

Semiparametric Mixed Effects with Hierarchical DP Mixture

April 21, 2007

R topics documented:

hdpm-package	1
hdpm	2
hdpmFitSetup	6
hdpmFit	8
hdpmPredictSetup	9
hdpmPredict	12

Index	15
--------------	-----------

hdpm-package	<i>Hierarchical mixture of Dirichlet process (HDPM) models for repeated measurement data</i>
--------------	--

Description

Inference for a mixed effects model with hierarchical DP mixture prior for the random effects. The functions are specific to the longitudinal data model for the application in *Mueller, Quintana and Rosner (2004)*.

Details

Package: hdpm
Type: Package
Version: 1.0
Date: 2007-04-15
License: GNU

The function `hdpm` initializes and carries out Markov chain Monte Carlo posterior simulation. Use the functions `hdpmPredictSetup` and `hdpmPredict` to obtain posterior predictive summaries for patients with given covariates under each of the studies. Use the functions `hdpmFitSetup` and `hdpmFit` to obtain fitted profiles for the observed patients.

The functions are specific to the longitudinal data model for the application in *Mueller, Quintana and Rosner (2004)*. See the paper for details of the model. In summary, the implemented model is as follows. We assume a normal non-linear regression for y_{jik} , the k -th observation for the i -th

patient in study j . The mean function $m(\cdot; \theta_{ji})$ is indexed by a patient-specific random effects vector θ_{ji}

$$p(y_{jik} | \theta_{ji}, \eta) = N(m(t_{jik}, \theta_{jik}), \sigma^2)$$

See equation (14) of the paper for details on the definition of the mean function. The random effect θ_{ji} arise from a random effects model for the j -th study, which in turn is a mixture of a measure F_0 that is common to all studies, and an idiosyncratic measure F_j that is specific to the j -th study.

$$\theta_{ji} \sim \epsilon F_0 + (1 - \epsilon) F_j$$

The random probability measures F_j in turn are given a Dirichlet process mixture of normal prior. We assume

$$F_j(\theta) = \int N(\mu, S) dG_j(\mu), \quad j = 0, 1, \dots, J$$

with $G_j \sim DP(G^*(\eta), \alpha)$. Here η are hyperparameters that index the base measure of the DP prior. We use a normal base measure and a conjugate hyperprior

$$G^*(\mu) = N(m, B), \quad \text{with } m \sim N(a, A), \quad \text{and } B^{-1} \sim \text{Wishart}(c, (cC)^{-1})$$

The Wishart prior is parametrized such that $E(B^{-1}) = C^{-1}$. Let δ_x denote a point mass at x . We complete the model with the hyperpriors

$$S^{-1} \sim W(q, (qR)^{-1}), \quad p(\epsilon) = \pi_0 \delta_0 + \pi_1 \delta_1 + (1 - \pi_0 - \pi_1) \text{Be}(a_\epsilon, b_\epsilon)$$

Regression on covariates in the random effects model can be achieved by including covariates in the random effects vector θ_{ji} . See *Mueller et al. (2004, section 3.3)* for details.

Author(s)

Peter Mueller

Maintainer: Peter Mueller <pm@wotan.mdacc.tmc.edu>

See Also

The functions in this package implement the application in *Mueller et al. (2004)*. Alternatively, the package [hdpmn](#) implements the same hierarchical mixture of Dirichlet process models for a multivariate normal sampling model and is useful for a wider class of applications.

Examples

hdpm

MCMC for the hierarchical DP mixture

Description

Initializes and runs posterior MCMC for the semiparametric mixed effects model with hierarchical DP mixture random effects distribution. The function is specific for the application in *Mueller et al. (2004)*.

Usage

```
hdpm(Y = NULL, Z = NULL,
      n = NULL, npatients = NULL, nstudies = NULL, py = NULL,
      n.iter = 1000, n.discard = 100, n.batch = 50, mcmc.c = 1,
      verbose = 3, seed1 = 981963, seed2 = 6869504,
      mcmc.eps = 0, sig = 1, as = 2, bs = 2,
      eps = 0.1, ae = 1, be = 1, pe1 = 0.1, pe0 = 0.1,
      pz = NULL, px = NULL,
      m.prior = 0, B.prior = 0, S.prior = 0,
      alpha.prior = 0, n.predupdate = 100,
      S.init = NULL, q = 5, R = NULL,
      B.init = NULL, cc = 5, C = NULL,
      m.init = NULL, a = NULL, A = NULL,
      alpha = 1, a0 = 1, b0 = 1,
      k0 = NULL, work.dir = NULL, header = T)
```

