

# Package ‘mpt’

August 29, 2011

**Version** 0.4-0

**Date** 2011-08-29

**Title** Multinomial Processing Tree (MPT) Models

**Author** Florian Wickelmaier

**Maintainer** Florian Wickelmaier <wickelmaier@web.de>

**Depends** R (>= 2.0.0), stats

**Description** Fitting and testing multinomial processing tree models, a class of statistical models for categorical data with latent parameters. These parameters are the link probabilities of a tree-like graph and represent the cognitive processing steps executed to arrive at observable response categories (Batchelder & Riefer, 1999; Erdfelder et al., 2009; Riefer & Batchelder, 1988).

**License** GPL (>= 2)

**URL** <http://www.r-project.org>,  
<http://homepages.uni-tuebingen.de/florian.wickelmaier>

**Repository** CRAN

**Date/Publication** 2011-08-29 17:37:53

## R topics documented:

logLik.mpt	2
mpt	3
mptEM	5
mptmodel	7
plot.mpt	8
proact	9
residuals.mpt	11
retroact	12
simulate.mpt	14

<b>Index</b>	<b>16</b>
--------------	-----------

---

`logLik.mpt`*Log-Likelihood of an mpt Object*

---

### Description

Returns the log-likelihood value of the (joint) multinomial processing tree model represented by `object` evaluated at the estimated parameters.

### Usage

```
## S3 method for class 'mpt'  
logLik(object, ...)
```

### Arguments

<code>object</code>	an object inheriting from class <code>mpt</code> , representing a fitted multinomial processing tree model.
<code>...</code>	some methods for this generic require additional arguments. None are used in this method.

### Value

The log-likelihood of the model represented by `object` evaluated at the estimated parameters.

### See Also

[mpt](#), [logLik.lm](#), [AIC](#).

### Examples

```
dat <- data.frame(freq = c(243, 64, 58, 55)) # from Riefer et al. (2002)  
mpt1 <- mpt(freq ~ list(  
  c*r,  
  (1 - c)*u^2,  
  2*(1 - c)*u*(1 - u),  
  c*(1 - r) + (1 - c)*(1 - u)^2),  
dat)  
  
logLik(mpt1)  
AIC(mpt1)
```

mpt

*Multinomial Processing Tree (MPT) Models***Description**

Fits a (joint) multinomial processing tree model using the EM algorithm.

**Usage**

```
mpt(formula, data, treeid = "treeid", constr = NULL,
    start = rep(0.5, length(all.vars(formula[[3]]))), ...)

## S3 method for class 'mpt'
summary(object, ...)

## S3 method for class 'mpt'
anova(object, ..., test = c("Chisq", "none"))
```

**Arguments**

formula	an object of class <code>formula</code> : a symbolic description of the model to be fitted. (See <a href="#">Details and Examples</a> .)
data	a data frame consisting at least of one variable that contains the absolute response frequencies. Alternatively, a (named) vector of frequencies.
treeid	name of the variable that identifies the processing trees of a joint multinomial model. Alternatively, a vector that identifies each tree.
constr	a named list of character vectors consisting of parameter names. The parameters in each vector are constrained to be equal.
start	a vector of starting values for the parameter estimates.
...	additional arguments passed to <a href="#">mptEM</a> .
object	an object of class <code>mpt</code> , typically the result of a call to <code>mpt</code> .
test	should the p-values of the chi-square distributions be reported?

**Details**

Multinomial processing tree models (Batchelder & Riefer, 1999; Erdfelder et al., 2009; Riefer & Batchelder, 1988) seek to represent the categorical responses of a group of subjects by a small number of latent (psychological) parameters. These models have a tree-like graph, the links being the parameters, the leaves being the response categories. The path from the root to one of the leaves represents the cognitive processing steps executed to arrive at a given response.

Each row in `data` corresponds to one response category. The model formula symbolically describes the model, for example it could look like

```
var ~ list(x + (1 - x)*y, (1 - x)*(1 - y), y, 1 - y, ...)
```

where the left hand side is the variable that contains the response frequencies. The right hand side gives the model equations in a list. Each component of the list represents the probability of a response in the corresponding category (link probabilities are multiplied, branch probabilities are added). Thus, the list usually has as many components as rows in the data frame.

