Package ‘IBclust’

July 1, 2025

Type Package

Title Deterministic Information Bottleneck Method for Clustering Mixed-Type Data

Version 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, Rdpack

LinkingTo Rcpp, RcppArmadillo, RcppEigen

RoxygenNote 7.3.2

RdMacros Rdpack

NeedsCompilation yes

Archs x64

Contents

AIBcat 2
AIBcont 4
AIBmix 5
DIBcat 8
DIBcont 10
DIBmix 12
GIBcat 14
GIBcont 17
GIBmix 19
IBcat 22
IBcont 24
IBmix 26

Index **29**

AIBcat *Cluster Categorical Data Using the Agglomerative Information Bottleneck Algorithm*

Description

The AIBcat function implements the Agglomerative Information Bottleneck (AIB) algorithm for hierarchical clustering of datasets containing categorical variables. This method merges clusters so that information retention is maximised at each step to create meaningful clusters, leveraging bandwidth parameters to handle different categorical data types (nominal and ordinal) effectively (Slonim and Tishby 1999).

Usage

```
AIBcat(X, lambda = -1)
```

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).
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.

Details

The AIBcat function applies the Agglomerative Information Bottleneck algorithm to do hierarchical clustering of datasets containing only categorical variables, both nominal and ordinal. The algorithm uses an information-theoretic criterion to merge clusters so that information retention is maximised at each step to create meaningful clusters with maximal 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 (Aitchison and Aitken 1976).

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

where ν is the bandwidth parameter for ordinal variables, accounting for the ordinal relationship between categories (Li and Racine 2003).

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:

merges	A data frame with 2 columns and n rows, showing which observations are merged at each step.
merge_costs	A numeric vector tracking the cost incurred by each merge $I(Z_m;Y) - I(Z_{m-1};Y)$.
partitions	A list containing n sub-lists. Each sub-list includes the cluster partition at each step.
I_Z_Y	A numeric vector including the mutual information $I(Z_m;Y)$ as the number of clusters m increases.
I_X_Y	A numeric value of the mutual information $I(X;Y)$ between observation indices and location.
info_ret	A numeric vector of length n including the fraction of the original information retained after each merge.
dendrogram	A dendrogram visualising the cluster hierarchy. The height is determined by the cost of cluster merges.

Author(s)

Efthymios Costa, Ioanna Papatsouma, Angelos Markos

References

Slonim N, Tishby N (1999). "Agglomerative Information Bottleneck." *Advances in Neural Information Processing Systems*, **12**.

Aitchison J, Aitken CG (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

AIBmix, AIBcont

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 AIBcat with automatic lambda selection
result <- AIBcat(X = X, lambda = -1)

# Print clustering results
plot(result$dendrogram, xlab = "", sub = "")  # Plot dendrogram
```

`AIBcont` *Cluster Continuous Data Using the Agglomerative Information Bottleneck Algorithm*

Description

The `AIBcont` function implements the Agglomerative Information Bottleneck (AIB) algorithm for hierarchical clustering of datasets containing categorical variables. This method merges clusters so that information retention is maximised at each step to create meaningful clusters, leveraging bandwidth parameters to handle different categorical data types (nominal and ordinal) effectively (Slonim and Tishby 1999).

Usage

```
AIBcont(X, s = -1, scale = TRUE)
```

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).
`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`.

Details

The `AIBcat` function applies the Agglomerative Information Bottleneck algorithm to do hierarchical clustering of datasets containing only continuous variables, both nominal and ordinal. The algorithm uses an information-theoretic criterion to merge clusters so that information retention is maximised at each step to create meaningful clusters with maximal information about the original 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:

`merges`	A data frame with 2 columns and n rows, showing which observations are merged at each step.
`merge_costs`	A numeric vector tracking the cost incurred by each merge $I(Z_m;Y) - I(Z_{m-1};Y)$.

partitions	A list containing n sub-lists. Each sub-list includes the cluster partition at each step.
I_Z_Y	A numeric vector including the mutual information $I(Z_m; Y)$ as the number of clusters m increases.
I_X_Y	A numeric value of the mutual information $I(X; Y)$ between observation indices and location.
info_ret	A numeric vector of length n including the fraction of the original information retained after each merge.
dendrogram	A dendrogram visualising the cluster hierarchy. The height is determined by the cost of cluster merges.

Author(s)

Efthymios Costa, Ioanna Papatsouma, Angelos Markos

References

Slonim N, Tishby N (1999). "Agglomerative Information Bottleneck." *Advances in Neural Information Processing Systems*, **12**.

Silverman BW (1998). *Density Estimation for Statistics and Data Analysis (1st Ed.)*. Routledge.

See Also

AIBmix, AIBcat

Examples

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

# Run AIBcont with automatic bandwidth selection
result <- AIBcont(X = X, s = -1, scale = TRUE)

# Print clustering results
plot(result$dendrogram, xlab = "", sub = "")  # Plot dendrogram
```

