# Package 'conclust'

October 12, 2022

**Type** Package

**Title** Pairwise Constraints Clustering

**Version** 1.1

**Date** 2016-08-15

**Author** Tran Khanh Hiep, Nguyen Minh Duc

**Maintainer** Tran Khanh Hiep <hieptkse03059@fpt.edu.vn>

**Description** There are 4 main functions in this package: ckmeans(), lcvqe(), mpckm() and ccls(). They take an unlabeled dataset and two lists of must-link and cannot-link constraints as input and produce a clustering as output.

**License** GPL-3

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2016-08-15 13:16:21

## R topics documented:

---

conclust-package | *Pairwise Constraints Clustering*

---

### Description

There are 4 main functions in this package: ckmeans(), lcvqe(), mpckm() and ccls(). They take an unlabeled dataset and two lists of must-link and cannot-link constraints as input and produce a clustering as output.

## Details

The DESCRIPTION file:

| | |
|---|---|
| Package: | conclust |
| Type: | Package |
| Title: | Pairwise Constraints Clustering |
| Version: | 1.1 |
| Date: | 2016-08-15 |
| Author: | Tran Khanh Hiep, Nguyen Minh Duc |
| Maintainer: | Tran Khanh Hiep <hieptkse03059@fpt.edu.vn> |
| Description: | There are 4 main functions in this package: ckmeans(), lcvqe(), mpckm() and ccls(). They take an unlabeled da |
| License: | GPL-3 |

Index of help topics:

```
ccls                   Pairwise Constrained Clustering by Local Search
ckmeans                COP K-means algorithm
conclust-package       Pairwise Constraints Clustering
lcvqe                  LCVQE algorithm
mpckm                  MPC K-means algorithm
```

There are 4 main functions in this package: ckmeans(), lcvqe(), mpckm() and ccls(). They take an unlabeled dataset and two lists of must-link and cannot-link constraints as input and produce a clustering as output.

## Author(s)

Tran Khanh Hiep, Nguyen Minh Duc

Maintainer: Tran Khanh Hiep <hieptkse03059@fpt.edu.vn>

## References

Wagstaff, Cardie, Rogers, Schrodl (2001), Constrained K-means Clustering with Background Knowledge Bilenko, Basu, Mooney (2004), Integrating Constraints and Metric Learning in Semi-Supervised Clustering Dan Pelleg, Dorit Baras (2007), K-means with large and noisy constraint sets

## See Also

Wagstaff, Cardie, Rogers, Schrodl (2001), Constrained K-means Clustering with Background Knowledge Bilenko, Basu, Mooney (2004), Integrating Constraints and Metric Learning in Semi-Supervised Clustering Dan Pelleg, Dorit Baras (2007), K-means with large and noisy constraint sets

## Examples

```
data = matrix(c(0, 1, 1, 0, 0, 0, 1, 1), nrow = 4)
mustLink = matrix(c(1, 2), nrow = 1)
cantLink = matrix(c(1, 4), nrow = 1)
k = 2
```

```
pred = ckmeans(data, k, mustLink, cantLink)
pred
pred = mpckm(data, k, mustLink, cantLink)
pred
pred = lcvqe(data, k, mustLink, cantLink)
pred
pred = ccls(data, k, mustLink, cantLink)
pred
```

---

ccls                        *Pairwise Constrained Clustering by Local Search*

---

### Description

This function takes an unlabeled dataset and two lists of must-link and cannot-link constraints as input and produce a clustering as output.

### Usage

```
ccls(data, k = -1, mustLink, cantLink, maxIter = 1, tabuIter = 100, tabuLength = 20)
```

### Arguments

| | |
|---|---|
| data | The unlabeled dataset. |
| k | Number of clusters. |
| mustLink | A list of must-link constraints |
| cantLink | A list of cannot-link constraints |
| maxIter | Number of iteration |
| tabuIter | Number of iteration in Tabu search |
| tabuLength | The number of elements in the Tabu list |

### Details

This algorithm minimizes the clustering cost function using Tabu search.

### Value

A vector that represents the labels (clusters) of the data points

### Note

This is the first algorithm for pairwise constrained clustering by local search.

### Author(s)

Tran Khanh Hiep Nguyen Minh Duc

## References

Tran Khanh Hiep, Nguyen Minh Duc, Bui Quoc Trung (2016), Pairwise Constrained Clustering by Local Search.

## See Also

Tran Khanh Hiep, Nguyen Minh Duc, Bui Quoc Trung (2016), Pairwise Constrained Clustering by Local Search.

