Package 'EBMAforecast'

August 17, 2014

Type Package

Title Ensemble BMA Forecasting

Version 0.5

Date 2014-08-17

Author Jacob M. Montgomery, Florian M. Hollenbach, and Michael D. Ward

Maintainer Jacob Montgomery <jacob.montgomery@wustl.edu>

Description Ensemble BMA for social science data

License GPL (>= 2)

Depends separationplot, ensembleBMA, plyr, methods, Hmisc, abind

Imports Rcpp (>= 0.11.2)

LinkingTo Rcpp

R topics documented:

`calibrateEnsemble` *Calibrate an ensemble Bayesian Model Averaging model*

Description

This function calibrates an EBMA model based on out-of-sample performance in the calibration time period. Given a dependent variable and calibration-sample predictions from multiple component forecast models in the `ForecastData` the `calibrateEnsemble` function fits an ensemble BMA mixture model. The weights assigned to each model are derived from the individual model's performance in the calibration period. Missing observations are allowed in the calibration period, however models with missing observations are penalized. When missing observations are prevalent in the calibration set, the EM algorithm is adjusted and model paprameters are estimated by maximizing a renormalized partial expected complete-data log-likelihood (Fraley et al. 2010).

Usage

```
calibrateEnsemble(.forecastData = new("ForecastData"), exp = 1,
  tol = sqrt(.Machine$double.eps), maxIter = 1e+06, model = "logit",
  method = "EM", const= 0,...)

fitEnsemble(.forecastData, tol = sqrt(.Machine$double.eps), maxIter = 1e+06,
  method = "EM", exp = 1, useModelParams = TRUE,
  predType = "posteriorMedian", ...)
```

Arguments

`.forecastData`	An object of class 'ForecastData' that will be used to calibrate the model.
`exp`	The exponential shrinkage term. Forecasts are raised to the (1/exp) power on the logit scale for the purposes of bias reduction. The default value is `exp=3`.
`tol`	Tolerance for improvements in the log-likelihood before the EM algorithm will stop optimization. The default is `tol= 0.01`, which is somewhat high. Researchers may wish to reduce this by an order of magnitude for final model estimation.
`maxIter`	The maximum number of iterations the EM algorithm will run before stopping automatically. The default is `maxIter=10000`.
`model`	The model type that should be used given the type of data that is being predicted (i.e., normal, binary, etc.).
`method`	The estimation method used. Currently only implements "EM".
`predType`	The prediction type used for the EBMA model under the normal model, user can choose either `posteriorMedian` or `posteriorMean`. Posterior median is the default.
`const`	A "wisdom of crowds" parameter that reflects the scholars prior belief that all models should receive some, but not necessarily equal, weight. Value can be between 0 and 1. If `const=0` then the weights are completely determined by the EBMA model. If `const=1` all models will receive the same weight.
`...`	Not implemented

Value

Returns a data of class 'FDatFitLogit' or FDatFitNormal, a subclass of 'ForecastData', with the following slots

`predCalibration`
: A matrix containing the predictions of all component models and the EBMA model for all observations in the calibration period.

`predTest`
: A matrix containing the predictions of all component models and the EBMA model for all observations in the test period.

`outcomeCalibration`
: A vector containing the true values of the dependent variable for all observations in the calibration period.

`outcomeTest`
: An optional vector containing the true values of the dependent variable for all observations in the test period.

`modelNames`
: A character vector containing the names of all component models. If no model names are specified, names will be assigned automatically.

`modelWeights`
: A vector containing the model weights assigned to each model.

`modelParams`
: The parameters for the individual logit models that transform the component models.

`logLik`
: The final log-likelihood for the calibrated EBMA model.

`exp`
: The exponential shrinkage term.

`tol`
: Tolerance for improvements in the log-likelihood before the EM algorithm will stop optimization.

`maxIter`
: The maximum number of iterations the EM algorithm will run before stopping automatically.

`method`
: The estimation method used.

`call`
: The actual call used to create the object.

