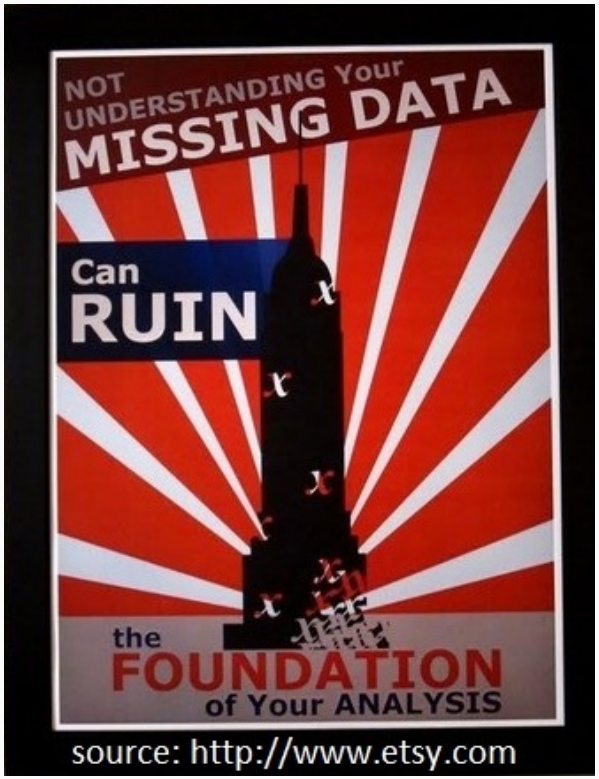
Supervised learning with missing values

Julie Josse
INRIA, Ecole Polytechnique

16 November 2020

Edinburgh - University Statistics Seminar



Introduction

Collaborators on supervised learning with missing values

- M. Le Morvan, Postdoc at INRIA, Paris.
- E. Scornet, Associate Professor at Ecole Polytechnique, IP Paris.

Topic: random forests.

- G. Varoquaux, Senior researcher at INRIA, Paris.

Topic: machine learning. Creator of Scikit-learn in python.



⇒ **Random Forests with missing values**

1. *Consistency of supervised learning with missing values. (2019). Revis JMLR.*

⇒ **Linear regression with missing values - MultiLayer perceptron**

2. *Linear predictor on linearly-generated data with missing values: non consistency and solutions. AISTAT2020.*

3. *Neumiss networks: differential programming for supervised learning with missing values. Neurips2020 (Oral).*

Traumabase project: decision support for trauma patients.

- 20000 trauma patients
- 250 continuous and categorical variables: **heterogeneous**
- 11 hospitals: **multilevel data**
- 4000 new patients/ year

Center	Accident	Age	Sex	Lactactes	BP	Shock	Platelet	...
Beaujon	fall	54	m	NM	180	yes	292000	
Pitie	gun	26	m	NA	131	no	323000	
Beaujon	moto	63	m	3.9	NR	yes	318000	
Pitie	moto	30	w	Imp	107	no	211000	
HEGP	knife	16	m	2.5	118	no	184000	
⋮								⋱

Traumabase project: decision support for trauma patients.

- 20000 trauma patients
- 250 continuous and categorical variables: **heterogeneous**
- 11 hospitals: **multilevel data**
- 4000 new patients/ year

Center	Accident	Age	Sex	Lactactes	BP	Shock	Platelet	...
Beaujon	fall	54	m	NM	180	yes	292000	
Pitie	gun	26	m	NA	131	no	323000	
Beaujon	moto	63	m	3.9	NR	yes	318000	
Pitie	moto	30	w	Imp	107	no	211000	
HEGP	knife	16	m	2.5	118	no	184000	
⋮								⋱

$\Rightarrow$ **Estimate causal effect**: Administration of the **treatment** "tranexamic acid" (within 3 hours after the accident) on the **outcome** mortality for traumatic brain injury patients. [1]

[1] Doubly robust treatment effect estimation with incomplete confounders. Mayer, Wager, J. Annals Of Applied Statistics 2020.

Traumabase project: decision support for trauma patients.

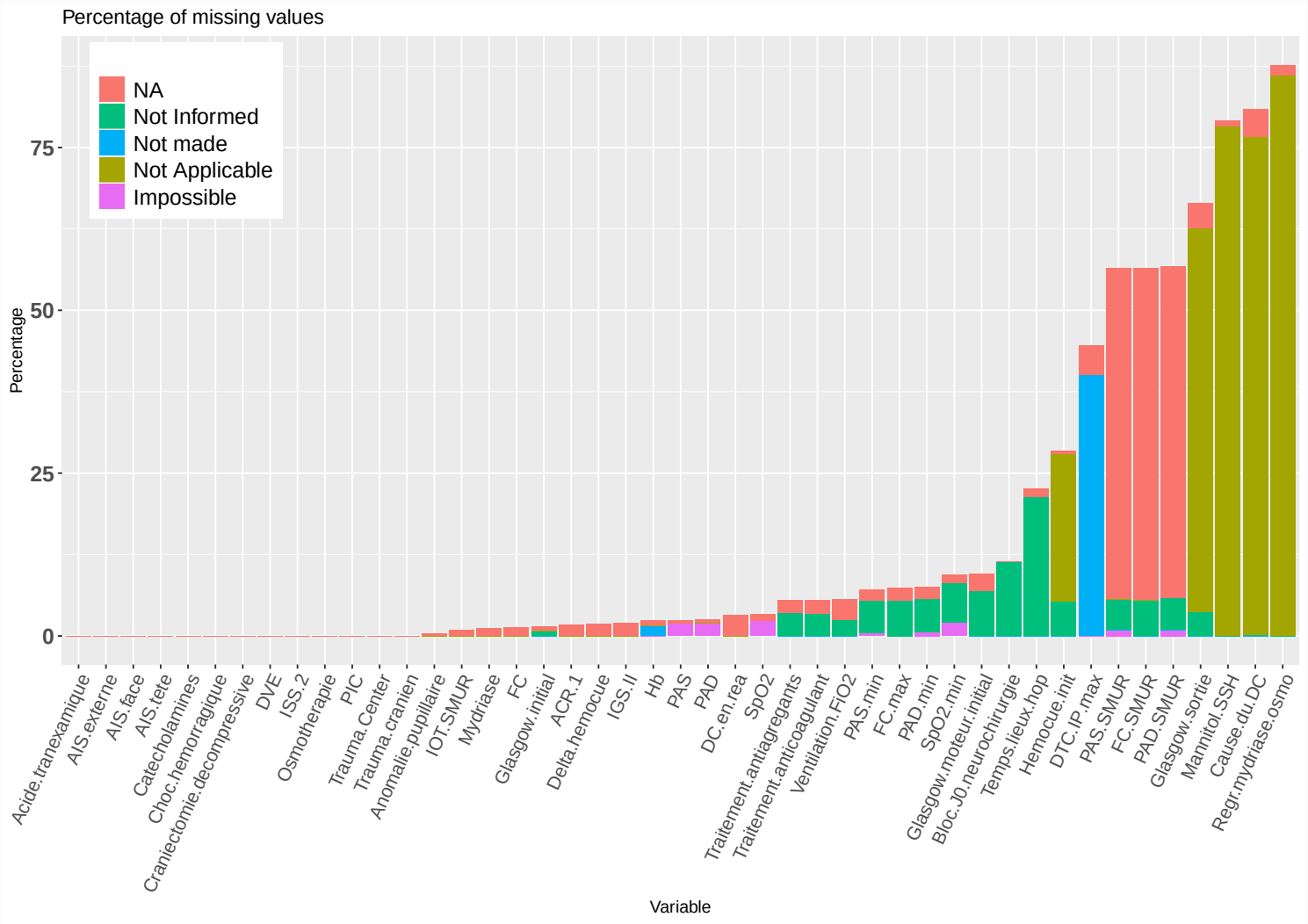
- 20000 trauma patients
- 250 continuous and categorical variables: **heterogeneous**
- 11 hospitals: **multilevel data**
- 4000 new patients/ year

Center	Accident	Age	Sex	Lactactes	BP	Shock	Platelet	...
Beaujon	fall	54	m	NM	180	yes	292000	
Pitie	gun	26	m	NA	131	no	323000	
Beaujon	moto	63	m	3.9	NR	yes	318000	
Pitie	moto	30	w	Imp	107	no	211000	
HEGP	knife	16	m	2.5	118	no	184000	
⋮								⋱

$\Rightarrow$ **Predict** platelet levels given pre-hospital features

Ex linear regression/ random forests with covariates with missing values

Missing values

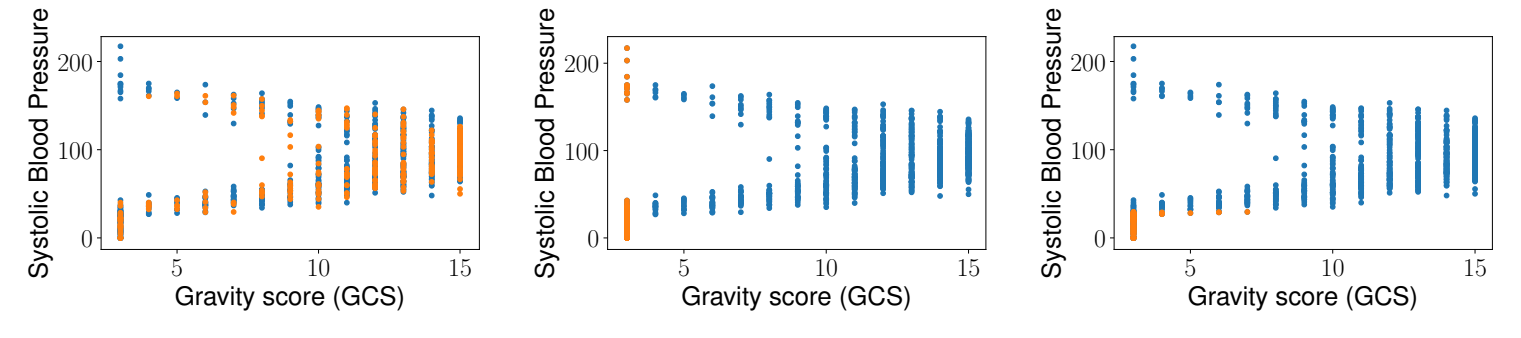


Different pattern: sporadic & systematic (missing variable in one hospital)

