^T \tilde{\boldsymbol{d}}_i + \delta^*}}{e^{\boldsymbol{\gamma}^T \tilde{\boldsymbol{d}}_i + \delta^*} + 1}. \tag{4.23}$$

In our MAR-injecting procedure, the probability of an observation to contain missing values varies across observations, as a function $\pi_{1,i}$ of an observation i's values and of parameters $\boldsymbol{\gamma}$ and δ^* (the latter, calculated in step 3, based on the chosen $\boldsymbol{\gamma}$). Step 2 ensures that

the missingness mechanism is truly MAR, by making $\pi_{1,i}$ depend on the only values of observation i that will certainly *not* be artificially removed, that is those whose corresponding variable indices do not appear in $\boldsymbol{\nu}_i$ (since in our experiments we defined vector $\boldsymbol{\gamma}$ as $(1, 2, \ldots, p, 0)^T$, $\pi_{1,i}$ was also not a function of the output value). Building on Theorem C.1 in Appendix C, the value of δ^* calculated in step 3 finally ensures that the expected proportion of observations with missing values is the desired π_1.

4.5.2 Real-world data

We used three fully-numeric real-world data sets for our experiments:

- BostonHousing: originally with $n = 506$ observations and $p = 13$ input variables, but reduced to $p = 12$, by removing binary variable "CHAS", and to $n = 100$ in our study (see also section 3.3.2). The reduced data set of $n = 100$ observations corresponds to a random sample of the original data set, with a different independent sample drawn for each nested cross-validation repeat. We artificially injected missingness into each of these samples using the same procedure and the same scenarios with either low ($\pi_1 = \pi_2 = 0.3$), medium ($\pi_1 = \pi_2 = 0.6$), or high ($\pi_1 = \pi_2 = 0.9$) missingness as for the artificial data (see section 4.5.1).

- Ozone: with $n = 366$ observations and $p = 12$ input variables. A total of 163 observations contain 1 or more missing values along the input variables, with most of them missing only 1 (see also section 3.3.2). Due to the large computational cost, we decreased the number of nested cross-validation repeats to 4 when processing this data set.

- Boys: with $n = 748$ observations and $p = 9$ input variables. A total of 525 observations contain 1 or more missing values along the input variables, with most of them missing 3 [114]. Due to the large computational cost, we decreased the number of nested cross-validation repeats to 2 when processing this data set.

4.5.3 Results

Artificial data

The experiments on the artificial data sets overall suggest that the amount of missing values in the data, as implied by π_1, π_2, plays a key role in explaining how each method performs and ranks with respect to the others. Less clear appears to be the role of variable correlation ρ. Notably, PSVR appears to be superior to all other methods when missingness is massive ($\pi_1, \pi_2 = 0.9$). As π_1, π_2 decrease, PSVR's performance becomes closer to that of MSVR, and, for low values π_1, π_2, also to that of CSVR, although PSVR appears to remain superior according to error measures $\bar{e}^c_{max}$ and/or $\bar{e}^u_{max}$. Interestingly, BoxSD and BoxQ perform worse than PSVR for any value of π_1, π_2, when errors are calculated on all observations (which include both certain and uncertain observations) or on the uncertain observations only. Details specific to each data set are presented below.

Normal *data set with low missingness* ($\pi_1, \pi_2 = 0.3$), Tables 4.1, 4.2: PSVR, MSVR, and CSVR show comparable error performance when errors are calculated on all observations ($\bar{e}_\cdot$ measures), or on the certain observations only ($\bar{e}^c_\cdot$ measures), although PSVR appears to be superior according to $\bar{e}^c_{maxa}$ when ρ is high. MSVR (as well as CSVR, if ρ is high), however, yields slighty better performance than PSVR when errors are calculated on the uncertain observations only ($\bar{e}^u_\cdot$ measures). Both BoxSD and BoxQ perform similarly to the other three methods according to $\bar{e}^c_\cdot$, but substantially worse according to $\bar{e}_\cdot$ and even more so according to $\bar{e}^u_\cdot$.

Normal *data set with medium missingness* ($\pi_1, \pi_2 = 0.6$), Tables 4.3, 4.4: PSVR performs overall comparably to MSVR if ρ is high and slightly worse than MSVR if ρ is low, although in the latter case PSVR appears to be superior to MSVR according to $\bar{e}^c_{maxa}$ and $\bar{e}^u_{maxa}$. CSVR performs worse than PSVR and MSVR, except according to some of the $\bar{e}^c_\cdot$ measures when ρ is high. Both BoxSD and BoxQ perform significantly worse than the other methods, especially according to $\bar{e}_\cdot$ and $\bar{e}^u_\cdot$.

Normal *data set with high missingness* ($\pi_1, \pi_2 = 0.9$), Tables 4.5, 4.6: PSVR is clearly superior to all other methods, according to any error measure, if ρ is low. With high ρ,

PSVR performs slightly better than MSVR according to $\bar{e}^u_{\cdot}$ (and especially $\bar{e}^u_{maxa}$), more or less comparably to MSVR according to $\bar{e}_{\cdot}$, and slightly worse than MSVR according to $\bar{e}^c_{\cdot}$. CSVR performs worse than PSVR and MSVR, except according to $\bar{e}^c_{\cdot}$ when ρ is low. Both BoxSD and BoxQ perform overall significantly worse than the other methods, with some exceptions for BoxSD according to $\bar{e}^c_{\cdot}$ (comparably to PSVR if ρ is low, and between PSVR and CSVR if ρ is high).

Table 4.1: Normal data, $\rho = 0.3$, $\pi_1 = \pi_2 = 0.3$

Error Measure	PSVR	MSVR	CSVR	BoxSD	BoxQ
$\bar{e}_{rms}$	12.587 ± 0.217	12.137 ± 0.159	12.964 ± 0.194	15.283 ± 0.173	15.287 ± 0.173
$\bar{e}_{ma}$	10.133 ± 0.213	9.848 ± 0.136	10.729 ± 0.175	12.534 ± 0.126	12.529 ± 0.117
$\bar{e}_{q.8a}$	15.201 ± 0.225	15.227 ± 0.227	16.186 ± 0.239	18.336 ± 0.268	18.468 ± 0.287
$\bar{e}_{maxa}$	28.436 ± 0.423	28.335 ± 0.556	28.003 ± 0.6	34.951 ± 0.765	34.078 ± 0.799
$\bar{e}^c_{rms}$	13.616 ± 0.21	13.078 ± 0.135	13.487 ± 0.146	13.459 ± 0.175	13.601 ± 0.162
$\bar{e}^c_{ma}$	10.98 ± 0.14	10.842 ± 0.129	11.111 ± 0.14	11.144 ± 0.14	10.997 ± 0.13
$\bar{e}^c_{q.8a}$	16.272 ± 0.306	16.872 ± 0.295	17.062 ± 0.42	17.198 ± 0.286	17.035 ± 0.295
$\bar{e}^c_{maxa}$	27.412 ± 0.402	27.367 ± 0.299	26.933 ± 0.386	28.042 ± 0.289	27.43 ± 0.328
$\bar{e}^u_{rms}$	10.845 ± 0.431	9.943 ± 0.404	11.155 ± 0.428	18.295 ± 0.362	18.058 ± 0.406
$\bar{e}^u_{ma}$	8.865 ± 0.333	8.283 ± 0.338	9.15 ± 0.35	15.363 ± 0.385	15.61 ± 0.401
$\bar{e}^u_{q.8a}$	13.178 ± 0.634	11.734 ± 0.44	13.652 ± 0.587	21.286 ± 0.562	21.481 ± 0.479
$\bar{e}^u_{maxa}$	18.059 ± 0.854	17.39 ± 0.819	19.526 ± 0.676	30.852 ± 0.749	31.122 ± 0.758

Table 4.2: Normal data, $\rho = 0.9$, $\pi_1 = \pi_2 = 0.3$

Error Measure	PSVR	MSVR	CSVR	BoxSD	BoxQ
$\bar{e}_{rms}$	19.06 ± 0.269	19.343 ± 0.256	19.118 ± 0.252	21.55 ± 0.271	21.723 ± 0.331
$\bar{e}_{ma}$	15.543 ± 0.262	15.788 ± 0.259	15.733 ± 0.197	17.69 ± 0.269	17.418 ± 0.235
$\bar{e}_{q.8a}$	23.685 ± 0.417	25.032 ± 0.41	24.859 ± 0.465	26.584 ± 0.456	26.335 ± 0.391
$\bar{e}_{maxa}$	41.352 ± 0.648	41.834 ± 0.719	42.276 ± 0.963	47.312 ± 0.917	49.31 ± 1.107
$\bar{e}^c_{rms}$	18.901 ± 0.323	19.302 ± 0.318	19.486 ± 0.407	18.765 ± 0.331	19.231 ± 0.34
$\bar{e}^c_{ma}$	15.523 ± 0.215	15.785 ± 0.276	15.805 ± 0.305	15.418 ± 0.247	15.661 ± 0.269
$\bar{e}^c_{q.8a}$	24.059 ± 0.583	23.921 ± 0.519	23.586 ± 0.525	23.889 ± 0.349	24.077 ± 0.375
$\bar{e}^c_{maxa}$	37.761 ± 0.839	39.171 ± 0.936	40.45 ± 1.172	38.603 ± 0.778	39.201 ± 0.915
$\bar{e}^u_{rms}$	20.692 ± 0.617	19.803 ± 0.522	19.763 ± 0.473	24.728 ± 0.739	24.282 ± 0.58
$\bar{e}^u_{ma}$	17.219 ± 0.65	16.31 ± 0.502	16.092 ± 0.457	20.206 ± 0.622	20.313 ± 0.602
$\bar{e}^u_{q.8a}$	26.067 ± 1.094	24.218 ± 0.844	23.411 ± 0.777	29.293 ± 0.917	29.458 ± 0.965
$\bar{e}^u_{maxa}$	35.536 ± 0.861	33.882 ± 0.794	33.941 ± 0.917	43.438 ± 1.251	42.14 ± 1.084

Table 4.3: Normal data, $\rho = 0.3$, $\pi_1 = \pi_2 = 0.6$

Error Measure	PSVR	MSVR	CSVR	BoxSD	BoxQ
$\bar{e}_{rms}$	11.835 ± 0.144	11.537 ± 0.2	13.248 ± 0.316	20.7 ± 0.264	20.811 ± 0.238
$\bar{e}_{ma}$	9.186 ± 0.213	9.039 ± 0.196	10.46 ± 0.244	16.86 ± 0.16	16.763 ± 0.199
$\bar{e}_{q.8a}$	14.307 ± 0.219	13.922 ± 0.28	15.952 ± 0.394	25.106 ± 0.387	25.469 ± 0.379
$\bar{e}_{maxa}$	28.065 ± 0.472	28.102 ± 0.55	31.94 ± 0.771	47.261 ± 0.774	47.734 ± 0.707
$\bar{e}^c_{rms}$	14.608 ± 0.293	14.249 ± 0.263	15.492 ± 0.36	15.591 ± 0.265	15.89 ± 0.308
$\bar{e}^c_{ma}$	11.441 ± 0.164	11.743 ± 0.258	12.809 ± 0.293	12.792 ± 0.193	13.025 ± 0.38
$\bar{e}^c_{q.8a}$	17.174 ± 0.399	17.563 ± 0.405	18.328 ± 0.592	18.88 ± 0.334	19.606 ± 0.375
$\bar{e}^c_{maxa}$	26.147 ± 0.583	27.574 ± 0.425	29.864 ± 0.6	28.712 ± 0.958	29.37 ± 0.841
$\bar{e}^u_{rms}$	8.868 ± 0.336	8.853 ± 0.251	11.472 ± 0.502	23.219 ± 0.292	23.393 ± 0.338
$\bar{e}^u_{ma}$	7.327 ± 0.263	7.084 ± 0.207	9.011 ± 0.4	19.42 ± 0.234	19.658 ± 0.268
$\bar{e}^u_{q.8a}$	10.933 ± 0.453	10.596 ± 0.393	13.873 ± 0.631	29.157 ± 0.347	29.154 ± 0.442
$\bar{e}^u_{maxa}$	18.501 ± 0.627	19.223 ± 0.646	24.935 ± 1.24	46.564 ± 0.751	45.972 ± 1.003

Table 4.4: Normal data, $\rho = 0.9$, $\pi_1 = \pi_2 = 0.6$

Error Measure	PSVR	MSVR	CSVR	BoxSD	BoxQ
$\bar{e}_{rms}$	18.098 ± 0.206	18.649 ± 0.248	19.401 ± 0.254	25.578 ± 0.225	25.626 ± 0.252
$\bar{e}_{ma}$	14.613 ± 0.202	14.91 ± 0.245	15.921 ± 0.251	20.691 ± 0.138	20.39 ± 0.18
$\bar{e}_{q.8a}$	23.678 ± 0.446	23.179 ± 0.453	24.761 ± 0.487	31.942 ± 0.25	31.99 ± 0.459
$\bar{e}_{maxa}$	41.261 ± 0.615	41.287 ± 0.809	43.288 ± 0.85	57.71 ± 0.68	58.824 ± 0.754
$\bar{e}^c_{rms}$	19.041 ± 0.413	20.304 ± 0.525	20.458 ± 0.468	21.868 ± 0.436	21.591 ± 0.383
$\bar{e}^c_{ma}$	15.91 ± 0.37	16.879 ± 0.512	16.68 ± 0.443	17.824 ± 0.394	18.655 ± 0.393
$\bar{e}^c_{q.8a}$	24.022 ± 0.669	24.6 ± 0.704	24.659 ± 0.675	26.178 ± 0.427	25.772 ± 0.499
$\bar{e}^c_{maxa}$	37.574 ± 1.091	37.019 ± 1.136	38.339 ± 0.946	41.231 ± 0.613	41.233 ± 0.797
$\bar{e}^u_{rms}$	16.977 ± 0.26	16.735 ± 0.391	18.481 ± 0.429	27.427 ± 0.341	26.247 ± 0.429
$\bar{e}^u_{ma}$	13.191 ± 0.254	13.329 ± 0.292	15.383 ± 0.348	21.26 ± 0.297	21.458 ± 0.393
$\bar{e}^u_{q.8a}$	20.418 ± 0.417	20.581 ± 0.413	23.512 ± 0.635	31.938 ± 0.524	32.928 ± 0.593
$\bar{e}^u_{maxa}$	35.514 ± 0.77	34.441 ± 0.918	37.973 ± 1.029	56.602 ± 0.718	55.663 ± 0.774

