Unsupervised learning - clustering

Jacek Tabor

GMUM

Journal Club 4.XI.2016

Tasks

Typically:

- clustering: divide the data into groups / clusters (unsupervised learning)
- semisupervised clustering: we have some partial information about clusters (must-be and cannot-be links)
- classification: we know (at least some) labels, we want to predict the labels for the unknown (supervised)

Clustering: it is hard to assess the quality (everyone has its own clustering, very data dependent), we typically use Rand index to compare the obtained clustering with some predefined (given in the data, for example from the UCI repository). Basic methods: k-means, hierarchical, density based (EM, DBSCAN), subspace clustering, etc.

Tasks

Typically:

- clustering: divide the data into groups / clusters (unsupervised learning)
- semisupervised clustering: we have some partial information about clusters (must-be and cannot-be links)
- classification: we know (at least some) labels, we want to predict the labels for the unknown (supervised)

Clustering: it is hard to assess the quality (everyone has its own clustering, very data dependent), we typically use Rand index to compare the obtained clustering with some predefined (given in the data, for example from the UCI repository). Basic methods: k-means, hierarchical, density based (EM, DBSCAN), subspace clustering, etc.

Classification: easy to measure the quality (divide the data into training and testing). Basic methods: SVM, Bayes, neural networks, regression, ELM (extreme learning machines), decision trees and random forests.

Tasks

Data Types for classification/clustering:

1. hard clustering (sets),
2. soft clustering (probabilities)
3. fuzzy clustering

In the case od clustering (not classificstion) we have two possible tasks: either cluster the existing data or construct a division of the whole space.

Why we need?

In fact we do not know what is the right distance for the data, but often not the one we have. Typically much better results are obtained after some kind of data preprocessing.

We have two methods:

- changing/creating the metric,
- embedding the data into a larger space.

Metric

Metric is a measure of distance

Definition

Distance (dissimilarity):

- $d(x, y) \geq 0$
- $d(x, y) = d(y, x)$ (symmetric)
- $d(x, x) = 0$

Semi-metric: $d(x, z) \leq d(x, y) + d(y, z)$.

Metric $d(x, y) = 0$ iff $x = y$.

Metric is defined on most used structures, in particular: standard space $\mathbb{R}^d$, texts, graphs, matrices, subspaces, sets.

Some examples

$\mathbb{R}^d$ (Euclidean space, when the usual norm):

$$d(x, y) = \|x - y\|_2 \text{ where } \|x\|_2^2 = \sum_i x_i^2.$$

Texts: cosine similarity measure

$$\cos(A, B) = \langle A, B\rangle/(\|A\| \cdot \|B\|).$$

Sets and bit sequences: for a set A by $|A|$ we denote the number of elements of A, Hamming distance: l^1 distance, the number of elements two sequences differ: $|B \setminus A| + |A \setminus B| = |A \cup B| - |A \cap B|$

Metric learning

PROBLEM: how to find the right metric for the data?

Find such a scalar product, that the data is spherical. The answer is given by Mahalanobis: $\|x\|_\Sigma^2 = x^T \Sigma^{-1} x$, equivalently

$$\|x\|_\Sigma = \|\Sigma^{-1/2} x\|,$$

where Σ is the estimator of the covariance of the data.

Dimension reduction

In some cases the dimension is to high from the numerical point of view, and we have to reduce it.

We can do it in the linear way - PCA, or nonlinear SOM. There is also an alternative approach of elastic maps or principal curves.

Linear: decorelation

We can construct a dual approach to Mahalanobis. Instead of changing the metric, we can change the data, by applying linear decorellation:

$$x \to \Sigma^{-1/2} x$$

Random embedding

Applied in particular in ELM and RBF. The idea is to enrich the space by taking nonlinear projections into reals, we typically take a metric in the data and use distance from chosen centers which are understood as neurons:

$$x \to d(x, w_i)$$

where w_i is chosen to be i-th neuron.

Kernel trick

Kernel space

Gaussian kernel: embedding into a larger (L^2 space):

$$x \to N(x, \sigma^2 I)$$

rysunek!

scalar product $\exp(-\|x-y\|^2/(2\sigma^2))$ and distance

$$d^2(x,y) = 2 - \exp(-\|x-y\|^2/(2\sigma^2)).$$

intuicja

k-means

The first and oldest method of clustering. One of the first papers about it was written by the famous polish mathematician H. Steinhaus.

