1. Simulation – Parameter Tuning

<model 1>

$Y_1 = \sin(\beta_1^T X) + \sin(\beta_2^T X) + 0.5\varepsilon$

n=300, p=40

Case (1) : true beta 0 개수 78개	Sparse DR

Case (1)

: $\beta_1 = (1, 0, \ldots, 0)^{\mathrm{T}}$ $\beta_2 = (0, 1, 0, \ldots, 0)^{\mathrm{T}}$

λ1 = 0.1, 0.05

λ2 = 0.05

slice 개수: 4개

method	p	corr1	corr2	mse1	mse2
dr	0.00	0.7092513	0.12164155	2.602801	1.965467
sparse dr	77.16	0.8865939	0.06533786	2.749936	1.929122

Case (1)

: $\beta_1 = (1, 0, \ldots, 0)^{\mathrm{T}}$ $\beta_2 = (0, 1, 0, \ldots, 0)^{\mathrm{T}}$

λ1 = 0.1, 0.01

λ2 = 0.01

slice 개수: 10개

method	p	corr1	corr2	mse1	mse2
dr	0.00	0.693193	0.1045944	2.838022	1.963565
sparse dr	62.33	0.892805	0.4696206	2.952623	2.441872

Case (1) –처음 결과

: $\beta_1 = (1, 0, \ldots, 0)^{\mathrm{T}}$ $\beta_2 = (0, 1, 0, \ldots, 0)^{\mathrm{T}}$

λ1 = 0.01, 0.01

λ2 = 0.01

slice 개수: 4개

method	p	corr1	corr2	mse1	mse2
dr	0.00	0.7092513	0.12164155	2.602801	1.965467
sparse dr	59.94	0.7548557	0.07432311	2.724729	2.125324

Case (2) : true beta 0 개수 60개	Sparse DR

Case (2)

: β_1 =(1,1,1,1,1,0.1,0.1,0.1,0.1,0.1,0,0...0,0)

β_2 =(0,0...0,0,0.1,0.1,0.1,0.1,0.1,1,1,1,1,1)

λ1 = 0.05, 0.01

λ2 = 0.01

slice: 10개

method	p	corr1	corr2	mse1	mse2
dr	0.00	0.1373907	0.1228866	2.074601	1.904770
sparse dr	62.66	0.1829674	0.1011862	2.230346	2.137052

Case (2)

: β_1 =(1,1,1,1,1,0.1,0.1,0.1,0.1,0.1,0,0...0,0)

β_2 =(0,0...0,0,0.1,0.1,0.1,0.1,0.1,1,1,1,1,1)

λ1 = 0.05, 0.01

λ2 = 0.01

slice: 4개

method	p	corr1	corr2	mse1	mse2
dr	0.00	0.1391564	0.1217259	2.140710	1.942375
sparse dr	66.65	0.1422226	0.1467648	2.127528	2.093356

Case (2)

: β_1 =(1,1,1,1,1,0.1,0.1,0.1,0.1,0.1,0,0...0,0)

β_2 =(0,0...0,0,0.1,0.1,0.1,0.1,0.1,1,1,1,1,1)

λ1 = 0.05, 0.05

λ2 = 0.01

slice: 4개

method	p	corr1	corr2	mse1	mse2
dr	0	0.1416286	0.1105666	2.086301	1.882642
sparse dr	78	0.1094201	0.1119920	2.069400	1.972753

lambda를 키웠더니 beta 0의 개수가 너무 많게 나옴

Case (2)—가장 처음 결과

: β_1 =(1,1,1,1,1,0.1,0.1,0.1,0.1,0.1,0,0...0,0)

β_2 =(0,0...0,0,0.1,0.1,0.1,0.1,0.1,1,1,1,1,1)

λ1 = 0.01, 0.01

λ2 = 0.01

slice: 4개

method	p	corr1	corr2	mse1	mse2
dr	0.00	0.1395863	0.1152813	2.099900	1.933958
sparse dr	55.27	0.1792127	0.1335033	2.245751	2.133104

