Package 'ensembleMOS'

July 22, 2025

Type Package

Title Ensemble Model Output Statistics

Version 0.8.2

Date 2018-03-21

Author RA Yuen, Sandor Baran, Chris Fraley, Tilmann Gneiting, Sebastian Lerch, Michael Scheuerer, Thordis Thorarinsdottir

Maintainer Sandor Baran <baran.sandor@inf.unideb.hu>

Depends R (>= 2.10.0), ensembleBMA, chron, evd

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 2018-03-22 09:50:57 UTC

Contents

`brierScore` *Brier Score*

Description

Computes the Brier score for the probability of exceedance of precipitation threshold values for univariate ensemble forecasting models.

Usage

```
brierScore(fit, ensembleData, thresholds, dates=NULL, ...)
```

Arguments

`fit`	A model fit to ensemble forecasting data, obtained using `fitMOS` or `ensembleMOS`. Only available for the censored and shifted gamma, and the censored generalized extreme value distribution model.
`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.
`thresholds`	Threshold values for which the probability of exceedance is evaluated, set to 0 to evaluate probability of precipitation forecasts.
`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

Note that the Brier scores are only available for EMOS models suitable for precipitation accumulation, i.e. the censored and shifted gamma, and the censored generalized extreme value distribution EMOS model.

Value

BScores is a vector giving the Brier scores 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("ensBMAtest", package = "ensembleBMA")

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

obs <- paste("PCP24","obs", sep = ".")
ens <- paste("PCP24", ensMemNames, sep = ".")
prcpTestData <- ensembleData(forecasts = ensBMAtest[,ens],
                             dates = ensBMAtest[,"vdate"],
                             observations = ensBMAtest[,obs],
                             station = ensBMAtest[,"station"],
                             forecastHour = 48,
                             initializationTime = "00")

prcpTestFitCSG0 <- ensembleMOScsg0(prcpTestData, trainingDays = 25,
                                   dates = "2008010100")

brierScore(prcpTestFitCSG0, ensembleData = prcpTestData,
           thresholds = 0)
```

---

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 obtained using fitMOS or ensembleMOS.

For the EMOS models that allow for point masses at 0, i.e. the censored and shifted gamma, and the censored generalized extreme value distribution EMOS model, the function contains an addition logical argument randomizeATzero that specifies whether the value of the CDF at zero should be chosen randomly from the interval between 0 and the value of the CDF at zero. The default choice if FALSE, setting randomizeATzero = TRUE is practical for computing randomized PIT values.

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("ensBMAtest", package = "ensembleBMA")

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

obs <- paste("T2", "obs", sep = ".")
ens <- paste("T2", ensMemNames, sep = ".")
tempTestData <- ensembleData(forecasts = ensBMAtest[,ens],
```

```
                              dates = ensBMAtest[,"vdate"],
                              observations = ensBMAtest[,obs],
                              station = ensBMAtest[,"station"],
                              forecastHour = 48,
                              initializationTime = "00")

tempTestFit <- ensembleMOS(tempTestData, trainingDays = 25,
                           model = "normal",
                           dates = "2008010100")

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

tempTestCDF
```

`controlMOScsg0` *Control parameters for censored and shifted gamma EMOS models*

Description

Specifies a list of values controling the censored and shifted gamma EMOS fit of ensemble forecasts.

Usage

```
controlMOScsg0(scoringRule = c("crps", "log"),
               optimRule = c("Nelder-Mead", "BFGS", "L-BFGS-B"),
               coefRule = c("square", "none", "positive"),
               varRule = c("square", "none"),
               start = list(a = NULL, B = NULL,
                            c = NULL, d = NULL, q = 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.
`optimRule`	Numerical optimization method to be supplied to `optim`. Options are "BFGS" for the Broyden-Fletcher-Goldfarb-Shanno algorithm and "Nelder-Mead" for the Nelder-Mead method, see optim for details. Note that these options are only available for `scoringRule = "log"`. In case of `scoringRule = "crps"`, the optimization method is set to "L-BFGS-B" by default.
`coefRule`	Method to control non-negativity of regression estimates. Options are: • `"square"` EMOS coefficients are parameterized as squares and thus guaranteed 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 scale parameters. Options "square" and "none" are the same as in coefRule.
start	A list of starting parameters, a, B, c, d and q 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

If no value is assigned to an argument, the first entry of the list of possibly choices will be used by default. Note that optimMethod options are only available for scoringRule = "log". In case of scoringRule = "crps", the optimization method is set to "L-BFGS-B" by default.

Given an ensemble of size m: $X_1, \ldots, X_m$, the following shifted gamma model left-censored at 0 is fit by ensembleMOScsg0:

$$Y \; Gamma_0(\kappa, \theta, q)$$

where $Gamma_0$ denotes the shifted gamma distribution left-censored at zero, with shape κ, scale θ and shift q. The model is parametrized such that the mean $\kappa\theta$ is a linear function $a + b_1 X_1 + \ldots + b_m X_m$ of the ensemble forecats, and the variance $\kappa\theta^2$ is a linear function of the ensemble mean $c + d\overline{f}$, see ensembleMOScsg0 for details.

Value

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

References

M. Scheuerer and T. M. Hamill, Statistical post-processing of ensemble precipitation forecasts by fitting censored, shifted gamma distributions. *Monthly Weather Review* 143:4578–4596, 2015.

S. Baran and D. Nemoda, Censored and shifted gamma distribution based EMOS model for probabilistic quantitative precipitation forecasting. *Environmetrics* 27:280–292, 2016.

See Also

ensembleMOScsg0, fitMOScsg0

Examples

