암 유전체 데이터 처리 및 다중 분류 모델에 따른 성능 비교 연구

성다훈[1], 임유진[2]
[1]숙명여자대학교 IT공학과 석사과정
[2]숙명여자대학교 인공지능공학부 교수
ekgns324@sookmyung.ac.kr, yujin91@sookmyung.ac.kr

Performance Comparison of Multiple Classification Models for Cancer Genomic Data Processing

Da-Hun Seong[1], Yujin Lim[2]
[1]Dept. of Information Technology Engineering, Sookmyung Women's University
[2]Div. of Artificial Intelligence Engineering, Sookmyung Women's University

요 약

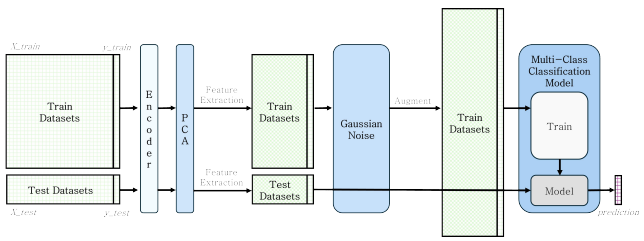
암 환자의 유전체 변이 데이터는 샘플 수가 적으나 많은 특성 정보를 가지는 의료 데이터로, 이러한 특성은 ML 모델의 성능 향상에 상당한 장벽으로 작용한다. 따라서 본 연구는 제한된 의료 데이터의 한계를 해결할 수 있는 적절한 모델 선택을 돕고자, 유전체 데이터를 이용하여 암종을 예측하는 다중 클래스 분류 문제에서 성능향상기법과 다중 클래스 분류 모델에 따른 성능을 분석하였다.

1. 서론

한국의 고령화 현상이 가속화되어 노령 인구는 늘어나고 있지만 의료 인력은 부족해짐에 따라, 의료 분야에 첨단기술을 접목하는 연구가 중요해지고 있다. 특히 데이터를 기반으로 학습하여 예측과 분류 작업을 하는 ML(Machine Learning) 모델은 의료 분야에 적용되기 위해 오랜 기간 활발히 연구되어 왔다[1]. ML 모델의 성능을 향상시키기 위해서는 대규모의 데이터가 필수적이나, 고품질 의료 데이터의 제한된 가용성은 상당한 장벽으로 작용한다[2]. 의료 데이터의 적은 샘플 수와 많은 특성 정보는 모델 학습을 어렵게 만드는 요인이여, 특히 이러한 경우 데이터의 전처리와 모델 선정에 따른 성능 차이가 클 것으로 예상된다.

암 환자의 유전체 변이 데이터는 적은 샘플 수와 많은 특성 정보를 갖는 대표적인 의료 데이터이다. 따라서 본 연구는 유전체 데이터를 이용하여 ML로 암종을 예측하는 다중 클래스 분류(Multi-class Classification) 문제에서, 데이터의 특성 추출(Feature Extraction) 및 증강(Augmentation)을 통한 모델 성능향상기법과 다중 클래스 분류 모델에 따른 성능을 분석해보고자 한다. 이를 통해 제한된 의료 데이터의 한계를 해결할 수 있는 적절한 모델 선택을 도울 수 있을 것이다.

2. 실험 설계



(그림 1)데이터 처리 및 모델 예측 과정

본 연구의 전체적인 데이터 처리 및 모델 예측 과정은 (그림 1)과 같다. 먼저 데이터 처리 과정은, 데이터를 인코더(Encoder)를 거쳐 범주형 데이터에서 수치형 데이터로 변환한 후, 이를 PCA(Principal Component Analysis)[3]를 통해 특성을 추출 한 뒤, 변환된 데이터를 가우시안 노이즈(Gaussian Noise)[4]를 이용하여 증강하는 과정을 거친다. 마지막으로 모델 예측 과정에서는 증강한 학습 데이터로

다중 클래스 분류 모델을 학습시킨 뒤, 학습된 모델에 테스트 데이터로 클래스를 예측한다.

