Model Selection and Evaluation

모델 선택과 평가

Dr. Saerom Park
Statistical Learning and Computational Finance Lab.
Department of Industrial Engineering
psr6275@snu.ac.kr
http://slcf.snu.ac.kr

목차

Reference

- **Reading:** [Raschka. (2017), chapter 6], [Müller. (2016), chapter 5 & 6].

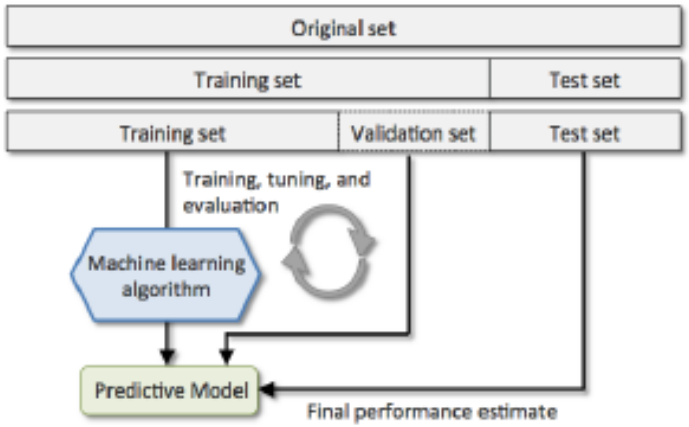
Introduction

- 모델 선택
  - 매개변수 선택: 다양한 모델 파라미터 중 가장 '적합한' 파라미터를 선택
    - Grid search
  - 학습 모델 선택: 학습시에 어떠한 종류의 모델을 사용할 것인지 선택
- 모델 평가
  - 과소 적합 과대 적합 판단: 학습 곡선과 검증 곡선
  - 어떠한 모델이 가장 '적합한' 모델인지를 판단
  - 선택 기준 (Evaluation metrics)
  - 데이터 분할
    - 학습/ 검증/ 시험 데이터
    - K-fold 교차 검증
- Pipeline
  - 데이터 변환 과정과 기계학습 모델을 연결
  - 매개 변수 선택 뿐만 아니라 학습 모델 선택을 위해서도 활용 가능

K-FOLD CROSS VALIDATION

k-fold 교차검증

- Holdout 교차검증
  - 여러 파라미터에 대해 평가할 때 동일한 test set을 이용하여 평가할 경우 그 test set에 overfit하는 model이 선택된다.
  - 마치 test set이 training에 부분적으로 이용되는 효과를 가져온다.
  - 따라서 test set이 아닌 다른 set을 이용하여 검증해야 한다.
  - Training set을 training set과 validation set으로 나눈다.
  - Train + Validation + Test



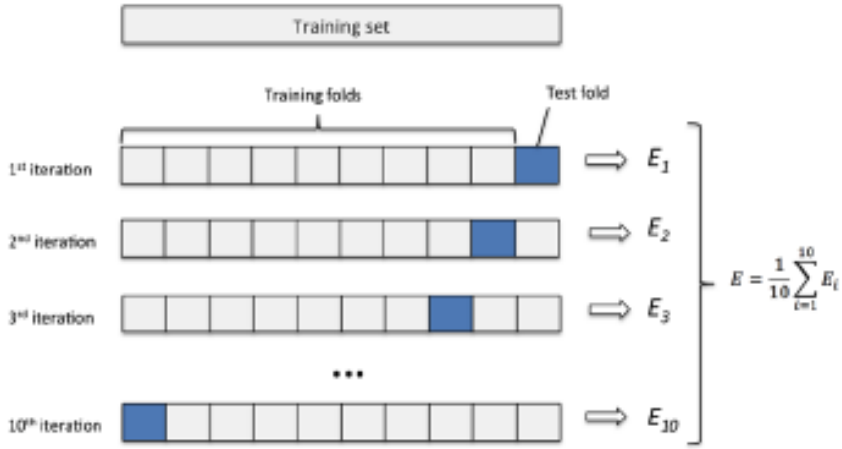
k-fold 교차검증

k-fold 교차검증

- 특정 set에 overfit하는 것을 방지하기 위해 여러 fold로 나누어 검증
- Validation set을 따로 둘 필요가 없음
- `from sklearn.model_selection import cross_val_score`
- `scores = cross_val_score(estimator=pipe_lr, X=X_train, y=y_train, cv=10, n_jobs=1)`

