Package 'ensembleMOS'

February 19, 2015

Type Package

Title Ensemble Model Output Statistics

Version 0.7

Date 2013-03-17

Author RA Yuen, Tilmann Gneiting, Thordis Thorarinsdottir, Chris Fraley

Maintainer RA Yuen <bobyuen@umich.edu>

Depends R (>= 2.10.0), ensembleBMA, chron

Suggests fields, maps

Description Ensemble Model Output Statistics to create probabilistic forecasts from ensemble forecasts and weather observations.

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2013-03-17 20:01:24

R topics documented:

cdf . . . 2
controlMOSnormal . . . 3
crps . . . 5
emosFit . . . 6
ensembleMOS . . . 7
ensembleMOSnormal . . . 9
ensMOStest . . . 11
fitMOS . . . 12
fitMOSnormal . . . 14
quantileForecast . . . 15
trainingData . . . 17

Index **19**

cdf *Cummulative Distribution Function for ensemble forcasting models*

Description

Computes the cumulative distribution function (CDF) of an ensemble forecasting model at observation locations.

Usage

```
cdf(fit, ensembleData, values, dates = NULL, ...)
```

Arguments

`fit`	A model fit to ensemble forecasting data, obtained using `fitMOS` or `ensembleMOS`.
`ensembleData`	An `ensembleData` object that includes ensemble forecasts, verification observations and possibly dates. Missing values (indicated by `NA`) are allowed. This need not be the data used for the model `fit`, although it must include the same ensemble members.
`values`	The vector of desired values at which the CDF of the ensemble forecasting model is to be evaluated.
`dates`	The dates for which the CDF will be computed. These dates must be consistent with `fit` and `ensembleData`. The default is to use all of the dates in `fit`. The dates are ignored if `fit` originates from `fitMOS`, which also ignores date information.
`...`	Included for generic function compatibility.

Details

This method is generic, and can be applied to any ensemble forecasting model.
Note the model may have been applied to a power transformation of the data, but that information is included in the input `fit`, and the output is transformed appropriately.

Value

A matrix of probabilities corresponding to the CDF at the desired values. Useful for determining propability of freezing, precipitation, etc.

References

T. Gneiting, A. E. Raftery, A. H. Westveld and T. Goldman, Calibrated probabilistic forecasting using ensemble model output statistics and minimum CRPS estimation. *Monthly Weather Review* 133:1098–1118, 2005.

See Also

ensembleMOS, fitMOS, quantileForecast

Examples

```
data(ensMOStest)

ensMemNames <- c("gfs","cmcg","eta","gasp","jma","ngps","tcwb","ukmo")

tempTestData <- ensembleData( forecasts = ensMOStest[,ensMemNames],
                              dates = ensMOStest[,"vdate"],
                              observations = ensMOStest[,"obs"],
                              station = ensMOStest[,"station"],
                              forecastHour = 48,
                              initializationTime = "00")


tempTestFit <- ensembleMOSnormal( tempTestData, trainingDays = 30)


tempTestForc <- quantileForecast( tempTestFit, tempTestData)
range(tempTestForc)

tempTestCDF <- cdf( tempTestFit, tempTestData,
                    values = seq(from=277, to=282, by = 1))

tempTestCDF
```

---

controlMOSnormal *Control parameters for Gaussian (normal) EMOS models*

Description

Specifies a list of values controling the Gaussian (normal) EMOS fit of ensemble forecasts.

Usage

```
controlMOSnormal(scoringRule = c("crps", "log"),
                 coefRule = c("square", "none", "positive"),
                 varRule = c("square", "none"),
                 start = list(a = NULL, B = NULL,
                              c = NULL, d = NULL),
                 maxIter = Inf)
```

Arguments

scoringRule	The scoring rule to be used in optimum score estimation. Options are "crps" for the continuous ranked probability score and "log" for the logarithmic score.

coefRule	Method to control non-negativity of regression estimates. Options are: • ``square'' - The EMOS coefficients are parameterized as squares and thus gauranteed to be non-negative. • ``positive'' finds non-negative coefficents iteratively by setting negative estimates at the current iteration to zero. • ``none'' no restriction on the coefficient estimates.
varRule	Method to control non-negativity of the variance parameters. Options ``square'' and ``none'' are the same as in coefRule.
start	A list of starting parameters, a, B, c and d specifying initial values for the intercept coefficient and variance parameters supplied to optim. See details.
maxIter	An integer specifying the upper limit of the number of iterations used to fit the model.

