논문스터디 1주차

장이준 이선민

Combining Labeled and Unlabeled Data with Co-Training

Avrim Blum, Tom Mitchell, 1998

목차

1. Semi-supervised Learning : Self-Training 과 Co-Training

1) Semi-supervised Learning

Supervised Learning의 한계

딥러닝의 가장 대표적인 방법론은 지도학습이지만, 지도 학습은 정답(label)이 있는 데이터의 패턴을 외우는 학습법에 불과하다. 즉, 전혀 새로운 데이터(label이 없는 데이터)에 대해서는 정답을 쉽게 맞춘다는 보장이 없다. 이러한 관점에서, 성공적인 딥러닝을 도입한 이미지 처리의 경우 역시 언제나 대용량 labeled 데이터가 확보되어야만 좋은 성능을 얻을 수 있다고 볼 수 있다.

그러나 labeled 데이터를 대용량으로 확보하기 어려운 경우에는 어떻게 해야할까? 가령 labeling에 고도의 전문성이 요구되거나 labeling 과정에서 오랜 시간이 걸리는 경우 효과적으로 labeled 데이터를 얻기 힘들 수 있다.

이렇게 충분한 labeled 데이터가 확보되지 않은 경우, 학습 데이터가 새로운 데이터(unlabeled data, 혹은 test data)의 분포 전체를 커버하지 못할 수 있기 때문에 성능이 떨어질 수 있다는 문제가 있다. 따라서 지도 학습은 train data에 대해서는 좋은 성능을 발휘할 수 있으나 이것이 전혀 새로운 data에 대한 성능을 보장해주지 못한다는 것이다.

Semi-supervised Learning

위와 같은 문제를 해결하기 위한 방법론으로, 적은 labeled data만 존재하는 상황에서, 추가로 활용될 수 있는 대용량의 unlabeled data를 이용하는 semi-supervised learning이 있다. 준지도학습의 핵심을 요약하자면,

“소량의 labeled data에는 supervised learning을 적용, 대량의 unlabeled data에는 unsupervised learning을 적용해 추가적인 성능 향상을 목표로 하는 방법론”

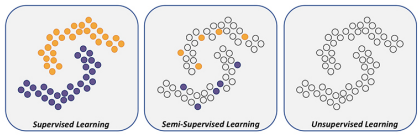
이다.

Semi-supervised learning에는 몇가지 가정과 방법론들이 있는데, 이는 reference를 참고하여 읽어보면 좋을 것 같다!

Semi-supervised learning 방법론 소개

안녕하세요. 이스트소프트 A.I. PLUS Lab입니다. 이번 포스팅에서는 머신러닝의 학습 방법 중 하나인 준지도학습(semi-supervised learning, SSL) 에 대해 다루어보려고 합니다. SSL 자체가 워낙 거대한 주제이기 때문에 이번 포스트에서 전체적인 내용을 모두 다루기

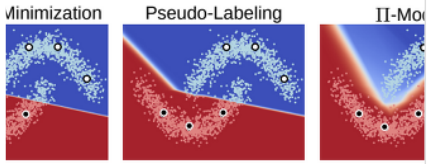
https://blog.est.ai/2020/11/ssl/



Semi-supervised learning (준지도학습): 개념과 방법론 톺아보기

Semi-supervised learning overview 논문 [14-16] 및 여러 방법론 관련 논문들 [1-13] 읽으며 얻은 지식을 바탕으로 글로 정리해보려고 한다. 오랜만에 쓰는 기술글이라 설렌다!! 나도 처음 공부하는 분야이기 때문에 부족한 부분도 있겠지만, 틀린 부분이나 덧붙여 설명

https://sanghyu.tistory.com/177



2) Self-Training

Self-Training은 이후의 Co-training과 함께 Semi-supervised Learning의 가장 기본적인 학습 방법이다.

Self-Training의 학습 과정

labeled data (X_l, y_l)과 unlabeled data X_u 가 주어졌을 때 self-training의 학습 과정을 살펴보겠다.

1) 주어진 labeled data (X_l, y_l) 를 이용하여 예측함수 f를 학습시킨다.

	x_1	x_2	$\cdots$	x_{d-1}	x_d	y
I_1	x_{11}	x_{12}	$\cdots$	$x_{1\,d-1}$	x_{1d}	y_1
I_2	x_{21}	x_{22}	$\cdots$	$x_{2\,d-1}$	x_{2d}	y_2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
I_l	x_{l1}	x_{l2}	$\cdots$	$x_{l\,d-1}$	x_{ld}	y_l

$$y_l = f(x_l)$$

2) 학습이 완료되면 예측 함수 f를 사용해 y값이 없는 unlabeled data X_u의 예측값 $\hat{y_u}$를 구한다.

	x_1	x_2	$\cdots$	x_{d-1}	x_d
I_{l+1}	$x_{l+1\,1}$	$x_{l+1\,2}$	$\cdots$	$x_{l+1\,d-1}$	$x_{l+1\,d}$
I_{l+2}	$x_{l+2\,1}$	$x_{l+2\,2}$	$\cdots$	$x_{l+2\,d-1}$	$x_{l+2\,d}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
I_{l+u}	$x_{l+u\,1}$	$x_{l+u\,2}$	$\cdots$	$x_{l+u\,d-1}$	$x_{l+u\,d}$

$\Downarrow$

	x_1	x_2	$\cdots$	x_{d-1}	x_d	y
I_{l+1}	$x_{l+1\,1}$	$x_{l+1\,2}$	$\cdots$	$x_{l+1\,d-1}$	$x_{l+1\,d}$	$\hat{y}_{l+1}$
I_{l+2}	$x_{l+2\,1}$	$x_{l+2\,2}$	$\cdots$	$x_{l+2\,d-1}$	$x_{l+2\,d}$	$\hat{y}_{l+2}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
I_{l+u}	$x_{l+u\,1}$	$x_{l+u\,2}$	$\cdots$	$x_{l+u\,d-1}$	$x_{l+u\,d}$	$\hat{y}_{l+u}$

3) 기존의 labeled data와 예측함수 f에 의해 새롭게 labeling된 unlabeled data를 합치고, 이를 사용해 새로운 예측함수 g를 학습한다.

즉, 예측함수 f를 학습하는데 사용됐던 기존의 labeled data와, 새롭게 labeling된 unlabeled data가 합쳐져 하나의 train data로서 작용하는 것이다.

	x_1	x_2	$\cdots$	x_{d-1}	x_d	y
I_1	x_{11}	x_{12}	$\cdots$	$x_{1\,d-1}$	x_{1d}	y_1
I_2	x_{21}	x_{22}	$\cdots$	$x_{2\,d-1}$	x_{2d}	y_2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
I_l	x_{l1}	x_{l2}	$\cdots$	$x_{l\,d-1}$	x_{ld}	y_l
I_{l+1}	$x_{l+1\,1}$	$x_{l+1\,2}$	$\cdots$	$x_{l+1\,d-1}$	$x_{l+1\,d}$	$\hat{y}_{l+1}$
I_{l+2}	$x_{l+2\,1}$	$x_{l+2\,2}$	$\cdots$	$x_{l+2\,d-1}$	$x_{l+2\,d}$	$\hat{y}_{l+2}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
I_{l+u}	$x_{l+u\,1}$	$x_{l+u\,2}$	$\cdots$	$x_{l+u\,d-1}$	$x_{l+u\,d}$	$\hat{y}_{l+u}$

여기서 잠깐! unlabeled data에서 새롭게 labeling된 unlabeled data는 어떤 기준으로 기존의 labeled data에 합류하는 걸까? 어떻게 합칠 것인지에 대한 여러가지 방법이 있다.

