# MACHINE LEARNING AND PATTERN RECOGNITION

# IMAGE AND VIDEO UNDERSTAND

## K-means & Spectral Clustering

## Image Segmentation

Dao Quang Hoan 871510

# Outline

1. K-means
2. Spectral Clustering
3. Elbow method
4. Apply to datasets
5. Image segmentation
6. F1 score & Conditional Entropy

## K-means clustering

Step 1: Specify number of clusters *K*

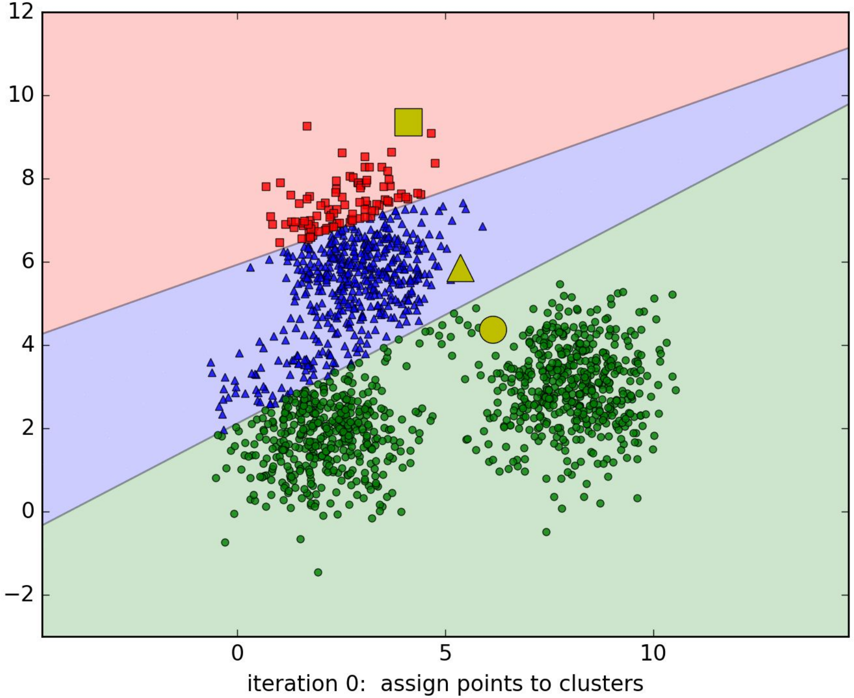
Step 2: Assign data points to the closest centroid

Step 3: Change the cluster centroid to the average of the assigned points

Step 4: Repeat step 2 and 3 until there is no change to the centroids

$$\mathrm{y}_i = \arg\min_{\mathrm{y}_i} \sum_{j=1}^{K} y_{ij} \|\mathrm{x}_i - \mathrm{m}_j\|_2^2$$

$$\text{subject to: } y_{ij} \in \{0,1\} \ \forall j; \quad \sum_{j=1}^{K} y_{ij} = 1$$



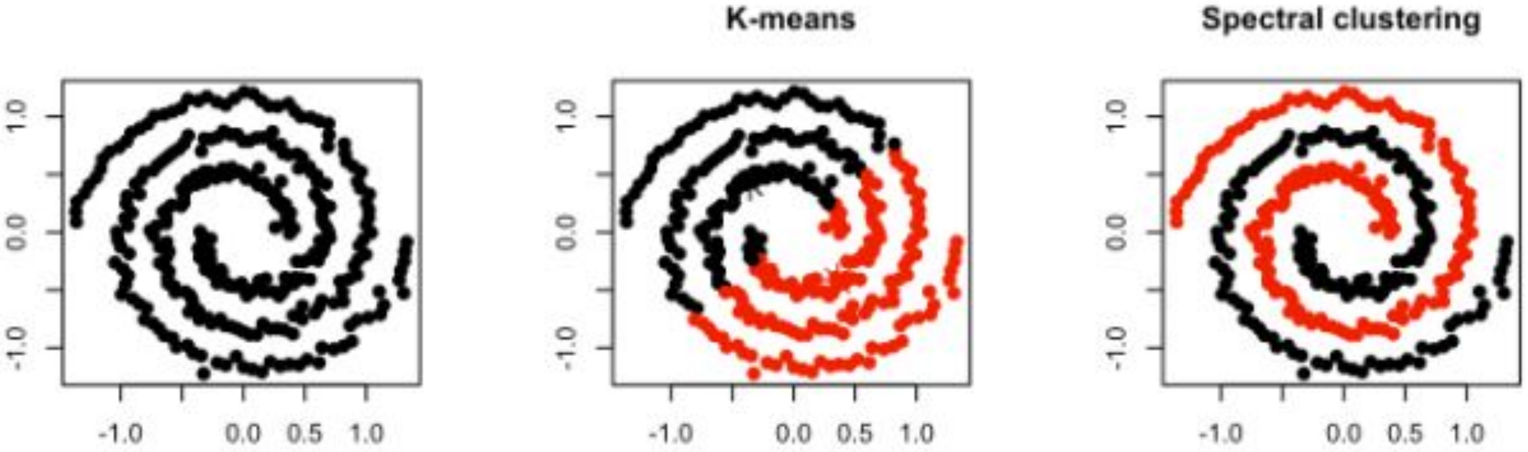
## Spectral clustering

Step 1: Construct a similarity graph (KNN graph)

