▣ **강의명:** CS109A: 데이터 과학 입문

▣ **주차:** Lecture 22

▣ **교수명:** Pavlos Protopapas, Kevin Rader, Chris Gumb

▣ **목적:** Lecture 22의 핵심 개념 학습

Contents

개요: 랜덤 포레스트와 변수 중요도

문서 핵심 요약

랜덤 포레스트(*Random Forest*)는 배깅(*Bagging*)의 변형으로, 다수의 결정 트리(*Decision Tree*)를 훈련하고 결과를 집계하여 예측의 분산(*Variance*)을 줄이는 앙상블 학습 기법입니다. 핵심은 각 트리의 분할 시마다 전체 특징(*Feature*, 변수)의 부분집합을 무작위로 선택하여, 트리 간의 상관관계를 낮춰 성능을 극대화하는 것입니다. 트리 간의 낮은 상관관계는 모델의 안정성(*Robustness*)을 높이고 과적합(*Overfitting*) 위험을 줄입니다. 변수 중요도(*Variable Importance*)는 모델의 해석력을 높이는 수단이며, 불순도 감소량 평균(*Mean Decrease in Impurity*, MDI)과 순열 중요도(*Permutation Importance*) 두 가지 방법으로 계산됩니다. 또한, 불균형 데이터(*Imbalanced Data*) 문제와 결측값(*Missing Data*) 처리 방법도 함께 다룹니다.

1 용어 정리

주요 용어 및 직관적 설명

2 핵심 개념: 랜덤 포레스트의 원리

2.1 배깅(*Bagging*)의 한계: 트리의 상관관계

배깅은 부트스트랩 샘플(*Bootstrap Sample*)을 사용하여 다수의 결정 트리를 학습하고 그 결과를 집계(*Aggregate*)하여 분산을 줄이는 앙상블 기법입니다. 그러나 배깅에서 생성된 트리는 서로 **상관관계(*Correlation*)가 높게** 나타나는 경향이 있습니다.

상관관계가 높은 이유에 대한 비유

강력한 예측 변수(*Strong Predictor*)가 있는 경우, 모든 부트스트랩 샘플에서도 이 예측 변수가 가장 높은 불순도 감소 효과를 보일 것입니다. 마치 모든 의사(*Estimator*)가 동일한 환자 데이터를 기반으로 훈련받고, '휴식, 수분 섭취'와 같은 가장 확실한 조언을 첫 번째 조치로 내리는 것과 같습니다. 따라서, 대부분의 트리는 **최상위 노드(*Root Node*)**에서 동일한 특징으로 분할을 시작하게 되어, 트리가 서로 비슷한 구조와 예측을 하게 됩니다. 이러한 상관관계는 앙상블이 기대하는 분산 감소 효과($1/\sqrt{n}$)를 약화시킵니다.

2.2 랜덤 포레스트: 비상관화(*Decorrelation*)의 도입

랜덤 포레스트는 배깅을 개선하여 트리 간의 상관관계를 낮추는 방법입니다. 트리의 상관관계를 낮추는 것이 분산(*Variance*)을 효과적으로 줄이는 핵심입니다.

랜덤 포레스트의 핵심 아이디어

각 분할 노드(*Split Node*)마다, 전체 특징 J개 중 무작위로 선택된 $J' < J$개의 특징 부분집합만 고려합니다. 즉, 트리가 자라날 때마다 최고의 특징을 선택할 때 **전체 특징이 아닌** 일부 특징 중에서만 선택하게 강제합니다. 이로 인해 강력한 특징이 매번 선택될 확률이 낮아지고, 결과적으로 트리가 다양해지고 서로 덜 유사해집니다.

랜덤 포레스트 요약 단계

1. B개의 부트스트랩 데이터셋을 생성합니다 (배깅과 동일).
2. B개의 결정 트리를 초기화합니다.
3. 각 트리 내의 **각 분할(*Split*)**마다:
   (a) 전체 특징 J개 중 **무작위로** J'개의 특징 부분집합을 선택합니다 ($J' < J$).
   (b) 이 J'개의 특징 중에서 최적의 특징과 최적의 임계값(*Threshold*)를 선택하여 분할합니다.
4. 최종적으로 모든 트리의 예측을 집계합니다 (분류는 다수결, 회귀는 평균).

이때 J'는 각 분할마다 **새롭게** 무작위로 선택됩니다.