```
data("ensBMAtest", package = "ensembleBMA")

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

obs <- paste("PCP24","obs", sep = ".")
ens <- paste("PCP24", ensMemNames, sep = ".")
prcpTestData <- ensembleData(forecasts = ensBMAtest[,ens],
                             dates = ensBMAtest[,"vdate"],
                             observations = ensBMAtest[,obs],
                             station = ensBMAtest[,"station"],
```

```
                              forecastHour = 48,
                              initializationTime = "00")

prcpTestFitCSG0 <- ensembleMOScsg0(prcpTestData, trainingDays = 25,
                                   dates = "2008010100",
                                   control = controlMOScsg0(maxIter = as.integer(100),
                                                            scoringRule = "log",
                                                            optimRule = "Nelder-Mead",
                                                            coefRule= "none",
                                                            varRule = "square"))
```

`controlMOSgev0` *Control parameters for censored generalized extreme value distribution EMOS models*

Description

Specifies a list of values controling the censored generalized extreme value distribution EMOS fit of ensemble forecasts.

Usage

```
controlMOSgev0(optimRule = c("Nelder-Mead", "L-BFGS-B", "BFGS"),
               coefRule = c("square", "none", "positive"),
               varRule = c("square", "none"),
               start = list(a = NULL, B = NULL,
                            s = NULL, c = NULL,
                            d = NULL, q = NULL),
               maxIter = Inf)
```

Arguments

`optimRule`	Numerical optimization method to be supplied to `optim`. Options are "BFGS" for the Broyden-Fletcher-Goldfarb-Shanno algorithm, "L-BFGS-B" for a constrained version thereof, and "Nelder-Mead" for the Nelder-Mead method, see optim for details.
`coefRule`	Method to control non-negativity of regression estimates. Options are: • `"square"` EMOS coefficients are parameterized as squares and thus guaranteed 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 scale parameters. Options `"square"` and `"none"` are the same as in `coefRule`.
`start`	A list of starting parameters, `a`, `B`, `s`, `c`, `d` and `q` specifying initial values for the location, scale and shape coefficients supplied to `optim`. See details.

`maxIter` An integer specifying the upper limit of the number of iterations used to fit the model.

Details

Note that only minimum CRPS estimation is available and chosen by default.

If no value is assigned to an argument, the first entry of the list of possibly choices will be used by default.

Given an ensemble of size m: $X_1, \ldots, X_m$, the following generalized extreme value distribution EMOS model left-censored at 0 is fit by `ensembleMOSgev0`:

$$Y \; GEV_0(\mu, \sigma, q)$$

where GEV_0 denotes the generalized extreme value distribution left-censored at zero, with location μ, scale σ and shape q. The model is parametrized such that the mean m is a linear function $a + b_1X_1 + \ldots + b_mX_m + sp_0$ of the ensemble forecats, where p_0 denotes the ratio of ensemble forecasts that are exactly 0, and the shape parameter σ is a linear function of the ensemble variance $c + dMD(X_1, \ldots, X_m)$, where $MD(X_1, \ldots, X_m)$ is Gini's mean difference. See ensembleMOSgev0 for details.

Value

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

References

M. Scheuerer, Probabilistic quantitative precipitation forecasting using ensemble model output statistics. *Quarterly Journal of the Royal Meteorological Society* 140:1086–1096, 2014.

See Also

`ensembleMOScsg0`, `fitMOScsg0`

Examples

```
data("ensBMAtest", package = "ensembleBMA")

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

obs <- paste("PCP24","obs", sep = ".")
ens <- paste("PCP24", ensMemNames, sep = ".")
prcpTestData <- ensembleData(forecasts = ensBMAtest[,ens],
                             dates = ensBMAtest[,"vdate"],
                             observations = ensBMAtest[,obs],
                             station = ensBMAtest[,"station"],
                             forecastHour = 48,
                             initializationTime = "00")

prcpTestFitGEV0 <- ensembleMOSgev0(prcpTestData, trainingDays = 25,
                        dates = "2008010100",
```

```
                    control = controlMOSgev0(maxIter = as.integer(100),
                       optimRule = "Nelder-Mead",
                       coefRule= "none",
                       varRule = "square"))
```

`controlMOSlognormal` *Control parameters for log-normal EMOS models*

Description

Specifies a list of values controling the log-normal EMOS fit of ensemble forecasts.

Usage

```
controlMOSlognormal(scoringRule = c("crps", "log"),
                    optimRule = c("BFGS","Nelder-Mead"),
                    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.
`optimRule`	Numerical optimization method to be supplied to `optim`. Options are "BFGS" for the Broyden-Fletcher-Goldfarb-Shanno algorithm and "Nelder-Mead" for the Nelder-Mead method, see optim for details.
`coefRule`	Method to control non-negativity of regression estimates. Options are: • `"square"` EMOS coefficients are parameterized as squares and thus guaranteed 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 scale parameters supplied to `optim`. See details.
`maxIter`	An integer specifying the upper limit of the number of iterations used to fit the model.

Details

If no value is assigned to an argument, the first entry of the list of possibly choices will be used by default. Given an ensemble of size m: $X_1, \ldots, X_m$, the following log-normal model is fit by `ensembleMOSlognormal`:

$$Y \; LN(\mu, \sigma)$$

where LN denotes the log-normal distrbution with `meanlog` parameter μ and `scalelog` parameter σ, see Lognormal. The model is parametrized such that the mean value of the log-normal distribution is a linear function $a + b_1 X_1 + \ldots + b_m X_m$ of the ensemble forecats, and the variance is a linear function $c + dS^2$. For transformations between μ, σ and mean and variance of the log-normal distribution, see Baran and Lerch (2015). See ensembleMOSlognormal for details.

