Package ‘effectFusion’

October 24, 2019

Version 1.1.2

Date 2019-10-24

Title Bayesian Effect Fusion for Categorical Predictors

Depends R (>= 3.3), mcclust

Imports Matrix, MASS, bayesm, cluster, GreedyEPL, gridExtra, ggplot2, methods, utils, stats

Description Variable selection and Bayesian effect fusion for categorical predictors in linear and logistic regression models. Effect fusion aims at the question which categories have a similar effect on the response and therefore can be fused to obtain a sparser representation of the model. Effect fusion and variable selection can be obtained either with a prior that has an interpretation as spike and slab prior on the level effect differences or with a sparse finite mixture prior on the level effects. The regression coefficients are estimated with a flat uninformative prior after model selection or by taking model averages. Posterior inference is accomplished by an MCMC sampling scheme which makes use of a data augmentation strategy (Polson, Scott & Windle (2013)) based on latent Polya-Gamma random variables in the case of logistic regression. The code for data augmentation is taken from Polson et al. (2013), who own the copyright.

License GPL-3

Encoding UTF-8

LazyData true

NeedsCompilation yes

RoxygenNote 6.1.1

Repository CRAN

Author Daniela Pauger [aut],
Magdalena Leitner [aut, cre],
Helga Wagner [aut] (<https://orcid.org/0000-0002-7003-9512>),
Gertraud Malsiner-Walli [aut] (<https://orcid.org/0000-0002-1213-4749>),
Nicholas G. Polson [ctb],
James G. Scott [ctb],
Jesse Windle [ctb],
Bettina Grün [ctb] (<https://orcid.org/0000-0001-7265-4773>)

Maintainer Magdalena Leitner <effectfusion.jku@gmail.com>

Date/Publication 2019-10-24 19:20:02 UTC

R topics documented:

`dic` *DIC*

Description

This function computes the DIC (deviance information criterion) for the estimated model in a `fusion` object.

Usage

```
dic(x)
```

Arguments

`x`	an object of class `fusion`

Details

The DIC can be easily computed from the MCMC output and is defined as $DIC = 2\overline{D(\theta)} - D(\bar{\theta})$, where $\overline{D(\theta)} = \frac{1}{M}\sum_{m=1}^{M} D(\theta^{(m)})$ is the average posterior deviance and $D(\bar{\theta})$ is the deviance evaluated at $\bar{\theta} = \frac{1}{M}\sum_{m=1}^{M} \theta^{(m)}$. $\theta^{(m)}$ are samples from the posterior of the model and M is the number of MCMC iterations.

Value

The DIC for the estimated model in the `fusion` object.

Author(s)

Daniela Pauger, Magdalena Leitner <effectfusion.jku@gmail.com>

References

Spiegelhalter, D., Best, N., Carlin, B., and van der Linde, A. (2002). Bayesian Measures of Model Complexity and Fit. *J. R. Statist. Soc. B*, **64(4)**, 583-639.

See Also

effectFusion

Examples

```
## see example for effectFusion
```

---

effectFusion *Bayesian effect fusion for categorical predictors*

Description

This function performs Bayesian variable selection and effect fusion for categorical predictors in linear and logistic regression models. Effect fusion aims at the question which categories of an ordinal or nominal predictor have a similar effect on the response and therefore can be fused to obtain a sparser representation of the model. Effect fusion and variable selection can be obtained either with a prior that has an interpretation as spike and slab prior on the level effect differences or with a sparse finite mixture prior on the level effects. The regression coefficients are estimated with a flat uninformative prior after model selection or by using model averaged results. For posterior inference, a MCMC sampling scheme is used that involves only Gibbs sampling steps. The sampling schemes for linear and logistic regression are almost identical as in the case of logistic regression a data augmentation strategy (Polson et al. (2013)) is used that requires only one additional step to sample from the Polya-Gamma distribution.