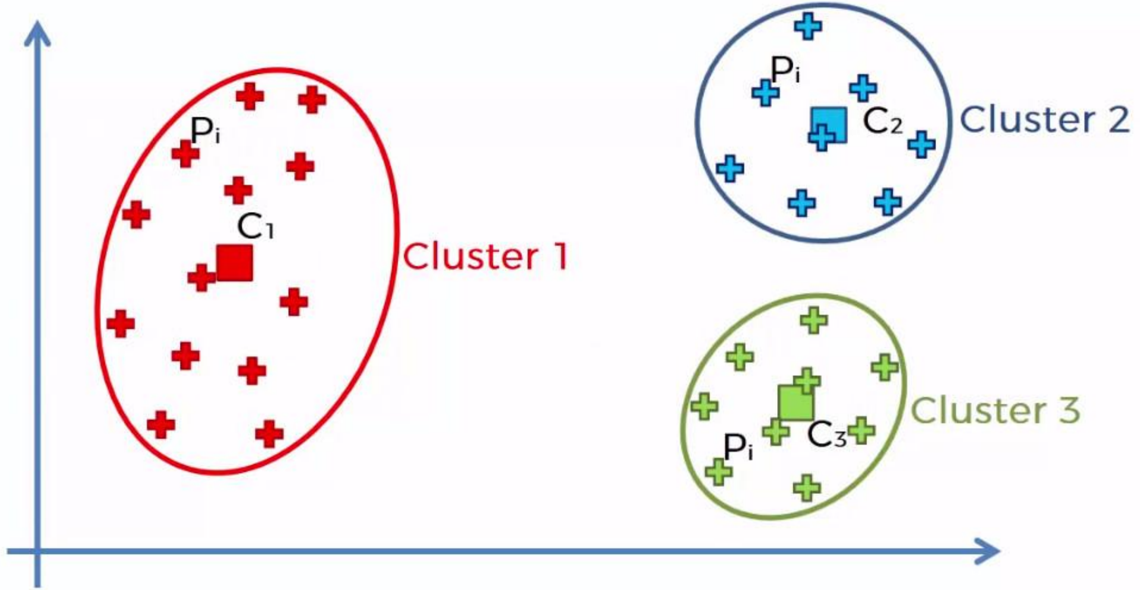
Step 2: Embed the data points in low dimensional space in which the clusters are

Step 3: Use the lowest eigenvalue in order to choose the eigenvector for clusters

The difference between the 2 can easily be shown by this illustration:

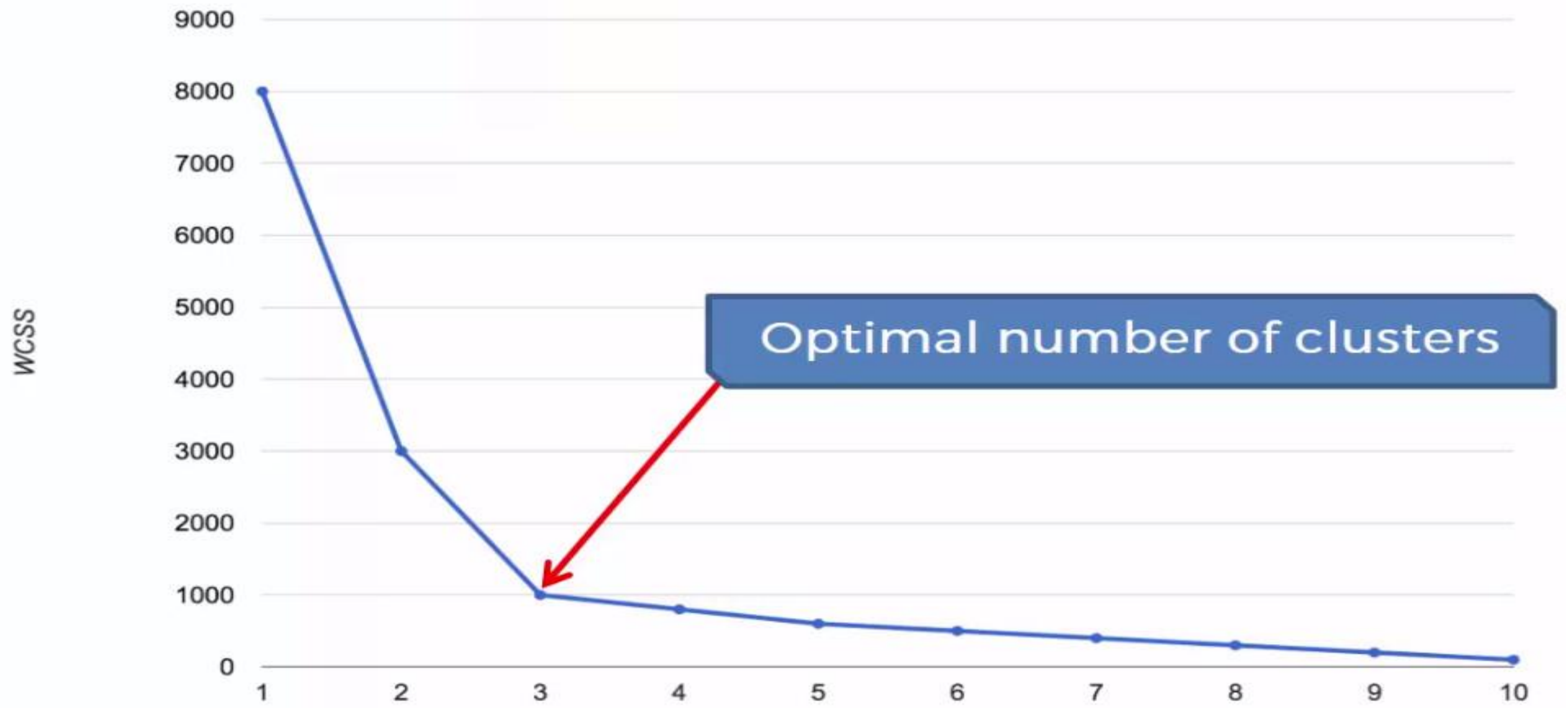


# Elbow method




$$\text{WCSS} = \sum_{\text{P}_\text{i} \text{ in Cluster 1}} \text{distance}(\text{P}_\text{i}, \text{C}_1)^2 + \sum_{\text{P}_\text{i} \text{ in Cluster 2}} \text{distance}(\text{P}_\text{i}, \text{C}_2)^2 + \sum_{\text{P}_\text{i} \text{ in Cluster 3}} \text{distance}(\text{P}_\text{i}, \text{C}_3)^2$$

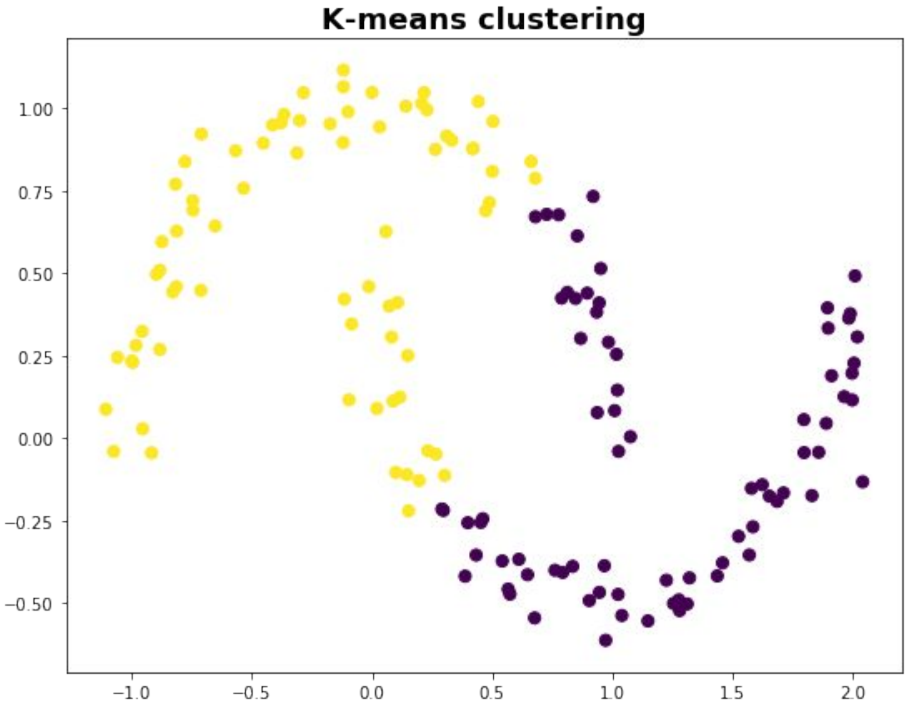
# The Elbow Method

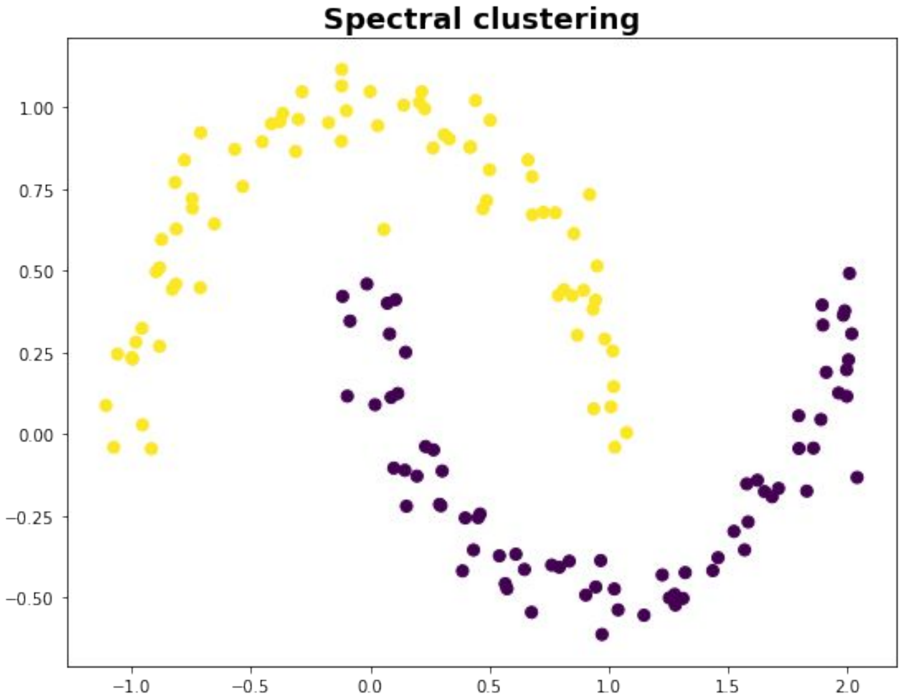


# Datasets

| Dataset | Num of instances | Num of attributes | Num of clusters |
|---|---|---|---|
| Moons | 150 | 2 | 2 |
| Iris | 150 | 4 | 3 |
| Seeds | 210 | 7 | 3 |