Different types: MCAR, MAR, MNAR

Missing values mechanism

Rubin's taxonomy Rubin, 1976

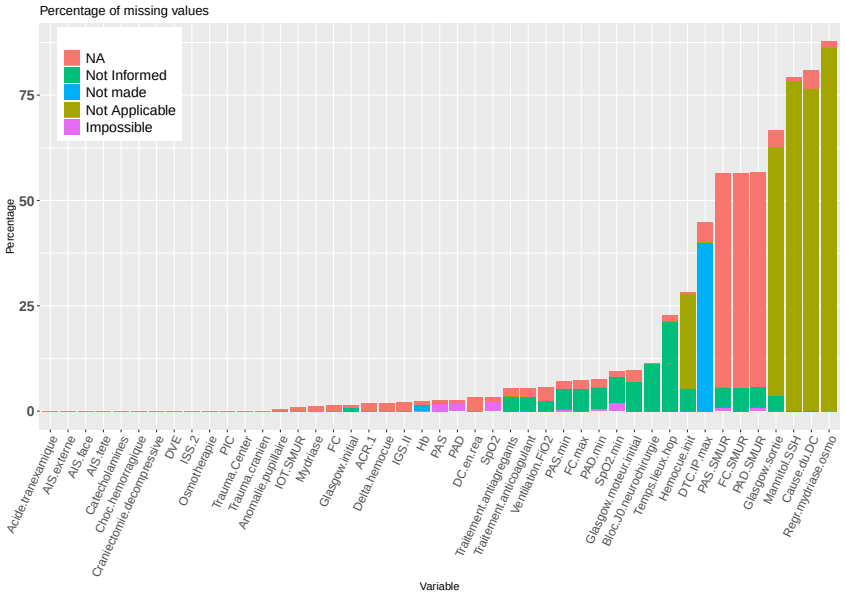


MCAR - MAR - MNAR

Orange: missing values for Systolic Blood Pressure - Gravity index (GCS) is always observed

MCAR (completely at random): Proba to be missing does not depend on SBP neither on gravity
MAR: Proba depends on gravity (we do not measure for too severe patients)
MNAR (not at random): Proba depends on SBP (low SBP not measured)

Complete-case analysis



`?lm, ?glm, na.action = na.omit`

"One of the ironies of Big Data is that missing data play an ever more significant role" (R. Sameworth, 2019)

An $n \times p$ matrix, each entry is missing with probability 0.01

$p = 5 \quad \Longrightarrow \approx 95\%$ of rows kept

$p = 300 \Longrightarrow \approx \ \ 5\%$ of rows kept

Random Forests with missing values

Missing values in a predictive framework (not inferential)

- Aim: target an outcome Y (not estimate parameters and their variance)

- Specificities: train & test sets with missing values. If not: distributional shift. Two data generating process (variables+missing mechanism)

[1] Rmistatic platform to organize ressources - Task view: more than 150 packages

Missing values in a predictive framework (not inferential)

- Aim: target an outcome Y (not estimate parameters and their variance)

- Specificities: train & test sets with missing values. If not: distributional shift. Two data generating process (variables+missing mechanism)

- Methods [1]: (in practice) imputation prior to prediction

  - Separate: impute train and test separately (with a different model)
  - Grouped/ semi-supervised: impute train and test simultaneously but the predictive model is learned only on the training imputed data set.
  - **Imputation train and test sets with the same model**
    Issue: methods (`missForest`) are "black-boxes" *i.e.* take as an input the incomplete data and output the completed data

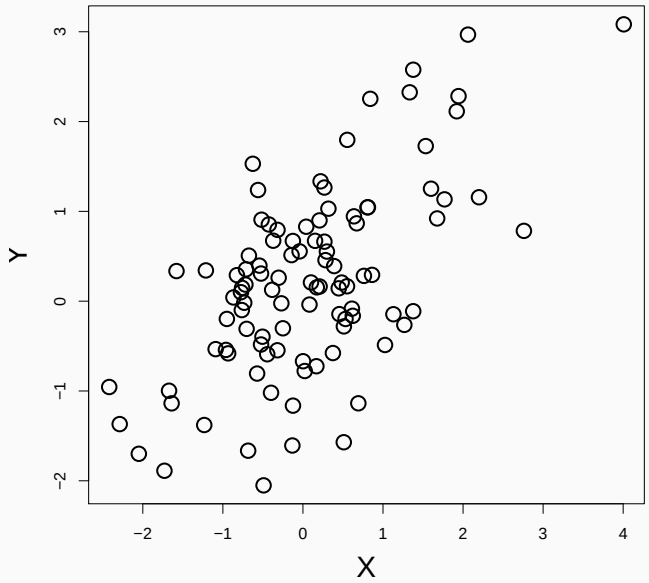
    Easy for univariate imputation: **mean of each colum of the train**.

[1] Rmistatic platform to organize ressources - Task view: more than 150 packages

Mean imputation

- $(x_i, y_i) \underset{\text{i.i.d.}}{\sim} \mathcal{N}_2((\mu_x, \mu_y), \Sigma_{xy})$

X	Y
-0.56	-1.93
-0.86	-1.50
.....	...
2.16	0.7
0.16	0.74



$\mu_y = 0$

$\sigma_y = 1$

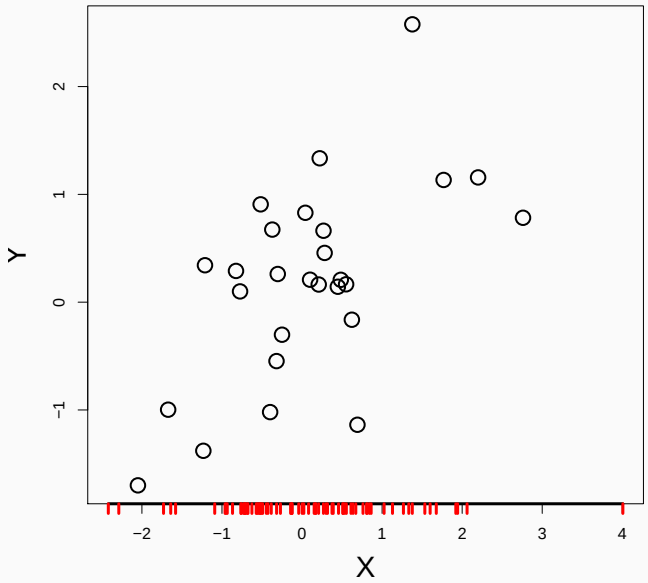
$\rho_{xy} = 0.6$

$\hat{\mu}_y = -0.01$
$\hat{\sigma}_y = 1.01$
$\hat{\rho} = 0.66$

Mean imputation

- $(x_i, y_i) \underset{\text{i.i.d.}}{\sim} \mathcal{N}_2((\mu_x, \mu_y), \Sigma_{xy})$
- 70 % of missing entries completely at random on Y

X	Y
-0.56	NA
-0.86	NA
.....	...
2.16	0.7
0.16	NA



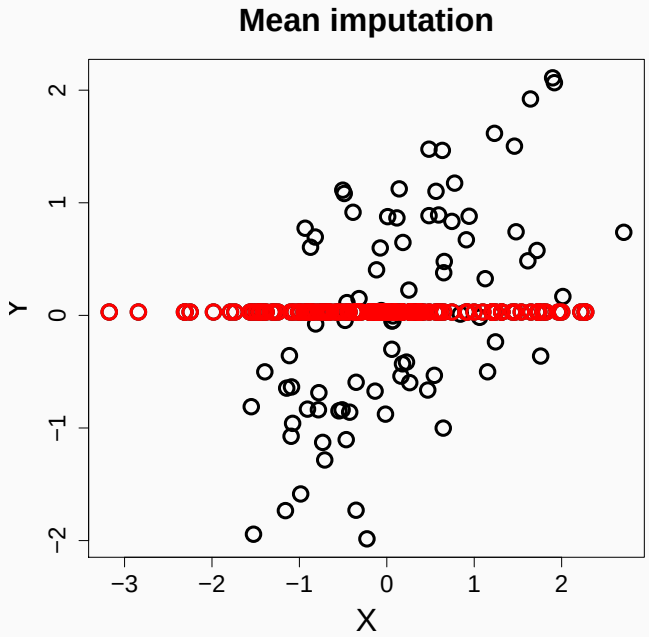
$\mu_y = 0$
$\sigma_y = 1$
$\rho_{xy} = 0.6$

$\hat{\mu}_y = 0.18$
$\hat{\sigma}_y = 0.9$
$\hat{\rho}_{xy} = 0.6$

Mean imputation

- $(x_i, y_i) \underset{\text{i.i.d.}}{\sim} \mathcal{N}_2((\mu_x, \mu_y), \Sigma_{xy})$
- 70 % of missing entries completely at random on Y
- Estimate parameters on the mean imputed data

X	Y
-0.56	**0.01**
-0.86	**0.01**
.....	...
2.16	0.7
0.16	**0.01**

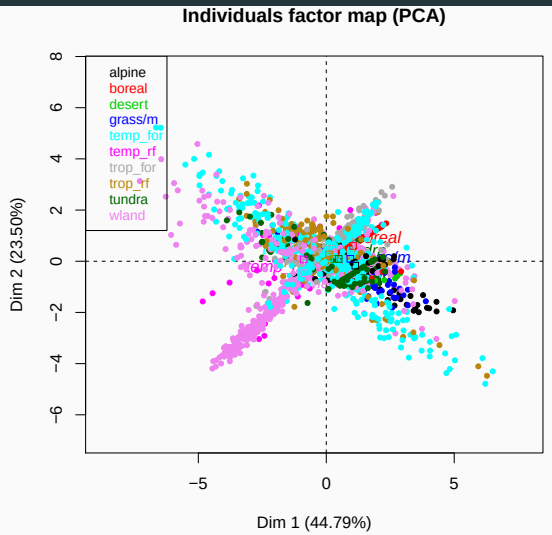


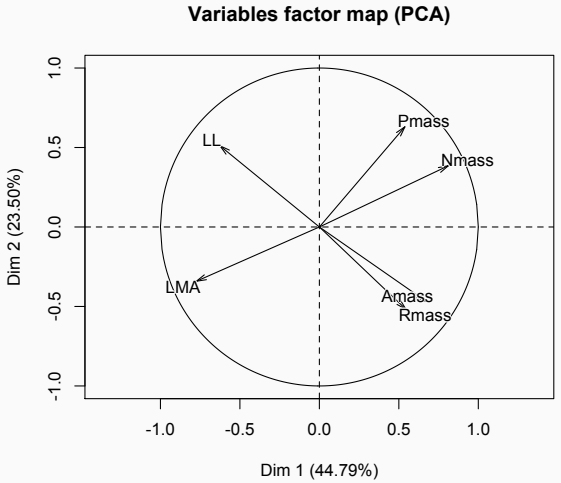
$\mu_y = 0$
$\sigma_y = 1$
$\rho_{xy} = 0.6$

