Learning From Data: 통계 학습 이론 정리

jongman@gmail.com

January 30, 2015

Abstract

이 노트는 통계 학습 이론에 관한 학부 수준의 좋은 교과서인 Learning From Data(Abu-Mostafa 외)에서 통계 학습 이론 관련된 내용만을 발췌 요약한 것이다. 개인적인 이해를 돕고 나중에 참고할 목적으로 쓴 노트이기 때문에, 틀린 내용이나 지나친 단순화가 많을 수 있다.

Contents

1 학습은 가능한가?

1.1 학습은 불가능하다!

기계 학습 알고리즘이 임의의 자료가 주어질 때 그 자료로부터 무언가를 배울 수 있을까? In-sample 자료를 살펴보면 그 밖의 자료에 대해서도 무언가를 알 수 있다고 우리가 수학적으로 보장할 수 있을까? 어떤 가정도 없이 자료로부터 무언가를 배우는 것은 불가능하다는 것을 다음 주장을 통해 알 수 있다.

크기 3의 진리값 벡터를 입력으로 갖는 이진 분류(binary classification) 문제가 있다고 하자. 따라서 8가지 가능한 입력이 있고, 존재 가능한 정답 함수는 2^8개가 있다. 만약 8가지 가능한 입력 중 5개가 학습용으로 주어졌다고 하자. 그러면 가능한 정답 함수 2^8개 중 2^3개는 이 5개의 예제 입력에 대한 정답을 모두 만족할 것이다. 과연 이 중 무엇이 정답에 더 가까울지 골라낼 수 있을까? 주어진 자료 하에서는 이 함수들은 구분할 수 없다. 따라서 우리는 아무것도 배울 수 없다!

과연 현실이 이렇게 우울할까? Out-of-sample 자료에 대해서는 우리는 어떠한 보장도 할 수 없을까?

1.2 확률적 접근

이와 같은 문제를 해결하기 위해 통계 학습 이론이 사용하는 중요한 가정은 in-sample 입력은 가능한 모든 입력 중에서 임의로 선택되었다는 것이다. 이와 같은 가정을 하면 우리는 이제 우리가 보지 못한 자료에 관해서도 알 수 있게 된다.

간단한 예제를 들어 보자. 커다란 상자에 엄청나게 많은 구슬이 들어 있는데, 이중 일부는 붉은색, 나머지는 초록색이라고 하자. 전체 구슬 중 붉은 색의 비율은 $\mu \times 100\%$ 이다. 이 상자에서 임의로 구슬을 N개 뽑았는데, 그 중 붉은 구슬의 비율은 $\nu \times 100\%$였다. 이 때 ν를 보면 μ에 대해 알 수 있는 것이 있을까? 물론이다. 만약 1000개의 구슬을 뽑았는데 붉은 구슬이 그 중 990개였다고 하자. 우리는 당연히 상자에는 붉은 구슬이 훨씬 많이 들어 있을 것이라고 예상할 수 있다. 이 직관을 어떻게 수학적으로 나타낼 수 있을까? 확률론의 힘을 빌리면 가능하다. 우리가 뽑은 구슬은 모두 임의로 뽑았기 때문에, 이것을 베르누이 확률 변수로 나타내자. 그러면 변수들의 관찰 평균값(ν)이 기대치(μ)에서 벗어날 확률을 계산하는 정리인 Hoeffding 부등식을 사용할 수 있는데, 다음과 같은 형태를 갖는다:

$$\mathrm{P}\left[|\nu - \mu| > \epsilon\right] \leq 2e^{-2\epsilon^2 N}$$

여기서 ϵ는 우리가 정하는 값으로, ν와 μ 사이의 오류를 나타낸다. 이 오류가 정해지면, Hoeffding 부등식을 이용해 우리가 확인한 ν가 참값인 μ에서 ϵ 이상 벗어날 확률을 ϵ과 N의 함수로 나타낼 수 있게 된다. 따라서 샘플의 크기가 커지면, 부등식의 우변이 줄어들기 때문에 ν가 μ에 점점 가까워지게 됨을 알 수 있다.

이 예와 학습 문제 사이에는 어떤 관련이 있을까? 우리가 어떤 형태의 가설 h를 세웠다고 하자. 이 가설의 형태는 중요하지 않다. 이 가설을 가능한 모든 데이터에 대해 적용하면, out-of-sample 오류율 $E_{out}(h)$를 얻게 될 것이다. 물론

모든 데이터를 얻는 것은 불가능하므로, 이 데이터 중 임의로 뽑은 in-sample 데이터에 대해 가설을 적용해 보고 오류율 $E_{in}(h)$를 계산해 보자. 샘플의 크기가 늘어나면 늘어날 수록 $E_{in}(h)$가 실제 오류율 $E_{out}(h)$에 가까워질 것이라는 신뢰를 가질 수 있게 된다. 따라서 이를 이용해, 우리가 아직 보지 못한 자료에 관해서 다음과 같이 말할 수 있게 된다:

> 우리가 정한 이 가설 h의 오류율이 이 범위를 벗어날 확률은 얼마이다.

