

12/ 1 (11/25 - 11/30)

Eigen analysis(고유 벡터)

기하적 관점 : 원점, 입출력이 평행한 상황에서 eigen vector가 형성된다. 이때 eigen vector는 방향이라고 이해하는 것이 수월하다.

또한 eigen vector 상에서는, 입력값이 출력값으로 나타나는 비율이 일정하다.

즉 $Av = (\text{상수})v$ 이는 기존에 배웠던 linear combination의 정의와도 일치한다.

실용적 관점

어떤 벡터가 있다고 하면, 그 벡터는 여러 가지 수치가 합해진 값이다. 하지만 eigen vector는 그 벡터의 고유성을 보여준다. 즉, 주어진 벡터는 컬럼 벡터의 수치들이 일정하지 않게 분포된 값이지만, eigen vector는 그 수치들이 어떤 경향성을 가지고 있는지 보여주는 것이다.

벡터를 통해서 두 수치 사이의 상관 관계를 구할 수도 있다.

보통 통계에서 어떤 자료들이 반비례/비례 관계를 보이면 그 상관성을 $-1 < r < 1$ 의 수치로 표현한다. 이는 자료들이 어떤 선을 이루고 있으면 $-1, 1$ 과 가깝고 전혀 그렇지 않을 때 0의 값을 갖는다. 신기하게도 이것을 벡터 차원으로 표현할 수 있다. 예를들어 80명 학생의 국어와 영어 점수에 대한 경향성을 구한다고 할 때, 영어 점수 전체는 80차원의 column vector로써 한 점으로 표현될 수 있다. 국어도 마찬가지다. 이 두 벡터 사이에 형성되는 각의 코사인 값이 위에서 이야기했던 r 값과 일치한다. 따라서 벡터 사이의 각이 두 수치 사이의 유사성을 설명해 줄 수 있는 것이다.

그렇다면 이 theta 값은 어떻게 구할까?

inner product, dot product 방법을 활용할 수 있다.

국어 벡터 a 값, 영어 벡터 b 값, 그리고 원점이 있다고 하자.

(b 에 a 를 프로젝트한 길이) * 원점에서 b 까지의 길이 = inner product 이다.

여기서 b 에 a 를 프로젝트 한 길이는 $a * \cos\theta$ 값과 일치한다.

따라서 inner product(a 행렬 * b 행렬)와 oa 의 길이 ob 의 길이를 통해 세타값을 구할 수 있다.

이렇게 얼마나 일치하는지를 알려주는 정보인 theta 값을 통해 spectrogram을 실제로 만들어낼 수 있다.

1. 페이지로 특정 주파수 만들기

2. 이 주파수가 만들고자 하는 spectrogram과 얼마만큼 일치하는지의 수치를 통해서 특정 주파수가 얼마만큼의 비중을 차지하고 있는지 파악할 수 있다.

3. 프랏한다.

하지만 여기에서도 문제가 생긴다. 같은 주파수를 갖고 있지만 시간이라는 차이로 그 상관성이 0이 되어 버리는 경우가 있다. (코사인과 사인의 관계)

따라서 조금 더 복잡한 페이지를 이용해야 한다.

각 HZ 별 phase를 만든 후, 그것을 target phase에 대해 inner product할 것이다.

이를 통해 각 HZ의 wave가 target phase와 얼마나 유사한지 알 수 있으며,

이를 통해 각 HZ가 얼마만큼의 비중으로 구성되었는지 파악할 수 있다.

1. HZ 별 phase 만들기

```
nSamp = len(s)
```

타겟 벡터의 수만큼으로 nsamp를 설정한다. 각 Hz 별 phase를 만들려면, 몇 개의 벡터로 구성되는지 알아야 하기 때문이다. 반다시 target phase와 비교하는 phase의 벡터 수가 같아야 한다!!

```
nFFT = nSamp
```

```
amp = [];
```

```
for n in range(0,nFFT):
```

```
    omega = 2*np.pi*n/nFFT # angular velocity
```

```
    #w = 2pi * n / 100
```

```
    z = np.exp(omega*1j)*(np.arange(0,nSamp))
```

```
    #컴플렉스를 만들어주는 코드
```

```
    amp.append(np.abs(np.dot(s,z)))
```

-> 100개의 데이터를 가지고 n주기를 돌게 된다.

2. inner product

```
amp.append(np.abs(np.dot(s,z)))
```

빈 리스트인 amp에 s와 z의 inner product 값을 계속 축전한다.

여기에서 중요한 것은 inner product 값에 절대값을 씌어야 한다는 것이다.

complex phasor로 작업했기 때문에 허수가 나올 가능성이 크다.

3. amp plotting

amp에 담긴 값은 y값이고, 이제는 x값을 설정해야 한다.

```
freq = np.arange(1,nFFT+1)*sr/nFFT;
```

이렇게 plot한 그래프가 나타나게 되는데,

여기서 하나의 bar 값은 하나의 inner product값과 같고,

총 bar의 개수 = sample의 개수

대칭의 형태 - nyquist frequency와 연관됨

S를 만들었던 과정이 그대로 드러나 있음

4. 효과적인 시각화

현재의 스펙트로그램은 매우 연하므로 보정을 할 것이다.

```
powspec = 1/nttf * (magspe**2)
```

#스펙트로그램에서 진한 부분은 1보다 큰 값을, 연한 부분은 1보다 작은 값을 취하기 때문에 제곱을 하면, 쉽게 로그를 적용할 수 있다.

로그를 적용하게 되면, 너무 작은 숫자나 너무 큰 숫자를 다룰 수 있는 범위 내로 만들어 주는 과정이다. 이를 통해 우리가 기존에 배웠던 스펙트로 그래프가 나타나게 된다.