MLSM 2021 - Quiz 4
02/11/2021 - 10/11/2021 at 23:59

Questions based on Lecture 6 and 7

(1) (1.0 pt.)

In Lecture 7, Kernel methods, it is discussed how to build new kernels from known ones, see Slide "Several ways to get to a kernel". Assume that we have a kernel function $\kappa(.,.)$ which satisfies the following condition $\kappa(\mathbf{x},\mathbf{z}) \leq \rho < 1$ for any pair $\mathbf{x}$ and $\mathbf{z}$. Let a new function be $\kappa^*(\mathbf{x},\mathbf{z}) = 1/(1-\kappa(\mathbf{x},\mathbf{z}))$. What could we say about the function κ^*?

Hint: Think about which of the kernel constructing operations might relate to this question. Could the new kernel be transformed into a sequence?

(1) κ^* can not be expressed by the kernel constructing operations.
(2) κ^* is not positive(semi) definite.
(3) κ^* is a valid positive(semi) definite kernel.
(4) κ^* is a valid kernel positive(semi) definite kernel if $\rho > 1$ only.

(2) (1.0 pt.)

A Gaussian(RBF) kernel has this form $\kappa_{rbf}(\mathbf{x},\mathbf{z}) = \exp(-||\mathbf{x}-\mathbf{z}||^2/(2\sigma^2))$. Let $\kappa(\mathbf{x},\mathbf{z}) = \langle\phi(\mathbf{x}),\phi(\mathbf{z})\rangle$ be any valid kernel with feature map ϕ.

Let us try to construct a new kernel out of the Gaussian one, namely we have $\kappa_{rbf\_mod}(\mathbf{x},\mathbf{z}) = \exp(-(\kappa(\mathbf{x},\mathbf{x}) + \kappa(\mathbf{z},\mathbf{z}) - 2\kappa(\mathbf{x},\mathbf{z}))/(2\sigma^2))$. Which of these statements is true?

Hint: How is the Gaussian kernel function defined? How is it built on the underlying Hilbert space?

(1) κ_{rbf_mod} is only a valid kernel if κ is a Gaussian one.
(2) κ_{rbf_mod} is a valid, positive semi-definite kernel for any κ.
(3) There are cases where κ_{rbf_mod} is not positive semi-definite.

(3) (1.0 pt.) Let a polynomial kernel, $\kappa_{pol}(\mathbf{x},\mathbf{z})$ be represented explicitly $(\mathbf{x}^T\mathbf{z})^q$ where q is the degree. What is the dimension of the explicit feature vector? Assume that the degree of the polynomial is 3, and the polynomial kernel is defined on the vector space of dimension 3.

Be careful, there is no constant term in this kernel definition! How can the feature dimension change if the constant is not included?

(1) 15
(2) 20
(3) 10
(4) 30

(4) (2.0 pt.)

In this question the behavior of algorithms developed to solve the Support Vector Machine problem are compared. The comparison is carried out on the breast cancer dataset of the sklearn package,

`from sklearn.datasets import load_breast_cancer`.

The labels of the Breast Cancer dataset are of $\{0, 1\}$ which need to be converted into $\{-1, +1\}$. Scale each of the input variables to have the maximum absolute value equal to 1. For example, these steps can be implemented by

```
import numpy as np
from sklearn.datasets import load_breast_cancer
from sklearn.svm import SVC
from sklearn.metrics import roc_auc_score

# load the data
X, y = load_breast_cancer(return_X_y=True)  ## X input, y output
print(X.shape, y.shape)

## to convert the {0,1} output into {-1,+1}
y = 2 * y -1

## X is the input matrix
mdata,ndim = X.shape
## normalization by L infinity norm
X/= np.outer(np.ones(mdata),np.max(np.abs(X),0))

## number of iteration
niter = 10

## penalty constant for the of the Stochastic Dual Coordinate Ascent algorithm
C = 1000
```

Three methods need to be applied. Those methods are the following ones:

- The "Stochastic Dual Coordinate Ascent for SVM" algorithm which is described in the lecture slides. In this algorithm, use the linear kernel on the top of the input data. The penalty constant, C, is equal to 1000. Repeat 10 times the algorithm on the full training dataset.
- The method *sklearn.svm.SVC* from the sklearn package with polynomial kernel, denoted by "poly" in the sklearn. The degree of the polynomial is 5, and the penalty constant, C, here is equal to 1, the default.
- The method *sklearn.svm.SVC* again from the sklearn package with Gaussian kernel, denoted by "rbf" in the sklearn. The "gamma" parameter of the rbf kernel is set to "scale", and the penalty constant, C, here is also equal to 1, the default.

Process the data in the original order of the examples appearing in the data set. No training and test split is applied. The original full data is used in the training and in the test phases as well.

Compute the Area Under the Curve(AUC) score for the three versions of the SVM solution methods, thus for the "Stochastic Dual Coordinate Ascent for SVM"', the sklearn.svm.SVC(kernel = "poly") and the sklearn.svm.SVC(kernel = "rbf"). Let the short names of the methods be SDCA, SVCPoly, SVCRbf.

Question: What is the order of the performances of the methods measured by Area Under the Curve(AUC)?

Hint: You might apply the `sklearn.metrics.roc_auc_score` function to compute the AUC score. The sklearn also contains other alternative approaches.

(1) $SDCA < SVCPoly < SVCRbf$
(2) $SDCA < SVCRbf < SVCPoly$
(3) $SVCRbf < SVCPoly < SDCA$
(4) $SVCRbf < SDCA < SVCPoly$