Note that in case of `scoringRule = "log"`, forecast cases in the training period with observation values of 0 are ignored in the model estimation as 0 is not included in the support of the log-normal distribution.

Value

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

References

S. Baran and S. Lerch, Log-normal distribution based Ensemble Model Output Statistics models for probabilistic wind-speed forecasting. *Quarterly Journal of the Royal Meteorological Society* 141:2289–2299, 2015.

See Also

`ensembleMOSlognormal`, `fitMOSlognormal`

Examples

```
data("ensBMAtest", package = "ensembleBMA")

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

obs <- paste("MAXWSP10","obs", sep = ".")
ens <- paste("MAXWSP10", ensMemNames, sep = ".")
windTestData <- ensembleData(forecasts = ensBMAtest[,ens],
                             dates = ensBMAtest[,"vdate"],
                             observations = ensBMAtest[,obs],
                             station = ensBMAtest[,"station"],
                             forecastHour = 48,
                             initializationTime = "00")

windTestFitLN <- ensembleMOSlognormal(windTestData, trainingDays = 25,
                                      dates = "2008010100",
                                 control = controlMOSlognormal(maxIter = as.integer(100),
                                                               scoringRule = "log",
                                                               optimRule = "BFGS",
```

```
                                                coefRule= "none",
                                                varRule = "square"))
```

---

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"),
                 optimRule = c("BFGS","Nelder-Mead"),
                 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.
optimRule	Numerical optimization method to be supplied to optim. Options are "BFGS" for the Broyden-Fletcher-Goldfarb-Shanno algorithm and "Nelder-Mead" for the Nelder-Mead method, see optim for details.
coefRule	Method to control non-negativity of regression estimates. Options are: • "square" EMOS coefficients are parameterized as squares and thus guaranteed 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

If no value is assigned to an argument, the first entry of the list of possibly choices will be used by default. Given an ensemble of size m: $X_1, \ldots, X_m$, the following Gaussian model is fit by `ensembleMOSnormal`:

$$Y \; N(a + b_1 X_1 + ... + b_m X_m, c + dS^2).$$

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

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("ensBMAtest", package = "ensembleBMA")

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

obs <- paste("T2", "obs", sep = ".")
ens <- paste("T2", ensMemNames, sep = ".")
tempTestData <- ensembleData(forecasts = ensBMAtest[,ens],
                             dates = ensBMAtest[,"vdate"],
                             observations = ensBMAtest[,obs],
                             station = ensBMAtest[,"station"],
                             forecastHour = 48,
                             initializationTime = "00")

tempTestFit <- ensembleMOSnormal(tempTestData, trainingDays = 25,
                                 dates = "2008010100",
                                 control = controlMOSnormal(maxIter = as.integer(100),
                                                            scoringRule = "log",
                                                            optimRule = "BFGS",
                                                            coefRule= "none",
                                                            varRule = "square"))
```

`controlMOStruncnormal` *Control parameters for truncated normal EMOS models*

Description

Specifies a list of values controling the truncated normal EMOS fit of ensemble forecasts.

Usage

```
controlMOStruncnormal(scoringRule = c("crps", "log"),
                      optimRule = c("BFGS","Nelder-Mead"),
                      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.
`optimRule`	Numerical optimization method to be supplied to `optim`. Options are "BFGS" for the Broyden-Fletcher-Goldfarb-Shanno algorithm and "Nelder-Mead" for the Nelder-Mead method, see optim for details.
`coefRule`	Method to control non-negativity of regression estimates. Options are: • `"square"` EMOS coefficients are parameterized as squares and thus guaranteed 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 scale 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

If no value is assigned to an argument, the first entry of the list of possibly choices will be used by default.

Given an ensemble of size m: $X_1, \ldots, X_m$, the following truncated normal model is fit by `ensembleMOStruncnormal`:

$$Y \; N_0(a + b_1 X_1 + ... + b_m X_m, c + dS^2),$$

where N_0 denotes the normal distribution truncated at zero, with location $a + b_1X_1 + ... + b_mX_m$ and squared scale $c + dS^2$. B is a vector of fitted regression coefficients $b_1, \ldots, b_m$. See ensembleMOStruncnormal for details.

Value

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

References

T. L. Thorarinsdottir and T. Gneiting, Probabilistic forecasts of wind speed: Ensemble model output statistics by using heteroscedastic censored regression. *Journal of the Royal Statistical Society Series A* 173:371–388, 2010.

See Also

ensembleMOStruncnormal, fitMOStruncnormal

Examples

```
data("ensBMAtest", package = "ensembleBMA")

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

obs <- paste("MAXWSP10","obs", sep = ".")
ens <- paste("MAXWSP10", ensMemNames, sep = ".")
windTestData <- ensembleData(forecasts = ensBMAtest[,ens],
                             dates = ensBMAtest[,"vdate"],
                             observations = ensBMAtest[,obs],
                             station = ensBMAtest[,"station"],
                             forecastHour = 48,
                             initializationTime = "00")

windTestFitTN <- ensembleMOStruncnormal(windTestData, trainingDays = 25,
                                        dates = "2008010100",
                                  control = controlMOStruncnormal(maxIter = as.integer(100),
                                                                  scoringRule = "log",
                                                                   optimRule = "BFGS",
                                                                     coefRule= "none",
                                                                  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, ...)
```

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.
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. Missing values in forecasts and/or observations result in NA values in the CRPS vector.