- Add all $(X_u, \hat{y_u})$ to labeled data
- Add a few most confident $(x_u, \hat{y_u})$ to labeled data
- Add all $(x_u, \hat{y_u})$ to labeled data, weight each by confidence

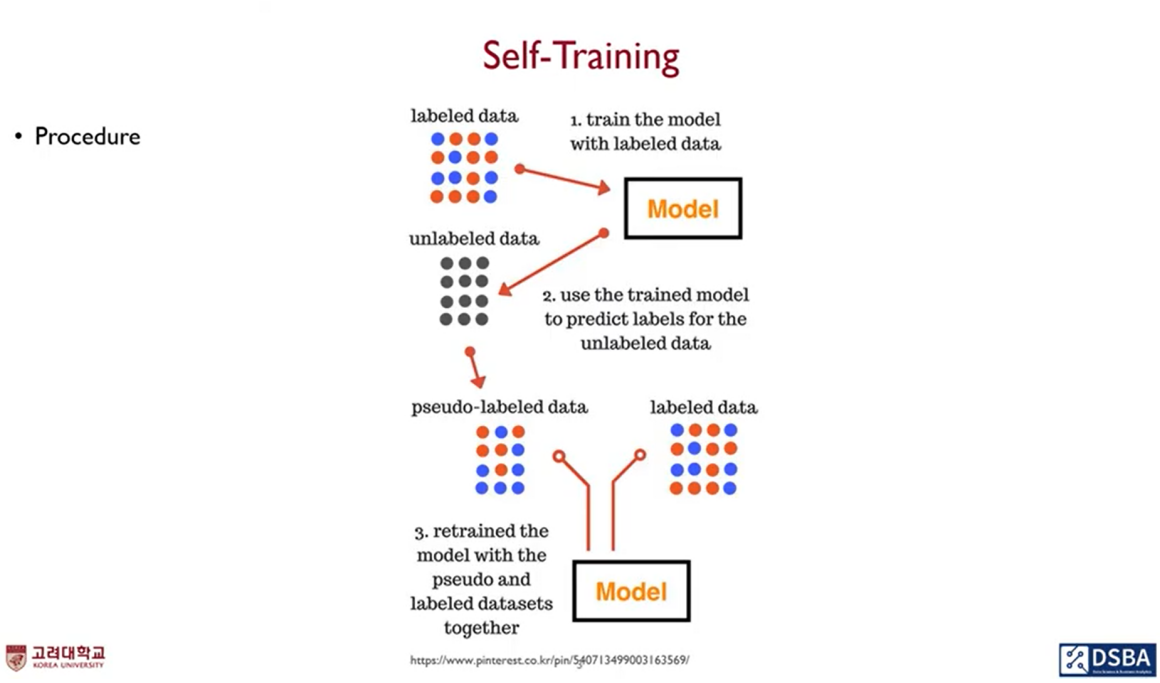
즉, 새롭게 labeling된 unlabeled data는 충분히 잘 추정된 값이라는 가정 하(high confidence) 에 labeled data에 합해진다. 여기서 self-training의 가장 기본적인 가정을 도출할 수 있다.

- One's own high confidence predictions are correct.

4) 위와 같은 과정을 unlabeled data가 없어지거나, 수렴할때까지 반복한다.

Self-Training의 학습 과정 정리

예측함수 f는 unlabeled data의 y값을 추정하는데 사용되고, 예측함수 g는 labeled & unlabeled data로 이루어진 전체 train data로 생성된 모델이며 이후 test 단계에서 사용된다.



Self-Training의 장단점

1) advantages

a. simplest semi-supervised learning method

b. a wrapper method, applies to existing complex classifers

c. often used in real tasks like natural language processing

2) disadvantages

a. early mistakes could reinforce themselves (강화 학습의 단점)

b. cannot say too much in terms of convergence

3) Co-Training

Collaborative training의 약자. 특정 object을 나타내는 여러 feature 중 mutually exclusive한, 즉 상호 배반적인 feature set이 있어야한다는 가정이 필요하다. 이러한 상호 배반적인 feature를 통해 각각 학습된 여러 개의 모델들이 서로 '협력'하며 학습을 진행하게 된다. 이때 하나의 object를 나타내는 다양한 feature들의 관점에서 바라보는 방법이므로 'multi-view algorithm'이라고도 한다.

이러한 co-training의 가정을 정리하면 다음과 같다.

- 하나의 object는 두 가지 이상의 feature set을 가지고 있어야 한다.
- 각 feature set은 서로에게 완전히 독립이어야 한다.(mutually exclusive)
- 각 feature set은 스스로 classifier의 생성이 가능해야 한다.

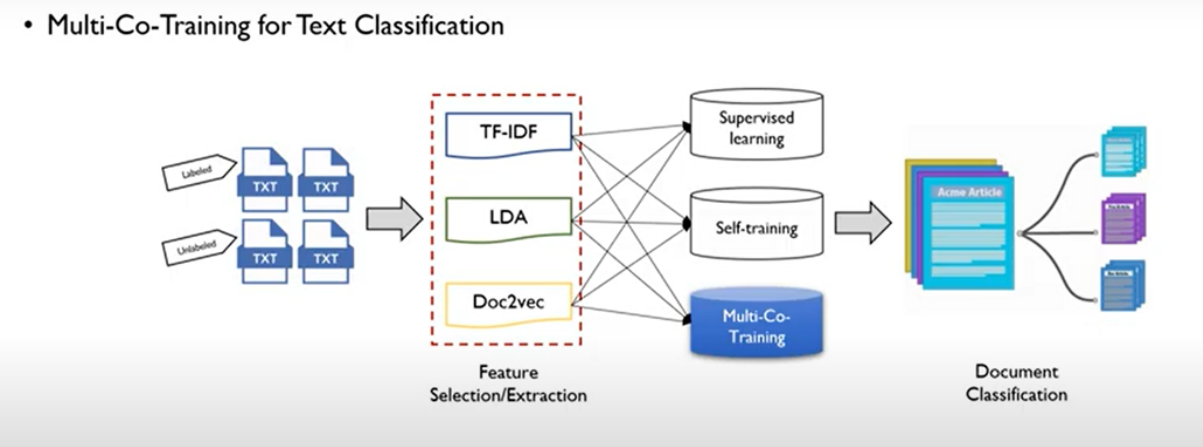
예를 들어, 어떤 객체의 feature로 image와 text가 있다고 가정할 때, image와 text라는 두 feature는 서로 관련성이 없으므로 위의 가정을 만족한다.

$$X = (x^{(1)}, x^{(2)})$$

$$x^{(1)} = image\ features$$

$$x^{(2)} = web\ page\ text$$

두 개의 구분되는 feature를 입력으로 사용해 학습하는 모델을 각각 만든다(supervised learning). 이 예시에서는 image classifier와 text classifier 두 가지 모델이 생성되는 것. 학습 과정은 self-training과 유사하다.

