NCTU Pattern Recognition, Homework 4

Deadline: June 12, 23:59

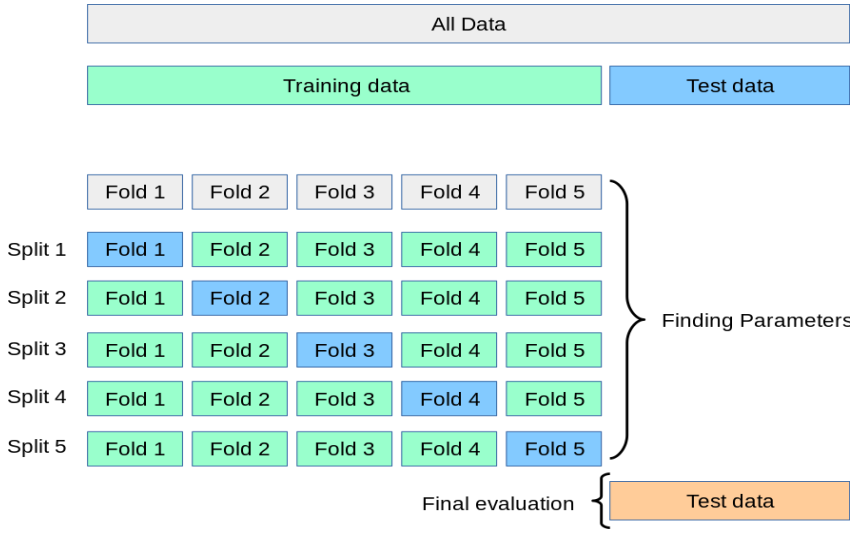
Part. 1, Coding (80%):

In this coding assignment, you need to implement the cross-validation and grid search by using only NumPy, then train the SVM model from scikit-learn on the provided dataset and test the performance with testing data. Find the sample code and data on the GitHub page https://github.com/NCTU-VRDL/CS_DCP3121/tree/master/HW4

Please note that only NumPy can be used to implement cross-validation and grid search. You will get no points by simply calling sklearn.model_selection.GridSearchCV.

1. (10%) K-fold data partition: Implement the K-fold cross-validation function. Your function should take K as an argument and return a list of lists (*len(list) should equal to K*), which contains K elements. Each element is a list contains two parts, the first part contains the index of all training folds (index_x_train, index_y_train), e.g. Fold 2 to Fold 5 in split 1. The second part contains the index of validation fold, e.g. Fold 1 in split 1 (index_x_val, index_y_val)

   Note: You need to handle if the sample size is not divisible by K. Using the strategy from sklearn. The first `n_samples % n_splits` folds have size `n_samples // n_splits + 1`, other folds have size `n_samples // n_splits`, where `n_samples` is the number of samples, `n_splits` is K, `%` stands for modulus, `//` stands for integer division. See this post for more details
   Note: Each of the samples should be used **exactly once** as the validation data
   Note: Please shuffle your data before partition