Arguments

Y	Data, as (n by py) matrix or file name. Column 1 is the patient index. Column 2 is the day t_{jik} . Column 3 is the response y_{jik} . The last column is a study index.
Z	Random effects (initial values) and covariates, as a (npatients by p) matrix or file name, where $p = pz + px$. The i-th row of Z reports the i-th patient. The first pz columns are the random effects vector. The last px columns are the covariates.
n	Total number of observations y_{jik} .
npatients	Total number of patients, counting across all studies.
nstudies	Number of studies, = J in $j = 1, \dots, J$.
py	Number of columns in the data matrix Y.
n.iter	Number of Markov chain Monte Carlo iterations.
n.discard	Initial burn-in to be discarded.
n.batch	Save imputed parameter values every n.batch iterations.
mcmc.c	Step size for Metropolis-Hastings random walk proposals.
verbose	0 is silent, 3 is verbose.
seed1	Random variate seed.
seed2	Random variate seed.
mcmc.eps	Indicator for resampling ϵ . If zero, ϵ will be fixed.
sig	Initial value for the residual standard deviation σ .
as	Prior parameters for an inverse Gamma prior $\sigma^2 \sim IG(as/2, bs/2)$.
bs	
eps	Initial value for ϵ .
ae	Prior parameters for a Beta prior on ϵ
be	
pe1	Point mass at $\epsilon = 1$
pe0	Point mass at $\epsilon = 0$
pz	Dimension of the random effects vector θ_{ji} , without the covariate vector.

<code>px</code>	Dimension of a subject-specific covariate vector.
<code>m.prior</code>	Indicator for resampling m , the mean of the normal base measure.
<code>B.prior</code>	Indicator for resampling B , the covariance matrix of the normal base measure.
<code>S.prior</code>	Indicator for resampling S , the covariance matrix of the normal kernel in the DP mixture.
<code>alpha.prior</code>	Indicator for resampling α , the total mass parameter of the DP prior.
<code>n.predupdate</code>	Batch size to update posterior predictive inference in the MCMC simulation.
<code>S.init</code>	initial value for S . <code>NULL</code> is allowed.
<code>q</code>	degrees of freedom in the inverse Wishart prior for S
<code>R</code>	matrix-variate parameter for the inverse Wishart prior for S
<code>B.init</code>	initial value for B , the covariance matrix in the DP base measure.
<code>cc</code>	degrees of freedom in the inverse Wishart prior for B
<code>C</code>	matrix-variate parameter for the inverse Wishart prior for B
<code>m.init</code>	initial value for m .
<code>a</code>	mean of the normal hyperprior for m .
<code>A</code>	covariance matrix of the normal hyperprior for m .
<code>alpha</code>	initial value of the total mass parameter in the DP prior.
<code>a0</code>	hyperparameters in the Gamma prior for $alpha$.
<code>b0</code>	
<code>k0</code>	initial number of clusters.
<code>work.dir</code>	directory where working files will be saved. <code>NULL</code> indicates to use the current working directory.
<code>header</code>	indicator whether the data files (for Y and Z) include a header.

Details

See [hdpm](#) for a statement of the probability model.

Value

The function returns no value. MCMC simulations are saved in files in the designated working directory.

Note

The only functions in the source code that are specific to the application in Mueller et al. (2004) are the definition of the top level sampling model in the file `lpost.c` and the functions for resampling (transition probabilities in the MCMC) for the random effects in the source file `hdpm.c`. Both source files are in the directory `src` of the package distribution. Users with experience in coding MCMC could appropriately change these functions and recompile the package for a different longitudinal mean function.

Author(s)

Peter Mueller

See Also

The functions in this package implement the application in *Mueller et al. (2004)*.

Alternatively, the package [hdpmn](#) implements the same hierarchical mixture of Dirichlet process models for a multivariate normal sampling model and is useful for a wider class of applications.

