Bagging과 Boosting 그리고 Stacking

19 JUL 2017 on Datascience

오늘은 머신러닝 성능을 최대로 끌어올릴 수 있는 앙상블 기법에 대해 정리해보았습니다.

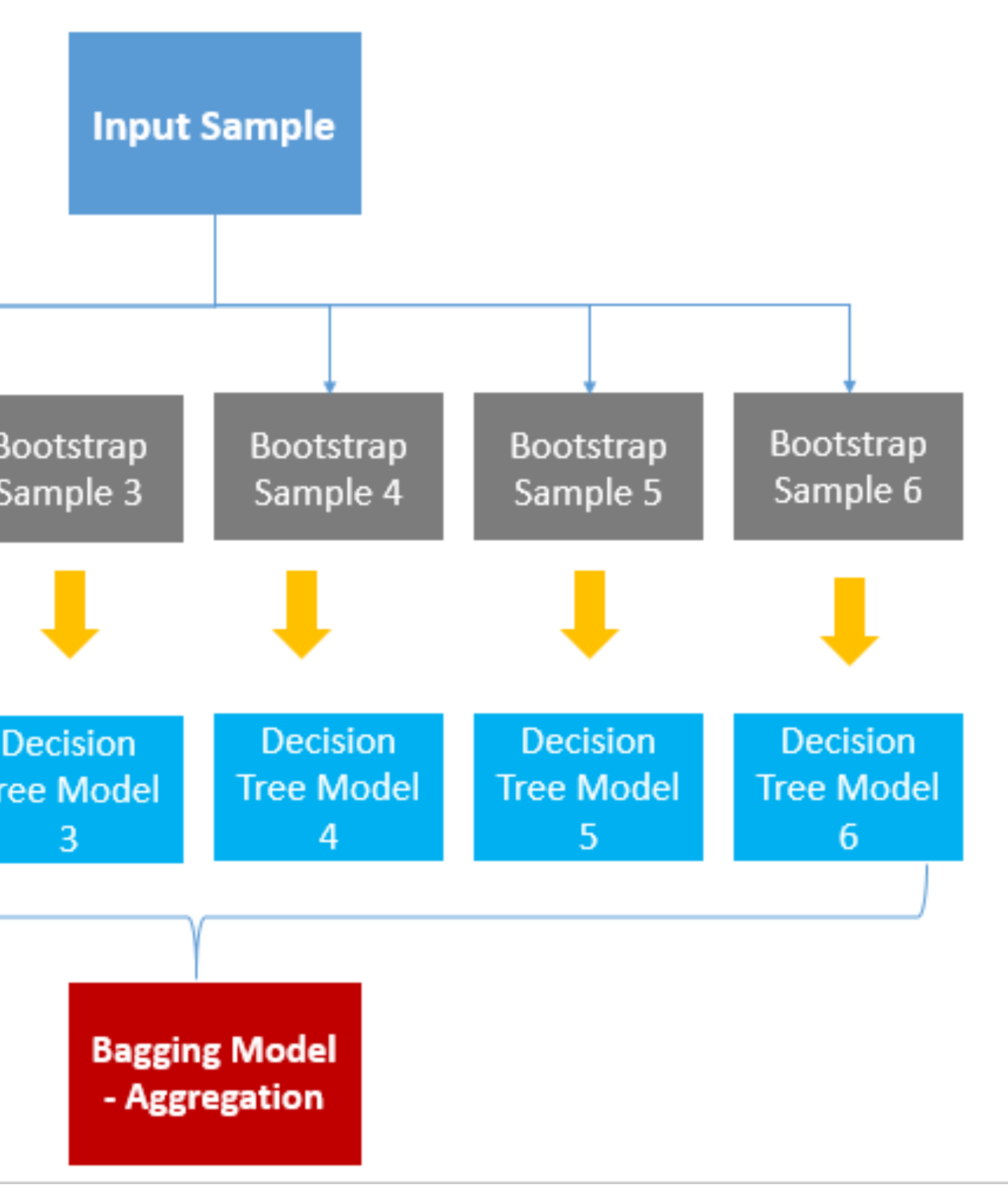
Ensemble, Hybrid Method

앙상블 기법은 동일한 학습 알고리즘을 사용해서 여러 모델을 학습하는 개념입니다. Weak learner를 결합한다면, Single learner보다 더 나은 성능을 얻을 수 있다는 아이디어입니다. **Bagging** 과 **Boosting** 이 이에 해당합니다.