http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_val_score.html

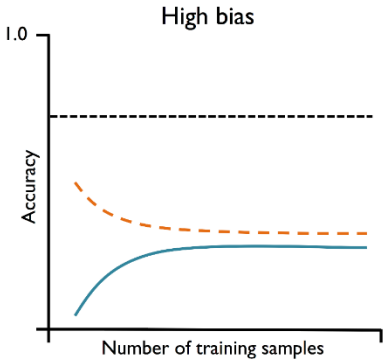
Parameters	
cv	k에 해당하는 교차검증 fold수
n_jobs	병렬처리 방법

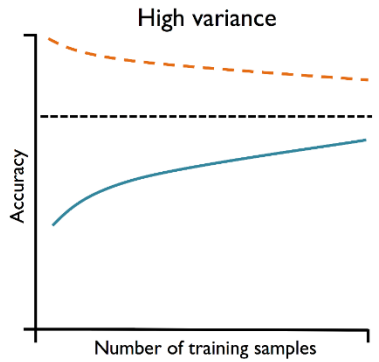


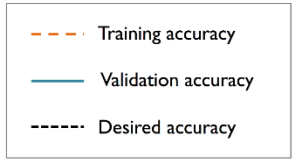
학습곡선과 검증곡선

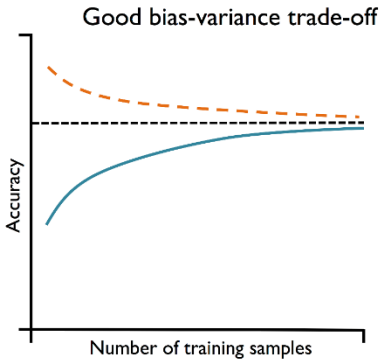
학습곡선과 검증곡선

- 학습곡선을 통해 underfitting인지 overfitting인지 판단할 수 있다.
- Underfitting의 해결방법
  - 추가 feature 수집
  - Regularization정도 낮춤
  - 파라미터 수 증가
  - 모델 복잡도 높임
- Overfitting의 해결방법
  - Data 더 수집
  - 차원축소
  - Feature 수 줄임
  - Regularization정도 높임
  - 파라미터 수 줄임
  - 모델 복잡도 낮춤







