Package 'fairpolicytree'

May 19, 2025

Title Fair and Interpretable Policy Learning with Decision Trees

Version 0.1.0

Description Extends the 'policytree' package by integrating fairness into algorithmic decision-making using policy trees, a class of interpretable decision rules. Useful for applications where both fairness and interpretability are essential, such as treatment assignment in public policy contexts.

Depends R (>= 3.5.0)

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Contents

---

mq_adjustment   *Marginal Quantile-Adjustment for Fair and Interpretable Policy Learning*

Description

Computes fairness-adjusted variables by MQ-adjustment

Usage

```
mq_adjustment(vars, sens, seed = 123456, ties.method = "random")
```

Arguments

vars	Numeric matrix or data.frame of variables to be adjusted (observations in rows, variables in columns).
sens	Matrix or data.frame of sensitive attributes. Must have the same number of rows as 'vars'.
seed	Integer scalar for reproducible random tie-breaking.
ties.method	Character string for ranking ties. One of "random", "average", "first", "last", "max", "min".

Value

A list with two data.frames:

vars_cdf CDF-adjusted variables denoted with suffix '_cdf'.

vars_mq MQ-adjusted variables denoted with suffix '_mq'.

---

plot.prob_split_tree   *Plot a Probabilistic Split Tree.*

Description

Plot a Probabilistic Split Tree.

Usage

```
## S3 method for class 'prob_split_tree'
plot(tree, leaf.labels = NULL)
```

Arguments

tree	A fitted object of class ''prob_split_tree''.
leaf.labels	An optional character vector of leaf labels for each treatment.

---

plot.prob_split_tree_list
                    Plot a List of Probabilistic Split Trees.

Description

Plot a List of Probabilistic Split Trees.

Usage

```
## S3 method for class 'prob_split_tree_list'
plot(tree_list, sens_names = NULL, leaf.labels = NULL)
```

Arguments

sens_names	The variables names of the sensitive attributes.
leaf.labels	An optional character vector of leaf labels for each treatment.
tree	A fitted object of class ''prob_split_tree_list''.

---

predict.prob_split_tree

Predict from a Probabilistic Split Tree

Description

Predict from a Probabilistic Split Tree

Usage

```
## S3 method for class 'prob_split_tree'
predict(
  tree,
  vars,
  tree_cdf = NULL,
  vars_cdf = NULL,
  type = "action.id",
  seed = 123456
)
```

Arguments

tree	A fitted object of class ''prob_split_tree'‘.
vars	A data.frame or matrix of the decision variables.
tree_cdf	Optional. A corresponding 'policy_tree' object fitted on CDF-adjusted decision variables.
vars_cdf	Optional. A data.frame or matrix of CDF-adjusted decision variables (same structure as 'vars').
type	Character. One of '"action.id"' (default) or '"leaf.id"'. Determines the type of prediction returned.
seed	Integer. Random seed used to resolve probabilistic splits when exact info is missing. Default is 123456.

Value

A numeric vector of predicted action or leaf IDs for each observation.

---

predict.prob_split_tree_list

Predict from a List of Probabilistic Split Trees

Description

Makes group-specific predictions using a list of probabilistic policy trees, each fitted for a unique combination of sensitive attributes.

Usage

```
## S3 method for class 'prob_split_tree_list'
predict(tree_list, A, sens, type = "action.id", seed = 123456)
```

Arguments

tree_list	A named list of fitted probabilistic split trees (e.g., from [prob_split_tree()]), where names are underscore-separated group identifiers (e.g., '"0_1"').
A	A matrix or data.frame of decision variables.
sens	A data.frame or matrix of sensitive attributes used for fairness adjustment. (must match the naming in 'tree_list').
type	Character. One of '"action.id"' (default) or '"leaf.id"'. Determines the type of prediction returned.
seed	Integer. Random seed used to resolve probabilistic splits when exact info is missing. Default is 123456.

Value

A vector of predicted actions or leaf IDs, one per row of 'A'.

---

print.prob_split_tree   *Print a Probabilistic Split Tree.*

Description

Print a Probabilistic Split Tree.

Usage

```
## S3 method for class 'prob_split_tree'
print(tree)
```

Arguments

tree	A fitted object of class ''prob_split_tree''.

---

print.prob_split_tree_list
                    Print a List of Probabilistic Split Trees.

Description

Print a List of Probabilistic Split Trees.

Usage

```
## S3 method for class 'prob_split_tree_list'
print(tree_list, sens_names = NULL)
```

Arguments

sens_names	The variables names of the sensitive attributes.
tree	A fitted object of class ''prob_split_tree_list''.

---

prob_split_tree	*Fit a Fair Probabilistic Split Tree*

Description

This function performs a cdf-fairness adjustment on decision variables, and optionally on policy score variables. It then fits a policy tree using the 'policytree' package, and adjusts split thresholds for each sensitive group to produce probabilistic split trees.

Usage

```
prob_split_tree(
  A,
  scores,
  sens,
  adjust_scores = FALSE,
  seed = 123456,
  ties.method = "random",
  depth = 2,
  search.depth = depth,
  split.step = 1,
  min.node.size = 1,
  verbose = TRUE
)
```

Arguments

A	A matrix or data.frame of decision variables.
scores	A data.frame or matrix of policy score variables (one column per treatment option).
sens	A data.frame or matrix of sensitive attributes used for fairness adjustment.
adjust_scores	Logical. Whether to apply fairness adjustment to 'scores' as well. Default is 'FALSE'.
seed	Integer seed for reproducibility. Default is 123456.
ties.method	Character string for ranking ties. One of "random", "average", "first", "last", "max", "min".
depth	Integer. Maximum depth of the output policy tree. Passed to 'policytree::policy_tree'.
search.depth	Integer. Only used if greater than 'depth'. If so, hybrid tree search is applied using 'policytree::hybrid_policy_tree'. Default is equal to 'depth'.
split.step	An optional approximation parameter, the number of possible splits to consider when performing tree search. split.step = 1 (default) considers every possible split, 'split.step = 10' considers splitting at every 10'th sample and may yield a substantial speedup for dense features. Manually rounding or re-encoding continuous covariates with very high cardinality in a problem specific manner allows for finer-grained control of the accuracy/runtime tradeoff and may in some cases be the preferred approach..

| `min.node.size` | An integer indicating the smallest terminal node size permitted. Default is 1. |
| `verbose` | Logical. Give verbose output. Default is 'TRUE'. |

Value

A list of probabilistic split trees, one per sensitive group.

`simulate_fairness_data`

Simulate Artificial Fairness Data

Description

Generates artificial data with: - Two binary sensitive attributes (correlated) - One binary and one continuous decision variable (both correlated with sensitive attributes) - Two continuous policy score variables (correlated with all other variables)

Usage

```
simulate_fairness_data(n = 1000, seed = 123456)
```

Arguments

| `n` | Integer. Number of observations to generate. Default is 1000. |
| `seed` | Integer. Random seed for reproducibility. Default is 123456. |

Value

A list with three data.frames:

sens Data frame with two binary sensitive attributes.

decision Data frame with one binary and one continuous decision variable.

scores Data frame with two continuous policy score variables.

Index