Usage

```
effectFusion(y, X, types, method, prior = list(), mcmc = list(),
  mcmcRefit = list(), family = "gaussian", modelSelection = "binder",
  returnBurnin = FALSE)
```

Arguments

y	a vector of the response observations (continuous if family = 'gaussian' and 0/1 if family = 'binomial')
X	a data frame with covariates, each column representing one covariate. Ordinal and nominal covariates should be of class factor
types	a character vector to specify the type of each covariate; 'c' indicates continuous or metric predictors, 'o' ordinal predictors and 'n' nominal predictors

`method` controls the main prior structure that is used for effect fusion. For the prior that has an interpretation as spike and slab prior on the level effect differences choose `method = 'SpikeSlab'` and for the sparse finite mixture prior on the level effects choose `method = 'FinMix'`. See details for a description of the two approaches and their advantages and drawbacks. For comparison purposes it is also possible to fit a full model without performing any effect fusion (`method = NULL`)

`prior` an (optional) list of prior settings and hyper-parameters for the prior (see details). The specification of this list depends on the chosen `method` and the selected `family`

`mcmc` an (optional) list of MCMC sampling options (see details)

`mcmcRefit` an (optional) list of MCMC sampling options for the refit of the selected model (see details)

`family` indicates whether linear (default, `family = 'gaussian'`) or logistic regression (`family = 'binomial'`) should be performed

`modelSelection` if `modelSelection = 'binder'` the final model is selected by minimising the expected posterior binder's loss using an algorithm of Lau and Green (2008) for the spike and slab model and an algorithm of Rastelli and Friel (2016) for the finite mixture approach. Alternatively, `modelSelection = 'pam'` can be specified for the sparse finite mixture model. In that case, the final model is selected by using pam clustering and the silhouette coefficient (see Malsiner-Walli et al., 2018 for details). If `modelSelection` is `NULL` no final model selection is performed and parameter estimates are model averaged results. `modelSelection = 'binder'` is the default value. `modelSelection = 'pam'` is only available for `method = 'FinMix'`. If `method = 'SpikeSlab'` and `modelSelection = 'pam'`, `modelSelection` is automatically set to `'binder'`. For the finite mixture approach we recommend to use `modelSelection = 'binder'`, as this algorithm provides - in contrast to pam clustering and the the silhouette coefficient - the opportunity to exclude whole covariates.

`returnBurnin` if `TRUE` (default is `FALSE`) the burn-in iterations of the MCMC sampling process are returned as well. This can be for example used to check convergence. Returning the burn-in does not influence the results of `dic`, `model`, `plot`, `print` and `summary`.

Details

This function provides identification of categories (of ordinal and nominal predictors) with the same effect on the response and their automatic fusion.

Two different prior versions for effect fusion and variable selection are available. The first prior version allows a priori for almost perfect as well as almost zero dependence between level effects. This prior has also an interpretation as independent spike and slab prior on all pairwise differences of level effects and correction for the linear dependence of the effect differences. Even though the prior is mainly designed for fusion of level effects, excluding some categories from the model as well as the whole covariate (variable selection) can also be easily accomplished. Excluding a category from the model corresponds to fusion of this category to the baseline and excluding the whole covariate consequently to fusion of all categories to the baseline.

The second prior is a modification of the usual spike and slab prior for the regression coefficients by combining a spike at zero with a finite location mixture of normal components. It enables detection of categories with similar effects on the response by clustering the regression effects. Categories with effects that are allocated to the same cluster are fused. Due to the specification with one component located at zero also automatic exclusion of levels and whole covariates without any effect on the outcome is provided. However, when using `modelSelection = 'pam'`, it is not possible to exclude whole covariates as the Silhouette coefficient does not allow for one cluster solutions. Therefore, we recommend to use `modelSelection = 'binder'`.

In settings with large numbers of categories, we recommend to use the sparse finite mixture prior for computational reasons. It is important to note that the sparse finite mixture prior on the level effects does not take into account the ordering information of ordinal predictors and treats them like nominal predictors, whereas in the spike and slab case fusion is restricted to adjacent categories for ordinal predictors.