2.3 하이퍼파라미터 튜닝

랜덤 포레스트는 여러 개의 하이퍼파라미터를 가지며, 이는 모델의 성능과 훈련 속도에 영향을 미칩니다.

주요 하이퍼파라미터

- **분할 시 무작위 선택할 특징의 개수** J': 트리 간의 상관관계를 조절하는 가장 중요한 파라미터입니다.
- **앙상블 내 트리의 총 개수** B: 분산 감소량을 결정합니다. 트리가 많을수록 분산은 줄어들지만 계산 시간이 늘어납니다.
- **트리 정지 조건**: 최대 깊이(*Maximum Depth*), 최소 리프 노드 크기(*Minimum Leaf Node Size*) 등.
- **분할 기준**: 지니 불순도(*Gini Impurity*) 또는 엔트로피(*Entropy*).

하이퍼파라미터 튜닝과 전문가의 권장 사항

1. **최적의 J' 선택**: 교차 검증(*Cross-Validation*)을 사용해야 하지만, 일반적인 경험 법칙(*Rule of Thumb*)이 있습니다.
   - **분류(*Classification*)**: $\sqrt{N_j}$ (전체 특징 개수의 제곱근)
   - **회귀(*Regression*)**: $N_j/3$
2. **트리의 개수** B: OOB 오류가 더 이상 감소하지 않고 안정화되는 지점까지 늘립니다. 트리의 개수는 **과적합(*Overfitting*)을 유발하지 않습니다**. 단지 분산만 감소시킬 뿐입니다.
3. **정지 조건 및 기준**: 보통 최대 깊이(*Maximum Depth*)나 지니 불순도(*Gini*)를 기본값으로 사용하고, 모델이 작동한 후에 더 세밀한 튜닝을 시도할 수 있습니다.

OOB 오류를 활용한 검증

랜덤 포레스트는 OOB(Out-of-Bag) 샘플, 즉 부트스트랩 과정에서 특정 트리 훈련에 사용되지 않은 데이터 포인트들을 사용하여 **별도의 검증 세트 없이** 모델을 평가할 수 있습니다. 이것이 교차 검증을 대체할 수 있는 효율적인 방법입니다.

3 변수 중요도(*Variable Importance*) 평가

앙상블 모델은 단일 결정 트리처럼 쉬운 규칙(*Rule*) 형태로 해석하기 어렵기 때문에, 어떤 특징이 예측에 가장 큰 영향을 미쳤는지 평가하여 모델의 **해석력(*Interpretability*)**을 높여야 합니다.

3.1 1. 불순도 감소량 평균(*Mean Decrease in Impurity*, MDI)

> MDI는 특정 특징이 트리의 불순도(예: 지니 불순도)를 평균적으로 얼마나 감소시켰는지를 측정하여 중요도를 산출하는 방법입니다.

MDI 계산 절차

1. **노드별 불순도 감소량 계산**: 단일 트리 내의 각 노드 q에서 분할로 인한 불순도 감소량 ΔI_q를 계산합니다.
$$\Delta I_q = (\frac{n}{N}) \left[Gini_n - \sum_{m \in \text{children}} (\frac{m}{n}) Gini_m \right]$$
(n: 노드의 샘플 수, N: 전체 데이터 샘플 수, m: 자식 노드의 샘플 수)
2. **특징별 중요도 합산**: 특정 특징 j가 사용된 모든 노드 n의 불순도 감소량을 합하여 해당 특징의 중요도 $F_j^{(t)}$를 계산합니다.
3. **정규화**: 중요도 합계를 모든 특징의 중요도 합계로 나누어 0에서 1 사이 값으로 정규화합니다.
4. **앙상블 평균**: 모든 트리 T에 대해 정규화된 특징 중요도 $\hat{F}_j^{(t)}$를 평균하여 최종 MDI 중요도를 구합니다.
$$\mathcal{F}_j = \frac{\sum_t \hat{F}_j^{(t)}}{T}$$

MDI의 장단점

- **장점**: 계산 속도가 빠릅니다. 모든 필요한 값은 랜덤 포레스트 훈련 중에 계산됩니다.
- **단점**: 수치형 특징(*Numerical Feature*)이나 범주형 특징 중 고유값이 많은 특징(*High Cardinality Categorical Feature*)에 **편향(*Bias*)**되어 중요도를 과대평가하는 경향이 있습니다. 이들은 분할 지점(*Split Point*)이 많기 때문에 우연히 불순도를 크게 감소시키는 분할을 찾을 가능성이 높습니다.

