**Classification by Coreset selection**

GSDS 석사과정 20224314

강현구

1. **Abstract**

Semi-supervised learning(이하 SSL)은 Labeling Cost 문제를 해결할 수 있는 주요 방안 중 하나이다. 하지만 모델을 학습하기에도 Label Data가 부족할 경우 Confirmation bias가 발생할 수 있다. 이를 방지하기 위해 기존의 연구들은 클래스별 Prior 확률을 활용하나, 이는 비현실적이다. 따라서 Dataset에 대한 사전 지식 없이 Label Data 문제를 해결할 필요가 있다.

본 연구에서는 Active learning의 coreset Selection을 활용한 새로운 Classification 방법을 제안한다. Coreset Selection 간 형성한 subgraph들은 기하적 관점에서 모델의 성능에 의존하지 않고 높은 신뢰도의 Classification을 가능하게 한다. 즉, 학습과정을 필요로 하지 않는 Classification을 통해 Pseudo labelling 함으로써 Label Data 부족 문제를 해결한다. 본 방법은 클래스의 수가 적고 데이터 분포가 밀접해 있는 데이터셋에 대해선 높은 효율 및 정확도를 갖춘다. 고차원의 데이터가 저차원의 Feature로 잘 반영되며, Distance base method가 적용될 수 있는 모든 분야에 적용할 수 있다.

**1. Introduction**

최근Deep learning(이하 DL)모델은 많은 양의 Labeled Data를 기반으로 다양한 분야에서 성과를 내고 있다. 하지만 모델이 필요로 하는 데이터가 증가함에 따라, Labeling Cost를 어떻게 해결할 것인가 중요한 화두가 되었다. Knowledge transfer은 Labeling이 된 Dataset과 그렇지 않은 Target 데이터셋으로 나뉜 시나리오를 가정한다(Argote & Ingram, 2000). Labeling dataset으로부터 Transferable한 representation을 학습하고, Learning pipeline을 통해 전달함으로써 target dataset의 Label data 부족 문제를 해결한다. Semi-supervised learning(이하 SSL)은 부족한 Label data와 매우 많은 unlabeled data 시나리오를 가정한다. 본 연구에서는 SSL을 통한 이미지 classification에 중점을 둔다.

SSL은 크게 Consistency regularization 방법과 Pseudo Labeling 으로 나뉜다(Yang, Song, King, & Xu, 2022). Consistency regularization 방법은 모델의 Input에 변형이 가해져도 모델의 결과는 일정하게 유지돼야 함을 가정한다. Label Data에 변형을 가한 데이터를 생산(augmentation) 함으로써 동일한 Class의 Label Data를 생성한다. Pseudo labeling은 학습된 모델을 통해 Unlabeled Data에 대해 예측하여 psuedu label을 부여한다. 이후 pseudo labeling 된 sample을 모델 학습에 활용한다. 두 방법은 SSL의 고질적인 문제인 Confirmation bias를 해소하기 위해 Unlabeled data를 활용하는 방법이 다르다. Confirmation bias이란 모델을 학습하는 과정에서 unlabeled sample에 대한 잘못된 예측값이 모델의 성능악화를 가중시키는 것을 의미한다. Confirmation bias를 해결하기 위해 warm-up phase using labeled data, uncertainty weighting, adversarial attacks, graph-consistency 방법 등을 활용한다.

본 paper에서는 pseudo labeling 방법에 초점을 잡는다. 최근 이미지 분류 연구에 있어 Consistency regularization의 높은 성능으로 인해 선호 받는다. 단, Consistency regularization과 Pseudo labeling을 통해 얻을 수 있는 dataset의 diversity는 성격이 달라 둘 다 필요하다. Consistency regularization은 label data를 다양한 방법으로 augmentation 하나, unlabeled 데이터를 직접적으로 활용하는데 제약이 있다. 반면 Pseudo labeling의 경우 Confirmation bias가 있으나 unlabeled data의 정보를 직접적으로 활용할 수 있다. 즉, Pseudo labeling은 Consistency regularization과 상반된 방법이 아닌 병행 가능한 방법이다. 각각의 방법에 대한 연구는 SSL 모델 전체의 성능 향상을 위해 필요하다. 또한 Confirmation bias를 잘 해소하여 Consistency regularization 보다 높은 성능을 낸 연구(Arazo, Ortego, Albert, O’Connor, & McGuinness, 2020)로 보아 Pseudo labeling의 잠재력을 엿볼 수 있다.

Pseudo labeling에 있어 가장 큰 난관은 confirmation bias를 잘 해소하는 것이다. 이때 핵심은 ‘어떻게 Pseudo labeling 하는가’ 이다. 기존의 연구들은 CNN 모델 예측 값에 대해 신뢰도에 대한 Threshold 설정(Cascante-Bonilla, Tan, Qi, & Ordonez, 2021), Soft labeling 부여 및 또는 초기 수렴을 위한 규제항을 도입한다.(Zhang, Cisse, Dauphin, & Lopez-Paz, 2017) 대부분의 조치들은 CNN 모델의 성능에 의존하고 있다. 단, CNN모델을 학습하기에도 Label data가 부족할 경우 모델의 성과를 신뢰하기 어렵다. 또한 Label data가 부족하여 dataset의 diversity를 반영하지 못할 수 있다. 모델이 학습하기에도 부족한 Label data는 confirmation bias를 방지하기 위한 조치들의 효과 또한 악화시킨다. 기존 연구에선 이를 방지하고자 데이터셋이 class-balance dataset일 것임을 가정함으로써 클래스별 사전 비율을 규제항으로 활용한다(Arazo et al., 2020). 하지만 실제 현장에선 Dataset이 Class-balance 할 것, 또는 클래스에 대한 사전 지식을 알 것이라 기대하긴 어렵다. 따라서 Label 데이터가 부족한 경우에 데이터셋의 사전 지식에 의존하지 않으면서 높은 성능을 내는 Classification 방법이 필요하다.

본 연구에선Active learning의 일환인 Coreset-selection 방법을 통해 모델 성능에 의존하지 않는 pseudo labeling 방법을 제안한다. Active learning은 모델을 학습함에 있어 유용한 Data를 Sampling 하는 방법이다. 어떤 Acquisition Strategy을 활용하냐에 따라 Dataset의 특정 정보가 반영된 Sample을 선정할 수 있다. Coreset selection은 Dataset의 Diversity를 보장하는 Data를 Sampling 한다. 또한 Sampling 간 Data들의 거리 정보를 활용하여 Neural Network 기반 모델에 의존하지 않는다. 즉, Coreset selection은 Dataset의 diversity 반영하며 Label data의 수에 영향을 받지 않는다. 더 나아가 본 연구에서 각 Sampling point들은 기하적 관점에서 Unlabeled Data에 대해 대표성을 띄고 있음을 보일 것이다. 또한 기하적 관계를 토대로 모델의 성능에 의존하지 않고 높은 신뢰도의 Classification이 가능함을 보일 것이다. 이때 active learning이 고차원 데이터에 대해 적용이 어렵다는 점, 그리고 기하적 관계를 단순화를 위해 unsupervised dimension reduction 방법을 적용하여 저차원으로 Mapping 할 것이다. 본 연구에서는 이미지 데이터의 구조 정보까지 반영할 수 있도록 Convolution autoencoder을 적용한다. 이로써 Coreset selection을 통한 Dataset의 diversity 반영, 모델에 의존하지 않는 높은 정확도의 classification을 통한 CNN 모델의 초기 학습 Confirmation bias 발생을 방지할 것이다.