Value

crps is a matrix giving the CRPS for each instance in the data for both the raw ensemble and the probabilistic forecast.

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("ensBMAtest", package = "ensembleBMA")

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

obs <- paste("T2", "obs", sep = ".")
ens <- paste("T2", ensMemNames, sep = ".")
tempTestData <- ensembleData(forecasts = ensBMAtest[,ens],
                             dates = ensBMAtest[,"vdate"],
                             observations = ensBMAtest[,obs],
                             station = ensBMAtest[,"station"],
                             forecastHour = 48,
```

```
                              initializationTime = "00")

tempTestFit <- ensembleMOS(tempTestData, trainingDays = 25,
                           dates = "2008010100",
                           model = "normal")

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

`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 corresponding control function has to be chosen in accordance with the selected `model`. For the Gaussian (normal) EMOS model see controlMOSnormal, for the truncated normal model see controlMOStruncnormal, for the log-normal model see controlMOSlognormal, for the censored and shifted gamma model see controlMOScsg0, and for the censored generalized extreme value distribution model see controlMOSgev0.
`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), `"truncnormal"` (typically used for wind speed data), `"lognormal"` (typically used for wind speed data), `"csg0"` (typically used for precipitation accumulation data), and `"gev0"` (typically used for precipitation accumulation data). For specific details on model fitting see ensembleMOSnormal, ensembleMOStruncnormal, ensembleMOSlognormal, ensembleMOScsg0, or ensembleMOSgev0.

`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 containing information on the training (length, lag and the number of instances used for training for each modeling date), the exchangeability, and vectors and/or matrics containing the estimated regression and variance coefficient values depending on the specified `model`.

References

Gaussian (normal) EMOS model:
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.

Truncated normal EMOS model:
T. L. Thorarinsdottir and T. Gneiting, Probabilistic forecasts of wind speed: Ensemble model output statistics by using heteroscedastic censored regression. *Journal of the Royal Statistical Society Series A* 173:371–388, 2010.

Log-normal EMOS model:
S. Baran and S. Lerch, Log-normal distribution based Ensemble Model Output Statistics models for probabilistic wind-speed forecasting. *Quarterly Journal of the Royal Meteorological Society* 141:2289–2299, 2015.

Censored and shifted gamma EMOS model:
M. Scheuerer and T. M. Hamill, Statistical post-processing of ensemble precipitation forecasts by fitting censored, shifted gamma distributions. *Monthly Weather Review* 143:4578–4596, 2015.
S. Baran and D. Nemoda, Censored and shifted gamma distribution based EMOS model for probabilistic quantitative precipitation forecasting. *Environmetrics* 27:280–292, 2016.

Censored generalized extreme value distribution EMOS model:
M. Scheuerer, Probabilistic quantitative precipitation forecasting using ensemble model output statistics. *Quarterly Journal of the Royal Meteorological Society* 140:1086–1096, 2014.

See Also

trainingData, ensembleMOSnormal, ensembleMOStruncnormal, ensembleMOSlognormal, ensembleMOScsg0, ensembleMOSgev0, controlMOSnormal, controlMOStruncnormal, controlMOSlognormal, controlMOScsg0, controlMOSgev0,

Examples

```
data("ensBMAtest", package = "ensembleBMA")

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

obs <- paste("T2", "obs", sep = ".")
ens <- paste("T2", ensMemNames, sep = ".")
tempTestData <- ensembleData(forecasts = ensBMAtest[,ens],
                             dates = ensBMAtest[,"vdate"],
                             observations = ensBMAtest[,obs],
                             station = ensBMAtest[,"station"],
                             forecastHour = 48,
                             initializationTime = "00")

tempTestFit <- ensembleMOS(tempTestData, trainingDays = 25,
                           model = "normal")

 ## Same as
 ## tempTestFit <- ensembleMOSnormal(tempTestData, trainingDays = 25)
```

---

ensembleMOScsg0 *Censored and shifted gamma EMOS modeling*

Description

Fits a censored and shifted gamma EMOS model to ensemble forecasts for specified dates.

Usage

```
ensembleMOScsg0(ensembleData, trainingDays, consecutive = FALSE,
                dates = NULL, control = controlMOScsg0(),
                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 specified via the function controlMOScsg0. For details and default values, see controlMOScsg0.
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 of size m: $X_1, \ldots, X_m$, the following shifted gamma model left-censored at 0 is fit by ensembleMOScsg0:

$$Y \; Gamma_0(\kappa, \theta, q)$$

where $Gamma_0$ denotes the shifted gamma distribution left-censored at zero, with shape κ, scale θ and shift q. The model is parametrized such that the mean $\kappa\theta$ is a linear function $a + b_1 X_1 + \ldots + b_m X_m$ of the ensemble forecats, and the variance $\kappa\theta^2$ is a linear function of the ensemble mean $c + d\overline{f}$, see Baran and Nemoda (2016) for details.

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

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	The fitted parameters for the variance, see details.
q	Fitted shift parameter, see details.

References

M. Scheuerer and T. M. Hamill, Statistical post-processing of ensemble precipitation forecasts by fitting censored, shifted gamma distributions. *Monthly Weather Review* 143:4578–4596, 2015.

S. Baran and D. Nemoda, Censored and shifted gamma distribution based EMOS model for probabilistic quantitative precipitation forecasting. *Environmetrics* 27:280–292, 2016.

See Also

controlMOScsg0, fitMOScsg0

Examples

```
data("ensBMAtest", package = "ensembleBMA")