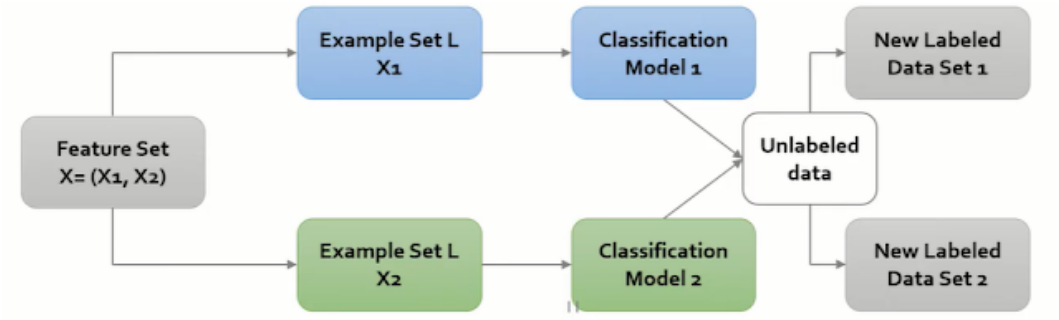


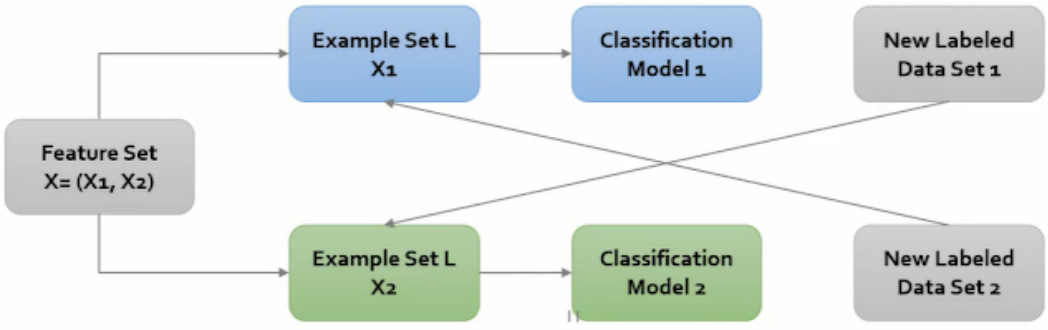
Co-Training의 학습 과정

1) 주어진 labeled data (X_l, y_l) 를 이용하여 예측함수 f를 학습시킨다.

2) 학습이 완료되면 예측 함수 f를 사용해 y값이 없는 unlabeled data X_u의 예측값 $\hat{y_u}$를 구한다.

3) 예측 결과 중 confidence가 높은 결과들을 '상대방' 모델의 재학습 데이터에 추가한다.





Confidence Measure

Confidence는 크게 두 가지 기준으로 표현된다.

1) Intra confidence $= -\ Entropy$

$-\ Engropy = \sum_{i=1} -p_i log_2\ p_i$

intra confidence는 현재 모델이 특정 class에 대해 얼마나 극단적인 예측을 하고 있는지 나타낸다.

2) Inter confidence = $-\ Training\ Error$

inter confidence는 각 모델의 예측율을 기반으로 얼마나 정확한지를 나타낸다.

3) Confidence

$Confidence\ measure = (-Entropy)\ *(-\ Training\ Error)$

즉, 해당 모델이 정확하면서 동시에 (위 예시에서는) 두 범주의 차이를 극단적으로 나타낼수록 해당 object들은 pseudo label (predicted label)이 정답일 가능성이 높은 것이다.

Self-Training & Co-Training

05_2 : Semi-Supervised Learning : Self-Training and Co-Training 우선 가장 기본적인 준지도학습 방법 Self-Training에 대하여 알아보자. 우리에게 주어진 정보는 Labeled data $(\mathbf{X}_l,y_l)$과 Unlabeled data $\mathbf{X}_u$이다. Self-training과정은 다음

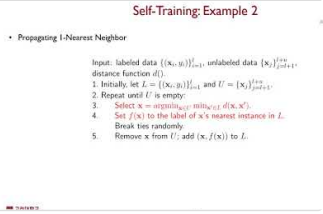
https://yupsung.blogspot.com/2021/02/self-training-co-training.html

$$\begin{array}{|c|c|c|c|c|}\hline x_1 & x_2 & \cdots & x_{d-1} & x_d \\ \hline x_{11} & x_{12} & \cdots & x_{1\,d-1} & x_{1d} \\ \hline x_{21} & x_{22} & \cdots & x_{2\,d-1} & x_{2d} \\ \hline \vdots & \vdots & \vdots & \vdots & \vdots \\ \hline x_{l1} & x_{l2} & \cdots & x_{l\,d-1} & x_{ld} \\ \hline \end{array}$$

05-2: Semi supervised Learning - Self-Training & Co-Training (자가학습 및 협동학습)

고려대학교 일반대학원 산업경영공학과비즈니스 애널리틱스5장: Semi-supervised LearningPart 2: Self-Training & Co-Traininghttps://github.com/pilsung-kang/Business-Analytics-IME654-

https://www.youtube.com/watch?v=5i-wVc8Jn-U

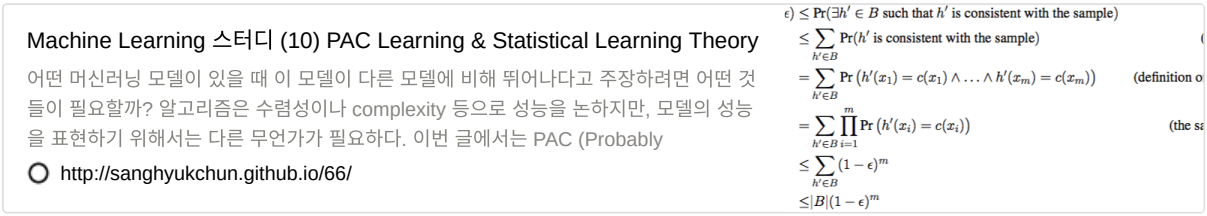


2. PAC Learning

Introduction

PAC Learning은 Probably Approximately Correct Learning의 약자로, 이론적으로 모델의 성능을 측정하는 방법 중 하나이다. 일반적으로 알고리즘은 수렴성이나 complexity 등으로 모델의 성능을 측정하지만, PAC learning은 개념적으로 모델이 언제, 왜 좋은지 설명할 수 있다. (따라서 pracital 하게 모델 성능을 논할 때 쓰이기는 힘들다!)

PAC는 Computational learning theory 분야에서 등장한 개념인데, 해당 개념의 등장 배경은 아래 reference를 참고해서 읽어보자.



