밑바닥부터 시작하는 데이터 과학

2회차

- 선형대수
- 최적화
- 차원축소
- 선형회귀 기초

1. 선형 대수

선형대수 (Linear Algebra)

벡터 공간을 다루는 수학의 한 분야

1. 선형 대수

데이터 분석에서의 행렬

데이터 분석에서는 기본적으로 아래와 같은 테이블 구조를 다룬다.

- 계산의 편의성을 위해 행렬로 변환하여 분석한다.

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	Southampton	no	False
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	Cherbourg	yes	False
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southampton	yes	True
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	Southampton	yes	False
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	Southampton	no	True

1. 선형 대수

데이터 유형

(1) 스칼라 (scalar)

- 하나의 숫자만으로 이루어진 데이터

(2) 벡터 (vector)

- 벡터끼리 더하거나 상수와 곱해지면 새로운 벡터를 생성하는 개념적인 도구.
- 일반적으로 1차원 배열을 벡터라고 한다.
- 하나의 벡터를 이루는 데이터의 개수를 차원(dimension)라고 한다.

사람을 표현하는 벡터

- [키, 몸무게, 나이] 인 3차원 벡터

(3) 행렬 (matrix)

- 행렬 : 2차원 배열
- 3차원 이상 배열은 텐서(tensor)라고 함

1. 선형 대수

벡터와 행렬 연산

(1) 스칼라 곱

$$c_1 = 5, \quad Y = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

$$c_1 Y = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} = 5 \cdot \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} = \begin{pmatrix} 5 \\ 10 \\ 15 \end{pmatrix}$$

(2) 벡터 덧셈

$$X = \begin{pmatrix} 1 \\ 3 \\ 5 \end{pmatrix}, \quad Y = \begin{pmatrix} 2 \\ -1 \\ 0 \end{pmatrix}$$

$$X + Y = \begin{pmatrix} 1 \\ 3 \\ 5 \end{pmatrix} + \begin{pmatrix} 2 \\ -1 \\ 0 \end{pmatrix} = \begin{pmatrix} 3 \\ -2 \\ 5 \end{pmatrix}$$

1. 선형 대수

벡터와 행렬 연산

(3) 전치행렬

- **전치(transpose)** 연산은 행렬에서 가장 기본이 되는 연산으로 행렬의 **행과 열을 바꾸는 연산**을 말한다.

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & x_{1,3} & x_{1,4} \\ x_{2,1} & x_{2,2} & x_{2,3} & x_{2,4} \\ x_{3,1} & x_{3,2} & x_{3,3} & x_{3,4} \\ x_{4,1} & x_{4,2} & x_{4,3} & x_{4,4} \\ x_{5,1} & x_{5,2} & x_{5,3} & x_{5,4} \\ x_{6,1} & x_{6,2} & x_{6,3} & x_{6,4} \end{bmatrix} \rightarrow X^T = \begin{bmatrix} x_{1,1} & x_{2,1} & x_{3,1} & x_{4,1} & x_{5,1} & x_{6,1} \\ x_{1,2} & x_{2,2} & x_{3,2} & x_{4,2} & x_{5,2} & x_{6,2} \\ x_{1,3} & x_{2,3} & x_{3,3} & x_{4,3} & x_{5,3} & x_{6,3} \\ x_{1,4} & x_{2,4} & x_{3,4} & x_{4,4} & x_{5,4} & x_{6,4} \end{bmatrix}$$

(4) 역행렬

- **정방행렬** A에 역행렬 A^{-1} 은 행렬 A에 대해 다음과 같은 관계를 만족한다.

$$A^{-1}A = AA^{-1} = I$$

- 역행렬은 다음과 같은 성질이 성립한다.

$$(AB)^{-1} = B^{-1}A^{-1}$$
$$(ABC)^{-1} = C^{-1}B^{-1}A^{-1}$$

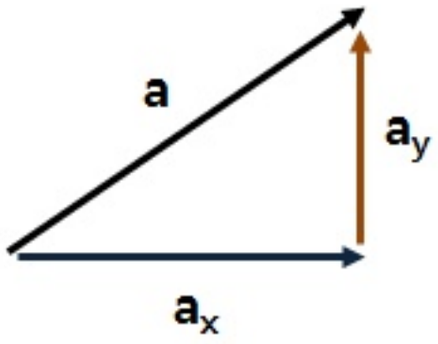
1. 선형 대수

벡터의 크기

벡터의 크기 (길이)

- 2차원 벡터 a의 길이는 피타고라스 정리를 이용하여 계산할 수 있으며 그 값은 벡터의 놈(norm) ‖ a ‖ 이다. N차원 벡터의 길이도 마찬가지로 벡터 놈으로 정의한다.

$$||a|| = \sqrt{a^T a} = \sqrt{a_1^2 + a_2^2 ... + a_N^2}$$



$$|\overrightarrow{AB}| = |\vec{a}| = \|\vec{a}\| = \sqrt{a_x{}^2 + a_y{}^2}$$

- 벡터의 크기가 1인 벡터를 **단위벡터 (unit vector)**라고 한다.

1. 선형 대수

투영 (projection)

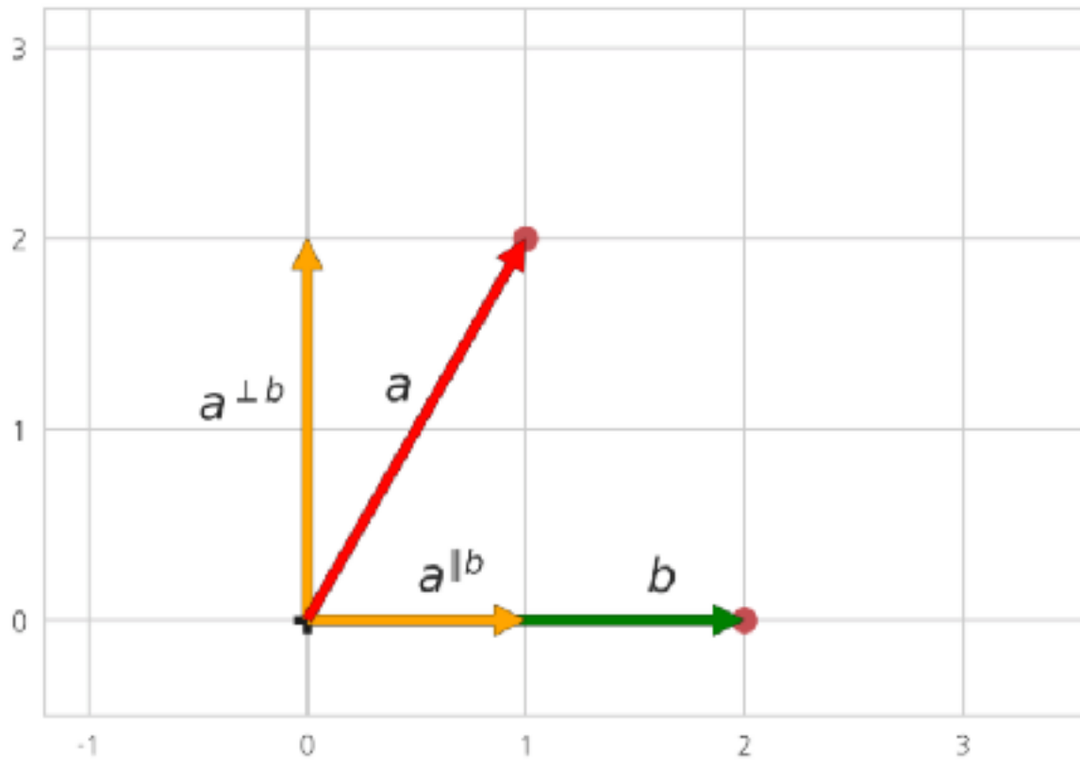
투영

- **사전적 의미**
  - 물체의 그림자를 어떤 물체 위에 비추거나 거기에 비친 그림자를 뜻하는 말.
- 벡터 a를 다른 벡터 **b에 직교하는 성분**과 **벡터 b에 평행한 성분**으로로 분해할 수 있는데,
- 벡터 b에 **수직인 성분**을 벡터 b에 대한 **리젝션 벡터(rejection vector)**,
- **평행한 성분**을 벡터 b에 대한 **프로젝션 벡터(projection vector)**라고 한다.

