Supervised Learning Under Informative Missingness

Mike Van Ness

May 2, 2022

Informative Missingness

Informative Missingness: when the fact that data is missing is informative to a predictive model.

Examples:

- Medical Data
  - Model: predict whether or not a patient has a disease
  - Missingness: doctor only takes a certain lab measurement if the patient is feared to have a certain disease.
- Political Survey Data.
  - Model: predict what candidate a partipant will vote for.
  - Missingness: participates more likely to omit a question depending on their party affiliation.

Missing Mechanisms

Similar concept: missingness mechanisms. For data X and missing indicators R, we have

- **MCAR:** $P(R = r \mid X = x) = P(R = r)$
- **MAR:** $P(R = r \mid X = x) = P(R = r \mid X_{\text{obs}_r})$.
- **MNAR:** $P(R = r \mid X = x) = P(R = r \mid X = x)$

Difference: informative missingness highlights impact on supervised response/label.

Supervised Learning Approaches

Common ways to deal with missing values in supervised learning models

- Impute missing values, then pretend data is complete.
- Use multiple imputation, fit several models for each set of imputations, average results.
- Use model that can handle missing values natively.
- Missing indicator method.

Linear Regression

Simple linear model:

$$Y = X\alpha + \epsilon, \quad X, Y \in \mathbb{R}$$

With full training data $X^{(1)}, \ldots, X^{(n)}$, fit ordinary least squares model. However, X is sometimes missing, leading to random variable R such that

$$R = \begin{cases} 1 & X \text{ is missing} \\ 0 & X \text{ is observed} \end{cases}$$

Additionally, define random variable Z such that

$$Z = \begin{cases} X & R = 0 \\ 0 & R = 1 \end{cases}$$

Linear Regression

Let $D = \begin{bmatrix} Z \\ R \end{bmatrix}$, then $DD^T = \begin{bmatrix} Z \\ R \end{bmatrix} \begin{bmatrix} Z & R \end{bmatrix} = \begin{bmatrix} Z^2 & 0 \\ 0 & R^2 \end{bmatrix}$ where the diagonals are always 0 by construction. We now fit the model $Y = X\alpha + R\beta + \epsilon$ as

$$
\begin{aligned}
\begin{bmatrix} \hat{\alpha} \\ \hat{\beta} \end{bmatrix} &= \left(\sum_{i=1}^{n} D^{(i)} D^{(i)\,T} \right)^{-1} \sum_{i=1}^{n} D^{(i)} Y^{(i)} \\
&= \begin{bmatrix} \left(\sum_{i=1}^{n} X^{(i)\,2} \right)^{-1} \sum_{i=1}^{n} X^{(i)} Y^{(i)} \\ \left(\sum_{i=1}^{n} R^{(i)} \right)^{-1} \sum_{i=1}^{n} R^{(i)} Y^{(i)} \end{bmatrix} \\
&= \begin{bmatrix} \left(\sum_{i=1}^{n} X^{(i)\,2} \right)^{-1} \sum_{i=1}^{n} X^{(i)} Y^{(i)} \\ \frac{1}{|\mathcal{M}_n|} \sum_{i \in \mathcal{M}_n} Y^{(i)} \end{bmatrix}
\end{aligned}
$$

where $\mathcal{M}_n = \{i : R_i = 1\}$

Linear Regression

Multiple linear regression model:

$$Y = \boldsymbol{X}^T \boldsymbol{\alpha} + \boldsymbol{R}^t \boldsymbol{\beta} + \epsilon$$

where $\boldsymbol{X} = (X_1, \ldots, X_p)^T$ and $\boldsymbol{R}, \boldsymbol{Z}, \boldsymbol{D}$ are defined similarly to before, except that $\boldsymbol{Z}$ and $\boldsymbol{R}$ are centered. Then

$$\begin{aligned}
\begin{bmatrix} \hat{\boldsymbol{\alpha}} \\ \hat{\boldsymbol{\beta}} \end{bmatrix} &= \left(\frac{1}{n} \sum_{i=1}^{n} \boldsymbol{D}^{(i)} \boldsymbol{D}^{(i)\,T} \right)^{-1} \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{D}^{(i)} Y^{(i)} \\
&\to \left(\mathbb{E}\left[\boldsymbol{D}\boldsymbol{D}^T \right] \right)^{-1} \mathbb{E}[\boldsymbol{D}Y] \\
&= \Sigma_{\boldsymbol{D}}^{-1} \mathbb{E}[\boldsymbol{D}Y]
\end{aligned}$$

Linear Regression

OLS coefficients under different scenarios:

- If the missingness is MCAR, then $\hat{\beta} \to 0$.
- If the missingness follows a self-masking mechanism, i.e. $P(\boldsymbol{R} \mid \boldsymbol{X}) = \prod_i P(R_i \mid X_i)$, and additionally $X_i \perp\!\!\!\perp X_j$ for $i \neq j$, then

$$\hat{\beta}_j \to \mathbb{E}[Y \mid X_j \text{ is missing}] - \mathbb{E}[Y \mid X_j \text{ is observed}]$$

- If the missingness is block independent in blocks $B_1, \ldots, B_d$, then for $j \in B_k$

$$\hat{\beta}_j \to \left(\mathbb{E}\left[\boldsymbol{D}_{\boldsymbol{B}_k} \boldsymbol{D}_{\boldsymbol{B}_k}^T\right]\right)^{-1} \mathbb{E}[\boldsymbol{D}_{\boldsymbol{B}_k} Y]$$

Simulated Data

Simulated data: $n = 10,000, p = 4$, binary classification.

