Conditional GAN 을 활용한 오버샘플링 기법

손민재, 정승원, 황인준
고려대학교 전기전자공학과
e-mail:smj5668@korea.ac.kr, jsw161@korea.ac.kr, ehwang04@korea.ac.kr

Oversampling scheme using Conditional GAN

Minjae Son, Seungwon Jung, Eenjun Hwang
School of Electrical Engineering, Korea University

요 약

기계학습 분야에서 분류 문제를 해결하기 위해 다양한 알고리즘들이 연구되고 있다. 하지만 기존에 연구된 분류 알고리즘 대부분은 각 클래스에 속한 데이터 수가 거의 같다는 가정하에 학습을 진행하기 때문에 각 클래스의 데이터 수가 불균형한 경우 분류 정확도가 다소 떨어지는 현상을 보인다. 이러한 문제를 해결하기 위해 본 논문에서는 Conditional Generative Adversarial Networks(CGAN)을 활용하여 데이터 수의 균형을 맞추는 오버샘플링 기법을 제안한다. CGAN 은 데이터 수가 적은 클래스에 속한 데이터 특징을 학습하고 실제 데이터와 유사한 데이터를 생성한다. 이를 통해 클래스별 데이터의 수를 맞춰 분류 알고리즘의 분류 정확도를 높인다. 실제 수집된 데이터를 이용하여 CGAN 을 활용한 오버샘플링 기법이 효과가 있음을 보이고 기존 오버샘플링 기법들과 비교하여 기존 기법들보다 우수함을 입증하였다.

1. 서론

분류 문제는 기계 학습 분야에서 중요한 연구 주제 중 하나로, 입력 데이터가 주어졌을 때 해당 데이터의 클래스를 예측하는 문제를 일컫는다. 분류 문제에서 가장 중요한 점은 분류의 정확도로, 이 정확도를 높이기 위해 다양한 기계 학습 알고리즘들이 제안되었다. 그러나 클래스별 데이터 수의 차이가 큰 데이터 집합을 분류하고자 할 때 이들 중 대부분은 클래스를 제대로 분류하지 못하는 현상을 겪는다. 이러한 문제를 클래스 불균형 문제라고 한다.

클래스 불균형 문제는 기계 학습 알고리즘 대부분이 각 클래스에 포함된 데이터의 비율이 거의 같다고 가정하기 때문에 발생한다[1]. 기계 학습 알고리즘들이 클래스별 데이터 수가 불균형한 데이터 집합을 학습하면 상대적으로 데이터의 수가 많은 다수 클래스(Majority Class)에 영향을 많이 받는다. 반면, 데이터 수가 적은 소수 클래스(Minority Class)에 대한 학습은 거의 이뤄지지 않기 때문에 다수 클래스의 데이터를 입력했을 때는 잘 분류하지만 소수 클래스의 데이터는 잘 분류하지 못하게 된다. 이처럼 다수 클래스의 데이터 수가 많아 전체적인 분류 정확도가 높게 측정되면 클래스 불균형 문제가 주로 발생하는 텔레마케팅 고객 탐지[2], 금융 사기 탐지[3], 질병 탐지[4] 등의 데이터들처럼 소수 클래스의 분류를 정확히 해야 할 경우 문제가 된다. 따라서 소수 클래스의 데이터를 정확히 분류하기 위해서는 클래스 불균형 문제 해결이 필수적이다[5].

클래스 불균형 문제를 해결하는 방법으로 클래스의 균형을 맞춰 분류기의 성능을 높이는 샘플링(Sampling) 기법[6]이 주로 사용된다. 샘플링 기법에는 크게 두 가지 방법이 있다. 첫 번째는 다수 클래스의 데이터 수를 소수 클래스에 속하는 데이터의 수에 맞추는 언더샘플링(Under-sampling) 기법이고, 두 번째는 소수 클래스의 데이터 수를 다수 클래스 데이터 개수에 맞춰 균형을 이루는 오버 샘플링(Over-sampling) 기법이다. 언더샘플링의 가장 대표적인 방법은 Random Under-sampling(RUS)으로, 소수 클래스의 데이터 수에 맞게 무작위로 다수 클래스의 데이터를 제거한다. 그러나 이 기법은 무작위로 데이터를 제거하기 때문에 학습할 수 있는 정보가 손실된다는 단점이 있다.

