Package: PTT (via r-universe)

October 21, 2026

Type Package

Title Pólya Tree Type Models

Version 1.0.1

Description Fits Bayesian nonparametric models based on Pólya tree processes, including adaptive Pólya trees, Markov adaptive Pólya trees, optional Pólya trees, and their conditional-density counterparts. Methods are described in Ma (2017) <doi:10.1214/16-BA1021>, Ma (2017) <doi:10.1214/17-EJS1254>, and Wong and Ma (2010) <doi:10.1214/09-AOS755>.

License GPL (>= 3)

URL https://github.com/MaStatLab/PTT

BugReports https://github.com/MaStatLab/PTT/issues

Encoding UTF-8

Imports Rcpp (>= 1.0.0)

LinkingTo Rcpp, RcppArmadillo

NeedsCompilation yes

Config/roxygen2/version 8.0.0

Repository https://mastatlab.r-universe.dev

Date/Publication 2026-07-22 23:56:02 UTC

RemoteUrl https://github.com/mastatlab/ptt

RemoteRef HEAD

RemoteSha a7cd602bb2665b0bc35f96a4d7a95b2a3089084c

Contents

apt 2
cond.apt 3
cond.opt 5
opt 7
plot_partition 8
plot_roc_curve 9

Index 11

`apt` *Fit an Adaptive Pólya Tree*

Description

Fits an adaptive Pólya tree (APT) or Markov APT to univariate or multivariate observations. The partition recursively bisects the sample space, while latent states control local shrinkage toward the uniform centering distribution.

Usage

```
apt(
  X,
  Xpred = NULL,
  Omega.type = "unit",
  max.resol = 10,
  rho0 = 0.2,
  rho0.mode = 0,
  tran.mode = 1,
  lognu.lb = -1,
  lognu.ub = 4,
  n.grid = 5,
  n.s = 5,
  beta = 0.1,
  n.post.samples = 0
)
```

Arguments

`X`	A numeric vector, matrix, or data frame. Rows are observations and columns are dimensions.
`Xpred`	Optional prediction locations. A vector is interpreted as many locations for univariate data and as one location for multivariate data. If ‘NULL‘, no predictive densities are computed.
`Omega.type`	Sample-space construction: ‘"unit"‘ uses the unit hypercube and ‘"standardized"‘ uses a slightly padded empirical range of ‘X‘ and ‘Xpred‘ in each dimension.
`max.resol`	Maximum partition depth. Must be an integer from 1 to 14.
`rho0`	Baseline prior probability of complete shrinkage. Must lie in ‘[0, 1]‘.
`rho0.mode`	How the complete-shrinkage probability changes with level: ‘0‘ keeps it constant; ‘1‘ uses ‘1 - (1 - rho0) / (level + 1)^2‘; and ‘2‘ uses ‘1 - (1 - rho0) / 2^level‘.

`tran.mode`	Shrinkage-state transition model. '0' makes non-stopping states independent across nodes; '1' imposes stochastically increasing shrinkage and treats complete shrinkage separately; '2' applies the increasing transition kernel to all states, including complete shrinkage.
`lognu.lb`, `lognu.ub`	Lower and upper support bounds for 'log10(nu)', where 'nu' is the beta precision (shrinkage) parameter.
`n.grid`	Number of midpoint quadrature points within each shrinkage state.
`n.s`	Number of non-stopping shrinkage states. Must be an integer from 1 to 65,534.
`beta`	Nonnegative stickiness parameter in the exponential transition kernel. 'beta = 0' gives uniform probability over admissible higher states.
`n.post.samples`	Number of posterior partition samples to draw.

Value

A list with 'logrho' (the log posterior complete-shrinkage probability at the root), 'logphi' (the log marginal likelihood), 'part_points_hmap' (the hierarchical MAP partition), 'predictive_densities', and 'Omega'. If posterior samples are requested, 'part_points_post_samples' is also returned. Partition bounds are integer cell indices on the depth-'max.resol' grid; 'Omega' records the physical sample-space bounds.

References

Ma, L. (2017). Adaptive shrinkage in Pólya tree type models. *Bayesian Analysis*, 12(3), 779–805. doi:10.1214/16BA1021.

See Also

[opt()], [cond.apt()]

Examples

```
set.seed(12345)
x <- c(rbeta(40, 3, 8), rbeta(40, 9, 3))
grid <- seq(0.05, 0.95, length.out = 25)
fit <- apt(x, Xpred = grid, max.resol = 4, n.grid = 3, n.s = 3)
head(fit$predictive_densities)
```

---

cond.apt *Fit a Conditional Adaptive Pólya Tree*

Description

Fits the two-stage conditional-density model of Ma (2017): the predictor space is recursively partitioned and an adaptive Pólya tree models the response distribution within each predictor block.

