

# Summary of Sparse Biclustering of Transposable Data

Yuchen Zeng

## 1 Biclustering

Biclustering, block clustering, co-clustering, or two-mode clustering is a data mining technique which allows simultaneous clustering of the rows and columns of matrix.

## 2 Assumptions

- each matrix element is normally distributed with a bicluster-specific mean;
- the biclusters partition the rows and columns of the matrix.

## 3 Sparse biclustering

### 3.1 Model assumptions

- The biclusters all are constant biclusters, in which all elements take on approximately a constant value.
- $X_{ij} \sim N(\mu_{kr}, \sigma^2)$  for  $i \in C_k, j \in D_r, k = 1, \dots, K$  and  $r = 1, \dots, R$ , and they are independent with each other.

### 3.2 Model

The model is:

$$X_{ij} = \mu_{kr} + \varepsilon_{ij} \text{ where } \varepsilon_{ij} \sim N(0, \sigma^2)$$

**Given parameters:**  $K, R, \lambda$ ;

**Unknown parameters:**  $\mu_{kr}, \{C_k\}, \{D_r\}$ .

Maximizing the log likelihood of the data under the model with inducing sparsity by using a LASSO penalty, we arrived at:

$$\underset{C_1, \dots, C_K, D_1, \dots, D_R, \mu \in R^{K \times R}}{\text{minimize}} \left\{ \frac{1}{2} \sum_{k=1}^K \sum_{r=1}^R \sum_{i \in C_k} \sum_{j \in D_r} (X_{ij} - \mu_{kr})^2 + \lambda \sum_{k=1}^K \sum_{r=1}^R |\mu_{kr}| \right\} \quad (1)$$

where  $\lambda$  is a non-negative tuning parameter.

### 3.3 An extension to tensor

#### 3.3.1 Model assumptions

- The clusters all are constant clusters, in which all elements take on approximately a constant value.
- $X_{ijm} \sim N(\mu_{krm}, \sigma^2)$  for  $i \in C_k, j \in D_r, m \in E_l, k = 1, \dots, K, r = 1, \dots, R, l = 1, \dots, L$ , and they are independent with each other.

#### 3.3.2 Model

The model is:

$$X_{ijm} = \mu_{krm} + \varepsilon_{ijm} \text{ where } \varepsilon_{ijm} \sim N(0, \sigma^2)$$

**Given parameters:**  $K, R, L, \lambda$ ;

**Unknown parameters:**  $\mu_{krm}, \{C_k\}, \{D_r\}, \{E_l\}$ .

Maximizing the log likelihood of the data under the model with inducing sparsity by using a LASSO penalty, we arrived at:

$$\begin{aligned} & \underset{C_1, \dots, C_K, D_1, \dots, D_R, E_1, \dots, E_L, \mu \in R^{K \times R \times L}}{\text{minimize}} \left\{ \frac{1}{2} \sum_{k=1}^K \sum_{r=1}^R \sum_{l=1}^L \sum_{i \in C_k} \sum_{j \in D_r} \sum_{m \in E_l} (X_{ijm} - \mu_{krm})^2 + \lambda \sum_{k=1}^K \sum_{r=1}^R \sum_{l=1}^L |\mu_{krm}| \right\} \\ & \quad (2) \end{aligned}$$

where  $\lambda$  is a non-negative tuning parameter.

#### 3.3.3 Algorithm

## 4 A spectral interpretation for biclustering

The optimization problem

$$\underset{A^T A = I_K, B^T B = I_R}{\text{maximize}} \quad \|A^T X B\|_F^2 \quad (4)$$

under two additional constraints:

- The elements of the  $k$ th column of A are 0 or  $\frac{1}{\sqrt{n_k}}$  with  $n_k \in Z^+, \sum_{k=1}^K n_k = n$ .
- The elements of the  $k$ th column of B are 0 or  $\frac{1}{\sqrt{p_r}}$  with  $p_r \in Z^+, \sum_{r=1}^R p_r = p$ .

makes (2) equivalent to the biclustering optimization problem (1) when  $\lambda = 0, K = R$ . So, with  $K = R$ , the biclustering problem (1) when  $\lambda = 0$  can be relaxed in order to yield the SVD.