오버 샘플링의 기본적인 방법으로는 Random Over-sampling(ROS)을 꼽을 수 있다. ROS 는 다수 클래스 레이블의 수에 맞춰 소수 클래스에서 무작위로 추출하여 복제하는 방법으로 다수 클래스와 소수 클래스의 데이터 수를 맞춘다. 그러나 소수 클래스의 데이터를 복제하기 때문에 학습 데이터에 과적합(Overfitting)되어 오히려 분류 정확도가 떨어지는 문제가 발생할 수 있다. 이 단점을 보완한 기법으로는 Synthetic Minority Over-sampling Technique(SMOTE)[7]가 제안되었다. SMOTE 는 ROS 처럼 데이터를 단순 복제 생성하지 않고 K-근접 이웃(K-nearest Neighbors) 알고리즘을 통하여 소수 클래스의 데이터와 가까운 거리에 데이터를 새로 생성한다. 따라서 ROS 와 달리 과적합을 피할 수 있지만, 이 기법 역시 문제점이 존재한다. 데이터 집합 내에는 다수 클래스의 데이터들보다 소수 클래스의 데이터들과 더 가까운 다수 클래스 데이터가 존재하기도 하는데 SMOTE 가 이러한 데이

터 근처에 데이터를 생성하는 경우가 종종 발생한다. 이 경우에 생성된 데이터는 기계 학습 알고리즘의 학습을 방해하는 요소가 될 수 있다.

본 논문에서는 클래스 불균형 문제를 해결하기 위해 ROS 나 SMOTE 와 달리 Conditional Generative Adversarial Networks(CGAN)[8]을 활용한 오버샘플링 기법을 제안한다. 주어진 데이터로 CGAN 을 학습시키고 학습된 CGAN 을 이용하여 오버샘플링한다. 이를 통해 다수 클래스와 소수 클래스 사이의 데이터 수 차이를 없앰으로써 클래스 불균형 문제를 해결한다. 실험을 통해 CGAN 을 이용한 오버샘플링이 클래스 불균형 문제를 해결할 수 있으며 기존의 오버 샘플링 기법인 ROS 와 SMOTE 보다 뛰어남을 입증한다.

본 논문의 구성은 다음과 같다. 2 장에서는 CGAN 을 활용한 샘플링 기법을 제안하고 3 장에서는 실험에 대한 결과를 제시한다. 이후 4 장에서는 결론을 맺고 향후 연구계획을 제시한다.

2. CGAN 을 활용한 샘플링 기법

Generative Adversarial Networks(GAN)[9]은 가상의 데이터를 생성하는 Generator 와 실제 데이터와 Generator 가 생성한 데이터를 구별하는 Discriminator 로 구성된 모델로, 이 두 모듈이 경쟁하면서 학습하게 되어 있다. Generator 는 최대한 실제 데이터같이 가상 데이터를 만들어 Discriminator 가 구별하기 힘들게 만들어야 하고 반대로, Discriminator 는 실제 데이터와 Generator 가 만들어낸 가상 데이터를 최대한 정확히 구별해내야 한다.

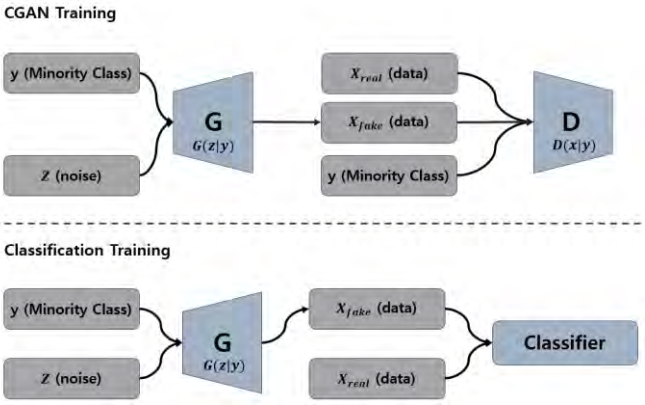
이를 반영한 GAN 의 손실함수 식은 식 (1)과 같다. 식에서 G 와 D 는 각각 Generator 와 Discriminator 을 의미한다. P_{data} 는 실제 데이터의 분포를 의미하고 x 는 실제 데이터 분포에서 뽑은 샘플 데이터이다. P_z 는 노이즈의 분포를 의미하며 z 는 P_z 에서 뽑은 임의의 노이즈 샘플 데이터를 의미한다. Discriminator 는 실제 데이터가 들어오면 1 을 출력하고 Generator 가 만든 가상의 데이터가 들어오면 0 을 출력하는 것을 목표로 한다. 따라서 식 (1)을 최대화하여 $D(x)$ 는 1, $D(G(z))$ 는 0 이 되도록 학습하려 한다. 반대로 Generator 는 Discriminator 가 자신이 생성한 데이터에도 실제 데이터처럼 1 을 출력하게끔 만들어야 하기 때문에 식 (1)을 최소화하여 $D(G(z))$ 이 1 이 되도록 학습하려 한다. 이론적으로 GAN 이 충분히 학습했다면 Generator 는 Discriminator 가 실제 데이터와 생성한 데이터를 구별해내지 못할 정도로 사실적인 데이터를 생성할 수 있게 된다.