본 연구는 Lable data가 부족한 상황에서도 높은 성능을 내는 Classification 방법을 제안한다는 측면에서 Contribution이 있다. 또한 Classification의 결과를 CNN 모델 학습에 적용함으로써 Label Data 부족 문제를 해소할 수 있다. Subgraph를 활용한 기하적 관점에서의 Classification은 클래스의 수가 적고 데이터 분포가 밀접해 있는 데이터셋에 대해선 높은 효율 및 정확도를 갖출 것으로 기대한다.

**2. Related works**

2-1). Semi supervised learning

SSL은 Labeling Cost 문제로 일부의 Label Data만을 구할 수 있는 현장에서 널리 활용된다. SSL은 크게 3가지 가정을 기반으로 한다(Chapelle, Scholkopf, & Zien, 2009). 첫째, 확률 밀도가 높은 지역의 입력값 x1, x2가 가깝다면 각각의 연관된 Label y1, y2도 가깝다. 둘, 모델의 결정 경계가 데이터의 확률 밀도가 높은 곳을 지나지 않는다. 마지막으로 고차원의 입력 데이터가 저차원의 Manifold를 따라 놓여 있다. 즉, SSL은 Data 간의 거리가 Data 특성의 유사성을 반영하며, 각 Class 별로 밀도를 통해 경계 지점을 파악할 수 있을 때 활용할 수 있다. 그 결과 Distance-based 방법들이 적용될 수 있는 다양한 영역에 폭 넓게 활용되고 있다.

SSL의 주요 방법 중 하나는 Consistency regularization 방법이다. 모델의 Input에 변형이 가해져도 모델의 결과는 일정하게 유지돼야 함을 가정한다. Data augmentation을 통한 Label data을 늘려 Confirmation bias를 효과적으로 제거할 수 있다. 구체적인 방법으로는 Randomize data augmentation(Sajjadi, Javanmardi, & Tasdizen, 2016), Co-training(Qiao, Shen, Zhang, Wang, & Yuille, 2018), contrastive loss(Luo, Zhu, Li, Ren, & Zhang, 2018) 등이 있다. 또 다른 SSL의 방식은 Pseudo labeling로, 본 연구에서 주로 다룬다. Pseudo labeling은 Unlabeled Data에 대한 예측을 leverage함으로써 모델의 성능을 향상시킨다. 초기에는 confirmation bias를 해결하지 못해 pseudo labeling을 모델의 fine tuning에 한정했다(Lee, 2013). 이후 confirmation bias를 해결하기 위해 uncertainty weight를 부여(Shi, Gong, Ding, Tao, & Zheng, 2018), Soft labeling, Mix-up data augmentation 적용(Arazo et al., 2020) 등 다양한 조치를 취해왔다.

단 각각의 조치들은 학습을 필요로 하는 Neural Network based model에 의존한다는 한계가 있다. 모델을 학습하기에도 Label data가 부족하여 모델의 성과를 신뢰할 수 없을 때가 많다. Label data가 dataset의 diversity를 반영하지 못할 수 있다. 모델이 학습하기에도 부족한 Label data는 confirmation bias를 방지하기 위한 조치들의 효과 또한 악화시킨다. 즉, labeled 데이터가 부족한 상황 속에서 uncertainty weight 및 soft labeling confirmation bias 해소는 한계가 있다. 한편 모델에 의존하지 않고도 confirmation bias를 해소할 수 있는 방안이 있다. Arazo의 연구에선 Soft labeling 외에도 Mix-up data augmentation & minimum batch, Dataset의 Class 사전 비율정보를 활용하여 Confirmation bias를 효과적으로 방지하였다 (Arazo et al., 2020). 해당 연구는CIFAR 10/100, SHVN 데이터셋에 대해 Consistency regularization 보다 높은 성능을 냈다. 단, Arazo의 연구에서도 한계가 있다. 해당 연구에선Class-balance scenario를 가정함으로써 데이터셋의 클래스 비율정보를 규제항으로 적용한다. 즉, Dataset의 클래스 비율에 대한 사전 지식을 얻을 수 없을 경우 초기 confirmation bias를 피할 수 없다는 한계가 있다.

2-2) Active learning

Active learning(Tong, 2001)(이하 AL) 또한 Labeling cost를 줄일 수 있는 방법이다. AL은 성능은 유지하면서 가능한 한 Labeling cost를 줄이기 위해 unlabeled dataset에서부터 가장 유용한 sample을 추출한다. AL의 유형은 크게 3가지가 있다(Ren et al., 2021). Membership query Synthesis은 Labeling을 요청할 데이터를 생성한다. Stream-based Selective sampling은 새로운 sample Data에 대해서 레이블이 필요한지를 판단한다. 마지막으로 Pool-based sampling은 주어진 Dataset에서 중요한 데이터를 선별한다. 본 연구에서는 Pool-based sampling에 초점을 둔다.

Pool-based sampling AL에서 Acquisition strategy에 따라 Unlabel Data를 선별한다. 크게 Uncertainty-based, Expected-based and Diversity-based approach로 나눠 볼 수 있다(Ren et al., 2021). Uncertainty-based approach는 대부분 neural network Model에 의거하여 각 Data별 Uncertainty를 측정한다(Gal, Islam, & Ghahramani, 2017; K. Wang, Zhang, Li, Zhang, & Lin, 2016). 이후 Uncertainty score가 높은 순으로 데이터를 Sampling 하여 Labeling을 한다. Uncertainty 가 높은 데이터들이 각 Class 별 경계 값 주변에 놓여 있는 것을 고려할 때, Uncertainty가 높은 데이터들은 서로 비슷한 상황에 놓여 있는 경우가 많다. 따라서 각 Class 별로 경계 영역을 학습하는 데에는 도움이 되나 Dataset의 Diversity를 반영하지 못한다는 한계점을 가진다. Expected-based approach는 모델 성능을 가장 향상시킬 것으로 기대되는 Data들을 선별한다(Sener & Savarese, 2017). 단, Uncertainty based approach와 동일하게 Acquisition strategy가 학습을 필요로 하는 neural network 기반 모델에 의거할 경우 Label data가 부족할 때엔 적용하기 어렵다. 마지막으로 Diversity-based approach는 데이터셋의 Diversity를 보장할 수 있는 데이터를 선별하는 방법이다. 대표적으론 Coreset selection이 있다(Sener & Savarese, 2017). Coreset은 Expected model의 접근 방식을 취하나 모든 data들을 덮는 최소 반지름을 가진 subgraph들을 뽑는 것과 결과가 동일함을 보인다. 즉, Coreset selection은 모델에 의존없이 sampling 가능한 AL로 본 연구에서 활용할 것이다.

