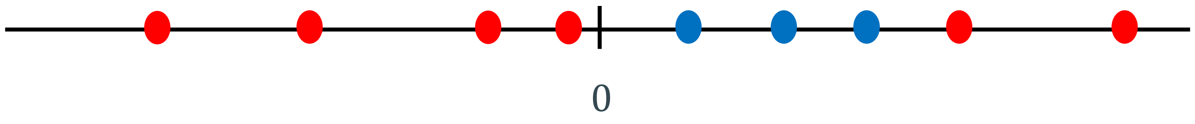
Foundation of Machine Learning 7주차

정재헌, 우지수 / 2023.03.02

Computational Data Science LAB

CONTENTS

How to Solve non-linear Problem

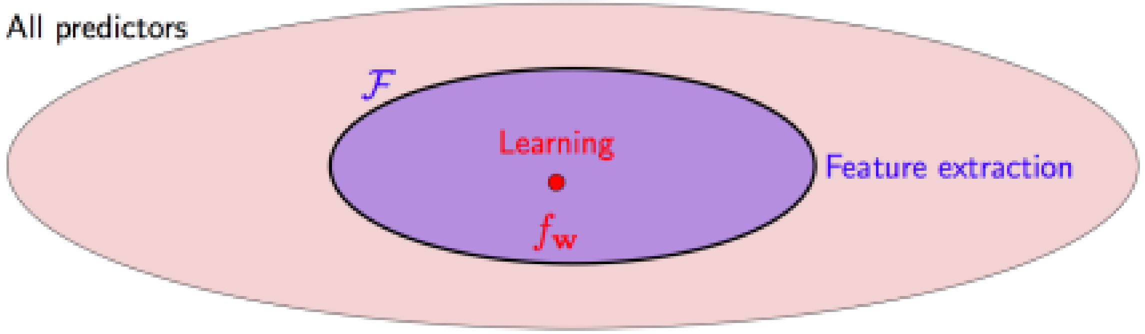


✓ 위와 같이 선형 분리가 불가능한 데이터의 경우, 어떻게 분류하는 것이 좋을까 ?

01 What is Feature Extraction?

How to express the data?

- 확장된 가설공간



✓ $\varphi : X \rightarrow R^d : F = \{f(x) = w^T \varphi(x)\}$

✓ 가설 공간(hypothesis space)은 학습 알고리즘이 가능한 가설(모델)의 집합

✓ 이 가설 공간을 얼마나 크게 만들 수 있느냐가 **모델의 표현 능력(expressivity)**을 결정

01 | What is Feature Extraction?

What is feature extraction?

- Feature Extraction

Raw Input → Feature Extraction → Feature Vector

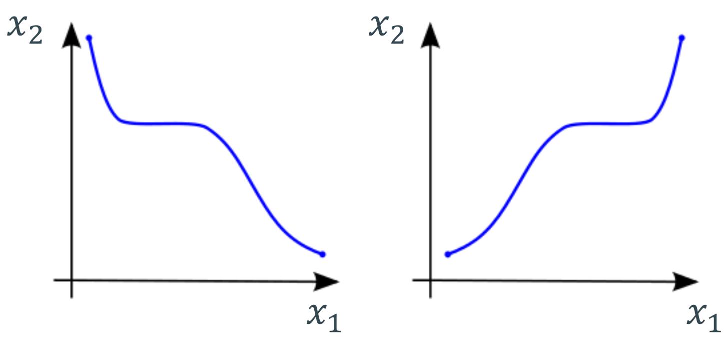
$(x_1, x_2) \rightarrow \emptyset(x) \rightarrow (x_1^2, x_2^2, \sqrt{2}x_1x_2\ )$

Mapping Function

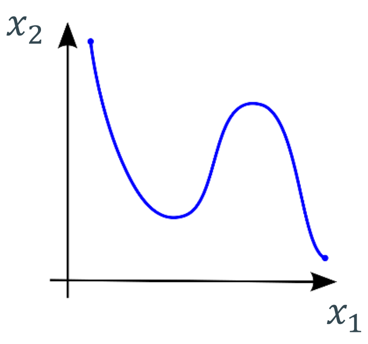
02 | What is non-linear ?