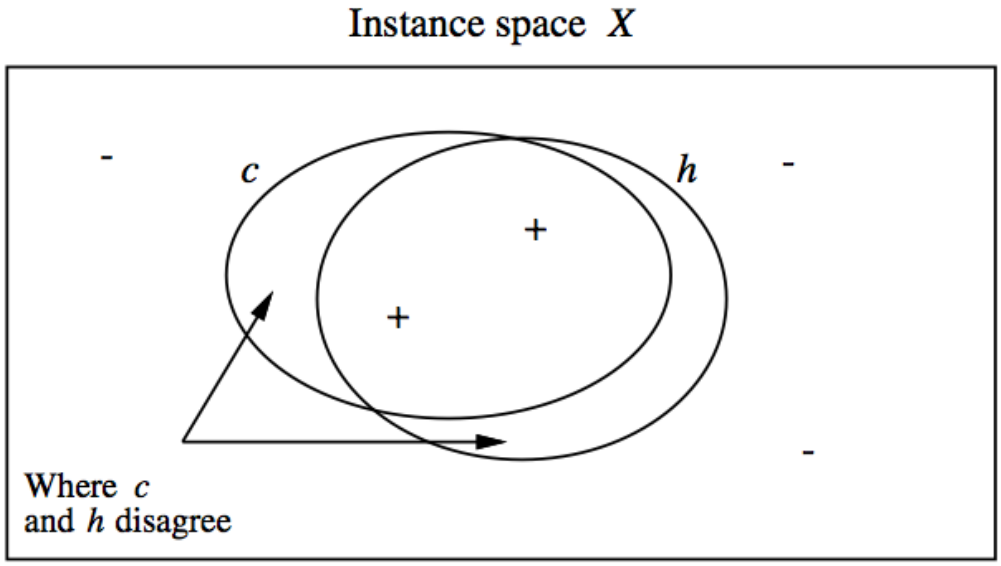
Terminology

PAC learning을 이해하는 데 필요한 용어에 대해 짚고 넘어가보도록 하겠다.

- $Instance\ X$: 모든 데이터의 공간. 예를 들어 28 by 28 사이즈의 흑백 이미지의 경우 2^{784} 크기의 set이 될 것이다.
- $Target\ Concept\ c$: target function으로 이해하면 된다. 주어진 데이터가 어떤 값을 가지는지 판단하는 함수
- $Hypothesis\ Space\ H$: 주어진 c와 최대한 비슷한 approximated function(즉, hypothesis) h가 속하는 space. 예를 들어, 함수가 linear라 가정한다면, Hypothesis space는 모둔 linear function의 function space가 된다.
- $Training\ Data\ D$: 모든 instance space를 다 볼 수 없으므로, 그 중 일부의 데이터만이 training data로 주어진다.

True error & Training error

위의 4가지 요소가 주어졌을 때, 모든 모델의 목표는 전체 데이터셋 $x \in X$ 에 대해 가장 target concept $c(x)$와 비슷한 hypothesis $h(x)$를 찾는 것이다. 하지만 실제로는 모집단이 아닌 sample $x \in D$ 만을 관측할 수 있으므로, 필연적으로 발생하는 'training error'를 고려하며 hypothesis $h(x)$를 찾아야 한다. 여기서 중요한 점은 전체 instance X의 subset 인 training instance D 에서 error를 최소화시키는 모델이, 전체 instance X에 대해서도 여전히 작은 error를 보일 것인지, 그렇지 않다면 training error와 true error가 얼마나 차이가 날 것인지이다. 이 둘의 관계를 그림으로 나타내면 다음과 같다. (만약 training error는 낮지만 test error, 혹은 true error가 높은 경우 overfitting이 발생하는 것)



이렇듯 절대 0이 될 수 없는 train error와test error간 차이의 존재는 머신 러닝에서 고려되어야 할 불가피한 문제이다. 따라서 “아주 작은 error를 높은 확률로 달성하기 위해 어느 정도의 데이터가 필요한지”에 대한 질문을 생각해볼 수 있다.

(true error와 training error에 대한 자세한 내용 역시 reference를 참고하자.)

이를 다음 예시로 이해해보자. 주어진 트레이닝 데이터에 대한 모델의 에러가 다음과 같을 때,

- 모델1 - training error가 0
- 모델2 - training error가 0
- 모델3 - training error가 0.1

모델3이 모델1, 2보다 더 좋다고 할 수 있을까? train data에 대해서는 모델 1, 2가 모델3 보다 좋은 것은 자명하지만, unseen data(즉 test data)에 대해서도 모델1, 2가 모델3 보다 완벽하게 예측할 수 있을 것이라는 보장은 없다. 따라서 특정 train data에 대해서 발생한 training error가 아닌, 일반적인 상황에서의 일반화된 오류를 계산하기 위한 이론이 바로 PAC Learning이다.

PAC Learning

PAC Learning은

“높은 확률로(Probably) 주어진 모델이 작은 error를 가진다(Approximately Correct)”

와 같은 분석을 진행할 수 있다.

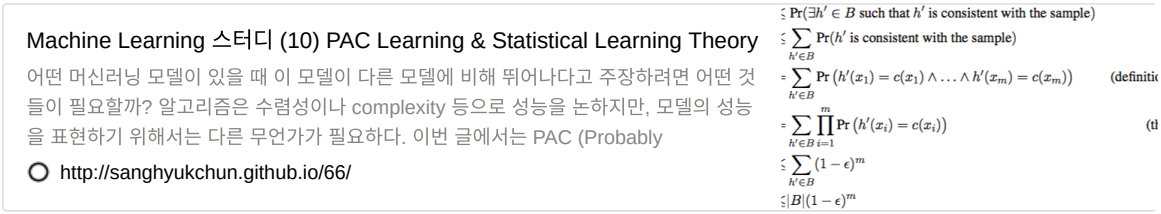
PAC bound는 모델의 성능, 즉 generalization과 overfitting에 대한 intuition을 나타내며, 이를 이해하기 위해서는 다음의 4가지 요소들이 필요하다.

- train data의 샘플 수 m
- train error와 true error의 gap ϵ, (이때 $error_{true}(h) \leq error_{train}(h) + \epsilon$)
- hypothesis space의 complexity $|H|$
- confidence of the relation : at least $(1 - \delta)$

PAC bound는 다음과 같이 주어진다.

$$P_r[error_{true}(h) \leq error_{train}(h) + \epsilon] \leq ||H||exp(-2m\epsilon^2)$$

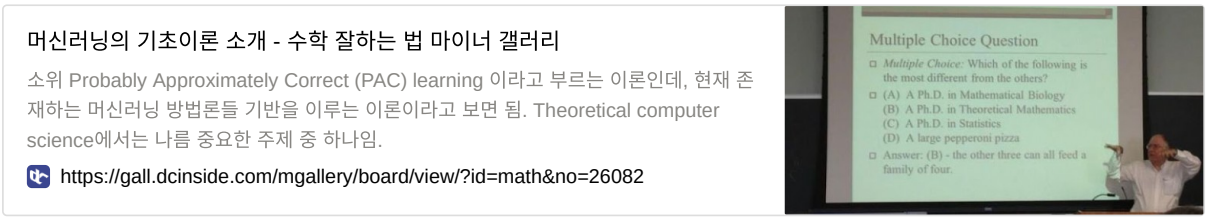
▼ 위 수식에 대한 증명은 reference 참고 부탁드립니다.



PAC Learning 정리

PAC Learning으로부터 얻을 수 있는 intuition은 다음과 같다.

- training data의 개수가 많을수록 모델은 더 generalize되고, 따라서 overfitting의 위험성도 작아진다.
- model complexity가 높을수록 모델의 generalization은 더 어려워지고, overfitting의 위험성도 커진다.
- model의 성능을 높이려면 train error를 줄이거나, training sample을 늘리거나, 모델의 complexity를 줄여야한다.



