CM146, Fall 2021
Problem Set 3: Clustering, Kernel, SVM, Bayesian Learning
Due Dec 5th, 2022 at 11:59 pm

1 Kernels [30 pts]

One way to construct complex kernels is to build them from simple ones, using the properties of the kernels. In the following, we will prove a list of properties of kernels (Rule 1 – Rule 3).

(a) **(10 pts)** Rule 1: Suppose we have $x \in \mathbb{X}, z \in \mathbb{X},\ \ g : \mathbb{X} \to \mathbb{R}$. Prove that $k(x, z) = g(x) \times g(z)$ is a valid kernel by constructing a feature map $\Phi(\cdot)$ and show that $k(x, z) = \Phi(x)^T \Phi(z)$.

Solution: Let $\Phi(x) = [g(x)]$ then $k(x, z) = g(x) \times g(z) = [g(x)]^T [g(z)] = \Phi(x)^T \Phi(z)$.

(b) **(10 pts)** Rule 2: Suppose we have a valid kernel $k_1(x, z) = \Phi_1(x)^T \Phi_1(z)$. Prove that $k(x, z) = \alpha \cdot k_1(x, z)\ \ \forall \alpha \geq 0$ is also a valid kernel by constructing a new feature map $\Phi(\cdot)$ using $\Phi_1(\cdot)$ and show that $k(x, z) = \Phi(x)^T \Phi(z)$.

Solution: Consider the new feature map: $\Phi = \sqrt{\alpha} \cdot \Phi_1$

Then we have

$$\begin{aligned}
\Phi(x)^T \Phi(z) &= \sqrt{\alpha} \cdot \Phi_1(x)^T \sqrt{\alpha} \cdot \Phi_1(z) \\
&= \alpha \cdot \Phi_1(x)^T \Phi_1(z) \\
&= \alpha \cdot k_1(x, z) \\
&= k(x, z)
\end{aligned}$$

(c) **(10 pts)** Rule 3: Suppose we have two valid kernels $k_1(x, z) = \Phi_1(x)^T \Phi_1(z)$ and $k_2(x, z) = \Phi_2(x)^T \Phi_2(z)$. Prove that $k(x, z) = k_1(x, z) + k_2(x, z)$ is also a valid kernel by constructing a new feature map $\Phi(\cdot)$ using $\Phi_1(\cdot)$ and $\Phi_2(\cdot)$ and show that $k(x, z) = \Phi(x)^T \Phi(z)$.

Solution: Consider the new feature map: $\Phi = [\Phi_1, \Phi_2]$, i.e., the concatenation of Φ_1 and Φ_2

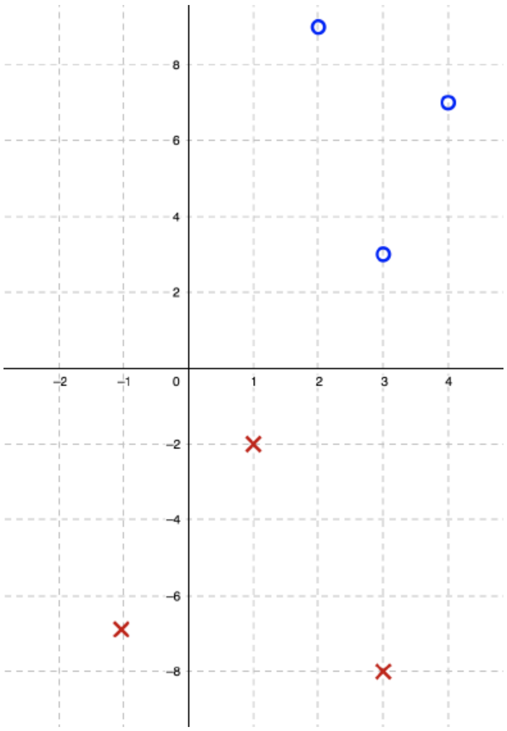
Then

$$\begin{aligned}
\Phi(x)^T \Phi(z) &= [\Phi_1(x), \Phi_2(x)]^T [\Phi_1(z), \Phi_2(z)] \\
&= \Phi_1(x)^T \Phi_1(z) + \Phi_2(x)^T \Phi_2(z) \\
&= k_1(x, z) + k_2(x, z) \\
&= k(x, z)
\end{aligned}$$

2 SVM [35 pts]

Suppose we have the following six training examples. x_1, x_2, x_3 are positive instances ($y = 1$) and x_4, x_5, x_6 are negative instances ($y = -1$) . Note: we expect you to use a simple geometric argument to narrow down the search and derive the same solution that an SVM optimization would find for the following two questions. You should NOT need to write a program to solve this problem.

