Package 'DIBclust'

June 27, 2025

Type Package

Title Deterministic Information Bottleneck Method for Clustering Mixed-Type Data

Version 0.1.2

Author Efthymios Costa, Ioanna Papatsouma, Angelos Markos

Maintainer Angelos Markos <amarkos@eled.duth.gr>

Description Implements the Deterministic Information Bottleneck method for clustering datasets with both categorical and continuous variables. The package handles data preprocessing, feature selection, and clustering optimization using information-theoretic approaches <doi:10.48550/arXiv.2407.03389>.

License MIT + file LICENSE

Encoding UTF-8

Imports Rcpp, RcppEigen, stats, utils, np, rje

LinkingTo Rcpp, RcppArmadillo, RcppEigen

RoxygenNote 7.3.2

NeedsCompilation yes

Archs x64

Contents

`DIBcat` *Cluster Categorical Data Using the Deterministic Information Bottleneck Algorithm*

Description

The DIBcat function implements the Deterministic Information Bottleneck (DIB) algorithm for clustering datasets containing categorical variables. This method balances information retention and data compression to create meaningful clusters, leveraging bandwidth parameters to handle different categorical data types (nominal and ordinal) effectively (Costa, Papatsouma & Markos, 2024).

Usage

```
DIBcat(X, ncl, randinit = NULL, lambda = -1,
       maxiter = 100, nstart = 100, select_features = FALSE,
       verbose = FALSE)
```

Arguments

X	A data frame containing the categorical data to be clustered. All variables should be categorical, either factor (for nominal variables) or ordered (for ordinal variables).
ncl	An integer specifying the number of clusters to form.
randinit	Optional. A vector specifying initial cluster assignments. If NULL, cluster assignments are initialized randomly.
lambda	A numeric value or vector specifying the bandwidth parameter for categorical variables. The default value is -1, which enables automatic determination of the optimal bandwidth. For nominal variables, the maximum allowable value of lambda is $(l-1)/l$, where l represents the number of categories. For ordinal variables, the maximum allowable value of lambda is 1.
maxiter	The maximum number of iterations for the clustering algorithm. Defaults to 100.
nstart	The number of random initializations to run. The best clustering result (based on the information-theoretic criterion) is returned. Defaults to 100.
select_features	Logical. If TRUE, uses an eigengap heuristic for feature selection, potentially improving clustering quality by reducing dimensionality. Defaults to FALSE.
verbose	Logical. Default to FALSE to suppress progress messages. Change to TRUE to print.

Details

The DIBcat function applies the Deterministic Information Bottleneck algorithm to cluster datasets containing only categorical variables, both nominal and ordinal. The algorithm optimizes an information-theoretic objective to balance the trade-off between data compression and the retention of information about the original distribution.

To estimate the distributions of categorical features, the function utilizes specialized kernel functions, as follows:

$$K_u(x = x'; \lambda) = \begin{cases} 1-\lambda, & \text{if } x = x' \\ \frac{\lambda}{\ell-1}, & \text{otherwise} \end{cases}, \quad 0 \leq \lambda \leq \frac{\ell-1}{\ell},$$

where ℓ is the number of categories, and λ controls the smoothness of the Aitchison & Aitken kernel for nominal variables.

$$K_o(x = x'; \nu) = \begin{cases} 1, & \text{if } x = x' \\ \nu^{|x-x'|}, & \text{otherwise} \end{cases}, \quad 0 \leq \nu \leq 1,$$

where ν is the bandwidth parameter for ordinal variables, accounting for the ordinal relationship between categories (Li & Racine kernel).

Here, λ, and ν are bandwidth or smoothing parameters, while ℓ is the number of levels of the categorical variable. The lambda parameter is automatically determined by the algorithm if not provided by the user. For ordinal variables, the lambda parameter of the function is used to define ν.

Value

A list containing the following elements:

- `Cluster`: An integer vector indicating the cluster assignment for each data point at convergence.
- `Entropy`: A numeric value representing the entropy of the cluster assignments at the end of the iterative procedure.
- `MutualInfo`: A numeric value representing the mutual information, $I(Y;T)$, between the data distribution and the cluster assignments.
- `lambda`: A numeric vector of bandwidth parameters for categorical variables, controlling how categories are weighted in the clustering.
- `beta`: A numeric vector of the final beta values used during the iterative optimization.
- `ents`: A numeric vector tracking the entropy values across iterations, providing insights into the convergence pattern.
- `mis`: A numeric vector tracking the mutual information values across iterations.