Usage

```
cond.apt(
  X,
  Y,
  Xpred = NULL,
  Ypred = NULL,
  OmegaX.type = "unit",
  OmegaY.type = "unit",
  max.resX = 5,
  max.resY = 8,
  rho0.X = 0.2,
  rho0.mode.X = 0,
  rho0.Y = 0.2,
  rho0.mode.Y = 0,
  tran.mode = 1,
  lognu.lb = -1,
  lognu.ub = 4,
  n.grid = 5,
  n.s = 5,
  beta = 0.1,
  n.post.samples = 0
)
```

Arguments

`X`, `Y`	Numeric vectors, matrices, or data frames containing paired predictor and response observations. They must have the same number of rows.
`Xpred`, `Ypred`	Optional paired predictor and response locations at which to evaluate the conditional predictive density. Supply both or neither; they must have the same number of rows.
`OmegaX.type`, `OmegaY.type`	Sample-space construction for predictors and responses. See 'Omega.type' in [apt()].
`max.resX`, `max.resY`	Maximum partition depths for predictor and response spaces, each between 1 and 14.
`rho0.X`, `rho0.Y`	Baseline prior complete-shrinkage probabilities for the predictor partition and response model.
`rho0.mode.X`, `rho0.mode.Y`	Level-dependent stopping modes for the predictor partition and response model. See 'rho0.mode' in [apt()].
`tran.mode`	Shrinkage-state transition model. '0' makes non-stopping states independent across nodes; '1' imposes stochastically increasing shrinkage and treats complete shrinkage separately; '2' applies the increasing transition kernel to all states, including complete shrinkage.
`lognu.lb`, `lognu.ub`	Lower and upper support bounds for 'log10(nu)', where 'nu' is the beta precision (shrinkage) parameter.

n.grid	Number of midpoint quadrature points within each shrinkage state.
n.s	Number of non-stopping shrinkage states. Must be an integer from 1 to 65,534.
beta	Nonnegative stickiness parameter in the exponential transition kernel. ‘beta = 0‘ gives uniform probability over admissible higher states.
n.post.samples	Number of posterior partition samples to draw.

Value

A list containing ‘logrho‘, ‘logphi‘, the predictor-space ‘part_points_hmap‘, ‘predictive_densities‘, ‘OmegaX‘, and ‘OmegaY‘. If requested, ‘part_points_post_samples‘ contains sampled predictor partitions. Partition bounds use integer cell indices at depth ‘max.resX‘.

References

Ma, L. (2017). Recursive partitioning and multi-scale modeling on conditional densities. *Electronic Journal of Statistics*, 11(1), 1297–1325. doi:10.1214/17EJS1254.

See Also

[cond.opt()], [apt()]

Examples

```
set.seed(12345)
x <- runif(80)
y <- rbeta(80, 2 + 6 * x, 8 - 5 * x)
fit <- cond.apt(
  x, y, Xpred = c(0.25, 0.75), Ypred = c(0.3, 0.7),
  max.resX = 3, max.resY = 4, n.grid = 3, n.s = 3
)
fit$predictive_densities
```

---

cond.opt *Fit a Conditional Optional Pólya Tree*

Description

A convenience wrapper around [cond.apt()] that uses an optional Pólya tree for the response distribution in each predictor block.

Usage

```
cond.opt(
  X,
  Y,
  Xpred = NULL,
  Ypred = NULL,
  OmegaX.type = "unit",
```

```
  OmegaY.type = "unit",
  max.resX = 7,
  max.resY = 7,
  rho0.X = 0.5,
  rho0.mode.X = 0,
  rho0.Y = 0.5,
  rho0.mode.Y = 0,
  n.post.samples = 0
)
```

Arguments

`X`, `Y`	Numeric vectors, matrices, or data frames containing paired predictor and response observations. They must have the same number of rows.
`Xpred`, `Ypred`	Optional paired predictor and response locations at which to evaluate the conditional predictive density. Supply both or neither; they must have the same number of rows.
`OmegaX.type`, `OmegaY.type`	Sample-space construction for predictors and responses. See 'Omega.type' in [apt()].
`max.resX`, `max.resY`	Maximum partition depths for predictor and response spaces, each between 1 and 14.
`rho0.X`, `rho0.Y`	Baseline prior complete-shrinkage probabilities for the predictor partition and response model.
`rho0.mode.X`, `rho0.mode.Y`	Level-dependent stopping modes for the predictor partition and response model. See 'rho0.mode' in [apt()].
`n.post.samples`	Number of posterior partition samples to draw.

Value

A list with the same structure as [cond.apt()].

See Also

[cond.apt()], [opt()]

Examples

```
set.seed(12345)
x <- runif(60)
y <- ifelse(x < 0.5, rbeta(60, 3, 8), rbeta(60, 8, 3))
fit <- cond.opt(
  x, y, Xpred = c(0.25, 0.75), Ypred = c(0.3, 0.7),
  max.resX = 3, max.resY = 4
)
fit$predictive_densities
```

`opt` *Fit an Optional Pólya Tree*

Description

A convenience wrapper around [apt()] that uses one non-stopping state with zero beta precision, reproducing the optional Pólya tree (OPT).