3.2 2. 순열 중요도(*Permutation Importance*)

> 순열 중요도는 특징의 값들을 무작위로 섞어(*Permute*) 해당 특징과 결과 변수 간의 관계를 끊었을 때, 모델의 검증/OOB 성능이 얼마나 하락하는지를 측정하여 중요도를 산출하는 방법입니다.

순열 중요도 계산 절차

1. **기준 성능 기록**: 원본 데이터셋에서의 OOB 또는 검증 정확도 s를 기록합니다.
2. **특징 순열**: 관심 있는 특징 j의 데이터 열을 무작위로 섞습니다.
3. **순열 성능 측정**: 섞인 데이터셋으로 모델을 실행하여 새로운 OOB/검증 정확도 $s_{k,j}$를 기록합니다.

4. **반복 및 평균**: 2, 3단계를 K 번 반복하고 평균 순열 정확도 s_j 를 계산합니다.

$$s_j = \frac{1}{K}\sum_{k=1}^{K} s_{k,j}$$

5. **중요도 산출**: 기준 성능과 평균 순열 성능의 차이를 계산하여 중요도를 산출합니다.

$$\text{특징 중요도} = s - s_j$$

순열 중요도의 장단점

- **장점**: MDI의 편향이 없으며, 실제 모델 성능에 미치는 영향을 직접적으로 측정하므로 **더 직관적이고 신뢰할 만한** 중요도를 제공합니다.
- **단점**: MDI보다 계산 비용이 더 많이 듭니다. (각 특징마다 K 번의 예측이 필요)

3.3 랜덤 포레스트 vs. 배깅 변수 중요도 비교

트리 간 상관관계가 높은 **배깅(*Bagging*)**은 소수의 강력한 예측 변수(예: `ChestPain`, `Ca`)에 중요도가 **집중**되는 경향을 보입니다. 반면, 비상관화 과정을 거친 **랜덤 포레스트(*Random Forest*)**의 중요도는 여러 특징에 걸쳐 **더 부드럽고 분산된** 분포를 보여줍니다. 이는 트리들이 다양한 특징을 고려하도록 강제되었기 때문입니다.

랜덤 포레스트가 잘 작동하지 않는 경우

전체 특징의 수가 매우 많지만, **실제로 중요한 특징의 수가 적을 때** (예: 1000개의 특징 중 10개만 중요), 랜덤 포레스트는 오히려 성능이 떨어질 수 있습니다. 각 분할에서 무작위로 선택된 특징 부분집합에 중요한 특징이 포함되지 않을 확률이 높아지기 때문에, 생성된 트리들이 대부분 '약한 모델'이 될 수 있습니다. 이 경우 PCA나 다른 특징 선택 기법을 사용해 불필요한 특징을 미리 제거하는 것이 좋습니다.

4 불균형 데이터(*Class Imbalance*) 처리

데이터셋에서 특정 클래스(소수 클래스)의 샘플 수가 다른 클래스(다수 클래스)에 비해 현저히 적을 때 발생하는 문제입니다. 이 경우 모델이 다수 클래스만 예측해도 높은 '정확도'를 얻을 수 있어, 실제로는 소수 클래스 예측 능력이 매우 낮아집니다.

불균형 데이터 문제 진단

정확도(*Accuracy*)는 좋은 평가 지표가 아닙니다. 99% 대 1%의 클래스 불균형에서 모델이 항상 99%의 다수 클래스만 예측해도 정확도는 99%가 나옵니다. 따라서, 불균형 데이터셋에서는 **F1-점수**나 **ROC 곡선 하 면적(*AUC*)**과 같은 지표를 사용해야 합니다.