3. 논문 정리

1) Introduction

이번 논문에서는 웹 페이지 분류 문제에 co-training을 적용하는 예시를 담고 있다. 특정 웹 페이지가 home page인지 아닌지를 분류하는 문제를 생각해보자. 특정 웹 페이지를 나타낼 수 있는 정보로는 두 가지 종류가 있고 이는 다음과 같다.

i. text appearing on the document itself

ii. anchor text attached to hyperlinks

첫째는 '홈페이지'임을 나타내는 웹 페이지 상의 문자이며, 두 번째는 하이퍼링크(url)에 나타나 있는 문자를 의미한다.

좀 더 자세히 예를 들어 보자. 아주 적은 labeled example들만을 가지고 웹 페이지 분류기를 학습시킨 결과, 'research interest'라는 웹 페이지 상의 문자는 해당 웹 페이지가 'home page'임을 나타내는 weak indicator 라고 할 수 있다. 왜 *weak* indicator 일까? 'research interest'라는 문구가 'home page'임을 보장하기 위한, 충분히 많은 labeled data의 학습 결과가 아니기 때문이다. 마찬가지로, 'my advisor'라는 웹 페이지 url에 속한 문구 역시 해당 웹 페이지가 'home page'임을 나타내는 weak indicator 라고 할 수 있다.

그렇다면 위와 같은 초기 시행에서, 'reseach interest'와 'my advisor' 모두 특정 웹 페이지가 'home page'임을 나타내지만, 확실히 보장하지 못하는 애매한 상황인 것이다.

> “Then, attempt to boostrap from these weak predictors using ***un***labeled data.”

즉 text classifier를 train 시킬 때, **기존 link 기반 predictor**의 학습데이터인 ‘my advisor’ 데이터를 ‘probably positive’ example로서 **word 기반 predictor**의 새로운 train data로 사용한다는 것이고 이러한 방법은 ‘research interest’의 labeling 결과에 confidence를 더해줄 수 있는 것이다. 이러한 co-training 개념을 적용하여 그 반대 절차 또한 가능하게 된다.

정리하자면, co-training을 통해 unlabeled data를 augement 하여 labeled data로 사용할 수 있는 것.

이러한 co-training이 효과적인 것일까? 이에 대한 답은 PAC Learning을 살펴보며 찾을 수 있다.

2) A Formal Framework

우선 논문을 설명하는데 필요한 용어들을 보도록 하겠다.

- views란 특정 instance를 표현할 수 있는 feature에 해당한다. 예를 들어, 우리는 document를 tf-idf를 통해 추출된 단어로도 표현할 수 있고, 혹은 LDA를 통해 추출된 토픽으로도 표현할 수 있다. 이때 tf-idf로 추출된 단어를 하나의 view, LDA를 통해 추출된 토픽들을 또 다른 view라고 볼 수 있다. 이때 우리는 하나의 view로도 정확한 분류를 진행할 수 있다고 가정한다.
- instance $X = (x1, x2)$라는 데이터로부터 D라는 분포가 나왔고,
- C_1과 C_2라는 concept class가 존재한다면(여기서 concept class란 각각의 views를 label로 매핑할 수 있는 모든 target 함수들 f들의 집합을 의미한다. 즉 $C : X \rightarrow Y$를 의미한다. $C_1(x)$는 $f(x_1) : X_1 \rightarrow Y$ 함수들의 집합, $C_2(x)$는 $f(x_2) : X_2 \rightarrow Y$ 함수들의 집합, 말한다.
- 이 D라는 분포 안의 값들은 $f(x) = f_1(x_1) = f_2(x_2) = l$이라는 조건을 만족해야한다. 이는 우리는 ‘compatibility’라고 정의한다. 즉 특정 instance가 존재할 때, 이를 $f_1(x_1)$에 넣든 $f_2(x_2)$에 넣든 동일한 결과를 보여줘야한다는 의미이다.

Concept class - Wikipedia

In computational learning theory in mathematics, a concept over a domain X is a total Boolean function over X. A concept class is a class of concepts. Concept classes are a subject of computational learning theory.

𝕎 https://en.wikipedia.org/wiki/Concept_class

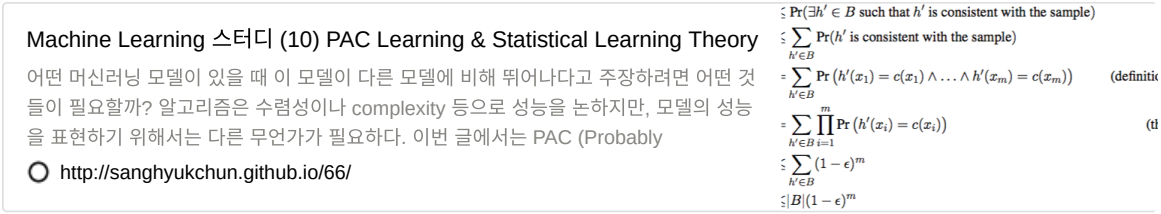
1) compatibility

필자는 앞으로 취급할 labeled data는 concept class들에 대해서 $f(x) = f_1(x_1) = f_2(x_2) = l$ 조건을 만족하는 noise를 최대한 줄인 데이터라고 정의하였다.

우리는 위에서 만들어진 작은 크기의 labeled 데이터로부터 unlabeled data의 label를 예측할 수 있는데, 이게 과연 도움이 될까?라는 근본적인 의문이 들 수 있다.

▼ computatinal learning theory: PAC learning

이를 이해하기전에, computational learning theory의 분야 중 하나인 PAC learning이라는 개념을 잠깐 짚고 넘어갈 필요가 있다.



우리는 학습에 필요한 데이터가 적거나, 모델의 정확도는 얼마나 높은지 등에 따라 이 모델의 "뛰어남"을 측정할 수 있다. 이러한 뛰어남을 연구하는 분야가 바로 computational learning(성능뿐만 아니라 비용과 시간도 따지겠다는 말이다.) 분야에 해당된다. PAC는 이 computational learning의 한 분야에 속한다.

PAC learning이란 "높은 확률로" 주어진 모델이 "작은 error를 가진다."와 같은 분석을 진행한다.

PAC bound는 다음과 같다.

$$Pr[error_{true}(h) \leq error_{train}(h) + \varepsilon] \leq |H|exp(-2m\epsilon^2)$$

$error_{true}(h) \leq error_{train}(h) + \varepsilon$, 즉 train error가 실제 모델의 error보다 클 확률이 $|H|exp(-2m\epsilon^2)$보다 크면 안된다는 것이고, 이때 $|H|$는 Hypothesis space의 complexity를 의미하고, m은 training example의 개수를 의미한다.

즉,

- Training data의 개수가 많으면 많을수록 (이 클수록) 모델은 더 generalize되고, overfitting의 위험성도 작아진다.
- Model complexity가 높을수록 (가 클수록) 모델 generalization은 더 어려워지고, overfitting의 위험성 역시 커진다.
- Model의 성능을 끌어올리려면 (true error를 줄이려면) train error를 줄이거나, training sample을 늘리거나, 모델 complexity를 줄여야한다.

는 의미를 지니고 있다.