Author(s)

Michael D. Ward <<michael.d.ward@duke.edu>> and Jacob M. Montgomery <<jacob.montgomery@wustl.edu>> and Florian M. Hollenbach <<florian.hollenbach@duke.edu>>

References

Montgomery, Jacob M., Florian M. Hollenbach and Michael D. Ward. (2012). Improving Predictions Using Ensemble Bayesian Model Averaging. *Political Analysis*. **20**: 271-291.

Raftery, A. E., T. Gneiting, F. Balabdaoui and M. Polakowski. (2005). Using Bayesian Model Averaging to calibrate forecast ensembles. *Monthly Weather Review*. **133**:1155–1174.

Sloughter, J. M., A. E. Raftery, T. Gneiting and C. Fraley. (2007). Probabilistic quantitative precipitation forecasting using Bayesian model averaging. *Monthly Weather Review*. **135**:3209–3220.

Fraley, C., A. E. Raftery, T. Gneiting. (2010). Calibrating Multi-Model Forecast Ensembles with Exchangeable and Missing Members using Bayesian Model Averaging. *Monthly Weather Review*. **138**:190–202.

Sloughter, J. M., T. Gneiting and A. E. Raftery. (2010). Probabilistic wind speed forecasting using ensembles and Bayesian model averaging. *Journal of the American Statistical Association*. **105**:25–35.

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

Examples

```
## Not run: data(calibrationSample)

data(testSample)

this.ForecastData <- makeForecastData(.predCalibration=calibrationSample[,c("LMER", "SAE", "GLM")],
.outcomeCalibration=calibrationSample[,"Insurgency"],.predTest=testSample[,c("LMER", "SAE", "GLM")],
.outcomeTest=testSample[,"Insurgency"], .modelNames=c("LMER", "SAE", "GLM"))

this.ensemble <- calibrateEnsemble(this.ForecastData, model="logit", tol=0.001, exp=3)

## End(Not run)
```

`calibrationSample` *Calibration sample data*

Description

This includes the calibration sample data for the insurgency forecasting example in Montgomery, Hollenbach and Ward (2012). It provides the predictions for the three models included in the Ensemble model, as well as the true values of the dependent variable for insurgency in 29 Asian countries. The calibration sample ranges from January 2008 to December 2009.

Details

The variables included in the dataset are:

- `LMER` The calibration sample predictions of the LMER model from the insurgency prediction example in Montgomery et. al. (2012). The LMER model is a generalized linear mixed effects model using the logistic link function. It includes two random effects terms and several other covariates.
- `SAE` The calibration sample prediction of the SAE model from the insurgency prediction example in Montgomery et. al. (2012). This is a model developed as part of the ICEWS project and was designed by *Strategic Analysis Enterprises*. It is a simple generalized linear model with 27 independent variables.
- `GLM` The calibration sample prediction of the GLM model from the insurgency prediction example in Montgomery et. al. (2012). This is a crude logistic model with only four independent variables.
- `Insurgency` The true values of the dependent variable in the calibration sample from the insurgency prediction example in Montgomery et. al. (2012). This is a binary variable indicating the actual ocurrence of insurgency for each observation in the calibration sample.

References

Montgomery, Jacob M., Florian M. Hollenbach and Michael D. Ward. (2012). Improving Predictions Using Ensemble Bayesian Model Averaging. *Political Analysis*. *Political Analysis*. **20**: 271-291.

compareModels *Function for comparing multiple models based on predictive performance*

Description

This function produces statistics to compare the predictive performance of the different models component models, as well as for the EBMA model itself, for either the calibration or the test period. It currently calculates the area under the ROC (auc), the brier score, the percent of observations predicted correctly (percCorrect), as well as the proportional reduction in error compared to some baseline model (pre) for binary models. For models with normally distributed outcomes the CompareModels function can be used to calculate the root mean squared error (rmse) as well as the mean absolute error (mae).

