Lab 4

Machine Learning Algorithms

Activity 1: K-Means. In this activity, we propose to implement the K-Means algorithm as described in the lecture notes. Given a set of n pixels $\mathcal{P} = \{p_0, \cdots, p_{n-1}\}$ and a predefined number of clusters k, the k-means algorithm assignes each pixel to a unique cluster. A pixel belong to a cluster if it has a similar color that the average color of this cluster. Each pixel are denoted $p_i := \{r, g, b, x, y\}$ with (r, g, b) are the color components, (x, y) are the 2D coordinates in the image.

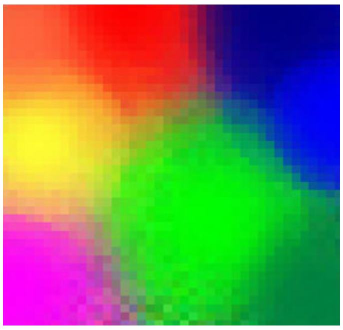

Figure 1: An input for K-Means (Image courtesy of dataaspirant.com).

A) Explain the key idea of the K-Means algorithm.

B) Explain the key steps of the classical pseudo-code of the K-Means algorithm. What is the input? What is the output? What are the subroutines or mathematical ingredients needed? What are the data structures needed?

C) Convert the pseudo-code of the K-Means algorithm into code using the language of your choice.

D) Construct or generate the corresponding dataset to provide as input. Apply the K-Means algorithm on a toy example.

E) Test and Evaluate the performance of your K-Means implementation.

Activity 2: K-Nearest Neighbors. In this activity, we propose to implement a classical machine learning algorithm named *K-Nearest Neighbors* as described in the lecture note. Given a set of n points $\mathcal{P} = \{p_0, \cdots, p_{n-1}\}$ in $\mathbb{R}^{2\times n}$ (also called the point cloud) and a given point of interest $q \in \mathcal{P}$. The KNN algorithm returns the subset $\mathcal{C} \subset \mathcal{P}$ of k nearest points to q in Euclidean sense. Each point $p_i = (x_i, y_i)$ is represented by its coordinates in 2D space.

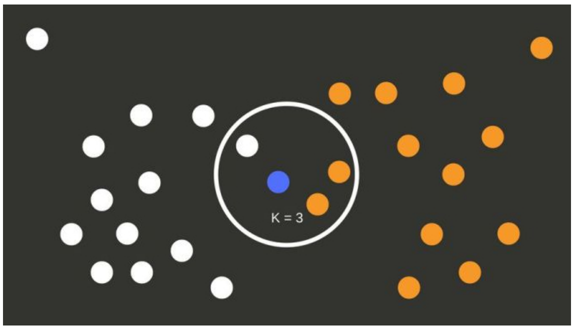

Figure 2: K-Nearest Neighbors (Image courtesy of dataaspirant.com).

A) Explain the key idea of the K-Nearest Neighbors algorithm.

B) Explain the key steps of the classical pseudo-code the K-Nearest Neighbors algorithm. What is the input? What is the output? What are the subroutines or mathematical ingredients needed? What are the data structures needed?

C) Convert the pseudo-code of the K-Nearest Neighbors algorithm into code using the language of your choice.

D) Construct or generate the corresponding dataset to provide as input. Apply the K-Nearest Neighbors algorithm on a toy example.

E) Test and evaluate the performance of your K-Nearest Neighbors implementation.

Activity 3: Drafting the coursework. In this activity, we propose to prepare the writing part of your coursework.

A) Have you checked that your selected papers are reliable? How do you motivate your choice?

B) Start to prepare an outline to structure the paragraphs. Each paragraph should focus on a specific aspect in which it is possible to compare both papers.

B) What is the most emerging aspect to highlight given your selected papers?

Extra Activity: Principal Component Analysis. In this activity, we propose to implement the Principal Component Analysis algorithm as described in the lecture.

A) Explain the key steps of the classical pseudo-code the Principal Component Analysis algorithm. What is the input? What is the output? What are the subroutines or mathematical ingredients needed? What are the data structures needed. Write the pseudo-code of the PCA algorithm.

B) Convert the pseudo-code of the PCA algorithm into code using the programming language of your choice.

C) Apply the PCA algorithm on small toy dataset that you will have previously generated or created.

D) Evaluate the accuracy of your PCA implementation. How to evaluate?