AIBmix — *Agglomerative Information Bottleneck Clustering for Mixed-Type Data*

Description

The `AIBmix` function implements the Agglomerative Information Bottleneck (AIB) algorithm for hierarchical clustering of datasets containing mixed-type variables, including categorical (nominal and ordinal) and continuous variables. This method merges clusters so that information retention is maximised at each step to create meaningful clusters, leveraging bandwidth parameters to handle different categorical data types (nominal and ordinal) effectively (Slonim and Tishby 1999).

Usage

```
AIBmix(X, catcols, contcols, lambda = -1, s = -1, scale = TRUE)
```

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).

`catcols` A vector indicating the indices of the categorical variables in `X`.

`contcols` A vector indicating the indices of the continuous variables in `X`.

`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`.

Details

The `AIBmix` function produces a hierarchical agglomerative clustering of the data while retaining maximal information about the original variable distributions. The Agglomerative Information Bottleneck algorithm uses an information-theoretic criterion to merge clusters so that information retention is maximised at each step, hence creating meaningful clusters with maximal information about the original distribution. Bandwidth parameters for categorical (nominal, ordinal) and continuous variables are adaptively determined if not provided. This 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*

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

- *Ordinal categorical variables: Li & Racine kernel*

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

Value

A list containing the following elements:

`merges` A data frame with 2 columns and n rows, showing which observations are merged at each step.

merge_costs	A numeric vector tracking the cost incurred by each merge $I(Z_m;Y)-I(Z_{m-1};Y)$.
partitions	A list containing n sub-lists. Each sub-list includes the cluster partition at each step.
I_Z_Y	A numeric vector including the mutual information $I(Z_m;Y)$ as the number of clusters m increases.
I_X_Y	A numeric value of the mutual information $I(X;Y)$ between observation indices and location.
info_ret	A numeric vector of length n including the fraction of the original information retained after each merge.
dendrogram	A dendrogram visualising the cluster hierarchy. The height is determined by the cost of cluster merges.

Author(s)

Efthymios Costa, Ioanna Papatsouma, Angelos Markos

References

Slonim N, Tishby N (1999). "Agglomerative Information Bottleneck." *Advances in Neural Information Processing Systems*, **12**.

Aitchison J, Aitken CG (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.

Silverman BW (1998). *Density Estimation for Statistics and Data Analysis (1st Ed.)*. Routledge.

See Also

AIBcat, AIBcont

Examples

```
# Example dataset with categorical, ordinal, and continuous variables
set.seed(123)
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 Hierarchical Clustering with Agglomerative IB
result <- AIBmix(X = data, catcols = 1:2, contcols = 3:4, lambda = -1, s = -1, scale = TRUE)

# Print clustering results
plot(result$dendrogram, xlab = "", sub = "")  # Plot dendrogram
```

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 et al. 2025).

Usage

```
DIBcat(X, ncl, randinit = NULL, lambda = -1,
       maxiter = 100, nstart = 100,
       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.
`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:

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

where ℓ is the number of categories, and λ controls the smoothness of the Aitchison & Aitken kernel for nominal variables (Aitchison and Aitken 1976).

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

where ν is the bandwidth parameter for ordinal variables, accounting for the ordinal relationship between categories (Li and Racine 2003).

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 (2025). “A Deterministic Information Bottleneck Method for Clustering Mixed-Type Data.” doi:10.48550/arXiv.2407.03389, arXiv:2407.03389, `https://arxiv.org/abs/2407.03389`.

Aitchison J, Aitken CG (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 et al. 2025).

Usage

```
DIBcont(X, ncl, randinit = NULL, s = -1, scale = TRUE,
        maxiter = 100, nstart = 100, 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`.
`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 (2025). “A Deterministic Information Bottleneck Method for Clustering Mixed-Type Data.” doi:10.48550/arXiv.2407.03389, arXiv:2407.03389, `https://arxiv.org/abs/2407.03389`.

