MINTY: Rule-based Models that Minimize the Need for Imputing Features with Missing Values

Lena Stempfle
Computer Science & Engineering
Chalmers University of Technology

Fredrik D. Johansson
Computer Science & Engineering
Chalmers University of Technology

Abstract

Rule models are often preferred in prediction tasks with tabular inputs as they can be easily interpreted using natural language and provide predictive performance on par with more complex models. However, most rule models' predictions are undefined or ambiguous when some inputs are missing, forcing users to rely on statistical imputation models or heuristics like zero imputation, undermining the interpretability of the models. In this work, we propose fitting concise yet precise rule models that learn to avoid relying on features with missing values and, therefore, limit their reliance on imputation at test time. We develop `MINTY`, a method that learns rules in the form of disjunctions between variables that act as replacements for each other when one or more is missing. This results in a sparse linear rule model, regularized to have small dependence on features with missing values, that allows a trade-off between goodness of fit, interpretability, and robustness to missing values at test time. We demonstrate the value of `MINTY` in experiments using synthetic and real-world data sets and find its predictive performance comparable or favorable to baselines, with smaller reliance on features with missing values.

1 INTRODUCTION

Linear rule models find extensive use in prediction tasks such as classification, regression, and risk scoring (Fürnkranz et al., 2012; Wei et al., 2019; Margot and Luta, 2021), and are particularly favored in domains where interpretability holds paramount importance. In the same domains, it is common for some of the variables used in the learned rules to be unobserved, missing at the time of prediction.

Established approaches to prediction with incomplete data at test time, include Bayesian modeling (Webb et al., 2010), fallback default rules (Twala et al., 2008; Chen and Guestrin, 2016), weighted estimating equations (Ibrahim et al., 2005), prediction with missingness indicators (Le Morvan et al., 2020a) and imputation (Rubin, 1976). Although imputation is general and powerful, it is not always optimal under test-time missingness (Le Morvan et al., 2020c) and often assumes that data is missing at random (MAR) (Rubin, 1976; Pedersen et al., 2017). If the distribution of missing values is preserved from training time to test time, the precise nature of the missingness mechanism is less important: when minimizing the expected error under a fixed distribution p, given a training set from p, the Bayes optimal predictive model is a function of the missingness mask and the input (Le Morvan et al., 2021b). However, such models may rely heavily on interactions between the mask and observed features, as well as imputed missing values.

A limitation of existing methods is that they either i) are specific to less interpretable model classes or ii) undermine the interpretability offered by rule-based models by relying on less interpretable auxiliary models (for imputation, estimation weighting) (Rubin, 1988) or on parameters associated with missingness itself (fallback rules, missingness mask) (Jones, 1996; Chen and Guestrin, 2016; Stempfle et al., 2023).

To address these shortcomings, we aim to *learn interpretable rule models that inherently limit the need for imputation of features with missing values.* We call our solution `MINTY`, which handles ***m****iss****in****gness* and provides ***int****erpretablit****y****y* by learning generalized linear rule models (GLRM) where literals of single variables are grouped in disjunctive rules so that the truth value of a rule can be determined when *one* of the literals is

Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238. Copyright 2024 by the author(s).

Predicting 2-year change in cognitive function (ADAS13)

Model rules	Coef.	Score
MMSE ≤ 26 OR Alzheimer's disease (AD)	+4	+4
TAU ≤ 191 OR PTAU ≤ 17	-5.2	-5.2
Married = TRUE	+3	+0
	Predicted change:	**-1.2**

Anna's features

MMSE	TAU	PTAU	MAR.	AD
24	170	N/A	N/A	No

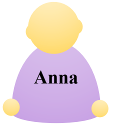

Figure 1: Illustrative example of scoring system predicting cognitive decline, measured by a change in the ADAS13 cognitive function score, using the *ADNI* data including incomplete data. The blue, underlined features indicate that these variables are observed for the specific patient, Anna, and the red shows that the observations for the variables are missing.

observed and true, no matter if the others are missing. This idea exploits redundancy in the covariate set inherent to many prediction tasks by allowing observed variables to be used as *replacements* for missing ones. We mitigate the reliance on imputation at test time by using a tunable regularization penalty on rules whose value can frequently not be determined.

Illustrative Example: Alzheimer's Progression. Figure 1 illustrates a disjunctive linear rule model for predicting cognitive decline. In the model, rules (left) are combined with coefficients (right) to calculate a predicted change in cognitive function (measured by ADAS13). The example shows the model's prediction for a patient, Anna, whose observed variables are displayed at the bottom. If at least one literal in each rule is observed and true, the added score is the same whether other variables in the rule are missing. For Anna, her TAU protein fragment level is observed to be in the range (`Tau` $\leq$ `191`), while a measurement for PTAU is missing. Despite this, the second rule can be evaluated and is true, contributing -5.2 to the final score. Similarly, the first rule is true, as we know that for Anna, `MMSE = 24`, even though she has not received a prior Alzheimer's disease (AD) diagnosis. In the case of a single-feature rule with a missing value, (e.g., `Married=True`), we default to zero-imputation, and no score is added to the total. This is common practice in the use of risk scores (Afessa et al., 2005) but may be possible to avoid by learning disjunctive rules whose value can be determined by a single observed feature and have to be zero-imputed less often.

Contributions Our contributions can be summarized as follows: 1) We propose `MINTY`, a generalized linear rule model, which uses disjunctive rules to exploit redundancy in the input variables, mitigating the need for imputation. 2) We optimize `MINTY` by adapting the column generation strategy of Wei et al. (2019), iteratively adding rules to the model based on a tunable trade-off between high predictive performance and small reliance on missing values. 3) We perform empirical experiments comparing `MINTY` to baselines that either handle missing values natively or rely on imputation. The results show that our proposed method achieves comparable prediction performance to larger black-box models and models that rely much more on features with missing values in prediction.

2 RULE MODELS & FEATURES WITH MISSING VALUES

We consider predicting an outcome $Y \in \mathbb{R}$ based on a vector of d input features $X = [X_1, ..., X_d]^\top \in \mathbb{R}^d$ when the value of any feature X_j may be missing *at training time or at test time*. Missingness is determined by a random binary mask $M = [M_1, ..., M_d]^\top \in \{0,1\}^d$ applied to a complete variable set X^*, such that $X_j = X_j^*$ if $M_j = 0$, and $X_j = \texttt{NA}$ if $M_j = 1$.

Our goal is to minimize the expected error in prediction, $R(h) := \mathbb{E}_p[L(h(X), Y)]$, over a distribution p, using a hypothesis h that handles missing values in the input X. L is a loss function such as the squared error or logistic loss. To learn, we are given a training set of examples $D = \{(x_i, m_i, y_i)\}_{i=1}^m$, assumed to be drawn i.i.d. from p. Here, $x_i = [x_{i1}, ...x_{id}]^\top$ is the (partially missing) feature vector of sample i, and m_i, y_i defined analogously. We let $\mathbf{X} \in (\{0,1\} \cup \{\texttt{NA}\})^{n \times d}, \mathbf{M} \in \{0,1\}^{n \times d}, \mathbf{Y} \in \mathbb{R}^{n \times 1}$ denote feature matrices, missingness masks and outcomes for all observations in D.

We say that a hypothesis h *relies on features with missing values* for an observation x_i if there is a feature j such that 1) $x_{ij} = \texttt{NA}$, and 2) computing $h(x_i)$ requires evaluating x_{ij} or its imputed value. We use a binary indicator $\rho_h(x_i) \in \{0,1\}$ to indicate reliance on x_i in h. For example, a dense *linear* model used with imputation (e.g., zero imputation or MICE) relies on features with missing values whenever its input x_i has any missing value. An XGBoost ensemble h has $\rho_h(x_i) = 1$ if x_i passes a "default" rule in its traversal through any of the model's trees. If the tree contains default rules, but x_i traverses neither of them, $\rho_h(x_i) = 0$. We denote the average reliance $\bar{\rho}(h) = \mathbb{E}_{X \sim p}[\rho(X)]$.

We propose `MINTY`, a learning algorithm that mitigates reliance on features with missing values by making predictions using *disjunctions* (or-clauses) of literals, e.g., "(Age > 60) or (Prior stroke)". If the value of "Age" is missing, but "Prior stroke" is True, the rule no longer depends on the value of "Age". This creates robustness by redundancy. Moreover, MINTY adds regularization to ensure that its rules can be evaluated with high probability despite missing values. We build our method on generalized linear rule models.

2.1 Generalized Linear Rule Models

In rule learning, features represent binary logical literals, where $X_{ij} = 1$ means that literal j is True for observation i. For instance, feature j may represent the literal $Age \geq 70$, and a subject i that is 73 years old would have $x_{ij} = 1$. There are standard ways to transform continuous and categorical values to literals, such as discretization by quantiles and dichotomization (Rucker et al., 2015).

Wei et al. (2019) defined generalized linear rule models (GLRM) using three components:

1. *Rule definitions* $z_k = [z_{1k}, ..., z_{dk}]^\top \in \{0,1\}^d$, for rules $k = 1, ..., K$, which define logical clauses in terms of inclusion indicators z_{jk} of literals $j \in [d]$.
2. *Rule activations* $a_i = [a_{i1}, ..., a_{iK}]^\top \in \{0,1\}^K$, where a_{ik} indicates whether rule k is satisfied ($a_{ik} = 1$) by observation x_i.
3. *Rule coefficients*, $\beta = [\beta_1, ..., \beta_K]^\top \in \mathbb{R}^K$, where β_k relates rule k to the predicted outcome. Letting rule 1 always be true, β_1 is the intercept.

