CDBSMOTE : 클래스와 밀도기반의 합성 소수 오버샘플링 기술

배경환*, 이경현**
*부경대학교 전산교육학과, **부경대학교 IT융합응용공학과
kyunghwan.abel.bae@gmail.com, yisecure@gmail.com

CDBSMOTE : Class and Density Based Synthetic Minority Oversampling Technique

Kyung-Hwan Bae*, Kyung-Hyune Rhee**
*Dept. of Computer Science Education, **Dept. of IT Convergence & Application Engineering, Pukyong National University

요 약

머신러닝의 성능 저하에 크게 영향을 미치는 데이터 불균형은 데이터를 증강하거나 제거하여 해결할 수 있다. 본 논문에서는 지도학습에서 쓰이는 정답 데이터를 기반으로 새로운 데이터 증강기법인 CDBSMOTE을 제안한다. CDBSMOTE을 사용하면 임의의 값을 사용하지 않고, 기존의 데이터 증강 기법의 문제점이었던 과적합을 최소화하며 지도학습 데이터를 효과적으로 증강시킬 수 있다.

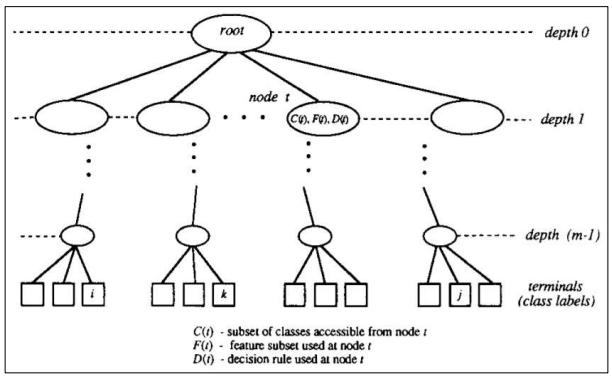
1. 서론

머신러닝은 크게 지도학습, 비지도학습, 강화학습으로 분류할 수 있다. 지도학습은 인간이 정의한 정답 데이터를 기반으로 전체 데이터를 학습하는 방식이며, 비지도학습은 인간이 정해둔 정답 없이 데이터를 학습한다. 본 논문에서는 지도학습에 사용하는 데이터를 비지도학습 알고리즘인 DBSCAN에 적극 반영하여 효과적인 데이터 증강을 하는 CDBSMOTE에 대해서 제안한다. 그리고 증강된 데이터들은 Decision Tree를 통해서 검증함으로 CDBSMOTE의 성능을 평가한다.

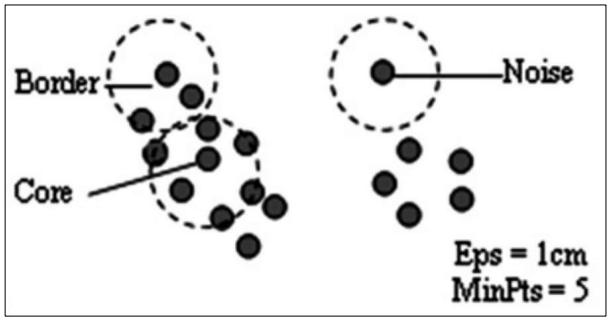
2. 배경

2.1 Decision Tree

Decision Tree는 (그림 1)과 같이 정보획득을 최대로 하는 방식으로 마치 가지를 치듯 트리를 형성해가는 지도학습 알고리즘이다. Decision Tree는 보통 데이터 분류를 위해서 사용되며, 정보획득이 불가능할 때까지 분기하게 되면 과적합이 일어나게 된다. 과적합을 해결하기 위해서, 대표적으로 '가지치기' 하여 데이터를 줄이거나 다수 클래스에 속한 데이터를 삭제하는 언더샘플링 기법을 적용한다. B.A Shawer[2]는 ENN을 포함한 다양한 언더샘플링 기법을 연구하여 과적합 문제를 해결하고 있다. 하지만 언더샘플링 기법은 정보가 소실되는 문제가 있다.