Silverman BW (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 et al. 2025).

Usage

```
DIBmix(X, ncl, catcols, contcols, randinit = NULL,
       lambda = -1, s = -1, scale = TRUE,
       maxiter = 100, nstart = 100,
       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.
`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*

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

- *Ordinal categorical variables: Li & Racine kernel*

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

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 containing 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

Costa E, Papatsouma I, Markos A (2025). "A Deterministic Information Bottleneck Method for Clustering Mixed-Type Data." doi:10.48550/arXiv.2407.03389, arXiv:2407.03389, https://arxiv.org/abs/2407.03389.

Aitchison J, Aitken CG (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.

Silverman BW (1998). *Density Estimation for Statistics and Data Analysis (1st Ed.)*. Routledge.

See Also

DIBcont, DIBcat

Examples

```
# Example dataset with categorical, ordinal, and continuous variables
set.seed(123)
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
```

GIBcat   *Cluster Categorical Data Using the Generalised Information Bottleneck Algorithm*

Description

The GIBcat function implements the Generalised Information Bottleneck (GIB) algorithm for fuzzy clustering of 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 (Strouse and Schwab 2019).

Usage

```
GIBcat(X, ncl, beta, alpha, randinit = NULL, lambda = -1,
       maxiter = 100, nstart = 100, 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).
`beta`	Regularisation strength.
`alpha`	Strength of relative entropy term.
`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.
`verbose`	Logical. Default to `FALSE` to suppress progress messages. Change to `TRUE` to print.

Details

The `GIBcat` function applies the Generalised Information Bottleneck algorithm to do fuzzy clustering of 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. Set $\alpha = 1$ and $\alpha = 0$ to recover the Information Bottleneck and its Deterministic variant, respectively. If $\alpha = 0$, the algorithm ignores the value of the regularisation parameter β.

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 (Aitchison and Aitken 1976).

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

where ν is the bandwidth parameter for ordinal variables, accounting for the ordinal relationship between categories (Li and Racine 2003).

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` A cluster membership matrix.
- `Entropy` A numeric value representing the entropy of the cluster assignment, $H(T)$.
- `RelEntropy` A numeric value representing the relative entropy of cluster assignment, given the observation weights $H(X \mid T)$.
- `MutualInfo` A numeric value representing the mutual information, $I(Y;T)$, between the original labels (Y) and the cluster assignments (T).
- `beta` A numeric value of the regularisation strength beta used.
- `alpha` A numeric value of the strength of relative entropy used.
- `lambda` A numeric vector of bandwidth parameters for categorical variables, controlling how categories are weighted in the clustering.
- `ht` A numeric vector tracking the entropy value of the cluster assignments across iterations.
- `hy_t` A numeric vector tracking the relative entropy values between the cluster assignments and observations weights 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

Strouse DJ, Schwab DJ (2017). "The Deterministic Information Bottleneck." *Neural Computation*, **29**(6), 1611–1630.

Aitchison J, Aitken CG (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

`GIBmix`, `GIBcont`

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 GIBcat with automatic lambda selection and multiple initializations
result <- GIBcat(X = X, ncl = 2, beta = 25, alpha = 0.75, lambda = -1, nstart = 20)

# Print clustering results
print(result$Cluster)       # Cluster membership matrix
print(result$Entropy)       # Entropy of final clustering
print(result$RelEntropy)    # Relative entropy of final clustering
print(result$MutualInfo)    # Mutual information between Y and T
```

GIBcont *Cluster Continuous Data Using the Generalised Information Bottleneck Algorithm*

Description

The `GIBcont` function implements the Generalised Information Bottleneck (GIB) algorithm for fuzzy clustering of continuous data. This method optimizes an information-theoretic objective to preserve relevant information while forming concise and interpretable cluster representations (Strouse and Schwab 2019).

Usage

```
GIBcont(X, ncl, beta, alpha, randinit = NULL, s = -1, scale = TRUE,
        maxiter = 100, nstart = 100,
        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.
`beta`	Regularisation strength.
`alpha`	Strength of relative entropy term.
`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`.
`verbose`	Logical. Default to `FALSE` to suppress progress messages. Change to `TRUE` to print.

