Linear predictor on linearly-generated data with missing values: non consistency and solutions

Marine Le Morvan [1 2] **Nicolas Prost**[1] **Julie Josse**[1 3] **Erwan Scornet**[3] **Gaël Varoquaux**[1 4]
[1] Université Paris-Saclay, Inria, CEA, Palaiseau, 91120, France
[2] Université Paris-Saclay, CNRS/IN2P3, IJCLab, 91405 Orsay, France
[3] CMAP, UMR7641, Ecole Polytechnique, IP Paris, 91128 Palaiseau, France
[4] Mila, McGill University, Montréal, Canada

Abstract

We consider building predictors when the data have missing values. We study the seemingly-simple case where the target to predict is a linear function of the fully-observed data and we show that, in the presence of missing values, the optimal predictor is not linear in general. In the particular Gaussian case, it can be written as a linear function of multiway interactions between the observed data and the various missing-value indicators. Due to its intrinsic complexity, we study a simple approximation and prove generalization bounds with finite samples, highlighting regimes for which each method performs best. We then show that multilayer perceptrons with ReLU activation functions can be consistent, and can explore good trade-offs between the true model and approximations. Our study highlights the interesting family of models that are beneficial to fit with missing values depending on the amount of data available.

1 Introduction

Increasing data sizes and diversity naturally entail more and more missing values. Data analysis with missing values has been extensively studied in the statistical literature, with the leading work of Rubin (1976). However, this literature (Little and Rubin, 2002; van Buuren, 2018; Josse et al., 2019a) does not address the questions of modern statistical learning. First it focuses on estimating parameters and their variance, of a distribution –joint or conditional– as in the linear model (Little, 1992; Jones, 1996). This is typically done using either likelihood inference based on expectation maximization (EM) algorithms (Dempster et al., 1977) or multiple imputation (van Buuren, 2018). Second, a large part of the literature only considers the restricted "missing at random" mechanism (Rubin, 1976) as it allows maximum-likelihood inference while ignoring the missing values distribution. "Missing non at random" mechanisms are much harder to address, and the literature is thin, focusing on detailed models of the missingness for a specific application such as collaborative filtering (Marlin and Zemel, 2009) or on missing values occurring only on few variables (Kim and Ying, 2018). Statistical estimation often hinges on parametric models for the data and the missingness mechanism (except, for example, Mohan and Pearl, 2019). Finally, only few notable exceptions study supervised-learning settings, where the aim is to predict a target variable given input variables and the missing values are both in the training and the test sets (Zhang et al., 2005; Pelckmans et al., 2005; Liu et al., 2016; Josse et al., 2019b). Machine-learning techniques have been extensively used to impute missing values (Lakshminarayan et al., 1996; Yoon et al., 2018). However imputation is a different problem from predicting a target variable and good imputation does not always lead to good prediction (Zhang et al., 2005; Josse et al., 2019b).

As surprising as it sounds, even the linear model, the simplest instance of regression models, has not been thoroughly studied with missing values and reveals unexpected challenges. This can be explained because data with missing values can be seen mixed continuous (observed values) and categorical (missing-values indicators) data. In comparison to decision trees for instance, linear models are less well-equipped by design to address such mixed data.

Proceedings of the 23rdInternational Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

After establishing the problem of risk minimization in supervised-learning settings with missing values, the first contribution of this paper is to develop the Bayes predictor under common Gaussian assumption. We highlight that the resulting problem of linear model with missing values is no longer linear. We use these results to introduce two approaches to estimate a predictor, one based directly on the Bayes-predictor expression, which boils down to performing one linear model per pattern of missing values, and one derived from a linear approximation, which is equivalent to imputing missing values by a constant and concatenating the pattern of missing values to the imputed design matrix. We derive new generalization bounds for these two estimates, therefore establishing the regimes in which each estimate has higher performance. Due to the complexity of the learning task, we study the benefit of using multilayer perceptron (MLP), a good compromise between the complexity of the first approach and the extreme simplicity of the second one. We show its consistency with enough hidden units. Finally, we conduct experimental studies that show that the MLP often gives the best prediction and can appropriately handle MNAR data.

2 Problem setting

Notation (Missing values)**.** Throughout the paper, missing values are represented with the symbol `na` satisfying, for all $x \in \mathbb{R}^\star$, $\texttt{na} \cdot \texttt{x} = \texttt{na}$ and $\texttt{na} \cdot \texttt{0} = \texttt{0}$.

Let us consider* a data matrix $\mathbf{x}_n \in \mathbb{R}^{n \times d}$ and a response vector $\mathbf{y}_n \in \mathbb{R}^n$, such as

$$\mathbf{x}_n = \begin{pmatrix} 9.1 & 8.5 \\ 2.1 & 3.5 \\ 6.7 & 9.6 \\ 4.2 & 5.5 \end{pmatrix}, \quad \mathbf{y}_n = \begin{pmatrix} 4.6 \\ 7.9 \\ 8.3 \\ 4.6 \end{pmatrix}.$$

