3.7. Inner Product of Functions 함수의 내적

-지금까지 내적의 속성을 보면서 길이, 각, 거리를 계산했음

-유한한 차원의 벡터들의 내적에 집중했음

-이번 챕터에서는 함수들의 내적에 대해 알아보겠음

-지금까지 논의한 내적은 유한한 숫자의 입력으로 만들어지는 벡터임

벡터 x ∈ R^n (n 함수값을 가지는 함수)

-'내적'이라는 컨셉은 무한한 숫자의 입력의 벡터와 continuously-valued functions,

즉 셀 수 있게 무한하고 셀 수 없게 무한한 벡터로 일반화될 수 있음

-그러면 벡터의 개별 구성 요소의 합은 적분이 됨

-두 개의 함수 u: R->R 과 v: R->R 의 내적은 아래의 정적분으로 정의될 수 있음

$$\langle u, v \rangle := \int_a^b u(x)v(x)dx$$

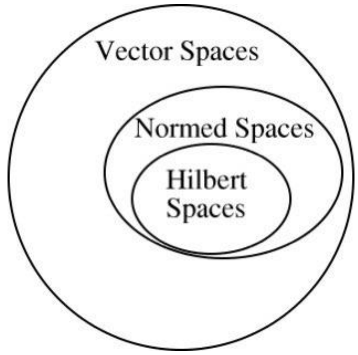
-우리의 일반적인 내적으로는 norms(벡터의 길이)와 orthogonality(직교)를 정의할 수 있음

-만약 위의 정적분이 0 이 된다면 함수 u 와 v 는 orthogonal 함

-선행하는 내적을 수학적으로 정확하게 하려면, 우리는 적분의 measures(측도)와 정의를 고려해야 하고, 'Hilbert space' (힐베르트 공간)로 나아가게 됨

*힐베르트 공간: -벡터를 복소 함수로 확장한 공간으로서, 편미분 방정식, 양자역학, 푸리에 분석 등에서 매우 중요

-선형 벡터 공간으로 벡터끼리 서로 더하거나 스칼라로 곱할 수 있는 벡터의 모음을 말하며 벡터를 복소 함수로 확장한 공간임



-더하여 유한한 차원의 내적과는 달리, 함수의 내적은 발산할 수 있음

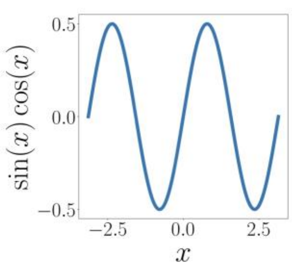
Example 3.9

-u=sin(x), v=cos(x)라 가정할 때, 적분 f(x)=u(x)v(x)는 기함수임 f(-x) = -f(x)

-그래서 a=-pi, b=pi 일 때 0 이 됨

-그리하여 sin 과 cos 은 orthogonal function(직교 함수)임

Figure 3.8 $f(x) = \sin(x)\cos(x)$.



3.8 Orthogonal Projections (직교 투영)

-Projection(투영)은 linear transformation(선형 변환)에서 중요한 클래스. graphics 에서 중요한 역할을 함(이론, 통계, 머신러닝을 만듦/구현/다룸)

-머신러닝에서는 고차원의 데이터를 주로 다룸. 고차원 데이터는 분석하거나 시각화하기 어려움.

-하지만 고차원의 데이터를 살펴보면 몇 차원에 대부분의 정보가 있고, 대부분의 차원은 데이터의 중요한 속성을 설명하기에 중요하지 않음

-고차원의 데이터를 압축하거나 시각화할 때, 정보를 잃게 됨. 그래서 정보를 최소한으로 잃기 위해 가장 정보를 많이 담고 있는 차원을 찾으려고 함

