KYOTO UNIVERSITY

Statistical Machine Learning Theory

Model Evaluation and Selection

Hisashi Kashima
kashima@i.Kyoto-u.ac.jp

DEPARTMENT OF INTELLIGENCE SCIENCE AND TECHNOLOGY

Model evaluation and selection framework: We want to predict final performance of models

- We are interested in the future performance of the obtained model when it is deployed
  - Model performance for training data and that for future data are different
- We often have some hyper-parameters to be tuned so that the final performance gets better
  - E.g. Training target of the ridge regression:

$$\text{minimize}_{\mathbf{w}} \ \|\mathbf{y} - \boldsymbol{X}\mathbf{w}\|_2^2 + \lambda\|\mathbf{w}\|_2^2$$

Hyperparameter

  - Hyper-parameters are not optimized in the training

The first principle:
Evaluation must use a dataset not used in training

- You *must not* evaluate your classifier based on the performance on the dataset you already used for training
- Usually, a given dataset must be divided into a *training dataset* and a *test dataset*
  1. Train a classifier using the training dataset
  2. Evaluate its performance on the test dataset
  - Partitioning should be done carefully
    - Time series data: if your dataset explicitly has time stamps and you are interested in predicting the future, you should divide it into past and future

A statistical framework for performance evaluation: Cross validation

- (K-fold) cross validation gives an estimate of the future performance of the classifier when it is deployed
- Divide a given dataset into K non-overlapping sets
  - Use $K - 1$ of them for training
  - Use the remaining one for testing
- Changing the test dataset results in K measurements
  - Take their average to get a final performance estimate

Statistical framework for tuning hyper-parameters: Cross validation (for model selection)

- Most of machine learning algorithms have hyper-parameters
  - Hyper-parameters are not automatically tuned in the training phase and must be given by users
- (K-fold) cross validation can also be used for this purpose:
  - Use $K-1$ of K sets for training models for various hyper-parameter settings
  - Use the remaining one for testing
  - Choose the hyper-parameter setting with the best averaged performance
    - Note that this is **NOT** the estimate of its final performance

Double-loop cross validation: Tuning hyper-parameters and performance evaluation at the same time

- Sometimes you want to do *both* hyper-parameter tuning and estimation of future performance
- Doing both with one K-fold cross validation is guilty 
  - You saw the test dataset for tuning hyper-parameters
- Double-loop cross validation:
  - Outer loop for performance evaluation
  - Inner loop for hyper-parameter tuning
  - High computational costs…

A simple alternative of double-loop cross validation: "Development set" approach

- A simple alternative for the double-loop cross validation
- "Development set" approach
  - Use $K-2$ of K sets for training
  - Use one for tuning hyper-parameters
  - Use one for testing