Hierarchical Mode Association Clustering Package

Jia Li
Email: jiali@psu.edu
The Pennsylvania State University

April 12, 2007

1 Introduction

This is a C package for the Hierarchical Mode Association Clustering (HMAC) algorithms described in [1]. The package is maintained at the Web site: `http://www.stat.psu.edu/~jiali/hmac`.

Mode association clustering (MAC) can be conducted either hierarchically or at one level. MAC is similar to mixture model based clustering in the sense of characterizing clusters by smooth densities. However, MAC requires no model fitting and uses a nonparametric kernel density estimation. The density of a cluster is not restricted to be parametric, for instance, Gaussian, but ensures uni-modality. The algorithm seems to combine the complementary merits of bottom-up clustering such as linkage and top-down clustering such as mixture modeling and k-means. It also tends to be robust against non-Gaussian shaped clusters. For details on the algorithm, we refer to [1]. We describe below the content in the package, how to install it, and the syntax for running the codes.

Please email comments to `jiali@stat.psu.edu` for usage of the codes, and refinement of the codes, algorithms, or documentation.

2 What Is Included?

The package includes:

1. C source files
2. Makefile
3. Readme and doc.pdf to explain the package
4. Some example data files which all have names *.dat

The algorithms of HMAC are implemented in three programs: `mtree, mtreeridge, mtreesep`. And linkage clustering is performed by `linkage`.

1. `mtree` is the basic version which performs the hierarchical mode association clustering (HMAC), and outputs the clustering results at each level of the hierarchy as well as the dendrogram created.

2. `mtreeridge` performs the same clustering procedure as in `mtree` first, then computes the ridgeline and the density values along the ridgeline for either a given pair of clusters at a given level, or for all the pairs of clusters at all the levels.

3. `mtreesep` performs the same clustering procedure as in `mtree` first, then conducts the separability and coverage rate based merging at either a given level, or all the levels of the dendrogram.

4. `linkage` performs agglomerative linkage clustering using different linkage schemes, e.g., single, complete, average, Ward's linkage.

3 How to Install?

To install the package, follow these steps:

1. Download the package, `hmac.tar`, from:
   `http://www.stat.psu.edu/~jiali/hmac`

2. Untar the package under command shell:
   `> tar -xvf hmac.tar`

   This will create a directory `hmac`

3. Go into the directory `hmac` and type this under command shell:
   `> make`

   Four executables should be created: `mtree, mtreeridge, mtreesep, linkage`.

4 Program Usages

The input data file that contains the instances to be clustered is of the same format for all the programs. An example data file is given by `exp1.dat`. The file should be in ascii format containing floating point numbers. Every row corresponds to one instance. Every column corresponds to one dimension. The values of different dimensions are separated by space. Readers are referred to [1] for a thorough understanding of the algorithms.

4.1 HMAC Algorithms

Syntax for `mtree, mtreeridge, mtreesep`

The order of the input arguments can be arbitrary.

1. `mtree -i` [input file] `-o` [output file] `-l` [input bandwidth file(optional)] `-p` [input weight file (optional)] `-d` [data dimension] `-n` [data size (optional)] `-f` [fraction for bandwidth step size] `-m` [factor for maximum bandwidth step size] `-h` [flag for hierarchical clustering if set] `-s` [flag for showing the modal EM path if set]

2. `mtreeridge -i` [input file] `-o` [output file] `-l` [input bandwidth file(optional)] `-p` [input weight file (optional)] `-d` [data dimension] `-n` [data size (optional)] `-f` [fraction for bandwidth step size] `-m` [factor for maximum bandwidth step size] `-h` [flag for hierarchical clustering if set] `-s` [flag for showing the modal EM path if set] `-w` [level to compute ridgeline] `-1` [label for the first cluster of the ridgeline] `-2` [label for the second cluster of the ridgeline]

3. `mtreesep -i` [input file] `-o` [output file] `-l` [input bandwidth file(optional)] `-p` [input weight file (optional)] `-d` [data dimension] `-n` [data size (optional)] `-f` [fraction for bandwidth step size] `-m` [factor for maximum bandwidth step size] `-h` [flag for hierarchical clustering if set] `-s` [flag for showing the modal EM path if set] `-w` [level to apply merging] `-v` [threshold for separability] `-c` [threshold for coverage rate]

Command line inputs

The common command input options for the three programs are explained below.