AL, DL의 각각의 장점을 합쳐 시너지를 내고자 한 연구들이 존재한다. DL은 고차원 데이터에 쉽게 적용할 수 있으며 데이터의 사람의 개입없이 feature engineering 할 수 있다. 하지만 모델 학습을 위해 유용한 Label data를 필요로 한다. 반면 AL은 유용한 Data를 우선적으로 sampling 하여 DL의 학습에 도움을 줄 수 있다. 하지만 고차원 데이터에는 적용하기 어렵다는 한계가 있으며, 고차원 데이터로부터 저차원의 feature을 산출할 능력이 없다(Tong, 2001). 이에 두 가지를 합쳐 이미지 Classification 간 Labeling cost를 문제를 해결한 연구들이 있다. DBL의 경우 Autoencoder로 고차원의 이미지 데이터를 저차원의 feature을 도출한 후, 각각의 data에 대한 representation과 uncertainty을 측정하여 sampling 한다(Liu, Zhang, & Eom, 2016). CEAL의 경우 CNN의 모델의 신뢰도가 낮은 unlabeled data에 대해 active learning을 하며, 신뢰도가 일정 이상일 경우에만 pseudo labeling 한다(K. Wang et al., 2016). BCNN의 경우 Bayesian Network을 활용하여 uncertainty을 계산하고 이를 acquisition strategy로 활용한다(Gal et al., 2017). 하지만 위의 연구들 또한 Labeled data가 적은 경우 Model의 성능을 신뢰할 수 없게 된다. Model 자체의 판단을 믿을 수 없는 상태에서 모델에 의거한 acquisition strategy는 신뢰할 수 없다.

반면 unsupervised Deep learning 방법을 통해 고차원의 데이터의 feature을 extraction 한 다음 AL을 적용한 연구들이 있다. 주로 고차원의 이미지 데이터를 Auto encoder을 통해서 저차원의 feature을 extraction 한다(Liu et al., 2016; Sun, Li, Wang, Plaza, & Chen, 2016). VAAL은 Variational auto encoder을 통해서 Latent space에서 충분히 다른 unlabeled data을 우선적으로 sampling 한다(Sinha, Ebrahimi, & Darrell, 2019). 해당 방법은 unsupervised learning으로 Label data가 부족한 경우에도 본래의 성능을 낼 수 있다. 따라서 본 연구에서도 고차원의 Image Data을 적용하기 위해 Unsupervised learning을 통한 feature extraction을 적용하겠다.

**3. Method**

**3.1) Model structure**

Notation은 Arazo의 연구를 참고한다(Arazo et al., 2020). 총 N개의 Dataset D에 대해 unlabeled set 과 labeled set s.t. 으로 표현하겠다 이때 은 에 대응하는 C Class에 대한 one-hot encoding label이다. 은 CNN 모델을, 은 모델의 파라미터를 의미한다. 본 연구에서 제시하는 2가지 classification 방법은 각각 편의상 CS(Coreset Selection)1, CS2라고 부르겠다. 첫번째 방법인 CS1은 Coreset selection간 subgraph 간의 기하적 특성을 활용하여 조건을 달성하는 특정 Data에 한해 Hard label 형태의 Classification을 한다. 두번째 방법인 CS2는 Coreset selection을 다회차 iteration 할 때, 각 데이터 별 subgraph의 중첩 정도에 따라 클래스별 확률 벡터 를 계산한다. 이에 주어진 label sample과 CS1 방법으로 labeling 한 은 Hard label로 고려하며, 그 외의 의 unlabel data는 CS2 방법을 통해 soft labeling을 적용할 것이다(s.t. ). 구체적인 방법은 뒤에서 구체적으로 서술하겠다. 모든 개의 unlabeled data는 모두 pseudo labeling이 가능하다고 가정한다. labeled sample와 라고 할 때, 우린 전체 Dataset D를  로 재정의할 수 있다.

CNN의 파라미터 는 categorical cross-entropy를 통해 최적화한다. 이때 Loss 함수는 다음과 같다.

(3.1)

이때 은 초기에는 로 적용하며, CNN 모델의 성능이 CS2의 방법보다 좋아졌다 판단될 때 을 적용할 것이다. 먼저 Label data을 활용한 1차 Warm training을 진행한다. 이후 CS1 방법을 통해 hard labeling 한 데이터 및 CS2 방법을 통해 soft labeling 한 데이터를 고려하여 2차 Warm training을 거친다. 이후 을 로 대체하겠다. 는 element-wise 하게 적용된다.

우리는 각 single class에 대한 확률 분포를 집중시키기 위해 규제항을 도입한다.

은 의 C class에 대한 softmax output을 의미한다. 이 규제항은 average per-sample entropy로 SSL에서는 잘 알려져 있는 규제항이다(Grandvalet & Bengio, 2004). 이로써 SSL Loss는 다음과 같이 표현할 수 있다.

(2)

이때 은 label data에 대한 Loss를, 은 CS1으로 pseudo label한 개의 데이터에 대한 Loss를, 마지막으로 은 개의 나머지 데이터에 대한 Loss를 의미한다. 각 들은 각 항목에 대한 영향력을 조절한다. [아직 $R\_H 적용 못함]

Confirmation bias를 방지하기 위해 Mix-up data augmentation과 setting a minimum number of labeled samples per mini-batch를 적용한다. Mix-up data augmentation은 data augmentation과 label smoothing을 결합한 강력한 규제 방법이다(Zhang et al., 2017). Mix-up data augmentation은 setting a minimum number of labeled samples for mini-batch k를 적용할 때 Confirmation bias가 효과적으로 예방된다(Arazo et al., 2020). 이때 높은 신뢰도를 가진 CS1 방법을 통해 구한 개의 sample에 대해서 k보다 작은 k’를 적용하겠다. 이로써 label data가 적어 의 경우에 대해서 pseudo labeled data가 labeled data보다 더욱 영향을 미치는 경우를 방지하겠다.

* 본 연구에서 적용할 수 없다면 제거하기. 추후 연구로 돌리기.

**3-2) Coreset selection**

Coreset Selection은 기대되는 모델 성능을 최대화하는 Data를 Sampling 한다. 이는 주어진 Sampling size b개로 전체 데이터를 덮을 수 있는 subgraph들을 구성할 때, 최소한의 반지름 를 가지게 만드는 데이터를 Sampling 하는 것과 동일하다(Sener & Savarese, 2017). 각 Sampling Data들은 Dataset의 밀집 정도에 무관하게 넓게 퍼져 있다. 이로 인해 Sampling point 는 특정 클래스의 값 만을 포함하는 것이 아닌 전반적인 데이터 셋을 반영한다. 또한 Data 간의 거리를 기반으로 Sampling 하여 학습을 필요로 하는 모델에 의존하지 않는다. 따라서 Label Data가 부족한 상황에서도 동일한 Sampling 성능을 낼 수 있다.

