

# NOISE-RESISTANT BICLUSTER RECOGNITION

---

Huan Sun\*, Gengxin Miao#, Xifeng Yan\*

\*Computer Science Department

#Electrical and Computer Engineering Department  
University of California, Santa Barbara



# Problem

- Biclustering: simultaneously cluster rows and columns in a matrix.

		Genes							
Conditions	0	0	0	0	0	0	0	0	0
	0	1	1	1	0	0	0	0	0
	0	-1	-1	-1	0	0	0	0	0
	0	1	1	1	0	0	0	0	0
	0	0	0	-1	1	1	-1	0	0
	0	0	0	1	-1	-1	1	0	0
	0	0	0	-1	1	1	-1	0	0
	0	0	0	0	0	0	0	0	0

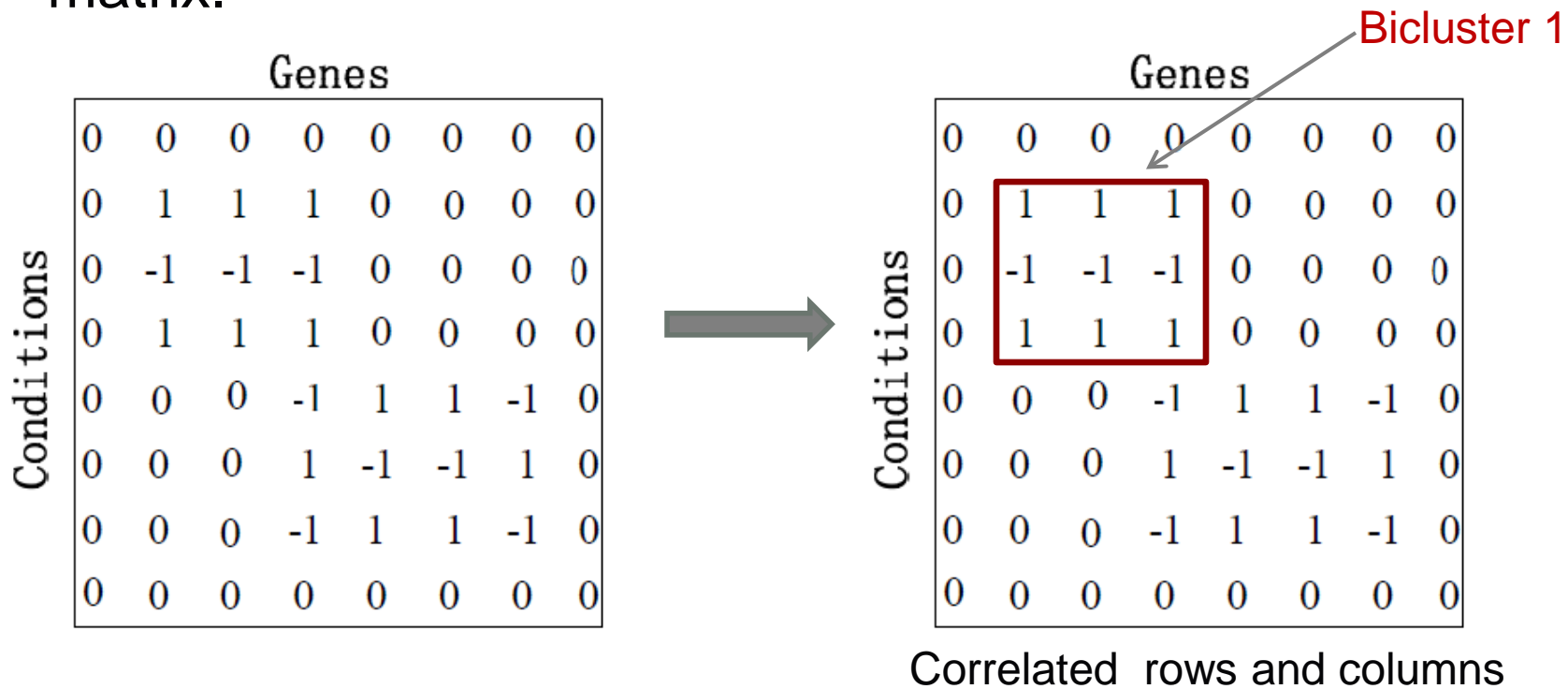
1: upregulating

0: unchanged

-1: downregulating

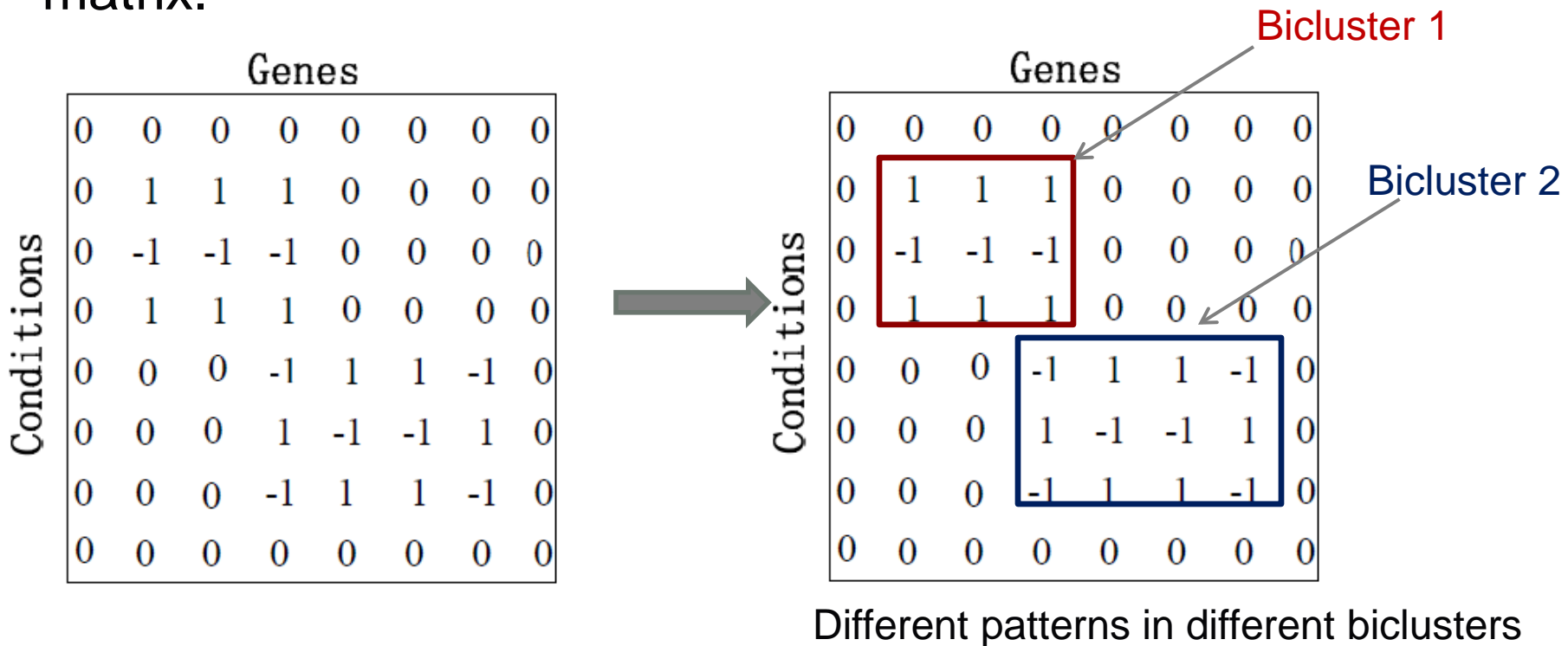
# Problem

- Biclustering: simultaneously cluster rows and columns in a matrix.