Table 4.5: Normal data, $\rho = 0.3$, $\pi_1 = \pi_2 = 0.9$

Error Measure	PSVR	MSVR	CSVR	BoxSD	BoxQ
$\bar{e}_{rms}$	18.816 ± 0.903	19.821 ± 0.994	28.873 ± 1.156	33.197 ± 0.554	33.183 ± 0.446
$\bar{e}_{ma}$	14.228 ± 0.696	15.414 ± 0.754	23.512 ± 1.029	26.534 ± 0.423	26.431 ± 0.416
$\bar{e}_{q.8a}$	21.511 ± 0.839	23.307 ± 1.162	37.164 ± 1.399	41.792 ± 0.686	42.024 ± 0.605
$\bar{e}_{maxa}$	46.462 ± 1.783	47.552 ± 1.876	64.517 ± 2.595	71.653 ± 0.604	74.654 ± 1.129
$\bar{e}^c_{rms}$	23.724 ± 1.14	29.513 ± 1.98	24.481 ± 1.095	26.863 ± 1.471	23.851 ± 1.115
$\bar{e}^c_{ma}$	21.466 ± 0.997	26.407 ± 1.903	22.672 ± 0.899	23.601 ± 1.105	20.661 ± 0.866
$\bar{e}^c_{q.8a}$	25.871 ± 1.431	31.265 ± 2.162	27.538 ± 1.227	29.62 ± 1.447	26.768 ± 1.312
$\bar{e}^c_{maxa}$	29.313 ± 1.482	35.295 ± 1.864	31.271 ± 1.366	34.414 ± 1.982	29.784 ± 1.241
$\bar{e}^u_{rms}$	15.691 ± 0.619	16.616 ± 0.746	29.6 ± 1.175	33.788 ± 0.548	34.047 ± 0.469
$\bar{e}^u_{ma}$	12.288 ± 0.45	13.942 ± 0.77	23.087 ± 1.127	26.966 ± 0.463	26.849 ± 0.417
$\bar{e}^u_{q.8a}$	18.608 ± 0.568	21.11 ± 1.169	36.805 ± 1.429	41.981 ± 0.65	41.584 ± 0.708
$\bar{e}^u_{maxa}$	35.034 ± 1.267	39.233 ± 1.759	61.876 ± 2.619	71.482 ± 0.547	70.881 ± 0.701

Table 4.6: Normal data, $\rho = 0.9$, $\pi_1 = \pi_2 = 0.9$

Error Measure	PSVR	MSVR	CSVR	BoxSD	BoxQ
$\bar{e}_{rms}$	16.705 ± 0.469	16.73 ± 0.337	26.429 ± 1.015	43.893 ± 0.529	45.756 ± 0.443
$\bar{e}_{ma}$	12.602 ± 0.35	12.763 ± 0.268	21.155 ± 0.874	34.697 ± 0.498	37.4 ± 0.425
$\bar{e}_{q.8a}$	18.496 ± 0.624	18.845 ± 0.503	30.799 ± 1.413	56.285 ± 0.846	56.007 ± 0.912
$\bar{e}_{maxa}$	41.338 ± 1.467	40.429 ± 0.996	61.475 ± 2.851	100.717 ± 1.739	110.869 ± 3.476
$\bar{e}^c_{rms}$	21.957 ± 1.085	20.495 ± 0.982	24.454 ± 1.522	22.102 ± 1.08	29.718 ± 1.131
$\bar{e}^c_{ma}$	18.941 ± 1.037	18.633 ± 0.973	21.295 ± 1.433	21.844 ± 1.011	26.107 ± 1.178
$\bar{e}^c_{q.8a}$	23.813 ± 1.159	22.443 ± 1.152	26.421 ± 1.758	24.737 ± 1.469	33.731 ± 1.42
$\bar{e}^c_{maxa}$	25.347 ± 1.327	25.478 ± 1.229	29.061 ± 2.165	28.853 ± 1.672	41.779 ± 1.924
$\bar{e}^u_{rms}$	14.539 ± 0.289	15.455 ± 0.354	26.665 ± 1.209	45.445 ± 0.633	46.4 ± 0.541
$\bar{e}^u_{ma}$	11.981 ± 0.291	12.179 ± 0.274	21.716 ± 0.74	36.057 ± 0.563	36.935 ± 0.445
$\bar{e}^u_{q.8a}$	17.641 ± 0.533	17.679 ± 0.508	30.87 ± 1.519	57.027 ± 0.832	57.113 ± 0.628
$\bar{e}^u_{maxa}$	32.261 ± 0.786	36.089 ± 0.645	60.29 ± 2.954	98.869 ± 1.621	111.421 ± 3.524

Real-world data

The experiments on the real-world data sets generally confirm the main findings of the experiments on the artificial data sets, especially with respect to the dependence of each method's performance on the amount of missing values in the data. Details specific to each data set are presented below.

BostonHousing *data set with low missingness* ($\pi_1, \pi_2 = 0.3$), Table 4.7: PSVR and MSVR exhibit overall similar performance, slightly better than CSVR according to $\bar{e}_{\cdot}$ and $\bar{e}^c_{\cdot}$, and more substantially so according to $\bar{e}^u_{\cdot}$. Both BoxSD and BoxQ perform similarly to PSVR and MSVR according to $\bar{e}^c_{\cdot}$, and similarly to CSVR according to $\bar{e}_{\cdot}$ and $\bar{e}^u_{\cdot}$.

BostonHousing *data set with medium missingness* ($\pi_1, \pi_2 = 0.6$), Table 4.8: The patterns are somewhat analogous to those observed with $\pi_1, \pi_2 = 0.3$, except for two facts: CSVR here performs similarly to PSVR and MSVR according to $\bar{e}^c_{\cdot}$; BoxSD and Box perform worse than CSVR according to $\bar{e}_{\cdot}$ and $\bar{e}^u_{\cdot}$.

BostonHousing *data set with high missingness* ($\pi_1, \pi_2 = 0.9$), Table 4.9: PSVR is slightly superior to MSVR according to $\bar{e}_{rms}$, $\bar{e}_{q.8a}$, $\bar{e}_{maxa}$, and $\bar{e}^c_{q.8a}$, whereas the two methods perform similarly according to the remaining error measures. CSVR performs worse than PSVR anb MSVR according to $\bar{e}_{\cdot}$ and $\bar{e}^u_{\cdot}$, but, somewhat surprisingly, better than both of them according to $\bar{e}^c_{\cdot}$. BoxSD and BoxQ perform roughly the same as PSVR and MSVR according to $\bar{e}^c_{\cdot}$, but substantially worse than all other methods according to the other error measures.

Ozone *data set*, Table 4.10: The performance is roughly equivalent across all methods. This is most likely due to the very limited amount of missing values in the data. Indeed, the fact that CSVR, which ignores observations with missing values, yields overall the same level of error as the other methods, which do process observations with missing values, seems to suggests that there is no benefit in learning from uncertain data when a substantial majority of the data is certain. This is essentially in accordance with what we observed on the Normal and BostonHousing data sets with $\pi_1, \pi_2 = 0.3$.

Boys *data set*, Table 4.11: PSVR and MSVR perform quite similarly overall, although there is some evidence of PSVR's superiority according to a few error measures ($\bar{e}_{rms}$, $\bar{e}^c_{maxa}$, and $\bar{e}^u_{maxa}$). CSVR performs about the same as PSVR according to $\bar{e}^c_{\cdot}$, but worse than PSVR (and MSVR) according to the other error measures. BoxSD and BoxQ perform overall substantially worse than both PSVR and MSVR; compared to CSVR, they underperform according to $\bar{e}^c_{\cdot}$ and overperform according to the other error measures.

Table 4.7: BostonHousing data, $\pi_1 = \pi_2 = 0.3$

Error Measure	PSVR	MSVR	CSVR	BoxSD	BoxQ
$\bar{e}_{rms}$	5.104 ± 0.127	5.077 ± 0.118	5.47 ± 0.119	5.653 ± 0.104	5.71 ± 0.115
$\bar{e}_{ma}$	3.565 ± 0.077	3.59 ± 0.083	3.878 ± 0.073	3.982 ± 0.066	3.996 ± 0.074
$\bar{e}_{q.8a}$	5.263 ± 0.136	5.148 ± 0.148	5.889 ± 0.15	5.778 ± 0.09	6.002 ± 0.109
$\bar{e}_{maxa}$	14.544 ± 0.419	14.275 ± 0.46	14.751 ± 0.443	16.307 ± 0.404	16.952 ± 0.54
$\bar{e}^c_{rms}$	4.365 ± 0.095	4.592 ± 0.137	4.74 ± 0.183	4.567 ± 0.124	4.714 ± 0.111
$\bar{e}^c_{ma}$	3.226 ± 0.073	3.232 ± 0.084	3.369 ± 0.1	3.269 ± 0.064	3.337 ± 0.062
$\bar{e}^c_{q.8a}$	4.799 ± 0.083	4.829 ± 0.137	4.893 ± 0.104	4.948 ± 0.075	5.019 ± 0.098
$\bar{e}^c_{maxa}$	11.386 ± 0.39	11.299 ± 0.363	12.111 ± 0.633	11.677 ± 0.427	11.94 ± 0.445
$\bar{e}^u_{rms}$	5.331 ± 0.226	5.286 ± 0.228	6.604 ± 0.245	6.7 ± 0.303	6.543 ± 0.295
$\bar{e}^u_{ma}$	4.041 ± 0.164	4.209 ± 0.177	5.251 ± 0.181	5.367 ± 0.24	5.368 ± 0.234
$\bar{e}^u_{q.8a}$	5.189 ± 0.255	5.762 ± 0.288	7.462 ± 0.315	7.096 ± 0.361	7.057 ± 0.275
$\bar{e}^u_{maxa}$	10.381 ± 0.49	10.32 ± 0.515	11.778 ± 0.438	12.625 ± 0.559	12.954 ± 0.644

Table 4.8: BostonHousing data, $\pi_1 = \pi_2 = 0.6$

Error Measure	PSVR	MSVR	CSVR	BoxSD	BoxQ
$\bar{e}_{rms}$	5.061 ± 0.187	5.146 ± 0.192	5.524 ± 0.194	6.232 ± 0.073	6.195 ± 0.098
$\bar{e}_{ma}$	3.34 ± 0.075	3.636 ± 0.107	4.006 ± 0.129	4.39 ± 0.053	4.385 ± 0.068
$\bar{e}_{q.8a}$	5.151 ± 0.134	5.103 ± 0.129	5.668 ± 0.216	6.418 ± 0.13	6.539 ± 0.127
$\bar{e}_{maxa}$	14.654 ± 0.595	14.772 ± 0.647	15.534 ± 0.683	17.03 ± 0.513	17.778 ± 0.529
$\bar{e}^c_{rms}$	5.07 ± 0.204	4.952 ± 0.147	5.184 ± 0.198	4.92 ± 0.141	5.154 ± 0.131
$\bar{e}^c_{ma}$	3.754 ± 0.135	4.057 ± 0.158	3.877 ± 0.138	3.653 ± 0.107	3.871 ± 0.119
$\bar{e}^c_{q.8a}$	5.934 ± 0.267	5.773 ± 0.172	5.588 ± 0.247	5.253 ± 0.156	5.605 ± 0.197
$\bar{e}^c_{maxa}$	11.067 ± 0.402	10.558 ± 0.255	11.058 ± 0.519	10.901 ± 0.351	10.826 ± 0.34
$\bar{e}^u_{rms}$	4.847 ± 0.255	4.827 ± 0.216	5.31 ± 0.245	6.555 ± 0.133	6.705 ± 0.117
$\bar{e}^u_{ma}$	3.243 ± 0.106	3.238 ± 0.122	3.832 ± 0.15	4.945 ± 0.084	5.053 ± 0.077
$\bar{e}^u_{q.8a}$	4.84 ± 0.131	4.625 ± 0.164	5.777 ± 0.275	6.725 ± 0.145	7.124 ± 0.134
$\bar{e}^u_{maxa}$	12.174 ± 0.592	11.972 ± 0.658	14.203 ± 0.989	15.761 ± 0.534	15.298 ± 0.469

Table 4.9: BostonHousing data, $\pi_1 = \pi_2 = 0.9$

Error Measure	PSVR	MSVR	CSVR	BoxSD	BoxQ
$\bar{e}_{rms}$	6.04 ± 0.177	6.447 ± 0.187	7.681 ± 0.211	9.33 ± 0.165	9.59 ± 0.143
$\bar{e}_{ma}$	4.359 ± 0.102	4.331 ± 0.109	5.51 ± 0.165	6.384 ± 0.079	6.598 ± 0.092
$\bar{e}_{q.8a}$	6.083 ± 0.247	6.542 ± 0.215	8.328 ± 0.319	9.897 ± 0.171	9.689 ± 0.192
$\bar{e}_{maxa}$	17.219 ± 0.801	19.681 ± 0.967	19.52 ± 0.566	26.587 ± 1.048	27.13 ± 0.773
$\bar{e}^c_{rms}$	8.8 ± 0.724	9.022 ± 0.708	7.062 ± 0.56	8.769 ± 0.556	8.714 ± 0.6
$\bar{e}^c_{ma}$	7.551 ± 0.68	7.947 ± 0.682	6.366 ± 0.491	7.692 ± 0.477	7.836 ± 0.591
$\bar{e}^c_{q.8a}$	9.408 ± 0.815	9.912 ± 0.81	7.865 ± 0.624	9.74 ± 0.65	9.543 ± 0.666
$\bar{e}^c_{maxa}$	11.471 ± 0.9	11.434 ± 0.941	8.836 ± 0.74	12.055 ± 0.917	11.034 ± 0.758
$\bar{e}^u_{rms}$	5.513 ± 0.196	5.441 ± 0.173	7.116 ± 0.226	8.827 ± 0.131	9.172 ± 0.123
$\bar{e}^u_{ma}$	3.901 ± 0.099	3.818 ± 0.098	5.345 ± 0.196	6.233 ± 0.123	6.571 ± 0.125
$\bar{e}^u_{q.8a}$	5.468 ± 0.203	5.714 ± 0.167	8.371 ± 0.357	9.826 ± 0.213	9.986 ± 0.223
$\bar{e}^u_{maxa}$	14.197 ± 0.533	14.075 ± 0.681	18.911 ± 0.622	22.492 ± 0.299	24.043 ± 0.627

Table 4.10: Ozone data

Error Measure	PSVR	MSVR	CSVR	BoxSD	BoxQ
$\bar{e}_{rms}$	4.371 ± 0.067	4.356 ± 0.051	4.394 ± 0.063	4.692 ± 0.044	4.717 ± 0.022
$\bar{e}_{ma}$	3.44 ± 0.047	3.463 ± 0.064	3.512 ± 0.066	3.684 ± 0.038	3.682 ± 0.03
$\bar{e}_{q.8a}$	5.605 ± 0.067	5.52 ± 0.055	5.628 ± 0.123	5.967 ± 0.169	5.937 ± 0.068
$\bar{e}_{maxa}$	12.156 ± 0.614	11.919 ± 0.494	12.163 ± 0.213	12.489 ± 0.763	12.464 ± 0.512
$\bar{e}^c_{rms}$	4.596 ± 0.125	4.53 ± 0.147	4.595 ± 0.086	4.689 ± 0.144	4.681 ± 0.102
$\bar{e}^c_{ma}$	3.721 ± 0.228	3.637 ± 0.112	3.663 ± 0.11	3.728 ± 0.1	3.712 ± 0.104
$\bar{e}^c_{q.8a}$	5.726 ± 0.164	5.518 ± 0.058	5.529 ± 0.154	5.818 ± 0.177	5.802 ± 0.122
$\bar{e}^c_{maxa}$	11.528 ± 0.587	10.899 ± 0.789	11.435 ± 0.524	11.506 ± 0.467	11.465 ± 0.72
$\bar{e}^u_{rms}$	4.217 ± 0.037	4.132 ± 0.034	4.226 ± 0.079	4.62 ± 0.094	4.628 ± 0.101
$\bar{e}^u_{ma}$	3.274 ± 0.12	3.254 ± 0.079	3.403 ± 0.07	3.679 ± 0.066	3.627 ± 0.043
$\bar{e}^u_{q.8a}$	5.308 ± 0.288	5.181 ± 0.121	5.654 ± 0.232	5.812 ± 0.12	5.819 ± 0.116
$\bar{e}^u_{maxa}$	10.527 ± 0.747	10.383 ± 0.634	10.359 ± 0.528	11.004 ± 0.417	10.841 ± 0.628