2-1. 데이터셋

<표 1> 데이터셋의 통계 정보

Datasets	Use	#Rows	#Cols	#Unique Labels	Data Type
Original	Train	4,962	4,385	26	Object
	Test	1,241	4,385	26	
Feature Extract	Train	4,962	2,000	26	Float64
	Test	1,241	2,000	26	
Feature Extract + Augmentation	Train	9,924	2,000	26	Float64
	Test	1,241	2,000	26	

본 연구에서 사용한 의료 데이터는 암 환자들의 유전체 변이 정보(학습될 특성)와 암종(정답 레이블)으로 구성되어있다. 이를 이용하여 지도학습 방법으로 암종별 클래스를 구분하고 예측하는 작업을 하고자 한다. 데이터셋의 통계정보는 <표 1>과 같다. 기존 데이터를 보면 분류 모델의 학습을 위한 샘플 수(#Rows)는 약 5천 개로, 높은 분류 모델의 성능을 이끌어 내기에는 부족하며, 데이터의 샘플 수에 비하여 특성의 수(#Cols)가 많은 특징을 가지고 있다. 실험에 사용한 데이터셋은 총 세 가지로, 기존 데이터셋, 특성 추출한 데이터셋, 특성 추출 및 데이터 증강한 데이터셋이 있다. 특성 추출에서는 특성의 수가 줄어들고, 데이터 증강까지 했을 때는 샘플 수가 늘어난 것을 확인할 수 있다.

2-2. 성능향상기법

본 연구의 유전체 데이터는 특성 정보는 많으나 샘플 수가 상대적으로 적기 때문에, 특성 추출과 데이터 증강 기법으로 모델 성능을 향상하고자 한다. 먼저 특성 추출 기법의 대표적인 방법인 PCA는 데이터의 특성에서 주요 성분을 찾아내는 기법으로, 고차원 데이터를 저차원 데이터로 변환하며 중요한 정보를 유지한다. PCA의 주성분은 기존 특성들의 선형 결합으로 나타내며, 주성분을 새로운 축으로 데이터를 투영함으로써 차원이 줄어든 공간에서 데이터를 나타내게 된다. 이렇게 얻어진 변수들은 데이터의 분산을 효과적으로 설명한다. 본 연구의 데이터는 유전체 변이 간 중요도가 다르기 때문에 순서형 인코딩(Ordinal Encoding)으로 데이터를 순서가 있는 숫자 데이터로 변환한 뒤, PCA 기법으로 변이 간 상대적 중요도를 포착하고자 하였다. 다음으로 대표적인 증강 기법으로는 가우시안 노이즈가 있다. 이는 데이터에 추가되는 랜덤한 값으로, 정규분포를 따르는 랜덤 값으로 구성된다. 본 연구에 사용되는 데이터는 테이블 데이터이며, 샘플 수가 적어 과적합(Overfitting)이 발생할 수 있는데, 가우시안 노이즈를 추가하면 기존 데이터의 분포를 크게 변형하지 않으면서도 다양하게 변형된 데이터 샘플을 생성하여 데이터 양을 증가시킬 수 있다.

