Knowledge Discovery and Data Mining

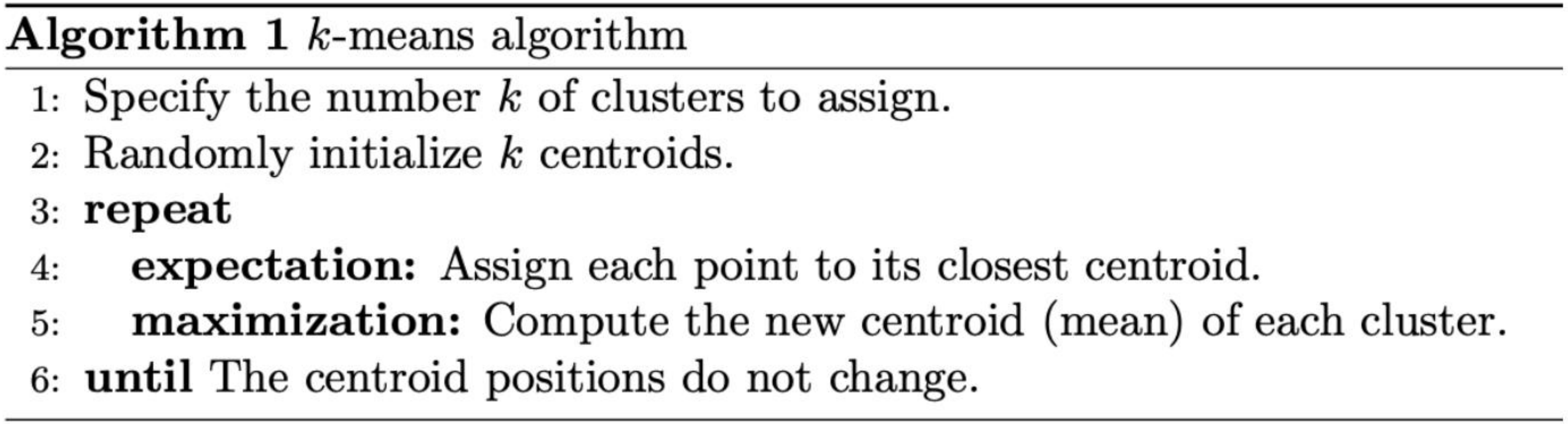
Lab 8 K-means

Xuan Song
Songx@sustech.edu.cn

Topics

Implement K-means with scikit-learn

K-means Algorithm



Algorithm 1 k-means algorithm

```
Specify the number $k$ of clusters to assign.
Randomly initialize $k$ centroids.
repeat
    expectation: Assign each point to its closest centroid.
    maximization: Compute the new centroid (mean) of each cluster.
until The centroid positions do not change.
```

Implementing K-means with Scikit-Learn

- sklearn.cluster.KMeans

class sklearn.cluster.KMeans(**n_clusters**=8, *, init='k-means++', n_init=10, **max_iter**=300, tol=0.0001, precompute_distances='deprecated’ , verbose=0, random_state=None, copy_x=True, n_jobs='deprecated', algorithm='auto')

- sklearn.cluster.MiniBatchKMeans

class sklearn.cluster.MiniBatchKMeans(**n_clusters**=8, *, init='k-means++', **max_iter**=100, **batch_size**=100, verbose=0, compute_labels=True, random_state=None, tol=0.0, max_no_improvement=10, init_size=None, n_init=3, reassignment_ratio=0.01)

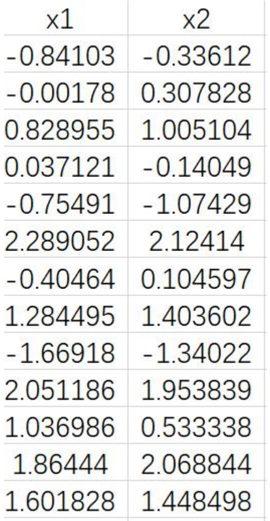
https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html#sklearn.cluster.KMeans
https://scikit-learn.org/stable/modules/generated/sklearn.cluster.MiniBatchKMeans.html#sklearn.cluster.MiniBatchKMeans