Table 4.11: Boys data

Error Measure	PSVR	MSVR	CSVR	BoxSD	BoxQ
$\bar{e}_{rms}$	1.094 ± 0.241	1.207 ± 0.264	3.705 ± 0.731	1.589 ± 0.291	1.581 ± 0.293
$\bar{e}_{ma}$	0.866 ± 0.19	0.869 ± 0.147	2.902 ± 0.691	1.06 ± 0.207	1.07 ± 0.232
$\bar{e}_{q.8a}$	1.476 ± 0.275	1.463 ± 0.327	5.772 ± 1.279	1.774 ± 0.284	1.837 ± 0.342
$\bar{e}_{maxa}$	3.044 ± 0.554	3.257 ± 0.656	6.354 ± 1.191	6.245 ± 1.438	6.73 ± 1.614
$\bar{e}^{c}_{rms}$	1.247 ± 0.245	1.356 ± 0.284	1.217 ± 0.212	1.647 ± 0.316	1.611 ± 0.258
$\bar{e}^{c}_{ma}$	1.011 ± 0.228	1.102 ± 0.224	1.012 ± 0.216	1.258 ± 0.301	1.255 ± 0.219
$\bar{e}^{c}_{q.8a}$	1.716 ± 0.386	1.788 ± 0.425	1.699 ± 0.335	1.877 ± 0.402	2.104 ± 0.436
$\bar{e}^{c}_{maxa}$	2.658 ± 0.506	3.108 ± 0.54	2.875 ± 0.657	4.753 ± 1.039	4.516 ± 0.726
$\bar{e}^{u}_{rms}$	1.065 ± 0.214	1.051 ± 0.17	4.349 ± 0.852	1.435 ± 0.335	1.441 ± 0.32
$\bar{e}^{u}_{ma}$	0.713 ± 0.117	0.726 ± 0.121	3.716 ± 0.833	0.954 ± 0.195	0.963 ± 0.156
$\bar{e}^{u}_{q.8a}$	1.221 ± 0.24	1.182 ± 0.211	5.924 ± 1.129	1.468 ± 0.274	1.536 ± 0.334
$\bar{e}^{u}_{maxa}$	2.841 ± 0.289	3.201 ± 0.399	6.354 ± 1.178	5.685 ± 1.224	5.638 ± 0.986

Effect of hyper-parameters on PSVR's performance

We studied the effect that PSVR's hyper-parameters have on the method's perfomance by investigating how validation error varies across an array of combinations of hyper-parameter values. We focused our analysis on a representative inner-cross validation fold of the Normal data set with $\rho = 0.3$ and $\pi_1 = \pi_2 = 0.9$, and investigated all combinations of the possible hyper-parameter values mentioned in section 4.5, with the following exceptions: we added 0.125, 0.7, and 0.85 to the list of possible values for ϵ and ϵ_u, to increase granularity; we added 0.05, 0.125, and 0.375 and removed 0.5, 0.75, and 1 from the list of possible values for s, to shift the analysis towards lower values, since higher ones appeared to yield systematically worse validation performance on the data set we considered.

For each error measure, for each hyper-parameter, and for each possible value of such hyper-parameter, we recorded the minimum error obtained over all possible combinations of values of the other hyper-parameters. Figures 4.3, 4.4, and 4.5 report such minimum error as a curve, given as a function of each hyper-parameter's value, when errors are calculated on all observations, on the certain observations only, and on the uncertain observations only, respectively; each subfigure considers a different error measure. In these figures, we normalized the values of all hyper-parameters to the same range, to facilitate comparisons across different curves along the (logarithmic) y-axis; note, however, that the range starts from 0 for all hyper-parameters.

With the exception of, possibly, ϵ_u, all hyper-parameters have a very clear impact

on PSVR's performance. Most notably, this holds for c_u and s, which are key to the exploitation of data-uncertainty information within the model. Further details are provided below.

Hyper-parameter c: The optimum is close to zero according to all error measures. Interestingly, but not surprisingly, when performance is calculated via $e^u_\cdot$, the optimal value of c is exactly 0. This result implies that the certain data is completely ignored when training a model tailored to the uncertain data.

Hyper-parameter c_u: The optimum is attained at medium to low values according to all error measures. It does not surprise that, when performance is calculated via $e^c_\cdot$, the non-zero optimal value of c_u offers little advantage with respect to $c_u = 0$. This result suggests that the uncertain data plays a marginal role when training a model tailored to the certain data.

Hyper-parameter ϵ: The optimum is attained at medium to high values according to all error measures. Somewhat interestingly, ϵ appears to play a particularly significant role when performance is calculated via $e^c_\cdot$.

Hyper-parameter ϵ_u: The impact on performance is usually not substantial, although it appears to allow some degree of fine-tuning according to a few error measures, such as e_{maxa} and $e^u_{q.8a}$, for which larger values of ϵ_u seem to yield smaller error.

Hyper-parameter s: The optimum is attained at medium-low values according to all error measures. It does not surprise that, when performance is calculated via $e^c_\cdot$, the non-zero optimal value of s provides little advantage with respect to $s = 0$. This result is in line with what we observed for c_u above.

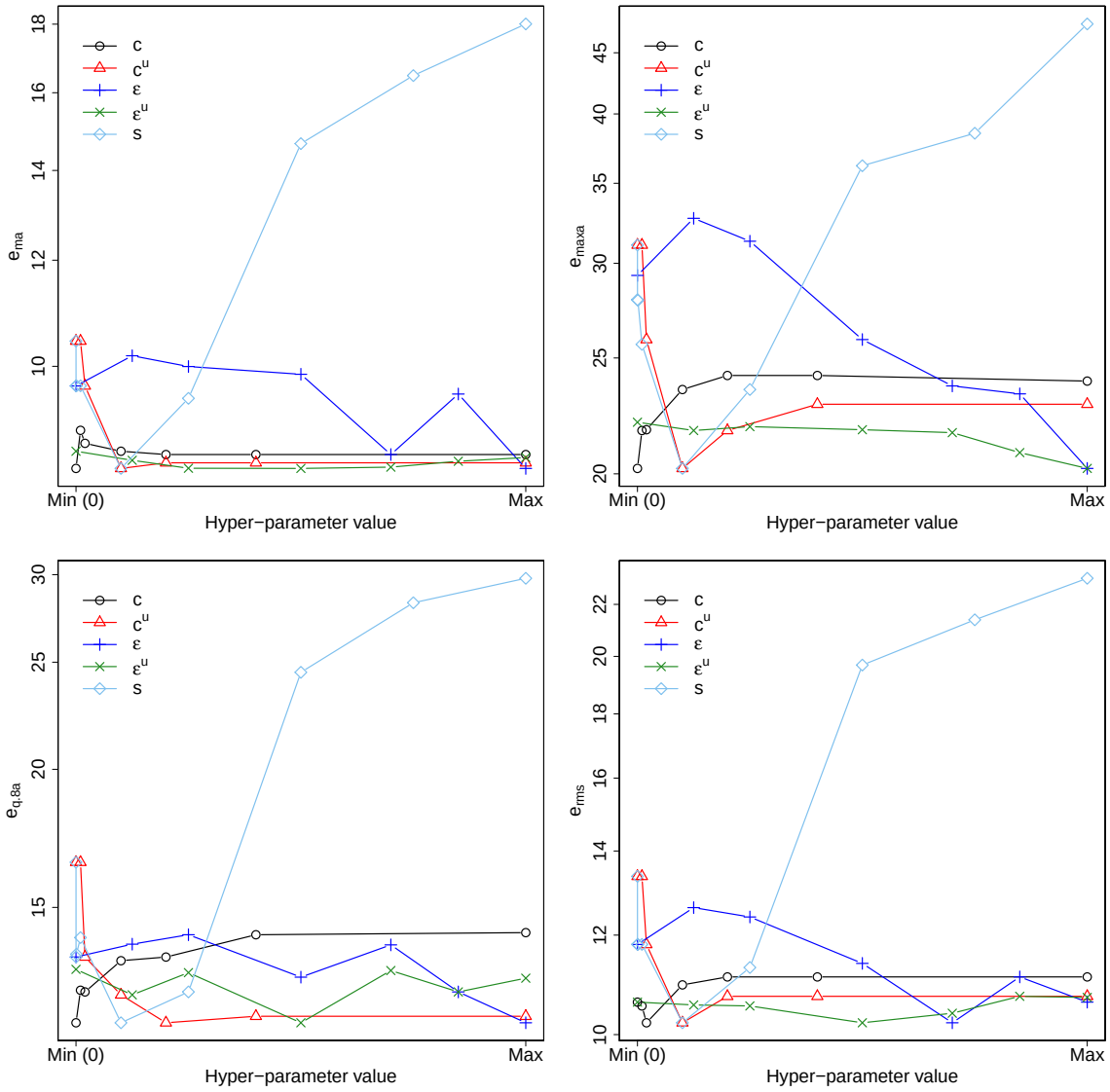

Figure 4.3: Validation error performance as a function of the hyper-parameter values, calculated on all observations.

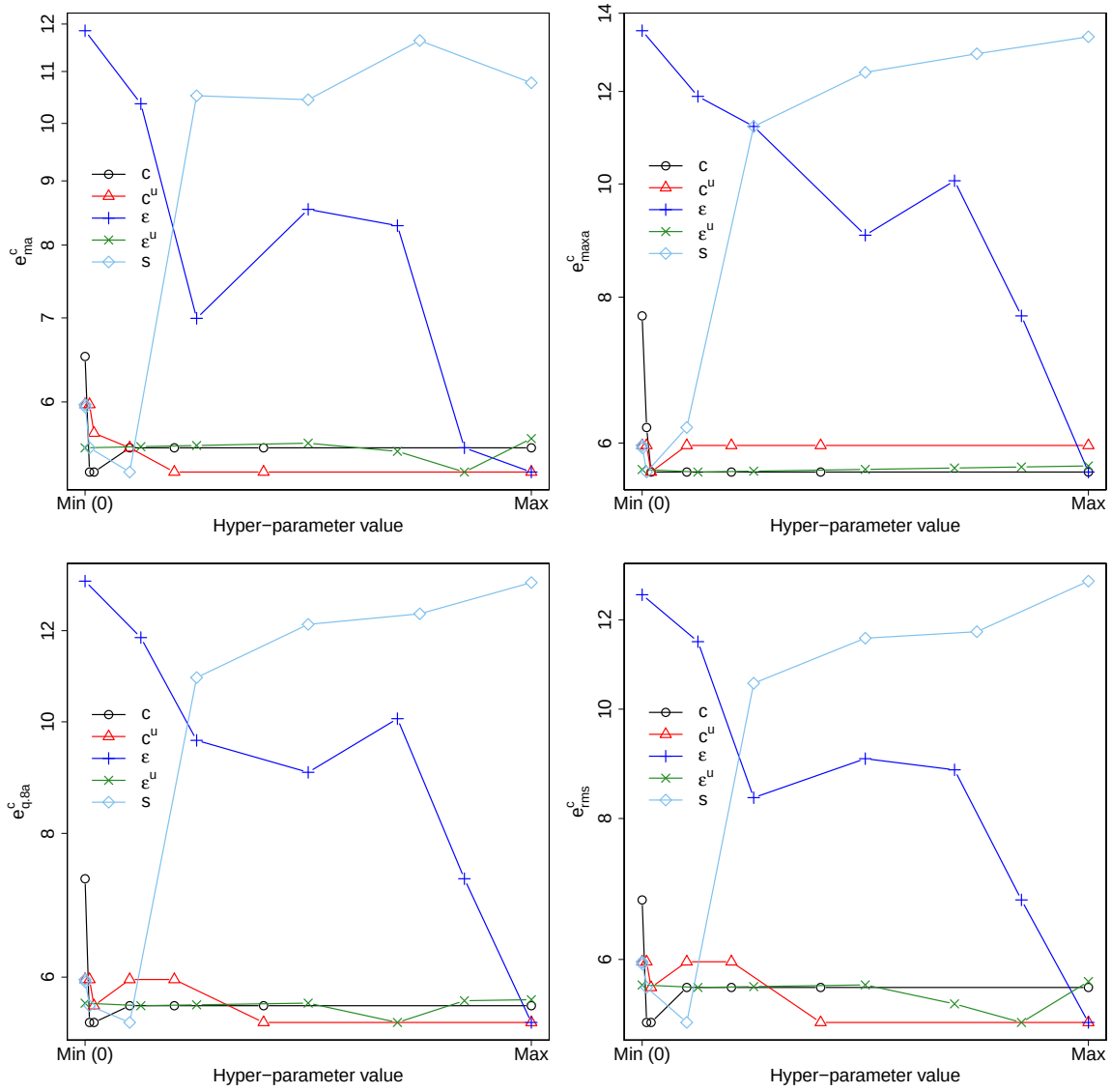

Figure 4.4: Validation error performance as a function of the hyper-parameter values, calculated on the certain observations only.