In this work, we use only *disjunctive* GLRMs, were the activation of rule k for complete x_i is defined as

$$a_{ik} := \bigvee_{j=1}^{d} x_{ij} z_{jk} = \max_{j \in [d]} x_{ij} z_{jk} \ .$$

In other words, $a_{ik} = 1$ if for *any* feature j, the literal is True ($x_{ij} = 1$) and j is included in rule k ($z_{jk} = 1$).

A GLRM predicts the outcome y_i for a *complete* input x_i as a generalized linear model of the rule indicators,

$$\hat{y}_i = \Phi'(\eta_i) \ \text{ where } \ \eta_i = a_i^\top \beta$$

where Φ is the log-partition function of the conditional distribution for an exponential family model $p(Y = y \mid X = x) = h(y) \exp(\eta y - \Phi(\eta))$. For linear regression, $\Phi'(\eta) = \eta$ and for logistic regression $\Phi'(\eta) = 1/(1 + \exp(-\eta))$ is the logistic function $\sigma(\eta)$.

2.2 Mitigating Reliance on Missing Features with Disjunctive Rules

GLRMs are not designed to handle missing values by default. In this work, we treat the truth value of rules as potentially missing as well, depending on the literals included in the disjunction. Concretely,

$$a_{ik} = \begin{cases} 1, & \exists j \in z_k : m_{ij} = 0 \land x_{ij} = 1 \\ 0, & \forall j \in z_k : m_{ij} = 0 \land x_{ij} = 0 \\ \texttt{NA}, & \forall j \in z_k : m_{ij} = 1 \lor x_{ij} = 0 \end{cases} .$$

where $(j \in z_k) \Leftrightarrow (z_{jk} = 1)$. For example,

$$(x_1 \lor x_2) = \begin{cases} 1, & x_1 = 1 \text{ or } x_2 = 1 \\ 0, & x_1 = 0 \ \text{ and } \ x_2 = 0 \\ \texttt{NA}, & (x_1 = 0 \ \text{ and } \ x_2 = \texttt{NA}) \ \text{ or } \\ & (x_1 = \texttt{NA} \ \text{ and } \ x_2 = 0) \end{cases} .$$

To predict using a rule k such that $a_{ik} = \texttt{NA}$, we would still need to impute some of the missing literals.

On the other hand, *evaluating the disjunction does not rely on all of its literals being observed.* As long as one literal is observed and True, we know that the value of the disjunction is True as well. Hence, the reliance $\bar{\rho}(h)$ for a disjunctive GLRM h can be lower than for, e.g., a linear model applied to the same features.

3 MINTY: RULE MODELS THAT AVOID IMPUTATION OF MISSING VALUES

We aim to learn a small set of rules $\mathcal{S}$ and coefficients β that minimize the regularized empirical risk, with a small expected reliance on features with missing values. Let $\mathcal{K}$ denote an index over *all possible disjunctions* of d binary features and let $\mathcal{S} \subseteq \mathcal{K}$ be the subset of rules used by our model, such that k defines z_k and thus a_{ik} for all observations i. Then, let $\rho_{ik} = \mathbb{1}[a_{ik} = \texttt{NA}]$ indicate the reliance of rule k on missing values in observation x_i.

We introduce a parameter $\gamma \geq 0$ to control the average reliance on missing features $\bar{\rho}_k$ for included rules k, and a general sparsity penalty $\lambda_k > 0$, and aim to solve,

$$\min_{\beta, \mathcal{S}} \frac{1}{n} \sum_{i=1}^{n} \left[(\beta^\top a_{i\mathcal{S}} - y_i)^2 + \sum_{k \in \mathcal{S}} (\gamma \rho_{ik} + \lambda_k) |\beta_k| \right] \quad (1)$$

Following Wei et al. (2019), we use an ℓ_1-penalty for controlling the size of the rule model, with parameter $\lambda_k = \lambda_0 + \lambda_1 \|z_k\|_1$. The latter term counts the number of literals in disjunction k. We include $\sum_i \rho_{ik}$ as a factor in the penalty to discourage models from using rules that both have a large influence on the prediction (high $|\beta_k|$) and frequent missingness (high $\sum_i \rho_{ik}$). By choosing $\lambda_0, \lambda_1, \gamma$, we can control the number, size and missingness reliance of rules used by the model.

If we let $\mathcal{S}$ be the set of all possible disjunctions $\mathcal{K} = \{0,1\}^d$, our learning problem reduces to a LASSO-like problem with active rules determined by the sparsity pattern in β, but with a number of rules and coefficients that grows exponentially with d. Even for moderate-size problems, these would be intractable to enumerate. Instead, we follow the column-generation strategy by Wei et al. (2019), which searches the space of disjunctions and builds up S incrementally.

The idea is to first solve problem (1) restricted to a small set of candidate rules $\hat{\mathcal{S}} = \mathcal{S}_0$, in our case just the intercept rule. Given a current set of disjunctions $\hat{\mathcal{S}}$ and estimated coefficients $\hat{\beta}$, a new rule is added by finding the disjunction that aligns the most with the residual of the current model, $\mathbf{R} = \mathbf{A}_{\hat{\mathcal{S}}}\hat{\beta} - \mathbf{Y}$, where $\mathbf{A}_{\hat{\mathcal{S}}} = [a_{1\cdot}, \ldots, a_{n\cdot}]^\top$ is the matrix of rule assignments for all observations in the training set w.r.t. $\hat{\mathcal{S}}$.

This procedure is justified by the optimality conditions of (1) which imply that at an optimal solution, the partial derivative with respect to both the positive and negative components of β must be non-negative. Optimality can therefore be determined by minimizing $\pm\frac{1}{n}\mathbf{R}^\top\mathbf{a} + \mathcal{R}(a)$ over the corresponding activations of a new rule $\mathbf{a} \in \{0,1\}^n$ (with $\mathcal{R}(a)$ corresponding to regularization terms, specified further below).

To avoid computation with `NA` values, we zero-impute X, defining $\bar{x}_{ij} = \mathbb{1}[m_{ij} = 0]x_{ij}$, keeping track of missing values in the mask M. In principle, other imputation could be used. We choose the next rule as defined by the minimizer z^* of the following two problems (±),

$$\begin{aligned} \underset{\substack{z\in\{0,1\}^d\\ a,\rho\in\{0,1\}^n}}{\text{minimize}} \quad & \pm\frac{1}{n}\sum_{i=1}^{n}(r_i a_i + \gamma\rho_i) + \lambda_0 + \lambda_1\sum_{j=1}^{d} z_j \\ \text{subject to} \quad & a_i = \sum_{k=1}^{K}\max(\bar{x}_{ij}z_j) \\ \forall i: \rho_i = & \underbrace{(1-\max_j[(1-M_{ij})z_j\bar{x}_{ij}])}_{(i)}\underbrace{(\max_j M_{ij}z_j)}_{(ii)} \end{aligned} \tag{2}$$

We let $z_{k^*}, a_{k^*}, \rho_{k^*}$ refer to the optimizers of (2), for the sign with smallest objective value, and δ_{k^*} to the corresponding objective. The first constraint in (2) makes sure that rule activations a_i correspond to a disjunction of literals $\bar{x}_{ij}$ as indicated by z. The constraint on ρ_i ensures that reliance on missing factors is counted only when (i) there is no observed True literal in the rule, and (ii) at least one literal is missing.

When no rule can be found with a negative solution to (2), or a maximum number of rules k_{max} has been reached, the algorithm terminates. We finish by solving (1) with respect to β for fixed $\hat{\mathcal{S}}$. The algorithm can be adapted to generalized linear models like logistic regression, without changing the rule generation procedure, as shown by Wei et al. (2019). We summarize our method, referred to as `MINTY`, in Algorithm 1.

Algorithm 1 `MINTY` learning algorithm

```
Input: $\mathbf{X}, \mathbf{M} \in \{0,1\}^{n\times d}$, $\mathbf{Y} \in \mathbb{R}^n$
Parameters: $\lambda_0, \lambda_1, \gamma \geq 0, k_{max} \geq 1$
Output: $\mathcal{S}$, $\beta$
Initialize $\hat{\mathcal{S}} = \{0\}$ where 0 is the intercept rule
Initialize $\delta_{k*} = -\infty$
Let $\bar{\mathbf{X}}$ be zero-imputed $\mathbf{X}$, $\bar{x}_{ij} = \mathbb{1}[m_{ij} = 0]x_{ij}$
Let $l = 0$
while $\delta_{k*} < 0$, $l < k_{max}$ do
    $\beta \leftarrow \arg\min_\beta \mathcal{O}(\bar{\mathbf{X}}, \mathbf{Y}, \hat{\mathcal{S}}, \lambda_0, \lambda_1, \gamma)$    $\triangleright$ (1)
    $a_{ik} = \max_{j\in[d]} z_{jk}\bar{x}_{ij}$ for $i \in [n], k \in \hat{\mathcal{S}}$
    $r_i = \sum_{k\in\hat{\mathcal{S}}} \beta_k a_{ik} - y_i$ for $i \in [n]$
    $z_{k^*}, \delta_{k^*} \leftarrow \text{ADD}(\bar{\mathbf{X}}, \mathbf{Y}, \mathbf{R}, \lambda_0, \lambda_1, \gamma)$    $\triangleright$ (2)
    if $\delta_{k^*} \geq 0$: then
        break. The current solution is optimal.
    else
        Append new rule $k^*$ to $\hat{\mathcal{S}}$,
        $l \leftarrow l + 1$
    end if
end while
$\hat{\beta} \leftarrow \arg\min_\beta \mathcal{O}(\bar{\mathbf{X}}, \mathbf{Y}, \hat{\mathcal{S}}, \lambda_0, \lambda_1, \gamma)$
return $\hat{\mathcal{S}}, \hat{\beta}$
```

As an alternative, $\bar{\rho}$ could be defined as the proportion of missing/undetermined components (rules) in a rule model, rather than the proportion of observations with ≥ 1 missing rule. Compared to our definition, such a penalty would be harsher on models for which multiple rules are missing for the same observation, and more lenient when only one rule is missing.