Details

The `GIBcont` function applies the Generalised Information Bottleneck algorithm to do fuzzy clustering of 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. Set $\alpha = 1$ and $\alpha = 0$ to recover the Information Bottleneck and its Deterministic variant, respectively. If $\alpha = 0$, the algorithm ignores the value of the regularisation parameter β.

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`	A cluster membership matrix.
`Entropy`	A numeric value representing the entropy of the cluster assignment, $H(T)$.
`RelEntropy`	A numeric value representing the relative entropy of cluster assignment, given the observation weights $H(X \mid T)$.
`MutualInfo`	A numeric value representing the mutual information, $I(Y;T)$, between the original labels (Y) and the cluster assignments (T).
`beta`	A numeric value of the regularisation strength beta used.
`alpha`	A numeric value of the strength of relative entropy used.
`s`	A numeric vector of bandwidth parameters used for the continuous variables.
`ht`	A numeric vector tracking the entropy value of the cluster assignments across iterations.
`hy_t`	A numeric vector tracking the relative entropy values between the cluster assignments and observations weights 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

Strouse DJ, Schwab DJ (2017). "The Deterministic Information Bottleneck." *Neural Computation*, **29**(6), 1611–1630.

Silverman BW (1998). *Density Estimation for Statistics and Data Analysis (1st Ed.)*. Routledge.

See Also

`GIBmix`, `GIBcat`

Examples

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

# Run GIBcont with automatic bandwidth selection and multiple initializations
result <- GIBcont(X = X, ncl = 2, beta = 50, alpha = 0.75, s = -1, nstart = 20)

# Print clustering results
print(result$Cluster)      # Cluster membership matrix
print(result$Entropy)      # Entropy of final clustering
print(result$RelEntropy)   # Relative entropy of final clustering
print(result$MutualInfo)   # Mutual information between Y and T
```

`GIBmix` *Generalised Information Bottleneck Clustering for Mixed-Type Data*

Description

The `GIBmix` function implements the Generalised Information Bottleneck (GIB) 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 (Strouse and Schwab 2017).

Usage

```
GIBmix(X, ncl, beta, alpha, catcols, contcols, randinit = NULL,
       lambda = -1, s = -1, scale = TRUE,
       maxiter = 100, nstart = 100,
       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.
`alpha`	Strength of relative entropy term.
`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.

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

Details

The `GIBmix` function produces a fuzzy clustering of the data while retaining maximal information about the original variable distributions. The Generalised 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. Set $\alpha = 1$ and $\alpha = 0$ to recover the Information Bottleneck and its Deterministic variant, respectively. If $\alpha = 0$, the algorithm ignores the value of the regularisation parameter β.

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*

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

- *Ordinal categorical variables: Li & Racine kernel*

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

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 containing the following elements:

`Cluster` A cluster membership matrix.

`Entropy` A numeric value representing the entropy of the cluster assignment, $H(T)$.

`RelEntropy` A numeric value representing the relative entropy of cluster assignment, given the observation weights $H(X \mid T)$.

MutualInfo	A numeric value representing the mutual information, $I(Y;T)$, between the original labels (Y) and the cluster assignments (T).
beta	A numeric value of the regularisation strength beta used.
alpha	A numeric value of the strength of relative entropy used.
s	A numeric vector of bandwidth parameters used for the continuous variables.
lambda	A numeric vector of bandwidth parameters used for the categorical variables.
ht	A numeric vector tracking the entropy value of the cluster assignments across iterations.
hy_t	A numeric vector tracking the relative entropy values between the cluster assignments and observations weights 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

Strouse DJ, Schwab DJ (2017). "The Deterministic Information Bottleneck." *Neural Computation*, **29**(6), 1611–1630.

Aitchison J, Aitken CG (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.

Silverman BW (1998). *Density Estimation for Statistics and Data Analysis (1st Ed.)*. Routledge.

See Also

GIBcat, GIBcont

Examples

```
# Example dataset with categorical, ordinal, and continuous variables
set.seed(123)
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 with Generalised IB
result <- GIBmix(X = data, ncl = 3, beta = 2, alpha = 0.5, catcols = 1:2, contcols = 3:4, nstart = 20)

# Print clustering results
print(result$Cluster)       # Cluster membership matrix
print(result$Entropy)       # Entropy of final clustering
print(result$RelEntropy)    # Relative entropy of final clustering
print(result$MutualInfo)    # Mutual information between Y and T
```

IBcat *Cluster Categorical Data Using the Information Bottleneck Algorithm*

Description

The `IBcat` function implements the Information Bottleneck (IB) algorithm for fuzzy clustering of 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 (Strouse and Schwab 2019).