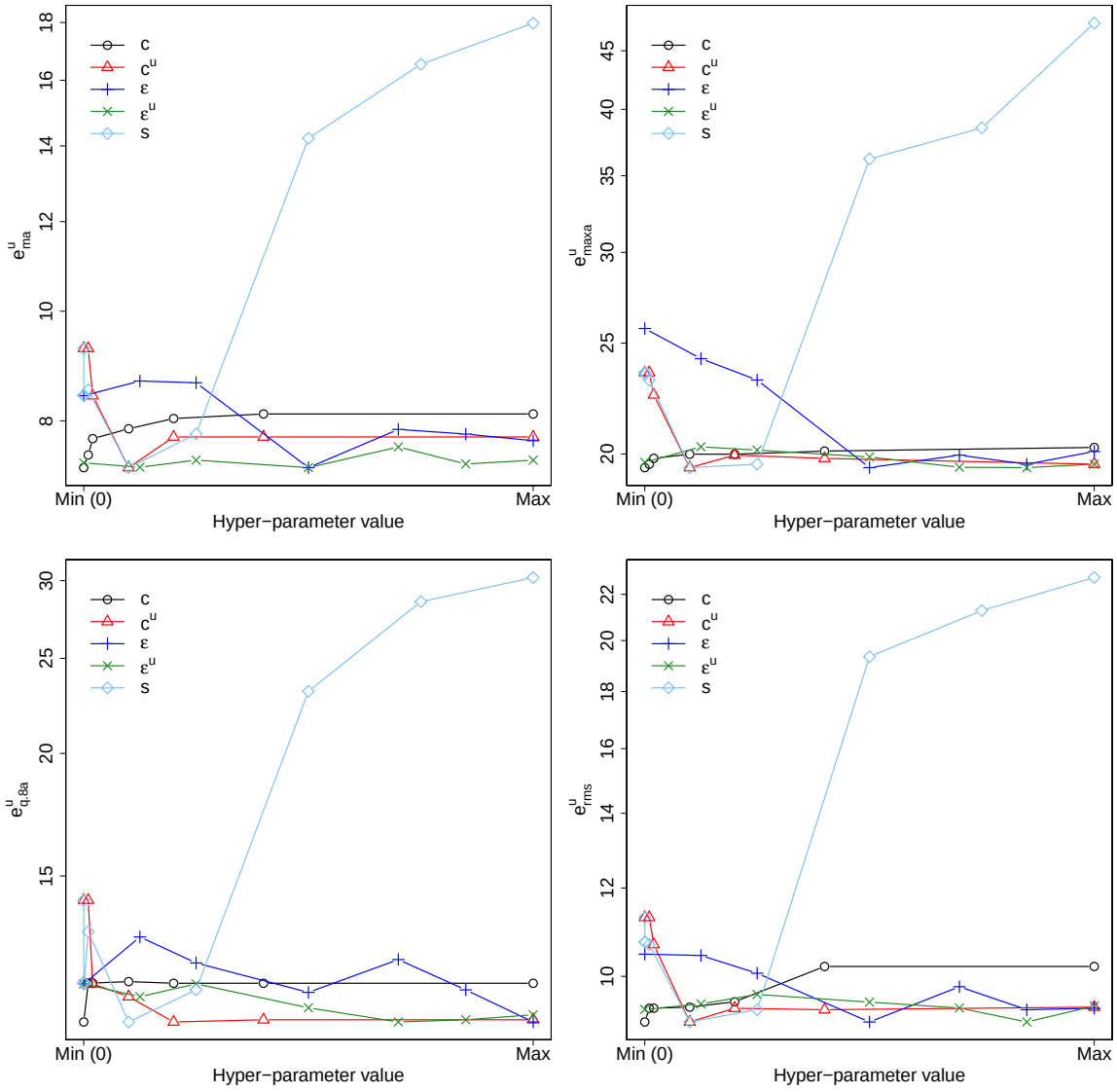

Figure 4.5: Validation error performance as a function of the hyper-parameter values, calculated on the uncertain observations only.

Chapter 5

Conclusions

In this dissertation we proposed a set of methods targeted to problems of variable importance, variable selection, and regression with uncertain information. The most important feature that all of these methods share is their capability of modeling and leveraging complex inter-variable dependences towards providing effective characterizations of the data and returning accurate predictions of unseen information. With extensive experimental validation in a multitude of real-world applications, we showed how this feature is key to our methods' performance, which is often superior to that of other state-of-the art approaches. The work we discussed here lends itself to several directions of development. Those of our current interest involve mainly the methods covered in Chapters 3 and 4 and are briefly presented below.

5.1 Extensions and future work on DBC-CPI and DBC-RCPI

The biased clustering approach behind the DBC-CPI measure is possibly the only unsupervised methodological component of this entire dissertation. A potential supervised extension of this approach may be targeted to calibrating the clustering effort on the $W^{(j)}$ space not only as a function of the dependence between X_j and the variables in $W^{(j)}$, but also as a function of the relevance that the variables in $W^{(j)}$ have towards Y. The basic conjecture that justifies this alternative approach is that, the larger the importance of $X_k \in W^{(j)}$, the

more the dependence between X_j and X_k will inflate X_j's non-conditional PI (see section 2.2.2). If that were indeed the case, it would be reasonable to exclude unimportant input variables from the space where the clustering of the data is performed. Alternatively, the non-conditional PI of $X_k \in W^{(j)}$ could be used to rescale weight $\boldsymbol{w}_{kj}$ prior to clustering, in such a way that within-group homogeneity be more strongly enforced on variables X_k that are both dependent on X_j and predictive of Y.

Despite being faster to execute than some of its competitor methods, DBC-RCPI remains a rather computationally intensive procedure. We envision two main directions to improve DBC-RCPI's computational performance: one implementational and one methodological. As per the former, the key observation is that DBC-RCPI is highly parallelizable. Each recursion, in fact, results in a total of p independent clustering problems (one per input variable), each of which inducing T (trees) $\times$ K (number of clusters) independent conditional permutations, which in turn rely on T independently trained trees. A parallelized implementation of DBC-RCPI, therefore, promises substantial computational gains. As per the methodological direction, we plan to incorporate in DBC-RCPI a hybrid importance measure I_j^H that carries both conditional and non-conditional information about X_j's importance. This hybrid measure could be defined, for example, as a convex combination $I_j^H = \tau I_j^P + (1-\tau) I_j^{DBC-CP}$, with $\tau \in [0,1]$ hyper-parameter. It has been already shown in [43] that, within each recursion of a backward-elimination algorithm relying on non-conditional PI (I_j^H with $\tau = 1$) for variable ranking, variable selection can be carried out both without the recalculation of PI and with the elimination of multiple variables. In section 3.2.2 we emphasized how, on the other hand, PI should be recalculated at every recursion if variable ranking is performed via the conditional variant (I_j^H with $\tau = 0$), with the obvious drawback that no more than one variable can be eliminated at a time. We conject, however, that, by appropriately tuning the value of τ, the number of recalculations of I_j^H may be reduced to less than p, and the corresponding number of variables eliminated at every recursion may be increased to more than one, without significantly affecting the quality of the solution currently returned by DBC-RCPI with $\tau = 0$ and p recalculations. This approach should obviously strike a reasonable balance between the computational gain resulting from the reduction of the number of recalculations/recursions, the computational

cost generated by the tuning of $\tau = 0$, and the properties of the resulting solution.

Interesting further extensions of DBC-CPI and DBC-RCPI may finally involve the replacement of RFs with other types of predictive models. In fact, the definition of variable importance by means of testing-data permutations and the measurement of the corresponding impact on predictive performance does not per se have to rely on the use of any specific model class. Analogously, it is conceivable to consider the substitution of K-means/medoids with different formulations of the clustering problem as a basis for devising partitions of the $W^{(j)}$ space. These lines of research are essentially unexplored in the literature on variable importance, and, as such, are open to potentially substantial innovations.

5.2 Extensions and future work on PSVR

SVR-like models are often processed in their dual formulation, primarily because that allows the solution of the optimal interpolating hyperplane problem within an abstract, arbitrarily-dimensional non-linear space, without incurring the additional computational cost of mapping the training points onto such space. This is made possible by the manipulation of the dual objective, which depends on the inner product of pairs of mapped training points. For certain types of maps, in fact, these inner products are equivalent to kernel functions, whose value can be efficiently calculated in the original space. To date, dual formulations have been proposed in the literature only for the special case of PSVR where uncertainty affects exclusively the output variable and can be modeled as an interval. Our intention is, therefore, to derive a dual PSVR formulation that can be applied to scenarios where uncertainty is modeled by arbitrary convex polyhedra, defined on both input and output variables.

Our use of MI for estimating polyhedra from datasets with missing values showed encouraging results in a variety of experiments. In the work we presented here, however, we limited our attention exclusively to MIs obtained via chained equations and predictive mean matching. A natural direction of further investigation, therefore, involves the study of the behavior of our approach as a function of different imputation models, and, possibly,

different MI techniques, which may be more or less appropriate depending on the nature of the data at hand.

Finally, an important open question is whether polyhedral data may be available to process as a given, rather than having to be estimated, in any real-world scenario in which numerical prediction or interpolation is of practical use. Interval data, that is, data geometrically represented by non-oriented box polyhedral data, are not uncommon in applications that involve measurements collected imprecisely or available as ranges. As per the more interesting general polyhedral data, a possible starting point may be the field of computer graphics, where convex polytopes play an important role in modeling real-world objects.

Appendices

Appendix A

R^2 modulation in a multiple linear model

Theorem A.1. *Let* $Y = \beta_0 + \sum_{j=1}^{\mathscr{P}} \beta_j X_j + \tilde{\epsilon}$ *be a multiple linear model, with* $\beta_0, \beta_1, \ldots, \beta_p \in \mathbb{R}$, $X_1, X_2, \ldots, X_{\mathscr{P}}$ *random variables with covariance matrix* $\mathbf{\Sigma} = \left(\sigma_{X_j X_k}\right) \in \mathbb{R}^{\mathscr{P}\times\mathscr{P}}$ *and correlation matrix* $\boldsymbol{R} = \left(\rho_{X_j X_k}\right) \in \mathbb{R}^{\mathscr{P}\times\mathscr{P}}$, $\tilde{\epsilon} = \beta_\epsilon \epsilon$, $\beta_\epsilon \in \mathbb{R}$, *and* ϵ *random variable such that* $\mathrm{Var}[\epsilon] = 1$ *and* $\mathrm{Cov}[X_j, \epsilon] = 0$, $\forall j, k \in \{1, 2, \ldots, \mathscr{P}\}$. *The two values of* β_ϵ *that yield a given coefficient of determination* $R^2 \in [0, 1]$ *in the multiple linear model are* $\pm\sqrt{\frac{\boldsymbol{c}^T \boldsymbol{R}^{-1} \boldsymbol{c}}{R^2} - b}$, *where* $b = \sum_{k=1}^{\mathscr{P}} \sum_{l=1}^{\mathscr{P}} \beta_k \beta_l \sigma_{X_k X_l}$, $\boldsymbol{c} = \left(\frac{a_1}{\sigma_{X_1}}, \frac{a_2}{\sigma_{X_2}}, \ldots, \frac{a_{\mathscr{P}}}{\sigma_{X_{\mathscr{P}}}}\right)^T$, *and* $a_j = \sum_{k=1}^{\mathscr{P}} \beta_k \sigma_{X_j X_k}$.

Proof. Let $\boldsymbol{\rho} = \left(\rho_{X_1,Y}, \rho_{X_2,Y}, \ldots, \rho_{X_p,Y}\right)^T$ be the vector of correlations between each of $X_1, X_2, \ldots, X_{\mathscr{P}}$ and Y. The assumptions in the theorem allow to rewrite $\forall j \in \{1, 2, \ldots, \mathscr{P}\}$:

$$
\begin{aligned}
\rho_{X_j,Y} &= \frac{\sigma_{X_j Y}}{\sigma_{X_j}\sigma_Y} \\
&= \frac{\mathrm{Cov}\left[X_j, \sum_{j=1}^{p} \beta_j X_j + \beta_\epsilon \epsilon\right]}{\sigma_{X_1}\sqrt{\mathrm{Var}\left[\sum_j \beta_j X_j + \beta_\epsilon \epsilon\right]}} \\
&= \frac{\sum_{k=1}^{p} \beta_k \sigma_{X_j X_k}}{\sigma_{X_1}\sqrt{\sum_{k=1}^{p}\sum_{l=1}^{p} \mathrm{Cov}\left[\beta_k X_k, \beta_l X_l\right] + \beta_\epsilon^2 \mathrm{Var}\left[\epsilon\right]}} \\
&= \frac{a_j}{\sigma_{X_j}\sqrt{b + \beta_\epsilon^2}}
\end{aligned}
$$

Letting $\boldsymbol{c}^T = \left(\frac{a_1}{\sigma_{X_1}}, \frac{a_2}{\sigma_{X_2}}, \ldots, \frac{a_{\mathscr{P}}}{\sigma_{X_{\mathscr{P}}}}\right)$, we can express the coefficient of determination R^2 as follows:

$$\begin{aligned} R^2 &= \boldsymbol{\rho}^T \boldsymbol{R}^{-1} \boldsymbol{\rho} \\ &= \frac{1}{b + \beta_\epsilon^2} \boldsymbol{c}^T \boldsymbol{R}^{-1} \boldsymbol{c} \end{aligned}$$

Solving the last equation for β_ϵ yields $\beta_\epsilon = \pm\sqrt{\frac{\boldsymbol{c}^T \boldsymbol{R}^{-1} \boldsymbol{c}}{R^2} - b}$.

□

Remark. Theorem A.1 applies to any multiple linear model. It is easy to see that is the case with (3.12), by simply setting $\mathscr{P} = p$; for (3.13), it is sufficient to set $\mathscr{P} = p + p^2$ and $X_j, j > p$ equal to the appropriate standardized product terms.

