

HW #10 SVD 이해

code

MathVision22
02221081 황지현

- Q1. Homogeneous system $Ax = 0$ 의 해는 다음과 같음을 설명 또는 증명하시오.
 - 0인 특이값이 존재하는 경우
 - $\sigma_i = 0$ 에 대응되는 right singular vector v_i 가 해

A의 SVD인 $A=U\Sigma V^T$ 의 양변에 v_i 를 곱하면

$$Av_i = U\Sigma V^T v_i = \sigma_i u_i \text{ 가 되는 것을 볼 수 있다.}$$

여기서 V는 orthogonal 행렬이기 때문에 i번째 행은 1, 나머지 행들은 전부 0,

$$Av_i = U\Sigma V^T v_i = \sigma_i u_i \Rightarrow Av_i = \sigma_i u_i$$

$Av_i = \sigma_i u_i$ 에서 0인 특이값이 존재하려면 i행의 특이값도 0이어야 한다.

따라서 i번째 특이값인 v_i 가 $Ax=0$ 의 해가 됨을 알 수 있다.

- 모든 특이값이 양수인 경우
 - 정확한 해는 존재하지 않으며
 - 최소 특이값에 대응되는 right singular vector(V의 가장 오른쪽 열 벡터)이 $\|Ax\|$ 를 최소화하는 해

$$Av_i = \sigma_i u_i$$

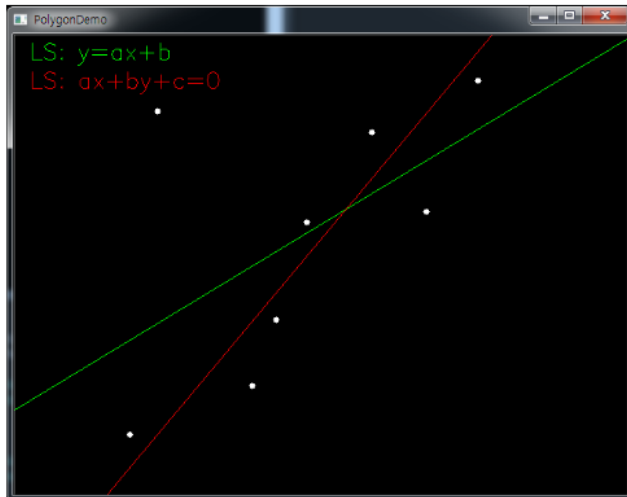
위 식의 우변에서 u_i 의 크기는 1 (orthogonal matrix)

$Av_i = \sigma_i$ 의 식으로 보았을 때 우변의 크기는 σ_i 의 특이값이라고 볼 수 있다.

$\|Ax\|$ 를 최소화하려면 σ_i 이 0에 수렴할수록 좋다.

따라서 모든 특이값이 양수인 경우 σ_i 의 크기가 0에 가까워질수록 Av_i 의 크기 또 0에 수렴하므로 $\|Ax\|=0$ 에 가장 가까운 해는 최소 특이값인 v_i 가 된다.

- Q2. 사용자로부터 $n(n \geq 2)$ 개의 점을 입력받은 후 입력 받은 점들의 근사 직선을 **SVD로 구하여** 화면에 도시하는 프로그램을 작성하시오.
 - 직선모델을 $y = ax+b$ 로 한 근사직선은 **초록색** 출력
 - 직선모델을 $ax + by + c = 0$ 으로 한 근사직선은 **붉은색** 출력
 - 리포트(패들릿) + 구현코드(GitHub) 제출