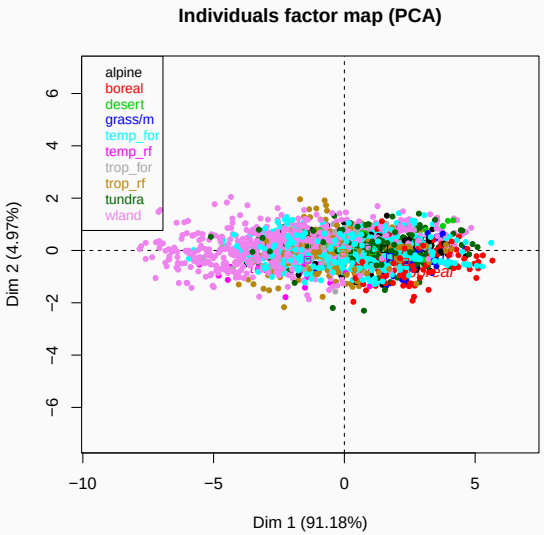
$\hat{\mu}_y = 0.01$
$\hat{\sigma}_y = 0.5$
$\hat{\rho} = 0.30$

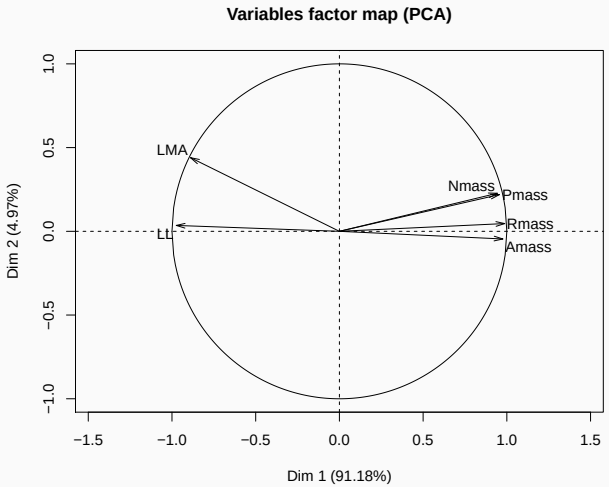
Mean imputation deforms joint and marginal distributions

Mean imputation is bad for estimation









PCA with mean imputation

```
library(FactoMineR)
PCA(ecolo)
Warning message: Missing
are imputed by the mean
of the variable:
You should use imputePCA
from missMDA
```

EM-PCA

```
library(missMDA)
imp <- imputePCA(ecolo)
PCA(imp$comp)
```

J. (2016). missMDA: Handling Missing Values in Multivariate Data Analysis, JSS.

Ecological data: [2] $n = 69000$ species - 6 traits. Estimated correlation between Pmass & Rmass ≈ 0 (mean imputation) or ≈ 1 (EM PCA)

[2] Wright, I. et al. (2004). The worldwide leaf economics spectrum. *Nature*.

Constant (mean) imputation is consistent for prediction

$\tilde{X} = X \odot (1 - M) + \texttt{NA} \odot M$. New feature space is $\widetilde{\mathbb{R}}^d = (\mathbb{R} \cup \{\texttt{NA}\})^d$.

$$Y = \begin{pmatrix} 4.6 \\ 7.9 \\ 8.3 \\ 4.6 \end{pmatrix} \quad \tilde{X} = \begin{pmatrix} 9.1 & \texttt{NA} & 1 \\ 2.1 & \texttt{NA} & 3 \\ \texttt{NA} & 9.6 & 2 \\ \texttt{NA} & 5.5 & 6 \end{pmatrix} \quad X = \begin{pmatrix} 9.1 & 8.5 & 1 \\ 2.1 & 3.5 & 3 \\ 6.7 & 9.6 & 2 \\ 4.2 & 5.5 & 6 \end{pmatrix} \quad M = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

Find a prediction function that minimizes the expected risk.

$$\text{Bayes rule: } f^* \in \underset{f:\ \widetilde{\mathbb{R}}^d \to \mathbb{R}}{\arg\min}\ \mathbb{E}\left[\left(Y - f(\tilde{X})\right)^2\right].$$

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

$\Rightarrow$ One model per pattern (2^d) (Rubin, 1984, generalized propensity score)

Constant (mean) imputation is consistent for prediction

$\tilde{X} = X \odot (1 - M) + \texttt{NA} \odot M$. New feature space is $\widetilde{\mathbb{R}}^d = (\mathbb{R} \cup \{\texttt{NA}\})^d$.

$$Y = \begin{pmatrix} 4.6 \\ 7.9 \\ 8.3 \\ 4.6 \end{pmatrix} \quad \tilde{X} = \begin{pmatrix} 9.1 & \texttt{NA} & 1 \\ 2.1 & \texttt{NA} & 3 \\ \texttt{NA} & 9.6 & 2 \\ \texttt{NA} & 5.5 & 6 \end{pmatrix} \quad X = \begin{pmatrix} 9.1 & 8.5 & 1 \\ 2.1 & 3.5 & 3 \\ 6.7 & 9.6 & 2 \\ 4.2 & 5.5 & 6 \end{pmatrix} \quad M = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

Find a prediction function that minimizes the expected risk.

$$\text{Bayes rule: } f^* \in \underset{f:\ \widetilde{\mathbb{R}}^d \to \mathbb{R}}{\arg\min}\ \mathbb{E}\left[\left(Y - f(\tilde{X})\right)^2\right].$$

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

$\Rightarrow$ One model per pattern (2^d) (Rubin, 1984, generalized propensity score)

Constant (mean) imputation is consistent

Framework - assumptions

- $Y = f(X) + \varepsilon$
- $X = (X_1, \ldots, X_d)$ has a continuous density $g > 0$ on $[0,1]^d$
- $\|f\|_\infty < \infty$
- Missing data MAR on X_1 with $M_1 \perp\!\!\!\perp X_1 | X_2, \ldots, X_d$.
- $(x_2, \ldots, x_d) \mapsto \mathbb{P}[M_1 = 1 | X_2 = x_2, \ldots, X_d = x_d]$ is continuous
- ε is a centered noise independent of (X, M_1)

(remains valid when missing values occur for several variables $X_1, \ldots, X_j$)

Constant (mean) imputation is consistent

Constant imputed entry $x' = (x'_1, x_2, \ldots, x_d)$: $x'_1 = x_1 \mathbb{1}_{M_1=0} + \alpha \mathbb{1}_{M_1=1}$

Theorem. (J. et al. 2019)

$$
\begin{aligned}
f^{\star}_{impute}(x') =& \mathbb{E}[Y|X_2 = x_2, \ldots, X_d = x_d, M_1 = 1] \\
& \mathbb{1}_{x'_1=\alpha} \mathbb{1}_{\mathbb{P}[M_1=1|X_2=x_2,\ldots,X_d=x_d]>0} \\
& + \mathbb{E}[Y|X = x'] \mathbb{1}_{x'_1=\alpha} \mathbb{1}_{\mathbb{P}[M_1=1|X_2=x_2,\ldots,X_d=x_d]=0} \\
& + \mathbb{E}[Y|X_1 = x_1, X_2 = x_2, \ldots, X_d = x_d, M_1 = 0] \mathbb{1}_{x'_1 \neq \alpha}.
\end{aligned}
$$

Prediction with mean is equal to the Bayes function almost everywhere

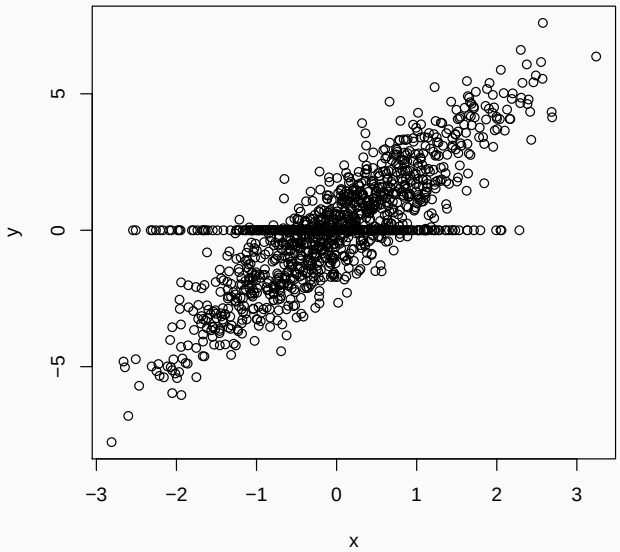
$$f^{\star}_{impute}(X') = f^{\star}(\tilde{X}) = \mathbb{E}[Y|\tilde{X} = \tilde{x}]$$

Rq: pointwise equality if using a constant out of range.