Joint (or product) multinomial models consist of more than one processing tree. The `treeid` should uniquely identify each tree. The model formula does not distinguish between joint and simple models: All model equations should go into a single list.

Parameter estimation is carried out by `mptEM` which implements the EM algorithm.

### Value

<code>coefficients</code>	a vector of parameter estimates.
<code>fitted.values</code>	the fitted response frequencies.
<code>loglik</code>	the log-likelihood of the fitted model.
<code>a</code>	structural constants passed to <code>mptEM</code> .
<code>b</code>	structural constants passed to <code>mptEM</code> .
<code>c</code>	structural constants passed to <code>mptEM</code> .
<code>goodness.of.fit</code>	the goodness of fit statistic including the likelihood ratio fitted vs. saturated model (G2), the degrees of freedom, and the p-value of the corresponding chi-square distribution.
<code>iter</code>	the number of iterations of the EM algorithm.
<code>pcat</code>	the predicted probabilities for each response category.
<code>pbranch</code>	the predicted probabilities for each branch in the tree.
<code>formula</code>	the model formula.
<code>ntrees</code>	the number of trees in a joint multinomial model.
<code>n</code>	the total number of observations per tree.
<code>y</code>	the vector of response frequencies.

### Author(s)

Florian Wickelmaier

### References

- Batchelder, W.H. & Riefer, D.M. (1999). Theoretical and empirical review of multinomial process tree modeling. *Psychonomic Bulletin & Review*, **6**, 57–86.
- Erdfelder, E., Auer, T., Hilbig, B.E., Assfalg, A., Moshagen, M., & Nadarevic, L. (2009). Multinomial processing tree models: A review of the literature. *Zeitschrift fuer Psychologie*, **217**, 108–124.
- Riefer, D.M., & Batchelder, W.H. (1988). Multinomial modeling and the measurement of cognitive processes. *Psychological Review*, **95**, 318–339.

**See Also**

[mptEM](#), [mptmodel](#), [simulate.mpt](#), [retroact](#), [proact](#), [plot.mpt](#), [residuals.mpt](#), [logLik.mpt](#).

**Examples**

```
## Storage-retrieval model for pair clustering (Riefer & Batchelder, 1988)
data(retroact)

mpt1 <- mpt(freq ~ list(
  c*r,
  (1 - c)*u^2,
  2*(1 - c)*u*(1 - u),
  c*(1 - r) + (1 - c)*(1 - u)^2,
  u,
  1 - u
), retroact[retroact$lists == 0,])

summary(mpt1) # parameter estimates, goodness of fit
plot(mpt1) # residuals versus predicted values
confint(mpt1) # approximate confidence intervals

plot(mpt1$coef, axes=FALSE, ylim=0:1, pch=16, xlab="",
      ylab="Parameter estimate (MPT model, 95% CI)")
axis(1, 1:3, names(mpt1$coef)); axis(2)
arrows(1:3, confint(mpt1)[,1], 1:3, confint(mpt1)[,2], .05, 90, 3)

## See ?retroact and ?proact for further examples.
```

---

mptEM

*EM Algorithm for Multinomial Processing Tree Models*


---

**Description**

Applies the EM algorithm to fit a multinomial processing tree model.

**Usage**

```
mptEM(theta, data, a, b, c, maxit = 1000, tolerance = 1e-8,
       stepsize = 1, verbose = FALSE)
```

**Arguments**

theta	a vector of starting values for the parameter estimates.
data	a vector of absolute response frequencies.
a	a three-dimensional array representing the model structure.
b	a three-dimensional array representing the model structure.

c	a matrix of structural constants.
maxit	the maximum number of iterations.
tolerance	the convergence criterion; the iterations converge when $\log Lik - \log Lik.old < tolerance$ .
stepsize	the step size defaulting to 1; slightly larger values may speed up convergence, but may also give errors; use with care.
verbose	logical indicating if output should be produced for each iteration.