$Example$	$feature_1$	$feature_2$	y
x_1	3	3	1
x_2	2	9	1
x_3	4	7	1
x_4	1	-2	-1
x_5	-1	-7	-1
x_6	3	-8	-1



(a) **(10 pts)** Suppose we are looking for a hard-SVM decision boundary $\boldsymbol{w}^T\boldsymbol{x}_n + b = 0$ passing through the origin (i.e., $b = 0$). In other words, we minimize $||\boldsymbol{w}||_2$ subject to $y_n\boldsymbol{w}^T\boldsymbol{x}_n \geq 1, n = 1, \ldots, N$. Identify the `support vectors` (data points that are actually used in the calculation of w and margin) in this training dataset.

Solution: Plotting the points on a 2D plane one can find that there are only two support vectors: one from the positive group, the other from the negative group. They should be samples that are closest to the hyper plane that separates the two classes. By visually inspecting the distribution of the training examples in the 2D plane, we can conclude that x_1 and x_4 are the supporting vectors.

Rubric:

- +3 points: gives only two data points (no matter they are correct or not).
- +7 points: correctly gives the right data points.

(b) **(10 pts)** Following part (a), what is $\boldsymbol{w}^* \in \mathbb{R}^2$ in this case and what is the margin: $\frac{1}{\|w^*\|_2}$?

Solution: In this case, the SVM uses both data points in (a) as support vectors such that $y_1\boldsymbol{w}^T\boldsymbol{x}_1 = 1$ and $y_4\boldsymbol{w}^T\boldsymbol{x}_4 = 1$. Solving the equations the corresponding $\boldsymbol{w}$ can be derived:

$$\boldsymbol{w}^* = [-\frac{1}{9}, \frac{4}{9}]^T \qquad \frac{1}{\|\boldsymbol{w}^*\|_2} = \frac{1}{\sqrt{\frac{17}{81}}} = 2.183,$$

Rubric:

- +2 points: gives a 2-dim weight vector $\mathbf{w}$ (no matter whether it is correct or not).
- +4 points: correctly gives the weight vector.
- +4 points: correctly gives the margin.
- Partial credit can be given if student provides reasonable description of how to derive the optimal weight.

(c) **(15 pts)** Suppose we now allow the offset parameter b to be non-zero. In other words, we minimize $||\boldsymbol{w}||_2$ subject to $y_n\boldsymbol{w}^T\boldsymbol{x}_n + b \geq 1, n = 1, \ldots, N$. How would the classifier and the actual margin change in the previous question? What are $\boldsymbol{w}^*, b^*, \frac{1}{\|\boldsymbol{w}^*\|_2}$,? Compare your solutions with problem (b).

Solution: In this case, we have a minimization problem with two equality constrains and 2 variables w, b.

$$\begin{aligned} &min \;\; \|w\|_2 \;\; s.t. \\ &y_1 \cdot (\boldsymbol{w}^T\boldsymbol{x}_1 + b) = 1 \\ &y_4 \cdot (\boldsymbol{w}^T\boldsymbol{x}_4 + b) = 1 \end{aligned}$$

from the two constrains we can represent w_1 and w_2 in terms of b:

$$w_1 = -\frac{5b+1}{9}$$

$$w_2 = \frac{2b+4}{9}$$

plug into the minimization problem and we have

$$min \;\; \|w\|_2 \;\; = min\sqrt{(\frac{5b+1}{9})^2 + (\frac{2b+4}{9})^2}$$

the equation under the square root is convex so we can find the global minimal by taking the derivative with respect to w and set to 0.

The corresponding $\boldsymbol{w}$, and b are:

$$\boldsymbol{w}^* = [\frac{4}{29}, \frac{10}{29}]^T \quad b^* = -\frac{13}{29} \quad \frac{1}{\|\boldsymbol{w}^*\|_2} = \frac{1}{\sqrt{\frac{116}{841}}} = 2.6926,$$

The margin for the classifier with offset is larger than the margin for the classifier without offset.

Rubric:

- +5 points: correctly provides weight vector.
- +5 points: correctly provides bias.
- +5 points: correctly describes the margin change with and without offset.
- Partial credit can be given if student provides reasonable description of how to derive the optimal weight.

3 Bayesian Learning [10 pts]

We are testing a set of light bulbs from the same manufacturer, and each light bulb has the same probability p to pass the test.

(a) **(5 pts)** We test 5 bulbs and find that only the first 3 bulbs pass the test. What is the most likely value of p based on MLE? Complete the following derivation.

The likelihood function that describes the observations as a function of p is $L(p) =$ __________. Therefore, the log-likelihood is $\log L(p) =$ __________. Maximizing the log-likelihood, we obtain $p_{MLE} =$ __________ (write down a real number rounded to 2 decimal places in the format of X.XX).

Solution:

blank #1: $p^3(1-p)^2$.
blank #2: $3\log p + 2\log(1-p)$.
blank #3: 0.60.

