Clustering

Machine Learning in Computer Vision

This report is what I’ve learn about clustering in Machine Learning and Computer Vision.

It is not a document about clustering, I just write down what I found out and try to prove it through examples, tests with skicit-learn so please feel free to correct me if something goes wrong.

A lot of images in this report are borrowed from [Wiki](https://en.wikipedia.org/wiki/Cluster_analysis) and [this online courses](https://www.udacity.com/course/introduction-to-computer-vision--ud810).

|  |  |
| --- | --- |
|  | **More information:** please email me at [14520040@gm.uit.edu.vn](https://mail.google.com/) |

# Overview

Let’s start with some numbers from [KDnuggest](https://www.kdnuggets.com/2016/09/poll-algorithms-used-data-scientists.html).

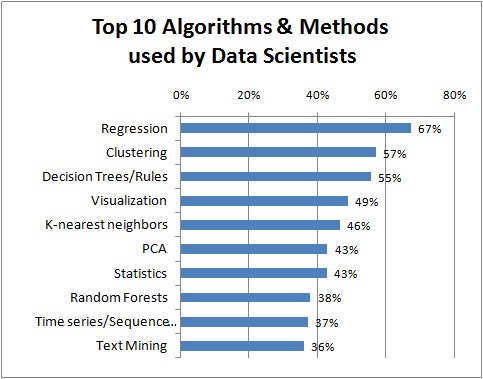


Figure 0. Top 10 algorithms and methods used by data Scientists

Hm… Clustering seems be an interesting stuff. Let’s find out why.

What is clustering?

Clustering is the task of dividing data (samples) into a number of groups (clusters). Samples in the same group are more similar to each other than to those in other groups.

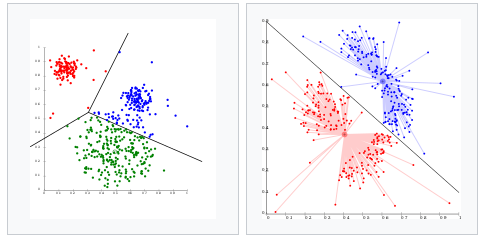


Figure 1. Clustering examples

Note: Since the task of clustering is subjective. It can be used in different ways to achieving different goals.

In the example below, we can use clustering for either grouping by gender or by relationship between people.

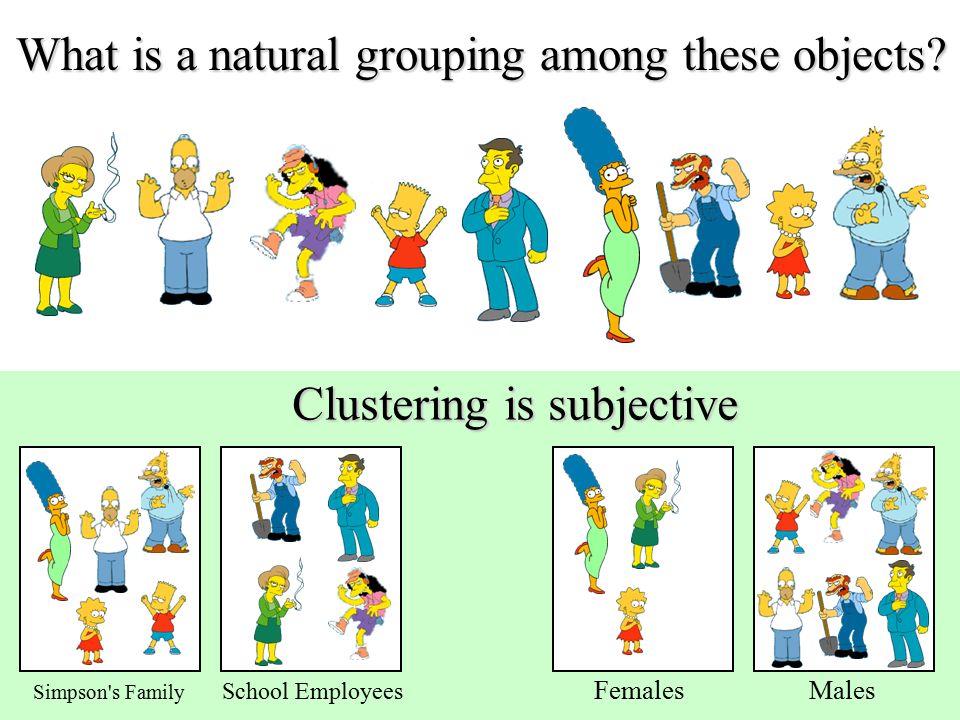


Figure 2. Clustering is subjective

Why do I need clustering in Machine Leanring?

In Machine learning, Clustering is used to find structure of unlabeled data. It is the most common form of unsupervised learning.

Given a dataset we don’t know anything about, a clustering algorithm can discover groups of objects where the average distances between the members of each cluster smaller than to those in other cluster.

How to measure performance a clustering algorithm?

There are two common ways to measure performance a clustering algorithm: Internal measure and External measure.

**Internal measure**: measuring without any external information

In comon, Objective function value is used.

Ex. Minimal SSD (Sum of square distance) or SSE (Sum of square distance error) between every data point in a cluster C to the center point .

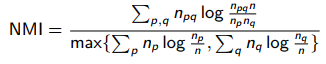
External measure: measuring agreement between two partions – the ground-truth labels and the result of clustering task. (used when we have data with ground-truth labels)

There are a lot of external functions can be used. It all calculates using different bettwen the structure of ground-truth and clustering reulst. Below is some of those.

* Clustering accuracy (CA): Find a maximum bipartite matching (using Hungarian Algorithm or Munkres Algorithm) between ground-truth labels and assigned labels.