$$||a^{||b}|| = ||a|| \cos\theta = \frac{||a||\,||b|| \cos\theta}{||b||} = \frac{a^T b}{||b||} = \frac{b^T a}{||b||}$$

$$a^{||b} = \frac{a^T b}{||b||}\,\frac{b}{||b||} = \frac{a^T b}{||b||^2} b$$

$$a^{\perp b} = a - a^{||b}$$



1. 선형 대수

벡터 내적과 행렬 곱

(1) 벡터 내적

- 벡터 간의 곱셈이라고 생각하면 된다.
- 두 벡터의 길이가 같아야 내적을 실행할 수 있다.
- 두 벡터의 곱은 같은 위치에 있는 원소들을 각각 곱한 다음에 그 값들을 다시 모두 더해서 하나의 스칼라값으로 만든다.

$$X=\begin{bmatrix}1\\3\end{bmatrix}, \qquad Y=\begin{bmatrix}1\\2\end{bmatrix}$$

$$X\cdot Y=1\times 1+3\times 2$$

$$x=(x_1,\ x_2,\ \dots\ ,\ x_n),\ y=(y_1,\ y_2,\ \dots\ ,\ y_n)$$

$$x\cdot y=x_1y_1+x_2y_2+\dots+x_ny_n$$

- 두 벡터의 크기(길이)와 벡터 사이의 각도 θ 로 정의할 수 있다.

$$X\cdot Y = ||X||_2\ ||Y||_2 \cos\theta$$

1. 선형 대수

벡터 내적과 행렬 곱

(2) 벡터 내적의 성질

$\vec{a} \cdot \vec{b} = \vec{b} \cdot \vec{a}$: 교환 법칙이 성립한다.

$(k\vec{a}) \cdot \vec{b} = \vec{a} \cdot (k\vec{b}) = k(\vec{a} \cdot \vec{b})$: 실수배는 자유롭게 이동 가능하다.

$\vec{a} \cdot (\vec{b} + \vec{c}) = \vec{a} \cdot \vec{b} + \vec{a} \cdot \vec{c}$: 분배 법칙이 성립한다.

주의할점

$(\vec{a} \cdot \vec{b}) \cdot \vec{c} \neq \vec{a} \cdot (\vec{b} \cdot \vec{c})$

1. 선형 대수

벡터 내적과 행렬 곱

(3) 행렬 곱

- 벡터의 곱을 이용해 행렬 곱을 정의할 수 있다.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11}+a_{12}b_{21}+a_{13}b_{31} & a_{11}b_{12}+a_{12}b_{22}+a_{13}b_{32} \\ a_{21}b_{11}+a_{22}b_{21}+a_{23}b_{31} & a_{21}b_{12}+a_{22}b_{22}+a_{23}b_{32} \\ a_{31}b_{11}+a_{32}b_{21}+a_{33}b_{31} & a_{31}b_{12}+a_{32}b_{22}+a_{33}b_{32} \\ a_{41}b_{11}+a_{42}b_{21}+a_{43}b_{31} & a_{41}b_{12}+a_{42}b_{22}+a_{43}b_{32} \end{bmatrix}$$

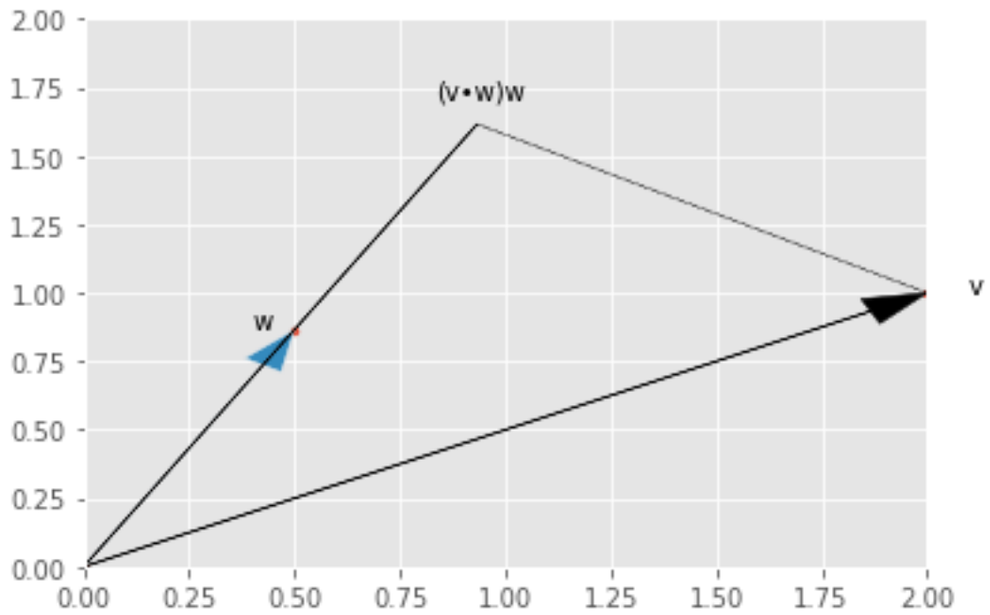
- n x n 행렬과 n x n 행렬의 곱 : n x n
- n x m 행렬과 m x n 행렬의 곱 : n x n
- 앞의 행렬의 **열의 수**와 뒷 행렬의 **행의 수**가 일치 해야지만 곱할 수 있다.

1. 선형 대수

보충설명

벡터 내적 (교재 54p)

- 내적은 벡터 v가 벡터 w 방향으로 얼마나 멀리 뻗어 나가는지를 나타낸다.



- w가 unit vector (||w|| = 1, 크기가 1)라면 (vw)w는 v가 w에 대해 투영된 벡터를 나타내고,
- vw (내적)은 v가 w에 투영된 벡터의 길이(크기)를 나타낸다.