Details

Given an ensemble of size m: $X_1, \ldots, X_m$, the following Gaussian model is fit by ensembleMOSnormal

$$Y_t \sim \mathcal{N}\left(a + b_1 X_1 + \cdots + b_m X_m, c + dS^2\right)$$

B is the array of fitted regression coefficients $b_1, \ldots, b_m$ for each date. See ensembleMOSnormal.

Value

A list whose components are the input arguments and their assigned values.

References

T. Gneiting, A. E. Raftery, A. H. Westveld and T. Goldman, calibrated probabilistic forecasting using ensemble model output statistics and minimum CRPS estimation. *Monthly Weather Review* 133:1098–1118, 2005.

See Also

ensembleMOSnormal, fitMOSnormal

Examples

```
data(ensMOStest)

ensMemNames <- c("gfs","cmcg","eta","gasp","jma","ngps","tcwb","ukmo")

tempTestData <- ensembleData( forecasts = ensMOStest[,ensMemNames],
                              dates = ensMOStest["vdate"],
                              observations = ensMOStest["obs"],
                              station = ensMOStest["station"],
                              forecastHour = 48,
                              initializationTime = "00")
```

```
tempTestFit1 <- ensembleMOSnormal(tempTestData, trainingDays = 30,
                          control = controlMOSnormal(maxIter = as.integer(100),
                          coefRule= "positive", varRule = "square"))
```

---

crps *Continuous Ranked Probability Score*

Description

Computes the continuous ranked probability score (CRPS) for univariate ensemble forecasting models.

Usage

```
crps(fit, ensembleData, dates=NULL, nSamples=NULL, seed=NULL, ...)
```

Arguments

`fit`	A model fit to ensemble forecasting data, obtained using `fitMOS` or `ensembleMOS`.
`ensembleData`	An `ensembleData` object that includes ensemble forecasts, verification observations and possibly dates. Missing values (indicated by `NA`) are allowed. This need not be the data used for the model `fit`, although it must include the same ensemble members.
`nSamples`	The number of simulation samples for CRPS via simulation.
`seed`	Argument to `set.seed` for random number generation in simulation.
`dates`	The dates for which the CRPS will be computed. These dates must be consistent with `fit` and `ensembleData`. The default is to use all of the dates in `fit`. The dates are ignored if `fit` originates from `fitMOS`, which also ignores date information.
`...`	Included for generic function compatibility.

Details

These methods are generic, and can be applied to all ensemble forecasting models.

For Gaussian (normal) models for temperature and pressure, calculation of the CRPS via simulation is not used and analytic computation is the only option.

Value

`crps` is a vector giving the CRPS for each instance in the data.

References

T. Gneiting and A. E. Raftery, Strictly proper scoring rules, prediction and estimation, *Journal of the American Statistical Association* 102:359–378 2007.

See Also

ensembleMOS, fitMOS

Examples

```
data(ensMOStest)

ensMemNames <- c("gfs","cmcg","eta","gasp","jma","ngps","tcwb","ukmo")

tempTestData <- ensembleData( forecasts = ensMOStest[,ensMemNames],
                              dates = ensMOStest[,"vdate"],
                              observations = ensMOStest[,"obs"],
                              station = ensMOStest[,"station"],
                              forecastHour = 48,
                              initializationTime = "00")


tempTestFit <- ensembleMOSnormal( tempTestData, trainingDays = 30)

crpsValues <- crps( tempTestFit, tempTestData)
mean(crpsValues)
```

---

emosFit *EMOS model fit.*

Description

A Gaussian (normal) EMOS model fitted to sampe data.

Usage

```
data(emosFit)
```

Format

An ensembleMOSnormal object fitted to the data set srft for forcast date 2004012900.