여기에서 가설은 우리가 알고 있는 것(in-sample 오류율)을 우리가 모르는 것(out-sample 오류율)으로 전달하는 역할을 한다. 우리는 목적 함수에 대해 안다고 말할 수 없으며, 우리가 가진 가설의 오류율에 대해서만 이야기할 수 있다.

마지막으로:

- 교과서의 논의는 분류에 대해서만 다뤘지만, 이와 같은 형태의 증명은 회귀 분석과 같은 다른 지도 학습(supervised learning)에도 적용될 수 있다(고 한다).
- Hoeffding 부등식은, 당연하게도, in-sample 데이터가 임의로 선택되었을 때만 적용된다. 모든 통계 학습 이론은 이 가정에 기반을 두고 있다.

1.3 다수의 가설과 합계 상한(Union Bound)

모든 통계 학습 문제는 여러 개의 가설 $\{h_i\}$중에서 가장 그럴 듯한 것을 선택하는 과정이다. (애초에 가설이 하나밖에 없다면, 학습이 아니라 검증에 불과할 것이다.) 이 때 어떤 가설을 고르는 것이 좋을까? 당연히 in-sample 오류율이 가장 낮은 가설을 택하는 것이 좋을 것이다. 이 가설을 g라고 부르자. 그러면 g의 성능, 다시 말해 $E_{out}(g)$에 대해 무슨 말을 할 수 있을까? Hoeffiding 부등식이 여기에도 적용될까? 안타깝게도, 그렇지 않다.

여기서 문제는 g는 우리가 in-sample 데이터를 본 후에야 결정된다는 것이다. 물론 h들의 집합을 알고 있으니, 이 중의 하나가 g가 된다는 것은 알지만, 이 중에 무엇일지 알 수가 없다. 따라서 $E_{out}(g)$의 분포는 $E_{out}(h)$의 분포와 달라지게 된다. 이 문제에 대한 적절한 설명을 하기 어려운데, 간단한 예만 들도록 하겠다. 어느 나라의 고3 학생들의 키를 모두 모아 보면 정규 분포를 따른다고 하자. 임의의 학생을 하나 고르면, 이 학생의 키는 해당 정규 분포를 따르게 된다. 그런데 각 학교마다 가장 큰 학생들을 모아 놓고, 이 중에서 임의의 학생을 뽑으면 어떨까? 원래의 정규 분포와는 당연히 다를 것이다.

이런 이유로, Hoeffding 부등식을 바로 적용하면 오류율을 엄청나게 낮춰잡는 결과를 가져오게 된다. 그러면 어떻게 할까? 이제는 확률 $\mathrm{P}[|E_{in}(g) - E_{out}(g)| > \epsilon]$에 대해 어떤 보장도 할 수 없을까? 무식한 방법 하나는 최종 가설 g는 항상 우리의 가설 집합 $\{h_i\}$중에 있다는 것을 이용하는 것이다. 따라서 만약 g의 오류율이 ϵ을 넘는다면, $\{h_i\}$중 최소한 하나는 ϵ을 넘는 오류율을 가져야 할 것이다. 사건 A가 B의 충분조건일 때, $\mathrm{P}(A) \leq \mathrm{P}(B)$ 이므로, 다음과 같은 상한을 얻을 수 있다.

$$\begin{aligned}
\mathrm{P}[|E_{in}(g) - E_{out}(g)| > \epsilon] &\leq \mathrm{P}[\cup_{i=1\sim M} |E_{in}(h_i) - E_{out}(h_i)| > \epsilon] \\
&\leq \sum_{i=1}^{M} \mathrm{P}[|E_{in}(h_i) - E_{out}(h_i)| > \epsilon] \\
&\leq 2Me^{-2\epsilon^2 N}
\end{aligned}$$

이 상한을 합계 상한(union bound)라고 부른다. 이 수식에는 이제 가능한 가설의 수인 M이 포함되어 있다는 것에 주목해 보자. 상한이 나온 것은 좋은데, 대부분의 기계 학습 알고리즘에는 무한한 수의 가설이 있다. 실수 인자(parameter)

하나만 있다고 하더라도 무한한 수의 가설을 갖게 되기 때문이다. 이 경우, 합계 상한은 무한대가 되기 때문에 아무런 의미가 없다. 뒤에서 우리는 VC-차원(VC-dimension)이라 부르는 값을 소개하는데, 이를 이용하면 유한한 확률 상한을 구할 수 있다.

1.4 학습 문제의 두 측면

기계 학습의 궁극적 목표는 out-of-sample 오류율 0을 달성하는 것이다. 그러나 Hoeffding 부등식은 단지 in-sample 오류율 E_{in}과 out-sample 오류율을 연결해 줄 뿐이다. 따라서 out-of-sample 오류율을 낮추는 일을 다음과 같이 두 단계로 나눌 수 있다.