# Moons dataset



# Iris dataset

1. sepal length in cm

2. sepal width in cm

3. petal length in cm

4. petal width in cm

5. Class:

-- Iris Setosa

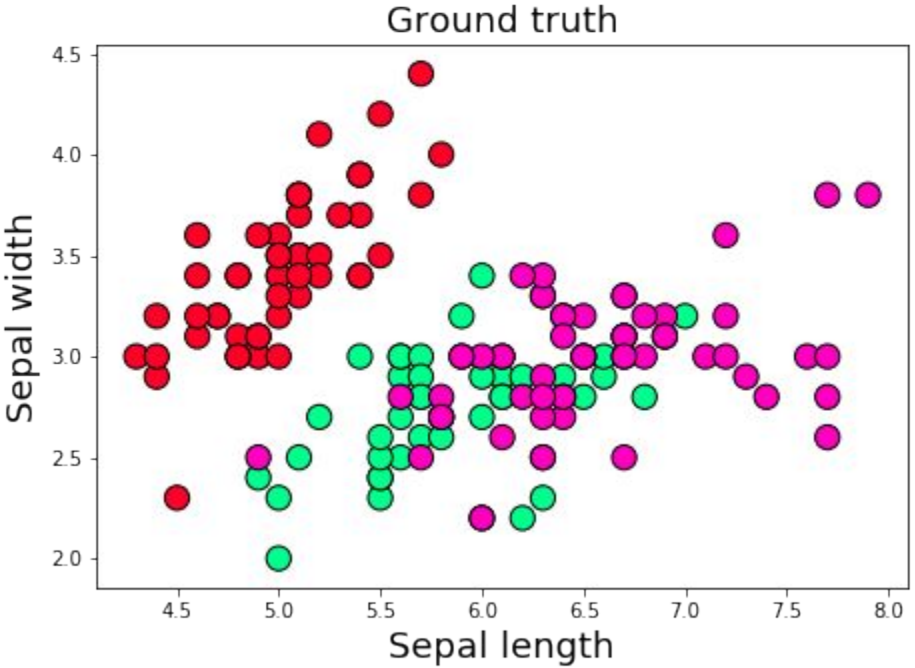
-- Iris Versicolour

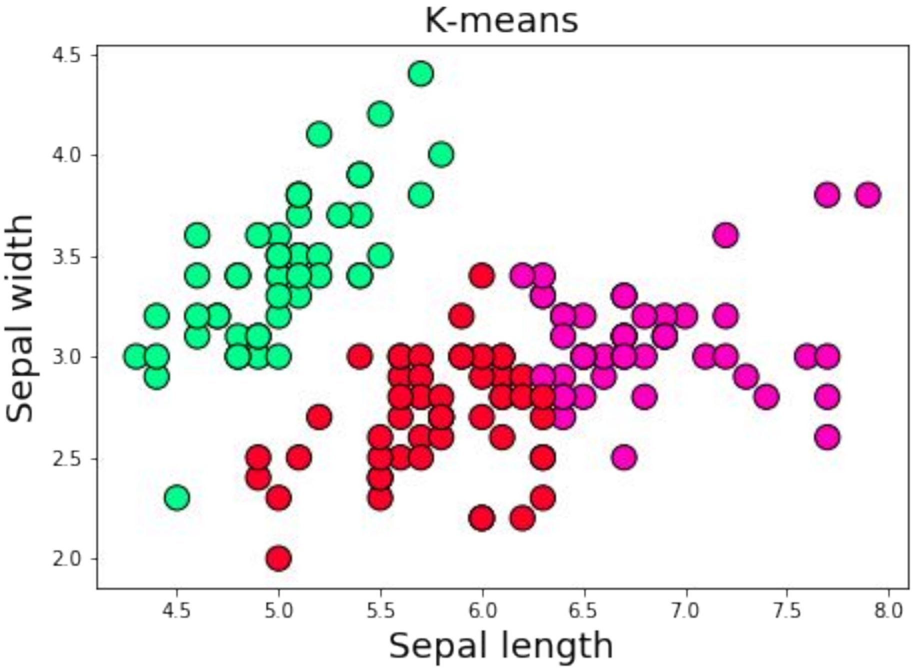
-- Iris Virginica

```
dataset.head()
```