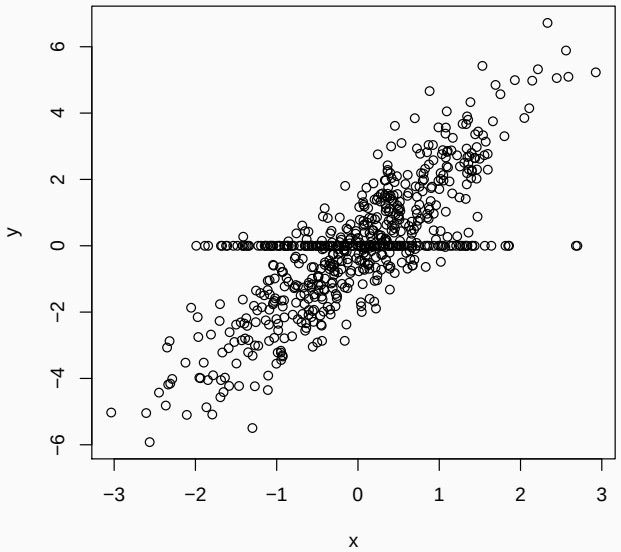
$\Rightarrow$ Learn on the mean-imputed training data, impute the test set with the same means and predict is optimal if the missing data are MAR and the **learning algorithm is universally consistent**

Consistency of supervised learning with NA: Rationale

- Specific value, systematic like a code for missing
- The learner detects the code and recognizes it at the test time
- With categorical data, just code "Missing"
- With continuous data, any constant:
- Need a lot of data (asymptotic result) and a super powerful learner



Train



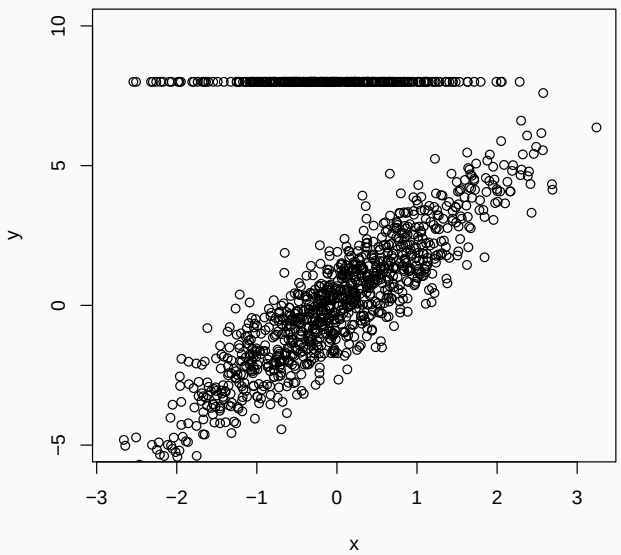
Test

Mean imputation not bad for prediction; it is consistent; despite its drawbacks for estimation - Useful in practice!

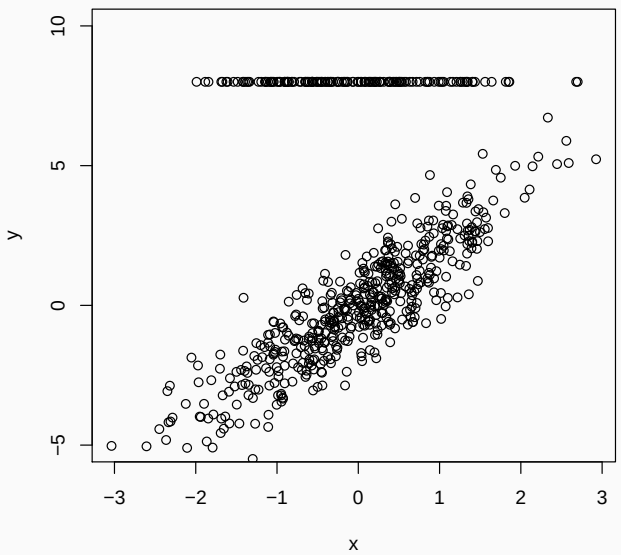
Empirically good results for MNAR

Consistency of supervised learning with NA: Rationale

- Specific value, systematic like a code for missing
- The learner detects the code and recognizes it at the test time
- With categorical data, just code "Missing"
- With continuous data, any constant: out of range
- Need a lot of data (asymptotic result) and a super powerful learner



Train



Test

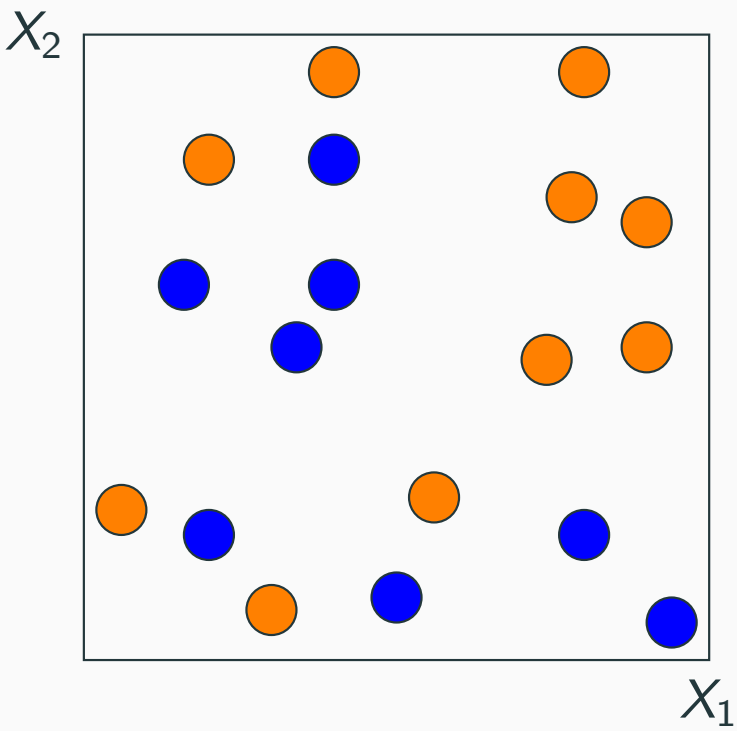
Mean imputation not bad for prediction; it is consistent; despite its drawbacks for estimation - Useful in practice!

Empirically good results for MNAR

CART (Breiman, 1984)

Built recursively by splitting the current cell into two children: Find the feature $j^\star$, the threshold $z^\star$ which minimize the (quadratic) loss

$$(j^\star, z^\star) \in \underset{(j,z)\in\mathcal{S}}{\arg\min}\ \mathbb{E}\Big[\big(Y - \mathbb{E}[Y|X_j \leq z]\big)^2 \cdot \mathbb{1}_{X_j \leq z} + \big(Y - \mathbb{E}[Y|X_j > z]\big)^2 \cdot \mathbb{1}_{X_j > z}\Big].$$

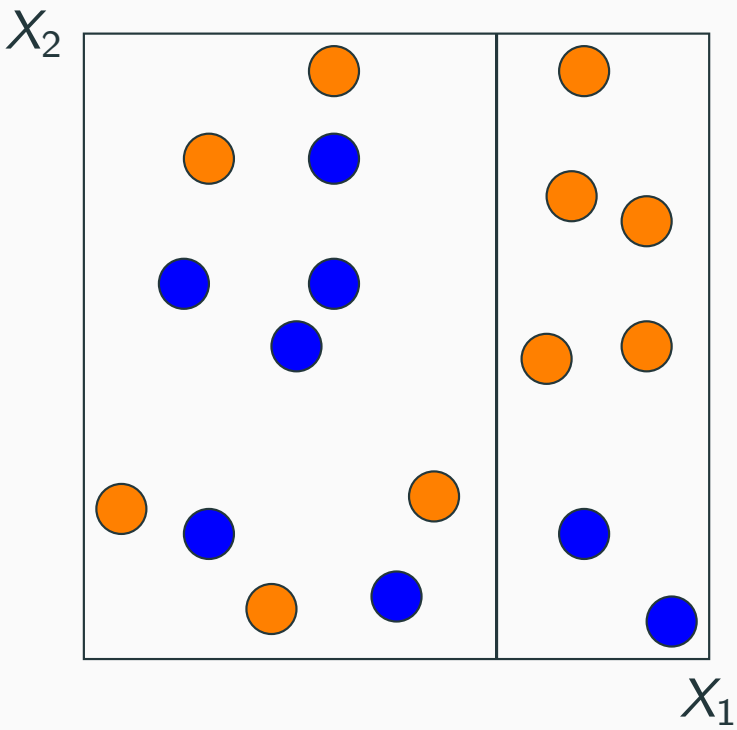


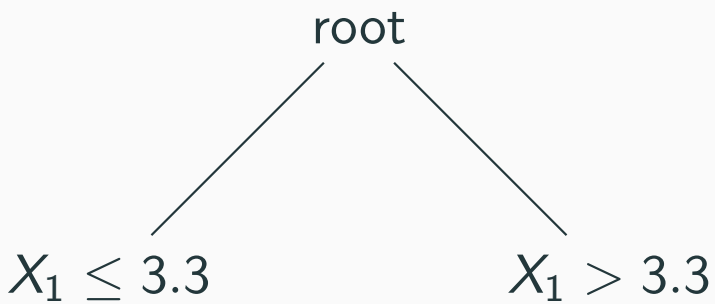
root

CART (Breiman, 1984)

Built recursively by splitting the current cell into two children: Find the feature $j^\star$, the threshold $z^\star$ which minimize the (quadratic) loss

$$(j^\star, z^\star) \in \underset{(j,z)\in\mathcal{S}}{\arg\min}\ \mathbb{E}\Big[\big(Y - \mathbb{E}[Y|X_j \leq z]\big)^2 \cdot \mathbb{1}_{X_j \leq z} + \big(Y - \mathbb{E}[Y|X_j > z]\big)^2 \cdot \mathbb{1}_{X_j > z}\Big].$$

