Discrete Mathematics

Mid-term Report

Clustering

指導教授 : 曾龍 教授

成功大學電機系 劉宥辰

- 簡介
- 應用
- 實做範例

・簡介

Clustering（群聚分析 ）

- Cluster群聚：一群dataobjects
- – 在同一群內相當相似
  - – 在不同群內非常不相似
    - Cluster analysis
    - – 把資料依相似性分群

- 簡介
- 應用
- 實做範例

- Marketing：

- Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs

- Insurance:

- Identifying groups of motor insurance policy holders with a high average claim cost

- City Planning：

- Identifying groups of houses according to their house type, value, and geographical location

- 簡介
- 應用
- 實做範例

・實作範例

- Reference material
- Clustering Problem（Training data）
- Implementation
- Algorithm : K-means

Credit 宥辰

- Reference material

tf.argmin(input, dimension, name=None)
函數解說：沿著需要的維度找尋最小值的索引值，最小由0開始

tf.reduce_sum()
reduce_sum() 就是求和，由於求和的對象是tensor，所以是沿着tensor的某些维度求和。reduction_indices是指沿tensor的哪些维度求和。
https://www.zhihu.com/question/51325408

tf.reshape(tensor,shape, name=None)
函数的作用是將tensor變換為參數shape的形式。
其中shape為一个列表形式，特殊的一點是列表中可以存在-1。-1代表的含義是不用我們自己指定這一维的大小，函数会自動计算，但列表中只能存在一個-1
好了我想说的重点还有一个就是根据shape如何变换矩阵。其实简单的想就是，
reshape（t, shape） => reshape(t, [-1]) =>reshape(t, shape)
首先将矩阵t变为一维矩阵，然后再对矩阵的形式更改就可以了。
http://blog.csdn.net/zeuseign/article/details/72742559

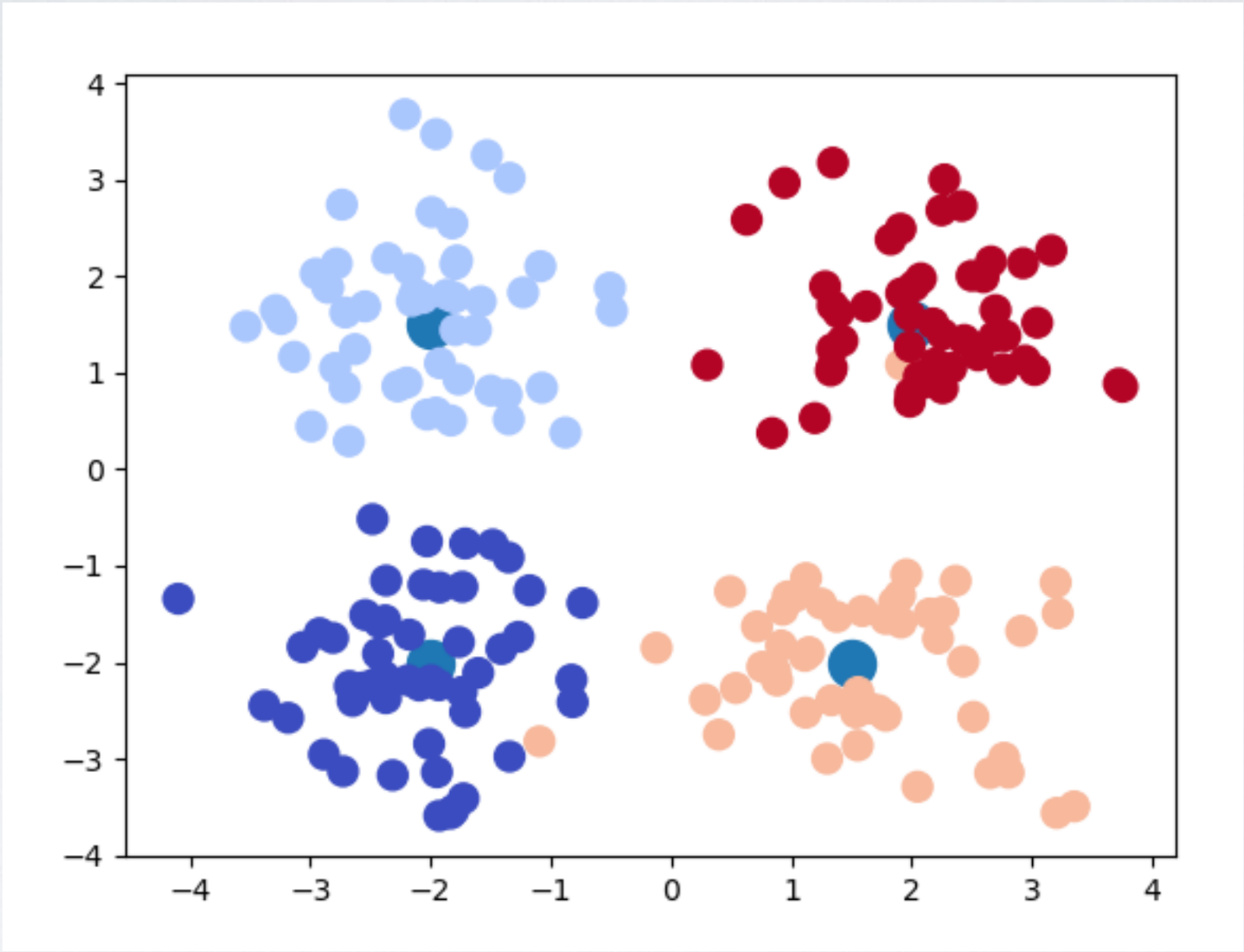
tile() 平鋪之意，用於在同一维度上的複製
http://blog.csdn.net/xwd18280820053/article/details/72867818