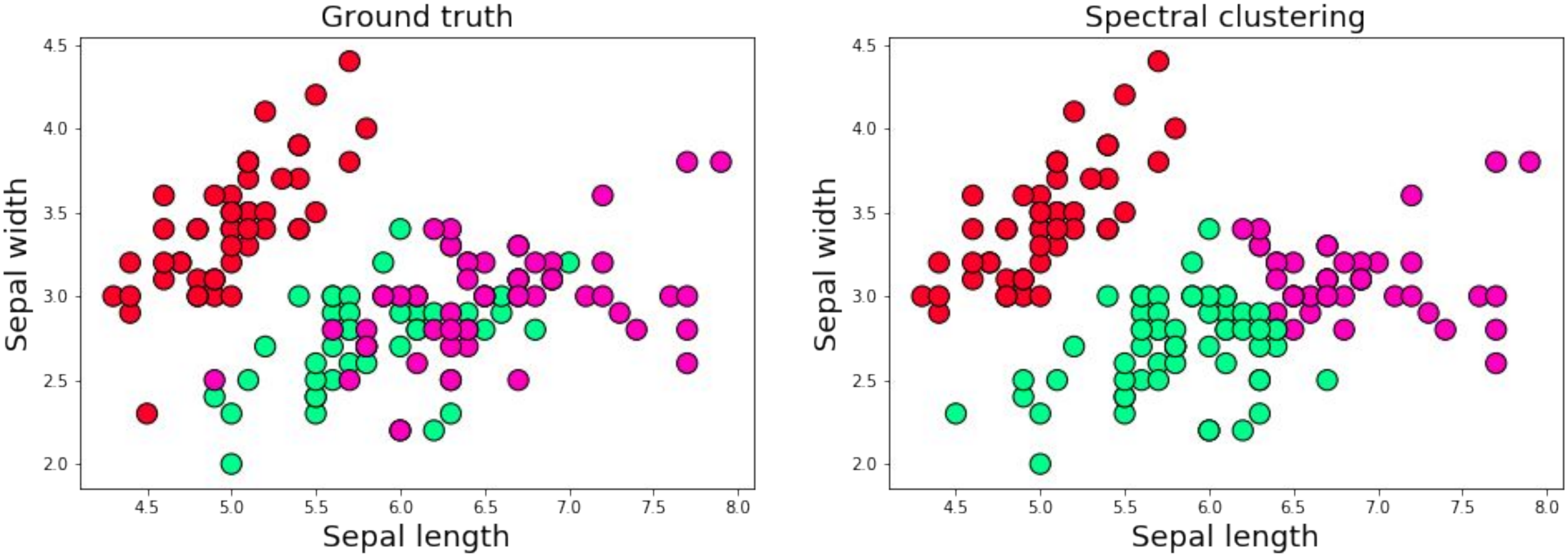
| | Id | SepalLengthCm | SepalWidthCm | PetalLengthCm | PetalWidthCm | Species |
|---|---|---|---|---|---|---|
| **0** | 1 | 5.1 | 3.5 | 1.4 | 0.2 | Iris-setosa |
| **1** | 2 | 4.9 | 3.0 | 1.4 | 0.2 | Iris-setosa |
| **2** | 3 | 4.7 | 3.2 | 1.3 | 0.2 | Iris-setosa |
| **3** | 4 | 4.6 | 3.1 | 1.5 | 0.2 | Iris-setosa |
| **4** | 5 | 5.0 | 3.6 | 1.4 | 0.2 | Iris-setosa |

# True label vs k-means

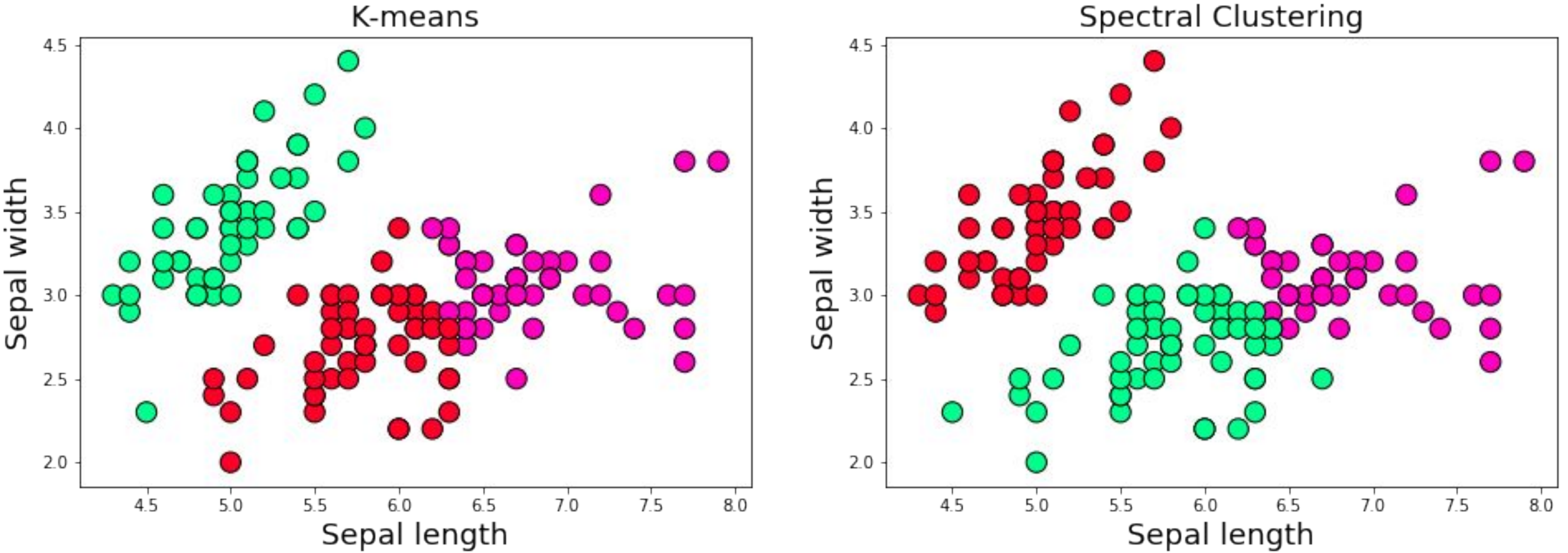


# True label vs Spectral clustering



# K-means vs Spectral clustering



# Seeds dataset

0. area A,

1. perimeter P,

2. compactness C = 4*pi*A/P^2,

3. length of kernel,

4. width of kernel,

5. asymmetry coefficient

6. lngth of kernel groove.