2. 최적화

최적화(optimization)

주어진 함수(오차함수)의 최댓값 혹은 최솟값을 찾는 문제

2. 최적화

최적화 문제를 풀기 위한 기본 개념

미분공식

(1) 상수

- 상수를 미분하면 0이 된다.
- $\frac{d}{dx}(c) = 0$

(2) 거듭제곱

- x의 n 제곱을 미분하면 $n-1$ 제곱으로 제곱수가 1씩 감소한다.
- 이 공식은 n이 자연수이거나 음의 정수일 때 성립한다. $n = 0$일 때는 성립하지 않는다.
- $\frac{d}{dx}(x^n) = nx^{n-1}$

2. 최적화

최적화 문제를 풀기 위한 기본 개념

미분공식

(3) 로그

- 로그함수를 미분하면 x^{-1} 이 된다.
- $\frac{d}{dx}(\log x) = \frac{1}{x}$

(4) 지수

- 밑이 오일러 수인 지수함수는 미분해도 변하지 않는다.
- $\frac{d}{dx}(e^x) = e^x$

2. 최적화

최적화 문제를 풀기 위한 기본 개념

미분공식

(5) 덧셈

$$\frac{d}{dx}[f(x)+g(x)] = f'(x)+g'(x)$$

(6) 곱셈

$$\frac{d}{dx}[f(x)g(x)] = f(x)g'(x)+g(x)f'(x)$$

(7) 체인룰 (chain rule)

$$\frac{d}{dx}[f(g(x))] = f'(g(x))g'(x)$$

2. 최적화

최적화 문제를 풀기 위한 기본 개념

편미분 (partial differentiation)

- 다변수 함수에서 특정 변수를 제외한 나머지 변수를 상수로 생각하여 미분하는 방식.

(1) 편미분 표기 방법

$$f_x(x,y) = \frac{\partial f}{\partial x}$$

$$f_y(x,y) = \frac{\partial f}{\partial y}$$

(2) 편미분 간단 예시

$$f(x,y) = x^2 + 4xy + 4y^2$$

$$f_x(x,y) = \frac{\partial f}{\partial x} = 2x + 4y$$

$$f_y(x,y) = \frac{\partial f}{\partial y} = 4x + 8y$$

2. 최적화

최적화 문제를 풀기 위한 기본 개념

행렬 미분

(1) 스칼라를 벡터로 미분하는 경우

- 데이터 분석에서는 보통 종속변수(**y**)가 한개이고, 독립변수(**x**)가 여러개인 경우(**다차원 벡터**)를 많이 다룬다.

$$\nabla y = \frac{\partial y}{\partial x} = \begin{bmatrix} \frac{\partial y}{\partial x_1} \\ \frac{\partial y}{\partial x_2} \\ \vdots \\ \frac{\partial y}{\partial x_N} \end{bmatrix}$$

(2) 벡터를 스칼라로 미분하는 경우

- 함수 y가 다차원이고, x가 스칼라인 경우

$$\frac{\partial y}{\partial x} = \begin{bmatrix} \frac{\partial y_1}{\partial x} & \frac{\partial y_2}{\partial x} & \cdots & \frac{\partial y_M}{\partial x} \end{bmatrix}$$

2. 최적화

최적화 문제를 풀기 위한 기본 개념

행렬 미분

(3) 벡터 미분 규칙 1

- 선형 모형을 미분하면 가중치 벡터가 된다.

$$\frac{\partial w^T x}{\partial x} = \frac{\partial x^T w}{\partial x} = w$$

(4) 벡터 미분 규칙 2

$$\frac{\partial x^T A x}{\partial x} = (A + A^T)x$$

2. 최적화

최적화 문제를 풀기 위한 기본 개념

행렬 미분

(5) 벡터를 벡터로 미분하는 경우

- 자코비안 행렬 (Jacobian Matrix)

$$J=\frac{dy}{dx}=\begin{bmatrix}\frac{\partial y_1}{\partial x}^T\\ \vdots \\ \frac{\partial y_M}{\partial x}^T\end{bmatrix}=\begin{bmatrix}\nabla y_1^T\\ \nabla y_2^T\\ \vdots \\ \nabla y_M^T\end{bmatrix}=\begin{bmatrix}\frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_N}\\ \vdots & \ddots & \vdots \\ \frac{\partial y_M}{\partial x_1} & \cdots & \frac{\partial y_M}{\partial x_N}\end{bmatrix}$$

(6) 다변수 함수의 2차 도함수

- 그래디언트 벡터를 독립변수로 미분
- 헤시안 행렬 (Hessian Matrix)

$$H=\begin{bmatrix}\frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_2 \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_N \partial x_1}\\ \frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_N \partial x_2}\\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_N} & \frac{\partial^2 f}{\partial x_2 \partial x_N} & \cdots & \frac{\partial^2 f}{\partial x_N^2}\end{bmatrix}$$

2. 최적화

최적화 문제를 풀기 위한 기본 개념

그래디언트 (Gradient)

- 다변수 함수의 편미분 벡터

(1) 그래디언트 표기 방법

$$\nabla f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2} \cdots \frac{\partial f}{\partial x_n} \right)$$

- 각 변수의 1차 편미분 값들로 구성된 벡터

(2) 그래디언트 의미

- **산이나 언덕**을 가정해보자. **어떤 지점(x,y)**에서의 **높이를 H(x,y)**로 표현하는 경우, 그래디언트는 가장 (위를 바라보는)경사가 가파른 방향과 그 경사의 크기를 나타낸다.
- **어떤 지점 : 스칼라, 가파른 경사의 방향과 크기 : 그래디언트**

2. 최적화

최적화 문제를 풀기 위한 기본 개념

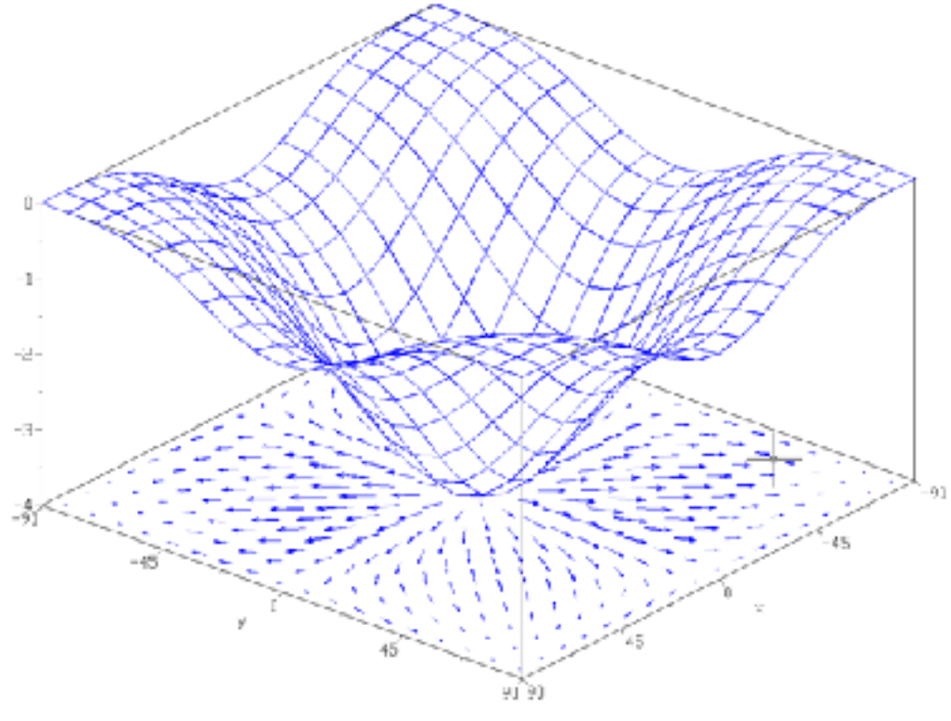
그래디언트 (Gradient)

- 다변수 함수의 편미분 벡터

(3) 그래디언트 벡터의 특징

- 그레디언트 벡터의 방향은 함수 곡면의 기울기가 가장 큰 방향, 즉 단위 길이당 함수 값(높이)이 가장 크게 증가하는 방향을 가리킨다.
- 그레디언트 벡터의 방향은 등고선(isoline)의 방향과 직교한다.
- 그레디언트 벡터의 크기는 기울기를 의미한다. 즉 벡터의 크기가 클 수록 함수 곡면의 기울기가 커진다.

