DATA MINING LECTURE 8

Clustering Validation
Minimum Description Length
Information Theory
Co-Clustering

CLUSTERING VALIDITY

Cluster Validity

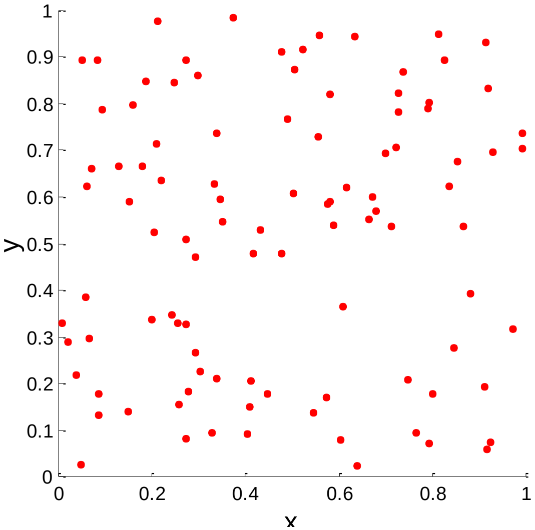
- How do we evaluate the “goodness” of the resulting clusters?

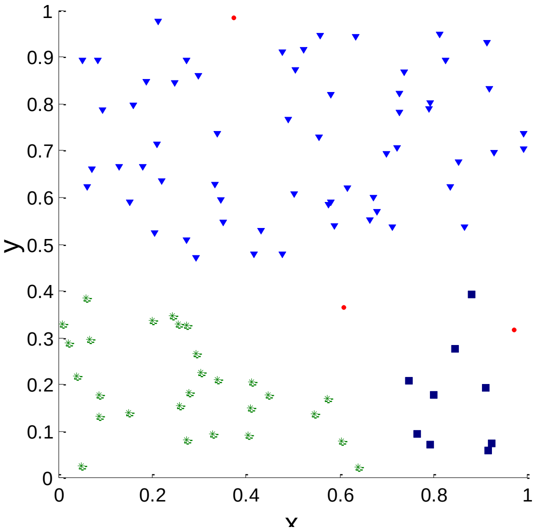
- But “clustering lies in the eye of the beholder”!

- Then why do we want to evaluate them?
  - To avoid finding patterns in noise
  - To compare clustering algorithms
  - To compare two clusterings
  - To compare two clusters

Clusters found in Random Data

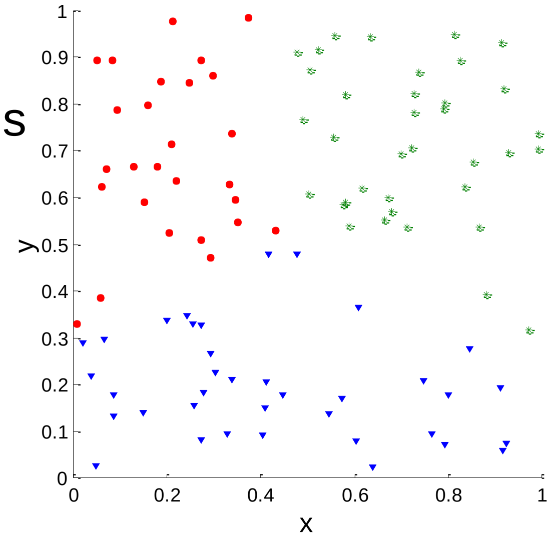
Random Points

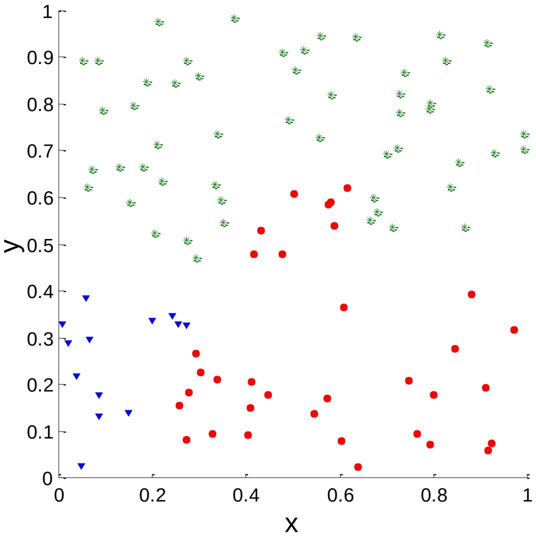




DBSCAN

K-means





Complete Link

Different Aspects of Cluster Validation

1. Determining the clustering tendency of a set of data, i.e., distinguishing whether non-random structure actually exists in the data.
2. Comparing the results of a cluster analysis to externally known results, e.g., to externally given class labels.
3. Evaluating how well the results of a cluster analysis fit the data *without* reference to external information.

   - Use only the data
4. Comparing the results of two different sets of cluster analyses to determine which is better.
5. Determining the 'correct' number of clusters.

For 2, 3, and 4, we can further distinguish whether we want to evaluate the entire clustering or just individual clusters.

Measures of Cluster Validity

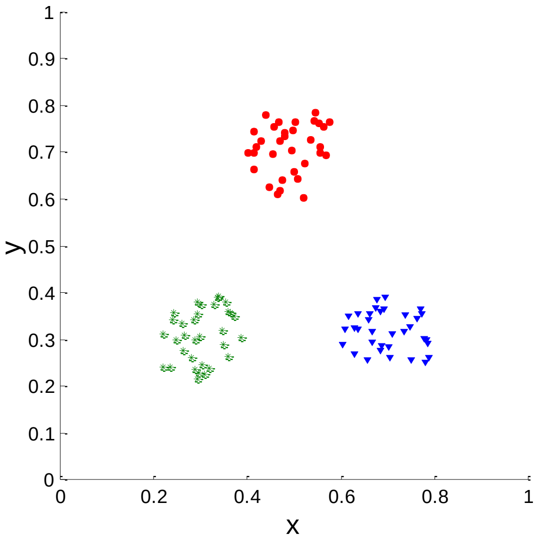
- Numerical measures that are applied to judge various aspects of cluster validity, are classified into the following three types.
  - External Index: Used to measure the extent to which cluster labels match externally supplied class labels.
    - E.g., entropy, precision, recall
  - Internal Index: Used to measure the goodness of a clustering structure without reference to external information.
    - E.g., Sum of Squared Error (SSE)
  - Relative Index: Used to compare two different clusterings or clusters.
    - Often an external or internal index is used for this function, e.g., SSE or entropy
- Sometimes these are referred to as criteria instead of indices
  - However, sometimes criterion is the general strategy and index is the numerical measure that implements the criterion.

Measuring Cluster Validity Via Correlation

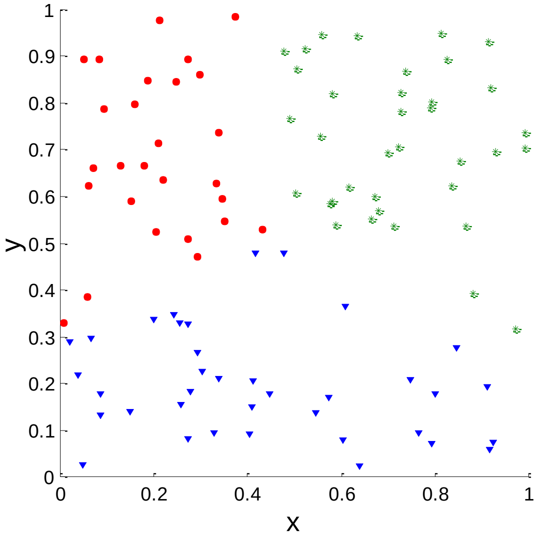
- Two matrices
  - Similarity or Distance Matrix
    - One row and one column for each data point
    - An entry is the similarity or distance of the associated pair of points
  - "Incidence" Matrix
    - One row and one column for each data point
    - An entry is 1 if the associated pair of points belong to the same cluster
    - An entry is 0 if the associated pair of points belongs to different clusters
- Compute the correlation between the two matrices
  - Since the matrices are symmetric, only the correlation between n(n-1) / 2 entries needs to be calculated.
- High correlation (positive for similarity, negative for distance) indicates that points that belong to the same cluster are close to each other.
- Not a good measure for some density or contiguity based clusters.

Measuring Cluster Validity Via Correlation

- Correlation of incidence and proximity matrices for the K-means clusterings of the following two data sets.



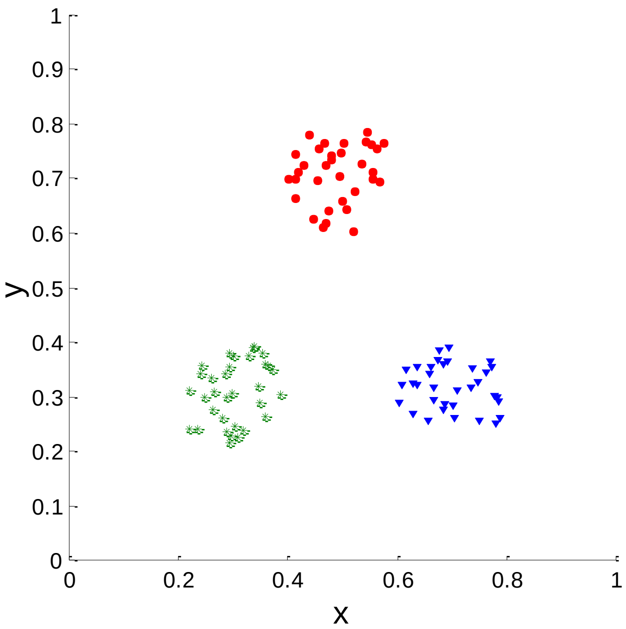
Corr = -0.9235

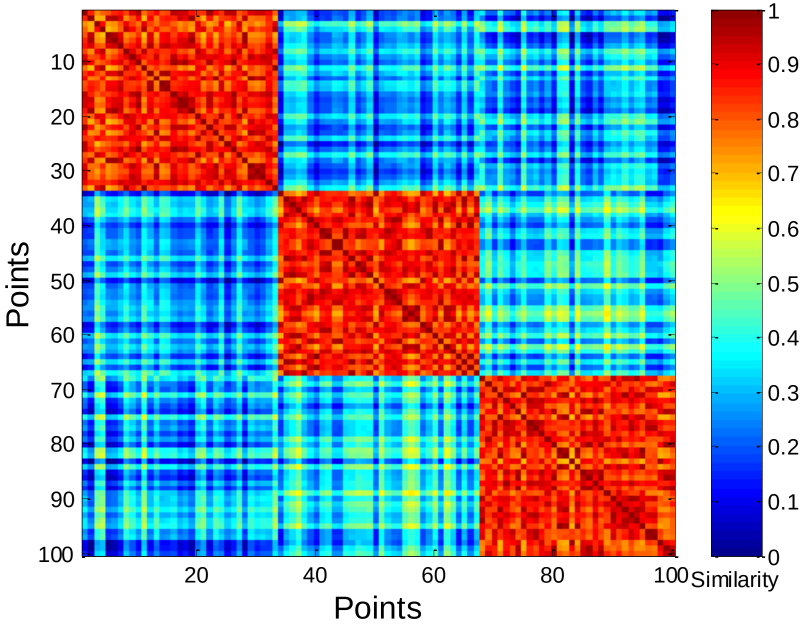


Corr = -0.5810

Using Similarity Matrix for Cluster Validation

- Order the similarity matrix with respect to cluster labels and inspect visually.