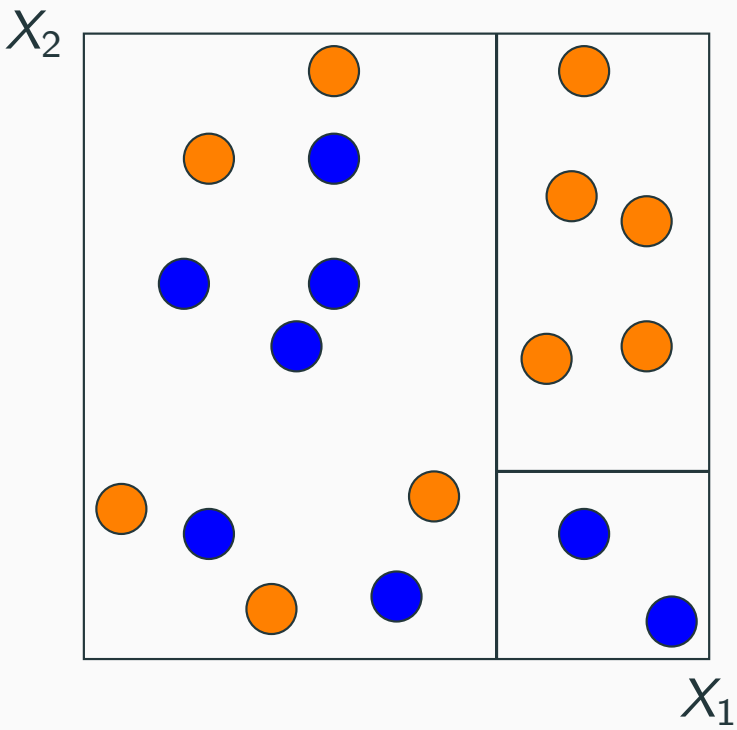


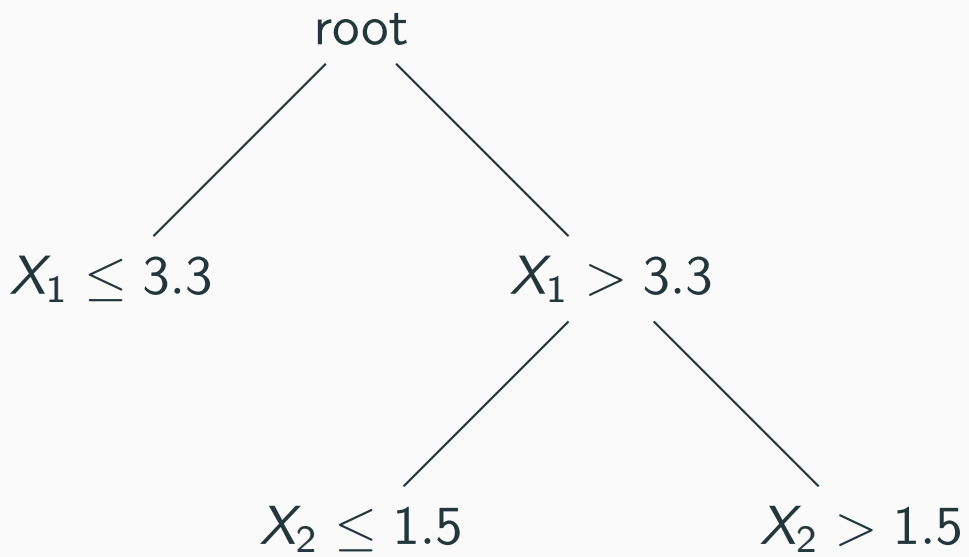


CART (Breiman, 1984)

Built recursively by splitting the current cell into two children: Find the feature $j^\star$, the threshold $z^\star$ which minimize the (quadratic) loss

$$(j^\star, z^\star) \in \underset{(j,z)\in\mathcal{S}}{\arg\min}\ \mathbb{E}\Big[\big(Y - \mathbb{E}[Y|X_j \leq z]\big)^2 \cdot \mathbb{1}_{X_j \leq z} + \big(Y - \mathbb{E}[Y|X_j > z]\big)^2 \cdot \mathbb{1}_{X_j > z}\Big].$$





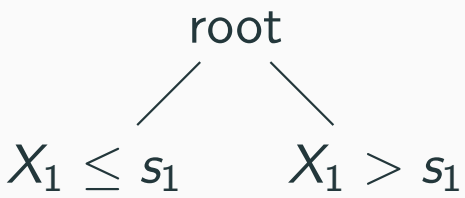
CART with missing values

root

	X_1	X_2	Y
1			
2	NA		
3	NA		
4			

CART with missing values

	X_1	X_2	Y
1			
~~2~~	~~NA~~		
~~3~~	~~NA~~		
4			

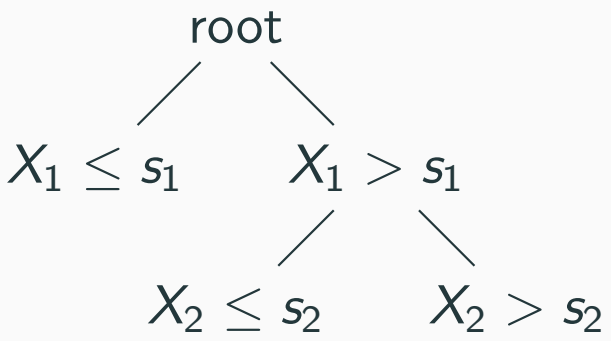


1) Select variable and threshold on observed values (1 & 4 for X_1)

$$\mathbb{E}\Big[\big(Y - \mathbb{E}[Y|X_j \leq z, M_j = 0]\big)^2 \cdot \mathbb{1}_{X_j \leq z, M_j = 0} + \big(Y - \mathbb{E}[Y|X_j > z, M_j = 0]\big)^2 \cdot \mathbb{1}_{X_j > z, M_j = 0}\Big].$$

CART with missing values

	X_1	X_2	Y
1			
2	~~NA~~		
3	~~NA~~		
4			



1) Select variable and threshold on observed values (1 & 4 for X_1)

$$\mathbb{E}\Big[\left(Y - \mathbb{E}[Y|X_j \leq z, M_j = 0]\right)^2 \cdot \mathbb{1}_{X_j \leq z, M_j = 0} + \left(Y - \mathbb{E}[Y|X_j > z, M_j = 0]\right)^2 \cdot \mathbb{1}_{X_j > z, M_j = 0}\Big].$$

2) Propagate observations (2 & 3) with missing values?

- Probabilistic split: $\mathcal{B}ernoulli(\frac{\#L}{\#L+\#R})$ (`Rweeka`)
- Block: Send all to a side by minimizing the error (`xgboost`, `lightgbm`)
- Surrogate split: Search another variable that gives a close partition (rpart)

Missing incorporated in attribute (Twala et al. 2008)

One step: select the variable, the threshold and propagate missing values

1. $\{\widetilde{X}_j \leq z \text{ or } \widetilde{X}_j = \texttt{NA}\}$ vs $\{\widetilde{X}_j > z\}$
2. $\{\widetilde{X}_j \leq z\}$ vs $\{\widetilde{X}_j > z \text{ or } \widetilde{X}_j = \texttt{NA}\}$
3. $\{\widetilde{X}_j \neq \texttt{NA}\}$ vs $\{\widetilde{X}_j = \texttt{NA}\}$.

- The splitting location z depends on the missing values
- **Missing values treated like a category** (well to handle $\mathbb{R} \cup \texttt{NA}$)
- Good for informative pattern (M explains Y)

Targets one model per pattern:

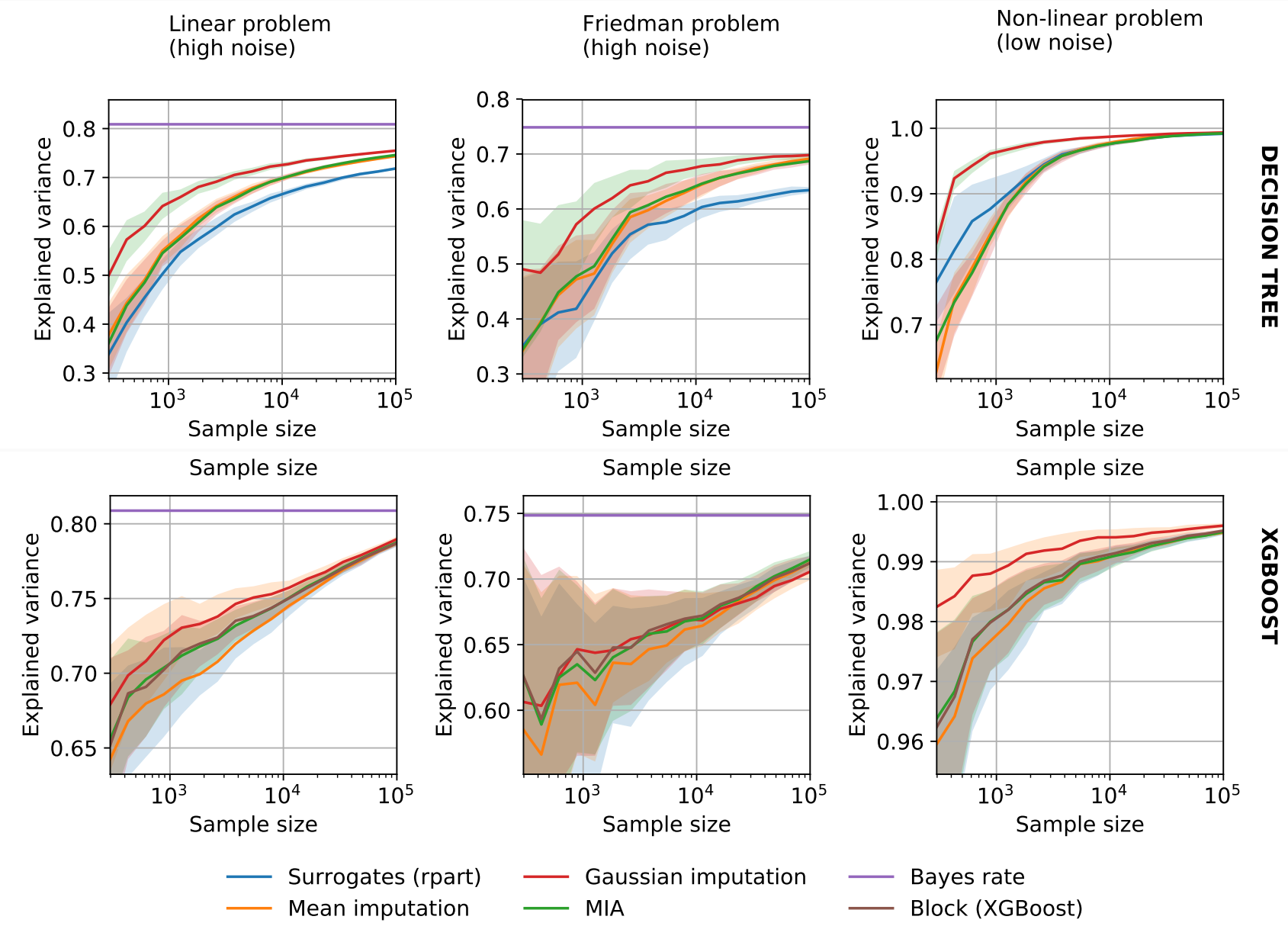
$$\mathbb{E}\left[Y \middle| \tilde{X}\right] = \sum_{m \in \{0,1\}^d} \mathbb{E}\left[Y | X_{obs(m)}, M = m\right] \mathbb{1}_{M=m}$$

- Implementation [3]: `grf package`, `scikit-learn`, `partykit`

$\Rightarrow$ Extremely **good performances** in practice **for any mechanism**.

