분류 복잡도를 활용한 오버 샘플링 비율 산출 알고리즘 개발

이도현*, 김경옥**
*서울과학기술대학교 데이터사이언스학과
**서울과학기술대학교 산업공학과
e-mail: skypo1000@ds.seoultech.ac.kr,
kyoungok.kim@seoultech.ac.kr

A Study on Calculating Over-sampling Ratio using Classification Complexity

Do-Hyeon Lee*, Kyoungok Kim**
*Dept. of Data Science, Seoul National University of Science and Technology
**Dept. of Industrial and Information Systems Engineering, Seoul National University of Science and Technology

요 약

불균형 데이터는 범주에 따른 데이터의 분포가 불균형한 데이터를 의미한다. 이런 데이터를 활용해 기존 분류 알고리즘으로 분류기를 학습하면 성능이 저하되는 문제가 발생한다. 오버 샘플링은 이를 해결하기 위한 기법 중 하나로 수가 적은 범주[이하 소수 범주]에 속한 데이터 수를 임의로 증가시킨다. 기존 연구들에서는 수가 많은 범주[이하 다수 범주]에 속한 데이터 수와 동일한 크기만큼 증가시키는 경우가 많다. 이는 증가시키는 샘플의 수를 결정할 때 범주 간 데이터 수 비율만 고려한 것이다. 그런데 데이터가 동일한 수준의 불균형 정도를 갖더라도 범주별 데이터 분포에 따라서 분류 복잡도가 다르며, 경우에 따라 데이터 분포에서 존재하는 불균형 정도를 완전히 해소하지 않아도 된다. 이에 본 논문은 분류 복잡도를 활용해 데이터 셋 별 적정 오버 샘플링 비율을 산출하는 알고리즘을 제안한다.

1. 서론

불균형 데이터는 범주에 따른 데이터의 분포가 불균형한 데이터를 뜻한다. 대부분의 분류 알고리즘은 범주 간 데이터의 수가 균형을 이룰 때 좋은 성능을 낼 수 있기 때문에 범주가 불균형한 분포를 가지는 데이터로 분류 알고리즘을 학습한 경우 해당 분류기의 성능이 저하된다. 그렇지만, 사기 감지, 질병 진단 등을 위한 많은 데이터들이 실제로 데이터의 수가 적은 소수 범주와 데이터 수가 많은 다수 범주로 이뤄진 사례가 많다. 이에 불균형 데이터에서도 높은 성능의 분류기를 학습할 수 있는 많은 연구가 진행되었다.

대표적으로 오버 샘플링은 수가 적은 범주에 속한 데이터 수를 임의로 증가시키며, 앙상블은 여러 개의 분류 모델을 학습시켜 이들을 종합적으로 활용하는 방식으로 불균형 정도를 해소한다. 최근에는 이들을 결합하는 연구들이 진행되었다. 대표적으로 Boosting에 오버 샘플링을 적용한 알고리즘을 예로 들 수 있다[1],[2],[3]. 매 약한 학습기 학습 시 이들의 입력 데이터에서 소수 범주에 속한 샘플 수를 증가시켜 매 학습에서 성능 개선이 가능하다.

그런데 오버 샘플링 시 사용자는 소수 범주에 속하는 데이터 수를 얼마나 늘릴지를 결정해야 한다. 기존의 연구들은 주로 다수 범주에 속하는 데이터 수와 동일하도록 그 수를 조절 한다[1],[2],[3],[6]. 즉, 이것은 다수 범주와 소수 범주 사이의 비율을 이용해서 오버 샘플링 비율을 결정하는 것이다.

그러나 동일한 데이터 수 불균형 정도를 갖는 데이터라도 서로 다른 분류 복잡도를 가지게 되고, 최종 분류기의 성능 또한 달라질 수 있다. 특히 Boosting에 오버 샘플링을 적용한 알고리즘의 경우 학습 시간에도 영향을 끼칠 수 있다.

앙상블 계열 기법에서는 이미 두 범주 간 데이터 분포가 꼭 완전 균형을 이루지 않아도 됨을 증명 한 바 있다[4].

이를 반영하여 본 논문에서는 불균형 정도와 분류 복잡도를 고려해, 완전 균형 상태보다 좋은 성능을 내지만 낮은 수준의 오버 샘플링 비율을 산출하는 알고리즘을 제시한다. 추가적으로 분류 복잡도를 구성하는 요인 간 비교 실험을 통해 더 나은 요인을

탐색한다.