Examples

```
## Not run:
require(hdpm)

## data files
data.dir <- system.file("data",package="hdpm")
work.dir <- paste(getwd(),"/tmp/",sep="")

## data files
Y <- file.path(data.dir,"CALGB.txt")
  ## data: responses
Z <- file.path(data.dir,"CALGBz.txt")
  ## random effects (initial values) and covariates
X <- file.path(data.dir,"CALGBz0.txt")
  ## same as Z, for future patients (for prediction)

## run MCMC -- save working files in work.dir
hdpm(Y=Y,Z=Z,px=3,q=15,cc=15,work.dir=work.dir)

## post-process MCMC output for predictive inference
## save posterior predictive summaries in work.dir
hdpmPredictSetup(X=X,px=3,
                 j=c(0,1,2,1,2),r=c(0,0,0,1,1),
                 xgrid=c(0,21),work.dir=work.dir)

## plot predictive for future patients in study 1 and 2,
for(i in c(5,7)){
  hdpmPredict(j=1,r=0,i=i)
  hdpmPredict(j=2,r=0,i=i)
}

## part of the predictive corresponding to the idiosyncratic measure
for(i in c(5,7)){
  hdpmPredict(j=1,r=1,i=i, work.dir=work.dir)
  hdpmPredict(j=2,r=1,i=i, work.dir=work.dir)
}

## post-process MCMC output for fitted mean curves
fits <- hdpmFitSetup(px=3,xgrid=c(0,21))

## plot fitted mean curves
par(mfrow=c(2,2),ask=T)
y <- as.matrix(read.table(Y,header=T)) # read in data
exs <- c(10:17,20:27,30:37,90:97)
for(i in exs){
  hdpmFit(fits,i=i)
  idx <- which(y[,1]==i-1)
  points(y[idx,2],y[idx,3],pch=19)
}
```

```
## End(Not run)
```

```
hdpmFitSetup          Fitted mean functions
```

Description

Computes posterior expected mean profiles and corresponding posterior standard deviations for observed patients

Usage

```
hdpmFitSetup(
  n = 100,
  px = NULL,
  sig = 1,
  nx = 25, xgrid = NULL,
  work.dir=NULL)
```

Arguments

<code>n</code>	Number of MCMC posterior simulations to be used for the fitted profiles.
<code>px</code>	Dimension of covariate vector
<code>sig</code>	Residual standard deviation (dummy)
<code>nx</code>	Size of grid on the time axis
<code>xgrid</code>	Range of grid, as pair of min and max.
<code>work.dir</code>	working directory to save posterior summaries.

Details

Evaluates $E[m(t; \theta_{ji}) \mid data]$ (and corresponding posterior standard deviations) as a function of time t for all patients in the data sets.

Value

The function returns a list with elements

<code>e</code>	(<code>npatients</code> by <code>nx</code>) matrix of posterior expected mean functions. Each row corresponds to a patient, each column corresponds to a time point in <code>xgrid</code> .
<code>s</code>	Same for posterior standard deviations.
<code>x</code>	(<code>npatients</code> by <code>px</code>) matrix of patient-specific covariates
<code>xgrid</code>	grid on the time axis

Note

The only functions in the source code that are specific to the application in Mueller et al. (2004) are the definition of the top level sampling model in the file `lpost.c` and the functions for resampling (transition probabilities in the MCMC) for the random effects in the source file `hdpm.c`. Both source files are in the directory `src` of the package distribution. Users with experience in coding MCMC could appropriately change these functions and recompile the package for a different longitudinal mean function.

Author(s)

Peter Mueller

See Also

The functions in this package implement the application in *Mueller et al. (2004)*. Alternatively, the package [hdpmn](#) implements the same hierarchical mixture of Dirichlet process models for a multivariate normal sampling model and is useful for a wider class of applications.

