Package 'MetaPersonalized'

October 4, 2017

Type Package

Title What the Package Does (Title Case)

Version 0.1.0

Author Who wrote it

Maintainer The package maintainer <yourself@somewhere.net>

Description More about what it does (maybe more than one line)
  Use four spaces when indenting paragraphs within the Description.

License What license is it under?

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

Imports glmnet,
    SGL

R topics documented:

MetaPersonalized   *Meta-analysis for Personalized Medicine*

Usage

```
MetaPersonalized(Xlist, Ylist, Trtlist, Plist, typelist = NULL,
  model = c("linear", "meta", "sparse group lasso", "group lasso", "lasso"),
  lambda1 = NULL, lambda2 = NULL, unique_rule_lambda = NULL,
  alpha = NULL, unique_rule = FALSE)
```

Arguments

Xlist	a list object with kth element denoting the covariate matrix of study k
Ylist	a list object with kth element denoting the response vector of study k
Trtlist	a list object with kth element denoting the treatment vector of study k (coded as 0 or 1)
model	the model to be used for the above framework, can be linear, meta-analysis, sparse group lasso, group lasso or lasso
lambda1	lambda1 in the framework above
lambda2	lambda2 in the framework above
unique_rule_lambda	
	λ_{uni} when unique treatment rule is required
alpha	alpha in the framework above
unique_rule	a logical value, whether a unique treatment rule is required
typlelist	a list object with kth element denoting the type of response in study k, can be continuous or binary, default is continuous

Details

Assume the total number of studies is K. This function implements meta-analysis for personalized medicine based on the following framework:

$$\min_{g_1,\ldots,g_K} \sum_{k=1}^{K} \sum_{i=1}^{n_k} \frac{|\hat{C}_k(X_i)|}{\sum_{i=1}^{n_k} |\hat{C}_k(X_i)|} \left[1\{\hat{C}_k(X_i) > 0\} - g_k(X_i)\right]^2 + h(g_1,\ldots,g_K)$$

Here the regularization function h is of the form of a sum of sparse group lasso and fused lasso penalty

$$h = (1-\alpha)\lambda_1\sqrt{q}\sum_{j=1}^{p} \|\boldsymbol{\beta_j}\|_2 + \alpha\lambda_1 \sum_{j=1}^{p} \|\boldsymbol{\beta_j}\|_1 + \lambda_2 \sum_{j=1}^{p} \sum_{1 \le a < b \le K} |\beta_{ja} - \beta_{jb}|$$

where $\boldsymbol{\beta_j} = (\beta_{j1},\ldots,\beta_{jK})$

If we would like a unique rule to be obtained, we should let $g_1 = \ldots = g_K$ and

$$h = \lambda_{uni}\|\beta\|_1$$

By setting $\lambda_1, \lambda_2, \alpha$ differently, different models could be obtained.

- If $\lambda_1, \lambda2 \neq 0$ and $alpha \neq 0$ or 1, we call it a complete meta-analysis model.
- If $\lambda_2 = 0$ and $\alpha \neq 0$ or 1, a sparse group lasso model is fitted.
- If $\lambda_2 = 0$ and $\alpha = 0$, a group lasso model is fitted.
- If $\lambda_2 = 0$ and $\alpha = 1$, a lasso model is fitted.
- If $\lambda_1, \lambda_2 = 0$, a linear model is fitted.

Value

an S3 object of class "mp", which contains the information of the fitted model. It could be supplied to the predict function

---

MetaPersonalized_cv *Meta-analysis for Personalized Medicine with Cross Validation*

Description

Meta-analysis for Personalized Medicine with Cross Validation

Usage

```
MetaPersonalized_cv(Xlist, Ylist, Trtlist, Plist, typelist, model = c("meta",
  "sparse group lasso", "group lasso", "lasso"), lambda1 = NULL,
  lambda2 = NULL, unique_rule_lambda = NULL, alpha = NULL,
  unique_rule = FALSE, cv_folds = 5)
```

Arguments

Xlist	a list object with kth element denoting the covariate matrix of study k
Ylist	a list object with kth element denoting the response vector of study k
Trtlist	a list object with kth element denoting the treatment vector of study k (coded as 0 or 1)
model	the model to be used for the above framework, can be meta-analysis, sparse group lasso, group lasso or lasso(linear does not need tuning)
lambda1	lambda1 in the framework above
lambda2	lambda2 in the framework above
unique_rule_lambda	
	λ_{uni} when unique treatment rule is required
alpha	alpha in the framework above
unique_rule	a logical value, whether a unique treatment rule is required
cv_folds	number of folds needed for cross-validation, default is 5
typelist	a list object with kth element denoting the type of response in study k, can be continuous or binary, default is continuous

Value

an S3 object of class "mp_cv", which contains the information of the model with the best fitted lambda. It can be supplied to the predict function.

---

predict.mp *Predict Function for "mp" object*

Description

This function predict the benefit scores and optimal treatment for new patients

Usage

```
## S3 method for class 'mp'
predict(mp, newx, weight = NULL, overall_rec = TRUE)
```

Arguments

mp	the fitted "mp" object
newx	the covariate matrix of the new patients
weight	a weight vector for the overall recommendation. If leave as NULL, a equally weighted recommendation will be made.
overall_rec	a logical value. If TRUE, an overall recommendation will be made weighted by the "weight" parameter.

Value

treatment	recommended treatment for each patient for each study/outcome. If overall_rec = TRUE, the weighted overall recommended treatment will be computed as well. If the overall recommened treatment is equal to 0.5, it means the sum of weight is equal for 0 and 1.
benefit_score	the benefit score computed from $g_1, \ldots, g_K$

---

predict.mp_cv *Predict Function for "mp_cv" object*

Description

This function predict the benefit scores and optimal treatment for new patients

Usage

```
## S3 method for class 'mp_cv'
predict(mp_cv, newx, weight = NULL, overall_rec = TRUE)
```

Arguments

mp_cv	the fitted "mp_cv" object
newx	the covariate matrix of the new patients
weight	a weight vector for the overall recommendation. If leave as NULL, a equally weighted recommendation will be made.
overall_rec	a logical value. If TRUE, an overall recommendation will be made weighted by the "weight" parameter.

Value

treatment   recommended treatment for each patient for each study/outcome. If `overall_rec = TRUE`, the weighted overall recommended treatment will be computed as well. If the overall recommened treatment is equal to 0.5, it means the sum of weight is equal for 0 and 1.

benefit_score   the benefit score computed from $g_1, \ldots, g_K$

---

simulated_dataset   *Simulated Dataset Generator*

Description

This function could generate a simulated dataset for the test and usage of this package.

Usage

```
simulated_dataset(n)
```

Arguments

n   sample size

Value

A simulated dataset.

Xlist   a list object with kth element denoting the covariate matrix of study k

Ylist   a list object with kth element denoting the response vector of study k

Trtlist   a list object with kth element denoting the treatment vector of study k (coded as 0 or 1)

Index