However, only the incomplete design matrix $\mathbf{z}_n$ is available. Letting $\widetilde{\mathbb{R}} = \mathbb{R} \cup \{\texttt{na}\}$, the incomplete design matrix $\mathbf{z}_n$ belongs to $\widetilde{\mathbb{R}}^{n \times d}$. More precisely, denoting by $\mathbf{m}_n \in \{0,1\}^{n \times d}$ the positions of missing entries in $\mathbf{z}_n$ (1 if the entry is missing, 0 otherwise), $\mathbf{z}_n$ can be written as $\mathbf{z}_n = \mathbf{x}_n \odot (\mathbf{1} - \mathbf{m}_n) + \texttt{na} \odot \mathbf{m}_n$, where $\odot$ is the term-by-term product. In summary, the observed data are given by

$$\mathbf{z}_n = \begin{pmatrix} 9.1 & 8.5 \\ 2.1 & \texttt{na} \\ \texttt{na} & 9.6 \\ \texttt{na} & \texttt{na} \end{pmatrix}, \quad \mathbf{m}_n = \begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{pmatrix}, \quad \mathbf{y}_n = \begin{pmatrix} 4.6 \\ 7.9 \\ 8.3 \\ 4.6 \end{pmatrix},$$

and are supposed to be n i.i.d. realizations of generic random variables Z, M, Y.

*Writing conventions used in this paper are detailed in appendix A.

Notation (Observed indices)**.** For all values m of a mask vector M, $obs(m)$ (resp. $mis(m)$) denote the indices of the zero entries of m (resp. non-zero). For instance, if $z = (3.4, 4.1, \texttt{na}, 2.6)$, then $m = (0,0,1,0)$, $mis(m) = \{2\}$ and $obs(m) = \{0,1,3\}$.

Throughout, the target Y depends linearly on X, that is, there exist $\beta_0 \in \mathbb{R}$ and $\beta \in \mathbb{R}^d$ such that

$$Y = \beta_0 + \langle X, \beta \rangle + \varepsilon, \text{ where } \varepsilon \sim \mathcal{N}(0, \sigma^2). \quad (1)$$

A natural loss function in the regression framework is the square loss $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_+$ defined as $\ell(y, y') = (y - y')^2$. The Bayes predictor $f^\star$ associated to this loss is the best possible predictor, defined by

$$f^\star \in \operatorname*{argmin}_{f : \widetilde{\mathbb{R}}^d \to \mathbb{R}} R(f), \quad (2)$$

where

$$R(f) \coloneqq \mathbb{E}[\ell(f(Z), Y)].$$

Since we do not have access to the true distribution of (X, Z, M, Y), an estimate $\hat{f}$ is typically built by minimizing the empirical risk over a class of functions $\mathcal{F}$ (Vapnik, 1992). This is a well-studied problem in the complete case: efficient gradient descent-based algorithms can be used to estimate predictors, and there are many empirical and theoretical results on how to choose a good parametric class of functions $\mathcal{F}$ to control the generalization error. Because of the semi-discrete nature of $\widetilde{\mathbb{R}}$, these results cannot be directly transposed to data with missing values.

3 Optimal imputation

The presence of missing values makes empirical risk minimization –optimizing an empirical version of (2)– untractable. Indeed, $\widetilde{\mathbb{R}}^d$ is not a vector space, therefore incapacitating gradient-based algorithms. Hence, solving (2) in presence of missing values requires a specific strategy.

Imputing by an optimal constant The simplest way to deal with missing values is to inject the incomplete data into $\mathbb{R}^d$. The easiest way to do so is to use constant imputation, *i.e.* impute each feature Z_j by a constant c_j: the most common choice is to impute by the mean or the median. However, it is also possible to optimise the constant with regards to the risk.

Proposition 3.1 (Optimal constant in linear model)**.** *The imputation constants $(c_j^\star)_{j \in [\![1,d]\!]}$ optimal to minimize the quadratic risk in a linear model can be easily computed by solving a linear model with a design matrix constructed by imputing X with zeros and concatenating the mask M as additional features (see Appendix A).*

In an inferential framework, Jones (1996) showed that constant imputation leads to regression parameters that are biased compared to parameters on the fully-observed data. We differ from Jones because our aim is prediction rather than estimation. Indeed, minimizing a prediction risk with missing values is different from recovering the behavior without missing values (Josse et al., 2019b). Nevertheless, the strategy of replacing missing values with a constant does not lead to Bayes-consistent predictors in the general setting, and even under a Gaussian assumption as shown in Section 4. In the general case, the problem can be illustrated by the following example which shows that the model is no longer linear when values are missing.

The best predictor need not be linear

Example 3.1 (Non-linear submodel)**.** *Let $Y = X_1 + X_2 + \varepsilon$, where $X_2 = \exp(X_1) + \varepsilon_1$. Now, assume that only X_1 is observed. Then, the model can be rewritten as*

$$Y = X_1 + \exp(X_1) + \varepsilon + \varepsilon_1,$$

where $f(X_1) = X_1 + \exp(X_1)$ is the Bayes predictor. In this example, the submodel for which only X_1 is observed is not linear.

From Example 3.1, we deduce that there exists a large variety of submodels for a same linear model. In fact, the submodel structures depend on the structure of X and on the missing-value mechanism. Therefore, an extensive analysis seems unrealistic. Below, we show that in the particular case of Gaussian generative mechanisms submodels can be easily expressed, hence the Bayes predictor for each submodel can be computed exactly.