1. E_{in}을 충분히 작게 만들기
2. E_{out}과 E_{in}을 충분히 가깝게 만들기

이렇게 문제를 쪼개 보면 많은 통찰을 얻을 수 있다. 그 한 예로 가설 집합의 복잡도가 갖는 트레이드오프에 대해 이해할 수 있게 된다. 아주 복잡하고 강력하며, 크기 M도 큰 가설 집합을 사용하면 E_{in}을 원하는 만큼 줄일 수 있을 것이다. 그러나 가설 집합의 크기가 크므로, 결과적으로 E_{out}이 E_{in}에 가까울 확률을 떨어뜨린다. 또 다른 예는 복잡한 목적 함수이다. 목적 함수는 문제 2를 어렵게 하진 않지만, 문제 1을 풀기 어렵게 만든다. 문제 1을 풀기 위해서 더 복잡한 가설 집합을 사용하면, 결과적으로 문제 2를 풀기 어려워지게 된다.

2 훈련과 테스트

이 챕터에서는 가설의 일반화(generalization) 속성, 즉 훈련 데이터에서 배운 것을 어떻게 테스트 데이터에 적용할 수 있는가를 나타낼 수 있는 여러 도구들에 대해 소개하고, 직관적인 이해를 돕는다.

2.1 합계 상한 고치기

최종 가설 g에 Hoeffding 부등식을 적용할 수 없었던 것은, 우리가 데이터를 본 후에야 g가 정해지기 때문이었다. 이 문제를 해결하기 위해, 합계 상한은 사건 $\mathcal{B}_i$를 i번 가설의 out-sample 오류율이 예상에서 크게 벗어나는 사건 $|E_{in}(h_i) - E_{out}(h_i) > \epsilon|$으로 둔 뒤, 다음과 같은 상한을 이용한다.

$$\mathrm{P}[\mathcal{B}_1 \vee \cdots \vee \mathcal{B}_M] \leq \sum_i \mathrm{P}[\mathcal{B}_i]$$

이 상한의 문제는 무엇일까? $\mathcal{B}_i$ 들의 사건 공간이 대개 겹친다면 우변의 상한이 좌변에 비해 너무 커져 버리고, 제대로 된 상한으로써의 의미를 잃게 된다. 그런데 이것은 많은 현실 세계의 문제에 대해서도 성립한다. 선형 회귀를 생각해 보면, 계수가 아주아주 조금 다른 수많은 다른 가설들이 있지만, 실질적으로 이들 간의 차이는 거의 없다.

2.2 증가 함수 (The Growth Function)

2.2.1 직관적 이해

교과서에서 다루는 분석은 오직 이진 분류 문제만을 다루고 있다. 강의에서는 이와 같은 방식의 접근으로 회귀 분석도 다룰 수 있다고 언급하는데, 논증 과정이 쓸데없이 복잡해 다루지 않는다고 한다.

우리는 이 절에서 증가 함수를 정의하는데, 이것은 합계 상한에서 가설의 수 M을 대체하게 될 함수로 가설 집합의 속성에 따라 변화한다. 그 정의는 약간 기계적이지만, 직관적으로는 위에서 말한 대로 서로 다른 가설들이라도 실제로 분류 결과가 다르지 않다면 의미가 없다는 점을 이용한다. 간단한 예를 들어 설명하자. 실수 집합 위의 점들을 분류하려 하는데, 우리의 가설들은 역치 a에 대해 다음 형태라고 하자:

$$h_a(x) = \begin{cases} 1 & \text{if } x \geq a \\ -1 & \text{otherwise} \end{cases}$$

가능한 역치는 무한히 많으므로, 우리가 사용 가능한 가설도 무한히 많다. 그러나 임의의 훈련 데이터가 주어질 때, 이 중 엄청나게 많은 수는 서로 다를 것이 없다. 훈련 데이터의 $\mathbf{x}$가 $\{1, 4, 6, 8\}$라 하면, $a = 5$와 $a = 5.00001$ 은 분명 다르지만 결과적으로 각 입력들을 똑같이 분류하기 때문에, 훈련 데이터 입장에서는 서로 다를 것이 없다.

증가 함수는 서로 유의미하게 다른 가설들의 수를 센다. 따라서 직관적으로는 우리가 가설의 수 M을 증가 함수 $m_{\mathcal{H}}(N)$로 바꾸는 것이 아주 말 되는 전략으로 보인다.

2.2.2 기술적 정의

특정 형태의 가설 집합이 있다고 하자. 크기가 N인 훈련 데이터를 만들면, 각 점을 이진 분류할 때 가능한 결과값은 2^N가지가 있다. 훈련 데이터는 임의로 만들 수 있다고 가정할 때, 2^N개의 결과값 중 이 가설 집합으로 최대 몇 개를 표현할 수 있을까? 증가 함수는 이 값을 나타낸다. 기술적으로는 다음과 같이 쓸 수 있을 것이다.

$$m_{\mathcal{H}}(N) = \max_{\mathbf{x}_1, \cdots, \mathbf{x}_N} |\mathcal{H}(\mathbf{x}_1, \cdots, \mathbf{x}_N)|$$

이 때 $\mathcal{H}()$는 주어진 훈련 데이터 $\{\mathbf{x}_i\}$에 대해, $\mathcal{H}$ 의 모든 가설들을 이용해 만들 수 있는 결과값들의 수를 나타낸다. 위에서 설명한 1차원 분류 문제와 가설 집합을 가져오면, $m_{\mathcal{H}}(N) = N + 1$이 될 것이다.