Case (3) : true beta 0 개수 60개

Sparse DR

Case (3)

: β_1 =(1,1,1,1,1,1,1,1,1,1 ,0,0...0,0)

β_2 =(0,0...0,0,0,1,1,1,1,1,1,1,1,1,1)

λ1 = 0.05, 0.05

λ2 = 0.05

slice: 10개

method	p	corr1	corr2	mse1	mse2
dr	0.00	0.1543846	0.1349226	2.146090	1.944538
sparse dr	77.96	0.1269302	0.1231912	2.113108	1.934631

lambda를 늘렸더니 beta 0의 개수가 너무 많게 나옴

Case (3)

: β_1 =(1,1,1,1,1,1,1,1,1,1 ,0,0...0,0)

β_2 =(0,0...0,0,0,1,1,1,1,1,1,1,1,1,1)

λ1 = 0.1, 0.01

λ2 = 0.01

slice: 10개

method	p	corr1	corr2	mse1	mse2
dr	0.00	0.1610913	0.1147005	2.078947	1.923570
sparse dr	63.01	0.2124721	0.1095257	2.088260	2.162515

Case (3)—가장 처음 결과

: β_1 =(1,1,1,1,1,1,1,1,1,1 ,0,0...0,0)

β_2 =(0,0...0,0,0,1,1,1,1,1,1,1,1,1,1)

λ1 = 0.01, 0.01

λ2 = 0.01

slice: 4개

method	p	corr1	corr2	mse1	mse2
dr	0.00	0.1407005	0.1310424	2.094963	1.959305
sparse dr	55.28	0.2043405	0.1515334	2.277170	2.051917

<model 2>

$Y_2 = \frac{\beta_1^T X}{0.5+(2\beta_2^T X+0.3)^2} + 0.3\varepsilon$, n=300, p=50

Case(1): : true beta 0 개수 98개	Sparse DR

Case (1)

: $\beta_1 = (1, 0, \ldots, 0)^{\mathrm{T}}$, $\beta_2 = (0, 1, 0, \ldots, 0)^{\mathrm{T}}$

λ1 = 0.1, 0.1

λ2 = 0.1

slice: 10개

method	p	corr1	corr2	mse1	mse2
dr	0	0.8344419	0.1473274	0.3430334	2.111377
sparse dr	98	1.0000000	0.5477459	0.0000000	1.223341

Case (1) – 가장 처음 결과

: $\beta_1 = (1, 0, \ldots, 0)^{\mathrm{T}}$, $\beta_2 = (0, 1, 0, \ldots, 0)^{\mathrm{T}}$

λ1 = 0.01, 0.01

λ2 = 0.01

slice: 4개

method	p	corr1	corr2	mse1	mse2
dr	0.00	0.8219249	0.2578787	0.36904806	1.966568
sparse dr	79.11	0.9833684	0.4017083	0.03622688	2.077622

Case(2) : : true beta 0 개수 80개	Sparse DR

Case (2)

β_1 =(1,1,1,1,1,0.1,0.1,0.1,0.1,0.1,0,0...0,0)

β_2 =(0,0...0,0,0.1,0.1,0.1,0.1,0.1,1,1,1,1,1)

λ1 = 0.05, 0.05

λ2 = 0.01

slice: 10개

method	p	corr1	corr2	mse1	mse2
dr	0.00	0.7676112	0.1834386	3.420411	2.004947
sparse dr	97.84	0.5064309	0.1217545	2.915175	2.048526

lambda를 늘렸더니 beta 0의 개수가 너무 많게 나옴

Case (2)

β_1 =(1,1,1,1,1,0.1,0.1,0.1,0.1,0.1,0,0...0,0)

β_2 =(0,0...0,0,0.1,0.1,0.1,0.1,0.1,1,1,1,1,1)