4 Bayes predictor

We now derive the expression of $\mathbb{E}[Y|Z]$ under Model (1) with missing values ($Z \in \widetilde{\mathbb{R}}^d$), as it gives the Bayes-optimal predictor for the square loss (Bishop, 2006).

The Bayes predictor can be written as

$$\begin{aligned} f^\star(Z) &= \mathbb{E}\left[Y \mid Z\right] \\ &= \mathbb{E}\left[Y \mid M, X_{obs(M)}\right] \\ &= \sum_{m \in \{0,1\}^d} \mathbb{E}\left[Y | X_{obs(m)}, M = m\right] \mathbb{1}_{M=m}. \end{aligned}$$

This formulation already highlights the combinatorial issues: as suggested by Rosenbaum and Rubin (1984, Appendix B), estimating $f^\star(Z)$ may require to estimate 2^d different submodels.

As shown by Example 3.1, controlling the form of $f^\star$ requires assumptions on the conditional relationships across the features X_j. To ground our theoretical derivations, we use the very common pattern mixture model (Little, 1993), with Gaussian distributions:

Assumption 4.1 (Gaussian pattern mixture model)**.** *The distribution of X conditional on M is Gaussian, that is, for all $m \in \{0,1\}^d$, there exist μ_m and Σ_m such that*

$$X \mid (M = m) \sim \mathcal{N}(\mu_m, \Sigma_m).$$

A particular case of this distribution is the case where X is Gaussian and independent of M.

Proposition 4.1 (Expanded Bayes predictor)**.** *Under Assumption 4.1 and Model (1), the Bayes predictor $f^\star$ takes the form*

$$f^\star(Z) = \langle W, \delta \rangle, \tag{3}$$

where the parameter $\delta \in \mathbb{R}^p$ is a function of β, $(\mu^m)_{m \in \{0,1\}^d}$ and $(\Sigma^m)_{m \in \{0,1\}^d}$, and the random variable $W \in \mathbb{R}^p$ is the concatenation of $j = 1, \ldots, 2^d$ blocks, each one being

$$\left(\mathbb{1}_{M=m_j} \ , \ X_{obs(m_j)} \mathbb{1}_{M=m_j}\right).$$

An interesting aspect of this result is that the Bayes predictor is a linear model, though not on the original data matrix X. Indeed, W and δ are vectors composed of 2^d blocks, for which only one block is "switched on" – the one corresponding to the observed missing pattern M. Elements of W of this block are the observed values for X and elements of δ of the same block are the linear coefficients corresponding to the observed missingness pattern. Equation (3) can thus be seen as the concatenation of each of the 2^d submodels, where each submodel corresponds to a missing pattern. Denoting by $\mathbf{x}^{(m_j)}$ the design matrix $\mathbf{x_n}$ restricted to the samples with missing data pattern m_j and to the columns $obs(m_j)$, then the matrix W looks like:

$$\begin{pmatrix} (1, \mathbf{x}^{(m_1)}) & 0 & & 0 \\ 0 & (1, \mathbf{x}^{(m_2)}) & \cdots & 0 \\ & \vdots & & \\ 0 & 0 & & (1, \mathbf{x}^{(m_{2^d})}) \end{pmatrix}.$$

The Bayes predictor can also be expressed in a second way, as shown in Proposition 4.2.

Proposition 4.2 (Factorized Bayes predictor)**.** *We have*

$$f^\star(Z) = \sum_{\mathcal{S} \subset [\![1,d]\!]} \left(\zeta_0^{\mathcal{S}} + \sum_{j=1}^{d} \zeta_j^{\mathcal{S}} (1 - M_j) X_j \right) \prod_{k \in \mathcal{S}} M_k, \tag{4}$$

where the parameter $\zeta \in \mathbb{R}^p$ is a function of δ. In addition, one can write

$$Y = f^\star(Z) + \text{noise}(Z),$$

with $\text{noise}(Z) = \varepsilon + < \sqrt{T_M}\Xi, \beta_{mis(M)} >$, *and* $\Xi \sim \mathcal{N}(0, I_d)$, *where* $T_M = \text{Var}\left(X_{mis(M)}|X_{obs(M)}, M\right)$ *and* $\sqrt{}$ *denotes the square root of a positive definite symmetric matrix.*

Expression (4) is a polynome of X and cross-products of M. As such, it is more convenient than expression (3) to compare to simpler estimates, as it can be truncated to low-order terms. This is done hereafter. Note that the multiplication $(1-M_j)X_j$ means that missing terms in X_j are imputed by zeros.

Proofs of Proposition 4.1 and 4.2 can be found in the Appendix B. Thanks to these explicit expressions, the Bayes risk can be computed exactly as shown in Appendix C. The value of the Bayes risk is extensively used in the Experiments (Section 7) to evaluate the performance of the different methods.

The integer p in equation (3) is the total number of parameters of the model which can be calculated by considering every sublinear model:

$$p = \sum_{k=0}^{d} \binom{d}{k} \times (k+1) = 2^{d-1} \times (d+2). \qquad (5)$$

Strikingly, the Bayes predictor gathers 2^d submodels. When d is not small, estimating it from data is therefore a high-dimensional problem, with computational and statistical challenges. For this reason, we introduce hereafter a low-dimensional linear approximation of $f^\star$, without interaction terms. Indeed, the expression in Proposition 4.1 is not linear in the original features and their missingness, but rather entails a combinatorial number of non-linearly derived features.