2-3. 다중 클래스 분류 모델 및 성능 지표

<표 2> 다중 클래스 분류 모델

Model	Accuracy	Description
RF [5]	Random Forest Classifier	여러 개의 의사결정나무(Decision Tree)를 통해 예측을 진행하며, 노이즈에 강하고 비선형 데이터 처리가 가능하다.
XGB [6]	eXtreme Gradient Boosting	성능 최적화와 실행 속도가 빠른 부스팅 알고리즘으로, 규제 기능이 추가되어 과적합을 방지한다.
LGBM [7]	Light Gradient Boosting Machine	메모리 효율성을 개선한 부스팅 알고리즘으로, 범주형 데이터 처리에 강하며 대규모 데이터에서 빠른 속도를 제공한다.
CB [8]	Categorical Boost Classifier	범주형 변수 처리를 위한 부스팅 알고리즘으로, 범주형 변수를 자동으로 처리한다.
SVC [9]	Support Vector Classifier	이진 분류에 기반을 둔 알고리즘으로, 고차원 데이터 처리에 강하며, 경계선이 명확하지 않은 데이터 분류에 적합하다.
KNN [10]	K Neighbors Classifier	새로운 데이터를 주변과의 거리에 따라 분류하는 직관적인 알고리즘으로, 데이터의 분포에 따라 유연하게 적용될 수 있다.
SNN [11]	Simple Neural Network	신경망(Neural Network)을 가진 딥러닝 모델로, 데이터의 복잡한 패턴을 학습할 수 있으며 확장성과 유연성을 갖는다.

본 실험에서 성능 비교에 사용된 다중 클래스 분류 모델은 7가지로, <표 2>와 같다. 다중 클래스 분류 모델 결과의 성능지표로는 Accuracy와 Macro F1를 사용하였으며, 이는 0에서 1 사이의 값을 갖는다. Accuracy는 전체 테스트 데이터 개수 중 정답을 맞춘 클래스의 비율을 의미하고, Macro F1 점수는 정확도(Precision)와 재현률(Recall)의 조화평균(F1)의 산술 평균으로서, 각 클래스의 중요도를 동일하게 취급하기 때문에 모든 클래스의 성능을 균형 있게 평가하고자 할 때 적합한 평가 지표이다. 본 연구에서는 두 가지 지표를 사용하여 단순 정답률뿐만 아니라 암종별 정답률도 평가하고자 하였다.

3. 실험 결과

실험에서는 <표 1>의 3가지 데이터를 각각 7가지 다중 클래스 분류 모델에 적용하여 2가지 성능지표로 평가하였다. 각각의 학습 데이터를 기준으로, 성능지표별 모델의 순위 결과를 살펴볼 것이다.

3-1. 기존 데이터 기준

<표 3> 기존 데이터로 학습한 모델별 성능

Ranking	Model	Accuracy	Model	Macro F1
1	CB	0.3529	CB	0.3371
2	LGBM	0.2981	LGBM	0.2211
3	KNN	0.1902	KNN	0.1271
4	RF	0.1813	RF	0.0543
5	SVC	0.1281	SVC	0.0332
6	XGB	0.0693	XGB	0.0050
7	SNN	0.0306	SNN	0.0023

먼저, 기존 데이터로 학습한 분류 모델의 성능지표별 순위 결과는 <표 3>과 같다. Accuracy와 Macro F1에서 모두 CB와 LGBM이 가장 우수한 성능을 보였는데, 이들은 사용한 데이터처럼 범주형 데이터 처리에 특화된 모델이기 때문으로 생각된다. LGBM은 CB과 성능 차이가 크지 않으나 더 빠르기 때문에 속도가 중요하다면 LGBM을, 정확도가 중요하다면 CB을 사용하는 것을 고려할 수 있다. SNN의 경우 딥러닝 모델이기 때문에 복잡한 차원 데이터 처리에 유리할 것 같지만, 성능이 가장 낮게 나왔다. 이러한 이유는 데이터의 양이 적어 과적합 문제가 발생했기 때문이다. 기존 데이터의 전체 결과를 살펴보면 전반적으로 성능이 낮으며, 성능향상기법을 도입할 필요가 있다는 것을 알 수 있다.