3.1 Solving the Rule Generation Problem

The problem in (2) is an integer linear program with nonlinear constraints. We consider two methods in experiments: Exact solutions using the off-the-shelf optimization toolkit Gurobi (Gurobi Optimization, LLC, 2023), and approximate solutions using a heuristic beam search algorithm, as used by Oberst et al. (2020).

For the beam search algorithm, we initialize the beam to contain all disjunctions of a single literal. We then retain the top-W_b of these, in terms of the objective in (2). Then, we generate the next set of candidates by adding one literal to all disjunctions, and evaluate these in the same way, retaining the top-W_b and proceeding in the same way until at most D_b literals have been added. Throughout, we keep track of the rule with the smallest objective, no matter its size, and return this once the beam has reached its maximum depth. In experiments, we let the beam width be $W_b = d$ (the number of features) and depth $D_b = 7$. The time complexity of the search is linear in $W_b D_b$.

3.2 MINTY in the Limits of Regularization

In our proposed method, we penalize reliance on missingness to disjunctive linear rule models, controlling the emphasis on observed literals within rules, with the parameter $\gamma \geq 0$. In the low limit, $\gamma = 0$, MINTY is equivalent to a disjunctive linear rule model with zero-imputation. In stationary environments, where $p(X, M, Y)$ doesn't change between training and testing, for sufficiently large data sets, learning with $\gamma = 0$ will result in the smallest error in general since this imposes the least constraints on the solution. This comes at the cost of reduced interpretability by relying on features with missing values in prediction.

In the limit $\gamma \to \infty$, MINTY imposes a hard constraint that no rule should be included in the model unless it can be evaluated for every example in the training set without relying on imputed values, $\forall i, k : \rho_{ik} = 0$. This could be appropriate in settings where there are *some* features that are never missing and would be preferred over features that are predictive but rarely measured. However, if any configuration $m \in \{0, 1\}^d$ of missing values is possible, MINTY will return an empty set of rules in the large-sample limit.

Observation 1. If the all-missing configuration has positive marginal probability, $\exists \epsilon > 0 : p(M = \mathbf{1}_d) > \epsilon$, the set of rules which have at least one literal measured for every example in the training set vanishes almost surely with a growing number of samples n. As a result, there is no non-trivial GLRM h with $\bar{\rho}(h) = 0$. An important special case of this is the Missingness-Completely-At-Random (MCAR) mechanism (Rubin, 1976) with missingness probability $q > \epsilon^{1/d}$. In other words, requiring *perfect* variable redundancy through rules by letting $\gamma \to \infty$ is too strict for many settings. Instead, we can aim to *limit* or *minimize* the reliance on missing values $\bar{\rho}$ by selecting a moderate γ.

3.3 Comparison With a Linear Model Trained on Complete Data

In many applications, interpretable risk scores trained on complete cases are deployed in settings where features are occasionally missing, necessitating the imputation of missing values with a constant, often 0 for binary variables. One example is the APACHE family of clinical risk scores (Afessa et al., 2005; Haniffa et al., 2018). It is natural to compare the bias of this approach to the bias of a model with inherently low reliance on missing values. Below, we do this for the case where the true outcome is a linear function and the variable set has a natural redundancy.

Assume that the outcome Y is linear in $X \in \{0, 1\}^d$ and has noise of bounded conditional variance,

$$Y = \beta^\top X + \epsilon(X), \text{ where } \mathbb{E}[\epsilon \mid X] = 0, \mathbb{M}[\epsilon \mid X] \leq \sigma^2 ,$$

with $\beta \in \mathbb{R}^d$. Next, assume that X has the following structure. For each X_i there is a paired "replacement" variable $X_{j(i)}$, with $j(j(i)) = i$, such that for $\delta \geq 0$, $p(X_i = X_{j(i)}) \geq 1 - \delta$, and that whenever X_i is missing, $X_{j(i)}$ is observed, $M_i = 1 \Rightarrow M_{j(i)} = 0$. Assume also that $\forall i, k \notin \{i, j(i)\} : X_i \perp\!\!\!\perp X_k$.

Proposition 1. *Under the conditions above, there is a GLRM h with d two-variable rules $\{\bar{X}_i \vee \bar{X}_{j(i)}\}_{i=1}^d$, where $\bar{X}_i = (1 - M_i)X_i$, with expected the squared error*

$$R(h) \leq \delta \|\beta\|_2^2 + \delta^2 \sum_{i,k \notin \{i,j(i)\}} |\beta_i \beta_k| + \sigma^2 \ .$$

Additionally, if $\beta_i \geq 0$ and $\mathbb{E}[X_i M_i] \geq \eta$ for all $i \in [d]$, using the ground truth β (the ideal complete-case model) with zero-imputed features $\bar{X}$ results in an expected squared error bounded from below as

$$R(\beta) \geq \eta \|\beta\|_2^2 + \sigma^2 \ ,$$

and a greater missingness reliance than the GLRM, $\bar{\rho}(\beta) \geq \bar{\rho}(h)$. Thus, with $a = \|\beta\|_2^2 / \sum_{i,k \notin \{i,j(i)\}} |\beta_i \beta_k|$, the GLRM is preferred when $\delta < (\sqrt{a^2 + 4\eta} - a)/2$.

A proof is given in the Appendix.

By Proposition 1, there are data-generating processes for which a disjunctive GLRM has a strictly smaller risk and smaller reliance on features with missing values than the ground-truth linear rule model used with zero imputation. For simplicity, the result is written for rules involving pairs of variables that are internally strongly correlated and independent of other pairs but can be generalized to disjunctions of variables in cliques of any size with the same property.

4 EMPIRICAL STUDY

We evaluate the proposed `MINTY` algorithm[1] on synthetic and real-world data, aiming to answer three main questions: i) How well can we learn rules when covariates are missing at training and test time? ii) How does the accuracy of `MINTY` compare to baseline models; iii) How does regularizing reliance on missing values affect performance and interpretability?

4.1 Experimental Setup

In our experiments, we solve the rule-generation subproblem of `MINTY` using beam search, as described in

[1] Code and instructions for reproducing experiments are available at `https://github.com/Healthy-AI/minty`.

Section 3.1. In the Appendix, we use a small synthetic data set to show that the predictive performance differs only minimally compared to solving the ILP in (2) exactly using Gurobi (Gurobi Optimization, LLC, 2023). To find optimal coefficients β, given rule definitions S, we use the `LASSO` implementation in scikit-learn (Buitinck et al., 2013), re-weighting covariates to achieve variable-specific regularization. Missing values were zero-imputed for `MINTY` with the original missingness mask informing the missingness reliance penalty.

The objective function regularizes each rule $z_{\cdot k}$ with strength $\lambda_k = \lambda_0 + \lambda_1 \|z_{\cdot k}\|_0$, limiting the reliance on missingness by minimizing the number of rules using zero-imputed features. The values of λ_0 and λ_1 range within $[10^{-3}, 0.1]$. We choose their best values through a grid search based on their validation set performance. The values for γ were chosen from $[0, 10^{-7}, 10^{-3}, 0.01, 0.1, 10000]$. The number of rules used by the model is set to 20. We present the result for several values of γ, to illustrate the tradeoff between performance and reliance on missing values. To mitigate the effects of scaling, outcome variables were normalized during training and re-scaled for evaluation to estimate the RMSE in the original scale of the outcome variable for each data set, respectively.

We compare `MINTY` to the baselines: Imputation + `LASSO` regression, Imputation + Decision Tree (`DT`), Imputation + `RULEFIT` (Friedman and Popescu, 2008), and XGBoost (`XGB`), where missing values are supported by default (Chen et al., 2019). Last, we compare to NeuMiss networks (`NEUMISS`) that use a new type of non-linearity: the multiplication by the missingness indicator (Le Morvan et al., 2020b). For imputation, we use zero (I_0) or multiple iterative imputation (I_{mice}) from Scikit-Learn (Pedregosa et al., 2011b; Van Buuren, 2018), that replaces missing values with multiple imputations using chained regressions. Iterative imputation was performed over 5 iterations. Details about implementations, hyperparameters, and evaluation metrics are given in Appendix A.1.

We report the average RMSE and R^2, and their respective standard deviations over 10 random train/test splits of the data. Additionally, we estimate the reliance on features with missing values, $\bar{\rho}$ of all methods on the test sets. For `LASSO`, this counts the fraction of observations with missing values among the features with non-zero coefficients. For `DT` and `RULEFIT`, we report the fraction of inputs with a feature that is both missing (and thus imputed) and used in a split to decide that input prediction. For `XGB`, we do the same, but count observations for which *any* of the trees rely on a missing value. `NEUMISS` uses all variables for prediction, and so $\bar{\rho}$ measures the fraction of observation with at least one missing value. For `MINTY`, we define $\bar{\rho}$ as explained in Section 3.