Author(s)

Efthymios Costa, Ioanna Papatsouma, Angelos Markos

References

Costa, E., Papatsouma, I., & Markos, A. (2024). A Deterministic Information Bottleneck Method for Clustering Mixed-Type Data. *arXiv:2407.03389 [stat.ME]*. Retrieved from https://arxiv.org/abs/2407.03389

Aitchison, J., & Aitken, C. G. G. (1976). Multivariate binary discrimination by the kernel method. *Biometrika*, *63*(3), 413-420.

Li, Q., & Racine, J. (2003). Nonparametric estimation of distributions with categorical and continuous data. *Journal of Multivariate Analysis*, *86*(2), 266-292.

See Also

`DIBmix`, `DIBcont`

Examples

```
# Simulated categorical data
set.seed(123)
X <- data.frame(
  Var1 = as.factor(sample(letters[1:3], 200, replace = TRUE)),  # Nominal variable
  Var2 = as.factor(sample(letters[4:6], 200, replace = TRUE)),  # Nominal variable
```

```
  Var3 = factor(sample(c("low", "medium", "high"), 200, replace = TRUE),
                levels = c("low", "medium", "high"), ordered = TRUE)  # Ordinal variable
)

# Run DIBcat with automatic lambda selection and multiple initializations
result <- DIBcat(X = X, ncl = 3, lambda = -1, nstart = 50)

# Print clustering results
print(result$Cluster)       # Cluster assignments
print(result$Entropy)       # Final entropy
print(result$MutualInfo)    # Mutual information
```

`DIBcont` *Cluster Continuous Data Using the Deterministic Information Bottleneck Algorithm*

Description

The `DIBcont` function implements the Deterministic Information Bottleneck (DIB) algorithm for clustering continuous data. This method optimizes an information-theoretic objective to preserve relevant information while forming concise and interpretable cluster representations (Costa, Papatsouma & Markos, 2024).

Usage

```
DIBcont(X, ncl, randinit = NULL, s = -1, scale = TRUE,
        maxiter = 100, nstart = 100, select_features = FALSE,
        verbose = FALSE)
```

Arguments

`X`	A numeric matrix or data frame containing the continuous data to be clustered. All variables should be of type `numeric`.
`ncl`	An integer specifying the number of clusters to form.
`randinit`	Optional. A vector specifying initial cluster assignments. If `NULL`, cluster assignments are initialized randomly.
`s`	A numeric value or vector specifying the bandwidth parameter(s) for continuous variables. The values must be greater than 0. The default value is -1, which enables the automatic selection of optimal bandwidth(s).
`scale`	A logical value indicating whether the continuous variables should be scaled to have unit variance before clustering. Defaults to `TRUE`.
`maxiter`	The maximum number of iterations allowed for the clustering algorithm. Defaults to 100.
`nstart`	The number of random initializations to run. The best clustering result (based on the information-theoretic criterion) is returned. Defaults to `100`.
`select_features`	Logical. If `TRUE`, uses an eigengap heuristic for feature selection, potentially improving clustering quality by reducing dimensionality. Defaults to `FALSE`.
`verbose`	Logical. Default to `FALSE` to suppress progress messages. Change to `TRUE` to print.

Details

The `DIBcont` function applies the Deterministic Information Bottleneck algorithm to cluster datasets comprising only continuous variables. This method leverages an information-theoretic objective to optimize the trade-off between data compression and the preservation of relevant information about the underlying data distribution.

The function utilizes the Gaussian kernel (Silverman, 1998) for estimating probability densities of continuous features. The kernel is defined as:

$$K_c\left(\frac{x - x'}{s}\right) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{(x - x')^2}{2s^2}\right\}, \quad s > 0.$$

The bandwidth parameter s, which controls the smoothness of the density estimate, is automatically determined by the algorithm if not provided by the user.

Value

A list containing the following elements:

- `Cluster`: An integer vector indicating the cluster assignment for each observation.
- `Entropy`: A numeric value representing the entropy of the cluster assignments at convergence.
- `MutualInfo`: A numeric value representing the mutual information, $I(Y;T)$, between the underlying data distribution and the cluster assignments.
- `beta`: A numeric vector of the final beta values used during the iterative optimization.
- `s`: A numeric value or vector of bandwidth parameters used for the continuous variables. Typically, this will be a single value if all continuous variables share the same bandwidth.
- `ents`: A numeric vector tracking the entropy values over the iterations, providing insight into the convergence process.
- `mis`: A numeric vector tracking the mutual information values over the iterations.