Metric predictors can be included in the model as well and variable selection will be performed also for these predictors.

If `modelSelection = NULL`, no final model selection is performed and model averaged results are returned. When `modelSelection = 'binder'` the final model is selected by minimizing the expected posterior Binder loss for each covariate separately. Additionally, there is a second option for the finite mixture prior (`modelSelection = 'pam'`) which performs model selection by identifying the optimal partition of the effects using PAM clustering and the silhouette coefficient.

For comparison purposes it is also possible to fit a full model instead of performing effect fusion (`method = NULL`). All other functions provided in this package, such as `dic` or `summary`, do also work for the full model.

Details for the model specification (see arguments):

`prior` a list (depending on used `method` and specified `family`). If `method = NULL`, all prior specifications are ignored and a flat, uninformative prior is assigned to the level effects

- `r` variance ratio of slab to spike component; default to 50000 if `family = 'gaussian'` and 5000000 if `family = 'binomial'`. `r` should be chosen not too small but still small enough to avoid stickiness of MCMC. We recommend a value of at least 20000.
- `g0` shape parameter of inverse gamma distribution on τ^2 when `tau2_fix = NULL` and `method = 'SpikeSlab'`; default to 5. The default value is a standard choice in variable selection where the tails of spike and slab component are not too thin to cause mixing problems in MCMC.
- `G0` scale parameter of inverse gamma distribution on τ^2 when `tau2_fix = NULL` and `method = 'SpikeSlab'`; default to 25. `G0` controls to some extend the sparsity of the model. Smaller values for `G0` help to detect also small level effect differences, but result in less fusion of categories.
- `tau2_fix` If `tau2_fix = NULL`, an inverse gamma hyper-prior is specified on τ^2. However, the value of the slab variance can also be fixed for each covariate instead of using a hyperprior. `tau2_fix` is only of interest if `method = 'SpikeSlab'`. Default to `NULL`. Similar to the scale parameter `G0`, the fixed variance of the slab component `tau2_fix` can control to some extend the sparsity.
- `e0` parameter of Dirichlet hyper-prior on mixture weights when `method = 'FinMix'`; default to 0.01. `e0` should be chosen smaller than 1 in order to encourage empty components. Small values such as 0.01 help to concentrate the model space on sparse solutions.

p prior parameter to control mixture component variances when method = 'FinMix'; values between 100 and 100000 led to good results in simulation studies; default to 100 if family = 'gaussian' and 1000 if family = 'binomial'. We recommend to try different values for this prior parameter and compare the models using dic. When hyperprior = FALSE, larger values of p lead to less sparsity and it should be chosen not smaller than 100. If a hyper-prior on the mixture component variances is used (hyperprior = TRUE), p has almost no effect on the sparsity of the model and it should again be not smaller than 100.

hyperprior logical value if inverse gamma hyper-prior on component variance should be specified when method = 'FinMix'; default to FALSE. The hyper-prior leads to robust results concerning the specification of p but also to very sparse solutions.

s0 hyper-parameter (shape) of inverse gamma distribution on error variance, used for both versions of method, but only for family = 'gaussian'; default to 0.

S0 hyper-parameter (scale) of inverse gamma distribution on error variance, used for both versions of method, but only for family = 'gaussian'; default to 0.

mcmc a list:

M number of MCMC iterations after the burn-in phase; default to 20000 for effect fusion models and 3000 for full models.

burnin number of MCMC iterations discarded as burn-in; default to 5000 for effect fusion models and 1000 for full models.

startsel number of MCMC iterations drawn from the model without performing effect fusion; default to 1000 for effect fusion models and 0 for full models.

mcmcRefit a list (not necessary if modelSelection = NULL or method = NULL):

M_refit number of MCMC iterations after the burn-in phase for the refit of the selected model; default to 3000.

burnin_refit number of MCMC iterations discarded as burn-in for the refit of the selected model; default to 1000.