4.1 처리 방법

1. **언더샘플링(*Under-sampling*): 다수 클래스 샘플 수 감소**
   - **무작위 언더샘플링**: 다수 클래스에서 무작위로 샘플을 제거하여 클래스 균형을 맞춥니다. (단점: 중요한 정보를 포함하는 샘플이 손실될 수 있습니다.)
   - **니어 미스(*Near Miss*)**: 결정 경계(*Decision Boundary*)에서 멀리 떨어진, 즉 분류에 덜 유익한 다수 클래스 샘플을 제거하여 정보 손실을 최소화합니다. (경계 근처의 샘플은 보존)
2. **오버샘플링(*Over-sampling*): 소수 클래스 샘플 수 증가**
   - **무작위 오버샘플링**: 소수 클래스의 샘플을 복원 추출(*with replacement*)하여 수를 늘립니다. (단점: 단순 복제이므로 과적합을 유발하거나 데이터의 다양성을 해칠 수 있습니다.)
   - **SMOTE (*Synthetic Minority Oversampling Technique*)**: 소수 클래스 샘플 주변에 **새로운 합성 데이터(*Synthetic Data*)**를 생성하여 수를 늘립니다. 데이터 영역을 확장하여 모델의 일반화 성능을 높입니다.
3. **클래스 가중치(*Class Weighting*)**
   - 모델의 손실 함수(*Loss Function*)에서 소수 클래스의 오류에 **더 높은 가중치**를 부여하여 학습 시 소수 클래스에 더 많은 관심을 기울이도록 합니다.
   - 사이킷런(*scikit-learn*)에서는 `class_weight='balanced'` 옵션을 통해 자동으로 클래스 빈도에 반비례하여 가중치를 조정할 수 있습니다.

$$W_K = \frac{N}{K \times N_K}$$

(N: 전체 샘플 수, K: 클래스 수, N_K: 클래스 K의 샘플 수)

실제 사용 전략

데이터 균형을 맞출 때에는 세 가지 방법(언더샘플링, 오버샘플링, 가중치) 중 하나 또는 그 조합을 사용합니다. 예를 들어, 오버샘플링과 언더샘플링을 결합하거나, 데이터 재조정 후 클래스 가중치를 적용할 수 있습니다. 중요한 것은 **항상 데이터를 균형 있게 처리**하는 것입니다 (40%/60% 불균형에서도 적용 권장).

5 결측값(*Missing Data*) 처리: 서러게이트 분할

결정 트리 기반 모델은 결측값을 처리하는 독특한 방법인 **서로게이트 분할(*Surrogate Split*)**을 사용할 수 있습니다.

5.1 서로게이트 분할의 개념

서로게이트 분할은 트리를 훈련하는 과정에서 **최적의 분할 특징(*Optimal Splitter*)**의 결측값(*Missing Value*)을 대신할 수 있는 **차선책의 특징(*Alternative Feature*)**을 미리 찾아 순위를 매겨 놓는 방법입니다.

작동 원리

1. **최적 분할 특징 선택**: 노드 N에서 불순도를 가장 크게 줄이는 최적의 분할 특징 P_{opt}를 찾습니다.
2. **분할 분포 기록**: P_{opt}를 사용해 노드를 분할했을 때, 각 자식 노드로 이동하는 반응 변수(*Response Variable*)의 분포(예: `Yes`/`No` 개수)를 기록합니다.
3. **대리 특징 찾기**: 나머지 모든 특징 P_i에 대해 P_{opt}와 동일한 분할을 시도했을 때, 반응 변수의 분포가 P_{opt}의 분포와 **가장 유사한** 특징을 찾습니다.
4. **유사도 측정**: 두 특징의 분할 분포가 얼마나 유사한지(즉, 분포를 일치시키기 위해 얼마나 많은 '플립'이 필요한지)를 측정하여 유사도를 산출하고 순위를 매깁니다.
5. **예측 시 사용**: 실제 예측 시점에 입력 데이터의 P_{opt} 값이 결측이면, 미리 정의된 순위에 따라 가장 유사한 서러게이트 분할 특징을 대신 사용하여 데이터를 분할합니다.

서러게이트 분할 예시

주요 특징이 '`Arteries Blocked`'라고 가정합니다. 이 특징으로 분할했을 때, 환자들의 '심장병 유무' 분포가 나옵니다.

- **최적 분할 특징 (`Arteries Blocked`)**: `FALSE` → [3 `No`, 1 `Yes`], `TRUE` → [0 `No`, 2 `Yes`]
- **대리 특징 후보 (`Chest Congested`)**: `FALSE` → [3 `No`, 2 `Yes`], `TRUE` → [0 `No`, 1 `Yes`]
- 두 분포를 비교하여 유사도를 측정합니다. 이 예시에서는 '`Chest Congested`'가 '`Arteries Blocked`'와 가장 유사한 분할 분포를 보였으므로 최적의 서러게이트가 됩니다.

