In the name of God

Machine Learning Course (Spring 2021)

Assignment #4: Kernel KNN and KSPCA

Due date: 10th of Khordad

In this assignment, you are to implement kernel k-nearest neighbor classifier. Recall that in 1NN classifier ($k = 1$), we just need to compute the Euclidean distance of a test instance to all the training instances, find the closest training instance, and find its label. The label of the nest instance will be identical to its nearest neighbor. This can be kernelized by observing that:

$$\|x_i - x_{i'}\|_2^2 = < x_i, x_i > + < x_{i'}, x_{i'} > - 2 < x_i, x_{i'} > = K(x_i, x_i) + K(x_{i'}, x_{i'}) - 2K(x_i, x_{i'})$$

This allows us to apply the nearest neighbor classifier to structured data objects.

Implement the KNN classifier and kernel KNN classifier with Linear, RBF (tune the σ parameter with cross-validation), Polynomial ($d = 1$), Polynomial ($d = 2$), and Polynomial ($d = 3$) kernels. Report the accuracy of classification for each data set with each classifier and compare the results. Split the data set into 70% and 30% for training and testing parts. You should report the mean of accuracies for 10 individual runs. Report the running time of your code (in seconds) in the second table.

You should report the results of your implementation in the following tables. In other words, please copy the following tables in your report and fill them with your results.

Datasets	**Accuracy of Algorithms**					
	1NN	1NN+Linear kernel	1NN+RBF kernel	1NN+Polynomial kernel ($d = 1$)	1NN+Polynomial kernel ($d = 2$)	1NN+Polynomial kernel ($d = 3$)
Wine						
Glass						
BreastTissue						
Diabetes						
Sonar						
Ionosphere						

Datasets	**Runing Time of Algorithms**					
	1NN	1NN+Linear kernel	1NN+RBF kernel	1NN+Polynomial kernel ($d = 1$)	1NN+Polynomial kernel ($d = 2$)	1NN+Polynomial kernel ($d = 3$)
Wine						
Glass						
BreastTissue						
Diabetes						
Sonar						
Ionosphere						

In the name of God

Machine Learning Course (Spring 2021)

Assignment #4: Kernel KNN and KSPCA

In the second part of the assignment, you should implement the Kernel Principal Component Analysis or KSPCA algorithm [1]. Implement the KSPCA algorithm according to the following pseudo-code.

Algorithm (*Kernel supervised PCA*).

Input: Kernel matrix of training data, **K**, kernel matrix of testing data, $\mathbf{K}_{test}$, kernel matrix of target variable, **L**, training data size, **n**.
Output: Dimension reduced training and testing data, **Z** and **z**.
1: $H \leftarrow I - n^{-1}\mathbf{e}\mathbf{e}^T$
2: $Q \leftarrow KHLHK$
3: **Compute basis:** $\beta \leftarrow$ generalized eigenvectors of (Q, K) corresponding to the top d eigenvalues.
4: **Encode training data:** $Z \leftarrow \beta^T[\Phi(X)^T\Phi(X)] = \beta^T K$
5: **Encode test example:** $\mathbf{z} \leftarrow \beta^T[\Phi(X)^T\Phi(\mathbf{x})] = \beta^T K_{test}$

Consider data matrix $X \in R^{p\times n}$ (p is the dimensionality of the data in original space and n is the number of training samples) and labels Y. All you need is to compute delta kernel L, matrix H ($H = I - \frac{1}{n}ee^T$ in which e is a vector of all ones, $e = [1,1,1,\dots,1]^T$), and kernel matrix K. I is the identity matrix.

- L is a $n \times n$ delta kernel over Y (labels), compute L with the following equation.

$$L(y_i, y_j) = \begin{cases} 1 & if\ y_i = y_j \\ 0 & otherwise \end{cases}$$

  For example, in a 2-class problem, if there are 5 data points such that the first 3 data points belong to class 1 and the fourth and the fifth data points are from class 2, $L(y, y')$ can be formed as follows:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

- K is a $n \times n$ RBF kernel over samples in X_{train}, compute K with the following equation.

$$K_{RBF}(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$$

- Split the data set into 70% and 30% for training and testing parts. K_{test} is a kernel over X_{train} and X_{test}. For example, if we have 70 samples in X_{train} and 30 samples in X_{test}, K_{test} is a matrix of 70×30, and K is a matrix of 70×70 over X_{train}.
- β can be computed via eigs command in Matlab or eigh in Python (scipy.linalg).

In the name of God
Machine Learning Course (Spring 2021)
Assignment #4: Kernel KNN and KSPCA

- Once you find the generalized eigenvectors of Q and K, select the first 2 columns ($d = 2$) and put them in β. Then you can find z and Z with matrix multiplication. z and Z are the test and train samples after projection.
- Provide scatter plots of all the datasets (in KSPCA folder) in original space and after projection. In other words, you should plot X_{train} and X_{test} in one plot and z and Z in another plot. Use different colors and symbols for different classes and different sets (train and test). You have 4 data sets, so you need to turn in 8 plots.
- For σ in RBF kernel, you should select it from the $\{0.1, 0.2, 0.3, \dots, 1\}$ set. Select the σ which has the best separation between classes (tune it empirically).

Questions:

- Why do we use kernels in different algorithms?
- Which kernel had the best results? Why?
- Which kernels had the same performance? Why?
- What is kernel trick? Why do we use kernels in learning algorithm?
- What is the difference between Delta and RBF kernels? Explain about the properties of these kernels.

Important Notes:

- You need to implement the main algorithms from scratch. Feel free to use your preferred programming languages.
- Pay extra attention to the due date. It will not be extended. Be advised that submissions after the deadline would not grade.
- Provide a report for your assignment and explain your code and report the results in each part.
- The name of the uploading file should be your **Lastname_Firstname.**
- Using other students' codes, using the codes available on the internet, or pay someone to write the code for you will lead to zero grades.
- Try to write your code efficiently.

Reference

[1] Barshan, E., Ghodsi, A., Azimifar, Z., & Jahromi, M. Z. (2011). Supervised principal component analysis: Visualization, classification and regression on subspaces and submanifolds. Pattern Recognition, 44(7), 1357-1371.