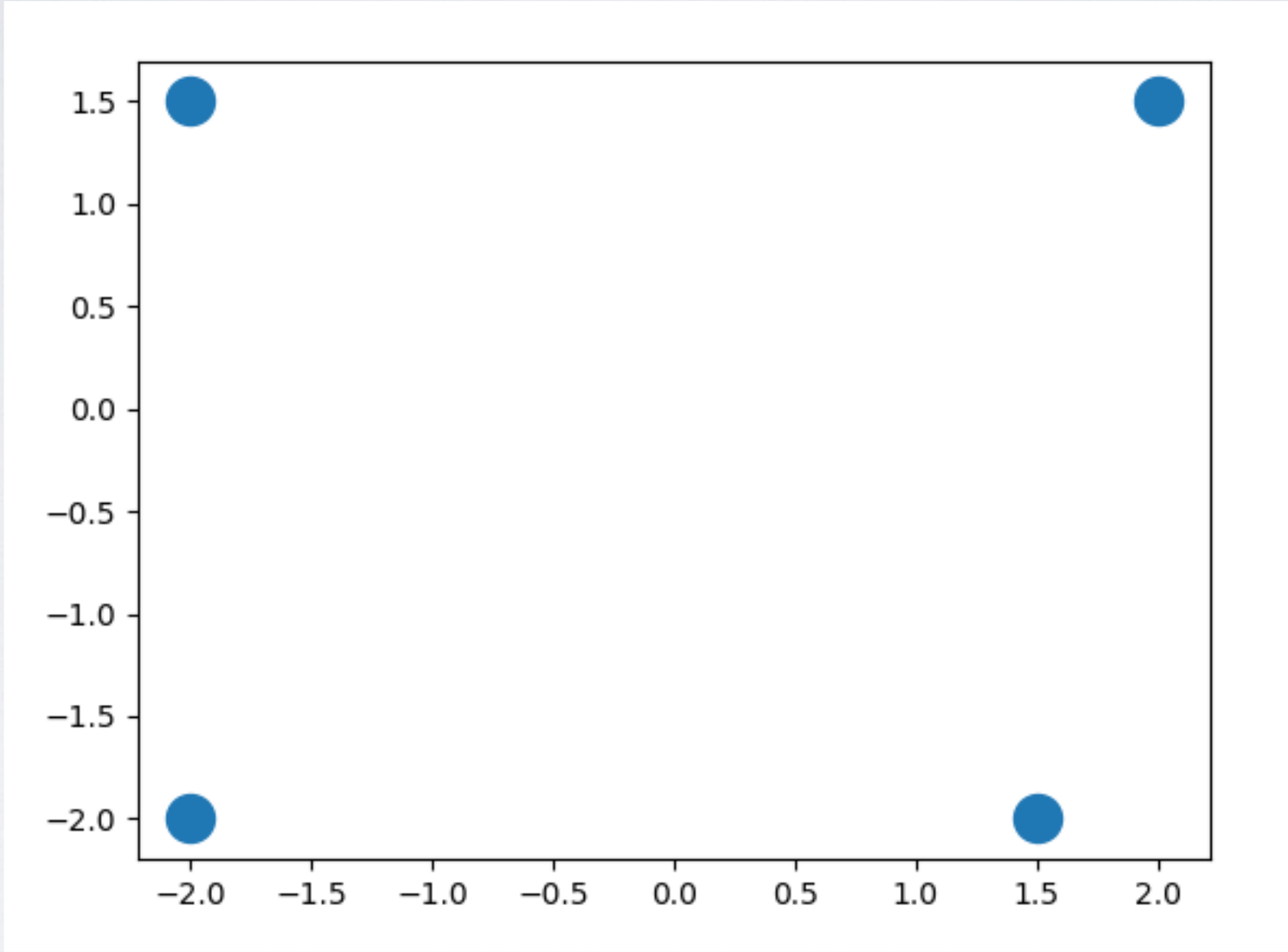
tf.reduce_any() 計算tensor中各個元素的邏輯或運算
http://blog.csdn.net/lhanchao/article/details/51442182

tf.unsorted_segment_sum(data, segment_ids, num_segments, name=None)