2. 관련연구

2.1 불균형 데이터 문제 해결을 위한 연구

불균형 데이터 문제 해결을 위해 제안된 방법론들은 전체적으로 전처리, 알고리즘 및 앙상블 기법으로 구분할 수 있다[6]. 전처리 기법은 임의로 각 범주에 속하는 데이터 수를 조정한다. 알고리즘 기법은 수가 극단적으로 적은 범주의 데이터를 올바르게 분류하도록 cost를 의도적으로 부여한다. 앙상블 기법에서는 여러 개의 분류 모델을 학습시켜 이들의 결과를 종합적으로 활용한다.

오버 샘플링은 전처리 기법 중 하나로, 소수 범주에 속한 데이터 수를 임의로 증가시켜 데이터 불균형 정도를 해소한다. 최근에는 이를 알고리즘 및 앙상블 단계에서 제안된 기법들과 결합하여 사용하는 연구가 진행되었다. 대표적으로 Boosting에 이를 적용한 알고리즘을 예로 들 수 있다.

앙상블 알고리즘 중 하나인 Boosting 알고리즘은 순차적으로 약한 학습자를 학습시키고, 이들의 결과를 결합하여 강한 학습자를 얻는다. 이 과정에서 오버 샘플링을 적용시키면 매 약한 학습자를 학습시킬 때마다 입력 데이터의 불균형을 해소하여 분류기의 성능 개선이 종합적으로 발생할 수 있다.

2.2 분류복잡도

분류 복잡도는 범주별 데이터들의 변수 분포에 따라 분류 경계가 명확한 정도를 의미한다[5]. 일반적으로 분류 모형은 학습을 통해 서로 다른 범주를 구분할 기준이 되는 분류 경계선을 생성한다. 이는 범주별 일종의 구역 할당을 의미하며, 분명하고 명확할수록, 기대 성능은 높아질 수 있다.

이들은 주로 근접 이웃을 통해 설정한 지역 내에서의 동일 범주 간 데이터 분포 밀도 정보 및 개별 변수 특성을 활용하여 표현할 수 있다.

근접 이웃을 사용한 지표는 대표적으로 *cohesion*(응집도)[7]와 *noise*(이상도)[8]가 있다. *cohesion*은 k 개의 근접 이웃 중 같은 범주에 속한 경우의 확률적 표현이다. 이는 동일 범주 데이터의 분포 밀도를 의미한다. 따라서 최종 산출 분류 복잡도와 역비례 관계를 가진다. 이는 T_1 으로 표기하며 수식 (1)로 정의된다. 추가적으로 다수 범주는 N, 다수범주 데이터 수는 n, 소수 범주는 P, 소수범주 데이터 수는 p, 각 데이터별로 활용할 근접 이웃의 수는 k로 표기한다.

$$T_1 = 1 - \frac{1}{n,k}\sum_{x\in P}^{n}\sum_{r=1}^{k} I_r(x, S = P \cup N) \quad (1)$$

$$I_r(x,S) = \begin{cases} 1, \text{if } x \in P \text{ and } NN_r(x,S) \in P \\ 0 \ otherwise \end{cases} \quad (2)$$

범주별 개별 변수 특성을 이용하는 지표는 변수별 정보를 활용하여 범주별 데이터 분포를 측정한다. 개별 변수 별로 측정 후, 대표 값을 선정한다. 대표적으로 Fisher's Discriminant Ratio(이하 $f1$)와 Feature Efficiency가 있다[5].