Non-linear problem’s Characteristics 1

1. 단조성이 아닌 경우(non-monotonicity)



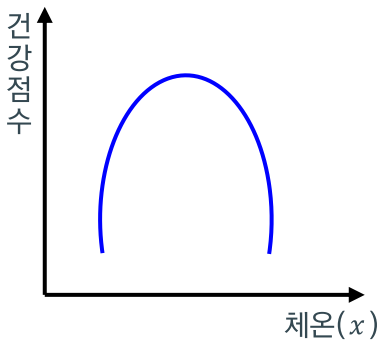
단조성



단조성이 아닌 경우

Non-linear problem's Characteristics 1

1. 단조성이 아닌 경우(non-monotonicity)



- 예시
  - ✓ $y = a$체온$(x)+b$
  - ✓ $y \in R(positive\ is\ good)$: 건강점수 예측
  - ✓ Hypothesis Space $F = \{affine\ functions\ of\ temperature\}$

건강 점수와 체온은 선형(affine)함수가 아니다.

- Transform 예시

Transform the input: $\varphi(x) = [1, \{temperature(x) - 37\}^2]$

즉, 도메인 지식을 이용해 input을 변환 → 결국 도메인 지식이 없으면 변환불가

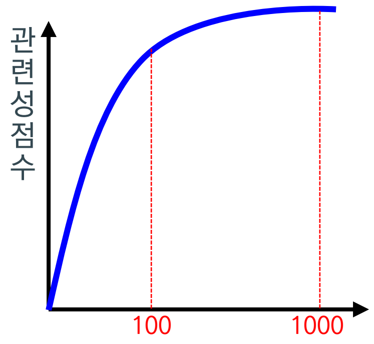
- Feature Extraction 예시

Feature Extraction the input: $\varphi(x) = [1, temperature(x), \left(temperature(x)\right)^2]$

가지고 있는 feature를 최대한 활용해서 transform 보다 상대적으로 **표현력이** 뛰어나게 할 수 있음

Non-linear problem' s Characteristics 2

2. 포화(saturation)



- 포화는 $N(x)$가 증가하고 관련성 점수도 증가할 때, $N(x)$는 계속해서 증가하고 관련성 점수가 임계점을 넘으면 더 이상 증가하지 않는 상황을 이야기함

- 예시

✓ 사용자의 검색어와 관련된 제품을 찾음

✓ Input: Product X

✓ y: X의 검색어와의 **관련성**을 점수로 평가

✓ $y = N(x) + b$, where $N(x) = x$를 구매한 사람들의 숫자

N(x)가 1000이 되었다고 해서 N(x)가 100일 때 보다 10배 더 신뢰할 수 있다고 말할 수 없음

Solution

- Transform 예시

Transform the input: $\varphi(x) = [1, \log\{1 + N(x)\}]$

로그함수를 이용해 input을 변환

- Feature Extraction 예시

Feature Extraction the input: $\varphi(x)$ = $(1(5 \leq N(x) < 10), 1(10 \leq N(x) < 100), 1(100 \leq N(x)))$

가지고 있는 feature를 구간별로 나누어 각 구간에 해당하는 이산적인 값으로 변환

Non-linear problem’s Characteristics 3

3. 특징간 상호작용(interactions between features)

- ✓ 두 개 이상의 feature가 함께 사용될 때 서로 영향을 미치는 경우를 의미

- 예시
- ✓ Input: Patient information x
- ✓ 건강점수 $y \in R$ (높을수록 좋습니다.)
- ✓ $y = \text{a}$신장$(x) + b$체중(x)
- ✓ Issue: 신장(x)에 관련된 체중(x)이 중요

두 feature가 각각 서로에게 영향을 미친다면 선형적으로 분리가 불가함

예를들어 원 함수에서 2m, 150kg이면 높은 건강점수를 주고

150cm, 40kg이면 낮은 건강점수를 줌

- Transform 예시

Transform the input: $이상체중(kg) = 52 + 1.9[신장(inch) - 60] \rightarrow f(x) = \left(52 + 1.9[신장(x) - 60] - 이상체중(x)\right)^2$