## 5 Tuning parameter selection

$$BIC = np \times \log(RSS) + (q + 1) \log(np)$$

---

**Algorithm 1** Selecting  $k, r, l$ 

---

1.  
**for** each  $time \in [1, T]$  (T-cross validation) **do**
  - (a) Let  $M$  denote a set containing  $npq/T$  elements of the form  $(i, j, m)$ , where  $M$  is disjoint for each iteration.
  - (b) Construct a new  $n \times p \times q$  array,  $X^*$ , for which the elements in  $M$  are "missing" and are imputed using the mean of the non-missing values:
  - (c)  
**for** each value  $(k, r, l)$  of interest: **do**
    - i. Perform sparse tensor clustering of  $X^*$  with  $k$  mode 1,  $r$  mode 2,  $l$  mode 3 clusters.
    - ii. Construct a  $n \times p \times q$  array  $A$  whose  $(i, j, m)$ th element equals the estimated value of  $\mu_{krl}$ , where  $i \in C_k$ ,  $j \in D_r$  and  $m \in E_L$ .  
Calculate the mean squared error that results from estimating the "missing" elements using the corresponding cluster means,

$$\sum_{(i,j,m) \in M} (X_{ijm} - A_{ijm})^2 / |M| \quad (3)$$

- end for**  
**end for**
2. For each value  $(k, r, l)$  that was considered in Step 1(c), compute  $m_{k,r,l}$ , the mean of the quantity (3) across all  $T$  iterations, as well as  $s_{k,r,l}$ , its standard error.
  3. Identify the  $(k, r, l)$  for which  $m_{k,r,l} \leq m_{k+1,r+1,l+1} + s_{k+1,r+1,l+1}$ .
  4. Select the  $(k, r, l)$  from step 3 for which  $k + r + l$  is smallest.
- 

---

**Algorithm 2** Classifying the labels

---

- Initialize  $C_1, \dots, C_K, D_1, \dots, D_R$  and  $E_1, \dots, E_L$  by performing one-way k-means clustering on the columns and on the rows of the data matrix  $X$ .
- repeat**
- (a) Holding  $C_1, \dots, C_K, D_1, \dots, D_R$  and  $E_1, \dots, E_L$  fixed, solve (1) with respect to  $\mu$  using LASSO regression.
  - (b) Holding  $\mu, D_1, \dots, D_R$  and  $E_1, \dots, E_L$  fixed, solve (1) with respect to  $C_1, \dots, C_K$ , by assigning the  $i$ th observation to the row cluster for which  $\sum_{r=1}^R \sum_{l=1}^L \sum_{j \in D_r} \sum_{m \in E_l} (X_{ijm} - \mu_{krl})^2$  is smallest.
  - (c) repeat (a).
  - (d) Holding  $\mu, C_1, \dots, C_K$  and  $E_1, \dots, E_L$  fixed, solve (1) with respect to  $D_1, \dots, D_R$ , by assigning the  $i$ th observation to the column cluster for which  $\sum_{k=1}^K \sum_{l=1}^L \sum_{i \in C_k} \sum_{m \in E_l} (X_{ijm} - \mu_{krl})^2$  is smallest.
  - (e) repeat (a).
  - (f) Holding  $\mu, C_1, \dots, C_K$  and  $D_1, \dots, D_R$  fixed, solve (1) with respect to  $E_1, \dots, E_L$ , by assigning the  $i$ th observation to the cluster of the third dimension for which  $\sum_{k=1}^K \sum_{r=1}^R \sum_{i \in C_k} \sum_{j \in D_r} (X_{ijm} - \mu_{krl})^2$  is smallest.
- until** Convergence
-

## 6 Simulation study

**Standard:** clustering error rate(CER), sparsity rate, sparsity error rate, proportion of correctly identified zeros(C.Zeros) and non-zeros(C.Non-zeros).

### 6.1 Definitions

**Clustering error rate (CER):**

Using adjusted rand index to measure the agreement between any two partitions for the data tensor. In this case, we have three kinds of CER in total: rowCER, columnCER and the CER of the third dimension. To be more specific, consider the rowCER. Denote  $S$  as the set of rows.  $T$  is the true partition of  $S$  and  $J$  is the clustering result with respect to rows. Here,

- $a$ , the number of pairs of elements/labels in  $S$  that in the same subset in  $T$  and in the same subset in  $J$ .
- $b$ , the number of pairs of elements/labels in  $S$  that in the different subsets in  $T$  and in the different subset in  $J$ .

$$rowCER = \frac{a + b}{C_n^2}$$

Intuitively,  $a + b$  can be considered as the number of agreements between  $T$  and  $J$  and  $c + d$  as the number of disagreements between  $T$  and  $J$ .

### 6.2 No bicluster means exactly equal to zero

**Conclusions:** The biclustering with  $\lambda = 0, 200$  leads to consistently better results than independent clustering of the rows and columns.

