k-means clustering

- supervised learning
- unsupervised learning
- hidden structure
- data groups

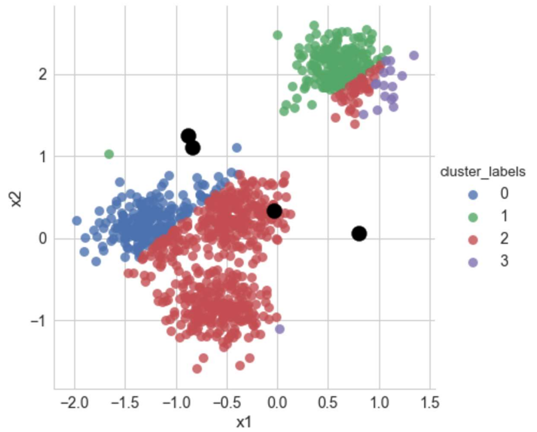
k-means clustering

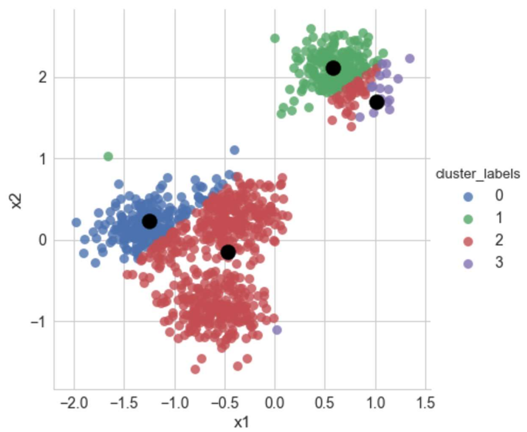
$$\mu_1^{(0)}, \ldots, \mu_k^{(0)} \in \mathbb{R}^n$$

$$c_i = \underset{j}{\operatorname{argmax}} \, \|x^{(i)} - \mu_j^{(s)}\|$$

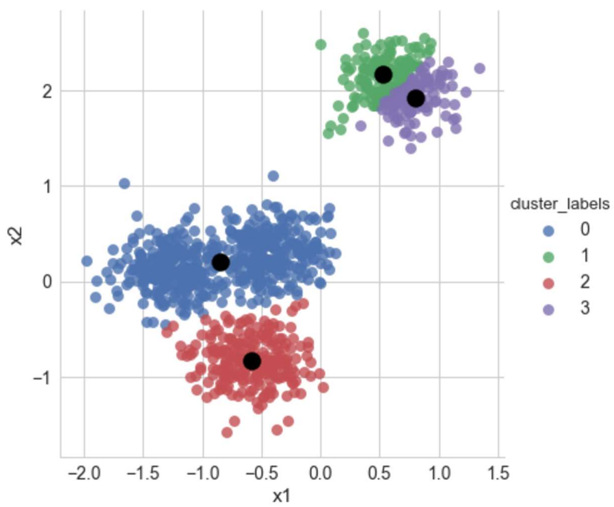
$$\mu_j^{(s+1)} = \frac{\sum_{i=1}^{n} I[c_i = j] x^{(i)}}{\sum_{i=1}^{n} I[c_i = j]}$$