Usage

```
compareModels(.forecastData, .period = "calibration",
  .fitStatistics = c("brier", "auc", "perCorrect", "pre"), .threshold = 0.5,
  .baseModel = 0, ...)
```

Arguments

.forecastData	An object of class 'ForecastData'.
.period	Can take value of "calibration" or "test" and indicates the period for which the test statistics should be calculated.
.fitStatistics	A vector naming statistics that should be calculated. Possible values include "auc", "brier", "percCorrect", "pre" for logit models and "mae","rsme" for normal models.
.threshold	The threshold used to calculate when a "positive" prediction is made by the model for binary dependent variables.
.baseModel	Vector containing predictions used to calculate proportional reduction of error ("pre").
...	Not implemented

Value

A data object of the class 'CompareModels' with the following slots:

fitStatistics	The output of the fit statistics for each model.
period	The period, "calibration" or "test", for which the statistics were calculated.
threshold	The threshold used to calculate when a "positive" prediction is made by the model.
baseModel	Vector containing predictions used to calculate proportional reduction of error ("pre").

Author(s)

Michael D. Ward <<michael.d.ward@duke.edu>> and Jacob M. Montgomery <<jacob.montgomery@wustl.edu>>

References

Montgomery, Jacob M., Florian M. Hollenbach and Michael D. Ward. (2012). Improving Predictions Using Ensemble Bayesian Model Averaging. *Political Analysis*. **20**: 271-291.

See Also

ensembleBMA, other functions

Examples

```
## Not run: data(calibrationSample)

data(testSample)

this.ForecastData <- makeForecastData(.predCalibration=calibrationSample[,c("LMER", "SAE", "GLM")],
.outcomeCalibration=calibrationSample[,"Insurgency"],.predTest=testSample[,c("LMER", "SAE", "GLM")],
.outcomeTest=testSample[,"Insurgency"], .modelNames=c("LMER", "SAE", "GLM"))

this.ensemble <- calibrateEnsemble(this.ForecastData, model="logit", tol=0.001, exp=3)

compareModels(this.ensemble,"calibration")

compareModels(this.ensemble,"test")

## End(Not run)
```

`EBMAforecast` *EBMAforecast*

Description

The EBMAforecast package (currently under development) allows users to increase the accuracy of forecasting models by pooling multiple component forecasts to generate ensemble forecasts. It includes functions to fit an ensemble Bayesian model averaging (EBMA) model using in-sample predictions, generate ensemble out-of-sample predictions, and create useful data visualizations. Currently, the package can only handle dichotomous outcomes or those with normally distributed errors, although additional models will be added to the package in the coming months. Missing observation are allowed in the calibration set, but models with many predictions missing are penalized.

Author(s)

Michael D. Ward <<michael.d.ward@duke.edu>> and Jacob M. Montgomery <<jacob.montgomery@wustl.edu>> and Florian M. Hollenbach <<florian.hollenbach@duke.edu>>

References

Montgomery, Jacob M., Florian M. Hollenbach and Michael D. Ward. (2012). Improving Predictions Using Ensemble Bayesian Model Averaging. *Political Analysis*. **20**: 271-291.

Raftery, A. E., T. Gneiting, F. Balabdaoui and M. Polakowski. (2005). Using Bayesian Model Averaging to calibrate forecast ensembles. *Monthly Weather Review*. **133**:1155–1174.

Sloughter, J. M., A. E. Raftery, T. Gneiting and C. Fraley. (2007). Probabilistic quantitative precipitation forecasting using Bayesian model averaging. *Monthly Weather Review*. **135**:3209–3220.

Fraley, C., A. E. Raftery, T. Gneiting. (2010). Calibrating Multi-Model Forecast Ensembles with Exchangeable and Missing Members using Bayesian Model Averaging. *Monthly Weather Review*. **138**:190–202.

Sloughter, J. M., T. Gneiting and A. E. Raftery. (2010). Probabilistic wind speed forecasting using ensembles and Bayesian model averaging. *Journal of the American Statistical Association*. **105**:25–35.