1. `-i` [input data file]
   This argument is mandatory. It is the file name of the data set to be clustered. There is no default value set for this argument.

2. `-o` [output file]
   This argument is mandatory. It is the file name of output. There is no default value set for this argument.

   (a) `mtree`: it contains the clustering results at all the levels of the dendrogram created, the bandwidths used, and the structure of the dendrogram.

   (b) `mtreeridge`: it contains the points on each ridgeline computed and the density values at these points.

   (c) `mtreesep`: it contains the clustering results obtained at one or multiple levels of the dendrogram after applying the separability and coverage rate based merging procedure.

3. `-l` [input file of kernel bandwidths]
   This argument is optional. If not set, the sequence of bandwidths used by HMAC is automatically generated. If given, it should be the file name of an ascii data file which contains in every row a single floating point value corresponding to a bandwidth. The bandwidths should be in increasing order. The first row should be an integer specifying the number of bandwidths. An example file is `bdw.dat` in the directory. Suppose 10 bandwidths are specified, this file should contain 11 lines where the first line should be 10 and the other lines each contain a bandwidth value.

4. `-p` [input file of instance weights]
   This argument is optional. If not set, the weights on the data points are equal. If set, this argument should be the file name of an ascii data file which contains in each row a single floating point value indicating the weight assigned to the corresponding data point. If the data size is n, this file should have n lines, the kth of which corresponds to the kth data point.

   When the data size is big, HMAC can be slow due to the nonparametric density estimation. A speed up approach is to perform a pre-clustering by, for instance, k-means or k-center on the data. Then represent each cluster by its centroid and assign the cluster a weight proportional to the number of points belonging to this cluster. Use the cluster centroids as data to be clustered and specify the weights in a data file and input it to the program using the `-p` option. As the number of clusters in the pre-clustering is much smaller than the data size, computation in HMAC is much reduced. On the other hand, this number should be much larger than the targeted number of clusters so that the final result of HMAC is nearly not affected. Details on the fast HMAC algorithm is referred to [1].

5. `-d` [data dimension]
   This argument is optional. If not set, the default value is 2. If the dimension of the data is not 2, this argument has to be set to the appropriate integer.

6. `-n` [number of samples]
   This argument is optional. If not set, the program automatically finds the number of samples according to the number of lines in the input data file.

7. `-f` [fraction factor]
   This argument is optional. If not set, the default is 0.1. This argument controls the step size of the bandwidth sequence. The smaller the value, the finer the bandwidth sequence.

8. `-m` [multiple factor]
   This argument is optional. If not set, the default is 2.0. This argument controls the maximum bandwidth in the sequence.

9. `-h` [flag for hierarchical clustering]
   This argument is optional. If not set, the program performs MAC at all the bandwidths individually. A nested hierarchy is not enforced. If set, nested hierarchy is enforced and a dendrogram is produced. For the HMAC algorithm described in [1], this flag should be set.

10. `-s` [flag for showing modal EM paths]
    This argument is optional. If not set, the program does not output the modal EM paths to `stdout`. If set, it does.

Command input options special to `mtreeridge`:

1. `-w` [dendrogram level]
   This argument is optional. If set to a positive integer, it specifies the level at which ridgelines between clusters will be calculated.

2. `-1` [cluster label]
   This argument is optional. If set to a non-negative integer, it specifies the label of one cluster among a pair for which the ridgeline will be calculated.

3. `-2` [cluster label]
   This argument is optional. If set to a non-negative integer, it specifies the label of the other cluster among a pair for which the ridgeline will be calculated.

If any of the above three values is not set, the program calculates ridgelines between all the pairs of clusters at all the levels of the dendrogram, which can be computationally costly. If the three values are set positive or non-negative, a ridgeline will be computed only for the clusters with the specified labels at the specified level. If clusters with such labels or the level specified do not exist, no ridgeline will be computed.

Command input options special to `mtreesep`:

1. `-w` [dendrogram level]
   This argument is optional. If set to a positive integer, it specifies the level at which separability and coverage rate based merging will be applied only to this level. If not set, merging will be applied to all the levels.