這個函數的作用是沿着segment_ids指定的维度，分割張量data中的值，並且返回累加值。
計算公式為:
其中，segment_ids[j] == i。这个API和SegmentSum最大的區別是，這個API不需要從0到k有序排列，可以亂序排列，並且該API不需要包含從0到k。
如果對於给定的分割區間ID i，output[i] = 0。那麼，num_segmetns應該等於不同的段ID的數量。
http://www.jianshu.com/p/4daafdbcdddf

- Clustering Problem（Training data）


4
3
2
1
0
−1
−2
−3
−4
−4
−3
−2
−1
0
1
2
3
4


1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

• Implementation

- Using Machine Learning（unsupervised learning）
- Framework : tensorflow

- Window 10
- Python 3.6
- GPU：GTX950

- Using Machine Learning（unsupervised learning）

- Window 10
- Python 3.6
- GPU：GTX950

- Supervised Learning：
- Training the model with “Labeled” data , which is a common way in implementation
- Advantage: Higher precision
- Disadvantage: More cost

- Unsupervised Learning：
- Training with “Un-Labeled” data , common in clustering , but pure unsupervised learning is not common in implementation
- Advantage : Less cost

- Algorithm : K-means

K-means Algorithm

Input:

Training set Training set: $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$

$x^{(i)} \in \mathbb{R}^n$ (drop $x_0 = 1$ convention, thus $x^{(i)} \notin \mathbb{R}^{n+1}$)

K: number of clusters

Randomly initialize K cluster centroids $\mu_1, \mu_2, \ldots, \mu_K \in \mathbb{R}^n$

Repeat {

// cluster assignment step

for i = 1 to m

$c^{(i)}$:= index (from 1 to K) of cluster centroid closest to $x^{(i)}$

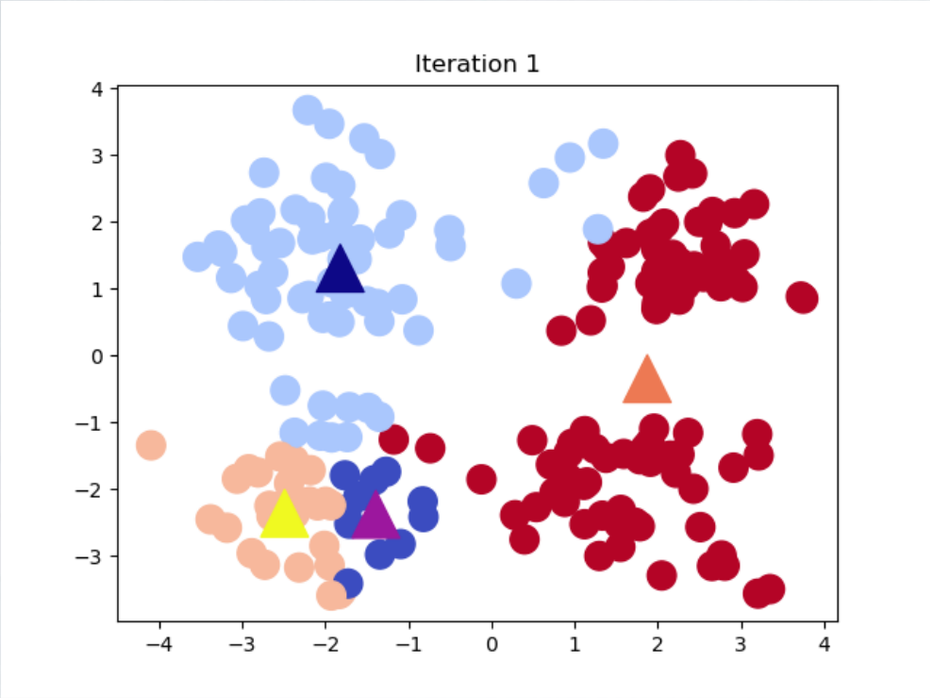
(或表示為 $c^{(i)} := min_k ||x^{(i)} - \mu_k||^2$)