### Details

Usually `mptEM` is automatically called by `mpt`.

The algorithm applies to MPT models where the probabilities of the  $i$ -th branch leading to the  $j$ -th category take the form

$$p_{ij}(\Theta) = c_{ij} \prod_{s=1}^S \theta_s^{a_{ijs}} (1 - \theta_s)^{b_{ijs}},$$

where  $\Theta$  is the parameter vector.

See Hu and Batchelder (1994) and Hu (1999) for details on the algorithm.

### Value

theta	the vector of parameter estimates.
loglik	the log-likelihood at termination of the algorithm.
pcat	a vector of predicted probabilities for each response category.
pbranch	a vector of predicted branch probabilities.
iter	the number of iterations of the algorithm.

### References

Hu, X., & Batchelder, W.H. (1994). The statistical analysis of general processing tree models with the EM algorithm. *Psychometrika*, **59**, 21–47.

Hu, X. (1999). Multinomial processing tree models: An implementation. *Behavior Research Methods, Instruments, & Computers*, **31**, 689–695.

### See Also

`mpt`.

---

`mptmodel`*Formulae for Some Prevalent MPT Models*

---

**Description**

Returns formulae for some prevalent MPT models.

**Usage**

```
mptmodel(which, replicates = 1, response = "freq")
```

**Arguments**

<code>which</code>	character string identifying the model. See Details.
<code>replicates</code>	number of replicates.
<code>response</code>	name of the response variable.

**Details**

Currently implemented are the following models:

1HT: the one-high threshold model (see Erdfelder et al., 2009).

2HT: the two-high threshold model (see Broeder & Schuetz, 2009).

PairAsso: the paired-associate learning model (Riefer & Batchelder, 1988).

SourceMon: the source-monitoring model (Batchelder & Riefer, 1990).

SR: the storage-retrieval model (Riefer & Batchelder, 1988).

**Value**

A formula that may be used as input for `mpt`.

**References**

Batchelder, W.H. & Riefer, D.M. (1990). Multinomial processing models of source monitoring. *Psychological Review*, **97**, 548–564.

Broeder, A. & Schuetz, J. (2009). Recognition ROCs are curvilinear—or are they? On premature arguments against the two-high-threshold model of recognition. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, **35**, 587–606.

Erdfelder, E., Auer, T., Hilbig, B.E., Assfalg, A., Moshagen, M., & Nadarevic, L. (2009). Multinomial processing tree models: A review of the literature. *Zeitschrift fuer Psychologie*, **217**, 108–124.

Riefer, D.M., & Batchelder, W.H. (1988). Multinomial modeling and the measurement of cognitive processes. *Psychological Review*, **95**, 318–339.

**See Also**

`mpt`.

**Examples**

```
## Fit one-high threshold model to data in Broeder & Schuetz (2009)
mpt(mptmodel("1HT"), c(t=55, t=35, d=45, d=765))
```

---

plot.mpt

*Diagnostic Plot for MPT Models*


---

**Description**

Plots MPT residuals against fitted values.

**Usage**

```
## S3 method for class 'mpt'
plot(x, showID = TRUE,
     xlab = "Predicted response probabilities", ylab = "Deviance residuals",
     ...)
```

**Arguments**

`x` an object of class `mpt`, typically the result of a call to `mpt`.  
`showID` logical. Should the `treeid`'s be plotted? Defaults to `TRUE`.  
`xlab, ylab, ...` graphical parameters passed to `plot`.

**Details**

The deviance residuals are plotted against the predicted response probabilities. If `showID` is `true`, plotting symbols are the `treeid`'s used when the model was fit.

**See Also**

[mpt, residuals.mpt](#).

