CS 446/ECE 449: Machine Learning

Shenlong Wang

University of Illinois at Urbana-Champaign, 2024

K-Means Clustering

Goals of this lecture

Goals of this lecture

- Understanding clustering concept

Goals of this lecture

- Understanding clustering concept
- Getting to know k-Means

Goals of this lecture

- Understanding clustering concept
- Getting to know k-Means

Reading material:

Goals of this lecture

- Understanding clustering concept
- Getting to know k-Means

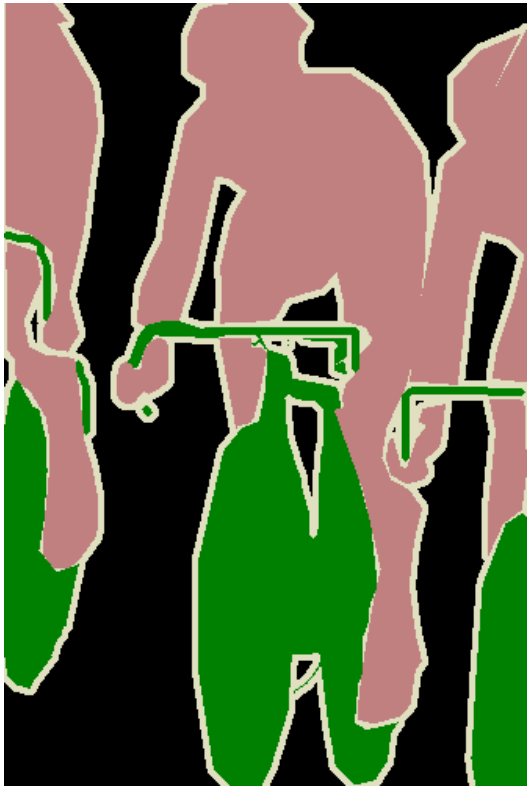
Reading material:

- K. Murphy; Machine Learning: A Probabilistic Perspective; Chapter 11

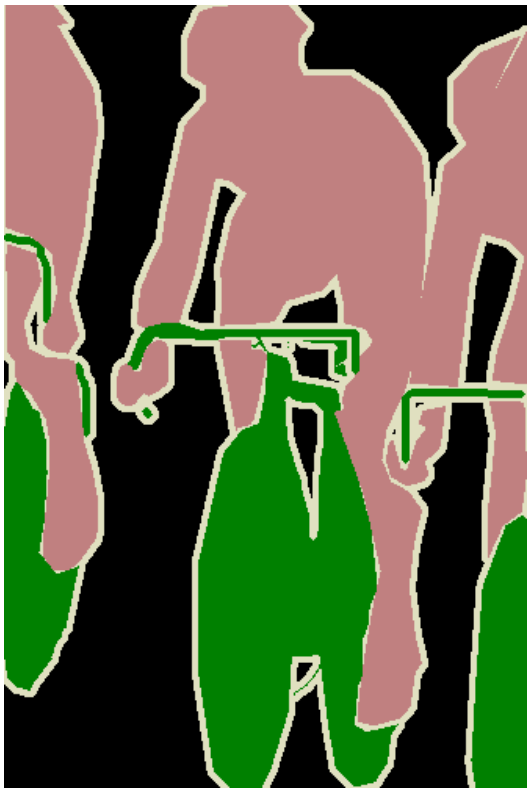
Recap: Semantic Image Segmentation

Recap: Semantic Image Segmentation

Recap: Semantic Image Segmentation



Recap: Semantic Image Segmentation



How did we do it?

What if we don't have labels?

What if we don't have labels?

Image

What if we don't have labels?

Image

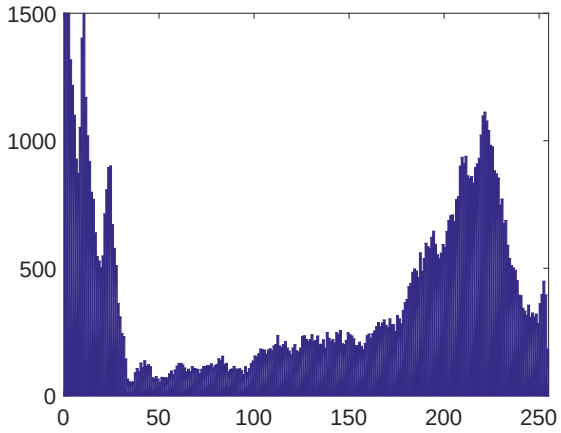
What if we don't have labels?

Image Intensities

What if we don't have labels?

Image

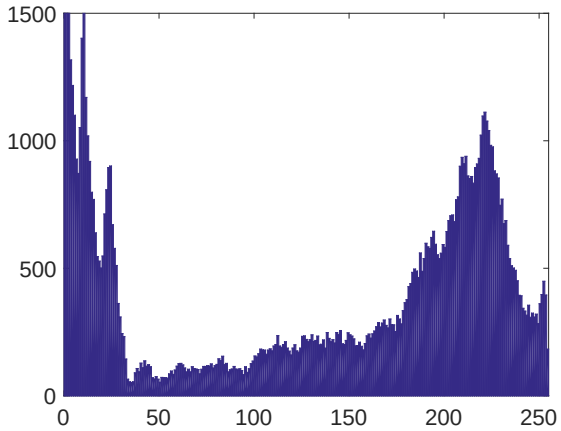
Intensities



What if we don't have labels?

