Package 'FindIt'

December 31, 2016

Version 1.0

Date 2016-12-22

Title Finding Heterogeneous Treatment Effects

Author Naoki Egami <negami@princeton.edu>, Marc Ratkovic <ratkovic@princeton.edu>, Kosuke Imai <kimai@princeton.edu>,

Maintainer Naoki Egami <negami@princeton.edu>

Depends R (>= 3.1.0), arm

Imports glmnet, lars, Matrix, quadprog, ggplot2, stats, graphics, utils

Description The heterogeneous treatment effect estimation procedure proposed by Imai and Ratkovic (2013)<DOI: 10.1214/12-AOAS593>. The proposed method is applicable, for example, when selecting a small number of most (or least) efficacious treatments from a large number of alternative treatments as well as when identifying subsets of the population who benefit (or are harmed by) a treatment of interest. The method adapts the Support Vector Machine classifier by placing separate LASSO constraints over the pre-treatment parameters and causal heterogeneity parameters of interest. This allows for the qualitative distinction between causal and other parameters, thereby making the variable selection suitable for the exploration of causal heterogeneity. The package also contains the function, CausalANOVA, which estimates the average marginal interaction effects by a regularized ANOVA as proposed by Egami and Imai (2016+).

LazyLoad yes

LazyData yes

License GPL (>= 2)

Repository CRAN

Date/Publication 2016-12-31 08:42:19

NeedsCompilation yes

R topics documented:

AMIE 2
Carlson 4
CausalANOVA 5
cv.CausalANOVA 8
FindIt 11
GerberGreen 15
LaLonde 16
plot.PredictFindIt 17
predict.FindIt 18

Index **19**

`AMIE` *Decomposing the Combination Effect into the AMEs and the AMIE.*

Description

`AMIE` shows the decomposition of the average combination effect into the AMEs and the AMIE. The function can decompose the conditional effect by setting the level of one factor equal to its baseline.

Usage

```
AMIE(object,fac.name,level.name,base.name=NULL,verbose=TRUE,...)
```

Arguments

`object`	an object from `CausalANOVA`.
`fac.name`	a character 2 dimensional vector indicating two factors of interest.
`level.name`	a character 2 dimensional vector indicating levels of each factor. When one of `level.name` is equal to `base.name`, the function decomposes the conditional effect instead of the combination effect.
`base.name`	a character 2 dimensional vector indicating baseline levels of each factor. If not specified, the function uses the last level defined by `levels()`.
`verbose`	whether it prints the results.
`...`	arguments passed to the function or arguments only for the internal use.

Details

Suggested workflow.

1. Specify the order of levels within each factor using `levels()`. Since the function places penalties on the differences between adjacent levels when levels are ordered, it is crucial to specify the order of levels within each factor carefully.

2. Run cv.CausalANOVA. Select the cost parameter minimizing the cross-validation error. Or choose largest value of cost such that error is within 1 standard error of the minimum. plot.cv.CausalANOVA can be used to investigate how cross-validation errors vary depending on cost parameters.
3. Run CausalANOVA. Run the main model with the chosen cost parameter and see summary by summary.CausalANOVA. If researchers want to compute selection probabilities, set select.prob=TRUE. Given it is computationally intensive, we recommend to compute selection probabilities when the model is finalized. The selection probability for the range of the AME (AMIE) is one minus the proportion of bootstrap replicates in which all coefficients for the corresponding factor (factor interaction) are estimated to be zero. The selection probability of the AME (AMIE) is the proportion of bootstrap replicates in which the sign of the effect is the same as the point estimate.
4. Investigate two-way interactions. Run plot.CausalANOVA and visualize the AMIEs by choosing two factors of interest. Run AMIE to examine decomposition of the average combination effect into the AMIE and AMEs.

Value

AMIE.main	an estimated AMIE.
baseline	baseline levels in each factor.
decompose	the decomposition of the combination effect into the AMEs and the AMIE. When one of level.name is equal to base.name, the function decomposes the conditional effect.