Author(s)

Efthymios Costa, Ioanna Papatsouma, Angelos Markos

References

Costa, E., Papatsouma, I., & Markos, A. (2024). A Deterministic Information Bottleneck Method for Clustering Mixed-Type Data. *arXiv:2407.03389 [stat.ME]*. Retrieved from https://arxiv.org/abs/2407.03389

Silverman, B. W. (1998). Density estimation for statistics and data analysis (1st ed.). Routledge.

See Also

`DIBmix`, `DIBcat`

Examples

```
# Generate simulated continuous data
set.seed(123)
X <- matrix(rnorm(1000), ncol = 5)  # 200 observations, 5 features

# Run DIBcont with automatic bandwidth selection and multiple initializations
result <- DIBcont(X = X, ncl = 3, s = -1, nstart = 50)
```

```
# Print clustering results
print(result$Cluster)       # Cluster assignments
print(result$Entropy)       # Final entropy
print(result$MutualInfo)    # Mutual information
```

DIBmix *Deterministic Information Bottleneck Clustering for Mixed-Type Data*

Description

The DIBmix function implements the Deterministic Information Bottleneck (DIB) algorithm for clustering datasets containing mixed-type variables, including categorical (nominal and ordinal) and continuous variables. This method optimizes an information-theoretic objective to preserve relevant information in the cluster assignments while achieving effective data compression (Costa, Papatsouma & Markos, 2024).

Usage

```
DIBmix(X, ncl, catcols, contcols, randinit = NULL,
       lambda = -1, s = -1, scale = TRUE,
       maxiter = 100, nstart = 100,
       select_features = FALSE,
       verbose = FALSE)
```

Arguments

X	A data frame containing the input data to be clustered. It should include categorical variables (factor for nominal and Ord.factor for ordinal) and continuous variables (numeric).
ncl	An integer specifying the number of clusters.
catcols	A vector indicating the indices of the categorical variables in X.
contcols	A vector indicating the indices of the continuous variables in X.
randinit	An optional vector specifying the initial cluster assignments. If NULL, cluster assignments are initialized randomly.
lambda	A numeric value or vector specifying the bandwidth parameter for categorical variables. The default value is -1, which enables automatic determination of the optimal bandwidth. For nominal variables, the maximum allowable value of lambda is $(l-1)/l$, where l represents the number of categories. For ordinal variables, the maximum allowable value of lambda is 1.
s	A numeric value or vector specifying the bandwidth parameter(s) for continuous variables. The values must be greater than 0. The default value is -1, which enables the automatic selection of optimal bandwidth(s).
scale	A logical value indicating whether the continuous variables should be scaled to have unit variance before clustering. Defaults to TRUE.
maxiter	The maximum number of iterations allowed for the clustering algorithm. Defaults to 100.
nstart	The number of random initializations to run. The best clustering solution is returned. Defaults to 100.

`select_features`
: A logical value indicating whether to perform feature selection based on the eigengap heuristic. Defaults to `FALSE`.

`verbose`
: Logical. Default to `FALSE` to suppress progress messages. Change to `TRUE` to print.

Details

The `DIBmix` function clusters data while retaining maximal information about the original variable distributions. The Deterministic Information Bottleneck algorithm optimizes an information-theoretic objective that balances information preservation and compression. Bandwidth parameters for categorical (nominal, ordinal) and continuous variables are adaptively determined if not provided. This iterative process identifies stable and interpretable cluster assignments by maximizing mutual information while controlling complexity. The method is well-suited for datasets with mixed-type variables and integrates information from all variable types effectively.

The following kernel functions are used to estimate densities for the clustering procedure:

- *Continuous variables: Gaussian kernel*

$$K_c\left(\frac{x-x'}{s}\right) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{(x-x')^2}{2s^2}\right\}, \quad s > 0.$$

- *Nominal categorical variables: Aitchison & Aitken kernel*

$$K_u\left(x = x'; \lambda\right) = \begin{cases} 1-\lambda & \text{if } x = x' \\ \frac{\lambda}{\ell-1} & \text{otherwise} \end{cases}, \quad 0 \leq \lambda \leq \frac{\ell-1}{\ell}.$$

- *Ordinal categorical variables: Li & Racine kernel*

$$K_o\left(x = x'; \nu\right) = \begin{cases} 1 & \text{if } x = x' \\ \nu^{|x-x'|} & \text{otherwise} \end{cases}, \quad 0 \leq \nu \leq 1.$$

Here, s, λ, and ν are bandwidth or smoothing parameters, while ℓ is the number of levels of the categorical variable. s and λ are automatically determined by the algorithm if not provided by the user. For ordinal variables, the lambda parameter of the function is used to define ν.