Image

Intensities

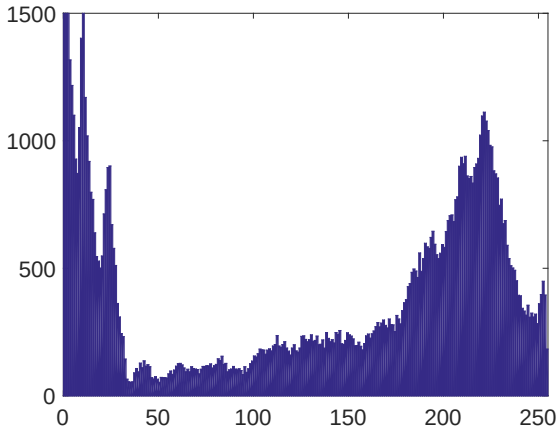


Clustering

What if we don't have labels?

Image

Intensities

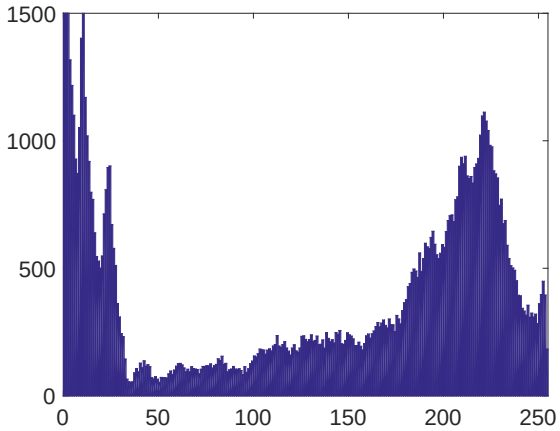


Clustering

What if we don't have labels?

Image

Intensities



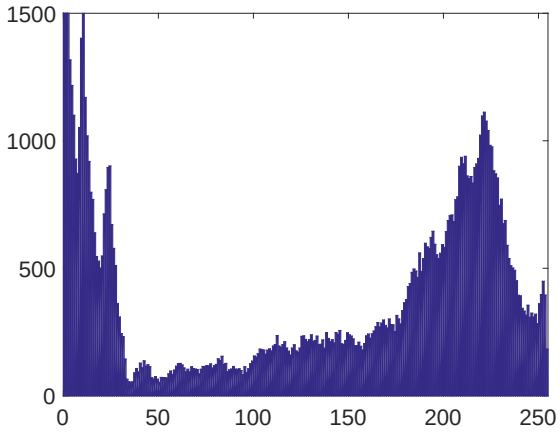
Clustering

Group perceptually or according to another metric similar regions

What if we don't have labels?

Image

Intensities



Clustering

Group perceptually or according to another metric similar regions

How can we do it?

kMeans/Lloyd's Algorithm (informal):

kMeans/Lloyd's Algorithm (informal):

- Initialize: pick K random points as cluster centers μ_k

kMeans/Lloyd's Algorithm (informal):

- Initialize: pick K random points as cluster centers μ_k
- Iterate:

kMeans/Lloyd's Algorithm (informal):

- Initialize: pick K random points as cluster centers μ_k
- Iterate:
  - Assign data points $x^{(i)}$ to closest cluster center according to some metric

