Package ‘BCBCSF’

October 12, 2022

Version 1.0-1

Title Bias-Corrected Bayesian Classification with Selected Features

Author Longhai Li <longhai@math.usask.ca>

Maintainer Longhai Li <longhai@math.usask.ca>

Depends R (>= 2.13.1), abind

Description Fully Bayesian Classification with a subset of high-dimensional features, such as expression levels of genes. The data are modeled with a hierarchical Bayesian models using heavy-tailed t distributions as priors. When a large number of features are available, one may like to select only a subset of features to use, typically those features strongly correlated with the response in training cases. Such a feature selection procedure is however invalid since the relationship between the response and the features has be exaggerated by feature selection. This package provides a way to avoid this bias and yield better-calibrated predictions for future cases when one uses F-statistic to select features.

License GPL (>= 2)

URL http://www.r-project.org, http://math.usask.ca/~longhai

NeedsCompilation yes

Repository CRAN

Date/Publication 2015-09-26 01:05:27

R topics documented:

- d1:bcbsfexamples ........................................ 2
- d2:fitpred .................................................. 3
- d3:evalpred ............................................... 6
- d4:analyzefit .......................................... 7
- d5:lymphoma ........................................... 8

Index 9
Examples of fitting models, predicting class labels, evaluating prediction, and analyzing fitting results

**Description**

These examples demonstrate how to use BCBCSF package. They use all prior and Markov chain sampling settings by default (except no_rmc as noted below). The methods for setting others can be found from documents for specific functions. However, the default settings may work well for a wide range of gene expression data.

**References**


**See Also**

`bcbcsf_fitpred`, `bcbcsf_pred`, `cross_vld`, `eval_pred`, `reload_fit_bcbcsf`, `bcbcsf_sumfit`, `bcbcsf_plotsumfit`

**Examples**

```r
## dontrun{
## load lymphoma microarray data
data (lymphoma)

## select some cases as testing data set
ts <- c (sort(sample (1:42,5)), 43:44, 61:62)

## training data
X_tr <- lymph.X[-ts,]
y_tr <- lymph.y[-ts]

## test data
X_ts <- lymph.X[ts,]
y_ts <- lymph.y[ts]

# fitting training data with top features selected by F-statistic
out_fit <- bcbcsf_fitpred (X_tr = X_tr, y_tr = y_tr, nos_fsel = c(20, 50), no_rmc = 100)

# note 1: if 'X_ts' is given above, prediction is made after fitting
# note 2: no_rmc = 100 is too small, omit it and use the default

# predicting class labels of test cases
out_pred <- bcbcsf_pred (X_ts = X_ts, out_fit = out_fit)
```
## evaluate prediction given true labels

eval_pred (out_pred = out_pred, y_ts = y_ts)

##########################################################################
# visualizing prediction results ##########################################
##########################################################################
## reload one bcbscf fit result from hardrive
fit_bcbscf <- reload_fit_bcbscf (out_fit$fitfiles[1])
## the fitting result for no_fsel = 50 can be retrieved directly from
## out_fit:
fit_bcbscf_fsel50 <- out_fit$fit_bcbscf
## summarize the fitting result
sum_fit <- bcbscf_sumfit (fit_bcbscf)

## visualize fitting result
bcbscf_plotsumfit (sum_fit)

##########################################################################
# cross validation ########################################################
##########################################################################
## doing cross validation with bcbscf_fitpred on lymphoma data
cv_pred <- cross_vld (
    fitpred_func = bcbscf_fitpred, X = lymph.X, y = lymph.y, nfold = 2,
    nos_fsel = c(20,50), no_rmc = 100 )
## note: no_rmc = 100 is too small, omit it and use the default in practice

## evaluate prediction given true labels
eval_pred (out_pred = cv_pred, y_ts = lymph.y)

## warning: this function is slow if nfold is large; if you have a
## computer cluster, you better parallel the cross validation folds.
##}

---

**d2:fitpred**

*Functions for fitting models with MCMC, predicting class labels of test cases, and finding predictive probabilities with cross-validation*

**Description**

bcbscf_fitpred trains models with Gibbs sampling for each number of retained features. The results are saved in files. This function also makes predictions for test cases if they are provided.

bcbscf_pred uses the posterior samples saved by bcbscf_fitpred to predict the class labels of test cases. Prediction results are an array of predictive probabilities array_probs_pred, whose rows for test cases, columns for classes, and the 3rd dimension for different numbers of retained features.