λ1 = 0.05, 0.01

λ2 = 0.01

slice: 10개

method	p	corr1	corr2	mse1	mse2
dr	0.0	0.7676112	0.1834386	3.420411	2.004947
sparse dr	80.7	0.4086287	0.1507352	2.542006	2.208538

Case (2) – 가장 처음 결과

β_1 =(1,1,1,1,1,0.1,0.1,0.1,0.1,0.1,0,0...0,0)

β_2 =(0,0...0,0,0.1,0.1,0.1,0.1,0.1,1,1,1,1,1)

λ1 = 0.01, 0.01

method	p	corr1	corr2	mse1	mse2
dr	0.00	0.6001518	0.1900158	2.916600	2.112913
sparse dr	75.04	0.6409069	0.2345991	2.955934	2.252680

<table>
<tr><td>$\lambda 2 = 0.01$
slice: 4개</td><td></td></tr>
</table>

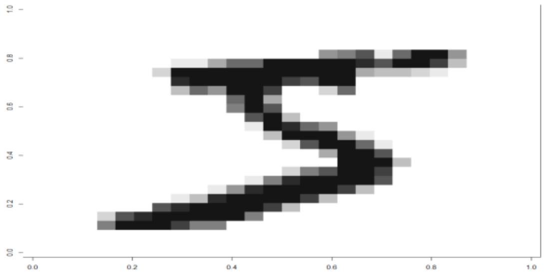
<table>
<tr><td>Case(3): : true beta 0 개수 80개</td><td>Sparse DR</td></tr>
<tr><td>Case (3)
β_1 =(1,1,1,1,...,1,1,1,0,0...0,0)
β_2 =(0,0...0,0,0,1,1,1,1,...,1,1)
λ 1 = 0.05, 0.01
$\lambda 2 = 0.01$
slice: 4개</td><td>
<table>
<tr><th>method</th><th>p</th><th>corr1</th><th>corr2</th><th>mse1</th><th>mse2</th></tr>
<tr><td>dr</td><td>0.00</td><td>0.3479419</td><td>0.1719531</td><td>2.255180</td><td>2.009467</td></tr>
<tr><td>sparse dr</td><td>85.56</td><td>0.2734218</td><td>0.1649450</td><td>2.192834</td><td>2.147575</td></tr>
</table>
</td></tr>
<tr><td>Case (3)
β_1 =(1,1,1,1,...,1,1,1,0,0...0,0)
β_2 =(0,0...0,0,0,1,1,1,1,...,1,1)
$\lambda 1 = 0.05, 0.01$
$\lambda 2 = 0.01$
slice: 10개</td><td>
<table>
<tr><th>method</th><th>p</th><th>corr1</th><th>corr2</th><th>mse1</th><th>mse2</th></tr>
<tr><td>dr</td><td>0.00</td><td>0.4743199</td><td>0.1638061</td><td>2.259702</td><td>2.030333</td></tr>
<tr><td>sparse dr</td><td>81.02</td><td>0.2930573</td><td>0.1406950</td><td>2.327612</td><td>2.197306</td></tr>
</table>
</td></tr>
<tr><td>Case (3) – 가장 처음 결과
β_1 =(1,1,1,1,...,1,1,1,0,0...0,0)
β_2 =(0,0...0,0,0,1,1,1,1,...,1,1)
$\lambda 1 = 0.01, 0.01$
$\lambda 2 = 0.01$
slice: 4개</td><td>
<table>
<tr><th>method</th><th>p</th><th>corr1</th><th>corr2</th><th>mse1</th><th>mse2</th></tr>
<tr><td>dr</td><td>0.00</td><td>0.3479419</td><td>0.1719531</td><td>2.255180</td><td>2.009467</td></tr>
<tr><td>sparse dr</td><td>73.48</td><td>0.3639681</td><td>0.1779225</td><td>2.360421</td><td>2.225217</td></tr>
</table>
</td></tr>
</table>

