Image Segmentation

Computer Vision

Exam Preparation

Overview

- Traditional Methods:
  - K-Means
  - KDE
- Deep Learning Methods
  - Metrics
  - Transposed convolution

K-Means

1. Analyzing K-means Algorithm Complexity

Question: What is the algorithmic complexity of the K-means algorithm per iteration, where K is the number of clusters and N is the number of data points?

a) $O(N^2)$

b) $O(K \cdot N)$

c) $O(K^2 \cdot N)$

d) $O(N \cdot \log N)$

K-Means

Algorithm 1 K-means Clustering Algorithm

Require: Data points $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, number of clusters K

Ensure: Cluster centroids $M = \{\mu_1, \mu_2, \ldots, \mu_K\}$ and assignments $f(\mathbf{x}_i)$

Init: Randomly initialize K cluster centroids $\mu_1, \mu_2, \ldots, \mu_K$

repeat

  for each data point $\mathbf{x}_i$ **do**

    E-Step: Assign $\mathbf{x}_i$ to the nearest cluster centroid:

$$f(\mathbf{x}_i) = \underset{j \in \{1,\ldots,K\}}{\arg\min} \|\mathbf{x}_i - \mu_j\|^2$$

  end for

  for each cluster $j = 1, \ldots, K$ **do**

    M-Step: Update the centroid μ_j by averaging all points assigned to it:

$$\mu_j = \frac{\sum_{i=1}^{N} I_j(\mathbf{x}_i) \cdot \mathbf{x}_i}{\sum_{i=1}^{N} I_j(\mathbf{x}_i)}$$

where

$$I_j(\mathbf{x}_i) = \begin{cases} 1 & \text{if } f(\mathbf{x}_i) = j \\ 0 & \text{otherwise.} \end{cases}$$

  end for

until convergence (i.e., cluster centroids no longer change or assignments remain stable)

return the final cluster centroids M and assignments $f(\mathbf{x}_i)$

K-Means

Correct Answer: b) $O(K \cdot N)$

Explanation: The K-means algorithm consists of two main steps in each iteration:

1. Assignment step: Each of the N data points must be compared to all K centroids to find the nearest centroid, requiring $O(K \cdot N)$ operations.

2. Update step: Computing new centroids requires averaging the points in each cluster, taking $O(N)$ operations.

Kernel Density Estimation (KDE)

2. Role of Bandwidth in Kernel Density Estimation (KDE)

Question: In Kernel Density Estimation (KDE), what is the primary role of the bandwidth parameter h?

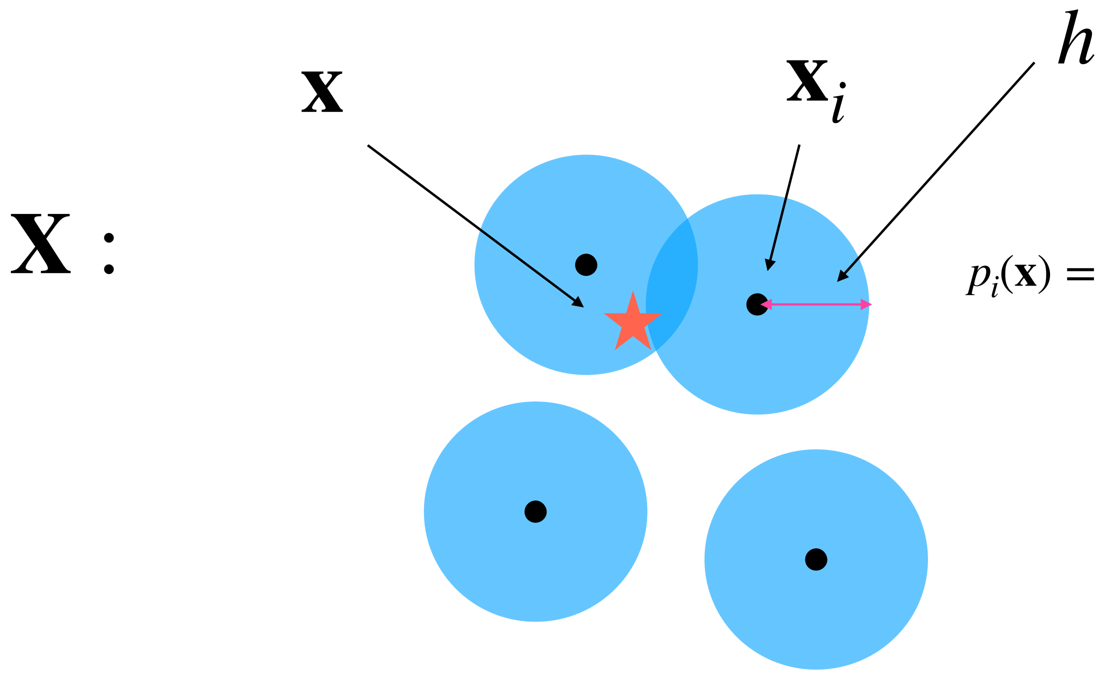
a) The smoothness of the estimated density function