따라서, 예측할 데이터에서 '`Arteries Blocked`' 값이 결측이면, 대신 '`Chest Congested`' 특징을 사용하여 트리를 따라 내려갑니다.

5.2 서러게이트 분할의 이점

- **해석력 향상**: 서러게이트는 최적 분할 특징과 비슷한 역할을 하는 보조 특징을 보여주므로, 주 분할기의 작동 방식을 이해하는 데 도움이 됩니다.
- **다중 공선성(*Multi-collinearity*) 활용**: 다중 공선성이 있는 경우(특징들이 서로 높은 상관관계를 가지는 경우), 대체할 서러게이트를 찾을 가능성이 높고 성능도 좋습니다.
- **명시적 대체 불필요**: 데이터 분석가가 별도의 결측값 대체(*Imputation*) 방법을 고민할 필요 없이, 트리가 자체적으로 결측값을 처리할 수 있습니다.

6 FAQ 및 초심자 체크리스트

6.1 자주 묻는 질문 (FAQ)

1. Q: 랜덤 포레스트에서 트리의 개수(*Number of Trees*)가 많아지면 과적합되나요?
   A: **아닙니다.** 트리의 개수를 늘리는 것은 **분산(*Variance*)**만 줄이는 역할을 합니다. 이는 앙상블의 예측을 안정화할 뿐이며, 다른 하이퍼파라미터처럼 모델의 복잡도(*Complexity*)를 제어하지 않으므로 과적합 위험을 증가시키지 않습니다.
2. Q: 랜덤 포레스트는 어떻게 클래스 예측 확률(*Class Probabilities*)을 반환하나요?
   A: 각 개별 트리는 최종적으로 클래스 예측 결과를 내놓습니다. 랜덤 포레스트 분류기(*Classifier*)는 모든 트리가 예측한 **클래스 예측의 평균**을 계산하여 이를 확률의 근사치(*Proxy*)로 사용합니다. 이 값을 임계값(*Threshold*)과 결합하여 ROC 곡선 등을 그릴 수 있습니다.
3. Q: MDI와 순열 중요도 중 어떤 것을 사용해야 하나요?
   A: **순열 중요도(*Permutation Importance*)**가 더 신뢰할 수 있습니다. MDI는 계산이 빠르다는 장점이 있지만, 수치형이나 고유값이 많은 범주형 특징에 편향되어 중요도를 과대평가하는 경향이 있습니다. 순열 중요도는 모델의 실제 성능 변화를 측정하므로 더 직관적이고 정확합니다.
4. Q: 결정 트리는 단일 모델로 불안정한데, 왜 배깅이나 랜덤 포레스트에서는 안정적인가요?
   A: 단일 결정 트리는 훈련 데이터의 작은 변화에도 민감하게 반응하여 매우 다른 구조로 학습될 수 있습니다(**고분산**). 앙상블은 부트스트랩을 통해 생성된 다양한 불안정한 트리를 평균하여, 그 '노이즈'로 인한 예측의 변동성(*Variability*)을 상쇄하고 **평균적인 안정적인 예측**을 얻습니다.

6.2 학습 및 구현 체크리스트

랜덤 포레스트 구현 전 점검 사항

- **핵심 원리 이해**: 랜덤 포레스트가 배깅과의 차이점(특징 무작위 부분집합 선택)과 분산 감소 메커니즘을 명확히 설명할 수 있는가?
- **하이퍼파라미터 튜닝**: 가장 중요한 하이퍼파라미터 J'와 B의 역할 및 권장 기본값($\sqrt{N_j}, N_j/3$)을 아는가?
- **평가 지표 선택**: 데이터 불균형 여부를 확인하고, 불균형 시 F1-점수 또는 AUC를 주 평가 지표로 사용하는가?
- **불균형 처리**: 언더샘플링(Near Miss)이나 오버샘플링(SMOTE) 또는 클래스 가중치 중 최소 한 가지 방법을 적용하여 데이터를 균형 있게 조정했는가?
- **변수 중요도 평가**: MDI의 편향성을 인지하고, 가능하면 순열 중요도를 사용하여 특징의 기여도를 평가했는가?
- **성능 비교**: 단일 결정 트리, 배깅, 랜덤 포레스트의 RMSE 또는 정확도를 비교하여 분산 감소 효과를 확인했는가?