Author(s)

Naoki Egami and Kosuke Imai.

References

Post, J. B. and Bondell, H. D. 2013. "Factor selection and structural identification in the interaction anova model." Biometrics 69, 1, 70–79.

Egami, Naoki and Kosuke Imai. 2016+. "Causal Interaction in Factorial Experiments: Application to Conjoint Analysis." Working paper. http://imai.princeton.edu/research/files/int.pdf

See Also

cv.CausalANOVA,CausalANOVA

Examples

```
data(Carlson)
## Specify the order of each factor
Carlson$newRecordF<- factor(Carlson$newRecordF,ordered=TRUE,
                        levels=c("YesLC", "YesDis","YesMP",
                            "noLC","noDis","noMP","noBusi"))
Carlson$promise <- factor(Carlson$promise,ordered=TRUE,levels=c("jobs","clinic","education"))
Carlson$coeth_voting <- factor(Carlson$coeth_voting,ordered=FALSE,levels=c("0","1"))
Carlson$relevantdegree <- factor(Carlson$relevantdegree,ordered=FALSE,levels=c("0","1"))
```

```
## Run cv.CausalANOVA
## Not run:
cv.fit <- cv.CausalANOVA(won ~ newRecordF + promise + coeth_voting + relevantdegree,
                         data=Carlson,
                         pair.id=Carlson$contestresp,diff=TRUE, nway=2)

cv.fit
plot(cv.fit)

## End(Not run)

fit <- CausalANOVA(won ~ newRecordF + promise + coeth_voting + relevantdegree,
                   data=Carlson,
                   pair.id=Carlson$contestresp,diff=TRUE, nway=2,cost=0.15)
## Or when we need selection probabilities.
## Not run:
fit <- CausalANOVA(won ~ newRecordF + promise + coeth_voting + relevantdegree,
                   data=Carlson,
                   pair.id=Carlson$contestresp,diff=TRUE,nway=2,cost=0.15,
                   select.prob=TRUE,boot=500,block.id=Carlson$respcodeS)

## End(Not run)
summary(fit)

## Not run:
## plot
plot(fit,fac.name=c("newRecordF","coeth_voting"))

## End(Not run)

## compute AMIEs
amie1 <- AMIE(fit,fac.name=c("promise","newRecordF"),
              level.name=c("jobs","noLC"),
              base.name=c("jobs","YesLC"))

amie2 <- AMIE(fit,fac.name=c("newRecordF","coeth_voting"),
              level.name=c("noBus","1"),
              base.name=c("noMP","0"))
```

`Carlson` *Data from conjoint analysis in Carlson (2015).*

Description

This data set gives the outcomes a well as treatment assignments of the conjoint analysis in Carlson (2015). Please Carlson (2015) and Egami and Imai (2016+) for more details.

Usage

```
data
```

Format

A data frame consisting of 7 columns (including a treatment assignment vector) and 3232 observations.

outcome	integer	whether a profile is chosen	0,1
newRecordF	factor	record as a politician	7 levels
promise	factor	platform	3 levels (job, clinic, education)
coeth_voting	factor	whether a profile is coethnic to a respodent	Yes, No
Degree	factor	job whether a profile has relevant degrees	4 Yes, No

Source

Data from Carlson (2015).

References

Carlson, E. 2015. "Ethnic voting and accountability in africa: A choice experiment in uganda." World Politics 67, 02, 353–385.

`CausalANOVA` *Estimating the AMEs and AMIEs with the CausalANOVA.*

Description

`CausalANOVA` estimates coefficients of the specified ANOVA with regularization. By taking differences in coefficients, the function recovers the AMEs and AMIEs.