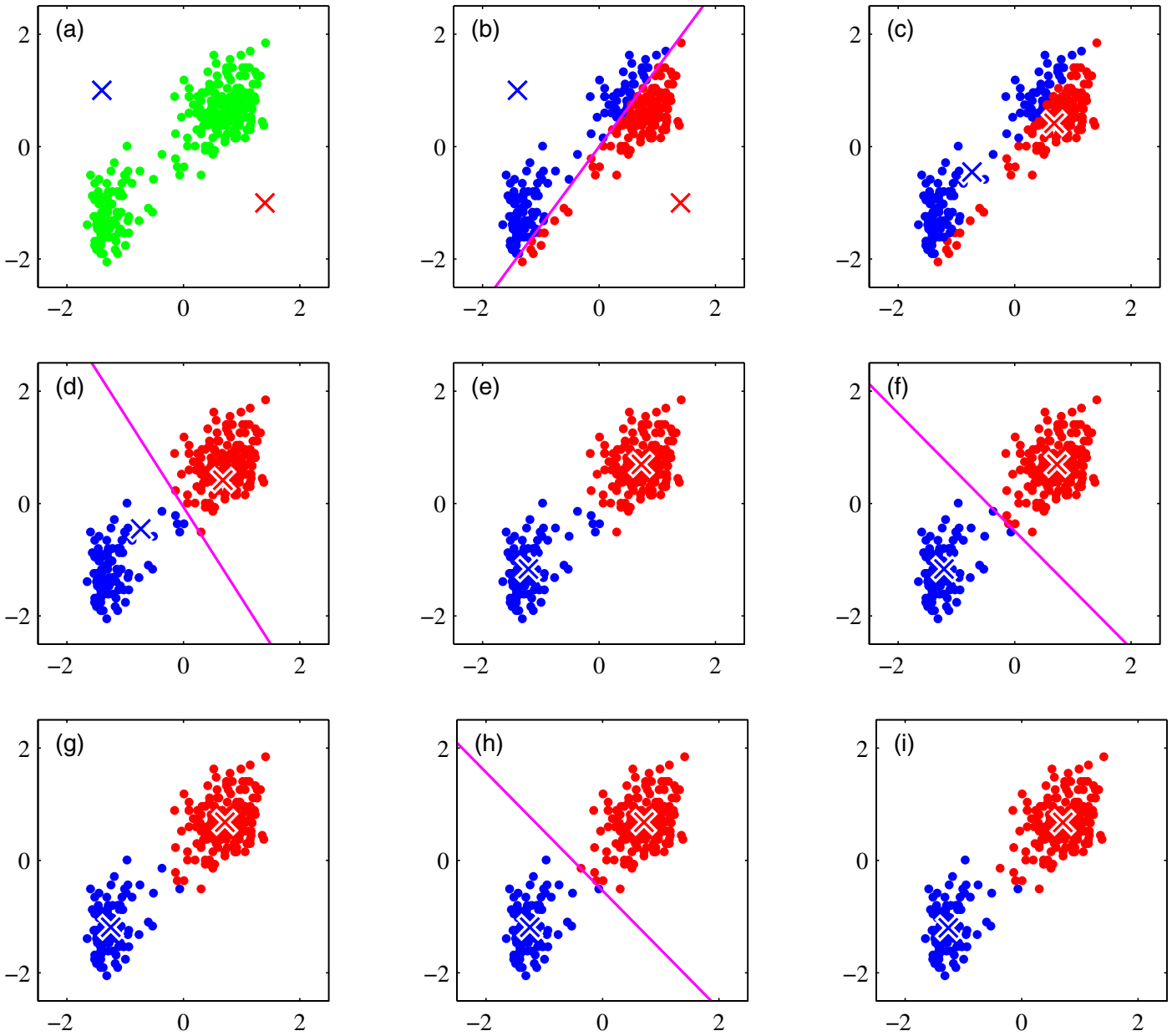
kMeans/Lloyd's Algorithm (informal):

- Initialize: pick K random points as cluster centers μ_k
- Iterate:
  - Assign data points $x^{(i)}$ to closest cluster center according to some metric
  - Update the cluster center to be the average of its assigned points

kMeans/Lloyd's Algorithm (informal):

- Initialize: pick K random points as cluster centers μ_k
- Iterate:
  - Assign data points $x^{(i)}$ to closest cluster center according to some metric
  - Update the cluster center to be the average of its assigned points
  - Stopping criterion: when no points' assignments change

2D Example:



Formal description:

Formal description:

What cost function does kMeans optimize?

Formal description:

What cost function does kMeans optimize? (distortion measure)

Formal description:

What cost function does kMeans optimize? (distortion measure)

$$\min_{\mu} \min_{r} \sum_{i \in \mathcal{D}} \sum_{k=1}^{K} \frac{1}{2} r_{ik} \|x^{(i)} - \mu_k\|_2^2 \quad \text{s.t.} \quad \begin{cases} r_{ik} \in \{0, 1\} & \forall i, k \\ \sum_{k=1}^{K} r_{ik} = 1 & \forall i \end{cases}$$

Formal description:

What cost function does kMeans optimize? (distortion measure)

$$\min_{\mu} \min_{r} \sum_{i \in \mathcal{D}} \sum_{k=1}^{K} \frac{1}{2} r_{ik} \|x^{(i)} - \mu_k\|_2^2 \quad \text{s.t.} \quad \begin{cases} r_{ik} \in \{0, 1\} & \forall i, k \\ \sum_{k=1}^{K} r_{ik} = 1 & \forall i \end{cases}$$

What does the constraint remind us of?

Formal description:

What cost function does kMeans optimize? (distortion measure)

$$\min_{\mu} \min_{r} \sum_{i \in \mathcal{D}} \sum_{k=1}^{K} \frac{1}{2} r_{ik} \|x^{(i)} - \mu_k\|_2^2 \quad \text{s.t.} \quad \begin{cases} r_{ik} \in \{0, 1\} & \forall i, k \\ \sum_{k=1}^{K} r_{ik} = 1 & \forall i \end{cases}$$

What does the constraint remind us of?

How to optimize this cost function?