Real-world Data Sets We used three different regression tasks for evaluation. The first task, *ADNI*, is sourced from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database and involves predicting the outcome of the ADAS13 cognitive test at a 2-year follow-up based on baseline data. In the second task, *Life*, we aim to predict life expectancy from various factors, including immunization, mortality, economic, and social factors (Roser et al., 2013). Last, the task called *Housing* involves predicting property prices in Ames, Iowa, using physical attributes and geographical features (De Cock, 2011). The data sets were discretized and used with binary data for `MINTY`. For the baselines, we use the original continuous values and one-hot encode the categorical values using the StandardScaler by Scikit-learn (Pedregosa et al., 2011a). More details can be found in the Appendix. We split the data randomly into a test set (20%) and a training set (80%), withholding a validation portion (20%) of the training set for selecting hyperparameters.

Missing Values *ADNI* has incomplete entries natively, indicated in results as "(Natural)" missingness. We added missing values to *Life* and *Housing* according to the Missing Completely at Random (MCAR) mechanism, where the probability that a feature X_j has a missing value is q, independent of other variables. In our experiments, we set q to 0.1. The same mechanism was used both during training and testing.

Synthetic Data In the Appendix, we also apply our algorithms to synthetic data where $n = 5000$ samples of $c = 30$ features are drawn from independent Bernoulli variables. Then, for each variable X_i, $i \in [c]$ a "replacement variable" X_{c+i}, is added which has the same value as X_i with probability 0.9. The outcome Y is a linear combination of all features with added noise. Missingness was added with different mechanisms—MCAR, Missing-At-Random (MAR), or Missing-Not-At-Random (MNAR) using the implementation by Mayer et al. (2019).

4.2 Results

We report the predictive performance of all models and their reliance on features with missing values in Tables 1–2, and comment on their interpretability.

Overall, `MINTY` achieves good held-out predictive performance (high R^2, low RMSE), comparable with other models across all data sets (*ADNI*, *Housing*, *Life*), while relying substantially less on features with missing values in the test set (smaller $\bar{\rho}$) than models with similar predictive accuracy. On *ADNI*, a `MINTY`

Table 1: Performance results for the real-world data sets *ADNI* and *Housing*. For MINTY using *ADNI* we use $\lambda_0 = 0.001, \lambda_1 = 0.01$, and for *Housing* we choose $\lambda_0 = 0.001$, $\lambda_0 = 0.001$ based on a 0.1 missingness proportion in the data. DT, XGB, RULEFIT, and LASSO are trained on non-discretized data, and all versions ofMINTY and NEUMISS on discretized data.

	ADNI (Natural)			HOUSING (MCAR)		
Model	R^2(std)	**RMSE** (std) ADAS13 score	$\bar{\rho}$	R^2 (std)	**RMSE** (std) $10k	$\bar{\rho}$
$\texttt{LASSO}_{I_{mice}(\text{A}),I_0(\text{H})}$	0.65 (0.02)	5.08 (0.12)	0.51	0.57 (0.06)	5.13 (0.58)	0.83
$\texttt{DT}_{I_0(\text{A}),I_{mice}(\text{H})}$	0.58 (0.03)	5.63 (0.26)	0.15	0.66 (0.06)	4.53 (0.58)	0.18
XGB	0.66 (0.02)	5.19 (0.16)	0.55	0.84 (0.05)	3.10 (0.08)	0.99
$\texttt{RULEFIT}_{I_0}$	0.64 (0.02)	5.15 (0.21)	0.43	0.68 (0.05)	4.49 (0.62)	0.60
NEUMISS	0.61 (0.04)	5.60 (0.30)	0.55	0.55 (0.04)	5.60 (0.24)	1.0
$\texttt{MINTY}_{\gamma=0}$	0.64 (0.02)	5.22 (0.19)	0.40	0.71 (0.04)	4.18 (0.49)	0.76
$\texttt{MINTY}_{\gamma=0.01(\text{A}),\gamma=0.1(\text{H})}$	0.63 (0.02)	5.27 (0.23)	0.27	0.72 (0.03)	4.05 (0.44)	0.49
$\texttt{MINTY}_{\gamma=1e4}$	0.62 (0.02)	5.27 (0.18)	0.0	0.47 (0.06)	5.64 (0.47)	0.0

Table 2: Performance results for real-world data set *Life* with $\lambda_0 = 0.001$, $\lambda_1 = 0.001$ for MINTY. The missingness proportion is 0.1. DT, XGB, RULEFIT, NEUMISS and LASSO are trained on non-discretized data, and all MINTY versions used discretized data.

	LIFE (MCAR)		
Model	R^2(std)	**RMSE** (std) years	$\bar{\rho}$
$\texttt{LASSO}_{I_0}$	0.89 (0.01)	3.00 (0.15)	0.86
$\texttt{DT}_{I_0}$	0.95 (0.01)	2.07 (0.19)	0.31
XGB	0.99 (0.01)	1.08 (0.10)	0.88
$\texttt{RULEFIT}_{I_0}$	0.76 (0.04)	4.55 (0.35)	0.27
NEUMISS	0.72 (0.31)	4.35 (0.57)	0.88
$\texttt{MINTY}_{\gamma=0}$	0.91 (0.01)	2.76 (0.12)	0.77
$\texttt{MINTY}_{\gamma=0.03}$	0.87 (0.01)	3.35 (0.13)	0.24
$\texttt{MINTY}_{\gamma=2.5}$	0.50 (0.05)	6.62 (0.33)	0.0

model with $\bar{\rho} = 0$ outperforms DT with higher performance and less reliance on missing values, despite DT having the lowest missingness reliance among the baselines. A MINTY model with $\bar{\rho} = 0.01$ achieves better R^2 than NEUMISS and similar performance as LASSO and XGB, models for which more than 50% for of the test samples must use default rules or be imputed, respectively. MINTY's results confirm that it is possible to learn to avoid imputation to a large degree while maintaining a competitive model. We see similar results on *Housing* and *Life*, despite the missingness being unstructured in these examples (MCAR). On *Life*, MINTY suffers from a coarse discretization (4 bins) of continuous values (e.g., Infant deaths); baselines use native rule generation strategies (e.g., decision tree splitting).

In Appendix Table 5, we compare all models on synthetic data in MCAR, MAR, and MNAR settings and see that MINTY$\gamma = 0.01$ is among the best-performing models regardless of the missingness mechanism. We note that XGB (tree ensemble) and NEUMISS (multilayer neural network) support prediction with missing values natively and perform well in all tasks, but can be difficult to interpret due to their large size and/or black-box nature. RULEFIT leverages random forests, breaking down each tree into decision rules for extra features in a Lasso model. Despite strong performance across data sets, its reliance on imputed values and low interpretability due to over 20 rules limit its utility.

In Appendix Figure 3, we report the R^2 values on *ADNI*, together with estimator-specific measures of complexity. These results, the results in Tables 1–2, and the model description in Table 3 confirm that MINTY can be used to learn (more) interpretable models while handling missing values at test time. Notably, across all data sets, DT also relies less on missing values than other baselines, simply because not every variable will be used to compute the prediction for every test instance. Building trees with explicit regularization for ρ is worth further investigation.

The Impact of Regularizing $\bar{\rho}$ For all data sets, there are values of $\gamma > 0$ such that $\texttt{MINTY}_{\gamma>0}$ and $\texttt{MINTY}_{\gamma=0}$ differ minimally in R^2 and RMSE values but where $\texttt{MINTY}_{\gamma>0}$ shows substantially lower reliance on imputation. For example, on *Housing*, $\texttt{MINTY}_{\gamma=0.1}$ achieves almost the same R^2 as $\texttt{MINTY}_{\gamma=0}$ but with reliance $\bar{\rho} = 0.49$ compared to $\bar{\rho} = 0.76$ for the unregularized model. As remarked previously, achieving $\bar{\rho} = 0$ with non-trivial predictive performance is not always possible: on *Life*, the upper extreme of $\gamma = 1000$ leads to a notably less effective model, since there were no rules which were always determined by observed values other than the intercept.

In Figure 2, we show the results of MINTY for 20 values of γ from a log-scale range over $[10^{-6}, 1000]$. For $\gamma = 1000$, the model disallows any use of missing values in the rules ($\bar{\rho} = 0$), which leads to worse predictive performance (bottom left in Figure). In the top right cor-

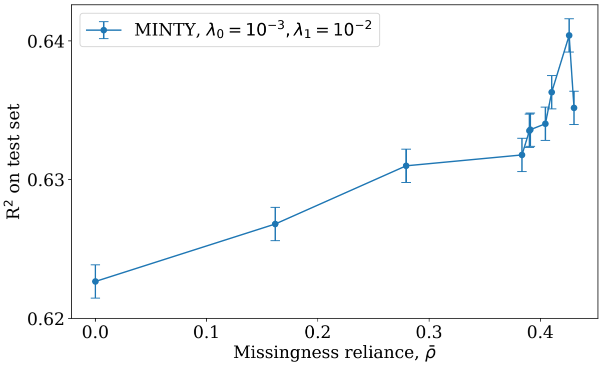

Figure 2: Predictive performance (R^2) and reliance on features with missing values $\bar{\rho}$ on *ADNI* for `MINTY` with γ chosen from a log scale over $[10^{-6}, 10^3]$.

ner, $\gamma = 0$ results in the best predictive performance, but the highest reliance on missing values. Regularizing the reliance $\bar{\rho}$ moderately ($\gamma = 0.01$) leads to a good balance of predictive accuracy ($R^2 = 0.63$) and reliance on imputation ($\bar{\rho} = 0.27$).

4.3 Interpreting Learned Rules on *ADNI*

In Table 3, we visualize the models learned by `MINTY` on *ADNI*, in the style of risk scores used in medicine or criminal justice, see, e.g., Ustun and Rudin (2019). On the left are rule definitions and on the right, their coefficients—the score added if the rule is true. The scores for each active rule are summed together with the intercept to form a prediction. The top table represents the learned set of rules using $\texttt{MINTY}_{\gamma=0}$ and the bottom one for $\texttt{MINTY}_{\gamma=0.01}$.