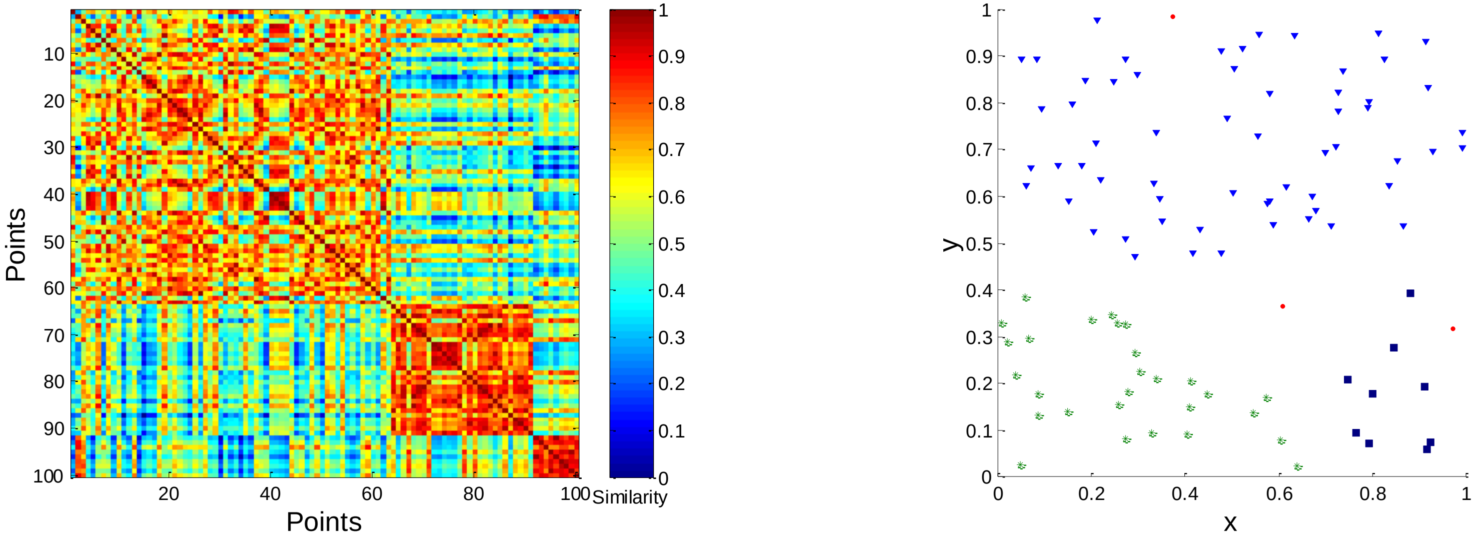




$$sim(i,j) = 1 - \frac{d_{ij} - d_{min}}{d_{max} - d_{min}}$$

Using Similarity Matrix for Cluster Validation

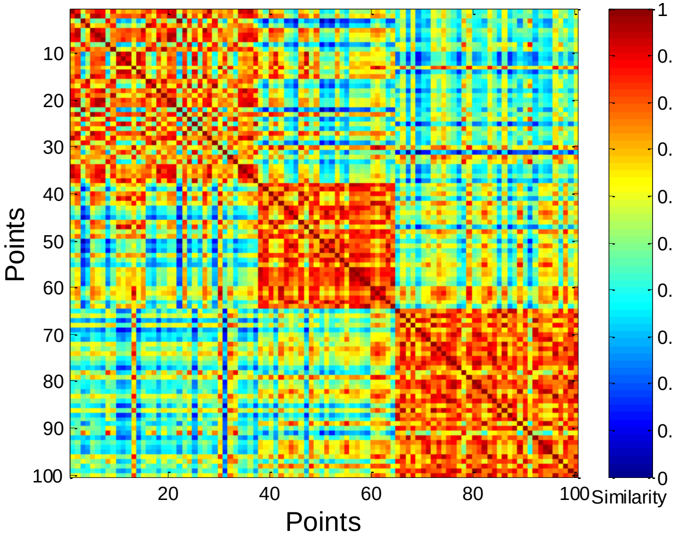
- Clusters in random data are not so crisp

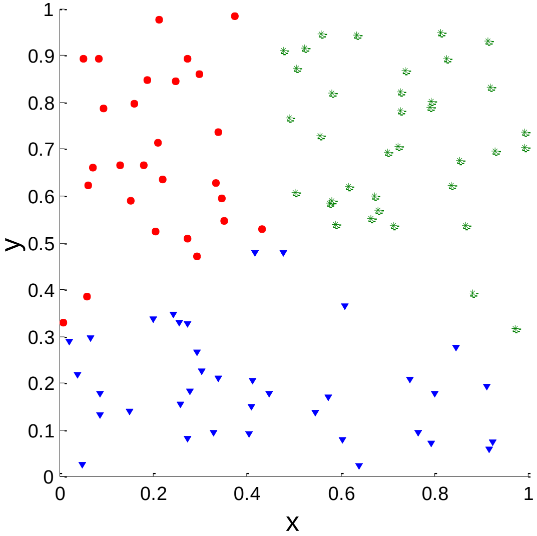


DBSCAN

Using Similarity Matrix for Cluster Validation

- Clusters in random data are not so crisp

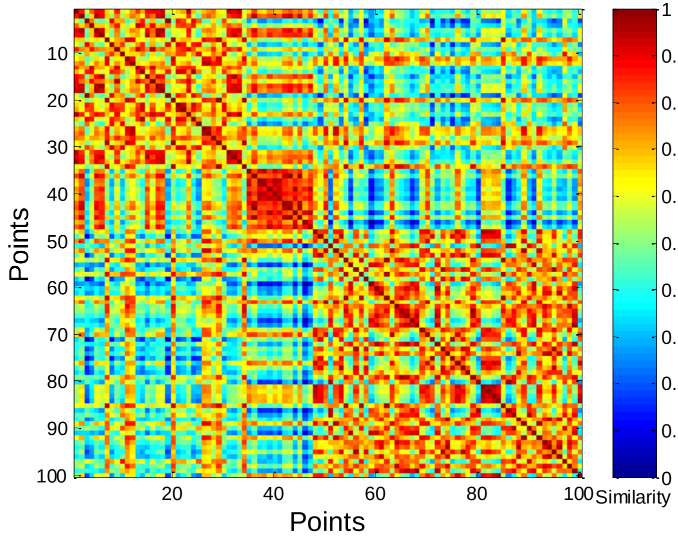


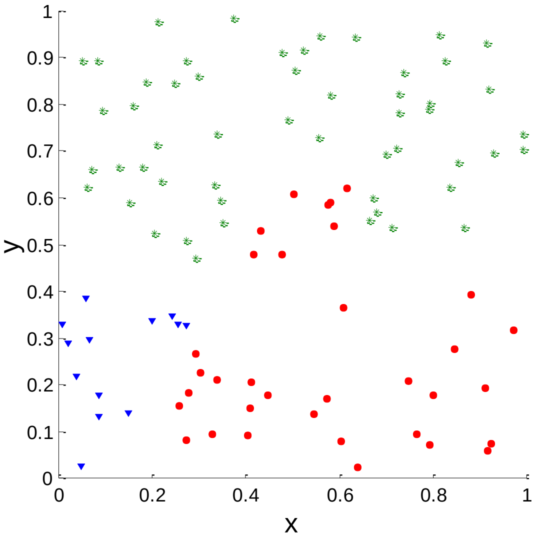


K-means

Using Similarity Matrix for Cluster Validation

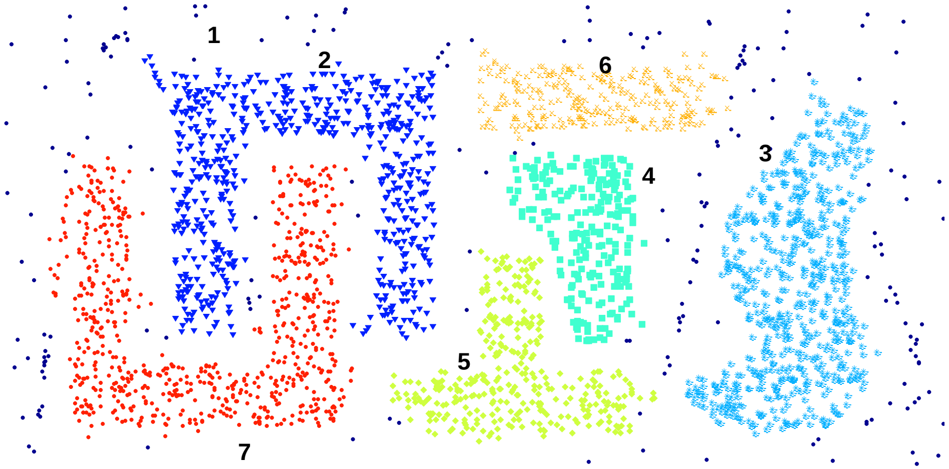
- Clusters in random data are not so crisp

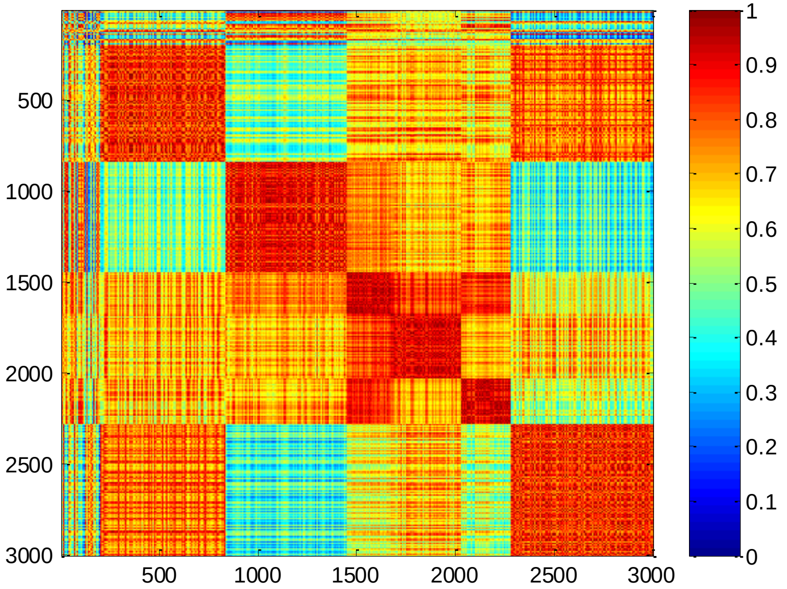




Complete Link

Using Similarity Matrix for Cluster Validation



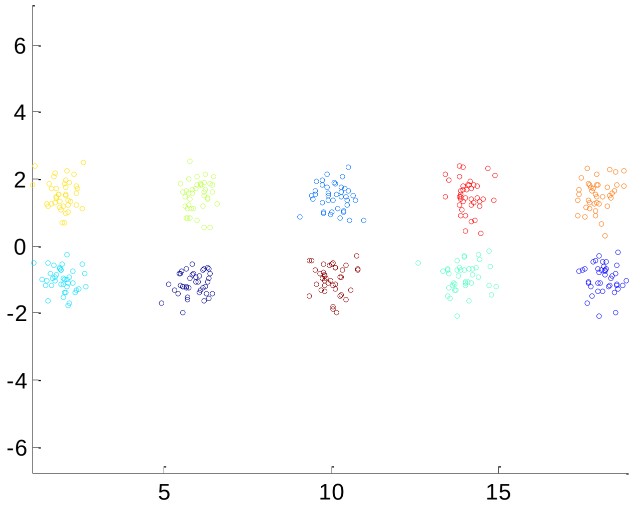


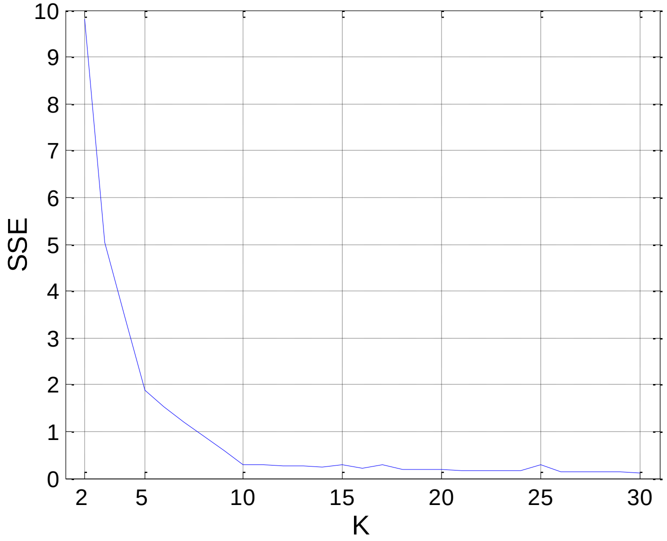
DBSCAN

- Clusters in more complicated figures aren’t well separated

Internal Measures: SSE

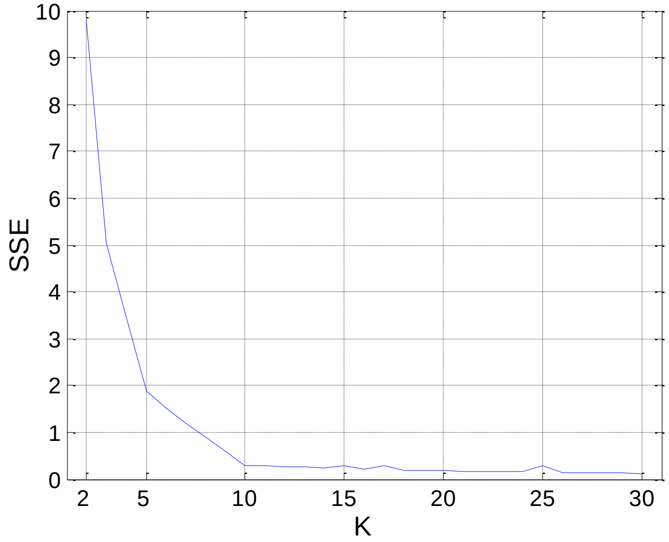
- Internal Index: Used to measure the goodness of a clustering structure without reference to external information
  - Example: SSE
- SSE is good for comparing two clusterings or two clusters (average SSE).
- Can also be used to estimate the number of clusters





Estimating the “right” number of clusters

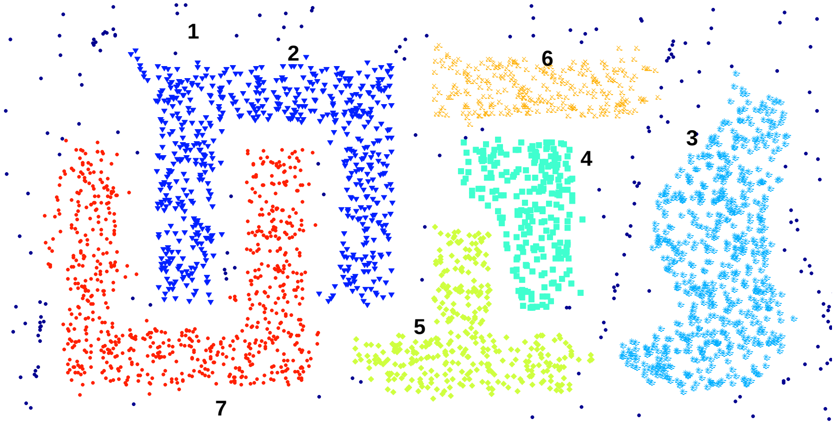
- Typical approach: find a “knee” in an internal measure curve.