More information: [Hans-Hermann Bock *Clustering Methods: A History of k-Means Algorithms*]

Problem (optimization)

Find such division of the set X into k subsets $X_1, \ldots, X_k$, to minimalize the cost function $\sum\limits_{i=1}^{k} ss(X_i)$ where ss (within cluster sum of squares):

$$ss(Y) := \sum_{y \in Y} \|y - m_Y\|^2,$$

and $m_Y = \frac{1}{|Y|} \sum_{y \in Y} y$ denotes the mean (“center of weight”) of data set Y.

Advantages: fast and easy to implement, easily scalable. Disadvantages: the result depends on the choice (change) of the coordinate system, does not find the right / optimal number of clusters, clusters are more or less of similar size.

Advantages: fast and easy to implement, easily scalable. Disadvantages: the result depends on the choice (change) of the coordinate system, does not find the right / optimal number of clusters, clusters are more or less of similar size.

The more clusters the better!

There are some adaptations which use BIC (or analogues) : Bayesian Information Criterion, to find the right number of groups.

Lloyds' method to finding the (locally) optimal solution

In general there is no easy way of finding optimal splittings, we describe some.

1. We choose from X in a random way k points $m_1, \ldots, m_k$ (initial cluster centers)
2. Each point from X we assign to those center (or more precisely its index) which is the nearest, i.e. to those for which the distance $\|x - m_i\|$ is minimal
3. for such constructed cluster we compute new centers, and if they differ from the previous we return to point two

Lloyds’ method to finding the (locally) optimal solution

In general there is no easy way of finding optimal splittings, we describe some.

1. We choose from X in a random way k points $m_1, \ldots, m_k$ (initial cluster centers)
2. Each point from X we assign to those center (or more precisely its index) which is the nearest, i.e. to those for which the distance $\|x - m_i\|$ is minimal
3. for such constructed cluster we compute new centers, and if they differ from the previous we return to point two

Warning: the method usually finds only local minima, and there there is a need for multiple initial stars (the problem of finding the global minima is NP-hard).

EM(=expectation maximization) method

[Geoffrey McLachlan,Thriyambakam Krishnan *The EM Algorithm and Extensions*]

EM method (in its most commonly found example of Gaussian Mixture Models) tries to fit (taking into account the ML=maximal likelihood method) to data X the "mixture" density of the form

$$X \sim p_1 f_1 + \ldots + p_k f_k,$$

where f_i belong to the predefined family of densities (typically Gaussians=normal densities). In other words it is a typical density estimation approach.

Then the point x belongs to those group for which the value of $p_i f_i(x)$ is maximal.

Advantages: does not depend on the scaling or affine change in the coordinate system, usually fits nicely to the data and returns the reliable densite estimation, one can use various Gaussian type models.

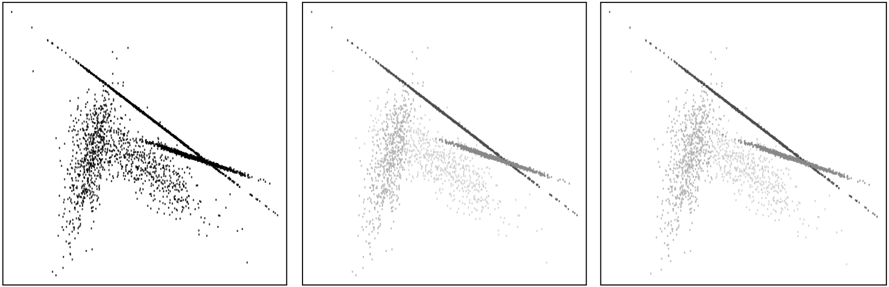
Disadvantages: usually slow (one cannot easily use the Hartigan approach), does not automatically detect the right number of clusters, use of Gaussian models need complicated nonlinear optimization.

Advantages: does not depend on the scaling or affine change in the coordinate system, usually fits nicely to the data and returns the reliable densite estimation, one can use various Gaussian type models.