(그림 1) Decision Tree의 구조[1]



(그림 2) DBSCAN의 구조[3]

2.2 DBSCAN

DBSCAN(Density-Based Spatial Clustering of

Applications with Noise)은 인간이 정해둔 정답데이터 없이 학습하는 비지도학습 알고리즘 중 하나이다. DBSCAN의 특징은 데이터의 밀도에 따라 클러스터링을 수행하는 알고리즘이다. (그림 2)와 같이 DBSCAN은 기준 데이터로부터의 반경인 Eps(ε)에 MinPts만큼의 데이터가 충족된다면 하나의 클러스터를 구성할 수 있다. 이때 Eps안에 MinPts를 모두 만족하면 핵심 데이터(Core Data)라고 하고, Eps안에 클러스터로 분류되는 데이터는 존재하되 MinPts를 만족하지 못하는 경우를 외곽 데이터(Border Data)라고 한다. 그리고 Eps안에 클러스터로 분류되는 데이터가 없는 경우 노이즈 데이터(Noise Data)라고 한다.

2.3 오버샘플링

데이터의 분포가 불균형할 때 사용하는 데이터 전처리 방식 중 데이터를 증강해서 데이터의 균형을 맞추는 방식이다. 데이터의 손실을 줄일 수 있으나 과적합을 일으킬 수 있는 단점이 있다. 대표적인 알고리즘은 SMOTE(Synthetic Minority Oversampling Technique)[4]이며, SMOTE는 소수 클래스의 데이터 중 일부를 무작위로 선택하여 데이터를 증강한다. 데이터의 비율은 높일 수 있으나 과적합을 초래한다. 이에 개선된 방식으로는 Borderline-SMOTE[5]이 대표적이다. Borderline-SMOTE에서는 다수 클래스와 소수 클래스의 경계 부분에서 데이터를 증강하여 과적합을 해결하려고 한다. 하지만 클래스의 경계는 사람에 의해서 정해진 것이며, 데이터의 밀도가 높을 수도 있기 때문에 증강을 한다고 해도 과적합을 해결할 수 있는 기준은 될 수 없다. DBSMOTE[6]은 DBSCAN을 기반으로 SMOTE을 하고 있지만, 데이터 증강에서 지도학습의 클래스 분포를 참고하는 것이 아니라 적절한 위치에서 무작위의 데이터를 생성한다. 무작위의 데이터 증강은 알고리즘이 실행할 때마다 다른 결과를 보여주며 노이즈로 처리되는 데이터를 생성해낼 가능성이 매우 높다.