Usage

```
CausalANOVA(formula,data,cost,pair.id=NULL,nway=2,diff=TRUE,
    select.prob=FALSE,boot=100,block.id=NULL,seed=1234,
    eps=1e-5,fac.level=NULL,ord.fac=NULL,verbose=TRUE)
```

Arguments

`formula`	a formula that specifies outcome and treatment variables.
`data`	an optional data frame, list or environment (or object coercible by 'as.data.frame' to a data frame) containing the variables in the model. If not found in 'data', the variables are taken from 'environment(formula)', typically the environment from which 'CausalANOVA' is called.

cost	a cost parameter ranging from 0 to 1. 1 corresponds to no regularization. It is recommended to choose the value using cross validation with cv.GashANOVA.
pair.id	unique identifiers for each pair of comparison. This option is used when dif=TRUE.
nway	"2" when the two way causal interactions are of interest and "3" when the three-way and two-way causal interactions are of interest. Default is 2.
diff	a logical indicating whether the outcome is the choice between a pair. If diff=TRUE, pair.id should specify a pair of comparison.
select.prob	a logical indicating whether selection probabilities are computed. We recommend to use this option when the model is finalized.
boot	the number of bootstrap replicates.
block.id	unique identifies for blocks that the function uses for block bootstrap.
seed	seed for bootstrap.
eps	a tolerance parameter in the internal optimization algorithm.
fac.level	optional. A vector containing the number of levels in each factor. The order of fac.level should match to the order of columns in the data. For example, when the first and second columns of the design matrix is "Education" and "Race", the first and second element of fac.level should be the number of levels in "Education" and "Race", respectively.
ord.fac	optional. logical vectors indicating whether each factor has ordered (TRUE) or unordered (FALSE) levels. When levels are ordered, the function uses the order given by function levels(). If levels are ordered, the function places penalties on the differences between adjacent levels. If levels are unordered, the function places penalties on the differences based on every pairwise comparison.
verbose	whether it prints the value of a cost parameter used.

Details

Suggested workflow.

1. Specify the order of levels within each factor using levels(). Since the function places penalties on the differences between adjacent levels when levels are ordered, it is crucial to specify the order of levels within each factor carefully.
2. Run cv.CausalANOVA. Select the cost parameter minimizing the cross-validation error. Or choose largest value of cost such that error is within 1 standard error of the minimum. plot.cv.CausalANOVA can be used to investigate how cross-validation errors vary depending on cost parameters.
3. Run CausalANOVA. Run the main model with the chosen cost parameter and see summary by summary.CausalANOVA. If researchers want to compute selection probabilities, set select.prob=TRUE. Given it is computationally intensive, we recommend to compute selection probabilities when the model is finalized. The selection probability for the range of the AME (AMIE) is one minus the proportion of bootstrap replicates in which all coefficients for the corresponding factor (factor interaction) are estimated to be zero. The selection probability of the AME (AMIE) is the proportion of bootstrap replicates in which the sign of the effect is the same as the point estimate.
4. Investigate two-way interactions. Run plot.CausalANOVA and visualize the AMIEs by choosing two factors of interest. Run AMIE to examine decomposition of the average combination effect into the AMIE and AMEs.

Value

intercept	an intercept of the estimated ANOVA model.If diff=TRUE, this should be close to 0.5.
coefs	a named vector of coefficients of the estimated ANOVA model.
formula	the formula used in the function.
cost	the cost parameter used in the function.
...	arguments passed to the function or arguments only for the internal use.

Author(s)

Naoki Egami and Kosuke Imai.

References

Post, J. B. and Bondell, H. D. 2013. "Factor selection and structural identification in the interaction anova model." Biometrics 69, 1, 70–79.

Egami, Naoki and Kosuke Imai. 2016+. "Causal Interaction in Factorial Experiments: Application to Conjoint Analysis." Working paper. http://imai.princeton.edu/research/files/int.pdf