(b) **(5 pts)** Now, we assume that the probability density function of the prior distribution of p is $P(p) = 2p,\ \ p \in [0,1]$. If we test 5 bulbs and find that only the first 3 bulbs pass the test (represented as the observation D), what is the most likely value of p based on maximum-a-posteriori (MAP) estimation? Complete the following derivation.

The posterior $P(p|D)$ is proportional to ______________ (write down as a function of p). Therefore, the MAP estimation of p is $p_{MAP} =$ ______________ (write down a real number rounded to 2 decimal places in the format of X.XX).

Solution:

blank #1: $P(p|D) \propto P(D|p)P(p) = p^3(1-p)^2 \times 2p \propto p^4(1-p)^2$.
blank #2: $\frac{4}{4+2} \approx 0.67$.

4 Clustering [25 pts]

4.1 Implement k-means manually[5 pts]

In this problem, we will perform the KMeans algorithm manually for two-dimensional data.

Assume we have data points as follows:

$x_1 = [1,2], x_2 = [4,1], x_3 = [0,2.5], x_4 = [3,-1]$.

We assume k=2, and the centers of the 2 clusters were initialized as $\mu_1 = [0,0]$ and $\mu_2 = [3,0]$. What are μ_1 and μ_2 after the model converge?

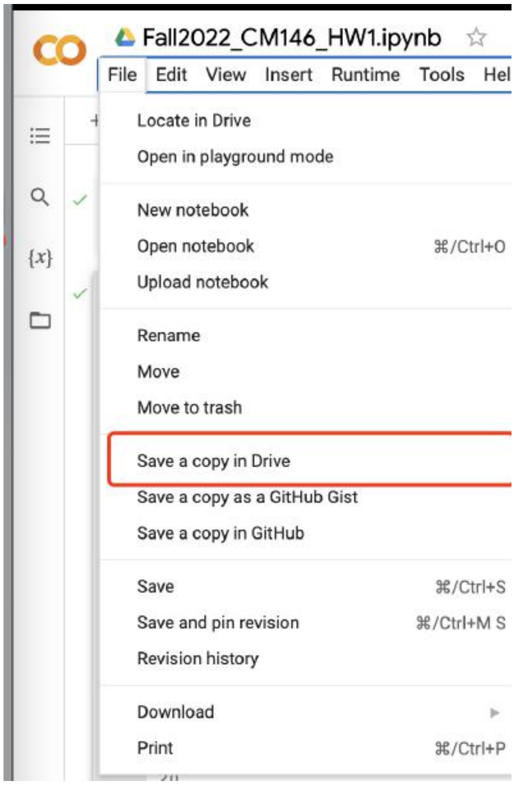
4.2 KMeans coding exercise [20 pts]

Introduction

In this problem, we will work on a mushroom clustering task. The dataset is adapted from the UCI Machine Learning Repository and contains descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms. Each mushroom is described in terms of physical characteristics, and the goal is to see how clustering algorithms could split the dataset properly and how the clustering match mushrooms labels as *edible* or *poisonous*. We will apply Kmenas and Kmedoids and here we use all 22 features in the original dataset.

For all the coding, please refer to the following Colab notebook Fall2022-CM146-HW3.ipynb.

Before executing or writing down any code, please make a copy of the notebook and save it to your own google drive by clicking the "File" → "Save a copy in Drive".



You will probably be prompted to log into your Google account. Make sure that all the work you implement is done on your own saved copy. You will not be able to make changes on the original notebook shared with the entire class.

The notebook has marked blocks where you need to code:

```
### ========== TODO : START ========== ###
### ========== TODO : END ========== ###
```

Submission instructions for programming problems

- Please save the execution output in your notebook. When submitting, export the notebook to a `.ipynb` file by clicking "File" → "Download .ipynb" and upload the notebook to BruinLearn.
- Your code should be commented appropriately. Importantly:
  - Your name should be at the top of the file.
  - Each class and method should have an appropriate doctsring.
  - Include some comments for anything complicated.

  There are many possible solutions to this assignment, which makes coding style and comments important for graders to conveniently understand the code.
- Please submit all the plots and the rest of the solutions (other than codes) to Gradescope.

Solution:

Code: `https://colab.research.google.com/drive/1lY9XtmMcIP43jgElaHvk3_Eeu6pTUfCr?usp=sharing`

Results may differ due to randomness and difference in implementation.

Documentation

- PCA: link
- KMeans: link
- KMedoids: link

The implementation for both Kmeans and Kmedoids was added. Use these two classes to apply clustering on the mushroom dataset instead of the KMeans implementation from scikit-learn.