-Ch.1 에서 언급한 것처럼 데이터는 벡터로 표현될 수 있고, 이번 챕터에서는 데이터 압축을 위한 근본적인 도구에 대해 논의할 예정임

-구체적으로, 원본의 고차원 데이터를 저차원 데이터 feature map(특징 공간)에 프로젝션하여 이 비교적 저차원 공간을 활용하여 데이터셋에 대해서 더 알아보고 관련 패턴을 추출할 수 있음

-머신러닝 알고리즘(PCA, DNN Auto-encoder 등)은 차원 축소라는 아이디어를 무게 있게 활용함

-이번에는 직교 프로젝션에 대해서 배우고, Ch.10 에서는 linear dimensionality reduction, Ch.12 에서는 classification 에 대해서 배움. Ch9 의 linear regression 도 직교 프로젝션을 가지고 해석할 수 있음

Definiton 3.10 (Projection). V 를 vector space, U 를 subspace of V 라 하자.

Linear mapping(선형사상/선형변환) pi: V -> U 는 pi^2=pi*pi=pi 일 때 projection(투영)이라고 불림

< Properties of projection matrix >

$$\overset{A}{\begin{bmatrix}1&1\\2&3\end{bmatrix}}\overset{x}{\begin{bmatrix}4\\7\end{bmatrix}} = 4\overset{col1}{\begin{bmatrix}1\\2\end{bmatrix}} + 7\overset{col2}{\begin{bmatrix}1\\3\end{bmatrix}} = \overset{b}{\begin{bmatrix}11\\29\end{bmatrix}}$$

x를 곱하는 것: A의 column space에 안착(landing)하는 것

∵ 결국 column vector의 선형조합으로 표현되기 때문.

어떤 두 벡터를 column vector × row vector로의 순서로 곱하면

(1×n) (n×1)

반드시 rank 1 행렬이 만들어진다.

∴ projection b onto a

(P : proj matrix)

$p = Pb$, $P = \dfrac{aa^T}{a^Ta}$

식에 나온 것처럼 column × row(aa^T)의 곱으로 만들어졌기 때문에

rank : 1

벡터 a가 행렬 P의 column space의 기저(basis)가

$\boxed{P^T = P}$

① 투영행렬은 대칭인가? OO

똑같은 벡터A의 column × row 순으로 곱하여 만들어진 행렬이기 때문에 대칭행렬이다.

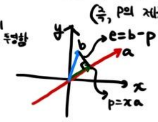
symmetric matrix

$\boxed{P^2 = P}$

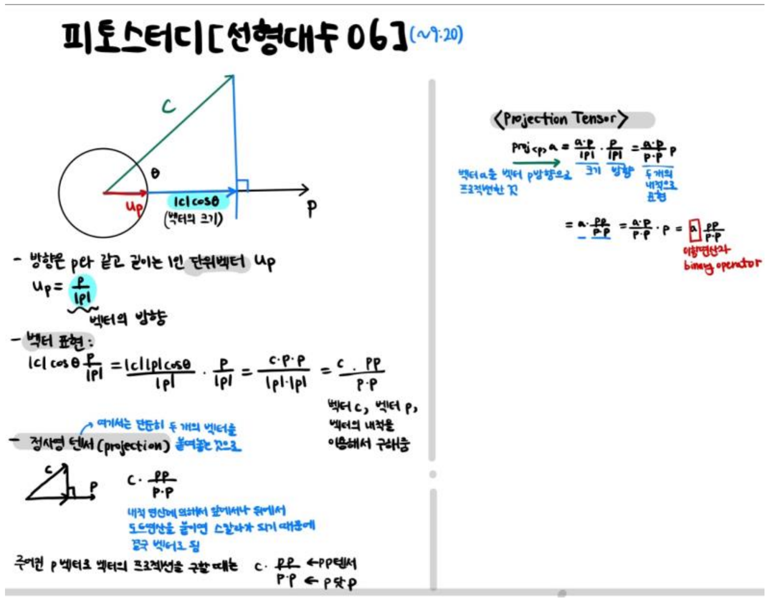
② 벡터 b를 투영 행렬 P에 두 번 투영시킨다면? Same

(즉, P의 제곱에 b를 곱하는 경우)