Value

The function returns an object of class fusion with methods dic, model, print, summary and plot.

An object of class fusion is a named list containing the following elements:

fit a named list containing the samples from the posterior distributions of the parameters depending on the used prior structure (method = 'SpikeSlab', method = 'FinMix' or method = NULL):

beta regression coefficients β_0 (intercept) and β

delta indicator variable δ for slab component when method = 'SpikeSlab'. The differences of the level effects are assigned either to the spike (delta = 0) or the slab component (delta = 1). If an effect difference is assigned to the spike component, the difference is almost zero and the corresponding level effects should be fused.

tau2 variance τ^2 of slab component when method = 'SpikeSlab'. If no hyperprior on τ^2 is specified, tau2 contains the fixed values for τ^2.

S latent allocation variable S for mixture components when method = 'FinMix'

eta mixture component weights η when method = 'FinMix'

eta0 weights of components located at zero η_0 when method = 'FinMix' mu mixture component means μ when method = 'FinMix'

sgma2 error variance σ^2 of the model (only for family = 'gaussian')

fit_burnin a named list containing the same elements as fit including the burnin-phase, if returnBurnin = TRUE, NULL otherwise. The elements that correspond to the model selection procedure, e.g. delta or S, are NA for the first startsel iterations.

refit a named list containing samples from the posterior distributions of the parameters of the model refit (only if method and modelSelection are unequal to NULL):

beta regression coefficients including the intercept in the model with fused levels

sgma2 error variance of the model with fused levels (only for family = 'gaussian')

X_dummy_fused the dummy coded design matrix with fused levels

model vector of zeros and ones representing the selected model based on pairs of categories

method see arguments

family see arguments

data a named list containing the data y, X, the dummy coded design matrix X_dummy, types and levelnames of ordinal and nominal predictors

model a named list containing information on the full, initial model

categories number of categories for categorical predictors

diff number of pairwise level effect differences

n_cont number of metric predictors

n_ord number of ordinal predictors

n_nom number of nominal predictors

prior see details for prior

mcmc see details for mcmc

mcmcRefit see details for mcmcRefit

modelSelection see arguments

returnBurnin see arguments

numbCoef number of estimated regression coefficients (based on the refitted model if effect fusion and final model selection is performed, otherwise based on model averaged results or the full model, respectively)

call function call

Note

The function can be used for ordinal and/or nominal predictors and metric covariates can additionally be included in the model. Binary covariates as a special case of nominal predictors can be included as well.

The sparse finite mixture prior approach does not take into account the ordering information of ordinal predictors. Ordinal predictors are treated as nominal predictors, whereas in the spike and slab case fusion is restricted to adjacent categories for ordinal predictors.

For large models and more than 15,000 MCMC iterations, some thinning of the MCMC when using the sparse finite mixture prior is performed due to computational issues.

Author(s)

Daniela Pauger, Magdalena Leitner <effectfusion.jku@gmail.com>, Helga Wagner, Gertraud Malsiner-Walli

References

Pauger, D., and Wagner, H. (2018). Bayesian Effect Fusion for Categorical Predictors. *Bayesian Analysis*, in print.

Malsiner-Walli, G., Pauger, D., and Wagner, H. (2018). Effect Fusion Using Model-Based Clustering. *Statistical Modelling*, **18(2)**, 175-196.

Polson, N.G., Scott, J.G., and Windle, J. (2013). Bayesian Inference for Logistic Models Using Polya-Gamma Latent Variables. *Journal of the American Statistical Association*, **108(504)**, 1339-1349.