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

Examples

```
## Not run: demo(EBMAforecast)
demo(presForecast)

## End(Not run)
```

`ForecastData-class` *An ensemble forecasting data object*

Description

Objects of class `ForecastData` are used in the `calibrateEnsemble` function. Datasets should be converted into an object of class `ForecastData` using the `makeForecastData` function. Individual slots of the `ForecastData` object can be accessed and changed using the `slot` function. Missing observations in the prediction calibration set are allowed.

Usage

```
setPredCalibration(object) <- value

setPredTest(object) <- value

setOutcomeCalibration(object) <- value

setOutcomeTest(object) <- value

setModelNames(object) <- value
```

Details

A data object of the class 'ForecastData' has the following slots:

- `predCalibration` An array containing the predictions of all component models for the observations in the calibration period.
- `predTest` An array containing the predictions of all component models for all the observations in the test period.
- `outcomeCalibration` A vector containing the true values of the dependent variable for all observations in the calibration period.

- outcomeTest A vector containing the true values of the dependent variable for all observations in the test period.
- modelNames A character vector containing the names of all component models.

Author(s)

Michael D. Ward <<michael.d.ward@duke.edu>> and Jacob M. Montgomery <<jacob.montgomery@wustl.edu>> and Florian M. Hollenbach <<florian.hollenbach@duke.edu>>

References

Montgomery, Jacob M., Florian M. Hollenbach and Michael D. Ward. (2012). Improving Predictions Using Ensemble Bayesian Model Averaging. *Political Analysis*. **20**: 271-291.

See Also

ensembleBMA

Examples

```
## Not run:  data(calibrationSample)

data(testSample)
this.ForecastData <- makeForecastData(.predCalibration=calibrationSample[,c("LMER", "SAE", "GLM")],
.outcomeCalibration=calibrationSample[,"Insurgency"],.predTest=testSample[,c("LMER", "SAE", "GLM")],
.outcomeTest=testSample[,"Insurgency"], .modelNames=c("LMER", "SAE", "GLM"))

### to acces individual slots in the ForecastData object
slot(.forecastData, "predCalibration")
slot(.forecastData, "OutcomeCalibration")
slot(.forecastData, "predTest")
slot(.forecastData, "outcomeTest")
slot(.forecastData, "modelNames")

### to assign individual slots, use set functions or the slot function

setPredCalibration(this.ForecastData)<-calibrationSample[,c("LMER", "SAE", "GLM")]
setOutcomeCalibration(this.ForecastData)<-calibrationSample[,"Insurgency"]
setPredTest(this.ForecastData)<-testSample[,c("LMER", "SAE", "GLM")]
setOutcomeTest(this.ForecastData)<-testSample[,"Insurgency"]
setModelNames(this.ForecastData)<-c("LMER", "SAE", "GLM")

## End(Not run)
```

---

makeForecastData *Build a ensemble forecasting data object*

Description

This function uses the component model forecasts and dependent variable observations provided by the user to create an object of class ForecastData, which can then be used to calibrate and fit the ensemble. Individual slots of the ForecastData object can be accessed and changed using the slot or set functions. Missing predictions are allowed in the calibration set.

Usage

```
makeForecastData(.predCalibration = array(NA, dim = c(0, 0, 0)),
  .predTest = array(NA, dim = c(0, 0, 0)), .outcomeCalibration = numeric(),
  .outcomeTest = numeric(), .modelNames = character(), ...)
```

Arguments

.predCalibration	A matrix with the number of rows being the number of observations in the calibration period and a column with calibration period predictions for each model.
.predTest	A vector with the number of rows being the number of observations in the test period and a column with test period predictions for each model.
.outcomeCalibration	A vector with the true values of the dependent variable for each observation in the calibration period.
.outcomeTest	A vector with the true values of the dependent variable for each observation in the test period.
.modelNames	A vector of length p with the names of the component models.
...	Additional arguments not implemented

Value

A data object of the class 'ForecastData' with the following slots:

predCalibration	An array containing the predictions of all component models for all observations in the calibration period.
predTest	An array containing the predictions of all component models for all observations in the test period.
outcomeCalibration	A vector containing the true values of the dependent variable for all observations in the calibration period.
outcomeTest	A vector containing the true values of the dependent variable for all observations in the test period.
modelNames	A character vector containing the names of all component models. If no model names are specified, names will be assigned automatically.