$$min_G max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[logD(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))] \quad (1)$$

CGAN 은 GAN 에서 파생된 모델로 GAN 처럼 Generator 와 Discriminator 를 활용한 학습 방식은 유사하나 조건이 존재한다는 점이 다르다. 이로 인해 사용자가 원하는 특성을 조건으로 반영하여 특성에 맞는 데이터를 생성해낼 수 있다. CGAN 의 손실함수 식은 식 (2)와 같다. GAN 과 달리 조건이 존재한다는 점 때문에 식 (1)에서 $D(x)$ 와 $D(G(z))$ 에 y 라는 조건이 추가된 것을 확인할 수 있다.

$$min_G max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[logD(x|y)] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z|y)))] \quad (2)$$

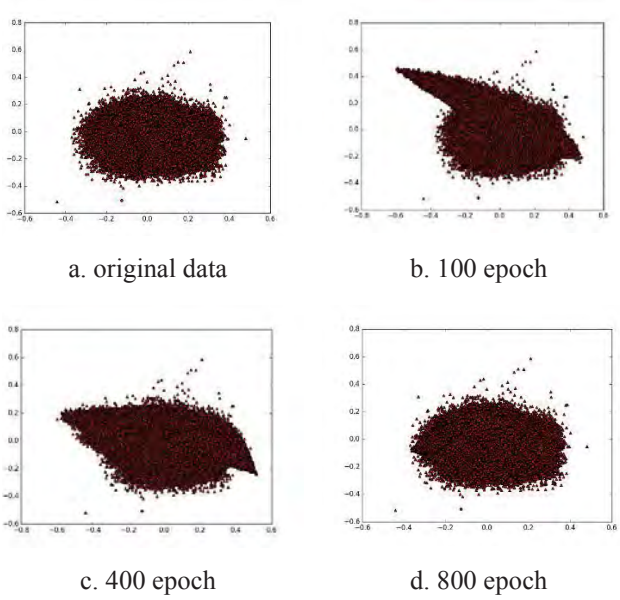
CGAN 이 사용자가 원하는 특성에 맞게 데이터를 생성해낼 수 있다는 점 때문에 본 논문에서는 각 클래스에 속한 데이터 수의 비율을 맞추기 위해 CGAN 을 활용한다. 제안하는 오버샘플링 기법은 그림 1 과 같다. 클래스 정보를 반영하여 CGAN 을 학습시키고 학습된 CGAN 을 통해 소수 클래스 특성을 가진 가상의 데이터를 생성한다. 다수 클래스의 데이터 수와 소수 클래스의 데이터 수가 같아질 때까지 반복 생성하며, 수가 같아졌다면 기계 학습 알고리즘을 학습시켜 분류를 수행한다.