**Examples**

```
## Compare two constrained MPT models

data(proact)

mpt1 <- mpt(freq ~ list(
  p1*q1*r1,
  p1*q1*(1 - r1),
  p1*(1 - q1)*r1,
  (1 - p1) + p1*(1 - q1)*(1 - r1),

  p2*q2*r2,
  p2*q2*(1 - r2),
```

```

p2*(1 - q2)*r2,
(1 - p2) + p2*(1 - q2)*(1 - r2),

p3*q3*r3,
p3*q3*(1 - r3),
p3*(1 - q3)*r3,
(1 - p3) + p3*(1 - q3)*(1 - r3)
), proact[proact$test == 1,], constr = list(p=c("p1", "p2", "p3"))

mpt2 <- mpt(mpt1$formula, proact[proact$test == 1,],
  constr = list(q=c("q1", "q2", "q3")))

par(mfrow = c(1,2)) # residuals versus fitted values
plot(mpt1, main = "p constrained", ylim = c(-3, 3.5)) # good fit
plot(mpt2, main = "q constrained", ylim = c(-3, 3.5)) # bad fit

```

---

proact

*Recall Frequencies for DaPolito's Experiment on Proactive Inhibition*


---

## Description

In DaPolito's experiment (Greeno, James, DaPolito & Polson, 1978), 60 subjects were presented with lists of stimulus-response associates to be learned, followed by a test in which only the stimuli were presented and the responses had to be recalled. Stimuli consisted of three-letter syllables, responses of the numbers from 1 to 30, so list items looked like, say, ESI-12, JOK-3, MAL-8, etc. Part of the items had two responses (A-B, A-C), the control items had only a single correct response. If the recall of C responses is poorer than that of control items, then proactive inhibition has occurred, that is interference with the recall by information that has been learned earlier.

Riefer and Batchelder (1988) analyzed only the A-B and A-C items. They investigated how repeated A-B presentation affects the B and C recall, respectively. The responses were classified into four categories and pooled across subjects.

## Usage

```
data(proact)
```

## Format

A data frame consisting of four variables:

`test` first or second test.

`abpres` the number of A-B presentations.

`resp` a factor giving the response category; `Bc` both B and C responses are correctly recalled, `Bc` only B is recalled, `cB` only C is recalled, `bc` neither response is recalled.

`freq` the aggregate recall frequencies per condition.

`treeid` an identifier for the single trees of the joint multinomial model.

## Source

Greeno, J.G., James, C.T., DaPolito, F., & Polson, P.G. (1978). *Associative learning: A cognitive analysis*. Englewood Cliffs, NJ: Prentice-Hall.

Riefer, D.M., & Batchelder, W.H. (1988). Multinomial modeling and the measurement of cognitive processes. *Psychological Review*, **95**, 318–339.

## Examples

```
data(proact)

## Testing hypotheses about the parameters
mpt1 <- mpt(freq ~ list(
  p1*q1*r1,
  p1*q1*(1 - r1),
  p1*(1 - q1)*r1,
  (1 - p1) + p1*(1 - q1)*(1 - r1),

  p2*q2*r2,
  p2*q2*(1 - r2),
  p2*(1 - q2)*r2,
  (1 - p2) + p2*(1 - q2)*(1 - r2),

  p3*q3*r3,
  p3*q3*(1 - r3),
  p3*(1 - q3)*r3,
  (1 - p3) + p3*(1 - q3)*(1 - r3),

  p4*q4*r4,
  p4*q4*(1 - r4),
  p4*(1 - q4)*r4,
  (1 - p4) + p4*(1 - q4)*(1 - r4),

  p5*q5*r5,
  p5*q5*(1 - r5),
  p5*(1 - q5)*r5,
  (1 - p5) + p5*(1 - q5)*(1 - r5),

  p6*q6*r6,
  p6*q6*(1 - r6),
  p6*(1 - q6)*r6,
  (1 - p6) + p6*(1 - q6)*(1 - r6)
), proact, maxit=2000, stepsize=1.175)

mpt2 <- mpt(mpt1$formula, proact,
  constr=list(q1 = c("q1", "q2", "q3"), q2 = c("q4", "q5", "q6")),
  start=coef(mpt1))

mpt3 <- mpt(mpt1$formula, proact,
  constr=list(r1 = c("r1", "r2", "r3"), r2 = c("r4", "r5", "r6")),
  start=coef(mpt1), stepsize=1.9)
```

```
anova(mpt2, mpt1) # q increases with number of A-B presentations
anova(mpt3, mpt1) # r remains constant
```

---

```
residuals.mpt      Residuals for MPT Models
```

---

## Description

Computes deviance and Pearson residuals for `mpt` objects.