Definition 4.1 (Linear approximation)**.** We define the linear approximation of $f^\star$ as

$$f^\star_{\text{approx}}(Z) = \beta^\star_{0,0} + \sum_{j=1}^{d} \beta^\star_{j,0} \mathbb{1}_{M_j=1} + \sum_{j=1}^{d} \beta^\star_j X_j \mathbb{1}_{M_j=0}$$

$f^\star_{\text{approx}}(Z)$ is a linear function of the concatenated vector (X, M) where X is imputed by zeros, enabling a study of linear regression with that input. Note that this approximation is the same as defined in Proposition 3.1.

5 Finite sample bounds for linear predictors

The above expression of the Bayes predictor leads to two estimation strategies with linear models. The first model is the direct empirical equivalent of the Bayes predictor, using a linear regression to estimate the terms in the expanded Bayes predictor (Proposition 4.1). It is a rich model, powerful in low dimension, but it is costly and has large variance in higher dimension. The second model is the approximation of the first given in Definition 4.1. It is has a lower approximation capacity but also a smaller variance since it contains fewer parameters.

For the theoretical analysis, we focus in this Section on the risk between the estimate and the Bayes predictor $f^\star(Z)$. We thus consider the new framework below to handle our analysis.

Assumption 5.1. *We have $Y = f^\star(Z) + noise(Z)$ as defined in Section 4, where $noise(Z)$ is a centred noise conditional on Z and such that there exists $\sigma^2 > 0$ satisfying $\mathbb{V}[Y|Z] \leq \sigma^2$ almost surely. Besides, assume that $\|f^\star\|_\infty < L$ and $Supp(X) \subset [-1, 1]^d$.*

For all $L > 0$ and for all function f, we define the clipped version $T_L f$ of f at level L by, for all x,

$$T_L f(x) = \begin{cases} f(x) & \text{if } |f(x)| \leq L \\ L \text{ sign}(f(x)) & \text{otherwise} \end{cases}$$

5.1 Expanded linear model

The expanded linear model is well specified, as the Bayes predictor detailed in Proposition 4.1 belongs to the model.

Theorem 5.1 (Expanded model)**.** *Grant Assumption 5.1. Let $f_{\hat{\beta}_{expanded}}$ be the linear regression estimate for the expanded model (see Proposition 4.1) computed via Ordinary Least Squares (OLS). Then, the risk of its predictions clipped at L satisfies*

$$R(T_L f_{\hat{\beta}_{expanded}}) \leq c \max\{\sigma^2, L^2\} \frac{2^{d-1}(d+2)(1+\log n)}{n} + \sigma^2.$$

Additionally, let $\mathcal{M} = \{m \in \{0,1\}^d, \mathbb{P}[M = m] > 0\}$ be the set of all possible missing patterns and assume that there exists $\alpha \in (0, 1]$ such that $|\mathcal{M}| = \alpha 2^d$. Then, there exists a constant $c_1 > 0$, such that for all n large enough,

$$R(T_L f_{\hat{\beta}_{expanded}}) \geq \sigma^2 + \frac{2^d c_1}{n+1}.$$

The proof is provided in Appendix D. Theorem 5.1 implies that the excess risk of the linear estimate of the expanded model is of order $\mathcal{O}(2^d/n)$ which grows exponentially fast with the original dimension of the problem.

5.2 Constant imputation via linear approximation

Theorem 5.2 (Linear approximation model)**.** *Grant Assumption 5.1. Let $f_{\hat{\beta}_{approx},L}$ be the linear regression estimate for the approximated model (Definition 4.1), computed via OLS. Then, the risk of its predictions clipped at L satisfies*

$$R(T_L f_{\hat{\beta}_{approx}}) \leq \sigma^2 + c\max\{\sigma^2, L^2\}\frac{2d(1+\log n)}{n} + 64(d+1)^2L^2$$

The proof is provided in Appendix D. A direct consequence of Theorem 5.2 is that the excess risk of the linear approximation of $f^\star$ (Definition 4.1) is $\mathcal{O}(d^2+d/n)$.

Comparing this upper bound to the one obtained for the expanded model, we see that the risk of the expanded model is lower than that of the approximated model when $n \gg \frac{2^d}{d}$. Therefore, the expanded model is only useful for very small data set (large n, small d), but for data sets of reasonable size, the linear approximation may be preferred. Nevertheless, as detailed in Section 6, multilayers neural nets can be used as a compromise between both approaches.