Examples

```
## Not run:
# ----------- Load simulated data set 'sim1' for linear regression
data(sim1)
y = sim1$y
X = sim1$X
types = sim1$types

# ----------- Bayesian effect fusion for simulated data set with spike and slab prior
m1 <- effectFusion(y, X, types, method = 'SpikeSlab')

# print, summarize and plot results
print(m1)
summary(m1)
plot(m1)

# evaluate model and model criteria
model(m1)
dic(m1)

# ----------- Use finite mixture prior for comparison
m2 <- effectFusion(y, X, types, method = 'FinMix')

# summarize and plot results
print(m2)
summary(m2)
plot(m2)
model(m2)
dic(m2)

# change prior parameter specification
m3 <- effectFusion(y, X, types, prior= list(p = 10^3), method = 'FinMix')
plot(m3)

# ------------  Use model averaged coefficient estimates
m4 <- effectFusion(y, X, types, method = 'SpikeSlab', modelSelection = NULL)
```

```
summary(m4)

# ------------  Estimate full model for comparison purposes
m5 <- effectFusion(y, X, types, method = NULL)
summary(m5)
plot(m5)
dic(m5)

# ----------- Load simulated data set 'sim3' for logistic regression
data(sim3)
y = sim3$y
X = sim3$X
types = sim3$types

# ----------- Bayesian effect fusion for simulated data set with finite mixture prior
m6 <- effectFusion(y, X, types, method = 'FinMix', prior = list(p = 10^4), family = 'binomial')

# look at the results
print(m6)
summary(m6)
plot(m6)
model(m6)
dic(m6)

# ----------- Use spike and slab prior for comparison
m7 <- effectFusion(y, X, types, method = 'SpikeSlab', family = 'binomial', returnBurnin = TRUE)

# summarize and evaluate results
print(m7)
summary(m7)
plot(m7)
model(m7)
dic(m7)

## End(Not run)
```

`model` *Selected model of a* `fusion` *object*

Description

The function displays for categorical covariates the selected model of an object of class `fusion` as list.

Usage

```
model(x)
```

Arguments

`x`	an object of class `fusion`

Details

The selected model for each categorical predictor is displayed as a list of length equal to the number of categories after fusion. Fused categories are shown with their original labelling in one list element. The function is only available if effect fusion (method in `effectFusion` is unequal to `NULL`) and final model selection (argument `modelSelection` in `effectFusion` is not NULL) is performed.

See `summary.fusion` for more details.

Author(s)

Daniela Pauger, Magdalena Leitner <effectfusion.jku@gmail.com>

See Also

`effectFusion`

Examples

```
## see example for effectFusion
```

`plot.fusion` *Plot an object of class* `fusion`

Description

This function provides plots of posterior means and 95%-HPD intervals for the regression effects. Plots are based on the refitted MCMC samples of the selected model in an object of class `fusion` or on model averaged results if no final model selection was performed.

Usage

```
## S3 method for class 'fusion'
plot(x, maxPlots = 4, ...)
```

Arguments

`x`	an object of class `fusion`
`maxPlots`	maximum number of plots on a single page, default argument to 4
`...`	further arguments passed to or from other methods (not used)

Details

If no effect fusion or no final model selection is performed, posterior means and HPD intervals are model averaged results. Otherwise, the parameters of the selected model are reestimated with a flat uninformative prior. Thus, fused categories have the same posterior mean and HPD interval. Single categories that are excluded from the model are fused to the reference category and therefore only a posterior mean at zero and no interval is plotted.

Author(s)

Daniela Pauger, Magdalena Leitner <effectfusion.jku@gmail.com>

See Also

`effectFusion`

Examples

```
## see example for effectFusion
```

`print.fusion` *Print object of class* `fusion`

Description

The default print method for a `fusion` object.

Usage

```
## S3 method for class 'fusion'
print(x, ...)
```

Arguments

`x`	an object of class `fusion`
`...`	further arguments passed to or from other methods (not used)

Details

Returns basic information about the full, initial model: the number of observations, the family, the number and types of used covariates with their number of categories and MCMC options.