Theorem A.2. *Let $X_1, X_2, \ldots, X_{\mathscr{P}}$ be standard Normal random variables with correlation matrix $\boldsymbol{R} = \left(\rho_{X_j X_k}\right) \in \mathbb{R}^{\mathscr{P} \times \mathscr{P}}$. The covariance matrix over $X_1, X_2, \ldots, X_{\mathscr{P}}$ and their second-degree transforms $X_j X_k$ is given by $\boldsymbol{\Sigma}_2 = \left(\begin{smallmatrix} \boldsymbol{R} & \boldsymbol{0} \\ \boldsymbol{0}^T & \boldsymbol{A} \end{smallmatrix}\right) \in \mathbb{R}^{\frac{\mathscr{P}(\mathscr{P}+3)}{2} \times \frac{\mathscr{P}(\mathscr{P}+3)}{2}}$, where $\boldsymbol{0} \in \mathbb{R}^{\mathscr{P} \times \frac{\mathscr{P}(\mathscr{P}-1)}{2}}$ is a zero matrix and $\boldsymbol{A} = \left(\rho_{X_j X_l} \rho_{X_k X_m} + \rho_{X_j X_k} \rho_{X_l X_m}\right) \in \mathbb{R}^{\mathscr{P}(\mathscr{P}+1) \times \mathscr{P}(\mathscr{P}+1)}$, $\forall j, k, l, m \in \{1, 2, \ldots, \mathscr{P}\} : \; j \leq k \leq l \leq m$.*

Proof. Matrix $\boldsymbol{R}$: $\boldsymbol{R}$ is, trivially, the matrix of covariances on the pairs of random variables (X_j, X_k), $\forall j, k \in \{1, 2, \ldots, \mathscr{P}\}$. Matrix $\boldsymbol{0}$: We observe we can write $X_k = r_{jk} X_j + s_{jk} Z_k$, $\forall r_{jk} \in [0, 1]$, $s_{jk} = 1 - r_{jk}^2$, with $Z_k \sim N(0, 1)$ independent of X_j. Since $\rho_{X_j X_k} \in [0, 1]$, we can set $r = \rho_{X_j X_k}$ and calculate:

$$\begin{aligned} \mathrm{Cov}[X_j, X_k X_l] &= \mathrm{E}[X_j X_k X_l] - \mathrm{E}[X_j]\mathrm{E}[X_k X_l] \\ &= \mathrm{E}\left[X_j (\rho_{X_j X_k} X_j + s_{jk} Z_k)(\rho_{X_j X_l} X_j + s_{jl} Z_l)\right] - 0 \cdot \mathrm{E}[X_k X_l] \\ &= \mathrm{E}\left[\rho_{X_j X_k} \rho_{X_j X_l} X_j^3 + \rho_{X_j X_k} s_{jl} X_j^2 Z_l + \rho_{X_j X_l} s_{jk} X_j^2 Z_k + s_{jk} s_{jl} X_j Z_k Z_l\right] \\ &= \rho_{X_j X_k} \rho_{X_j X_l} \mathrm{E}\left[X_j^3\right] + \rho_{X_j X_k} s_{jl} \mathrm{E}\left[X_j^2 Z_l\right] + \rho_{X_j X_l} s_{jk} \mathrm{E}\left[X_j^2 Z_k\right] + s_{jk} s_{jl} \mathrm{E}\left[X_j Z_k Z_l\right] \\ &= 0 + 0 + 0 + s_{jk} s_{jl} \left(\mathrm{E}\left[\left(X_j Z_k - \mathrm{E}[X_j Z_k]\right)\left(Z_l - \mathrm{E}[Z_l]\right)\right)\right] + \mathrm{E}[X_j Z_k]\mathrm{E}[Z_l]\right) \\ &= 0 \end{aligned}$$

where we have used the fact that $\mathrm{E}[X_j^q] = (q-1)!!$ for q even and $\mathrm{E}[X_j^q] = 0$ for q odd, and that $\mathrm{E}[X_j^q Z_\cdot^t] = \mathrm{E}[X_j^q]\mathrm{E}[Z_\cdot^t]$, $\forall q, t \in \mathbb{N}$, because functions of independent random variables are themselves independent. This result can be used to build the matrix $\mathbf{0}$ of covariances on the pairs of random variables $(X_j, X_k X_l)$, $\forall j,k,l \in \{1,2,\ldots,p\} : j \leq k \leq l$. Matrix $\boldsymbol{A}$:

$$
\begin{aligned}
&\mathrm{Cov}[X_j X_k, X_l X_m] = \\
&= \mathrm{E}[X_j X_k X_l X_m] - \mathrm{E}[X_j X_k]\mathrm{E}[X_l X_m] \\
&= \mathrm{E}\left[X_j(\rho_{X_jX_k}X_j + s_{jk}Z_k)(\rho_{X_jX_l}X_j + s_{jl}Z_l)(\rho_{X_jX_m}X_j + s_{jm}Z_m)\right] \\
&\qquad - \mathrm{E}[X_j(\rho_{X_jX_k}X_j + s_{jk}Z_k)]\mathrm{E}[X_l(\rho_{X_lX_m}X_l + s_{lm}Z_m)] \\
&= \mathrm{E}\big[\rho_{X_jX_k}\rho_{X_jX_l}\rho_{X_jX_m}X_j^4 + \rho_{X_jX_k}\rho_{X_jX_l}s_{jm}X_j^3Z_m + \rho_{X_jX_k}\rho_{X_jX_m}s_{jl}X_j^3Z_l \\
&\qquad + \rho_{X_jX_k}s_{jl}s_{jm}X_j^2Z_lZ_m + \rho_{X_jX_l}\rho_{X_jX_m}s_{jk}X_j^3Z_k + \rho_{X_jX_l}s_{jk}s_{jm}X_j^2Z_kZ_m \\
&\qquad + \rho_{X_jX_m}s_{jk}s_{jl}X_j^2Z_kZ_l + s_{jk}s_{jl}s_{jm}X_1Z_kZ_lZ_m\big] \\
&\qquad - \mathrm{E}\left[\rho_{X_jX_k}X_j^2 + s_{jk}X_jZ_k\right]\mathrm{E}\left[\rho_{X_lX_m}X_l^2 + s_{lm}X_lZ_m\right] \\
&= \rho_{X_jX_k}\rho_{X_jX_l}\rho_{X_jX_m}\mathrm{E}\left[X_j^4\right] + \rho_{X_jX_k}\rho_{X_jX_l}s_{jm}\mathrm{E}\left[X_j^3Z_m\right] + \rho_{X_jX_k}\rho_{X_jX_m}s_{jl}\mathrm{E}\left[X_j^3Z_l\right] \\
&\qquad + \rho_{X_jX_k}s_{jl}s_{jm}\mathrm{E}\left[X_j^2Z_lZ_m\right] + \rho_{X_jX_l}\rho_{X_jX_m}s_{jk}\mathrm{E}\left[X_j^3Z_k\right] + \rho_{X_jX_l}s_{jk}s_{jm}\mathrm{E}\left[X_j^2Z_kZ_m\right] \\
&\qquad + \rho_{X_jX_m}s_{jk}s_{jl}\mathrm{E}\left[X_j^2Z_kZ_l\right] + s_{jk}s_{jl}s_{jm}\mathrm{E}\left[X_1Z_kZ_lZ_m\right] \\
&\qquad - \left(\rho_{X_jX_k}\mathrm{E}\left[X_j^2\right] + s_{jk}\mathrm{E}\left[X_jZ_k\right]\right)\left(\rho_{X_lX_m}\mathrm{E}\left[X_l^2\right] + s_{lm}\mathrm{E}\left[X_lZ_m\right]\right) \\
&= 3\rho_{X_jX_k}\rho_{X_jX_l}\rho_{X_jX_m} + 0 + 0 + \rho_{X_jX_k}s_{jl}s_{jm}\mathrm{E}\left[X_j^2\right]\mathrm{E}\left[Z_lZ_m\right] + 0 \\
&\qquad + \rho_{X_jX_l}s_{jk}s_{jm}\mathrm{E}\left[X_j^2\right]\mathrm{E}\left[Z_kZ_m\right] + \rho_{X_jX_m}s_{jk}s_{jl}\mathrm{E}\left[X_j^2\right]\mathrm{E}\left[Z_kZ_l\right] + 0 \\
&\qquad - (\rho_{X_jX_k} + 0)(\rho_{X_lX_m} + 0) + 0 + \rho_{X_jX_l}s_{jk}s_{jm}\mathrm{E}\left[X_j^2\right]\mathrm{E}\left[Z_kZ_m\right] \\
&= \rho_{X_jX_l}\rho_{X_kX_m} + \rho_{X_jX_k}\rho_{X_lX_m},
\end{aligned}
$$

where last step follows from

$$
\mathrm{E}\left[Z_{(.)}Z_{(..)}\right] = \mathrm{E}\left[\frac{\left(X_{(.)} - \rho_{X_jX_{(.)}}X_j\right)\left(X_{(..)} - \rho_{X_jX_{(..)}}X_j\right)}{s_{j(.)}s_{j(..)}}\right] = \frac{\rho_{X_{(.)}X_{(..)}}}{s_{j(.)}s_{j(..)}},
$$

by $X_{(.)} = \rho_{X_j X_{(.)}} X_j + s_{j(.)} Z_{(.)}$ and $X_{(..)} = \rho_{X_j X_{(..)}} X_j + s_{j(..)} Z_{(..)}$. This result can be used to build the matrix $\boldsymbol{A}$ of covariances on the pairs of random variables $(X_j X_k, X_l X_m)$, $\forall j,k,l,m \in \{1,2,\ldots,\mathscr{P}\} : j \leq k \leq l \leq m$.

□

Corollary A.3. *Let $X_1, X_2, \ldots, X_{\mathscr{P}}$ be as in Theorem A.2. The covariance/correlation matrix over $X_1, X_2, \ldots, X_p$ and their standardized second-degree transforms $\widetilde{X_j X_k}$ is the same as $\boldsymbol{\Sigma_2}$, except $\boldsymbol{A} = \left(-\rho_{X_j X_k} \rho_{X_l X_m} \Big/ \left(\rho^2_{X_j X_k} + 1\right)^{\frac{1}{2}} \left(\rho^2_{X_l X_m} + 1\right)^{\frac{1}{2}} \right)$.*

Proof. Matrix $\boldsymbol{R}$ is obvious. Matrix $\mathbf{0}$:

$$
\begin{aligned}
\mathrm{Cov}[X_j, \widetilde{X_k X_l}] &= \mathrm{E}\left[X_j \widetilde{X_k X_l}\right] - \mathrm{E}[X_j]\mathrm{E}\left[\widetilde{X_k X_l}\right] \\
&= \mathrm{E}\left[X_j \frac{X_k X_l - \mathrm{E}\left[X_k X_l\right]}{\sqrt{\mathrm{Var}\left[X_k X_l\right]}}\right] - 0 \cdot \mathrm{E}\left[\widetilde{X_k X_l}\right] \\
&= \mathrm{E}\left[X_j \frac{X_k X_l - \mathrm{E}\left[X_k \left(\rho_{X_k X_l} X_k + s_{kl} Z_l\right)\right]}{\sqrt{\mathrm{Var}\left[X_k \left(\rho_{X_k X_l} X_k + s_{kl} Z_l\right)\right]}}\right] \\
&= \mathrm{E}\left[X_j \frac{X_k X_l - \rho_{X_k X_l}}{\sqrt{\rho^2_{X_k X_l} + 1}}\right] \\
&= \frac{1}{\sqrt{\rho^2_{X_k X_l} + 1}} \left(\mathrm{E}\left[X_j X_k X_l\right] - \rho_{X_k X_l} X_j\right) \\
&= \frac{1}{\sqrt{\rho^2_{X_k X_l} + 1}} (0 - 0) \\
&= 0
\end{aligned}
$$

Matrix $\boldsymbol{A}$:

$$
\begin{aligned}
&\mathrm{Cov}[\widetilde{X_jX_k}, \widetilde{X_lX_m}] = \\
&= \mathrm{E}[\widetilde{X_jX_k}\widetilde{X_lX_m}] - \mathrm{E}[\widetilde{X_jX_k}]\mathrm{E}[\widetilde{X_lX_m}] \\
&= \frac{-\rho_{X_jX_k}\rho_{X_lX_m}}{\sqrt{\left(\rho^2_{X_jX_k}+1\right)\left(\rho^2_{X_lX_m}+1\right)}} - 0 \cdot 0 \\
&= \frac{-\rho_{X_jX_k}\rho_{X_lX_m}}{\sqrt{\left(\rho^2_{X_jX_k}+1\right)\left(\rho^2_{X_lX_m}+1\right)}}
\end{aligned}
$$

□

Appendix B

Characterization of brain-network connectivity

B.1 A brief neuro-scientific overview

The anatomical architecture and the integrated physiological activity of large-scale complex neuronal networks are fundamental factors that affect brain functionality. Studies show that alterations in brain network organization may act as predictors of neurological dysfunctions. Of particular interest are alterations that affect the connectivity between and within elements of the brain network. Connectivity may take place in three forms: structurally, effectively, and functionally [116, 117, 118]. Structural connectivity is determined by physical connections, such as fiber pathways and synaptic links, whereas effective and functional connectivity are identified by specific relationships between temporal dynamics of neuronal activity (NA) in different brain areas, causal in nature for the former, similarity-related for the latter.

Functional connectivity, which our work specifically focuses on, can be analyzed both in a task-positive and in a task-negative framework. Depending on whether a subject is performing a task or remaining idle while being examined, different functional networks arise in the brain. In the former case, the so called "cognitive control network" becomes

apparent [119] and the focus of the analysis is typically on how activation and network connections vary depending on the task being performed. In the latter case, the so called "default mode network" (DMN) becomes dominant [120], and the focus of the analysis is on the characterization of the typical local NA patterns and distributed connections at rest.

DMN plays a crucial role in cognition. Studies have, for example, shown a clear relationship between DMN organization, defined by the pattern of interconnections between elements of the network, and several measures of cognitive performance, such as intelligence quotient and episodic memory. Reorganization of connections within DMN, generally taking the form of reduced connectivity levels, arises with decline in cognitive performance (e.g., aging), disturbed cognitive function (e.g., schizophrenia), and degenerative disorders (e.g., Alzheimer's, multiple sclerosis).

A DMN-dependent medical condition that is particularly relevant to our work is mild cognitive impairment (MCI). MCI is a brain function syndrome that falls between normal cognitive decline caused by aging and abnormal cognitive decline induced by neurological pathologies, such as Alzheimer's, of which MCI may be a precursor. Diagnosis of MCI is made on a symptomatic basis, with the drawback that clear signs of MCI appear once the condition has already reached an advanced stage. Standard pre-symptomatic diagnostic criteria to assess the risk of developing the MCI or early detecting its progression are still not available, preventing timely medical treatment of the disorder. A variety of data-driven approaches have been proposed to mitigate this issue [121, 122, 123, 124, 125, 126, 127, 128, 129].

The characterization of distributed NA and the organization of brain networks boomed with the development of functional magnetic resonance imaging (fMRI), which emerged in the early 1990s as an evolution of conventional magnetic resonance imaging (MRI). MRI is a biomedical imaging technology that employs a combination of radio waves and strong magnetic fields to acquire detailed images of a subject's bodily structure by means of non-invasive scans. fMRI focuses on mapping NA over time, via the collection of sequences of individual MRI images that allow the detection of specific physiological changes in the

scanned tissues [130]. NA has been shown to be closely connected to local fluctuations in blood oxygenation level. In fact, when a neuron becomes active, its metabolic rate increases and so does its demand of oxygen, causing a shift in the relative concentration of oxygenated and deoxygenated hemoglobin in the blood flowing to the proximate vessels. This quantity, which is referred to as BOLD (Blood Oxygenation Level Dependent) signal, can be measured by an MRI scanner, since the magnetic susceptibility of blood is a function of its oxygenation level, and is therefore used as a proxy of NA [131, 132].

fMRI images can be seen as brain "slices", acquired along three orthogonal x, y, z directions (top-down, front-back, left-right with respect to the body of the subject). Each of them is uniformly partitioned into a grid of (typically, 64 x 64 or 128 x 128) cubic cells called voxels, with x- y- z-dimensions in the order 3 x 3 x 4 mm. A given image records the BOLD signal of every voxel of a brain slice at a given time point; a volumetric snapshot of the NA of the entire brain can be obtained by stacking together all images (typically in the order of 30 of them, each located at a different z-coordinate) acquired at the same moment. Throughout the course of an fMRI experiment, the NA of each voxel is measured at uniformly spaced time points; the time distance between two successive observations, known as "repetition time" (TR), typically ranges from 500 to 4000 milliseconds. In conclusion, the output of an fMRI experiment is a volume of spatially resolved time series, each of which describing the dynamics of NA at a different brain location.