# Problem

- Biclustering: simultaneously cluster rows and columns in a matrix.



# Many existing algorithms

---

- Combinatorial algorithms: *e.g.*, QUBIC, COALESCE *etc.*
  - Probabilistic models: *e.g.*, SAMBA, FABIA *etc.*
  - Matrix factorization: *e.g.*, SSVD, S4VD *etc.*
-

# Challenges

Easy but not the real case!!!

Genes

0	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0
0	-1	-1	-1	0	0	0	0
0	1	1	1	0	0	0	0
0	0	0	-1	1	1	-1	0
0	0	0	1	-1	-1	1	0
0	0	0	-1	1	1	-1	0
0	0	0	0	0	0	0	0

- Noise

Genes							
0	0	0	0	0	0	0	0
0	1	1	1	0	<del>x</del>	0	0
0	-1	<del>x</del>	-1	0	0	0	<del>x</del>
0	1	1	1	-1	-1	1	0
0	-1	-1	<del>x</del>	1	1	-1	0
0	0	1	1	-1	<del>x</del>	1	0
0	0	-1	-1	1	1	-1	0
<del>x</del>	0	0	0	0	0	0	0

- Overlap

Genes							
0	0	0	0	0	0	0	0
0	1	1	1	0	-1	0	0
0	-1	0	-1	0	0	0	1
0	1	<del>1</del>	<del>1</del>	-1	-1	1	0
0	-1	<del>-1</del>	<del>0</del>	1	1	-1	0
0	0	1	1	-1	0	1	0
0	0	-1	-1	1	1	-1	0
1	0	0	0	0	0	0	0

---

Since neural networks have achieved great successes in data classification,  
*can neural networks do biclustering?*

---

---

Since neural networks have achieved great successes in data classification.

*can neural networks do biclustering?*

**Yes!**

**Not only that, it can significantly outperform  
all the existing methods.**

---



# Intuition

feature 1

Genes

Conditions

0	0	0	0	0	0	0	0
0	1	1	1	0	-1	0	0
0	-1	0	-1	0	0	0	1
0	1	1	1	-1	-1	1	0
0	-1	-1	0	1	1	-1	0
0	0	1	1	-1	0	1	0
0	0	-1	-1	1	1	-1	0
1	0	0	0	0	0	0	0

repetition of feature 1

Genes

Conditions

0	0	0	0	0	0	0	0
0	1	1	1	0	-1	0	0
0	-1	0	-1	0	0	0	1
0	1	1	1	-1	-1	1	0
0	-1	-1	0	1	1	-1	0
0	0	1	1	-1	0	1	0
0	0	-1	-1	1	1	-1	0
1	0	0	0	0	0	0	0

# Intuition

Similarly,

feature 2

Genes

0	0	0	0	0	0	0	0
0	1	1	1	0	-1	0	0
0	-1	0	-1	0	0	0	1
0	1	1	1	-1	-1	1	0
0	-1	-1	0	1	1	-1	0
0	0	1	1	-1	0	1	0
0	0	-1	-1	1	1	-1	0
1	0	0	0	0	0	0	0