물론 이 bound는 $h(x)$, hypothesis space가 유한할 때만 쓸 수 있는 bound에 해당된다. 당장 $f(x) = ax + b$를 hypothesis space로 가정하면 해당 모델을 무한개가 될 수 있기 때문이다. 그래서 VC dimension이라는 개념이 도입되었다.

- Instance X는 모든 데이터의 공간이라 할 수 있다. 즉, 모든 픽셀이 0 또는 1인 28 by 28 흑백 이미지라고 한다면 2784278 크기의 set이 될 것이다.

Instance X에 대해 $VC(H)$는 H로 shatter할 수 있는 가장 큰 X의 finite subset 크기와 같다. 이때 shatter란 labeling에 상관없이 정확하게 classify하는 것을 의미한다. **(어떤 상황에서, 최대 몇개의 데이터까지를 오차없이 무조건 분류해낼 수 있는지를 의미한다. 이 크기가 커질수록 우리는 굉장히 복잡한 모델이라고도 생각할 수 있다. 과적합의 위험성이 존재하기 때문이다.)**

- 먼저 linear model은 d차원에서 VC dimension이 d + 1 이다.
- KNN에서 K가 1인, 즉 1-NN의 VC dimension은 무한대이다.
- k level의 decision tree는 총 2^k개의 leaf node를 사용해 서로 다른 2^k개 sample을 분류할 수 있으므로 VC dimension은 2^K 이다.

이 경우에서의 PAC bound는 하단과 같이 변한다.

$$\text{error}_{true}(h) < \text{error}_{train}(h) + \sqrt{\frac{VC(H)(\ln \frac{2m}{VC(H)} + 1) + \ln \frac{4}{\delta}}{m}}.$$

실제 svm도 이러한 error값을 줄이는 방향으로 학습이 된다고 한다.

PAC learning의 핵심 개념은 위에서 자세하게 보는 것으로 하고, 간단하게만 설명하면, PAC learning은 특정 데이터의 수로, 그리고 모델의 복잡성이 적은 방향으로 error가 작은 모델을 성능이 "뛰어난"모델이라고 정의한다. 즉, 작은 크기의 labeled 데이터가 오류가 없는 데이터라면 우리는 PAC learning 관점에서 더 뛰어난 모델을 찾았다고 할 수 있을 것이다. 그리고 co-training은 이러한 데이터들만 사용하는 것이 핵심이다.

아무튼, 다시 본론으로 돌아가자면, 우리는 위와 같은 점들때문에 C_1과 C_2가 굉장히 복잡한 함수들을 포함함에도 불구하고, distribution D(concept class들에 대해서 error가 존재하지 않는, noise를 최대한 줄인 데이터) 에 대해서는 모델의 복잡성이나 사이즈가 더 간단해지고 작아질 수 있다는 장점이 있다.

2) independency

추가로, X_1 과 X_2 데이터로 co-training을 진행할 때 이 둘은 독립적인 관계를 띄어야 함을 알 수 있다. 예를 들어보겠다. $X_1 = X_2 = \{0,1\}^n$이고 $C_1 = C_2 =$conjunction over $\{0,1\}^n$ (0과 1로 이루어진 데이터들간의 "AND" 함수)이라고 가정해보자.

함수가 conjuction이라는 개념이 와닿지 않을 수 있어서 간단한 예시를 들고 왔다. 만약 $f_1(x)$**=** x_1 **And** x_k **And** x_n **And..**라고 가정해보자. 이때 $x_1 = 0$이라면, $f_1(x)$에 어떤 값이 오든 x_1 **And** x_k **And** x_n **And..** **은** 0이므로 결과값도 0이 등장하게 된다.

위에서 보았던 것처럼 X_1이 0일 때는 $f_1(x_1) = 0$은 자명한 사실이다. 여러 항들 중 하나만 0이라도 AND에 대한 결과값은 0이기 때문이다. 이때, $f_1(x_1) = 0$이므로, 해당 데이터값은 $f_2(x_2)$=0에 대한 학습값으로도 사용가능하다. 하지만 만약 D가 $x_1 = x_2$라는 , x_1으로부터 x_2를 추정할 수 있는 가정을 지닌 분포라면 이는 f_2의 학습에 대해 별로 도움이 되지않는 데이터에 해당된다. 왜냐하면, 모델을 잘 학습시키기 위해서는 random하고 새로운 데이터를 계속 입력하여 학습시켜주는 것이 좋은데, 와같은 데이터는 결국 f_2에 대해 완전히 random한 데이터라고 보기 힘들기 때문에 좋은 분포라고 볼 수 없다. ($(x1, x2)$ 데이터 분포에 $x_1 = x_2$라는 조건이 애초에 존재했다면, $x_2 = 0$일 때 $f_2(x_2) = 0$이라는 것이 당연하기 때문) 반대의 상황을 보겠다.

그런데 그러한 가정($x_1 = x_2$라는 등의) 없이 $x_1 = 0$일 때 $f_1(x_1) = 0$이면 이때의 x_2의 값은 완전히 random한 값이기 때문에 우리는 $f(x_2 = ?) = 0$에 대한 새로운 관계를 찾아낼 수 있기 때문이다.

결론적으로, x_1과 x_2가 독립적인 성질을 띄어야 한다는 것이다. 독립적이라는 상황 아래서는 x_1의데이터가 충분히 f_2 학습에 도움이 되는 데이터라는 뜻이다.

더 나아가, D가 independent하고 target class들이 random classification noise를 가진 경우에만 co-training이 의미가 있는데 이러한 개념은 session 5에서 수식적으로 더 다뤄볼 예정이다.

▼ 기타 참고용 3. A HIGH LEVEL VIEW AND RELATION TO OTHER APPROACHES

일단 해당 논문에서는 data distribution와 concept 사이의 compatibility의 개념을 담으려고 하였다. 그리고 주어진 분포안에서는 target concept들이 compatible하여야 함을 말하고, 이것은 unlabeled data가 class C를 더 작은 C'으로 줄일 수 있음을 말한다. 무슨 말이냐, 고 하면 unlabeled data로부터 형성된

concept class C'은 기존의 distribution D로 부터 파생된 class C와 unlabeled data도 포함된 distribution D로 부터 파생된 class C_D와의 교집합이기 때문에 더 작은 집합에 해당된다. 물론, 이러한 compatiibility 정의말고, 다른 류의 compatibility 정의도 존재한다.

필자는 필자의 co-training과 다른 사람들의 model(label과 unlabeled data를 합치는 모델)을 비교해보았다.

첫번째 접근법은 바로, 데이터가 간단한 parametric model로부터 만들어졌다고 가정하는 것이다. 이러한 form 아래, 베이지안 최적화 모델로부터 labeled data와 unlabeled data의 적절한 양을 매길 수 있다. Missing information을 가진 data로부터 학습을 하는 EM 알고리즘이 이러한 setting을 갖고 있다. 예를 들어, 가장 일반적인 가정은 positive example(결과값이 1인 데이터)들은 n차원의 가우시안 분포 D_+ 에서 θ_+를 중심으로 생성이 되었고, negative example(결과값이 0인 데이터)들은 n차원의 가우시안 분포 D_- 에서 θ_-를 중심으로 생성이 되었다. (여기서 θ_+와 θ_-는 모델에 알려지지 않은 것들에 해당된다.)이러한 데이터들은 D_+로부터 positive point를 고르거나, D_- 로부터 negative point를 고르면서 생성된다. 이러한 케이스에서 베이즈 최적화 모델은 선분 $\theta_+\theta_-$과 수직이고 이등분하는 hyperplane에 해당된다.