Examples

```
## Not run: # R check
  data(srft)

  labels <- c("CMCG","ETA","GASP","GFS","JMA","NGPS","TCWB","UKMO")

  srftData <- ensembleData(forecasts = srft[,labels],
                        dates = srft$date, observations = srft$obs,
                        latitude = srft$lat, longitude = srft$lon,
                        forecastHour = 48, initializationTime = "00")
  emosFit <- ensembleMOS(srftData, date = "2004012900", trainingDays = 25,
                        model = "normal")

  data(emosFit)

  emosForc <- quantileForecast(emosFit, srftData, date = "2004012900",
                               quantiles = c(.1, .5, .9))

  data(srftGrid)

  memberLabels <- c("CMCG","ETA","GASP","GFS","JMA","NGPS","TCWB","UKMO")
  srftGridData <- ensembleData(forecasts = srftGrid[,memberLabels],
                            latitude = srftGrid[,"latitude"],
                            longitude = srftGrid[,"longitude"],
                            forecastHour = 48, initializationTime = "00")

  gridForc <- quantileForecast(emosFit, srftGridData, date = "2004012900",
                            quantiles = c( .1, .5, .9))

  library(fields)

  plotProbcast(gridForc[,"0.5"],lon=srftGridData$lon,
            lat=srftGridData$lat, type="image",
            col=rev(rainbow(100,start=0,end=0.85)))
  title("Median Grid Forecast for Surface Temperature", cex = 0.5)

  probFreeze <- cdf(emosFit, srftGridData,  date = "2004012900",
                value = 273.15)

  plotProbcast(probFreeze, lon=srftGridData$lon, lat=srftGridData$lat,
            type="image",col=gray((32:0)/32))
  title("Probability of Freezing", cex = 0.5)


## End(Not run)
```

---

ensembleMOS *EMOS modeling*

Description

Fits a EMOS model to ensemble forecasts. Allows specification of a model, training rule, and forecasting dates.

Usage

```
ensembleMOS(ensembleData, trainingDays, consecutive = FALSE,
            dates = NULL, control = NULL, warmStart = FALSE,
            model = NULL, exchangeable = NULL)
```

Arguments

`ensembleData`	An `ensembleData` object including ensemble forecasts with the corresponding verifying observations and their dates. Missing values (indicated by `NA`) are allowed.
`trainingDays`	An integer giving the number of time steps (e.g. days) in the training period. There is no default.
`consecutive`	If `TRUE` then the sequence of dates in the training set are treated as consecutive, i.e. date gaps are ignored
`dates`	The dates for which EMOS forecasting models are desired. By default, this will be all dates in `ensembleData` for which modeling is allowed given the training rule.
`control`	A list of control values for the fitting functions. The default is `controlMOSnormal()` for Gaussian (normal) models.
`warmStart`	If `TRUE`, then starting values for parameters in optimization are set to the estimates of the preceding date's fit.
`model`	A character string describing the EMOS model to be fit. Current choices are `"normal"`, typically used for temperature or pressure data. For specific details on model fitting see ensembleMOSnormal
`exchangeable`	A numeric or character vector or factor indicating groups of ensemble members that are exchangeable (indistinguishable). The model fit will have equal weights and parameters within each group. The default determines exchangeability from `ensembleData`.

Details

If dates are specified in `dates` that cannot be forecast with the training rule, the corresponding EMOS model parameter outputs will be missing (`NA`) but not `NULL`.
The training rule uses the number of days corresponding to its `length` regardless of whether or not the dates are consecutive.

Value

A list with the following output components:

training	A list containing information on the training length and lag and the number of instances used for training for each modeling date.
a	A vector of fitted EMOS intercept parameters for each date.
B	A matrix of fitted EMOS coefficients for each date.
c,d	Vectors of the fitted variance parameters for each date, see ensembleMOSnormal for details.

References

T. Gneiting, A. E. Raftery, A. H. Westveld and T. Goldman, Calibrated probabilistic forecasting using ensemble model output statistics and minimum CRPS estimation. *Monthly Weather Review* 133:1098–1118, 2005.