(4) 그래디언트 시각화



2. 최적화

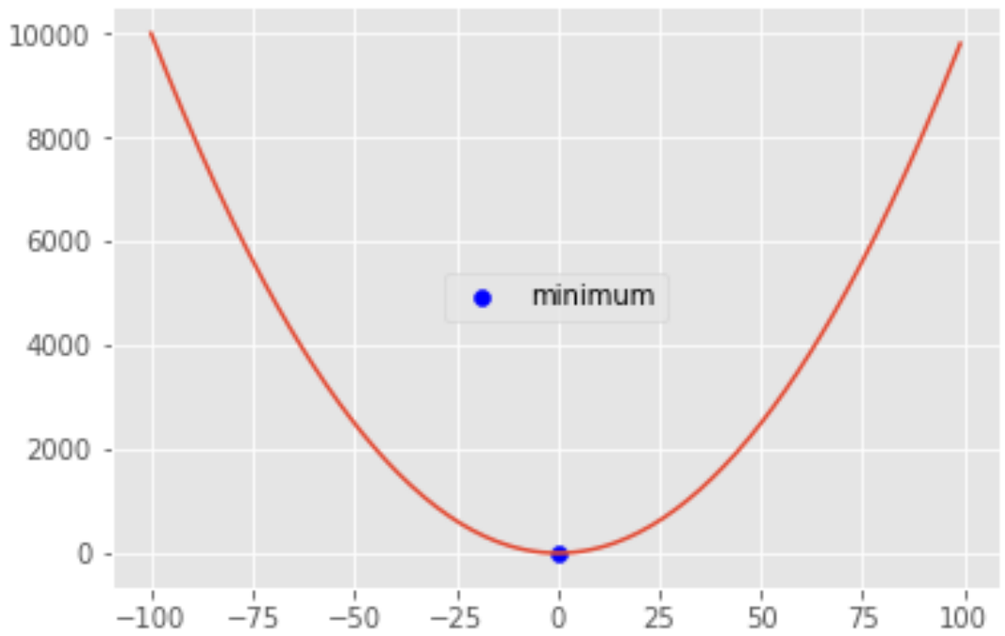
최적화 문제

최적화 문제

- 주어진 함수의 최솟값(혹은 최댓값)을 찾는 문제
  - 보통 최소화의 경우만 고려함.
  - 함수 f의 값을 최소화하는 x의 값 (x*)

$$x^* = arg\min_x f(x)$$

  - 최소화하려는 함수는 목적함수 (objective function, O), 비용함수 (cost function, C), 손실함수 (loss function, L) 라고 부른다.



2. 최적화

최적화 문제를 풀기 위한 방법

(1) 미분

- 편미분 벡터가 0인 지점은 최적점이다.

$$\nabla f = 0$$

- 이 조건은 최솟값이 되기위한 필요 조건이다.
- 기울기가 0이라고 반드시 최솟값이 되지는 않지만, 모든 최솟값은 기울기가 0이다.

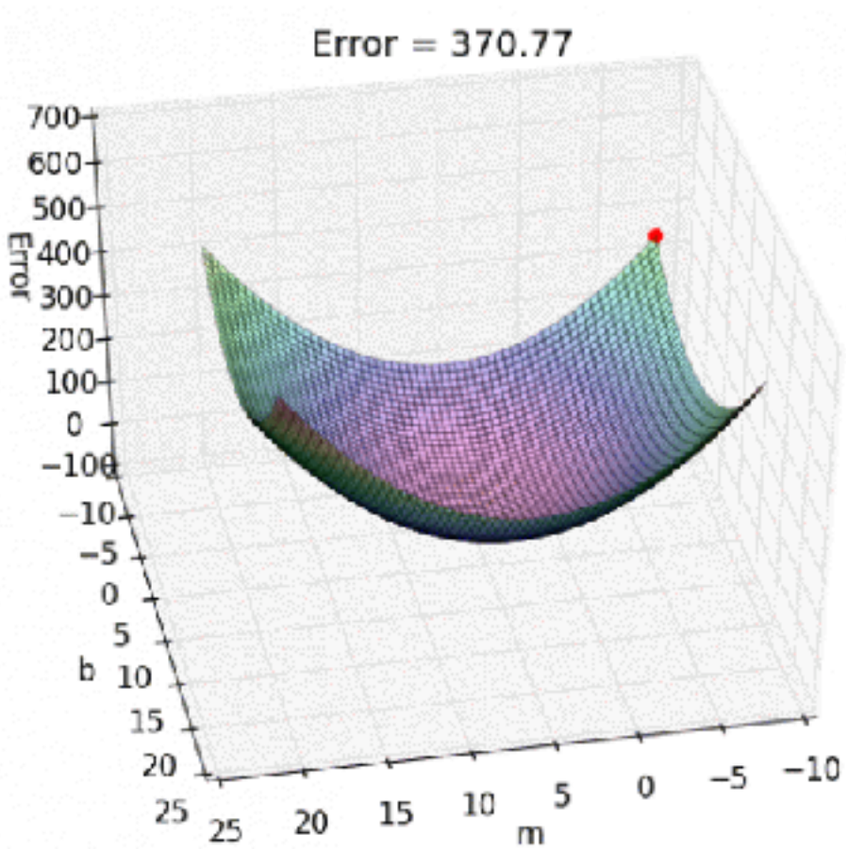
2. 최적화

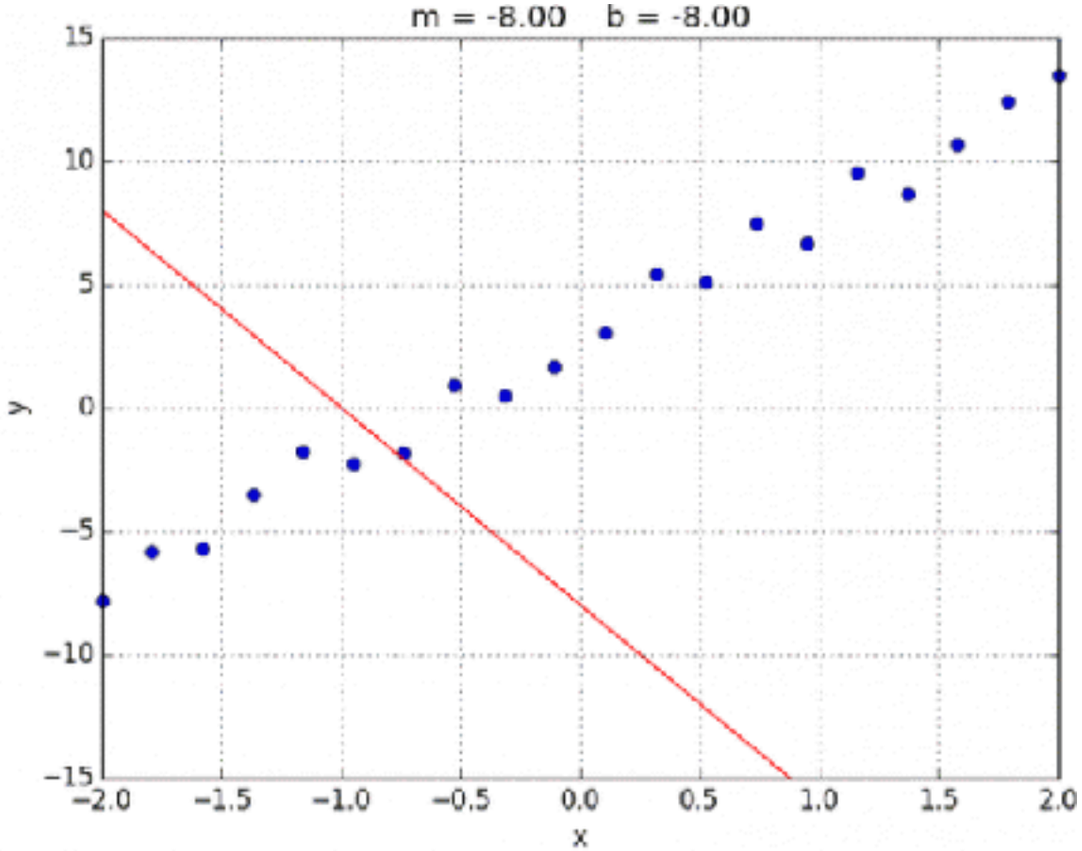
최적화 문제를 풀기 위한 방법

(2) 경사하강법(Gradient Descent)