Conditions

repetition of feature 2

Genes

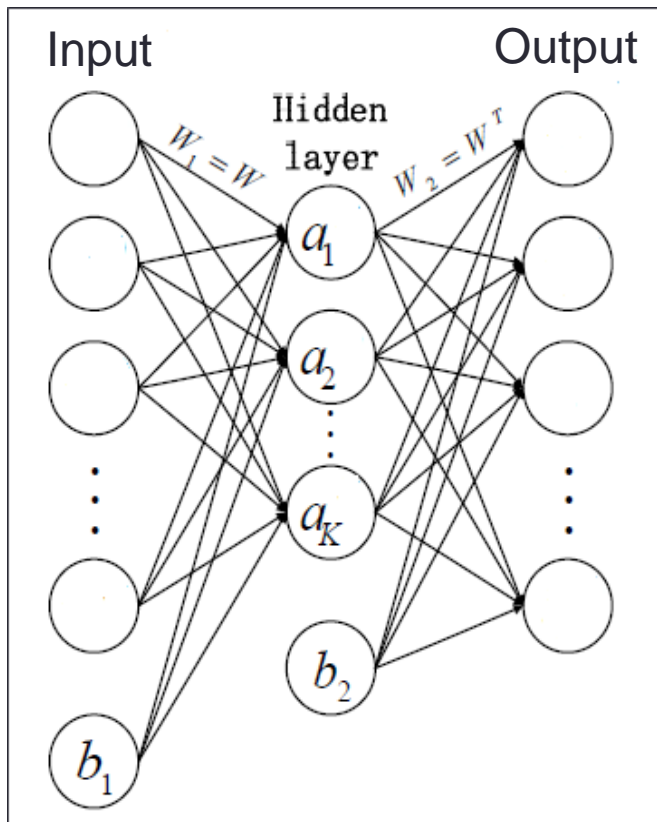
0	0	0	0	0	0	0	0
0	1	1	1	0	-1	0	0
0	-1	0	-1	0	0	0	1
0	1	1	1	-1	-1	1	0
0	-1	-1	0	1	1	-1	0
0	0	1	1	-1	0	1	0
0	0	-1	-1	1	1	-1	0
1	0	0	0	0	0	0	0

Conditions



# Feature learning: Sparse Autoencoder (SAE)

Map Sparse Autoencoder (SAE) to biclustering:



One hidden neuron corresponds to one feature

**$W$** : the contribution of each row in activating hidden neurons

⇒ membership of a row in hidden neurons

**$A = [a_1, a_2, \dots, a_K]^T$**  : activation of hidden neurons for a column

⇒ membership of a column in hidden neurons

Row and column members in the same hidden neuron compose a bicluster.

# Our Method: AutoDecoder (AD)

Based on Sparse Autoencoder, we enhance:

- **Robustness against noise**

*Intuition:*

- Noise outside bicluster pattern:

0	0	0	1	0	0	1	0
0	1	1	1	1	1	0	0
-1	-1	-1	-1	-1	-1	0	0
0	1	1	1	1	1	0	0
0	0	0	0	-1	0	0	1

- Noise inside bicluster pattern:

0	0	0	0	0	0	0	0
0	1	1	1	0	1	0	0
0	-1	0	-1	-1	-1	0	0
0	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0

Bicluster pattern

# Our Method: AutoDecoder (AD)

Based on Sparse Autoencoder, we enhance:

- **Robustness against noise**

*Intuition:*

- Noise outside bicluster pattern:

0	0	0	1	0	0	1	0
0	1	1	1	1	1	0	0
-1	-1	-1	-1	-1	-1	0	0
0	1	1	1	1	1	0	0
0	0	0	0	-1	0	0	1

Allow more false negative reconstruction errors

0	0	0	0	0	0	0	0
0	1	1	1	1	1	0	0
0	-1	-1	-1	-1	-1	0	0
0	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0

- Noise inside bicluster pattern:

0	0	0	0	0	0	0	0
0	1	1	1	0	1	0	0
0	-1	0	-1	-1	-1	0	0
0	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0

Allow more false positive reconstruction errors

0	0	0	0	0	0	0	0
0	1	1	1	1	1	0	0
0	-1	-1	-1	-1	-1	0	0
0	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0

“Decode”

# Our Method: AutoDecoder (AD)

Based on sparse autoencoder, we enhance:

- **Robustness against overlap**

*Intuition:*

		Genes							
		0	0	0	0	0	0	0	0
		0	1	1	1	0	-1	0	0
		0	-1	0	-1	0	0	0	1
Conditions	Condition i	0	1	1	1	-1	-1	1	0
		0	-1	-1	0	1	1	-1	0
		0	0	1	1	-1	0	1	0
	Condition j	0	0	-1	-1	1	1	-1	0
		1	0	0	0	0	0	0	0