See Also

trainingData, ensembleMOSnormal,

controlMOSnormal,

Examples

```
data(ensMOStest)

ensMemNames <- c("gfs","cmcg","eta","gasp","jma","ngps","tcwb","ukmo")

tempTestData <- ensembleData( forecasts = ensMOStest[,ensMemNames],
                              dates = ensMOStest["vdate"],
                              observations = ensMOStest["obs"],
                              station = ensMOStest["station"],
                              forecastHour = 48,
                              initializationTime = "00")

tempTestFit <- ensembleMOS( tempTestData, trainingDays = 30,
                            model = "normal")

## Same as
## tempTestFit <- ensembleMOSnormal( tempTestData, trainingDays = 30)
```

---

ensembleMOSnormal *Gaussian (normal) EMOS modeling*

Description

Fits a Gaussian (normal) EMOS model to ensemble forecasts for specified dates.

Usage

```
ensembleMOSnormal(ensembleData, trainingDays, consecutive = FALSE,
                  dates = NULL, control = controlMOSnormal(),
                  warmStart = FALSE, exchangeable = NULL)
```

Arguments

ensembleData	An ensembleData object including ensemble forecasts with the corresponding verifying observations and their dates. Missing values (indicated by NA) are allowed.
trainingDays	An integer giving the number of time steps (e.g. days) in the training period. There is no default.
consecutive	If TRUE then the sequence of dates in the training set are treated as consecutive, i.e. date gaps are ignored.
dates	The dates for which EMOS forecasting models are desired. By default, this will be all dates in ensembleData for which modeling is allowed given the training rule.
control	A list of control values for the fitting functions. The defaults are given by the function controlMOSnormal.
warmStart	If TRUE, then starting values for parameters in optimization are set to the estimates of the preceding date's fit.
exchangeable	A numeric or character vector or factor indicating groups of ensemble members that are exchangeable (indistinguishable). The modeling will have equal parameters within each group. The default determines exchangeability from ensembleData.

Details

Given an ensemble forecast of size m: $X_1, \ldots, X_m$, the following Gaussian predictive distribution is fit by ensembleMOSnormal

$$Y \sim \mathcal{N}\left(a + b_1 X_1 + \cdots + b_m X_m, c + dS^2\right)$$

B is a vector of fitted regression coefficients: $b_1, \ldots, b_m$. Specifically, $a, b_1, \ldots, b_m, c, d$ are fitted to optimize control$scoringRule over the specified training period using optim with method = "BFGS".

Value

A list with the following output components:

training	A list containing information on the training length and lag and the number of instances used for training for each modeling date.
a	A vector of fitted EMOS intercept parameters for each date.
B	A matrix of fitted EMOS coefficients for each date.
c,d	Vectors of the fitted variance parameters for each date, see details.

References

T. Gneiting, A. E. Raftery, A. H. Westveld and T. Goldman, Calibrated probabilistic forecasting using ensemble model output statistics and minimum CRPS estimation. *Monthly Weather Review* 133:1098–1118, 2005.

See Also

`controlMOSnormal`, `fitMOSnormal`

Examples

```
data(ensMOStest)

ensMemNames <- c("gfs","cmcg","eta","gasp","jma","ngps","tcwb","ukmo")

tempTestData <- ensembleData( forecasts = ensMOStest[,ensMemNames],
                              dates = ensMOStest[,"vdate"],
                              observations = ensMOStest[,"obs"],
                              station = ensMOStest[,"station"],
                              forecastHour = 48,
                              initializationTime = "00")

tempTestFit <- ensembleMOSnormal( tempTestData, trainingDays = 30)
```

---

ensMOStest *Ensemble MOS Test Data Set*

Description

This data set gives 48-hour ahead forecasts for 2-m temperature over the last 24 hours at SeaTac (KSEA) and Portland (PDX) airports in 2007/2008 initialized at 00 hours UTC using a 12km grid. The forecasts are based on an 8 member version of the University of Washington mesoscale ensemble (Grimit and Mass 2002; Eckel and Mass 2005).

Format

A data frame with 66 rows and 16 columns:
`idate` the initialization date of each forecast/observation, format YYYYMMDDHH (categorical).
`vdate` the validation date of each forecast/observation, format YYYYMMDDHH (categorical).
`latitude` the latitude of each forecast/observation (numeric).
`longitude` the longitude of each forecast/observation (numeric).
`elevation` the elevation (in meters) above sea level (numeric).
`station` weather station identifier (categorical).
`network` weather network identifier (categorical). `gfs,cmcg,eta,gasp,jma,ngps,tcwb` forecasts from the 8 members of the ensemble (numeric). `obs` observed values for the weather parameters.