cross_vld uses cross-validation to obtain predictive probabilities for all cases of a data set. This generic function can be used with bcbscf_fitpred and other classifiers.
Usage

```r
cbcbcsf_fitpred (  
    ## arguments specifying info of data sets  
    X_tr, y_tr, nos_fsel = ncol(X_tr),  
    X_ts = NULL, standardize = FALSE, rankf = FALSE,  
    ## arguments for prediction  
    burn = NULL, thin = 1, offset_sdxj = 0.5,  
    ## arguments for Markov chain sampling  
    no_rmc = 1000, no_imc = 5, no_mhwmux = 10,  
    fit_cbcbcsf_filepre = ",.fitcbcbcsf_",  
    ## arguments specifying priors for parameters and hyperparameters  
    w0_mu = 0.05, alpha0_mu = 0.5, alpha1_mu = 3,  
    w0_x = 1.00, alpha0_x = 0.5, alpha1_x = 10,  
    w0_nu = 0.05, alpha0_nu = 0.5, prior_psi = NULL,  
    ## arguments for metropolis sampling for wmu, wx  
    stepadj_mhwmux = 1, diag_mhwmux = FALSE,  
    ## arguments for computing adjustment factor  
    bcor = 1, cut_qf = exp(-10), cut_dpoi = exp(-10), nos_sim = 1000,  
    ## whether look at progress  
    monitor = TRUE)  
```

```r
cbcbcsf_pred (X_ts, out_fit, burn = NULL, thin = 1, offset_sdxj = 0.5)  
```

```r
cross_vld (X, y, nfold = 10, folds = NULL,  
    fitpred_func = cbcbcsf_fitpred, ...)  
```

Arguments

- **X_tr, X_ts, X** matrices containing gene expression data; rows should be for the cases, and columns for different genes; `X_tr` are training data, `X_ts` are test data or future data for which prediction are needed, `X` are a data set used for cross-validation.

- **y_tr,y** class labels in training or test data set, or just a data set.

- **nos_fsel** a vector of numbers of features to be retained.

- **burn,thin** burn of Markov chain (super)iterations will be discarded for prediction, and only every `thin`th are used; by default, 20% of (super)iterations are burned, and `thin`=1.

- **offset_sdxj** a value between 0 and 1; 100*offset_sdxj% quantile of the samples of all standard deviations $\sqrt{w^2_{xj}}$ is added to the all standard deviations; this is to remedy the non-normality in real gene expression data sets, and especially offset some very small standard deviations; by default, median is used.

- **no_rmc, no_imc** no_rmc of super Markov chain transitions are run, with no_imc Markov chain iterations for each; only the last state of each super transition is saved.

- **fit_cbcbcsf_filepre** a string added to the names of files saving Markov chain fitting results; the actual file names contain also the data dimension and number of retained features;
when `fit_bcbcsf_filepre` is set to NULL, no fitting file will be created, and `bcbcsf_fitpred` returns only the fitting result corresponding to the last number of retained features in `nos_fsel`, which is always returned regardless of the value of `fit_bcbcsf_filepre`.

\[
\begin{align*}
&\text{\(w_0\mu,\alpha_0\mu,\alpha_1\mu, w_0\nu,\alpha_0\nu,\alpha_1\nu\)} \\
&\text{settings of priors for means and variances of genes; they are denoted by } w_0^\mu, \alpha_1^\mu, \\
&\text{\(\alpha_1^\nu, w_0^\nu, \alpha_0^\nu\)} \text{ in the reference.}
\end{align*}
\]

`prior_psi` a vector of length the number of classes, specifying the Dirichlet prior distribution for probabilities of classes; it is denoted by \(c_{1:G}\) in the reference; by default, they are all equal to 1.

`no_mhwx, stepadj_mhwx, diag_mhwx` arguments specifying Metropolis sampling for \(\log(w_\mu)\) and \(\log(w_x)\); respectively the number of iterations, stepsize adjustment, and an indicator representing whether one wants to pause and look into this sampling.

`bcor` taking value 0 or 1, indicating whether bias-correction is to be applied.

`cut_qf, cut_dpoi, nos_sim` arguments specifying approximation of adjustment factor; `cut_qf` is \(f_\ell\) in the reference, `cut_dpoi` is the threshold below which Poisson probabilities are omitted, `nos_sim` is the number of random \(\Lambda\).

`nfold, folds` folds should be a list of test cases for different folds; if `folds` is NULL (by default), `folds` will be generated by the software, with `nfold` is set to the smaller value of the given value and the smallest number of cases in all classes.

`out_fit` a list returned by `bcbcsf_fitpred`, which are used to make prediction for test cases.

`standardize` if it is set to TRUE, the original gene expression values are centralized and divided by the pooled standard deviation; by default, it is FALSE.

`rankf` if it is set to TRUE, the original features will be re-ordered by F-statistic; by default, it is FALSE.

`monitor` if it is set to TRUE, progress of fitting is shown on screen

`fitpred_func` an R function that can fit with training data, and predict for test data; the arguments of `fitpred_func` must include \(X_{tr}, y_{tr}, X_{ts}\), and the outputs of `fitpred_func` must include `array_probs_pred`

\[
\text{... arguments passed to classifier `fitpred_func`}
\]

**Value**

`nos_fsel` a vector of numbers of features retained.