$f\ (x) = 3.61신장(x)^2 - 3.8신장(x)체중(x) - 235.6신장(x) + 체중(x)\hat{}2 + 124체중(x) + 3844$

구글에서 검색한 " 키에 따른 이상체중 " → 외부도구가 없으면 변환 불가

- Feature Extraction 예시

Transform the input: $\varphi(x) = [1, h(x), w(x), h(x)\hat{}2, w(x)\hat{}2, h(x)w(x)(교차\ 항)]$

상대적으로 **유연해지며** 구글과 같은 검색엔진을 **필요로 하지 않음**

Problem

- Feature Extraction 을 통해 만들어진 거대한 feature space는 풍부한 표현력을 가질 수 있음

- 거대한 feature space의 문제점
  1. 과적합
  2. 메모리 및 계산 비용

- 과적합은 정규화를 통해 계산이 가능
- 메모리 및 계산 비용을 줄이기 위한 kernel methods

Kernel Methods: SVM

- SVM에서 kernel method에 대한 예시
- Liner SVM에서 매핑을 통해 input space를 고차원의 feature space로 보냄

$$k(x_i, x_j) = < \psi(x_i), \psi(x_j) >$$

$$\sup_{\alpha} \textstyle\sum_{i=0}^{n} \ \alpha_i - \frac{1}{2} \displaystyle\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \psi(x_i)^T \psi(x_i)$$

$$s.t. \sum_{i=1}^{n} \alpha_i \, y_i = 0$$

$$\alpha_i \in [0, \frac{c}{n}], \ i = 1, \dots, n$$

〈Feature Extraction〉

$$\sup_{\alpha} \textstyle\sum_{i=0}^{n} \ \alpha_i - \frac{1}{2} \displaystyle\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

$$s.t. \sum_{i=1}^{n} \alpha_i \, y_i = 0$$

$$\alpha_i \in [0, \frac{c}{n}], \ i = 1, \dots, n$$

〈Kernel Method〉

04 | Kernel Methods

Kernelized

- Kernelized: 입력 공간(input space)의 내적으로 나타내는 값을 기반으로 커널함수를 사용하는 방법

$$\psi(x_i)^T \psi(x_i) \text{ for } x_i, x_j \in X$$

- kernel function은 ψ 와 내적(inner product) $<\cdot\,,\cdot>$ 으로 나타냄

$$k(x_i, x_j) = <\psi(x_i), \psi(x_j)>$$

- 매핑함수를 각각 계산하지 않고 한번에 계산이 가능

예시: Quadratic Kernel

- $x = (a_1, a_2, a_3), \mathrm{x}' = (\mathrm{b}_1, \mathrm{b}_2, \mathrm{b}_3)$
- Feature map : $\psi(x) = (x_1, \ldots, x_d, x_1^2, \ldots, x_d^2, \sqrt{2}x_1x_2, \ldots, \sqrt{2}x_{d-1}x_d)$
- $\psi(x), \psi(x')$ 각각 계산 하는 경우:

$$\psi(x) = a_1, a_2, a_3, a_1^2, a_2^2, a_3^2, \sqrt{2}a_1a_2, \sqrt{2}a_2a_3, \sqrt{2}a_3a_1$$

$$\psi(x') = b_1, b_2, b_3, b_1^2, b_2^2, b_3^2, \sqrt{2}b_1b_2, \sqrt{2}b_2b_3, \sqrt{2}b_3b_1$$

  - 후에 $\psi(x)^T\psi(x')$

- 커널함수를 이용하는 경우: 실제 데이터를 고차원에 매핑시키지 않아도 저차원에서 내적연산이 가능함(KERNEL TRICK)

$$K(x, x') = <\psi(x), \psi(x')> = <x, x'> + <x, x'>^2$$

$$(a_1 \cdot b_1) + (a_2 \cdot b_2) + (a_3 \cdot b_3) + (a_1 \cdot b_1)^2 + (a_2 \cdot b_2)^2 + (a_3 \cdot b_3)^2$$

Inner Product Space(Pre-Hilbert Space)