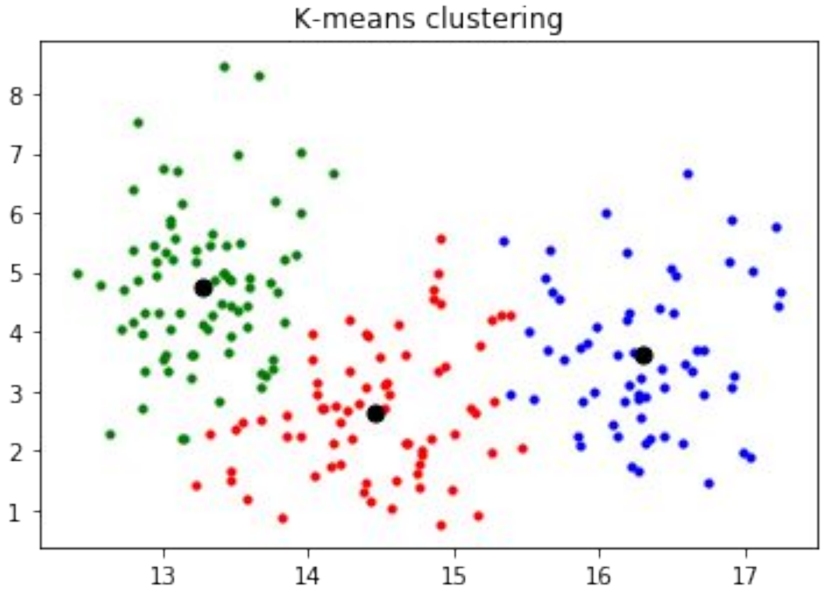
7. Labels

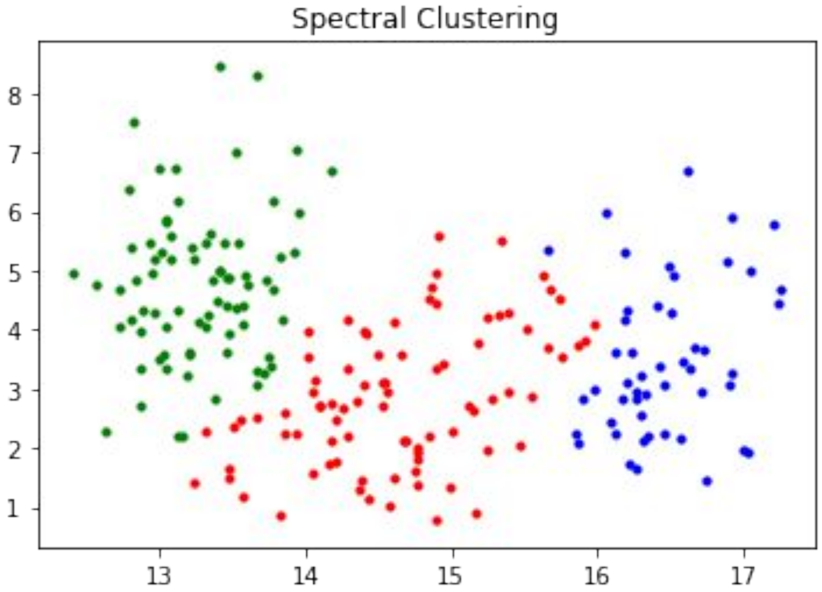
All of these parameters were real-valued continuous.

```
data.head()
```

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| **0** | 15.26 | 14.84 | 0.8710 | 5.763 | 3.312 | 2.221 | 5.220 | 1 |
| **1** | 14.88 | 14.57 | 0.8811 | 5.554 | 3.333 | 1.018 | 4.956 | 1 |
| **2** | 14.29 | 14.09 | 0.9050 | 5.291 | 3.337 | 2.699 | 4.825 | 1 |
| **3** | 13.84 | 13.94 | 0.8955 | 5.324 | 3.379 | 2.259 | 4.805 | 1 |
| **4** | 16.14 | 14.99 | 0.9034 | 5.658 | 3.562 | 1.355 | 5.175 | 1 |

# Seeds dataset



# K-means vs Spectral Clustering

**K-Means**:

- Efficient
- Fast and simple
- Need to pick K
- Sensitive to outliers
- Only finds “spherical” clusters