Usage

```
opt(
  X,
  Xpred = NULL,
  Omega.type = "unit",
  max.resol = 10,
  rho0 = 0.5,
  rho0.mode = 0,
  n.post.samples = 0
)
```

Arguments

`X`	A numeric vector, matrix, or data frame. Rows are observations and columns are dimensions.
`Xpred`	Optional prediction locations. A vector is interpreted as many locations for univariate data and as one location for multivariate data. If ‘NULL‘, no predictive densities are computed.
`Omega.type`	Sample-space construction: ‘"unit"‘ uses the unit hypercube and ‘"standardized"‘ uses a slightly padded empirical range of ‘X‘ and ‘Xpred‘ in each dimension.
`max.resol`	Maximum partition depth. Must be an integer from 1 to 14.
`rho0`	Baseline prior probability of complete shrinkage. Must lie in ‘[0, 1]‘.
`rho0.mode`	How the complete-shrinkage probability changes with level: ‘0‘ keeps it constant; ‘1‘ uses ‘1 - (1 - rho0) / (level + 1)^2‘; and ‘2‘ uses ‘1 - (1 - rho0) / 2^level‘.
`n.post.samples`	Number of posterior partition samples to draw.

Value

A list with the same structure as [apt()].

References

Wong, W. H. and Ma, L. (2010). Optional Pólya tree and Bayesian inference. *The Annals of Statistics*, 38(3), 1433–1459. doi:10.1214/09AOS755.

See Also

[apt()], [cond.opt()]

Examples

```
set.seed(12345)
x <- rbeta(60, 4, 7)
fit <- opt(x, Xpred = c(0.25, 0.5, 0.75), max.resol = 4)
fit$predictive_densities
```

`plot_partition` *Plot a Two-Dimensional Partition*

Description

Draws rectangular partition cells with optional density-based fill colors. This helper uses grid graphics and is intended for two-dimensional HMAP or posterior partitions after their integer bounds have been converted to the desired plotting coordinates.

Usage

```
plot_partition(
  xy.part,
  border = FALSE,
  xlim = c(0, 1),
  ylim = c(0, 1),
  zlim = grDevices::extendrange(xy.part[, "den"]),
  main = "",
  nlevels = 100,
  plot.den = TRUE,
  color.fun = grDevices::terrain.colors,
  plot.scale = TRUE,
  color = NA,
  newpage = TRUE
)

## S3 method for class 'part'
plot(x, ...)
```

Arguments

`xy.part`	A matrix or data frame with columns ‘xmin‘, ‘xmax‘, ‘ymin‘, ‘ymax‘, and ‘den‘.
`border`	‘FALSE‘ uses the fill colors as borders, ‘TRUE‘ uses black, and a single color string uses that border color.
`xlim`, `ylim`	Horizontal and vertical plotting limits.
`zlim`	Density limits used to map ‘den‘ to colors.

main	Plot title.
nlevels	Number of colors in the palette; at least two.
plot.den	Whether to color cells by ‘den‘.
color.fun	A function that returns ‘nlevels‘ colors.
plot.scale	Whether to draw a density color scale.
color	Optional fixed fill color. The default ‘NA‘ uses ‘color.fun‘.
newpage	Whether to start a new grid graphics page.
x	A partition object for the legacy ‘plot.part()‘ S3 method.
...	Additional arguments passed to [plot_partition()].

Value

Invisibly, the clipped partition data with an added ‘fill‘ column.

Examples

```
cells <- data.frame(
  xmin = c(0, 0.5), xmax = c(0.5, 1),
  ymin = c(0, 0), ymax = c(1, 1), den = c(0.6, 1.4)
)
plot_partition(cells, plot.scale = FALSE)
```

---

plot_roc_curve *Plot an Empirical ROC Curve*

Description

Uses empirical null-score quantiles as thresholds. The curve treats smaller scores as stronger evidence for the positive class.

Usage

```
plot_roc_curve(
  p,
  p0,
  fpr = seq(0, 1, by = 0.01),
  col = "black",
  xlim = c(0, 1),
  ylim = c(0, 1),
  main = "",
  lty = 1,
  lwd = 2
)

## S3 method for class 'roc'
plot(x, ...)
```

Arguments

p	Numeric scores under the alternative or positive class.
p0	Numeric scores under the null or negative class.
fpr	False-positive-rate values in ‘[0, 1]‘.
col, lty, lwd	Graphical parameters passed to [graphics::plot()].
xlim, ylim	Axis limits.
main	Plot title.
x	Scores for the legacy ‘plot.roc()‘ S3 method.
...	Additional arguments passed to [plot_roc_curve()].

Value

Invisibly, a data frame containing ‘fpr‘, ‘threshold‘, and ‘tpr‘.

Examples

```
set.seed(12345)
null <- rnorm(100)
alternative <- rnorm(100, mean = -1)
roc <- plot_roc_curve(alternative, null)
head(roc)
```

Index

apt, 2

cond.apt, 3
cond.opt, 5

opt, 7

plot.part *(*plot_partition*)*, 8
plot.roc *(*plot_roc_curve*)*, 9
plot_partition, 8
plot_roc_curve, 9