- 내적 공간(Inner Product Space)은 벡터 공간(V)과 내적 연산이 함께 정의된 공간

$$<\cdot,\cdot>: \mathcal{V} \times \mathcal{V} \longrightarrow \boldsymbol{R}$$

- 실수 a, b와 벡터공간(V)의 모든 x, y, z에 대해 다음 3가지 속성을 갖음

1. 대칭성(Symmetry): $< x, y > = < y, x >$
2. 선형성(Linearity): $< ax + by >, < z > = a < x, z > \; + b < y, z >$
3. 양의 정부호성(Positive-definiteness): $< x, x > \geq 0 \; and \;\; < x, x > = \; 0 \; \Leftrightarrow \; x \; = \; 0$

Norm

- 내적 공간(Inner Product Space)에서의 norm

$$||x|| = \sqrt{< x, x >} = \sqrt{x^T x}$$

- 예시)

$$< x, y > := x^T y, \ \forall x, y \in \boldsymbol{R}^d$$

Pythagorean Theorem

- 정의
  - $< x, x' > = 0$이면, 두벡터는 직교하고, 다음과같이 표기됨 $x \perp x'$
  - 집합 S의 모든 $\vec{x}$에 대해 $\vec{x} \perp S$이면 $\vec{x}$는 집합 S의 모든 원소와 직교
- Pythagorean Theorem
  - $x \perp x'$이면 다음식이 성립

$$||x + x'||^2 = ||x||^2 + ||x'||^2$$

$$||x + x'||^2 = < x + x', x + x' >$$

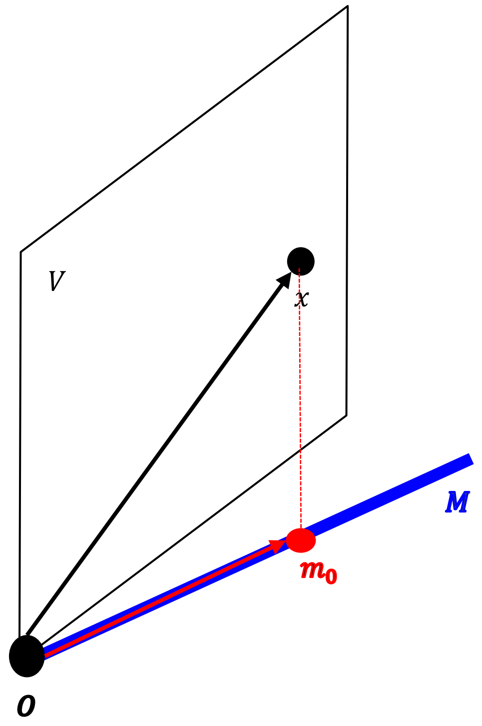
$$= < x, x > + < x, x' > + < x', x > + < x', x' >$$

$$= \boldsymbol{||x||^2 + ||x'||^2} \cdots (1)$$

Projection onto a Plane

- 내적공간($\mathcal{V}$)의 일부 x를 선정
- M을 내적공간($\mathcal{V}$)의 부분공간(subspace)이라 가정
- m_0는 M위의 x의 투영(projection)
  - ➢ $m_0 \in M$이면 M의 x와 가장 가까운 점
- M에 속하는 모든 m에 대해 다음식을 만족

$$||x - m_0|| \leq ||x - m||$$



Hilbert Space

- Hilbert 공간은 내적을 갖는 공간, 무한차원의 내적공간에서 투영이 가능
  - 이때, 무한차원에서 모든 수열이 수렴한다는 성질인 완비성(completeness)이 존재
- 따라서 Hilbert 공간에서 내적과 투영을 이용한 다양한 연산이 가능

Kernel Method에서 **입력데이터**를 저차원 공간에서 무한차원의 Hilbert 공간으로 매핑하고,
매핑된 데이터를 사영을 통해 내적계산을 하여 분류나 회귀를 수행하게 됨