[3] implementation trick, J. Tibshirani, duplicate the incomplete columns, and replace the missing entries once by $+\infty$ and once by $-\infty$

Consistency: 40% missing values MCAR



Linear regression with missing values (using MLP)

Linear model with missing values

Linear model:

$Y = \beta_0 + \langle X, \beta \rangle + \varepsilon, \quad X \in \mathbb{R}^d, \ \varepsilon$ gaussian.

Existing solutions

- ML with EM algo. (available implementation struggles for large d)
- Multiple imputation (few aggregation strategies for predictive models)

$\Rightarrow$ Mainly to estimate parameters in Missing At Random setting

Aim: Predict Y (out of sample) with any missing value mechanism

$\tilde{X} = X \odot (1 - M) + \texttt{NA} \odot M$. New feature space is $\widetilde{\mathbb{R}}^d = (\mathbb{R} \cup \{\texttt{NA}\})^d$.

$$\text{Bayes rule: } f^* \in \underset{f:\ \widetilde{\mathbb{R}}^d \to \mathbb{R}}{\arg\min}\ \mathbb{E}\left[\left(Y - f(\tilde{X})\right)^2\right].$$

$$f^*(\tilde{X}) = \mathbb{E}\left[Y \mid \tilde{X}\right] = \sum_{m \in \{0,1\}^d} \mathbb{E}\left[Y | X_{obs(m)}, M = m\right] \ \mathbb{1}_{M=m}$$

$\Rightarrow$ One model per pattern (2^d) (Rubin, 1984, generalized propensity score)

Linear model with missing values not necessarely linear

Example

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.

$\Rightarrow$ There exists a large variety of submodels for a same linear model. Depend on the structure of X and on the missing-value mechanism.

Explicit Bayes predictor with missing values

Linear model:

$$Y = \beta_0 + \langle X, \beta \rangle + \varepsilon, \quad X \in \mathbb{R}^d, \ \varepsilon \text{ gaussian.}$$

Bayes predictor for the linear model:

$$\begin{aligned} f^\star(\tilde{X}) &= \mathbb{E}[Y|\tilde{X}] = \mathbb{E}[\beta_0 + \beta^\mathsf{T} X \mid \ M, X_{obs(M)}] \\ &= \beta_0 + \beta_{obs(M)}^\mathsf{T} X_{obs(M)} + \beta_{mis(M)}^\mathsf{T} \ \mathbb{E}[X_{mis(M)} \mid M, X_{obs(M)}] \end{aligned}$$

Assumptions on covariates and missing values

1. Gaussian pattern mixture model, PMM: $X \mid (M = m) \sim \mathcal{N}(\mu_m, \Sigma_m)$
2. Gaussian assumption $X \sim \mathcal{N}(\mu, \Sigma)$ + MCAR and MAR
3. (Also for Gaussian assumption + MNAR self mask gaussian)

Under Assump. the Bayes predictor is linear per pattern

$$f^\star(X_{obs}, M) = \beta_0^\star + \langle \beta_{obs}^\star, X_{obs} \rangle + \langle \beta_{mis}^\star, \mu_{mis} + \Sigma_{mis,obs}(\Sigma_{obs})^{-1}(X_{obs} - \mu_{obs}) \rangle$$

use of *obs* instead of *obs*(M) for lighter notations - Expression for 2.

Expanded Bayes predictor

Under GPMM, bayes predictor is linear per pattern $\Leftrightarrow$ linear model in W

$$f^*(\tilde{X}) = \langle W, \delta \rangle$$

W an expansion (2^d blocks) & parameters $\delta \in \mathbb{R}^d$ function of β, μ_m, Σ_m

$$\tilde{X} = \left(\begin{array}{c|cc} 1 & x_{1,1} & x_{1,2} \\ 1 & x_{2,1} & x_{2,2} \\ \hline 1 & x_{3,1} & \text{NA} \\ 1 & x_{4,1} & \text{NA} \\ \hline 1 & \text{NA} & x_{5,2} \\ 1 & \text{NA} & x_{6,2} \\ \hline 1 & \text{NA} & \text{NA} \\ 1 & \text{NA} & \text{NA} \end{array}\right) \quad W = \left(\begin{array}{ccc|cc|cc|c} 1 & x_{1,1} & x_{1,2} & 0 & 0 & 0 & 0 & 0 \\ 1 & x_{2,1} & x_{2,2} & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & x_{3,1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x_{4,1} & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 1 & x_{5,2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & x_{6,2} & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right)$$

$W = (\mathbb{1}_{M=(0,0)}, X_1\mathbb{1}_{M=(0,0)}, X_2\mathbb{1}_{M=(0,0)}, \mathbb{1}_{M=(0,1)}, X_1\mathbb{1}_{M=(0,1)}, \mathbb{1}_{M=(1,0)}, X_2\mathbb{1}_{M=(1,0)}, \mathbb{1}_{M=(1,1)})$.

Expanded Bayes predictor

Under GPMM, bayes predictor is linear per pattern $\Leftrightarrow$ linear model in W

$$f^*(\tilde{X}) = \langle W, \delta \rangle$$

W an expansion (2^d blocks) & parameters $\delta \in \mathbb{R}^d$ function of β, μ_m, Σ_m

$$\tilde{X} = \left(\begin{array}{c|cc} 1 & x_{1,1} & x_{1,2} \\ 1 & x_{2,1} & x_{2,2} \\ \hline 1 & x_{3,1} & \text{NA} \\ 1 & x_{4,1} & \text{NA} \\ \hline 1 & \text{NA} & x_{5,2} \\ 1 & \text{NA} & x_{6,2} \\ \hline 1 & \text{NA} & \text{NA} \\ 1 & \text{NA} & \text{NA} \end{array}\right) \quad W = \left(\begin{array}{ccc|cc|cc|c} 1 & x_{1,1} & x_{1,2} & 0 & 0 & 0 & 0 & 0 \\ 1 & x_{2,1} & x_{2,2} & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & x_{3,1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x_{4,1} & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 1 & x_{5,2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & x_{6,2} & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right)$$

$W = (\mathbb{1}_{M=(0,0)}, X_1\mathbb{1}_{M=(0,0)}, X_2\mathbb{1}_{M=(0,0)}, \mathbb{1}_{M=(0,1)}, X_1\mathbb{1}_{M=(0,1)}, \mathbb{1}_{M=(1,0)}, X_2\mathbb{1}_{M=(1,0)}, \mathbb{1}_{M=(1,1)})$.

Problem: Dim of W is $p = \sum_{k=0}^{d} \binom{d}{k} \times (k+1) = 2^{d-1} \times (d+2)$.

Need to approximate it: Linear + MLP approximation + Neumiss

Linear Approximation

The Bayes predictor can be expressed as a polynome of X and M, which can be truncated to a lower-dimensional approximation.

$$f^\star_{\text{approx}}(\tilde{X}) = \beta^\star_{0,0} + \sum_{j=1}^{d} \beta^\star_{j,0} M_j + \sum_{j=1}^{d} \beta^\star_j X_j (1 - M_j).$$

$$\left(\begin{array}{c|cc|cc}
1 & X_1 \odot (1 - M_1) & X_2 \odot (1 - M_2) & M_1 & M_2 \\
\hline
1 & x_{1,1} & x_{1,2} & 0 & 0 \\
1 & x_{2,1} & x_{2,2} & 0 & 0 \\
\hline
1 & x_{3,1} & 0 & 0 & 1 \\
1 & x_{4,1} & 0 & 0 & 1 \\
\hline
1 & 0 & x_{5,2} & 1 & 0 \\
1 & 0 & x_{6,2} & 1 & 0 \\
\hline
1 & 0 & 0 & 1 & 1 \\
1 & 0 & 0 & 1 & 1
\end{array}\right)$$

Imputing X by 0 and concatenate M

Linear Approximation

Impute X by 0 and concatenate $M \Leftrightarrow$ optimize an imputation constant.

$$\text{Given } \begin{array}{cc} X_1 & X_2 \\ \begin{pmatrix} 1.1 & 3.2 \\ \texttt{NA} & 0.1 \\ 4.6 & \texttt{NA} \\ 4.0 & 0.9 \\ \texttt{NA} & 2.2 \end{pmatrix} \end{array}, \quad \begin{array}{cccc} X_1 & X_2 & M_1 & M_2 \\ \begin{pmatrix} 1.1 & 3.2 & 0 & 0 \\ 0 & 0.1 & 1 & 0 \\ 4.6 & 0 & 0 & 1 \\ 4.0 & 0.9 & 0 & 0 \\ 0 & 2.2 & 1 & 0 \end{pmatrix} \end{array} \Leftrightarrow \begin{array}{cc} X_1 & X_2 \\ \begin{pmatrix} 1.1 & 3.2 \\ C_1 & 0.1 \\ 4.6 & C_2 \\ 4.0 & 0.9 \\ C_1 & 2.2 \end{pmatrix} \end{array}$$

Indeed,

$$\beta_j \{X_j(1 - M_j) + c_j M_j\} = \beta_j X_j(1 - M_j) + \{\beta_j c_j\} M_j.$$

Expanded model VS Linear approximation

expanded

$$\left(\begin{array}{ccc|cc|cc|c} 1 & x_{1,1} & x_{1,2} & 0 & 0 & 0 & 0 & 0 \\ 1 & x_{2,1} & x_{2,2} & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & x_{3,1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x_{4,1} & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 1 & x_{5,2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & x_{6,2} & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right)$$

vs

linear approximation

$$\left(\begin{array}{c|cc|cc} 1 & x_{1,1} & x_{1,2} & 0 & 0 \\ 1 & x_{2,1} & x_{2,2} & 0 & 0 \\ \hline 1 & x_{3,1} & 0 & 0 & 1 \\ 1 & x_{4,1} & 0 & 0 & 1 \\ \hline 1 & 0 & x_{5,2} & 1 & 0 \\ 1 & 0 & x_{6,2} & 1 & 0 \\ \hline 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \end{array}\right)$$

Two estimations strategies:

• Linear reg. to estimate the expanded bayes predictor: rich model, powerful in low dimension. Costly, large variance in high dimension

• Linear approximation: lower approximation capacity smaller variance since it contains fewer parameters

Finite sample bounds - Excess of risk

• Expanded: $\mathcal{O}\left(\frac{2^d}{n}\right)$

• Linear approximation: $\mathcal{O}\left(d^2 + \frac{d}{n}\right)$

Comparing the upper bounds: Risk of expanded is lower than risk of approximation when $n >> \frac{2^d}{d}$

Bayes consistency of the MLP

Theorem. Bayes consistency of a MLP. Le Morvan et al. (2020)

Under linear model + Gaussian pattern mixture model, a MLP:

- with one hidden layer containing 2^d hidden units
- ReLU activation functions
- fed with $[X \odot (1 - M), M]$ ($\tilde{X}$ imputed by 0 concatenated with mask)

can achieve the Bayes rate.