## Examples

```
data = matrix(c(0, 1, 1, 0, 0, 0, 1, 1), nrow = 4)
mustLink = matrix(c(1, 2), nrow = 1)
cantLink = matrix(c(1, 4), nrow = 1)
k = 2
pred = ckmeans(data, k, mustLink, cantLink)
pred
```

---

ckmeans                                 *COP K-means algorithm*

---

## Description

This function takes an unlabeled dataset and two lists of must-link and cannot-link constraints as input and produce a clustering as output.

## Usage

```
ckmeans(data, k, mustLink, cantLink, maxIter = 100)
```

## Arguments

| | |
|-----------|----------------------------------------|
| data      | The unlabeled dataset.                 |
| k         | Number of clusters.                    |
| mustLink  | A list of must-link constraints        |
| cantLink  | A list of cannot-link constraints      |
| maxIter   | Number of iteration                    |

## Details

This algorithm produces a clustering that satisfies all given constraints.

## Value

A vector that represents the labels (clusters) of the data points

## Note

The constraints should be consistent in order for the algorithm to work.

## Author(s)

Tran Khanh Hiep Nguyen Minh Duc

## References

Wagstaff, Cardie, Rogers, Schrodl (2001), Constrained K-means Clustering with Background Knowledge

## See Also

Wagstaff, Cardie, Rogers, Schrodl (2001), Constrained K-means Clustering with Background Knowledge

## Examples

```
data = matrix(c(0, 1, 1, 0, 0, 0, 1, 1), nrow = 4)
mustLink = matrix(c(1, 2), nrow = 1)
cantLink = matrix(c(1, 4), nrow = 1)
k = 2
pred = ckmeans(data, k, mustLink, cantLink)
pred
```

---

lcvqe                         *LCVQE algorithm*

---

## Description

This function takes an unlabeled dataset and two lists of must-link and cannot-link constraints as input and produce a clustering as output.

## Usage

```
lcvqe(data, k, mustLink, cantLink, maxIter = 10)
```

## Arguments

| | |
|---|---|
| data | The unlabeled dataset. |
| k | Number of clusters. |
| mustLink | A list of must-link constraints |
| cantLink | A list of cannot-link constraints |
| maxIter | Number of iteration |

## Details

This algorithm finds a clustering that satisfies as many constraints as possible

## Value

A vector that represents the labels (clusters) of the data points

## Note

This algorithm can handle noisy constraints.

## Author(s)

Tran Khanh Hiep Nguyen Minh Duc

## References

Dan Pelleg, Dorit Baras (2007), K-means with large and noisy constraint sets

## See Also

Dan Pelleg, Dorit Baras (2007), K-means with large and noisy constraint sets

## Examples

```
data = matrix(c(0, 1, 1, 0, 0, 0, 1, 1), nrow = 4)
mustLink = matrix(c(1, 2), nrow = 1)
cantLink = matrix(c(1, 4), nrow = 1)
k = 2
pred = lcvqe(data, k, mustLink, cantLink)
pred
```

---

mpckm  *MPC K-means algorithm*

---

## Description

This function takes an unlabeled dataset and two lists of must-link and cannot-link constraints as input and produce a clustering as output.

## Usage

```
mpckm(data, k, mustLink, cantLink, maxIter = 10)
```

## Arguments

| | |
|---|---|
| `data` | The unlabeled dataset. |
| `k` | Number of clusters. |
| `mustLink` | A list of must-link constraints |
| `cantLink` | A list of cannot-link constraints |
| `maxIter` | Number of iteration |

## Details

This algorithm finds a clustering that satisfies as many constraints as possible

## Value

A vector that represents the labels (clusters) of the data points

## Note

This is one of the best algorithm for clustering with constraints.

## Author(s)

Tran Khanh Hiep Nguyen Minh Duc

## References

Bilenko, Basu, Mooney (2004), Integrating Constraints and Metric Learning in Semi-Supervised Clustering

## See Also

Bilenko, Basu, Mooney (2004), Integrating Constraints and Metric Learning in Semi-Supervised Clustering

## Examples

```
data = matrix(c(0, 1, 1, 0, 0, 0, 1, 1), nrow = 4)
mustLink = matrix(c(1, 2), nrow = 1)
cantLink = matrix(c(1, 4), nrow = 1)
k = 2
pred = mpckm(data, k, mustLink, cantLink)
pred
```

# Index