$f1$은 범주 별 구별 가능성을 측정한다. 이는 수식 (4)로 정의 되며, 범주별 분산 합의 제곱 대비 범주별 평균의 차이의 제곱의 비율을 의미한다[5]. 범주별 평균의 차이는 범주 간 구별 가능성, 분산 합은 데이터 산포도 정도를 의미한다. $f1$에 기반을 둔 분류 복잡도 T_2는 수식 (3)으로 정의된다. 변수 수는 Z, 개별 변수의 $f1$은 $f1_z$, 개별 범주에서의 소수범주 및 다수 범주 평균과 표준 편차는 각각 μ_{P_z}, μ_{N_z}, σ_{P_z}, σ_{N_z}로 표기한다.

$$T_2 = \frac{1}{1+\max(f1_z)} \quad (3)$$

$$f1_z = \frac{(\mu_{P_z} - \mu_{N_z})^2}{(\sigma_{P_z})^2 + (\sigma_{N_z})^2} \quad (4)$$

$f2$는 개별 변수에서의 범주별 최솟값 중 최댓값과 최댓값 중 최솟값 사이에 모든 범주 데이터 셋의 개수를 측정한다[5]. 이는 데이터의 중첩의 정도를 나타낸다. 높은 중첩의 정도는 범주별 뚜렷한 구분 없이 데이터가 분포됨을 의미한다. 대표 $f2$은 개별 변수들의 $f2$ 평균값으로 선정한다.

3. 제안 알고리즘

3.1 용어정의

불균형 정도란 소수범주 대비 다수범주 데이터 수의 비율을 의미한다. 오버샘플링 비율(O_r)이란 오버샘플링을 통해 달성하고자 하는 다수범주 대비 소수범주 데이터 수의 비율을 의미한다. O_r이 1일 때, 데이터의 분포가 균형 함을 의미한다.

3.2 분류복잡도 요인 선정

데이터 밀도 정보 및 개별 변수 별 특성을 반영하는 요인을 하나씩 선정한다. 전자는 데이터 간 분포 밀도를 반영하므로, 전체 변수 정보를 모두 고려하는 반면. 후자는 특정 변수에서의 데이터 분포를 고려한다. 이를 통해 다른 성격의 분류 복잡도 요인

간 비교가 용이해진다. 이들은 모두 아래 수식에서 T로 표기한다.

3.2 알고리즘 설계

기본적으로 알고리즘을 통해 산출되는 오버 샘플링 비율(O_r)은 앞서 구한 분류 복잡도와 비례 관계를 가지므로 T 로 나타낸다. 단 산출된 오버 샘플링 비율을 적용했을 때, 소수 범주의 최종 크기가 기존 보다 작으면 이를 기존 소수 범주 크기가 되도록 조정했다. 이는 수식 (5)으로 정의된다.

$$O_r = \begin{cases} T, & \text{if } T > \frac{p}{n} \\ \frac{p}{n} & otherwise \end{cases} \quad (5)$$

4. 실험

4.1 실험 데이터

본 실험에서는 "UCI Machine Learning Repository"에서 공개된 불균형 데이터 셋들을 사용하였다. 이들의 특성은 표 1.에 정리되어 있다. 특히 다중 분류 데이터는 기존 연구들과 동일하게 소수 범주 선정 후, 데이터 명의 괄호 안에 표기하였다. 오버샘플링 및 분류학습 시 범주형 독립변수는 제외하였다.

4.2 실험설계

<표 1> 데이터 셋 특성요약

데이터명 (소수범주명)	다수범주 데이터수	소수범주 데이터수	변수 수	불균형 정도
Abalone(7)	3786	391	7	9.68
Cm	449	48	22	9.35
Ecoli(imU)	301	35	8	8.60
Glass(table)	205	9	9	22.78
Ionosphere	225	126	34	1.79
Isolet(A,B)	7197	600	617	12.00
Kc	1783	326	21	5.47
Letter Img(Z)	19266	734	16	26.25
Mammography	10923	260	6	42.01
Oil	896	41	47	21.85
Optical Digits (8)	5066	554	64	9.14
Pc	4817	113	6	42.63
Satimage(4)	5809	626	36	9.28
Segment	1980	330	20	6.00
Spectrometer (44이상)	486	45	93	10.80
Us Crime (0.65초과)	1844	150	100	12.29
Wine Quality (4이하)	4715	183	11	25.77
Yeast Me2 (ME2)	1433	51	8	28.10

본 논문에서 제안한 알고리즘과 완전 균형 상태(O_r이 1인 상태) 간 분류 성능 평가를 위해 4.1에서 제시한 18개의 데이터 셋을 사용하였다. 산출된 오버 샘플링 비율은 소수점 한 자리로 올림 후 사용하였다. 분류 학습자로 Boosting에 오버 샘플링을 적용한 알고리즘(SMOTEBoost [1], RAMOBoost [2], WOTBoost [3])을 사용했다. 또한 강건성이 높은 AUC(Area under the ROC Curve)를 주요 성능지표로 사용하였다. 일반적으로 확률 기반 분류 학습자는 최종 분류에 앞서, 각 범주에 속할 확률을 출력하는데, AUC는 모든 cutoff(기준선)에서의 결과를 반영하기 때문이다.

