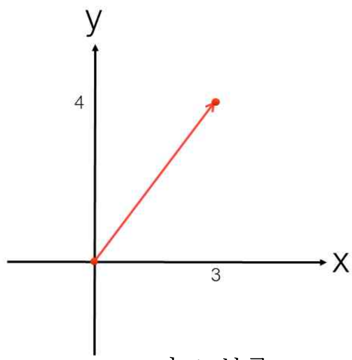
1119, 1121 영어음성학 Summary : 선형대수 (Linear Algebra)
(ipynb 형식으로 정리할 수 없어서 pdf로 정리하겠습니다.)

2018130816 강이수

데이터는 벡터의 형태로 되어 있어야 한다.
음성이 들어가서 텍스트가 나옴 : 음성인식
텍스트가 들어가서 음성이 나옴 : 음성합성
일본어 텍스트가 들어가서 한국어 텍스트가 나옴 : 기계번역
‘들어가는 곳’에 해당되는, 음성을 텍스트로 바꿔주고 텍스트를 음성으로 바꿔주는 그 중간 매체는 어떤 형태를 지닐까? -> “행렬의 형태를 지님.”
인공지능 : 행렬의 형태(행렬의 곱), 입력벡터를 출력벡터로 바꿔줌
즉, 인공지능(행렬)의 곱셈을 할 수 있게 해주기 위해 데이터가 벡터 형태가 되어야 한다는 것

1. Vector = a sequence of numbers

$$\begin{bmatrix} 3 \\ 4 \end{bmatrix}$$



위와 같은 m by 1 matrix : column vector라고 부름.

2. Vector multiplication

scalar*vector의 형태, scalar도 일종의 1 by 1 matrix라고 생각하고 곱해주면 됨.

$0.5\begin{bmatrix} 6 \\ 8 \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$

3. Vector addition

같은 위치에 있는 정보끼리 더해 줌.
$\begin{bmatrix} 3 \\ 4 \end{bmatrix} + \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ 5 \end{bmatrix}$

4. Linear combination (c*v+d*w)

c,d는 scalars, v,w는 vectors
간단하게 linear combination이란 스칼라와 벡터를 곱하고, 그 값들을 더하는 것임.
나누거나 제곱한 형태는 linear combination이라고 부르지 않음.

5. Vector spaces

vector space : 무수하게 많은 벡터들의 linear combination이 만들어내는 공간.
vector space는 n차원의 '모든' 공간이 되어야지, 그 일부분만은 될 수 없다.
(2차원 공간에서 제1사분면만 vector space라고 할 수는 없다는 것)
vector space의 조건 : linear combinations still stay in the space
즉, vector space 내의 어떠한 원소들을 linear combination 해도 그 결과값 역시 그 vector space 안에 있어야 한다.
Rn : n개의 component를 가진 모든 벡터가 채워져 있는 공간. (n차원의 공간)
R1은 선 line, R2는 평면 plain

6. Column space

column space C(A) : matrix A의 column들이 spanning해서 만드는 공간, column들이 linear combination해서 만들어내는 모든 공간.

A = $\begin{bmatrix} 2 & -1 \\ 1 & 3 \end{bmatrix}$

column vector 두 개, 이 두 column들로 linear combination을 무한대로 하면, 모든 2차원 공간이 채워짐. 이 공간이 column vector에 의한 column space
column space는 column vector보다 차원이 높을 수 없음. 낮거나 같다.
원점이 있고 두 column vector가 있는데, 모두 연결해서 상상하면 삼각형이 만들어짐. 그 삼각형을 확장하면(spanning) column space가 됨.

7. Dependent/ independent

두 column이 같은 선상에 있지 않으면 independent하고, 같은 선상에 놓이면 dependent하다. 즉 하나의 column이 다른 column의 linear combination으로 표현되면 independent한 것이다.
1) A=$\begin{bmatrix} 2 & -1 \\ 1 & 3 \end{bmatrix}$
whole space : vector 자체가 갖고 있는 dimension (행렬 A의 경우 2차원)
column space가 whole space가 아닌 경우 : column들이 linearly dependent한 경우
(즉, column들이 dependent하다면 linear combination으로 전체공간을 표현할 수 없음.)