단, AL은 이미지, Text 등의 고차원 데이터로 확장하는데 어려움을 겪는다(Tong, 2001). 이로 인해 대부분의 AL에 대한 연구들은 저차원의 문제에 집중하거나, 사전에 추출한 고차원의 데이터의 feature을 활용한다. 이때 후자의 경우 AL은 데이터의 feature을 추출하는 능력이 없어 추가적인 방안을 적용할 필요가 있다. 이에 고차원의 데이터를 가공하며 사람의 개입없이 feature extraction 할 수 있는 능력을 갖춘 Deep learning과 AL을 함께 사용함으로써 시너지를 향상시킬 수 있다(Ren et al., 2021).

**3-3) Dimension reduction**

고차원 이미지 데이터에 대해 Coreset을 적용하기 위해선 feature extraction을 통한 dimension reduction이 필요하다. 또한 현 연구에서 제안하는 classification의 방법은 subgraph 간의 접점 유무를 활용한다는 점에서 2차원, 또는 3차원까지의 차원 축소를 필요로 한다. Classification의 자세한 과정은 뒤에 서술하며, 지금은 데이터 차원에 따른 Classification의 조건만 한정적으로 다루겠다. 정확도와 CS1 방법을 적용할 수 있는 Data의 개수는 Trade-off 관계를 띈다. 그리고 데이터의 차원이 이를 조절하는 요인이다. 접하고 있는 subgraph의 수가 많을수록 classification의 제한조건이 강화되어 정밀도가 올라간다. 반면 조건이 강화됨에 따라 Classification이 가능한 unlabeled data의 수가 줄어든다. 접하는 subgraph의 수에 대해 적절한 guideline이 필요하다. 한편 subgraph의 차원이 증가할수록 접하는 subgraph의 수가 기하급수적으로 증가한다. 2차원의 경우 각 subgraph 마다 접할 수 있는 subgraph의 수는 기껏해야 7~8개이다. 하지만 3차원의 경우 20~30개로 증가하며 이는 차원이 증가할수록 기하급수적으로 증가할 것이다. 이는 Adjacency 유무를 각 subgraph의 중점 간의 거리로 판단하기 때문이다. 작은 이미지 데이터 셋인 MNIST의 이미지조차 784 차원으로 표현된다. 따라서 정확도 및 classification 가능 데이터 간의 관계를 유념하며 차원 축소를 할 필요가 있다. 본 연구에서는 간단한 이미지 데이터셋을 활용하여 2차원으로 줄여도 높은 정확도의 classification이 가능했다. 이에 본 연구에서는 2차원으로 한정한다.

본 연구에서는 차원 축소를 위해 Convolution autoencoder(이하 CAE)을 활용한다. 본 연구에서는 dimension reduction 방법을 통해 축소했을 때 각 클래스별 정보를 잘 보존하여 클러스트링 될 것이라 가정한다. Auto encoder는 Dimension reduction의 역할을 뿐만 아니라 반복되는 구조를 포착할 수 있다(Y. Wang, Yao, & Zhao, 2016). 또한 Unsupervised dimension reduction 방법들을 활용하여 k-nearest neighbor 방식으로 classification을 하였을 때 CAE가 85% 가량의 정확도로 SOTA 방식과 유사한 결과를 가졌다(Hurtik, Molek, & Perfilieva, 2020). 이로 보아 CAE를 통해 dimension reduction을 하였을 때 유사한 클래스의 데이터들을 clustering 하는 효과가 높다고 유추할 수 있다. 이는 CAE가 각 feature 뿐만 아니라 구조적인 정보까지 담고 있기 때문으로 판단된다. 추후 CNN 모델의 Fine tuning에도 활용하겠다.

**3-4) Coreset selection based classification (CS1 & CS2)**

Confirmation bias를 막기 위해선 CNN 모델에 대한 과신뢰를 줄이거나, 새로운 높은 성능의 Classification 방법을 고안할 필요가 있다. 먼저 전자의 방법으로서 mix-up augmentation을 적용했다. 본 연구에서는 후자의 관점에서 Coreset selection 간 형성된 subgraph를 활용하여 신뢰도 높은 Classification을 제안한다. 전체 과정은 Figure을 참고하기 바란다. 먼저 주어진 dataset을 CAE을 활용하여 2차원의 형태로 feature extraction 시킨다. 이후 2차원 feature 간의 distance를 기반으로 Coreset selection을 진행한다. 개의 Labeled data들은 Coreset selection을 통해 sampling한 결과라고 한다. 총 p번의 Coreset selection을 한다 했을 때, 를 p번째 Coreset selection을 통해 Sampling 한 data set이라고 하며, 을 각 Sampling point 라고 하자. 이때 을 성립한다. 각 을 중점으로 삼는 subgraph 의 반지름을 라고 하자. Coreset selection은 K-centers algorithm과 동일하여 subgraph 의 반지름 을 최소화시키는 unlabeled data 를 로 Sampling한다. 이로써 는 을 만족한다. 각 subgraph의 밀도는 해당 subgraph에 포함된 Data 의 개수로 측정할 수 있다.

CS1 방법은 subgraph 간의 기하적 관계와 밀도를 통해 Classification 한다. 편의 상 이번 coreset 간 표기를 로 단순화하겠다. 확률 밀도가 높은 지역의 입력값 x1, x2가 가깝다면 각각의 연관된 Label y1, y2도 가깝다는 SSL의 가정을 통해, 밀도가 높은 subgraph 에 속한 Data s.t. j ∈ {1,…, | }들은 높은 확률로 중점인 와 클래스가 동일할 것이다. 즉, 은 subgraph에 속한 unlabel Data를 대표할 수 있다. 더 나아가 Active selection을 통해 형성한 subgraph들이 Dataset을 촘촘하게 덮을 수 있을 정도로 충분히 작은 반지름 을 가졌다고 가정한다. s.t. k ∈ {1,…, | } 를 와 접하고 있는 subgraph라고 하자. 각 subgraph의 중점 , 들의 클래스는 알고 있다. 이때 기하적 관점에서 의 Class가 다를 경우 와 가 서로 다른 클래스들의 경계 지역에 있다고 유추해볼 수 있다. 반대로 모든 와 의 클래스가 동일할 때 특정 클래스의 중심부에 위치할 것이다. 즉, Figure1과 같이 subgraph 간의 기하적 관계와 각 중점의 클래스를 통해서 각 subgraph들이 클래스별 데이터 분포간 어디에 위치해 있는지 유추할 수 있다. 즉, 이며, 을 성립하는 subgraph 에 속한 unlabeled Data 에 대해 와 동일한 Class로 Classification을 할 수 있다. 는 hyperparameter이다. 이때 추가 조건으로 주변부에 떨어진 2개의 subgraph가 우연에 의해 겹쳐 잘못 Classification 하는 경우를 방지하기 위해 접하는 subgraph의 개수 또한 하이퍼 파라미터 이상 일 것을 제한 조건으로 추가한다.