- 반복적 시행 착오(trial and error)에 의해 최적화 필요조건을 만족하는 값을 찾는 방법인 수치적 최적화(numerical optimization) 방법 중 하나.

(최적화 필요조건 : 편미분값 = 0 (최대 혹은 최소가 되는 지점))





3. 차원축소

차원 축소 (Dimensionality Reduction)

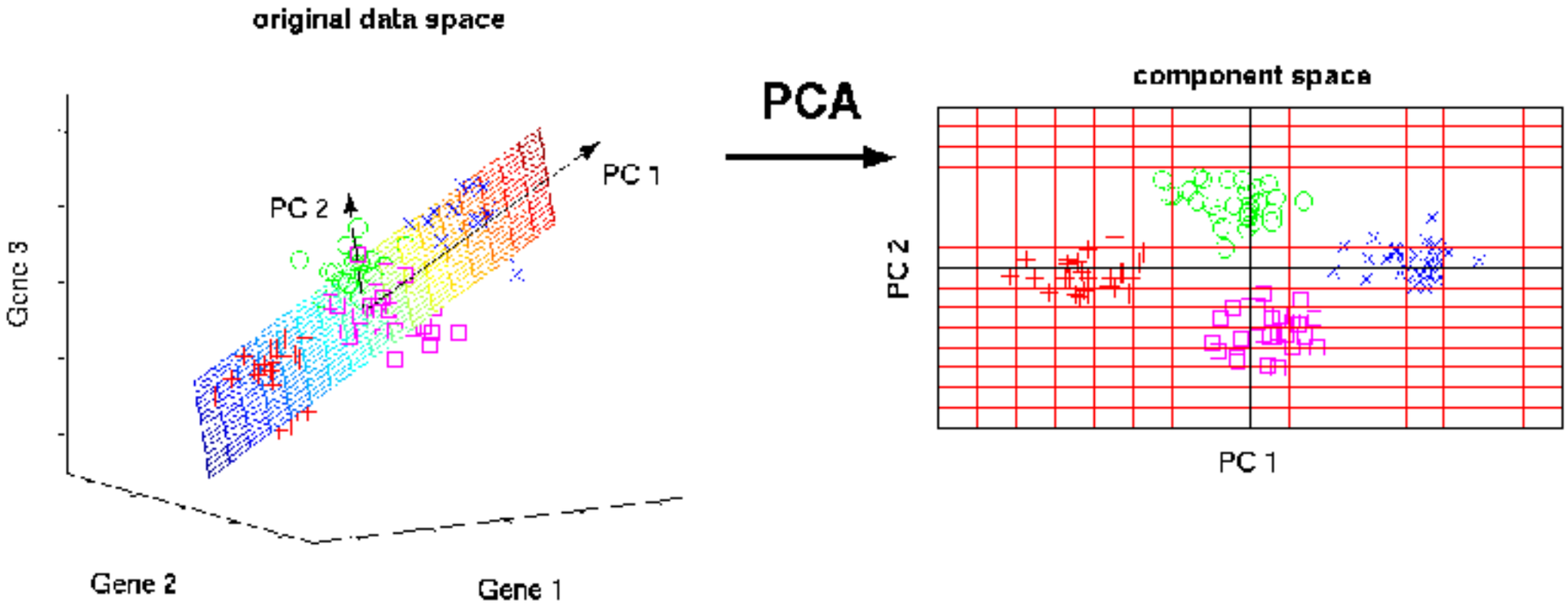
데이터의 의미를 제대로 표현하는 특징을 추려내는 것

3. 차원축소

주성분 분석 (PCA)

주성분분석은 **데이터의 분포**를 가장 잘 표현하는 성분을 찾아주는 것이다.

- 실제 우리가 관찰한 특징들 (data space)중 에서 숨겨져있는 진짜 특징 (latent space)를 찾는 것.
- 데이터의 분포를 가장 잘 표현하는 성분 = 분산이 가장 큰 벡터



3. 차원축소

Mathematical Background

1. **공분산 행렬**
   - 벡터의 각 차원 간의 공분산을 나타낸 행렬
   - 평균을 중심으로 각 자료들이 어떻게 분포되어 있는지 크기와 방향성을 같이 보여준다

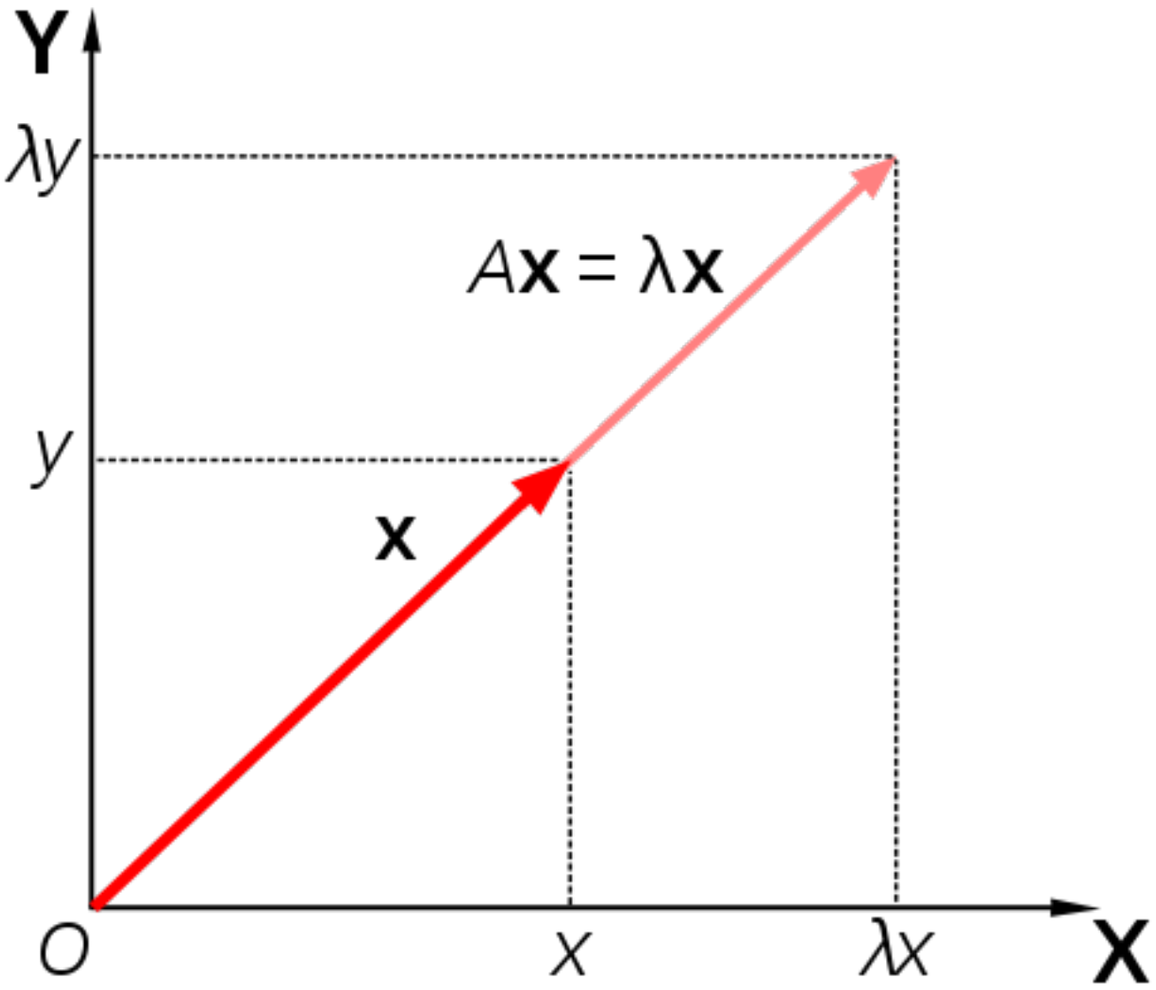
$$cov(x,\ y) = E[(X-\overline{X})(Y-\overline{Y})]$$

