LMCLUS Documentation

Release 1.1.0

Art Wild

Aug 14, 2018

Contents

Many clustering algorithms are based on the concept that a cluster has a a single center point. Clusters could be considered as groups of points compact around a linear manifold. A linear manifold of dimension 0 is a point. So clustering around a center point is a special case of linear manifold clustering.

Linear manifold clustering algorithm identifies subsets of the data which are embedded in arbitrary oriented lower dimensional linear manifolds, not nessesaraly zero dimensional. Minimal subsets of points are repeatedly sampled to construct trial a linear manifold and isolete points around it based of the proximity of points to the found manifold. Using top-down aproach, the linear manifold clustering alogorithm iterativly partitions dataset and discovers clusters embedded into low-dimensioanl linear subspaces[1].

LMCLUS.jl is a Julia package for linear manifold clustering.

Contents:

[1] Haralick, R. & Harpaz, R. "Linear manifold clustering in high dimensional spaces by stochastic search", Pattern recognition, Elsevier, 2007, 40, 2672-2684, DOI:10.1016/j.patcog.2007.01.020

CHAPTER 1

Linear Manifold Clustering

Linear manifold clustering algorithm (LMCLUS) discovers clusters which are described by a following model:

$$x = \mu^{N\times 1} + B^{N\times K}\phi^{K\times 1} + \bar{B}^{N\times N-K}\epsilon^{N-K\times 1}$$

where N is a dimension of the dataset, K is dimension of the manifold, $\mu \in \mathbb{R}^N$ is a linear manifold translation vector, B is a matrix whose columns are orthonormal vectors that span $\mathbb{R}^K$, $\bar{B}$ is a matrix whose columns span subspace orthogonal to spanned by columns of B, ϕ is a zero-mean random vector whose entries are i.i.d. from a support of linear manifold, ϵ is a zero-mean random vector with small variance independent of ϕ.

1.1 Clustering

This package implements the *LMCLUS* algorithm in the `lmclus` function:

`lmclus`(*X*, *p*)
Performs linear manifold clustering over the given dataset.

Parameters

- **`X`** – The given sample matrix. Each column of `X` is a sample.
- **`p`** – The clustering parameters as instance of *LMCLUSParameters*.

This function returns an `LMCLUSResult` instance.

1.2 Results

Let `M` be an instance of `Manifold`, `n` be the number of observations, and `d` be the dimension of the linear manifold cluster.

`indim`(*M*)
Returns a dimension of the observation space.

outdim(*M*)
Returns a dimension of the linear manifold cluster which is the dimension of the subspace.

size(*M*)
Returns the number of points in the cluster which is the size of the cluster.

points(*M*)
Returns indexes of points assigned to the cluster.

mean(*M*)
Returns the translation vector μ which contains coordinates of the linear manifold origin.

projection(*M*)
Returns the basis matrix with columns corresponding to orthonormal vectors that span the linear manifold."

separation(*M*)
Returns the instance of *Separation* object.

1.3 Example

```
using LMCLUS

# Load test data, remove label column and flip
X = readdlm(Pkg.dir("LMCLUS", "test", "testData"), ',')[:,1:end-1]'

# Initialize clustering parameters with
# maximum dimensionality for clusters.
# I should be less then original space dimension.
params = LMCLUSParameters(5)

# perform clustering and returns a collection of clusters
clust = lmclus(X, params)

# pick the first cluster
M = manifold(clust, 1)

# obtain indexes of points assigned to the cluster
l = points(M)

# obtain the linear manifold cluster translation vector
mu = mean(M)

# get basis vectors that span manifold as columns of the returned matrix
B = projection(M)

# get separation properties
S = separation(M)
```

CHAPTER 2

Parameters

The clustering properties set in `LMCLUSParameters` instance, which is defined as follows:

```
type LMCLUSParameters
    min_dim::Int                    # Minimum cluster dimension
    max_dim::Int                    # Maximum cluster dimension
    number_of_clusters::Int         # Nominal number of resulting clusters
    hist_bin_size::Int              # Fixed number of bins for the distance␣
↪histogram.
    min_cluster_size::Int           # Minimum cluster size
    best_bound::Float64             # Best bound
    error_bound::Float64            # Error bound
    max_bin_portion::Float64        # Maximum histogram bin size
    random_seed::Int64              # Random seed
    sampling_heuristic::Int         # Sampling heuristic
    sampling_factor::Float64        # Sampling factor
    histogram_sampling::Bool        # Sample points for distance histogram
    zero_d_search::Bool             # Enable zero-dimensional manifold␣
↪search
    basis_alignment::Bool           # Manifold cluster basis alignment
    dim_adjustment::Bool            # Manifold dimensionality adjustment
    dim_adjustment_ratio::Float64   # Ratio of manifold principal subspace␣
↪variance
    mdl::Bool                       # Enable MDL heuristic
    mdl_model_precision::Int        # MDL model precision encoding constant
    mdl_data_precision::Int         # MDL data precision encoding constant
    mdl_quant_error::Float64        # Quantization error of a bin size␣
↪calculation
    mdl_compres_ratio::Float64      # Cluster compression ration
    log_level::Int                  # Log level (0-5)
end
```