See Also

cv.CausalANOVA,AMIE

Examples

```
data(Carlson)
## Specify the order of each factor
Carlson$newRecordF<- factor(Carlson$newRecordF,ordered=TRUE,
                        levels=c("YesLC", "YesDis","YesMP",
                            "noLC","noDis","noMP","noBusi"))
Carlson$promise <- factor(Carlson$promise,ordered=TRUE,levels=c("jobs","clinic","education"))
Carlson$coeth_voting <- factor(Carlson$coeth_voting,ordered=FALSE,levels=c("0","1"))
Carlson$relevantdegree <- factor(Carlson$relevantdegree,ordered=FALSE,levels=c("0","1"))

## Run cv.CausalANOVA
## Not run:
cv.fit <- cv.CausalANOVA(won ~ newRecordF + promise + coeth_voting + relevantdegree,
                        data=Carlson,
                        pair.id=Carlson$contestresp,diff=TRUE, nway=2)

cv.fit
plot(cv.fit)

## End(Not run)

fit <- CausalANOVA(won ~ newRecordF + promise + coeth_voting + relevantdegree,
                    data=Carlson,
                    pair.id=Carlson$contestresp,diff=TRUE, nway=2,cost=0.15)
## Or when we need selection probabilities.
```

```
## Not run:
fit <- CausalANOVA(won ~ newRecordF + promise + coeth_voting + relevantdegree,
                   data=Carlson,
                   pair.id=Carlson$contestresp,diff=TRUE,nway=2,cost=0.15,
                   select.prob=TRUE,boot=500,block.id=Carlson$respcodeS)

## End(Not run)
summary(fit)

## Not run:
## plot
plot(fit,fac.name=c("newRecordF","coeth_voting"))

## End(Not run)

## compute AMIEs
amie1 <- AMIE(fit,fac.name=c("promise","newRecordF"),
              level.name=c("jobs","noLC"),
              base.name=c("jobs","YesLC"))

amie2 <- AMIE(fit,fac.name=c("newRecordF","coeth_voting"),
              level.name=c("noBus","1"),
              base.name=c("noMP","0"))
```

`cv.CausalANOVA` *Cross validation for the CausalANOVA.*

Description

`cv.CausalANOVA` implements cross-validation for `CausalANOVA` to select the cost parameter.

Usage

```
cv.CausalANOVA(formula,data,cv.cost=c(0.1,0.3,0.5,0.7,1.0), type="bin",
               pair.id=NULL,nway=2,diff=TRUE,eps=1e-5,nfolds=10,seed=1234)
```

Arguments

`formula`	a formula that specifies outcome and treatment variables.
`data`	an optional data frame, list or environment (or object coercible by 'as.data.frame' to a data frame) containing the variables in the model. If not found in 'data', the variables are taken from 'environment(formula)', typically the environment from which 'CausalANOVA' is called.
`cv.cost`	a vector containing candidates for a cost parameter ranging from 0 to 1. 1 corresponds to no regularization and the smaller value corresponds to the stronger regularization. Default is `c(0.1,0.3,0.5,0.7,1.0)`.

type	When the outcome is binary, set type to "bin". Cross-validation error is based on misclassification. WHen the outcome is continuous, set type to "cont". Cross-validation error is based on the mean squared error.
pair.id	unique identifiers for each pair of comparison. This option is used when dif=TRUE.
nway	"2" when the two way causal interactions are of interest and "3" when the three-way and two-way causal interactions are of interest. Default is 2.
diff	a logical indicating whether the outcome is the choice between a pair. If diff=TRUE, pair.id should specify a pair of comparison.
eps	a tolerance parameter in the internal optimization algorithm.
nfolds	number of folds - default is 10. Although nfolds can be as large as the sample size (leave-one-out CV), it is not recommended for large datasets.
seed	an argument for set.seed().

Details

Suggested workflow.

1. Specify the order of levels within each factor using levels(). Since the function places penalties on the differences between adjacent levels when levels are ordered, it is crucial to specify the order of levels within each factor carefully.
2. Run cv.CausalANOVA. Select the cost parameter minimizing the cross-validation error. Or choose largest value of cost such that error is within 1 standard error of the minimum. plot.cv.CausalANOVA can be used to investigate how cross-validation errors vary depending on cost parameters.
3. Run CausalANOVA. Run the main model with the chosen cost parameter and see summary by summary.CausalANOVA. If researchers want to compute selection probabilities, set select.prob=TRUE. Given it is computationally intensive, we recommend to compute selection probabilities when the model is finalized. The selection probability for the range of the AME (AMIE) is one minus the proportion of bootstrap replicates in which all coefficients for the corresponding factor (factor interaction) are estimated to be zero. The selection probability of the AME (AMIE) is the proportion of bootstrap replicates in which the sign of the effect is the same as the point estimate.
4. Investigate two-way interactions. Run plot.CausalANOVA and visualize the AMIEs by choosing two factors of interest. Run AMIE to examine decomposition of the average combination effect into the AMIE and AMEs.