2) B=$\begin{bmatrix} 2 & -1 \\ 1 & -0.5 \end{bmatrix}$

B의 두 column들은 dependent하다. (같은 선상에 있고, (column 2)*(-2)=(column 1))
원점과 두 column vector을 연결했을 때 삼각형이 나오지 않으므로, spanning도 불가능
아무리 linear combination을 통해 확장시켜봤자 line이 확장될 뿐이지, 나머지 전체공간 (2차원)을 채울 수 없음.
이 경우에는 column space: L (line)

cf) whole space의 dimension : 몇 개의 row가 있는지 찾으면 됨. (=column vector 안의 component의 수)
column space의 dimension : 몇 개의 independent한 column이 있는지.

3) C= $\begin{bmatrix} 1 & 3 & -2 \\ 2 & 3 & 0 \\ 4 & 1 & 5 \end{bmatrix}$
->세 개의 column이 모두 independent, Whole space: R3, Column space: R3

4) D= $\begin{bmatrix} 1 & 2 & -2 \\ 2 & 4 & 0 \\ 4 & 8 & 5 \end{bmatrix}$
->(column 2)=(column 1)*2이므로 dependent
Whole space: R3, Column space: P

5) E= $\begin{bmatrix} 1 & 1 & 2 \\ 2 & 1 & 3 \\ 4 & 1 & 5 \end{bmatrix}$
->(Column 3)=(column 1)+(column 2)이므로 dependent
Whole space: R3, Column space: P

6) F= $\begin{bmatrix} 1 & 2 & 4 \\ 2 & 4 & 8 \\ 4 & 8 & 16 \end{bmatrix}$
->column 1,2,3이 모두 dependent함 (각각 *2)
Whole space: R3, Column space: L (line 1차원)

7) A= $\begin{bmatrix} 1 & 3 \\ 2 & 3 \\ 4 & 1 \end{bmatrix}$
Whole space: R3, Column space: P
(3차원 속 column vector 2개)
AT (행렬 A의 transpose) = $\begin{bmatrix} 1 & 2 & 4 \\ 3 & 3 & 1 \end{bmatrix}$
Whole space: R2, Column space: R2
위의 경우엔 column vector 3개지만, 아무리 spanning해도 2차원을 벗어날 수 없음. (전체공간이 2차원이므로)
transpose하면 whole space도 달라지고 숫자도 다 달라지지만, column space는 언제나 같

다. (2차원)

8. Four spaces in a matrix

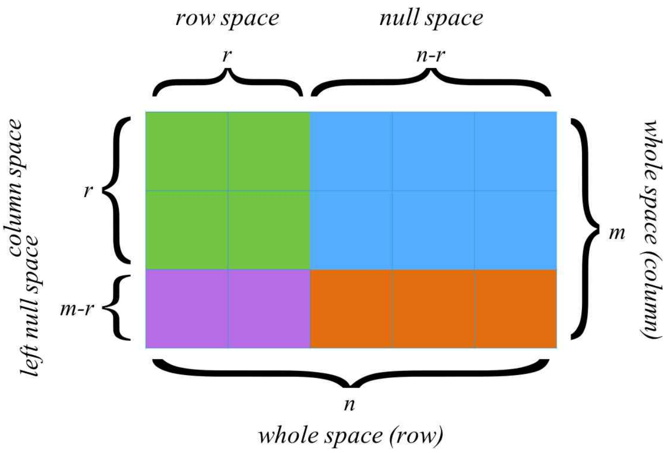
지금까지 column vector의 관점에서 whole space는 row vector의 수였음.
그러나, row vector의 관점에서도 보면 mxn 행렬은 두 가지의 전체공간을 지님: Rm, Rn

A= $\begin{bmatrix} 1 & 2 \\ 2 & 4 \\ 3 & 6 \end{bmatrix}$

column 방향으로 생각하면 whole space는 3차원
column들을 linear combination해봤자 1차원임 (두 column이 dependent하므로)
그러면 전체 3차원에서 C(A)를 뺀 2차원은 무엇일까? -> 이를 Nullspace라고 함.
수학적으로, 어떤 matrix에 무엇을 곱하면, 그 값이 0이 되게 하는 모든 '무엇'들을 모은 공간을 null space라고 함. Ax=0을 충족하는 모든 x
위의 예시에서는,

$\begin{bmatrix} 1 & 2 \\ 2 & 4 \\ 3 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$

가 되게 하는 x1, x2의 값은 (0,0), (2,-1) 등이 있고, 이것들을 모으고 linear combination 한 공간이 nullspace임.
이때 x가 왼쪽에 곱해지면 그 값이 달라지는데, x가 왼쪽에 곱해질 때에는 left nullspace라고 함. (column space 관점에서 nullspace = left nullspace)



즉, mxn matrix의 4대 부분공간은
row space와 null space가 더해져서 Rn, column space와 left nullspace가 더해져서 Rm
column space와 left nullspace는 orthogonal하기도 함. (직교)
기하적으로, 3차원에서 spanning 된 plain (column space, 2차원)에 orthogonal한 1차원의 line을 left nullspace라고 부름.

