Ensemble Learning
앙상블 학습

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

Table of Contents

- 목차

Reference

- **Reading:** [Raschka. (2017), chapter 7], [GÉRON. (2017), chapter 7].

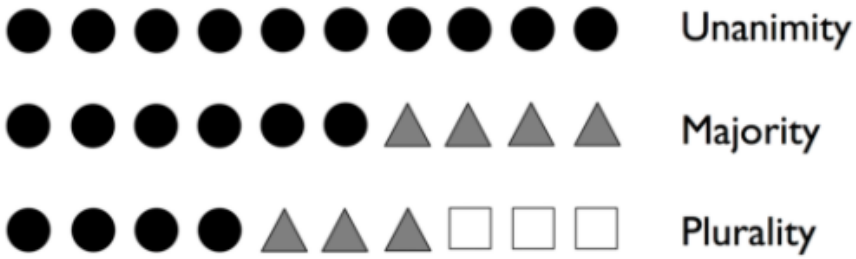
Introduction

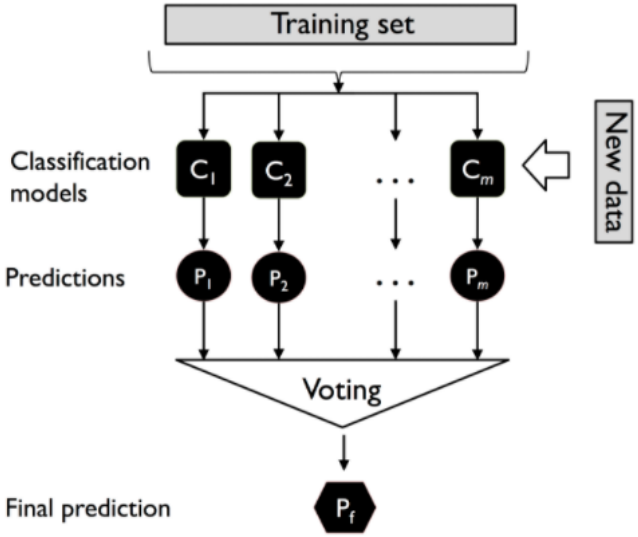
- Ensemble method
  - 서로 다른 분류기들을 혼합해 더 나은 성능을 가지는 meta-classifier를 만듦
    - 예를 들어, 10명의 전문가가 특정 문제에 대해 예측하고 결과를 잘 섞어서 각 개인의 예측보다 더욱 정확하고 견고한 결과를 얻을 수 있다.
  - 앙상블 대상
    - 여러 알고리즘
      - Majority voting, random forest, stacking
    - 여러 데이터
      - Sampling: bagging, pasting
      - Feature 선택: random forest
  - Boosting
    - 몇 개의 weak learners을 결합하여 strong learner를 만듦
    - 모델들을 순차적으로 학습함
    - 모델
      - Adaboost
      - Gradient boosting
  - Stacking
    - 예측된 결과를 결합하는 모델을 학습함