Value

cv.error	The mean cross-validated error - a vector of length length(cv.t).
cv.min	value of t that gives minimum cv.missclass.
cv.1sd	largest value of t such that error is within 1 standard error of the minimum.
cv.each.mat	a matrix containing cross-validation errors for each fold and cost parameter.
cv.cost	the cv.cost used in the function.

Author(s)

Naoki Egami and Kosuke Imai.

References

Post, J. B. and Bondell, H. D. 2013. "Factor selection and structural identification in the interaction anova model." Biometrics 69, 1, 70–79.

Egami, Naoki and Kosuke Imai. 2016+. "Causal Interaction in Factorial Experiments: Application to Conjoint Analysis." Working paper. http://imai.princeton.edu/research/files/int.pdf

See Also

CausalANOVA,AMIE.

Examples

```
data(Carlson)
## Specify the order of each factor
Carlson$newRecordF<- factor(Carlson$newRecordF,ordered=TRUE,
                        levels=c("YesLC", "YesDis","YesMP",
                            "noLC","noDis","noMP","noBusi"))
Carlson$promise <- factor(Carlson$promise,ordered=TRUE,levels=c("jobs","clinic","education"))
Carlson$coeth_voting <- factor(Carlson$coeth_voting,ordered=FALSE,levels=c("0","1"))
Carlson$relevantdegree <- factor(Carlson$relevantdegree,ordered=FALSE,levels=c("0","1"))

## Run cv.CausalANOVA
## Not run:
cv.fit <- cv.CausalANOVA(won ~ newRecordF + promise + coeth_voting + relevantdegree,
                        data=Carlson,
                        pair.id=Carlson$contestresp,diff=TRUE, nway=2)

cv.fit
plot(cv.fit)

## End(Not run)

fit <- CausalANOVA(won ~ newRecordF + promise + coeth_voting + relevantdegree,
                    data=Carlson,
                    pair.id=Carlson$contestresp,diff=TRUE, nway=2,cost=0.15)
## Or when we need selection probabilities.
## Not run:
fit <- CausalANOVA(won ~ newRecordF + promise + coeth_voting + relevantdegree,
                    data=Carlson,
                    pair.id=Carlson$contestresp,diff=TRUE,nway=2,cost=0.15,
                    select.prob=TRUE,boot=500,block.id=Carlson$respcodeS)

## End(Not run)
summary(fit)

## Not run:
## plot
```

```
plot(fit,fac.name=c("newRecordF","coeth_voting"))

## End(Not run)

## compute AMIEs
amie1 <- AMIE(fit,fac.name=c("promise","newRecordF"),
              level.name=c("jobs","noLC"),
              base.name=c("jobs","YesLC"))

amie2 <- AMIE(fit,fac.name=c("newRecordF","coeth_voting"),
              level.name=c("noBus","1"),
              base.name=c("noMP","0"))
```

FindIt *FindIt for Estimating Heterogeneous Treatment Effects*

Description

FindIt returns a model with the most predictive treatment-treatment interactions or treatment-covariate interactions.