B.2 Functional-connectivity feature extraction algorithms

Our functional-connectivity feature-extraction algorithms (FEAs) process matrices whose elements represent correlations between a the BOLD time series of a row-voxel and that of a column-voxel. Let a (b) be a list of row (column) indices. We define a submatrix of a matrix $\boldsymbol{M}$ with either $\boldsymbol{M}[a, b]$ or $\boldsymbol{M}[-a, -b]$, depending on whether the submatrix is obtained selecting or removing a-indexed rows and b-indexed columns from $\boldsymbol{M}$. We use two special types of lists of indices: sequential lists and empty lists. Sequential lists identify chunks of adjacent rows (or columns) and are denoted by $i{:}j$, where i and j are the first

and last row (or column) in the chunk, with $1 \leq i \leq j$; if $i = j$, we omit the ': j' part. Empty lists leave rows (or columns) of the array untouched, and are denoted by a colon (':'). Analogous notation is used for vectors, that are treated as one-column matrices; however, when defining subvectors we simply use one list of row indices to identify the elements to select or remove.

Other data manipulation operators used in the FEAs are the following: $abs(\boldsymbol{M})$, which returns the matrix obtained extracting the absolute value from the elements of $\boldsymbol{M}$; $nrows(\boldsymbol{M})$ ($ncols(\boldsymbol{M})$), which returns the number of rows (columns) of $\boldsymbol{M}$; $mean(\boldsymbol{v})$, $sd(\boldsymbol{v})$, $skew(\boldsymbol{v})$, which respectively return the mean, the standard deviation and the nonparametric skew of the elements of vector $\boldsymbol{v}$ (disregarding any missing elements); $freq(\boldsymbol{v})$, which returns the vector containing the relative frequency of all values taken on by the elements of $\boldsymbol{v}$; $round(s)$, which rounds scaler s to the nearest integer; $order(\boldsymbol{v})$, which returns a list containing the permutation of the indices of the elements of $\boldsymbol{v}$ that sorts such elements in increasing order; $intersect(a, b)$, which returns a list containing the indices shared by a and b.

The input of the FEAs consists of a certain number of parameters and one of three different types of matrices. If we let $\boldsymbol{R} = (r_{kl})$ be the Pearson's linear correlation matrix computed on the BOLD time series of every voxel pair (k, l), and let $a = \{$list of indices of voxels in ROI $i\}$, $\bar{a} = \{$list of indices of voxels in ROIs other than $i\}$, and $b = \{$list of indices of voxels in ROI $j\}$, the three types of matrices are: $\boldsymbol{R}_i^i = \boldsymbol{R}[a, a]$, $\boldsymbol{R}_i^{\bar{i}} = \boldsymbol{R}[a, \bar{a}]$, $\boldsymbol{R}_i^j = \boldsymbol{R}[a, b]$.

Although targeted to the DMN, these FEAs can be used to characterize functional connectivity within any brain network. For a network with N ROIs, the FEAs extract a total of $\frac{3}{2}N(3N + 7)$ features. The $N = 6$ ROIs of the DMN, for example, results in a total of 225 features. Details and pseudocode pertinent to each FEA are presented below.

Intra-ROI Connectivity (IaC): It computes the average absolute correlation of every voxel (referred to as "seed" voxel) in a given ROI with all other voxels (referred to as "target" voxels) in the same ROI and returns average, standard deviation, and

nonparametric skew of such averages across all seed voxels, for a total of $3N$ features.

procedure IAC($\boldsymbol{R}_i^i$)
  $n \leftarrow nrows(\boldsymbol{R}_i^i)$
  $\boldsymbol{v} \leftarrow n$-dimensional empty vector
  $\boldsymbol{R}_i^i \leftarrow abs(\boldsymbol{R}_i^i)$
  for $k = 1, 2, ..., n$ **do**
    $\boldsymbol{w} \leftarrow \boldsymbol{R}_i^i[k, -k]$
    $\boldsymbol{v}[k] \leftarrow mean(\boldsymbol{w})$
  end for
  return $mean(\boldsymbol{v})$, $sd(\boldsymbol{v})$, $skew(\boldsymbol{v})$
end procedure

Intra-ROI Top Positive Connectivity (IaTC): It finds the top positive correlations (via subroutine TPC) of every seed voxel in a given ROI with all other target voxels in the same ROI and computes their average as well as their proportion over all correlation values. It finally returns average, standard deviation, and nonparametric skew of (a) such averages and (b) such proportions across all seed voxels, for a total of $6N$ features.

procedure IATC($\boldsymbol{R}_i^i$, s)
  $n \leftarrow nrows(\boldsymbol{R}_i^i)$
  $m_s \leftarrow round(s(n-1))$
  $\boldsymbol{v}_1 \leftarrow n$-dimensional empty vector
  $\boldsymbol{v}_2 \leftarrow n$-dimensional empty vector
  for $k = 1, 2, ..., n$ **do**
    $\boldsymbol{w} \leftarrow \boldsymbol{R}_i^i[k, -k]$
    $\boldsymbol{w} \leftarrow$TPC($\boldsymbol{w}, m_s$)
    if $\boldsymbol{w}$ is not empty **then**
      $\boldsymbol{v}_1[k] \leftarrow mean(\boldsymbol{w})$
      $\boldsymbol{v}_2[k] \leftarrow nrows(\boldsymbol{w})/m_s$

end if

end for

return $mean(\boldsymbol{v}_1)$, $sd(\boldsymbol{v}_1)$, $skew(\boldsymbol{v}_1)$, $mean(\boldsymbol{v}_2)$, $sd(\boldsymbol{v}_2)$, $skew(\boldsymbol{v}_2)$

end procedure

Inter-ROI Connectivity (IeC): It computes the average absolute correlation of every seed voxel in a given ROI (referred to as "seed" ROI) with all voxels in another ROI (referred to as "target" ROI) and returns the average of such averages, for a total of $N(N-1)/2$ features (since swapping seed and target ROI does not change the returned value, IeC is run only once per unordered pair of ROIs).

procedure IeC($\boldsymbol{R}_i^j$)

$n \leftarrow nrows(\boldsymbol{R}_i^j)$

$m \leftarrow ncols(\boldsymbol{R}_i^j)$

for $k = 1, 2, ..., n$ **do**

$\boldsymbol{w} \leftarrow \boldsymbol{R}_i^j[k, :]$

$\boldsymbol{v}[k] \leftarrow mean(\boldsymbol{w})$

end for

return $mean(\boldsymbol{v})$, $sd(\boldsymbol{v})$, $skew(\boldsymbol{v})$

end procedure

Inter-ROI Top Positive Connectivity (IeTC): It finds the set of top positive correlations of every seed voxel in a given seed ROI with all voxels in a given target ROI and computes their average as well as their proportion over all correlation values. It finally returns the average of (a) such averages and (b) such proportions across all seed voxels, for a total of $2N(N-1)$ features. Any seed voxels whose set of top positive correlations is empty are disregarded.

procedure IeTC($\boldsymbol{R}_i^j$, s)

$n \leftarrow nrows(\boldsymbol{R}_i^j)$

$m \leftarrow ncols(\boldsymbol{R}_i^j)$

$m_s \leftarrow round(sm)$

$\boldsymbol{v}_1 \leftarrow n$-dimensional empty vector

$\boldsymbol{v}_2 \leftarrow n$-dimensional empty vector

for $k = 1, 2, ..., n$ **do**

$\boldsymbol{w} \leftarrow \boldsymbol{R}_i^j[k, :]$

$\boldsymbol{w} \leftarrow$TPC$(\boldsymbol{w}, m_s)$

if $\boldsymbol{w}$ is not empty **then**

$\boldsymbol{v}_1[k] \leftarrow mean(\boldsymbol{w})$

$\boldsymbol{v}_2[k] \leftarrow nrows(\boldsymbol{w})/m_s$

end if

end for

return $mean(\boldsymbol{v}_1)$, $mean(\boldsymbol{v}_2)$

end procedure

Maximum Inter-ROI Connectivity Region (MIeCR): It computes the average absolute correlation of every seed voxel in a given seed ROI with all voxels in each of all possible target ROIs, and identifies the target ROI for which such average is maximum. It finally returns the relative frequency of every target ROI as region of maximum average correlation, calculated across all seed voxels, for a total of $N(N-1)$ features.

procedure MIeCR($\{\boldsymbol{R}_i^j : j \neq i\}$)

$\boldsymbol{R}_i^j = abs(\boldsymbol{R}_i^j)\ \forall j \neq i$

$n \leftarrow nrows(\boldsymbol{R}_i^1)$

$\boldsymbol{v} \leftarrow n$-dimensional empty vector

for $k = 1, 2, ..., n$ **do**

$\boldsymbol{m} \leftarrow mean(\boldsymbol{R}_i^j[k, :])$

$\boldsymbol{v}[k] = \arg\max_{j \neq i} \boldsymbol{m}$

end for

return $freq(\boldsymbol{v})$

end procedure

Maximum Inter-ROI Top Positive Connectivity Region (MIeTCR): It finds the top positive correlations of every seed voxel in a given seed ROI with all voxels in a given target ROI, calculates their average, and identifies the target ROI for which such average is maximum. It finally returns the relative frequency of every target ROI as region of maximum average top positive correlation, calculated across all seed voxels, for a total of $N(N-1)$ features.

procedure MIeTCR($\{\boldsymbol{R}_i^j : j \neq i\}$)
  $\boldsymbol{R}_i^j = abs(\boldsymbol{R}_i^j)\ \forall j \neq i$
  $n \leftarrow nrows(\boldsymbol{R}_i^1)$
  $v \leftarrow n$-dimensional empty vector
  for $k = 1, 2, ..., n$ **do**
    for $j \neq i$ **do**
      $m \leftarrow ncols(\boldsymbol{R}_i^j)$
      $m_s \leftarrow round(s(m-1))$
      $\boldsymbol{w}_j \leftarrow \boldsymbol{R}_i^j[k, :]$
      $\boldsymbol{w}_j \leftarrow$TPC$(\boldsymbol{w}_j, m_s)$
    end for
    $\boldsymbol{m} \leftarrow mean(\boldsymbol{w}_j)$
    $\boldsymbol{v}[k] = \arg\max_{j \neq i} \boldsymbol{m}$
  end for
  return $freq(\boldsymbol{v})$
end procedure

Intra/Inter-ROI Connectivity Ratio (IaIeCR): It calculates the average absolute correlation of every seed voxel in a given ROI with all target voxels in the same ROI, as well as the average absolute correlation of every seed voxel with all voxels in any target ROI, and computes the ratio of those two quantities. It finally returns average, standard deviation, and nonparametric skew of such ratios across all seed voxels, for a total of $3N$ features.

procedure IaIeCR($\boldsymbol{R}_i^j$, $\boldsymbol{R}_i^{\bar{i}}$)

$n \leftarrow nrows(\boldsymbol{R}_i^i)$

$\boldsymbol{v} \leftarrow n$-dimensional empty vector

$\boldsymbol{R}_i^i \leftarrow abs(\boldsymbol{R}_i^i)$

$\boldsymbol{R}_i^{\bar{i}} \leftarrow abs(\boldsymbol{R}_i^{\bar{i}})$

for $k = 1, 2, ..., n$ **do**

$\boldsymbol{w}_1 \leftarrow \boldsymbol{R}_i^i[k, -k]$

$\boldsymbol{w}_2 \leftarrow \boldsymbol{R}_i^j[k, :]$

$\boldsymbol{v}_1[k] \leftarrow mean(\boldsymbol{w}_1)$

$\boldsymbol{v}_2[k] \leftarrow mean(\boldsymbol{w}_2)$

$\boldsymbol{v}[k] \leftarrow \boldsymbol{v}_1[k]/\boldsymbol{v}_2[k]$

end for

return $mean(\boldsymbol{v})$, $sd(\boldsymbol{v})$, $skew(\boldsymbol{v})$

end procedure

Intra/Inter-ROI Top Positive Connectivity Ratio (IaIeTCR): It finds the top positive correlations of every seed voxel in a given seed ROI with all other target voxels in the same ROI and computes their average; it then finds the top positive correlations of each of the same seed voxels with all voxels in any target ROI, and computes its average. It finally computes the ratio of those two quantities and their returns average, standard deviation, and nonparametric skew across all seed voxels, for a total of $3N$ features.

procedure IaIeTCR($\boldsymbol{R}_i^j$, $\boldsymbol{R}_i^{\bar{i}}$, s)

$n \leftarrow nrows(\boldsymbol{R}_i^i)$

$n_s \leftarrow round(s(n-1))$

$m \leftarrow ncols(\boldsymbol{R}_i^j)$

$m_s \leftarrow round(sm)$

$\boldsymbol{v} \leftarrow n$-dimensional empty vector

$\boldsymbol{v}_1 \leftarrow n$-dimensional empty vector

$\boldsymbol{v}_2 \leftarrow n$-dimensional empty vector

$\boldsymbol{R}_i^i \leftarrow abs(\boldsymbol{R}_i^i)$

$R_i^{\bar{i}} \leftarrow abs(\boldsymbol{R}_i^{\bar{i}})$

for $k = 1, 2, ..., n$ **do**

$\boldsymbol{w}_1 \leftarrow \boldsymbol{R}_i^i[k, -k]$

$\boldsymbol{w}_1 \leftarrow$TPC$(\boldsymbol{w}_1, n_s)$

$\boldsymbol{w}_2 \leftarrow \boldsymbol{R}_i^j[k, :]$

$\boldsymbol{w}_2 \leftarrow$TPC$(\boldsymbol{w}_2, m_s)$

if $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ are both non empty **then**

$\boldsymbol{v}_1[k] \leftarrow mean(\boldsymbol{w}_1)$

$\boldsymbol{v}_2[k] \leftarrow mean(\boldsymbol{w}_2)$

$\boldsymbol{v}[k] \leftarrow \boldsymbol{v}_1[k]/\boldsymbol{v}_2[k]$

end if

end for

return $mean(\boldsymbol{v})$, $sd(\boldsymbol{v})$, $skew(\boldsymbol{v})$

end procedure

Top Positive Correlation (TPC) subroutine: For a given threshold s, $0 < s < 1$, it selects the largest $s \cdot 100\%$ of the elements of an input vector $\boldsymbol{v}$ by absolute value and returns a vector containing those of such elements that have positive value.

procedure TPC($\boldsymbol{v}$, s)

$n \leftarrow nrows(\boldsymbol{v})$

$n \leftarrow round(s \cdot n)$

$\boldsymbol{v}_A \leftarrow abs(\boldsymbol{v})$

$a \leftarrow order(\boldsymbol{v}_A)$

$a \leftarrow a[1 : n_s]$

$b \leftarrow$ {list of indices of voxels associated to positive elements in $\boldsymbol{v}$}

$c \leftarrow intersect(a, b)$

$\boldsymbol{v} \leftarrow \boldsymbol{v}[c]$

return $\boldsymbol{v}$

end procedure

Appendix C

MAR missingness modulation

Theorem C.1. *Consider a data matrix $\tilde{\boldsymbol{D}}$ containing n random independent observations $\tilde{\boldsymbol{d}}_i$, $i = 1, 2, \ldots, n$ on $p+1$ variables. Let $\pi_{1,i} = \frac{e^{\boldsymbol{\gamma}^T \tilde{\boldsymbol{d}}_i + \delta^*}}{e^{\boldsymbol{\gamma}^T \tilde{\boldsymbol{d}}_i + \delta^*} + 1}$ be the probability that $\tilde{\boldsymbol{d}}_i$ contains missing values, for a given vector $\boldsymbol{\gamma} \in \mathbb{R}^{p+1}$. For any $\pi_1 \in [0, 1]$, if $\delta^* \in \mathbb{R}$ is the unique solution of*

$$\boldsymbol{e}^T \left[e^{\left[-(\tilde{\boldsymbol{D}}\boldsymbol{\gamma} + \delta \boldsymbol{e})\right]} + \boldsymbol{e} \right]^{-1} - n\pi_1 = 0, \tag{C.1}$$

where $\boldsymbol{e}$ is an n-dimensional vector of ones, and $e^{[.]}$ and $[.]^{-1}$ are exponential and power functions applied element-wise to their argument, then the expected proportion of observations containing missing values is π_1.