In the *ADNI* task, the goal is to predict the cognitive decline measured by a change in the cognitive test score ADAS13 (high score means low cognitive ability, a positive change means deteriorating ability) from baseline to a 2-year follow-up. The learned coefficients match expectations as, for example, diagnoses of Alzheimer's disease (AD) or mild cognitive impairment (LMCI) are associated with higher cognitive decline (positive coefficients). Similarly, `MMSE` $\geq$ `29` (normal cognitive ability) is associated with a smaller decline in ADAS13 (negative coefficient).

The two models with $\gamma = 0$ and $\gamma = 0.01$ learn similar rules with similar coefficients but with different reliance on features with missing values ($\bar{\rho} = 0.40$ vs $\bar{\rho} = 0.27$). The rules, `TAU` $\leq$ `191.1 OR Hippocampus` $\geq$ `7721.0` and `FDG` $\leq$ `1.163` are not included in the second model ($\gamma = 0.01$), since they are missing for 0.33% and 0.27% of all individuals in the data set. By using a higher γ we achieve a more robust solution with less dependence on imputed values.

Table 3: `MINTY` models learned on *ADNI* using $\gamma = 0$ (top) and $\gamma = 0.01$ (bottom). The R^2 for the two models were 0.64 and 0.63 respectively, the latter with smaller reliance on features with missing values ($\bar{\rho} = 0.28$ vs $\bar{\rho} = 0.40$). Two rules in the top model are not in the bottom model due to more frequent missingness; the bottom model adds two rules with less missingness.

Rules by `MINTY` with $\gamma = 0$	Coeff.
AD diagnosis OR LMCI diagnosis	+0.35
MMSE $\leq$ 26.0 OR LMCI diagnosis	+0.23
LDELTOTAL $\leq$ 3.0	+0.63
AD diagnosis	+0.65
Hippocampus $\leq$ 6071.0 OR Sex = Male	+0.18
MMSE $\geq$ 29.0	−0.16
Entorhinal $\leq$ 3022.0	+0.18
LDELTOTAL score $3 - 8$	+0.27
TAU $\leq$ 191.1 OR Hippocampus $\geq$ 7721.0	−0.19
FDG $\leq$ 1.163	+0.17
Intercept	-0.57

Rules by `MINTY` with $\gamma = 0.01$	Coeff.
AD diagnosis OR LMCI diagnosis	+0.36
MMSE $\leq$ 26.0 OR LMCI diagnosis	+0.22
LDELTOTAL $\leq$ 3.0	+0.67
AD diagnosis	+0.68
Hippocampus $\leq$ 6071.0 OR Sex = Male	+0.19
MMSE $\geq$ 29	−0.17
Entorhinal $\leq$ 3022.0	+0.17
LDELTOTAL score $\in [3, 8]$	+0.28
Hippocampus $\geq$ 7721.0	-0.16
APOE4 = 1	+0.08
Intercept	-0.61

For $\texttt{MINTY}_{\gamma=0.1}$, which achieves $\bar{\rho} = 0$, shared in Table 6 in the Appendix, we see that the learned rules contain mostly features that are *always* measured such as demographics and cognitive test scores, following the constraint that rules should not be included unless it can be evaluated for every example. We also show an example in Table 7 in the Appendix, where the true rules produced by synthetic data are recovered.

5 RELATED WORK

Predicting with missing values The rich literature on learning from data with missing values, see e.g., Little and Rubin (2019); Mayer et al. (2019), studies both a) settings in which complete inputs are expected at test time but have missing values during training, and b) predictive settings where missing values are expected also during testing (Josse et al., 2019). Studies of the first category have produced impressive results that give inference guarantees under different missingness mechanisms, such as MCAR, MAR, MNAR (Rubin, 1976) and have often focused on imputing missing values with model-based techniques (Van Buuren, 2018). Our work falls firmly in the second category, born out of supervised learning: rather than assuming that a particular mechanism generated missingness, we assume that the mechanism is preserved at test time (Josse et al., 2019).

Two common strategies in our setting are to i) impute-then-regress—to impute missing values and proceed as if they were observed, or ii) build models that explicitly depend on the missingness mask M, indicators for missing values (Little and Rubin, 2019). The former approach can introduce avoidable bias even with powerful imputation methods in the setting where values are missing in the same distribution during testing as during training (Le Morvan et al., 2021a). Josse et al. (2019) showed that pairing constant imputations, for example with 0, with a sufficiently expressive model leads to consistent learning. A drawback of this is that the optimal imputation or regression models are often complex and challenging to interpret.

The second strategy has resulted in diverse methods, many of which incorporate the missingness mask in deep learning (Bengio and Gingras, 1995; Che et al., 2018; Le Morvan et al., 2020c; Nazabal et al., 2020). Recently, NeuMiss networks (Le Morvan et al., 2020b) introduced a deep neural network architecture that applies a non-linearity based on the missingness mask to learn optimal linear predictive models. Another approach is the so-called Missing Incorporated in Attribute (MIA) (Twala et al., 2008) which uses missingness itself as a splitting criterion in tree learning, as used by e.g., XGBoost (Chen and Guestrin, 2016). A drawback of these methods is that they are difficult to interpret due to their complexity. In concurrent work, Chen et al. (2023) addressed missing values with explainable machine learning but focused on a different model class from ours, using explainable boosting machines (EBMs) to gain insights without relying on imputation or specific missingness mechanisms.

Rule models and missing values Rule-based models, such as decision trees, are relatively easy to understand because they mimic human decision-making processes (Molnar, 2022). Examples of rule-based methods aimed at interpretability include RuleFit, which utilizes rule ensembles by a linear model of tree-based decision rules (Friedman and Popescu, 2008) (more detail in the Appendix A.1), and Node Harvest (Meinshausen, 2010) that merges the benefits of individual trees and tree ensembles, producing sparse, interpretable results, particularly in low signal-to-noise situations. Most rule-based models do not natively handle missing values at test time, but there are notable exceptions, such as XGBoost, described earlier. Node Harvest handles missing values at test time by letting observations be "members" of a node only if all of the characterizing features of the node are observed. This strategy is reminiscent of `MINTY`, in which rule activations are affected only by observed features.

6 DISCUSSION

We have proposed `MINTY`, a generalized linear rule model that mitigates reliance on missing values by a) using disjunctive rules whose values can be computed as long as one of its literals is observed and true, and b) regularizing the inclusion of rules whose values can frequently not be determined. We demonstrated in experiments on real-world data that `MINTY` often has similar accuracy to black-box estimators and outperforms competitive baselines while maintaining interpretability and minimizing the reliance on missing values.

`MINTY`'s design takes inspiration from the widely-used structure of risk scores, with rules defined by disjunctions of literals. If a disjunction includes literals that are often missing, and the value of the rule cannot be determined, its coefficient may be smaller than that of a rarely missing rule. Thus, coefficients may reflect patterns of missingness more than the association of the features with the outcome itself. We acknowledge this challenge of interpreting `MINTY`'s coefficients as causal effects, however, this issue is not unique to `MINTY`. Similar challenges arise in all (generalized) linear models when faced with model misspecification or the presence of unobserved variables, whether due to selection biases or data generation processes. Specifically, if there is a correlation between an observed and an unobserved variable, it may affect the observed variable's coefficient, causing it to increase or decrease.

Limitations in our work include the heuristic approximation algorithm used to solve the column generation problem in `MINTY`—an optimal solution could yield different rules and coefficients. Although our code is equipped to utilize an exact solver, this option was not used in real-world experiments due to time and computational resource constraintsFurthermore, we evaluated methods on *Housing* and *Life* only using synthetic MCAR missingness; only *ADNI* data set holds natural missingness. Future work on prediction with test-time missingness would do well to establish challenging benchmarks with natively missing values. Finally, although the examples in our paper were all from healthcare, `MINTY` has potential uses in various fields, such as finance for fraud detection or e-commerce for recommendations. While the linear parameterization may be limiting, future work could explore applying the same principle of mitigating reliance on missing values in other model classes, such as decision trees.

Acknowledgements

This work was supported in part by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. Computations were enabled by resources provided by the Swedish National Infrastructure for Computing (SNIC) partially funded by the Swedish Research Council through grant agreement no. 2018-05973, and the National Academic Infrastructure for Supercomputing in Sweden (NAISS).

Data used in preparation of this article were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (adni.loni.usc.edu). As such, the investigators within the ADNI contributed to the design and implementation of ADNI and/or provided data but did not participate in the analysis or writing of this report. A complete listing of ADNI investigators can be found at: `http://adni.loni.usc.edu/wp-content/uploads/how_to_apply/ADNI_Acknowledgement_List.pdf`

References

Bekele Afessa, Mark T Keegan, Ognjen Gajic, Rolf D Hubmayr, and Steve G Peters. The influence of missing components of the acute physiology score of apache iii on the measurement of icu performance. *Intensive care medicine*, 31:1537–1543, 2005.

Yoshua Bengio and Francois Gingras. Recurrent neural networks for missing or asynchronous data. *Advances in neural information processing systems*, 8, 1995.

Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 8(1):1–12, 2018.

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*, pages 785–794, 2016.

Tianqi Chen, Tong He, Michael Benesty, and Vadim Khotilovich. Package 'xgboost'. *R version*, 90, 2019.

Zhi Chen, Sarah Tan, Urszula Chajewska, Cynthia Rudin, and Rich Caruna. Missing values and imputation in healthcare data: Can interpretable machine learning help? In *Conference on Health, Inference, and Learning*, pages 86–99. PMLR, 2023.