Details

Temperature is given in degrees Kelvin.
This is a small dataset provided for the purposes of testing. Typically forecasting would be performed on much larger datasets.

References

F. A. Eckel and C. F. Mass, Effective mesoscale, short-range ensemble forecasting, *Weather and Forecasting* 20:328–350, 2005.

E. P. Grimit and C. F. Mass, Initial results of a mesoscale short-range ensemble forecasting system over the Pacific Northwest, *Weather and Forecasting* 17:192–205, 2002.

Examples

```
data(ensMOStest)

ensMemNames <- c("gfs","cmcg","eta","gasp","jma","ngps","tcwb","ukmo")

tempTestData <- ensembleData( forecasts = ensMOStest[,ensMemNames],
                              dates = ensMOStest[,"vdate"],
                              observations = ensMOStest[,"obs"],
                              station = ensMOStest[,"station"],
                              forecastHour = 48,
                              initializationTime = "00")

tempTestFit <- ensembleMOSnormal( tempTestData, trainingDays = 30)
```

---

fitMOS *EMOS model fit to a training set*

Description

Fits Ensemble Model Output Statistics to a given training set.

Usage

```
fitMOS(ensembleData, control = NULL, model = NULL,
       exchangeable = NULL)
```

Arguments

ensembleData	An ensembleData object including ensemble forecasts and verification observations. Missing values (indicated by NA) are allowed. Dates are ignored if they are included. This is the training set for the model.
control	A list of control values for the fitting functions. The default is controlMOSnormal() for Gaussian (normal) models.

model	A character string describing the EMOS model to be fit. Current choices are "normal", typically used for temperature or pressure data. For specific details on model fitting see fitMOSnormal
exchangeable	A numeric or character vector or factor indicating groups of ensemble members that are exchangeable (indistinguishable). The model fit will have equal weights and parameters within each group. The default determines exchangeability from ensembleData.

Value

A list with the following output components:

a	The fitted intercept.
B	The fitted EMOS coefficients.
c,d	The fitted variance parameters for each date, see details.

References

T. Gneiting, A. E. Raftery, A. H. Westveld and T. Goldman, Calibrated probabilistic forecasting using ensemble model output statistics and minimum CRPS estimation. *Monthly Weather Review* 133:1098–1118, 2005.

See Also

fitMOSnormal controlMOSnormal

Examples

```
data(ensMOStest)

ensNames <- c("gfs","cmcg","eta","gasp","jma","ngps","tcwb","ukmo")

tempTestData <- ensembleData( forecasts = ensMOStest[,ensNames],
                              observations = ensMOStest[,"obs"],
                              station = ensMOStest[,"station"],
                              dates = ensMOStest[,"vdate"],
                              forecastHour = 48,
                              initializationTime = "00")

tempTrain <- trainingData( tempTestData, trainingDays = 30,
                           date  = "2008010100")

tempTrainFit <- fitMOS( tempTrain, model = "normal")

## equivalent to
##    tempTrainFit <- fitMOSnormal( tempTrain)
```

`fitMOSnormal` *Gaussian (normal) EMOS model fit to a training set*

Description

Fits a Gaussian (normal) EMOS model to a given training set.

Usage

```
fitMOSnormal(ensembleData, control = controlMOSnormal(),
             exchangeable = NULL)
```

Arguments

`ensembleData`	An `ensembleData` object including ensemble forecasts and verification observations. Missing values (indicated by `NA`) are allowed. Dates are ignored if they are included. This is the training set for the model.
`control`	A list of control values for the fitting functions. The defaults are given by the function `controlMOSnormal`.
`exchangeable`	An optional numeric or character vector or factor indicating groups of ensemble members that are exchangeable (indistinguishable). The models have equal EMOS coefficients within each group. If supplied, this argument will override any specification of exchangeability in `ensembleData`.