Implementing K-means with Scikit-Learn

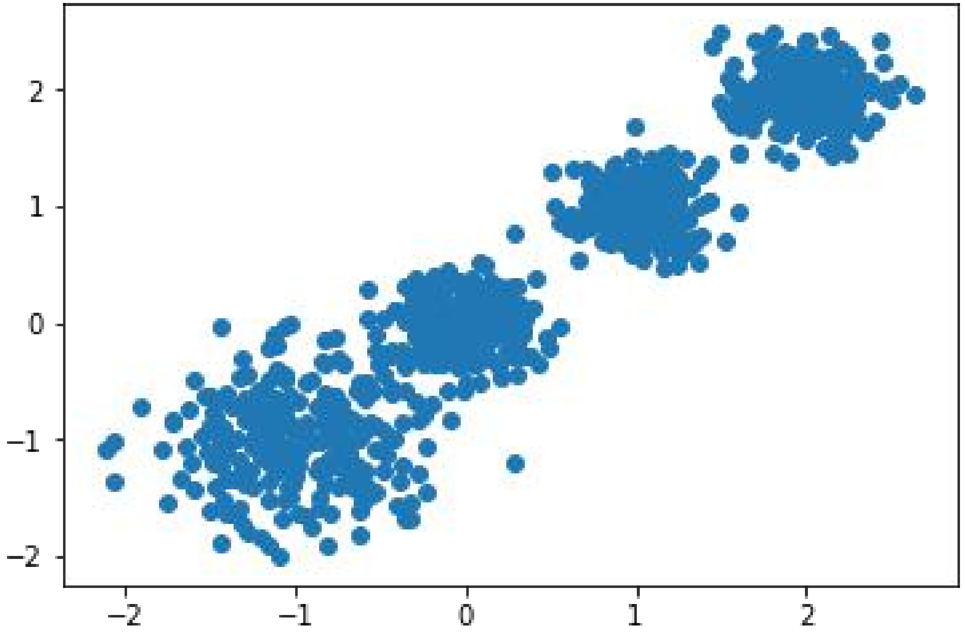
- Sample data: cluster1.csv

cluster1.csv

Attribute



x1	x2
-0.84103	-0.33612
-0.00178	0.307828
0.828955	1.005104
0.037121	-0.14049
-0.75491	-1.07429
2.289052	2.12414
-0.40464	0.104597
1.284495	1.403602
-1.66918	-1.34022
2.051186	1.953839
1.036986	0.533338
1.86444	2.068844
1.601828	1.448498



Implementing K-means with Scikit-Learn

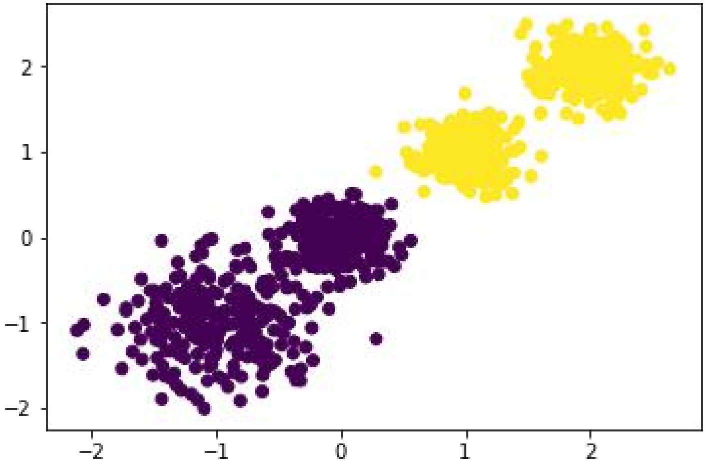
1. Load data from csv files.
2. Data cleaning.
3. Get features.
4. Build a K-means model with scikit-learn.

```
from sklearn.cluster import KMeans
kmeans= KMeans(n_clusters=2, random_state=9)
```

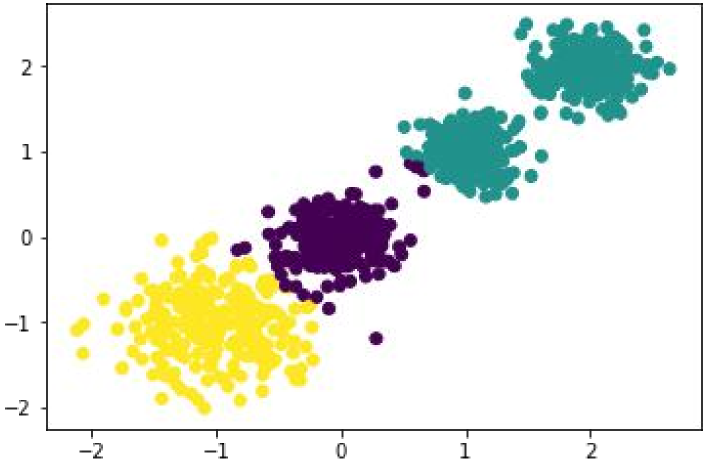
Implementing K-means with Scikit-Learn

5. Visualize the cluster result.

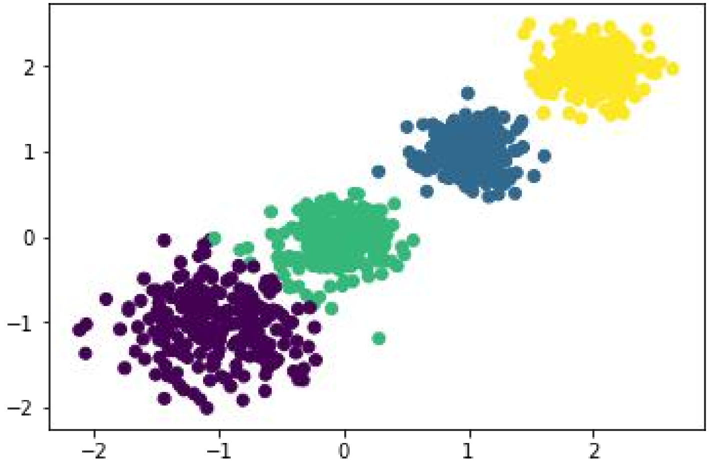
```
y_pred = kmeans.fit_predict(X)
plt.scatter(X[:, 0], X[:, 1], c=y_pred)
plt.show()
```



K= 2



K= 3



K= 4

Implementing K-means with Scikit-Learn

6. Evaluate the clustering

```
from sklearn import metrics
metrics.calinski_harabasz_score(X, y_pred)
```

```
k= 4, Calinski-Harabasz_score  = 5924.050613464895
k= 2, Calinski-Harabasz_score  = 3116.170676416667
k= 3, Calinski-Harabasz_score  = 2931.6250302645562
```

Task1:

- Implementing K-means based on the given dataset.

cluster_task1.csv

Extra

- Implement the k-means algorithm with python, and any clustering library is prohibited.

SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

南方科技大学

End of Lab8