Dean De Cock. Ames, iowa: Alternative to the boston housing data as an end of semester regression project. *Journal of Statistics Education*, 19(3), 2011.

Jerome H Friedman and Bogdan E Popescu. Predictive learning via rule ensembles. *The annals of applied statistics*, pages 916–954, 2008.

Johannes Fürnkranz, Dragan Gamberger, and Nada Lavrač. *Foundations of rule learning*. Springer Science & Business Media, 2012.

Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2023. URL `https://www.gurobi.com`.

Rashan Haniffa, Ilhaam Isaam, A Pubudu De Silva, Arjen M Dondorp, and Nicolette F De Keizer. Performance of critical care prognostic scoring systems in low and middle-income countries: a systematic review. *Critical care*, 22:1–22, 2018.

Joseph G Ibrahim, Ming-Hui Chen, Stuart R Lipsitz, and Amy H Herring. Missing-data methods for generalized linear models: A comparative review. *Journal of the American Statistical Association*, 100 (469):332–346, 2005.

Michael P. Jones. Indicator and stratification methods for missing explanatory variables in multiple linear regression. *Journal of the American Statistical Association*, 91(433):222–230, 1996. ISSN 01621459.

Julie Josse, Nicolas Prost, Erwan Scornet, and Gaël Varoquaux. On the consistency of supervised learning with missing values. *arXiv preprint arXiv:1902.06931*, 2019.

Marine Le Morvan, Julie Josse, Thomas Moreau, Erwan Scornet, and Gaël Varoquaux. Neumiss networks: differentiable programming for supervised learning with missing values. *Advances in Neural Information Processing Systems*, 33:5980–5990, 2020a.

Marine Le Morvan, Julie Josse, Thomas Moreau, Erwan Scornet, and Gaël Varoquaux. NeuMiss networks: differentiable programming for supervised learning with missing values. *arXiv:2007.01627*, 2020b.

Marine Le Morvan, Nicolas Prost, Julie Josse, Erwan Scornet, and Gael Varoquaux. Linear predictor on linearly-generated data with missing values: non consistency and solutions. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 3165–3174. PMLR, 26–28 Aug 2020c.

Marine Le Morvan, Julie Josse, Erwan Scornet, and Gaël Varoquaux. What's a good imputation to predict with missing values? *Advances in Neural Information Processing Systems*, 34:11530–11540, 2021a.

Marine Le Morvan, Julie Josse, Erwan Scornet, and Gael Varoquaux. What's a good imputation to predict with missing values? In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 11530–11540. Curran Associates, Inc., 2021b.

Roderick JA Little and Donald B Rubin. *Statistical analysis with missing data*, volume 793. John Wiley & Sons, 2019.

Vincent Margot and George Luta. A new method to compare the interpretability of rule-based algorithms. *AI*, 2(4):621–635, 2021.

Imke Mayer, Aude Sportisse, Julie Josse, Nicholas Tierney, and Nathalie Vialaneix. R-miss-tastic: a unified platform for missing values methods and workflows, 2019.

Nicolai Meinshausen. Node harvest. *The Annals of Applied Statistics*, pages 2049–2072, 2010.

Samaneh A Mofrad, Astri J Lundervold, Alexandra Vik, and Alexander S Lundervold. Cognitive and MRI trajectories for prediction of Alzheimer's disease. *Scientific Reports*, 11(1):1–10, 2021.

Christoph Molnar. *Interpretable Machine Learning*. https://christophm.github.io/interpretable-ml-book, 2 edition, 2022.

Alfredo Nazabal, Pablo M Olmos, Zoubin Ghahramani, and Isabel Valera. Handling incomplete heterogeneous data using vaes. *Pattern Recognition*, 107:107501, 2020.

Michael Oberst, Fredrik Johansson, Dennis Wei, Tian Gao, Gabriel Brat, David Sontag, and Kush Varshney. Characterization of overlap in observational studies. In *International Conference on Artificial Intelligence and Statistics*, pages 788–798. PMLR, 2020.

World Health Organization et al. Ghe: Life expectancy and healthy life expectancy. *The Global Health Observatory [Internet].[cited 26 Aug 2022].*, 2021.

Alma B Pedersen, Ellen M Mikkelsen, Deirdre Cronin-Fenton, Nickolaj R Kristensen, Tra My Pham, Lars Pedersen, and Irene Petersen. Missing data and multiple imputation in clinical epidemiological research. *Clinical epidemiology*, pages 157–166, 2017.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830, 2011a.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830, 2011b.

Max Roser, Esteban Ortiz-Ospina, and Hannah Ritchie. Life expectancy. *Our World in Data*, 2013. https://ourworldindata.org/life-expectancy.

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

Donald B Rubin. An overview of multiple imputation. In *Proceedings of the survey research methods section of the American statistical association*, volume 79, page 84. Citeseer, 1988.

Derek D Rucker, Blakeley B McShane, and Kristopher J Preacher. A researcher's guide to regression, discretization, and median splits of continuous variables. *Journal of Consumer Psychology*, 25(4):666–678, 2015.

Lena Stempfle, Ashkan Panahi, and Fredrik D Johansson. Sharing pattern submodels for prediction with missing values. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37–8, pages 9882–9890, 2023.

Bheki ETH Twala, MC Jones, and David J Hand. Good methods for coping with missing data in decision trees. *Pattern Recognition Letters*, 29(7):950–956, 2008.

Berk Ustun and Cynthia Rudin. Learning optimized risk scores. *Journal of Machine Learning Research (JMLR)*, 20(150):1–75, 2019.

Stef Van Buuren. *Flexible Imputation of Missing Data (2nd ed.)*. Chapman and Hall/CRC, Boca Raton, FL, 2018.

Geoffrey I Webb, Eamonn Keogh, and Risto Miikkulainen. Naïve bayes. *Encyclopedia of machine learning*, 15(1):713–714, 2010.

Dennis Wei, Sanjeeb Dash, Tian Gao, and Oktay Gunluk. Generalized linear rule models. In *International Conference on Machine Learning*, pages 6687–6696. PMLR, 2019.

Michael W. Weiner, Paul S. Aisen, Clifford R. Jack Jr., William J. Jagust, John Q. Trojanowski, Leslie Shaw, Andrew J. Saykin, John C. Morris, Nigel Cairns, Laurel A. Beckett, Arthur Toga, Robert Green, Sarah Walter, Holly Soares, Peter Snyder, Eric Siemers, William Potter, Patricia E. Cole, Mark Schmidt, and Alzheimer's Disease Neuroimaging Initiative. The alzheimer's disease neuroimaging initiative: Progress report and future plans. *Alzheimer's & Dementia*, 6(3):202–211.e7, 2010.

Checklist

Please do not modify the questions. Note that the Checklist section does not count towards the page limit. Not including the checklist in the first submission won't result in desk rejection, although in such case we will ask you to upload it during the author response period and include it in camera ready (if accepted).

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes, the setting is described in Section 2 and the algorithm in Section 3.]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [No, we have not performed a formal time/space complexity analysis as this is not relevant to our contributions. We comment very briefly on the complexity of our beam search column generation in Section 3]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes, we have included a reference to a repository]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes/No/Not Applicable]
   (b) Complete proofs of all theoretical results. [Yes. A proof of proposition 1 is included in the Appendix.]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Code to run the algorithm is included in the supplement. The data is either public (and can be found in links provided) or not permitted to be shared (ADNI).]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes, splits are determined by the random seeds in the code. Their sizes are described. Hyperparameters are described as well.]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes. We used up to 50 non-GPU compute nodes with 2x Intel Xeon Gold 6130 CPUs and used around 25000 CPU hours.]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Yes]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable. The ADNI consortium collected all the human subjects data and informed them in this process.]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable. No additional IRB approval was needed beyond the approval for the original ADNI data collection.]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Appendix

A Additional experimental details

A.1 Baseline models

The baselines are trained by the following parameters. The best values for these hyperparameters are chosen based on the validation test set.

`LASSO`: The values of alpha indicating a ℓ_1 regularization term on weights range within $[0.01, 0.4]$, where increasing this value will make model more conservative. We allow to fit an intercept and set the precompute parameter to *TRUE* to get the precomputed Gram matrix to speed up calculations (Buitinck et al., 2013). `LASSO` is trained with zero and MICE imputation and chosen based on the validation performance.

`XGB`: In `XGB` we range the learning rate (η) between $[0.001, 0.01, 0.1, 0.2, 0.3, 0.5]$ where the shrinking step size is used in the update to prevent overfitting. After each boosting step, we can get the weights of new features directly, and η shrinks the feature weights to make the boosting process more conservative. The maximum depth of the trees is from $[4, 6, 10]$ while increasing this value will make the model more complex and more likely to overfit (Chen and Guestrin, 2016). The hyperparameters λ represent the $\ell 2$ regularization term on weights and α indicates the $\ell 1$ regularization term. We choose λ from $[0.01, 0.1, 0.5, 1]$ and α values between $[0, 0.1, 0.2, 0.3]$. Increasing this value will make a model more conservative. `XGB` does not rely on imputation and chooses a default direction for missing values learned during training.

`DT`: For `DT` we set the criterion to measure the quality of a split using the 'squared error' and used 'best' as the strategy to choose the split at each node. The minimum number of samples per leaf can range between [10, 20, 50]. A node will be split if this split induces a decrease of the impurity greater than or equal to 0.1. Complexity parameter 'ccp alpha' is used for Minimal Cost-Complexity Pruning where the subtree with the largest cost complexity that is smaller than 0.005 will be chosen (Buitinck et al., 2013). We use zero imputation for all `DT`s.