### 6.3 Some bicluster means exactly equal to zero

**Conclusions:** IP fails to identify any biclusters in this simulation set-up. SSVD and LAS perform comparably in this setting. But by far the best overall performance is achieved by sparse biclustering proposal with a large value of  $\lambda$ .

### 6.4 Multiplicative biclusters

**Conclusions:** SSVD has the best results in this simulation set-up, as in this set-up there are multiplicative biclusters.

### 6.5 Overlapping multiplicative biclusters

**Conclusions:** Both SSVD and sparse biclustering performs pretty good though the set-up violates the assumptions of sparse biclustering.

## A Additional biclustering results of Table 2

True value of (K,R)	n	p	Overall Accuracy	Selected K	Selected R
K=2, R=4	250	100	74%	2(0.0000)	3.7(0.0769)
K=2, R=4	20	50	16%	2.02(0.0318)	2.74(0.1090)

## B Additional simulation biclustering results of Table 3

Method	n	p	Row CER	Column CER	Sparsity Rate
k-means	20	50	0.3621(0.0223)	0.3407(0.0046)	0
Bicluster $\lambda = 0$	20	50	0.3509(0.0220)	0.3217(0.0058)	0
Bicluster $\lambda = 200$	20	50	0.3654(0.0206)	0.4136(0.0155)	0.4455(0.0260)
Bicluster $\lambda = 400$	20	50	0.4841(0.0099)	0.6751(0.0217)	0.8074(0.0553)
Bicluster $\lambda = 800$	20	50	0.4909(0.0061)	0.7478(0.0017)	1(0)
k-means	250	100	0.1202(0.0188)	0.1649(0.0089)	0
Bicluster $\lambda = 0$	250	100	0.1077(0.0177)	0.0958(0.0103)	0
Bicluster $\lambda = 200$	250	100	0.1104(0.0178)	0.0982(0.0105)	0.0610(0.0123)
Bicluster $\lambda = 400$	250	100	0.1119(0.0181)	0.1074(0.0097)	0.1192(0.0161)
Bicluster $\lambda = 800$	250	100	0.1171(0.0185)	0.1358(0.0098)	0.1889(0.0212)

## C Simulation tensor clustering result when k,r,l are given

n	p	q	k	r	l	noise	lambda	iteration	modeI CER	modeII CER	modeIII CER
50	20	20	5	2	2	2	0.01	50	0.1164082	0	0
50	20	20	5	2	2	2	0	50	0.02844082	0	0
50	20	20	5	2	2	0	0	50	0	0	0
50	50	50	3	3	3	0	0	50	0	0	0
50	50	50	3	3	3	3	0	50	0.005240816	0.010693878	0.010318367

Table 1: The simulation result in tensor clustering

## D Two results in tensor clustering under noise = 2

Both of the result are of 0 CERs in three modes.

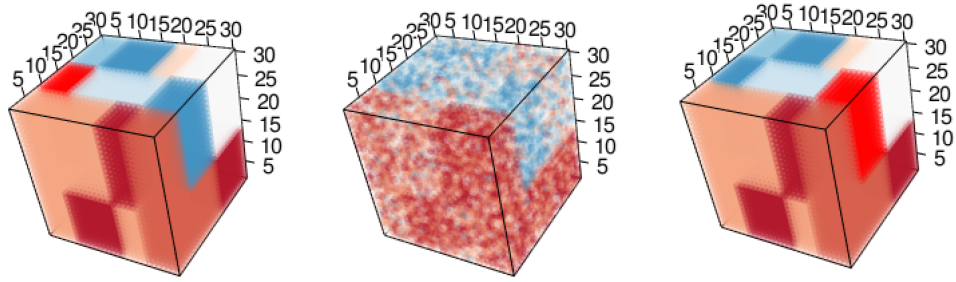


Figure 1: first simulation (truth, input, output)

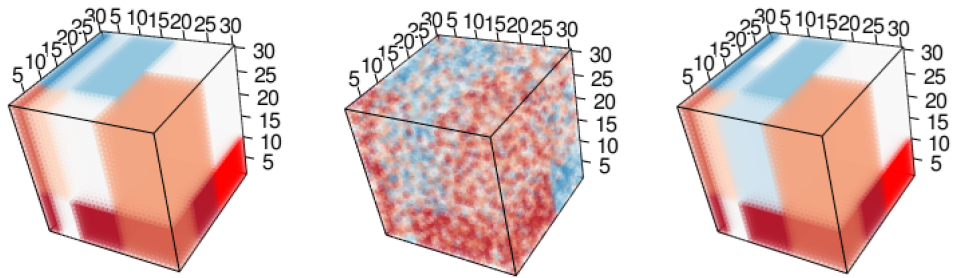


Figure 2: second simulation (truth, input, output)