Disadvantages: usually slow (one cannot easily use the Hartigan approach), does not automatically detect the right number of clusters, use of Gaussian models need complicated nonlinear optimization.

The more clusters the better!

Example of GMM



(a) Data coming from the mixture of 4 gaussians.

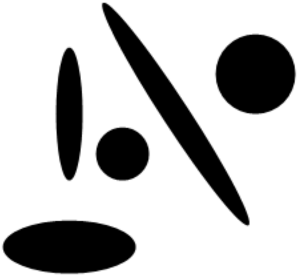
(b) Clustering with the use of EM with gaussians.

(c) CEC starting with 10 gaussians.

Figure: Comparison of clustering of the mixture of 4 gaussians by EM (with $k = 4$) and Gaussowskim CEC(=cross-entropy clustering) starting from initially $k = 10$ clusters.

Gaussian models

We can specify what class of gaussians we are interested in. Most common is clearly the class if all gaussians, but we ca encounter also radial(spherical) ones, with diagonal covariance, etc.



Density clustering: DBScan.

In general we can also cluster the data if we have a estimation of the density, for example by kernel density approach. Since in this case we do not have natural clusters, we fix some level and we divide the data according to this level, wheren the points below we track as outliers.

In the case of DBScan, we simply look at the graph of distances.

rysunek!

Outliers/anomaly detection

Useful in particular in industry, but also in data preprocessing. Both discriminative and generative models. We want to find two clusters, but one should correspond to the main data, and the other to noise/errors/outliers.

Typical solutions: discriminative: one class SVM, generative- one estimated the density, and takes the points with small probability.

Ockham's razor

Among competing hypotheses, the one with the fewest assumptions should be selected. Other, more complicated solutions may ultimately prove correct, but—in the absence of certainty—the fewer assumptions that are made, the better.

Figure: William of Ocham 1287–1347.

Morse code

Morse Code/telegraph [1836 S. Morse, A. Vail]: Vail determined the frequency of use of letters in the English language by counting the movable type he found in the type-cases of a local newspaper in Morristown: *the code of typical text should be as short as possible*, i.e.: the more common the letter, the shorter should be its code.

Morse code

Morse Code/telegraph [1836 S. Morse, A. Vail]: Vail determined the frequency of use of letters in the English language by counting the movable type he found in the type-cases of a local newspaper in Morristown: *the code of typical text should be as short as possible*, i.e.: the more common the letter, the shorter should be its code.

ANECDOTE: Ernest Hemingway [newspaper notes from war in Spain]

Morse code

Morse Code/telegraph [1836 S. Morse, A. Vail]: Vail determined the frequency of use of letters in the English language by counting the movable type he found in the type-cases of a local newspaper in Morristown: *the code of typical text should be as short as possible*, i.e.: the more common the letter, the shorter should be its code.

ANECDOTE: Ernest Hemingway [newspaper notes from war in Spain]
READER (with admiration): how is it possible to learn to write the notes so short, but so wonderfully informative?

Morse code

Morse Code/telegraph [1836 S. Morse, A. Vail]: Vail determined the frequency of use of letters in the English language by counting the movable type he found in the type-cases of a local newspaper in Morristown: *the code of typical text should be as short as possible*, i.e.: the more common the letter, the shorter should be its code.

ANECDOTE: Ernest Hemingway [newspaper notes from war in Spain]
READER (with admiration): how is it possible to learn to write the notes so short, but so wonderfully informative?
HEMINGWAY: easy – you just have to pay one dollar for each word send by the telegraph

Morse code

Morse Code/telegraph [1836 S. Morse, A. Vail]: Vail determined the frequency of use of letters in the English language by counting the movable type he found in the type-cases of a local newspaper in Morristown: *the code of typical text should be as short as possible*, i.e.: the more common the letter, the shorter should be its code.

ANECDOTE: Ernest Hemingway [newspaper notes from war in Spain]
READER (with admiration): how is it possible to learn to write the notes so short, but so wonderfully informative?
HEMINGWAY: easy – you just have to pay one dollar for each word send by the telegraph

Information=money

Entropy

C. Shannon [1948. "A Mathematical Theory of Communication". Bell System Technical Journal 27] Precise formulation of the idea of Morse leads to the formal definition of Shannon's entropy: *in the optimal coding (that is those with the shortest code-length), if the signal appears with probability p, its code-length should equal to* $-\log_2 p$.