3-2. 특성 추출 데이터 기준

<표 4> 특성 추출 후 학습한 모델별 성능

Ranking	Model	Accuracy	Model	Macro F1
1	CB	▽ 0.3465	CB	▽ 0.3285
2	LGBM	▽ 0.2305	LGBM	▽ 0.1657
3	XGB	▲ 0.1861	XGB	▲ 0.1539
4	RF	▲ 0.1861	SNN	▲ 0.0860
5	SNN	▲ 0.1772	KNN	▽ 0.0588
6	SVC	▲ 0.1531	SVC	▲ 0.0540
7	KNN	▽ 0.1273	RF	▽ 0.0502

다음으로 PCA로 특성을 추출한 데이터를 기준으로 분류 모델의 성능 지표 순위 결과를 보면 <표 4>와 같다. 세모 표시(▲/▽)는, 원본 데이터로 모델을 학습한 결과 대비 기법을 적용한 데이터로 학습한 결과의 성능 변화를 표시한 것으로, (▲) 표시는 기존 데이터보다 성능이 증가된 경우를 의미하고, (▽) 표시는 기법을 적용했을 때 기존 데이터보다 오히려 성능이 낮아진 경우를 의미한다.

성능 결과는 기존 데이터의 학습 결과와 마찬가지로 Accuracy와 Macro F1 모두 CB가 가장 높았고, 그다음으로 LGBM이 순위를 이었다. 두 모델이 여전히 좋은 성능을 유지한 것은, PCA를 통해 차원이 축소되더라도 여전히 범주형 특성의 패턴을 가지고 있기 때문으로 해석된다. 다만 두 모델 모두 기존 데이터에 비해 성능이 저하된 이유는, 두 모델은 고차원 데이터의 다양한 특성을 활용하여 성능을 내지만 PCA 이후 차원이 축소되어 정보가 손실되었기 때문으로 보인다. 또한, KNN에서도 성능이 저하되었는데 그 이유는, KNN은 거리 기반 알고리즘인데 차원 축소 후 거리 계산의 의미가 변질되었기 때문으로 생각된다. 반대로 그 외의 모델에서 성능이 향상된 이유는, 차원 축소가 과적합을 방지하고 학습에 유리한 정보만 남기면서 모델의 복잡성을 줄였기 때문이다.

PCA를 적용한 전반적인 결과를 살펴보면, PCA 사용이 성능향상에 기여하는 바가 컸으나, 기존 데이터의 성능을 능가하는 결과는 없었다. 이는 차원을 축소했음에도 학습에 필요한 샘플 수 자체가 작은 문제 때문인 것으로 보인다. 그럼에도, 성능이 저하된 모델의 경우 두 성능 지표 결과에서 큰 차이가 없었고, PCA를 통해 차원을 줄일 수 있어 메모리 사용량이 줄고 계산 속도가 향상될 수 있기 때문에 필요에 따라 수행하는 것을 고려할 수 있을 것이다. 다만 여전히 PCA를 수행한 결과 또한 전반적으로 성능이 높지 않기 때문에 다른 성능향상기법을 추가하여 성능을 향상할 필요가 있다.

3-3. 특성 추출 및 증강 데이터 기준

<표 5> 특성 추출 및 데이터 증강 후 모델별 성능

Ranking	Model	Accuracy	Model	Macro F1
1	XGB	▲ 0.7521	SVC	▲ 0.8028
2	CB	▲ 0.7473	CB	▲ 0.7910
3	SVC	▲ 0.7392	XGB	▲ 0.7908
4	LGBM	▲ 0.6344	LGBM	▲ 0.6935
5	SNN	▲ 0.4502	SNN	▲ 0.2809
6	RF	▲ 0.2378	KNN	▽ 0.1150
7	KNN	▽ 0.1773	RF	▲ 0.1137

마지막으로, PCA로 특성을 추출하고 가우시안

노이즈로 데이터를 증강하여 학습한 분류 모델의 성능 결과는 <표 5>와 같다. 여기서는 XGB와 SVC가 각각 Accuracy와 Macro F1에서 가장 높은 점수를 기록했다. XGB는 트리 기반 모델로, 데이터를 분류할 때 데이터의 빈도에 따라 더 많은 분기를 갖기 때문에 다수 클래스에 대한 높은 정확도를 보인 것으로 해석된다. 반면, SVC는 데이터의 경계를 찾는 방식으로 작동하며, 클래스 간 경계를 최대한 명확히 하려는 특성으로 인해 소수 클래스에서도 균형 잡힌 성능을 나타낸 것으로 보인다. 결과적으로, 기존 데이터에 비해 두 가지 성능향상기법을 적용한 데이터에서 KNN을 제외하고는 유의미한 성능 향상 결과를 보였다.