2. **Eigen vector (고유벡터)**
   - 어떤 행렬(A)와의 선형 변환 (행렬곱)의 결과가 자기 자신의 상수배가 되는 0이 아닌 벡터

3. **Eigen value (고유값) : 그때의 상수배**
   - Ax = λx 가 성립할 때, (A = 정방행렬 (n x n))
     - **λ : A의 eigen value**
     - **x : λ에 대응되는 eigen vector**
       - x와 λ는 최대 n개까지 존재할 수 있다.

3. 차원축소

Mathematical Background



- λ : A의 eigen value
- x : λ에 대응하는 eigen vector

- 행렬 A의 eigen vector는 선형 변환 A에 의해 방향은 보존되고 스케일 (scale)만 변화되는 방향 벡터를 나타내고, eigen value은 그 eigen vector의 변화되는 스케일 정도를 나타내는 값이다.

3. 차원축소

Mathematical Background

4. **RSS (Residual Sum of Square)**
   - 실제 값과 예측 값의 제곱합
     - 잔차 제곱합이라고도 부른다.

$$\sum_{i}^{n}(y-\hat{y})^2$$

5. **제약 조건하에 최적화**
   - **라그랑쥬 승수법**
     - 원 함수 (f)에 제약식 (g) 을 추가하여 최적값을 구하는 방법
     - 제약식이 추가된 함수에 각 변수로 미분한 값과 람다로 미분한 값을 통해 최적값을 도출한다.

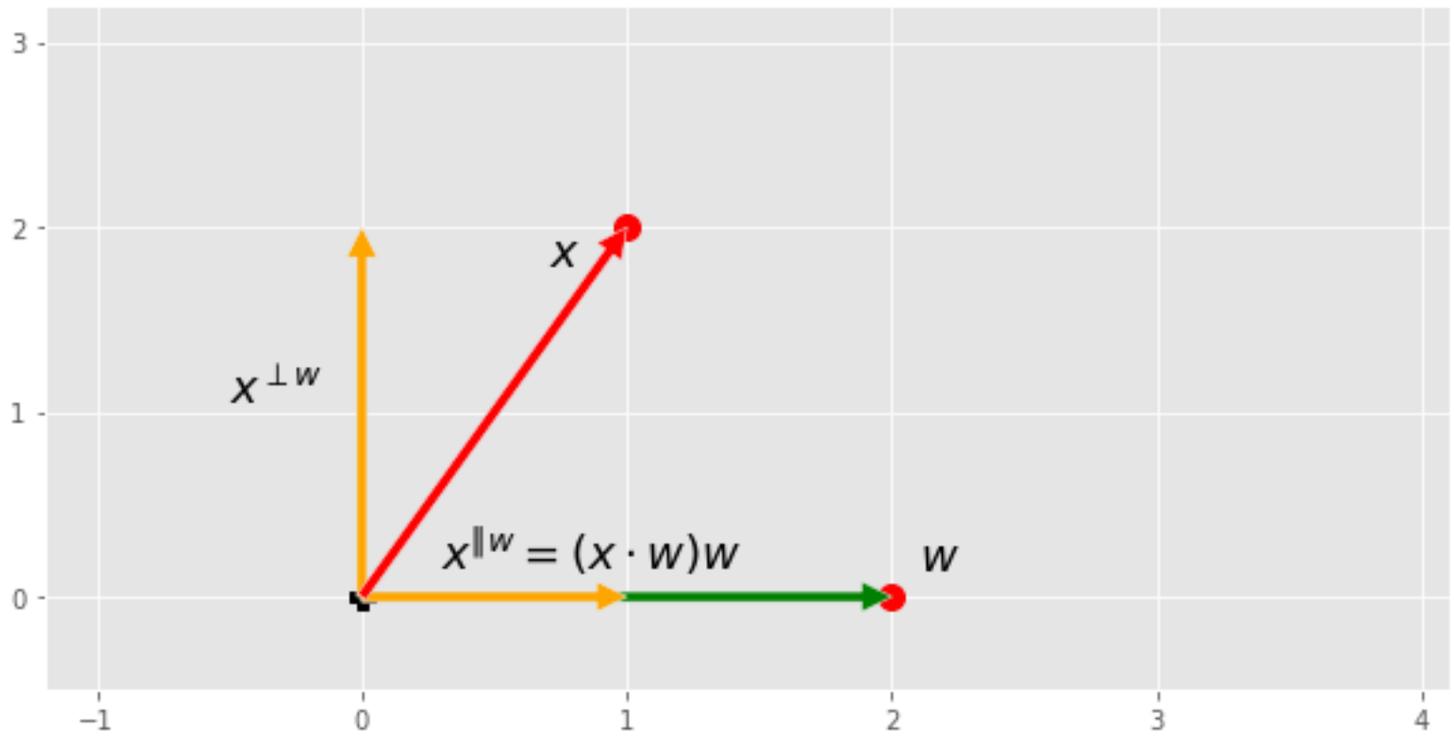
$$f=x^2+y^2$$
$$g=x+y-2=0$$
$$L=f-\lambda(g)=x^2+y^2-\lambda(x+y-2)$$

3. 차원축소

PCA mathematics

Minimize RSS

- 차원 축소는 차원이 축소된 뒤, 원래 벡터가 가지는 정보를 얼마나 잘 가지고 있는지에 관심이 있다.
  - 원 벡터 x가 w에 대해 투영되었을 때, 투영된 프로젝션 벡터가 x를 얼마나 잘 표현하는가?
    - 프로젝션 벡터는 x의 예측값이 된다.
    - 원벡터와 프로젝션 벡터의 차이는 residual이 된다.
    - RSS !