Details

Given an ensemble forecast of size m: $X_1, \ldots, X_m$, the following Gaussian predictive distribution is fit by `fitMOSnormal`

$$Y \sim \mathcal{N}\left(a + b_1 X_1 + \cdots + b_m X_m, c + dS^2\right)$$

`B` is a vector of fitted regression coefficients: $b_1, \ldots, b_m$. Specifically, $a, b_1, \ldots, b_m, c, d$ are fitted to optimize `control$scoringRule` over the specified training set using `optim` with `method = "BFGS"`.

Value

A list with the following output components:

`a`	The fitted intercept.
`B`	The fitted EMOS coefficients.
`c,d`	The fitted variance parameters, see details.

References

T. Gneiting, A. E. Raftery, A. H. Westveld and T. Goldman, Calibrated probabilistic forecasting using ensemble model output statistics and minimum CRPS estimation. *Monthly Weather Review* 133:1098–1118, 2005.

See Also

controlMOSnormal, ensembleMOSnormal,

Examples

```
data(ensMOStest)

ensNames <- c("gfs","cmcg","eta","gasp","jma","ngps","tcwb","ukmo")

tempTestData <- ensembleData( forecasts = ensMOStest[,ensNames],
                              observations = ensMOStest[,"obs"],
                              station = ensMOStest[,"station"],
                              dates = ensMOStest[,"vdate"],
                              forecastHour = 48,
                              initializationTime = "00")

tempTrain <- trainingData( tempTestData, trainingDays = 30,
                           date = "2008010100")

tempTrainFit <- fitMOSnormal( tempTrain)

## equivalent to
##    tempTrainFit <- fitMOS( tempTrain, model = "normal")
```

---

quantileForecast *Quantile forecasts at observation locations*

Description

Computes quantiles for the probability distribution function (PDF) for ensemble forecasting models.

Usage

```
quantileForecast(fit, ensembleData, quantiles = 0.5, dates = NULL, ...)
```

Arguments

fit	A model fit to ensemble forecasting data.
ensembleData	An ensembleData object that includes ensemble forecasts, verification observations and possibly dates. Missing values (indicated by NA) are allowed. \ This need not be the data used for the model fit, although it must include the same ensemble members.
quantiles	The vector of desired quantiles for the PDF of the EMOS model.
dates	The dates for which the quantile forecasts will be computed. These dates must be consistent with fit and ensembleData. The default is to use all of the dates in fit. If ensembleData does not include dates, they will be inferred from fit and dates.

... Included for generic function compatibility.

Details

This method is generic, and can be applied to any ensemble forecasting model.
Note the model may have been applied to a power transformation of the data, but that information is included in the input `fit`, and the output is transformed appropriately.
This can be used to compute prediction intervals for the PDF.

Value

A matrix of forecasts corresponding to the desired quantiles.

References

T. Gneiting, A. E. Raftery, A. H. Westveld and T. Goldman, Calibrated probabilistic forecasting using ensemble model output statistics and minimum CRPS estimation. *Monthly Weather Review* 133:1098–1118, 2005.

See Also

ensembleMOS, fitMOS, cdf

Examples

```
data(ensMOStest)

ensMemNames <- c("gfs","cmcg","eta","gasp","jma","ngps","tcwb","ukmo")

tempTestData <- ensembleData( forecasts = ensMOStest[,ensMemNames],
                              dates = ensMOStest[,"vdate"],
                              observations = ensMOStest[,"obs"],
                              station = ensMOStest[,"station"],
                              forecastHour = 48,
                              initializationTime = "00")


tempTestFit <- ensembleMOSnormal( tempTestData, trainingDays = 30)


tempTestForc <- quantileForecast( tempTestFit, tempTestData)


## Not run:  # R check

data(srft)

labels <- c("CMCG","ETA","GASP","GFS","JMA","NGPS","TCWB","UKMO")

srftData <- ensembleData( forecasts = srft[ ,labels],
                          dates = srft$date,
```

```
                              observations = srft$obs,
                              latitude = srft$lat,
                              longitude = srft$lon,
                              forecastHour = 48,
                              initializationTime = "00")

  srftFit <- ensembleMOSnormal(srftData, date = "2004012900",
                               trainingDays = 25)

  data(srftGrid)

  srftGridData <- ensembleData(forecasts = srftGrid[ ,labels],
                               latitude = srftGrid$lat,
                               longitude = srftGrid$lon,
                               forecastHour = 48,
                               initializationTime = "00")

  srftGridForc <- quantileForecast( srftFit, srftGridData,
                    date = "2004012900")

## End(Not run)
```

`trainingData` *Extract Training Data*

Description

Extracts a subset of an `ensembleData` object corresponding to a given date and number of training days.