[Figure 1. Subgraph 간 기하적 관계]

본 Classification 방법은 클래스의 종류가 적고, 특정 클래스가 Clustering 될 수록 성능이 향상된다. 충분히 작은 반지름 , 클래스별 데이터 밀집이 보장된다면 각 클래스별 중심부에 해당하는 데이터들을 높은 신뢰도로 Classification을 할 수 있을 것으로 유추된다. 단, 추후 실험 결과로 밝히겠지만 반지름 의 크기와 subgraph의 밀도는 Classification 정확도에 영향을 미치지 않는다. 오히려 밀도 조건에 추가함에 따라 Classification이 가능한 Data의 수를 줄이는 악영향만을 미친다. 은 크기에 상관없이 항상 높은 성능의 classification이 가능했다. 마지막으로 sampling을 보다 활용할 수 있도록, 다음 회차의 Coreset selection과 Labeling 못한 개의 Unlabeled data에 한정하여 진행한다.

Coreset selection을 다회차 진행하여 subgraph들이 중첩되었을 때 Cs1으로 pseudo labeling 되지 않은 에 대해 class 별 probability를 계산할 수 있다. 1번의 Coreset selection을 통해 모든 들은 1개 이상의 subgraph에 포함된다. 따라서 Coreset selection이 총 P번 진행된다면 모든 들은 최소 P개의 subgraph안에 속하게 된다. 은 특정 Class의 중점을 가진 subgraph에 많이 속할수록 동일한 클래스일 확률이 높을 것이다. 또한 각 subgraph들의 반지름 에 반비례하여 확률이 높을 것이다. 각 클래스 별로 subgraph에 속하는 횟수 와, 반지름을 기반으로 softmax를 통해 확률화한다. 또한 반지름의 크기가 매우 커 각각의 확률값이 낮게 나오는 것을 방지하고자 p 회차 중 가장 작은 반지름 대비 각 의 길이를 활용한다. 또한 최근의 Coreset selection에 비중을 주기 위해 i번째 Iteration에 대해 의 비중을 추가하겠다.

(3.1)

이때 데이터의 클래스를 가장 높은 확률 값으로 고정할 경우 Confirmation bias가 발생할 수 있다. 따라서 soft labeling을 통해 예측에 대한 과신뢰를 방지한다. 또한 충분한 횟수의 subgraph의 중첩이 생기기 전엔 틀린 pseudo label을 부여할 확률이 높다. 따라서 접하는 subgraph의 수 가  일 것을 제한 조건으로 둔다.

[Figure 2. P 회차 Coreset selection을 통해 에 중첩된 subgraph]

마지막으로 주어진 Sampling size 에 대해서 어떻게 개별 Coreset sampling 크기와 Iteration의 균형을 잡을지 논의하겠다. I 번째의 Coreset selection의 크기 가 클수록 CS1 방법의 정확도와 Classification이 가능한 데이터의 크기가 향상된다. 반면 P의 크기가 늘어날수록 한 데이터에 대해 중첩하는 subgraph의 개수가 늘어나 CS2 방법의 정확도가 올라간다. CS1의 방법과 SC2의 방법은 개별적으로 적용할 수 있으며, 또는 CS1을 적용한 후 남은 unlabeled dataset에 한정하여 CS2을 적용할 수 있다. CS1 또한 한번에 개의 데이터를 Coreset selection 하는 것보다 = … = 처럼 p번에 나눠 Coreset selection을 할 때와 결과가 달라질 수 있다. 이에 주어진 에 대해 P번의 Coreset selection으로 나누면 좋을지 경험적으로 검증하도록 하겠다.

4. 연구 환경

4-1) Dataset and training

우린 총 4개의 이미지 classification dataset으로 MNIST, CIFAR10/100, 그리고 SVHN을 우리의 접근 방식을 확인하고자 사용할 것이다. 각각의 데이터셋은 CIFAR 100을 제외하곤 모두 10개의 클래스를 가진다. CIFAR 100은 100개의 클래스를 가진다. MNIST dataset은 총 60K의 Training Image와 10K의 Test Image로 이뤄져 있다. 모두 흑백 이미지이며 해상도는 28x28이다. CIFAR10/100, SVHN은 각각 50K,50K, 73257개의 Training Image와 10K, 10K, 26032개의 Test 데이터로 이뤄져 있다. 모두 컬러 이미지이며 해상도는 32x32이다. 클래스의 개수가 10개인 MNIST, CIFAR10, SVHN의 경우 = 10, 50, 100, 250, 1K을 적용하겠다. CIFAR 100은 순으로 적용하겠다. 각각의 데이터셋에 대해 CNN 모델은 잘 알려진 13-CNN모델 구조를 적용하겠다. 각 데이터셋의 해상도에 맞춰 Convolution autoencoder을 적용한다. Encoder 부분은 13-CNN 모델을 차용했으나, Decoder는 각 Encoder의 CNN 과정과 맞췄다.

각 데이터셋 별로 동일한 하이퍼 파라미터를 적용했다. 각 데이터셋 별로 Convolution autoencoder의 경우 총 100회의 epoch를 적용하였으며, learning rate는 최초에는 0.001를, optimizer로는 Adam을 적용하였다. 이후 각 CNN 모델을 학습함에 있어 label dataset과 CS1, CS2로 labeling 한 dataset을 따로 학습하였다. {Arazo, 2020 #9}의 연구 내용에 따라 mixup data augmentation의 hyperparameter alpha의 값은 4로 설정하였다. 또한 Label dataset의 train batch size는 8, SC1과 SC2의 train batch\_size는 각각 50, 100으로 설정하였다. 또한 각각에 대해 warm training을 위해 label data는 10 epoch를 학습한 후, 각각의 SC1, SC2 label data를 통해서 5 epoch 학습을 진행하였다. MNIST보다 해상도가 높은 CIFAR10/100, SVHN 에서는 각 20회, 10회 epoch 간 학습했다.