성능검증 방법으로는 층화 추출을 통한 5-겹 교차을 20번 반복하였다. 오버 샘플링 및 분류 복잡도 산출에 필요한 근접 이웃 수(k)는 5로 설정하였다.

4.3 실험 결과

요인 별 산출 오버 샘플링 비율과 최종 분류 성능 결과는 표 2.에 정리되어 있다. 특히 최종 분류성능 비교에서 분류 학습자 별로 가장 성능이 좋은 경우는 굵은 글씨로 표시했다. 또한 표 2.의 Win Count 항목에서 요인 간 분류 성능 비교 결과를 나타냈다.

먼저 Glass, Oil, Yest Me2를 제외한 모든 실험 데이터 셋에서 하나 이상의 분류 학습자를 대상으로 완전 균형일 때 보다 높은 성능을 보였다. 성능이 낮은 경우에서도 아주 근소한 차이를 보였다. Cm, Segment의 경우에서 낮은 비율로도 높은 성능을 기대할 수 있음을 알 수 있다. 추가적으로 진행한 요인 간 비교 실험에서 산출 비율에서의 차이는 대체로 0.5 이하로 나타났다. $cohesion$의 경우 산출 비율이 높아질수록 공통적으로 낮은 수준의 분류 성능을 기대할 수 있음을 알 수 있다. 각 요인 별 Win Count 및 분류 복잡도 설명력을 종합했을 때, $cohesion$이 더 나은 요인임을 알 수 있다.

5. 결론

Boosting에 오버 샘플링을 적용한 알고리즘에 관한 기존 연구에서는 일반적으로 오버 샘플링 비율을 데이터 불균형 정도만을 고려하여 완전 균형(O_r이 1인 상태)로 설정하였다. 본 논문에서는 분류 복잡도를 고려해 오버 샘플링 비율을 산출하는 알고리즘을 제안하였다. 완전 균형과의 분류 성능 실험 비교 결과 Glass, Oil, Yest Me2를 제외한 모든 데이터 셋에서 완전 균형 이상의 분류 성능을 보였다. 특히

<표 2> 요인 별 산출 오버샘플링 비율 및 분류성능(AUC) 비교

비교항목	산출 비율		분류 성능								
			Smote			Ramo			Wot		
요인구분 / 데이터명	cohesion	f1	cohesion	f1	균형	cohesion	f1	균형	cohesion	f1	균형
Abalone	0.7	0.5	**0.8532**	0.8507	0.8529	**0.8474**	0.8473	0.8457	0.8483	**0.8486**	0.8472
Cm	0.2	0.3	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
Ecoli	0.3	0.4	0.9961	0.9960	**0.9968**	**0.9961**	0.9959	0.9960	0.9960	**0.9967**	0.9959
Glass	0.5	0.5	0.9863	0.9863	**0.9878**	0.9865	0.9865	**0.9871**	0.9862	0.9862	**0.9873**
Ionosphere	0.6	0.6	**0.9551**	**0.9551**	0.9544	**0.9515**	**0.9515**	0.9490	**0.9507**	**0.9507**	0.9473
Isolet	0.3	0.6	**0.9731**	0.9727	0.9715	**0.9730**	0.9715	0.9691	**0.9728**	0.9715	0.9691
Kc	0.7	0.6	0.7889	0.7876	**0.7909**	0.7825	**0.7863**	0.7791	0.7853	**0.7868**	0.7798
Letter Img	0.1	0.4	**0.9948**	0.9922	0.9828	**0.9950**	0.9945	0.9932	**0.9950**	0.9948	0.9930
Mammography	0.4	0.5	**0.9247**	0.9222	0.9188	0.9326	0.9318	**0.9341**	0.9314	0.9323	**0.9333**
Oil	0.6	0.4	0.8613	0.8527	**0.8656**	0.8637	0.8475	**0.8824**	0.8578	0.8561	**0.8944**
Optical Digits	0.2	0.6	0.9797	**0.9795**	0.9775	**0.9792**	**0.9792**	0.9775	**0.9796**	0.9794	0.9779
Pc	1.0	0.9	0.7051	**0.7100**	0.7051	0.7069	**0.7070**	0.7069	0.7041	**0.7106**	0.7041
Satimage	0.3	0.7	0.9268	**0.9283**	0.9258	**0.9249**	0.9233	0.9215	**0.9257**	0.9244	0.9230
Segment	0.2	0.2	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
Spectrometer	0.3	0.5	0.9578	0.9560	**0.9583**	0.9559	0.9548	**0.9565**	0.9550	**0.9585**	0.9557
Us Crime	0.7	0.3	0.8978	0.8975	**0.9026**	0.8837	**0.8884**	0.8732	0.8817	**0.8905**	0.8724
Wine Quality	0.8	0.8	**0.8202**	**0.8202**	0.8191	**0.8077**	**0.8077**	0.8061	**0.8071**	**0.8071**	0.8063
Yeast Me2	0.8	0.4	0.8503	0.8265	**0.8672**	0.8431	0.8188	**0.8537**	0.8425	0.8164	**0.8596**
Win Count	분류 학습자별 측정		9	2	7	9	4	5	8	6	4