Rationale: The MLP produces a prediction function piecewise affine. Since the Bayes predictor is linear per pattern, MLP good candidate.

We show that there exists a configuration of the parameters of the MLP so that the resulting predictor is the Bayes predictor.

Number of parameters: $(d + 1)2^{d+1} + 1$.

$\Rightarrow$ Provides a natural way to reduce the model capacity by reducing the number of hidden units. (Trading off estimation and approximation error)

Neumiss Networks to approximate the covariance matrix

The Bayes predictor is linear per pattern (Gaussian+ M(C)AR)

$$f^\star(X_{obs}, M) = \beta_0^\star + \langle \beta^\star_{obs}, X_{obs} \rangle + \langle \beta^\star_{mis}, \mu_{mis} + \Sigma_{mis,obs}(\Sigma_{obs})^{-1}(X_{obs} - \mu_{obs}) \rangle$$

Order-ℓ approx of $(\Sigma^{-1}_{obs(m)})$ for any m defined recursively:

$$S^{(\ell)}_{obs(m)} = (Id - \Sigma_{obs(m)}) S^{(\ell-1)}_{obs(m)} + Id.$$

Neuman Series, $S^{(0)} = Id$, $\ell = \infty$: $(\Sigma_{obs(m)})^{-1} = \sum_{k=0}^{\infty} (Id - \Sigma_{obs(m)})^k$

Neumiss Networks to approximate the covariance matrix

Order-ℓ approx of the Bayes predictor in MAR

$$f_\ell^\star(X_{obs}, M) = \langle \beta_{obs}, X_{obs} \rangle + \langle \beta_{mis}, \mu_{mis} + \Sigma_{mis,obs} S^{(\ell)}_{obs(m)} (X_{obs} - \mu_{obs}) \rangle.$$

Order-ℓ approx of $(\Sigma^{-1}_{obs(m)})$ for any m defined recursively:

$$S^{(\ell)}_{obs(m)} = (Id - \Sigma_{obs(m)}) S^{(\ell-1)}_{obs(m)} + Id.$$

Neuman Series, $S^{(0)} = Id$, $\ell = \infty$: $(\Sigma_{obs(m)})^{-1} = \sum_{k=0}^{\infty} (Id - \Sigma_{obs(m)})^k$

Proposition (Risk of the Order-ℓ approx)

Let ν be the smallest eigenvalue of Σ. Assume linear model with Gaussian covariates, M(C)AR, and that the spectral radius of Σ is < 1. Then, for all $\ell \geq 1$,

$$\mathbb{E}\left[\left(f_\ell^\star(X_{obs}, M) - f^\star(X_{obs}, M) \right)^2 \right] \leq \frac{(1-\nu)^{2\ell} \|\beta^\star\|_2^2}{\nu} \mathbb{E}\left[\left\| Id - S^{(0)}_{obs(M)} \Sigma_{obs(M)} \right\|_2^2 \right]$$

The error of the order-ℓ approximation decays exponentially fast with ℓ.

Neumiss Networks to approximate the covariance matrix

Order-ℓ approx of the Bayes predictor in MAR

$$f_\ell^\star(X_{obs}, M) = \langle \beta_{obs}, X_{obs} \rangle + \langle \beta_{mis}, \mu_{mis} + \Sigma_{mis,obs} S^{(\ell)}_{obs(m)} (X_{obs} - \mu_{obs}) \rangle.$$

Order-ℓ approx of $(\Sigma^{-1}_{obs(m)})$ for any m defined recursively:

$$S^{(\ell)}_{obs(m)} = (Id - \Sigma_{obs(m)}) S^{(\ell-1)}_{obs(m)} + Id.$$

Neuman Series, $S^{(0)} = Id$, $\ell = \infty$: $(\Sigma_{obs(m)})^{-1} = \sum_{k=0}^{\infty} (Id - \Sigma_{obs(m)})^k$

$\Rightarrow$ **Neural network architecture to approximate the Bayes predictor**

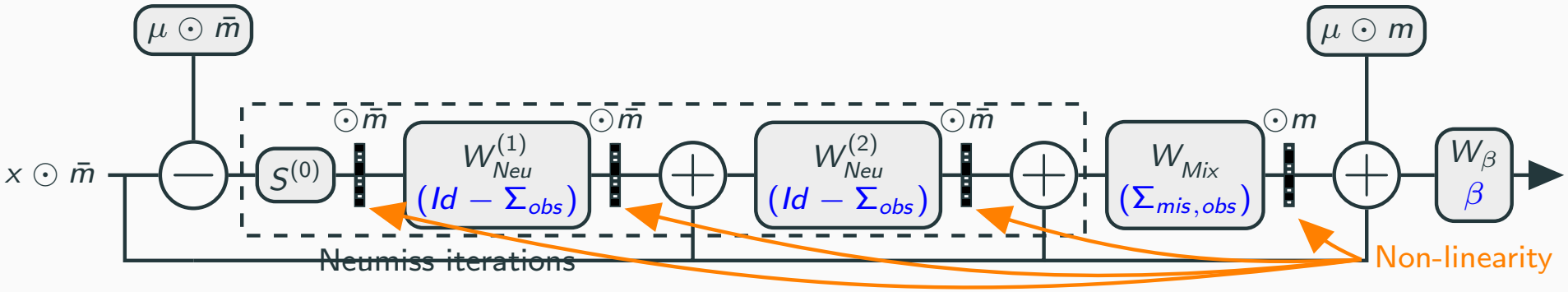


Figure 1: Depth of 3, $\bar{m} = 1 - m$. Each weight matrix $W^{(k)}$ corresponds to a simple transformation of the covariance matrix indicated in blue.

Neumiss Networks to approximate the covariance matrix

Order-ℓ approx of the Bayes predictor in MAR

$$f_\ell^\star(X_{obs}, M) = \langle \beta_{obs}, X_{obs} \rangle + \langle \beta_{mis}, \mu_{mis} + \Sigma_{mis,obs} S^{(\ell)}_{obs(m)} (X_{obs} - \mu_{obs}) \rangle.$$

Order-ℓ approx of $(\Sigma^{-1}_{obs(m)})$ for any m defined recursively:

$$S^{(\ell)}_{obs(m)} = (Id - \Sigma_{obs(m)}) S^{(\ell-1)}_{obs(m)} + Id.$$

Neuman Series, $S^{(0)} = Id$, $\ell = \infty$: $(\Sigma_{obs(m)})^{-1} = \sum_{k=0}^{\infty} (Id - \Sigma_{obs(m)})^k$

$\Rightarrow$ **Neural network architecture to approximate the Bayes predictor**

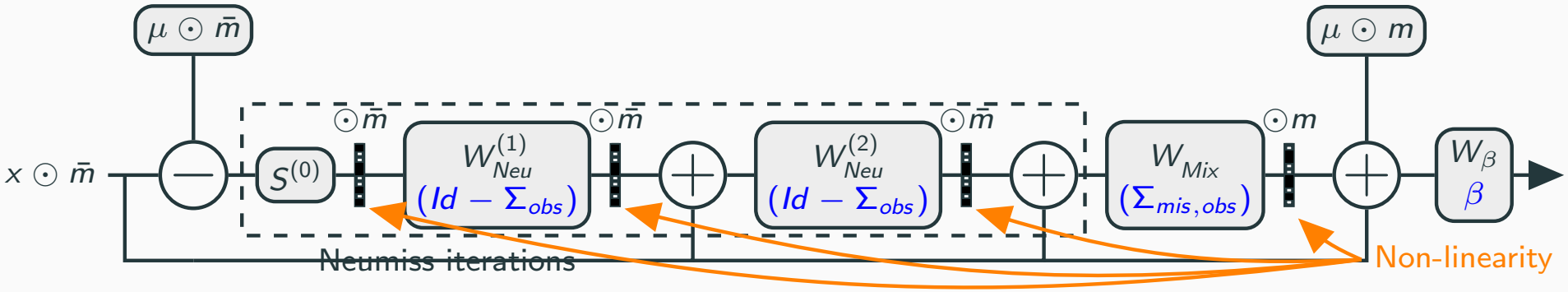
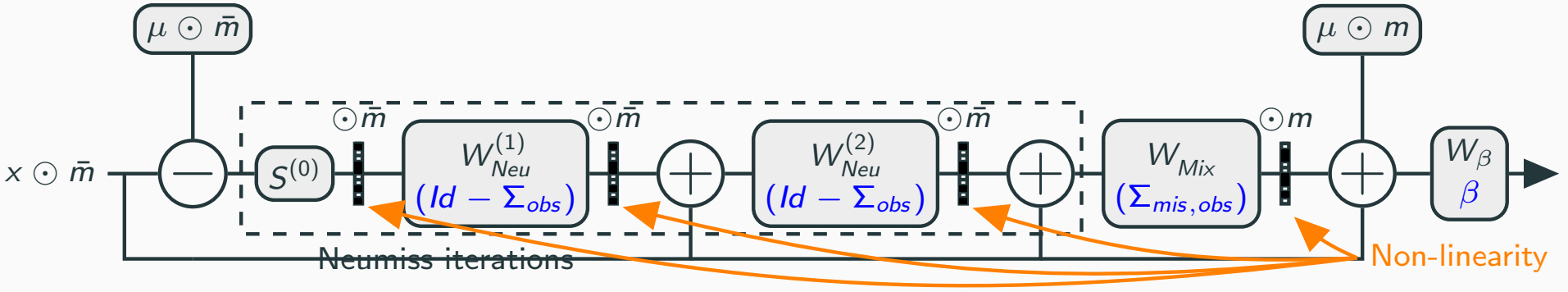


Figure 1: Depth of 3, $\bar{m} = 1 - m$. Each weight matrix $W^{(k)}$ corresponds to a simple transformation of the covariance matrix indicated in blue.