Cost function:

$$\min_{\mu} \min_{r} \sum_{i \in \mathcal{D}} \sum_{k=1}^{K} \frac{1}{2} r_{ik} \|x^{(i)} - \mu_k\|_2^2 \quad \text{s.t.} \quad \begin{cases} r_{ik} \in \{0, 1\} & \forall i, k \\ \sum_{k=1}^{K} r_{ik} = 1 & \forall i \end{cases}$$

Cost function:

$$\min_{\mu} \min_{r} \sum_{i \in \mathcal{D}} \sum_{k=1}^{K} \frac{1}{2} r_{ik} \|x^{(i)} - \mu_k\|_2^2 \quad \text{s.t.} \quad \begin{cases} r_{ik} \in \{0, 1\} & \forall i, k \\ \sum_{k=1}^{K} r_{ik} = 1 & \forall i \end{cases}$$

Alternate optimization:

Cost function:

$$\min_{\mu} \min_{r} \sum_{i \in \mathcal{D}} \sum_{k=1}^{K} \frac{1}{2} r_{ik} \|x^{(i)} - \mu_k\|_2^2 \quad \text{s.t.} \quad \begin{cases} r_{ik} \in \{0, 1\} & \forall i, k \\ \sum_{k=1}^{K} r_{ik} = 1 & \forall i \end{cases}$$

Alternate optimization:

- Optimize for r given μ

Cost function:

$$\min_{\mu} \min_{r} \sum_{i \in \mathcal{D}} \sum_{k=1}^{K} \frac{1}{2} r_{ik} \|x^{(i)} - \mu_k\|_2^2 \quad \text{s.t.} \quad \begin{cases} r_{ik} \in \{0, 1\} & \forall i, k \\ \sum_{k=1}^{K} r_{ik} = 1 & \forall i \end{cases}$$

Alternate optimization:

- Optimize for r given μ

$$r_{ik} = \begin{cases} 1 & \text{if } k = \arg\min_{k \in \{1,\ldots,K\}} \|x^{(i)} - \mu_k\|_2^2 \\ 0 & \text{otherwise} \end{cases}$$

Cost function:

$$\min_{\mu} \min_{r} \sum_{i \in \mathcal{D}} \sum_{k=1}^{K} \frac{1}{2} r_{ik} \|x^{(i)} - \mu_k\|_2^2 \quad \text{s.t.} \quad \begin{cases} r_{ik} \in \{0,1\} & \forall i, k \\ \sum_{k=1}^{K} r_{ik} = 1 & \forall i \end{cases}$$

Alternate optimization:

- Optimize for r given μ

$$r_{ik} = \begin{cases} 1 & \text{if } k = \arg\min_{k \in \{1,\ldots,K\}} \|x^{(i)} - \mu_k\|_2^2 \\ 0 & \text{otherwise} \end{cases}$$

- Optimize for μ given r

Cost function:

$$\min_{\mu} \min_{r} \sum_{i \in \mathcal{D}} \sum_{k=1}^{K} \frac{1}{2} r_{ik} \|x^{(i)} - \mu_k\|_2^2 \quad \text{s.t.} \quad \begin{cases} r_{ik} \in \{0,1\} & \forall i, k \\ \sum_{k=1}^{K} r_{ik} = 1 & \forall i \end{cases}$$

Alternate optimization:

- Optimize for r given μ

$$r_{ik} = \begin{cases} 1 & \text{if } k = \arg\min_{k \in \{1,\ldots,K\}} \|x^{(i)} - \mu_k\|_2^2 \\ 0 & \text{otherwise} \end{cases}$$

- Optimize for μ given r

$$\nabla_{\mu_k} :$$

Cost function:

$$\min_{\mu} \min_{r} \sum_{i \in \mathcal{D}} \sum_{k=1}^{K} \frac{1}{2} r_{ik} \|x^{(i)} - \mu_k\|_2^2 \quad \text{s.t.} \quad \begin{cases} r_{ik} \in \{0,1\} & \forall i,k \\ \sum_{k=1}^{K} r_{ik} = 1 & \forall i \end{cases}$$

Alternate optimization:

- Optimize for r given μ

$$r_{ik} = \begin{cases} 1 & \text{if } k = \arg\min_{k \in \{1,\ldots,K\}} \|x^{(i)} - \mu_k\|_2^2 \\ 0 & \text{otherwise} \end{cases}$$

- Optimize for μ given r

$$\nabla_{\mu_k}: \qquad \sum_{i \in \mathcal{D}} r_{ik}(x^{(i)} - \mu_k) = 0$$

Cost function:

$$\min_{\mu} \min_{r} \sum_{i \in \mathcal{D}} \sum_{k=1}^{K} \frac{1}{2} r_{ik} \|x^{(i)} - \mu_k\|_2^2 \quad \text{s.t.} \quad \begin{cases} r_{ik} \in \{0,1\} & \forall i,k \\ \sum_{k=1}^{K} r_{ik} = 1 & \forall i \end{cases}$$

Alternate optimization:

- Optimize for r given μ

$$r_{ik} = \begin{cases} 1 & \text{if } k = \arg\min_{k \in \{1,\ldots,K\}} \|x^{(i)} - \mu_k\|_2^2 \\ 0 & \text{otherwise} \end{cases}$$