3. 차원축소

PCA mathematics

Minimize RSS

- 전체 평균이 0인 임의의 벡터 x 하나와 어떤 방향을 가지는 unit vector w를 가정하자.
- x와 프로젝션 벡터와의 차이는 다음과 같이 정의한다.

$$|| \vec{x_i} - (\vec{x_i} \cdot \vec{w}) \vec{w} ||^2 = || \vec{x_i} ||^2 - (\vec{x_i} \cdot \vec{w})^2$$

- 이를 바탕으로 RSS를 도출할 수 있다.

$$RSS = \sum_i^n (|| \vec{x}_i ||^2 - (\vec{x}_i \cdot \vec{w})^2)$$

$$= \sum_i^n (|| \vec{x}_i ||^2) - \sum_i^n ((\vec{x}_i \cdot \vec{w})^2)$$

- RSS에서 w에 영향을 받는 것은 마지막 항이다.
- 즉, RSS를 최소화 시키기 위해서는 마지막 항을 최대화 시켜야된다.

3. 차원축소

PCA mathematics

Minimize RSS

- 마지막항을 조금 변형 시켜보자.

$$\sum_{i}^{n} (\vec{x}_i \cdot \vec{w})^2 \quad -> \quad \frac{1}{n} \sum_{i}^{n} (\vec{x}_i \cdot \vec{w})^2$$

- 이제 $\frac{1}{n} \sum_{i}^{n} (\vec{x}_i \cdot \vec{w})^2$ 을 최대화 하는 문제로 바뀌었다.

- 그런데, **분산 = 제곱의 평균 - 평균의 제곱**
- **제곱의 평균 = 평균의 제곱 + 분산**
- x의 평균이 0이므로 우리가 최대화 시켜야되는 $\frac{1}{n} \sum_{i}^{n} (\vec{x} \cdot \vec{w})^2$ 는 **분산**이 된다.

$$\frac{1}{n} \sum_{i}^{n} (\vec{x}_i \cdot \vec{w})^2 = (\frac{1}{n} \sum_{i}^{n} \vec{x}_i \cdot \vec{w})^2 + Var[\vec{x}_i \cdot \vec{w}]$$

3. 차원축소

PCA mathematics

Maximize Variation

$$\sigma^2_{\vec{w}} = \frac{1}{n}\sum_i^n (\vec{x_i} \cdot \vec{w})^2$$

$$= \frac{1}{n}(\mathbf{Xw})^T(\mathbf{Xw})$$

$$= \frac{1}{n}\mathbf{w}^T\mathbf{X}^T\mathbf{Xw}$$

$$= \mathbf{w}^T\frac{\mathbf{X}^T\mathbf{X}}{n}\mathbf{w}$$

$$= \mathbf{w}^T\mathbf{Vw}$$

$$C=\begin{pmatrix} cov(x,x) & cov(x,y) \\ cov(x,y) & cov(y,y) \end{pmatrix}$$

$$=\begin{pmatrix} \frac{1}{n}\sum(x_i-m_x)^2 & \frac{1}{n}\sum(x_i-m_x)(y_i-m_y) \\ \frac{1}{n}\sum(x_i-m_x)(y_i-m_y) & \frac{1}{n}\sum(y_i-m_y)^2 \end{pmatrix}$$

- V는 X벡터의 **공분산 행렬**이 된다.
- 분산을 w에 대한 함수라고 하면, ||w||=1 이라는 제약 조건하에 최적화를 하는 문제와 같아진다.
- 우리는 제약 조건이 있는, w와 λ에 대한 함수 u를 얻을 수 있다.

$$u(w,\lambda) = f(w) - \lambda(g(w) - c)$$

3. 차원축소

PCA mathematics

최적화

$$u(w,\lambda) = f(w) - \lambda(g(w) - c)$$

- 라그랑쥬 승수법을 활용해 최적값을 찾는다.

$$\frac{\partial u}{\partial w} = 0 = \frac{\partial f}{\partial w} - \lambda \frac{\partial g}{\partial w}$$

$$\frac{\partial u}{\partial \lambda} = 0 = -(g(w) - c)$$

$$\begin{aligned} u &= \mathbf{w}^T\mathbf{V}\mathbf{w} - \lambda(\mathbf{w}^T\mathbf{w} - 1) \\ \frac{\partial u}{\partial \mathbf{w}} &= 2\mathbf{V}\mathbf{w} - 2\lambda\mathbf{w} = 0 \\ \mathbf{V}\mathbf{w} &= \lambda\mathbf{w} \end{aligned}$$

$$\frac{w^T V w}{\partial w} = (V^T + V)w$$

$$\frac{\lambda w^T w}{\partial w} = 2\lambda w$$

3. 차원축소

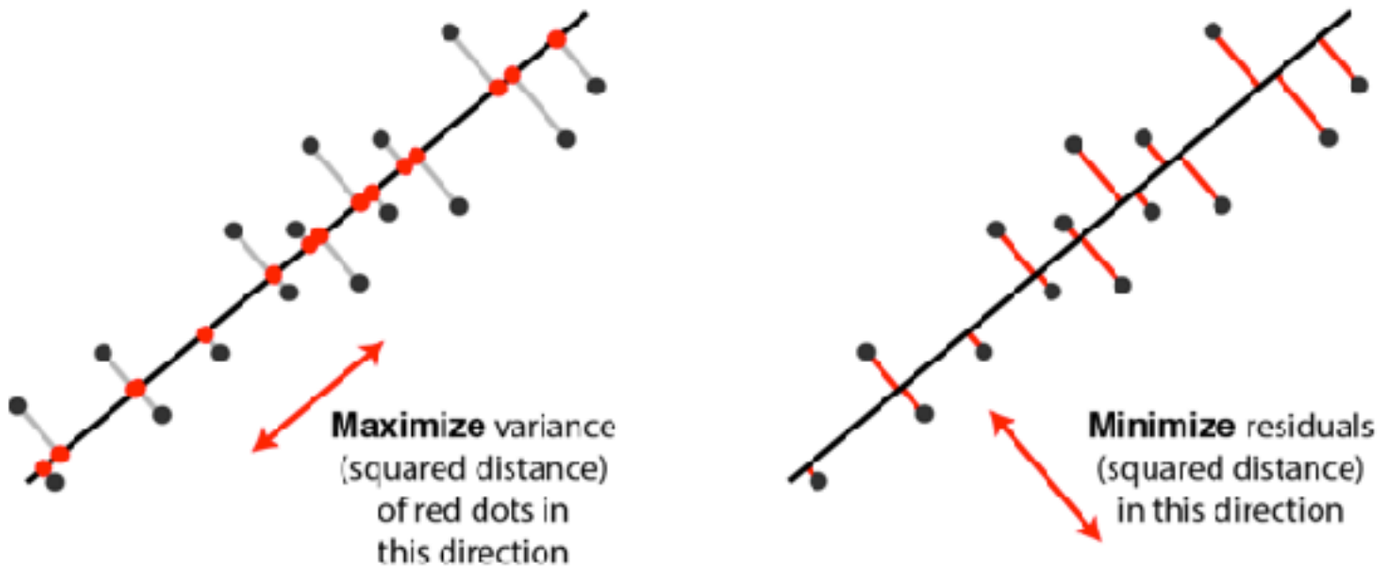
PCA mathematics

최적화

- w는 X의 공분산 행렬 V의 eigen vector가 된다.

$$\mathbf{V}\mathbf{w} = \lambda\mathbf{w}$$

- 따라서, 원벡터와 프로젝션 벡터의 차이를 최소화 시키는 w는 X의 eigen vector를 구하면 된다.



Two equivalent views of principal component analysis.