벡터a에 벡터 b를 투영함



3.8.1 Projection onto One-Dimensional Subspaces (Lines) 1 차원(선)에 프로젝션)



6.3 Orthogonal and orthonormal vectors

DEFINITION. *We say that 2 vectors are orthogonal if they are perpendicular to each other. i.e. the dot product of the two vectors is zero.*

https://www.ucl.ac.uk/~ucahmdl/LessonPlans/Lesson10.pdf

- 프로젝션은 c 에 제일 가까운데, 가깝다는 말은 ||프로젝션-c|| 거리가 가장 작다는 것을 의미

- '프로젝션-c' 세그먼트는 subspace(부분공간)에 직교하고, 그러므로 기저 벡터 b 임

-직교 조건은 벡터 간의 각은 내적으로 정의되기 때문에 ⟨프로젝션 - c, b⟩ = 0 을 산출해냄

-c 를 부분공간에 직교하는 프로젝션은 반드시 부분집합의 요소가 되어야 하고, 그리하여 부분집합에서 span 하는 기저 벡터 b 의 배수도 마찬가지다.

프로젝션 = λb, $\lambda \in R$

-좌표 λ, 프로젝션 ∈부분공간, 프로젝션 매트릭스을 정함

3.8.2 Projection onto General Subspaces

더 낮은 차원의 부분공간에 정사영(Orthogonal projection)함

프로젝션으로 Ax=b 해결, b 는 A 의 span 에 있는 것이 아님. B 는 a 의 칼럼으로 spanned 된 부분공간에 놓여져 있지 않음.

솔루션: 최소제곱해(least squares solution)

3.8.3 Gram-Schmidt Orthogonalization

Remark. We just looked at projections of vectors $\boldsymbol{x}$ onto a subspace U with basis vectors $\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k\}$. If this basis is an ONB, i.e., (3.33) and (3.34) are satisfied, the projection equation (3.58) simplifies greatly to

$$\pi_U(\boldsymbol{x}) = \boldsymbol{B}\boldsymbol{B}^\top \boldsymbol{x} \tag{3.65}$$

since $\boldsymbol{B}^\top \boldsymbol{B} = \boldsymbol{I}$ with coordinates

$$\boldsymbol{\lambda} = \boldsymbol{B}^\top \boldsymbol{x}\,. \tag{3.66}$$

This means that we no longer have to compute the inverse from (3.58), which saves computation time. ◇

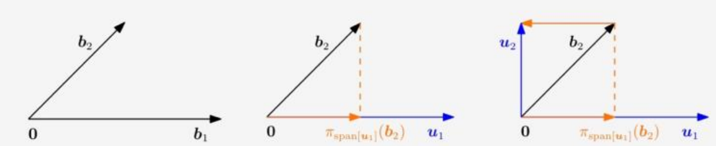
주어진 벡터들에 대한 직교기저(orthogonal basis) 또는 정규직교기저(orthonormal basis)를 구하는 과정₩

그람-슈미트 직교화(Gram-Schmidt orthogonalization): 주어진 벡터 v1, v2, ... 로부터 이 벡터들을 생성할 수 있는 직교기저(orthogonal basis)를 구하는 과정

그람-슈미트 정규직교화(Gram-Schmidt orthonormalization): 주어진 벡터 v1, v2, ... 로부터 이 벡터들을 생성할 수 있는 정규직교기저(orthonormal basis)를 구하는 과정

3.8.4 Projection onto Affine Subspaces

Example 3.12 (Gram-Schmidt Orthogonalization)



(a) Original non-orthogonal basis vectors $\boldsymbol{b}_1, \boldsymbol{b}_2$.

(b) First new basis vector $\boldsymbol{u}_1 = \boldsymbol{b}_1$ and projection of $\boldsymbol{b}_2$ onto the subspace spanned by $\boldsymbol{u}_1$.