동일한 학습 알고리즘을 사용하는 방법을 앙상블이라고 한다면, 서로 다른 모델을 결합하여 새로운 모델을 만들어내는 방법도 있습니다. 대표적으로 **Stacking** 이 있으며, 최근 Kaggle 에서 많이 소개된 바 있습니다.

Bagging

Bagging은 샘플을 여러 번 뽑아 각 모델을 학습시켜 결과를 **집계(Aggregating)** 하는 방법입니다. 아래의 그림을 통해 자세히 알아보겠습니다.



먼저 대상 데이터로부터 복원 랜덤 샘플링을 합니다. 이렇게 추출한 데이터가 일종의 표본 집단이 됩니다. 이제 여기에 동일한 모델을 학습시킵니다. 그리고 학습된 모델의 예측변수들을 집계하여 그 결과로 모델을 생성해냅니다.

이러한 방식을 **Bootstrap Aggregating** 이라고 부릅니다.

이렇게 하는 이유는 "알고리즘의 안정성과 정확성을 향상시키기 위해서" 입니다. 대부분 학습에서 나타나는 오류는 다음과 같습니다.

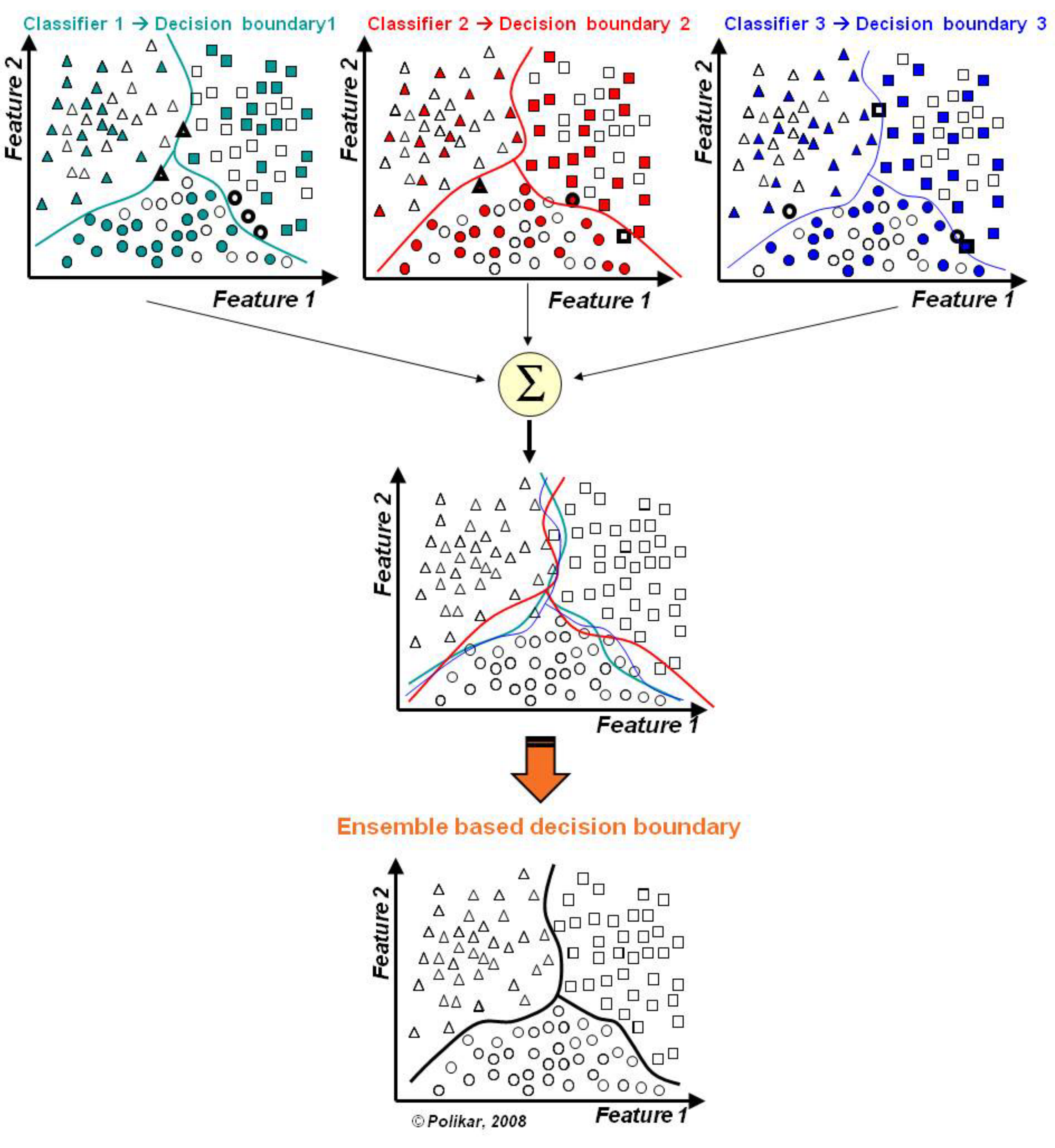
1. 높은 bias로 인한 Underfitting
2. 높은 Variance로 인한 Overfitting