Usage

```
trainingData( ensembleData, trainingDays, consecutive = FALSE, date)
```

Arguments

`ensembleData`	An `ensembleData` object that includes ensemble forecasts, observations and dates.
`trainingDays`	An integer specifying the number of days in the training period.
`consecutive`	If `TRUE` then dates in training set are treated as consecutive, i.e. date gaps are ignored.
`date`	The date for which the training data is desired.

Value

An `ensembleData` object corresponding to the training data for the given date relative to `ensembleData`.

References

A. E. Raftery, T. Gneiting, F. Balabdaoui and M. Polakowski, Using Bayesian model averaging to calibrate forecast ensembles, *Monthly Weather Review* 133:1155-1174, 2005.

J. M. Sloughter, A. E. Raftery, T. Gneiting and C. Fraley, Probabilistic quantitative precipitation forecasting using Bayesian model averaging, *Monthly Weather Review* 135:3309–3320, 2007.

C. Fraley, A. E. Raftery, T. Gneiting and J. M. Sloughter, `ensembleBMA`: An `R` Package for Probabilistic Forecasting using Ensembles and Bayesian Model Averaging, Technical Report No. 516R, Department of Statistics, University of Washington, December 2008.
Available at: http://www.stat.washington.edu/research/reports/

C. Fraley, A. E. Raftery and T. Gneiting, Calibrating multi-model forecast ensembles with exchangeable and missing members using Bayesian model averaging, *Monthly Weather Review* 138:190-202, 2010.

See Also

`ensembleMOSnormal`, `fitMOSnormal`

Examples

```
data(ensMOStest)

ensNames <- c("gfs","cmcg","eta","gasp","jma","ngps","tcwb","ukmo")

tempTestData <- ensembleData( forecasts = ensMOStest[,ensNames],
                              observations = ensMOStest[,"obs"],
                              station = ensMOStest[,"station"],
                              dates = ensMOStest[,"vdate"],
                              forecastHour = 48,
                              initializationTime = "00")

tempTrain <- trainingData( tempTestData, trainingDays = 30,
                           date  = "2008010100")

tempTrainFit <- fitMOSnormal( tempTrain)
```

Index

∗Topic **datasets**
emosFit, 6
ensMOStest, 11
∗Topic **models**
cdf, 2
controlMOSnormal, 3
crps, 5
ensembleMOS, 7
ensembleMOSnormal, 9
fitMOS, 12
fitMOSnormal, 14
quantileForecast, 15
trainingData, 17

cdf, 2, *16*
controlMOSnormal, 3, *9–11*, *13*, *15*
crps, *5*

emosFit, 6
ensembleMemberLabels *(*ensembleMOS*)*, 7
ensembleMOS, *2, 3*, *5, 6*, 7, *16*
ensembleMOSnormal, *4*, *6*, *8, 9*, 9, *15*, *18*
ensembleNobs *(*ensembleMOS*)*, 7
ensembleObsLabels *(*ensembleMOS*)*, 7
ensembleSize *(*ensembleMOS*)*, 7
ensembleValidDates *(*ensembleMOS*)*, 7
ensembleVerifObs *(*ensembleMOS*)*, 7
ensMOStest, 11

fitMOS, *2, 3*, *5, 6*, 12, *16*
fitMOSnormal, *4*, *11*, *13*, 14, *18*

getExchangeable *(*ensembleMOS*)*, 7
getHH *(*ensembleMOS*)*, 7

matchEnsembleMembers *(*ensembleMOS*)*, 7
matchITandFH *(*ensembleMOS*)*, 7

quantileForecast, *3*, 15

srft, *6*

temp *(*ensMOStest*)*, 11
trainingData, *9*, 17