Networks with missing values: $\odot M$ nonlinearity

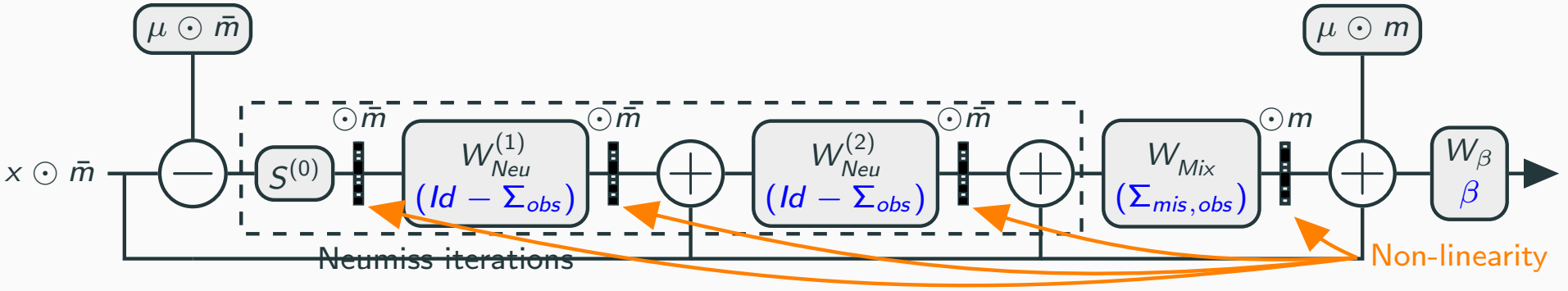


- Implementing a network with the matrix **weights** $W^{(k)} = (I - \Sigma_{obs(m)})$ **masked differently for each sample** can be challenging
- Masked weights is **equivalent to masking input & output vector.**

Let v a vector, $\bar{m} = 1 - m$. $(W \odot \bar{m}\bar{m}^\top)v = (W(v \odot \bar{m})) \odot \bar{m}$

Classic network with multiplications by the mask nonlinearities $\odot M$

Networks with missing values: $\odot M$ nonlinearity



- Implementing a network with the matrix **weights** $W^{(k)} = (I - \Sigma_{obs(m)})$ **masked differently for each sample** can be challenging
- Masked weights is **equivalent to masking input & output vector.**

Let v a vector, $\bar{m} = 1 - m$. $(W \odot \bar{m}\bar{m}^\top)v = (W(v \odot \bar{m})) \odot \bar{m}$

Classic network with multiplications by the mask nonlinearities $\odot M$

Proposition (equivalence MLP - depth-0 Neumiss network)

A MLP with ReLU activations, one hidden layer of d hidden units, and which operates on the $[X \odot (1 - M), M]$, the input X imputed by 0 concatenated with the mask M, is equivalent to the 0-depth NN

Experiments for linear regression with missing values

- $Y = X\beta^* + \varepsilon$, ε chosen such as SNR $= 10$.
- $X \sim \mathcal{N}(\mu, \Sigma)$
- $\Sigma = UU^\top + \text{diag}(\epsilon')$, $U \in \mathbb{R}^{d \times \frac{d}{2}}$, $U_ij \sim \mathcal{N}(0,1)$ $\epsilon' \sim \mathcal{U}(10^{-2}, 10^{-1})$
- 50% of MCAR, MAR, Probit self-masking.

- **Max Likelihood**: to estimate the parameters of the joint Gaussian distribution $(X_1, ..., X_d, Y)$ with EM. Predict by conditional expectation of Y given X_{obs}.

- **ICE + LR**: conditional imputation with an iterative imputer followed by linear regression.

- **MLP**: take as input the data imputed by 0 concatenated with the mask $[X \odot (1 - M), M]$ with ReLU nonlinearity,
  - **MLP-Wide**: one hidden layer with width increased (between d & 2^d)
  - **MLP-Deep**: 1 to 10 hidden layers of d hidden units

- **Neumiss**: The Neumiss architecture with the $\odot M$, choosing the depth on a validation set.

Results

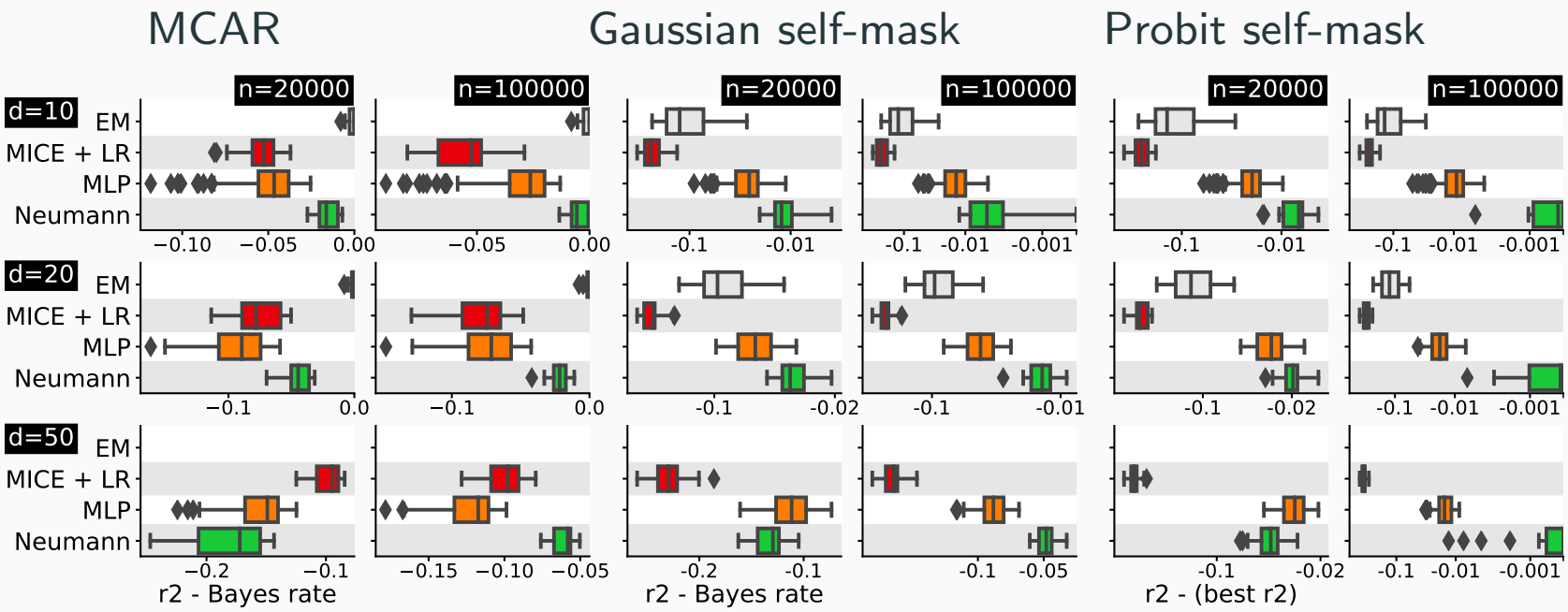


Figure 2: Predictive performances in various scenarios — varying missing-value mechanisms, number of samples n, and number of features d.

$\Rightarrow$ Best performances for MNAR scenario (50% of NA on all variables)

• More effective to increase the capacity of the Neumiss network (depth) than to increase the capacity (width) of MLP Wide.

Discussion - challenges

Take-home message. Supervised learning with missing values.

Supervised learning different from usual inferential probabilistic models. Solutions useful in practice robust to the missing-value mechanisms but needs powerful model.

Powerful learner with missing values

- Incomplete train and test $\rightarrow$ same imputation model
- Single constant imputation is consistent with a powerful learner
- Tree-based models : Missing Incorporated in Attribute
- To be done: nonasymptotic results, uncertainty, distributional shift: No NA in the test? Proofs in MNAR

Linear regression with missing values

- The Bayes predictor is explicit under Gaussian assumptions/ MAR and gaussian self mask but high-dimensional.
- Approx include MLP which can be consistent and Neumiss Network
- New architecture for network with missing data: $\odot M$ nonlinearity.

Ressources

R-miss-tastic https://rmisstastic.netlify.com/R-miss-tastic

J., I. Mayer, N. Tierney & N. Vialaneix

Project funded by the R consortium (Infrastructure Steering Committee)[4]

Aim: a reference platform on the theme of missing data management

- list existing packages
- available literature
- tutorials
- analysis workflows on data
- main actors

$\Rightarrow$ Federate the community

$\Rightarrow$ Contribute!

[4] https://www.r-consortium.org/projects/call-for-proposals

Ressources

Examples:

- Lecture [5] - General tutorial : Statistical Methods for Analysis with Missing Data (Mauricio Sadinle)
- Lecture - Multiple Imputation: `mice` by Nicole Erler [6]
- Longitudinal data, Time Series Imputation (Steffen Moritz - very active contributor of r-miss-tastic), Principal Component Methods[7]

[5] https://rmisstastic.netlify.com/lectures/

[6] `https://rmisstastic.netlify.com/tutorials/erler_course_multipleimputation_2018/erler_practical_mice_2018`

[7] `https://rmisstastic.netlify.com/tutorials/Josse_slides_imputation_PCA_2018.pdf`