Classical Projection Theorem

- H 는 Hilbert space
- M 는 H의 닫힌(수렴하는) 부분공간
- 임의의 $x \in H$에 대해, 아래식과 같은 고유한 $m_0 \in M$가 존재
- m_0는 x를 M위로 투영
- M위의 m_0는 M에대한 x의 투영

$$||x - m_0|| \leq ||x - m||, \qquad \forall m \in M$$

$$x - m_0 \perp M \cdots (\mathbf{2})$$

Projection Reduces Norm

- M을 H 의 닫힌 부분공간으로 가정할 때, 임의의 $x \in H$의 경우,
- $m_0 = Proj_M x$를 M에대한 $\vec{x}$의 투영이라고 할 때, 다음식을 만족

$$||m_0|| \le ||x||$$

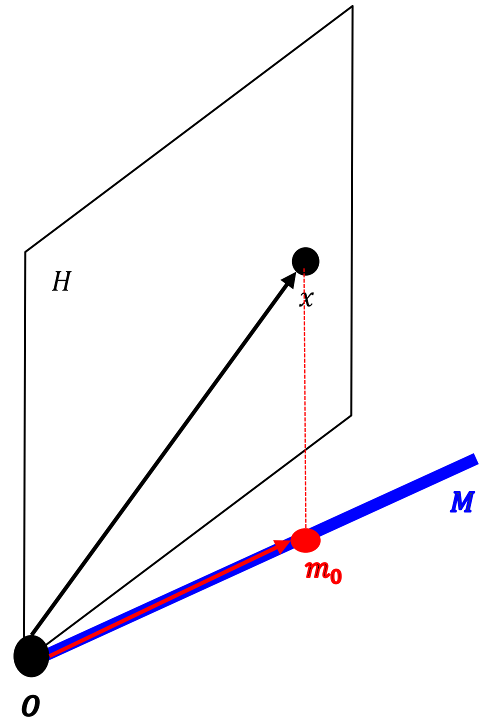
- $m_0 \;=\; x$ 때 같은 값을 갖는다면 다음 식을 만족

$$||x||^2 = ||m_0 + (x - m_0)||^2 \;\; ((x \;-\; m0) \perp m0) \text{ 투영이론에 의해} \cdots (2)$$

$$= ||m_0|| + || \; x - m_0 \; ||^2 \text{ 피타고라스 이론에 의해} \cdots (1)$$

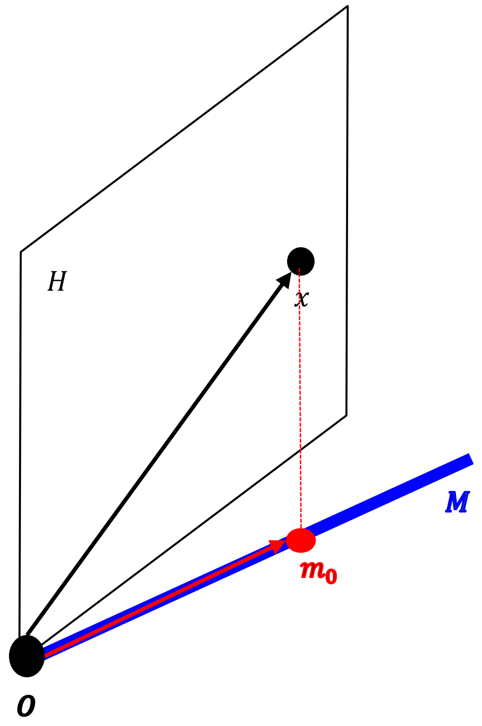
$$\therefore || \; \boldsymbol{m_0} \; ||^2 = || \; \boldsymbol{x} \; ||^2 - || \; \boldsymbol{x} - \boldsymbol{m_0} \; ||^2$$

$|| \; x - m_0 \; ||^2 \ge 0$ 일 때, $|| \; m_0 \; ||^2 \le || \; x \; ||^2$, $\;|| \; x - m_0 \; ||^2 = 0$ 일 때, $x = m_0$



Kernel Trick

- 실제 데이터를 고차원에 매핑시키지 않아도 저차원에서 내적연산이 가능함
- 이걸 Kernel Trick이라 함
- 이 때 사용되는 kernel은 polynomial kernel, RBF kernel 등이 있음
- 내적연산을 사영을 통해 계산 비용을 줄일 수 있다는 insight를 얻을 수 있음



Q&A

감사합니다.