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

obs <- paste("PCP24","obs", sep = ".")
ens <- paste("PCP24", ensMemNames, sep = ".")
prcpTestData <- ensembleData(forecasts = ensBMAtest[,ens],
                             dates = ensBMAtest[,"vdate"],
                             observations = ensBMAtest[,obs],
                             station = ensBMAtest[,"station"],
                             forecastHour = 48,
                             initializationTime = "00")

fitDates <- c("2008010100", "2008010200")
prcpTestFitGEV0 <- ensembleMOSgev0(prcpTestData, trainingDays = 25,
                                   dates = fitDates)
```

---

ensembleMOSgev0 *Censored generalized extreme value distribution EMOS modeling*

Description

Fits a Censored generalized extreme value distribution EMOS model to ensemble forecasts for specified dates.

Usage

```
ensembleMOSgev0(ensembleData, trainingDays, consecutive = FALSE,
                dates = NULL, control = controlMOSgev0(),
                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 specified via the function controlMOSgev0. For details and default values, see controlMOSgev0.
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 of size m: $X_1, \ldots, X_m$, the following generalized extreme value distribution EMOS model left-censored at 0 is fit by ensembleMOSgev0:

$$Y \; GEV_0(\mu, \sigma, q)$$

where GEV_0 denotes the generalized extreme value distribution left-censored at zero, with location μ, scale σ and shape q. The model is parametrized such that the mean m is a linear function $a + b_1X_1 + \ldots + b_mX_m + sp_0$ of the ensemble forecats, where p_0 denotes the ratio of ensemble forecasts that are exactly 0, and the shape parameter σ is a linear function of the ensemble variance $c + dMD(X_1, \ldots, X_m)$, where $MD(X_1, \ldots, X_m)$ is Gini's mean difference. See ensembleMOSgev0 for details.

B is a vector of fitted regression coefficients: $b_1, \ldots, b_m$. Specifically, $a, b_1, \ldots, b_m, s, c, d, q$ are fitted to optimize the mean CRPS over the specified training period using optim.

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.
s	A vector of fitted EMOS coefficients for p_0 for each date, see details.
c, d	The fitted coefficients for the shape parameter, see details.
q	Fitted shape parameter, see details.

References

M. Scheuerer, Probabilistic quantitative precipitation forecasting using ensemble model output statistics. *Quarterly Journal of the Royal Meteorological Society* 140:1086–1096, 2014.

See Also

controlMOSgev0, fitMOSgev0

Examples

```
data("ensBMAtest", package = "ensembleBMA")

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

obs <- paste("PCP24","obs", sep = ".")
ens <- paste("PCP24", ensMemNames, sep = ".")
prcpTestData <- ensembleData(forecasts = ensBMAtest[,ens],
                             dates = ensBMAtest[,"vdate"],
                             observations = ensBMAtest[,obs],
                             station = ensBMAtest[,"station"],
                             forecastHour = 48,
                             initializationTime = "00")

prcpTestFitGEV0 <- ensembleMOSgev0(prcpTestData, trainingDays = 25,
                                   dates = "2008010100")
```

`ensembleMOSlognormal` *Log-normal EMOS modeling*

Description

Fits a log-normal EMOS model to ensemble forecasts for specified dates.

Usage

```
ensembleMOSlognormal(ensembleData, trainingDays, consecutive = FALSE,
                     dates = NULL, control = controlMOSlognormal(),
                     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 specified via the function controlMOStruncnormal. For details and default values, see controlMOStruncnormal.
`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 of size m: $X_1, \ldots, X_m$, the following log-normal model is fit by ensembleMOSlognormal:

$$Y \; LN(\mu, \sigma)$$

where LN denotes the log-normal distrbution with meanlog parameter μ and scalelog parameter σ, see Lognormal. The model is parametrized such that the mean value of the log-normal distribution is a linear function $a + b_1 X_1 + \ldots + b_m X_m$ of the ensemble forecats, and the variance is a linear function $c + dS^2$. For transformations between μ, σ and mean and variance of the log-normal distribution, see Baran and Lerch (2015). See ensembleMOSlognormal for details. 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$scoringRule over the specified training period using optim with method = control$optimRule.

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 — The fitted parameters for the variance, see details.

References

S. Baran and S. Lerch, Log-normal distribution based Ensemble Model Output Statistics models for probabilistic wind-speed forecasting. *Quarterly Journal of the Royal Meteorological Society* 141:2289–2299, 2015.

See Also

controlMOSlognormal, fitMOSlognormal

Examples

```
data("ensBMAtest", package = "ensembleBMA")

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

obs <- paste("MAXWSP10","obs", sep = ".")
ens <- paste("MAXWSP10", ensMemNames, sep = ".")
windTestData <- ensembleData(forecasts = ensBMAtest[,ens],
                             dates = ensBMAtest[,"vdate"],
```

```
                              observations = ensBMAtest[,obs],
                              station = ensBMAtest[,"station"],
                              forecastHour = 48,
                              initializationTime = "00")

windTestFitLN <- ensembleMOSlognormal(windTestData, trainingDays = 25)
```

`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 specified via the function controlMOSnormal. For details and default values, see 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 of size m: $X_1, \ldots, X_m$, the following Gaussian model is fit by `ensembleMOSnormal`:

$$Y \; N(a + b_1 X_1 + ... + b_m X_m, c + dS^2).$$

`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$scoringRule` over the specified training period using `optim` with `method = control$optimRule`.

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("ensBMAtest", package = "ensembleBMA")

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

obs <- paste("T2", "obs", sep = ".")
ens <- paste("T2", ensMemNames, sep = ".")
tempTestData <- ensembleData(forecasts = ensBMAtest[,ens],
                             dates = ensBMAtest[,"vdate"],
                             observations = ensBMAtest[,obs],
                             station = ensBMAtest[,"station"],
                             forecastHour = 48,
                             initializationTime = "00")

tempTestFit <- ensembleMOSnormal(tempTestData, trainingDays = 25)
```

`ensembleMOStruncnormal` *Truncated normal EMOS modeling*

Description

Fits a truncated normal EMOS model to ensemble forecasts for specified dates.