Examples

```
## Not run:
require(hdpm)

## data files
data.dir <- system.file("data",package="hdpm")
work.dir <- paste(getwd(),"/tmp/",sep="")

## data files
Y <- file.path(data.dir,"CALGB.txt")
  ## data: responses
Z <- file.path(data.dir,"CALGBz.txt")
  ## random effects (initial values) and covariates
X <- file.path(data.dir,"CALGBz0.txt")
  ## same as Z, for future patients (for prediction)

## run MCMC -- save working files in work.dir
hdpm(Y=Y,Z=Z,px=3,q=15,cc=15,work.dir=work.dir)

## post-process MCMC output for predictive inference
## save posterior predictive summaries in work.dir
hdpmPredictSetup(X=X,px=3,
                 j=c(0,1,2,1,2),r=c(0,0,0,1,1),
                 xgrid=c(0,21),work.dir=work.dir)

## plot predictive for future patients in study 1 and 2,
for(i in c(5,7)){
  hdpmPredict(j=1,r=0,i=i)
  hdpmPredict(j=2,r=0,i=i)
}

## part of the predictive corresponding to the idiosyncratic measure
for(i in c(5,7)){
  hdpmPredict(j=1,r=1,i=i, work.dir=work.dir)
  hdpmPredict(j=2,r=1,i=i, work.dir=work.dir)
}

## post-process MCMC output for fitted mean curves
fits <- hdpmFitSetup(px=3,xgrid=c(0,21))

## plot fitted mean curves
par(mfrow=c(2,2),ask=T)
y <- as.matrix(read.table(Y,header=T)) # read in data
exs <- c(10:17,20:27,30:37,90:97)
for(i in exs){
  hdpmFit(fits,i=i)
```

```

idx <- which(y[,1]==i-1)
points(y[idx,2],y[idx,3],pch=19)
}
## End(Not run)

```

hdpmFit

Plot posterior mean profiles

Description

Plots the posterior mean profile for observed patients.

Usage

```
hdpmFit(fit.outp = NULL, i = NULL, data.dir = NULL)
```

Arguments

<code>fit.outp</code>	list returned from <code>hdpmFitSetup</code> .
<code>i</code>	index of vector of indices of patients.

Details

Plots posterior mean profile $E(m(t; \theta_{ji}) \mid data)$ for t in `xgrid`, with plus/minus one posterior standard deviation as dashed curves.

Value

No return value.

Note

The only functions in the source code that are specific to the application in Mueller et al. (2004) are the definition of the top level sampling model in the file `lpost.c` and the functions for resampling (transition probabilities in the MCMC) for the random effects in the source file `hdpm.c`. Both source files are in the directory `src` of the package distribution. Users with experience in coding MCMC could appropriately change these functions and recompile the package for a different longitudinal mean function.

Author(s)

Peter Mueller

See Also

The functions in this package implement the application in Mueller et al. (2004). Alternatively, the package [hdpmn](#) implements the same hierarchical mixture of Dirichlet process models for a multivariate normal sampling model and is useful for a wider class of applications.

Examples

```

## Not run:
require(hdpm)

## data files
data.dir <- system.file("data", package="hdpm")
work.dir <- paste(getwd(), "/tmp/", sep="")

## data files
Y <- file.path(data.dir, "CALGB.txt")
  ## data: responses
Z <- file.path(data.dir, "CALGBz.txt")
  ## random effects (initial values) and covariates
X <- file.path(data.dir, "CALGBz0.txt")
  ## same as Z, for future patients (for prediction)

## run MCMC -- save working files in work.dir
hdpm(Y=Y, Z=Z, px=3, q=15, cc=15, work.dir=work.dir)

## post-process MCMC output for predictive inference
## save posterior predictive summaries in work.dir
hdpmPredictSetup(X=X, px=3,
                 j=c(0,1,2,1,2), r=c(0,0,0,1,1),
                 xgrid=c(0,21), work.dir=work.dir)

## plot predictive for future patients in study 1 and 2,
for(i in c(5,7)){
  hdpmPredict(j=1, r=0, i=i)
  hdpmPredict(j=2, r=0, i=i)
}

## part of the predictive corresponding to the idiosyncratic measure
for(i in c(5,7)){
  hdpmPredict(j=1, r=1, i=i, work.dir=work.dir)
  hdpmPredict(j=2, r=1, i=i, work.dir=work.dir)
}

## post-process MCMC output for fitted mean curves
fits <- hdpmFitSetup(px=3, xgrid=c(0,21))

## plot fitted mean curves
par(mfrow=c(2,2), ask=T)
y <- as.matrix(read.table(Y, header=T)) # read in data
exs <- c(10:17, 20:27, 30:37, 90:97)
for(i in exs){
  hdpmFit(fits, i=i)
  idx <- which(y[,1]==i-1)
  points(y[idx,2], y[idx,3], pch=19)
}
## End(Not run)

```

Description

Carries out postprocessing of the MCMC output to allow posterior predictive inference for patients with different covariates in different studies.