(그림 1) CGAN 을 활용한 샘플링 기법 구성도

3. 실험 및 결과

실험에 사용된 데이터 집합은 신용카드 사기 데이터[10]이다. 이 데이터는 거래 시간, 금액 외에 보안상 처리된 28 개의 변수로 구성되어있다. 분류하고자 하는 클래스는 신용카드 사기 여부로, 284807 건 중 492 건만 사기를 당해 전체 데이터의 약 0.17%가 소수 클래스, 반대로 약 99.83%가 다수 클래스에 속한다. 실험에 사용된 CGAN 은 Generator 와 Discriminator 모두 2 층으로 구성되었고 Activation Function 은 ReLU, Optimizer 는 Adam[11]을 사용하였다. 비교한 오버샘플링 기법에는 ROS 와 SMOTE 가 사용되었고, 샘플링된 데이터를 대상으로 분류를 수행한 분류기는 Support Vector Machine(SVM)[12], Random Forest(RF)[13], Multi-Layer Perceptron(MLP)[14] 세 종류로 실험을 진행하였다. 실험은 python 3.5 에서 수행되었고, 라이브러리는 sklearn 0.19.1 을 사용하였다. SVM 모델은 C=1.0 으로 설정하고 RBF kernel 을 사용했으며 RF 모델은 트리 100 개로 구성하였다. MLP 모델은 은닉층의 각 층이 25 개, 10 개, 5 개 노드를 갖도록 구성되었으며, Activation Function 과 Optimizer 는 각각 ReLU, Adam 으로 CGAN 과 동일하게 사용하였다.



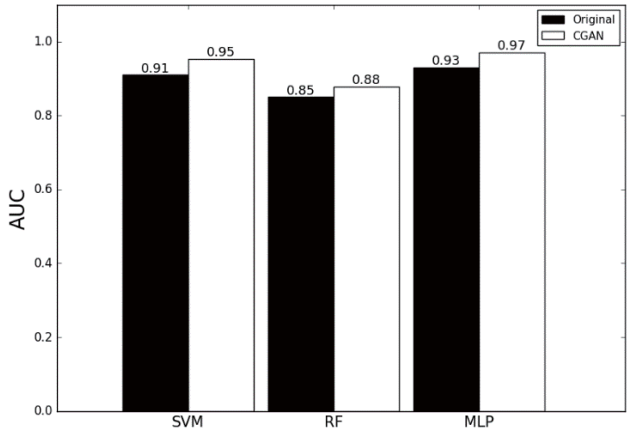
a. original data b. 100 epoch

c. 400 epoch d. 800 epoch

(그림 2) Epoch 에 따른 PCA1-PCA2 그래프

CGAN 이 제대로 가상 데이터를 생성하는지 시각적으로 확인하기 위해 실제 데이터에서 Principal Component Analysis(PCA)로 주성분을 추출하고 이 축을 이용하여 실제 데이터와 CGAN 으로 생성된 데이터를 그려보았다. 그림 2 는 이 실험의 결과로, CGAN 의 epoch 에 따른 데이터를 그린 그래프이다. 그래프의 x 축은 첫 번째 주성분, y 축은 두 번째 주성분의 값이다. 그림에서 a 는 오버샘플링을 하지 않은 원본 데이터의 분포를 나타내며, b, c, d 는 각각 CGAN 모델이 100 회, 400 회, 800 회 학습했을 때 생성된 데이터의 분포를 나타낸 것이다. 그림에서 같이 100 회, 400 회에서는 CGAN 이 실제 데이터 분포에서 벗어난 데이터를 생성하는 경우가 존재했지만 800 회 이상 학습했을 때는 원본 데이터의 분포 안에서 잘 만들어짐을 확인할 수 있었다. 따라서 CGAN 모델의 epoch 를 1000 으로 설정하여 분류 정확도 측정 실험을 진행하였다.

모델 정확도의 척도로는 클래스 불균형 데이터에 가장 많이 사용되는 Area Under the ROC Curve (AUC)를 사용하였다[5]. 그림 3 은 샘플링 전의 데이터로 기계 학습 알고리즘을 학습시켰을 때와 CGAN 을 활용하여 오버샘플링한 후에 기계 학습 알고리즘을 학습시켰을 때 분류 정확도를 그린 것이다. 불균형 데이터의 분류에 가장 많이 이용되는 SVM 의 경우 0.91 에서 0.95 로 0.04 만큼 향상되었고, MLP 의 경우에도 0.93 에서 0.97 로 0.04 만큼 상승한 것을 확인하였다. 기존의 예측 정확도가 제일 낮았던 RF 경우에도 0.88 로 상승하여 평균적으로 AUC 가 0.03 이상 상승하였다.