Matrix의 rank : independent한 row, column의 숫자 (둘은 항상 같음)

9. Linear transformation

Ax=b에서 A는 x의 차원도 바꾸고 숫자도 바꿈 -> linear transformation
행렬: 대문자, 벡터: 소문자로 표기
x: 입력벡터, b: 출력벡터
Ax=b에서 입력벡터를 출력벡터로 바꿔주는 transformation matrix: A

ex) $\begin{bmatrix} 0.9 & -0.4 \\ 0.4 & 0.9 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 1.3 \end{bmatrix}$
(1,1)이 입력벡터, 앞의 matrix가 인공지능의 역할
위의 식을 기하적으로도 확인할 수 있음.
(x를 A grid로 옮긴다고 표현, 옮겨진 x는 원래 grid의 위치에서 b가 됨)

ex2) $\begin{bmatrix} 1 & 0.5 \\ 0.25 & 0.125 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1.5 \\ 0.375 \end{bmatrix}$
이 식을 기하적으로 봤을 때, 두 column vector가 dependent하므로 하나의 line 위에 있음. 그러므로 x를 다시 b로 옮기려고 할 때, 원래 b의 위치를 알 수 없게 됨.

10. Inverse matrix

A^{-1}b=x
x를 b로 바꾸는 matrix A의 역행렬은 b를 다시 x로 바꿀 수 있음.

Ax=b를 기하적으로 나타내는 과정에서, A의 column들을 공간에 표현하면 만들어지는 평행사변형 하나의 면적이 determinant와 동일함. 이 면적이 0이 된다는 말은 같은 선상에 column vector가 두 개가 된다는 말이고, 이는 두 column vector가 linearly dependent하다는 말. 이 경우에 inverse matrix는 존재하지 않음.
determinant는 기하학적으로 A의 column들로 만들어지는 평행사변형의 면적과 같고, 0이 되면 역행렬이 존재하지 않는다.

$\begin{bmatrix} 1 & 0.5 \\ 0.25 & 0.125 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1.5 \\ 0.375 \end{bmatrix}$ 그러므로 이 식에서 A의 두 column (1,0.25), (0.5,0.125)은 dependent하고, (1.5, 0.375)를 다시 (1,1)로 돌려놓을 수 없음. invertible하지 않다.

11. Eigenvector (아이겐벡터)

Av=b를 만족하는 모든 v 중 몇몇 v는 Av와 평행한데, 이 v를 eigenvector라고 한다.

1126, 1128 영어음성학 Summary : 선형대수 (Linear Algebra)

1. Nullspace

수학적 해석 : Ax=0을 만족시키는 모든 x
기하적 해석 : row space에 orthogonal한 공간
응용적 해석 : 어떤 입력이 들어오든 출력에는 영향을 미치지 않는다. (0이므로)
영향을 미치지 않는다-nullspace
ex) 서로 다른 강아지 사진을 넣어도 다 '강아지'라고 인식-> 입력이 달라도 출력이 같다. (nullspace 이용)
위와 같이 인공지능과 선형대수와의 관계를 nullspace로 설명 가능.

2. 상관관계

상관관계 : 수치로 표현할 수 있는데, -1부터 1까지 숫자로 표현 가능 (-1 : 반비례, 1: 비례)
상관관계는 0에서 제일 낮음.
-1, 1 : 기울기가 어쨌든, 점들이 하나의 선상에 있기만 하면 됨.
점들이 원처럼 동그랗게 분포 -> 상관관계 전혀 없음, 상관관계가 0인 경우

ex) 85명의 국어, 영어 점수-> 국어벡터 하나, 영어벡터 하나 : 85차원
한 점에는 85개의 정보가 들어있음.
차원을 모두 지우고 원점과 국어, 영어 두 점만 남기면 원점과 각각의 점을 이을 수 있음.
원점과 각각의 점을 이은 두 선분 사이의 각도 = 상관관계 r
cos 0-> 상관관계 r: 1
원점과 그 점을 이은 선분이 멀어질수록(각도가 커질수록) 상관관계가 없다.
cos 90=> 상관관계 0 : orthogonal할수록 관련이 없다.
--> 선형대수와 통계와의 관련성

3. inner product(dot product)