Author(s)

Michael D. Ward <<michael.d.ward@duke.edu>> and Jacob M. Montgomery <<jacob.montgomery@wustl.edu>> and Florian M. Hollenbach <<florian.hollenbach@duke.edu>>

References

Montgomery, Jacob M., Florian M. Hollenbach and Michael D. Ward. (2012). Improving Predictions Using Ensemble Bayesian Model Averaging. *Political Analysis*. **20**: 271-291.

See Also

ensembleBMA

Examples

```
data(calibrationSample)

## Not run: data(testSample)
this.ForecastData <- makeForecastData(.predCalibration=calibrationSample[,c("LMER", "SAE", "GLM")],
.outcomeCalibration=calibrationSample[,"Insurgency"],.predTest=testSample[,c("LMER", "SAE", "GLM")],
.outcomeTest=testSample[,"Insurgency"], .modelNames=c("LMER", "SAE", "GLM"))

### to acces individual slots in the ForecastData object
slot(.forecastData, "predCalibration")
slot(.forecastData, "OutcomeCalibration")
slot(.forecastData, "predTest")
slot(.forecastData, "outcomeTest")
slot(.forecastData, "modelNames")

### to assign individual slots, use set functions or slot function

setPredCalibration(this.ForecastData)<-calibrationSample[,c("LMER", "SAE", "GLM")]
setOutcomeCalibration(this.ForecastData)<-calibrationSample[,"Insurgency"]
setPredTest(this.ForecastData)<-testSample[,c("LMER", "SAE", "GLM")]
setOutcomeTest(this.ForecastData)<-testSample[,"Insurgency"]
setModelNames(this.ForecastData)<-c("LMER", "SAE", "GLM")

## End(Not run)
```

---

plot,FDatFitLogit-method

Plotting function for ensemble models of the class "FDatFitLogit" or "FDatFitNormal", which are the objects created by the `calibrateEnsemble()` *function.*

Description

Default plotting for objectes created by the "calibrateEnsemble" function. See details below.

Usage

```
## S4 method for signature 'FDatFitLogit'
plot(x, y = NULL, period = "calibration", ...)
```

Arguments

x	An object of class "FDatFitLogit" or "FDatFitNormal"
period	Can take value of "calibration" or "test" and indicates the period for which the plots should be produced.
subset	The row names or numbers for the observations the user wishes to plot. Only implemented for the subclass "FDatFitNormal"
mainLabel	A vector strings to appear at the top of each predictive posterior plot. Only implemented for the subclass "FDatFitNormal"
xLab	The label for the x-axis. Only implemented for the subclass "FDatFitNormal"
yLab	The label for the y-axis. Only implemented for the subclass "FDatFitNormal"

`cols` A vector containing the color for plotting the predictive pdf of each component model forecast. Only implemented for the subclass "FDatFitNormal"

`y` the y coordinates of points in the plot, *optional* if `x` is an appropriate structure.

`...` Arguments to be passed to methods, such as graphical parameters (see `par`). Many methods will accept the following arguments:

`type` what type of plot should be drawn. Possible types are

- `"p"` for **p**oints,
- `"l"` for **l**ines,
- `"b"` for **b**oth,
- `"c"` for the lines part alone of `"b"`,
- `"o"` for both '**o**verplotted',
- `"h"` for '**h**istogram' like (or 'high-density') vertical lines,
- `"s"` for stair **s**teps,
- `"S"` for other **s**teps, see 'Details' below,
- `"n"` for no plotting.

All other `type`s give a warning or an error; using, e.g., `type = "punkte"` being equivalent to `type = "p"` for S compatibility. Note that some methods, e.g. `plot.factor`, do not accept this.

`main` an overall title for the plot: see `title`.

`sub` a sub title for the plot: see `title`.

`xlab` a title for the x axis: see `title`.