4-2) Performance of CS1 & CS2

CS1 과 CS2 각 classification성능을 검토하겠다. 먼저 일 때, 한번에 1000개를 Sampling 한 경우와 100개씩 나눠 10번 Iteration 한 경우에 대해 각 데이터셋 별로 CS1, CS2 그리고 둘 다 적용한 결과는 Table1 과 같다. 이때 CS2의 Threshold는 0.5로 설정하였다.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Dataset | (||, P) | CS1 Only ( | | SC2 Only ( | |
| MNIST  (1 x 28 x 28) | 1000 (1000, 1) | 11992 | 99.59(%) | X | X |
| 1000 (100, 10) | 15305 | 99.97(%) | 14237(%) | 94.19(%) |
| FashionMNIST  (1 x 28 x 28) | 1000 (1000, 1) | 4217 | 96.84(%) | X | X |
| 1000 (100, 10) | 6492 | 95.56(%) | 10765(%) | 65.54(%) |
| EMNIST-Letter  (1 x 28 x 28) | 1000 (1000, 1) | 4697 | 93.52(%) | X | X |
| 1000 (100, 10) |  |  | 0 | 0 |
| CIFAR10  (3 x 32 x 32) | 1000 (1000, 1) | 15 | 66.67(%) | X | X |
| 1000 (100, 10) | 46 | 73.91(%) | 2250(%) | 20.56(%) |
| CIFAR100  (3 x 32 x 32) | 1000 (1000, 1) | 9 | 33.33(%) | x | X |
| 1000 (100, 10) | 51 | 76.46(%) | 0 | 0 |

[Table1 각 데이터셋 별 CS1, CS2 결과]

CS1은 저해상도의 데이터셋인 MNIST, Fashion EMNIST에서 아니라 95% 가량의 정확도를 가진다. 반면 CS2의 경우 MNIST 데이터셋 외에서는 성능이 좋지 않다. 이때 FashionMNIST, EMNIST-Letter의 경우 Threshold를 어떻게 설정하는 가에 따라 정확도가 향상될 수 있다. 자세한 내용은 4-3)에서 다루겠다. 하지만 높은 해상도 데이터셋인 Cifar 10 / 100 에서는 성능이 크게 악화되는 것을 확인할 수 있다. 이는 고해상도의 이미지의 경우 CAE model의 성능에 한계가 있기 때문으로 볼 수 있다. CS1, CS2각 방법은 Convolution auto encoder의 성능에 크게 의존한다. CAE의 성능이 충분히 좋다면 각 Class별로 Cluster 하면서 차원 축소할 것이란 가정 하에 CS1, CS2 방법을 적용한다. 즉, 해상도가 낮은 MNIST, FashionMNIST, EMNIST-Digit의 경우 13-CNN 기반의 CAE로 충분했으나, 해상도가 높은 CIFAR 10, CIFAR 100에 대해선 그렇지 않았다는 것이다. 해당 부분에 대해서는 ResNet 18 또는 또다른 dimension reduction 및 Clustering 적용에 대해 검토할 필요가 있다.

추후 충분한 CAE의 성능을 확보했다 가정할 때, 다양한 Sampling size 과 Sampling 및 Iteration의 관계를 확인하겠다. 현재 MNIST Dataset에서 CS1, CS2의 성능이 좋게 나오는 점으로 보아, 현재 CAE가 충분한 성능을 가진 것으로 볼 수 있다. 이에 각 경우에 대해 결과를 Table 2에서 확인할 수 있다.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| (||, P) | CS1 Only ( | | SC2 Only ( | | Both ( | |
| 50 (50,1) | 738(x14) | 99.04(%) | X | X | X | X |
| 100 (100, 1) | 3886(x38) | 99.84(%) | X | X | X | X |
| 250 (250, 1) | 4024(x16) | 98.03(%) | X | X | X | X |
| 500 (500, 1) | 9354(x18) | 99.66(%) | X | X | X | X |
| 750 (750, 1) | 10430(x14) | 99.75(%) | X | X | X | X |
| 1000 (1000, 1) | 11992(x12) | 99.59(%) | X | X | X | X |
| 100 (10, 10) | 4505(x45) | 100(%) | 4268(x42) | 99.5(%) | 4971(+466) | 99.02(%) |
| 250 (25, 10) | 8874(x35) | 99.92(%) | 8244(x33) | 99.68(%) | 9961(+1087) | 99.56(%) |
| 500 (50, 10) | 11486(x22) | 99.85(%) | 10672(x21) | 98.2(%) | 12200(+714) | 98.97(%) |
| 750 (75, 10) | 13012(x17) | 99.84(%) | 13810(x18) | 99.17(%) | 13012(+260) | 99.84(%) |
| 1000 (100, 10) | 15305(x15) | 99.78(%) | 14237(x14) | 94.19(%) | 15305(+252) | 99.47(%) |

[Table 2. MNIST 에서의 CS1, CS2 결과]

Cs1, CS2의 성능은 서로 유사하다. 대부분의 경우 정확도가 99%에 달하는 것을 확인할 수 있다. 즉, 정확도에 있어서 Sampling size 및 Iteration 유무는 큰 영향을 미치지 않는다. 각 Sampling size에 대해서 CS1, CS2 방법 둘 다 unlabeled train dataset에서 Classification이 가능한 데이터는 주어진 label data에 10~40배 정도이다. CS1을 적용한 후, 추가적으로 CS2을 unlabeled data에 적용했을 때 추가적으로 classification이 가능하다. 이때 sampling size를 키움에 따라 추가적인 CS2의 Classification의 개수가 줄어든다. 이로 보아 CS1, CS2는 유사한 조건의 unlabeled data에 대해서 classification을 하나, 일부 영역이 다름을 알 수 있다. 단, Sampling size가 커짐에 따라 두 방법 간의 classification 영역은 줄어드는 것을 확인할 수 있다. 또한 Sampling size가 증가함에 따라서 Classification이 가능한 개수와의 배수는 점차 줄어드는 것을 확인할 수 있다. 이는 경계부 또는 서로 다른 class 데이터들이 중첩된 곳에 대해서 Sampling size를 늘린다고 해도 classification이 불가능하기 때문으로 판단된다.

주어진 Sampling size를 P 번의 Iteration에 나눠 Sampling 할 때 SC1을 통해 Classification 가능한 data의 수가 늘어난다. 이는 Coreset selection 과정을 통해 설명할 수 있다. Coreset selection에 따라 데이터를 1개씩 Sampling 할 때마다 subgraph의 관계와 radius가 바뀐다. 즉, 100개의 데이터를 Sampling 했을 때와 101개의 데이터를 sampling 했을 때 SC1의 대상이 되는 data는 변화한다. Coreset selection 관점에서 100개의 Data을 Sampling 하는 것과 이를 10번으로 나눠 10개씩 Sampling하는 것 사이에는 차이점이 없다. 차이는 각 순간의 subgraph 관계를 통해 SC1을 적용할 수 있는 횟수만 다를 뿐이다. 즉, P번의 Iteration은 총 의 Coreset selection 간 P번의 순간을 SC1을 통해 Snapshot 한 것이다. 또한 Iteration 횟수가 충분히 있을 때 SC2 방법 또한 같이 적용할 수 있다. 이로써 SC1, SC2 방법 각각이 정확도를 보장할 수 있을 때 Iteration 횟수를 늘릴수록 보다 많은 data를 Classification할 수 있게 된다.