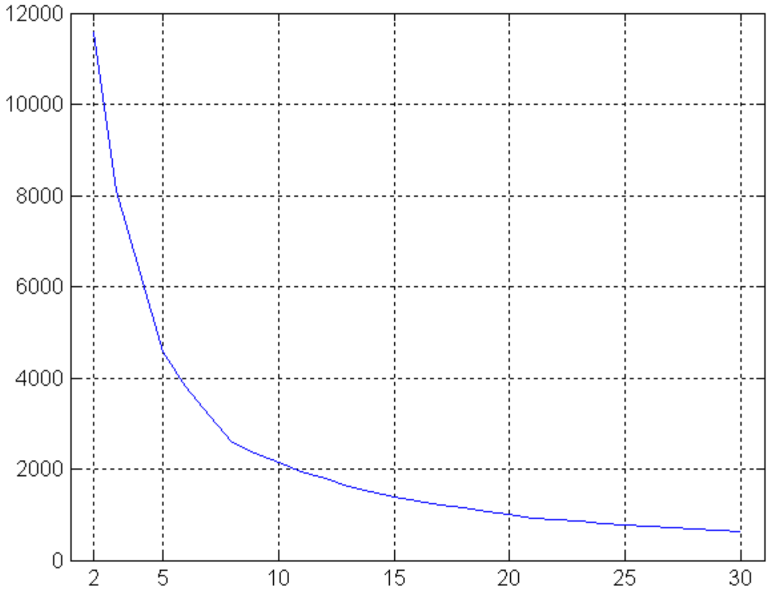


- Question: why not the k that minimizes the SSE?
    - Forward reference: minimize a measure, but with a “simple” clustering
- Desirable property: the clustering algorithm does not require the number of clusters to be specified (e.g., DBSCAN)

Internal Measures: SSE

- SSE curve for a more complicated data set





SSE of clusters found using K-means

Internal Measures: Cohesion and Separation

- Cluster Cohesion: Measures how closely related are objects in a cluster
- Cluster Separation: Measure how distinct or well-separated a cluster is from other clusters
- Example: Squared Error
  - Cohesion is measured by the within cluster sum of squares (SSE)

$$WSS = \sum_{i} \sum_{x \in C_i} (x - c_i)^2$$

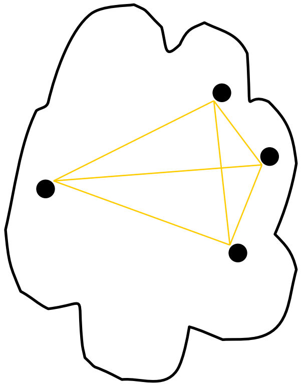
  - Separation is measured by the between cluster sum of squares

$$BSS = \sum_{i} m_i (c - c_i)^2$$

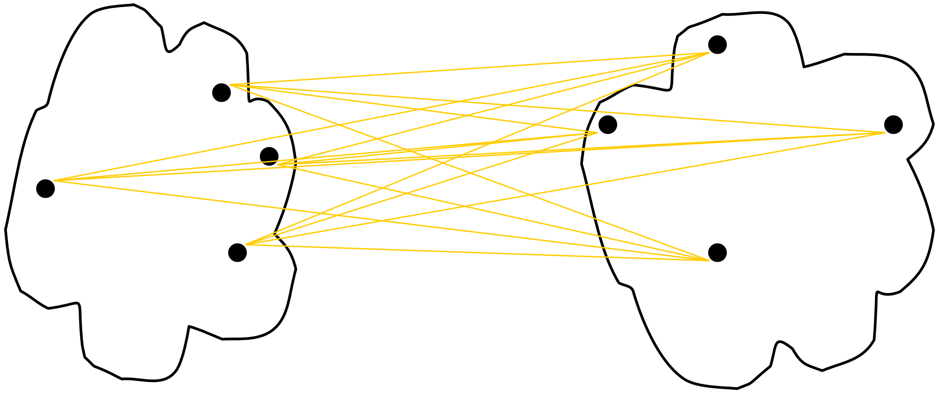
    - Where m_i is the size of cluster i

Internal Measures: Cohesion and Separation

- A proximity graph based approach can also be used for cohesion and separation.
  - Cluster cohesion is the sum of the weight of all links within a cluster.
  - Cluster separation is the sum of the weights between nodes in the cluster and nodes outside the cluster.



cohesion



separation

Internal measures – caveats

- Internal measures have the problem that the clustering algorithm did not set out to optimize this measure, so it is will not necessarily do well with respect to the measure.

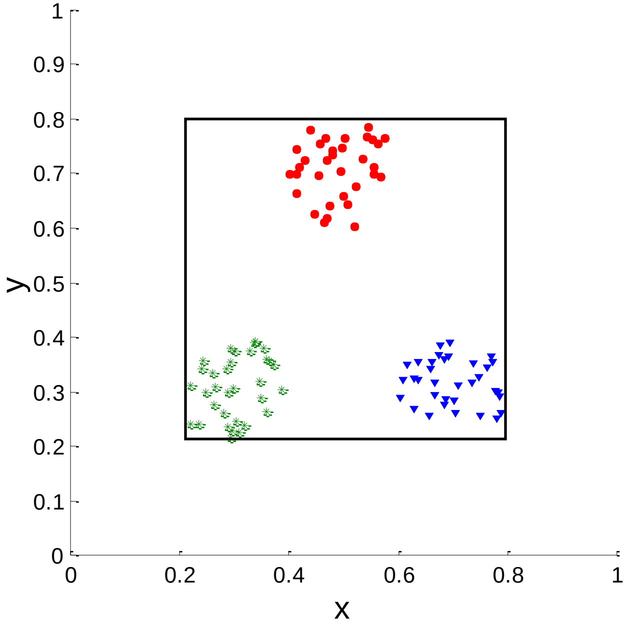
- An internal measure can also be used as an objective function for clustering

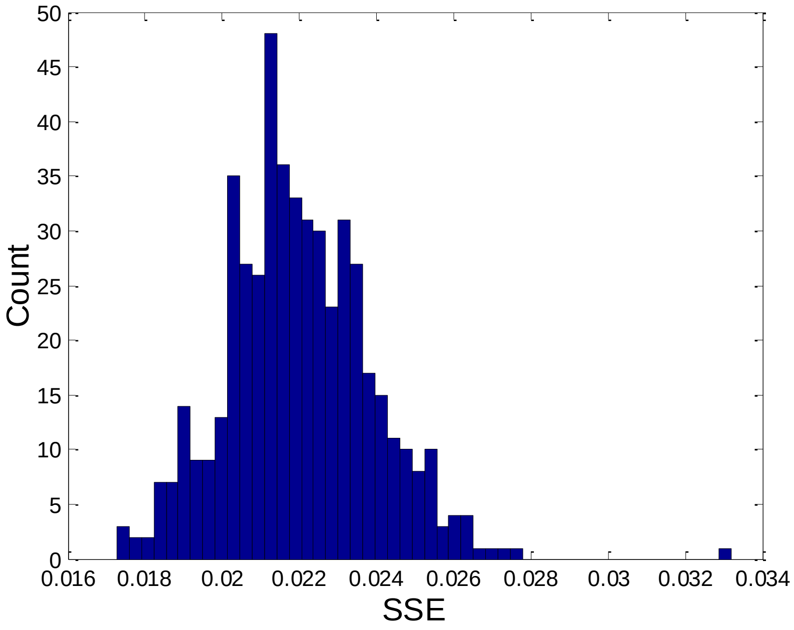
Framework for Cluster Validity

- Need a framework to interpret any measure.
  - For example, if our measure of evaluation has the value, 10, is that good, fair, or poor?
- Statistics provide a framework for cluster validity
  - The more “non-random” a clustering result is, the more likely it represents valid structure in the data
  - Can compare the values of an index that result from random data or clusterings to those of a clustering result.
    - If the value of the index is unlikely, then the cluster results are valid
- For comparing the results of two different sets of cluster analyses, a framework is less necessary.
  - However, there is the question of whether the difference between two index values is significant

Statistical Framework for SSE

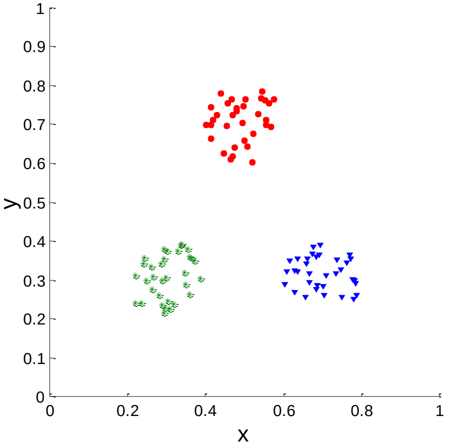
- Example
  - Compare SSE of 0.005 against three clusters in random data
  - Histogram of SSE for three clusters in 500 random data sets of 100 random points distributed in the range 0.2 – 0.8 for x and y
    - Value 0.005 is very unlikely



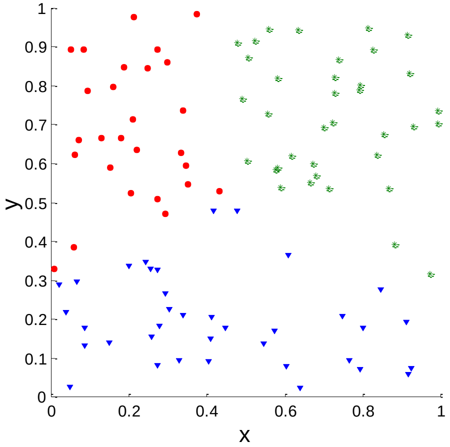


Statistical Framework for Correlation

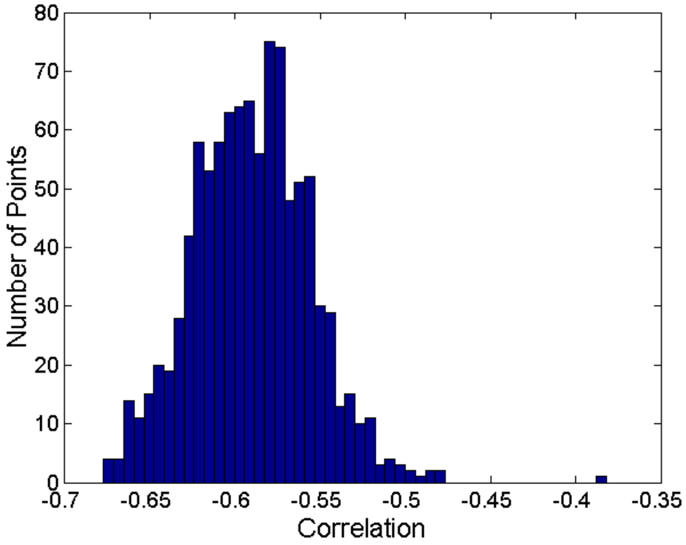
- Correlation of incidence and proximity matrices for the K-means clusterings of the following two data sets.



Corr = -0.9235



Corr = -0.5810



Empirical p-value

- If we have a measurement v (e.g., the SSE value)
- ..and we have N measurements on random datasets
- …the empirical p-value is the fraction of measurements in the random data that have value less or equal than value v (or greater or equal if we want to maximize)
    - i.e., the value in the random dataset is at least as good as that in the real data

- We usually require that p-value ≤ 0.05

- Hard question: what is the right notion of a random dataset?