`ylab` a title for the y axis: see `title`.

`asp` the y/x aspect ratio, see `plot.window`.

Details

For objects of the class "FDatFitLogit", this function creates separation plots for each of the fitted models, including the EBMA model. Observations are ordered from left to right with increasing predicted probabilities, which is depicted by the black line. Actual occurrences are displayed by red vertical lines. Plots can be displayed for the test or calibration period. For objects of the class "FDatFitNormal", this function creates a plot of the predictive density distribution containing the EBMA PDF and the PDFs for all component models (scaled by their model weights). It also plots the prediction for the ensemble and the components for the specified observations.

Author(s)

Michael D. Ward <<michael.d.ward@duke.edu>> and Jacob M. Montgomery <<jacob.montgomery@wustl.edu>>

References

Raftery, A. E., T. Gneiting, F. Balabdaoui and M. Polakowski. (2005). Using Bayesian Model Averaging to calibrate forecast ensembles. *Monthly Weather Review*. **133**:1155–1174.

Greenhill, B., M.D. Ward, A. Sacks. (2011). The Separation Plot: A New Visual Method For Evaluating the Fit of Binary Data. *American Journal of Political Science*.**55**: 991–1002.

Montgomery, Jacob M., Florian M. Hollenbach and Michael D. Ward. (2012). Improving Predictions Using Ensemble Bayesian Model Averaging. *Political Analysis*. **20**: 271-291.

See Also

`separationplot`

Examples

```
data(calibrationSample)

data(testSample)

this.ForecastData <- makeForecastData(.predCalibration=calibrationSample[,c("LMER", "SAE", "GLM")],
.outcomeCalibration=calibrationSample[,"Insurgency"],.predTest=testSample[,c("LMER", "SAE", "GLM")],
.outcomeTest=testSample[,"Insurgency"], .modelNames=c("LMER", "SAE", "GLM"))

this.ensemble <- calibrateEnsemble(this.ForecastData, model="logit", tol=0.001, exp=3)

plot(this.ensemble, period="calibration")
plot(this.ensemble, period="test")
```

`presidentialForecast` *Sample data Presidential Election*

Description

This includes the data for the presidential election forecasting example in Montgomery, Hollenbach and Ward (2012). The data ranges from 1952 to 2008 and includes predictions for the six different component models included in the Ensemble model. Users may split the sample into calibration and test sample.

Details

The variables included in the dataset are:

- `Campbell` Predictions of Campbell's "Trial-Heat and Economy Model" (Campbell 2008).
- `Abramowitz` Predictions of Abramowitz's "Time for Change Model" (Abramowitz 2008).
- `Hibbs` Predictions for the "Bread and Peace Model" created by Douglas Hibbs (2008).
- `Fair` Forecasts from Fair's presidential vote share model (2010).
- `Lewis-Beck/Tien` Predictions from the "Jobs Model Forecast" by Michael Lewis-Beck and Charles Tien (2008).
- `EWT2C2` Predictions from the model in Column 2 in Table 2 by Erickson and Wlezien (2008).
- `Actual` The true values of the dependent variable, i.e. the incumbent-party voteshare in each presidential election in the sample.

References

Montgomery, Jacob M., Florian M. Hollenbach and Michael D. Ward. (2012). Improving Predictions Using Ensemble Bayesian Model Averaging. *Political Analysis*. **20**: 271-291.

Campbell, James E. 2008. The trial-heat forecast of the 2008 presidential vote: Performance and value considerations in an open-seat election. *PS: Political Science & Politics* **41**:697-701.

Hibbs, Douglas A. 2000. Bread and peace voting in U.S presidential elections. *Public Choice* **104**:149-180.

Fair, Ray C. 2010. Presidential and Congressional vote-share equations: November 2010 update. Working Paper. Yale University.

Lewis-Beck, Michael S. and Charles Tien. 2008. The job of president and the jobs model forecast: Obama for '08? *PS: Political Science & Politics* **41**:687-690.