2.2.3 증가 함수 계산하기

당연하게도 서로 다른 가설 집합은 서로 다른 증가 함수를 가지며, 이들을 직접 계산하기란 아주 까다롭다. 따라서 증가 함수의 상한을 계산해 이를 이용하도록 한다. 이 때 유용하게 사용되는 것이 부순다는 개념이다. 크기 N인 입력이 주어질 때, 어떤 가설 집합 $\mathcal{H}$를 이용해 이 입력에 대한 어떤 출력도 만들 수 있다면(다시 말해 2^N가지의 결과값을 모두 만들 수 있다면) 이 가설 집합이 이 입력을 부순다고 말한다. 당연하지만 N이 증가할 수록 입력을 부수기는 어려워진다. 크기가 k인 어떤 입력도 부술 수 없는 최초의 k를 $\mathcal{H}$의 실패점(breaking point)이라고 부른다. 예를 들어, 2D 평면의 퍼셉트론은 입력이 3개 주어지면 그 중 일부 형태는 부술 수 있지만 (삼각형은 부술 수 있지만, 일직선 상의 세 점은 부술 수 없다), 입력이 4개 주어지면 절대로 부술 수 없다. 따라서 퍼셉트론의 실패점은 4이다.

실패점을 이용하면 증가 함수의 상한을 계산할 수 있다. 자세한 증명 과정은 $m_{\mathcal{H}}(N)$의 상한이 되는 새로운 함수 $B(N,k)$를 정의하고 이 함수에 대한 재귀식을 찾아 푸는 과정을 거치는데, 결과적으로 다음과 같은 결과를 얻을 수 있다.

$$m_{\mathcal{H}}(N) \leq \sum_{i=0}^{k-1} \binom{N}{i}$$

유의할 것은 이 식이 N의 $k-1$차 다항식이라는 것이다. 다항식!!!

2.3 VC 차원 (VC-dimension)

실패점은 주어진 가설 집합의 일반화 속성을 표현하는 중요한 숫자이므로, 좀더 멋진 이름으로 부르기로 한다. 바로 Vapnik-Chervonenkis 차원(VC-dimension)이 그것이다[1]. 가설 집합 $\mathcal{H}$의 VC-차원 d_{VC}는 실패점 - 1, 즉 $m_{\mathcal{H}}(N) = 2^N$ 인 최대의 N으로 정의된다. 따라서 위의 상한을 N의 d_{VC}차 다항식으로 표현할 수 있다.

$$m_{\mathcal{H}}(N) = O\left(N^{d_{VC}}\right)$$

이제 이 상한을 어떻게 이용해 Hoeffding 부등식을 고칠 수 있을까? 원래 우리가 원하던 것은 M 을 $m_{\mathcal{H}}(N)$으로 바꾸는 것였으니, 다음과 같이 쓸 수 있을까?

$$E_{out} \leq E_{in} + \sqrt{\frac{1}{2N} \ln \frac{2m_{\mathcal{H}}(N)}{\delta}}$$

이 상한은 기술적인 이유로 완전히 정확하진 않지만, 상수를 제외하면 대체적으로 맞는다. 정확한 상한은 아래와 같다:

VC Generalization Bound

주어진 오류 $\delta > 0$에 대해

$$E_{out}(g) \leq E_{in}(g) + \sqrt{\frac{8}{N} \ln \frac{4m_{\mathcal{H}}(N)}{\delta}}$$

일 확률은 $1-\delta$ 이상이다.

이 상한을 어떻게 해석해야 할까? $m_{\mathcal{H}}$가 N에 대한 다항식이라고 하자. 이것은 꽤 큰 수($\frac{1}{\delta}$)에 곱해지지만, 로그가 씌워지고 나면 작은 수가 된다. 결과적으로 N이 커지면 $\frac{8}{N}$이 훨씬 작아져서 뒤의 $O(k \lg N)$항을 압도하게 되고, E_{out}과 E_{in}이 가까워지게 된다. 이를 이용해 우리는 무한한 크기의 가설 집합에서도 N이 커지면 E_{in}이 E_{out}에 가까워진다고 말할 수 있게 된다!

2.3.1 현실적인 고려들

VC 차원을 이용하면 확률의 상한을 구할 수 있지만, 이 상한은 여러 개의 상한을 연결해 만들었기 때문에, 실용적인 의미는 크지 않다. 예를 들어 증가 함수 $m_{\mathcal{H}}(N) = N + 1$을 갖는 아주 간단한 증가 함수를 생각해 보자. 이 모델의

[1] Vapnik은 SVM의 Vapnik이 맞다.

VC차원은 1로, 생각할 수 있는 가장 단순한 모델 중 하나일 것이다. 100개의 훈련 데이터가 주어질 때, E_{out}이 E_{in}에서 0.1 이상 벗어날 확률은 얼마나 될까? 다음 수식을 δ에 대해 풀면

$$\sqrt{\frac{8}{100}\ln\frac{4\cdot 101}{\delta}} = 0.1$$

356.52를 얻을 수 있다. 확률 상한으로는 별로 의미가 없는 수치이다. :-p 그래서 VC 차원을 이용한 일반화 성질은 이론적인 결과에 훨씬 가깝지만, 교과서에서는 몇 개의 휴리스틱 규칙을 제안하고 있다.