Entropy

C. Shannon [1948. "A Mathematical Theory of Communication". Bell System Technical Journal 27] Precise formulation of the idea of Morse leads to the formal definition of Shannon's entropy: *in the optimal coding (that is those with the shortest code-length), if the signal appears with probability p, its code-length should equal to* $-\log_2 p$.

Thus if the symbols $x_1, \ldots, x_n$ appear with probabilities $p_1, \ldots, p_n$:

$$\text{entropy} = \text{minimal code length} = p_1 \cdot -\log_2 p_1 + \ldots + p_n \cdot -\log_2 p_n.$$

In practice leads to Huffman's coding (used for example in jpg).

Minimum description length (MDL) principle

Formulation of the MDL principle Jorma Rissanen [1978. "Modeling by shortest data description". Automatica 14]: the best hypothesis for a given set of data is the one that leads to the best compression of the data.

Minimum description length (MDL) principle

Formulation of the MDL principle Jorma Rissanen [1978. "Modeling by shortest data description". Automatica 14]: the best hypothesis for a given set of data is the one that leads to the best compression of the data.

P. Grünwald, [1998]: "any regularity in a given set of data can be used to compress the data, i.e. to describe it using fewer symbols than needed to describe the data literally."

Minimum description length (MDL) principle

Formulation of the MDL principle Jorma Rissanen [1978. "Modeling by shortest data description". Automatica 14]: the best hypothesis for a given set of data is the one that leads to the best compression of the data.

P. Grünwald, [1998]: "any regularity in a given set of data can be used to compress the data, i.e. to describe it using fewer symbols than needed to describe the data literally."

Connected to the notion of Kolmogorov complexity [1963. "On Tables of Random Numbers". Sankhya Ser. A. 2]: the complexity of a string is the length of the shortest possible description of the string in some fixed universal description language.

How many clusters?

In most clustering methods, one has to specify the number of clusters. This implies, that the procedure does not return the right number of clusters (or reduce unnecessary clusters on-line during the clustering procedure).

How many clusters?

In most clustering methods, one has to specify the number of clusters. This implies, that the procedure does not return the right number of clusters (or reduce unnecessary clusters on-line during the clustering procedure).

Advantages of the use of MDL principle in clustering:

- automatically reduces the complexity of the model (number of clusters)
- has high adaptability
- (often) small requirements on the the data: (we do not require vector or metric structures, but only the existence of encoding or compression methods)
- easily allows potential modifications

MDL principle in clustering

Requirements
Data type which we want to cluster, available compression methods $\mathcal{W}$.

MDL principle in clustering

Requirements

Data type which we want to cluster, available compression methods $\mathcal{W}$.

Determination of the overall cost of the memory

Let us assume that message $X = (x_1, \ldots, x_N)$ and the compression methods $w_1, \ldots, w_K \in \mathcal{W}$ are given. We denote the algorithm that encodes point x_l (defines the cluster it belongs to) with w_{k_l}, where $k_l \in \{1, \ldots, K\}$.
Then the memory cost of coding the message X equals

$$\sum_{i=1}^{K} \text{memory cost of } w_k + \sum_{l=1}^{N} (\text{cost of identification of } k_l + \text{the amount of memory algorithm } w_{k_l} \text{ uses to code } x_l).$$

MDL principle in clustering

Requirements
Data type which we want to cluster, available compression methods $\mathcal{W}$.

Determination of the overall cost of the memory
Let us assume that message $X = (x_1, \ldots, x_N)$ and the compression methods $w_1, \ldots, w_K \in \mathcal{W}$ are given. We denote the algorithm that encodes point x_l (defines the cluster it belongs to) with w_{k_l}, where $k_l \in \{1, \ldots, K\}$.
Then the memory cost of coding the message X equals

$$\sum_{i=1}^{K} \text{memory cost of } w_k + \sum_{l=1}^{N} (\text{cost of identification of } k_l + \text{the amount of memory algorithm } w_{k_l} \text{ uses to code } x_l).$$

Memory minimization step
We seek K, compression methods $w_1, \ldots, w_K$ and indices k_l, such that the total cost needed to encode the message X is minimal.