6 Multilayer perceptron

6.1 Consistency

Theorem 6.1 (MLP)**.** *Assume that the Bayes predictor takes the form described in proposition 4.1, and that the support of X is finite. A MLP i) with one hidden layer containing 2^d hidden units, ii) ReLU activation functions iii) which is fed with the concatenated vector $(X \odot (\mathbf{1} - M), M)$, is Bayes consistent.*

The complete proof of Theorem 6.1 is given in Appendix E, but we provide here the main ideas. The activation for each hidden unit is a linear function of a design matrix constructed by imputing X with zeros and concatenating with the mask M. Thus, just like for the optimal imputation problem of Proposition 3.1, the weights of the linear function can be seen as either regular linear regression weights for the observed variables or learned imputation constants for the missing ones. In the context of a MLP with 2^d hidden units, we have 2^d such linear functions, meaning that each hidden unit is associated with one learned imputation vector. It turns out, as shown in the proof, that it is possible to choose the imputation vector of each hidden unit so that one hidden unit is always activated by points with a given missing-values pattern m but never by points with another missing-values pattern $m' \neq m$. As a result, all points with a given missing-values pattern fall into their own affine region, and it is then possible to adjust the weights so that the slope and bias of each affine region equals those of the Bayes predictor.

The number of parameters of a MLP with one hidden layer and 2^d units is $(d+1)2^{d+1}+1$. Compared to the expanded Bayes predictor which has $(d+1)2^{d-1}$ parameters, this is roughly 4 times more. This comes from the fact that the MLP does not directly estimate the slope and bias of a linear model per missing-values pattern. First, for each affine region associated to a missing-values pattern, it estimates a slope for all variables, and not only for the observed ones. This doubles the number of parameters to be estimated. Second, it also needs to estimate the imputation constants for each hidden unit (or equivalently missing-values pattern) which again doubles the number of parameters to be estimated. As a result, the MLP should require more samples than the expanded Bayes predictor to achieve the Bayes rate. However, as we discuss below the parametrization of the MLP provides a natural way to control the capacity of the model. By contrast, there is no such easy and natural way to control the capacity of the expanded Bayes predictor.

6.2 Trading off estimation and approximation error.

The prediction function of a MLP with one hidden layer and n_h hidden units is a piecewise affine function with at most $\sum_{j=0}^{d}\binom{n_h}{j}$ regions (Pascanu et al., 2013). Thus, choosing $n_h = d$, we obtain 2^d affine regions, so potentially one per missing-value pattern. However, the slopes and biases of these affine regions are not independent, since they are linear combinations of the weights associated to each hidden unit. Yet, if the data-generating process has more regularities, 2^d different slopes may not be necessary to approximate it well. Varying the number of hidden units n_h thus explores an interesting tradeoff between model complexity –which comes at the cost of estimation error– and approximation error, to successfully address medium sample sizes problems.

7 Empirical study

We now run an empirical study to illustrate our theoretical results, but also to explore how the different bias-variance trade-offs of the various models introduced affect prediction errors depending on the amount of data available. The code to reproduce the experiments in available on GitHub [†].

[†] https://github.com/marineLM/linear_predictor_missing

7.1 Experimental settings

Simulation models The data (X, M) is generated according to three different models. Two of them are instances of Assumption 4.1 while the third one is a classical Missing Non At Random (MNAR) model (Little and Rubin, 2002).

mixture 1 The first model assumes that the data is generated according to Assumption 4.1 with only one Gaussian component shared by all missing-values patterns. This boils down to a classical Missing Completely At Random (MCAR) setting, where $X \sim \mathcal{N}(X|\mu, \Sigma)$ and missing values are introduced uniformly at random, independent of X.

mixture 3 The second model assumes that the data is generated according to Assumption 4.1 with three Gaussian components. Each missing-values pattern is associated to one of the three Gaussian components in such a way that each component is associated with the same number of missing-values patterns.

selfmasking The last model assumes that the data is generated according to a single Gaussian, and that missing values are introduced according to a probit model parametrized by λ and μ_0 as $P(M = 1|X_j) = \text{Probit}(\lambda_j(X_j - \mu_0))$. This model allows to increase the probability of introducing missing values when the variable increases (or decreases), hence the denomination *selfmasking*. It is a classical instance of a Missing Non At Random (MNAR) problem. Estimation in MNAR settings is notoriously difficult as most approaches, such as EM, rely on ignoring –marginalizing– the unobserved data which then introduces biases.

For the three models, covariances for the Gaussian distributions are obtained as $BB^\top + D$ where $B \in \mathbb{R}^{d \times \lfloor \frac{d}{2} \rfloor}$ is drawn from a standard normal distribution and D is a diagonal matrix with small positive elements to make the covariance matrix full rank. This gives covariance matrices with some strong correlations.

For *mixture 1* and *mixture 3*, missing values are introduced in such a way that each missing-values pattern in equiprobable. For *selfmasking*, the parameters of the probit function are chosen so that the missing rate for each variable is 25%.

The response Y is generated by a linear combination of the input variables as in Equation 1. Note that to generate Y, we use the complete design matrix (without missing values). In these experiments, the noise ε is set to 0 and the regression coefficients β, β_0 are chosen equal to $\beta_0 = 1$ and $\beta = (1, 1, \ldots, 1)$.

Estimation approaches For these three simulation scenarios, we compare four approaches:

ConstantImputedLR: optimal imputation method described in Proposition 3.1 The regression is performed using ordinary least squares.