## Usage

```
## S3 method for class 'mpt'
residuals(object, type = c("deviance", "pearson"), ...)
```

## Arguments

<code>object</code>	an object of class <code>mpt</code> , typically the result of a call to <a href="#">mpt</a> .
<code>type</code>	the type of residuals which should be returned; the alternatives are: "deviance" (default) and "pearson".
<code>...</code>	further arguments passed to or from other methods. None are used in this method.

## Details

See [residuals.glm](#) for details.

## Value

A named vector of residuals having as many elements as response categories. The names are given by the `treeid`.

## See Also

[mpt](#), [residuals.glm](#), [plot.mpt](#).

## Examples

```
data(retroact) # retroactive inhibition data

mpt1 <- mpt(freq ~ list( # fit storage-retrieval model
  c*r,
  (1 - c)*u^2,
  2*(1 - c)*u*(1 - u),
  c*(1 - r) + (1 - c)*(1 - u)^2,
  u,
  1 - u
), retroact[retroact$lists == 0,])
```

```
sum( resid(mpt1)^2 )           # likelihood ratio G2
sum( resid(mpt1, "pearson")^2 ) # Pearson X2
```

---

retroact

*Recall Frequencies for a Retroactive Inhibition Experiment*


---

## Description

The experiment is described in Riefer and Batchelder (1988). Each of the 75 subjects was presented with either one, two, three, four, or five successive lists of words (15 subjects per group). These words were shown in random order on a computer screen, one word at a time, at a rate of 5 s per word. Each list contained 25 words, consisting of 10 categories (with 2 associate words per category) and five singletons. Subjects were given 1.5 min to recall in writing the 25 words from each individual list. After all of the lists had been presented, a final free-recall test was given in which subjects attempted to recall the words from all of the previous lists. Subjects were given up to 5 min for this final written recall.

The focus here is on the recall of the first-list words during the final recall task. The responses were classified into six categories and pooled across subjects.

## Usage

```
data(retroact)
```

## Format

A data frame consisting of four variables:

`lists` the number of interpolated lists.

`treeid` an identifier for the single trees of the joint multinomial model.

`resp` a factor giving the response category; E1 pair is recalled adjacently, E2 pair is recalled non-adjacently, E3 one word in a pair is recalled, E4 neither word in a pair is recalled, F1 recall of a singleton, F2 non-recall of a singleton.

`freq` the aggregate recall frequencies per condition.

## Source

Riefer, D.M., & Batchelder, W.H. (1988). Multinomial modeling and the measurement of cognitive processes. *Psychological Review*, **95**, 318–339.