Usage

```
IBcat(X, ncl, beta, randinit = NULL, lambda = -1,
      maxiter = 100, nstart = 100, 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).
`beta`	Regularisation strength.
`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.
`verbose`	Logical. Default to `FALSE` to suppress progress messages. Change to `TRUE` to print.

Details

The `IBcat` function applies the Information Bottleneck algorithm to do fuzzy clustering of 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 (Aitchison and Aitken 1976).

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

where ν is the bandwidth parameter for ordinal variables, accounting for the ordinal relationship between categories (Li and Racine 2003).

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`	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 value of the regularisation strength beta used.
`lambda`	A numeric vector of bandwidth parameters for categorical variables, controlling how categories are weighted in the clustering.
`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

Strouse DJ, Schwab DJ (2019). “The information bottleneck and geometric clustering.” *Neural Computation*, **31**(3), 596–612.

Aitchison J, Aitken CG (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

`IBmix`, `IBcont`

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 IBcat with automatic lambda selection and multiple initializations
result <- IBcat(X = X, ncl = 3, beta = 15, lambda = -1, nstart = 20)

# 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
```

`IBcont` *Cluster Continuous Data Using the Information Bottleneck Algorithm*

Description

The `IBcont` function implements the Information Bottleneck (IB) algorithm for fuzzy clustering of continuous data. This method optimizes an information-theoretic objective to preserve relevant information while forming concise and interpretable cluster representations (Strouse and Schwab 2019).

Usage

```
IBcont(X, ncl, beta, randinit = NULL, s = -1, scale = TRUE,
       maxiter = 100, nstart = 100, 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.
`beta`	Regularisation strength.
`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`.
`verbose`	Logical. Default to `FALSE` to suppress progress messages. Change to `TRUE` to print.

Details

The `IBcont` function applies the Information Bottleneck algorithm to do fuzzy clustering of 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`	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 value of the regularisation strength beta used.
`s`	A numeric vector of bandwidth parameters used for the continuous 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

Strouse DJ, Schwab DJ (2019). "The information bottleneck and geometric clustering." *Neural Computation*, **31**(3), 596–612.

Silverman BW (1998). *Density Estimation for Statistics and Data Analysis (1st Ed.)*. Routledge.

See Also

`IBmix`, `IBcat`

Examples

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

# Run IBcont with automatic bandwidth selection and multiple initializations
result <- IBcont(X = X, ncl = 3, beta = 50, s = -1, nstart = 20)

# 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
```

`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 (Strouse and Schwab 2019).

Usage

```
IBmix(X, ncl, beta, catcols, contcols, randinit = NULL,
       lambda = -1, s = -1, scale = TRUE,
       maxiter = 100, nstart = 100,
       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.
`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 containing 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 value of the regularisation strength beta used.
`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

Strouse DJ, Schwab DJ (2019). "The information bottleneck and geometric clustering." *Neural Computation*, **31**(3), 596–612.

Aitchison J, Aitken CG (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.

Silverman BW (1998). *Density Estimation for Statistics and Data Analysis (1st Ed.)*. Routledge.

See Also

DIBcont, DIBcat

Examples

```
# Example dataset with categorical, ordinal, and continuous variables
set.seed(123)
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, nstart = 20)

# 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

∗ **clustering**
- AIBcat, 2
- AIBcont, 4
- AIBmix, 5
- DIBcat, 8
- DIBcont, 10
- DIBmix, 12
- GIBcat, 14
- GIBcont, 17
- GIBmix, 19
- IBcat, 22
- IBcont, 24
- IBmix, 26

AIBcat, 2, *5*, *7*
AIBcont, *3*, 4, *7*
AIBmix, *3*, *5*, 5

DIBcat, 8, *11*, *14*, *28*
DIBcont, *9*, 10, *14*, *28*
DIBmix, *9*, *11*, 12

GIBcat, 14, *18*, *21*
GIBcont, *16*, 17, *21*
GIBmix, *16*, *18*, 19

IBcat, 22, *25*
IBcont, *23*, 24
IBmix, *23*, *25*, 26