



Os parâmetros do Affinity Propagation

☰ Ciclo	Ciclo 08: Outros algoritmos Clusterização
# Aula	69
🕒 Created	@June 27, 2023 2:03 PM
☑ Done	<input type="checkbox"/>
☑ Ready	<input checked="" type="checkbox"/>

Objetivo da Aula:

- ☐ Parâmetros Affinity Propagation
- ☐ Affinity Propagation fine-tuning
- ☐ Próxima aula

Conteúdo:

▼ 1. Parâmetros Affinity Propagation

▼ max_iter:

Número máximo de iterações. O valor padrão de fábrica é 200

▼ convergence_iter:

Número de iterações que interrompe o treinamento, quando não houver mais mudanças no número estimado de cluster

▼ copy:

Faz uma cópia dos dados de entrada

▼ preference:

O parâmetro "preference" controla a prioridade ou preferência que os pontos têm em se tornar centros de cluster. Valores

▼ damping:

O fator de amortecimento no intervalo `[0, 5, 1, 0]` controla a taxa de atualização das responsabilidades e disponibilidade

▼ affinity:

Afinidade é a métrica de similaridade entre os pontos do conjunto de dados. No momento, são suportados 'precompute

▼ random_state:

Gerador de números pseudo-aleatórios para controlar o estado inicial. Use um int para resultados reproduzíveis em cha

▼ 2. Affinity Propagation fine-tuning

```
# libraries
import time
import numpy as np

from IPython import display
from sklearn import datasets as dt
from sklearn import metrics as m
from sklearn import cluster as c

from matplotlib import pyplot as plt

# dataset
X, clusters = dt.make_blobs( n_samples=300, centers=4, cluster_std=0.60,
                             random_state=0 )
```

```

# show figure
plt.scatter( X[:,0], X[:,1], alpha=0.7, edgecolors='b' )

# testing parameter
p = np.arange( -1, -61, -1 )
ss_list = []

fig = plt.figure()
for i in range( len( p ) ):
    # model definition
    model = c.AffinityPropagation( preference=p[i] )

    # model training
    model.fit( X )

    # clustering
    labels = model.predict( X )

    # plot
    plt.scatter( X[:,0], X[:,1], c=labels, cmap='rainbow', alpha=0.7,
                edgecolors='b' )

display.clear_output( wait=True )
plt.show()
plt.gcf()
time.sleep( 0.05 )

```

▼ 3. Próxima aula

Convergencia do Affinity Propagation