Erikson, Robert S. and Christopher Wlezien. 2008. Leading economic indicators, the polls, and the presidential vote. *PS: Political Science & Politics* **41**:703-707.

`summary,FDatFitLogit-method`

Summary Function

Description

This function summarizes the Ensemble models that have been fit previously by the user.

Usage

```
## S4 method for signature 'FDatFitLogit'
summary(object, period = "calibration",
  fitStatistics = c("brier", "auc", "perCorrect", "pre"), threshold = 0.5,
  baseModel = 0, showCoefs = TRUE, ...)
```

Arguments

`object`	An object of the subclass "FDatFitLogit" or "FDatFitNormal"
`period`	The period for which the summary should be provided, either "calibration" or "test".
`fitStatistics`	A vector naming statistics that should be calculated. Possible values for objects in the "FDatFitLogit" subclass include "auc", "brier", "percCorrect", "pre". Possible values for objects in the "FDatFitNormal" subclass include "rmse" and "mae." Additional metrics will be made available in a future release of this package.
`threshold`	The threshold used to calculate when a "positive" prediction is made for a model. Not used for objects of the "FDatFitNormal" subclass.
`baseModel`	A vector containing predictions used to calculate proportional reduction of error ("pre"). Not used for objects of the "FDatFitNormal" subclass.
`showCoefs`	A logical indicating whether model coefficients from the ensemble should be shown.
`...`	Not implemented

Value

A data object of the class 'SummaryForecastData' with the following slots:

`summaryData`	Under the default, the function produces a matrix containing one row for each model plus one row for the EBMA forecast. The first column is always the model weights assigned to the component models. The second and third columns display the model parameters for the transformation of the component models. The remaining columns are the requested fit statistics for all models, as calculated by the `copareModels` function. If `showCoefs=FALSE`, then the model parameters will not be shown.

Author(s)

Michael D. Ward <<michael.d.ward@duke.edu>> and Jacob M. Montgomery <<jacob.montgomery@wustl.edu>>

Examples

```
## Not run:  data(calibrationSample)

data(testSample)

this.ForecastData <- makeForecastData(.predCalibration=calibrationSample[,c("LMER", "SAE", "GLM")],
.outcomeCalibration=calibrationSample[,"Insurgency"],.predTest=testSample[,c("LMER", "SAE", "GLM")],
.outcomeTest=testSample[,"Insurgency"], .modelNames=c("LMER", "SAE", "GLM"))

this.ensemble <- calibrateEnsemble(this.ForecastData, model="logit", tol=0.001,exp=3)

summary(this.ensemble, period="calibration")

summary(this.ensemble, period="test",showCoefs=FALSE)

## End(Not run)
```

`testSample` *Test sample data*

Description

This includes the test sample data for the insurgency forecasting example in Montgomery, Hollenbach and Ward (2012). It provides the predictions for the three models included in the Ensemble model, as well as the true values of the dependent variable for insurgency in 29 Asian countries. The test sample ranges ranges from January 2010 to December 2010.

Details

The variables included in the dataset are:

- `LMER` The test sample predictions of the LMER model from the insurgency prediction example in Montgomery et. al. (2012). The LMER model is a generalized linear mixed effects model using the logistic link function. It includes two random effects terms and several other covariates.
- `SAE` The test sample prediction of the SAE model from the insurgency prediction example in Montgomery et. al. (2012). This is a model developed as part of the ICEWS project and was designed by *Strategic Analysis Enterprises*. It is a simple generalized linear model with 27 independent variables.
- `GLM` The test sample prediction of the GLM model from the insurgency prediction example in Montgomery et. al. (2012). This is a crude logistic model with only four independent variables.
- `Insurgency` The true values of the dependent variable in the test sample from the insurgency prediction example in Montgomery et. al. (2012). This is a binary variable indicating the actual ocurrence of insurgency for each observation in the test sample.

References

Montgomery, Jacob M., Florian M. Hollenbach and Michael D. Ward. (2012). Improving Predictions Using Ensemble Bayesian Model Averaging. *Political Analysis*. **20**: 271-291.

Index