Here is a description of algorithm parameters and their default values:

name	description	default
min_dim	Low bound of a cluster manifold dimension.	`1`
max_dim	High bound of a cluster manifold dimension. *It cannot be larger then a dimensionality of a dataset.*	
number_of_clusters	Expected number of clusters. *Requred for the sampling heuristics.*	`10`
hist_bin_size	Number of bins for a distance histogram. *If this parameter is set to zero, the number of bins in the distance histogram determined by parameter* `max_bin_portion`.	`0`
min_cluster_size	Minimum size of a collection of data points to be considered as a proper cluster.	`20`
best_bound	Separation best bound value is used for evaluating a goodness of separation characterized by a discriminability and a depth between modes of a distance histogram.	`1.0`
error_bound	Sampling error bound determines a minimal number of samples required to correctly identify a linear manifold cluster.	`1e-4`
max_bin_portion	Sampling error bound determines a minimal number of samples required to correctly identify a linear manifold cluster. *Value should be selected from a (0,1) range.*	`0.1`
random_seed	Random number generator seed. *If not specified then RNG will be reinitialized at every run.*	`0`
sampling_heuristic	The choice of heuristic method: 1. algorithm will use a probabilistic heuristic which will sample a quantity exponential in `max_dim` and `cluster_number` parameters 2. will sample fixed number of points 3. the lesser of the previous two	`3`
sampling_factor	Sampling factor used in the sampling heuristics (see above, options 2 & 3) to determine a number of samples as a percentage from a total dataset size.	`0.01`
histogram_sampling	Turns on a sampling for a distance histogram. Instead of computing the distance histogram from a whole dataset, the algorithm draws a small sample for the histogram construction, thus improving a its performance. This parameter depends on a `cluster_number` value.	`false`
zero_d_search	Turn on/off zero dimensional manifold search.	`false`

2.1 Suggestions

Particular settings could impact performance of the algorithm:

- If you want a persistent clustering results fix a `random_seed` parameter. By default, RNG is reinitialized every time when algorithm runs.
- If a dimensionality of the data is low, a histogram sampling could speeding up calculations.
- Value `1` of `sampling_heuristic` parameter should not be used if parameter `max_dim` is large, as it will generate a very large number of samples.
- Increasing value of `max_bin_portion` parameter could improve an efficiency of the clustering partitioning, but as well could degrade overall performance of the algorithm.

2.2 Parallelization

This implementation of LMCLUS algorithm uses parallel computations during a manifold sampling stage. You need add additional workers before executing the algorithm.

CHAPTER 3

Separations

When linear manifold is formed, a distance from every point of dataset to the manifold is calculated, and a histograms of point distances to each trial manifold are computed. If the resulting histogram contains multiple modes then the mode near zero is isolated in histogram[1]. The isolated part of histogram is used to determine a separation criteria, and the data points are partitioned from the rest of the dataset on the basis of such separation.

The separation properties defined in `Separation` instance, which is defined as follows:

```
type Separation
    depth::Float64              # Separation depth (depth between separated␣
↪histogram modes)
    discriminability::Float64   # Separation discriminability (width between␣
↪separated histogram modes)
    threshold::Float64          # Distance threshold value
    globalmin::Int              # Global minimum as histogram bin index
    hist_range::Vector{Float64} # Histogram ranges
    hist_count::Vector{UInt32}  # Histogram counts
    bin_index::Vector{UInt32}   # Point to bin assignments
end
```

Separation criteria and distance threshold value can be accessed through following functions:

`criteria`(S)
: Returns separation criteria value which is product of depth and discriminability.

`threshold`(S)
: Returns distance threshold value for separation calculated on histogram of distances. It is used to determine which points belong to formed cluster.

References

[1] 10. Kittler & J. Illingworth: "Minimum Error Thresholding", Pattern Recognition, Vol 19, nr 1. 1986, pp. 41-47, DOI:10.1016/0031-3203(86)90030-0

CHAPTER 4

Utilities

Linear Manifold Clustering Algorithm relies on multiple search and optimization methods:

kittler (*X*, *bins*, *tol*)
A minimum error thresholding method for multimodal histograms[1].

otsu (*X*, *bins*)
A gray-level thresholding method for multimodal histograms[2].

mdl (*M*, *X; Pm = 32*, *Pd = 16*, *T = :Empirical*, *= 1e-4*)
Performs calculation of the minimum description length for the linear manifold cluster.

Parameters

- **M** – Linear manifold cluster description as `Manifold` type instance.
- **X** – Linear manifold cluster data as `Matrix` with points as its columns.
- **Pm** – Precision encoding constant for the model, i.e. number of bits required for encoding on element of the model description. Default value is 32 which corresponds to `Float32`.
- **Pd** – Precision encoding constant for the data.
- **T** – Type of a dataset encoding model as symbol: `:Gausian`, `:Uniform`, `:Empirical`.
- – Error tolerance for bin quantization used in an empirical model encoding

Returns number of bits required to encode linear manifold cluster with the MDL schema.

[1] 10. Kittler & J. Illingworth: "Minimum Error Thresholding", Pattern Recognition, Vol 19, nr 1. 1986, pp. 41-47, DOI:10.1016/0031-3203(86)90030-0

[2] 14. Otsu: "A threshold selection method from gray-level histograms", Automatica, 1975, 11, 285-296, DOI:10.1109/TSMC.1979.4310076

References

Notes:

All methods implemented in this package adopt the column-major convention: in a data matrix, each column corresponds to a sample/observation, while each row corresponds to a feature (variable or attribute).

References

Index