Usage

```
ensembleMOStruncnormal(ensembleData, trainingDays, consecutive = FALSE,
                       dates = NULL, control = controlMOStruncnormal(),
                       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 specified via the function controlMOStruncnormal. For details and default values, see controlMOStruncnormal.
`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 of size m: $X_1, \ldots, X_m$, the following truncated normal model is fit by `ensembleMOStruncnormal`:

$$Y \; N_0(a + b_1 X_1 + ... + b_m X_m, c + dS^2),$$

where N_0 denotes the normal distribution truncated at zero, with location $a + b_1 X_1 + ... + b_m X_m$ and squared scale $c + dS^2$. `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$scoringRule` over the specified training period using `optim` with `method = control$optimRule`.

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	The fitted parameters for the squared scale, see details.

References

T. L. Thorarinsdottir and T. Gneiting, Probabilistic forecasts of wind speed: Ensemble model output statistics by using heteroscedastic censored regression. *Journal of the Royal Statistical Society Series A* 173:371–388, 2010.

See Also

controlMOStruncnormal, fitMOStruncnormal

Examples

```
data("ensBMAtest", package = "ensembleBMA")

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

obs <- paste("MAXWSP10","obs", sep = ".")
ens <- paste("MAXWSP10", ensMemNames, sep = ".")
windTestData <- ensembleData(forecasts = ensBMAtest[,ens],
                             dates = ensBMAtest[,"vdate"],
                             observations = ensBMAtest[,obs],
                             station = ensBMAtest[,"station"],
                             forecastHour = 48,
                             initializationTime = "00")

windTestFitTN <- ensembleMOStruncnormal(windTestData, trainingDays = 25)
```

---

fitMOS *EMOS model fit to a training set*

Description

Fits an EMOS model 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 corresponding control function has to be chosen in accordance with the selected model. For the Gaussian (normal) EMOS model see controlMOSnormal, for the truncated normal model see controlMOStruncnormal, for the log-normal model see controlMOSlognormal, for the censored and shifted gamma model see controlMOScsg0, and for the censored generalized extreme value distribution model see controlMOSgev0.
model	A character string describing the EMOS model to be fit. Current choices are "normal" (typically used for temperature or pressure data), "truncnormal" (typically used for wind speed data), "lognormal" (typically used for wind speed data), "csg0" (typically used for precipitation accumulation data), and "gev0" (typically used for precipitation accumulation data). For specific details on model fitting see ensembleMOSnormal, ensembleMOStruncnormal, ensembleMOSlognormal, ensembleMOScsg0, or ensembleMOSgev0.
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 estimated coefficient values. The specific content depends on the chosen model.

References

Gaussian (normal) EMOS model:
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.

Truncated (normal) EMOS model:
T. L. Thorarinsdottir and T. Gneiting, Probabilistic forecasts of wind speed: Ensemble model output statistics by using heteroscedastic censored regression. *Journal of the Royal Statistical Society Series A* 173:371–388, 2010.

Log-normal EMOS model:
S. Baran and S. Lerch, Log-normal distribution based Ensemble Model Output Statistics models for probabilistic wind-speed forecasting. *Quarterly Journal of the Royal Meteorological Society* 141:2289–2299, 2015.

Censored and shifted gamma EMOS model:
M. Scheuerer and T. M. Hamill, Statistical post-processing of ensemble precipitation forecasts by fitting censored, shifted gamma distributions. *Monthly Weather Review* 143:4578–4596, 2015.
S. Baran and D. Nemoda, Censored and shifted gamma distribution based EMOS model for probabilistic quantitative precipitation forecasting. *Environmetrics* 27:280–292, 2016.

Censored generalized extreme value distribution EMOS model:
M. Scheuerer, Probabilistic quantitative precipitation forecasting using ensemble model output statistics. *Quarterly Journal of the Royal Meteorological Society* 140:1086–1096, 2014.

See Also

fitMOSnormal fitMOStruncnormal fitMOSlognormal fitMOScsg0 fitMOSgev0 controlMOSnormal controlMOStruncnormal controlMOSlognormal controlMOScsg0 controlMOSgev0

Examples

```
data("ensBMAtest", package = "ensembleBMA")

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

obs <- paste("T2", "obs", sep = ".")
ens <- paste("T2", ensMemNames, sep = ".")
tempTestData <- ensembleData(forecasts = ensBMAtest[,ens],
                             dates = ensBMAtest[,"vdate"],
                             observations = ensBMAtest[,obs],
                             station = ensBMAtest[,"station"],
                             forecastHour = 48,
                             initializationTime = "00")

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

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

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

---

fitMOScsg0 *Censored and shifted gamma EMOS modeling*

Description

Fits a censored and shifted gamma EMOS model to a given training set.