MAJORITY VOTING

Majority voting

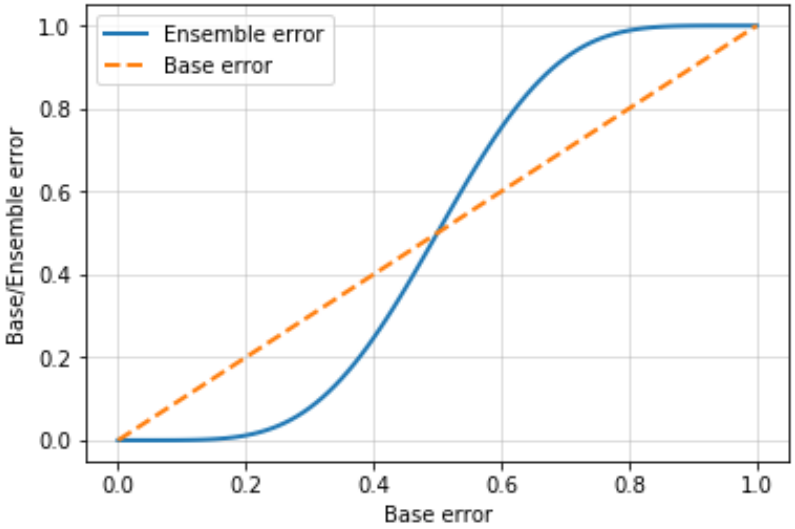
- Majority voting classifier
  - Majority voting이란 단순히 분류기의 majority(50% 이상)가 선택한 클래스 label을 선택하는 것이다.
  - Multi-class 상황에 대해서는 plurality voting(가장 많이 선택된 것)을 사용한다.
  - 각각의 분류기는 서로 다른 분류 알고리즘을 사용할 수 있다.
    - Decision trees, support vector machines, logistic regression classifiers.
  - 같은 알고리즘을 서로 다른 트레이닝 셋에 적용시킬 수도 있다.
    - Random forest





Majority voting

- Ensemble method의 성능
  - n개의 분류기가 이진 분류를 실행하는 경우, 각 분류기가 동일한 에러 비율 ϵ을 가진다고 가정하자.
  - Ensemble했을 때의 에러 확률은 다음과 같다.
    - $\epsilon_{ensemble} = \sum_{k=\lceil n/2 \rceil}^{n} \binom{n}{k} \epsilon^k (1-\epsilon)^{n-k}$
    - Majority가 에러를 내지 않는 한, ensemble method는 에러를 내지 않는다.
  - N=11, ϵ=0.25일 때 $\epsilon_{ensemble} = \sum_{k=6}^{11} \binom{11}{k} 0.25^k (1-0.25)^{11-k} = 0.034$
  - ϵ값을 조절해 가며 ensemble 에러 확률을 구하면 다음과 같다.
    - $\epsilon < 0.5$인 이상 ensemble method가 더 나은 에러를 가진다.



Implementing a simple majority vote classifier

- 각 분류기가 동일한 weight를 가지는 경우:
  - $\hat{y} = mode\{C_1(x), C_2(x), \ldots, C_m(x)\}$
  - Mode는 집합에서 가장 빈번한 값을 반환하며, C는 각 분류기의 예측 값이다.
- 각 분류기에 대해 weight 값을 부여하는 경우:
  - $\hat{y} = arg\max_i \sum_{j=1}^{m} w_j \chi_A(C_j(x) = i)$
    - 이 때, χ 함수는 $C_j(x) = i$이면 1, 아니면 0을 반환한다.
  - 분류기 1, 2, 3의 weight값이 각각 0.2, 0.2, 0.6이고, 각 분류기의 예측 값이 0, 0, 1일 때, i=0이면 0.4, i=1이면 0.6이 되므로 ensemble method의 예측 값은 1이 된다.
- 각 분류기가 클래스에 대한 확률을 반환하는 경우:
  - $\hat{y} = arg\max_i \sum_{j=1}^{m} w_j p_{ij}$
    - 이 때, p_{ij}는 j번째 분류기가 클래스 i에 대해 반환하는 확률
  - 3개 분류기가 입력 데이터 x에 대해 반환한 확률이 [0.9, 0.1], [0.8, 0.2], [0.4, 0.6]
    - $P(i = 0|x) = 0.2 * 0.9 + 0.2 * 0.8 + 0.6 * 0.4 = 0.58$, $P(i = 1|x) = 0.2 * 0.1 + 0.2 * 0.2 + 0.6 * 0.6 = 0.42$ 이므로 $\hat{y} = 0$

Implementing a simple majority vote classifier

- 예제 코드 구현 – 알아야 할 것들
  - numpy.bincount(*x*, *weights=None*, *minlength=0*)
    - https://docs.scipy.org/doc/numpy-1.13.0/reference/generated/numpy.bincount.html