(a) **(5 pts)** Based on the code provided, apply Kmedoids (with K=2) with 50 iterations and evaluate its performance. Kmedoids is a clustering algorithm, and we assume the data is unlabeled. However, we can use the labels of the data to evaluate its performance. Specifically, we define purity as follows. We assign a label to each cluster based on the most frequent class in it. The purity is then defined as the number of correctly matched class and cluster labels divided by the number of total data points. For example, if the Kmedoids algorithm outputs two clusters, cluster 1 has 20 positive examples, and 10 negative examples; cluster 2 has 10 positive example, and 35 negative example, the purity is (20+35)/(20+10+10+35)=73%.

Based on the above description, report the purity of the clusters generated by the kmedoids algorithm on the mushroom data.

Solution: the number might vary depending on initialization, mainly the implementation will be checked based on the purity definition. The range of purity is from 50% to 100%

(b) **(5 pts)** Based on the code provided, apply Kmeans (with K=2) with 50 iterations and evaluate its performance. Report the purity of the clusters generated by the KMeans algorithm on the mushroom data.

Solution: the number might vary depending on initialization, mainly the implementation will be checked based on the purity definition. The range of purity is from 50% to 100%

(c) **(10 pts)** *Visualizing the clusters.* Principle Component Analysis (PCA) is a technique that maps high-dimensional data into a low-dimension space while preserving as much information as possible. For example, Figure 3 shows the original data in 3D and Figure 4 shows the transformed data in 2D using PCA.

The mushroom dataset has 22 features and we cannot visualize the clusters generated by Kmeans on 22D space. Therefore, in this question, we will use PCA to map the data from 22D into 2D to visualize the clusters.

Please plot the following two figures:

In one figure, plot the data with two colors representing the clusters assigned by the KMeans algorithm.

In another figure, plot the data while using two colors showing their actual labels on the mushroom dataset.

Report both figures in your answer and discuss your findings.

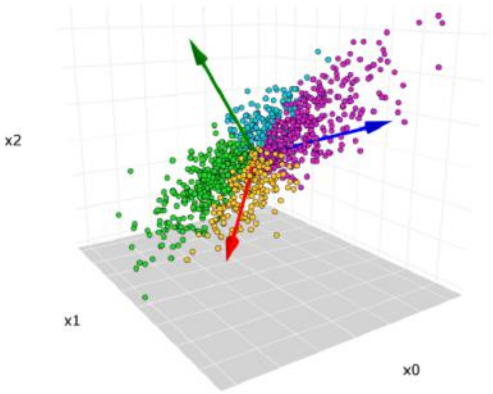


Figure 1: Original high dimensional data

Solution: the plots show that the data is not well clustered in both cases and that might be due to the nature of the data. in each case, comparing figure 1 and 2 demonstrate that the data around the center are hard to cluster with both algorithms

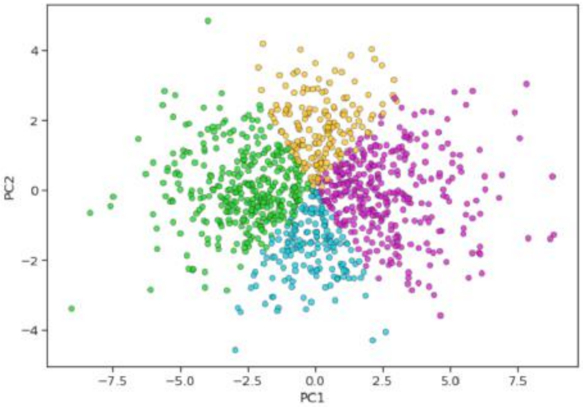


Figure 2: Transformed data into 2D using PCA

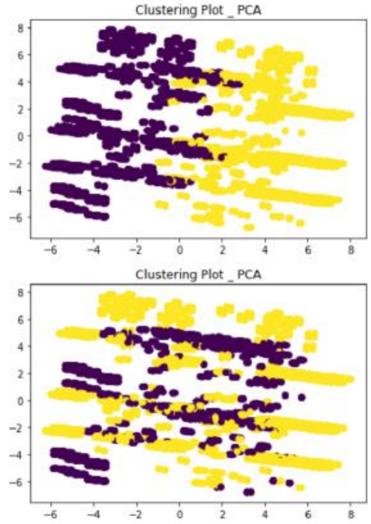


Figure 3: kmeans

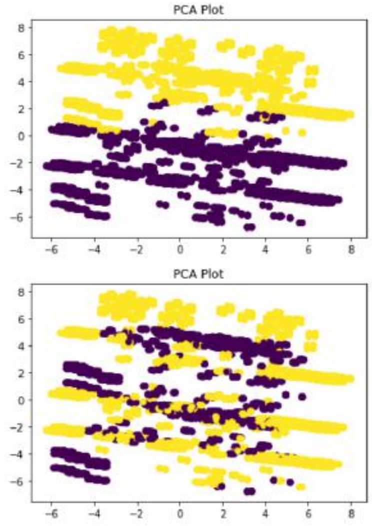


Figure 4: Kmedoids