Proof. Let M_i be a random variable with value 1 if observation i, $i = 1, 2, \ldots, n$ contains missing values and value 0 otherwise, and let $S_n = \sum_{i=1}^{n} M_i$. Trivially, the data set will include an expected proportion of π_1 observations with missing values if

$$\mathrm{E}[S_n] = n\pi_1 \tag{C.2}$$

Now, M_i, $i = 1, 2, \ldots, n$, are Bernoulli distributed with parameter $\pi_{1,i}$, and are independent of each other since $\tilde{\boldsymbol{d}}_i$, and consequently $\pi_{1,i}$, $i = 1, 2, \ldots, n$, are independent. Therefore, S_n

is Poisson Binomial distributed with parameters $\pi_{1,1}, \pi_{1,2}, \ldots, \pi_{1,n}$ and $\mathrm{E}[S_n] = \sum_{i=1}^{n} \pi_{1,i}$. Therefore we can write:

$$\begin{aligned}
\mathrm{E}[S_n] &= \sum_{i=1}^{n} \frac{e^{\boldsymbol{\gamma}^T \tilde{\boldsymbol{d}}_i + \delta^*}}{e^{\boldsymbol{\gamma}^T \tilde{\boldsymbol{d}}_i + \delta^*} + 1} \\
&= \sum_{i=1}^{n} \frac{1}{e^{-\left(\boldsymbol{\gamma}^T \tilde{\boldsymbol{d}}_i + \delta^*\right)}} \\
&= \boldsymbol{e}^T \left[e^{\left[-\left(\tilde{\boldsymbol{D}}\boldsymbol{\gamma} + \delta^* \boldsymbol{e}\right)\right]} + \boldsymbol{e} \right]^{-1}. \qquad \text{(C.3)}
\end{aligned}$$

The RHS of (C.2) is the sum of n monotone increasing functions of δ, each with range $[0, 1]$, and is therefore also a monotone increasing function, with range $[0, n]$. Equating the RHS of (C.2) to the RHS of (C.3) yields an equation with at most one solution, due to the monotonicity of (C.3). This equation, moreover, always has a solution, since the range of (C.2) is a subset of the range of (C.3), which proves the theorem. □

Bibliography

[1] U. Grömping, "Variable importance in regression models," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 7, no. 2, pp. 137–152, 2015.

[2] J. W. Johnson and J. M. Lebreton, "History and use of relative importance indices in organizational research," *Organizational Research Methods*, vol. 7, no. 3, pp. 238–257, 2004.

[3] J. Bi, "A review of statistical methods for determination of relative importance of correlated predictors and identification of drivers of consumer liking," *Journal of Sensory Studies*, vol. 27, no. 2, pp. 87–101, 2012.

[4] J. Johnson, "A heuristic method for estimating the relative weight of predictor variables in multiple regression," *Multivariate Behavioral Research*, vol. 35, no. 1, pp. 1–19, 2000.

[5] R. H. Lindeman, P. F. Merenda, and R. Z. Gold, *Introduction to Bivariate and Multivariate Analysis*. Glenview, IL: Scott Foresman, 1980.

[6] R. Azen and D. V. Budescu, "The dominance analysis approach for comparing predictors in multiple regression," *Psychological Methods*, vol. 8, no. 2, pp. 129–148, 2003.

[7] B. E. Feldman, "Relative importance and value." https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2255827, 2005. Accessed 11 October 2018.

[8] U. Grömping, "Estimators of relative importance in linear regression based on variance decomposition," *The American Statistician*, vol. 61, no. 2, pp. 139–147, 2007.

[9] L. Auret and C. Aldrich, "Empirical comparison of tree ensemble variable importance measures," *Chemometrics and Intelligent Laboratory Systems*, vol. 105, no. 2, pp. 157–170, 2011.

[10] K. J. Archer and R. V. Kimes, "Empirical characterization of random forest variable importance measures," *Computational Statistics and Data Analysis*, vol. 52, no. 4, pp. 2249–2260, 2008.

[11] D. Liakhovitski, Y. Bryukhov, and M. Conklin, "Relative importance of predictors: comparison of random forests with Johnson's relative weights," *Model Assisted Statistics and Applications*, vol. 5, no. 4, pp. 235–249, 2010.

[12] U. Grömping, "Variable importance assessment in regression: linear regression versus random forest," *The American Statistician*, vol. 63, no. 4, pp. 308–319, 2009.

[13] P. F. Zantek, G. P. Wright, and R. D. Plante, "Process and product improvement in manufacturing systems with correlated stages," *Management Science*, vol. 48, no. 5, pp. 591–606, 2002.

[14] F. Tsung, Y. Li, and M. Jin, "Statistical Process control for multistage manufacturing and service operations: a review and some extensions," *International Journal of Services Operations and Informatics*, vol. 3, no. 2, pp. 191–204, 2008.

[15] J. Shi and S. Zhou, "Quality control and improvement for multistage systems: a survey," *IIE transactions*, vol. 41, no. 9, pp. 744–753, 2009.

[16] J. Liu, "Variation reduction for multistage manufacturing processes: a comparison survey of statistical-process-control vs stream-of-variation methodologies," *Quality and Reliability Engineering International*, vol. 26, no. 7, pp. 645–661, 2010.

[17] G. Gazzola, J. Choi, D.-S. Kwak, B. K. Kim, D. M. Kim, S. H. Tong, and M.-K. Jeong, "Integrated variable importance assessment in multi-stage processes," *IEEE Transactions on Semiconductor Manufacturing*, vol. 31, no. 3, pp. 343–355, 2018.

[18] H. Wang, C. K. Chang, H.-I. Yang, and Y. Chen, "Estimating the relative importance of nodes in social networks," *Journal of Information Processing*, vol. 21, no. 3, pp. 414–422, 2013.

[19] S. Huang, T. Lv, X. Zhang, Y. Yang, W. Zheng, and C. Wen, "Identifying node role in social network based on multiple indicators," *PloS one*, vol. 9, no. 8, p. e103733, 2014.

[20] B. R. Memon, "Identifying important nodes in weighted covert networks using generalized centrality measures," in *Proceedings of the 2012 European Intelligence and Security Informatics Conference (EISIC)*, pp. 131–140, IEEE, 2012.

[21] K. Taha and P. Yoo, "SIIMCO: A forensic investigation tool for identifying the influential members of a criminal organization," *IEEE Transactions on Information Forensics and Security*, pp. 1–1, 2016.

[22] K. Taha and P. Yoo, "Using the spanning tree of a criminal network for identifying its leaders," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 2, pp. 445–453, 2016.

[23] C. Bergstrom, "Eigenfactor: measuring the value and prestige of scholarly journals," *College & Research Libraries News*, vol. 68, no. 5, pp. 314–316, 2007.

[24] Y. Ding, E. Yan, A. Frazho, and J. Caverlee, "Pagerank for ranking authors in cocitation networks," *Journal of the Association for Information Science and Technology*, vol. 60, no. 11, pp. 2229–2243, 2009.

[25] D. Kim, B. Lee, H. J. Lee, S. P. Lee, Y. Moon, and M. K. Jeong, "Automated detection of influential patents using singular values," *IEEE Transactions on Automation Science and Engineering*, vol. 9, no. 4, pp. 723–733, 2012.

[26] A. Rodriguez, B. Kim, J.-M. Lee, B.-Y. Coh, and M. K. Jeong, "Graph kernel based measure for evaluating the influence of patents in a patent citation network," *Expert Systems With Applications*, vol. 42, no. 3, pp. 1479–1486, 2015.

[27] J. B. Glattfelder and S. Battiston, "Backbone of complex networks of corporations: the flow of control," *Physical Review E*, vol. 80, no. 3, p. 036104, 2009.

[28] S. Vitali, J. B. Glattfelder, and S. Battiston, "The network of global corporate control," *PLoS ONE*, vol. 6, no. 10, p. e25995, 2011.

[29] E. Muller, P. I. Sánchez, Y. Mulle, and K. Bohm, "Ranking outlier nodes in subspaces of attributed graphs," in *Proceedings of the 2013 IEEE 29th International Conference on Data Engineering (ICDE)*, pp. 216–222, 2013.

[30] P. I. Sánchez, E. Müller, O. Irmler, and K. Böhm, "Local context selection for outlier ranking in graphs with multiple numeric node attributes," in *Proceedings of the 26th International Conference on Scientific and Statistical Database Management*, p. 16, 2014.

[31] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[32] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning. Data Mining, Inference, and Prediction, 2nd ed.* New York, NY: Springer, 2009.

[33] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & Software, 1984.

[34] C. Strobl, A.-L. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis, "Conditional variable importance for random forests," *BMC Bioinformatics*, vol. 9, no. 1, p. 307, 2008.

[35] K. Nicodemus, J. Malley, C. Strobl, and A. Ziegler, "The behaviour of random forest permutation-based variable importance measures under predictor correlation," *BMC Bioinformatics*, vol. 11, no. 1, p. 110, 2010.

[36] D. N. Reshef, Y. A. Reshef, H. K. Finucane, S. R. Grossman, G. McVean, P. J. Turnbaugh, E. S. Lander, M. Mitzenmacher, and P. C. Sabeti, "Detecting novel associations in large data sets," *Science*, vol. 334, no. 6062, pp. 1518–1524, 2011.

[37] T. Hothorn, K. Hornik, and M. A. Van De Wiel, "A Lego system for conditional inference," *The American Statistician*, vol. 60, no. 3, 2006.

[38] H. Strasser and C. Weber, "On the asymptotic theory of permutation statistics," *Mathematical Methods of Statistics*, vol. 2, no. 8, pp. 220–250, 1999.

[39] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, no. Mar, pp. 1157–1182, 2003.

[40] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers and Electrical Engineering*, vol. 40, no. 1, pp. 16–28, 2014.

[41] R. Sheikhpour, M. A. Sarram, S. Gharaghani, and M. A. Z. Chahooki, "A survey on semi-supervised feature selection methods," *Pattern Recognition*, vol. 64, pp. 141–158, 2017.

[42] R. Genuer, J.-M. Poggi, and C. Tuleau-Malot, "Variable selection using random forests," *Pattern Recognition Letters*, vol. 31, no. 14, pp. 2225–2236, 2010.

[43] V. Svetnik, A. Liaw, C. Tong, and T. Wang, "Application of Breiman's random forest to modeling structure-activity relationships of pharmaceutical molecules," in *Multiple Classifier Systems, Lecture Notes in Computer Science* (J. Kittler and T. Windeatt, eds.), vol. 3077, pp. 334–343, 2004.

[44] H. Jiang, Y. Deng, H. S. Chen, and L. Tao, "Joint analysis of two microarray gene-expression data sets to select lung adenocarcinoma marker genes," *BMC Bioinformatics*, vol. 5, no. 1, p. 81, 2004.

[45] R. Díaz-Uriarte and S. A. De Andres, "Gene selection and classification of microarray data using random forest," *BMC Bioinformatics*, vol. 7, no. 1, p. 3, 2006.

[46] H. Deng and G. Runger, "Gene selection with guided regularized random forest," *Pattern Recognition*, vol. 46, no. 12, pp. 3483–3489, 2013.

[47] H. Deng, "Guided random forest in the rrf package," *arXiv preprint arXiv:1306.0237*, 2013.

[48] E. Tuv, A. Borisov, G. Runger, and K. Torkkola, "Feature selection with ensembles, artificial variables, and redundancy elimination," *Journal of Machine Learning Research*, vol. 10, no. Jul, pp. 1341–1366, 2009.

[49] A. Altmann, L. Toloşi, O. Sander, and T. Lengauer, "Permutation importance: a corrected feature importance measure," *Bioinformatics*, vol. 26, no. 10, pp. 1340–1347, 2010.

[50] A. Hapfelmeier and K. Ulm, "A new variable selection approach using random forests," *Computational Statistics and Data Analysis*, vol. 60, no. 1, pp. 50–69, 2013.

[51] W. Rodenburg, A. Heidema, J. M. A. Boer, I. M. J. Bovee-Oudenhoven, E. J. M. Feskens, E. C. M. Mariman, and J. Keijer, "A framework to identify physiological responses in microarray-based gene expression studies: selection and interpretation of biologically relevant genes.," *Physiological Genomics*, vol. 33, no. 1, pp. 78–90, 2008.

[52] M. B. Kursa and W. R. Rudnicki, "Feature selection with the boruta package," *Journal of Statistical Software*, vol. 36, no. 11, pp. 1–13, 2010.

[53] Y.-S. Jeong, I.-H. Kang, M.-K. Jeong, and D. Kong, "A new feature selection method for one-class classification problems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, vol. 42, no. 6, pp. 1500–1509, 2012.

[54] Y.-S. Jeong, K. S. Shin, and M. K. Jeong, "An evolutionary algorithm with the partial sequential forward floating search mutation for large-scale feature selection problems," *Journal of the Operational Research Society*, vol. 66, no. 4, pp. 529–538, 2015.

[55] E. Tapia, P. Bulacio, and L. Angelone, "Sparse and stable gene selection with consensus svm-rfe," *Pattern Recognition Letters*, vol. 33, no. 2, pp. 164–172, 2012.

[56] F. Kovács, C. Legány, and A. Babos, "Cluster validity measurement techniques," in *6th International Symposium of Hungarian Researchers on Computational Intelligence*, pp. 388–393, 2005.

[57] J. C. Gower, "A general coefficient of similarity and some of its properties," *Biometrics*, vol. 27, no. 4, pp. 857–871, 1971.

[58] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *International Joint Conference on Artificial Intelligence*, pp. 1137–1145, 1995.

[59] J.-H. Kim, "Estimating classification error rate: repeated cross-validation, repeated hold-out and bootstrap," *Computational Statistics and Data Analysis*, vol. 53, no. 11, pp. 3735–3745, 2009.

[60] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: a K-means clustering algorithm," *Journal of the Royal Statistical Society, Series C*, vol. 28, no. 1, pp. 100–108, 1979.

[61] L. Kaufman and P. Rousseeuw, "Clustering by means of medoids," in *Statistical Data Analysis Based on the L_1-Norm and Related Methods* (Y. Dodge, ed.), pp. 405–416, North-Holland: Amsterdam, 1987.