Usage

```
FindIt(model.treat, model.main, model.int,data = NULL,
       type = "binary", treat.type = "multiple", nway,
       search.lambdas = TRUE, lambdas = NULL,
       make.twoway = TRUE, make.allway = TRUE,
       wts = 1, scale.c = 1, scale.int = 1, fit.glmnet = TRUE,
       make.reference = TRUE,reference.main = NULL,threshold = 0.999999)
```

Arguments

model.treat	A formula that specifies outcome and treatment variables.
model.main	An optional formula that specifies pre-treatment covariates to be adjusted.
model.int	A formula specifying pre-treatment covariates to be interacted with treatment assignments when treat.type="single".
data	An optional data frame, list or environment (or object coercible by 'as.data.frame' to a data frame) containing the variables in the model. If not found in 'data', the variables are taken from 'environment(formula)', typically the environment from which 'FindIt' is called.
type	"binary" for a binary outcome variable, which needs to be integer class; "continuous" for a continuous outcome variable.
treat.type	"single" for interactions between a single treatment variable, which needs to be integer class, and multiple pre-treatment covariates specified with model.int; "multiple" is used when treatment-treatment interactions are of interest and treat is a matrix of multiple treatments.

nway	An argument passed to makeallway when treat.type="multiple". FindIt generates treatment-treatment interactions up to the order specified with this argument. In general, it is recommended to use the number of factorial treatments. The current version covers up to four way interactions.
search.lambdas	Whether to search for the tuning parameters for the LASSO constraints. If FALSE, lambdas must be supplied.
lambdas	Tuning parameters to be given to FindIt; only used if search.lambdas=FALSE.
make.twoway	If make.twoway=TRUE, all possible two-way interactions for the pre-treatment covariates specified in model.main and model.int are generated within FindIt. The default is set to be TRUE.
make.allway	If make.allway=TRUE, all possible treatment-treatment interactions for multiple treatments are generated when treat.type="multiple". Interactions of the order up to the value of nway is computed.
wts	An optional set of scaling weights. The default is 1.
scale.c	A set of weights for recaling the pre-treatment covariates; only used if make.twoway=FALSE. maketwoway is useful for generating these.
scale.int	A set of weights for recaling the covariates to be interacted with treatment variables ; only used if make.twoway=FALSE. maketwoway is useful for generating these.
fit.glmnet	Whether to fit using the coordinate descent method in glmnet (TRUE) or the regularization path method of LARS (FALSE).
make.reference	Whether to make a reference matrix to check which columns are dropped when makeallway=TRUE.
reference.main	If make.allway=FALSE and researchers generate a matrix of all possible interactions between factorial treatments, reference from makeallway function is better to be passed to FindIt through this argument.
threshold	An argument passed to makeallway when treat.type="multiple". Threshold to drop correlated columns when makeallway is used.

Details

Implements the alternating line search algorithm for estimating the tuning parameters, as described in Imai and Ratkovic (2013).

Value

coefs	A named vector of scaled coefficients
coefs.orig	A vector of coefficients on the original scale, if scale.c and scale.t was used
fit	Fitted values on an SVM scale
names.out	Names of the coefficients
y	A vector of observed outcomes
X.c	A matrix of pre-treatment covariates to be adjusted
X.t	A matrix of treatments and treatment-treatment interactions, or treatment-covariate interactions

GCV	GCV statistic at the minimum
ATE	When treat.type="single", the estimated ATE. When treat.type="multiple", the estimated treatment effect of each unique treatment combination
lambdas	Tuning parameters used for the fit
reference	When treat.type="multiple", after making all interaction terms, columns with no variation or columns perfectly correlated with one of other columns are automatically dropped. reference shows which columns are kept and dropped.

Author(s)

Naoki Egami, Marc Ratkovic and Kosuke Imai.

References

Imai, Kosuke and Marc Ratkovic. 2013. "Estimating Treatment Effect Heterogeneity in Randomized Program Evaluation." Annals of Applied Statistics, Vol.7, No.1(March), pp. 443-470. http://imai.princeton.edu/research/files/svm.pdf

Egami, Naoki and Kosuke Imai. 2015. "Causal Interaction in High-Dimension." Working paper. http://imai.princeton.edu/research/files/int.pdf