External Measures for Clustering Validity

- Assume that the data is labeled with some class labels
  - E.g., documents are classified into topics, people classified according to their income, senators classified as republican or democrat.
- In this case we want the clusters to be homogeneous with respect to classes
  - Each cluster should contain elements of mostly one class
  - Also each class should ideally be assigned to a single cluster
- This does not always make sense
  - Clustering is not the same as classification
- But this is what people use most of the time

Measures

- n = number of points
- m_i = points in cluster i
- c_j = points in class j
- m_{ij}= points in cluster i coming from class j
- $p_{ij} = m_{ij}/m_i$= prob of element from class j in cluster i
- Entropy:
  - Of a cluster i: $e_i = -\sum_{j=1}^{L} p_{ij} \log p_{ij}$
    - Highest when uniform, zero when single class
  - Of a clustering: $e = \sum_{i=1}^{K} \frac{m_i}{n} e_i$
- Purity:
  - Of a cluster i: $p_i = \max_j p_{ij}$
  - Of a clustering: $purity = \sum_{i=1}^{K} \frac{m_i}{n} p_i$

	Class 1	Class 2	Class 3	
Cluster 1	m_{11}	m_{12}	m_{13}	m_1
Cluster 2	m_{21}	m_{22}	m_{23}	m_2
Cluster 3	m_{31}	m_{32}	m_{33}	m_3
	c_1	c_2	c_3	n

Measures

- Precision:
  - Of cluster i with respect to class j: $Prec(i,j) = p_{ij}$
    - For the precision of a clustering you can take the maximum
- Recall:
  - Of cluster i with respect to class j: $Rec(i,j) = \frac{m_{ij}}{c_j}$
    - For the precision of a clustering you can take the maximum
- F-measure:
  - Harmonic Mean of Precision and Recall:

$$F(i,j) = \frac{2 * Prec(i,j) * Rec(i,j)}{Prec(i,j) + Rec(i,j)}$$

Good and bad clustering

	Class 1	Class 2	Class 3	
Cluster 1	2	3	85	90
Cluster 2	90	12	8	110
Cluster 3	8	85	7	100
	100	100	100	300

	Class 1	Class 2	Class 3	
Cluster 1	20	35	35	90
Cluster 2	30	42	38	110
Cluster 3	38	35	27	100
	100	100	100	300

Purity: (0.94, 0.81, 0.85) – overall 0.86
Precision: (0.94, 0.81, 0.85)
Recall: (0.85, 0.9, 0.85)

Purity: (0.38, 0.38, 0.38) – overall 0.38
Precision: (0.38, 0.38, 0.38)
Recall: (0.35, 0.42, 0.38)

Another bad clustering

	Class 1	Class 2	Class 3	
Cluster 1	0	0	35	35
Cluster 2	50	77	38	165
Cluster 3	38	35	27	100
	100	100	100	300

Cluster 1:

Purity: 1
Precision: 1
Recall: 0.35

External Measures of Cluster Validity: Entropy and Purity

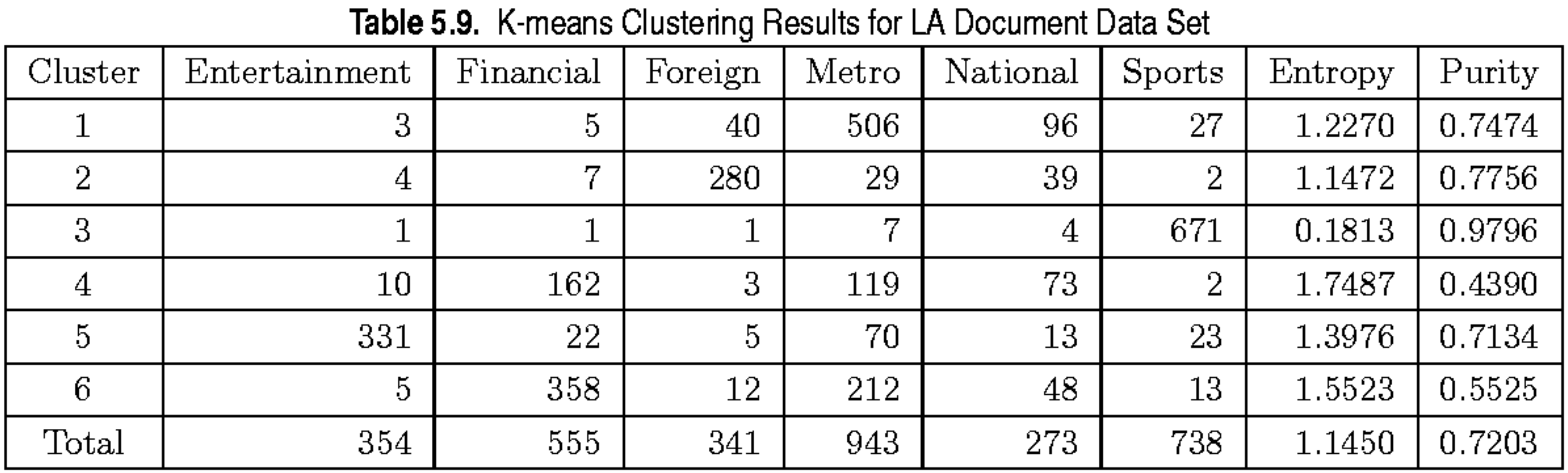
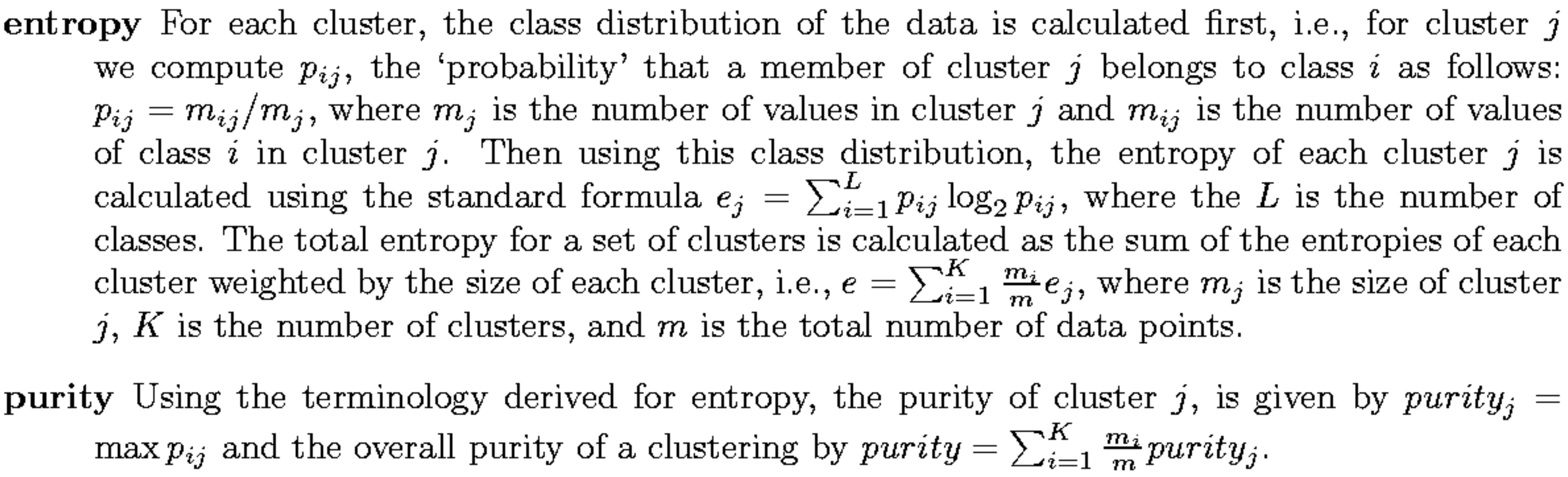


Table 5.9. K-means Clustering Results for LA Document Data Set

Cluster	Entertainment	Financial	Foreign	Metro	National	Sports	Entropy	Purity
1	3	5	40	506	96	27	1.2270	0.7474
2	4	7	280	29	39	2	1.1472	0.7756
3	1	1	1	7	4	671	0.1813	0.9796
4	10	162	3	119	73	2	1.7487	0.4390
5	331	22	5	70	13	23	1.3976	0.7134
6	5	358	12	212	48	13	1.5523	0.5525
Total	354	555	341	943	273	738	1.1450	0.7203



entropy For each cluster, the class distribution of the data is calculated first, i.e., for cluster j we compute p_{ij}, the 'probability' that a member of cluster j belongs to class i as follows: $p_{ij} = m_{ij}/m_j$, where m_j is the number of values in cluster j and m_{ij} is the number of values of class i in cluster j. Then using this class distribution, the entropy of each cluster j is calculated using the standard formula $e_j = \sum_{i=1}^{L} p_{ij} \log_2 p_{ij}$, where the L is the number of classes. The total entropy for a set of clusters is calculated as the sum of the entropies of each cluster weighted by the size of each cluster, i.e., $e = \sum_{i=1}^{K} \frac{m_i}{m} e_j$, where m_j is the size of cluster j, K is the number of clusters, and m is the total number of data points.

purity Using the terminology derived for entropy, the purity of cluster j, is given by $purity_j = \max p_{ij}$ and the overall purity of a clustering by $purity = \sum_{i=1}^{K} \frac{m_i}{m} purity_j$.

Final Comment on Cluster Validity

“The validation of clustering structures is the most difficult and frustrating part of cluster analysis.

Without a strong effort in this direction, cluster analysis will remain a black art accessible only to those true believers who have experience and great courage.”

Algorithms for Clustering Data, Jain and Dubes

MINIMUM DESCRIPTION LENGTH

Occam’s razor

- Most data mining tasks can be described as creating a model for the data
  - E.g., the EM algorithm models the data as a mixture of Gaussians, the K-means models the data as a set of centroids.
- What is the right model?
- Occam’s razor: All other things being equal, the simplest model is the best.
  - A good principle for life as well

Occam's Razor and MDL

- What is a simple model?

- Minimum Description Length Principle: Every model provides a (lossless) encoding of our data. The model that gives the shortest encoding (best compression) of the data is the best.
  - Related: Kolmogorov complexity. Find the shortest program that produces the data (uncomputable).
  - MDL restricts the family of models considered

  - Encoding cost: cost of party A to transmit to party B the data.

Minimum Description Length (MDL)

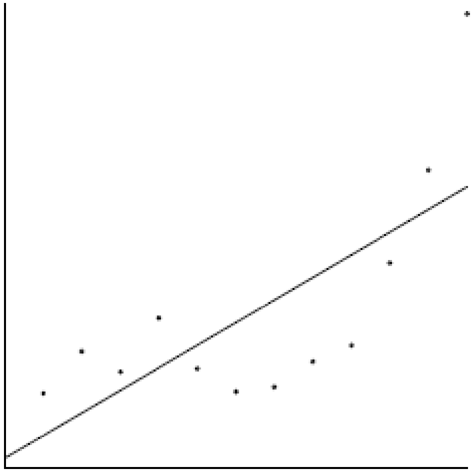
- The description length consists of two terms
  - The cost of describing the model (model cost)
  - The cost of describing the data given the model (data cost).
  - L(D) = L(M) + L(D|M)

- There is a tradeoff between the two costs
  - Very complex models describe the data in a lot of detail but are expensive to describe the model
  - Very simple models are cheap to describe but it is expensive to describe the data given the model

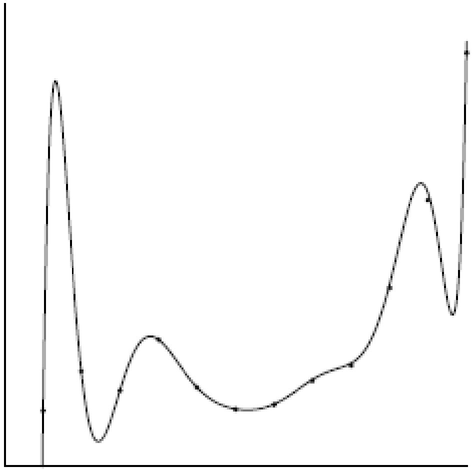
- This is generic idea for finding the right model
  - We use MDL as a blanket name.

Example

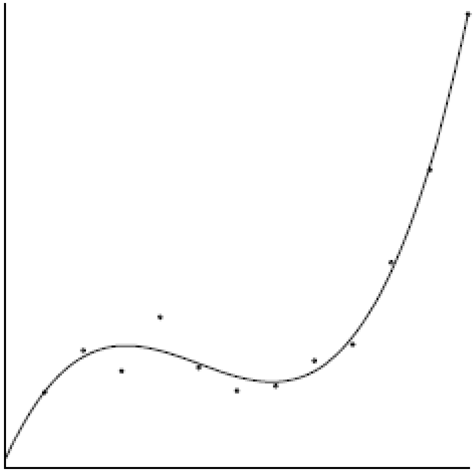
- Regression: find a polynomial for describing a set of values
  - Model complexity (model cost): polynomial coefficients
  - Goodness of fit (data cost): difference between real value and the polynomial value



Minimum model cost
High data cost



High model cost
Minimum data cost



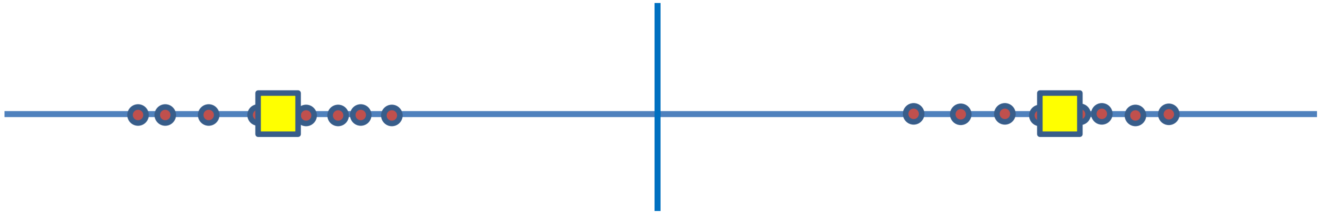
Low model cost
Low data cost

MDL avoids overfitting automatically!

Source: Grunwald et al. (2005) *Tutorial on MDL.*

Example

- Suppose you want to describe a set of integer numbers
  - Cost of describing a single number is proportional to the value of the number x (e.g., logx).
  - How can we get an efficient description?



- Cluster integers into two clusters and describe the cluster by the centroid and the points by their distance from the centroid
  - Model cost: cost of the centroids
  - Data cost: cost of cluster membership and distance from centroid

- What are the two extreme cases?