* Normalized mutual information (NMI): Compare the distributions defined by ground-truth partitioning and clustering result.



: Number of data points with p-th ground-truth label

: Number of data points with q-th assigned label

: Number of data points with p-th ground-truth label and q-th assignedlabel

* Purity:
* Entropy:

Note: Clustering task is just grouping so the labels from clustering result is just the name of the group, it will not be the same with the labels in ground truth and External functions above also do not use directly the labels value for calculating.

Ex. Labels\_true = [1, 1, 2, 2, 3] // the ground-truth

Labels\_pred = [3, 3, 1, 1, 2] // the clustering result

NMI (labels\_true, labels\_pred) = 1 //it means clustering absolutely right

As we can see, the clustering result is just a group name, not really a labels. The really important thing we can get from result is structure of data.

One more thing

I don’t really know how those external functions were created. For now, I just use it as a measure.

The formulas are copied from [here](http://www.math.ucla.edu/~dakuang/cse6040/lectures/6040_lecture13.pdf).

Which clustering algorithm should I use?

The answer is It depends on what data we have, and what we are going to do with the data.

There are several different clustering algorithms we can easy to find with just a name such as Kmeans, Spectral clustering, DBSCAN, etc.

In this project, I will go on detail about K-means and just an over view about other algorithms. The reason of choosing K-means will be metioned in below.

Then, what I can really do with clustering?

Hm… I can’t even count. There are a lot of cool stuffs we can do with clustering.

We just need to remember that clustering is used to find out the struture of data. Therefore, whenever we have data and want to do something cool with it but we don’t know what we are really doing, then just using clustering to have a better view.

Ex. Clustering can be used to group all the shopping items available on the web into a set of unique products (like eBay)

Or in the study of social networks, clustering may be used to recognize communities within large groups of people

Or even cluster analysis can be used to identify areas where there are greater incidences of particular types of crime.

Or at least, we can used it for school project (Ex. Image Segmentation)

K-means Clustering

Kmeans is a common clustering algorithm used in Computer Vision. I have no number for this one but just see how many Computer Vision courses have “Kmean” as an basic method for segmentation and orther tasks.

Basic idea

Minimize SSD between all points and their nearest clustering center. (This is an internal measure above).

Algorithm

1. Randomly initializes K cluster centers
2. Determine points in each cluster.

For each point p in dataset, find closest , put p into cluster i

1. Given points in each cluster, calculate center again.
2. If any has changed, repeat step 2.

Ex. Implement Kmeans would be something like this (from a to f)

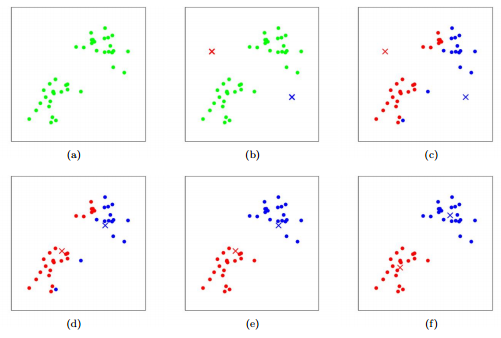


Figure 3. An example of Kmean clustering

Pros and Cons

Pros

Very simple algorthm, easy to implement, widely usage.

Converges to a local minimun of error function (that how the basic idea of Kmean effects).

Cons

Need to pick K.

* *It can be easy improved by picking diffrerent K and see which is good (depend on our goal of clustering).*

Sensitive to initialization

*To see how initialization effects on Kmeans, I implement k-mean few times with a simple dataset, here is some result*

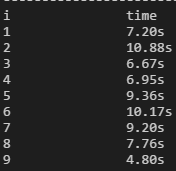


Figure 4. Execution time has changed by initialization with the same data.

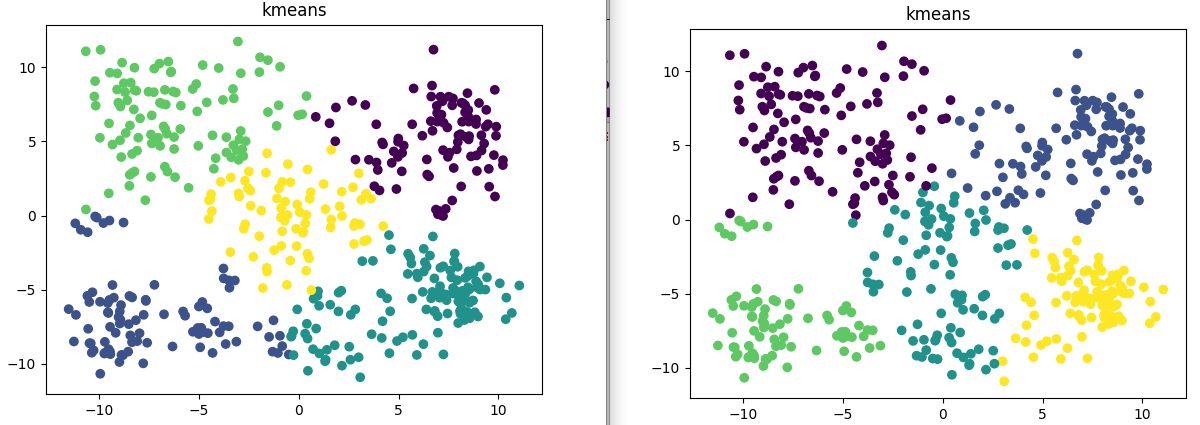


Figure 5. The result has also changed by initialization

To simplest way to avoid this problem I’ve found is re-runing Kmeans and “hope” the random initialization will be better in next time (just run it again if we feel the result is not good enough).

Only finds “spherical” clusters

Sensitive to outliers

# Some other clustering algorithm

.

Simple Comparison

.

# Concluson