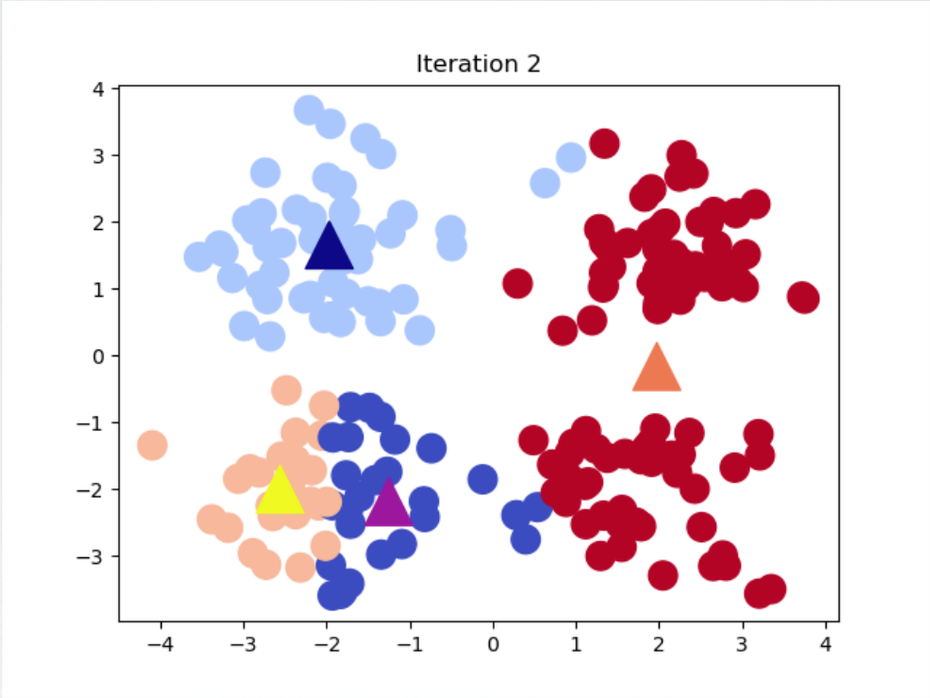
// move centroid step

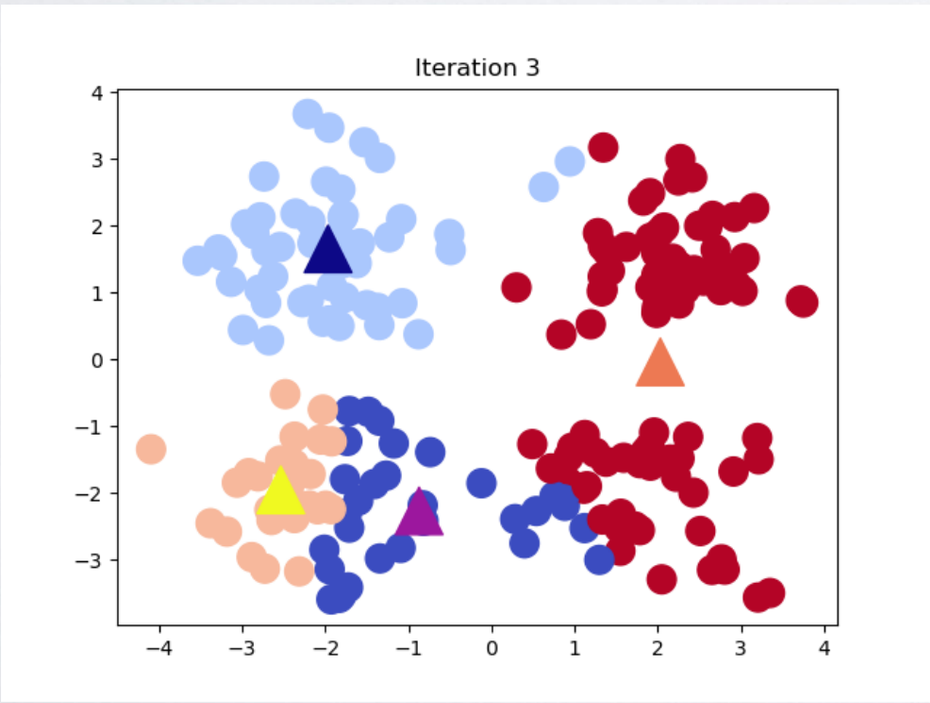
for k = 1 to K

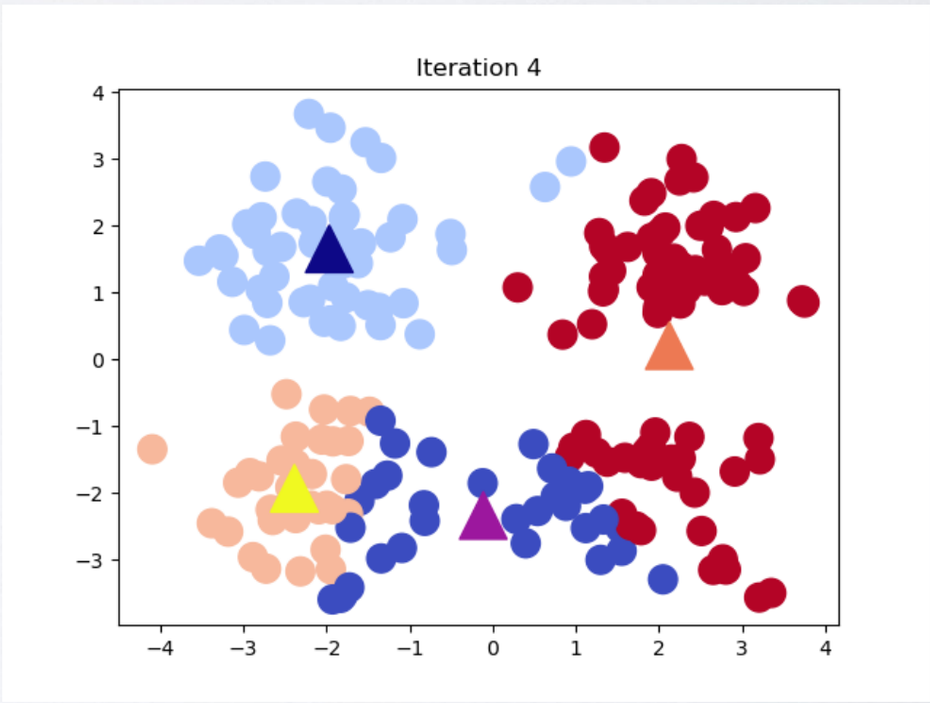
μ_k := average (mean) of points assigned to cluster k