Condition i } similar → likely in the same bicluster

Condition j }

# Our Method: AutoDecoder (AD)

Based on sparse autoencoder, we enhance:

- **Robustness against overlap**

*Intuition:*

		Genes							
		0	0	0	0	0	0	0	0
Conditions	Condition k	0	1	1	1	0	-1	0	0
		0	-1	0	-1	0	0	0	1
	Condition i	0	1	1	1	-1	-1	1	0
		0	-1	-1	0	1	1	-1	0
		0	0	1	1	-1	0	1	0
	Condition j	0	0	-1	-1	1	1	-1	0
		1	0	0	0	0	0	0	0

similar  $\longrightarrow$  likely in the same bicluster

Condition i is similar to conditions in different biclusters; then condition i should simultaneously belong to these biclusters as well (**overlap**).



# Our Method: AutoDecoder (AD)

- Objective function

SAE

$$\begin{aligned}
 & \arg \min_{W, b_1, b_2} \mathcal{H} \\
 &= \underbrace{\frac{1}{2N} \sum_{i=1}^N \sum_{m=1}^M (\hat{x}_m^{(i)} - x_m^{(i)})^2}_{\text{(I)}} \\
 &+ \underbrace{\beta \sum_{k=1}^K \text{KL}(\rho_k \| \hat{\rho}_k)}_{\text{(II)}} \\
 &+ \underbrace{\lambda \|W\|_1}_{\text{(III)}}
 \end{aligned}$$

AD

$$\begin{aligned}
 & \arg \min_{W, b_1, b_2} \mathcal{H} \\
 &= \underbrace{\frac{1}{2N} \sum_{i=1}^N \sum_{m=1}^M [I_{m,i} + \alpha(1 - I_{m,i})] (\hat{x}_m^{(i)} - x_m^{(i)})^2}_{\text{(I) Robustness against noise}} \\
 &+ \underbrace{\beta \sum_{k=1}^K \text{KL}(\rho_k \| \hat{\rho}_k)}_{\text{(II) Sparsity term as in SAE}} \\
 &+ \underbrace{\frac{\gamma}{2} \sum_{k=1}^K \sum_{i \neq j} S_{i,j} (|W_{k,i}| - |W_{k,j}|)^2}_{\text{(III) Robustness against overlap}} \\
 &+ \underbrace{\lambda \|W\|_1}_{\text{(IV) Overfitting regularizer as in SAE}}
 \end{aligned}$$

# Our Method: AutoDecoder (AD)

- Solution

Backpropagation + L-BFGS optimization algorithm:

Error terms:

$$\delta_{m,n}^{(2)} = [I_{m,n} + \alpha(1 - I_{m,n})](\hat{x}_m^{(n)} - x_m^{(n)})$$

$$\delta_{k,n}^{(1)} = \left[ \sum_{m=1}^M W_{k,m} \delta_{m,n}^{(2)} + \beta \left( -\frac{\rho_k}{\hat{\rho}_k} + \frac{1 - \rho_k}{1 - \hat{\rho}_k} \right) \right] a_k^{(n)} (1 - a_k^{(n)})$$

Gradients:

$$\frac{\partial \mathcal{H}}{\partial b_{1k}} = \frac{1}{N} \sum_{n=1}^N \delta_{k,n}^{(1)}$$

$$\frac{\partial \mathcal{H}}{\partial b_{2m}} = \frac{1}{N} \sum_{n=1}^N \delta_{m,n}^{(2)}$$

$$\frac{\partial \mathcal{H}}{\partial W_{k,m}} = \frac{1}{N} \sum_{n=1}^N (\delta_{k,n}^{(1)} x_m^{(n)} + \delta_{m,n}^{(2)} a_k^{(n)})$$

$$+ (\gamma L_{m,(\cdot)} W_{k,(\cdot)}^T + \lambda) \text{sgn}(W_{k,m})$$

# Experiments

---

- Existing algorithms
    - **QUBIC** [Li *et al.*, Nucleic Acids Research, 2009]
      - Combinatorial algorithm
    - **COALESCE** [Huttenhower *et al.*, Bioinformatics, 2009]
      - Combinatorial algorithm
    - **FABIA** [Hochreiter *et al.*, Bioinformatics, 2010]
      - Probabilistic models
    - **S4VD** [Sill *et al.*, Bioinformatics, 2011]
      - Matrix factorization
-