2. `-v` [threshold for separability]
   This argument is optional. The default value is 0.5. It can be set to a floating point number between 0 and 1. A pair of clusters generated by modal clustering with separability smaller than this threshold will be merged. If no merging is preferred regardless of this threshold, set this value to 0.

3. `-c` [threshold for coverage rate]
   This argument is optional. The default value is 0.95. It can be set to a floating point number between 0 and 1. If this value is set to, for instance, 95%, then 5% data are allowed to be outliers and coverage rate based merging is performed. If no merging is preferred regardless of this threshold, set this value to 1.

Note that a two-step merging is performed, one based on separability between pairs of cluster, the other on coverage rate, allowing outlier instances. Either of the two merging steps can be bypassed by setting the corresponding threshold to an invalid value, that is, 0 for the separability threshold and 1 for the coverage rate threshold.

Example usages

1. HMAC enforcing nested dendrogram
   ```
   > mtree -i exp1.dat -o out.dat -d 2 -h
   ```

2. Multi-level modal clustering done individually for each level. The clustering result is not ensured to be nested.
   ```
   > mtree -i exp1.dat -o out.dat -d 2
   ```

3. Instead of automatically generating the sequence of bandwidths, use bandwidths input from a file.
   ```
   mtree -i exp1.dat -o out.dat -d 2 -h -l bdw.dat
   ```

4. If weights on the points are unequal, an input file is expected to specify the weights for each point.
   ```
   mtree -i exp1.dat -o out.dat -d 2 -h -p prior.dat
   ```
   The example file containing weights is `prior.dat`. The number of lines in the file equals the data size. Line k contains the intended weight for the kth data point in floating point format.

5. HMAC with ridgelines between all pairs of clusters at all the levels of the dendrogram.
   ```
   mtreeridge -i exp1.dat -o out.dat -d 2 -h
   ```

6. HMAC with ridgeline at a given level between two given clusters.
   ```
   mtreeridge -i exp1.dat -o out.dat -d 2 -h -w 2 -1 0 -2 1
   ```

7. HMAC with separability and coverage rate based merging at all the levels.
   ```
   mtreesep -i exp1.dat -o out.dat -d 2 -h
   ```

8. HMAC with separability and coverage rate based merging at a given level specified by the `-w` option.
   ```
   mtreesep -i exp1.dat -o out.dat -d 2 -h -w 2
   ```

4.2 Linkage Clustering

Syntax for `linkage`

`linkage -i` [input data file] `-o` [output clustering result file] `-t` [output dendrogram file] `-f` [flag for type of linkage] `-d` [data dimension] `-n` [number of instances (optional)]

Command line inputs

1. `-i` [input data file]
   This argument is mandatory. It specifies the file name of the input data to be clustered. Format is the same as that for `mtree`.
2. `-o` [output clustering result file]
   This argument is mandatory. It outputs the clustering results at all the levels of the dendrogram generated by linkage clustering.
3. `-t` [output dendrogram structure file]
   This argument is optional. If given, it is an output file name. The dendrogram generated by linkage clustering is recorded in this file.
4. `-d` [data dimension]
   This argument is optional. The default is 2. If data dimension is not 2, a proper integer should be specified.
5. `-n` [number of samples]
   This argument is optional. If not set, the program automatically finds the number of samples according to the number of lines in the input data file.
6. `-f` [flag for type of linkage clustering]
   This argument is optional. The default is 0, that is, single linkage. Valid values for the flag option:
   (a) 0: single linkage
   (b) 1: complete linkage
   (c) 2: average linkage, unweighted
   (d) 3: average linkage, weighted
   (e) 4: centroid clustering, unweighted
   (f) 5: centroid clustering, weighted
   (g) 6: Ward's clustering

Example usages

1. Average linkage clustering
```
> linkage -i exp1.dat -o cls.dat -t tree.dat -f 3 -d 2
```

References

[1] J. Li, S. Ray, B. G. Lindsay, "A nonparametric statistical approach to clustering via mode identification," *Journal of Machine Learning Research*, 8(8):1687-1723, 2007.