(1,2,3) • (4,5,6) = 1*4+2*5+3*6 = 32
inner product의 기하적 해석: 하나의 선분에서 다른 한 선분에 수직인 점을 내리고, 원점에서부터 그 점까지의 거리(a*cos theta)와 다른 선분의 길이 (b의 길이)를 곱하는 것과 같음.
벡터의 길이 : 전부 제곱해서 더해주고 루트 씌우면 됨.
ex) a=(1,2,3)의 길이 : 루트 (1제곱+2제곱+3제곱)

우리는 소리에서 몇Hz의 소리가 어느 정도 들어 있느냐 알 수 있어야 함.
상관관계가 크면 inner product값이 크게 나올 거고, 상관관계가 작으면 inner product 작게 나올 것임.

--> inner product와 음성분석의 상관관계!

4. eigenvector Av=λv

λ(람다)는 상수 (eigenvalue)
given matrix A에 의해 transformation된 결과가 원래 벡터의 상수배라면, 즉 원래의 벡터와 A에 의해 바뀐 벡터가 한 직선상에 있다면, 그때의 벡터를 eigenvector라고 한다.

5. inner product (2)
a= $[1\,2\,3]$, b= $[2\,4\,7]$을 곱하려면?

1) b를 transpose : $[1\,2\,3]\begin{bmatrix}2\\4\\7\end{bmatrix}$ = [31]

2) a를 transpose : $\begin{bmatrix}1\\2\\3\end{bmatrix}[2\,4\,7]=\begin{bmatrix}2&4&7\\4&8&14\\6&12&21\end{bmatrix}$

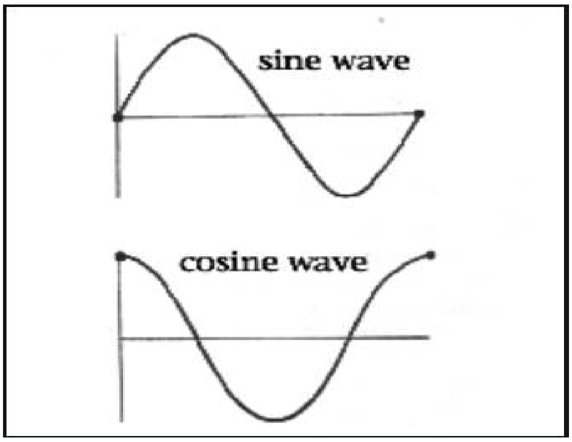
이때 1)의 경우가 inner product이다. 결과값은 scalar 하나가 나와야 한다.

*inner product의 기하적 해석
: 하나의 선분에서 다른 한 선분에 수직인 점을 내리고, 원점에서부터 그 점까지의 거리(a*cos theta)와 다른 선분의 길이 (b의 길이)를 곱하는 것과 같음.
즉 벡터 a • b=|a|*cos θ*|b|
cos θ=(a • b)/(|a|*|b|)

만약 벡터 a와 b가 orthogonal하여 theta가 90이라면, cos theta의 값이 0이 되므로 a와 b의 inner product 값도 0이다.

6. cos similarity
벡터 a와 b가 얼마나 유사한지 나타내는 개념. a와 b 사이의 각도(theta)를 구하고, cos theta값으로 유사성을 판단한다.

7. inner product와 음성 분석
하나의 complex phasor에서 Hz에 따라 wave를 뽑아내는 방법: inner product
원래의 complex phasor와 각 Hz의 wave를 inner product하여 그 값을 구하면, 하나의 complex phasor에 어느 주파수가 얼마나 있는지 그 스펙트로그램을 구할 수 있다.
벡터 a와 b의 wave가 같이 움직인다면, 부호가 항상 같아 inner product 값이 상당히 크게 나온다. 반면, a와 c가 반대로 움직여, a가 올라갈 때 c가 내려가는 불일치가 초래된다면 값이 줄어든다. **그러므로 똑같은 성분이 많을수록 inner product값이 크게 나온다.** (이 원리로 하나의 complex phasor에서 주파수 성분을 뽑아낼 수 있는 것임.)



sine wave끼리 inner product하면 그 값이 4가 나옴 (0+1+0+1+0+1+0+1+0)
sine wave와 cosine wave를 inner product하면 0이 나옴.

sine wave를 90도 이동했는데 inner product 값이 0이 되었다-> inner product는 어떤 주파수가 많은지 알려는 것이 목적인데, 움직임에 따라 결과값이 예민하게 변함.

-> 그래서 complex phasor을 써서 inner product 해야한다.