MYONGJI UNIVERSITY

Artificial Intelligence

Lecture 9. K-fold Cross Validation

Spring 2022

Prof. Jonghoon Chun, Ph.D.

E-mail : jchun@mju.ac.kr
Lecture Note : http://lms.mju.ac.kr

Agenda

- K-fold cross validation 개념
- 예제

K-FOLD CROSS VALIDATION

Model Evaluation and Selection

- Evaluation metrics: How can we measure accuracy?
- Use **validation test set** of class-labeled tuples instead of training set when assessing accuracy
- Methods for estimating a classifier's accuracy:
  - Holdout method, random subsampling
  - Cross-validation
  - Bootstrap

Confusion Matrix

- Confusion matrix

Actual class\Predicted class	C_1	¬ C_1
C_1	**True Positives (TP)**	**False Negatives (FN)**
¬ C_1	**False Positives (FP)**	**True Negatives (TN)**

- Example

Actual class\Predicted class	buy_computer = yes	buy_computer = no	Total
buy_computer = yes	**6954**	**46**	7000
buy_computer = no	**412**	**2588**	3000
Total	7366	2634	10000

Evaluation Metrics

- Classifier Accuracy, or recognition rate: percentage of test set tuples that are correctly classified

  Accuracy = (TP + TN)/All

- Error rate: *1 – accuracy*, or

  Error rate = (FP + FN)/All

- Sensitivity: True Positive recognition rate

  Sensitivity = TP/P

- Specificity: True Negative recognition rate

  Specificity = TN/N

Evaluation Metrics

- Precision: exactness – what % of tuples that the classifier labeled as positive are actually positive

$$precision = \frac{TP}{TP + FP}$$

- Recall: completeness – what % of positive tuples did the classifier label as positive?

$$recall = \frac{TP}{TP + FN}$$

- Inverse relationship between precision & recall
- F measure (F_1 or F-score): harmonic mean of precision and recall,

$$F = \frac{2 * precision * recall}{precision + recall}$$

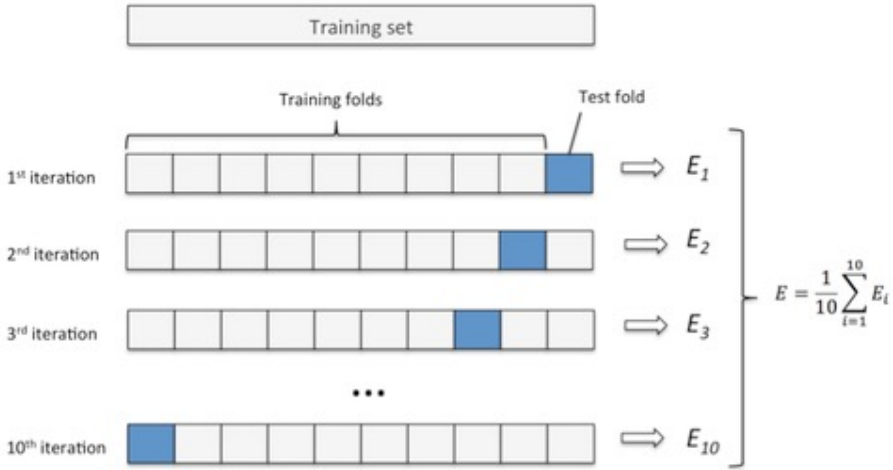
Evaluation Metrics – An Example

Actual Class\Predicted class	cancer = yes	cancer = no	Total	Recognition(%)
cancer = yes	**90**	**210**	300	30.00 (*sensitivity)*
cancer = no	**140**	**9560**	9700	98.56 (*specificity)*
Total	230	9770	10000	96.40 (*accuracy*)