- VC 차원 분석은 가설 집합의 형태와 상관없이 느슨한 상한이기 때문에, (경험적으로) 많은 경우 VC 차원은 가설 집합의 일반화 성능과 연관된다. 따라서, VC 차원이 작은 가설 집합은 좀 더 잘 일반화될 가능성이 높다.
- 또 다른 중요한 휴리스틱 규칙은, 만약 당신의 가설 집합의 VC 차원이 d_{VC}라면 최소한 $10\times d_{VC}$개의 훈련 데이터는 사용하는 게 좋다는 것이다. 이건 꽤 작은 숫자인 듯하다.

2.3.2 VC 차원에 대한 직관

수업에서는 VC 차원은 모델의 effective degree of freedom을 측정한다고 말하고 있다. 물론 인자가 늘어난다고 모델의 degree of freedom이 증가하는 것은 아니지만, VC 차원과 인자의 수가 대략적으로 비례한다는 직관은 꽤 유용하다.

2.3.3 좀 더 직접적으로 E_{out} 측정하기

물론 VC 차원을 이용해 E_{out}을 예측할 수도 있지만, 좀 더 직접적으로 그를 예측하는 방법은 별도의 테스트 데이터를 사용하는 것이다. 테스트 데이터는 훈련 과정에서 사용되지 않았기 때문에, 단순한 Hoeffiding 부등식을 사용할 수 있게 한다.

2.4 Bias-Variance 트레이드오프

E_{out}을 분석하는 다른 방법으로 Bias-Variance 분석이 있다. 텍스트에서는 회귀 분석 문제를 중심으로 이 분석을 설명한다.

우리의 훈련 데이터 $\mathcal{D}$가, 모집단에서 임의로 선택된 확률 변수라고 하자. 그러면 우리가 선택할 최종 가설 g 또한 $\mathcal{D}$에 의존적인 확률 변수가 된다. 이것을 명시적으로 $g^{(\mathcal{D})}$와 같이 표현하자. 그러면 out-of-sample 오류의 기대치를 다음과 같이 쓸 수 있다.

$$E_{out}\left(g^{(\mathcal{D})}\right) = \mathbb{E}_{\mathbf{x}}\left[\left(g^{(\mathcal{D})}\left(\mathbf{x}\right) - f\left(\mathbf{x}\right)\right)^2\right]$$

이것을 $\mathcal{D}$에 대한 기대치로 정리하면 다음과 같다:

$$\begin{aligned}
\mathbb{E}_{\mathcal{D}}\left[E_{out}\left(g^{(\mathcal{D})}\right)\right] &= \mathbb{E}_{\mathcal{D}}\left[\mathbb{E}_{\mathbf{x}}\left[\left(g^{(\mathcal{D})}\left(\mathbf{x}\right) - f\left(\mathbf{x}\right)\right)^2\right]\right] \\
&= \mathbb{E}_{\mathbf{x}}\left[\mathbb{E}_{\mathcal{D}}\left[\left(g^{(\mathcal{D})}\left(\mathbf{x}\right) - f\left(\mathbf{x}\right)\right)^2\right]\right] \\
&= \mathbb{E}_{\mathbf{x}}\left[\mathbb{E}_{\mathcal{D}}\left[g^{(\mathcal{D})}\left(\mathbf{x}\right)\right]^2 - 2\mathbb{E}_{\mathcal{D}}\left[g^{(\mathcal{D})}\left(\mathbf{x}\right)\right] f\left(\mathbf{x}\right) + f\left(\mathbf{x}\right)^2\right]
\end{aligned}$$

위 형태에서 식 $\mathbb{E}_{\mathcal{D}}\left[g^{(\mathcal{D})}(\mathbf{x})\right]$에 주목하자. 이것은 평균적인 가설이 내놓는 결과값이다. 이는 $\mathcal{D}$를 모집합에서 샘플링해서 가설을 학습하는 것을 반복한 뒤, 결과로 얻어지는 모델들의 결과를 평균한 것을 말한다. (이 "평균" 가설은 더 이상 원래 가설 집합에 포함되어 있지 않을 수도 있다는 것에 유의하라!) 이 가설을 $\bar{g}(x)$라고 하면, 위를 $\bar{g}$에 대해 다음과 같이 쓸 수 있다.

$$\begin{aligned}
\mathbb{E}_{\mathcal{D}}\left[E_{out}\left(g^{(\mathcal{D})}\right)\right] &= \mathbb{E}_{\mathbf{x}}\left[\mathbb{E}_{\mathcal{D}}\left[g^{(\mathcal{D})}(\mathbf{x})\right]^2 - 2\bar{g}(\mathbf{x})f(\mathbf{x}) + f(\mathbf{x})^2\right] \\
&= \mathbb{E}_{\mathbf{x}}\left[\mathbb{E}_{\mathcal{D}}\left[g^{(\mathcal{D})}(\mathbf{x})\right]^2 - \bar{g}(\mathbf{x}) + \bar{g}(\mathbf{x}) - 2\bar{g}(\mathbf{x})f(\mathbf{x}) + f(\mathbf{x})^2\right] \\
&= \mathbb{E}_{\mathbf{x}}\left[\underbrace{\mathbb{E}_{\mathcal{D}}\left[g^{(\mathcal{D})}(\mathbf{x}) - \bar{g}(\mathbf{x})\right]^2}_{\text{variance}} + \underbrace{(\bar{g}(\mathbf{x}) - f(\mathbf{x}))^2}_{\text{bias}}\right]
\end{aligned}$$

이것을 Bias-variance 분해라고 한다.