- * 리포트 포함 내용: 실행화면 캡처
- * 점들이 x축, y축 평행인 경우, 등 다양한 경우 결과 비교
- * 결론

핵심 코드

Pseudo inverse (의사 역행렬) : $ax + b = 0$ 구하기

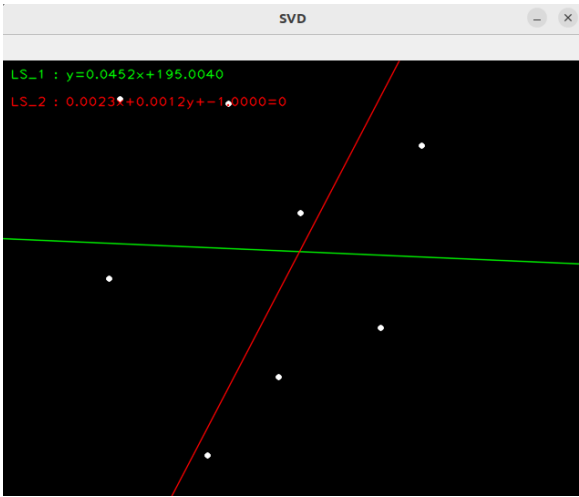
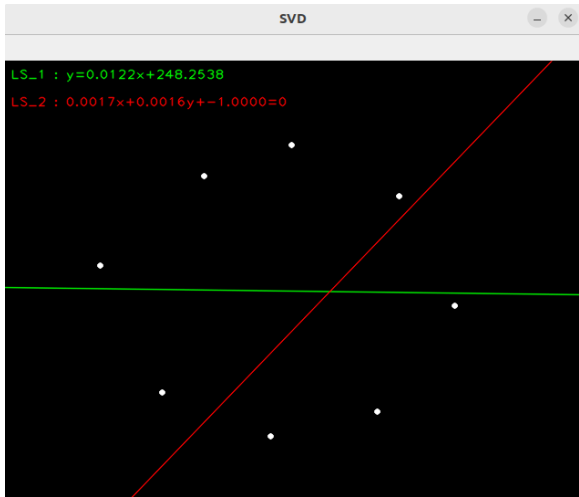
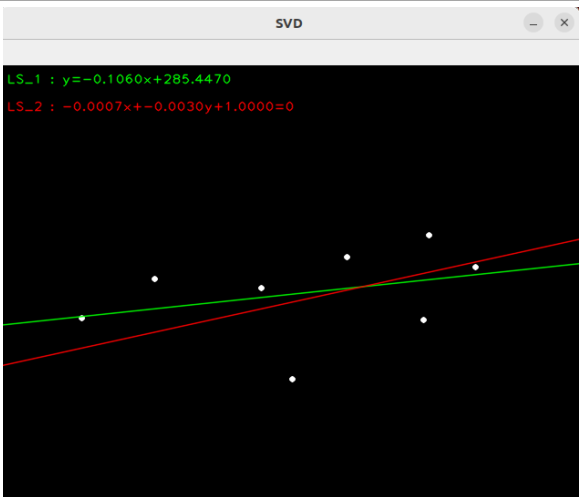
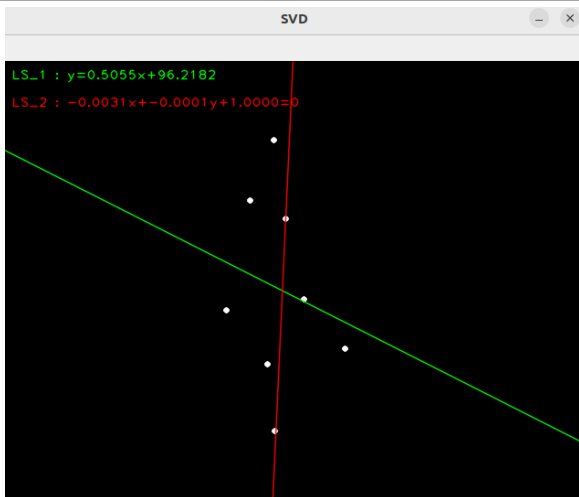
```
# Pseudo inverse
pi = np.linalg.inv(p1_l.T @ p1_l) @ p1_l.T
result = pi @ p2_r
a,b = result
cv2.putText(window, f'LS_1 : y={a:.4f}x+{b:.4f}', (10, 20), cv2.FONT_HERSHEY_PLAIN, 1.0, (0,255,0))
x1, x2 = 0, 640
y1, y2 = int(a*x1+b), int(a*x2+b)
cv2.line(window, (x1,y1), (x2,y2), (0,255,0), 1)
```

SVD (특이값 분해) : $ax + by + c = 0$ 구하기

```
# SVD
U,D,V = np.linalg.svd(np.array(p3))
a,b,c = V[-1]
cv2.putText(window, f'LS_2 : {a:.4f}x+{b:.4f}y+{c:.4f}=0', (10, 50), cv2.FONT_HERSHEY_PLAIN, 1.0, (0,0,255))
if np.abs(b) < 1e-9 :
    x1 = -c/a
    cv2.line(window, (x1,0), (x2,H), (0,0,255), 1)
else :
    y1 = int((-c - (a*x1))/ b)
    y2 = int((-c - (a*x2))/ b)
    cv2.line(window, (x1,y1), (x2,y2), (0,0,255), 1)
cv2.imshow(title, window)
```

실행화면 캡처 (n=8 으로 통일)

화면 상단에서 직선 방정식을 확인할 수 있다.

1. 무작위	2. 원형
	
3. 가로	4. 세로
	

결과

점들이 가로로(x 축과 평행) 분포한 경우에는 결과가 잘 출력되지만,
세로로(y 축과 평행) 분포하는 경우는 제대로 출력되지 않는다는 것을 볼 수 있었다.
수업시간에 나왔던 3 차원 공간에서의 수행이 필요하다는 것을 느꼈다.

참고 링크

1. <https://darkpgmr.tistory.com/106>
2. <https://datascienceschool.net/02%20mathematics/03.04%20%ED%8A%B9%EC%9E%87%EA%B0%92%20%EB%B6%84%ED%95%B4.html>