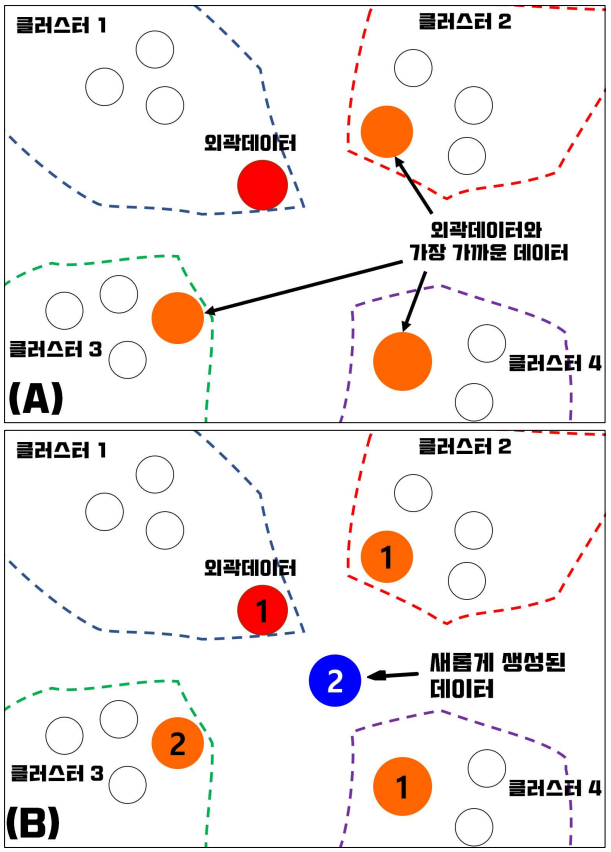
3. CDBSMOTE

본 논문에서 제안하는 알고리즘인 CDBSMOTE은 지도학습의 정답데이터를 기반으로 비지도학습인 DBSCAN의 클러스터링을 설계한다. 그리고 클러스터링을 통해 데이터의 밀도가 가장 낮은 곳에서 지도학습의 정답데이터를 기반으로 가장 소수 클래스의 데이터를 증강함으로 전체 데이터의 균형을 맞추는 것이다. 증강된 데이터는 지도학습 알고리즘 중 Decision Tree를 통해서 정확도를 검출할 수 있다.

```
알고리즘: CDBSMOTE
입력: D: 정답데이터가 있는 데이터세트
출력: O: D에서 증강된 데이터

1.  [Class] = Normalize(D)
2.  For MinPts = minCount(Class) to 3  step -1
3.    For ε = 0 to 1 step 0.01
4.      [Cluster] = DBSCAN(D, ε, MinPts)
5.      If Class의 정답 종류와 생성된 Cluster의 개수가
        같다 then
6.        [C] = Cluster // C : 데이터 증강에 사용할
          클러스터 집합
7.        break
8.    End for
9.    If C가 존재한다 then
10.     break
11. End for
12. borders = findBorder(C)
13. For j = 1 to count(borders)
14.   b = borders_j
15.   closestItems = findClosestItems(b)
16.   minCountClassLabel = findMinCountClassLabel(closestItem)
17.   O_j = Augment(average(closestItems), minCountClassLabel)
18.   O = ∪ O_j
19. End for
```

(그림 3) CDBSMOTE 알고리즘 의사코드



(그림 4) (A) 데이터 증강에 활용 될 데이터
(B) 새롭게 생성된 데이터의 값과 클래스

(그림 3)의 알고리즘을 통해 오버샘플링되는 과정을 이해할 수 있다. 먼저 데이터는 0~1 사이로 정규화되고(1번째 줄), 주어진 지도학습 데이터에 따라 클러스터링이 이뤄진다(2~11번째 줄). 여기에서 DBSCAN의 클러스터들이 가장 정확하기 분류되었다고 하기 위해서는 지도학습 데이터의 클래스 분포와 일치해야 한다. 그래서 DBSCAN에서 클러스터링을 할 MinPts의 최대로 허용되는 개수는 지도학습의 소수 클래스의 개수라고 할 수 있으며, 클러스터 생성을 위한 최소 개수인 3까지의 범위를 가지게 된다(2번째 줄). Epsilon(ε)은 데이터 전처리 과정을 통해 정규화된 데이터 때문에 0~1 사이의 값을 가진다(3번째 줄). 적합한 클러스터를 찾기 위해서 클러스터링을 하기 어려운 최대의 MinPts와 최소의 ε으로 시작하여, 반대되는 최소의 MinPts와 최대의 ε으로 클러스터링을 진행한다. 반복문은 결과로 도출된 클러스터의 개수가 지도학습데이터의 클래스가 처음으로 일치할 때 종료되며, 그때의 클러스터링 된 데이터를 기준으로 데이터 증강을 한다.