MDL and Data Mining

- Why does the shorter encoding make sense?
  - Shorter encoding implies regularities in the data
  - Regularities in the data imply patterns
  - Patterns are interesting

- Example

000010000100001000010000100001000010000100001000010000100001

- Short description length, just repeat 12 times 00001

0100111001010011011010100001110101111011011010101110010011100

- Random sequence, no patterns, no compression

Is everything about compression?

- Jürgen Schmidhuber: A theory about creativity, art and fun
    - Interesting Art corresponds to a novel pattern that we cannot compress well, yet it is not too random so we can learn it
    - Good Humor corresponds to an input that does not compress well because it is out of place and surprising
    - Scientific discovery corresponds to a significant compression event
        - E.g., a law that can explain all falling apples.

- Fun lecture:
    - Compression Progress: The Algorithmic Principle Behind Curiosity and Creativity

Issues with MDL

- What is the right model family?
  - This determines the kind of solutions that we can have
    - E.g., polynomials
    - Clusterings

- What is the encoding cost?
  - Determines the function that we optimize
  - Information theory

INFORMATION THEORY

A short introduction

Encoding

- Consider the following sequence

AAABBBAAACCCABACAABBAACCABAC

- Suppose you wanted to encode it in binary form, how would you do it?

50% A
25% B
25% C

A is 50% of the sequence
We should give it a shorter representation

A → 0
B → 10
C → 11

This is actually provably the best encoding!

Encoding

- Prefix Codes: no codeword is a prefix of another

A → 0
B → 10
C → 11

Uniquely directly decodable

For every code we can find a prefix code of equal length

- Codes and Distributions: There is one to one mapping between codes and distributions
  - If P is a distribution over a set of elements (e.g., {A,B,C}) then there exists a (prefix) code C where $L_C(x) = -\lceil \log P(x) \rceil, x \in \{A, B, C\}$
  - For every (prefix) code C of elements {A,B,C}, we can define a distribution $P(x) = 2^{-C(x)}$

- The code defined has the smallest average codelength!

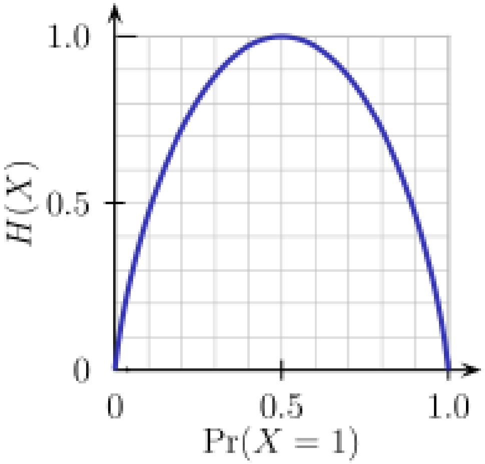
Entropy

- Suppose we have a random variable X that takes n distinct values
$$X = \{x_1, x_2, \dots, x_n\}$$
that have probabilities $\mathrm{P(X)} = \{p_1, \dots, p_n\}$

- This defines a code C with $L_C(x_i) = -\lceil \log p_i \rceil$. The average codelength is
$$-\sum_{i=1}^{n} p_i \lceil \log p_i \rceil$$
- This (more or less) is the entropy $H(X)$ of the random variable X
$$H(X) = -\sum_{i=1}^{n} p_i \log p_i$$
- Shannon's theorem: The entropy is a lower bound on the average codelength of any code that encodes the distribution P(X)
  - When encoding N numbers drawn from P(X), the best encoding length we can hope for is $N * H(X)$
  - Reminder: Lossless encoding

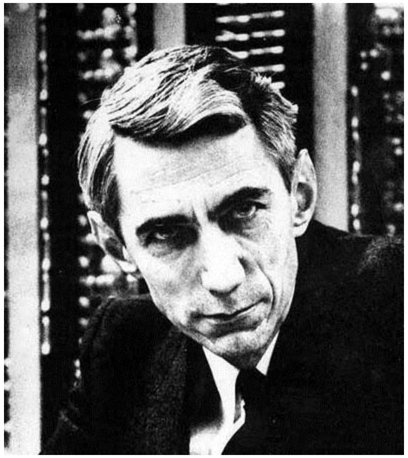
Entropy

$$H(X) = -\sum_{i=1}^{n} p_i \log p_i$$



- What does it mean?
- Entropy captures different aspects of a distribution:
  - The compressibility of the data represented by random variable X
    - Follows from Shannon's theorem
  - The uncertainty of the distribution (highest entropy for uniform distribution)
    - How well can I predict a value of the random variable?
  - The information content of the random variable X
    - The number of bits used for representing a value is the information content of this value.

Claude Shannon



Father of Information Theory

Envisioned the idea of communication of information with 0/1 bits

Introduced the word “bit”

The word entropy was suggested by Von Neumann

- Similarity to physics, but also
- “nobody really knows what entropy really is, so in any conversation you will have an advantage”

Some information theoretic measures

- Conditional entropy H(Y|X): the uncertainty for Y given that we know X

$$H(Y|X) = -\sum_{x} p(x) \sum_{y} p(y|x) \log p(y|x)$$

$$= -\sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)}$$

- Mutual Information I(X,Y): The reduction in the uncertainty for Y (or X) given that we know X (or Y)

$$I(X,Y) = H(Y) - H(Y|X) = H(X) - H(X|Y)$$

Some information theoretic measures

- Cross Entropy: The cost of encoding distribution P, using the code of distribution Q

$$-\sum_{x} P(x) \log Q(x)$$

- KL Divergence KL(P||Q): The increase in encoding cost for distribution P when using the code of distribution Q

$$KL(P||Q) = -\sum_{x} P(x) \log Q(x) + \sum_{x} P(x) \log P(x)$$

  - Not symmetric
  - Problematic if Q not defined for all x of P.

Some information theoretic measures

- Jensen-Shannon Divergence JS(P,Q): distance between two distributions P and Q
  - Deals with the shortcomings of KL-divergence

- If M = ½ (P+Q) is the mean distribution

$$JS(P,Q) = \frac{1}{2}KL(P||M) + \frac{1}{2}KL(Q||M)$$

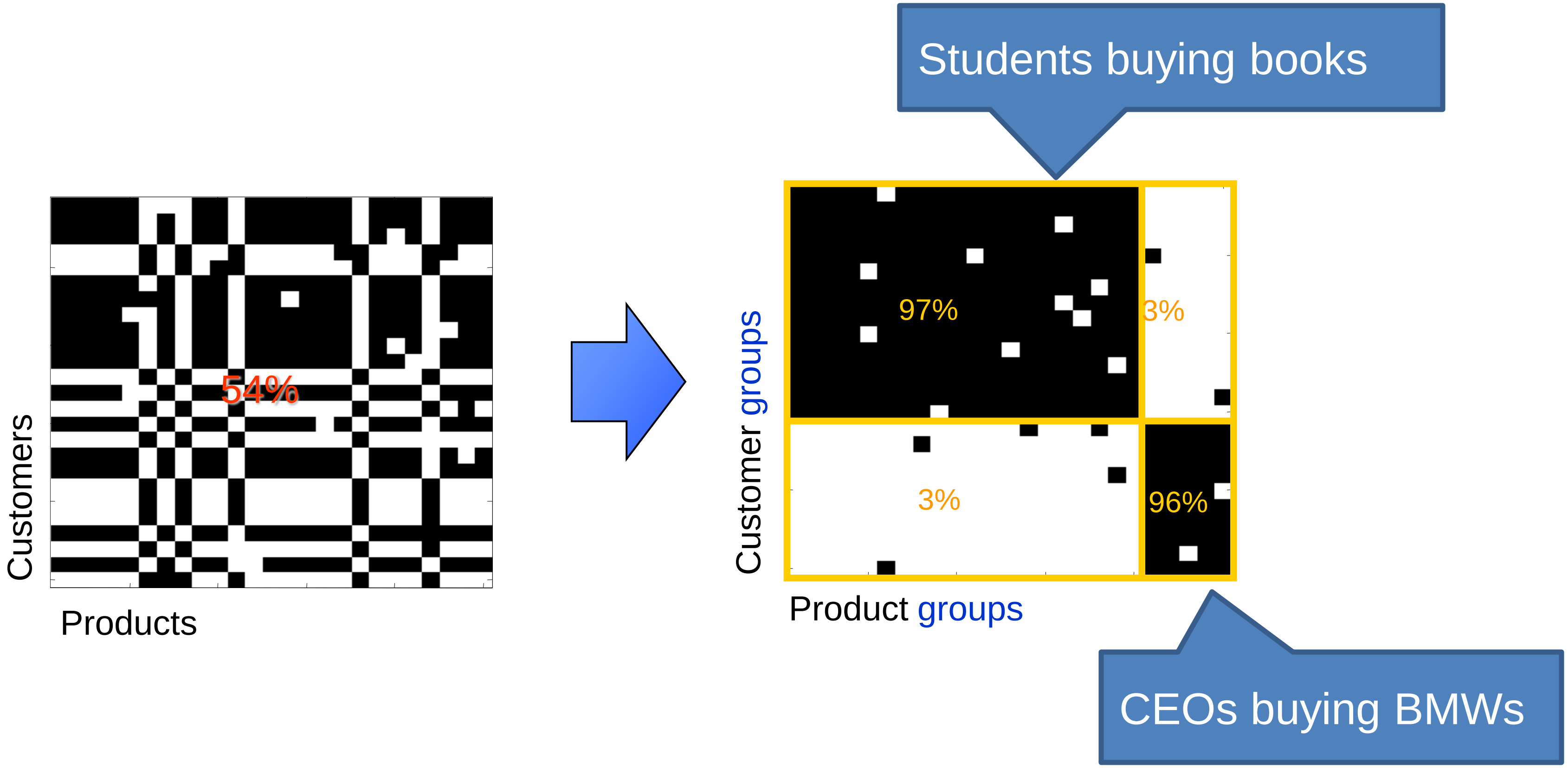
- Jensen-Shannon is a metric

USING MDL FOR CO-CLUSTERING (CROSS-ASSOCIATIONS)

Thanks to Spiros Papadimitriou.

Co-clustering

- Simultaneous grouping of rows and columns of a matrix into homogeneous groups



Students buying books

CEOs buying BMWs

Co-clustering

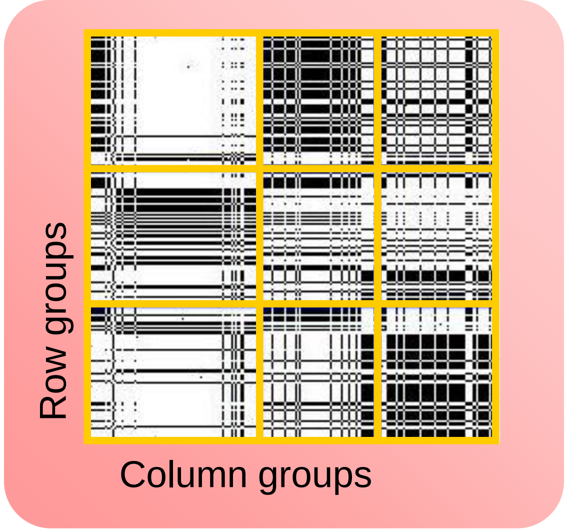
- **Step 1**: How to define a “good” partitioning?

  Intuition and formalization

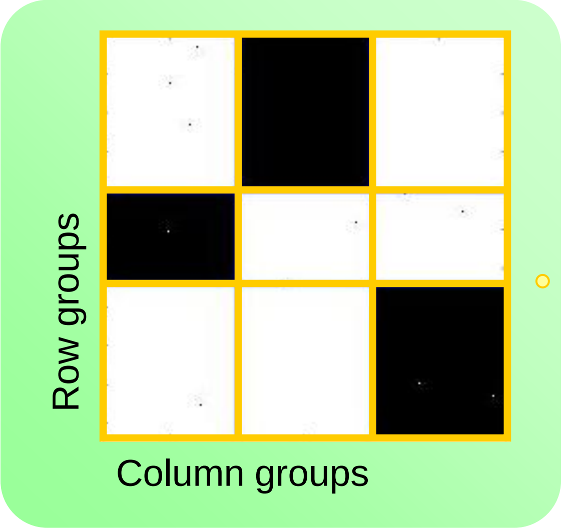
- **Step 2**: How to find it?

Co-clustering

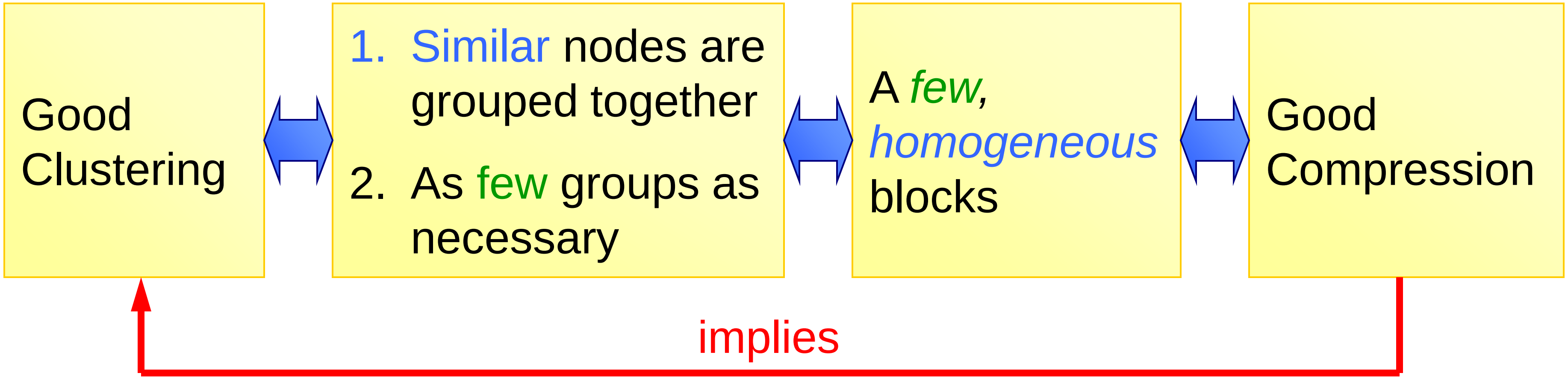
Intuition



versus



Why is this better?