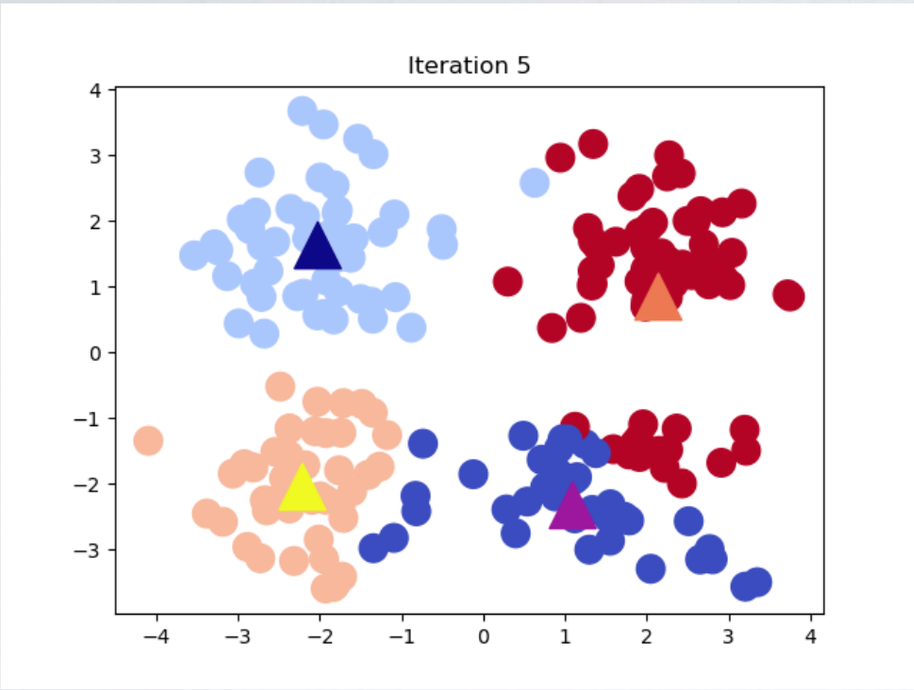
}

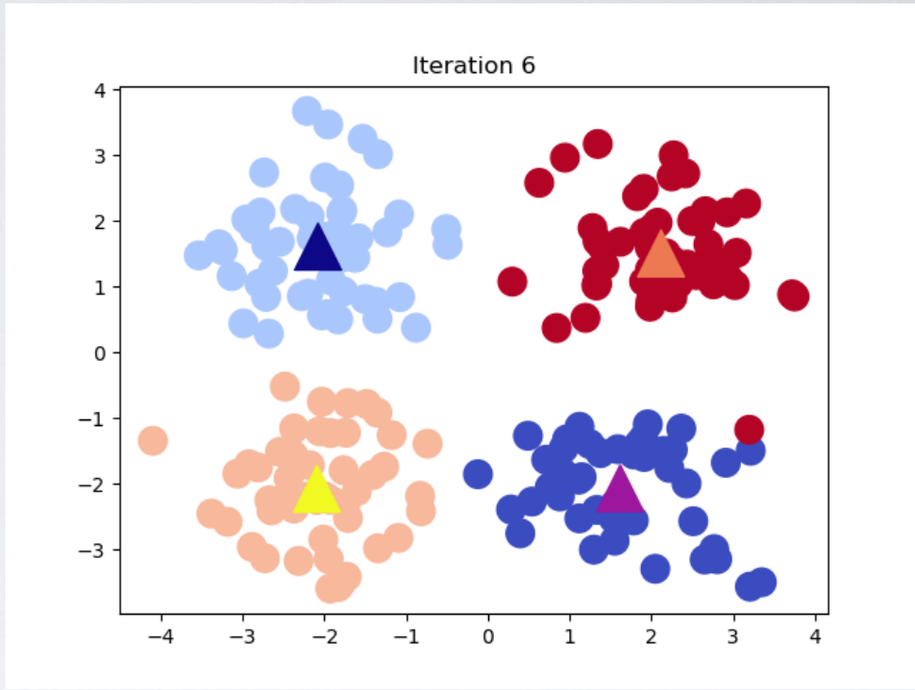

Iteration 1
4
3
2
1
0
−1
−2
−3
−4
−3
−2
−1
0
1
2
3
4

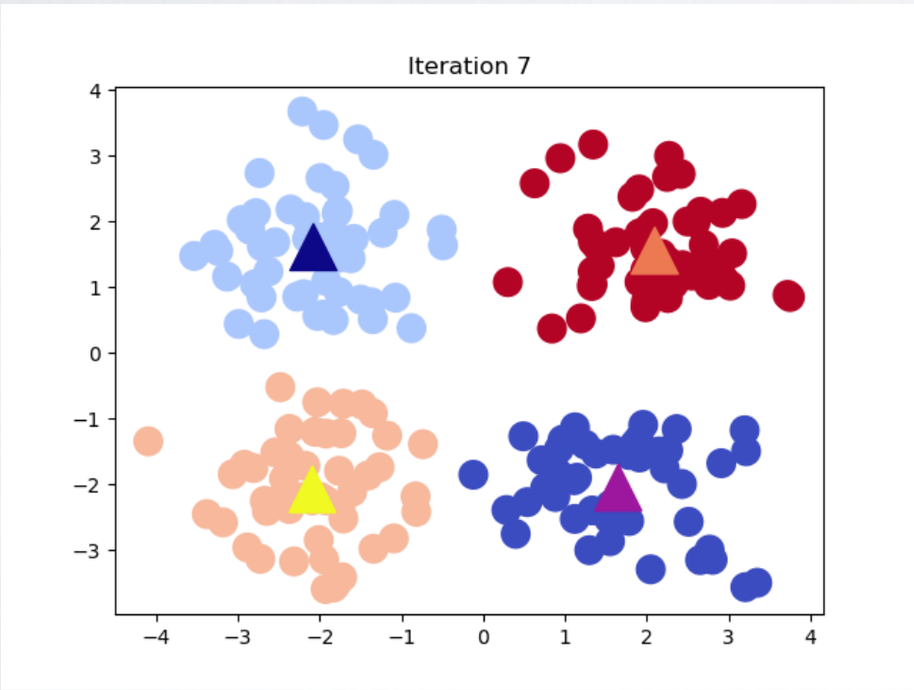

Iteration 2
4
3
2
1
0
−1
−2
−3
−4
−3
−2
−1
0
1
2
3
4

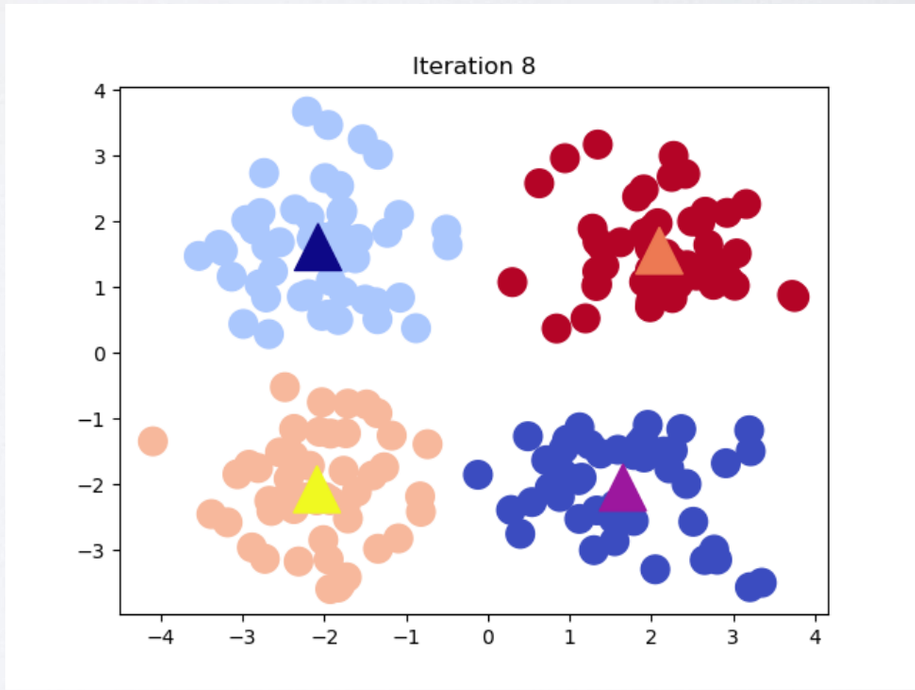

Iteration 3
4
3
2
1
0
−1
−2
−3
−4
−3
−2
−1
0
1
2
3
4


Iteration 4
4
3
2
1
0
−1
−2
−3
−4
−3
−2
−1
0
1
2
3
4

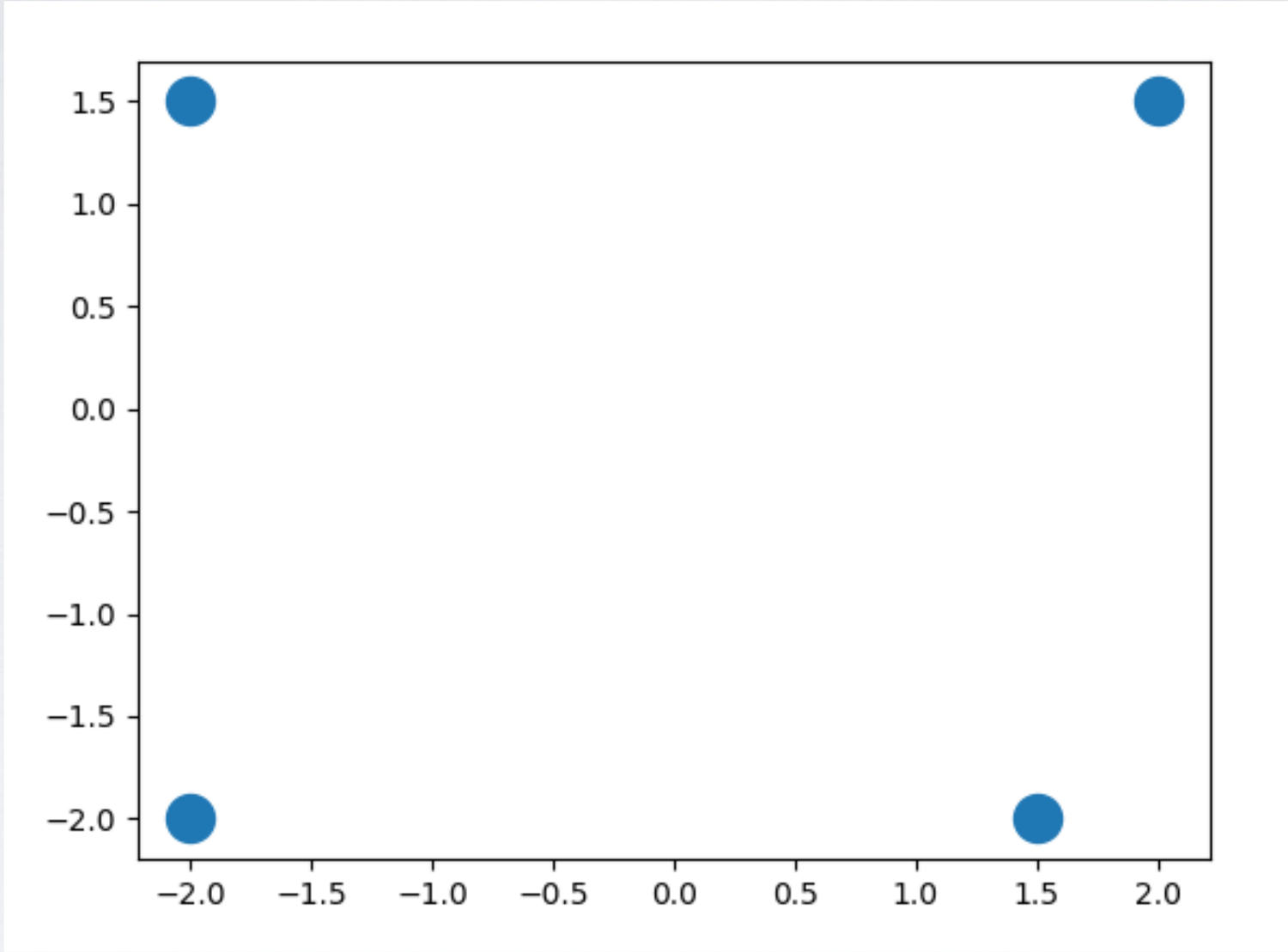

Iteration 5
4
3
2
1
0
−1
−2
−3
−4
−3
−2
−1
0
1
2
3
4


Iteration 6
4
3
2
1
0
−1
−2
−3
−4
−3
−2
−1
0
1
2
3
4


Iteration 7
4
3
2
1
0
−1
−2
−3
−4
−3
−2
−1
0
1
2
3
4


Iteration 8
4
3
2
1
0
−1
−2
−3
−4
−3
−2
−1
0
1
2
3
4

```
PS D:\Programing\machine learning> python .\k-means\k-means.py
2017-11-16 15:14:28.842545: I C:\tf_jenkins\home\workspace\rel-win\M\windows-gpu\PY\36\tensorflow\core\platform\cpu_feature_guard.cc:137] Your CPU supports instructions tha