4. 선형회귀 기초

선형회귀 기초 (Linear Regression)

4. 선형회귀 기초

예측 문제의 수학적 표현

예측 모형은 입력 데이터 X를 받아서 출력 데이터Y 를 만든다는 점에서 **수학의 함수(function)**와 유사하다.

예측 문제의 최종 목표는 **X와 Y의 관계 함수 f(X)** 를 구하는 것이다.

$$Y=f(X) \qquad Y=a_1x_1+a_2x_2+a_3x_3+a_4x_4+....$$

하지만 현실적으로는 데이터의 개수가 너무 많고, x 변수의 개수가 한정적인 경우가 많으므로 정확한 f를 구할 수 없다.

(**연립 방정식**을 생각하면 쉽다 !)

따라서, 우리는 정확한 f 보다는 f와 가장 유사하고, 재현 가능한 $\hat{f}$ 를 구하는 것에 집중한다.

이러한 $\hat{f}$ 을 가지고 있다면, 그리고 f와 아주 유사하다면, 새로운 데이터가 들어왔을 때,

실제 y와 아주 유사한 $\hat{y}$ 을 구할 수 있다.

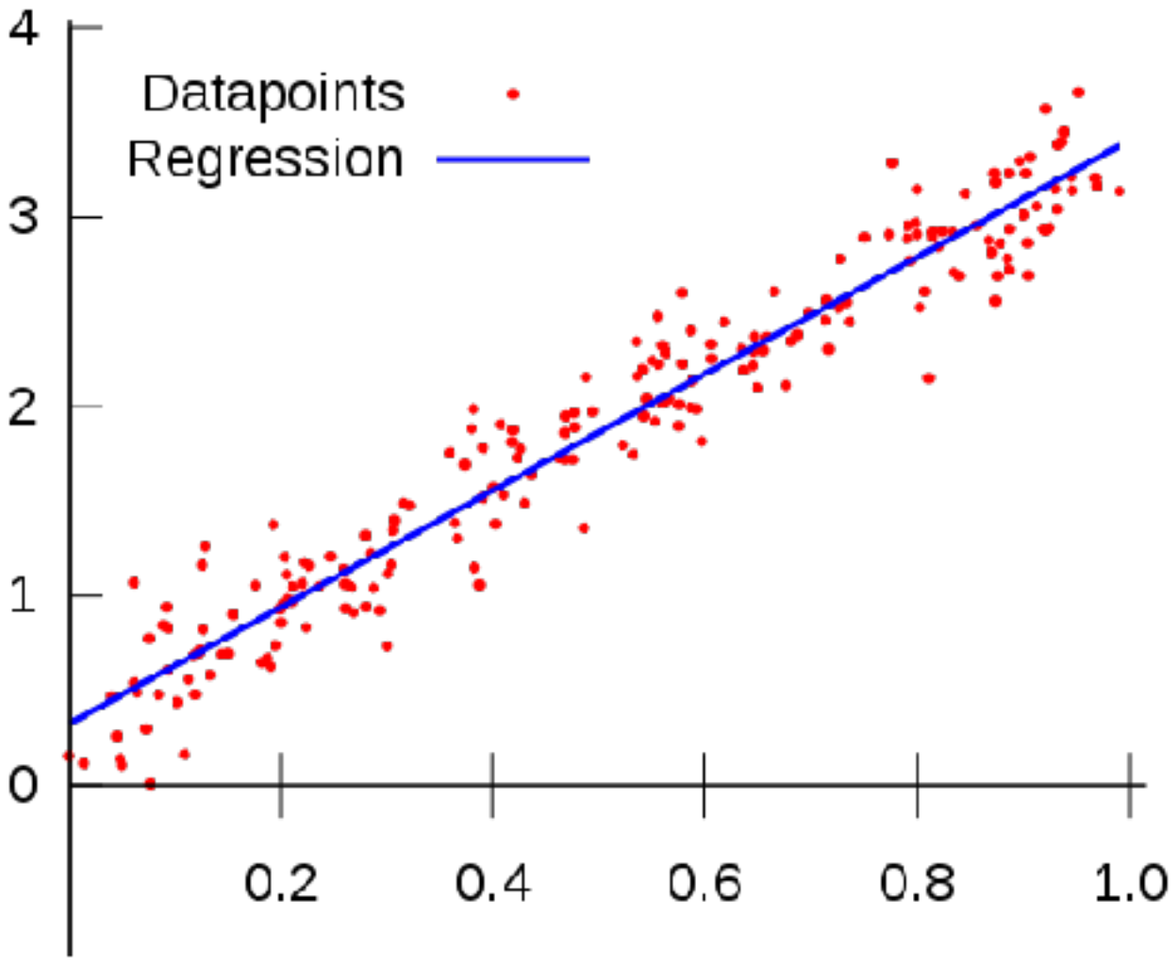
$$Y \approx \hat{f}(X)$$

4. 선형회귀 기초

선형회귀

수치형 **설명변수 X**와 **연속형 숫자**로 이뤄진 **종속변수 Y**간의 관계를 **선형**으로 가정하고
이를 가장 잘 표현할 수 있는 **회귀계수**를 데이터로부터 추정하는 모델

- **예시 : 집 크기 (x)에 대한 집 값 (y) 예측**



$$Y = Xw$$

4. 선형회귀 기초

선형회귀식의 표기

$$Y=Xw$$

- 설명 변수 x가 D개가 있고, 데이터가 N개가 있을 때, x에 대한 전체 데이터를 오른쪽과 같이 (D X N) 행렬로 표기가 가능하다.

$$x_i=\begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{iD} \end{bmatrix} \rightarrow X=\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1D} \\ x_{21} & x_{22} & \cdots & x_{2D} \\ \vdots & \vdots & \vdots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{ND} \end{bmatrix}$$

- 전체 수식은 설명 변수 벡터(x)와 가중치 벡터(w)의 내적으로 간단하게 나타낼 수 있다.

$$f(x)=w_1x_1+w_2x_2+\cdots+w_Dx_D=\begin{bmatrix} x_1 & x_2 & \cdots & x_D \end{bmatrix}\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_D \end{bmatrix}=x_a^Tw_a=w_a^Tx_a$$

4. 선형회귀 기초

목적함수

RSS (Residual Sum of Squares)

- 실제값 - 예측값

$$\hat{y} = Xw$$

$$e = y - \hat{y} = y - Xw$$

$$\begin{aligned} RSS &= e^T e \\ &= (y - Xw)^T (y - Xw) \\ &= y^T y - 2y^T Xw + w^T X^T Xw \end{aligned}$$

- 선형회귀분석도 마찬가지로 RSS 를 minimize 하기 위한 w 를 구하는 것이다.

4. 선형회귀 기초

최적화

- RSS의 최솟값을 구하기 위해 그레디언트 벡터를 구한다. (RSS를 w에 대해 미분한다)

$$\frac{dRSS}{dw} = -2X^T y + 2X^T Xw$$

- RSS가 최소가 되는 최적화 조건은 그레디언트 벡터가 0벡터이어야 하므로 다음 식이 성립한다.

$$\frac{dRSS}{dw} = 0$$

$$X^T Xw^* = X^T y$$

- X^TX의 역행렬이 존재한다면 최적 가중치 벡터 w를 구할 수 있다.

$$w^* = (X^T X)^{-1} X^T y$$

- 이와 같은 방식으로 명시적 해를 단번에 추정할 수 있다.

다음에
또!
같이!
만나요!

learnit Flip your study