Value

A list with the following elements:

`Cluster`
: An integer vector giving the cluster assignments for each data point.

`Entropy`
: A numeric value representing the entropy of the cluster assignments at convergence.

`MutualInfo`
: A numeric value representing the mutual information, $I(Y;T)$, between the original labels (Y) and the cluster assignments (T).

`beta`
: A numeric vector of the final beta values used in the iterative procedure.

`s`
: A numeric vector of bandwidth parameters used for the continuous variables.

`lambda`
: A numeric vector of bandwidth parameters used for the categorical variables.

`ents`
: A numeric vector tracking the entropy values across iterations.

`mis`
: A numeric vector tracking the mutual information values across iterations.

Author(s)

Efthymios Costa, Ioanna Papatsouma, Angelos Markos

References

Aitchison, J., & Aitken, C. G. G. (1976). Multivariate binary discrimination by the kernel method. *Biometrika*, *63*(3), 413-420.

Costa, E., Papatsouma, I., & Markos, A. (2024). A Deterministic Information Bottleneck Method for Clustering Mixed-Type Data. *arXiv:2407.03389 [stat.ME]*. Retrieved from https://arxiv.org/abs/2407.03389

Silverman, B. W. (1998). Density estimation for statistics and data analysis (1st ed.). Routledge.

Li, Q., & Racine, J. (2003). Nonparametric estimation of distributions with categorical and continuous data. *Journal of Multivariate Analysis*, *86*(2), 266-292.

See Also

DIBcont, DIBcat

Examples

```
# Example dataset with categorical, ordinal, and continuous variables
data <- data.frame(
 cat_var = factor(sample(letters[1:3], 100, replace = TRUE)),    # Nominal categorical variable
  ord_var = factor(sample(c("low", "medium", "high"), 100, replace = TRUE),
                   levels = c("low", "medium", "high"),
                   ordered = TRUE),                                  # Ordinal variable
  cont_var1 = rnorm(100),                                          # Continuous variable 1
  cont_var2 = runif(100)                                           # Continuous variable 2
)

# Perform Mixed-Type Clustering
result <- DIBmix(X = data, ncl = 3, catcols = 1:2, contcols = 3:4)

# Print clustering results
print(result$Cluster)       # Cluster assignments
print(result$Entropy)       # Final entropy
print(result$MutualInfo)    # Mutual information
```

IBmix *Information Bottleneck Clustering for Mixed-Type Data*

Description

The IBmix function implements the Information Bottleneck (IB) algorithm for clustering datasets containing mixed-type variables, including categorical (nominal and ordinal) and continuous variables. This method optimizes an information-theoretic objective to preserve relevant information in the cluster assignments while achieving effective data compression (Costa, Papatsouma & Markos, 2024).

Usage

```
IBmix(X, ncl, beta, catcols, contcols, randinit = NULL,
      lambda = -1, s = -1, scale = TRUE,
      maxiter = 100, nstart = 100,
      select_features = FALSE,
      verbose = FALSE)
```

Arguments

X	A data frame containing the input data to be clustered. It should include categorical variables (`factor` for nominal and `Ord.factor` for ordinal) and continuous variables (`numeric`).
ncl	An integer specifying the number of clusters.
beta	Regularisation strength.
catcols	A vector indicating the indices of the categorical variables in `X`.
contcols	A vector indicating the indices of the continuous variables in `X`.
randinit	An optional vector specifying the initial cluster assignments. If `NULL`, cluster assignments are initialized randomly.
lambda	A numeric value or vector specifying the bandwidth parameter for categorical variables. The default value is -1, which enables automatic determination of the optimal bandwidth. For nominal variables, the maximum allowable value of `lambda` is $(l-1)/l$, where l represents the number of categories. For ordinal variables, the maximum allowable value of `lambda` is 1.
s	A numeric value or vector specifying the bandwidth parameter(s) for continuous variables. The values must be greater than 0. The default value is -1, which enables the automatic selection of optimal bandwidth(s).
scale	A logical value indicating whether the continuous variables should be scaled to have unit variance before clustering. Defaults to `TRUE`.
maxiter	The maximum number of iterations allowed for the clustering algorithm. Defaults to 100.
nstart	The number of random initializations to run. The best clustering solution is returned. Defaults to 100.
select_features	A logical value indicating whether to perform feature selection based on the eigengap heuristic. Defaults to `FALSE`.
verbose	Logical. Default to `FALSE` to suppress progress messages. Change to `TRUE` to print.