Usage

```
hdpmPredictSetup(
  j = 1, r = 0,
  n = 100,
  sig = 1,
  nx = 22, xgrid = NULL,
  npa = NULL, p = NULL, px = NULL, X = NULL,
  work.dir = NULL, header = T)
```

Arguments

<code>j</code>	study
<code>r</code>	indicator for including (0) or not (1) the common measure.
<code>n</code>	Number of imputed posterior simulations to use.
<code>sig</code>	Residual standard deviation.
<code>nx</code>	Size of the grid over the time axis to use.
<code>xgrid</code>	Pair with lower and upper end of the grid.
<code>p</code>	number of columns for X, i.e., dimension of the random effects vector including the covariates.
<code>px</code>	Dimension of covariate vector. The last <code>px</code> columns of X are assumed to be the covariates. The first <code>p-px</code> columns are assumed to be random effects.
<code>npa</code>	Number of rows in X, i.e., number of patients for posterior predictive inference.
<code>X</code>	Random effects (dummy values) and covariates for future patients, (<code>n</code> by <code>p</code>) matrix. X can be a matrix or a file name.
<code>work.dir</code>	directory to save working files.
<code>header</code>	indicator for a header line in X

Details

Must run `hdpm` first to carry out MCMC posterior simulation.

For `npa` assumed future patients with given covariates the function computes posterior predictive inference of future responses. Posterior predictive inference is (naturally) conditional on the fixed covariate values *and* conditional on the first random effect. This is meaningful if the first random effect is interpretable as a baseline. It makes posterior predictive inference comparable across covariates. If no conditioning is desired, make the first random effect a dummy.

The `npa` by `p` matrix X specifies the covariates for the future patients. The subvector of random effects in X is a dummy to match the dimension.

See [hpdm-package](#) for a statement of the probability model.

Value

No values are returned. All posterior predictive simulations are saved in working files in the directory `work.dir`.

Note

The only functions in the source code that are specific to the application in Mueller et al. (2004) are the definition of the top level sampling model in the file `lpost.c` and the functions for resampling (transition probabilities in the MCMC) for the random effects in the source file `hdpm.c`. Both source files are in the directory `src` of the package distribution. Users with experience in coding MCMC could appropriately change these functions and recompile the package for a different longitudinal mean function.

Author(s)

Peter Mueller

See Also

The functions in this package implement the application in *Mueller et al. (2004)*. Alternatively, the package [hdpmn](#) implements the same hierarchical mixture of Dirichlet process models for a multivariate normal sampling model and is useful for a wider class of applications.

Examples

```
## Not run:
require(hdpm)

## data files
data.dir <- system.file("data", package="hdpm")
work.dir <- paste(getwd(), "/tmp/", sep="")

## data files
Y <- file.path(data.dir, "CALGB.txt")
  ## data: responses
Z <- file.path(data.dir, "CALGBz.txt")
  ## random effects (initial values) and covariates
X <- file.path(data.dir, "CALGBz0.txt")
  ## same as Z, for future patients (for prediction)

## run MCMC -- save working files in work.dir
hdpm(Y=Y, Z=Z, px=3, q=15, cc=15, work.dir=work.dir)

## post-process MCMC output for predictive inference
## save posterior predictive summaries in work.dir
hdpmPredictSetup(X=X, px=3,
                 j=c(0, 1, 2, 1, 2), r=c(0, 0, 0, 1, 1),
                 xgrid=c(0, 21), work.dir=work.dir)

## plot predictive for future patients in study 1 and 2,
for(i in c(5, 7)){
  hdpmPredict(j=1, r=0, i=i)
  hdpmPredict(j=2, r=0, i=i)
}

## part of the predictive corresponding to the idiosyncratic measure
for(i in c(5, 7)){
  hdpmPredict(j=1, r=1, i=i, work.dir=work.dir)
  hdpmPredict(j=2, r=1, i=i, work.dir=work.dir)
}
```

```
## post-process MCMC output for fitted mean curves
fits <- hdpmFitSetup(px=3,xgrid=c(0,21))

## plot fitted mean curves
par(mfrow=c(2,2),ask=T)
y <- as.matrix(read.table(Y,header=T)) # read in data
exs <- c(10:17,20:27,30:37,90:97)
for(i in exs){
  hdpmFit(fits,i=i)
  idx <- which(y[,1]==i-1)
  points(y[idx,2],y[idx,3],pch=19)
}

## End(Not run)
```

hdpmPredict	<i>Posterior predictive response</i>
-------------	--------------------------------------

Description

Plots posterior predictive response for assumed future patients. Must run `hdpmPredictSetup` first.