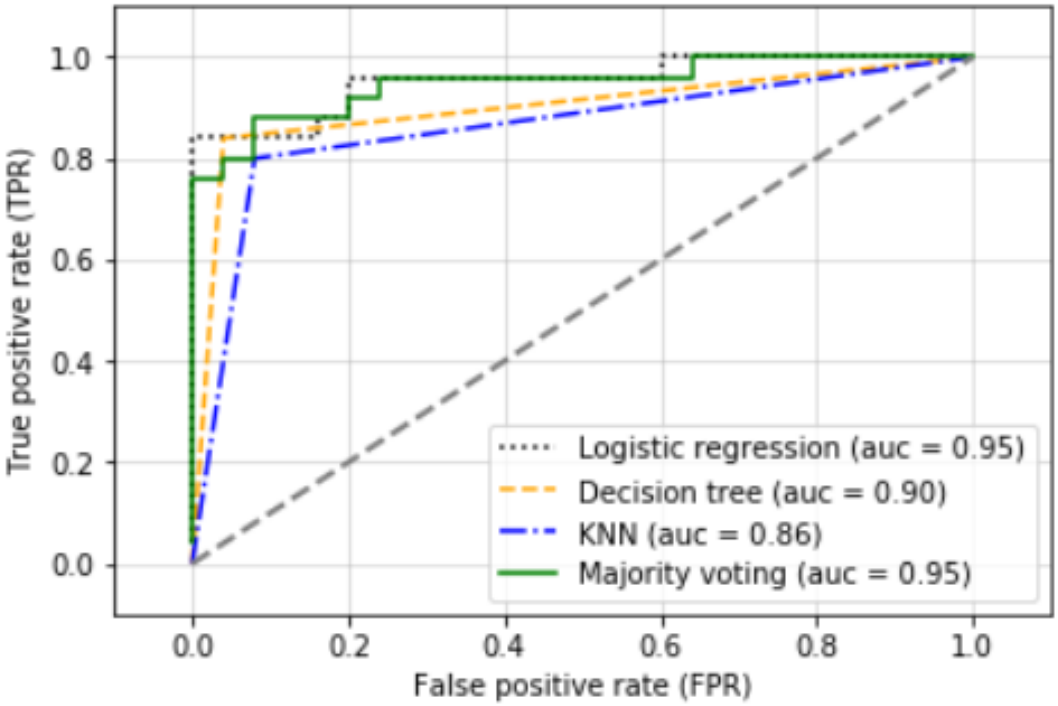
Parameters	
x	Array of nonnegative ints
weights	Weights, array of same length as x

Returns	
out	0부터 max(x)까지 각 정수가 몇 개인지 반환하는 array. weight가 있을 경우, 개수 대신 weight의 합산을 반환한다.

  - sklearn.pipeline.Pipeline (class)
    - 일련의 변환(transform) 과정과 최종 estimator를 순차적으로 적용시켜준다.
    - 하나의 estimator와 같이 사용할 수 있다.
    - 자세한 사항은 http://scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline.html 참조