Author(s)

Daniela Pauger, Magdalena Leitner <effectfusion.jku@gmail.com>

See Also

`effectFusion`

Examples

```
## see example for effectFusion
```

sim1 *Simulated data set 1*

Description

The simulated data set sim1 illustrates a setting with 500 observations from a linear regression model with normal response, 4 ordinal and 4 nominal predictors. Two regressors have 8 and two have 4 categories for each type of covariate (ordinal and nominal). Regression effects are set to $\beta_1 = (0, 1, 1, 2, 2, 4, 4)$ and $\beta_3 = (0, -2, -2)$ for the ordinal and $\beta_5 = (0, 1, 1, 1, 1, -2, -2)$ and $\beta_7 = (0, 2, 2)$ for the nominal covariates, and $\beta_h = 0$ for h = 2, 4, 6, 8. Levels of the predictors are generated with probabilities $\pi_h = (0.1, 0.1, 0.2, 0.05, 0.2, 0.1, 0.2, 0.05)$ and $\pi_h = (0.1, 0.4, 0.2, 0.3)$ for regressors with 8 and 4 levels, respectively. For more details on the simulation setting see Pauger and Wagner (2018).

Usage

```
data(sim1)
```

Format

A named list containing the following four variables:

y vector with 500 observations of a normal response variable

X matrix with 8 categorical predictors

beta vector with coefficients used for data generation

types character vector with types of covariates, 'o' for ordinal and 'n' for nominal covariates

References

Pauger, D., and Wagner, H. (2018). Bayesian Effect Fusion for Categorical Predictors. *Bayesian Analysis*, in print.

See Also

effectFusion

sim2 *Simulated data set 2*

Description

The simulated data set `sim2` illustrates a setting with 4000 observations from a linear regression model. The model has four independent predictors with either 10 or 100 categories and uniform prior class probabilities. The first covariate with 10 categories has three levels with no effects, three levels with effects of size 0.5 and the remaining three levels have effects of size one. The second covariate with 10 categories has 8 levels with no effects and only one level with an effect of size one. The final covariate with 10 categories has only levels without any effect on the outcome. Analogue to the first one, the covariate with 100 categories has 33 levels with no effects, 33 levels with effects of size 0.5 and 33 levels with effects of size 1.

Usage

```
data(sim2)
```

Format

A named list containing the following four variables:

`y` vector with 4000 observations of a normal response variable

`X` matrix with 4 categorical predictors

`beta` vector with coefficients used for data generation

`types` character vector with types of covariates, 'o' for ordinal and 'n' for nominal covariates

See Also

`effectFusion`

sim3 *Simulated data set 3*

Description

The simulated data set `sim3` considers a setting with 2000 observations from a logistic regression model. The number and types of predictors, the regression effects and the level probabilites of the predictors are the same as for `sim1`. The number of observations was increased as the uncertainty is usually higher for logistic regression compared to linear regression with normal response.

Usage

```
data(sim3)
```

Format

A named list containing the following four variables:

`y` vector with 2000 observations of a binary response variable

`X` matrix with 8 categorical predictors

`beta` vector with coefficients used for data generation

`types` character vector with types of covariates, 'o' for ordinal and 'n' for nominal covariates

See Also

`effectFusion`,`sim1`

`summary.fusion` *Summary of object of class* `fusion`

Description

Returns basic information about the model and the priors, MCMC details and posterior means from the refit of the selected model or model averaged results as well as 95%-HPD intervals for the regression effects.

Usage

```
## S3 method for class 'fusion'
summary(object, ...)
```

Arguments

`object`	an object of class `fusion`
`...`	further arguments passed to or from other methods (not used)

Details

The model selected with function `effectFusion` is refitted with a flat uninformative prior to get estimates for the regression coefficients `beta`. The posterior means and 95%-HPD intervals resulting from this refit are shown with this function. Fused categories have the same regression coefficient estimates and the same HPD intervals.

If a full model is fitted (`method` in `effectFusion` is NULL) or no final model selection is performed (argument `modelSelection` in `effectFusion` is NULL), the coefficient estimates are model averaged results.

Author(s)

Daniela Pauger, Magdalena Leitner <effectfusion.jku@gmail.com>

See Also

`effectFusion`

Examples

```
## see example for effectFusion
```

Index