## E Three examples of selecting $k, r, l$

### Example 1:

True  $k, r, l$ : 3,2,3;

Estimated  $k, r, l$ : 3,2,3.

```
n=30;p=30;q=30;k=3;r=2;l=3
data = get.data(n,p,q,k,r,l,error=2,sort=TRUE)
test = data$x

range.k = 2:4; range.r = 2:4; range.l = 2:4
sparse.choosekrl(test,range.k,range.r,range.l,trace=TRUE)

$estimated_krl
      [,1] [,2] [,3]
[1,]     3     2     3

$results.se
, , L = 2

      R = 2      R = 3      R = 4
K = 2 116.6043 115.0035 113.9306
K = 3 114.1632 117.3038 111.5404
```

```
K = 4 113.7263 123.2596 120.3845
```

```
, , L = 3
```

```
      R = 2      R = 3      R = 4
K = 2 146.6639 144.8598 147.0858
K = 3 141.6482 144.5811 135.6438
K = 4 140.5085 144.1642 150.5537
```

```
, , L = 4
```

```
      R = 2      R = 3      R = 4
K = 2 144.8609 145.5596 152.6614
K = 3 138.9845 140.0927 138.1418
K = 4 136.4456 148.6868 140.0982
```

```
$results.mean
```

```
, , L = 2
```

```
      R = 2      R = 3      R = 4
K = 2 27632.04 27655.18 27665.42
K = 3 27253.61 27260.49 27288.64
K = 4 27272.04 27291.61 27297.49
```

```
, , L = 3
```

```
      R = 2      R = 3      R = 4
K = 2 24671.30 24688.94 24712.19
K = 3 22635.90 22645.72 22676.05
K = 4 22647.39 22672.96 22674.25
```

```
, , L = 4
```

```
      R = 2      R = 3      R = 4
K = 2 24684.75 24704.75 24716.35
K = 3 22655.69 22672.62 22700.28
K = 4 22677.69 22694.95 22698.40
```

## Example 2:

True k,r,l: 4,3,2;

Estimated k,r,l: 4,4,4.

```
n=40;p=30;q=20;k=4;r=3;l=2
data = get.data(n,p,q,k,r,l,error=2,sort=TRUE)
test = data$x
range.k = 2:4; range.r = 2:4; range.l = 2:4
sparse.choosekrl(test,range.k,range.r,range.l,trace=TRUE)
```

```
$bestK
```

```
[1] 4
```

```
$bestR
[1] 4
```

```
$bestL
[1] 4
```

### Example 3:

True k,r,l: 3,3,3

Estimated k,r,l: 3,3,3

```
n=30;p=30;q=30;k=3;r=3;l=3
data = get.data(n,p,q,k,r,l,error=2,sort=TRUE)
test = data$x
range.k = 2:4; range.r = 2:4; range.l = 2:4
sparse.choosekrl(test,range.k,range.r,range.l,trace=TRUE)
```

```
$estimated_krl
      [,1] [,2] [,3]
[1,]      3      3      3
```

```
$results.se
, , L = 2
```

```
      R = 2      R = 3      R = 4
K = 2 219.9770 149.3796 148.4654
K = 3 215.8733 157.0098 158.6694
K = 4 215.5330 153.5680 153.0606
```

```
, , L = 3
```

```
      R = 2      R = 3      R = 4
K = 2 298.1318 231.2240 232.7078
K = 3 261.5005 202.5743 201.8399
K = 4 260.5464 199.7686 195.5321
```

```
, , L = 4
```

```
      R = 2      R = 3      R = 4
K = 2 296.5098 225.4298 231.0916
K = 3 267.6983 197.6522 209.2797
K = 4 258.7976 197.9091 198.2777
```

```
$results.mean
, , L = 2
```

```
      R = 2      R = 3      R = 4
K = 2 34958.74 30413.91 30427.48
K = 3 32687.69 27287.17 27309.28
K = 4 32693.21 27291.23 27302.11
```

```
, , L = 3
```



	R = 2	R = 3	R = 4
K = 2	31339.87	26439.73	26458.68
K = 3	28157.20	22273.70	22298.02
K = 4	28160.83	22278.93	22293.68

, , L = 4

	R = 2	R = 3	R = 4
K = 2	31345.99	26447.07	26477.48
K = 3	28184.74	22288.72	22305.65
K = 4	28195.24	22296.37	22326.64