Usage

```
fitMOScsg0(ensembleData, control = controlMOScsg0(),
           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 specified via the function controlMOScsg0. For details and default values, see controlMOScsg0.
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 of size m: $X_1, \ldots, X_m$, the following shifted gamma model left-censored at 0 is fit by ensembleMOScsg0:

$$Y \; Gamma_0(\kappa, \theta, q)$$

where $Gamma_0$ denotes the shifted gamma distribution left-censored at zero, with shape κ, scale θ and shift q. The model is parametrized such that the mean $\kappa\theta$ is a linear function $a + b_1 X_1 + \ldots + b_m X_m$ of the ensemble forecats, and the variance $\kappa\theta^2$ is a linear function of the ensemble mean $c + d\overline{f}$, see Baran and Nemoda (2016) for details.

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$scoringRule over the specified training period using optim with method = control$optimRule.

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	The fitted parameters for the variance, see details.
q	Fitted shift parameter, see details.

References

M. Scheuerer and T. M. Hamill, Statistical post-processing of ensemble precipitation forecasts by fitting censored, shifted gamma distributions. *Monthly Weather Review* 143:4578–4596, 2015.

S. Baran and D. Nemoda, Censored and shifted gamma distribution based EMOS model for probabilistic quantitative precipitation forecasting. *Environmetrics* 27:280–292, 2016.

See Also

controlMOScsg0, ensembleMOScsg0,

Examples

```
data("ensBMAtest", package = "ensembleBMA")

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

obs <- paste("PCP24","obs", sep = ".")
ens <- paste("PCP24", ensMemNames, sep = ".")
prcpTestData <- ensembleData(forecasts = ensBMAtest[,ens],
                             dates = ensBMAtest[,"vdate"],
                             observations = ensBMAtest[,obs],
                             station = ensBMAtest[,"station"],
                             forecastHour = 48,
                             initializationTime = "00")

prcpTrain <- trainingData(prcpTestData, trainingDays = 30,
                             date = "2008010100")

prcpTestFit <- fitMOScsg0(prcpTrain)
```

`fitMOSgev0` *Censored generalized extreme value distribution EMOS modeling*

Description

Fits a censored generalized extreme value distribution EMOS model to a given training set.

Usage

```
fitMOSgev0(ensembleData, control = controlMOSgev0(),
           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 specified via the function controlMOSgev0. For details and default values, see controlMOSgev0.
`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 of size m: $X_1, \ldots, X_m$, the following generalized extreme value distribution EMOS model left-censored at 0 is fit by `ensembleMOSgev0`:

$$Y \; GEV_0(\mu, \sigma, q)$$

where GEV_0 denotes the generalized extreme value distribution left-censored at zero, with location μ, scale σ and shape q. The model is parametrized such that the mean m is a linear function $a + b_1X_1 + \ldots + b_mX_m + sp_0$ of the ensemble forecats, where p_0 denotes the ratio of ensemble forecasts that are exactly 0, and the shape parameter σ is a linear function of the ensemble variance $c + dMD(X_1, \ldots, X_m)$, where $MD(X_1, \ldots, X_m)$ is Gini's mean difference. See ensembleMOSgev0 for details.

`B` is a vector of fitted regression coefficients: $b_1, \ldots, b_m$. Specifically, $a, b_1, \ldots, b_m, s, c, d, q$ are fitted to optimize the mean CRPS over the specified training period using `optim`.

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.
`s`	A vector of fitted EMOS coefficients for p_0 for each date, see details.
`c, d`	The fitted coefficients for the shape parameter, see details.
`q`	Fitted shape parameter, see details.

References

M. Scheuerer, Probabilistic quantitative precipitation forecasting using ensemble model output statistics. *Quarterly Journal of the Royal Meteorological Society* 140:1086–1096, 2014.

See Also

`controlMOSgev0`, `ensembleMOSgev0`,

Examples

```
data("ensBMAtest", package = "ensembleBMA")

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

obs <- paste("PCP24","obs", sep = ".")
ens <- paste("PCP24", ensMemNames, sep = ".")
prcpTestData <- ensembleData(forecasts = ensBMAtest[,ens],
                             dates = ensBMAtest[,"vdate"],
                             observations = ensBMAtest[,obs],
                             station = ensBMAtest[,"station"],
```

```
                              forecastHour = 48,
                              initializationTime = "00")

prcpTrain <- trainingData(prcpTestData, trainingDays = 30,
                              date = "2008010100")

prcpTestFit <- fitMOSgev0(prcpTrain)
```

`fitMOSlognormal` *Log-normal EMOS model fit to a training set*

Description

Fits a log-normal EMOS model to a given training set.

Usage

```
fitMOSlognormal(ensembleData, control = controlMOSlognormal(),
                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 specified via the function controlMOSlognormal. For details and default values, see controlMOSlognormal.
`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 of size m: $X_1, \ldots, X_m$, the following log-normal model is fit by `ensembleMOSlognormal`:

$$Y \ LN(\mu, \sigma)$$

where LN denotes the log-normal distrbution with `meanlog` parameter μ and `scalelog` parameter σ, see Lognormal. The model is parametrized such that the mean value of the log-normal distribution is a linear function $a + b_1 X_1 + \ldots + b_m X_m$ of the ensemble forecats, and the variance is a linear function $c + dS^2$. For transformations between μ, σ and mean and variance of the log-normal distribution, see Baran and Lerch (2015). See ensembleMOSlognormal for details.

Value

A list with the following output components:

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

References

S. Baran and S. Lerch, Log-normal distribution based Ensemble Model Output Statistics models for probabilistic wind-speed forecasting. *Quarterly Journal of the Royal Meteorological Society* 141:2289–2299, 2015.

See Also

controlMOSlognormal, ensembleMOSlognormal,

Examples

```
data("ensBMAtest", package = "ensembleBMA")