**Spectral clustering:**

- Excellent quality under many different data forms
- Much slower than KMeans

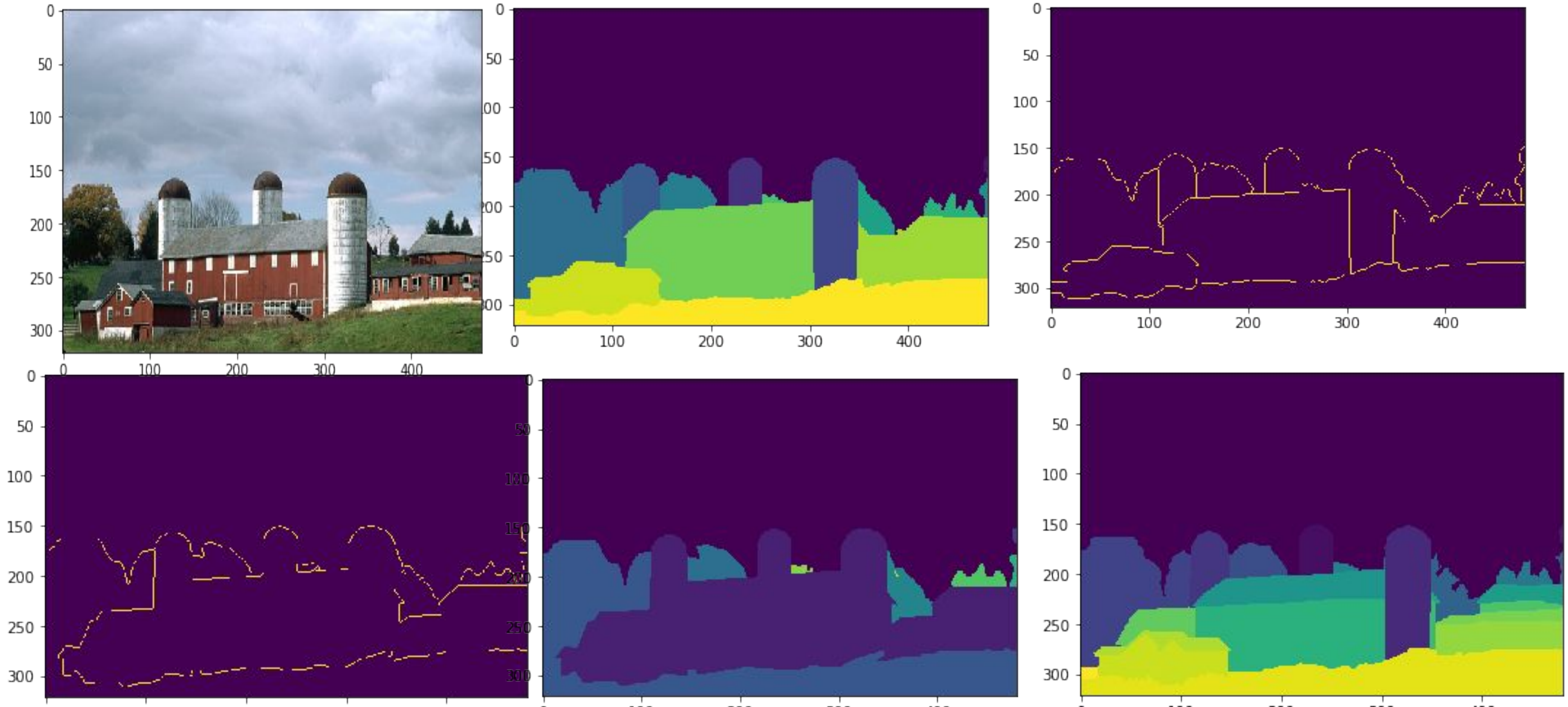
# Image Segmentation

- We will use Berkeley Segmentation Benchmark:
    - Original images (.jpg) and ground truth (.mat)
    - Each image have 5 ground truth segmentations

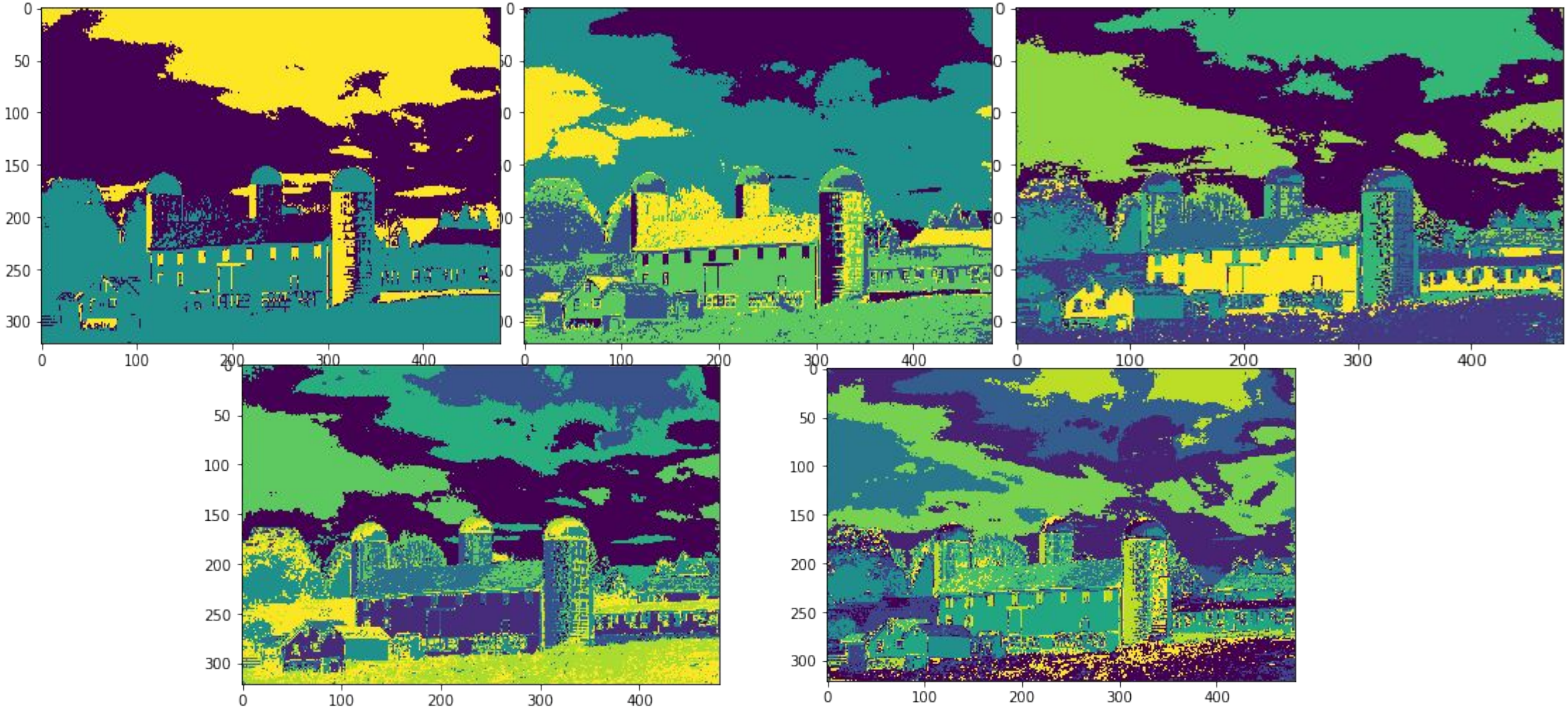
- Apply k-means and spectral clustering algorithms