앙상블 기법은 이러한 오류를 최소화하는데 도움이 됩니다. 특히 Bagging은 각 샘플에서 나타난 결과를 일종의 중간값으로 맞추어 주기 때문에, Overfitting을 피할 수 있습니다.

일반적으로 Categorical Data인 경우, 투표 방식 (Voting)으로 집계하며 Continuous Data인 경우, 평균 (Average)으로 집계합니다.

대표적인 Bagging 알고리즘으로 `RandomForest` 모델이 있습니다. 원래 단일 DecisionTree 모델은 boundary가 discrete 한 모양일 수 밖에 없지만, RandomForest는 여러 트리 모델을 결합하여 이를 넘어설 수 있게 되었습니다.

결과는 아래와 같습니다.



Boosting

Bagging이 일반적인 모델을 만드는데 집중되어있다면, Boosting은 맞추기 어려운 문제를 맞추는데 초점이 맞춰져 있습니다.

수학 문제를 푸는데 9번 문제가 엄청 어려워서 계속 틀렸다고 가정해보겠습니다. Boosting 방식은 9번 문제에 가중치를 부여해서 9번 문제를 잘 맞춘 모델을 최종 모델로 선정합니다. 아래 그림을 통해 자세히 알아보겠습니다.

Boosting도 Bagging과 동일하게 복원 랜덤 샘플링을 하지만, 가중치를 부여한다는 차이점이 있습니다. Bagging이 병렬로 학습하는 반면, Boosting은 순차적으로 학습시킵니다. 학습이 끝나면 나온 결과에 따라 가중치가 재분배됩니다.

오답에 대해 높은 가중치를 부여하고, 정답에 대해 낮은 가중치를 부여하기 때문에 오답에 더욱 집중할 수 있게 되는 것 입니다. Boosting 기법의 경우, 정확도가 높게 나타납니다. 하지만, 그만큼 Outlier에 취약하기도 합니다.

AdaBoost, XGBoost, GradientBoost 등 다양한 모델이 있습니다. 그 중에서도 XGBoost 모델은 강력한 성능을 보여줍니다. 최근 대부분의 Kaggle 대회 우승 알고리즘이기도 합니다.

Stacking

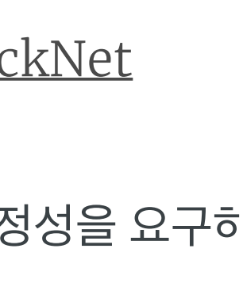
Meta Modeling 이라고 불리기도 하는 이 방법은 위의 2가지 방식과는 조금 다릅니다. "Two heads are better than one" 이라는 아이디어에서 출발합니다.