k-means clustering

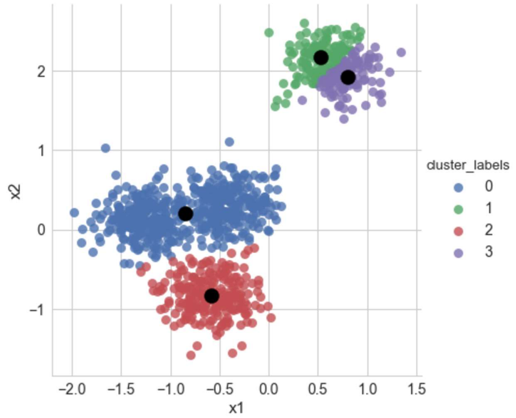




k-means clustering

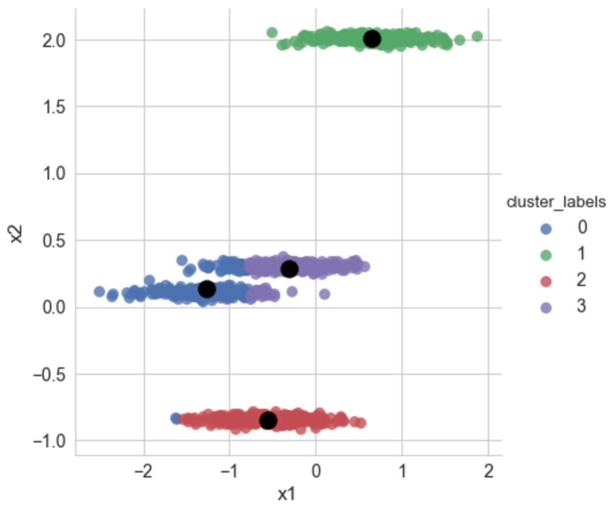


k-means++ clustering



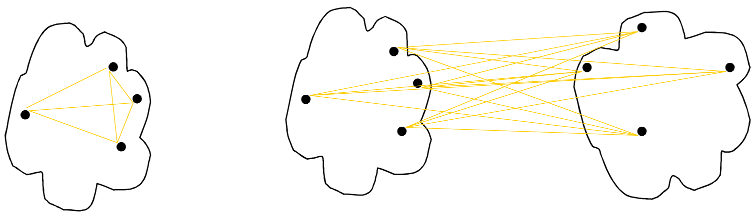
- the first centroid is chosen uniformly at random from the data points that are being clustered
- each subsequent centroid is chosen from the remaining data points with probability proportional to its squared distance from the point's closest existing centroid
- better spread of the initial centroids

k-means++ clustering



k-means++ clustering: finding k

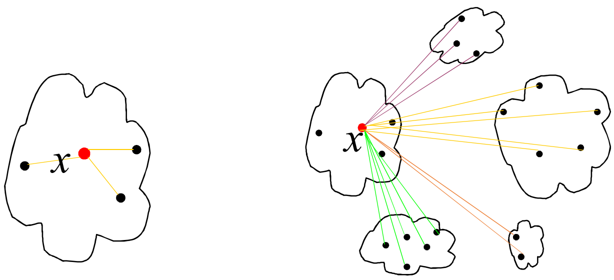
- **Cohesion**: measures how closely related are objects in a cluster
- **Separation**: measure how well-separated a cluster is from other clusters



cohesion separation

k-means++ clustering: finding k

- **Cohesion a(x)**: the mean distance between the data point and all other points in the same cluster
- **Separation b(x)**: the mean distance between the data point and all other points in the next nearest cluster



cohesion separation

k-means++ clustering: finding k

- **Silhouette score s(x)**:

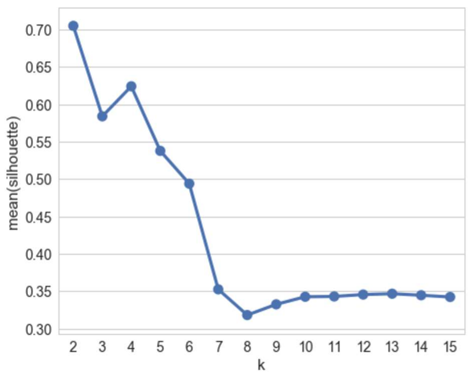
$$s(x) = \frac{b(x) - a(x)}{max(a(x), b(x))}$$

- **Silhouette coefficient *SC***:

$$SC = \frac{1}{n} \sum_{i=1}^{n} s(x)$$

- **Intertia**: the sum of squared distance for each point to it's closest cluster centroid, i.e. its assigned cluster

k-means++ clustering: finding k



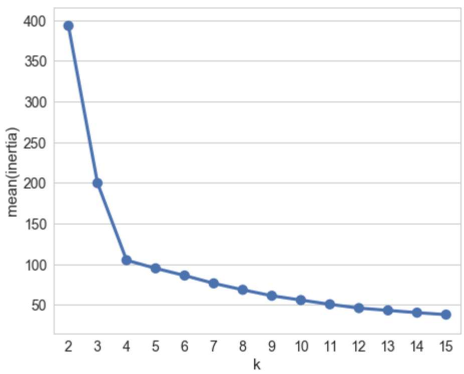


Image segmentation:

- Each image is represented in the *RGB* color space.
- An image pixel is represented as a 3D color vector
- Pixels are clustered to find the segments.

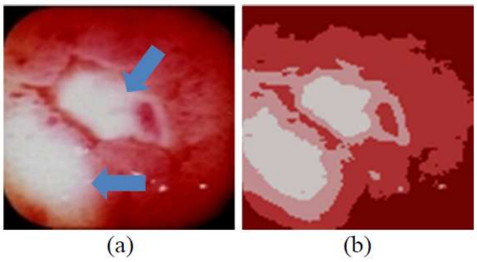


FIGURE 1. Results of segmentation on an image diagnosed with duodenal ulcer: (a) Original image; (b) Image segmented with 4 regions.