(그림 4)에서는 데이터 증강을 하는 방식을 표현했다. 먼저 (그림 4)의 (A)를 보면, 데이터 밀도가 낮은 곳을 찾기 위해서 클러스터의 결과 중 추출한 외곽데이터를 추출한다(12번째 줄). 그리고 각 클러스터에서 추출한 외곽데이터와 가장 가까운 데이터를 찾는다(15번째 줄). 다음 (그림 4)의 (B)에서는 선택된 외곽데이터와 가장 가까운 데이터의 값들의 유클리디안 거리를 기준으로 평균값을 도출해 데이터를 증강한다. 그리고 증강된 데이터의 클래스는 데이터 증강에 사용한 데이터들의 클래스를 조사하여 가장 소수로 집계되는 값으로 클래스를 결정한다(16~17번째 줄). 증강된 데이터는 기존의 데이터세트에 추가되며(18번째 줄), 각각의 외곽의 데이터들에 대해서 같은 방식으로 데이터 증강을 시도한다.

4. 실험

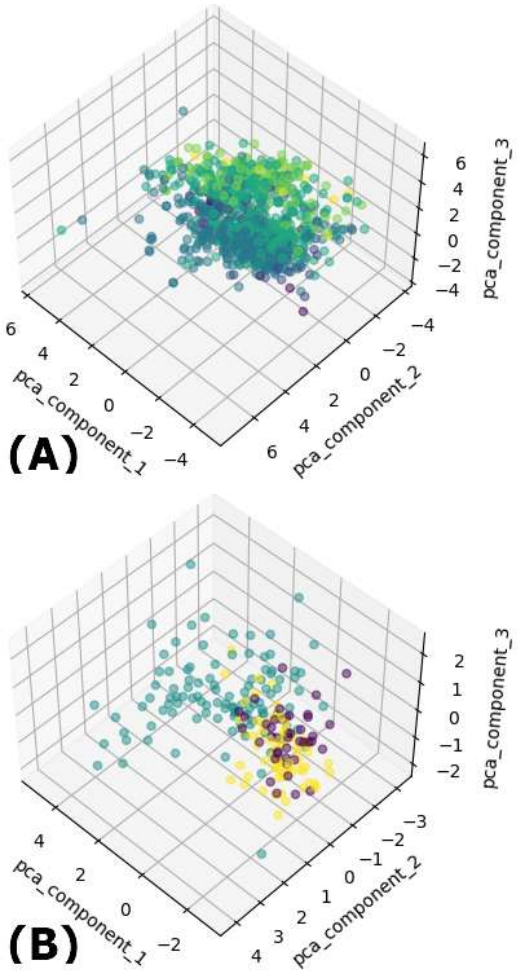
실험용 데이터세트는 UCI Machine Learning 저장소[7]에서 5개와 Kaggle의 'Star Type Classification / NASA' 데이터[8] 1개로 구성되어있으며, 데이터 증강의 결과와 Decision Tree의 정확도를 측정하여 (표 1)과 같이 정리하였다.

실험에서는 소수 클래스가 정상적으로 증강된 6번 데이터세트는 정확도가 상승하였으며, 다른 1~4번 데이터세트들은 데이터가 증강하였으나 정확도가 상승하거나 기존의 정확도를 유지하였다. 5번 데이터세트는 데이터 증강이 소수 클래스가 아닌 다수 클래스에서 이뤄져서 오히려 정확도가 떨어지는 결과를 보여준다.

(표 1) CDBSMOTE의 데이터증강 결과

데이터세트		데이터 증강 전		데이터 증강 후	
		데이터 분포	정확도	데이터 분포	정확도
1	Stars	0: 40 1: 40 2: 40 3: 40 4: 40 5: 40	0.736	0: 40 1: 47 2: 47 3: 46 4: 47 5: 47	0.819
2	glass	1: 70 2: 76 3: 17 5: 13 6: 9 7: 29	0.585	1: 70 2: 76 3: 31 5: 13 6: 9 7: 31	0.585
3	Iris	0: 50 1: 50 2: 50	0.911	0: 69 1: 67 2: 50	0.911
4	breast cancer diagnostic	0: 212 1: 357	0.930	0: 301 1: 376	0.930
5	wine quality	3: 10 4: 53 5: 681 6: 638 7: 199 8: 18	0.565	3: 10 4: 53 5: 681 6: 658 7: 199 8: 18	0.560
6	biomechanical features of orthopedic patients	0: 60 1: 150 2: 100	0.828	0: 100 1: 150 2: 100	0.860