Good Clustering

1. Similar nodes are grouped together
2. As few groups as necessary

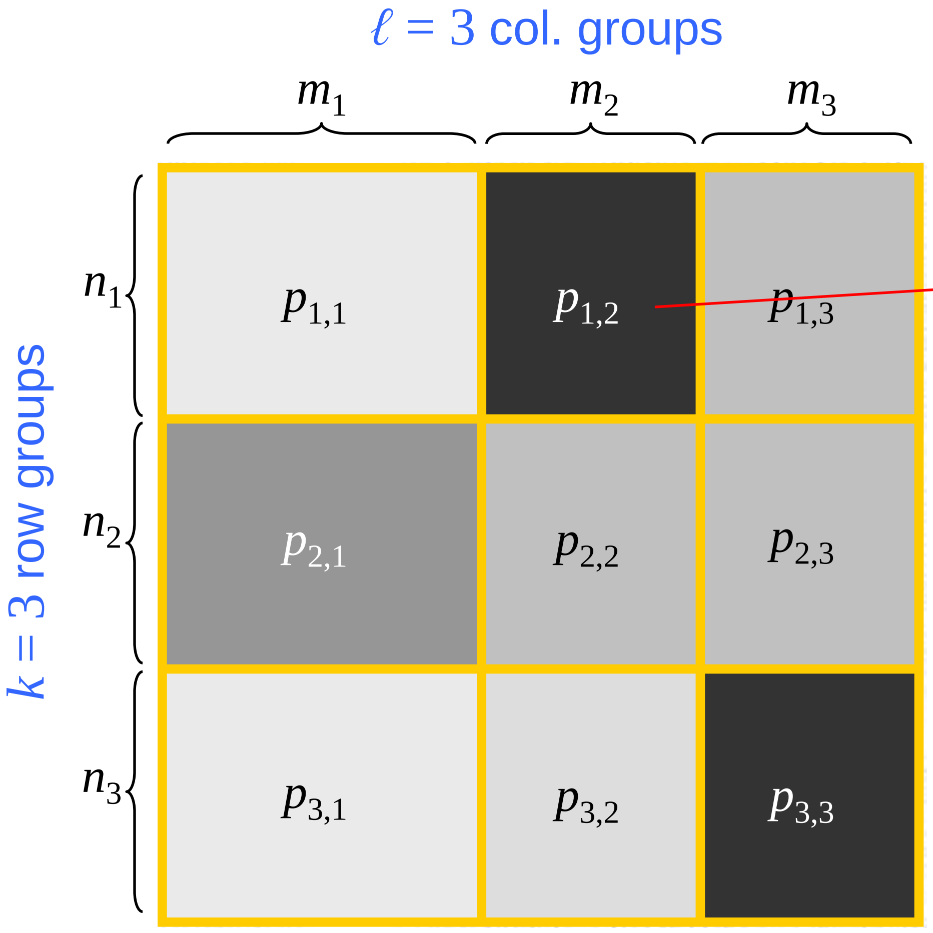
A *few*, *homogeneous* blocks

Good Compression

implies

Co-clustering

MDL formalization—Cost objective



n × m matrix

$$p_{i,j} := \frac{e_{i,j}}{n_i m_j}$$

density of ones

$n_1 m_2 \, H(p_{1,2})$ bits for (1,2)

block size

entropy

$$\underbrace{\sum_{i,j} n_i m_j \, H(p_{i,j})}_{\text{data cost}}$$ bits total

+

$$\overbrace{nH\left(\frac{n_1}{n}, \ldots, \frac{n_k}{n}\right) + mH\left(\frac{m_1}{m}, \ldots, \frac{m_\ell}{m}\right)}^{\text{model cost}}$$

row-partition description

col-partition description

$$+ \log^* k + \log^* \ell + \sum_{i,j} \lceil \log n_i m_j \rceil$$

transmit #partitions

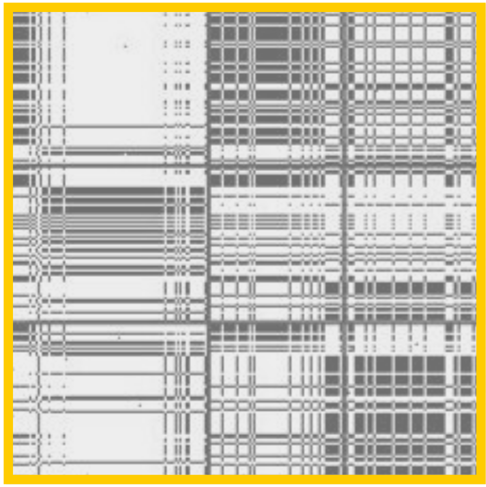
transmit #ones $e_{i,j}$

Co-clustering

MDL formalization—Cost objective

one row group
one col group

n row groups
m col groups



high

low

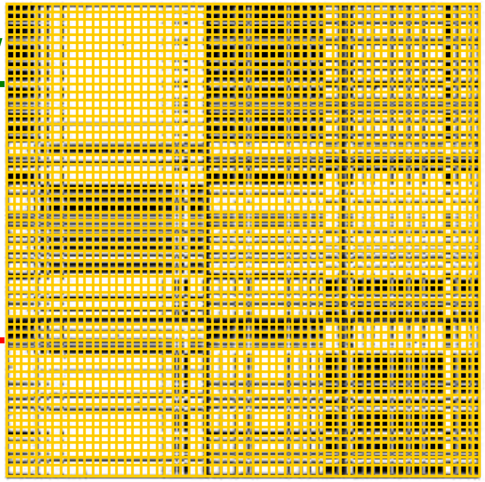
code cost
(block contents)

+

description cost
(block structure)

low

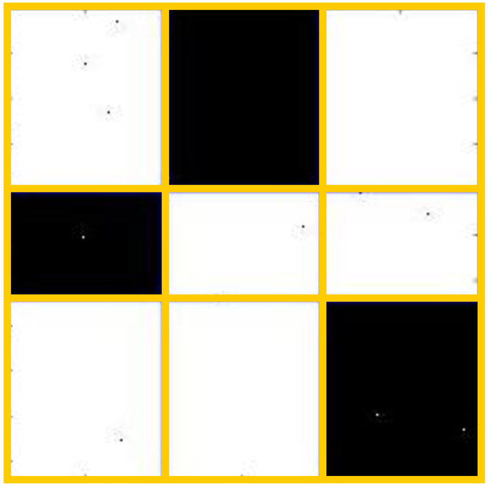
high



Co-clustering

MDL formalization—Cost objective

$k = 3$ row groups ✓
$\ell = 3$ col groups



low →

code cost
(block contents)

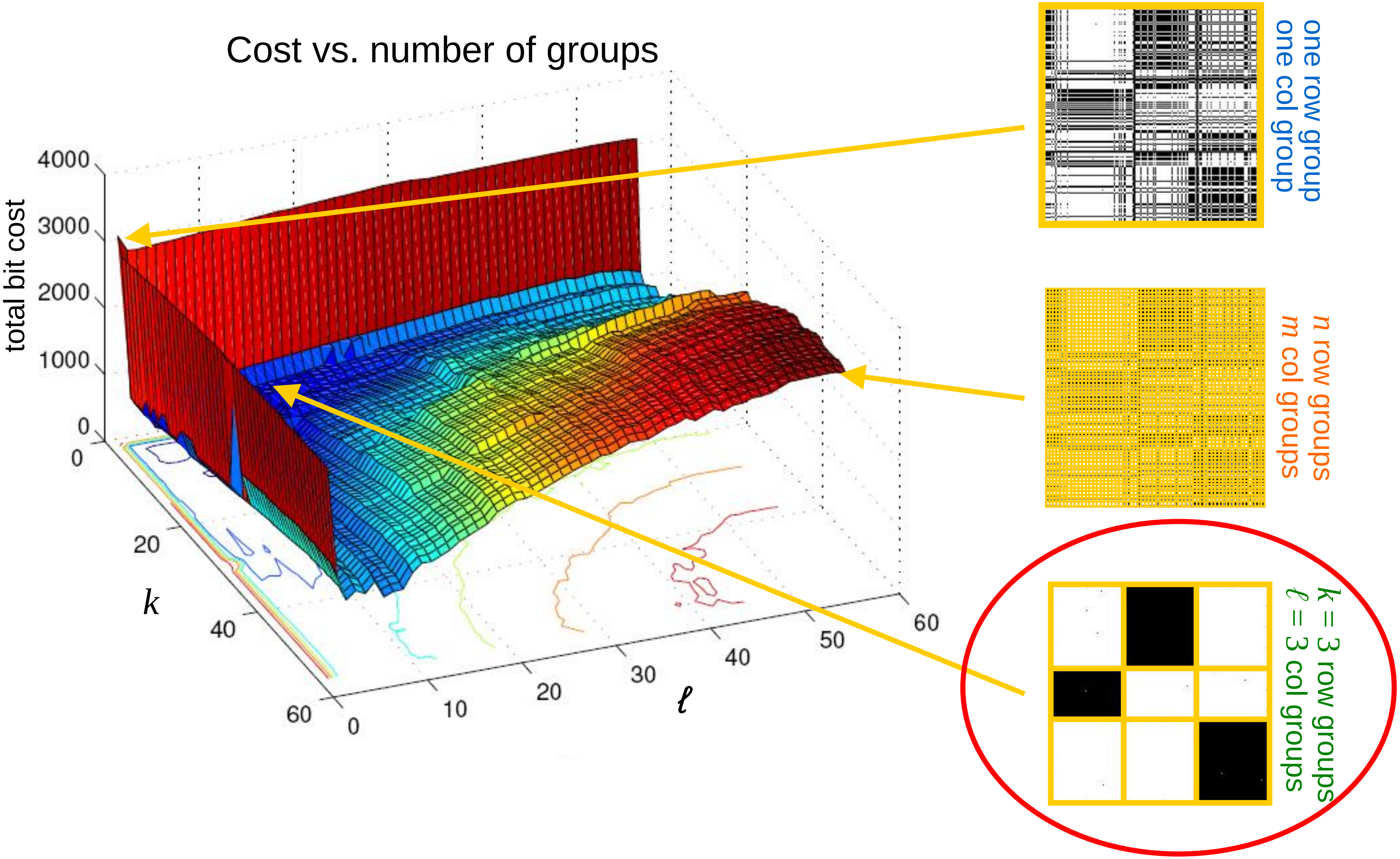
+

low →

description cost
(block structure)

Co-clustering

MDL formalization—Cost objective



Cost vs. number of groups

one row group
one col group

n row groups
m col groups

k = 3 row groups
ℓ = 3 col groups

Co-clustering

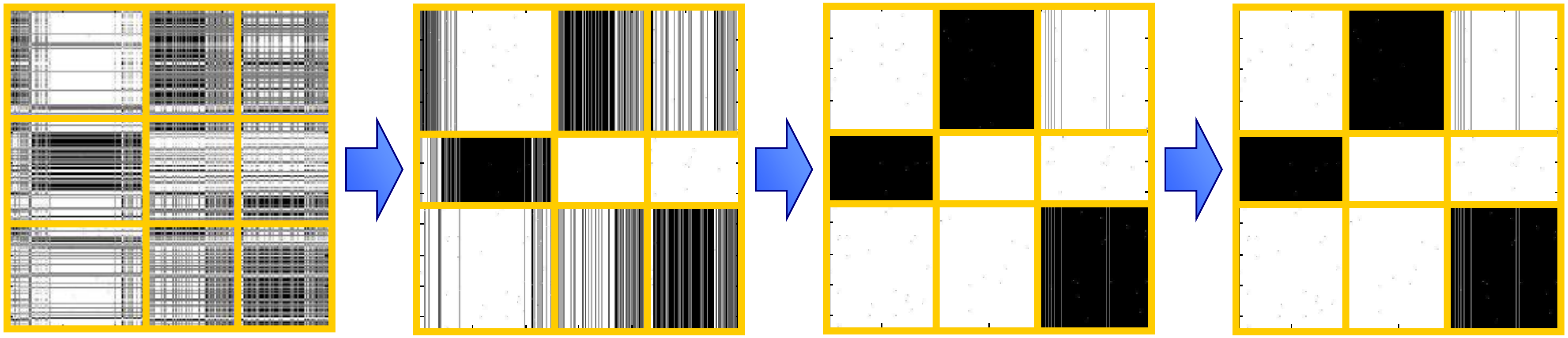
- **Step 1**: How to define a “good” partitioning?

