python-cluster Documentation

Release 1.3.0

Michel Albert

December 10, 2015

Contents

CHAPTER 1

Index

1.1 Changelog

1.1.1 Release 1.3.0

- Performance improvments for hierarchical clustering (at the cost of memory)
- Cluster instances are now iterable.It will iterate over each element, resulting in a flat list of items.
- New option to specify a progress method. This method will be called on each iteration for hierarchical clusters. It gives users a way to present to progress on screen.
- The library now also has a `__version__` member.

CHAPTER 2

Introduction

Implementation of cluster algorithms in pure Python.

As this is exacuted in the Python runtime, the code runs slower than similar implementations in compiled languages. You gain however to run this on pretty much any Python object. The different clustering methods have different prerequisites however which are mentioned in the different implementations.

CHAPTER 3

Example for K-Means Clustering

```
from cluster import KMeansClustering
data = [
    (8, 2),
    (7, 3),
    (2, 6),
    (3, 5),
    (3, 6),
    (1, 5),
    (8, 1),
    (3, 4),
    (8, 3),
    (9, 2),
    (2, 5),
    (9, 3)
]
cl = KMeansClustering(data)
cl.getclusters(2)
```

The above code would give the following result:

```
[
    [(8, 2), (8, 1), (8, 3), (7, 3), (9, 2), (9, 3)],
    [(3, 5), (1, 5), (3, 4), (2, 6), (2, 5), (3, 6)]
]
```

CHAPTER 4

Example for Hierarchical Clustering

```
from cluster import HierarchicalClustering
data = [791, 956, 676, 124, 564, 84, 24, 365, 594, 940, 398,
        971, 131, 365, 542, 336, 518, 835, 134, 391]
cl = HierarchicalClustering(data)
cl.getlevel(40)
```

The above code would give the following result:

```
[
    [24],
    [84, 124, 131, 134],
    [336, 365, 365, 391, 398],
    [676],
    [594, 518, 542, 564],
    [940, 956, 971],
    [791],
    [835],
]
```

Using `getlevel()` returns clusters where the distance between each cluster is no less than *level*.

Note: Due to a bug in earlier releases, the elements of the input data *must be* sortable!

CHAPTER 5

API

CHAPTER 6

Indices and tables

- genindex
- modindex
- search