`fitfiles` a string vector of length `nos_fsel`, each saving file name of Markov chain fitting result for a number of retained features in `nos_fsel`; the `fitfiles` returned by `cross_vld` is for the training in the last fold.

`array_probs_pred` an array of predictive probabilities, whose rows for test cases, columns for classes, and the 3rd dimension for different numbers of retained features.
fit_bcbcsf   a list of Markov chain sampling results from the fitting with number of retained features equal to the last number in nos_fsel. Note that, the fitting results for other numbers (including the last one) of retained feature are saved in harddrive files if fit_bcbcsf_filepre isn’t empty, and can be retrieved using function reload_fit_bcbcsf. Particularly, the list component of fit_bcbcsf has fsel saving the indice of features selected by F-statistic.

**d3:evalpred**  

*A function for evaluating arrays of predictive probabilities with the true class labels of test cases*

**Description**

This function is used to find error rate, amlp, loss and predictive probabilities at true labels.

**Usage**

`eval_pred(out_pred, y_ts, Mloss = NULL)`

**Arguments**

- `out_pred` a list returned by function bcbcsf_fitpred with X_ts given, or bcbcsf_pred, or by cross_vld.
- `y_ts` a vector of true class labels.
- `Mloss` a matrix indicting loss function, with element $m_{ij}$ saving the losss from predicting class $i$ with class label $j$; by default, it is NULL.

**Value**

- `probs_at_truelabels` a matrix of predictive probabilities at true labels, with rows for cases, and columns for different numbers of retained features
- `summary` a data frame, with rows for different numbers of retained features, and columns: Error.Rate: fraction of cases misclassified with fair threshold, and AMLP: minus average log probabilities at true labels, often called "deviation", and Loss (if Mloss is given): average loss.
d4:analyze fit

Functions for analyzing and visualizing a BCBCSF fitting result

Description

These functions are used to look at the fitting results, especially plot the gene signals.

Usage

reload_fit_bcbcsf (fit_bcbcsf_afile)

bcbcsf_sumfit (fit_bcbcsf = NULL, fit_bcbcsf_afile = NULL,
               burn = NULL, thin = 1)

bcbcsf_plotsumfit (sum_fit)

Arguments

fit_bcbcsf_afile
   a string of name of a file saving a Markov chain fitting result; it can be found from the value fitfiles of function bcbcsf_fitpred.

fit_bcbcsf
   a list of Markov chain fitting result, returned by function reload_fit_bcbcsf and bcbcsf_fitpred; if it is NULL, it will be retrieved by running reload_fit_bcbcsf with value in fit_bcbcsf_afile.

burn, thin
   burn of Markov chain (super)iterations will be discarded (burned) for evaluation, and only every thinth are used; by default, 20% of (super)iterations are burned, and thin=1.

sum_fit
   a list returned by function bcbcsf_sumfit

Value

reload_fit_bcbcsf returns a list of Markov chain fitting results, including how to do feature selection and data preprocessing.

bcbcsf_sumfit returns a list of point estimates of means and variances.

bcbcsf_plotsumfit returns nothing; it plots the normalized means (for each gene, original expression means subtracted by their means and divided by the common standard deviation), and overall signals (Euclid distance of normalized means) for the selected features.
Lymphoma Microarray Data

Description

This is one of the microarray data sets used to demonstrate BCBCSF in the reference article. Information about this data set can be found from the reference.

Usage

data(lymphoma)

Value

lymph.X  a matrix of gene expression data for p = 4026 genes on n = 62 cases in G=3 classes
lymph.y  a vector of class labels coded by 1,2,3.
Index

* classif
  d2:fitpred, 3
  d3:evalpred, 6

bcbsf_fitpred, 2, 7
bcbsf_fitpred(d2:fitpred), 3
bcbsf_plotsumfit, 2
bcbsf_plotsumfit(d4:analyzeft), 7
bcbsf_pred, 2
bcbsf_pred(d2:fitpred), 3
bcbsf_sumfit, 2
bcbsf_sumfit(d4:analyzeft), 7
bcbsfexamples (d1:bcbsfexamples), 2

cross_vld, 2
cross_vld(d2:fitpred), 3

d1:bcbsfexamples, 2
d2:fitpred, 3
d3:evalpred, 6
d4:analyzeft, 7
d5:lymphoma, 8

eval_pred, 2
eval_pred(d3:evalpred), 6

lymph.X (d5:lymphoma), 8
lymph.y (d5:lymphoma), 8
lymphoma (d5:lymphoma), 8

reload_fit_bcbsf, 2, 6
reload_fit_bcbsf(d4:analyzeft), 7