- *Precision* = 90/230 = 39.13% *Recall* = 90/300 = 30.00%

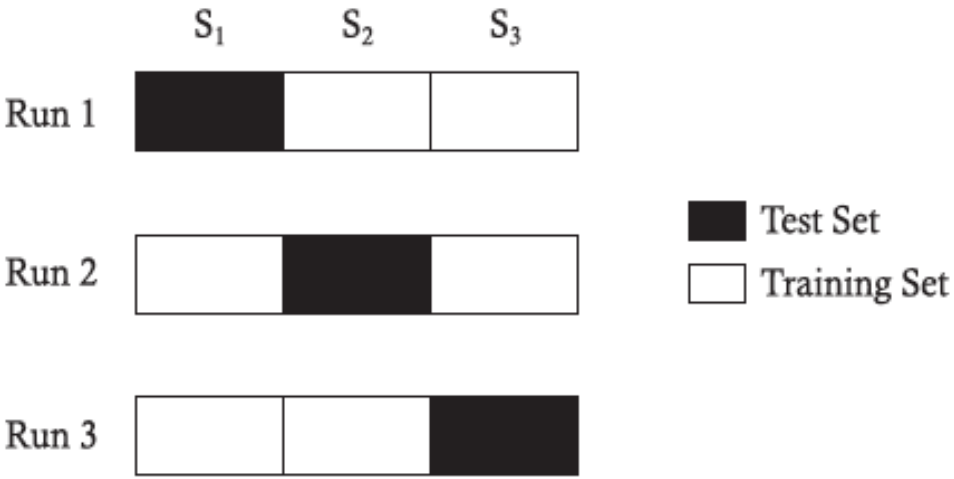
K-fold Cross Validation

- **Cross-validation** (*k*-fold, where k = 10 is most popular)
  - Randomly partition the data into *k mutually exclusive* subsets, each approximately equal size
  - At *i*-th iteration, use D_i as test set and others as training set
  - Leave-one-out: *k* folds where *k* = # of tuples, for small sized data



Cross-validation Example

- 3-fold cross-validation



Holdout Method

- **Holdout Method (Holdout Cross Validation)**
  - Given data is randomly partitioned into two independent sets
    - Reserve k% for training and (100-k)% for testing
    - Training set (e.g., 2/3) for model construction
    - Test set (e.g., 1/3) for accuracy estimation
  - Holdout Cross Validation (by Random subsampling)
    - a variation of holdout
    - Repeat holdout k times
    - accuracy = avg. of the accuracies obtained from k validations

- **Stratified cross-validation**
  - folds are stratified so that class dist. in each fold is approx. the same as that in the initial data

K-FOLD CROSS VALIDATION 예제

K-fold Cross Validation

- sklearn.model_selection.cross_val_score 사용
  - parameters: estimator, data, target, cv
  - estimator: 모델 (estimator to use to fit the data)
  - cv (cross validation): k-fold의 k값 지정, default k = 3
  - https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_val_score.html

- time( )을 사용하여 execution time 측정

```
from time import time
start = time()
 . . .
end = time()
print("Execution time(seconds) :", str(round((end - start), 2)))
```

K-fold Cross Validation 예제

In [1]:
```
import pandas as pd
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn import metrics, model_selection
from time import time

mr = pd.read_csv("mushroom.csv", header = None)

df = pd.DataFrame(mr.iloc[:, 0])
df = df.join(pd.get_dummies(mr.iloc[:, 1:]))

data = df.iloc[:, 1:]
label = df.loc[:, 0]

# 시작 시간을  설정
start = time()

# random vector 갯수를 설정
clf = RandomForestClassifier(n_estimators = 5)
clf.fit(data,label)

# Cross validation 설정, cv = k, k-fold cross validation
scores = model_selection.cross_val_score(clf, data, label, cv = 5)

# 종료 시간을 측정
end = time()

print("Execution time(seconds)", str(round((end - start), 2)))
print("각 validation 정답률 = ", scores)
print("평균 정답률 :", "%.2f"%scores.mean())
```

```
Execution time(seconds) 0.17
각 validation 정답률 =  [0.84246154 1.         0.99261538 1.         0.79248768]
평균 정답률 : 0.93
```

K-fold Cross Validation 예제

In [2]:
```
import pandas as pd
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn import metrics, model_selection
from time import time

mr = pd.read_csv("mushroom.csv", header = None)

df = pd.DataFrame(mr.iloc[:, 0])
df = df.join(pd.get_dummies(mr.iloc[:, 1:]))

data = df.iloc[:, 1:]
label = df.loc[:, 0]

# 시작 시간을  설정
start = time()

# random vector 갯수를 설정
clf = RandomForestClassifier(n_estimators = 100)
clf.fit(data,label)

# Cross validation 설정, cv = k, k-fold cross validation
scores = model_selection.cross_val_score(clf, data, label, cv = 10)

# 종료 시간을 측정
end = time()

print("Execution time(seconds)", str(round((end - start), 2)))
print("각 validation 정답률 = ", scores)
print("평균 정답률 :", "%.2f"%scores.mean())
```

```
Execution time(seconds) 3.54
각 validation 정답률 =  [0.68511685 1.         1.         1.         1.         1.
 1.         1.         0.97044335 1.        ]
평균 정답률 : 0.97
```

MYONGJI UNIVERSITY

END

MYONGJI UNIVERSITY