- The evaluation using F-measures and Conditional Entropy

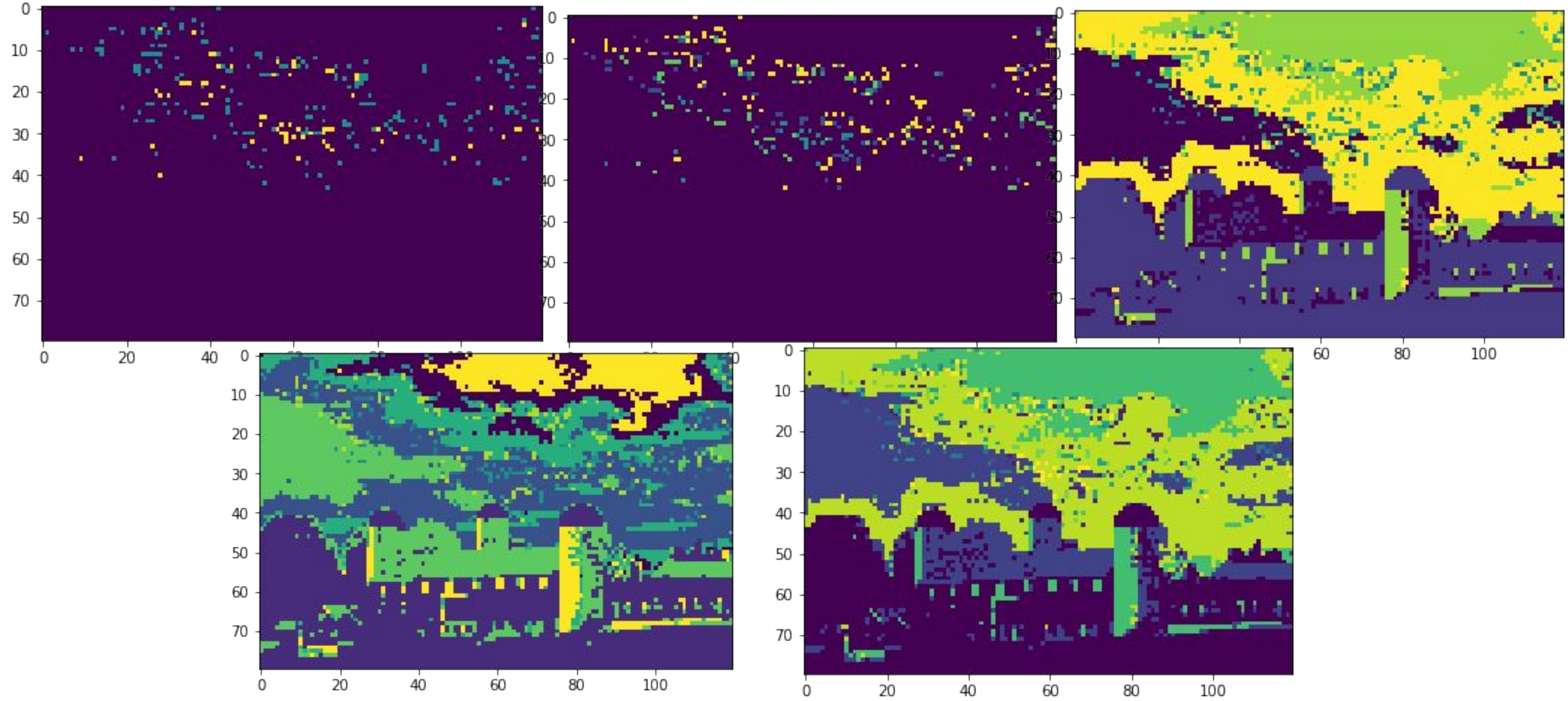
# Original image and GroundTruth



# K-means: k=3,5,7,9,11



# Spectral clustering: k=3,5,7,9,11



# F1 score & Conditional Entropy

| k | k=2 | k=3 | k=4 | k=5 | k=6 | k=7 | k=8 | k=9 | k=10 | k=11 |
|---|---|---|---|---|---|---|---|---|---|---|
| **F1 score** | 0.00045 | 0.02893 | 0.23084 | 0.04445 | 0.03253 | 0.28533 | 0.28359 | 0.26506 | 0.02397 | 0.02072 |
| **Condition Entropy** | 0.20140 | 0.23542 | 0.24351 | 0.26916 | 0.53136 | 0.41086 | 0.46782 | 0.64620 | 0.44827 | 0.69523 |

# Thank you!