4. 결론

본 연구는 의료 데이터의 특성을 반영하여 효율적으로 동작할 수 있는 모델 선택을 돕고자 연구를 진행하였다. 이를 위해 암 환자 유전체 데이터를 기반으로 성능향상기법과 분류 모델에 따른 암종 분류 실험을 진행하여 성능을 분석하였다. 그 결과 특성 추출 및 데이터 증강 기법의 병합으로 다양한 특성 값에 비해 상대적으로 적은 양의 데이터 문제를 해결할 수 있음을 확인하였다. 특히 PCA와 가우시안 노이즈를 결합한 기법이 전반적으로 성능 향상에 기여했음을 확인할 수 있었다. 다만, 분류 모델에 따라 적절한 성능향상기법을 선택해야 하며, 분류 모델 선택은 데이터의 특성에 따라 달라져야 한다. 또한, 모델 선택 시 속도와 정확도의 균형을 고려해 모델을 유연하게 사용할 수 있을 것이다. 정확도의 경우, 다중 클래스 분류 문제에서 전체 클래스를 동일한 우선순위로 평가한다면 Macro F1 지표를 우선적으로 고려하는 것이 성능 해석에 유리할 수 있다. 향후 연구로는 데이터에 특성 추출과 데이터 증강 기법의 순서에 따른 성능 비교하고자 하며, 다양한 축소 기법과 증강 기법과 이들 간의 조합에 대한 성능도 비교분석하고자 한다.

사사문구

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 ICT혁신인재4.0 사업의 연구결과로 수행되었음 (IITP-2024-RS-2022-00156299)

참고문헌

[1] M. E. Ozer, P. O. Sarica and K. Y. Arga, “New Machine Learning Applications to Accelerate Personalized Medicine in Breast Cancer: Rise of the Support Vector Machines,” Omics: A Journal of Integrative Biology, Vol.24, No.5, pp.241-246, 2020.

[2] W. Zhu, L. Xie, J. Han and X. Guo, “The Application of Deep Learning in Cancer Prognosis Prediction,” Cancers, Vol.12, No.3, pp.603, 2020.

[3] I. T. Jolliffe, “Principal Component Analysis and Factor Analysis,” Principal Component Analysis, pp.150-166, 2002.

[4] C. M. Bishop and N. M. Nasrabadi, “Pattern Recognition and Machine Learning,” New York, Springer, 2006.

[5] A. Liaw and M. Wiener, “Classification and Regression by RandomForest,” R News, Vol.2, No.3, pp.18-22, 2002.

[6] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, USA, 2016, pp.785-794.

[7] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma and T. Y. Liu, “LightGBM: A Highly Efficient Gradient Boosting Decision Tree,” In Advances in Neural Information Processing Systems, Long Beach, USA, 2017, pp.3146-3154.

[8] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush and A. Gulin, “CatBoost: Unbiased Boosting with Categorical Features,” In Advances in Neural Information Processing Systems, Montreal, Canada, 2018, pp.6638-6648.

[9] V. Vapnik, “Support-Vector Networks,” Machine Learning, pp.273-297, 1995.

[10] T. Cover and P. Hart, “Nearest Neighbor Pattern Classification,” IEEE Transactions on Information Theory, Vol.13, No.1, pp.21-27, 1967.

[11] I. Goodfellow, Y. Bengio and A. Courville, “Deep Learning,” Cambridge, MIT Press, 2016.