2.4.1 Bias-Variance의 해석

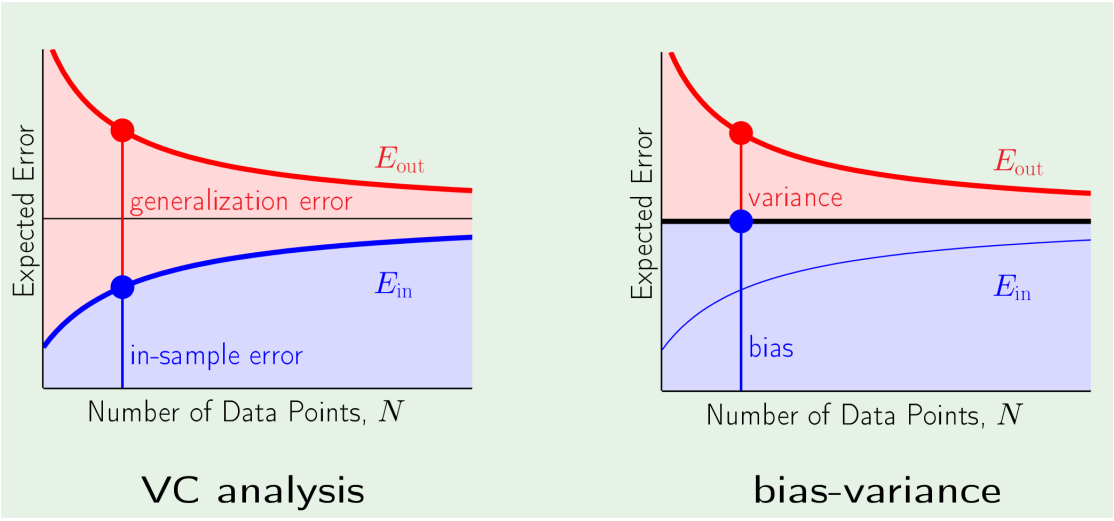
이 분해를 어떻게 해석할 수 있을까?

- Bias는, 대략적으로, 평균 가설 $\bar{g}$와 목표 함수 사이의 오류를 의미한다. 다시 말해 우리가 평균적으로 얼마나 틀릴 것인가를 나타내는데, 이것은 우리의 가설 집합이 목표 함수를 비슷하게 근사할 수 있을 정도로 유연한지를 나타낸다.
- Variance는 임의의 $\mathcal{D}$를 이용해 얻은 $g^{(\mathcal{D})}$가 $\bar{g}$에서 얼마나 멀리 떨어지는가를 나타낸다. 다시 말해, 우리의 최종 가설 g가 $\mathcal{D}$에 얼마나 민감한지를 나타낸다. 직관적으로는, variance가 학습 모델의 "안정성"을 말한다고 할 수도 있다.

이 두 값이 어떻게 트레이드오프 관계를 갖는지 아는 것은 어렵지 않다. 단순한 가설 집합은 높은 bias를 갖지만, variance는 작을 것이다. 반면 복잡한 모델은 variance가 커지지만, bias가 작아질 것이다. 물론 훈련 데이터의 크기가 커질 수록 variance는 점점 감소하며, 결과적으로는 bias가 E_{out}의 대부분을 차지하게 될 것이다.

2.5 VC 차원 분석과 Bias-variance 분석의 비교

강의 자료에서 훔쳐온 아래 그림은 두 개의 분석이 E_{out}을 어떻게 나눠보는지를 잘 대조해 보여준다.



3 오버피팅

이 장에서는 오버피팅이 발생하는 이유, 그리고 오버피팅에 대처하는 가장 기초적인 방법인 정규화(regularization)와 검증(validation)에 대해 언급한다.

3.1 오버피팅이 발생할 때

오버피팅은 언제 발생할까? 가장 당연하고 직관적인 경우는, 실제 목적 함수를 표현하는 데 필요한 것보다 복잡한 가설 집합을 사용하는 것이다. 지나치게 복잡한 가설 집합에는 어떤 문제가 있을까? 복잡한 가설 집합은 너무 유연해서, 입력에 주어지는 노이즈들을 피팅해 버린다.