# Experiments

---

- Evaluation measures

Commonly used measures in biclustering: Relevance and Recovery.

- Relevance: How relevant the discovered biclusters are to the true ones

- Recovery: To what degree the true biclusters are recovered

- F score: 
$$F = \frac{2 \times \text{Relevance} \times \text{Recovery}}{\text{Relevance} + \text{Recovery}}$$

- Gene set enrichment analysis on real data sets.
-

# Experiments

---

- Real data sets
    - Diffuse Large-B-Cell Lymphoma (DLBCL)
      - 3795 genes, 58 samples
    - Lung Cancer
      - 12625 genes, 56 samples
    - Breast Cancer
      - 1213 genes, 97 samples
    - Multiple Tissue
      - 5565 genes, 102 samples
-

# Experiments

- F score on four real data sets

Methods	DLBCL	Lung Cancer	Breast Cancer	Multiple Tissue
AD	<b>0.50</b>	<b>0.92</b>	0.43	<b>0.82</b>
QUBIC	0.34	0.62	0.21	0.63
COALESCE	0.38	0.36	0.41	0.47
FABIA	0.27	0.85	<b>0.57</b>	0.77
S4VD	0.28	0.72	0.49	0.10
<b>Relative Improvement</b>	<b><math>\geq 31\%</math></b>	<b><math>\geq 8.2\%</math></b>	\	<b><math>\geq 6.5\%</math></b>

# Experiments

- Biological significance evaluation

Data sets	P-value of gene sets discovered by AD (the smaller, the better)
Multiple Tissue	$1.2 \times 10^{-20} \sim 6.9 \times 10^{-5}$
DLBCL	$6.3 \times 10^{-10} \sim 5.3 \times 10^{-4}$
Lung Cancer	$9.2 \times 10^{-26} \sim 2.2 \times 10^{-6}$

The gene sets discovered by AD are also biologically significant.

# Experiments

---

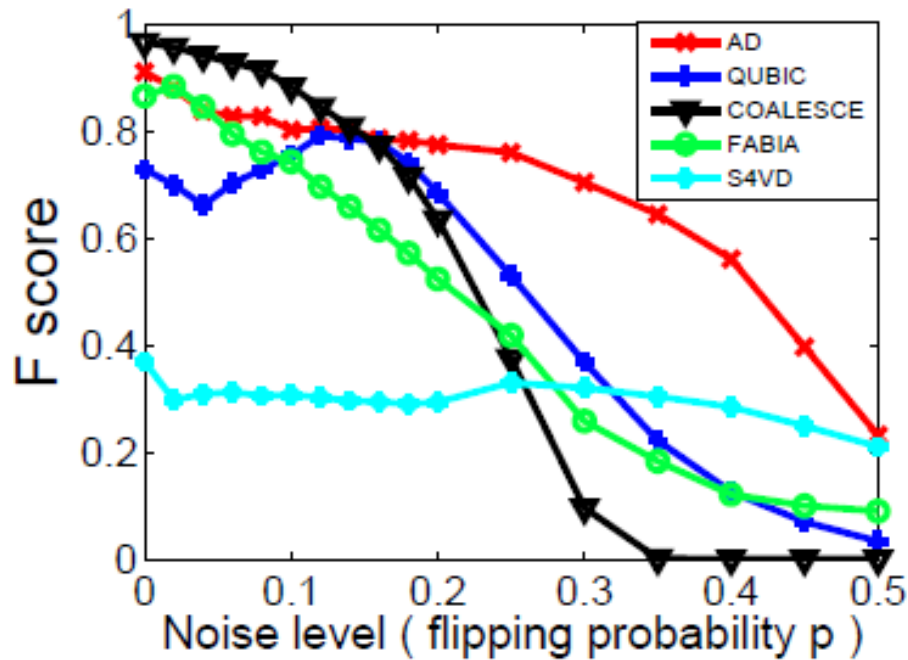
- Synthetic data sets
    - Dataset synthesis procedure:
      1. Set matrix size as  $100 \times 500$ , initially filled with 0's;
      2. Implant one bicluster by:
        - Select the number of rows  $r$  in this bicluster from  $[10, 30]$ ;
        - Select the number of columns  $c$  in this bicluster from  $[50, 100]$ ;
        - Randomly choose  $r$  rows and  $c$  columns as the members of the bicluster;
        - Fill this bicluster with 1's.
      3. Totally implant  $K$  biclusters;
      4. Inject noise to the matrix by flipping the 1's inside a bicluster to 0 with probability  $p$  and flipping the 0's outside the biclusters to 1 or -1 respectively with probability  $p/2$ .
    - Vary  $K$  and  $p$  in our experiments ( $K$  controls the overlap while  $p$  denotes the noise degree).
-



