Contents

1 Linear Dependence Model

The linear data generation process simulates mixed continuous and categorical variables via a latent multivariate normal factor model. Given parameters

$$n, \quad p_{\text{cont}} = \lfloor p/2 \rfloor, \quad p_{\text{cat}} = p - p_{\text{cont}}, \quad d = \max\bigl(p_{\text{cont}}, \texttt{max\_levels}\bigr),$$

and `max_levels` the maximum categories per variable, we proceed:

1. Generate a random $d \times d$ matrix M with entries

$$M_{ij} \sim \mathcal{U}(0.05, 0.95) \quad \text{and set} \quad \Sigma = M\,M^\top$$

   to ensure a positive–definite covariance.

2. Draw latent factors

$$\mathbf{Z} \in \mathbb{R}^{n \times d} \;\sim\; \mathcal{N}\bigl(\mathbf{0}, \Sigma\bigr).$$

3. Split into *continuous* and *categorical* blocks:

Continuous: Select p_{cont} distinct columns of $\mathbf{Z}$ at random and label them $\{X_1, \ldots, X_{p_{\text{cont}}}\}$.

Categorical: For each $j = 1, \ldots, p_{\text{cat}}$:

(a) Sample the number of categories $K_j \in \{2, \ldots, \texttt{max\_levels}\}$.

(b) Choose $s = K_j - 1$ columns of $\mathbf{Z}$ as predictors, forming $\mathbf{Z}_{\text{pred}} \in \mathbb{R}^{n \times s}$.

(c) Build a coefficient matrix $B \in \mathbb{R}^{s \times K_j}$ by

$$B \;=\; \bigl[\,\mathbf{0} \bigm| \beta\bigr], \quad \beta_{ik} \sim \mathcal{U}(-3, 3),$$

where the first column of zeros is the reference.

(d) Add independent noise $\boldsymbol{\varepsilon} \in \mathbb{R}^{n \times K_j}$, $\varepsilon_{ik} \sim \mathcal{N}(0, 1)$.

(e) Compute latent utilities

$$Y \;=\; \mathbf{Z}_{\text{pred}}\,B \;+\; \boldsymbol{\varepsilon} \;\in\; \mathbb{R}^{n \times K_j}.$$

(f) Assign each observation

$$C_j(i) \;=\; \arg\max_{k}\; Y_{ik}.$$

(g) If any category has fewer than `min_obs` observations, reassign those cases by choosing the next-largest utility among the remaining valid categories. Finally, relabel categories to consecutive integers $1, \dots, K'_j$.

Return a data frame containing $\{X_1, \dots, X_{p_{\text{cont}}}\}$ as numeric and $\{C_1, \dots, C_{p_{\text{cat}}}\}$ as factors.

2 Hierarchical Tree-Based Dependence Model

The hierarchical generator alternates continuous and categorical features by recursive binary splitting on all previously generated variables.
Given bounds $[a, b]$ for the continuous features and parameters `max_depth` $= 5$, `min_split` $= 200$, `min_bucket` $= 50$, the procedure for the generation of the total set of variables $\{V_1, \dots, V_p\}$ is:

1. *Initialize* $X_1 \sim \mathcal{U}(a, b)$ independently.

2. For $j = 2, \dots, p$:

(a) *Build a binary tree* on the index set $\{1, \dots, n\}$ using all previously generated variables $\{V_1, \dots, V_{j-1}\}$ as predictors. Call this `build_tree`($\{1, \dots, n\}$, $\text{data}_{1:(j-1)}$), which:

- Stops splitting a node if its size $<$ `min_split` or if its depth $>$ `max_depth`.
- Attempts up to 5 random splits per node:
  - If the chosen predictor is continuous, pick a split threshold at a random truncated-normal quantile of the node's values. The random quantile q is drawn from $\mathcal{N}(0.5, 0.2)$ and then truncated to lie in $[0.1, 0.9]$, ensuring the split always falls between the 10^{th} and 90^{th} percentiles to reduce the likelihood of creating nodes with a sparse observation count.
  - If it is categorical, split by a random nonempty proper subset of its levels.
- Accepts a split only if both children have $\geq$ `min_bucket` observations; otherwise it retries or makes the node a leaf if it attempted the split five times.

(b) **Continuous** (j odd:) Let the resulting tree's leaves correspond to intervals $[L_i, U_i]$. The intervals bounds are specified by the previous hierarchy. The base intervals starts with $[0, 10]$ at the tree's root and then every split parts the previous node's interval in two equally sized new intervals. For each observation t in leaf i, sample

$$X_k(t) \;\sim\; \mathcal{N}\big(\tfrac{L_i+U_i}{2},\; \tfrac{U_i-L_i}{8}\big)\,.$$

(c) **Categorical** (j even:) Collect all L leaves and choose a number of categories $K \sim \{2, \ldots, 7\}$. Assign each leaf one of the K labels, ensuring every label is used at least once. Then for each t in leaf i:

$$C_k(t) = \begin{cases} \ell(i), & \text{with probability } 0.8, \\ \text{a different label in } [1..K] \setminus \{\ell(i)\}, & \text{with probability } 0.2. \end{cases}$$

If after all leafs are assigned to the predefined categories, any category ends up with < 20 observations, the entire assignment is retried; if still invalid, the function errors.

3. Return $\{X_1, \ldots, X_{\lceil p/2 \rceil}\}$ as numeric columns and $\{C_1, \ldots, C_{\lfloor p/2 \rfloor}\}$ as factors.

This yields a sequence of features where each is conditioned on a random tree built from all previous ones, producing rich, hierarchical dependencies between continuous and categorical variables.

3 Simulation Grid

The complete dataset variations resulting for varying $type \in \{linear, hierarchical\}$, $p \in \{6, 12, 18\}$ and $seed \in \{1, 2, 3, 4, 5\}$

$$2 \text{ (types)} \times 3 \ (p) \times 5 \text{ (seeds)} = 30 \text{ simulated datasets.}$$