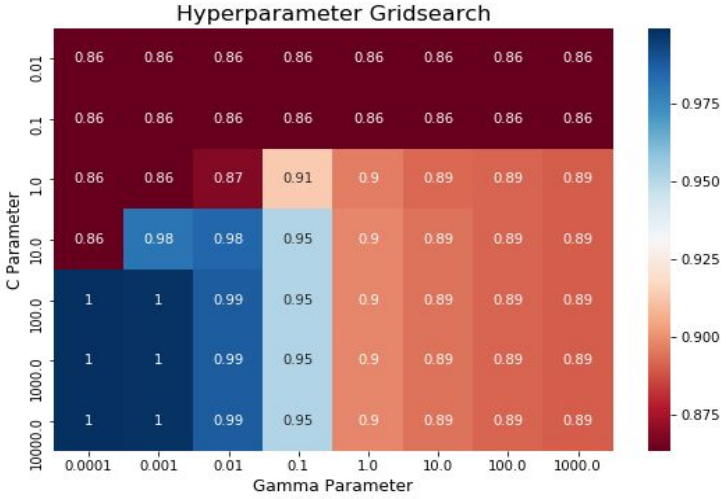


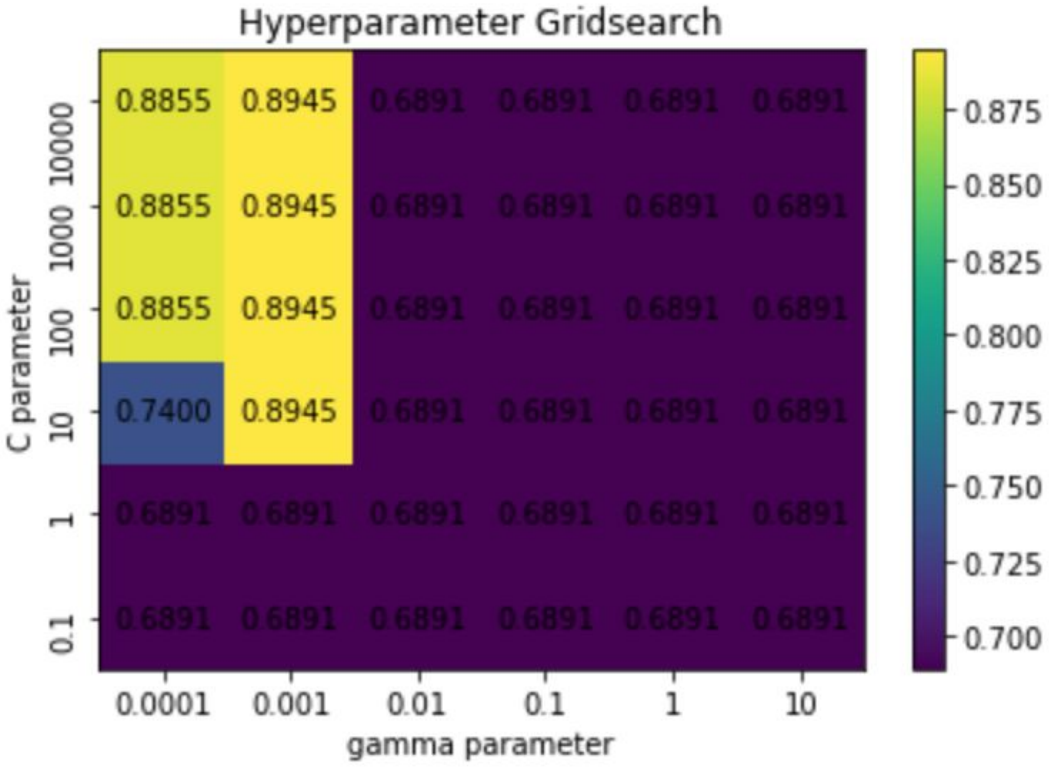
2. (30%) Grid Search & Cross-validation: using sklearn.svm.SVC to train a classifier on the provided train set and conduct the grid search of "C" and "gamma", "kernel'='rbf"

to find the best hyperparameters by cross-validation. Print the best hyperparameters you found.
Note: We suggest use K=5
The best parameters are C = 10000.0, gamma = 0.0010

3. (10%) Plot the grid search results of your SVM. The x, y represents the hyperparameters of "gamma" and "C", respectively. And the color represents the average score of validation folds.

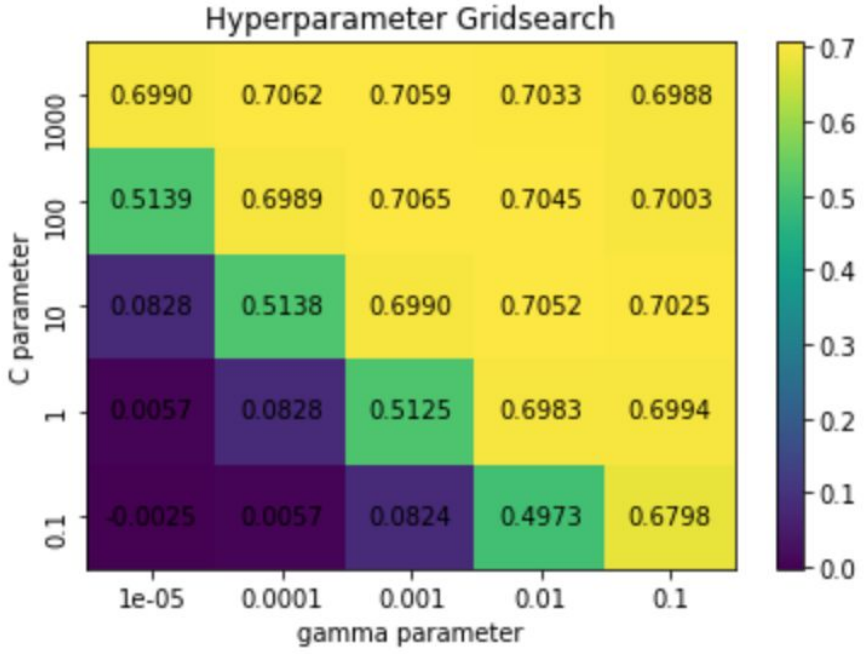
Note: This image is for reference, not the answer
Note: matplotlib is allowed to use





4. (15%) Train your SVM model by the best hyperparameters you found from question 2 on the whole training set and evaluate the performance on the test set.
Note: Your accuracy scores should be higher than 0.85
Accuracy score: 0.8958333333333334

5. (15%) Consider the dataset used in HW1 for regression. Please redo the above questions 2 ~ 4 with the dataset replaced by that used in HW1, while the task is changed from classification to regression. You should use the SVM regression model RBF kernel with grid search for hyperparameters and K-fold cross-validation (you can use any K for cross-validation). Then compare the linear regression model you have implemented in HW1 with SVM by showing the Mean Square Errors of both models on the test set.
The best parameters are C = 100.0, gamma = 0.0010



Square error of Linear regression: 0.06870743256403333
Square error of SVM regresssion model: 0.07332255968213075

Part. 2, Questions (20%):

1. Given a valid kernel $k_1(x, x')$, prove that 1) $k(x, x') = ck_1(x, x')$ and 2) $k(x, x') = f(x)\, k_1(x, x')\, f(x')$ are valid kernels, where $c > 0$ is a positive constant and $f(\cdot)$ is any real-valued function.

1) $\because k_1(x,x')$ is a valid kernel

$\therefore k_1(x,x') = \phi(x) \cdot \phi(x')$

$\phi'(x) = \sqrt{c}\,\phi(x)$

$\Rightarrow k(x,x') = c k_1(x,x') = \phi'(x) \cdot \phi'(x')$

2) $k(x,x') = f(x) k_1(x,x') f(x')$

$= f(x) f(x') \phi(x) \cdot \phi(x')$

$= (f(x)\phi(x)) \cdot (f(x')\phi(x'))$

$= \phi'(x) \cdot \phi''(x')$