[62] S. J. Smith, N. Ellis, and C. R. Pitcher, "Conditional variable importance in r package extendedforest." R vignette, http://gradientforest.r-forge.r-project.org/, 2014.

[63] K. W. Schaie, S. L. Willis, and G. I. L. Caskie, "The Seattle Longitudinal Study: relationship between personality and cognition," *Neuropsychology, Development, and Cognition Section B, Aging, Neuropsychology and Cognition*, vol. 11, no. 2–3, pp. 304–324, 2004.

[64] R. A. Fisher, *The Design of Experiments*. Edinburgh: Oliver And Boyd, 1937.

[65] A. Buchanan, G. Gazzola, and M. A. Bedau, "Evolutionary design of a model of self-assembling chemical structures," *Studies in Multidisciplinarity*, vol. 5, pp. 79–100, 2008.

[66] J. N. Cawse, G. Gazzola, and N. Packard, "Efficient discovery and optimization of complex high-throughput experiments," *Catalysis today*, vol. 159, no. 1, pp. 55–63, 2011.

[67] F. Caschera, G. Gazzola, M. A. Bedau, C. B. Moreno, A. Buchanan, J. Cawse, N. Packard, and M. M. Hanczyc, "Automated discovery of novel drug formulations using predictive iterated high throughput experimentation," *PLoS One*, vol. 5, no. 1, p. e8546, 2010.

[68] W. B. Powell, *Approximate dynamic programming: solving the curses of dimensionality*. New York: Wiley, 2007.

[69] P. O. Brown and D. Botstein, "Exploring the new world of the genome with dna microarrays," *Nature Genetics*, vol. 21, no. 1 s., p. 33, 1999.

[70] L. Breiman and J. H. Friedman, "Estimating optimal transformations for multiple regression and correlation," *Journal of the American Statistical Association*, vol. 80, no. 391, pp. 580–598, 1985.

[71] "Statlib archive." http://lib.stat.cmu.edu/datasets/. Accessed 11 October 2018.

[72] M. A. Little, P. E. McSharry, S. J. Roberts, D. A. Costello, and I. M. Moroz, "Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection," *BioMedical Engineering OnLine*, vol. 6, no. 1, p. 23, 2007.

[73] H. Drucker, C. J. C. Burges, L. Kaufman, A. J. Smola, and V. N. Vapnik, "Support vector regression machines," *Advances in Neural Information Processing Systems*, vol. 9, pp. 155–161, 1997.

[74] V. N. Vapnik, *The Nature of Statistical Learning*. New York: Springer-Verlag, 1995.

[75] A. J. Smola and B. Scholkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, 2004.

[76] H. Yang, L. Chan, and I. King, "Support vector machine regression for volatile stock market prediction," in *International Conference on Intelligent Data Engineering and Automated Learning*, pp. 391–396, 2002.

[77] C.-H. Wu, C.-C. Wei, D.-C. Su, M.-H. Chang, and J.-M. Ho, "Travel time prediction with support vector regression," *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 4, pp. 276–281, 2004.

[78] E. Myasnikova, A. Samsonova, M. Samsonova, and J. Reinitz, "Support vector regression applied to the determination of the developmental age of a drosophila embryo from its segmentation gene expression patterns," *Bioinformatics*, vol. 18, no. s1, pp. 87–95, 2002.

[79] Y. M. Wang, R. T. Schultz, R. T. Constable, and L. H. Staib, "Nonlinear estimation and modeling of fmri data using spatio-temporal support vector regression," in *Biennial International Conference on Information Processing in Medical Imaging*, pp. 647–659, 2003.

[80] J. D. Martín-Guerrero, G. Camps-Valls, E. Soria-Olivas, A. J. Serrano-López, J. J. Pérez-Ruixo, and N. V. Jiménez-Torres, "Dosage individualization of erythropoietin using a profile-dependent support vector regression," *IEEE Transactions on Biomedical Engineering*, vol. 50, no. 10, pp. 1136–1142, 2003.

[81] A. J. Smola, *Regression estimation with support vector learning machines*. PhD thesis, Master's thesis, Technische Universität München, 1996.

[82] D. Kim, C. Lee, S. Hwang, and M. K. Jeong, "A robust support vector regression with a linear-log concave loss function," *Journal of the Operational Research Society*, vol. 67, no. 5, pp. 735–742, 2016.

[83] O. P. Panagopoulos, P. Xanthopoulos, T. Razzaghi, and O. Şeref, "Relaxed support vector regression," *Annals of Operations Research*, pp. 1–20, 2018.

[84] C. C. Chang and C. J. Lin, "Training *nu*-support vector regression: theory and algorithms," *Neural Computation*, vol. 14, no. 8, pp. 1959–1978, 2002.

[85] C. Chen-Chia, S. Shun-Feng, J. Jin-Tsong, and H. Chih-Ching, "Robust support vector regression networks for function approximation with outliers," *IEEE Transactions on Neural Networks*, vol. 13, no. 6, pp. 1322–1330, 2002.

[86] S. Hwang, D. Kim, M. K. Jeong, and B.-J. Yum, "Robust kernel-based regression with bounded influence for outliers," *Journal of the Operational Research Society*, vol. 66, no. 8, pp. 1385–1398, 2015.

[87] J. I. Park, N. Kim, M. K. Jeong, and K. S. Shin, "Multiphase support vector regression for function approximation with break-points," *Journal of the Operational Research Society*, vol. 64, no. 5, pp. 775–785, 2013.

[88] K. Lee, N. Kim, and M. K. Jeong, "The sparse signomial classification and regression model," *Annals of Operations Research*, vol. 216, no. 1, pp. 257–286, 2014.

[89] P. Shivaswamy, C. Bhattacharyya, and A. Smola, "Second order cone programming approaches for handling missing and uncertain data," *The Journal of Machine Learning Research*, vol. 7, pp. 1283–1314, 2006.

[90] G. Huang, S. Song, C. Wu, and K. You, "Robust support vector regression for uncertain input and output data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 11, pp. 1690–1700, 2012.

[91] T. B. Trafalis and R. C. Gilbert, "Robust classification and regression using support vector machines," *European Journal of Operational Research*, vol. 173, no. 3, pp. 893–909, 2006.

[92] T. B. Trafalis and S. A. Alwazzi, "Support vector regression with noisy data: a second order cone programming approach," *International Journal of General Systems*, vol. 36, no. 2, 2007.

[93] D. H. Hong and C. Hwang, "Support vector fuzzy regression machines," *Fuzzy Sets and Systems*, vol. 138, no. 2, pp. 271–281, 2003.

[94] D. H. Hong and C. Hwang, "Extended fuzzy regression models using regularization method," *Information Sciences*, vol. 164, no. 1–4, pp. 31–46, 2004.

[95] E. Carrizosa, J. Gordillo, and F. Plastria, "Support vector regression for imprecise data," *Dept. MOSI, Vrije Univ. Brussel, Belgium, Tech. Rep.*, 2007.

[96] E. Carrizosa and J. Gordillo, "Kernel support vector regression with imprecise output," *Dept. MOSI, Vrije Univ. Brussel, Belgium, Tech. Rep.*, 2008.

[97] O. Mangasarian, J. Shavlik, and E. Wild, "Knowledge-based kernel approximation," *The Journal of Machine Learning Research*, vol. 5, pp. 1127–1141, 2004.

[98] O. Mangasarian and E. Wild, "Nonlinear knowledge in kernel approximation," *IEEE Transactions on Neural Networks*, vol. 18, no. 1, pp. 300–306, 2007.

[99] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge university press, 2004.

[100] D. B. Rubin, *Multiple Imputation for Nonresponse in Surveys*. New York:Wiley, 1987.

[101] D. B. Rubin, "Multiple imputation after 18+ years," *Journal of the American Statistical Association*, vol. 91, no. 434, pp. 473–518, 1996.

[102] S. Van Buuren, J. P. L. Brand, C. G. M. Groothuis-Oudshoorn, and D. B. Rubin, "Fully conditional specification in multivariate imputation," *Journal of Statistical Computation and Simulation*, vol. 76, no. 12, pp. 1049–1064, 2006.

[103] S. Van Buuren, H. C. Boshuizen, and D. L. Knook, "Multiple imputation of missing blood pressure covariates in survival analysis," *Statistics in Medicine*, vol. 18, no. 6, pp. 681–694, 1999.

[104] S. Van Buuren, "Multiple imputation of discrete and continuous data by fully conditional specification," *Statistical Methods in Medical Research*, vol. 16, no. 3, pp. 219–242, 2007.

[105] T. E. Raghunathan, J. M. Lepkowski, and J. Van Hoewyk, "A multivariate technique for multiply imputing missing values using a sequence of regression models," *Survey Methodology*, vol. 27, no. 1, pp. 85–95, 2001.

[106] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philosophical Magazine*, vol. 2, pp. 559–572, 1901.

[107] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *Journal of Educational Psychology*, vol. 24, pp. 417–441, 1933.

[108] I. T. Jolliffe, *Principal Component Analysis, 2nd ed.* New York: Spinger, 2002.

[109] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[110] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore and London: JHU Press, 2012.

[111] C. Dimitrov, D.and Knauer, K. Kriegel, and G. Rote, "On the bounding boxes obtained by principal component analysis," in *22nd European Workshop on Computational Geometry*, pp. 193–196, 2006.

[112] R. Little, "Missing-data adjustments in large surveys," *Journal of Business and Economic Statistics*, vol. 6, no. 3, pp. 287–296, 1988.

[113] N. Schenker and J. M. G. Taylor, "Partially parametric techniques for multiple imputation," *Computational Statistics and Data Analysis*, vol. 22, no. 4, pp. 425–446, 1996.

[114] S. V. Buuren and K. Groothuis-Oudshoorn, "mice: multivariate imputation by chained equations in R," *Journal of Statistical Software*, vol. 45, no. 3, pp. 1–68, 2010.

[115] R. J. Little and D. B. Rubin, *Statistical Analysis with Missing Data*. New York: Wiley, 2014.

[116] K. G. Frinston, "Functional and effective connectivity in neuroimaging: a synthesis," *Human Brain Mapping*, vol. 2, no. 1–2, pp. 56–78, 1994.

[117] O. Sporns, G. Tononi, and R. Kotter, "The human connectome: A structural description of the human brain," *PLoS Computational Biology*, vol. 1, no. 4, p. e42, 2005.

[118] B. Horwitz, "The elusive concept of brain connectivity," *NeuroImage*, vol. 19, no. 2, pp. 466–470, 2003.

[119] M. W. Cole and W. Schneider, "The cognitive control network: integrated cortical regions with dissociable functions," *Neuroimage*, vol. 37, no. 1, pp. 343–360, 2007.

[120] R. L. Buckner, J. R. Andrews-Hanna, and D. L. Schacter, "The brain's default network: anatomy, function, and relevance to disease," *Annals of the New York Academy of Sciences*, vol. 1124, pp. 1–38, 2008.

[121] C. Y. Wee, P.-T. Yap, K. Denny, J. N. Browndyke, G. G. Potter, K. A. Welsh-Bohmer, L. Wang, and D. Shen, "Resting-state multi-spectrum functional connectivity networks for identification of MCI patients," *PLosOne*, vol. 7, no. 5, p. e37828, 2012.

[122] W. A. Chaovalitwongse, D. Won, S. O., P. R. Borghesani, M. K. Askren, S. L. Willis, and T. J. Grabowski, "Network optimization of functional connectivity within default mode network regions to detect cognitive decline," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 7, pp. 1079–1089, 2017.

[123] C. Y. Wee, P.-T. Yap, K. Denny, J. N. Browndyke, G. G. Potter, K. A. Welsh-Bohmer, L. Wang, and D. Shen, "Identification of MCI individuals using structural and functional connectivity networks," *NeuroImage*, vol. 59, no. 3, pp. 2045–2056, 2012.

[124] B. A. Gutman, X. Hua, P. Rajagopalan, Y.-Y. Chou, Y. Wang, I. Yanovsky, A. W. Toga, C. R. Jack, M. W. Weiner, and P. M. Thompson, "Maximizing power to track Alzheimer's disease and MCI progression by LDA-based weighting of longitudinal ventricular surface features," *NeuroImage*, vol. 70, pp. 386–401, 2013.

[125] Y. Fan, N. Batmanghelich, C. M. Clark, and C. Davatziko, "Spatial patterns of brain atrophy in MCI patients, identified via high-dimensional pattern classification, predict subsequent cognitive decline," *NeuroImage*, vol. 39, no. 4, pp. 1731–1743, 2007.

[126] N. Singh, A. Y. Wang, P. Sankaranarayanan, P. T. Fletcher, and S. Joshi, "Genetic, structural and functional imaging biomarkers for early detection of conversion from MCI to AD," *International Conference on Medical Image Computing and Computer-Assisted Intervention, LNCS*, vol. 7510, pp. 132–140, 2012.

[127] D. Zhang, Y. Wang, L. Zhou, H. Yuan, and D. Shen, "Multimodal classification of Alzheimer's disease and mild cognitive impairment," *NeuroImage*, vol. 55, no. 3, pp. 856–867, 2011.

[128] C. Y. Wee, P.-T. Yap, L. W., K. Denny, J. N. Browndyke, G. G. Potter, K. A. Welsh-Bohmer, L. Wang, and D. Shen, "Enriched white matter connectivity networks for accurate identification of MCI patients," *NeuroImage*, vol. 54, no. 3, pp. 1812–1822, 2011.

[129] W. A. Chaovalitwongse, G. Presnyakov, Y. Cao, S. Sujitnapitsatham, D. Won, T. Madhyastha, K. E. Weaver, P. R. Borghesani, and T. J. Grabowski, "Diagnostic network modeling of neural connectivity using functional magnetic resonance imaging," in *Healthcare Intelligence: Turning Data into Knowledge* (H. Yang and E. Kundakcioglu, eds.), pp. 62–65, New York: IEEE Computer Society, 2014.

[130] G. Gazzola, J. Chou, W. A. Chaovalitwongse, and M. J. Jeong, "An introduction to the analysis of functional magnetic resonance imaging data," in *Optimization and Data Analysis in Biomedical Informatics* (M. P. Pardalos, T. F. Coleman, and P. Xanthopoulous, eds.), pp. 131–151, New York: Springer, 2012.

[131] K. K. Kwong, J. W. Belliveau, D. A. Chesler, I. E. Goldberg, R. M. Weissko, B. P. Poncelet, D. N. Kennedy, B. E. Hoppel, M. S. Cohen, R. Turner, H. M. Cheng, T. J. Brady, and B. R. Rosen, "Dynamic magnetic resonance imaging of human brain

activity during primary sensory stimulation," *Proceedings of the National Academy of Sciences (USA)*, vol. 89, pp. 5675–5679, 1992.

[132] N. Logothetis and A. Wandell, "Interpreting the BOLD signal," *Annual Review of Physiology*, vol. 66, pp. 735–769, 2004.