- Optimize for μ given r

$$\nabla_{\mu_k}: \qquad \sum_{i \in \mathcal{D}} r_{ik}(x^{(i)} - \mu_k) = 0$$

$$\mu_k = \frac{\sum_{i \in \mathcal{D}} r_{ik} x^{(i)}}{\sum_{i \in \mathcal{D}} r_{ik}}$$

Properties of Algorithm:

Properties of Algorithm:

- Local optimum is found

Properties of Algorithm:

- Local optimum is found
- Guaranteed to converge in a finite number of iterations

Properties of Algorithm:

- Local optimum is found
- Guaranteed to converge in a finite number of iterations
- Running time per iteration:

Properties of Algorithm:

- Local optimum is found
- Guaranteed to converge in a finite number of iterations
- Running time per iteration:
  - Assign data points to closest cluster center

Properties of Algorithm:

- Local optimum is found
- Guaranteed to converge in a finite number of iterations
- Running time per iteration:
  - Assign data points to closest cluster center

$$O(KNd)$$

Properties of Algorithm:

- Local optimum is found
- Guaranteed to converge in a finite number of iterations
- Running time per iteration:
  - Assign data points to closest cluster center

$$O(KNd)$$

  - Change the cluster center to the average of its assigned points

Properties of Algorithm:

- Local optimum is found
- Guaranteed to converge in a finite number of iterations
- Running time per iteration:
  - Assign data points to closest cluster center

$$O(KNd)$$

  - Change the cluster center to the average of its assigned points

$$O(Nd)$$

Extensions:

Extensions:

- kMeans is very sensitive to initialization: kMeans++

Extensions:

- kMeans is very sensitive to initialization: kMeans++
- kMeans depends on the feature space: Kernels

Extensions:

- kMeans is very sensitive to initialization: kMeans++
- kMeans depends on the feature space: Kernels
- Evaluation

kMeans++: How to initialize kMeans

kMeans++: How to initialize kMeans

- Randomly choose first center

kMeans++: How to initialize kMeans

- Randomly choose first center
- Pick new center with probability proportional to $\|x^{(i)} - \mu_k\|_2^2$ (contribution of $x^{(i)}$ to total error)

kMeans++: How to initialize kMeans

- Randomly choose first center
- Pick new center with probability proportional to $\|x^{(i)} - \mu_k\|_2^2$ (contribution of $x^{(i)}$ to total error)
- Repeat until K centers are chosen

kMeans++: How to initialize kMeans

- Randomly choose first center
- Pick new center with probability proportional to $\|x^{(i)} - \mu_k\|_2^2$ (contribution of $x^{(i)}$ to total error)
- Repeat until K centers are chosen

Try multiple initializations and choose the best

Distance measure:

Distance measure:

- Euclidean (most commonly used)

Distance measure:

- Euclidean (most commonly used)
- Cosine

Distance measure:

- Euclidean (most commonly used)
- Cosine
- Non-linear (via Kernels): $\phi(x^{(i)})$

How to evaluate clusters?

How to evaluate clusters?

- Generative: How well are points reconstructed from the clusters

Distortion

How to evaluate clusters?