EMLR: EM is used to fit a multivariate normal distribution for the $p + 1$-dimensional random variable $(X_1, \ldots, X_p, Y)$. Denoting by $\mu = (\mu_X, \mu_Y) \in \mathbb{R}^{p+1}$ the estimated mean and by Σ the estimated covariance matrix (with blocks $\Sigma_{XX} \in \mathbb{R}^{p \times p}$, $\Sigma_{XY} \in \mathbb{R}^{p \times 1}$, and $\Sigma_{YY} \in \mathbb{R}$), the predictor used is:

$$\mathbb{E}\left\{Y|X_{obs(M)}, M\right\} = \mu_Y + \Sigma_{Y,obs(M)}\Sigma^{-1}_{obs(M)}\left(X_{obs(M)} - \mu_{obs(M)}\right)$$

as can be obtained from the conditional Gaussian formula. Since EM directly estimates β, μ and Σ, it only has to estimate $(d + 1) + d + \frac{d(d+1)}{2} = \frac{d(d+5)}{2} + 1$ parameters, while ExpandedLR needs to estimate $2^{d-1}(d+2)$ parameters. However, this method is expected to perform well only in the *mixture 1* setting.

MICELR: Multivariate Imputation by Chained Equations (MICE) is a widely used and effective conditional imputation method. We used the `IterativeImputer` of scikit-learn which adapts MICE to out-of-sample imputation.

ExpandedLR: full linear model described in Proposition 4.1. When the number of samples is small compared to the number of missing-values patterns, it may happen that for a given pattern, we have fewer samples than parameters to estimate. For this reason, we used ridge regression on the expanded feature set, and the regularization parameter was chosen by cross-validation over a small grid of values (10^{-3}, 1, 10^3). The data is standardized before fitting the model.

MLP: Multilayer perceptron with one hidden layer whose size is varied between and 1 and 2^d hidden units. The input that is fed to the MLP is (X, M) where X is imputed with zeros and M is the mask. Rectified Linear Units (ReLU) were used as activation functions. The MLP was optimized with Adam, and a batch size of 200 samples. Weight decay was applied and the regularization parameter chosen by cross-validation over a small grid of values $(10^{-1}, 10^{-2}, 10^{-4})$. The data is standardized before fitting the model.

When referring to a MLP in the figures, we use the notation *MLP_Wx* where x is a value indicating the

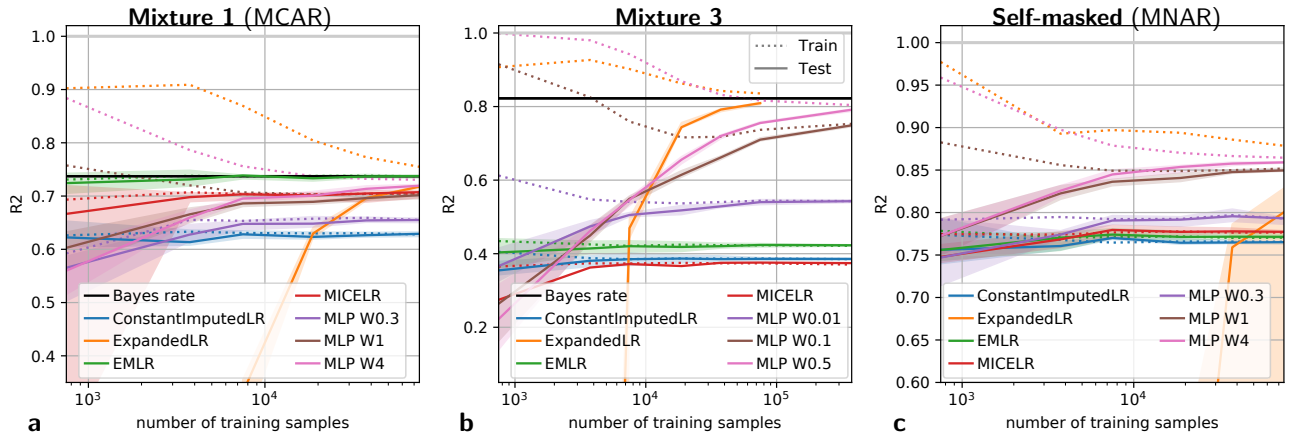

Figure 1: **Learning curves** Train and test R2 scores (respectively in dotted and in plain lines) as a function of the number of training samples, for each data type. Experiments were carried out in dimension $d = 10$. The curves display the mean and 95% confidence interval over 5 repetitions. The black horizontal line represents the Bayes rate (best achievable performance). Figure 3, in the supp. mat. gives a box plot at $n = 75\,000$.