2. Application

Data1 : mnist data

https://github.com/jlmelville/mnist

grayscale의 28*28 data (60000개 – training data, 10000개 – test data)



이미지 데이터를 불러와 70000*784 matrix로 변환하여 진행

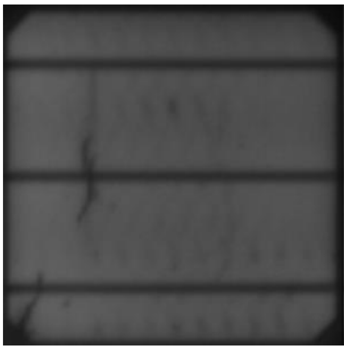
<X train set(60000*784) matrix – sparse dr 결과>

Sparse dr 후 beta (784*5 matrix)

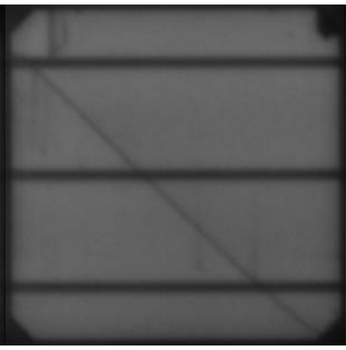
```
> ssdr.result1 <- ssdr.lambda(x.tra, y.tra, method="dr", d=5, nslices=10, lambda1=c(0.01,0.01,0.01,0.01,0.01), lambda2=0.01, m
ax.iter=100, eps.conv=1e-7)
> beta = beta.order(ssdr.result1$beta)
> beta
               [,1]         [,2]          [,3]        [,4]         [,5]
 [1,]  0.000000e+00  0.000000000  0.000000e+00  0.00000000 0.0000000000
 [2,]  0.000000e+00  0.000000000  0.000000e+00  0.00000000 0.0000000000
 [3,]  0.000000e+00  0.000000000  0.000000e+00  0.00000000 0.0000000000
 [4,]  0.000000e+00  0.000000000  0.000000e+00  0.00000000 0.0000000000
 [5,]  0.000000e+00  0.000000000  0.000000e+00  0.00000000 0.0000000000
 [6,]  0.000000e+00  0.000000000  0.000000e+00  0.00000000 0.0000000000
 [7,]  0.000000e+00  0.000000000  0.000000e+00  0.00000000 0.0000000000
 [8,]  0.000000e+00  0.000000000  0.000000e+00  0.00000000 0.0000000000
 [9,]  0.000000e+00  0.000000000  0.000000e+00  0.00000000 0.0000000000
[10,]  0.000000e+00  0.000000000  0.000000e+00  0.00000000 0.0000000000
[11,]  0.000000e+00  0.000000000  0.000000e+00  0.00000000 0.0000000000
[12,]  0.000000e+00  0.000000000  0.000000e+00  0.00000000 0.0000000000
[13,]  0.000000e+00  0.000000000  0.000000e+00  0.00000000 0.0000000000
[14,]  0.000000e+00  0.000000000  0.000000e+00  0.00000000 0.0000000000
[15,]  0.000000e+00  0.000000000  0.000000e+00  0.00000000 0.0000000000
[16,]  0.000000e+00  0.000000000  0.000000e+00  0.00000000 0.0000000000
[17,]  0.000000e+00  0.000000000  6.986332e-02  0.00000000 0.0000000000
[18,]  0.000000e+00  0.148874260  0.000000e+00  0.00000000 0.0000000000
[19,]  0.000000e+00  0.000000000  0.000000e+00  0.00000000 0.0000000000
[20,]  0.000000e+00  0.004863110  0.000000e+00  0.00000000 0.0000000000
[21,]  0.000000e+00  0.000000000  0.000000e+00  0.00000000 0.0000000000
[22,]  0.000000e+00  0.000000000  0.000000e+00  0.00000000 0.0000000000
[23,]  0.000000e+00  0.000000000  5.196173e-04  0.00000000 0.0000000000
[24,]  0.000000e+00 -0.003701105  0.000000e+00 -0.19249843 0.0000000000
[25,]  0.000000e+00  0.000000000  0.000000e+00  0.51898810 0.0000000000
[26,]  0.000000e+00  0.000000000  0.000000e+00  0.00000000 0.0000000000
[27,]  0.000000e+00  0.000000000  0.000000e+00  0.00000000 0.0017398978
[28,]  0.000000e+00  0.000000000  0.000000e+00  0.00000000 0.0000000000
[29,]  0.000000e+00  0.534223866  8.791777e-02  0.00000000 0.0000000000
[30,]  0.000000e+00  0.000000000  0.000000e+00  0.00000000 0.0000000000
[31,]  0.000000e+00  0.000000000  0.000000e+00  0.00000000 0.0000000000
[32,]  0.000000e+00  0.000000000  0.000000e+00 -0.31181533 0.0000000000
[33,]  0.000000e+00  0.000000000  0.000000e+00  0.00000000 0.0000000000
[34,]  0.000000e+00  0.000000000  0.000000e+00  0.00000000 0.0000000000
[35,]  0.000000e+00  0.000000000  0.000000e+00  0.00000000 0.0000000000
[36,]  0.000000e+00  0.000000000  0.000000e+00  0.00000000 0.0000000000
[37,]  0.000000e+00  0.000000000  0.000000e+00  0.00000000 0.0000000000
[38,]  0.000000e+00  0.000000000  0.000000e+00  0.00000000 0.0000000000
[39,] -2.515690e-03  0.000000000  0.000000e+00  0.00000000 0.0000000000
[40,]  0.000000e+00  0.000000000  0.000000e+00  0.00000000 0.0000000000
[41,] -7.020160e-02  0.000000000  0.000000e+00  0.00000000 0.0000000000
[42,] -5.376417e-02  0.000000000  0.000000e+00  0.00000000 0.0000000000
[43,]  0.000000e+00  0.000000000  0.000000e+00  0.00000000 0.0000000000
[44,]  0.000000e+00  0.000000000  0.000000e+00  0.00000000 0.0000000000
[45,]  0.000000e+00  0.000000000  0.000000e+00  0.00000000 0.0000000000
[46,]  0.000000e+00  0.000000000  0.000000e+00  0.00000000 0.0000000000
[47,] -8.042546e-02  0.000000000  0.000000e+00  0.00000000 0.0000000000
[48,] -7.105272e-02  0.000000000  0.000000e+00  0.00000000 0.0000000000
[49,] -1.066256e-02  0.000000000  0.000000e+00  0.00000000 0.0000000000
```