(c) Orthogonal basis vectors $\boldsymbol{u}_1$ and $\boldsymbol{u}_2 = \boldsymbol{b}_2 - \pi_{\text{span}[\boldsymbol{u}_1]}(\boldsymbol{b}_2)$.

Consider a basis $(\boldsymbol{b}_1, \boldsymbol{b}_2)$ of $\mathbb{R}^2$, where

$$\boldsymbol{b}_1 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \quad \boldsymbol{b}_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}; \tag{3.69}$$

see also Figure 3.12(a). Using the Gram-Schmidt method, we construct an orthogonal basis $(\boldsymbol{u}_1, \boldsymbol{u}_2)$ of $\mathbb{R}^2$ as follows (assuming the dot product as the inner product):

$$\boldsymbol{u}_1 := \boldsymbol{b}_1 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \tag{3.70}$$

$$\boldsymbol{u}_2 := \boldsymbol{b}_2 - \pi_{\text{span}[\boldsymbol{u}_1]}(\boldsymbol{b}_2) \overset{(3.45)}{=} \boldsymbol{b}_2 - \frac{\boldsymbol{u}_1\boldsymbol{u}_1^\top}{\|\boldsymbol{u}_1\|^2}\boldsymbol{b}_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{3.71}$$

3.8.4 Projection onto Affine Subspaces

When given an affine space L=x0 + U, b1 and b2 are basis vectors of U

To determine the orthogonal projection of pi(x) of x onto L, we transform the problem into **the problem of projection onto a vector space**

$$\pi_L(\boldsymbol{x}) = \boldsymbol{x}_0 + \pi_U(\boldsymbol{x} - \boldsymbol{x}_0)\,, \tag{3.72}$$

where $\pi_U(\cdot)$ is the orthogonal projection onto the subspace U, i.e., the direction space of L; see Figure 3.13(c).

From Figure 3.13, it is also evident that the distance of $\boldsymbol{x}$ from the affine space L is identical to the distance of $\boldsymbol{x} - \boldsymbol{x}_0$ from U, i.e.,

$$d(\boldsymbol{x}, L) = \|\boldsymbol{x} - \pi_L(\boldsymbol{x})\| = \|\boldsymbol{x} - (\boldsymbol{x}_0 + \pi_U(\boldsymbol{x} - \boldsymbol{x}_0))\| \tag{3.73a}$$

$$= d(\boldsymbol{x} - \boldsymbol{x}_0, \pi_U(\boldsymbol{x} - \boldsymbol{x}_0)) = d(\boldsymbol{x} - \boldsymbol{x}_0, U)\,. \tag{3.73b}$$

3.9 Rotations

Length and angle preservations

Rotation: a linear mapping that rotates a plane by an angle about the origin (fixed point)

3.9.4 Properties of Rotations

Rotations exhibit a number of useful properties, which can be derived by considering them as orthogonal matrices (Definition 3.8):

- Rotations preserve distances, i.e., $\|\boldsymbol{x}-\boldsymbol{y}\| = \|\boldsymbol{R}_\theta(\boldsymbol{x})-\boldsymbol{R}_\theta(\boldsymbol{y})\|$. In other words, rotations leave the distance between any two points unchanged after the transformation.
- Rotations preserve angles, i.e., the angle between $\boldsymbol{R}_\theta\boldsymbol{x}$ and $\boldsymbol{R}_\theta\boldsymbol{y}$ equals the angle between $\boldsymbol{x}$ and $\boldsymbol{y}$.
- Rotations in three (or more) dimensions are generally not commutative. Therefore, the order in which rotations are applied is important, even if they rotate about the same point. Only in two dimensions vector rotations are commutative, such that $\boldsymbol{R}(\phi)\boldsymbol{R}(\theta) = \boldsymbol{R}(\theta)\boldsymbol{R}(\phi)$ for all $\phi, \theta \in [0, 2\pi)$. They form an Abelian group (with multiplication) only if they rotate about the same point (e.g., the origin).