`NEUMISS`: For `NEUMISS` models we define the dimension of inputs and outputs of the NeuMiss block (n-features), choose the number of layers (Neumann iterations) in the NeuMiss block (depth) between $[2, 3, 4, 5, 6, 7, 8, 10]$ and range the number of hidden layers in the MLP (mlp depth) between [3,5,6,7,9,10] and set the width of the MLP (mlp width) to the number of covariates for each data set (Le Morvan et al., 2020a).

`RULEFIT` The `RULEFIT` algorithm, proposed by (Friedman and Popescu, 2008), blends tree-based decisions and linear modeling to predict outcomes from input data. It starts by generating a tree ensemble through gradient boosting and then converts the decision paths into binary rules reflecting input feature influences. These rules, along with input variables, are included in a Lasso linear model, which evaluates rule impacts on the target variable and applies L1-regularization to simplify the model by reducing many coefficients to zero, enhancing interpretability. Used hyperparameters were the maximum number of rules between [7, 15, 20, 30, 100] indicating the total number of terms included in the final model (both linear and rules) and the tree size varying between [5, 10, 15] considering interpretablity objectives. We also allowed for *lin standardise=True*, indicating the linear terms will be standardized by multiplying the winsorized variable by 0.4/standard deviation and *exp rand tree size=True* we set that each boosted tree will have a different maximum number of terminal nodes based on an exponential distribution of tree size.

B Real-world data sets

ADNI The data is obtained from the publicly available Alzheimer's Disease Neuroimaging Initiative (ADNI) database. ADNI collects clinical data, neuroimaging, and genetic data (Weiner et al., 2010). The regression task aims to predict the outcome of the ADAS13 (Alzheimer's Disease Assessment Scale) (Mofrad et al., 2021) cognitive test at a 2-year follow-up based on available data at baseline.

Life The data set related to *life* expectancy, has been collected from the WHO data repository website(Organization et al., 2021), and its corresponding economic data was collected from the United Nations website. The data can be publicly accessed trough (Roser et al., 2013). In a regression task, we aim to predict the life expectancy in years from 193 countries considering data from the years 2000-2025. The final dataset consists of 20 columns and 2864 samples where all predicting variables were then divided into several broad categories: immunization factors, mortality factors, economic factors, and social factors.

Housing The Ames *housing* data set was obtained from (http://www.kaggle.com) and describes the selling price of individual properties, various features, and details of each home in Ames, Iowa, USA from 2006 to 2010 (De Cock, 2011). We selected 51 variables on the quality and quantity of physical attributes of a property such as measurements of area dimensions for each observation, including the sizes of lots, rooms, porches, and garages or some geographical categorical features related to profiling properties and the neighborhood. In a regression task, we used 1460 observations.

C Additional results

Table 4: Performance results for Synthetic data of 500 samples and 15 covariates over 10 seeds using Gurobi or beam-search as a solver for the optimization. $\lambda_0 = 0.01$ and $\lambda_1 = 0.01$ were chosen.

	Synthetic (MNAR), *ILP*			Synthetic (MNAR), *beam search*		
Model	R^2	**MSE**	$\bar{\rho}$	R^2	**MSE**	$\bar{\rho}$
`MINTY`$_{\gamma=0}$	0.72 (0.20)	1.32 (0.31)	0.36	0.73 (0.2)	1.30 (0.32)	0.36
`MINTY`$_{\gamma=0.01}$	0.72 (0.21)	1.32 (0.32)	0.34	0.73 (0.2)	1.29 (0.39)	0.26
`MINTY`$_{\gamma=10000}$	-0.00 (0.20)	4.71 (0.31)	0.03	-0.01 (0.2)	4.74 (0.63)	0.00

We show in Table 4 the comparison between the optimal solution found by the Gurobi (Gurobi Optimization, LLC, 2023) solver (left in table), and the approximate solutions using a heuristic beam search algorithm. We see that when using beam-search, we achieve almost the same results as with Gurobi.

Complexity vs. predictiveness Results are shown in Figure 3, comparing the R^2s with estimator-specific complexity measurements. We observe that `MINTY`$_{\gamma=0.1}$ balances the trade-off between good predictive performance with a small number of non-zero coefficients which in turn ensures lower model complexity (15 coefficients). One reason why `MINTY`$_{\gamma=0.10}$ performs better than `MINTY`$_{\gamma=0}$ (essentially zero-imputation) is that it can choose from a bigger set of rules. However, this also increases the reliance on imputed values and some level of bias in the model. `NEUMISS` which shows the lowest complexity, however, depends on imputation, and cannot be interpreted due to its black-box nature. Similary for `DT`, which performs the best on the ADNI data but perhaps lacks some interpretability with almost 40 numbers of leaves. In a `DT`, neighboring leaves are similar to each other as they share the path in the tree. As the number of leaves increases, variance in the performance increases and perhaps compromises interpretability. `XGB` achieves consistent performance across estimators, but could be difficult to interpret with a larger number of estimators (and an even larger number of parameters). While `LASSO` is the simplest model, its performance is the lowest.

Customized Rules We use simulated data X_{sim} by sampling $n \times d$ independent binary input features. However, we add some conditional dependence between columns 0 and 4 to illustrate the process of generating replacement variables focusing on predictive performance and interpretability. Each element of X_{sim} is randomly set to 0 or 1 based on whether a random value drawn from a standard normal distribution is greater than 0. The outcome Y is based on the values in columns 0 and 4 of X_{sim}, adding a constant term of 1 and some random noise drawn from a standard normal distribution.

Table 5: Performance results for synthetic data with 10 iterations and 7000 samples and 15 columns. The missingness proposition of 0.1 together with 0.1 for replacement disagreement probability, as described in 7 for three different missingness mechanisms.

	Synthetic data								
	MAR, $\lambda_0 = 0.001$, $\lambda_1 = 0.01$			MNAR, $\lambda_0 = 0.01$, $\lambda_1 = 0.01$			MCAR, $\lambda_0 = 0.01$, $\lambda_1 = 0.01$		
Model	R^2	**MSE**	$\bar{\rho}$	R^2	**MSE**	$\bar{\rho}$	R^2	**MSE**	$\bar{\rho}$
$\texttt{LASSO}_{I_0}$	0.55 (0.04)	3.81 (0.23)	0.65	0.51 (0.17)	4.13 (0.23)	0.70	0.47 (0.04)	4.64 (0.21)	0.82
$\texttt{DT}_{I_0}$	0.43 (0.05)	4.94 (0.23)	0.23	0.41 (0.58)	5.18 (0.17)	0.29	0.36 (0.05)	5.60 (0.22)	0.44
XGB	0.80 (0.03)	1.64 (0.10)	0.92	0.79 (0.35)	1.76 (0.18)	0.96	0.76 (0.02)	2.04 (0.12)	0.99
RULEFIT	0.60 (0.03)	1.97 (0.06)	0.44	0.53 (0.03)	2.13 (0.07)	0.58	0.61 (0.03)	1.94 (0.08)	0.52
NEUMISS	0.81 (0.03)	1.52 (0.12)	0.92	0.75 (0.46)	1.97 (0.11)	0.96	0.75 (0.03)	2.07 (0.23)	0.99
$\texttt{MINTY}_{\gamma=0}$	0.69 (0.04)	2.69 (0.18)	0.51	0.67 (0.41)	2.69 (0.24)	0.64	0.66 (0.03)	2.94 (0.24)	0.81
$\texttt{MINTY}_{\gamma=0.01}$	0.69 (0.03)	2.69 (0.21)	0.49	0.66 (0.35)	2.89 (0.25)	0.64	0.66 (0.04)	2.99 (0.24)	0.80
$\texttt{MINTY}_{\gamma=10000}$	0.25 (0.05)	6.60 (0.27)	0.00	-0.00 (-0.1)	8.81 (0.61)	0.00	-0.00 (0.06)	8.98 (0.26)	0.00

Table 6: Customized rule sets for predictions using ADNI data using $\gamma = 0$ (top) and $\gamma = 0.01$ (bottom). The R^2 for the two models were .64 and .63 respectively, but the latter had significantly smaller reliance on features with missing values ($\bar{\rho} = 0.28$ vs $\bar{\rho} = 0.40$). The red rules in the top model are not present in the bottom and have larger missingness in the data. The blue rules in the bottom model are not present in the top and have less missingness.

Learned Rules by $\texttt{MINTY}_{\gamma=0.1}$	Coeff.
LDELTOTAL $\in [8 - 12]$ OR LDELTOTAL ≥ 12 OR Cognitive normal diagnosis	-0.81
LDELTOTAL ≤ 3.0 OR LDELTOTAL $\in [8 - 12]$ OR Alzheimer's diagnosis	+0.42
LDELTOTAL ≥ 12 OR MMSE ≤ 26	+0.13
AGE ≥ 78.5 OR MMSE $\in [26 - 28]$ OR Alzheimer's diagnosis	+0.35
MMSE ≤ 26 OR SEX = Male	+0.17
AGE $\in [73.5 - 78.5]$ OR APOE4= 2.0 OR Alzheimer's diagnosis	+0.25
68.9 AGE $\in [68.9 - 73.5]$ OR MMSE $\in [26 - 28]$ OR Alzheimer's diagnosis	+0.22
MMSE $\in [26 - 28]$ OR MMSE ≥ 29.0 OR Race=Black	-0.22
LDELTOTAL ≤ 3.0 OR APOE4 = 1.0	+0.14
MMSE $\in [26 - 28]$ OR MMSE $\in [28 - 29]$ OR Cognitive normal diagnosis OR EMCI diagnosis	-0.15
Intercept	-0.09

In Table 7, we compare a set of learned rules (right Table) to the ground truth rules (left Table) from generated data. We interpret the results by saying that the model perfectly produces the correct rules, e.g. variable 1 and variable 4. Moreover, the coefficients and intercept are also identical if rounded.