Self-masking mechanism:

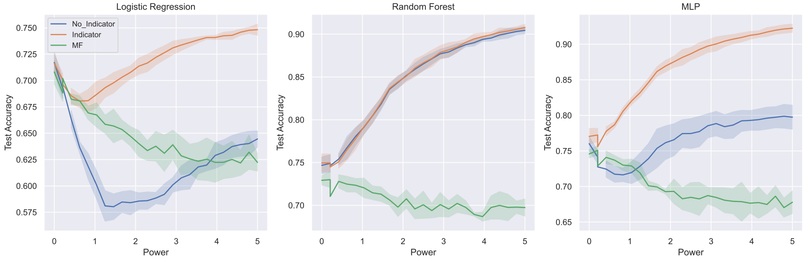
$$R_j \mid X_j \sim \text{Bernoulli}\left(p_j = \frac{1}{1 + \exp(-\gamma X_i)}\right)$$

where γ is a power parameter that controls the steepness of the sigmoid.

Methods:

- Impute with mean
- Impute with mean, add missing indicators
- Impute with missforest

Simulated Data

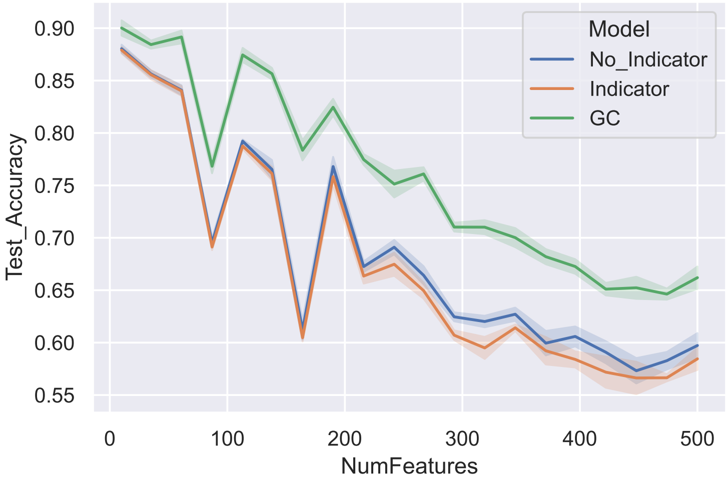


Times (in training seconds):

- No Indicator: 0.012
- MF: 11.467
- Indicator: 0.030

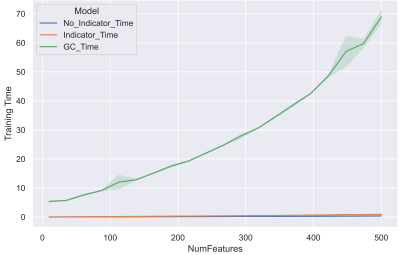
Simulated Data

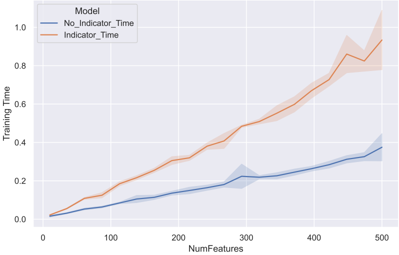
MCAR results by number of features:



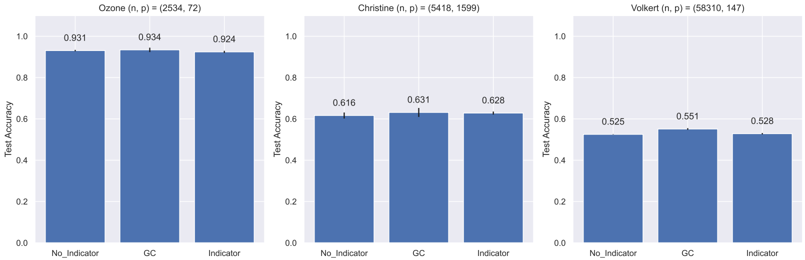
Simulated Data

MCAR times by number of features:





OpenML Datasets



In The Works

Things I hope to get to before the NeurIPS deadline:

- Explain difference between simulated high dimensional data and OpenML high dimensional data.
- Explain why convergence to 0 under MCAR is very slow.
- More interesting neural network architecture based on self-attention.

Questions?

Thank you!