- Generative: How well are points reconstructed from the clusters

  Distortion
- Discriminative: How well do the clusters correspond to labels

  Purity

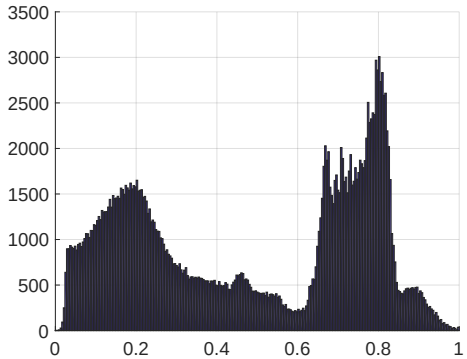
Applications

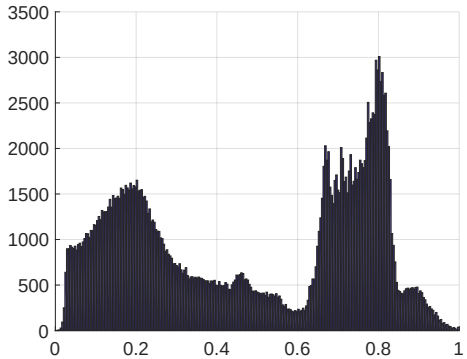
Applications

- Clustering

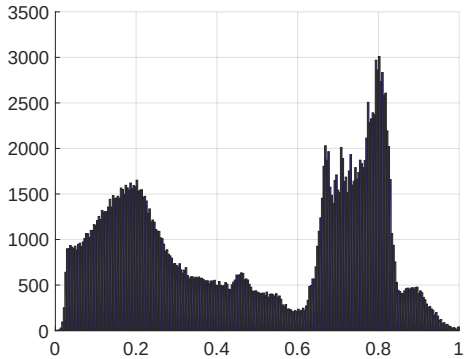
Applications

- Clustering
- Super-pixel segmentation


3500
3000
2500
2000
1500
1000
500
0
0
0.2
0.4
0.6
0.8
1

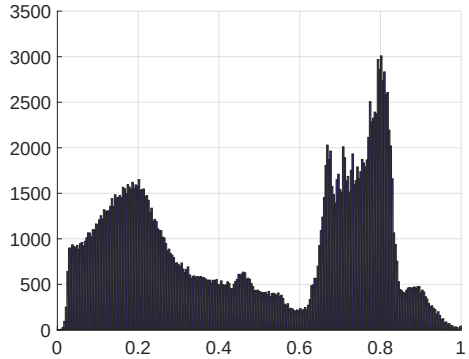


- Grayscale $x^{(i)} \in \mathbb{R}$



2 clusters 3 clusters

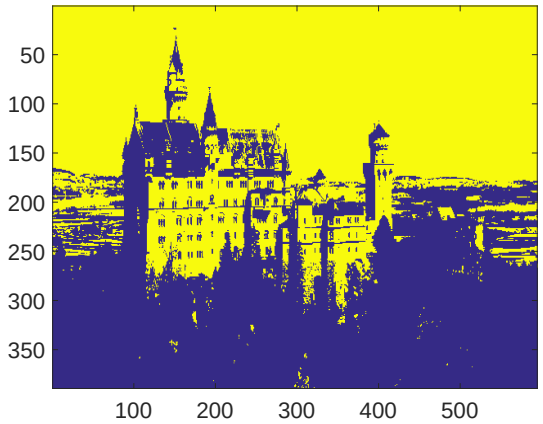
- Grayscale $x^{(i)} \in \mathbb{R}$

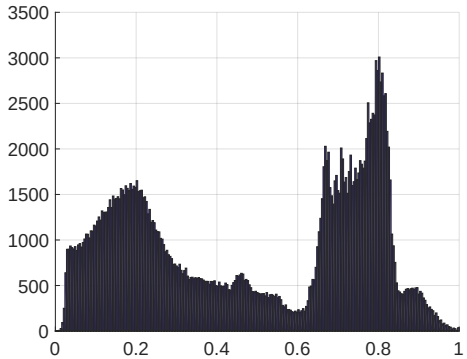


2 clusters

3 clusters

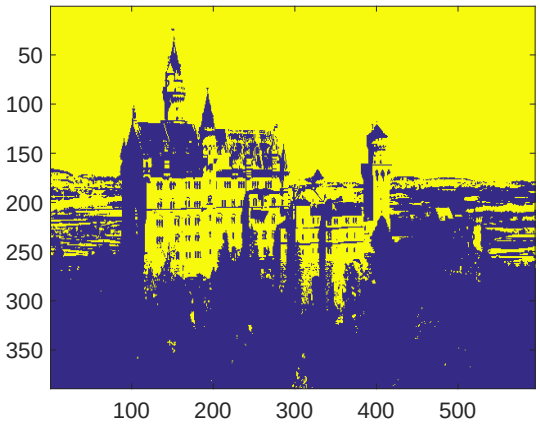
- Grayscale $x^{(i)} \in \mathbb{R}$



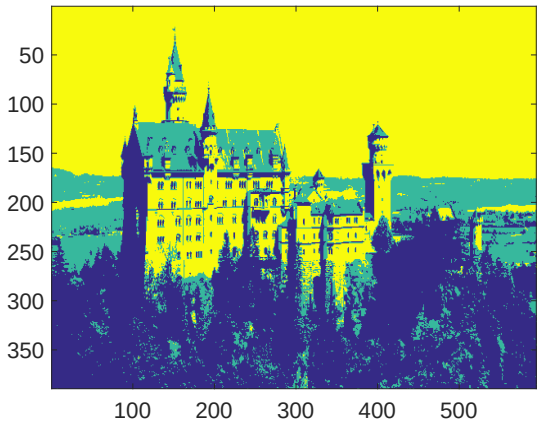


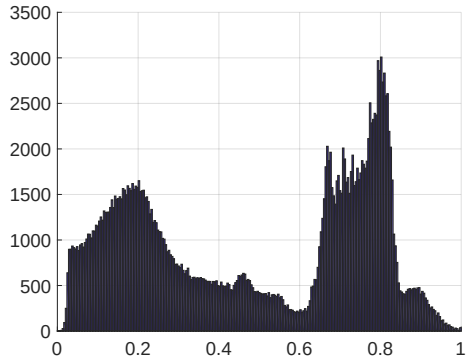
- Grayscale $x^{(i)} \in \mathbb{R}$

2 clusters



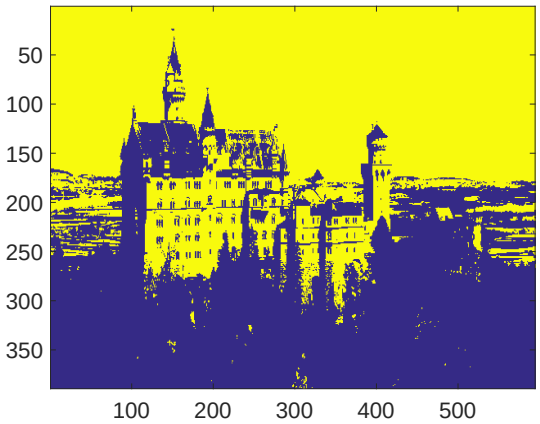
3 clusters



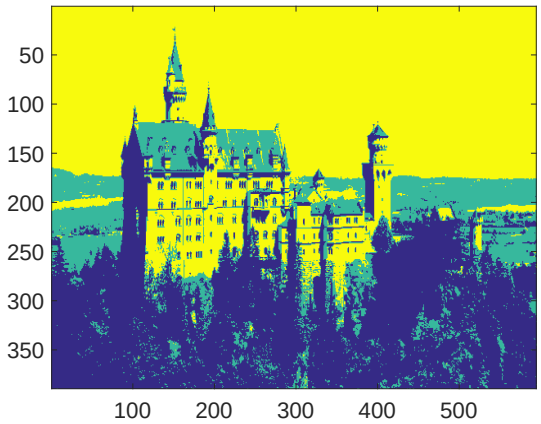


- Grayscale $x^{(i)} \in \mathbb{R}$

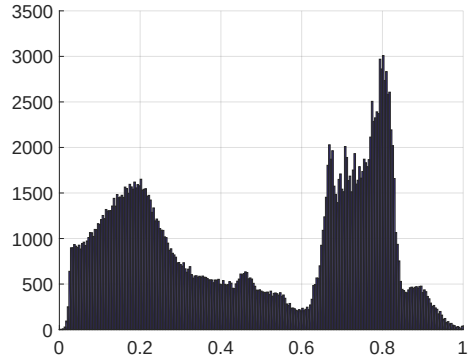
2 clusters



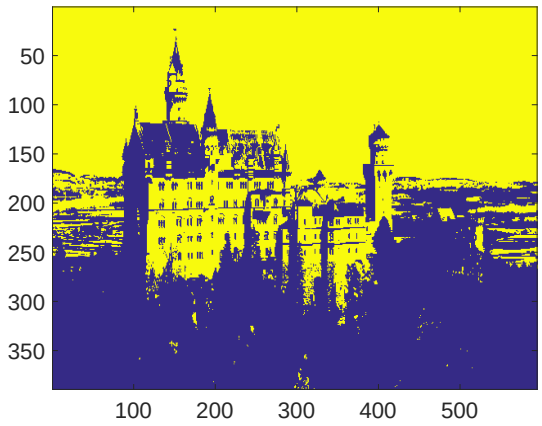
3 clusters

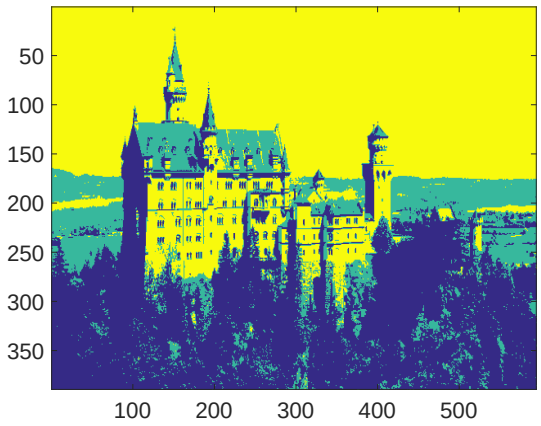