5. PAC LEARNING IN LARGE INPUT SPACES

Distribution D에서 조건부 독립성 가정이 주어졌을 때, 만약 target class가 random noise로 학습이 가능하다고 하다면, 약한 예측기가 co-training을 통하여 boosted될 수 있다고 말할 수 있다. 이때, target function f_1과 f_2, 그리고 distribution가 동시에 conditional independence, 조건부 독립을 만족시키게 하는 조건을 볼 것이다. 이는

$$Pr[x_1 = \hat{x_1}|x_2 = \hat{x_2}] = Pr[x_1 = \hat{x_1}|f_2(x_2) = f_2(\hat{x_2})]$$

$$Pr[x_2 = \hat{x_2}|x_1 = \hat{x_1}] = Pr[x_2 = \hat{x_2}|f_1(x_1) = f_1(\hat{x_1})]$$

이 의미는 바로, x_1과 x_2가 라벨에 대해 서로 독립적이라는 말이다. 예를 들어, 웹페이지 P를 구분하는데 있어서 page안의 word와 hyperlink는 서로 독립적인 관계라는 말이다.

"약한 예측기가 co-training을 통하여 boosted될 수 있다고 말할 수 있다"라는 theorem1에 대해 논의하기 위해서 우선은 function f의 'weakly-useful predictor' h를 하단과 같이 정의하였다.

$$1. \quad Pr_D[h(x) = 1] \geq \epsilon$$

$$2. \quad Pr_D[f(x) = 1|h(x) = 1] \geq Pr_D[f(x) = 1] + \epsilon$$

예를 들어 어떤 웹 페이지의 "handout"이라는 단어가 웹페이지에 등장한다는 weakly useful predictor h가 될 수 있다고 가정해보자.

1번 식은 "handout"이라는 단어가 그 웹페이지에서 무시할 수 없는 부분이라는 조건이고, 2번 식은 "handout"이라는 단어가 그 웹 페이지에 나타났을 때 그 웹페이지가 강의 홈페이지일 확률이 "handout"이라는 단어가 나타나지 않았을 때의 확률보다 높다는 조건이다. 이러한 조건들로 weakly predictor h를 정의하였고, 정리하자면 *h* 를 이용하면 그냥 0 또는 1로 아무거나 찍는 것보다 현저하게 나은 결과가 나온다.

이제 Theorem 1을 봐보자.

> 만약 C_2가 PAC 모델에서 classification noise와 함께 learnable하다면, 그리고 조건부 독립성 가정이 만족한다면, C_1과 C_2는 weakly-useful predictor인 $h(x_1)$으로부터 시작된 co-training model안에서 학습이 가능해진다.

정리하자면, 조건부 독립성 가정을 만족하고, classification noise와 함께 learnable한 concept class들은 unlabeled data와 weakly useful predictor로 학습이 가능하다는 말이다.

이 theorem을 증명하기 위해 알아야하는 개념과 부명제에 대해서 새로 짚고 넘어가겠다. (α, β) classification noise가 존재한다고 해보자. 이때 α는 positive으로 분류된 데이터들의 오류, β는 negative으로 분류된 데이터들의 오류에 해당된다.

본격적으로 theoreom 1을 증명해나가기 위해서는 우리는 하단의 부명제를 이용해야 한다.

Lemma 1은 다음과 같다.

> concept class C(목적 함수들의 집합)가 standard classification noise model에서 학습이 가능하다면 C는 $\alpha + \beta < 1$일 때 (α, β) ⇒ classification noise로 학습이 가능하다.

본격적으로 Theorem 1을 증명하기에 앞서 $f(x)$를 목적 함수(target function), $p = Pr_D[f(x) = 1]$ 는 랜덤하게 추출한 D에서 positive의 확률, $q = Pr_D(f(x) = 1|h(x_1) = 1), c = Pr_D(h(x_1) = 1)$라고 정의한다. 이제 p, q, c를 이용해서 α, β를 표현해본 후, 그렇게 표현된 $\alpha+\beta$가 1을 넘지 않는지를 확인할 것이다. 만약 1을 넘지 않는다면 C_2는 classification noise로 학습이 가능하다는 뜻이고, 그렇다면 theoreom 1에 의해 $h(x_1)$으로 cotraining이 가능해진다는 말이다.

$$Pr_D[h(x_1) = 1|f(x) = 1] = \frac{Pr_D[f(x) = 1|h(x_1) = 1]Pr_D[h(x_1) = 1]}{Pr_D[f(x) = 1]} = \frac{qc}{p}$$

이 식은 실제 class가 1일 때 가설이 1로 예측할 확률을 베이지안 확률 정의로 표현한 것이고, 앞서 정의한 p, q, c로 나타내었다.

$$Pr_D[h(x_1) = 1|f(x) = 0] = \frac{(1 - q)c}{1 - q}$$

위의 식은 실제 class가 0일 때 x_1으로 예측한 가설이 1로 잘못 예측할 확률을 나타내며, 앞서 정의한 p, q, c로 나타내었다.

조건부 독립 가정에 의해 random example $x = (x_1, x_2)$에 대해 $h(x_1)$ (첫 번째 view의 instance x_1 로 만든 predictor)는 x_2와 독립이다. “handout” 예시에서 말한 것처럼 “handout”이라는 단어로 예측한 웹 페이지의 class는 그 웹 페이지의 하이퍼링크, 하이퍼링크의 라벨과는 독립이다. 그렇기 때문에 만약 $h(x_1)$ 을 x_2 의 잘

못 labeling된 label로 사용한다면, 이것은 (α, β) -classification noise와 같아진다. 여기서 $\alpha = 1 - \frac{qc}{p}$이고, $\beta = \frac{(1-q)c}{(1-p)}$이다. 이것을 이용하면

$$\alpha + \beta = 1 - \frac{qc}{p} + \frac{(1-q)c}{(1-p)}$$

식을 이렇게 쓸 수 있다. 앞에서 언급했던, weakly-useful predictor *h*를 정의하기 위해 썼던 가정들을 p, q, c로 나타내면 $Pr_D[h(x) = 1] \geq \epsilon$ 이고 $c = Pr_D(h(x_1) = 1)$이므로 $c \geq \epsilon$로 표현할 수 있다. 마찬가지로 $Pr_D[f(x) = 1|h(x) = 1] \geq Pr_D[f(x) = 1] + \epsilon$이므로 이항하면 $p - q \geq \epsilon$로 나타낼 수 있다. 그러니까 $c = \epsilon, p - q = \epsilon$라고 하면 $\alpha + \beta$의 값은 최대 $1 - \frac{\epsilon^2}{p(1-p)}$가 되며, 숫자로 치면 $1 - 4\epsilon^2$ 이다. (*f*가 unbiased 하여p = 1/2 라고 생각하면) 따라서 $\alpha + \beta < 1$이라는 Lemma 1에 적용하여 조건부 독립 가정이 성립한다면 weakly useful predictor인 $h(x_1)$ 이 주어졌을 때 (C_1, C_2) 가 unlabeled data만 있는 상황에서 Co-training model 학습이 가능하다는 Theorem1을 확인할 수 있다.