Stacking은 서로 다른 모델들을 조합해서 최고의 성능을 내는 모델을 생성합니다. 여기에서 사용되는 모델은 SVM, RandomForest, KNN 등 다양한 알고리즘을 사용할 수 있습니다. 이러한 조합을 통해 서로의 장점은 취하고 약점을 보완할 수 있게 되는 것 입니다.

Stacking은 이미 느끼셨겠지만 필요한 연산량이 어마어마합니다. 적용해보고 싶다면 아래의 StackNet을 사용하시는 걸 강력하게 추천합니다.

https://github.com/kaz-Anova/StackNet

문제에 따라 정확도를 요구하기도 하지만, 안정성을 요구하기도 합니다. 따라서, 주어진 문제에 적절한 모델을 선택하는 것이 중요합니다.



Swalloow
Interested in Data Science & Engineering
Seoul, Korea swalloow.github.io/

Share this post

댓글 7건 swalloow 1 로그인

♡ 추천 7 Tweet f 공유 인기순

토론 참여하기

다음으로 로그인 또는 디스커스에 가입하세요. ?

이름

mingyu park
22일 전
잘 보고 갑니다! 배깅과 부스팅에 대해 핵심을 잘 설명해 주셔서 이해에 도움이 많이 되었습니다.
일반적으로 학습에서 높은 bias로 인한 Underfitting이 일어나고 앙상블 기법을 통해 이러한 오류를 최소화 할 수 있다고 쓰셨는데, 혹시 이에 대한 설명 해주실 수 있으신가요?
답글

Geon Kim
한달 전
감사합니다!
답글

안재형
3달 전
bagging과 boosting 항상 헷갈렸는데 명쾌하게 정리가 되네요.
답글

Curycu
4달 전
부트스트랩을 이용하는 기법들이라 비복원추출이 아닌 복원추출아닌가요
답글

> **Junyoung Park** → Curycu
> 4달 전
> 잘못 표기되어 있네요. 본문 내용 수정했습니다 감사합니다!
> 답글

Ashtray Kim
4달 전
gradient boosting에서 가중치가 어떤 것인지 궁금합니다.
가중치 개념은 adaboost에 있는 걸로 알고 있는데요..
답글

> **Junyoung Park** → Ashtray Kim
> 4달 전
> pseudo-residuals을 생각하며 썼던거 같은데 가중치라는 단어가 적절하지 않은 듯 합니다.
> 내용 수정하겠습니다 감사합니다.
> 답글

SWALLOOW의 다른 댓글.

JWT를 구현하면서 마주치게 되는 고민들
댓글 8건 • 2년 전
Lawrence Kim — 구현하기 나름일 것 같습니다. 쿠키에 저장하셔도 되고, 로컬스토리지에 저장하셔도 됩니다.https://stormpath.com/blog/...

넷플릭스 본사 방문 후기
댓글 4건 • 일년 전
Goun Na — 우왓! 저도 데이터엔지니어인데 부럽습니다!

Pandas DataFrame을 병렬처리 하는 방법
댓글 3건 • 2년 전
rightx2 — 실제로 몇개이상의 row에 대해서 성능 이득을보나요? 데이터가 딱히 마땅한게 없어서 경험들어보고 싶습니다ㅠ

Jupyter에서 Scala로 Spark 사용하는 방법
댓글 3건 • 2년 전
오영택 — 와 진짜 유용하게 잘썼습니다

READ THIS NEXT

Hive Metastore 구축 관련 문제와 해결과정

최근 Hive Metastore를 구축하면서 겪은 이슈와 해결 과정을 기록해두려고 합니다. 사용 환경은 Spark 2.1.1, Hive...

YOU MIGHT ENJOY

Spark DataFrame을 MySQL에 저장하는 방법

Spark에서 MySQL에 접근하고 DataFrame을 read, write 하는 방법에 대해 정리해보았습니다. 참고로 저는 Spark 2.1.0...