Implementing a simple majority vote classifier

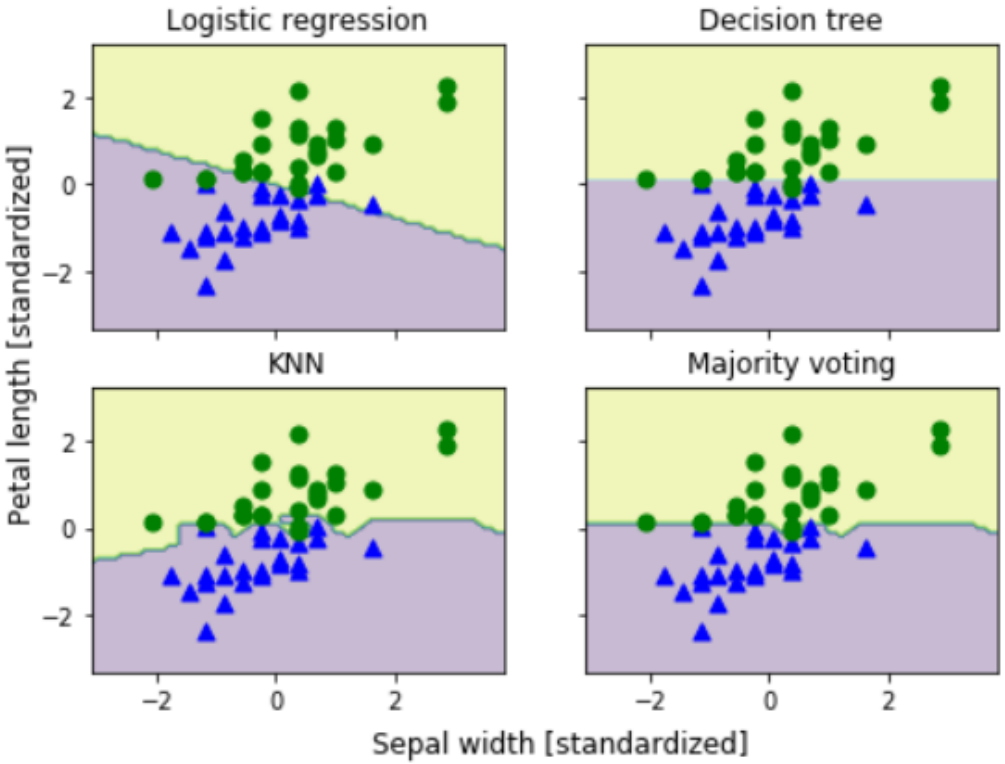
- 결과 예제 코드-ROC curve



- ROC curve는 왼쪽 위 꼭지점에 가까울수록 좋은 성능을 나타낸다.
  - Majority voting의 성능이 우수함을 확인할 수 있다.

Implementing a simple majority vote classifier

- 결과 예제 코드-Decision boundary



- 전체적으로 세 가지 분류기의 boundary가 섞인 형태
  - Decision tree와 유사하나, KNN의 nonlinearity를 포함

BAGGING

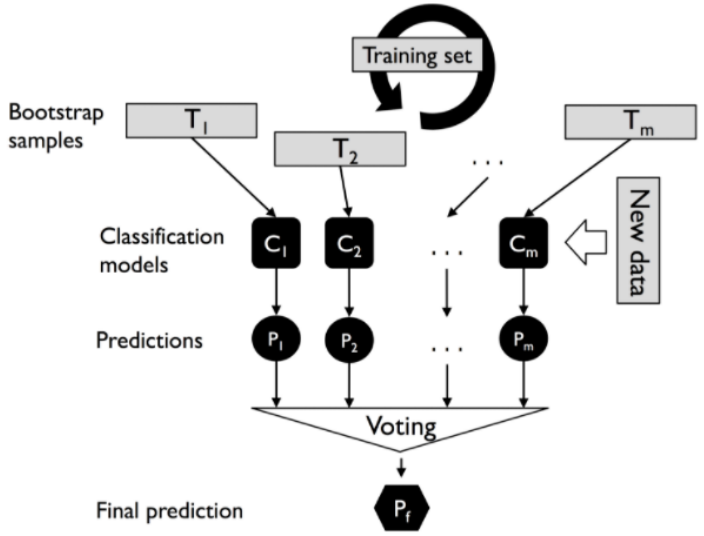
Bagging

Bagging and Pasting

- ensemble learning 기술의 하나로 각 분류기가 다른 학습 데이터로 학습됨
- Bootstrap aggregation이라고도 부른다.
- 초기 트레이닝 셋으로부터 sample을 만들어낸다.
  - Bagging: random sample with replacement (Bootstrap)
  - Pasting: random sample without replacement
- Aggregation 방법
  - Voting: 분류 문제
  - Averaging: 회귀 문제