  Intuition and formalization

- **Step 2**: How to find it?

Search for solution

Overview: assignments w/ fixed number of groups (shuffles)



original groups

row shuffle

reassign all rows, holding column assignments fixed

column shuffle

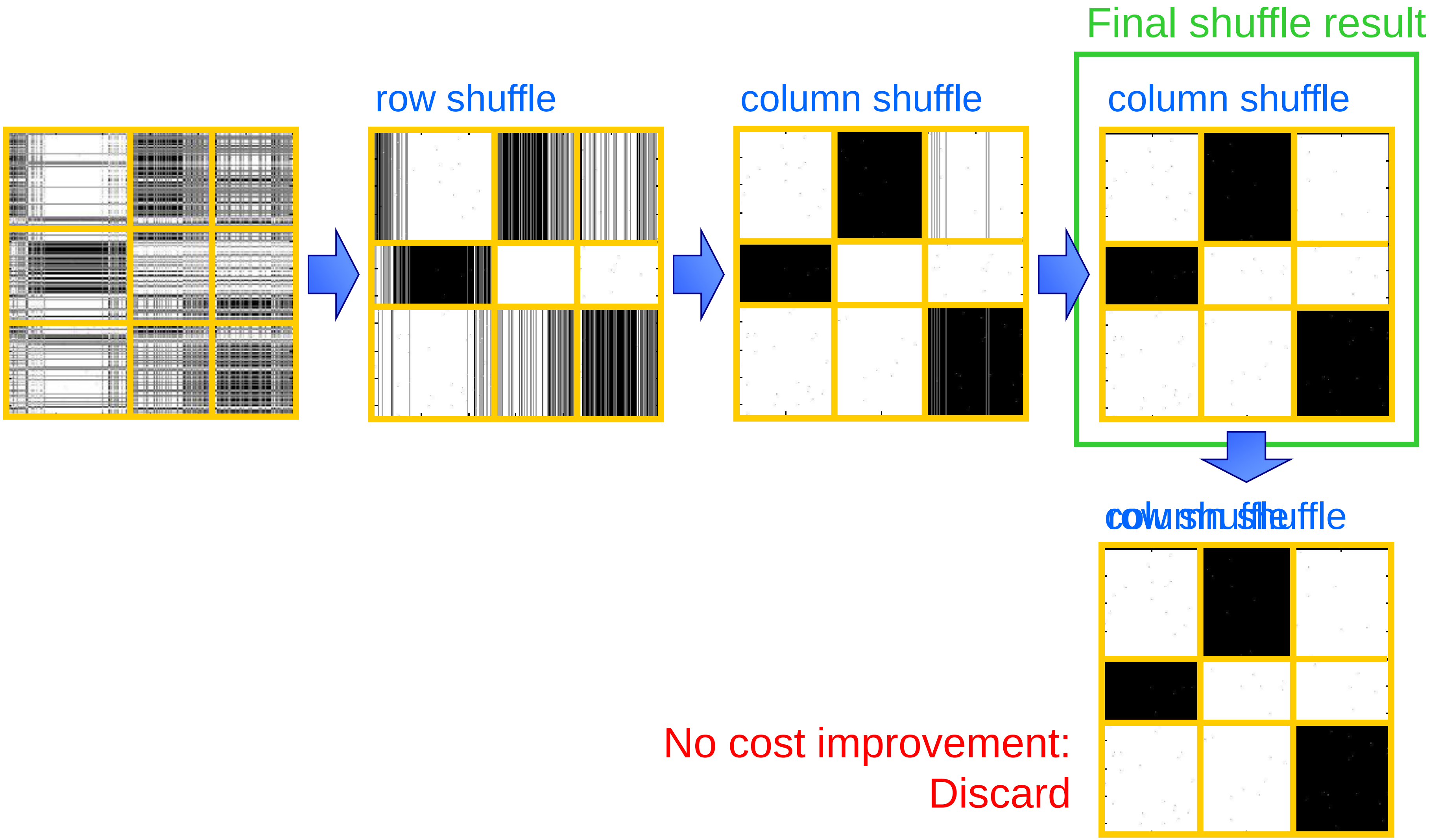
reassign all columns, holding row assignments fixed

row shuffle

No cost improvement: Discard

Search for solution

Overview: assignments w/ fixed number of groups (shuffles)



Final shuffle result

row shuffle

column shuffle

column shuffle

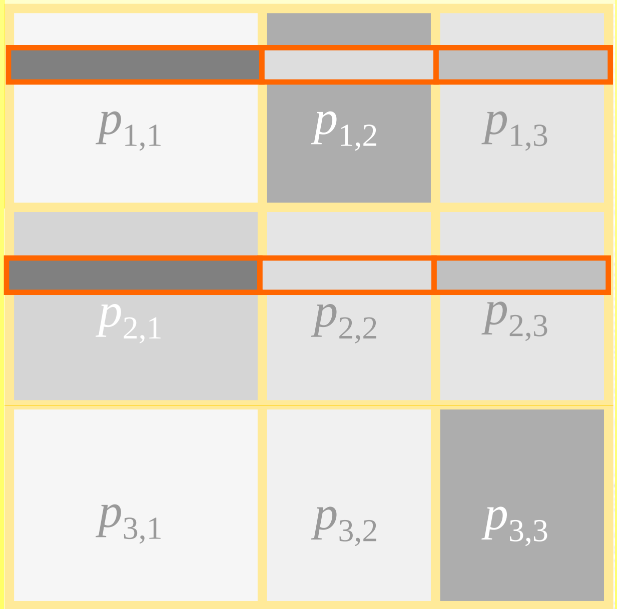
row shuffle

column shuffle

No cost improvement:
Discard

Search for solution

Shuffles



Similarity (“KL-divergences”)
of row fragments
to blocks of a row group

Assign to second row-group

teration

ach part

that, for all

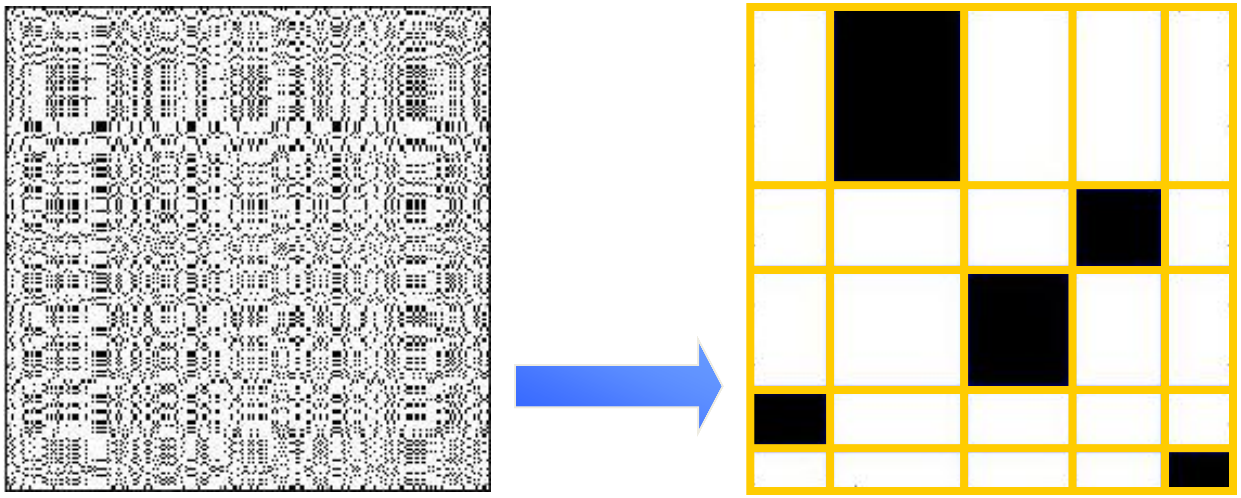
$$-\sum_{j=1}^{\ell}\Big(\nu_j \log p_{i^*,j} + (n-\nu_j)\log(1-p_{i^*,j})\Big)$$

$$\leq -\sum_{j=1}^{\ell}\Big(\nu_j \log p_{i,j} + (n-\nu_j)\log(1-p_{i,j})\Big)$$

Search for solution

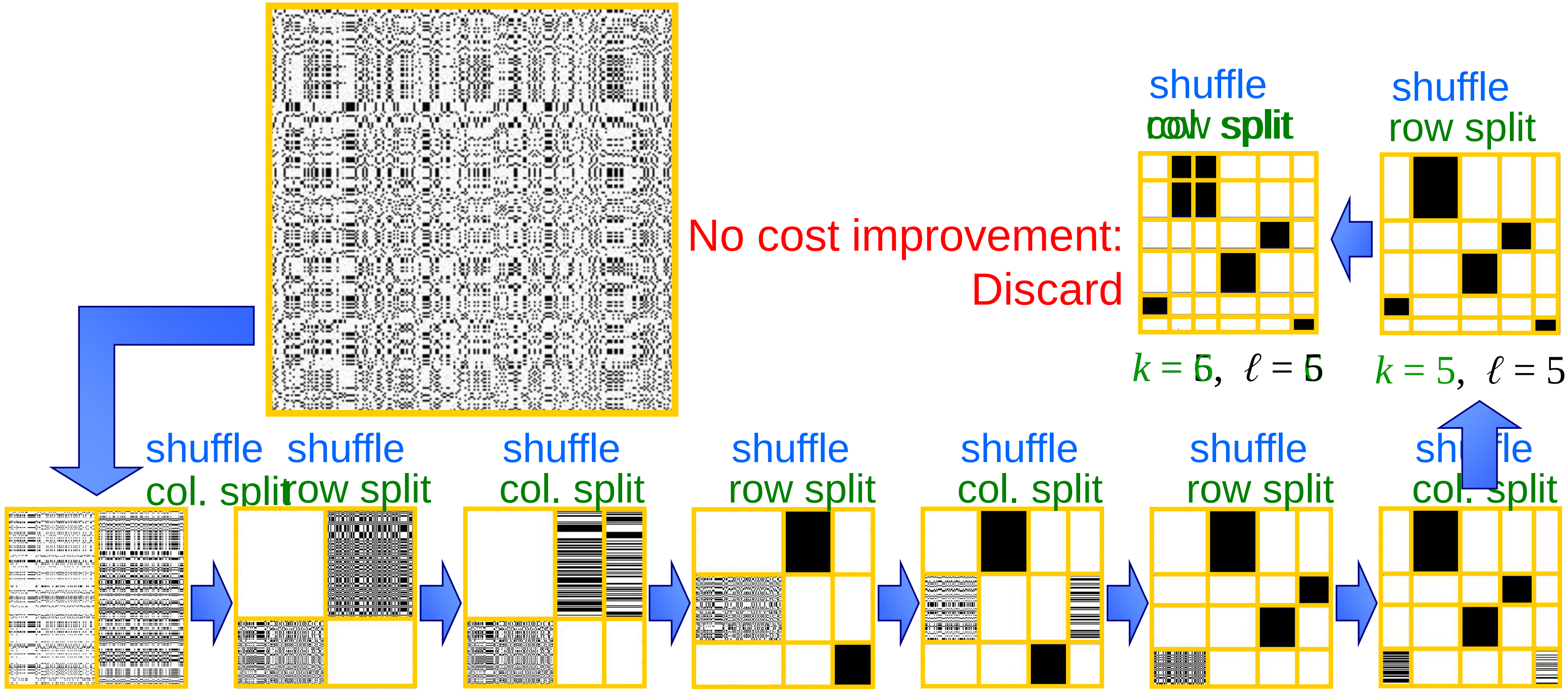
Overview: number of groups k and ℓ (splits & shuffles)

$k = 5, \;\; \ell = 5$



Search for solution

Overview: number of groups k and ℓ (splits & shuffles)