Usage

```
hdpmPredict(j = 1, r = 0,
            i = NULL,
            work.dir = NULL, plt = T)
```

Arguments

<code>j</code>	study
<code>r</code>	indicator for including (0) or not (1) the common measure.
<code>i</code>	index or vector of indices of future patients specified in the X in the earlier call to <code>hdpmPredictSetup</code> , i.e., row numbers in X.
<code>work.dir</code>	working directory where <code>hdpmPredictSetup</code> saved the posterior predictive simulations.
<code>plt</code>	indicator for plotting the predictive profiles.

Details

Finds and plots posterior predictive mean and (point-wise) standard deviation of future responses for the indicated future patients.

See [hdpm-package](#) for a statement of the probability model.

Value

Returns a list with

xgrid	grid on the time axis
e	matrix of posterior predictive means with each row reporting $n \times$ mean responses for one patient.
s	posterior predictive s.d.
x	matrix with the covariates of the future patients, with one line per patient.

Note

The only functions in the source code that are specific to the application in Mueller et al. (2004) are the definition of the top level sampling model in the file `lpost.c` and the functions for resampling (transition probabilities in the MCMC) for the random effects in the source file `hdpm.c`. Both source files are in the directory `src` of the package distribution. Users with experience in coding MCMC could appropriately change these functions and recompile the package for a different longitudinal mean function.

Author(s)

Peter Mueller

See Also

The functions in this package implement the application in *Mueller et al. (2004)*. Alternatively, the package `hdpmn` implements the same hierarchical mixture of Dirichlet process models for a multivariate normal sampling model and is useful for a wider class of applications.

Examples

```
## Not run:
require(hdpm)

## data files
data.dir <- system.file("data", package="hdpm")
work.dir <- paste(getwd(), "/tmp/", sep="")

## data files
Y <- file.path(data.dir, "CALGB.txt")
  ## data: responses
Z <- file.path(data.dir, "CALGBz.txt")
  ## random effects (initial values) and covariates
X <- file.path(data.dir, "CALGBz0.txt")
  ## same as Z, for future patients (for prediction)

## run MCMC -- save working files in work.dir
hdpm(Y=Y, Z=Z, px=3, q=15, cc=15, work.dir=work.dir)

## post-process MCMC output for predictive inference
## save posterior predictive summaries in work.dir
hdpmPredictSetup(X=X, px=3,
                 j=c(0, 1, 2, 1, 2), r=c(0, 0, 0, 1, 1),
                 xgrid=c(0, 21), work.dir=work.dir)
```

```
## plot predictive for future patients in study 1 and 2,
for(i in c(5,7)){
  hdpmPredict(j=1,r=0,i=i)
  hdpmPredict(j=2,r=0,i=i)
}

## part of the predictive corresponding to the idiosyncratic measure
for(i in c(5,7)){
  hdpmPredict(j=1,r=1,i=i, work.dir=work.dir)
  hdpmPredict(j=2,r=1,i=i, work.dir=work.dir)
}

## post-process MCMC output for fitted mean curves
fits <- hdpmFitSetup(px=3,xgrid=c(0,21))

## plot fitted mean curves
par(mfrow=c(2,2),ask=T)
y <- as.matrix(read.table(Y,header=T)) # read in data
exs <- c(10:17,20:27,30:37,90:97)
for(i in exs){
  hdpmFit(fits,i=i)
  idx <- which(y[,1]==i-1)
  points(y[idx,2],y[idx,3],pch=19)
}
## End(Not run)
```

Index

*Topic **kw1**

hdpm, [2](#)

*Topic **kw2**

hdpm, [2](#)

*Topic **package**

hdpm-package, [1](#)

hdpm, [2](#), [4](#), [10](#)

hdpm-package, [1](#)

hdpmFit, [8](#)

hdpmFitSetup, [5](#)

hdpmn, [2](#), [4](#), [7](#), [8](#), [11](#), [13](#)

hdpmPredict, [12](#)

hdpmPredictSetup, [9](#)

hdpm-package, [10](#), [12](#)