Sample indices	Bagging round 1	Bagging round 2	…
1	2	7	…
2	2	3	…
3	1	2	…
4	3	1	…
5	7	1	…
6	2	7	…
7	4	7	…

C_1 C_2 C_m



Bagging

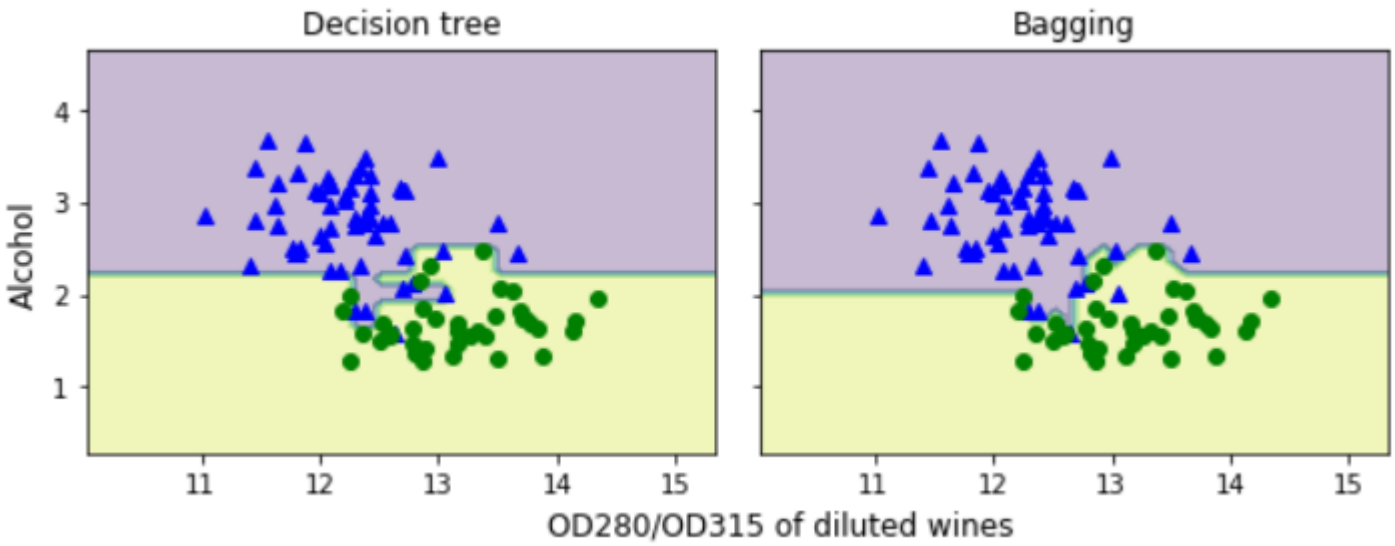
- Sklearn에서 bagging 사용 방법
  - sklearn.ensemble.BaggingClassifier

Parameters	
base_estimator	The base estimator to fit
n_estimators	The number of base estimators
max_samples	The number of samples to draw from X to train each base estimator
max_features	The number of features to draw from X to train each base estimator
bootstrap	Whether samples are drawn with replacement
bootstrap_features	Whether features are drawn with replacement

  - 자세한 사용법은 http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.BaggingClassifier.html 참조

Bagging

- 예제 코드 결과



- Bagging을 통해 decision tree의 decision boundary가 더 매끄러워졌음을 확인할 수 있다.

STACKING

Stacking

- Stacking multiple predictors
  - Voting 등의 trivial한 함수를 사용하는 대신에, ensemble 모델 자체를 학습
- Stacking 과정
  - 트레이닝 셋을 두 subset으로 나누고, 하나를 이용해 분류기들을 학습
  - 남은 subset을 분류기들의 입력값으로 하여 그 출력값을 가지고 blender(최종 예측값을 반환하는 모델)을 학습
- Stacking with multiple layers
  - Ex)트레이닝 셋을 3개로 나눈다.