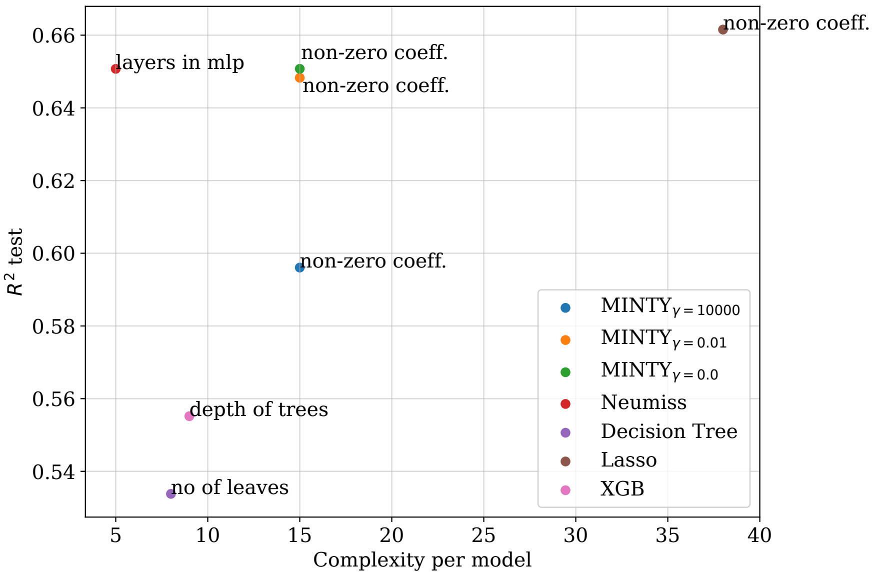

Figure 3: Performance against complexity measurement on *ADNI* data. As a criterion for complexity, we use for `MINTY` models and `LASSO` the number of non-zero coefficients achieved by regularisation. `NEUMISS` does not aim at a sparse solution and therefore we give the complexity by the number of layers in the MLP network. Note, that there might be more parameters to optimize for. The depth of the trees defines the complexity for `XGB`, and for `DT` we describe the number of leaves.

Table 7: Customized rule sets for predictions based on the ground true rule set (Top table). Learned rules set with corresponding coefficients in the bottom table are based on `MINTY`. The results are based on a generated data set with $n = 7000$ samples and a $p_{miss} = 0.1$

True Rules	Coeff.
Variable 1 OR Variable 4	2
Intercept	+1

Learned Rules	Coeff.
Variable 1 OR Variable 4	1.63
Intercept	+1.14

D Proof of Proposition 1

Proposition 1. *Assume that an outcome Y is linear in X with noise of bounded conditional variance,*

$$Y = \beta^\top X + \epsilon(X), \text{ where } \mathbb{E}[\epsilon \mid X] = 0, \mathbb{M}[\epsilon \mid X] \leq \sigma^2 \ ,$$

with $\beta \in \mathbb{R}^d$ and $X \in \{0,1\}^d$ a multivariate binary variable with the following structure. For each X_i there is a paired "replacement" variable $X_{j(i)}$, with $j(j(i)) = i$, such that for $\delta \geq 0$, $p(X_i = X_{j(i)}) \geq 1 - \delta$, and that whenever X_i is missing, $X_{j(i)}$ is observed, $M_i = 1 \Rightarrow M_{j(i)} = 0$. Assume also that $\forall i, k \notin \{i, j(i)\} : X_i \perp\!\!\!\perp X_k$. Then, there is a GLRM h with two-variable rules $\{\bar{X}_i \vee \bar{X}_{j(i)}\}_{i=1}^d$, where $\bar{X}_i = (1 - M_i)X_i$, with risk

$$R(h) \leq \delta\|\beta\|_2^2 + \delta^2 \sum_{i,k \notin \{i,j(i)\}} |\beta_i \beta_k| + \sigma^2 \ .$$

under the squared error. Additionally, if $\beta_i \geq 0$ and $\mathbb{E}[X_i M_i] \geq \eta$ for all $i \in [d]$, using the ground truth β with zero-imputed features $\bar{X}$ yields a risk bounded from below as

$$R(\beta) \geq \eta\|\beta\|_2^2 + \sigma^2 \ ,$$

and a greater missingness reliance than the GLRM, $\bar{\rho}(\beta) \geq \bar{\rho}(h)$.

Proof. Let $\mu(X) = \mathbb{E}[Y \mid X]$. The risk of any hypothesis $h(X)$ can be decomposed as

$$R(h) = \mathbb{E}[L(h(X), Y)] = \mathbb{E}[(h(X) - Y)^2] = \mathbb{E}[(h(X) - \mu(X))^2] + \underbrace{\mathbb{E}[\epsilon^2]}_{\leq \sigma^2} \ .$$

Now, consider a GRLM h where each variable pair $i, j(i)$ is represented by a rule $(\bar{X}_i \vee \bar{X}_{j(i)})$, used in place of X_i and X_j in a linear model, and a coefficient $\tilde{\beta}_i = \beta_i + \beta_{j(i)}$. Then, for each i, define the bias variable

$$\Delta_i = (\bar{X}_i \vee \bar{X}_{j(i)}) - X_i = \begin{cases} 1, & \text{if } \bar{X}_{j(i)} = 1 \wedge X_i = 0 \\ 0, & \text{otherwise} \end{cases} \ .$$

In other words, bias is introduced, $\Delta_i = 1$, only if the zero-imputed replacement $\bar{X}_{j(i)}$ is 1 but X_i is 0. $\bar{X}_{j(i)}$ is only equal to 1 if $j(i)$ is observed. Thus, $\mathbb{E}[\Delta_i] = p(X_{j(i)} = 1, X_i = 0) \leq \delta$, by assumption. As a result,

$$\begin{aligned}
\mathbb{E}[(h(X) - \mu(X))^2] &= \mathbb{E}\left[\left(\sum_{i=1}^d (\beta_i(\bar{X}_i \vee \bar{X}_{j(i)}) - \beta_i X_i)\right)^2\right] \\
&= \mathbb{E}\left[\sum_{i,j=1}^d \beta_i \beta_j \Delta_i \Delta_j\right] \\
&= \sum_{i,j=1}^d \mathbb{E}[\beta_i \beta_j \Delta_i \Delta_j] \\
&= \sum_{i=1}^d \left(\mathbb{E}[\beta_i^2 \Delta_i^2] + \mathbb{E}[\beta_i \beta_{j(i)} \underbrace{\Delta_i \Delta_{j(i)}}_{=0}] + \sum_{k \notin \{i,j(i)\}} \mathbb{E}[\beta_i \beta_k \Delta_i \Delta_k]\right) \\
&= \sum_{i=1}^d \left(\mathbb{E}[\beta_i^2 \Delta_i] + \sum_{k \notin \{i,j(i)\}} \mathbb{E}[\beta_i \Delta_i]\mathbb{E}[\beta_k \Delta_k]\right) \\
&\leq \sum_{i=1}^d \left(\beta_i^2 \mathbb{E}[\Delta_i] + \sum_{k \notin \{i,j(i)\}} \beta_i \beta_k \mathbb{E}[\Delta_i]\mathbb{E}[\Delta_k]\right) && \text{By independence, } X_i \perp\!\!\!\perp X_k \\
&\leq \delta\|\beta\|^2 + \delta^2 \sum_{k \notin \{i,j(i)\}} |\beta_i \beta_k| \ .
\end{aligned}$$

We can generalize the result by placing a bound on the cross-moment of the replacement bias $\mathbb{E}[\Delta_i \Delta_k]$, rather than assuming that $X_i \perp\!\!\!\perp X_k$.

There is also a lower bound for the ground-truth model applied to zero-imputed data with missingness. Its bias is

$$B = \mathbb{E}[(\beta^\top X - \beta^\top \bar{X})^2]) = \mathbb{E}[(\beta^\top (M \odot X))^2]$$

If all coefficiencts are positive, $\beta \in \mathbb{R}^d_+$, and hence all terms in the bias,

$$B \geq \sum_{i=1}^{d} \mathbb{E}[(\beta_i M_i X_i)^2] = \sum_{i=1}^{d} \beta_i^2 \mathbb{E}[M_i X_i]$$

By the assumption that $\mathbb{E}[M_i X_i] \geq \eta$ for some $\eta > 0$, it follows that

$$B \geq \eta \|\beta\|_2^2 \ .$$

The reliance on features with missing values $\bar{\rho}(h)$ of the GLRM h is determined by events where a replacement variable $j(i)$ has the value 0 when the variable i is unobserved, $\exists i : \mathbb{1}[M_i = 1, X_{j(i)} = 0)]$. If this is true for any i, $\rho = 1$. For the ground-truth model, it is sufficient that a variable is missing, $\exists i : \mathbb{1}[M_i = 1]$. Hence, the expected reliance on features with missing values is greater for $\beta^\top \bar{X}$ than for h.

In conclusion, the GLRM is preferred whenever

$$\delta \|\beta\|^2 + \delta^2 \sum_{i,k \notin \{i,j(i)\}} |\beta_i \beta_k| < \eta \|\beta\|^2 \ .$$

Letting $a = \|\beta\|^2 / (\sum_{i,k \notin \{i,j(i)\}} |\beta_i \beta_k|)$ and solving for δ, we get

$$\delta < (\sqrt{a^2 + 4\eta} - a)/2 \ .$$

□