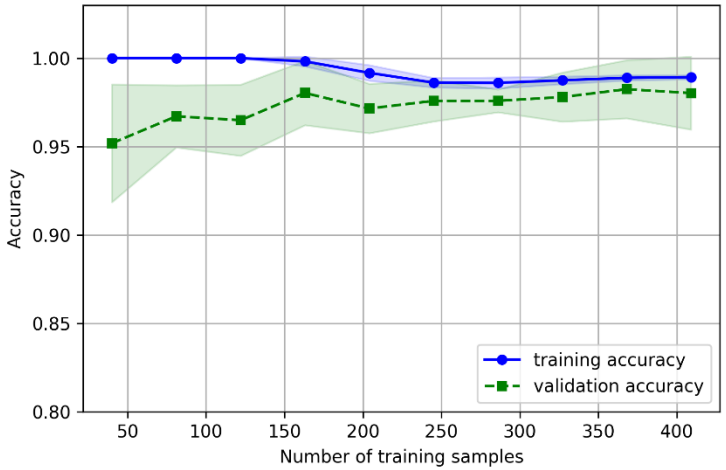


학습곡선과 검증곡선

- 학습곡선 예시

  - ```
    train_sizes, train_scores, test_scores =
    learning_curve(estimator=pipe_lr, X=X_train, y=y_train,
    train_sizes=np.linspace(0.1, 1.0, 10), cv=10, n_jobs=1)
    ```

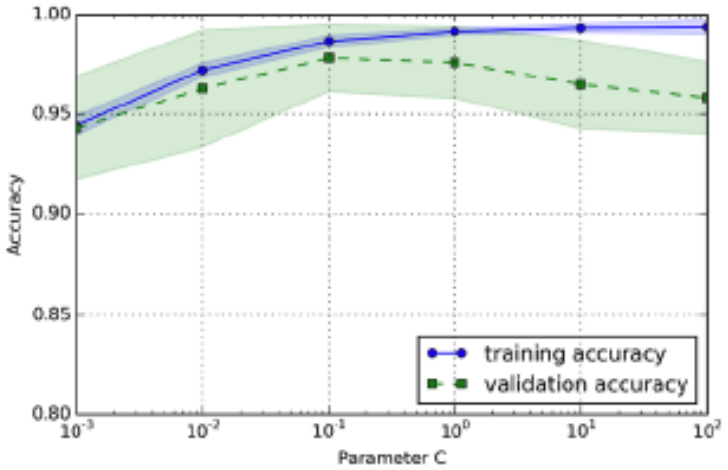
http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.learning_curve.html


    ```

학습곡선과 검증곡선

- 검증곡선
  - 파라미터에 따른 training accuracy와 validation accuracy를 보고 가장 적합한 파라미터를 선택
  - `train_scores, test_scores = validation_curve(estimator=pipe_lr, X=X_train, y=y_train, param_name='logisticregression__C', param_range=param_range, cv=10)`

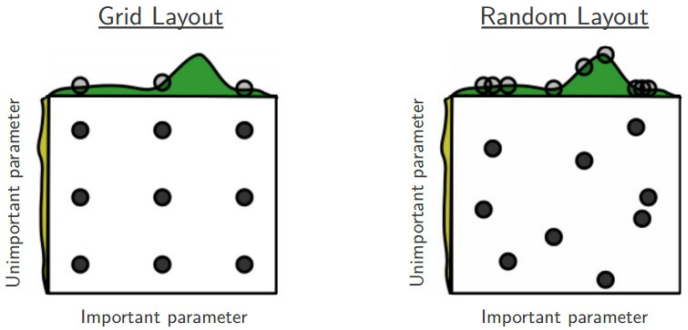
http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.validation_curve.html



GRID SEARCH

그리드 검색

- 그리드 검색
  - 그리드를 만들어서 모든 경우의 수에 대해 교차검증
  - Random layout은 변수들 간에 의존성을 제거해 더 다양한 변수 탐색 가능
  - ```
    param_grid = [{'svc__C': param_range,
                   'svc__kernel': ['linear']},
                  {'svc__C': param_range,
                   'svc__gamma': param_range,
                   'svc__kernel': ['rbf']}]
    ```




  - ```
    gs = GridSearchCV(estimator=pipe_svc, param_grid=param_grid,
    scoring='accuracy', cv=10, n_jobs=-1)
    ```

http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

PERFORMANCE MEASURES

성능 메트릭

분류

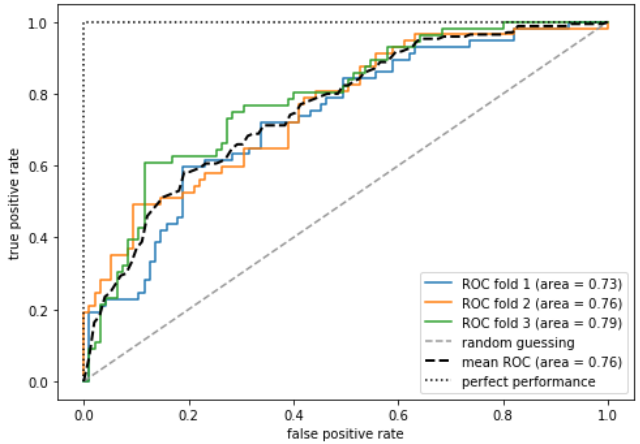
- 오차 행렬(Confusion matrix)
  - 진양성(TP), 진음성(TN), 가양성(FP), 가음성(FN)을 나타낸 행렬
  - 다중 분류일 경우 각 행은 정답 레이블이며 열은 예측 레이블

		Predicted class: P	Predicted class: N
Actual Class	P	True Positives (TP)	False Negatives (FN)
	N	False Positives (FP)	True Negatives (TN)

- 정확도(Accuracy) $= \frac{TP+TN}{TP+TN+FP+FN}$
- 정밀도(Precision) $= \frac{TP}{TP+FP}$: 거짓 양성의 수를 줄이는 것이 목표 ↑
- 재현율(Recall) $= \frac{TP}{TP+FN}$: 모든 양성 샘플을 식별해야 할 때 ↑
- $F1 = \frac{2}{\frac{1}{Precision}+\frac{1}{Recall}}$: 정밀도와 재현율 요약 ↑

ROC 곡선

- 여러 임계값에서 분류기의 특성을 분석
  - TPR = 재현율, FPR $= \frac{FP}{FP+TN}$
  - 왼쪽 위로 가까울수록 이상적
  - AUC(Area Under Curve)



PIPELINE

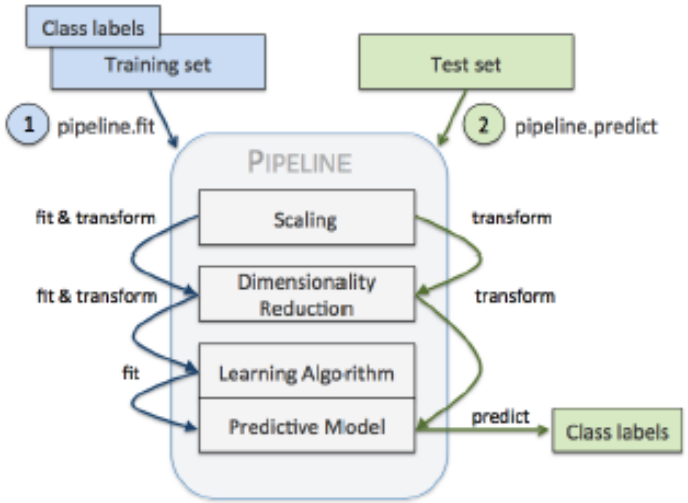
파이프라인

- 파이프라인 = 전처리 + 학습 + 예측
  - 여러 처리 단계를 하나의 scikit-learn 추정기 형태로 묶어주는 파이썬 클래스
    - cross_val_score나 GridSearchCV에 파이프라인을 하나의 추정기 처럼 사용할 수 있음