Data2: 태양 전지 결함 데이터 – 전기 발광 이미지 (Visual Identification of Defective Solar Cells in Electroluminescence Imagery)

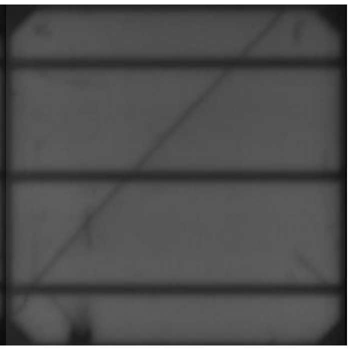
https://github.com/zae-bayern/elpv-dataset : 300*300, n=2624



cell0001.png



cell0002.png



cell0003.png

labels.csv

attribute	Description
Image url	이미지 저장 이름
Defect probability	결함 확률(0, 0.33333, 0.66666, 1) 1로 가까워질수록 결함이 있는 태양 전지이다.
Solar module type	태양 전지의 종류(mono/poly) Mono: 단일 실리콘 결정으로 만들어진 검은색 태양전지를 가지고 있으며, 일반적으로 효율이 더 높다. 가격이 비싸다. Poly: 여러 개의 실리콘 결정으로 만들어지며 파란색 셀을 가지고 있다. 효율은 다소 떨어지지만 가격이 저렴하다.