t this TensorFlow binary was not compiled to use: AVX AVX2
2017-11-16 15:14:29.650957: I C:\tf_jenkins\home\workspace\rel-win\M\windows-gpu\PY\36\tensorflow\core\common_runtime\gpu\gpu_device.cc:1030] Found device 0 with properties
:
name: GeForce GTX 950M major: 5 minor: 0 memoryClockRate(GHz): 1.124
pciBusID: 0000:01:00.0
totalMemory: 2.00GiB freeMemory: 1.65GiB
2017-11-16 15:14:29.651066: I C:\tf_jenkins\home\workspace\rel-win\M\windows-gpu\PY\36\tensorflow\core\common_runtime\gpu\gpu_device.cc:1120] Creating TensorFlow device (/d
evice:GPU:0) -> (device: 0, name: GeForce GTX 950M, pci bus id: 0000:01:00.0, compute capability: 5.0)
WARNING:tensorflow:From C:\Users\User\AppData\Local\Programs\Python\Python36\lib\site-packages\tensorflow\python\util\tf_should_use.py:107: initialize_all_variables (from t
ensorflow.python.ops.variables) is deprecated and will be removed after 2017-03-02.
Instructions for updating:
Use `tf.global_variables_initializer` instead.
Found in 20.09 seconds 8 iterations
Centroids:
[[ 1.65289262 -2.04643427]
 [-2.0763623   1.61204964]
 [-2.08862822 -2.07255306]
 [ 2.09831502  1.55936014]]
Cluster assignment [2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 3 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3]
PS D:\Programing\machine learning>
```


1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

Source Code

Import all we need

```
import tensorflow as tf
import numpy as np
import time

#help us to graph
import matplotlib
import matplotlib.pyplot as plt

#import datasets we need by scikit-learn
from sklearn.datasets.samples_generator import make_blobs
from sklearn.datasets.samples_generator import make_circles
```

Set up Data set

#set up data type , here i choose blobs to make it simpler
DATA_TYPE = "blobs"

#Set up Number of clusters in train data , if we choose circle,2 is enough
K = 4
if(DATA_TYPE == "circle"):
K = 2
else:
K = 4

Set up Data set

Using ski-learn
for blobs:
n_samples : number of data , which means we have 200 points
centers = [(-2, -2), (-2, 1.5), (1.5, -2), (2, 1.5)]
centers = centers
n_features = dimmension , here we choose plane so = 2
cluster_std = std
shuffle : if we mix up samples,here I choose false
random_state : random seed
for circles:
noise : random noise data set up to the sample set
factor : the ratio factor between circle data set

Conclusion

Using unsupervised learning takes advantage of Clustering.
As the result , the common way in implenmentation is Clustering first before supervised learnig which called “Semi-supervised learning”
In this case , I choose an easy way to generate a data, and imply unsupervised learning which can help me to Label the data first , considered a pre-handling stage
(different color represent different Lebel)