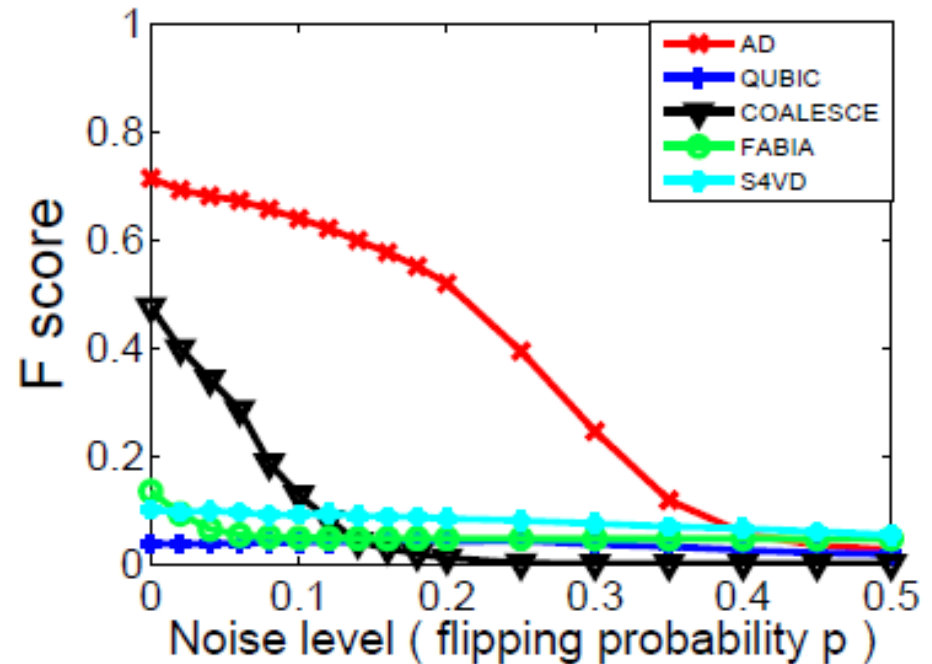
# Experiments

- Noise-and-overlap resistance testing on synthetic data sets

K=6



K=20



AD significantly outperforms other methods when there are more biclusters and heavier noise.

# Conclusions

---

- A novel model, AutoDecoder (AD), for biclustering
    - neural networks ----->feature learning-----> biclustering
    - more robust against noise and overlap in both real and synthetic data sets.
  - Project homepage:
    - <http://grafica.cs.ucsb.edu/autodecoder/>
    - Source code and data sets
-

*Thank You!*

---