위와 같은 정의만 들으면, 오버피팅은 목적 함수보다 복잡한 가설 집합에서 온다는 것으로 이해하기 쉽다. 그러나, 가설 집합이 목적 함수보다 단순한 경우에도 오버피팅은 발생할 수 있다. 교과서에서는 간단한 예를 들어 보인다. 10차 방정식 + 작은 노이즈가 목적 함수라고 하자. 다항식 피터를 사용해 2차 혹은 10차 방정식으로 훈련 데이터를 피팅한다. 어느 쪽이 더 결과가 좋을까? 10차 방정식을 10차 방정식으로 피팅하니 당연히 10차가 더 좋을 것 같지만, 이들은 노이즈에 너무 큰 영향을 받아서 결과적으로 더 높은 E_{out}을 갖게 된다. (좀 pathological 한 예인 것 같기는 한데, 요점은 일단 그런 것으로) 놀라운 것은 2차 방정식을 사용할 때 우리는 목표 함수를 표현할 수 있는 가능성을 아예 포기했음에도 더 나은 결과를 얻었다는 것이다. 왜 그럴까? E_{out}과 E_{in}을 입력의 크기에 대해 그려볼 때, 복잡한 가설 집합은 결과적으로 더 낮은 오류율에서 E_{in}과 E_{out}이 수렴하지만, 초반에는 E_{in}과 E_{out} 간의 간격이 너무 크다. 따라서 우리가 사용하려고 했던 10차 방정식을 사용하기에는 입력의 개수가 너무 작았다고 볼 수 있다.

3.1.1 결정적 노이즈와 비결정적 노이즈

오버피팅이 노이즈의 결과물이라면, 노이즈가 없이 정확히 측정된 입력을 사용하는 문제에서는 오버피팅이 없을까? 그렇지 않다. 교과서는 결정적 노이즈(deterministic noise)의 개념으로 이 현상을 설명한다. 목적 함수가 가설 집합에 포함되어 있을 가능성은 적다. 그렇다면 가설 집합 중에서 가장 좋은 핏(best fit)을 갖는 가설을 찾게 되는데, 이 때 가설과

목적 함수 간의 차이는 가설 입장에서는 노이즈와 다름없다. 이 노이즈는 일반적으로 우리가 $N\left(0,\epsilon^2\right)$의 분포를 가진다고 가정하는 비결정적 노이즈(stochastic noise)와는 다르게 목적 함수에 의해 결정적으로(deterministically) 정해지지만, 학습 알고리즘 입장에서는 똑같은 노이즈일 뿐이다.

비결정적 노이즈는 목적 함수의 복잡도가 커질 수록 증가하고, 따라서 오버피팅은 목적 함수가 복잡해질 수록 발생하기 쉬워진다. 이것은 나의 직관과 달랐는데, 내 직관에서는 목적 함수가 충분히 복잡하다면 아무리 복잡한 가설 집합을 던져도 이 목적 함수를 다 표현하기 어려우니 오버피팅이 발생할 가능성이 적었기 때문이다.

3.2 정규화 (regularization)

어떤 학습 알고리즘을 사용하건 간에, 오버피팅과 싸우기 위한 가장 효과적인 무기는 정규화이다. 정규화를 이해하기 위한 한 가지 방법은, 우리가 사용하는 모델의 복잡도에 대해 페널티를 부여하는 것으로 이해하는 것이다. VC Generalization Bound에서, 우리는 E_{in}과 E_{out}의 차이가 증가 함수 $m_{\mathcal{H}}(N)$에 좌우된다고 보았다. 증가 함수는 가설 집합이 복잡할 수록 커지므로, 가설이 복잡할 수록 페널티를 준다는 개념은 VC Generalization Bound와 일맥상통한다. 그래서, 모델의 복잡도에 불이익을 주는 augmented error E_{aug}는 E_{in}보다 E_{out}을 더 잘 예측할 수 있다는 것이 정규화의 요점이다. (단 VC 차원에서는 가설 집합의 복잡도를 쟀다면, 정규화에서는 특정 가설의 복잡도를 측정한다는 차이가 있다.)

3.2.1 Soft Threshold와 Ridge Regression

교과서에서 첫 번째로 드는 정규화의 예는 (당연하게도) Tikhonov 정규화이다. 여러 번 본 내용이지만 다시 정리해 보자. 선형 회귀의 정규화에서 가중치 벡터 $\mathbf{w}$에 대한 Soft threshold는 다음과 같은 조건을 의미한다.

$$\mathbf{w}^T\mathbf{w} \leq C$$

이 조건은 $\mathbf{w}$가 가질 수 있는 값들을 제한하므로, 가설 집합의 크기를 줄여 VC 차원을 줄이는 효과를 준다. 잘 알다시피, 이 조건은 ridge regression과 동치인데, 이것을 다음과 같이 간단히 보일 수 있다.