## Examples

```
data(retroact)
```

```
## Fitting individual storage-retrieval models per condition
formula <- freq ~ list(
  c*r,
```

```

      (1 - c)*u^2,
      2*(1 - c)*u*(1 - u),
      c*(1 - r) + (1 - c)*(1 - u)^2,
      u,
      1 - u
    )

pars <- sapply(0:4,
  function(x) coef(mpt(formula, retroact[retroact$lists == x,])))

## Figure 3 in Riefer & Batchelder (1988)
plot(pars["c",] ~ I(0:4), pch=16, type="b", ylim=c(.3, 1),
  xlab="Number of interpolated lists, j",
  ylab="Parameter estimate (Storage-retrieval model)")
points(pars["r",] ~ I(0:4), type="b", lty=2)
text(3, .89, expression("Storage of clusters," ~ hat(c)[j]))
text(3, .46, expression("Retrieval of clusters," ~ hat(r)[j]))

## Testing hypotheses about the parameters
mpt1 <- mpt(freq ~ list(
  c0*r0,
  (1 - c0)*u0^2,
  2*(1 - c0)*u0*(1 - u0),
  c0*(1 - r0) + (1 - c0)*(1 - u0)^2,
  u0,
  1 - u0,

  c1*r1,
  (1 - c1)*u1^2,
  2*(1 - c1)*u1*(1 - u1),
  c1*(1 - r1) + (1 - c1)*(1 - u1)^2,
  u1,
  1 - u1,

  c2*r2,
  (1 - c2)*u2^2,
  2*(1 - c2)*u2*(1 - u2),
  c2*(1 - r2) + (1 - c2)*(1 - u2)^2,
  u2,
  1 - u2,

  c3*r3,
  (1 - c3)*u3^2,
  2*(1 - c3)*u3*(1 - u3),
  c3*(1 - r3) + (1 - c3)*(1 - u3)^2,
  u3,
  1 - u3,

  c4*r4,
  (1 - c4)*u4^2,
  2*(1 - c4)*u4*(1 - u4),
  c4*(1 - r4) + (1 - c4)*(1 - u4)^2,
  u4,

```

```

      1 - u4
    ), retroact)

mpt2 <- mpt(mpt1$formula, retroact, constr=list(r = paste("r", 0:4, sep="")))
mpt3 <- mpt(mpt1$formula, retroact, constr=list(c = paste("c", 0:4, sep="")))

anova(mpt2, mpt1) # r decreases the more lists have been interpolated
anova(mpt3, mpt1) # c remains constant

```

---

simulate.mpt

*Simulate Responses from MPT Models*


---

## Description

Simulates responses from the distribution corresponding to a fitted `mpt` model object.

## Usage

```

## S3 method for class 'mpt'
simulate(object, nsim, seed, pool = TRUE, ...)

```

## Arguments

<code>object</code>	an object of class <code>mpt</code> , typically the result of a call to <code>mpt</code>
<code>nsim</code>	currently not used
<code>seed</code>	currently not used
<code>pool</code>	logical, if <code>TRUE</code> (default), pooled responses (summed across respondents) are returned
<code>...</code>	further arguments passed to or from other methods. None are used in this method.

## Details

Responses are simulated by (repeatedly) applying `rmultinom` with sizes taken from the original sample and probabilities computed from the model object.

## Value

A named vector of (pooled) responses. Names identify the tree from which responses were simulated.

## See Also

[mpt](#).

**Examples**

```
data(retroact)

mpt1 <- mpt(freq ~ list(
  c*r,
  (1 - c)*u^2,
  2*(1 - c)*u*(1 - u),
  c*(1 - r) + (1 - c)*(1 - u)^2,
  u,
  1 - u
), retroact[retroact$lists == 1,])

simulate(mpt1)

## Perform parametric bootstrap
LR.stat <- replicate(200, mpt(formula(mpt1), simulate(mpt1))$good[1])

hist(LR.stat, col="lightgrey", border="white", freq=FALSE, breaks=20,
     main="Parametric bootstrap")
curve(dchisq(x, df=mpt1$good[1]), add=TRUE)
abline(v=mpt1$good[1], lty=2)
```

# Index

## \*Topic **datasets**

proact, 9

retroact, 12

## \*Topic **models**

logLik.mpt, 2

mpt, 3

mptEM, 5

mptmodel, 7

plot.mpt, 8

residuals.mpt, 11

simulate.mpt, 14

AIC, 2

anova.mpt (*mpt*), 3

logLik.lm, 2

logLik.mpt, 2, 5

mpt, 2, 3, 6–8, 11, 14

mptEM, 3, 4, 5, 5

mptmodel, 5, 7

plot.mpt, 5, 8, 11

print.mpt (*mpt*), 3

print.summary.mpt (*mpt*), 3

proact, 5, 9

residuals.glm, 11

residuals.mpt, 5, 8, 11

retroact, 5, 12

rmultinom, 14

simulate.mpt, 5, 14

summary.mpt (*mpt*), 3

vcov.mpt (*mpt*), 3