- Color space $x^{(i)} \in \mathbb{R}^3$

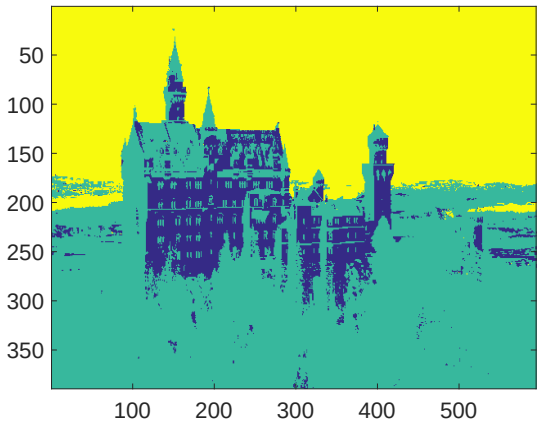


- Grayscale $x^{(i)} \in \mathbb{R}$ 2 clusters 3 clusters





- Color space $x^{(i)} \in \mathbb{R}^3$



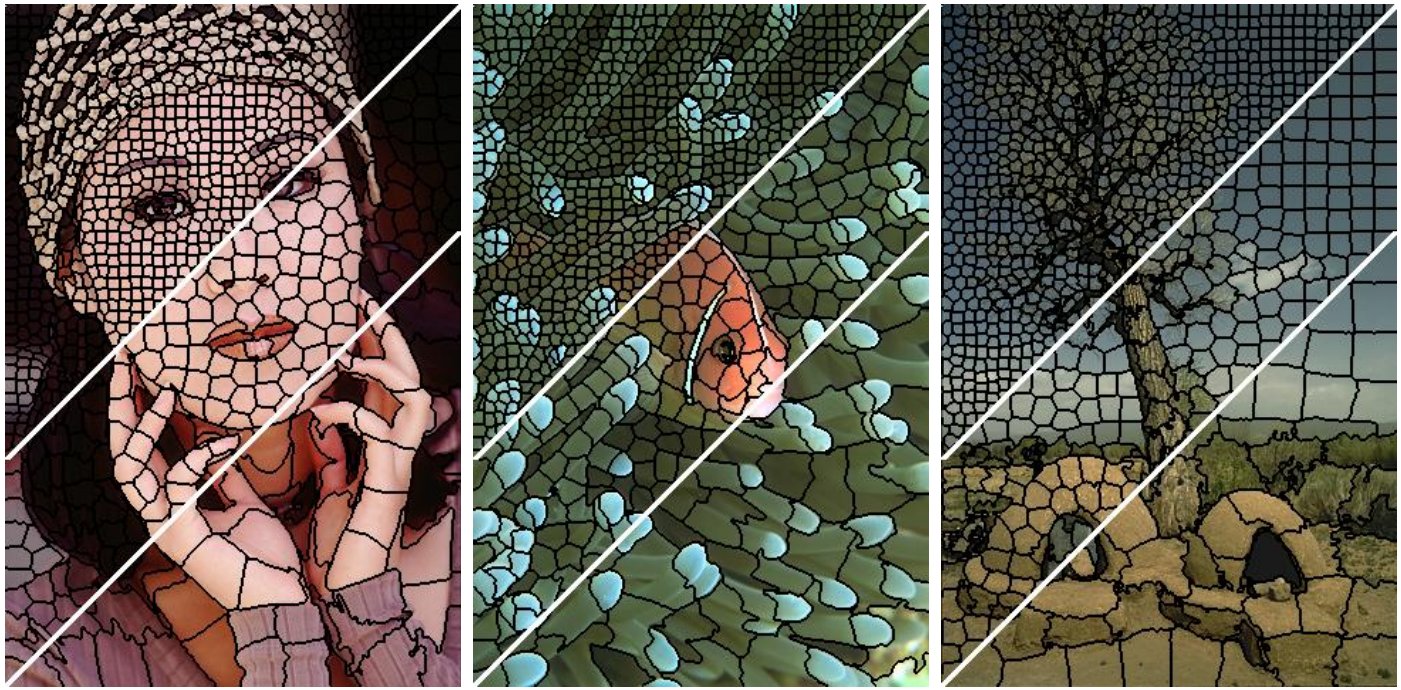
How to obtain spatial smoothness?

How to obtain spatial smoothness?

Augment the feature space $\phi(x^{(i)})$ to contain spatial coordinates in addition to intensities

Superpixel segmentation

Superpixel segmentation



Superpixel segmentation

Superpixel segmentation

- 5D feature space (lab color space, 2d spatial coordinates)

Superpixel segmentation

- 5D feature space (lab color space, 2d spatial coordinates)
- Distance metric treats color dimensions and spatial dimensions differently

Superpixel segmentation

- 5D feature space (lab color space, 2d spatial coordinates)
- Distance metric treats color dimensions and spatial dimensions differently
- Initial cluster centers are spaced regularly on the image and slightly perturbed to avoid edges

Summary:

Pros:

Summary:

Pros:

- Simple

Summary:

Pros:

- Simple
- Easy to implement

Summary:

Pros:

- Simple
- Easy to implement

Cons:

Summary:

Pros:

- Simple
- Easy to implement

Cons:

- Need to choose K

Summary:

Pros:

- Simple
- Easy to implement

Cons:

- Need to choose K
- Sensitive to outliers

Summary:

Pros:

- Simple
- Easy to implement

Cons:

- Need to choose K
- Sensitive to outliers
- Can get stuck in local minima

Summary:

Pros:

- Simple
- Easy to implement

Cons:

- Need to choose K
- Sensitive to outliers
- Can get stuck in local minima
- All cluster centers have same parameters (non-adaptive)

Summary:

Pros:

- Simple
- Easy to implement

Cons:

- Need to choose K
- Sensitive to outliers
- Can get stuck in local minima
- All cluster centers have same parameters (non-adaptive)
- Can be slow $O(KNd)$

Quiz:

Quiz:

- What is the cost function for kMeans?

Quiz:

- What is the cost function for kMeans?
- What are the steps of the kMeans algorithm?

Quiz:

- What is the cost function for kMeans?
- What are the steps of the kMeans algorithm?
- What are the guarantees of the kMeans algorithm?

Important topics of this lecture

Important topics of this lecture

- Understanding kMeans

Important topics of this lecture

- Understanding kMeans
- Getting to know different mechanisms to adjust kMeans