Handling Missing Values in Tree-based Models:
A Brief Survey

Yibin Xiong

October 18, 2021

Table of Contents

Missing Data Mechanisms [6]

- **Missing Completely at Random (MCAR)**
  Missingness of attribute $\mathcal{A}$ do not depend on the values of $\mathcal{A}$ itself and other attributes.
- **Missing at Random (MAR)**
  Missingness of attribute $\mathcal{A}$ depend on the values of other attributes, but not the value of $\mathcal{A}$ itself.
- **Missing Not at Random (MNAR)**
  Missingness of attribute $\mathcal{A}$ depend on the value of $\mathcal{A}$ itself.
  e.g. People with higher income tend not to disclose their salary

Categories of Solutions

- Discard
  ◦ Result in *overfitting* when there are too many missing data
  ◦ Result in *bias* when missing values are NOT completely random(depend on values of existing attributes or the missing attributes themselves)
- "Leave it empty": build a new category for missing values
  ◦ Only for discrete random variables
  ◦ For continuous random variables, we can assign 0 to the missing values and add a *binary dummy variable* associated with the imputed attribute
  e.g. CatBoost [3]
- Imputation
  ◦ Statistical or learning-based estimates for missing values

Table of Contents

Typical Imputation Techniques

- Mean/Median/Mode Imputations

  ▷ Global mean or mean of KNN

  ◦ This makes **strong assumption** about the data, for instance all attribute variables are independent to each other.

  e.g. Random Forests [1]

Typical Imputation Techniques

- Mean/Median/Mode Imputations
  ▷ Global mean or mean of KNN
  ◦ This makes **strong assumption** about the data, for instance all attribute variables are independent to each other.
  e.g. Random Forests [1]
- Surrogate Tests
  ▷ Use other "relevant", value-existing attributes to predict the missing values of an attribute.
  ◦ "Relevant" to what degree?
  i) If some other attributes can completely replace the attribute $\mathcal{A}$ where there are missing values, then we do not even need $\mathcal{A}$.
  ii) If other attributes are not so relevant to $\mathcal{A}$, then the effect of surrogate test is bad.
  e.g. CART [8]

Typical Imputation Techniques

- Default Directions
  ▷ Assign all instances with missing data in an attribute to a *default* child node. Choose the default that maximizes the evaluation metric for a split.
  ◦ It requires test data to have the same pattern of missing values and may overfit to the missingness pattern of the training data
  e.g. XGBoost [2]

Typical Imputation Techniques

- Default Directions

  ▷ Assign all instances with missing data in an attribute to a *default* child node. Choose the default that maximizes the evaluation metric for a split.

  ◦ It requires test data to have the same pattern of missing values and may overfit to the missingness pattern of the training data

  e.g. XGBoost [2]

- Soft Alignment

  Let $O_1, \dots O_n$ be the outcomes of a test. If x has missing value in this attribute, we assign x to each T_i (corresponding to O_i) by probability $w_{i,x}$, where

$$\begin{aligned} w_{i,x} &= P(x \in T)\ P(x_{\mathcal{A}} \in O_i | x \in T) \\ &= P(x \in T)\ P(\bar{x} \in T_i | x \in T) \\ &= w_x\ P(\bar{x} \in T_i | x \in T) \end{aligned}$$

Typical Imputation Techniques

We estimate $P(x_{\mathcal{A}} \in O_i|x)$ using instances that has values of attribute $\mathcal{A}$. Let T_c be the set of instances with value in attribute $\mathcal{A}$

$$P(x_{\mathcal{A}} \in O_i|x) = \frac{\sum_{y \in T_c} w_y \cdot \mathbb{1}\{y_{\mathcal{A}=O_i}\}}{\sum_{y \in T_c} w_y}$$

∘ Assumes that the unknown test outcome are distributed probabilistically according to the relative frequency of the known outcomes.

∘ If $|T_c|$ is small, then the estimate is not accurate (sample estimate has high variance).

e.g. C4.5 [5]

Multiple Imputation Methods

Framework [7]:

- Propose *multiple* possible values (drawn from a distribution) to fill the missing entries and get complete data.
- Get the learned model parameters $\hat{Q}^{(i)}$ associated with each proposal. Combine the results to compute a pooled estimate $\bar{Q}$ and its variance.

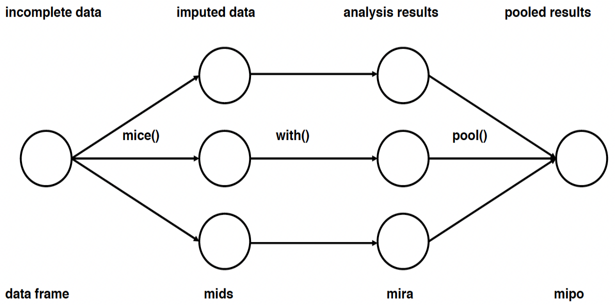

Figure 1: Main steps used in multiple imputation.