Examples

```
#################################################
## Example 1: Treatment-Covariate Interaction
#################################################
data(LaLonde)

## The model includes a treatment variable,
## nine covariates to be interacted with the treatment variable,
## and the same nine covariates to be adjusted.

## Not run:

## Run to find the LASSO parameters
F1  <-FindIt(model.treat= outcome ~ treat,
             model.main= ~ age+educ+black+hisp+white+
             marr+nodegr+log.re75+u75,
             model.int= ~ age+educ+black+hisp+white+
             marr+nodegr+log.re75+u75,
             data = LaLonde,
             type="binary",
             treat.type="single")

## End(Not run)

## Fit with uncovered lambda parameters.
F1  <-FindIt(model.treat= outcome ~ treat,
             model.main= ~ age+educ+black+hisp+white+
             marr+nodegr+log.re75+u75,
```

```
              model.int= ~ age+educ+black+hisp+white+
              marr+nodegr+log.re75+u75,
              data = LaLonde,
              type="binary",
              treat.type="single",
              search.lambdas=FALSE,
              lambdas = c(-3.8760,-4.0025) )

summary(F1)

## Returns all the estimated treatment effects.
pred1 <- predict(F1)
## Top10
head(pred1$data, n=10)
## Bottom 10
tail(pred1$data ,n=10)

## Visualize all the estimated treatment effects.
## Not run:
plot(pred1)

## End(Not run)

#################################################
## Example 2: Treatment-Treatment Interaction
#################################################

## Not run:
data(GerberGreen)

## The model includes four factorial treatments and
## all two, three, four-way interactions between them.
## Four pre-treatment covariates are adjusted.

## Run to search for lambdas.
F2<- FindIt(model.treat= voted98 ~ persngrp+phnscrpt+mailings+appeal,
              nway=4,
              model.main= ~ age+majorpty+vote96.1+vote96.0,
              data = GerberGreen,
              type="binary",
              treat.type="multiple")

## Fit, given selected lambdas.
F2<- FindIt(model.treat= voted98 ~ persngrp+phnscrpt+mailings+appeal,
              nway=4,
              model.main= ~ age+majorpty+vote96.1+vote96.0,
              data = GerberGreen,
              type="binary",
              treat.type="multiple",
              search.lambdas=FALSE,
              lambdas=c(-15.000,-6.237))

## Returns coefficient estimates.
```

```
summary(F2)

## Returns predicted values for unique treatment combinations.
pred2 <- predict(F2,unique=TRUE)
## Top 10
head(pred2$data, n=10)
## Bottom 10
tail(pred2$data, n=10)

## Visualize predicted values for each treatment combination.
plot(pred2)

## End(Not run)
```

GerberGreen *Data from the 1998 New Haven Get-Out-the-Vote Experiment*

Description

This data set contains the most recent corrected data from the field experiment analyzed in Gerber and Green (2000).

Usage

```
data
```

Format

A data frame consisting of 9 columns and 29,380 observations.

voted98	integer	voted in 1998	0,1
persngrp	factor	personal contact attempted	0,1
phnscrpt	factor	script read to phone respondents	7 levels
mailings	factor	number of mailings sent	0 - 3
appeal	factor	content of message	3 levels
age	integer	age of respondent	
majorpty	factor	Democratic or Republican	
voted96.1	factor	voted in 1996	0,1
voted96.0	factor	abstained in 1996	0,1

Note: The levels of phnscrpt and appeal are follows.

phnscrpt: Script read to phone respondents

0	No phone
1	Civic-Blood
2	Civic
3	Civic or Blood-Civic

4 Neighbor
5 Neighbor or Civic-Neighbor
6 Close

appeal: Content of message

1 Civic Duty
2 Neighborhood Solidarity
3 Close Election

References

Gerber, A. S. and Green, D. P. 2000 . "The effects of canvassing, telephone calls, and direct mail on voter turnout: A field experiment." American Political Science Review, Vol.94, No.3, pp. 653-663.

Imai, K. 2005 . "Do get-out-the-vote calls reduce turnout?: The importance of statistical methods for field experiments."" American Political Science Review, Vol.99, No.2, pp. 283-300.

LaLonde *National Supported Work Study Experimental Data*

Description

This data set gives the outcomes a well as treatment assignments and covariates for the National Supported Work Study, as analyzed in LaLonde (1986).