Details

The `IBmix` function produces a fuzzy clustering of the data while retaining maximal information about the original variable distributions. The Information Bottleneck algorithm optimizes an information-theoretic objective that balances information preservation and compression. Bandwidth parameters for categorical (nominal, ordinal) and continuous variables are adaptively determined if not provided. This iterative process identifies stable and interpretable cluster assignments by maximizing mutual information while controlling complexity. The method is well-suited for datasets with mixed-type variables and integrates information from all variable types effectively.

The following kernel functions are used to estimate densities for the clustering procedure:

- *Continuous variables: Gaussian kernel*

$$K_c\left(\frac{x-x'}{s}\right) = \frac{1}{\sqrt{2\pi}}\exp\left\{-\frac{(x-x')^2}{2s^2}\right\}, \quad s > 0.$$

- *Nominal categorical variables: Aitchison & Aitken kernel*

$$K_u\left(x = x'; \lambda\right) = \begin{cases} 1-\lambda & \text{if } x = x' \\ \frac{\lambda}{\ell-1} & \text{otherwise} \end{cases}, \quad 0 \leq \lambda \leq \frac{\ell-1}{\ell}.$$

- *Ordinal categorical variables: Li & Racine kernel*

$$K_o\left(x = x'; \nu\right) = \begin{cases} 1 & \text{if } x = x' \\ \nu^{|x-x'|} & \text{otherwise} \end{cases}, \quad 0 \leq \nu \leq 1.$$

Here, s, λ, and ν are bandwidth or smoothing parameters, while ℓ is the number of levels of the categorical variable. s and λ are automatically determined by the algorithm if not provided by the user. For ordinal variables, the lambda parameter of the function is used to define ν.

Value

A list with the following elements:

`Cluster`	A cluster membership matrix.
`InfoXT`	A numeric value representing the mutual information, $I(X;T)$, between the original observations weights (X) and the cluster assignments (T).
`InfoYT`	A numeric value representing the mutual information, $I(Y;T)$, between the original labels (Y) and the cluster assignments (T).
`beta`	A numeric vector of the final beta values used in the iterative procedure.
`s`	A numeric vector of bandwidth parameters used for the continuous variables.
`lambda`	A numeric vector of bandwidth parameters used for the categorical variables.
`ixt`	A numeric vector tracking the mutual information values between original observation weights and cluster assignments across iterations.
`iyt`	A numeric vector tracking the mutual information values between original labels and cluster assignments across iterations.
`losses`	A numeric vector tracking the final loss values across iterations.

Author(s)

Efthymios Costa, Ioanna Papatsouma, Angelos Markos

References

Aitchison, J., & Aitken, C. G. G. (1976). Multivariate binary discrimination by the kernel method. *Biometrika*, *63*(3), 413-420.

Costa, E., Papatsouma, I., & Markos, A. (2024). A Deterministic Information Bottleneck Method for Clustering Mixed-Type Data. *arXiv:2407.03389 [stat.ME]*. Retrieved from https://arxiv.org/abs/2407.03389

Silverman, B. W. (1998). Density estimation for statistics and data analysis (1st ed.). Routledge.

Li, Q., & Racine, J. (2003). Nonparametric estimation of distributions with categorical and continuous data. *Journal of Multivariate Analysis*, *86*(2), 266-292.

Examples

```
# Example dataset with categorical, ordinal, and continuous variables
data <- data.frame(
 cat_var = factor(sample(letters[1:3], 100, replace = TRUE)),    # Nominal categorical variable
  ord_var = factor(sample(c("low", "medium", "high"), 100, replace = TRUE),
                   levels = c("low", "medium", "high"),
                   ordered = TRUE),                                 # Ordinal variable
  cont_var1 = rnorm(100),                                          # Continuous variable 1
  cont_var2 = runif(100)                                           # Continuous variable 2
)

# Perform Mixed-Type Fuzzy Clustering
result <- IBmix(X = data, ncl = 3, beta = 2, catcols = 1:2, contcols = 3:4)

# Print clustering results
print(result$Cluster)       # Cluster membership matrix
print(result$InfoXT)       # Mutual information between X and T
print(result$InfoYT)    # Mutual information between Y and T
```

Index