Parameter Estimation: Joint and Fully Conditional

Let $Y = \{Y_1, \ldots Y_n\}$ be the attribute variables. Y_{-j} be all but the j-th variable.

1. Model the *joint* probability of attribute variables given missingness
2. Model the *univariate conditional* probability iteratively

Here is one iteration of iterative fully conditional specification methods

Algorithm 1 Iterative FCS sampler from Liu et al. (2014)

For $1 \leq j \leq p$,

1. Sample $\theta_j \sim \pi_j(\theta_j \mid Y_{j,\text{obs}}, Y_{(-j),\text{imp}}) \propto g_j(Y_{j,\text{obs}} \mid Y_{(-j),\text{imp}}, \theta_j)\pi_j(\theta_j)$
2. Sample $Y_{j,\text{imp}} \sim g_j(Y_{j,\text{imp}} \mid Y_{j,\text{obs}}, Y_{(-j),\text{imp}}, \theta_j)$

MI Methods: Probabilistic Full Imputation [4]

Idea: Try every possible imputation and combine the results weightedly according to $P(X)$ (probability of completed data).

Advantage: Does not need strong assumptions; Tree-type independent

- Training time: minimize the expected loss over all imputations

$$\mathcal{L}(\Theta; \mathrm{D}_{\text{train}}) = \frac{1}{|\mathrm{D}_{\text{train}}|} \sum_{\mathbf{x}^o, y \in \mathrm{D}_{\text{train}}} \mathbb{E}_{p_\Phi(\mathbf{X}^m|\mathbf{x}^o)} \left[l(y, f_\Theta(\mathbf{x})) \right]$$

For MSE loss, the optimal parameter has the following closed-form:

$$\theta_\ell^* = \frac{\sum_{\mathbf{x}^o, y \in \mathrm{D}_{\text{train}}} y \cdot p_\ell(\mathbf{x}^o)/p(\mathbf{x}^o)}{\sum_{\mathbf{x}^o, y \in \mathrm{D}_{\text{train}}} p_\ell(\mathbf{x}^o)/p(\mathbf{x}^o)}$$

for each leaf $\ell \in \text{leaves}(\mathcal{T})$.

MI Methods: Probabilistic Full Imputation

- Test time: find the expected prediction over all imputations

Proposition 1 (Expected predictions for decision trees). *Given a decision tree $(\mathcal{T}, \Theta)$ encoding $f_\Theta(\mathbf{x})$, a distribution $p(\mathbf{X})$, and a partial assignment $\mathbf{x}^o$, the expected prediction of f w.r.t. p can be computed as follows:*

$$\mathbb{E}_{\mathbf{x}^m \sim p(\mathbf{X}^m | \mathbf{x}^o)} \left[f_\Theta(\mathbf{x}^o, \mathbf{x}^m) \right] = \frac{1}{p(\mathbf{x}^o)} \sum_{\ell \in \mathsf{leaves}(\mathcal{T})} \theta_\ell \cdot p_\ell(\mathbf{x}^o) \tag{4}$$

where $p_\ell(\mathbf{x}^o) = p(\mathbf{x}^{\mathsf{path}(\ell)}, \mathbf{x}^o)$ and $\mathbf{x}^{\mathsf{path}(\ell)}$ is the assignment to the RVs in $\mathsf{path}(\ell)$ *that evaluates $\mathcal{I}_\ell(\mathbf{x}') = \prod_{(n,j) \in \mathsf{path}(\ell)} [\![x'_n = j]\!]$ to 1.*

To model $p_\ell(x^o)$, we need to marginalize over RVs that are not on path ℓ.

◦ Possible models for tractable marginalization: Gaussian, GMM, decomposable probabilistic circuits(PC)

References I

- Leo Breiman. "Random forests". In: *Machine learning* 45.1 (2001), pp. 5–32.
- Tianqi Chen and Carlos Guestrin. "Xgboost: A scalable tree boosting system". In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 2016, pp. 785–794.
- Anna Veronika Dorogush, Vasily Ershov, and Andrey Gulin. "CatBoost: gradient boosting with categorical features support". In: *arXiv preprint arXiv:1810.11363* (2018).
- Pasha Khosravi et al. "Handling missing data in decision trees: A probabilistic approach". In: *arXiv preprint arXiv:2006.16341* (2020).
- J Ross Quinlan. *C4. 5: programs for machine learning*. Elsevier, 2014.
- Donald B Rubin. "Inference and missing data". In: *Biometrika* 63.3 (1976), pp. 581–592.

References II

Stef Van Buuren and Karin Groothuis-Oudshoorn. "mice: Multivariate imputation by chained equations in R". In: *Journal of statistical software* 45.1 (2011), pp. 1–67.

Xindong Wu et al. "Top 10 algorithms in data mining". In: *Knowledge and information systems* 14.1 (2008), pp. 1–37.