```
from sklearn.pipeline import make_pipeline
pipe_lr = make_pipeline(StandardScaler(), PCA(n_components=2),
LogisticRegression(random_state=1))
pipe_lr.fit(X_train, y_train)
y_pred = pipe_lr.predict(X_test)
```

- Pipeline의 각 step은 transform과 fit method

http://scikit-learn.org/stable/modules/generated/s



Model Selection 실습

- WDBC datasets
  - LDA
    - classification수행
    - 가장 좋은 dimension을 Grid search 를 통해 찾기
    - 다양한 performance metrics로 성능 비교
  - Logistic regression
    - Classification 수행
    - No-regularization, L1, L2 regularization에 대해서 가장 좋은 경우와 가장 좋은 parameter 찾기
    - Pipeline을 통해 PCA도 함께 수행하여 가장 좋은 차원도 함께 찾기
    - 다양한 performance metrics로 최종적으로 구한 모델의 성능 비교

Reference

- [Raschka. (2017)] Raschka, Sebastian, and Vahid Mirjalili. *Python machine learning*. Packt Publishing Ltd, 2017.
- [Müller. (2016)] Müller, Andreas C., and Sarah Guido. *Introduction to machine learning with Python: a guide for data scientists*. 2016.
- [GÉRON. (2017)] GÉRON, Aurélien. *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems.* Sebastopol, CA: O, 2017.