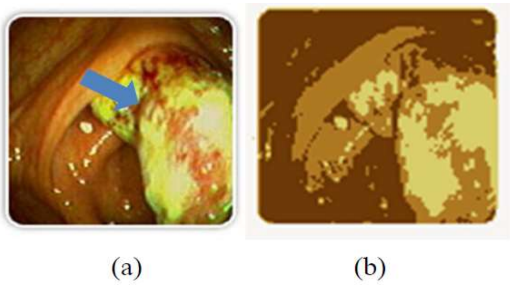
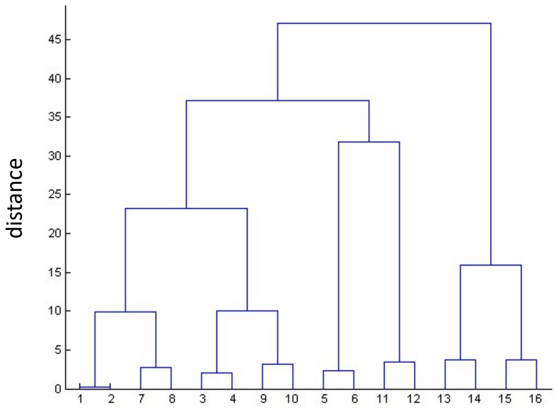


FIGURE 3. Results of segmentation on an image diagnosed with colon cancer: (a) Original image; (b) Image segmented with 4 regions.

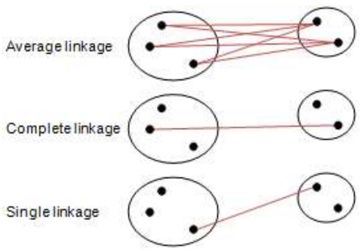
```python
from sklearn import metrics

X = dataset_blops[['x1','x2']].values

clf = KMeans(init='k-means++')

s_sil = []
s_in = []
for k in range(2,16):
    clf.n_clusters = k
    clf.fit(X)
    labels = clf.labels_
    s_sil.append(metrics.silhouette_score(X, labels))
    s_in.append(clf.inertia_)

cluster_evaluation = pd.DataFrame()
cluster_evaluation['k'] = range(2,16)
cluster_evaluation['silhouette'] = s_sil
cluster_evaluation['inertia'] = s_in

plt.figure(figsize=(16,6))
plt.subplot(1,2,1)
sns.pointplot(x="k", y="silhouette", data=cluster_evaluation)
plt.subplot(1,2,2)
sns.pointplot(x="k", y="inertia", data=cluster_evaluation)
plt.show()
```

hierarchical clustering



- agglomerative clustering
- no need to set *k* in advance
- start with a singleton cluster
- clusters are iteratively merged until one single cluster remains
- cluster tree or dendogram
- works on distance matrix

hierarchical clustering



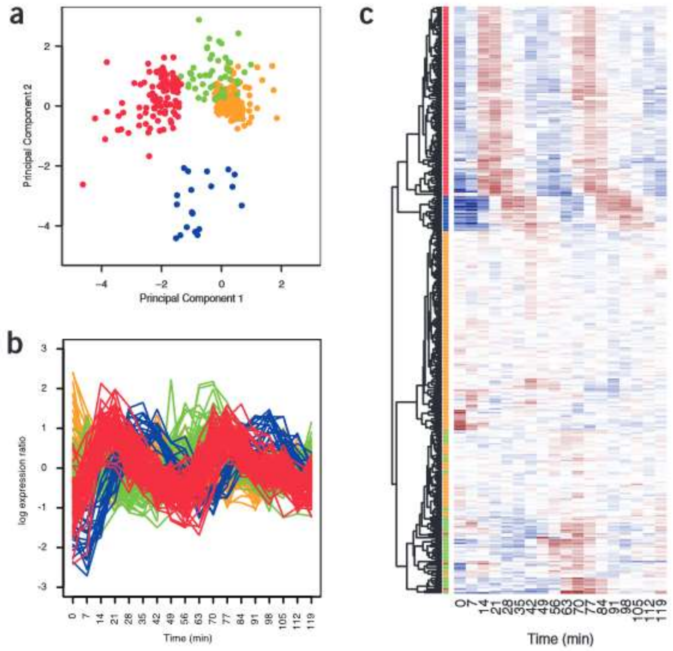
$$d(A,B) = \frac{1}{|A||B|} \sum_{A_i \in A} \sum_{B_j \in B} dist(A_i, B_j)$$

$$d(A,B) = \max(dist(A_i, B_j))$$

$$d(A,B) = \min(dist(A_i, B_j))$$

1. represent each data point as a singleton cluster
2. merge the two closest clusters
3. repeat step 2. until one single cluster remains

*"**Visualization of gene expression profiles.***
Expression of 320 transcripts from S. cerevisiae, collected over 18 time points throughout the cell cycle 80. Colors indicate cluster membership based on a k-means clustering (k= 4)"



Gehlenborg N. *et al.* (2010) Visualization of omics data for systems biology. *Nat Methods* 7: S56–68

```
cg = sns.clustermap(dataset_hc,method="complete",figsize=(6,10))
plt.setp(cg.ax_heatmap.yaxis.get_majorticklabels(), rotation=0)
plt.show()
```