Usage

```
data
```

Format

A data frame consisting of 12 columns (including a treatment assignment vector) and 2787 observations.

outcome	integer	whether earnings in 1978 are larger than in 1975	0,1
treat	integer	whether the individual received the treatment	0,1
age	numeric	age in years	
educ	numeric	education in years	
black	factor	black or not	0,1
hisp	factor	hispanic or not	0,1
white	factor	white or not	0,1
marr	factor	married or not	0,1
nodegr	factor	an indicator for no high school degree	0,1
log.re75	numeric	log of earnings in 1975	
u75	factor	unemployed in 1975	0,1
wts.extrap	numeric	extrapolation weights to the 1978 Panel Study for Income Dynamics dataset	

Source

Data from the National Supported Work Study. A benchmark matching dataset. 1975 earnings are pre-treatment.

References

LaLonde, R.J. 1986. "Evaluating the econometric evaulations of training programs with experimental data."" American Economic Review, Vol.76, No.4, pp. 604-620.

`plot.PredictFindIt` *Plot estimated treatment effects or predicted outcomes for each treatment combination.*

Description

Plot estimated treatment effects when `treat.type="single"` and predicted outcomes for each treatment combination when `treat.type="multiple"`.

Usage

```
## S3 method for class 'PredictFindIt'
plot(x,main,xlab,ylab, ...)
```

Arguments

`x`	output from `predict.FindIt`.
`main`	the argument specifying the main title of the plot.
`xlab`	the argument specifying the name of x axis.
`ylab`	the argument specifying the name of y axis.
`...`	further arguments passed to or from other methods.

Details

Plot estimated treatment effects when `treat.type="single"` and predicted outcomes for each treatment combination when `treat.type="multiple"`.

Value

`plot`	Plot estimated treatment effects when `treat.type="single"` and predicted outcomes for each treatment combination when `treat.type="multiple"`.

Author(s)

Naoki Egami, Marc Ratkovic and Kosuke Imai.

Examples

```
## See the help page for FindIt() for an example.
```

---

predict.FindIt *Computing predicted values for each sample in the data.*

Description

predict.FindIt takes an output from FindIt and returns estimated treatment effects when treat.type="single" and predicted outcomes for each treatment combination when treat.type="multiple".

Usage

```
## S3 method for class 'FindIt'
predict(object, newdata, sort = TRUE, decreasing = TRUE,wts = 1, unique = FALSE, ...)
```

Arguments

object	An output object from FindIt.
newdata	An optional data frame in which to look for variables with which to predict. If omitted, the data used in FindIt is used.
sort	Whether to sort samples according to estimated treatment effects.
decreasing	When sort=TRUE, whether to sort the output in descending order or not.
wts	Weights.
unique	If unique=TRUE, predict returns estimated treatment effects or predicted outcomes for unique samples.
...	further arguments passed to or from other methods.

Details

Useful for computing estimated treatment effects or predicted outcomes for each treatment combination. By using newdata, researchers can compute them for any samples.

Value

data	A matrix of estimated treatment effects when treat.type="single" and predicted outcomes for each treatment combination when treat.type="multiple".

Author(s)

Naoki Egami, Marc Ratkovic and Kosuke Imai.

Examples

```
## See the help page for FindIt() for an example.
```

Index

∗Topic **datasets**
Carlson, 4
GerberGreen, 15
LaLonde, 16

AMIE, 2, *7*, *10*

Carlson, 4
CausalANOVA, *3*, 5, *10*
cv.CausalANOVA, *3*, *7*, 8

FindIt, 11

GerberGreen, 15

LaLonde, 16

plot.CausalANOVA *(*CausalANOVA*)*, 5
plot.cv.CausalANOVA *(*cv.CausalANOVA*)*, 8
plot.PredictFindIt, 17
predict.FindIt, 18

summary.CausalANOVA *(*CausalANOVA*)*, 5
summary.FindIt *(*FindIt*)*, 11