본 실험에서 사용한 Boosting에 오버 샘플링을 적용한 알고리즘의 경우 학습 시간이 약한 학습자 수와 곱에 비례하는 특성을 고려했을 때, 해당 논문의 알고리즘은 학습 시간 단축 면에서도 활용 가능하다. 추가적으로 진행한 요인 간 비교 실험을 통해 요인 간 산출 비율에서의 차이는 대체로 0.5 이하로 나타났다. cohesion의 경우 산출 비율이 높아질수록 공통적으로 낮은 수준의 분류 성능을 기대할 수 있음을 알 수 있다. 요인 별 산출 오버 샘플링 경향과 분류 성능을 종합했을 때, *cohesion*이 더 나은 요인이라고 결론지을 수 있다.

추후 연구에서는 요인 별 상반된 비율에 대한 원인 분석을 수행하고자 한다. 또한 Boosting에 적용할 오버 샘플링 기법의 다양성 측면에서도 연구를 수행하고자 한다. 본 논문에서 사용한 분류 학습자는 초기 기법(Smote)을 기반으로 설계되었기 때문이다. 마지막으로 분류 복잡도 산출에 필요한 근접 이웃 수(k)에 대한 파라미터 분석(Parameter Analysis)을 수행하고자 한다.

참고문헌

[1] Chawla, Nitesh V., et al, "SMOTEBoost: Improving prediction of the minority class in boosting.", European conference on principles of data mining and knowledge discovery. Springer, Berlin, Heidelberg, 2003, pp. 107-119.

[2] Chen, Sheng, Haibo He, and Edwardo A. Garcia, "RAMOBoost: ranked minority oversampling in boosting.", IEEE Transactions on Neural Networks 21.10, 2010, pp. 1624-1642.

[3] Zhang, Wenhao, Ramin Ramezani, and Arash Naeim, "WOTBoost: Weighted Oversampling Technique in Boosting for imbalanced learning.", arXiv preprint, 2019, arXiv:1910.07892 .

[4] Hido, Shohei, Hisashi Kashima, and Yutaka Takahashi. "Roughly balanced bagging for imbalanced data." Statistical Analysis and Data Mining: The ASA Data Science Journal 2.5 6, 2009, pp. 412-426.

[5] Lorena, Ana C., et al, "How Complex is your classification problem? A survey on measuring classification complexity.", ACM Computing Surveys (CSUR) 52.5, 2019, pp. 1-34.

[6] Shakeel, Fatima, A. Sai Sabhitha, and Seema Sharma, "Exploratory review on class imbalance problem: An overview.", 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT). IEEE, 2017.

[7] Alkhalid, Abdulaziz, Mohammad Alshayeb, and Sabri Mahmoud, "Software refactoring at the function level using new Adaptive K-Nearest Neighbor algorithm.", Advances in Engineering Software 41.10-11, 2010, pp. 1160-1178.

[8] Han, Hui, Wen-Yuan Wang, and Bing-Huan Mao., "Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning.", International conference on intelligent computing. Springer, Berlin, Heidelberg, 2005.