하지만 각 Iteration 마다 sampling size를 일정 이상 낮출 순 없다. 우연에 의해 클래스간 경계부에 있음에도 우연에 의해 접하는 subgraph의 Class가 모두 동일한 경우가 존재한다. 이때 CS1와 CS2의 정확도는 대폭 악화된다. 이를 방지하기 위해선 Sampling size를 키워 가능한 접하는 subgraph의 수가 많도록 한다. 또한 Sampling size가 클수록 Radius의 값이 작아져 보다 촘촘하게 모든 데이터를 덮을 수 있다. 단 고정된 에 대해서 Sampling size는 Iteration 과 Trade off 관계로 적절히 조절할 필요가 있다. 그 외로 접하는 subgraph의 수에 대해 제한을 두거나, subgraph의 density에 조건을 둠으로써 방지할 수 있을 것으로 보인다. 자세한 내용은 4-3)에서 다루겠다.

4-3) Sensitivity test about hyperparameter

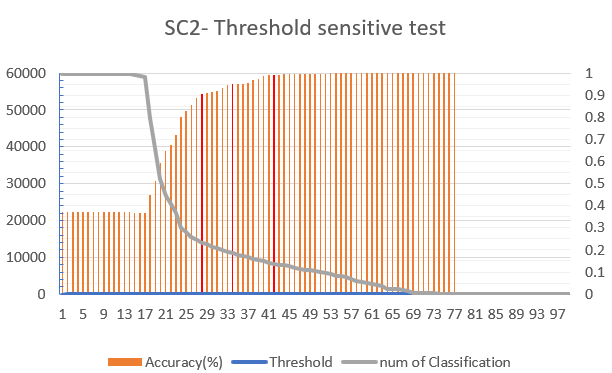
데이터셋 MNIST을 통해 각각의 Hyperparameter가 어떠한 영향을 미치는 지 알 수 있다면, 추후 다른 데이터셋에서 보다 효율적으로 hyperparameter 세팅이 가능할 것이다. SC1에서의 Subgraph 밀도, 접하는 subgraph의 수, 그리고 SC2에서의 Threshold에 대한 sensitivity test를 진행한다. K가 250인 상황에서의 각 hyper parameter의 값에 따른 정확도와 Classification 가능한 수는 다음과 같다.

MNIST Dataset 한정.

Good case : 밀도 / subgraph 수에 따른 정확도 및 Classification 성능

Bad case : 이것을 방지할 수 있는가?

Fashion MNIST / EMNIST Dataset에서도 Threshold 점검 필요



SC2 Classification에서 각각 빨간색으로 칠해진 부분은 각각 0.29, 0.37, 0.42로 각각 정확도가 85%, 90%, 99%가 넘은 지점이다. 이를 볼 때 Threshold는 각 특정 값이 속할 확률을 그대로 반영하는 것 보다 엄격하게 보는 지표라고 볼 수 있다. 한편 Classification이 가능한 data의 수와 Accauracy는 Trade off 관계임을 알 수 있다. 단, 적정한 Threshold는 Iteration의 크기에 따라 달라질 수 있다. 이에 Threshold가 0.5로 최소 확률이 과반수를 넘기도록 설정하는 것이 필요하겠다.

4-4) 문제점 발생

MNIST 데이터셋에서 CS1, CS2을 통해 높은 정확도의 Labeled data를 CNN 모델 학습에 활용할 때 오히려 Test dataset에 대한 정확도가 떨어진다.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| (||, P) | CS1 | CS2 | Total Acc | CNN Acc  ( only) | CNN Acc  (+) | CNN Acc  (+) |
| 100 (10, 10) | 4505 | 466 | 99.02 | 49.84 | 32.87 | 31.65 |
| 250 (25, 10) | 8874 | 1087 | 99.56 | 54.45 | 39.23 | 44.52 |
| 500 (50, 10) | 11486 | 714 | 98.97 | 68.26 | 52.35 | 58.03 |
| 750 (75, 10) | 13012 | 260 | 99.84 | 63.03 | 54.77 | 55.42 |
| 1000 (100, 10) | 15305 | 252 | 99.47 | 67.60 | 58.17 | 53.12 |

[Table 3. CS1, CS2 Labeling 활용]

이에 대해선 크게 3가지 해석이 가능하다. 먼저 CS1, CS2 방법을 통해 Classification 하는 데이터들은 클래스 중심부에 속하는 Data들로 다들 유사한 특성을 지닐 것이다. 따라서 유사한 데이터들을 학습에 활용한다고 해서 CNN 모델에 큰 영향을 미치지 못한다. 두번째, CS1, CS2의 방법은 다른 Class와 잘 구별되는 특정 Class의 데이터만을 Classification 한다. MNIST의 경우 0, 4, 6 클래스 data에 대해서는 SC1, SC2의 classification 조건에 잘 부합했으나, 다른 클래스의 경우 소수에 불가했다 즉, 모델의 일반화가 아닌 특정 클래스에 대한 과적합을 시키게 된다. 특히 본 paper에는 담지 않았으나 CS1, CS2 이 100%의 정확도를 가질 때에도 성능이 동일하거나 오히려 악화되는 경우가 있었다. 이로 보아 label 데이터 분포가 unbalanced할 때 모델 학습에 악영향을 미치는 것으로 보인다. 세번째, 오히려 유사한 특성이나 잘못된 Classification 된 값이 모델 전체 학습에 큰 영향을 미친다는 것이다. CAE의 성능 부족으로 인해 feature들은 유사했으나, 원본 데이터는 큰 차이를 띌 수 있다. 또한 학습 간 잘못 labeling 된 데이터가 들어갔을 때 모델의 Loss값이 증가되는 것을 확인할 수 있었다.

5. 보완 및 개선할 점

현재 SC1, SC2 방법의 가장 큰 한계점은 해상도가 낮은 데이터셋에 한정적으로 적용할 수 있다는 점이다. 이에 Active learning을 적용할 수 있는 저차원으로 mapping 하면서 동시에 각 클래스별로 Clustering이 잘 되도록 하는 방안을 강구할 필요가 있다. 현 논문에서는 13 CNN 모델에 의존했다. 추후 ResNet 18, unsupervised clustering, unsupervised representation learning method for image data 인 DGI 등 다양한 representation 방법을 적용해볼 필요가 있다. 다음으로 Train dataset에 대한 높은 Classification이 CNN 모델의 성능 향상으로 이어지지 않는다는 한계점이다. 해당 문제점은 보다 성능이 좋은 Representation learning을 적용하면 일부 해소될 수 있다. Features extraction을 했을 때 각 Class 별로 잘 구분되도록 Clustering된다면, 다양한 Class 데이터에 대해 SC1, SC2을 적용할 수 있다. 또는 Coreset selection을 통해 뽑은 Label data의 Class 비율이 본래 dataset의 Class 비율과 유사하다 가정함으로써, Unbalanced 한 Cs1, CS2로 Labeling 한 dataset의 영향력을 조절할 수 있다. 또한 소량의 misclassification data에 대해서도 outlier detection을 적용할 여지가 있다. 잘못 labeling한 데이터의 경우 loss 값이 크게 튀어오른 것을 확인하였다. 따라서 Loss의 값이 일정 값 아래인 데이터에 한정하여 학습한다면 잘못 labeling 한 것을 방지할 수 있다. 본 방법은 Cs1, CS2 의 정확도가 95% 이상으로 소수의 mislabeling Outlier을 포착할 수 있을 것으로 보인다.