b) The number of clusters in the resulting distribution

c) The dimensionality of the feature space

d) The mean of the underlying data distribution

Kernel Density Estimation (KDE)



$$p_i(\mathbf{x}) = K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right) = \mathcal{N}(\mathbf{x} \,|\, \mathbf{x}_i, \Sigma = h^2 I)$$

Key intuition: every point defines an isotropic Gaussian, **with fixed variance defined by bandwidth *h***. The density at the point is then estimated as a mixture of these Gaussians.

Kernel Density Estimation (KDE)

Correct Answer: a) The smoothness of the estimated density function
Explanation: The bandwidth parameter h determines:

- The spread of each kernel function around its data point
- Larger h values result in smoother density estimates where each data point influences a wider area
- Smaller h values produce more detailed, peaked estimates that closely follow individual data points
- Serves as a key parameter balancing between over-smoothing and under-smoothing of the density estimate

Metrics

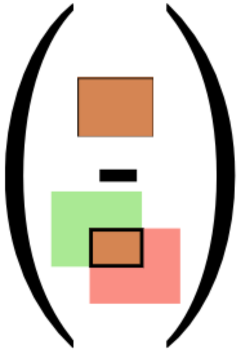
3. Calculating mIoU and mPA from Ground Truth and Predictions

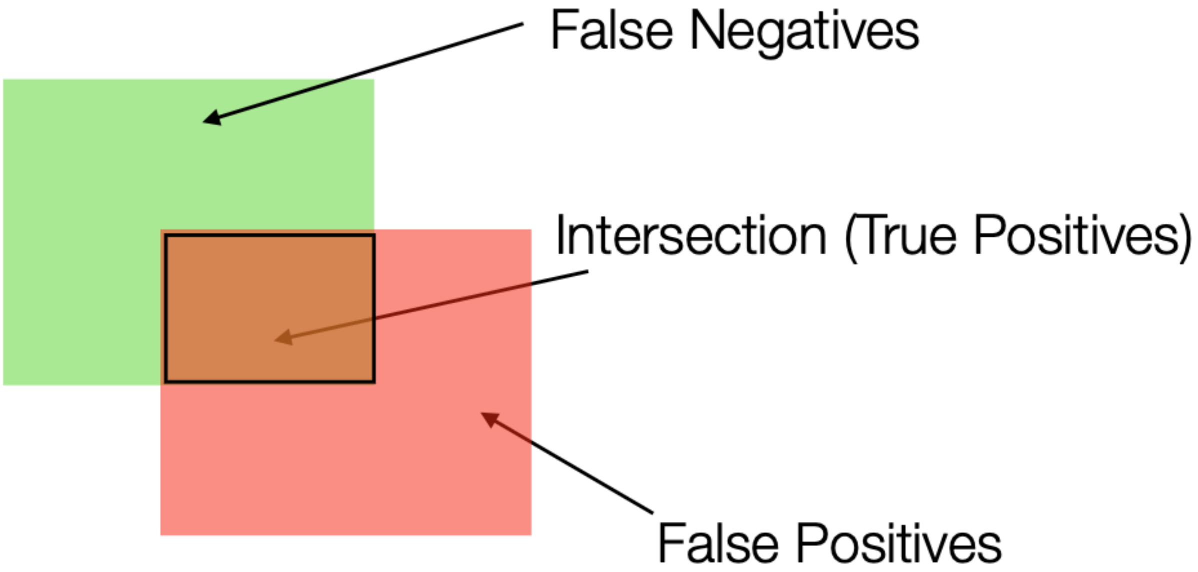
Question: Given the ground truth matrix Y and prediction matrix P:

$$Y = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} \quad P = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Calculate the mean Intersection over Union (mIoU) and mean Pixel Accuracy (mPA).