Stacking

- **Stacking Structures**

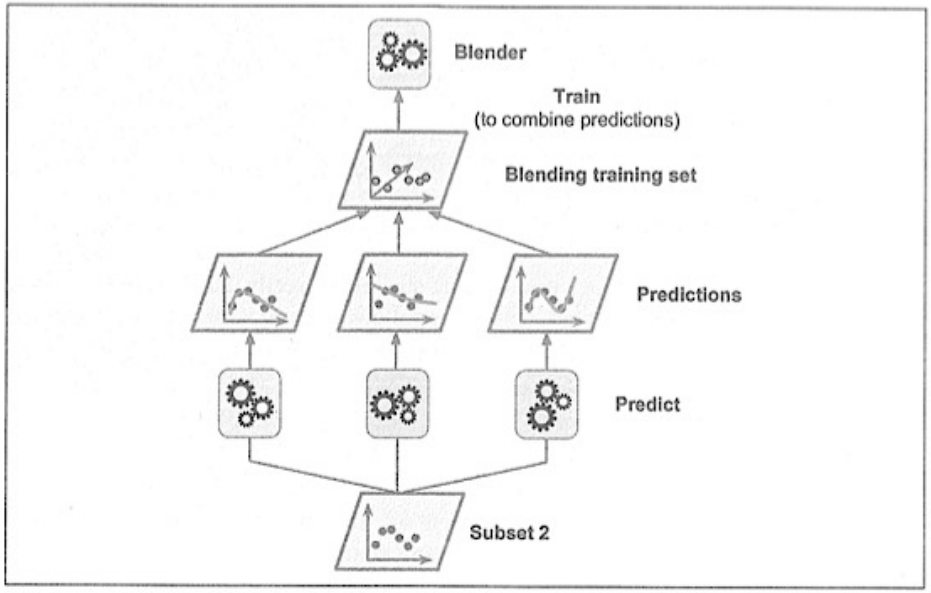


Figure 7-14. Training the blender

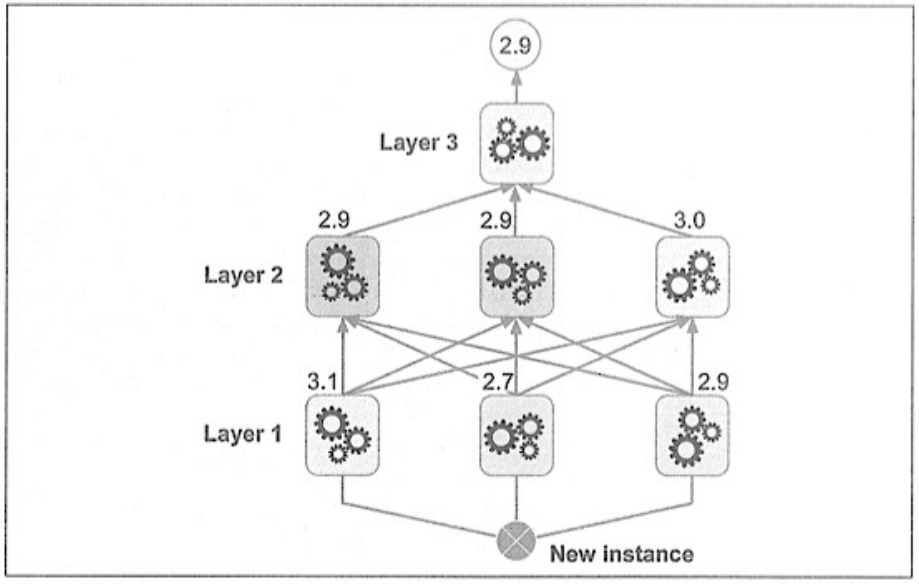


Figure 7-15. Predictions in a multilayer stacking ensemble

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.