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

obs <- paste("MAXWSP10","obs", sep = ".")
ens <- paste("MAXWSP10", ensMemNames, sep = ".")
windTestData <- ensembleData(forecasts = ensBMAtest[,ens],
                             dates = ensBMAtest[,"vdate"],
                             observations = ensBMAtest[,obs],
                             station = ensBMAtest[,"station"],
                             forecastHour = 48,
                             initializationTime = "00")

windTrain <- trainingData(windTestData, trainingDays = 30,
                          date = "2008010100")

windTestFit <- fitMOSlognormal(windTrain)
```

---

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 specified via the function controlMOSnormal. For details and default values, see 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 of size m: $X_1, \ldots, X_m$, the following Gaussian model is fit by ensembleMOSnormal:

$$Y \; N(a + b_1 X_1 + ... + b_m X_m, c + dS^2).$$

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$scoringRule over the specified training period using optim with method = control$optimRule.

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("ensBMAtest", package = "ensembleBMA")

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

obs <- paste("T2", "obs", sep = ".")
ens <- paste("T2", ensMemNames, sep = ".")
tempTestData <- ensembleData(forecasts = ensBMAtest[,ens],
                             dates = ensBMAtest[,"vdate"],
```

```
                              observations = ensBMAtest[,obs],
                              station = ensBMAtest[,"station"],
                              forecastHour = 48,
                              initializationTime = "00")

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

tempTrainFit <- fitMOSnormal(tempTrain)
```

`fitMOStruncnormal` *Truncated normal EMOS model fit to a training set*

Description

Fits a truncated normal EMOS model to a given training set.

Usage

```
fitMOStruncnormal(ensembleData, control = controlMOStruncnormal(),
                  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 specified via the function controlMOStruncnormal. For details and default values, see controlMOStruncnormal.
`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 of size m: $X_1, \ldots, X_m$, the following truncated normal model is fit by `ensembleMOStruncnormal`:

$$Y \sim N_0(a + b_1 X_1 + ... + b_m X_m, c + dS^2),$$

where N_0 denotes the normal distribution truncated at zero, with location $a + b_1 X_1 + ... + b_m X_m$ and squared scale $c + dS^2$. `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$scoringRule` over the specified training period using `optim` with `method = control$optimRule`.

Value

A list with the following output components:

a	The fitted intercept.
B	The fitted EMOS coefficients.
c, d	The fitted parameters for the squared scale, see details.

References

T. L. Thorarinsdottir and T. Gneiting, Probabilistic forecasts of wind speed: Ensemble model output statistics by using heteroscedastic censored regression. *Journal of the Royal Statistical Society Series A* 173:371–388, 2010.

See Also

controlMOStruncnormal, ensembleMOStruncnormal,

Examples

```
data("ensBMAtest", package = "ensembleBMA")

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

obs <- paste("MAXWSP10","obs", sep = ".")
ens <- paste("MAXWSP10", ensMemNames, sep = ".")
windTestData <- ensembleData(forecasts = ensBMAtest[,ens],
                             dates = ensBMAtest[,"vdate"],
                             observations = ensBMAtest[,obs],
                             station = ensBMAtest[,"station"],
                             forecastHour = 48,
                             initializationTime = "00")

windTrain <- trainingData(windTestData, trainingDays = 30,
                          date = "2008010100")

windTestFit <- fitMOStruncnormal(windTrain)
```

---

pars *Forecast distribution parameters*

Description

Computes the parameters of the forecast distribution resulting for univariate ensemble forecasting models.

Usage

```
pars(fit, ensembleData, 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.
dates	The dates for which the parameters 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.

Value

pars is a matrix giving the distribution parameter values for each instance in the data. The returned parameters denoted in the column names depend on the model used to generate the fit object.

See Also

ensembleMOS, fitMOS

Examples

```
data("ensBMAtest", package = "ensembleBMA")

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

obs <- paste("T2", "obs", sep = ".")
ens <- paste("T2", ensMemNames, sep = ".")
tempTestData <- ensembleData(forecasts = ensBMAtest[,ens],
                             dates = ensBMAtest[,"vdate"],
                             observations = ensBMAtest[,obs],
                             station = ensBMAtest[,"station"],
                             forecastHour = 48,
                             initializationTime = "00")

tempTestFit <- ensembleMOS(tempTestData, trainingDays = 25,
                           dates = c("2008010100", "2008010200"),
                           model = "normal")

parValues <- pars(tempTestFit, tempTestData)
parValues
```

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. 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("ensBMAtest", package = "ensembleBMA")

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

obs <- paste("T2", "obs", sep = ".")
ens <- paste("T2", ensMemNames, sep = ".")
tempTestData <- ensembleData(forecasts = ensBMAtest[,ens],
                             dates = ensBMAtest[,"vdate"],
                             observations = ensBMAtest[,obs],
                             station = ensBMAtest[,"station"],
                             forecastHour = 48,
                             initializationTime = "00")

tempTestFit <- ensembleMOS(tempTestData, trainingDays = 25,
                           dates = c("2008010100", "2008010200"),
                           model = "normal")

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

---

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("ensBMAtest", package = "ensembleBMA")

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

obs <- paste("T2", "obs", sep = ".")
ens <- paste("T2", ensMemNames, sep = ".")
tempTestData <- ensembleData(forecasts = ensBMAtest[,ens],
                             dates = ensBMAtest[,"vdate"],
                             observations = ensBMAtest[,obs],
                             station = ensBMAtest[,"station"],
                             forecastHour = 48,
                             initializationTime = "00")

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

tempTrainFit <- fitMOSnormal(tempTrain)
```

Index