MDL principle in clustering

Requirements
Data type which we want to cluster, available compression methods $\mathcal{W}$.

Determination of the overall cost of the memory
Let us assume that message $X = (x_1, \ldots, x_N)$ and the compression methods $w_1, \ldots, w_K \in \mathcal{W}$ are given. We denote the algorithm that encodes point x_l (defines the cluster it belongs to) with w_{k_l}, where $k_l \in \{1, \ldots, K\}$.
Then the memory cost of coding the message X equals

$$\sum_{i=1}^{K} \text{memory cost of } w_k + \sum_{l=1}^{N} (\text{cost of identification of } k_l + \text{the amount of memory algorithm } w_{k_l} \text{ uses to code } x_l).$$

Memory minimization step
We seek K, compression methods $w_1, \ldots, w_K$ and indices k_l, such that the total cost needed to encode the message X is minimal.

Construction of the clustering
Points which are coded by the same algorithm are assigned to the same cluster.

MDL principle in clustering

Observe that in the above approach the use of any encoding method takes memory (needed for its determination), as a consequence we obtain the upper limit for the amount of possible clusters. Moreover, in practice – when the amount of elements encoded by a given algorithm is small – it will be worthwhile to give up the use of this algorithm, which leads to a reduction of the complexity of the constructed model (understood as a number of clusters).

Differential entropy

The coder adapted to the data generated by the continuous random variable with the density f.

Differential entropy is the limiting case of entropy (smaller and smaller bins):

$$h(f) := \int f(x) \cdot -\log_2(f(x))dx.$$

Differential entropy

The coder adapted to the data generated by the continuous random variable with the density f.

Differential entropy is the limiting case of entropy (smaller and smaller bins):

$$h(f) := \int f(x) \cdot -\log_2(f(x))dx.$$

Cross-entropy

$$h(f) := \int g(x) \cdot -\log_2(f(x))dx.$$

Gives the memory cost of compressing the data with the density g by the coder optimally adapted to the density f.

Gaussian models

In practice, we can compute the above for the Gaussian models!

Gaussian models

In practice, we can compute the above for the Gaussian models!

There is only the need for the knowledge of the mean of the data and the covariance matrix.

Papers

TABOR, SPUREK:
Cross-entropy clustering, Pattern Recognition 2014
TABOR, MISZTAL:
Detection of elliptical shapes via cross-entropy clustering (IbPRIA 2013)
SPUREK, TABOR, ZAJĄC:
Detection of disk-like particles in electron microscopy images (CORES 2013)
ŚMIEJA, TABOR:
Image segmentation with use of cross-entropy clustering (CORES 2013)

Available at: `http://www.ii.uj.edu.pl/~tabor`.

EM vs CEC

[Geoffrey McLachlan,Thriyambakam Krishnan *The EM Algorithm and Extensions*]

Fit the data X by

$$X \sim p_1 f_1 + \ldots + p_k f_k,$$

where f_i belong to the fixed density family.

CEC:

$$X \sim \max(p_1 f_1, \ldots, p_k f_k),$$

Cluster reduction

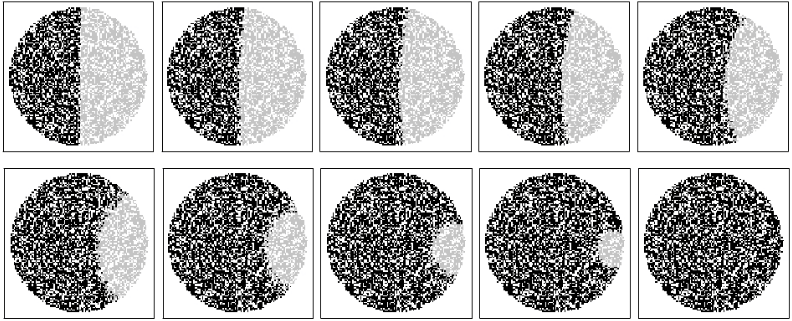


Figure: Cluster reduction.

Packages in R

CEC R - (P. Spurek)

GMUM-R (W. Czarnecki)