그 외에 현재 CS1, CS2의 방법을 향상할 여지가 존재한다. CS1 방법은 주어진 Sampling size를 P 번의 Iteration에 나눠 Sampling 할 때 Classification 가능한 data의 수가 늘어났다. 이는 Coreset selection에 따라 데이터를 1개씩 Sampling 할 때마다 subgraph의 관계와 radius가 바뀐다. 즉, 100개의 데이터를 Sampling 했을 때와 101개의 데이터를 sampling 했을 때 SC1의 대상이 되는 data는 변화한다. P번의 Iteration은 총 의 Sampling 간 P번의 순간을 SC1을 통해 Snapshot 한 것이다. 이를 확장하면 개의 데이터를 Sampling 하는 것은 각 1개씩 총 번의 Iteration을 적용할 수 있다. 이는 SC1 방법의 Classification 개수를 향상시킨다. 무엇보다 각 Iteration 마다 subgraph을 형성하기 때문에, 각 점은 최소 개의 subgraph들에 포함된다. 이로써 SC2 방법은 운의 요소에 의존했던 기존의 방법에서 벗어나 매 순간 뛰어난 성능을 보장할 수 있을 것이다. 마지막으로 현 연구에서는 CS1, CS2의 방법을 Train dataset 내에서의 Classification에 초점을 잡았다. 하지만 Train dataset의 크기가 충분히 커 모집단의 역할을 할 수 있다면, Train dataset에서 학습한 representation learning은 Test dataset에서도 적용할 수 있다. 이는 SC1, SC2의 방법을 Test Dataset에도 적용할 여지가 있음을 시사한다.

**5. 예상 기대효과**

본 연구는 Coreset selection의 Subgraph를 기하적인 관점에서 활용한 Classification 방법을 제시한다. 이를 Pseudo labeling으로 활용함으로써 Label Data 부족으로 인한 confirmation bias를 해소할 수 있다. 더 나아가 Classification의 성능을 보장하기 위해 필요로 하는 Sample size에 대한 bound를 제시한다. 이는 현장에서 Classification 성능 확보까지 필요로 하는 Labeling Cost를 계산할 수 있는 여지를 제공한다.

본 방식은 Dataset의 특성에 따라 성능이 변한다. 데이터셋의 크기가 크고 클래스가 적어 밀도가 높은 Dataset일수록 Classification 성능이 뛰어날 것이다. 한편 본 방법은 Distance을 측정할 수 있으며, distance가 데이터 간의 특성의 유사성을 반영하는 모든 Dataset에 적용할 수 있다. 더불어 representation learning과 함께 활용한다면 시너지를 낼 수 있다는 점에서 가치가 있다.

**<참고문헌>**

Arazo, E., Ortego, D., Albert, P., O’Connor, N. E., & McGuinness, K. (2020). *Pseudo-labeling and confirmation bias in deep semi-supervised learning.* Paper presented at the 2020 International Joint Conference on Neural Networks (IJCNN).

Argote, L., & Ingram, P. (2000). Knowledge transfer: A basis for competitive advantage in firms. *Organizational behavior and human decision processes, 82*(1), 150-169.

Cascante-Bonilla, P., Tan, F., Qi, Y., & Ordonez, V. (2021). *Curriculum labeling: Revisiting pseudo-labeling for semi-supervised learning.* Paper presented at the Proceedings of the AAAI Conference on Artificial Intelligence.

Chapelle, O., Scholkopf, B., & Zien, A. (2009). Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks, 20*(3), 542-542.

Gal, Y., Islam, R., & Ghahramani, Z. (2017). *Deep bayesian active learning with image data.* Paper presented at the International Conference on Machine Learning.

Grandvalet, Y., & Bengio, Y. (2004). Semi-supervised learning by entropy minimization. *Advances in neural information processing systems, 17*.

Hurtik, P., Molek, V., & Perfilieva, I. (2020). Novel dimensionality reduction approach for unsupervised learning on small datasets. *Pattern Recognition, 103*, 107291.

Lee, D.-H. (2013). *Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks.* Paper presented at the Workshop on challenges in representation learning, ICML.

Liu, P., Zhang, H., & Eom, K. B. (2016). Active deep learning for classification of hyperspectral images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 10*(2), 712-724.

Luo, Y., Zhu, J., Li, M., Ren, Y., & Zhang, B. (2018). *Smooth neighbors on teacher graphs for semi-supervised learning.* Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

Qiao, S., Shen, W., Zhang, Z., Wang, B., & Yuille, A. (2018). *Deep co-training for semi-supervised image recognition.* Paper presented at the Proceedings of the european conference on computer vision (eccv).

Ren, P., Xiao, Y., Chang, X., Huang, P.-Y., Li, Z., Gupta, B. B., . . . Wang, X. (2021). A survey of deep active learning. *ACM computing surveys (CSUR), 54*(9), 1-40.

Sajjadi, M., Javanmardi, M., & Tasdizen, T. (2016). Regularization with stochastic transformations and perturbations for deep semi-supervised learning. *Advances in neural information processing systems, 29*.

Sener, O., & Savarese, S. (2017). Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*.

Shi, W., Gong, Y., Ding, C., Tao, Z. M., & Zheng, N. (2018). *Transductive semi-supervised deep learning using min-max features.* Paper presented at the Proceedings of the European Conference on Computer Vision (ECCV).

Sinha, S., Ebrahimi, S., & Darrell, T. (2019). *Variational adversarial active learning.* Paper presented at the Proceedings of the IEEE/CVF International Conference on Computer Vision.

Sun, Y., Li, J., Wang, W., Plaza, A., & Chen, Z. (2016). *Active learning based autoencoder for hyperspectral imagery classification.* Paper presented at the 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS).

Tong, S. (2001). *Active learning: theory and applications*: Stanford University.

Wang, K., Zhang, D., Li, Y., Zhang, R., & Lin, L. (2016). Cost-effective active learning for deep image classification. *IEEE Transactions on Circuits and Systems for Video Technology, 27*(12), 2591-2600.

Wang, Y., Yao, H., & Zhao, S. (2016). Auto-encoder based dimensionality reduction. *Neurocomputing, 184*, 232-242.

Yang, X., Song, Z., King, I., & Xu, Z. (2022). A survey on deep semi-supervised learning. *IEEE Transactions on Knowledge and Data Engineering*.

Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2017). mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.