MLSM 2023 - Quiz 4

24/10/2023 - 01/11/2023 at 23:59

Questions based on Lecture 4, 6 and 7

(1) (1.0 pt.) Let a polynomial kernel, $\kappa_{pol}(\mathbf{x}, \mathbf{z})$ be represented explicitly $(\mathbf{x}^T\mathbf{z} + c)^q$ where q is the degree. Let the dimension of the explicit feature space be 15.

Question: What are the degree of the polynomial and the dimension of that vector space on which the polynomial kernel is defined?

Enumerate all possible pairs with the explicit feature space dimension 15. Write the pairs in this form:

`(degree of the polynomial, dimension of the vector space on which the polynomial defined)`

(1) (2,4), (4,2)
(2) (3,3)
(3) (4,3), (3,5)
(4) (2,2), (3,3)

(2) (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 function $\kappa(x, x') = \sum_{n=0}^{\infty}\left(\frac{\cos(\alpha(\mathbf{x}, \mathbf{x}'))}{\lambda}\right)^n$ where α is the angle between the two vectors, $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^d$ where d is a fixed dimension, and λ is a real number.

Question: Does the function κ define a positive (semi)definite kernel?

Hint: Think about which of the kernel constructing operations might relate to this question.

(1) κ is not positive (semi)definite kernel.
(2) κ is a valid positive(semi)definite kernel if $\lambda < 1$.
(3) κ is a valid positive (semi)definite kernel if $\lambda > 1$.
(4) κ is a valid positive (semi)definite kernel for any λ.

(3) (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 $\{p(\cdot|\boldsymbol{\theta})\boldsymbol{\theta} \in \Theta\}$ be a parametric family of probability density functions defined on a fixed data space $\mathcal{X} \subset \mathbb{R}^m$ and on the parameter space $\Theta \subset \mathbb{R}^n$.

Let the feature vector be defined by $\mathbf{f}(\mathbf{x}|\tilde{\boldsymbol{\theta}}) = \nabla_\theta p(\mathbf{x}|\boldsymbol{\theta})|_{\theta=\tilde{\theta}}$, which is the gradient of the density function with respect to the parameter vector at a fixed parameter $\tilde{\boldsymbol{\theta}}$. Here we assume that the gradient of p exists for all θ.

Now let us try to construct a new kernel by combining the Gaussian kernel with the gradient of the probability density function $\kappa_{rbf\_mod}(\mathbf{x}, \mathbf{x}') = \exp(-(||\mathbf{f}(\mathbf{x}|\tilde{\boldsymbol{\theta}}) - \mathbf{f}(\mathbf{x}'|\tilde{\boldsymbol{\theta}})||^2)/(2\sigma^2))$.

Question: Which of these statements is true?

Hint: How is the Gaussian kernel function defined on the underlying feature space? What is the space of the gradients of a given function?

(1) κ_{rbf_mod} is only a valid positive (semi)definite kernel if the density function p is a Gaussian one.
(2) κ_{rbf_mod} is a valid, positive (semi)definite kernel for any density function.
(3) There are cases where κ_{rbf_mod} is not positive (semi)definite independently from the density function.

(4) (2.0 pt.)

The task in this question is to implement the complete nested cross-validation procedure on a hyper-paramer. See the details in Section "Cross-validation" of Lecture 4

The methods applied in the cross-validation are algorithms developed to solve the Support Vector Machine problem. The dataset is the breast cancer dataset of the sklearn package, The labels of the Breast Cancer dataset are of $\{0, 1\}$ which need to be converted into $\{-1, +1\}$, and scale each of the input variables to have the maximum absolute value equal to 1 similarly to Question 3 of Quiz 3.

Three methods need to be applied:

- The "Stochastic gradient descent algorithm for SVM" algorithm which is described in Lecture 6 discussing the SVM method. The kernel applied is linear. It is called later on as "primal" method.
- The method *sklearn.svm.SVC* which is taken from from the sklearn package, and the linear kernel is used here as well. It is called as "svc" method.

The example code `quiz4_question_4_hint.py` contains a template, a detailed skeleton of the cross-validation method. In the code the comments show those points where you need to include the concrete realization of the primal method, and compute the corresponding training, test and evaluation for both the primal and the svc methods.

In the implementation of the primal method, process the data in the original order of the examples appearing in the dataset. This is indicated in the example code.

In the cross-validation the best C hyper-parameter which controls the balance between the complexity model and the empirical error is looked for.

Here a summary of the procedure is described, all the details are given in the example code.

In the cross-validation you can apply three nested loops:

- Outer loop; which selects the training and the test sets.
- The hyper-parameter selecting loop; which enumerates the values of hyperparameter C. Those values are given by the list "lC" in the example code.
- Inner loop; in which the training is split into inner training and validation set.

The number of folds for both the outer and inner loop is given in the example code.

The methods are evaluated in the following way:

- In the inner loop, the F1 score is computed on all validation sets and for both methods for a fixed C value.
- After the inner loop, the mean of the F1 score is computed on the validation sets for both methods.
- After the hyper-parameter selecting loop, that C is selected for each method which gives the highest mean F1 score.
- In the outer loop, the training and the prediction are executed on the best C parameters selected for both methods.
- Compute the F1 score for both methods and for all of the 5 test sets.
- After the outer loop compute the mean and the standard deviation of the F1 scores corresponding to the 5 test sets for both methods.

Check carefully the example code which shows the nested loops and the points where the evaluation steps need to be implemented.

Question: What is the ratio of the standard deviations of the F1 scores computed on the tests sets in the outer loop. The ratio is computed in this way:

`standard deviation of the primal / standard deviation of the svc.`

(1) 0.84
(2) 1.22
(3) 2.34
(4) 1.49