$\mathbf{w}_{lin}$을 OLS가 반환하는 가중치 벡터라고 할 때, $\mathbf{w}_{lin}^T\mathbf{w}_{lin} \leq C$ 라면 이미 조건이 만족되었으므로, 정규화된 선형 회귀 결과 $\mathbf{w}_{reg}$=$\mathbf{w}_{lin}$으로 둘 수 있다. 반면 $\mathbf{w}_{lin}^T\mathbf{w}_{lin} > C$라면 어떨까? $\mathbf{w}_{lin}^T\mathbf{w}_{lin} = C$가 될 때까지 $\mathbf{w}_{lin}$의 원소들의 크기를 줄여나가야 할 것이다. 이 때 $\mathbf{w}^T\mathbf{w} = C$인 벡터 중 $E_{in}(\mathbf{w})$를 최소화하는 $\mathbf{w}$를 $\mathbf{w}_{reg}$로 채택하게 된다. 따라서 이것은 제약 있는 최적화 문제(constrainted optimization)로, 라그랑주 승수법(Lagrange multiplier)을 이용해 풀 수 있다.

라그랑주 승수법을 쓰기 위해 다음과 같이 정의하자: $f(\mathbf{w}) = E_{in}(\mathbf{w})$, $g(\mathbf{w}) = \mathbf{w}^T\mathbf{w} - C$. 따라서 우리의 목적은 $g(\mathbf{w}) = 0$일 때 $f(\mathbf{w})$를 최소화하는 것이다. 라그랑주 승수법의 중요한 통찰은 우리가 원하는 최소점에서는∇f와 ∇g가 서로 평행이라는 것이다. 만약 이들이 서로 평행이 아니라면, ∇g의 수직 방향으로 조금 움직이면 $g = 0$ 조건을 유지하면서 f를 줄이거나 늘릴 수 있게 되기 때문이다. 따라서 이 조건을 풀어 쓰면, 어떤 λ에 대해 다음이 성립해야 한다고 쓸 수 있다.

$$\nabla f = \lambda \nabla g$$

f와 g의 정의를 대입하면

$$\nabla E_{in}(\mathbf{w}) = -2\lambda_C \mathbf{w}^T$$

로 쓸 수 있다. (상수 2를 도입한 이유는 뒤에 명확해지며, 음수 부호는 λ_C는 양수로 만들기 위해 도입했다. 실제로 두 등위곡선이 만나는 지점에서 두 함수의 gradient[2]는 서로 반대 방향이기 때문이다.)

그런데 이 조건은 다시 쓰면

$$\nabla E_{in}(\mathbf{w}) + 2\lambda_C \mathbf{w}^T = 0 \iff \nabla \left(E_{in}(\mathbf{w}) + \lambda_C \mathbf{w}^T \mathbf{w} \right) = 0$$

즉 CvxOpt에서 언급했던 라그랑지안(Lagrangian)의 최소점이라는 것을 알 수 있다. 이 라그랑지안의 형태는 결국 Tikhonov regression에서 최소화하기를 요구하는 값과 같으므로, 두 방법은 서로 동등하다. 물론 이 때 λ_C와 C 사이에는 자명한 관계는 없으며, 둘의 관계는 데이터에 의해 결정된다. 물론 둘 사이에 대략적인 반비례 관계가 존재한다는 것은 직관적으로 이해할 수 있다: 보다 큰 C는 가중치들을 더 클 수 있도록 허용하며, 따라서 λ_C가 더 작아지게 한다.

3.2.2 어떻게 정규화 형태(regularizer)를 고를 것인가?

CvxOpt에서 이미 다뤘듯이, 다른 regularizer는 베이지안 관점에서 인자의 분포에 대한 다른 믿음(prior)를 나타낸다. 따라서 정규화 형태를 선택하는 것은 휴리스틱에 가까우며, 가장 좋은 방법이란 없다. 수업에서는 다음과 같은 규칙을 제안한다: 비결정적 노이즈는 high-frequency이고, 결정적 노이즈 또한 불연속적이다. 따라서 가능한한 매끄러운smooth 가설을 장려하는 정규화를 선택하는 것이 좋다.

3.2.3 λ의 선택

정규화 파라미터 λ의 선택은, 물론, 검증 데이터를 통해 이뤄진다.

3.3 검증 (validation)

이 절에는 따로 적을 만한 내용이 별로 없어, 간단하게만 요약한다. 훈련 데이터에서 일부를 떼어 놨다가, 마지막에 파라미터 튜닝에 사용하는 것이 검증이다. 검증의 가장 큰 사용처는 모델 선택이다: 여러 개의 가설 집합 중 어느 것을 쓸 것인가?

이런 질문을 할 수 있다: 신경망의 계수나, SVM의 지지 벡터나 파라미터나 모두 데이터에서 배우는 것인데, 그렇다면 검증 데이터와 훈련 데이터는 무엇이 다른가? 이와 같은 질문은 충분히 의미있는 것인데, 훈련과 검증 사이의 경계는 사실 희미하고 검증은 엄밀한 수학적 기반을 둔 기술이라기보다는 휴리스틱에 가까운 것이기 때문이다. 단, 검증의 목표는 가능한 E_{out}에 대한 좋은 예측을 얻는 것이기 때문에, 검증 데이터를 이용해 최적화하는 파라미터의 수를 최소화하는 것이 좋다. 너무 많은 파라미터를 검증을 통해 학습하게 되면 검증 셋이 너무 "더럽혀져서contaminated" 검증 셋으로서의 의미를 잃게 된다.

[2] 아.. 한글로 좀 쓰려고 해봤는데 못해먹겠다.