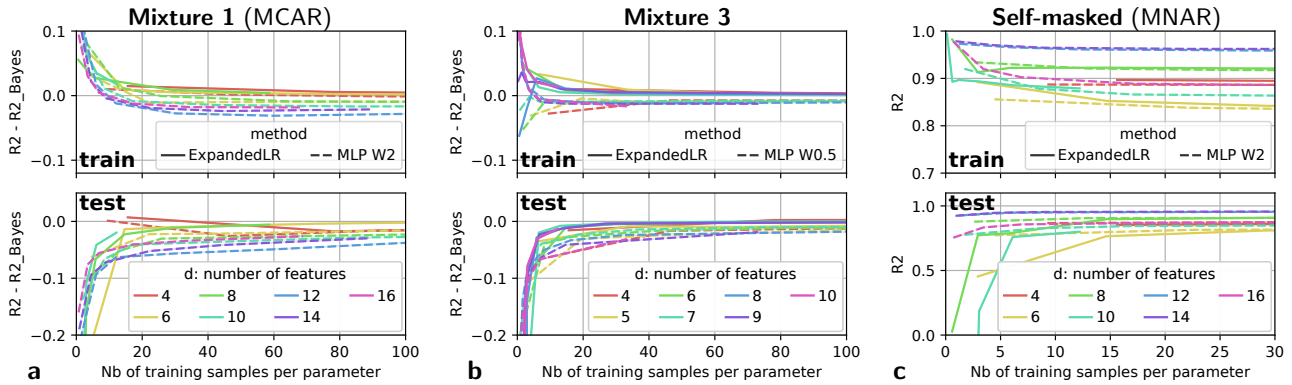

Figure 2: **Learning curves varying the dimensionality** for the MLPs and ExpandedLR. For a given number of features d, the number of hidden units used is $2d$ for mixture 1 and self-masked and 0.5×2^d for mixture 3.

number of units used. For an experiment in dimension d, *MLP_Wx* refers to a MLP with $x \times 2^d$ hidden units for *mixture 3*, or a MLP with $x \times d$ hidden units for *MCAR* and *MNAR*. The reason why we use this notation is because to achieve the same performance level for different dimensions, we must use a number of hidden units that is proportional to 2^d or to d according to the data type (See Appendix F.2). In the experiments, we test three different hidden layer sizes, to compare low, medium and high capacities.

All models are compared to the Bayes rate computed in Appendix C whenever possible, i.e. for *mixture 1* and *mixture 3*. In all experiments, datasets are split into train and test set (75% train and 25% test) and the performance reported is the R2 score of the predictor on the test set (or the difference between the R2 score of the predictor and that of the Bayes predictor).

7.2 Results

Mixture 1* and *Mixture 3 We first focus on the data types satisfying Assumption 4.1, i.e, *mixture 1* and *mixture 3*, since the theory developed in this paper applies for these cases.

Figure 1 (a and b) presents the learning curves for the various methods, as the number of samples increases from 10^3 to 10^5 (5×10^5 for *mixture 3*) and the dimension is fixed to $d = 10$. First of all, this figure experimentally confirms that ExpandedLR and the MLP are Bayes consistent. With enough samples, both methods achieve the best possible performance. It also confirms that ExpandedLR cannot be used in the small n large d regime. Indeed, between 10,000 and 20,000 samples are required for ExpandedLR to reach acceptable performances.

Overall, the learning curves (Figure 1) highlight three sample size regimes. We have a small sample size

regime ($n < 1,000$) where EM is the best option. Indeed, EM is Bayes consistent for *mixture 1*. For *mixture 3*, EM performs badly but is not worse than the other methods in the small sample size regime. It is slightly better than ConstantImputedLR which still remains a reasonable option in this regime. MICELR follows the behavior of EM with slightly worse performance.

For $n > 30,000$ in *MCAR* and $n > 10,000$ in *mixture 3*, we are in a large sample size regime where ExpandedLR is an excellent choice, with performances on par or better than those of the MLP. The observation of small and large sample regimes support the theoretical analysis of Section 5. The fact that ExpandedLR outperforms the MLP for a larger sample range in *mixture 3* ($n > 10,000$) compared to *mixture 1* ($n > 30,000$) is explained by the fact that, to reach a given performance, the MLP needs fewer parameters in *mixture 1* than in *mixture 3*, and thus fewer samples.

Finally, for $1,000 < n < 10,000$ or $1,000 < n < 30,000$, we have a last regime where the MLP should be the preferred option, since it outperforms both ConstantImputedLR and ExpandedLR. It shows that for medium size samples, it is possible to adapt the width of the hidden layer to reach a beneficial compromise between estimation and approximation error. This is particularly useful since many real datasets fall into this medium size sample regime.

Figure 2 demonstrates that the sample complexity is directly related to the number of parameters to learn. In particular, it shows that ExpandedLR requires around 15 samples per parameter to achieve the Bayes rate (whatever the dimension). Since in dimension 10, the number of parameters of this model is $2^{d-1}(d+2) = 6144$, we need $15 \times 2^{d-1}(d+2) \approx 100,000$ samples to achieve the Bayes rate and around $10,00$ samples to achieve a reasonable performance. By comparison, the MLP requires as many samples per parameter as ExpandedLR but it doesn't need to have as many parameters as ExpandedLR to perform well. For example in figure 1 (*mixture 1*), *MLP W1* has $2d(d+1)+1 = 111$ parameters, which suffice to obtain good performances.

Self-masked (MNAR) Self-masked (MNAR) does not satisfy Assumption 4.1. Therefore under this data generating scheme, the expression of the Bayes predictor derived in earlier sections is not valid, and ExpandedLR and the MLP with 2^d hidden units need not be Bayes consistent. Self-masking, where the probability of missingness depends on the unobserved value, is however a classical missing-values mechanism, and it is useful to assess the performance of the different methods in this setting. As shown in the right panel of Figure 1, the MLP outperforms all other methods for this data type. This reflects the versatility of this method, which can adapt to all data generating schemes. ExpandedLR caps at a performance close to that of ConstantImputedLR, which highlights the dependence of this method on Assumption 4.1. Imputation, explicit with MICE or implicit with EM, performs poorly. This is expected as in MNAR it does not ground good prediction (Josse et al., 2019b).