5.1 relaxing this assumptions

5.1절에서는, 조건부 독립성 가정은 그대로 유지하되, distribution D에 적용되었던, 목적함수 $f(x_1, x_2)$에 대해 $f_1(x_1) \neq f_2(x_2)$를 만족하는 (x_1, x_2)도 존재한다고 보고, weakly useful predictor를 unlabeled data로 boost할 수 있음을 보여준다. 증명과정은 하단에 첨부해놓겠다.

6) Experiments

웹 페이지 분류 실험의 데이터는 4개 대학교에서 수집한 1051개의 Computer Science Department 웹 사이트. X_1은 웹 페이지 컨텐츠를, X_2는 hyperlink를 나타낸다. x_1과 x_2 각각에 대한 분류기는 나이브 베이즈 알고리즘을 사용하여 학습되며, 이를 각각 page-based classifier 와 hyperlink-based classifier라 부른다. (나이브 베이즈 알고리즘은 텍스트 분류기에 유용한 알고리즘으로 사용되어진다.)

L은 labeled data set, U은 unlabeled data set을 나타내고, $U^{'}$은 u 개의 unlabeled data set 을 의미한다. 알고리즘의 진행 과정은 다음과 같다.(u=75)

1) L 은 두 개의 분류기 h_1과 h_2를 학습하는데 사용된다.

이때 h_1은 x_1 만을 가지고 학습한 것이고, h_2도 마찬가지이다.

2) 각 분류 모델은 unlabeled data set인 $U^{'}$의 label을 예측하게 되는데, positive로 간주된 가장 confident한 example과 negative로 간주된 confident exmaple을 선택한다.

이 실험에서는 $p = 1, n = 3$이다.

3) 이렇게 각 분류기에서 선택된 p개의 positive한 example과 n개의 negative한 example들이 다시 labeled data set인 L에 더해지며, 이때 U에 속한 2p + 2n 개의 example들이 랜덤으로 $U^{'}$에 더해지게 된다.

4) $p + n$ 개의 데이터가 L로 옮겨진다.

5) 위 과정을 k 번 반복한다.(k=30)

Given:

- a set L of labeled training examples
- a set U of unlabeled examples

Create a pool U' of examples by choosing u examples at random from U
Loop for k iterations:

Use L to train a classifier h_1 that considers only the x_1 portion of x
Use L to train a classifier h_2 that considers only the x_2 portion of x
Allow h_1 to label p positive and n negative examples from U'
Allow h_2 to label p positive and n negative examples from U'
Add these self-labeled examples to L
Randomly choose $2p + 2n$ examples from U to replenish U'

co-training과의 결과를 비교하기 위해 각 분류기를 supervised training을 진행해보았고, 두 분류기를 combined한 combined classifier를 생성해봤다.

test error는 초기 데이터를 랜덤하게 분할하여 실시한 error의 평균값이며, 그 결과는 다음과 같다.

	Page-based classifier	Hyperlink-based classifier	Combined classifier
Supervised training	12.9	12.4	11.1
Co-training	6.2	11.6	5.0

전체적으로 co-training의 test error rate이 더 작았으며, 심지어 co-training의 test error rate은 supervised training의 test error rate의 절반 수준에 해당하기도 한다. text data와 hyperlink data를 구분지어 두 개의 분류기를 사용했을 때보다, 두 데이터를 섞어 생성한 combined classifier를 사용했을 때의 성능이 더 좋은 것을 확인할 수 있다.

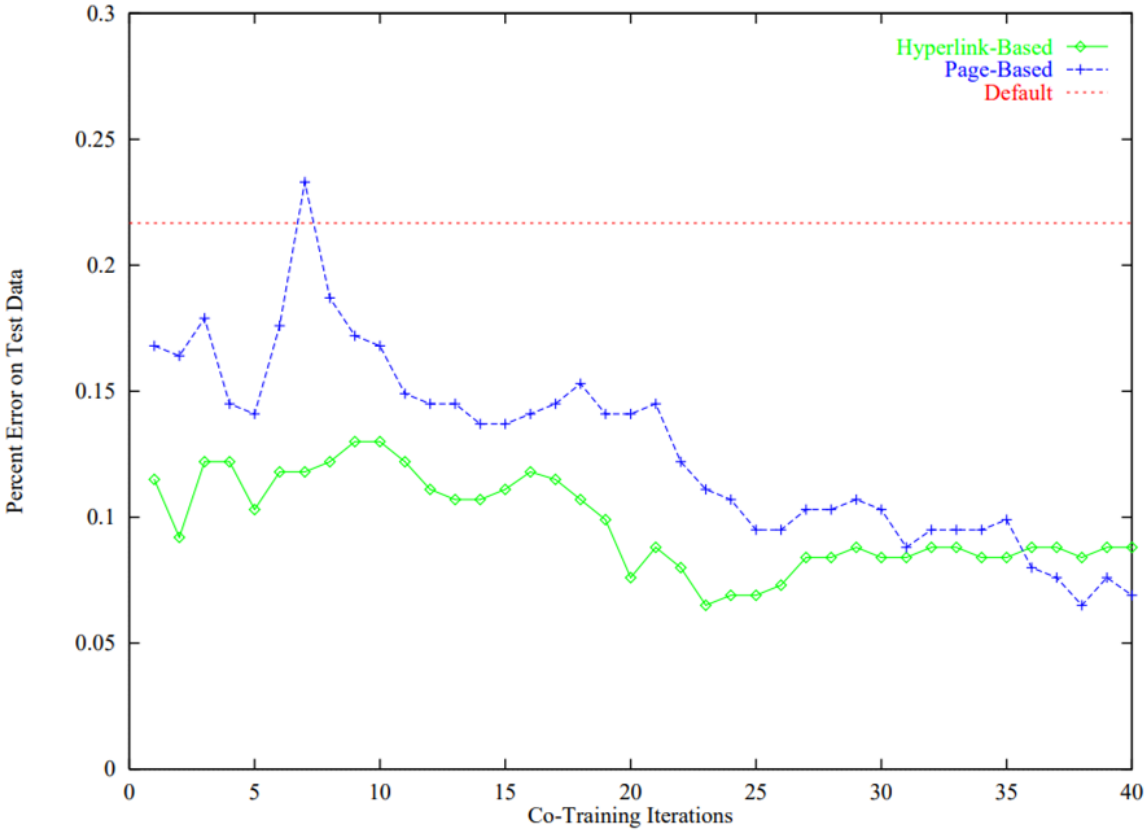


Figure 2: Error versus number of iterations for one run of co-training experiment.

7) Conclusions and Open Questions

co-training은 서로 상관관계가 전혀 없는 두 속성 집합을 이용하여 unlabeled data를 labeled data로 포함시켜, labeled data를 대용량으로 얻는데 드는 비용을 줄여 모델의 효용성을 제고하는 방법론이다. 단 이 방법론은 현실 문제를 지나치게 단순화시킨 면이 있으며, 상관관계가 전혀 없는 두 속성 집합이 구성 가능한 문제가 실제로 많지 않다는 단점이 있다. 이러한 이유로 완전하게 상호 배타적이지 않더라도, 두 속성의 상관관계가 미미하다면 co-training을 사용하기도 한다.