(그림 3) CGAN 을 활용한 샘플링 기법의 성능 검증

표 1 은 기존에 많이 사용되는 오버샘플링 기법들과 본 논문에서 제안하는 CGAN 을 활용한 오버샘플링 기법의 효과를 비교한 표이다. RF 와 MLP 에서 CGAN 을 사용한 오버샘플링이 ROS 와 SMOTE 보다 AUC 가 0.01 이상 좋았다. 특히 딥러닝 기반 모델인 MLP 를 사용하여 분류하였을 때 0.971 로 가장 높은 AUC 를 기록하였다. 반면, SVM 에서는 SMOTE 가 CGAN 을 사용한 오버샘플링보다 0.001 만큼 더 좋았다. 하지만 전반적으로 보았을 때 CGAN 을 사용한 오버샘플링 기법이 ROS 나 SMOTE 보다 전반적으로 우수하다고 할 수 있다.

<표 1> 오버샘플링 수행 후 분류기별 성능 비교 실험

Model	Original	ROS	SMOTE	CGAN
SVM	0.91	0.950	**0.954**	0.953
RF	0.85	0.859	0.867	**0.878**
MLP	0.93	0.959	0.961	**0.971**

4. 결론 및 향후 연구

본 논문에서는 클래스 불균형 문제를 해결하기 위해 CGAN 을 활용하여 오버샘플링하는 기법을 제시하였고, 기존의 다른 오버샘플링 기법들과 비교 실험을 진행하여 제안하는 오버샘플링 기법의 우수함을 입증하였다. 향후 연구는 CGAN 을 활용한 오버샘플링 기법을 다른 데이터 집합에도 적용하여 오버샘플링으로서의 성능을 검증할 것이며, 하이퍼 파라미터 최적화, 특징 선택, 특징 추출 및 군집화 알고리즘 등을 적용하여 분류 모델의 성능을 보다 안정적으로 향상시키는 연구를 진행할 계획이다.

사사의 글

이 논문은 2018 년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원(No.R0190-16-2012, 빅데이터 처리 고도화 핵심기술개발 사업 총괄 및 고성능 컴퓨팅 기술을 활용한 성능 가속화 기술 개발)을 받아 수행된 연구임.

참고문헌

[1] O'Brien, Robert C. "A Random Forests Quantile Classifier for Class Imbalanced Data." (2018).
[2] Moro, Sérgio, Paulo Cortez, and Paulo Rita. "A data-driven approach to predict the success of bank telemarketing." Decision Support Systems 62 (2014): 22-31.
[3] Stolfo, Salvatore J., et al. "Cost-based modeling for fraud and intrusion detection: Results from the JAM project." COLUMBIA UNIV NEW YORK DEPT OF COMPUTER SCIENCE (2000).
[4] Kumar, M., and H. Sheshadri. "On the classification of imbalanced datasets." International Journal of Computer Applications 44 (2012).
[5] Haixiang, Guo, et al. "Learning from class-imbalanced data: Review of methods and applications." Expert Systems with Applications 73 (2017): 220-239.
[6] Van Hulse, Jason, Taghi M. Khoshgoftaar, and Amri Napolitano. "Experimental perspectives on learning from imbalanced data." Proceedings of the 24th international conference on Machine learning. ACM (2007).
[7] Chawla, Nitesh V., et al. "SMOTE: synthetic minority over-sampling technique." Journal of artificial intelligence research 16 (2002): 321-357.
[8] Mirza, Mehdi, and Simon Osindero. "Conditional generative adversarial nets." arXiv preprint arXiv:1411.1784 (2014).
[9] Goodfellow, Ian, et al. "Generative adversarial nets." Advances in neural information processing systems (2014).
[10] Dal Pozzolo, Andrea, and Gianluca Bontempi. "Adaptive machine learning for credit card fraud detection." (2015).
[11] Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).
[12] Hearst, Marti A., et al. "Support vector machines." IEEE Intelligent Systems and their applications 13.4 (1998): 18-28.
[13] Liaw, Andy, and Matthew Wiener. "Classification and regression by randomForest." R news 2.3 (2002): 18-22.
[14] Haykin, Simon S., et al. Neural networks and learning machines. Vol. 3. Upper Saddle River, NJ, USA:: Pearson (2009).