8 Discussion and conclusion

We have studied how to minimize a prediction error on data where the response variable is a linear function of a set of partially observed features. Surprisingly, with these missing values, the Bayes-optimal predictor is no longer a linear function of the data. Under Gaussian mixtures assumptions we derive a closed-form expression of this Bayes predictor and used it to introduce a consistent estimation procedure of the prediction function. However, it entails a very high-dimensional estimation. Indeed, our generalization bounds –to our knowledge the first finite-sample theoretical results on prediction with missing values– show that the sample complexity scales, in general, as 2^d. We therefore study several approximations, in the form of constant imputation or a multi-layer perceptron (MLP), which can also be consistent given a sufficient number of hidden units. A key benefit of the MLP is that tuning its number of hidden units enables reducing model complexity and thus decreasing the number of samples required to estimate the model. Our experiments indeed show that in the finite-sample regime, using a MLP with a reduced number of hidden units leads to the best prediction. Importantly, the MLP adapts naturally to the complexity of the data-generating mechanism: it needs fewer hidden units and less data to predict well in a missing completely at random situation.

Our approach departs strongly from classical missing-values approaches, which rely on EM or imputation to model unobserved values. Rather, we tackle the problem with an empirical risk minimization strategy. An important benefit of this approach is that it is robust to the missing-values mechanism, unlike most strategies which require missing-at-random assumptions. Our theoretical and empirical results are useful to guide the choice of learning architectures in the presence of missing-values: with a powerful neural architecture, imputing with zeros and adding features indicating missing-values suffices.

Acknowledgments Funding via ANR-17-CE23-0018 DirtyData and DataIA MissingBigData grants.

References

Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer Cambridge, UK.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38.

Györfi, L., Kohler, M., Krzyzak, A., and Walk, H. (2006). *A distribution-free theory of nonparametric regression*. Springer Science & Business Media.

Jones, M. P. (1996). Indicator and stratification methods for missing explanatory variables in multiple linear regression. *Journal of the American statistical association*, 91(433):222–230.

Josse, J., Mayer, I., Nicholas, T., and Nathalie., V. (2019a). R-miss-tastic: a unified platform for missing values methods and workflows. *arXiv preprint*.

Josse, J., Prost, N., Scornet, E., and Varoquaux, G. (2019b). On the consistency of supervised learning with missing values. *arXiv preprint arXiv:1902.06931*.

Kim, J. K. and Ying, Z. (2018). *Data Missing Not at Random, special issue*. Statistica Sinica. Institute of Statistical Science, Academia Sinica.

Lakshminarayan, K., Harp, S. A., Goldman, R. P., Samad, T., and Others (1996). Imputation of Missing Data Using Machine Learning Techniques. In *KDD*, pages 140–145.

Little, R. J. A. (1992). Regression with missing X's: a review. *Journal of the American Statistical Association*, 87(420):1227–1237.

Little, R. J. A. (1993). Pattern-mixture models for multivariate incomplete data. *Journal of the American Statistical Association*, 88(421):125–134.

Little, R. J. A. and Rubin, D. B. (2002). *Statistical analysis with missing data*. John Wiley & Sons.

Liu, Y., Wang, Y., Feng, Y., and Wall, M. M. (2016). Variable selection and prediction with incomplete high-dimensional data. *Ann. Appl. Stat.*, 10(1):418–450.

Marlin, B. M. and Zemel, R. S. (2009). Collaborative prediction and ranking with non-random missing data. In *Proceedings of the third ACM conference on Recommender systems*, pages 5–12. ACM.

Mohan, K. and Pearl, J. (2019). Graphical Models for Processing Missing Data. Technical Report R-473-L, Department of Computer Science, University of California, Los Angeles, CA.

Pascanu, R., Montufar, G., and Bengio, Y. (2013). On the number of response regions of deep feed forward networks with piece-wise linear activations. *arXiv:1312.6098*.

Pelckmans, K., De Brabanter, J., Suykens, J. A. K., and De Moor, B. (2005). Handling missing values in support vector machine classifiers. *Neural Networks*, 18(5-6):684–692.

Rosenbaum, P. R. and Rubin, D. B. (1984). Reducing bias in observational studies using subclassification on the propensity score. *Journal of the American Statistical Association*, 79(387):516–524.

Rubin, D. B. (1976). Inference and missing data. *Biometrika*, 63(3):581–592.

Tsybakov, A. B. (2003). Optimal rates of aggregation. In *Learning theory and kernel machines*, pages 303–313. Springer.

van Buuren, S. (2018). *Flexible Imputation of Missing Data*. Chapman and Hall/CRC, Boca Raton, FL.

Vapnik, V. (1992). Principles of risk minimization for learning theory. In *Advances in neural information processing systems*, pages 831–838.

Yoon, J., Jordon, J., and Schaar, M. (2018). GAIN: Missing Data Imputation using Generative Adversarial Nets. In *International Conference on Machine Learning*, pages 5675–5684.

Zhang, S., Qin, Z., Ling, C. X., and Sheng, S. (2005). " Missing is useful": missing values in cost-sensitive decision trees. *IEEE transactions on knowledge and data engineering*, 17(12):1689–1693.