실험 결과를 분석하기 위해, 대표적으로 성능이 좋지 않게 나온 5번 데이터세트와 성능이 좋게 나온 6번 데이터세트를 PCA차원감소를 하여 3차원으로 매칭을 하면 (그림 5)와 같이 확인할 수 있다. (그림 5)의 (A)에서는 다수의 클래스를 가지고 있으며 클래스와 무관하게 데이터끼리 밀집해있다. 또한, 데이터 불균형이 크다. 이러면 외곽데이터를 기반으로 한 데이터 증강에서 소수 클래스를 찾을 수 없게 된



(그림 5) (A) 5번 데이터세트, (B) 6번 데이터세트

다. (B)에서는 데이터가 밀집해있는 구간이 있으나 데이터의 불균형이 크지 않다. 그래서 외곽데이터와 가까운 데이터 중에서 소수 클래스의 정보를 가지고 있는 데이터가 많아 소수 클래스에 대한 데이터증강이 가능하다.

5. 결론

CDBSMOTE에서는 지도학습에 사용하는 데이터를 적극 DBSCAN에 활용하여 데이터를 증강했다. CDBSMOTE의 데이터 중 임의로 결정되는 값은 없으며, 데이터의 밀도가 낮은 곳에서 소수 클래스의 데이터를 증강함으로 성능의 저하를 최소화한다. CDBSMOTE를 이용한 지도학습 데이터의 증강으로 항상 Decision Tree의 성능이 좋아진다고는 할 수 없지만, 성능이 개선될 가능성이 높고 과적합을 최소화하여 잘못된 증강이 이뤄져도 기존의 성능을 유지하는 모습을 기대할 수 있다. 하지만 지도학습에 사용하는 데이터의 분포가 아주 불균형하며 밀집해 있는 구조에서는 소수 클래스의 데이터가 외곽데이터에 의해서 발견될 가능성이 낮으므로 기존의 다수 클래스의 데이터를 증강하여 전체 성능이 저하될 수도 있다. 이 부분에 대해서는 다수 클래스에 속하는 데이터는 전체 데이터세트에 추가하지 않는다거나, 먼저 언더샘플링을 진행하고 난 뒤 데이터를 증강하는 식으로 알고리즘을 더욱 개선할 수 있을 것이다.

참고문헌

[1] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," IEEE Transactions on Systems, Man, and Cybernetics, vol. 21, no. 3, pp. 660-674, 1991

[2] Bayan Abu Shawar, Mohamed Habib, Khalil El Hindi, Mousa Al-Akhras, Asma' Amro "Instance Reduction for Avoiding Overfitting in Decision Trees" Journal of Intelligent Systems 30 (1):438-459, 2021

[3] Derya Birant, Alp Kut, "ST-DBSCAN: An algorithm for clustering spatial - temporal data", Data & Knowledge Engineering, Volume 60, Issue 1, Pages 208-221, 2007

[4] N.V. Chawla, K.W. Bowyer, L.O. Hall, and W.P. Kegelmeyer, "SMOTE: Synthetic Minority Over-Sampling Technique," J. Artificial Intelligence Research, vol. 16, pp. 321-357, 2002

[5] H. Han, W.Y. Wang, and B.H. Mao, "Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning," Proc. Int'l Conf. Intelligent Computing, pp. 878-887, 2005

[6] Bunkhumpornpat, C., Sinapiromsaran, K. & Lursinsap, C. DBSMOTE: Density-Based Synthetic Minority Over-sampling TEchnique. Appl Intell 36, 664 - 684, 2012

[7] C. Blake and C. Merz. UCI Repository of Machine Learning Databases [Online]. Available: https://archive.ics.uci.edu/ml/index.php

[8] Baris Dincer, Star Type Classification / NASA, Kaggle [Online]. Available: https://www.kaggle.com/brsdincer/star-type-classification