Metrics (IoU)





$$\left(\frac{\text{True Positives}_k}{\text{True Positives}_k + \text{False Positives}_k + \text{False Negatives}_k}\right)$$

Metrics (Pixel Accuracy)

Pixel Accuracy:

$$PA = \frac{\sum_{i=1}^{N} I(y_i = p_i)}{N}$$

$\left(\frac{\text{\# correctly classified pixels}}{\text{\# all pixels}}\right)$

Class Pixel Accuracy:

$$PA_k = \frac{\sum_{i=1}^{N} I(y_i = k) I(y_i = p_i)}{N_k}$$

(Same, but for a specific class)

Mean Pixel Accuracy (mPA):

$$\text{mPA} = \frac{1}{K} \sum_{k=1}^{K} \text{PA}_k$$

(Averaged across all classes)

Metrics

Question: Given the ground truth matrix Y and prediction matrix P:

$$Y = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} \quad P = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Calculate the mean Intersection over Union (mIoU) and mean Pixel Accuracy (mPA).

Solution:

1. **Intersection over Union (IoU):**

   - Class 1:

$$\text{IoU}_1 = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}} = \frac{1}{1+1+1} = \frac{1}{3}$$

   - Class 0:

$$\text{IoU}_0 = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}} = \frac{1}{1+1+1} = \frac{1}{3}$$

   - mIoU:

$$\text{mIoU} = \frac{\text{IoU}_0 + \text{IoU}_1}{2} = \frac{1/3 + 1/3}{2} = \frac{1}{3} \approx 0.333$$

2. **Pixel Accuracy (PA):**

   - Class 1:

$$\text{PA}_1 = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{1}{2} = 0.5$$

   - Class 0:

$$\text{PA}_0 = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{1}{2} = 0.5$$

   - mPA:

$$\text{mPA} = \frac{\text{PA}_0 + \text{PA}_1}{2} = \frac{0.5 + 0.5}{2} = 0.5$$

Transposed Convolution

4. Output Size Calculation for Transposed Convolution

Question: Calculate the output dimensions of a transposed convolution with the following parameters:

- Input size: 2×2
- Kernel size (K): 2×2
- Stride (S): 1
- Padding (P): 0

Transposed Convolution

Transposed convolution with **stride 1**, **kernel size 2**:

$$\begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix} \longrightarrow \begin{bmatrix} y_{11} & y_{12} & y_{13} \\ y_{21} & y_{22} & y_{23} \\ y_{31} & y_{32} & y_{33} \end{bmatrix}$$

$$W = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix}, b$$

(Our learnable **kernel**)

$$y_{22} = \boxed{x_{11} \cdot w_{22}} + \boxed{x_{12} \cdot w_{21}} + \boxed{x_{21} \cdot w_{12}} + \boxed{x_{22} \cdot w_{11}} + b$$

This behaviour can lead to “**checkerboard artifacts**” [1] when stride is selected improperly.

[1] Odena, Augustus, Vincent Dumoulin, and Chris Olah. "Deconvolution and checkerboard artifacts." *Distill* 1.10 (2016): e3.

Transposed Convolution

Convolution Arithmetic

General Formula

W - input size,
K - kernel size,
S - stride size,
P - (input) padding size,
O - output size.

Convolution:

$$O = \frac{W - K + 2P}{S} + 1$$

Transposed Convolution:

$$O = S \cdot (W - 1) + K - 2P$$

Transposed Convolution

Solution: For transposed convolution, the output size (O) is given by:

$$O = S \cdot (I - 1) + K - 2P$$

where I is the input size.

Substituting the values:

$$O = 1 \cdot (2 - 1) + 2 - 2 \cdot 0 = 1 + 2 = 3$$

Therefore, the output feature map will be 3×3.