$k = 1, \; \ell = 1$

No cost improvement: Discard

shuffle
col split

$k = 6, \; \ell = 5$

shuffle
row split

$k = 5, \; \ell = 5$

shuffle
col. split

k=1, ℓ=2

shuffle
row split

k=2, ℓ=2

shuffle
col. split

k=2, ℓ=3

shuffle
row split

k=3, ℓ=3

shuffle
col. split

k=3, ℓ=4

shuffle
row split

k=4, ℓ=4

shuffle
col. split

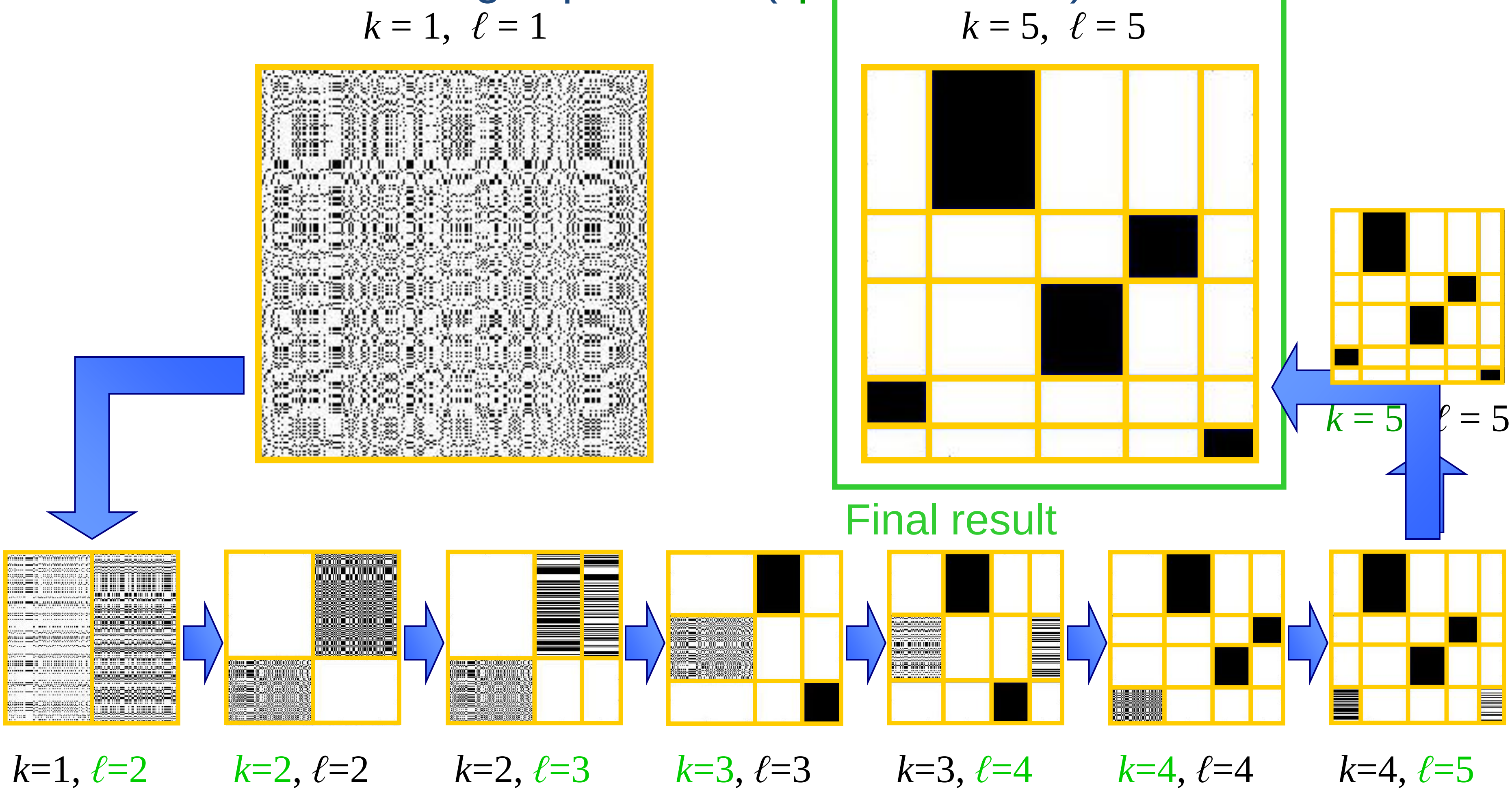
k=4, ℓ=5

Split:
Increase k or ℓ

Shuffle:
Rearrange rows or cols

Search for solution

Overview: number of groups k and ℓ (splits & shuffles)



$k = 1,\ \ell = 1$

$k = 5,\ \ell = 5$

Final result

$k = 5,\ \ell = 5$

k=1, ℓ=2

k=2, ℓ=2

k=2, ℓ=3

k=3, ℓ=3

k=3, ℓ=4

k=4, ℓ=4

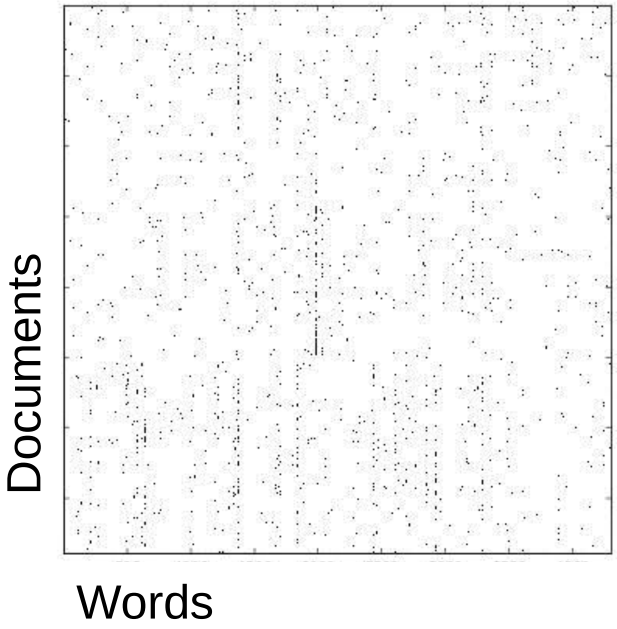
k=4, ℓ=5

Split:
Increase k or ℓ

Shuffle:
Rearrange rows or cols

Co-clustering

CLASSIC



CLASSIC corpus

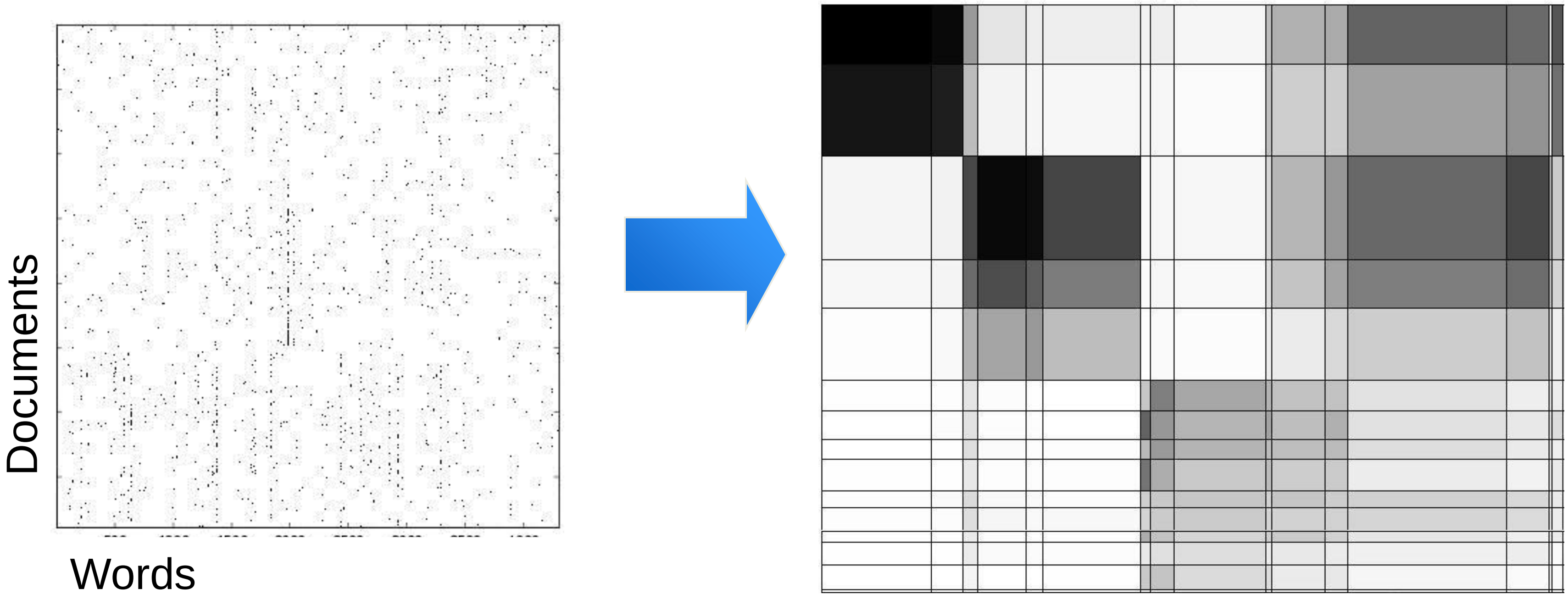
- 3,893 documents
- 4,303 words
- 176,347 “dots” (edges)

Combination of 3 sources:

- MEDLINE (medical)
- CISI (info. retrieval)
- CRANFIELD (aerodynamics)

Graph co-clustering

CLASSIC

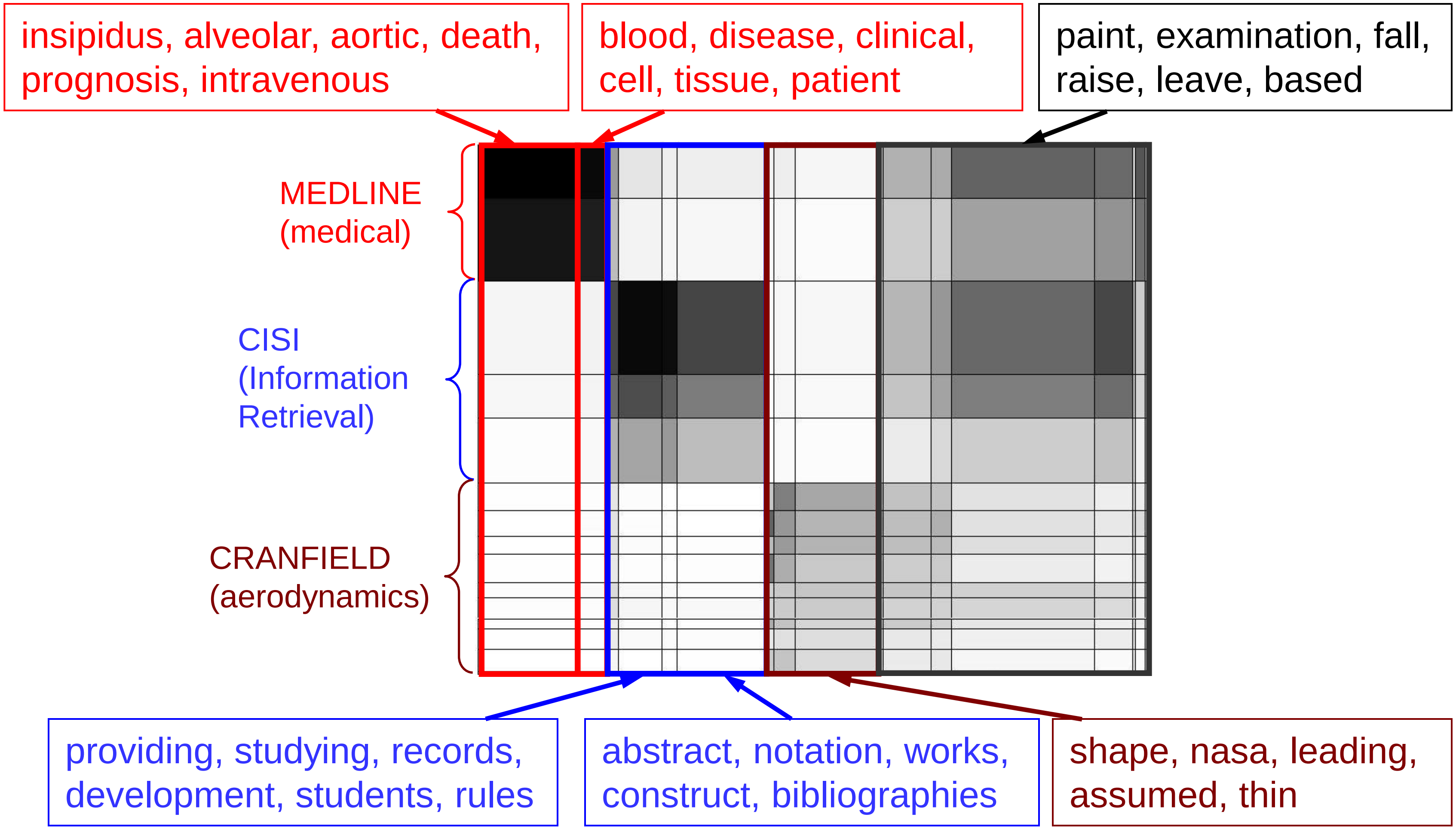


“CLASSIC” graph of documents & words:

$k = 15, \ell = 19$

Co-clustering

CLASSIC



insipidus, alveolar, aortic, death, prognosis, intravenous

blood, disease, clinical, cell, tissue, patient

paint, examination, fall, raise, leave, based

MEDLINE (medical)

CISI (Information Retrieval)

CRANFIELD (aerodynamics)

providing, studying, records, development, students, rules

abstract, notation, works, construct, bibliographies

shape, nasa, leading, assumed, thin

“CLASSIC” graph of documents & words:
$k = 15$, $\ell = 19$

Co-clustering

CLASSIC

Document cluster #	Document class			Precision
	CRANFIELD	CISI	MEDLINE	
1	0	1	390	0.997
2	0	0	610	1.000
3	2	676	9	0.984
4	1	317	6	0.978
5	3	452	16	0.960
6	207	0	0	1.000
7	188	0	0	1.000
8	131	0	0	1.000
9	209	0	0	1.000
10	107	2	0	0.982
11	152	3	2	0.968
12	74	0	0	1.000
13	139	9	0	0.939
14	163	0	0	1.000
15	24	0	0	1.000
Recall	0.996	0.990	0.968	

0.999

0.975

0.987

0.94-1.00

0.97-0.99